Atmospheric Emissions Associated with the Use of Biogas in Ontario

by

Savannah Bindas

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Civil Engineering

Waterloo, Ontario, Canada, 2024

© Savannah Bindas 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This study aims to quantify the atmospheric emissions associated with an energy-from-waste transition in Ontario. Specifically, it explores the emissions from using livestock and food waste to produce biogas as a source of renewable natural gas. Biogas is one potential "closed loop" solution to waste management; however, there is potential for additional emissions associated with transitioning to anaerobic digestion as a waste management strategy. Each step along the anaerobic digestion process, from the transportation of feedstock to the storage of post-processed digestate, can release both greenhouse gas (GHG) and air pollutant emissions. Here, we quantified the net effects on emissions of a biogas transition in Ontario. We evaluated scenarios using up to 100% of available waste feedstocks in the province and compared these emissions to conventional manure management and landfilling. We found that emissions from current manure management strategies dominated GHG emissions, and, as more manure was utilized in the biogas system, there was a drastic decrease in corresponding emissions of CH_4, N_2O, and CO_2, all of which are potent GHGs. All scenarios showed emission reductions compared to the traditional practice. By the 75% biogas scenario, GHG emissions associated with the biogas process are balanced by the potential offsets from avoided synesthetic fertilizer production, leading to negligible net GHG emissions from this system. In the 100% scenario, we observed SOx, VOCs, NH_3, and $PM_{2.5}$ were increasingly offset by emissions savings in the natural gas production and synthetic fertilizer production. The important exceptions were the significant NH_3 and $PM_{2.5}$ emissions associated with conventional manure management. Due to this, we did not see net emissions savings from the biogas scenarios until the 100% run. These results highlight the atmospheric impacts of conventional waste management and demonstrate the potential for anaerobic digestion to mitigating these emissions.

Acknowledgements

I would like to express gratitude to my supervisor Dr. Rebecca Saari, who supported me throughout the process of completing this research. She was instrumental in guiding me through these past two years, and I thank her for her patience, kindness, and for inspiring me to continue to work towards my goals.

I also want to acknowledge and thank the readers of my thesis, Dr. Nandita Basu and Dr. Bruce Macvicar, who took time to review this work and provide valuable feedback.

There are many people who should be thanked for helping me with this work, beginning with Ushnik Mukherjee. Ushnik's work served as the basis for my research, and without the work he did to develop his optimization model, this work would not have been possible. Additionally, he continued to advise and support me throughout the research process. I am also thankful for the input of my lab members, Amir Reza Salehi and Matt Sparks, for the support they provided in all aspects of this work, from providing feedback on the very beginning stages of my research to helping me revise the final draft of this thesis.

I want to thank my beautiful partner, Gabriel Gallan, who held my hand as I worked long hours, who continually reminded me to find joy outside of my work, and whose unwavering support gave me the confidence to get to this moment. Gabe, thank you for everything you do.

I also want to thank my Mom, for her long phone conversations and never-ending love, and my Dad, for his confidence in my abilities and assurance that "all grad students feel like this".

My friends, both in Canada and abroad, deserve my heartfelt thanks as well. Thank you for always listening to me talk about my work, even though sometimes I know it bores you to tears. I am forever grateful to have you in my life.

Finally, I would like to acknowledge the entire SolutionScapes team for their contributions to this work.

Table of Contents

Author's Declaration ..... *ii*

Abstract ..... *iii*

Acknowledgements ..... *iv*

List of Figures ..... *vii*

List of Tables ..... *viii*

Chapter 1 ..... *1*

1.1 Atmospheric Emissions **1**

1.2 Environmental Impact of Current Waste Management Practices **3**

1.3 Potential Solution: Biogas **5**

1.4 Potential Emissions Issues **7**

1.5 Objective **10**

Chapter 2 ..... *11*

2.1 Nutrient Recycling Potential **11**

2.2 Feasibility **13**

2.3 Energy Potential and Utilization Pathways **15**

2.4 Novelty of Research **17**

Chapter 3 ..... *19*

3.1 GHG Emissions Estimation **22**

3.1.1 Baseline Emissions 22

3.1.2 Biogas Scenarios 28

3.2 Air Pollutant Emissions Estimation **37**

3.2.1 Baseline Emissions 37
3.2.2 Biogas Scenario 41

Chapter 4 46

4.1 GHG Emissions 46

4.2 Air Pollutant Emissions 49

Chapter 5 58

References 60

Appendix A 65

Appendix B 71

Appendix C 72

List of Figures

Figure 1. Process diagram distinguishing the difference in pathways between the baseline and biogas scenarios (created by Nettie Wallace, Ushnik Mukherjee, Savannah Bindas, and Camila Granito) 20

Figure 2. Process diagram describing the sources of emissions (red boxes) and offsets (green boxes) associated with the biogas scenarios (created by Nettie Wallace, Ushnik Mukherjee, Savannah Bindas, and Camila Granito). 21

Figure 3. GHG emissions (in Mtonnes of CO_{2E}) across the baseline and 25%, 50%, 75%, and 100% biogas scenarios. Emissions are shown by source (manure management, food waste, biogas upgrading, transportation, manure storage) and offsets (due to avoided production of natural gas and fertilizer). Red markers and lines represent the net GHG emissions, taking into account both potential emissions and potential offsets from natural gas and fertilizer production 47

Figure 4. Barchart showing the emissions from manure management and food waste landfilling in the baseline scenario. Emissions are reported in tonnes. 50

Figure 5. Barchart representing air pollutant emissions associated with a 25% biogas scenario. Emissions from transportation, upgrading, digestate storage, food waste landfilling, natural gas production offset, and synthetic fertilizer production offset are reported in tonnes, whereas emissions from conventional manure management are reported in kilotonnes 51

Figure 6. Barchart representing air pollutant emissions associated with a 50% biogas scenario. Emissions from transportation, upgrading, digestate storage, food waste landfilling, natural gas production offset, and synthetic fertilizer production offset are reported in tonnes, whereas emissions from conventional manure management are reported in kilotonnes 53

Figure 7. Barchart representing air pollutant emissions associated with a 75% biogas scenario. Emissions from transportation, upgrading, digestate storage, food waste landfilling, natural gas production offset, and synthetic fertilizer production offset are reported in tonnes, whereas emissions from conventional manure management are reported in kilotonnes 54

Figure 8. Barchart representing air pollutant emissions associated with a 100% biogas scenario. Emissions from transportation, upgrading, digestate storage, food waste landfilling, natural gas production offset, and synthetic fertilizer production offset are reported in tonnes, whereas emissions from conventional manure management are reported in kilotonnes 56

Figure 9. Barchart representing net emissions by scenario in ktonnes 57

List of Tables

Table 1. Atmospheric emissions examined and their impacts to health and climate; adapted from (Shindell 2015) and (Nasir 2018)..... 2

Table 2. Manure Managed in Conventional Systems in Ontario..... 23

Table 3. Share of each manure type managed by differing management systems in Canada (% f values (f*100)) (Government of Canada 2022a)..... 24

Table 4. Livestock manure volatile solid (VS) and maximum methane generation potential (B0) for each livestock type..... 24

Table 5. Methane conversion factors (MCF) for different manure management strategies in Canada (Government of Canada 2022a)..... 25

Table 6. N_2O emissions factors for manure management systems by livestock type (Government of Canada 2022a)26

Table 7. Parameter values for estimating methane generation at landfills (Environment and Climate Change Canada 2023)..... 27

Table 8. Parameter values for estimating methane emitted at landfills..... 28

Table 9. Food waste sent to landfill by scenario..... 28

Table 10. Feedstock transportation emissions factors for GHG and air pollutants associated with manure hauling vehicles assuming speeds between 44.25 kph and 52.30 kph on rural unrestricted roads during average spring temperatures (Wang 2020)..... 30

Table 11. Total distance travelled to move feedstock and total feedstock moved by biogas scenario and feedstock type..... 30

Table 12. Upgrade to pipeline injection on-site criteria pollutant and GHG emissions factors (EPA 2016)..... 31

Table 13. Amount of natural gas produced by scenario..... 32

Table 14. Well-to-Gate GHG emissions for conventional natural gas sources in Canada (Liu et al. 2021)..... 33

Table 15. N and P mass ratios for each manure type (United States Department of Agriculture 2009)..... 34

Table 16. Fertilizer types and N and P mass ratios for each fertilizer type. Those denoted with N/A have no ratios associated with that nutrient type. 35

Table 17. Percentage of each type of fertilizer used in Ontario. Diammonium phosphate (DAP) and monoammonium phosphate (MAP) percentages are determined using only the amounts of DAP and MAP used in Ontario, as they are the only fertilizers included that have phosphorus content. Urea, ammonia, and Urea Ammonium Nitrate (UAN) are similarly determined together, because they only include nitrogen content (Statistics Canada 2016)..... 35

Table 18. Fertilizer type and emissions factor EF (Bentrup 2018) 35
Table 19. Emissions factors (NH_3, VOC, $PM_{2.5}$) for differing animal types and manure management systems, shown in grams per animal unit (AU) 39
Table 20. Annual average manure generated by each animal type (United States Department of Agriculture 2009)40
Table 21. Number of animals per EPA animal unit (AU) (US EPA and National Academies of Sciences 2003) 40
Table 22. Percentage of manure (F_{MMS}) managed in differing manure management systems, separated by animal type, slightly altered to reflect available emission factors (Government of Canada 2022) 40
Table 23. Emissions factors (CO, NOx, $PM_{2.5}$) for food waste landfilling ("WebFire Search | US EPA," n.d.) 41
Table 24. Emissions factor (SO_2) for natural gas production offset (US EPA 1995b) 43
Table 25. Emissions factors for synthetic fertilizer production air pollutant emissions ("WebFire Search | US EPA," n.d.) 44
Table 26. Emissions factors for digestate storage post-digestion (EEA 2019) 45
Table 27. Net air pollutant emissions associated with the 25%, 50%, 75% and 100% scenarios. 52
Table 28. Feedstock transportation emissions factors for GHG and air pollutants associated with food waste hauling vehicles assuming speeds between 84.50 kph and 92.52 kph on rural unrestricted roads during average spring temperatures (Wang 2020) 67

Chapter 1

Introduction

Air pollution and climate change are issues that impact communities on all scales; local, national, and global. As such, they are the topic of much discussion and evaluation. Estimating emissions associated with policy changes or engineering choices is one way to evaluate the impact these actions have on the global climate and human health. This study evaluates the emissions, both of GHGs and air pollutants, associated with the potential adoption of anaerobic digestion across Ontario. The results of this study could suggest whether the use of biogas could be an opportunity for emissions mitigation, as well as evaluate the impact that current waste management strategies have on atmospheric emissions.

1.1 Atmospheric Emissions

Atmospheric releases refer to the release of chemical species into the atmosphere, specifically those releases potentially impacting global climate and human and environmental health. Species of atmospheric releases are labelled as greenhouse gases (GHGs), or air pollutants based on their ability to harm human health and environmental quality, or their ability to trap heat in the atmosphere, respectively. However, that does not mean that these categories are mutually exclusive. In fact, there are several air pollutants that have impacts on global mean temperature, as well as GHGs that negatively impact human and environmental health. Table 1 lists all GHGs, and air pollutants analyzed in this study and illustrates their impact on both global mean temperature and health. The GHGs analyzed in this study include CO_2, CH_4, and N_2O, all of which have significant potential to trap heat in the atmosphere by absorbing infrared radiation, leading to an increase in the effects of climate change. These three GHGs were chosen for analysis both because of their prevalence in the waste management sector and because they are identified by the IPCC as three of the most significant GHGs and contributors to climate change (IPCC 2022). These pollutants were converted into CO2e to compare their impact directly. This was done by utilizing their global warming potentials (GWP) as defined by the IPCC (IPCC 2022). The air pollutants in this study include nitrogen oxides (NOx), sulfur oxides (SOx), ammonia (NH_3), fine particulate matter ($PM_{2.5}$), volatile organic compounds (VOCs), and carbon monoxide (CO). $PM_{2.5}$ is further defined as solid or liquid particles with aerodynamic diameters 2.5 micrometers or smaller. $PM_{2.5}$ is a pollutant of major concern because of

its connection to mortality and morbidity, with an estimate 1 million premature deaths each year associated with short-term exposure to $PM_{2.5}$ between 2000 and 2019 (Yu et al. 2024). Other pollutants are included because they are typical emissions of concern and include the main precursors to the formation of secondary $PM_{2.5}$ (Jacob 1999).

Table 1. Atmospheric emissions examined and their impacts to health and climate; adapted from (Shindell 2015) and (Nasir 2018).

	Effect on Global Mean Temperature	Pathway to $PM_{2.5}$ health impacts
CO_2	Warming	
CH_4	Warming	
N_2O	Warming	
NOx	Warming and Cooling	Surface $PM_{2.5}$
SOx	Cooling	Surface $PM_{2.5}$
NH_3	Cooling	Surface $PM_{2.5}$
$PM_{2.5}$	Warming and Cooling*	Surface $PM_{2.5}$
VOCs	Warming	Surface $PM_{2.5}$
CO	Warming	

*PM2.5 has both organic and inorganic components, leading to potential warming or cooling effects on global temperature

As illustrated in Table 1, impacts between air pollutants and GHGs are not mutually exclusive, but rather intertwined. This means that policies that target reductions in either GHGs or air pollutants will affect both groups of atmospheric releases, and thus affect both climate change and air quality. These potential mutual benefits are referred to as co-benefits (or co-impacts) (Huang et al. 2023; Karlsson, Alfredsson, and Westling 2020; Shindell 2015). For this reason, the current thesis seeks to investigate implications of biogas for both GHGs and air pollutants, to further understand the potential co-benefits associated with this waste-to-energy strategy. The following sections describe known environmental impacts of conventional waste management practices and biogas and positions this thesis's contribution in the literature.

1.2 Environmental Impact of Current Waste Management Practices

Agriculture, particularly as it relates to waste management and fertilizer application, continues to be a concern including in the areas of public health, air quality, water quality, and climate change (Domingo et al. 2021; Lelieveld et al. 2015; Guo and Chen 2022; Smit and Heederik 2017; Sampat, Ruiz-Mercado, and Zavala 2018). Several studies have identified agricultural emissions as a major contributor to premature death associated with $PM_{2.5}$. In the United States, $PM_{2.5}$ emissions are responsible for 100,000 premature deaths each year, one-fifth of which can be linked to agricultural emissions (Domingo et al. 2021). The same study reported that measures to reduce emissions from agriculture could potentially prevent 7,900 deaths each year, and most of these savings come from changes to waste management and fertilizer application practices. Lelieveld et al. (2015) used the global atmospheric chemistry model ECHAM5/MESSy to determine the link between premature mortality and several emissions sources, including agriculture. Agriculture was determined to be the second largest source of premature mortality, contributing one-fifth of the total premature deaths included in this study. In the USA, they found that 29% of the premature mortalities linked to outdoor air pollution in 2010 were associated with agriculture, the second leading cause following power generation. In Russia, Ukraine, Germany, Türkiye, and Japan, agriculture was responsible for the largest share of outdoor air quality related deaths. Authors indicate that these deaths are likely linked to agriculture's large impact on $PM_{2.5}$ concentrations, which, with ozone, remains the largest contributor to air quality related deaths worldwide. Impacts from agriculture are also not only found at the source of the emissions. Smit and Heedrick (2017) found that agricultural emissions of ammonia are major contributors to increased $PM_{2.5}$ concentrations both at the farm and in downwind urban areas.

Within the agricultural sector, we find that agriculture-related manure is associated with emissions of major greenhouse gas (GHG) and air pollutants. Manure storage is linked to emissions of methane (CH_4), nitrous oxide (N_2O), ammonia (NH_3), hydrogen sulfide (H_2S), volatile organic compounds (VOCs), and particulate matter (PM) (Guo and Chen 2022; Cuéllar and Webber 2008). Synthetic fertilizer use leads to significant emissions of N_2O and NH_3. N_2O emissions associated with synthetic fertilizer use in Canada in 2020 amounted to 17 Mt CO_{2eq} (Food and Agriculture Organization of the United Nations 2020). Emissions from the animal agriculture sector continue to grow with each year, associated with the increasing demand for animal products. CH_4 emissions from

manure management in the US in 2021 amounted to 66 MMT CO_{2eq}, a 69% increase from 1990 levels. N_2O emissions from manure management increased 40% between 1990 and 2021, with 17.4 MMT CO_{2eq} emitted in 2021(EPA 2023).

Agriculture is a major source of emissions in Canada. In 2021, agriculture was responsible for 94% of the emissions of NH_3. Agriculture was also responsible for emitting 3,600,000 tonnes of total particulate matter (TPM) and 460,000 tonnes of NH_3. Agriculture was responsible for 24% of the 1,500,000 tonnes of $PM_{2.5}$ emitted, the second largest source behind all open dust (e.g., from fields and unpaved roads) (Government of Canada 2022b).

Canadian agricultural emissions are increasing. Emissions of N_2O and CH_4 from manure management amounted to 7580.22 kt CO_{2eq} in 2020, an 11% increase from 1990 levels (Food and Agriculture Organization of the United Nations 2020). NH_3 emissions from agriculture in Canada also increased 30% between 1990 and 2021(Government of Canada 2022b).

Emissions from agriculture, particularly NH_3 and N_2O, differ depending on the type of agriculture dominant in the given area. In Canada, in areas with a high livestock population, such as Ontario and Alberta, ammonia emissions from animal activity dominate, whereas in areas with high crop output, such as Saskatchewan, ammonia emissions from fertilizer use are the main concern. In Ontario in 2021, animal production was responsible for emissions of 64,073 tonnes of NH_3 whereas crop production was responsible for 20,925 tonnes of NH_3 that same year. In Saskatchewan, animal production was responsible for 37,583 tonnes of NH_3 in 2021 and crop production was responsible for 70,027 tonnes of NH_3, likely from the impact of synthetic fertilizer use (Government of Canada 2022). This information indicates that, within Ontario, focusing on decreasing emissions associated with animal production would likely be the most effective way to reduce the impact of agriculture on air quality and climate change.

In addition to the environmental impact of agriculture, the environmental impact of landfills is a concern that has gathered global interest due to large emissions of CH_4 and other GHGs. In Canada, landfills in 2020 accounted for 17 Mt CO_{2eq} emissions, ~3% of the total reported CO_{2e} emissions from that year (Government of Canada 2020). Estimates of CH_4 emissions from global

food waste disposal in 2020 accounted for 1.2 Gt CO_{2eq}. In Canada, food waste disposal amounted to 6.4Mt CO_{2eq} in 2020, most of which came from emissions of CH_4 (Food and Agriculture Organization of the United Nations 2020). In 2021, emissions from landfills in Ontario amounted to 135 tonnes of $PM_{2.5}$, 1,726 tonnes of VOCs, and 1,155 tonnes of CO (Government of Canada 2022a).

1.3 Potential Solution: Biogas

The issues with waste management outlined above call for a better approach for waste management: a "closed-loop" solution to waste that could mitigate some of the outlined issues. Anaerobic digestion and biogas production has been viewed as a potential closed-loop recycling management system that has the potential to achieve multiple goals at once (Vaneeckhaute et al. 2018; Metson, Sundblad, et al. 2022; Metson, Feiz, et al. 2022). Anaerobic digestion is defined as the microbial decomposition of organic matter in an oxygen-depleted environment, producing biogas and a nutrient-rich digestate product that can be utilized as a natural fertilizer (Vasco-Correa et al. 2018; Astill and Shumway 2016; Appels et al. 2011; Höhn et al. 2014; Scarlat et al. 2018; Valenti et al. 2018). Biogas production via anaerobic digestion gained popularity in the twentieth century, with interest waning in the 1950s due to the low prices and high accessibility of fossil fuels. Design issues, limited knowledge, and various management issues, along with high costs and high failure rates, further slowed the implementation of biogas production as a renewable energy source in the 1970s. Currently, after having addressed the above issues slowing the implementation of biogas, anaerobic digestion is again gaining interest due to its potential as a closed-loop waste management strategy, as well as its potential as a source of renewable energy and a replacement for synthetic fertilizer usage (Vasco-Correa et al. 2018).

Anaerobic digestion is promoted as a GHG mitigation tool for various reasons. First, anaerobic digestion has the potential to reduce odors and GHG emissions associated with conventional manure management, especially as it concerns the emissions of CH_4 and N_2O, which have 25 and 298 times the global warming potential of CO_2, respectively (Vasco-Correa et al. 2018). Second, the creation of renewable biogas can serve as an alternative to fossil fuels. Third, the carbon-rich digestate created in the process helps to store carbon in soil organic matter when applied to the land, increasing atmospheric CO_2 removal. Fourth, the nitrogen supplied from the digestate also reduces the need for synthetic fertilizer usage and offers opportunities for nutrient recycling (Ebner et

al. 2015; Scarlat et al. 2018). Fiez et al. (2021) used a material flow analysis and case study approach to determine the potential of local biogas production on nutrient recovery. They found that switching to an anaerobic system can lead to a significant increase in the recovery potential of nutrients as opposed to existing waste management systems and saw that biogas can play a key role in the creation of a biobased, or circular, economy. Studies estimating GHG mitigation potential, specifically, are described later in 1.4.

In addition to its role in waste management and emission reductions, biogas is a potential source of renewable energy. In 2014, the global energy potential for biogas was ~1.3 million TJ, with an annual average increase in biogas production of 13.2% (Vasco-Correa et al. 2018). Driven mostly by stricter regulations regarding waste management, Europe is currently the world leader in anaerobic digestion technology. In 2014, ~4% of the total energy supply in Europe was produced via biogas (Pierie et al. 2016). In 2019, Europe had 18,943 combined heat and power plants (CHP) and 725 biomethane injection plants for a total production of 167 TWh biogas and 26 TWh biomethane. Biomethane energy potential is expected to increase in Europe, potentially to 370 TWh in 2030 and 1170 TWh in 2050 (Malet et al. 2023). In the United States, there are currently ~2,100 digestors operating, with that number expected to increase as the recognition of the value of anaerobic digestion as a closed-loop solution to waste management becomes more widely recognized (Vasco-Correa et al. 2018). In 2011, the EPA estimated that on-farm anaerobic digestion units in the U.S. prevented the emissions of 1.2 million metric tons CO_{2eq} (Thompson, Wang, and Li 2013). It is estimated that if all animal waste was treated in an anaerobic digestion unit, the energy created from the biogas created in the process could equal 1.8-3% of the total annual U.S. electricity consumption, around 1015 BTU of energy (Thompson, Wang, and Li 2013).

Anaerobic digestion can be deployed on multiple scales, from household digestors, to farm-scale, to centralized systems (Appels et al. 2011; Vasco-Correa et al. 2018). Centralized anaerobic digestion systems typically codigest animal manure from multiple farms with other organic waste, including food waste (Vasco-Correa et al. 2018). Codigestion of food waste and manure is particularly economically attractive because of the associated increased biogas yield, as well as additional revenue in the form of “tipping fees” generated from importing food waste (Ebner et al. 2015; Appels et al. 2011; Bidart, Fröhling, and Schultmann 2014; Valenti et al. 2018). Studies have

suggested that anaerobic digestion systems only become economically viable when they include some type of codigestion (Astill and Shumway 2016). The energy potential in food waste alone is $33/ton, assuming an electricity cost of $0.10/kWh and a 35% efficiency rate. Additionally, food waste has 3 times the CH_4 production potential of biosolids, including animal wastes, making the codigestion of manure and food waste an attractive option (Kuo and Dow 2017).

Vanneeckhaute et al. (2018) identified the environmental, economic, and social sustainability of utilizing digestate instead of raw animal manure or synthetic fertilizers in Southern Sweden. To quantify the economic dimension of their study, they used net present values and internal rate of return (IRR) as economic indicators. They found that the IRR of digestate handling over a 20-year period increased by 18.5% when manure was codigested with other organic waste, from a 5.1% IRR from monodigestion of manure to a 23.6% IRR when codigestion was considered. Similarly, Astill and Shumway (2016) found that the most profitable scenario was one where animal waste was being codigested with another form of organic waste and suggested that monodigestion was not economically viable under most conditions.

1.4 Potential Emissions Issues

While biogas has the potential to be a 'closed-loop' solution to waste management issues, there are also potential GHG and air pollution emissions concerns that need to be investigated before adopting biogas on a wider scale. Many studies have been conducted to try and estimate the impact of anaerobic digestion on climate change and air quality. Transportation of waste to the digester units and of the digestate is suggested to be a source of significant GHG and criteria air pollutant emissions (Kuo and Dow 2017). Malet et al. (2023) conducted a survey of 30 anaerobic digestion plants in France to determine the emissions from these plants compared to a baseline. The baseline in this case was a scenario where biomass undergoes 'conventional' biomass use, such as soil burying or composting. They found that usage of anaerobic digestion in this study did not lead to significant mitigation of GHG emissions as compared to the baseline. They attributed this to the cost of transporting material to and from the digestor units, which accounted for a sizable portion of emissions. They also found significant variability in their results depending on the type of feedstock used as well as the distance traveled by truck to and from the digestor site. Ebner et al. (2015) also investigated the impact of transportation to the digestor unit on GHG mitigation potential by using

lifecycle assessment methodology to analyze greenhouse gas impacts for an anaerobic digestor that codigests manure and food waste located in the Western New York. The emission areas considered in this study included food waste hauling, digester operation, digestate storage, and land application of digestate. The results indicated that the impact from hauling the food waste was outweighed by the benefits from avoiding landfill emissions. However, only the transportation of the food waste to the digestor unit was investigated, not the transportation of the digestate or the manure also included in the codigestion process.

In addition to concerns involving transportation, there can be GHG emissions associated with the digestion process and the storage of biomass and digestate. Generally, the emissions of CO_2 from biogas combustion and biomass/digestate storage are calculated as neutral as it is considered a biogenic source of emissions. However, emissions of CH_4 and N_2O, both potent greenhouse gases, continue to be a concern. While the current practice of manure waste management leads to increased CH_4 emissions associated with agriculture, CH_4 can also be released during incomplete combustion, as well as during biomass storage and digestate management (Paolini et al. 2018). Hijazi et al. (2016) compared the results of several studies to evaluate the environmental impacts associated with the use of biogas within Europe. Fifteen life-cycle analysis studies were considered, and despite issues in comparing LCA studies with differing methods for presenting results, they were able to find that CH_4 emissions from leaks or overpressure values can be a significant source of GHG emissions associated with anaerobic digestion. Additionally, they found that open storage of digestate can be a hot spot for GHG emissions of both CH_4 and N_2O, but using closed storage instead seems to mitigate this issue.

Vasco-Correa et al. (2018) reviewed the current state of knowledge on anaerobic technology worldwide and some environmental impacts of using anaerobic digestors as a waste management strategy. They found that leakages during biogas production and digestate storage led to a substantial portion of emissions associated with biogas. Additionally, they suggested the digestate created in the anaerobic digestion process must be handled in an environmentally responsible manner, or it can be a source of environmental contamination. Of the studies reviewed in this paper, they found a range of net GHG emissions from biogas (-4 to 3kg CO_{2eq}/m^3 biogas produced), but they note that for most of these studies, this range of emissions was lower than the reference case used. The range they found for animal manure feedstock was between -7 and 2.5 kg CO_{2eq}/m^3 of biogas and between -4 and 2.5

kg CO_{2eq}/m^3 of biogas for food waste feedstock. This study did not report findings regarding codigestion of food waste and animal manure.

Several studies have also investigated the impact of anaerobic digestion on criteria air pollutants, such as NOx, CO, O_3, and $PM_{2.5}$. Paolini et al. (2018) found that both CO and SO_2 concentrations can increase around the location of a biogas plant. CO is a by-product of the incomplete combustion of biogas, whereas SO_2 spikes are likely attributable to H_2S oxidation from diffusive emissions associated with the anaerobic digestion process. Preble et al. (2020) used in situ measurements of an anaerobic digestion unit in California to determine their impact on air quality. Flare emissions of NOx and CO were found to be in the ranges of 5.3-13.9 mgNOx/kg and 6.0-14.8 mgCO/kg. They found that their measurements were 7-14 times greater than the emissions factors from the literature reviewed, indicating high variability in the emissions from anaerobic digestors depending on a variety of factors including geography, feedstock composition, etc. Carreras-Sospedra et al. (2016) studied the emissions of both greenhouse gases and air pollutants from new and existing bioenergy capacity in the state of California. The results of their study indicated that use of maximum bioelectricity production in California could potentially increase NOx emissions by 10%, as well as increases of over 6 ppb in ozone and 2 ug/m3 in PM concentrations within the Central Valley of California. This spike in $PM_{2.5}$ is likely due to emission of both NOx and ammonia, both of which can contribute to the creation of secondary particulate matter (Battini et al. 2014; Preble et al. 2020; Guo and Chen 2022; Smit and Heederik 2017). The increases in ozone found in Carreras-Sospedra et al. (2016) are mirrored in Battini et al. (2014), where they attribute this spike in ozone to increased NOx emissions associated with biogas combustion. They found a 41.6-42.3% increase in ozone concentrations following the installation of an anaerobic digestion unit on a cattle farm in Italy, but these NOx emissions could potentially be mitigated by upgrading the biogas into biomethane. Despite findings that the creation of renewable natural gas through biogas upgrading yields significant energy efficient and energy production, biomethane production is also associated with the highest emissions impacts due to the increased external energy demands (Pierie et al. 2016).

Overall, the studies conducted indicate a need to understand both the emissions of air pollutants and greenhouse gases associated with anaerobic digestion as part of a biogas producing system. Carreras-Sospedra, Williams, and Dabdub (2016) and Battini et al. (2014) considered both air

pollutant and GHG emissions. They found emissions of $PM_{2.5}$ and ozone precursor pollutants, such as NOx, as well as emissions of CO_2. Both studies focused on comparably smaller areas, with Carreras-Sospedra focusing on the Central Valley of California and Battini focusing on a single cattle farm in Italy. There are a variety of factors that influence emissions associated with biogas, including digestate and biomass storage, transportation distances, and utilization pathways. The type of feedstock used also impacts the emissions potential. Therefore, it is crucial to explore the emissions implications of using all biogenic feedstocks at scale, within a spatial economic optimization framework to consider the net potential emissions of a biogas transition.

1.5 Objective

This thesis seeks to evaluate the GHG and air pollutant emissions associated with the use of biogas as a potential replacement for conventional natural gas. We examine the case of Ontario, which we selected as a major agricultural producer whose potential for biogas production has been previously identified (Canadian Biogas Association 2019). We evaluated several biogas uptake scenarios are so trends in emissions can be seen depending on the amount of feedstock consumed. These scenarios vary by the amount of available energy produced, using up to 100% of the feedstock available in the province of Ontario. Resulting biogas plants are cited based on a least-cost optimization that was developed as a part of an ongoing project called Solutionscapes. Solutionscapes is led by Dr. Nandita Basu. The optimization model was developed by Dr. Ushnik Mukherjee in collaboration with Dr. Basu and others. This thesis focuses only on estimating the resulting emissions. It reports the findings of this emissions evaluation and discusses the implication of these findings for air quality, climate change, and the future of waste management in Ontario.

Chapter 2

Background

This thesis examines the emissions implications of optimal biogas siting. Other work related to the analysis of optimal site selection for biogas operation has varied across multiple scales (national, regional, local), methodologies, scopes, and focuses. The work seeks to answer similar questions to that literature: what options exist to help encourage nutrient recycling in agricultural settings, what is the potential for the use of biogas operation, and where might the optimal locations be for biogas use? The following review of relevant literature is separated into three sections based on the following topics: nutrient recycling potential, biogas feasibility, and biogas potential and utilization pathways.

2.1 Nutrient Recycling Potential

Proper nutrient recycling and management is crucial in mitigating the environmental impact of agriculture, and several studies have been conducted on the potential for large-scale nutrient recycling. These studies suggest areas of possible pitfalls and significant potential. Spiegal et al. (2020) used a GIS-based approach to determine the potential of nutrient recycling on multiple scales and a variety of environmental, social, economic, and technological factors. They looked at potential nutrient sources and sinks on a farm, county, and regional level. They found that nutrient density greatly impacted the feasibility of transporting nutrients from sources to sinks. They also suggest that consolidation of manure into a solid state before transportation from source to sink is a critical step in optimizing nutrient recycling. The study calls for nationwide spatial analysis of nutrient recycling potential, considering road networks, economic costs of transport, and source-sink analysis to create optimized nutrient recycling (Spiegal et al. 2020).

Klingmair et al. (2017) adapted a phosphorus (P) material flow analysis (MFA) to analyze the potential to optimize P management in Denmark using a linear programming approach. They write that the recovery and recycling of P from agricultural waste streams, such as manure, could potentially reduce dependence on mineral fertilizer use and close the anthropogenic P cycle. They identify that transporting the manure from regions with surplus manure to regions needing synthetic fertilizer replacements presents a challenge in overall emissions reductions. The optimization model

used in this study was structured to minimize the import of mineral fertilizers while still satisfying the agricultural demand for P in the regions used for the study. The results indicate that despite the issues with transportation costs, P recovered from waste streams such as manure hold significant potential in reducing the requirements for mineral fertilizer use in agricultural settings (Klinglmair et al. 2017). Sampat et al. (2018) used a multiproduct supply chain framework to analyze optimal technology deployment strategies for recovery of value-added products from waste generated at CAFOs. Their study case, involving dairy farms in the state of Wisconsin, indicated that the most effective strategy for waste management and nutrient recycling was to adopt the use of anaerobic digestion units to achieve the goals of energy and nutrient recovery simultaneously (Sampat, Ruiz-Mercado, and Zavala 2018).

Also focused on the potential of nutrient recycling to meet crop nutrient demands, Metson et al. (2020) used both a spatial optimization model and a life cycle analysis-based approach to assess how biogas solutions may contribute to sustainability and circular economy efforts in Sweden. They compared the ability of organic waste (manure and human excreta) applied directly to fields to meet crop nitrogen, phosphorus, and potassium demand versus a scenario where the waste is first processed in an anaerobic digestion unit and the digestate is then applied to fields. The found that a sizable portion of Sweden's nitrogen and phosphorus demand can be met through recyclable sources. They also emphasized the need to minimize transportation costs. As such, the optimization model used was aimed at minimizing transportation distances, but the authors did an additional LCA analysis to determine the potential environmental and economic impact of the proposed anaerobic digestion units. Their LCA analysis found that processing waste through biogas plants resulted in significantly more GHG emissions savings as compared to direct application of waste onto fields. In addition to meeting the crop nutrient demands in Sweden with the recycled waste products, 3.4 TWh of upgraded biogas was produced in the process. Overall, the results of the study indicated that processing waste through biogas plants is a net benefit for the selected economic and environmental indicators, despite the cost of transporting feedstock to the units and digestate to agricultural lands (Metson et al. 2020).

As we work towards meeting more sustainability goals, food systems are likely to change, which could impact the ability for nutrient recycling to meet changing crop demands. Metson et al.

(2022b) used a spatial optimization model and an LCA-based approach to investigate how requirements for waste recycling change under different future visions of Sweden's food system, as compared to a 2015 baseline. The optimization model was set up to minimize transportation costs while also maximizing nutrient and energy recycling through biogas plants. They utilized two stakeholder created forecasts of what Nordic food systems could look like in 2030. Their study highlights the importance of acknowledging how food systems may change to further sustainability efforts and how these changes will impact waste recycling implemented under our current food system structure. Across all three scenarios, the authors found that digestate from waste recycling was able to meet crop demand. While the amount of animal manure was suggested to decrease under future scenarios, this decrease was not shown to impact the number of plant locations.

2.2 Feasibility

The feasibility of biogas plants depends on many factors and should be implemented on sites that satisfy environmental, social, technical, and economic constraints. A holistic approach that accounts for these factors should be taken when evaluating the feasibility of anaerobic digestion units within a region or country (Escalante et al. 2016; Metson et al. 2020; Metson, Feiz, et al. 2022; Metson, Sundblad, et al. 2022; Scarlat et al. 2018; Ma et al. 2005; Cuéllar and Webber 2008). Ma et al. (2005) used a GIS-based approach to identify locations most suitable for bio-energy systems on a county scale. Their model incorporated varying environmental, economic, and social constraints to identify the most suitable locations for anaerobic digestion facilities in Tompkins County, New York. The factors used to help determine these optimal locations included distance from existing dairy farms, distance from roads, distance from transmissions lines, distance from existing natural gas pipelines, and distance from powerplants and substations. They determined that areas with high suitability for biogas plants had relatively less environmental impact, higher economic benefits, and easier access to roads and existing power systems.

Escalante et al. (2016) used a GIS-based approach to develop a spatial decision support system to evaluate the potential for anaerobic digester units on a regional scale in Colombia. The objectives of the study included spatial assessment of crop residue availability, assessment of potential methane yields, determination of potential anaerobic digestor locations, and assessment of

economic and technical feasibility. The authors sought to create this tool to better inform stakeholders and policy makers when they consider implementing anaerobic digestor units and begin to answer questions about feasibility. They found that the economics of biogas use changes at different sizes and feedstock compositions and highlight the importance of spatial variability in determining anaerobic digestion feasibility on any large scale.

Similarly to Escalante et al. (2016), Metson et al. (2022) recognized the impact of social, geographic, environmental, and economic factors on biogas placement and feasibility. Metson et al. (2022) developed a four-stage methodological approach to determine which localization factors most constrains potential biogas locations on a national, regional, and local scale, and how the introduction of several biogas plants, as opposed to a solitary unit, affect the optimal locations. Their methods included a multitude of constraints, including social constraints related to proximity to residential areas and protected areas, as well as geographic constraints such as proximity to roads capable to supporting heavy trucks and located in an area with enough manure. At a national, regional, and local level, access to roads remained the most significant constraint. They also concluded that placing more than one plant at a time did in fact affect the optimal location for plants, which indicates that biogas plants have an impact on the location selection of other biogas plants. The authors note that they did not include proximity to existing electrical infrastructure in their study, which may also impact the distribution of biogas plants.

Thompson et al. (2013) used a GIS-based approach to identify the optimal sites for centralized anaerobic digestor units in Addison County, Vermont. The authors note that a major obstacle in optimizing for centralized anaerobic digestor units on a large-scale are issues with manure transportation and handling costs potentially offsetting the economic savings of biogas. The study uses a land suitability model framework to evaluate environmental, social, and economic constraints to selecting optimal sites for biogas plants within Addison County. Their analysis concluded a maximum transportation distance of 22km was required to maximize economic benefits and minimize environmental costs. They found that the optimal number of digestor units needed to service this county was 6, servicing 116 dairy farms. They found that increasing the number of digestor units in this area above 6 did not allow additional farms to be serviced but reduced the transportation distances from the included farms. Therefore, the cost of increasing the number of units beyond 6

potentially outweighed the benefits, if the goal was cost reduction. Their results show the balance that must be struck between minimizing economic costs and environmental costs in determining optimal sites for centralized biogas production (Thompson, Wang, and Li 2013).

2.3 Energy Potential and Utilization Pathways

Biogas site optimization studies often included analysis of biogas energy potential, though the studies differ in terms of scale, focus, and methodology. On a continental scale, Scarlat et al. (2018) performed a spatial analysis of biogas potential from livestock manure in Europe at a high level of resolution (1km spatial resolution). They determined suitable areas for biogas production facilities to be areas with a high concentration of feedstock and close to an existing road network. They found that the manure produced within Europe has a biogas potential of 25.7 million m^3 CH_4, which is 4.5% of the overall natural gas consumption in the continent. In terms of biomethane potential, they determine that livestock manure has the potential to yield 17.8 billion m^3 biomethane in Europe, which is the equivalent of the annual natural gas consumption in Belgium. They also found that different countries within Europe had differing biogas potential, indicating a need to investigate biogas feasibility on a case-by-case basis. Their results suggest that using site optimization on a large, continental scale is possible, but it is likely more effective on a regional scale.

Valenti et al. (2018) used a GIS modeling approach to determine the potential of regional biogas production via codigestion of food and agricultural waste in the Catalina province of Sicily. They determined the optimal locations of potential biogas plants in Catalina based on type and quality of feedstocks, locations of farms and processing facilities, and proximity to existing road networks. The optimal biogas power generation system suggested four biogas plants in this region, capable of handling 211,000 t/year of biomass and generating 23.1 GWh-e/year, which would satisfy 27% of the total agricultural electricity demand in this region.

Dagnall et al. (2000) used a GIS-based approach to help inform optimal locations for centralized anaerobic digestor units in the UK, and specifically look at the biogas energy potential for East Anglia sites. The study focused on CHP plants as opposed to biomethane creation and the feedstock used for analysis was livestock manure, primarily poultry litter and pig manure. A major

constraint mentioned in this study was proximity to existing road networks and the authors only considered sites within 1km of a major roadway. Additionally, proximity to existing electricity grid connection points were considered a site constraint and only sites within 5km of a grid substation were considered as a potential site for centralized biogas plants. They found the energy potential of optimized site in East Anglia was 17.2 MWe, assuming all available resources could be collected and converted to energy. The authors notes that to maximize the benefits associated with biogas energy creation, the centralized sites needed to be located near the feedstock and near electrical grid inputs. However, the study did not analyze potential social and environmental constraints to optimal biogas site selection (Dagnall, Hill, and Pegg 2000).

Also on a regional scale, Batzias et al. (2004) used a GIS-based approach to estimate the regional distribution of biogas potential in Greece based on livestock residue. The authors created a tool (ABEPE) which they described as a "livestock manure assessment tool", which they used in conjunction with statistical data to determine the biogas potential from manure in Greece. They were able to determine which regions in Greece has the largest biogas potential, based upon the availability and type of feedstock in the region. They were also able to determine that biogas potential within Greece is likely to increase with each passing year. They suggest that this biogas potential could be utilized to meet the increasing demand for natural gas within the country by upgrading the biogas into biomethane (Batzias, Sidiras, and Spyrou 2005).

Differing biogas utilization pathways also influence both site selection and energy potential. Pierie et al. (2016) used a process optimization and LCA-based approach to investigate the potential of biogas production pathways in the EU. They analyzed three possible utilization pathways; green gas (renewable natural gas) production, combined heat and power (CHP), and waste treatment, which produces both green gas and CHP while also creating fossil-equivalent quality fertilizers from the digestate. They found that the renewable natural gas, or green gas, utilization pathway was the most energy efficient, with an average of 73% of energy extracted as green gas, compared to 57% as CHP and 44% in the waste management pathway. The green gas pathway could retain the largest share of energy from the feedstock, but the authors acknowledge the trade-offs involved with the increased energy requirement associated with the biogas upgrading process. Despite this trade-off, the green gas utilization pathway was found to be the most capable of achieving higher energy production levels.

Bidart et al. (2014) used a GIS-based approach to determine the biogas potential in Chile based on the utilization of manure and crop residue. They analyzed two possible utilization pathways; biogas-to-energy (CHP) and biogas-to-upgrade (renewable natural gas). They found that biogas potential varied by region in Chile, with one region accounting for 64% of the biogas-to-upgrade production potential in the country. They expressed the necessity for biogas potential to be investigated on a case-by-case basis, drawing that conclusion from the fact that biogas potential in the country was clustered within one region. They also found that few of the farms included in the study had adequate conditions for developing biogas plants, indicating a need for a network of biogas plants handling the waste from multiple farms. They also indicate that the process of codigestion increased the biogas potential within the country, increasing the overall possible methane yield as compared to the monodigestion of manure.

Focused entirely on biomethane potential, Hohn et al. (2014) used a GIS-based approach to determine the energy potential and optimal locations for biomethane within Finland. Suitable locations for biogas plants were determined via the distribution of biomass sources and biomethane usage end points, emphasizing the importance of minimizing transportation distances to reduce the associated emissions. Biomethane potential was studied on a regional scale in three regions in southern Finland. In regions with access to an existing natural gas grid, locations with high availability of biomass sources close to the grid were chosen. In regions without access to a natural gas grid, locations were chosen by the availability and proximity of multiple biomass feedstock, with the assumption that it was more feasible to produce the biomethane close to the source of the feedstock and transport it from there to the consumer market, as opposed to transporting the feedstock to a location closer to the consumer market first. They found that the biomethane potential of these three regions was 2.8 TWh energy, most of which was derived from agricultural materials, such as livestock manure (Höhn et al. 2014).

2.4 Novelty of Research

The majority of the research conducted on site optimization of biogas emphasizes that social and environmental factors, such as potential emissions, should be taken into consideration when

considering large-scale anaerobic digestion potential and feasibility. Many studies have been conducted investigating the potential and feasibility of large-scale biogas production, including the potential energy creations and limitations of the biomethane utilization pathway, though none reviewed have investigated the emissions savings of biogas as a waste management strategy versus the potential emissions associated with anaerobic digestion plants. This reflects a potential gap in research that this study hopes to fill.

Chapter 3

Methods

This section will describe the process of estimating emissions from a potential biogas transition in Ontario. The effect of emissions is based on the difference between a baseline scenario, and biogas adoption scenarios. Figure 1 displays the difference between the baseline and biogas scenarios. In the baseline scenario, all waste is being handled via conventional waste management methods. Specifically, all manure is sent to conventional manure management systems, and all food waste is landfilled. In the biogas scenarios, biogas plants divert some of these feedstocks to digesters that create digestate and biogas. Biogas is upgraded and injected into the existing natural gas pipeline network. Digestate is stored for land application.

Biogas scenarios were developed using an optimization framework as part of an ongoing project outside this thesis, in collaboration with Prof. Nandita Basu and Dr. Ushnik Mukherjee. Each scenario is based on a least-cost optimization to produce some fraction of the total potential energy from biogas in Ontario. The total potential energy is based on utilizing 100% of the available feedstock, including agricultural manure and food waste. In addition to that, this thesis also evaluates emissions from producing 25%, 50%, and 75% of the potential energy from biogas, respectively.

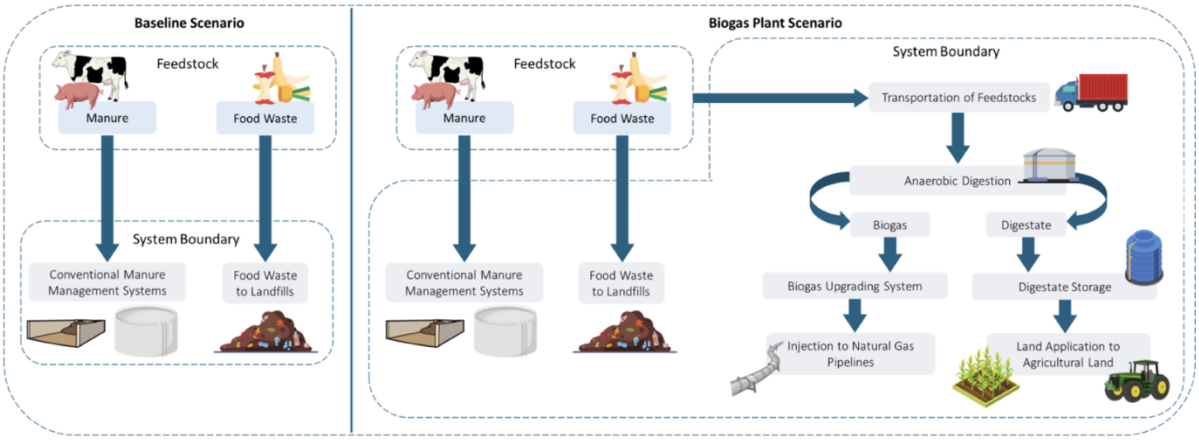


Figure 1. Process diagram distinguishing the difference in pathways between the baseline and biogas scenarios (created by Nettie Wallace, Ushnik Mukherjee, Savannah Bindas, and Camila Granito)

Figure 2 describes the areas for potential emissions sources and offsets associated with the biogas scenarios. This section will describe the methods for estimating the emissions associated with the steps in the biogas scenarios, as outlined in Figure 2. Additionally, this section will also describe the method for estimating emissions associated with the baseline scenario, which include emissions associated with conventional manure management and food waste landfilling.

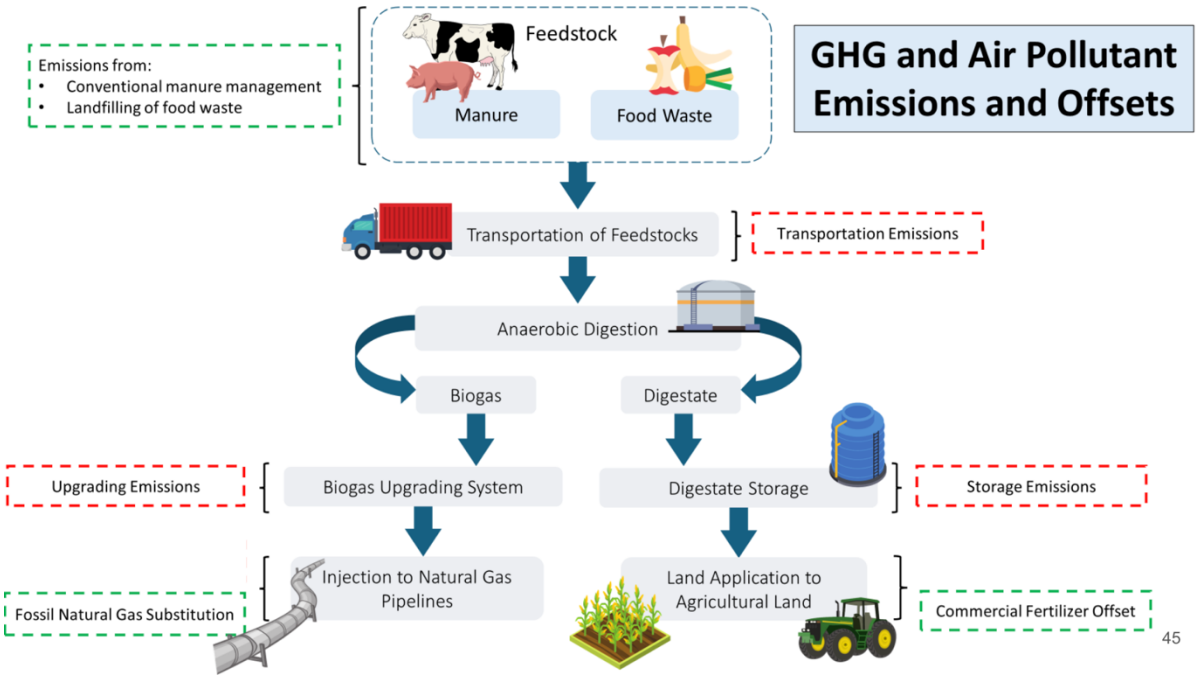


Figure 2. Process diagram describing the sources of emissions (red boxes) and offsets (green boxes) associated with the biogas scenarios (created by Nettie Wallace, Ushnik Mukherjee, Savannah Bindas, and Camila Granito)

As stated in Chapter 1, the aim of this thesis is to estimate emissions of air pollutants and greenhouse gases of a biogas transition. Emissions identification and estimation is based on an in-depth search of relevant literature, reports, and databases to develop a list of potential equations and emissions factors. When multiple approaches were available, the most relevant and accurate approach was chosen based on comparison with measurements, where possible, or following the most recent and relevant literature. Most emissions are estimated based on emission factors. Emission factors were selected based on quality and similarity to the source in question. This was assessed based on the emission factor quality rating, recency, and specific source properties, such as location, process details, and emission control. The remainder of this chapter provides precise details of all emission estimation methods for each source. Example calculations are provided in Appendix A. The Jupyter Notebook used to perform all calculations is attached as Appendix C.

3.1 GHG Emissions Estimation

This section will outline the GHG emissions associated with the baseline and biogas scenarios, specifically emissions of CH_4, N_2O, and CO_2.

Emissions from all greenhouse gasses were converted to their CO_2 equivalent emissions using their respective global warming potential (GWP):

$$E_{CO2e} = \sum_{GHG} \mathrm{E_{GHG}} \times GWP_{GHG}$$

Equation 1

Where:

GHG: The three major long-lived GHGs, CH_4, N_2O, and CO_2

E_{CO2e}: CO_2 equivalent emissions for each GHG, namely CH_4, N_2O, and CO_2

GWP: 100-year Global Warming Potential of each gas. For CH_4 it is 25; for N_2O it is 298, and for CO_2 it is 1.

Emissions are estimated for the entire Province of Ontario based on inputs provided by Dr. Ushnik Mukherjee. Those inputs reflect a spatial optimization covering the province with a 5 by 5 km grid reflecting the locations of feedstocks, the natural gas pipeline network, and optimal placement of biogas plants. This thesis calculates emissions for the entire province based on provincial totals of relevant activities. For example, emissions from manure management are based on the total amount of manure processed from each animal type in each management system. Similarly, emissions from transportation of feedstocks are based on the total distance travelled for all trips.

3.1.1 Baseline Emissions

The baseline GHG emissions will be described first, specifically focusing on the CH_4 and N_2O emissions associated with conventional manure management and food waste landfilling.

3.1.1.1 CH4 Manure Management Emissions

To determine a more precise CH_4 manure management emissions factor, the IPCC Tier 2 approach was used via the equation below (Metson et al. 2020).

$$E_{CH4,MM} = \sum_{i,k,MMS} m_i \times VS_i \times B_{0,i} \times \rho_{CH_4} \times MCF_{MMS,ik} \times f_{MMS,ik}$$

Equation 2

Where:

$E_{CH4,MM}$: Methane emissions from Manure Management (MM) across all livestock types for the province (kg of CH_4)

i: Feedstock type. For this source, only livestock feedstocks apply, namely beef cattle, dairy cattle, hog, broiler/poultry

m_i: Mass of manure (kg) by livestock type

k: Manure management system type (liquid, solid, pasture, other)

VS_i: Volatile solids in manure by livestock type (kg VS/kg of manure).

$B_{0,i}$: Maximum methane generation potential of manure from livestock type (m^3 CH_4/kg of VS in manure).

$MCF_{MMS,ik}$: Methane conversion factor (%) is the portion of the methane generation potential realized by a manure management system and livestock type.

$f_{MMS,ik}$: Fraction of livestock manure managed in manure management system

ρCH_4: Density of methane (kg/m^3)

The mass of manure managed in conventional systems by livestock type is provided in Table 2. The same procedure is used to estimate emissions from this source for the baseline scenario as all biogas scenarios. Note that, by the 100% scenario, all manure is sent to biogas systems, so the amount managed in conventional systems is zero.

Table 2. Manure Managed in Conventional Systems in Ontario

Livestock type	Baseline (tonnes)	25% Scenario (tonnes)	50% Scenario (tonnes)	75% Scenario (tonnes)
Beef Cattle	8,359,902	7,955,516	6,122,545	3,783,211
Dairy Cattle	30,261,825	29,797,098	22,960,068	11,888,524
Hog	5,663,619	4,861,996	1,566,461	286,804
Broiler/Chicken	558,382	353,272	119,910	21,608

Within each manure type, different management systems are used. Table 3, Table 4, and Table 5 show the different f, VS, B0, and MCF values for each manure type or management system.

Table 3. Share of each manure type managed by differing management systems in Canada (% f values (f*100)) (Government of Canada 2022a)

Livestock type	Liquid Systems (%)	Solid storage and Drylot (%)	Pasture, Range and Paddock (%)	Other Systems (%)
Beef Cattle	5.3	45	45	4.2
Dairy Cattle	64	18	16	2.9
Hog	97	3	0	0
Broiler/Chicken	7	92	0.6	0.6

Table 4. Livestock manure volatile solid (VS) and maximum methane generation potential (B0) for each livestock type

Livestock type	VS (kg VS/kg of manure) *	B_0 (m^3 CH_4/kg of VS in manure) +
Beef Cattle	0.06545	0.247
Dairy Cattle	0.15768	0.19
Hog	0.0847	0.48
Broiler/Chicken	0.432	0.39

*(Canadian Biogas Association 2019) +(EPA 2009)

Table 5. Methane conversion factors (MCF) for different manure management strategies in Canada (Government of Canada 2022a)

Manure management system	Manure management subsystem	MCF (Fraction)
Liquid	Earthen Basin – No Crust	0.2
	Earthen Basin – Crust	0.13
	Tank – No Crust	0.2
	Tank – Crust	0.13
	Slatted Floor	0.2
Solid	Solid storage and drylot	0.02
	Pasture, Range and Paddock	0.01
	Pasture, Range and Paddock	0.015*
Other systems	NA	0.01
Other systems	NA	0.015*
*Applicable only to Broiler and Chicken livestock types		

An example calculation showing the total estimated emissions of CH_4 for the province is provided in Appendix A.

3.1.1.2 N_2O Emissions from Manure Management

Manure nitrogen loss from manure management systems leads to direct and indirect (such as volatilization and leaching) emissions of N_2O. To determine the amount of direct and indirect N_2O emitted from manure management, an emissions factor approach was used by utilizing Equation 3, the emissions factors in Table 6, and the amounts of manure by feedstock type, m_i, in Table 2.

$$E_{N2O,MM} = m_i \times EF$$

Equation 3

Table 6. N_2O emissions factors for manure management systems by livestock type (Government of Canada 2022a)

Livestock type	Direct N_2O emissions (kg N_2O/tonne manure)	Indirect N_2O emissions (Volatilization and leaching, kg N_2O/tonne manure)
Beef Cattle	0.13	0.013
Dairy Cattle	0.019	0.005
Hog	0.014	0.025
Broiler/Chicken	1.02	0.28

3.1.1.3 Food Waste Landfilling

Similarly to section 3.1.1.1, the IPCC Tier 2 approach was used to determine CH_4 emissions from food waste via Equation 4, shown below (Hiraishi et al. 2018).

$$E_{CH_{4,LF}}(t) = \left[m_{CH4,gen}(t) - R(t)\right] \times (1 - OX) + \left[m_{CH4,flare}(t) \times \left(1 - \eta_{flare}\right)\right]$$

Equation 4

Where:

$E_{CH4,LF}(t)$: Methane emissions from Landfilling (LF) for the province by year (tonnes of CH_4). Note that only a single inventory year is calculated, so t = 1.

$m_{CH4,gen}(t)$: Methane generated at year of inventory t (tonnes of CH_4/y)

$R(t)$: Methane recovered at inventory year t (tonnes of CH_4/y)

OX: Fraction of CH_4 oxidized within the soil or any other material covering the waste deposited in the landfill

$m_{CH4,flare}(t)$: Methane flared in the landfill gas recovery system in year t (tonnes of CH_4/y)

η_{flare}: Efficiency of the flare landfill gas recovery system (%)

To complete Equation 4, an additional equation (Equation 5) is needed to determine CH_4 generated.

$$m_{CH4,gen}(t) = \sum_{x}^{t}\left[A \times k \times FW_x \times MCF \times DOC_x \times DOC_f \times F \times \frac{16}{12} \times e^{-k(t-x)}\right]$$

Equation 5

Where:

t : Denotes year of inventory (here, t=1)

x : Denotes every year when waste is added to the landfill (here, x =1)

A : Denotes a normalization factor for correcting the summation

k : Decay rate constant for food waste (1/year)

FW_x : Food waste sent to landfill in year x

MCF : Methane conversion factor in year x

DOC_x : Degradable organic carbon in food waste deposited in year x (tonne of C/tonne of food waste)

DOC_f : Share of DOC that decays

F : Fraction of CH4 in landfill gas

Table 7 was used to help fill in the elements needed to compute Equation 5, specifically k, MCF, DOC_x, DOC_f, and F.

Table 7. Parameter values for estimating methane generation at landfills (Environment and Climate Change Canada 2023)

Methane Generation Estimation Parameters	Values
k	0.06 (Dry conditions), 0.185 (Wet conditions)
MCF	1
DOC	0.15
DOC_F	0.7
F	0.5

Table 8 was used to fill in the parameters needed to compute Equation 4, specifically $\eta flare$ and OX.

Table 8. Parameter values for estimating methane emitted at landfills

Methane Emitted Estimation Parameters	Values
$\eta flare$	99.7%
OX	0.1

Finally, Table 9 shows the amount of food waste sent to landfill, *FWx.* The same procedure is used to estimate emissions from this source for the baseline scenario as all biogas scenarios. Note that food waste has a high energy content, and is fully consumed in the biogas system by the 50% scenario, so the amount landfilled is zero.

Table 9. Food waste sent to landfill by scenario.

Scenario	Food Waste Landfilled (tonnes)
Baseline	3,090,878
25%	133,824

Starting with Equation 5, the parameters from Table 7 and the output of the model showing total food waste (Table 9) were used to determine CH_4 generated in tonnes for the model year. Following Equation 5's completion, Equation 4 was computed using the results for CH_4 generated and the parameters in Table 8. Additionally, three emissions scenarios were considered; a 44% methane recovery scenario, a 66% methane recovery scenario, and a scenario where no methane was recovered.

3.1.2 Biogas Scenarios

Next, the emissions and offsets associated with the biogas scenarios were evaluated. The sources of these emissions are outlined below but can also be found in Figure 2. The procedures outlined below were conducted 4 times to represent each biogas scenario; 25%, 50%, 75%, and 100%.

3.1.2.1 Transportation Emissions

An emissions factor approach was used to determine potential emissions from transportation. Equation 6 yields the total emissions for the province for a given emission (e.g., GHG or air pollutant) and biogas scenario.

$$E_{GHG,TRAN} = \sum_i \frac{m_i \times VKT_i}{\gamma_i} \times EF_{GHG,TRAN}$$

Equation 6

Where:

$E_{GHG,TRAN}$: Emissions of each type of GHG (or air pollutant) from transportation for each scenario (g)
$EF_{GHG,TRAN}$: Emissions factor for GHG (or air pollutant) (g/km)
m_i: Mass of manure transported by feedstock type by scenario (tonnes) (Table 11)
VKT_i: Vehicle kilometers travelled (km) by scenario; total distance travelled by all trucks to transport all feedstock
γ_i: Truck carrying capacity by feedstock type (22.7 tonnes for manure hauling vehicles and 20 tonnes for food waste hauling vehicles)

Remaining inputs for Equation 6 are in Table 2, Table 10, and Table 11. The mass of manure transported by feedstock type by scenario (m_i) can be calculated from Table 2. That table shows the amount of manure still handled by conventional manure management systems in each biogas scenario. The amount of manure consumed in the biogas system can be found by taking the difference between these numbers, and the baseline. Table 10 shows the emissions factors used in Equation 6. Example calculations and alternative emission factors for different speeds are provided in Appendix A. Table 11 shows the total distance travelled to move feedstock by biogas scenario and feedstock type (VKT_i).

Table 10. Feedstock transportation emissions factors for GHG and air pollutants associated with manure hauling vehicles assuming speeds between 44.25 kph and 52.30 kph on rural unrestricted roads during average spring temperatures (Wang 2020)

Emission Type	Emissions Factor (g/km)
CO_2	1331
CH_4	0.03
N_2O	0.002
NH_3	0.01
NOx	6.32
SO_2	0.01
$PM_{2.5}$ (Primary exhaust)	0.28
VOCs	0.27

Table 11. Total distance travelled to move feedstock and total feedstock moved by biogas scenario and feedstock type.

Feedstock type	25% Scenario (km) and (tonnes)		50% Scenario (km) and (tonnes)		75% Scenario (km) and (tonnes)		100% Scenario (km) and (tonnes)	
Beef Cattle	5787	404386	9659	2237357	27022	4576690	71193	8359902
Broiler/Chicken	13822	205110	15348	438472	15663	536773	15277	558382
Dairy Cattle	1309	464727	4566	7301757	16074	18373300	24110	30261825
Hog	2820	801623	7315	4097158	13537	5376815	12617	5663619
Food Waste	142962	2957053	118542	3090878	217775	3090878	281452	3090878

3.1.2.2 Upgrading Emissions

Similarly to transportation emissions, an emissions factor approach was used to determine upgrading emissions for each biogas scenario and emission type (GHG or air pollutant) (Equation 7).

$$E_{GHG,UG} = \theta \times EF_{GHG,UG} \times \eta_{UG} \times \frac{0.947817\ MMBtu}{GJ}$$

Equation 7

Where:

$E_{GHG,UG}$: Emissions of each type of GHG (or air pollutant) from biogas upgrading for each scenario (lbs)

$EF_{GHG,UG}$: Emissions factor for GHG (or air pollutant) from upgrading biogas (lbs/MMBtu input)

θ: Natural gas produced by scenario (GJ) (Table 13)

η_{UG}: Efficiency of upgrading process (0.85)

Emissions factors for biogas upgrading can be found in Table 12. The amount of natural gas produced by scenario is shown in Table 13. This is the amount produced per year, but emissions are evaluated for a single year.

Table 12. Upgrade to pipeline injection on-site criteria pollutant and GHG emissions factors (EPA 2016)

Emissions Type	**Emissions Factors (lb/MMBtu input to process)**
NOx	0.0051
CO	0.0042
PM	0.0011
VOC	0.0006
SOx	0.0036
CH_4	0.4358
N_2O	0.00022
CO_2	86.1

Table 13. Amount of natural gas produced by scenario.

Scenario	Natural gas generated (GJ/yr)
25%	201,414,597
50%	218,254,563
75%	31,954,459
100%	8,359,902

3.1.2.3 Leftover Manure (Manure Management Emissions)

In several of the scenarios analyzed (25%, 50%, 75%), there was additional manure not utilized by the biogas system. The assumption is that this leftover manure would be handled in the conventional way, creating emissions like those found in the baseline. The same process used to determine baseline manure management emissions is used here to determine emissions from the manure leftover. How much manure was leftover was determined by adding the manure used (tonnes/yr) and subtracting from the total manure value determined by the baseline. Methods for determining baseline manure management emissions can be found in 3.1.1.1 and 3.1.1.2.

3.1.2.4 Leftover Food Waste

The same procedure outlined for baseline food waste emissions was utilized here to determine the emissions from food waste not consumed by the biogas system. It is assumed that whatever food waste was leftover was transported to a landfill and handled the same way as the baseline. The leftover food waste was determined by adding together all the food waste used in the scenario and subtracting from the total amount assumed in the baseline. Methods for determining emissions from food waste landfilling can be found in 3.1.1.3.

3.1.2.5 Energy Production Offset

The emissions offset from utilizing biogas-generated natural gas instead of conventional natural gas is assumed to come from the avoided emissions associated with the production of conventional natural

gas. An emissions factor approach was again utilized here, with an emission factor based on the amount of natural gas produced (in MJ) in each scenario, and for each emission type (GHG or air pollutant) (Equation 8).

$$E_{GHG,NGOFF} = \theta \times EF_{GHG,NGOFf} \times \frac{1000MJ}{GJ}$$

Equation 8

Where:

$E_{GHG,NGOFF}$: Emissions of each type of GHG (or air pollutant) from the offset of natural gas production for each scenario (lbs). A value of 4 g CO_{2e} / MJ was used.
$EF_{GHG,NGOFF}$: Emissions factor for GHG (or air pollutant) for natural gas production (g $CO_{2,e}$/MJ)
θ: Energy content of natural gas produced by scenario (GJ) (see **Error! Reference source not found.**)

The emissions factors in Table 14 are taken from Liu et al. 2021 and were used to determine potential emissions offset from avoided natural gas production.

Table 14. Well-to-Gate GHG emissions for conventional natural gas sources in Canada (Liu et al. 2021)

Natural Gas Source	**Well-to-gate emission factor (g $CO_{2,e}$/MJ)**
Western Canada	3.1 – 4 (Calculated by Liu et al.) Previous Canadian studies: 6.2 – 12

3.1.2.6 Fertilizer Offset

The following analysis was used to determine the emissions avoided from synthetic fertilizer production. It is assumed that the nutrients found in the digestate would replace those found in synthetic fertilizer, thus offsetting the emissions associated with the production of those fertilizers. This replacement was determined on a nutrient basis, with mass of nitrogen and phosphorus in manure (Table 15) replacing nitrogen and phosphorus (Table 16) for synthetic fertilizers used in Ontario (Table 17). Since phosphorus-containing fertilizers DAP and MAP also contain nitrogen,

these fertilizers were replaced first, and the remaining nitrogen available then replaced the nitrogen-only fertilizers. An emissions factor approach was used here as well, illustrated in Equation 9.

$$E_{GHG,fert} = \sum_{j,k} \left(\frac{m_j \times R_l}{r_{j,l}} \right) \times EF_{GHG,l}$$

Equation 9

Where:

$E_{GHG,fert}$ = Avoided CO_2e emissions from fertilizer production (tonnes)

m_j = mass of nutrient replaced for $j = N$ for nitrogen or K for phosphorus (tonnes) (Equation 10 and Equation 11)

R_l = mass ratio of each fertilizer type l used in Ontario to total fertilizer use in the province (for nitrogen, this excludes phosphorus-containing fertilizers DAP / MAP) (Table 17)

$EF_{GHG,l}$ = Emissions factor for GHGs for each fertilizer type (excluding DAP / MAP) (ton CO_2e / ton fertilizer produced) (Table 18)

$r_{j,l}$ = mass ratio of nutrient in each fertilizer (for j = *nitrogen*, l excludes DAP / MAP) (Table 16)

Table 15. N and P mass ratios for each manure type (United States Department of Agriculture 2009)

Livestock Type	P to Manure Ratio	N to Manure Ratio
Beef	0.001	0.003
Dairy	0.001	0.007
Hog	0.001	0.008
Broiler	0.003	0.011

Table 16. Fertilizer types and N and P mass ratios for each fertilizer type. Those denoted with N/A have no ratios associated with that nutrient type.

Fertilizer Type	N ratio	P ratio
DAP	0.18	0.201
MAP	0.11	0.227
Urea	0.46	N/A
Ammonia	0.82	N/A
UAN	0.28	N/A

Table 17. Percentage of each type of fertilizer used in Ontario. Diammonium phosphate (DAP) and monoammonium phosphate (MAP) percentages are determined using only the amounts of DAP and MAP used in Ontario, as they are the only fertilizers included that have phosphorus content. Urea, ammonia, and Urea Ammonium Nitrate (UAN) are similarly determined together, because they only include nitrogen content (Statistics Canada 2016)

	Fertilizer Type	Percentage
P-containing	DAP	47.3
	MAP	52.7
N-containing	Urea	51.4
	Ammonia	1.3
	UAN	47.3

Table 18. Fertilizer type and emissions factor EF (Bentrup 2018)

Fertilizer Type	EF (ton CO2e / ton product)
DAP	0.67
MAP	0.46
Urea	1.01
Ammonia	2.49
UAN	1.33

The mass of nutrient in synthetic fertilizer replaced by digestate was determined using the nutrients available in the digestate, beginning with the phosphorus containing fertilizers (DAP / MAP). Equation 10 was used to determine the amount of phosphorus mass available in the digestate to replace phosphorus in synthetic fertilizer.

$$m_K = \sum_i m_i \times r_{K,i}$$

Equation 10

Where:

m_i = Feedstock used in digestor (tonnes) by feedstock type (beef, dairy, hog, broiler) (Table 11)

$r_{K,i}$ = Phosphorus ratios for each feedstock (beef, dairy, hog, broiler) (mass of nutrient per mass of manure) (Table 15)

Because phosphorus-containing fertilizers also contain nitrogen, any nitrogen available in the digestate was first allocated to replacing phosphorus-containing fertilizer. However, following this allocation, there remained additional nitrogen in the digestate that could replace other nitrogen-containing fertilizers and thus avoid GHG emissions associated with their production. That remaining nitrogen mass was found using Equation 11, found below.

$$m_{NL} = \sum_k m_K \times \left(\frac{\left(R_l \times r_{N,l} \right)}{r_{P,l}} \right)$$

Equation 11

Where:

m_{NL} = mass of nitrogen leftover to replace nitrogen-containing fertilizer

m_K = phosphorus available (tonnes) (result from Equation 10)

R_l = mass ratio of fertilizer type *l* used in Ontario to total fertilizer use in the province, here, for *l* = DAP, MAP (Table 17)

$r_{N,l}$ = ratio of nitrogen in DAP / MAP (Table 16)

$r_{P,l}$ = ratio of phosphorus in DAP / MAP (Table 16)

3.1.2.7 Digestate Storage

Storage emissions from digestate were calculated using a similar method to determine methane emissions from manure, outlined in 3.1.1.1. However, differing MCF and f_{MMS} values were used. The MCF value for storage of digestate was changed to 0.01 (Gavrilova et al. 2019). The f_{MMS} values were changed to reflect the percentage of each type of manure processed within the scenario. Equation 12, below, outlines how the f_{MMS} values were determined for beef, broiler, hog, and dairy manure. After determining the new f_{MMS} and MCF values, the same process outlined in 3.1.1.1 was used.

$$f_{MMS,i,k} = \frac{m_{i,k}}{\sum_i m_{i,k}}$$

Equation 12

3.2 Air Pollutant Emissions Estimation

This section will outline the air pollutant emissions associated with the baseline and biogas scenarios, specifically emissions of $PM_{2.5}$, NOx, SOx, VOCs, NH_3, and CO. As with emissions of GHGs, pollutant emissions are estimated using inputs from a spatial optimization model. Those inputs (e.g., manure transported, methane generated) are first summed across the model grid to obtain provincial totals, so that emissions are estimated at the provincial scale. Unlike GHG emissions, the emissions of air pollutants cannot readily be summed into a total metric like CO_2 equivalent to reflect their combined impact and are thus presented separately.

3.2.1 Baseline Emissions

The baseline air pollutant emissions will be described first, specifically focusing on the emissions associated with conventional manure management and food waste landfilling.

3.2.1.1 Manure Management

To determine air pollutant emissions from conventional manure management, an emissions factor approach was used. Equation 13 yields the total emissions for the province for a particular air pollutant and biogas scenario. Table 19 shows the emissions factors used for each type of conventional manure management system. In this analysis, emissions of NH_3, $PM_{2.5}$, and VOCs are evaluated.

$$E_{AP,MM} = \sum_{i,k} \frac{\left(\frac{m_i}{\mu_i}\right)}{EPA\ AU_i} \times f_{MMS,ik} \times EF_{AP,MM,ik}$$

Equation 13

Where:

$E_{AP,MM}$: Emissions of each type of air pollutant from manure management for each scenario (g)

k: Manure management system type (here: liquid, solid, pasture. "Other" is excluded due to lack of emissions factors)

$EF_{AP,MM,i,k}$: Emissions factors for air pollutant by livestock type and manure management system (g/AU) (see Table 19).

m_i: Mass of manure generated by livestock type (kg) (see Table 2)

μ_i: Annual average manure generated per animal by livestock type (kg per year per animal) (see

$EPA\ AU_i$: Animals per Animal Unit as defined by the U.S. Environmental Protection Agency (EPA) (see Table 21)

$f_{MMS,i,k}$: Fraction of livestock manure managed in manure management system (see Table 22, modified from Table 3).

Emissions factors (see Table 19) were not available for all manure management systems (e.g., no values for "Other Systems"). Thus, for these calculations, $f_{MMS,i,k}$ is altered slightly from Table 3 to produce Table 22. Hog manure was unaffected. Emission factors were missing for less than 2% and 3% of manure from chicken and dairy cattle, respectively. As those amounts were deemed small, these emissions were not estimated. For beef cattle, 10% of the manure (stored in liquid or "other" systems) was re-assigned equally to solid and pasture systems.

Table 19. Emissions factors (NH_3, VOC, $PM_{2.5}$) for differing animal types and manure management systems, shown in grams per animal unit (AU)

Manure Management System	NH_3 emissions factor (g / AU)	VOC emissions factor (g / AU)	$PM_{2.5}$ emissions factor (g / AU)
Beef Cattle (solid system)	9966		5753
Beef Cattle (pasture)	119*		
Dairy Cattle (solid system)	4620		1041
Dairy Cattle (liquid system)	18120	2038	
Dairy Cattle (pasture)	119*		
Hog (solid system)	7791		3624
Hog (liquid system)	6795	1087	
Chicken (solid system)	996		
Chicken (liquid system)	14496	1812	

*Emissions factors taken from (Hristov et al. 2011), all other emissions factors taken from (US EPA 2001). US EPA emissions factors were converted from lb / AU to g / AU.

Table 20. Annual average manure generated by each animal type (United States Department of Agriculture 2009)

Animal Type	Annual average manure generated by animal (kg / year / animal)
Beef cattle	8,600
Dairy cattle	46,000
Hog	1,600
Chicken	11

Table 21. Number of animals per EPA animal unit (AU) (US EPA and National Academies of Sciences 2003)

Species Type	Head of Livestock
Beef Cattle	1
Dairy Cattle	0.7
Hog	2.5
Chicken	100

Table 22. Percentage of manure (f_{MMS}) managed in differing manure management systems, separated by animal type, slightly altered to reflect available emission factors (Government of Canada 2022)

Livestock type	Liquid Systems (%)	Solid storage and Drylot (%)	Pasture, Range and Paddock (%)	Other Systems (%)
Beef Cattle	N/A	50	50	N/A
Dairy Cattle	64	18	16	N/A
Hog	97	3	0	0
Broiler/Chicken	7	92	N/A	N/A

3.2.1.2 Food Waste Landfilling Emissions

For food waste landfilling, emissions of CO, NOx, and $PM_{2.5}$ were considered. An emissions factor approach was used (Equation 14). Emissions factors used for food waste landfilling are found in Table 23. To begin, the amount of methane generated by the landfilling of food waste was determined using Equation 5 in 3.1.1.3. That result, given in tonnes of methane generated, was converted from to million m^3 and from million m^3 to million standard cubic feet (mscf). Those values were multiplied by the emissions factors found in Table 23. Those values were then converted from pounds to grams and from grams to tonnes.

$$E_{AP,FW} = m_{CH4,gen}(t) \times EF_{AP,FW}$$

Equation 14

Where:

$E_{AP,FW}$ = Emissions of each type of air pollutant from food waste for each scenario (lbs)

$m_{CH4,gen}(t)$ = Methane generated at landfill (mscf). See Equation 5.

$EF_{AP,FW}$ = Emissions factor for air pollutants (lbs / mscf CH_4)

Table 23. Emissions factors (CO, NOx, $PM_{2.5}$) for food waste landfilling ("WebFire Search | US EPA," n.d.)

Emission Type	Emissions Factor (lb / Million Standard Cubic Feet)
CO	750
NOx	40
PM2.5	17

3.2.2 Biogas Scenario

Similarly to the GHG emissions estimation section, air pollutant emissions were evaluated for four biogas scenarios: 25%, 50%, 75%, and 100%.

3.2.2.1 Transportation Emissions

Air pollutant emissions for transportation were estimated using the methods outlined in 3.1.2.1.

3.2.2.2 Upgrading Emissions

Air pollutant emissions from biogas upgrading were estimated using the methods outlined in 3.1.1.2.

3.2.2.3 Leftover Manure Management Emissions

In the 25%, 50%, and 75% scenarios, there is manure that is not utilized by the biogas system. In those cases, it is assumed that the manure is handled in the conventional way. Methods for estimating air pollutant emissions from conventional manure management can be found in 3.2.1.1.

3.2.2.4 Leftover Food Waste Landfilling Emissions

In the 25% scenario, there is food waste that is not utilized by the biogas system. In that case, it is assumed that the food waste is handled using conventional methods. Methods for estimating air pollutant emissions from food waste landfilling can be found in 3.2.1.2.

3.2.2.5 Energy Production Offset

To determine the potential emissions offset associated with natural gas production, an emissions factor approach was also used, shown in Equation 15, using the emission factors found in Table 24.

$$E_{AP,NG} = \theta \times EF_{AP,NG}$$

Equation 15

Where:

$E_{AP,NG}$: Emissions of SO_2 from natural gas production for each scenario (kg)

θ: Natural gas output in each scenario from **Error! Reference source not found.** (in GJ) converted to million m^3 (using 26.853 m^3/GJ / 10^6)

$EF_{AP,NG}$: Emissions factor for SO_2 (kg / million m^3 processed).

Table 24 shows that the emission factor for SO2 depends on the volume fraction of H_2S in natural gas (referred to as "H_2S percentage" in the table). For the purposes of this analysis, the volume fraction of 3.3 was used.

Table 24. Emissions factor (SO_2) for natural gas production offset (US EPA 1995b)

Emissions Type	Emissions Factor (kg / million m^3 processed)
SO_2	26.98 * H_2S percentage*

*H2S percentage of 3.3 was used, taken from (Kidnay and Parrish 2006).

3.2.2.6 Synthetic Fertilizer Production Offset

To determine the potential emissions offset from synthetic fertilizer production, it is first assumed that all the nitrogen and phosphorus contained in the digestate generated in the model scenario would be used to directly replace synthetic fertilizer. An emissions factor approach was also utilized in this process, shown below in Equation 16.

$$E_{AP.fert} = m_l \times EF_{AP,fert}$$

Equation 16

Where:

$E_{AP,fert}$ = Emissions savings each type of air pollutant from synthetic fertilizer production (tonnes)

m_l = mass of each type of fertilizer replaced (for DAP / MAP, see Equation 17 for P_2O_5 mass, which is used here to represent DAP / MAP mass)

$EF_{ap,\ fert}$ = Emissions factors for air pollutants for each type of fertilizer (Table 25)

Table 25. Emissions factors for synthetic fertilizer production air pollutant emissions ("WebFire Search | US EPA," n.d.)

Emissions Type	MAP (kg / Mg P2O5)	DAP (kg / Mg P2O5)	Urea (lbs / tonne)	Ammonia (lbs / tonne)
SOx	0.040	0.040		0.005
PM	0.34	0.34	3.8	0.144
NH3				
CO				0.136
NOx				5.4
VOC			0.020	0.012

To determine the potential offset, we first had to determine the amount of nitrogen and phosphorus available in the digestate for each scenario (outlined in 3.1.2.6 of GHG methods). Next, the amount of DAP and MAP fertilizer potentially replaced by the phosphorus available in the digestate was determined, again using the same methods as 3.1.2.6 of the GHG methods. These same methods were used again to determine the amount of urea, ammonia, and UAN potentially replaced by the leftover nitrogen in the digestate, after the amounts replaced by MAP and DAP were removed.

To utilize the emissions factors in Table 25, the amount of phosphorus replaced by MAP and DAP had to be converted into Mg P_2O_5. Equation 17, below, outlines this process.

$$m_{P_2O_5} = \left(\mathrm{M}_{P_2O_5} \div M_P\right) \times m_K$$

Equation 17

Where:

$P_2O_5\ molar\ mass$ = 0.283889 kg / mol

$P\ molar\ mass$ = 0.12388 kg / mol

m_K = amount of phosphorus replaced by DAP and MAP in scenario

After converting total P to its mass in P_2O_5, this result can be multiplied by the DAP and MAP emissions factors found in Table 25, as represented in Equation 16.

In order to estimate emissions associated with the production of UAN fertilizer, an implied emissions factor of 4 kg NH_3 / tonne UAN produced was used based on values taken from Statistics Canada and the Government of Canada (Statistics Canada 2016; Canada 2021). An alternative method for estimating UAN emissions, which yielded much larger NH_3 emissions, can be found in Appendix A.

3.2.2.7 Digestate Storage Emissions

To determine emissions associated with the storage of digestate post-digestion, an emissions factor approach was used, illustrated by E19, shown below.

$$E_{NH_3} = m_N \times EF_{NH_3}$$

Where:

E_{NH_3} = Emissions of NH_3 associated with digestate storage (kg)

m_N = Nitrogen available in feedstock (kg)

EF_{NH_3} = Emissions factor for NH_3 emissions from digestate (Table 26)

Table 26. Emissions factors for digestate storage post-digestion (EEA 2019)

Emissions Type	**Digestate Storage (kg NH3 / kg N in feedstock)**
NH_3	0.0266

After determining the amount of nitrogen available in the feedstock, this value can then be multiplied by the value found in Table 26 to find the amount of NH_3 emissions associated with that amount of nitrogen. NH_3 amounts were then converted from kg to tonnes.

Chapter 4

Results and Discussion

The above methodology was utilized to determine both the GHG and air pollutant emissions associated with the potential use of biogas as a source of renewable natural gas in Ontario. The results from the GHG analysis include evaluations of CH_4, N_2O, and CO_2; all of which were converted into their CO_2 equivalents for more appropriate comparison. GHG analysis was undergone for all sections outlined in the methodology, representing all opportunities for emissions in the baseline and biogas scenarios within the system boundary. Air pollutant emissions (CO, $PM_{2.5}$, NOx, SOx, VOCs, NH_3) were also evaluated using the above methodology for the same areas of potential emissions. Results are further contextualized using relevant literature, and results from Canada's National Inventory Reports (NIR) documenting sources and sinks of GHGs.

4.1 GHG Emissions

For both the baseline and biogas scenarios, GHG emissions were estimated to understand the trends across scenarios, Figure 3, shown below, presents the trends in GHG emissions across the baseline and biogas scenarios.

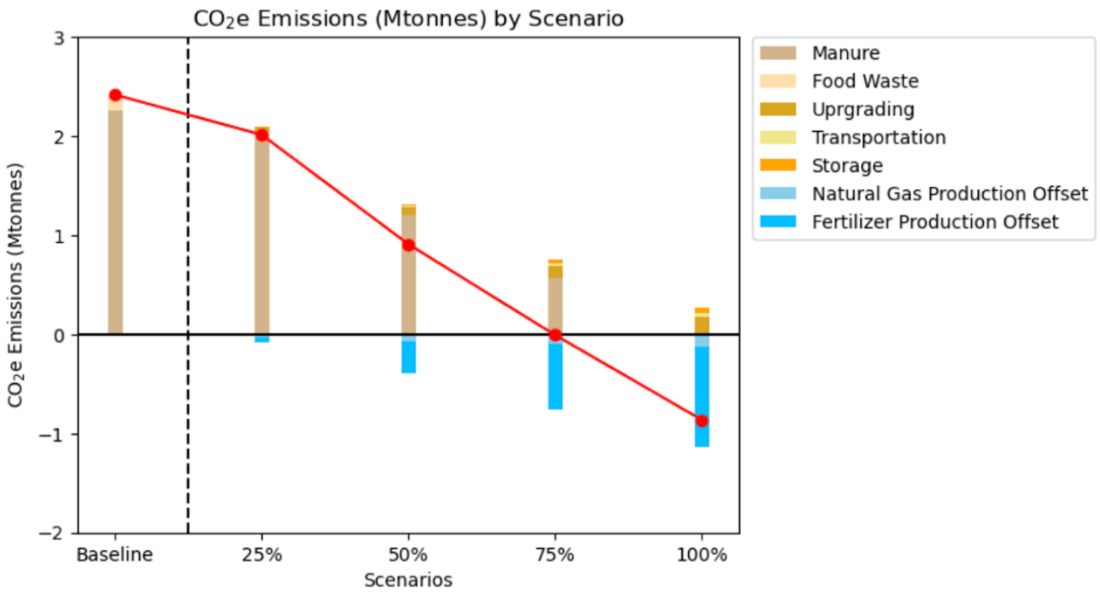


Figure 3. GHG emissions (in Mtonnes of CO_{2e}) across the baseline and 25%, 50%, 75%, and 100% biogas scenarios. Emissions are shown by source (manure management, food waste, biogas upgrading, transportation, manure storage) and offsets (due to avoided production of natural gas and fertilizer). Red markers and lines represent the net GHG emissions, taking into account both potential emissions and potential offsets from natural gas and fertilizer production

In the baseline scenario, emissions from current manure and food waste management are considered. Emissions from manure management in the baseline scenario are estimated to be 2.3 Mtonnes CO_{2e}, which agrees well with the 1.9 Mtonnes CO_{2e} reported by the NIR (Environment and Climate Change Canada 2023). Food waste landfilling in the baseline scenario was estimated to be 0.16 Mtonnes CO_{2e}. The comparable estimate from the 2023 NIR report is for all landfilling in Ontario and is reported to be 5.0 Mtonnes CO_{2e} in 2021 (Environment and Climate Change Canada 2023). Based on the assumption that food waste makes up only a fraction of the total waste disposed of in Ontario landfills, the difference between the model's estimation and the NIR report appears reasonable. By the 50% scenario, all the available food waste is being utilized by the biogas system, and by the 100% scenario, all the available manure is also being utilized by the biogas system. This leads to a significant reduction in emissions between the 75% to the 100% scenario, as the largest

fraction of emissions belongs to the leftover manure being handled by conventional means. Once the biogas system utilizes all available manure, the emissions savings become even more clear.

As Figure 3 moves from the baseline scenario to the biogas scenarios, starting with the 25% scenario, GHG emissions are shown to decrease. By the 75% scenario, the emissions from leftover manure, biogas processes, and transportation are outweighed by the potential emissions savings from natural gas and fertilizer production offset, leading to negligible net emissions from this scenario. In the 100% scenario, emissions have decreased to the point where emissions savings begin, with a net value of -0.8 Mtonnes CO_{2e} emissions. The 100% scenario is also the scenario with the largest upgrading and transportation emissions, due to the larger amount of feedstock being transported and consumed in this scenario. Other studies reported larger emissions from transportations, e.g., Malet (2023) reported that transportation was likely responsible for the emissions associated with the biogas process. The transportation in that study included 30km feedstock trips, as well as digestate spreading trips, which explains the larger transportation emissions reported in that study. In this work, trip distances were optimized and therefore much shorter that what was reported in Malet (2023).

The transportation emissions in the 100% scenario are estimated to be 0.03 Mtonnes CO_{2e}. Metson et al. 2020 reported transportation emissions of 1.2 Mtonnes CO_{2e}. There are several potential reasons for which their higher value is expected. Their work used a Life Cycle Assessment approach, adopting a standard database value for diesel emissions, reflecting life-cycle stages beyond on-road emission. Conversely, this work is based on on-road emission factors developed using the Motor Vehicle Emission Simulator (MOVES) model for Ontario (Wang et al. 2020), which is the same model used to develop the NIR. Another factor lowering emissions in this work was the use of a spatial techno-economic optimization approach for biogas siting. This means that trips are optimized. For example, trip distances in Metson (2020) were 112 km. In this work, trip distances are much lower (median distances of 9 km for most feedstocks, as communicated by Ushnik Mukherjee). Finally, their analysis included both transportation of the feedstock from the source to the biogas location as well as digestate movement from the biogas location to the application location. This work did not include any movement of excess digestate. These additional routes, plus other factors mentioned, likely explain the larger emissions reported in Metson et al. 2020.

Similarly, for biogas upgrading, emissions were estimated to be 0.18 Mtonnes CO_{2e}, while Metson et al. 2022 reports 3.7 Mtonnes CO_{2e} emissions from what they refer to as “other processes' ', which included biogas upgrading to natural gas. This category refers to the production and distribution of biogas, as well as upgrading. Because of these additional categories included with biogas upgrading, the larger value from their report is within expectations.

4.2 Air Pollutant Emissions

For both the baseline and biogas scenarios, air pollutant emissions were estimated to understand the changes across scenarios. Figure 4, shown below, illustrates the sources of emissions in the baseline scenario. Baseline manure management emissions of NH_3 are reported as 32 kilotonnes. The Canadian national Air Pollutant Emission Inventory (APEI) reported value for NH_3 emissions from animal agriculture was 64 kilotonnes in Ontario in 2022, meaning that our baseline estimate underestimates the amount of NH_3 being produced by animal agriculture (Government of Canada 2022a). This could be explained by the difference in methods between the APEI and our baseline manure management assumptions, specifically that we are calculating NH_3 emissions from only conventional manure management, whereas the APEI does not report their numbers from manure management alone, but from animal agriculture as a whole. Our baseline values for $PM_{2.5}$ and VOCs follow a similar trend with the reported APEI values for 2022, with our model estimating 2.85 kilotonnes of $PM_{2.5}$ and 3.6kilotonnes VOCs and the APEI reporting 23 kilotonnes of $PM_{2.5}$ and 30 kilotonnes of VOCs (Government of Canada 2022a). Overall, these values indicate that our results do not indicate the full impact of animal agriculture but are focused instead on the impact from manure management.

Figure 4 also presents the air pollutant emissions associated with food waste, but they are dwarfed by the emissions associated with manure management. Our model indicates that in the baseline scenario, there are 107 tonnes of CO, 5.7 tonnes of NOx, and 2.4 tonnes of $PM_{2.5}$ associated with food waste landfilling. The APEI reports 2.7 kilotonnes of CO, 318 tonnes of NOx, and 202 tonnes of $PM_{2.5}$ in Ontario in 2022 associated with landfilling (Government of Canada 2022a). These values are for all landfilling, not for food waste specifically, so the larger values are not unexpected. The emissions do follow a similar trend to those reported in the APEI, with the largest emissions

being CO, followed by NOx and then $PM_{2.5}$, lending some confidence in our estimations of food waste landfilling emissions.

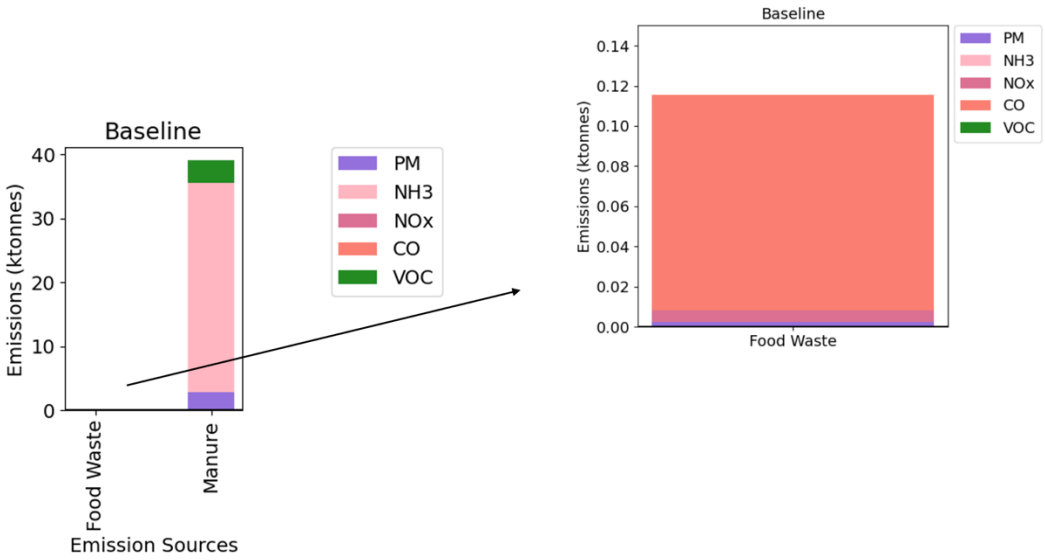


Figure 4. Air pollutant emissions from manure management and food waste landfilling in the baseline scenario.

Figure 5 depicts the emissions from various sources under a 25% biogas scenario. From this figure, it can be seen that emissions from conventional management of manure not utilized by the biogas scenario is responsible for the largest share of emissions. In the figure, those emissions are given in Mtonnes of emissions as opposed to tonnes like the other emissions sources. In the 25% scenario, net emissions were estimated to be 28 kilotonnes, of which manure management is responsible for a sizable portion. Without taking the emissions of leftover manure and food waste into account, and just solving for the net emissions associated with the biogas process itself, net emissions come out to only 42 tonnes, or 0.042 kilotonnes (Table 27). The largest fraction of emissions associated with the biogas process was upgrading, followed by transportation. A significant offset

can already be seen in synthetic fertilizer production, despite only 25% of the available feedstock being converted to digestate capable of potentially replacing fertilizer.

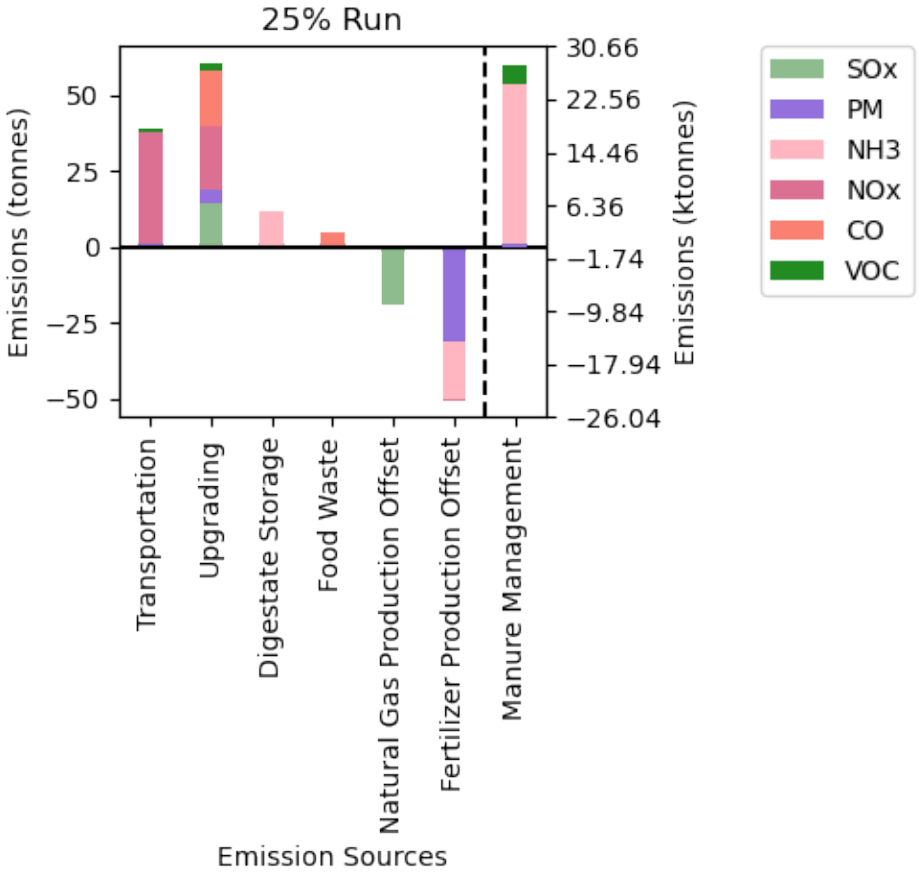


Figure 5. Air pollutant emissions associated with a 25% biogas scenario. Emissions from transportation, upgrading, digestate storage, food waste landfilling, natural gas production offset, and synthetic fertilizer production offset are reported in tonnes, whereas emissions from conventional manure management are reported in kilotonnes

Table 27. Net air pollutant emissions associated with the 25%, 50%, 75% and 100% scenarios.

	Net Emissions (kilotonnes)
25% Run	27.9
50% Run	16.4
75% Run	9.6
100% Run	-0.5

As we move from the 25% scenario to the 50% and 75% scenario, those significant synthetic fertilizer production offsets become clearer (Figure 6, Figure 7). If not for the impact of conventional manure management, the offsets from replacing the production of synthetic fertilizer would completely offset any potential emissions from the biogas scenario by the 50% biogas scenario. As it stands, with the inclusion of emissions from leftover manure being managed using conventional means, the total net emissions for the 50 and 75% scenarios are 16 kilotonnes and 9.5 kilotonnes, respectively (Table 27). There are significant decreases in emissions between the 25, 50, and 75% scenarios, due in part to both the increase in manure being utilized by the biogas system and an increase in the potential digestate available to replaced synthetic fertilizer. The significant emissions associated with manure management, as shown in Figure 5, Figure 6, and Figure 7, highlight the importance of implementing new strategies to deal with animal waste. Other studies corroborate these findings, including one that investigated the potential mortalities associated with the entire food production system in the United States. They found that the largest share of mortalities were associated with NH_3 emissions from livestock waste, further highlighting the importance of an improved waste management strategy (Domingo et al. 2021). By the 50% scenario, all available food waste is being used by the biogas system, and by the 100% scenario, all available manure is also being utilized, impacting the emissions significantly.

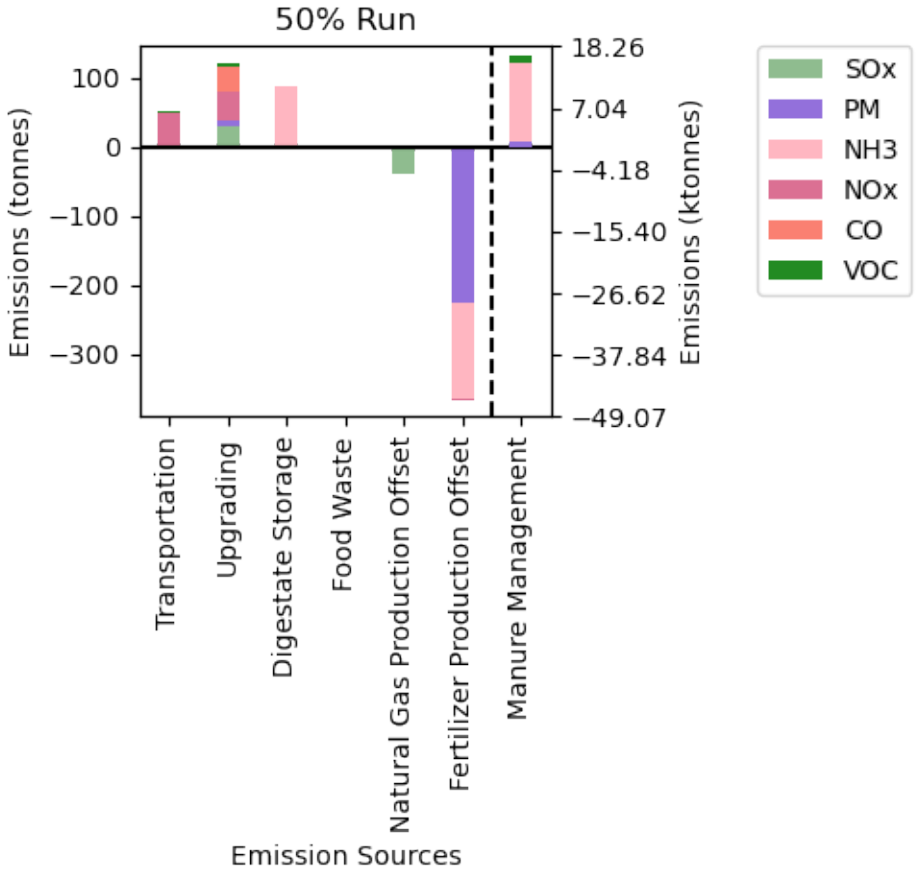


Figure 6. Air pollutant emissions associated with a 50% biogas scenario. Emissions from transportation, upgrading, digestate storage, food waste landfilling, natural gas production offset, and synthetic fertilizer production offset are reported in tonnes, whereas emissions from conventional manure management are reported in kilotonnes

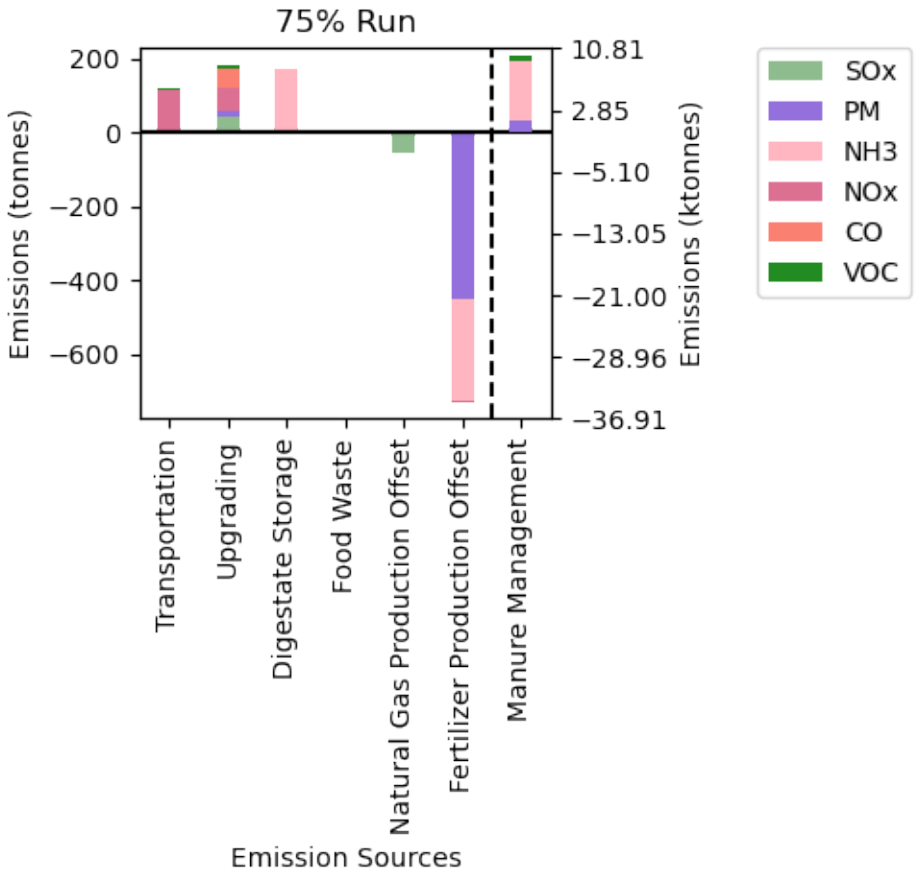


Figure 7. Air pollutant emissions associated with a 75% biogas scenario. Emissions from transportation, upgrading, digestate storage, food waste landfilling, natural gas production offset, and synthetic fertilizer production offset are reported in tonnes, whereas emissions from conventional manure management are reported in kilotonnes

Figure 8 illustrates the emissions in the 100% biogas scenario, where all available food waste and manure is being utilized by the biogas system as opposed to traditional manure management systems. Because of the increase in feedstock being transported to and utilized at the digestion sites, we do see increases in emissions associated with transportation, upgrading, and digestate storage. However, these emissions are completely offset by the potential emissions savings from the avoided production of synthetic fertilizer. Without the large emissions from leftover manure seen in the previous scenarios, the 100% scenario's net emissions were estimated to be -0.0510

kilotonnes (Table 27). In this scenario, we estimated 1,099 tonnes of emissions offsets related to synthetic fertilizer offset. While some of these emissions savings will be seen out of province, such as production of MAP and DAP, some emissions savings would be seen in Ontario, particularly emissions of NH_3 associated with UAN production (Statistics Canada 2016). In 2016, 157 tonnes of NH_3 associated with nitrogen-containing fertilizer production was reported in Ontario (Government of Canada 2016). In the 100% scenario, there are an estimated 410 tonnes of NH_3 emissions offset generated from UAN production, meaning that use of digestate could potentially offset the emissions associated with UAN production in Canada. According to Statistics Canada, 1.4 kt of UAN was produced in Canada in 2016, while our model suggests that we would be able to potentially replace 120 kt of UAN based on the amount of nitrogen and phosphorus that is present in the digestate (Statistics Canada 2016). The discrepancy between the emissions reported and the tonnes of UAN reported is likely due to the fact that the facility that produces UAN in Ontario also generates other fertilizer products that were not evaluated in this study.

The air pollution emission reductions found in this report differ from some previous reports in the literature of degraded air quality associated with biogas production, including increases in NOx emissions from biogas combustion (Carreras-Sospedra, Williams, and Dabdub 2016; Battini et al. 2014). Here, we also find an increase in NOx emissions due to transportation and the upgrading process, although emissions of pollutants decrease overall. We do not include emissions from biogas combustion. Instead, we consider upgraded biogas to be a substitute for fossil natural gas, which will have similar emissions. We consider any effect on combustion emissions to be minimal, given that the 100% biogas scenario could replace less than 3% of Ontario's current natural gas demand. Ultimately, the effect of increases in NOx and decreases in $PM_{2.5}$ and ammonia on ambient air quality and human health should be evaluated as a subject of future work.

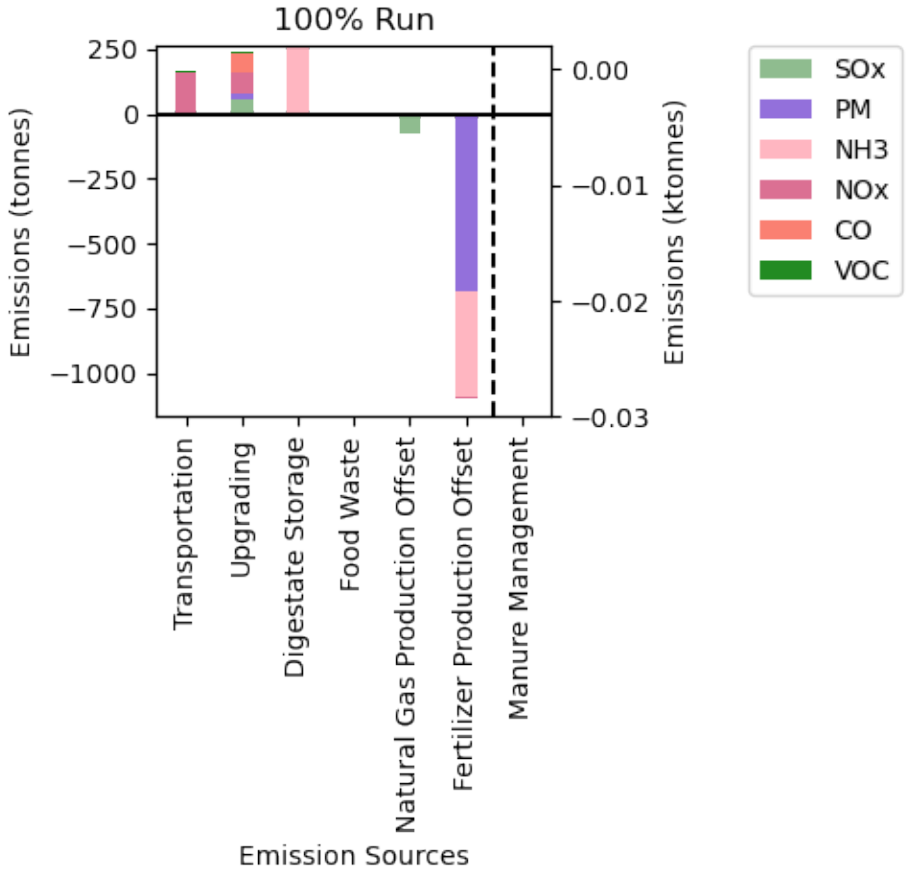
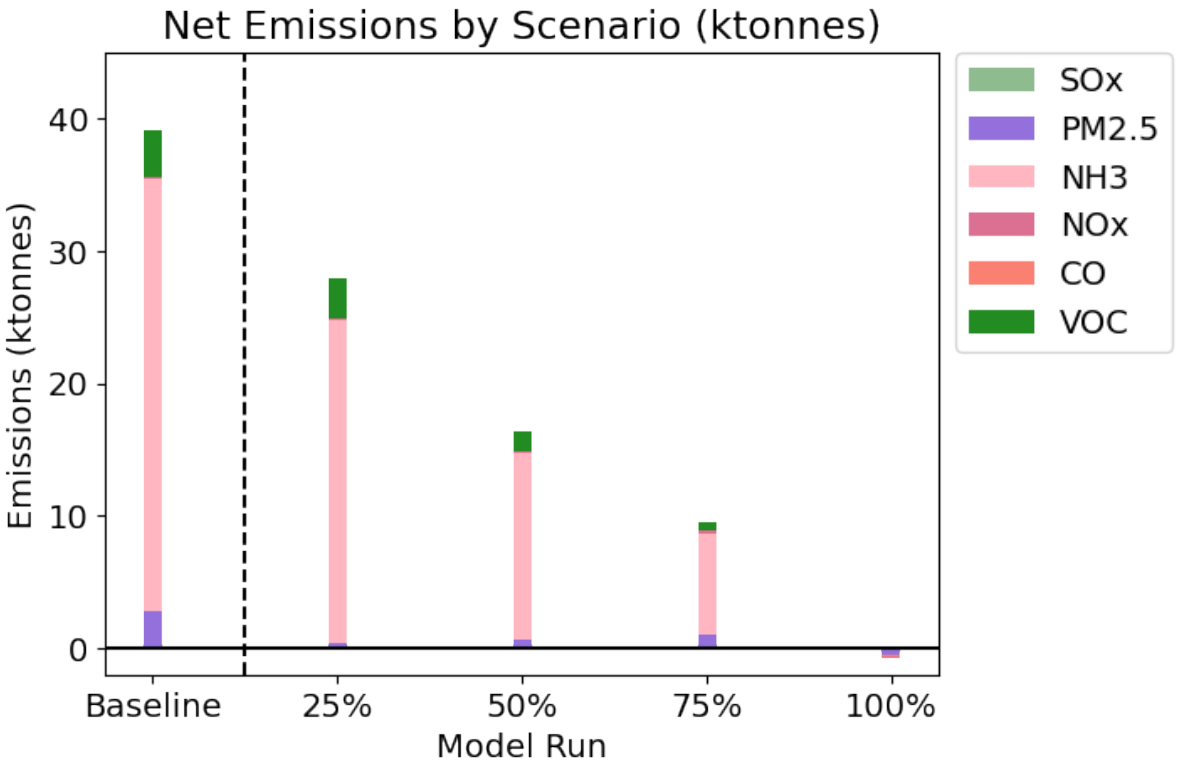


Figure 8. Air pollutant emissions associated with a 100% biogas scenario. Emissions from transportation, upgrading, digestate storage, food waste landfilling, natural gas production offset, and synthetic fertilizer production offset are reported in tonnes, whereas emissions from conventional manure management are reported in kilotonnes

We see drastic reductions in NH_3 as we move from the baseline into the biogas scenarions (Figure 9). Figure 9 shows us the net emissions by scenario, with each evaluated pollutant represented. The largest reductions are NH_3, which is the major pollutant associated with conventional manure management. As more manure is incorporated into the biogas system, as opposed to being managed using conventional means, we see more reductions in NH_3 emissions, along with emissions of VOCs, although not on the same scale as NH_3. As stated previously, by the 100% scenario, when all of the available manure is now being incorporated into the biogas system, we see no positive emissions, and

net negative emissions of NH_3 and $PM_{2.5}$, meaning that any emissions associated with the biogas process are outweighed by the potential emissions savings from synthetic fertilizer production offset and natural gas production offset by the 100% scenario.

Figure 9. Net air pollutant emissions by scenario



Chapter 5

Conclusion

Many studies agree that the use of biogas as a strategy to deal with issues surrounding current waste management strategies has the potential to reduce GHG emissions, but there are no studies that look specifically at the potential co-benefits related to reductions in air pollutant emissions. Additionally, this study reflects the most comprehensive emissions estimation associated with the adoption biogas within a Canadian context, considering all processes associated with the creation of renewable natural gas via anaerobic digestion, including transportation of the feedstock, the process of upgrading the biogas into biomethane, and the storage of the digestate post-digestion. The results of this study offer insights into the potential consequences of adopting biogas on a provincial scale in Ontario and highlight the issues associated with current waste management practices, specifically manure management.

The results of this study highlight the significant impact that current manure management practices have on emissions. When looking at both GHGs and air pollutants, the most significant source of emissions were associated with current manure management practices. Once all the available manure was utilized in the biogas scenario, the emissions reductions were even more significant, especially when considering air pollutant emissions. This suggests that while biogas is touted as a GHG mitigation strategy, there are significant air pollution co-benefits when it comes to biogas use.

There are several limitations associated with this study, including the uncertainty associated with the use of emissions factors. The emissions factors used in this study were highly rated via the EPA emissions factor rating system, but there are potential uncertainties inherently involved with the use of emissions factors (US EPA 1995b; Frey 2007). The letter rating given by the EPA (A-E) is based on the quality of the test data and the ability of the factor to represent national average emissions for that category. As emissions can vary with process details that may vary or be unknown, the emissions factors used in this study could be either under or over emphasizing the impacts from those sources (Frey 2007). The ranges in potential emission factors were included within the Python code in the Appendix. Additionally, there is some discussion about how best to go about accounting

for emissions of CO_2. In reference to the emissions associated with biogas upgrading, the EPA suggests that any carbon emissions would be considered biogenic and therefore were not included in the analysis of GHG emissions for this study (EPA 2016). Inclusion of those carbon emissions would certainly increase the GHG emissions associated with the biogas upgrading process. Additionally, we are not accounting for the cooling effect of aerosol precursors (NH_3, SOx, VOCs, NOx, CH_4, $PM_{2.5}$) as referenced in Table 1, when accounting for the potential climate impact of GHG emissions from the baseline and biogas scenarios. While Hijazi et al. (2016) and Vasco-Correa et al. (2018) do mention that leakages from the biogas production site could be a potential source of emissions, these emissions are not included in this analysis. Potential leakage from digestate storage, as referenced in Vasco-Correa et al. (2018), is captured by the emissions estimation associated with digestate storage in the methods and results section.

While this study does evaluate the emissions of GHGs and air pollutants associated with the use of biogas in Ontario, there are several suggestions for how to improve this research going forward. First, an analysis of emissions associated with digestate application would bolster the argument that the use of digestate could be an improvement over the use of synthetic fertilizer. This study suggests significant emissions savings are associated with avoiding synthetic fertilizer production, but there is also the potential for reductions in emissions associated with synthetic fertilizer application. An investigation of the difference in emissions associated with digestate use vs. synthetic fertilizer use would help to evaluate the argument that digestate is a better alternative. Additionally, an estimate of the economic value of these GHG and air pollutant emissions reductions would be helpful when discussing this waste management solution with policymakers.

References

Appels, Lise, Joost Lauwers, Jan Degrève, Lieve Helsen, Bart Lievens, Kris Willems, Jan Van Impe, and Raf Dewil. 2011. "Anaerobic Digestion in Global Bio-Energy Production: Potential and Research Challenges." Renewable and Sustainable Energy Reviews 15 (9): 4295–4301. https://doi.org/10.1016/j.rser.2011.07.121.

Ashtari, Ahmad Kalbasi, Amir M. Samani Majd, Gerald L. Riskowski, Saqib Mukhtar, and Lingying Zhao. 2016. "Removing Ammonia from Air with a Constant pH, Slightly Acidic Water Spray Wet Scrubber Using Recycled Scrubbing Solution." Frontiers of Environmental Science & Engineering 10 (6): 3. https://doi.org/10.1007/s11783-016-0869-3.

Astill, Gregory M., and C. Richard Shumway. 2016. "Profits from Pollutants: Economic Feasibility of Integrated Anaerobic Digester and Nutrient Management Systems." Journal of Environmental Management 184 (December):353–62. https://doi.org/10.1016/j.jenvman.2016.10.012.

Battini, F., A. Agostini, A. K. Boulamanti, J. Giuntoli, and S. Amaducci. 2014. "Mitigating the Environmental Impacts of Milk Production via Anaerobic Digestion of Manure: Case Study of a Dairy Farm in the Po Valley." Science of The Total Environment 481 (May):196–208. https://doi.org/10.1016/j.scitotenv.2014.02.038.

Batzias, F. A., D. K. Sidiras, and E. K. Spyrou. 2005. "Evaluating Livestock Manures for Biogas Production: A GIS Based Method." Renewable Energy 30 (8): 1161–76. https://doi.org/10.1016/j.renene.2004.10.001.

Bentrup, Frank. 2018. "Updated Carbon Footprint Values for Mineral Fertilizer from Different World Regions." In *Global Food Challenges towards Sustainable Consumption and Production*.

Bidart, Christian, Magnus Fröhling, and Frank Schultmann. 2014. "Livestock Manure and Crop Residue for Energy Generation: Macro-Assessment at a National Scale." Renewable and Sustainable Energy Reviews 38 (October):537–50. https://doi.org/10.1016/j.rser.2014.06.005.

Bist, Ramesh Bahadur, Sachin Subedi, Lilong Chai, and Xiao Yang. 2023. "Ammonia Emissions, Impacts, and Mitigation Strategies for Poultry Production: A Critical Review." Journal of Environmental Management 328 (February):116919. https://doi.org/10.1016/j.jenvman.2022.116919.

Canadian Biogas Association. 2019. "Biogas and Renewable Natural Gas in Ontario: 2019 Market Overview and Outlook." https://www.biogasassociation.ca/images/uploads/documents/2019/2019-Market-Overview.pdf.

Cuéllar, Amanda D., and Michael E. Webber. 2008. "Cow Power: The Energy and Emissions Benefits of Converting Manure to Biogas." Environmental Research Letters 3 (3): 034002. https://doi.org/10.1088/1748-9326/3/3/034002.

Dagnall, Steve, Jon Hill, and David Pegg. 2000. "Resource Mapping and Analysis of Farm Livestock Manures—Assessing the Opportunities for Biomass-to-Energy Schemes." Bioresource Technology 71 (3): 225–34. https://doi.org/10.1016/S0960-8524(99)00076-0.

Domingo, Nina G. G., Srinidhi Balasubramanian, Sumil K. Thakrar, Michael A. Clark, Peter J. Adams, Julian D. Marshall, Nicholas Z. Muller, et al. 2021. "Air Quality–Related Health Damages of Food." Proceedings of the National Academy of Sciences 118 (20). https://doi.org/10.1073/pnas.2013637118.

Ebner, Jacqueline H., Rodrigo A. Labatut, Matthew J. Rankin, Jennifer L. Pronto, Curt A. Gooch, Anahita A. Williamson, and Thomas A. Trabold. 2015. "Lifecycle Greenhouse Gas Analysis of an Anaerobic Codigestion Facility Processing Dairy Manure and Industrial Food Waste." Environmental Science & Technology 49 (18): 11199–208. https://doi.org/10.1021/acs.est.5b01331.

EEA. 2019. “Biological Treatment of Waste - Anaerobic Digestion at Biogas Facilities.”
Environment and Climate Change Canada. 2023. “National Inventory Report 1990-2021: Greenhouse Gas Sources and Sinks in Canada.” Environment and Climate Change Canada.
EPA. 2009. “Technical Support Document for Manure Management Systems: Proposed Rule for Mandatory Reporting of Greenhouse Gases.” https://www.epa.gov/sites/default/files/2015-07/documents/tsd_manure_020409.pdf.
———. 2016. “Evaluating the Air Quality, Climate & Economic Impacts of Biogas Management Technologies.”
———. 2023. “DRAFT Inventory of U.S. Greenhouse Gas Emissions and Sink 1990-2021.” https://www.epa.gov/ghgemissions/draft-inventory-us-greenhouse-gas-emissions-and-sinks-1990-2021.
Escalante, Humberto, Liliana Castro, Paola Gauthier-Maradei, and Reynel Rodríguez De La Vega. 2016. “Spatial Decision Support System to Evaluate Crop Residue Energy Potential by Anaerobic Digestion.” Bioresource Technology 219 (November):80–90. https://doi.org/10.1016/j.biortech.2016.06.136.
Food and Agriculture Organization of the United Nations. 2020. “FAOSTAT.” https://www.fao.org/faostat/en/#data/GT.
Frey, H Christopher. 2007. “Quantification of Uncertainty in Emission Factors and Inventories.” https://www3.epa.gov/ttnchie1/conference/ei16/session5/frey.pdf.
Gavrilova, Olga, Adrian Leip, Hongmin Dong, James Macdonald, Douglas, Carlos Alfredo Gomez, Bravo, Barbara Amon, Rolando Rosales, Barahona, et al. 2019. “Chapter 10: Emissions from Livestock and Manure Management - 2019 Refinement to the 2006 IPCC Guidelines for National Greenhouse Gas Inventories.”
Government of Canada. 2016. “National Pollutant Release Inventory Dashboard.” https://www.canada.ca/en/environment-climate-change/services/national-pollutant-release-inventory/tools-resources-data/all-year-dashboard.html.
———. 2020. “Canada’s Air Pollutant Emissions Inventory - Open Government Portal.” 2020. https://open.canada.ca/data/en/dataset/fa1c88a8-bf78-4fcb-9c1e-2a5534b92131.
———. 2022a. “Canada’s Official Greenhouse Gas Inventory - Environment and Climate Change Canada Data.” April 14, 2022. https://data.ec.gc.ca/data/substances/monitor/canada-s-official-greenhouse-gas-inventory/D-Emission-Factors/?lang=en.
———. 2022a. “Canada’s Air Pollutant Emissions Inventory—Environment and Climate Change Canada Data: 2020.” https://data-donnees.ec.gc.ca/data/substances/monitor/canada-s-air-pollutant-emissions-inventory/APEI_Tables_Canada_Provinces_Territories/?lang=en.
Government of Canada, Public Services and Procurement Canada. 2022b. “Canada’s Air Pollutant Emissions Inventory Report : Government of Canada Publications.” 2022. https://publications.gc.ca/site/eng/9.869731/publication.html.
Guo, Li, and Weiwei Chen. 2022. “Atmospheric Pollution of Agriculture-Dominated Cities.” Atmosphere 13 (6): 900. https://doi.org/10.3390/atmos13060900.
Hadlocon, Lara Jane, Roderick Manuzon, and Lingying Zhao. 2014. “Optimization of Ammonia Absorption Using Acid Spray Wet Scrubbers.” Transactions of the ASABE (American Society of Agricultural and Biological Engineers) 57 (February):647–59. https://doi.org/10.13031/trans.57.10481.
Hiraishi, Taka, Buruhani Nyenzi, José Domingos Miguez, A D Bhide, and Riitta Pipatti. 2018. “IPCC Good Practice Guidance and Uncertainty Management in National Greenhouse Gas Inventories, Chapter 5: Waste.”

Höhn, J., E. Lehtonen, S. Rasi, and J. Rintala. 2014. “A Geographical Information System (GIS) Based Methodology for Determination of Potential Biomasses and Sites for Biogas Plants in Southern Finland.” Applied Energy 113 (January):1–10. https://doi.org/10.1016/j.apenergy.2013.07.005.
Huang, Xinyuan, Vivek Srikrishnan, Jonathan Lamontagne, Klaus Keller, and Wei Peng. 2023. “Effects of Global Climate Mitigation on Regional Air Quality and Health.” Nature Sustainability, May, 1–13. https://doi.org/10.1038/s41893-023-01133-5.
Karlsson, Mikael, Eva Alfredsson, and Nils Westling. 2020. “Climate Policy Co-Benefits: A Review.” Climate Policy 20 (3): 292–316. https://doi.org/10.1080/14693062.2020.1724070.
Kidnay, A.J., and W.R. Parrish. 2006. Fundamentals of Natural Gas Processing. CRC Press.
Klinglmair, Manfred, Carl Vadenbo, Thomas Fruergaard Astrup, and Charlotte Scheutz. 2017. “An MFA-Based Optimization Model for Increased Resource Efficiency: Phosphorus Flows in Denmark.” Resources, Conservation and Recycling 122 (July):1–10. https://doi.org/10.1016/j.resconrec.2017.01.012.
Kuo, Jeff, and Jason Dow. 2017. “Biogas Production from Anaerobic Digestion of Food Waste and Relevant Air Quality Implications.” Journal of the Air & Waste Management Association 67 (9): 1000–1011. https://doi.org/10.1080/10962247.2017.1316326.
Lelieveld, J., J. S. Evans, M. Fnais, D. Giannadaki, and A. Pozzer. 2015. “The Contribution of Outdoor Air Pollution Sources to Premature Mortality on a Global Scale.” Nature 525 (7569): 367–71. https://doi.org/10.1038/nature15371.
Ma, Jianguo, Norman R. Scott, Stephen D. DeGloria, and Arthur J. Lembo. 2005. “Siting Analysis of Farm-Based Centralized Anaerobic Digester Systems for Distributed Generation Using GIS.” Biomass and Bioenergy 28 (6): 591–600. https://doi.org/10.1016/j.biombioe.2004.12.003.
Malet, Nicolas, Sylvain Pellerin, Romain Girault, and Thomas Nesme. 2023. “Does Anaerobic Digestion Really Help to Reduce Greenhouse Gas Emissions? A Nuanced Case Study Based on 30 Cogeneration Plants in France.” Journal of Cleaner Production 384 (January):135578. https://doi.org/10.1016/j.jclepro.2022.135578.
Metson, Geneviève S., Roozbeh Feiz, Ida Lindegaard, Tove Ranggård, Nils-Hassan Quttineh, and Emma Gunnarsson. 2022. “Not All Sites Are Created Equal – Exploring the Impact of Constraints to Suitable Biogas Plant Locations in Sweden.” Journal of Cleaner Production 349 (May):131390. https://doi.org/10.1016/j.jclepro.2022.131390.
Metson, Geneviève S., Roozbeh Feiz, Nils-Hassan Quttineh, and Karin Tonderski. 2020. “Optimizing Transport to Maximize Nutrient Recycling and Green Energy Recovery.” Resources, Conservation & Recycling: X 9–10 (June):100049. https://doi.org/10.1016/j.rcrx.2021.100049.
Metson, Geneviève S., Anton Sundblad, Roozbeh Feiz, Nils-Hassan Quttineh, and Steve Mohr. 2022. “Swedish Food System Transformations: Rethinking Biogas Transport Logistics to Adapt to Localized Agriculture.” Sustainable Production and Consumption 29 (January):370–86. https://doi.org/10.1016/j.spc.2021.10.019.
Nasir, Filzah. 2018. “Integration of Environmental Costs in Ontario’s Pavement Management Systems.” Waterloo, Ontario: University of Waterloo. http://hdl.handle.net/10012/13709.
Paolini, Valerio, Francesco Petracchini, Marco Segreto, Laura Tomassetti, Nour Naja, and Angelo Cecinato. 2018. “Environmental Impact of Biogas: A Short Review of Current Knowledge.” Journal of Environmental Science and Health, Part A 53 (10): 899–906. https://doi.org/10.1080/10934529.2018.1459076.
Pierie, F., R. M. J. Benders, J. Bekkering, W. J. Th. van Gemert, and H. C. Moll. 2016. “Lessons from Spatial and Environmental Assessment of Energy Potentials for Anaerobic Digestion Production Systems

Applied to the Netherlands." Applied Energy 176 (August):233–44. https://doi.org/10.1016/j.apenergy.2016.05.055.
Preble, Chelsea V., Sharon S. Chen, Toshifumi Hotchi, Michael D. Sohn, Randy L. Maddalena, Marion L. Russell, Nancy J. Brown, Corinne D. Scown, and Thomas W. Kirchstetter. 2020. "Air Pollutant Emission Rates for Dry Anaerobic Digestion and Composting of Organic Municipal Solid Waste." Environmental Science & Technology 54 (24): 16097–107. https://doi.org/10.1021/acs.est.0c03953.
Sampat, Apoorva M., Gerardo J. Ruiz-Mercado, and Victor M. Zavala. 2018. "Economic and Environmental Analysis for Advancing Sustainable Management of Livestock Waste: A Wisconsin Case Study." ACS Sustainable Chemistry & Engineering 6 (5): 6018–31. https://doi.org/10.1021/acssuschemeng.7b04657.
Scarlat, Nicolae, Fernando Fahl, Jean-François Dallemand, Fabio Monforti, and Vicenzo Motola. 2018. "A Spatial Analysis of Biogas Potential from Manure in Europe." Renewable and Sustainable Energy Reviews 94 (October):915–30. https://doi.org/10.1016/j.rser.2018.06.035.
Shindell, Drew. 2015. "The Social Cost of Atmospheric Release." Climatic Change 130 (2): 313–26. https://doi.org/10.1007/s10584-015-1343-0.
Smit, Lidwien A. M., and Dick Heederik. 2017. "Impacts of Intensive Livestock Production on Human Health in Densely Populated Regions." GeoHealth 1 (7): 272–77. https://doi.org/10.1002/2017GH000103.
Spiegal, Sheri, Peter J. A. Kleinman, Dinku M. Endale, Ray B. Bryant, Curtis Dell, Sarah Goslee, Robert J. Meinen, et al. 2020. "Manuresheds: Advancing Nutrient Recycling in US Agriculture." Agricultural Systems 182 (June):102813. https://doi.org/10.1016/j.agsy.2020.102813.
Statistics Canada. 2016. "Canadian Fertilizer Production, by Product Type and Fertilizer Year, Cumulative Data."
Thompson, Ethan, Qingbin Wang, and Minghao Li. 2013. "Anaerobic Digester Systems (ADS) for Multiple Dairy Farms: A GIS Analysis for Optimal Site Selection." Energy Policy 61 (October):114–24. https://doi.org/10.1016/j.enpol.2013.06.035.
United States Department of Agriculture. 2009. "Chapter 4 - Agricultural Waste Characteristics - Agricultural Waste Management Field Handbook."
US EPA. 1995a. "Ammonium Nitrate."
———. 1995b. "Emissions Factors - Introduction."
———. 1995c. "Natural Gas Processing."
Valenti, Francesca, Simona M. C. Porto, Bruce E. Dale, and Wei Liao. 2018. "Spatial Analysis of Feedstock Supply and Logistics to Establish Regional Biogas Power Generation: A Case Study in the Region of Sicily." Renewable and Sustainable Energy Reviews 97 (December):50–63. https://doi.org/10.1016/j.rser.2018.08.022.
Vaneeckhaute, Céline, David Styles, Thomas Prade, Paul Adams, Gunnar Thelin, Lena Rodhe, Inga Gunnarsson, and Tina D'Hertefeldt. 2018. "Closing Nutrient Loops through Decentralized Anaerobic Digestion of Organic Residues in Agricultural Regions: A Multi-Dimensional Sustainability Assessment." Resources, Conservation and Recycling 136 (September):110–17. https://doi.org/10.1016/j.resconrec.2018.03.027.
Vasco-Correa, Juliana, Sami Khanal, Ashish Manandhar, and Ajay Shah. 2018. "Anaerobic Digestion for Bioenergy Production: Global Status, Environmental and Techno-Economic Implications, and Government Policies." Bioresource Technology 247 (January):1015–26. https://doi.org/10.1016/j.biortech.2017.09.004.
Wang, Wilson, Rebecca K. Saari, Chris Bachmann, and Ushnik Mukherjee. 2020. "Estimating Transboundary Economic Damages from Climate Change and Air Pollution for Subnational Incentives for Green On-

Road Freight." Transportation Research Part D: Transport and Environment 82 (May):102325. https://doi.org/10.1016/j.trd.2020.102325.
"WebFire Search | US EPA." n.d. Accessed July 22, 2024. https://cfpub.epa.gov/webfire/SearchEmissionFactor/searchpage.cfm.

Appendix A

This appendix shows example calculations for Chapter 3.

Methane from Manure Management

The equation below, taken from Equation 2, was used to determine the CH_4 emissions per kg of manure for each management system within a manure type. For example, for beef cattle manure, the equation was performed for liquid management systems, solid storage, pasture, and other management systems. Each time, the f value and MCF value was changed to reflect the percentage of manure managed under that management system in Canada and the differing methane conversion factors.

$$Beef\ Liquid\ CH_4\ emissions\ per\ kg\ of\ manure\ =\ 0.06545 \times 0.247 \times 0.657 \times 0.2 \times 0.053$$
$$=\ 0.0001$$

Equation 18

Where:

VS= 0.06545 (Table 4)

B_0= 0.247 (Table 4)

ρ= 0.657

$\boldsymbol{MCF}$= 0.2 (Error! Reference source not found.)

f= 0.053 (Table 3)

The results of Equation 18 produce CH_4 emissions per kg of manure.

$$Beef\ Liquid\ CH_4\ emissions\ (kg)\ =\ (Manure\ (tonnes) \times 1000kg) \times \frac{0.0001\ kg\ CH_4}{kg\ manure}$$
$$=\ 941{,}193.73\ kg$$

Equation 19

The result of Equation 19 produces CH_4 emissions from beef liquid systems in kg, where manure (tonnes) is estimated to be 8,359,902 tonnes (Table 2).

$$Beef\ Liquid\ CH_4\ emissions\ (tonnes)\ =\ 941193.73\ kg\ \div \left(\frac{tonne}{1000kg}\right)\ =\ 941.19\ tonnes$$

Equation 20

The result of Equation 20 gives the result of tonnes of CH_4 emissions for the amount of manure generated. This same procedure is repeated for each type of manure management system beef manure undergoes. These values are then added together.

$$Beef\ total\ CH_4\ emissions\ (tonnes)\ =\ BL\ +\ BS\ +\ BP\ +\ BO\ =\ 2{,}177.18\ tonnes$$

Equation 21

Where:

BL: Beef Liquid CH_4 emissions

BS: Beef Solid CH_4 emissions

BP: Beef Pasture CH_4 emissions

BO: Beef Other CH_4 emissions

This entire process is repeated for each manure type. The values for each manure type are then added together to produce total CH_4 emissions for manure management.

$$Total\ manure\ CH_4\ emissions\ (tonnes)\ =\ Beef\ +\ Dairy\ +\ Hog\ +\ Broiler\ =\ 113{,}960.30$$

Equation 22

The value from Equation 22 is then multiplied by 25 to convert to CO_{2e} based on the Global Warming Potential of Methane, and converted into Mtonnes, after being combined with the emissions from direct and indirect N_2O emissions.

Food Waste Landfilling

Equation 23, below, shows the 44% recovery scenario.

$$CH_4\ emitted\ (tonnes)\ =\ (12599.91\ -\ 5543.96)(1-0.1)+\left(5543.96(1-0.997)\right)$$

Equation 23

Equation 24 below shows what Equation 5 looks like with all parameters included.

$$CH_4\ generated\ (tonnes)\ =\ (3{,}090{,}878)(1)(0.06)(0.15)\left(\frac{(1-e^{-0.06})}{0.06}\right)\left(\frac{16}{12}\right)(e^{-0.06\times0})$$

Equation 24

The result of Equation 24 shows CH_4 generated at the landfill for the model year in tonnes. This value was then multiplied by 0.44 and 0.66 respectively to indicate the CH_4 generated in a 44% recovery and 66% recovery scenario. The results of each of these scenarios were then multiplied by 25 to convert from CH_4 to CO_{2e} and then divided by 1,000,000 to convert to Mtonnes of CO_{2e}.

Transportation Emissions

Table 28 shows alternative emissions factors used in Equation 6.

Table 28. Feedstock transportation emissions factors for GHG and air pollutants associated with food waste hauling vehicles assuming speeds between 84.50 kph and 92.52 kph on rural unrestricted roads during average spring temperatures (Wang 2020)

Emission Type	Emissions Factor (g/km)
CO_2	1000
CH_4	0.02
N_2O	0.001
NH_3	0.02
NOx	4.88
SO_2	0.008
$PM_{2.5}$ (Primary exhaust)	0.13
VOCs	0.19

To calculate emissions from transportation, the following series of calculations was used. Equation 25, below, shows the procedure for determining the number of trips needed to transport the feedstock based on the carrying capacity of the trucks. In Equation 25, the carrying capacity is determined to be 22.7 tonnes for manure hauling vehicles and 20 tonnes for food waste hauling vehicles.

$$Trips\ Needed\ = \frac{Feedstock\ Moved\ (tonnes)}{Carrying\ Capacity\ (tonnes)}$$

Equation 25

Equation 26, below, represents Equation 25 with the relevant parameters included, resulting in the number of trips needed to transport beef manure in the 25% scenario.

$$Trips\ Needed\ (Beef)\ =\ \frac{404{,}385.77}{22.7}$$

Equation 26

Next, the number of trips needed was multiplied by the distance traveled, and then that product was multiplied by the relevant emissions factor, to produce transportation emissions in grams, as shown in Equation 27.

$$Transportation\ emissions\ (g)\ =\ \left(Trips\ Needed \times Feedstock\ Dis\tan c\ e\ (km)\right) \times EF$$

Equation 27

Equation 28, below, shows Equation 27 with the relevant parameters included, using CH_4 emissions from beef transportation as an example.

$$Transportation\ emissions\ (Beef)\ CH_4\ (g)\ =\ (17{,}814.35 \times 5{,}780\ km) \times 0.026$$

Equation 28

After this step was completed for all three GHGs (CH_4, N_2O, and CO_2), they were multiplied by the GWP (Global Warming Potential) for each pollutant (25 for CH_4, 298 for N_2O) and added together to produce transportation emissions in grams of CO_{2e}.

This process was completed for each feedstock type, then those values were added together to get a total amount.

Upgrading Emissions

To begin calculating emissions from the upgrading process, the methane output from the model had to be converted from GJ to MMBtu, as shown in Equation 29 (Calculator et al., n.d.).

$$Methane\ output\ (MMBtu)\ =\ Methane\ Output\ (GJ)\ \times\ 0.947817$$

Equation 29

Assuming an 85% efficiency rate, Equation 30, below, was repeated for both CH_4 and N_2O, using the emissions factors from **Table 12**. CO_2 emissions from the upgrading process are considered biogenic and were not included in this analysis.

$$Upgrading\ emissions\ (lbs)\ =\ (Methane\ output\ (MMBtu)\ \times\ 0.85)\ \times\ EF$$

Equation 30

Equation 31 shows Equation 30 with the relevant paraments filled in, using CH_4 emissions in the 25% scenario as an example.

$$Upgrading\ emissions\ (lbs)\ =\ (8{,}090{,}922.71\ MMBtu\ \times\ 0.85)\ \times\ 0.4358$$

Equation 31

After being converted from pounds to grams, CH_4 and N_2O emissions were then converted to CO_{2e} and summed to find total upgrading emissions in grams of CO_{2e}. Afterward, they were further converted to Mtonnes CO_{2e}.

Energy Offset

Equation 32, below, shows this analysis with the relevant parameters included, using the 25% scenario as an example.

$$Natural\ Gas\ production\ offset\ (gCO2_e)\ =\ 8{,}090{,}922{,}712\ MJ\ \times\ 4\frac{gCO2_e}{MJ}$$

Equation 32

The results from Equation 32 were then converted to Mtonnes CO_{2e}.

Fertilizer Offset

Equation 33 and Equation 34 show with the relevant parameters included, using urea fertilizer in the 25% scenario as an example.

$$Urea\ replaced\ (tonnes)\ =\ 11657.06\ \times\ 0.514$$

Equation 33

$$Avoided\ CO_{2e}\ emissions\ (tonnes)\ =\ 5992.8\ \times\ \left(\frac{1.01}{0.46}\right)$$

Equation 34

CO_{2e} results from each fertilizer type are then added together and converted to Mtonnes. The results are reflected in the negative to represent emissions offset.

Appendix B

Table S-1. Emissions factors for UAN fertilizer production air pollutant emissions (US EPA 1995a)

Emissions Type	**UAN Neutralizer (kg/tonne)***	**UAN Solid (kg/tonne)***	**UAN Cooler (kg/tonne) ***	**UAN Evaporator (kg/tonne)***
PM	0.0002, 0.22	0.60, 0.26, 0.22, 0.02	0.01, 0.26, 0.57, 0.08, 0.18	
NH3	0.42, 18.0	28.6, 0.13, 29.7, 0.07	0.02, 015, 0.8	0.27, 16.7

*mean values were used for this analysis

The replaced amount of UAN was also multiplied by the emissions factors presented in Table S-1. Table S-1 shows potential emissions from each step in the UAN production process. Emissions associated with each step were then added together to represent total UAN emissions. These values were then multiplied by 0.85 to represent an assumed 85% scrubbing efficiency (Bist et al. 2023; Hadlocon, Manuzon, and Zhao 2014; Ashtari et al. 2016).

Appendix C

```
import matplotlib.pyplot as plt
import numpy as np
%pylab inline
import pylab
import pandas as pd
import math
```

```
%pylab is deprecated, use %matplotlib inline and import the required libra
ries.
Populating the interactive namespace from numpy and matplotlib
```

```
#Baseline Manure and Food Waste Amounts
beef_baseline_manure_tonnes = 8359902
dairy_baseline_manure_tonnes = 30261825
broiler_baseline_manure_tonnes = 558382
hog_baseline_manure_tonnes = 5663619
foodwaste_baseline_tonnes = 3090878

#Unit conversions
kg2tonne = 1000
tonne2kg = 1000
tonne2mtonne = 1000000
grams2tonnes = 1000000
gj2mmbtu = 0.947817
lbs2grams = 453.592
m32gj = 0.0373
gj2m3 = 26.853
gj2j = 1000000000
kg2g = 1000
m32millionm3 = 1000000
kg2m3 = 1.406
m32ft3 = 35.3147
gj2mj = 1000
p2o5_molarmass = 0.283889 #kg/mol
p_molarmass = 0.12388
tonnes2ktonnes = 1000

#Emission Factors


VS_beef = 0.15768 #kg VS / kg manure
B0_beef = 0.19 #m3 CH4 / kg VS
VS_dairy = 0.06545 #kg VS / kg manure
B0_dairy = 0.24 #m3 CH4 / kg VS
```

```
VS_hog = 0.0847 #kg VS / kg manure
B0_hog = 0.48 #m3 CH4 / kg VS
VS_broiler = 0.432 #kg VS / kg manure
B0_broiler = 0.36 #m3 CH4 / kg VS

CH4_density = 0.657 #kg /m3

fMMS_LB = 0.053 #percent beef manure managed in liquid system
fMMS_SB = 0.45 #percent beef manure managed in solid system
fMMS_PB = 0.45 #percent beef manure managed in pasture
fMMS_OB = 0.042 #percent beef manure managed in other systems

fMMS_LD = 0.64 #percent dairy manure managed in liquid system
fMMS_SD = 0.18 #percent dairy manure managed in solid system
fMMS_PD = 0.16 #percent dairy manure managed in pasture
fMMS_OD = 0.029 #percent dairy manure managed in other systems

fMMS_LH = 0.97 #percent hog manure managed in liquid system
fMMS_SH = 0.03 #percent manure managed in solid system

fMMS_LBr = 0.07 #percent broiler manure managed in liquid system
fMMS_SBr = 0.92 #percent broiler manure managed in solid system
fMMS_PBr = 0.006 #percent broiler manure managed in pasture
fMMS_OBr = 0.006 #percent broiler manure managed in other systems

MCFBL = 0.2 #MCF for beef (liquid) %
MCFBS = 0.02 #MCF for beef (solid) %
MCFBP = 0.01 #MCF for beef (pasture) %
MCFBO = 0.01 #MCF for beef (other) %

MCFDL = 0.2 #MCF for dairy (liquid) %
MCFDS = 0.01 #MCF for dairy (solid) %
MCFDP = 0.01 #MCF for dairy (pasture) %
MCFDO = 0.01 #MCF for dairy (other) %

MCFHL = 0.2 #MCF for hog (liquid) %
MCFHS = 0.01 #MCF for hog (solid) %

MCFBrL = 0.2 #MCF for broiler (liquid) %
MCFBrS = 0.015 #MCF for broiler (solid) %
MCFBrP = 0.015 #MCF for broiler (pasture) %
MCFBrO = 0.015 #MCF for broiler (other) %


#manure AP = NH3, PM, VOCs
```

```
beef_S_NH3 = 22 * lbs2grams #g / year / AU, converted this to grams but or
iginal doc is lbs
beef_S_PM = 12.7 * lbs2grams

beef_P_NH3 = 119 #g / AU

dairy_S_NH3 = 10.2 * lbs2grams #g
dairy_S_PM = 2.3 * lbs2grams

dairy_L_NH3 = 40 * lbs2grams #g
dairy_L_VOC = 4.5 * lbs2grams

dairy_P_NH3 = 119 #g

hog_L_NH3 = 15.1 * lbs2grams #g
hog_L_VOC = 2.4 * lbs2grams

hog_S_NH3 = 17.2 * lbs2grams #g
hog_S_PM = 8.0 * lbs2grams

broiler_S_NH3 = 2.2 * lbs2grams #g

broiler_L_NH3 = 32 * lbs2grams #g
broiler_L_VOC = 4 * lbs2grams

h2s_content = 3.3 #percent
so2_natgas_ef = 26.98 * 3.3 #kg / million m3 processed


EF_CH4_pipe = 0.05 #leakage from pipeline transfer in g/m3 #source:

HHV_NG = 50000000 #J / kg

UAN_scrubber = 0.85 #85% scrubbing efficiency

GWP_CH4 = 25 #global warming potential methane
GWP_N2O = 298 #global warming potential N2O

BN2O_direct = 0.13 #beef direct N2O EF, kg N2O / tonne manure
DN2O_direct = 0.019 #dairy direct N2O EF
BrN2O_direct = 1.02 #broiler direct N2O EF
HN2O_direct = 0.014 #hog direct N2O EF

BN2O_indirect = 0.013 #beef indirect N2O EF, kg N2O / tonne manure
DN2O_indirect = 0.005 #dairy
BrN2O_indirect = 0.28 #broiler
```

```
HN2O_indirect = 0.025 #hog

CC_M = 22.7 #carrying capacity of manure hauling trucks, tonnes
CC_FW = 20 #carrying capacity of FW hauling trucks, tonnes

transpo_CH4_M = 0.0261192135513 #g CH4 / km
transpo_N2O_M = 0.0017141333083 #g N2O / km
transpo_CO2_M = 1330.9086914062 #g CO2 / km
transpo_CH4_FW = 0.0158117227256
transpo_N2O_FW = 0.00093498342903
transpo_CO2_FW = 999.68060302734

transpo_NH3_M = 0.0143314292654 #g / km
transpo_NOx_M = 6.3233842849731
transpo_SO2_M = 0.0112092262134
transpo_PM25_M = 0.2761988639832
transpo_VOC_M = 0.2708128392696

transpo_NH3_FW = 0.015540804714 #g / km
transpo_NOx_FW = 4.8768501281738
transpo_SO2_FW = 0.0084195798263
transpo_PM25_FW = 0.1337221860886
transpo_VOC_FW = 0.193175598979

upgrade_CH4 = 0.4358 #lb CH4 / MMBtu input to process
upgrade_N2O = 0.00022 #lb N2O / MMBtu input to process

upgrade_NOx = 0.0051  #lb / MMBtu input to process
#upgrade_Co = 0.0045
upgrade_VOC = 0.0006
upgrade_SOx = 0.0036
upgrade_PM = 0.001

upgrade_CO = 0.0045

pipeline_CH4 = 1.9 #g CH4 / m3
pipeline_N2O = 0.05 #g N2O / m3

pipeline_NOx = 175 # g / GJ
pipeline_CO = 45 # g / GJ
pipeline_VOC = 1.6 # g / GJ

k = 0.06 #decay rate constant (1/year)
A = (1-math.exp(-k))/k #normalization factor
MCF = 1 #methane correction factor
DOC = 0.15 #degradable organic carbon, tonne c / tonne food waste
```

```
DOCf = 0.7 #share of DOC that decays
f = 0.5 #fraction of CH4 in landfill gas
e = math.exp(-k*0)
ox = 0.1
nflare = 0.997

beef_manure_Pratio = 0.000776 #tonnes (because manure is reported in tonne
s)
dairy_manure_Pratio = 0.001162008273
broiler_manure_Pratio = 0.003159956331
hog_manure_Pratio = 0.001418140124

beef_manure_Nratio = 0.00336
dairy_manure_Nratio = 0.006884479317
broiler_manure_Nratio = 0.01079651746
hog_manure_Nratio = 0.008327801581

MCF_storage = 0.01

DAP_N_ratio = 0.18 #percentage of N in type of fertilizer
DAP_P_ratio = 0.201 #percentage of P in type of fertilizer
MAP_N_ratio = 0.11
MAP_P_ratio = 0.227
Urea_N_ratio = 0.46
Ammonia_N_ratio = 0.82
UAN_N_ratio = 0.28

DAP_used = 15000+50000+44000+21000 #tonnes of fertilizer type used in "eas
tern provinces" in 2022/2023
MAP_used = 41000+36000+44000+24000
Urea_used = 51000+158000+189000+58000
Ammonia_used = 11000
UAN_used = 47000+105000+183000+85000
total_fert_N = Urea_used+Ammonia_used+UAN_used
total_fert_NP = DAP_used+MAP_used

DAP_ratio = DAP_used / total_fert_NP
MAP_ratio = MAP_used / total_fert_NP

Urea_ratio = Urea_used / total_fert_N
Ammonia_ratio = Ammonia_used / total_fert_N
UAN_ratio = UAN_used / total_fert_N

#redo ratio

#DAP_N_used = DAP_used * DAP_N_ratio #tonnes of N via DAP used
```

```
#DAP_P_used = DAP_used * DAP_P_ratio
#MAP_N_used = MAP_used * MAP_N_ratio
#MAP_P_used = MAP_used * MAP_P_ratio
#Urea_N_used = Urea_used * Urea_N_ratio
#Ammonia_N_used = Ammonia_used * Ammonia_N_ratio
#UAN_N_used = UAN_used * UAN_N_ratio

#N_total_used = DAP_P_used+MAP_P_used+Urea_N_used+Ammonia_N_used+UAN_N_use
d #total amount of N used
#P_total_used = DAP_P_used+MAP_P_used #total amount of P used

#Urea_N = Urea_N_used / N_total_used
#Ammonia_N = Ammonia_N_used / N_total_used
#UAN_N = UAN_N_used / N_total_used

#DAP_P = DAP_P_used / P_total_used
#MAP_P = MAP_P_used / P_total_used



DAP_EF = 0.67 #ton CO2e / ton product
MAP_EF = 0.46
Urea_EF = 1.01
Ammonia_EF = 2.49
UAN_EF = 1.33

FW_CO_EF = 750 #lb / Million Standard Cubic Feet Methane
FW_NOx_EF = 40
FW_PM25_EF = 17

NH3_storage = 0.0009

residential_NOx_EF = 94 #lbs NOx / million scf natural gas
energy_SO2 = 0.6

energy_savings_EF = 4 #grams CO2e / MJ

fert_so2_EF = 0.040 #kg / Mg P2O5
fert_pm_EF = 0.34
fert_hap_EF = 0.022

#fert_nh3_EF = 18.4 #lbs / tonne
#uan_neutralizer_pm = mean([0.002, 0.22])#kg / Mg (which is the same as kg
/ tonne)
#uan_neutralizer_nh3 = mean([0.42, 18.0])
#uan_evaporation_nh3 = mean([0.27, 16.7])
```

```
#uan_solid_pm = mean([0.60, 0.26, 0.22, 0.02])
#uan_solid_nh3 = mean([28.6, 0.13, 29.7, 0.07])
#uan_cooler_pm = mean([0.01, 0.26, 0.57, 0.08, 0.18])
#uan_cooler_nh3 = mean([0.02, 0.15, 0.8])

uan_ef_try = 4.0 #kg / Mg

ammonia_co_EF = 0.136 #lbs / tonne
ammonia_nox_EF = 5.4 #lbs / tonne
ammonia_pm_EF = 0.144 #lbs / tonne
ammonia_voc_EF = 0.012 #lbs / tonne
ammonia_sox_EF = 0.00480 #lbs / tonne

urea_pm_EF = 3.8 #lbs / tonne
urea_voc_EF = 0.020 #lbs / tonne



beef_au = 8.6 #tonnes manure / year / animal
dairy_au = 46
broiler_au = 0.011
hog_au = 1.6



#define functions
def MANURE_CH4_EMISSION(VS, B0, CH4_density, MCF, fMMS, manure_tonnes,tonn
e2kg, kg2tonne):
    EF = VS*B0*CH4_density*MCF*fMMS #kg CH4/kg manure
    CH4_kg = EF*((manure_tonnes*fMMS)*tonne2kg) #kg CH4 emissions
    CH4_tonnes = CH4_kg/kg2tonne #tonne CH4 emissions

    return CH4_tonnes


def MANURE_N2O(manure_tonnes, N2OEF):
    N2O_kg = manure_tonnes*N2OEF #kg N2O

    return N2O_kg

def FW_METHANE(fw_tonnes):
    CH4_generated = (A)*(k)*(fw_tonnes)*(MCF)*(DOC)*(DOCf)*(f)*(16/12)*(e)
#tonnes CH4 generated
    rt = CH4_generated*0.44
    ch4_flared = CH4_generated*0.44
    CH4_emitted = (CH4_generated-rt)*(1-ox)+(ch4_flared*(1-nflare)) #tonne
```

```
s CH4 emitted
    CO2e = CH4_emitted * GWP_CH4
    mtonnes = CO2e / tonne2mtonne

    return(mtonnes)

def FW_JUST_METHANE(fw_tonnes):
    CH4_generated = (A)*(k)*(fw_tonnes)*(MCF)*(DOC)*(DOCf)*(f)*(16/12)*(e)
#tonnes CH4 generated
    rt = CH4_generated*0.44
    ch4_flared = CH4_generated*0.44
    CH4_emitted = (CH4_generated-rt)*(1-ox)+(ch4_flared*(1-nflare))

    return CH4_emitted

def FW_AP(methane_fw):
    kg = methane_fw * kg2tonne #tonnes x 1000 to convert to kg
    m3 = kg * kg2m3 #kg x 1.406 to convert to m3
    million_m3 = m3 / m32millionm3
    million_cft = million_m3*m32ft3
    CO_all = million_cft * FW_CO_EF #lbs
    CO_grams = CO_all * lbs2grams
    CO_t = CO_grams / grams2tonnes
    CO = CO_t / tonnes2ktonnes
    NOx_all = million_cft * FW_NOx_EF #lbs
    NOx_grams = NOx_all * lbs2grams
    NOx_t = NOx_grams / grams2tonnes
    NOx = NOx_t / tonnes2ktonnes
    PM25_all = million_cft * FW_PM25_EF #lbs
    PM25_grams = PM25_all * lbs2grams
    PM25_t = PM25_grams / grams2tonnes
    PM25 = PM25_t / tonnes2ktonnes


    return CO, NOx, PM25

def FW_AP_baseline(methane_fw):
    kg = methane_fw * kg2tonne #tonnes x 1000 to convert to kg
    m3 = kg * kg2m3 #kg x 1.406 to convert to m3
    million_m3 = m3 / m32millionm3
    million_cft = million_m3*m32ft3
    CO_all = million_cft * FW_CO_EF #lbs
    CO_grams = CO_all * lbs2grams
    CO_t = CO_grams / grams2tonnes
    CO = CO_t / tonnes2ktonnes
    NOx_all = million_cft * FW_NOx_EF #lbs
```

```
    NOx_grams = NOx_all * lbs2grams
    NOx_t = NOx_grams / grams2tonnes
    NOx = NOx_t / tonnes2ktonnes
    PM25_all = million_cft * FW_PM25_EF #lbs
    PM25_grams = PM25_all * lbs2grams
    PM25_t = PM25_grams / grams2tonnes
    PM25 = PM25_t / tonnes2ktonnes

    return CO, NOx, PM25


#def ENERGY_SAVINGS(methane_output):
    #pin = (methane_output.sum())*gj2j #J / year
    #mf = pin / HHV_NG #kg NG / year ?
    #CO2 = mf * 0.75 * (44/12) #kg CO2 / year
    #tonnes = CO2/kg2tonne
    #mtonnes = tonnes/tonne2mtonne

    #return mtonnes

def ENERGY_SAVINGS(methane_output):
    mj = (methane_output.sum()) * gj2mj
    grams = mj * energy_savings_EF #grams CO2e / mj
    tonnes = grams / grams2tonnes
    mtonnes = tonnes / tonne2mtonne

    return mtonnes


def ENERGY_SAVINGS_AP(methane_output):
    m3 = (methane_output.sum()) * gj2m3
    ft3 = m3 * m32ft3
    millionft3 = ft3 / m32millionm3
    SO2_lbs = millionft3 * energy_SO2
    SO2_grams = SO2_lbs * lbs2grams
    SO2_t = SO2_grams / grams2tonnes
    SO2 = SO2_t / tonnes2ktonnes
    residential_mm3 = millionft3 * (1/3)
    residential_NOx_lbs = residential_mm3 * residential_NOx_EF
    residential_NOx_grams = residential_NOx_lbs * lbs2grams
    residential_NOx = residential_NOx_grams / grams2tonnes

    return SO2, residential_NOx

def TRANSPO(moved, cc, distance, EF_CH4, EF_N2O, EF_CO2):
    total_moved = moved/cc
```

```
    CH4 = (total_moved * distance) * EF_CH4 #grams CH4
    N2O = (total_moved * distance) * EF_N2O #grams N2O
    CO2 = (total_moved * distance) * EF_CO2 #grams CO2
    CO2e = (CH4 * GWP_CH4) + (N2O * GWP_N2O) + CO2 #grams CO2e
    CO2e_sum = CO2e.sum()
    tonnes = CO2e_sum / grams2tonnes
    mtonnes = tonnes / tonne2mtonne

    return mtonnes

def TRANSPO_MOVED(moved, cc):
    total_moved = moved/cc

    return total_moved

def TRANSPO_AP(moved, cc, distance, EF_NH3, EF_NOx, EF_SO2, EF_PM25, EF_VO
C):
    total_moved = moved/cc
    NH3_all = (total_moved * distance) * EF_NH3 #grams NH3
    NH3_t = (NH3_all.sum())/grams2tonnes #tonnes
    NH3 = NH3_t / tonnes2ktonnes
    NOx_all = (total_moved * distance) * EF_NOx
    NOx_t = (NOx_all.sum())/grams2tonnes
    NOx = NOx_t / tonnes2ktonnes
    SO2_all = (total_moved * distance) * EF_SO2
    SO2_t = (SO2_all.sum())/grams2tonnes
    SO2 = SO2_t / tonnes2ktonnes
    PM25_all = (total_moved * distance) * EF_PM25
    PM25_t = (PM25_all.sum())/grams2tonnes
    PM25 = PM25_t / tonnes2ktonnes
    VOC_all = (total_moved * distance) * EF_VOC
    VOC_t = (VOC_all.sum())/grams2tonnes
    VOC = VOC_t / tonnes2ktonnes

    return NH3, NOx, SO2, PM25, VOC

def UPGRADE(methane_output):
    mmbtu = methane_output * gj2mmbtu
    CH4 = (mmbtu/0.85) * upgrade_CH4 #lbs
    N2O = (mmbtu/0.85) * upgrade_N2O #lbs
    CH4_grams = CH4.sum() * lbs2grams #grams
    N2O_grams = N2O.sum() * lbs2grams #grams
    CO2e = (CH4_grams * GWP_CH4) + (N2O_grams * GWP_N2O)
    tonnes = CO2e / grams2tonnes
    mtonnes = tonnes / tonne2mtonne
```

```
    return mtonnes

def UPGRADE_AP(methane_output):
    mmbtu = methane_output * gj2mmbtu
    NOx_all = (mmbtu/0.85) * upgrade_NOx #lbs
    NOx_grams = (NOx_all.sum()) * lbs2grams
    NOx_t = NOx_grams / grams2tonnes
    NOx = NOx_t / tonnes2ktonnes
    CO_all = (mmbtu/0.85) * upgrade_CO
    CO_grams = (CO_all.sum()) * lbs2grams
    CO_t = CO_grams / grams2tonnes
    CO = CO_t / tonnes2ktonnes
    VOC_all = (mmbtu/0.85) * upgrade_VOC
    VOC_grams = (VOC_all.sum()) * lbs2grams
    VOC_t = VOC_grams / grams2tonnes
    VOC = VOC_t / tonnes2ktonnes
    SOx_all = (mmbtu/0.85) * upgrade_SOx
    SOx_grams = (SOx_all.sum()) * lbs2grams
    SOx_t = SOx_grams / grams2tonnes
    SOx = SOx_t / tonnes2ktonnes
    PM_all = (mmbtu/0.85) * upgrade_PM
    PM_grams = (PM_all.sum()) * lbs2grams
    PM_t = PM_grams / grams2tonnes
    PM = PM_t / tonnes2ktonnes

    return NOx, CO, VOC, SOx, PM

def UPGRADE_AP_GRIDDED(methane_output):
    mmbtu = methane_output * gj2mmbtu
    NOx_all = (mmbtu/0.85) * upgrade_NOx #lbs
    NOx_grams = (NOx_all) * lbs2grams
    NOx = NOx_grams / grams2tonnes
    CO_all = (mmbtu/0.85) * upgrade_CO
    CO_grams = (CO_all) * lbs2grams
    CO = CO_grams / grams2tonnes
    VOC_all = (mmbtu/0.85) * upgrade_VOC
    VOC_grams = (VOC_all) * lbs2grams
    VOC = VOC_grams / grams2tonnes
    SOx_all = (mmbtu/0.85) * upgrade_SOx
    SOx_grams = (SOx_all) * lbs2grams
    SOx = SOx_grams / grams2tonnes
    PM_all = (mmbtu/0.85) * upgrade_PM
    PM_grams = (PM_all) * lbs2grams
    PM = PM_grams / grams2tonnes

    return NOx, CO, VOC, SOx, PM
```

```
def PIPELINE(methane_output):
    m3 = (methane_output.sum()) * gj2m3
    CH4 = m3 * pipeline_CH4
    N2O = m3 * pipeline_N2O
    CO2e = (CH4 * GWP_CH4) + (N2O * GWP_N2O) #grams
    tonnes = CO2e/grams2tonnes
    mtonnes = tonnes/tonne2mtonne

    return mtonnes

def PIPELINE_AP(methane_output):
    NOx_all = (methane_output.sum()) * pipeline_NOx #grams NOx
    NOx = NOx_all / grams2tonnes
    CO_all = (methane_output.sum()) * pipeline_CO
    CO = CO_all / grams2tonnes
    VOC_all = (methane_output.sum()) * pipeline_VOC
    VOC = VOC_all / grams2tonnes

    return NOx, CO, VOC

def TOTAL_N(beef, dairy, broiler, hog):
    total_N_beef = beef * beef_manure_Nratio #tonnes
    total_N_dairy = dairy * dairy_manure_Nratio
    total_N_broiler = broiler * broiler_manure_Nratio
    total_N_hog = hog * hog_manure_Nratio
    total_N = total_N_beef+total_N_dairy+total_N_broiler+total_N_hog

    return total_N

def TOTAL_P(beef, dairy, broiler, hog):
    total_P_beef = beef * beef_manure_Pratio #tonnes
    total_P_dairy = dairy * dairy_manure_Pratio
    total_P_broiler = broiler * broiler_manure_Pratio
    total_P_hog = hog * hog_manure_Pratio
    total_P = total_P_beef+total_P_dairy+total_P_broiler+total_P_hog

    return total_P


def TOTAL_FERT_USED(N_total_used, P_total_used):
    replaced_N = N_total_used #tons of N/P potentially replaced by N gener
ated in scenario
    replaced_P = P_total_used
```

```
    DAP_replaced_P = replaced_P * DAP_ratio #amount of DAP used based on t
he amount of P available in scenario
    DAP_CO2e = DAP_replaced_P * (DAP_EF / DAP_P_ratio) #tonnes of CO2e rep
laced via DAP #divide EF by P_ratio
    DAP_replaced_N = replaced_P * ((DAP_ratio * DAP_N_ratio)/DAP_P_ratio)
#amount of N replaced via DAP based on P available in scenario

    MAP_replaced_P = replaced_P * MAP_ratio #amount of MAP used based on t
he amount of P available in scenario
    MAP_CO2e = MAP_replaced_P * (MAP_EF / MAP_P_ratio) #tonnes of CO2e rep
laced via MAP #divide EF by P_ratio
    MAP_replaced_N = replaced_P * ((MAP_ratio * MAP_N_ratio)/MAP_P_ratio)
#amount of N replaced via MAP based on P availabe in scenario

    totalNused = DAP_replaced_N+MAP_replaced_N #total amount of N used via
MAP and DAP in scenario
    leftoverN = replaced_N - totalNused #amount of N leftover from MAP and
DAP (unaccounted for)

    Urea_replaced = leftoverN * Urea_ratio
    Urea_CO2e = Urea_replaced * (Urea_EF / Urea_N_ratio)

    Ammonia_replaced = leftoverN * Ammonia_ratio
    Ammonia_CO2e = Ammonia_replaced * (Ammonia_EF / Ammonia_N_ratio) #EF d
ivided by N_ratio

    UAN_replaced = leftoverN * UAN_ratio
    UAN_CO2e = UAN_replaced * (UAN_EF / UAN_N_ratio)

    total_CO2e_replaced = - (DAP_CO2e+MAP_CO2e+Urea_CO2e+Ammonia_CO2e+UAN_
CO2e)
    mtonnes = total_CO2e_replaced / tonne2mtonne

    return mtonnes

def FERT_REPLACED(P_total_used, N_total_used):
    replaced_P = P_total_used #tonnes, which is bascially the same as Mg
    replaced_N = N_total_used
    DAP_replaced_P = replaced_P * DAP_ratio #tonnes
    MAP_replaced_P = replaced_P * MAP_ratio #tonnes
    total_P = (DAP_replaced_P + MAP_replaced_P) * tonne2kg #convert from t
onnes to kg

    DAP_replaced_N = replaced_P * ((DAP_ratio * DAP_N_ratio)/DAP_P_ratio)
#tonnes
    MAP_replaced_N = replaced_P * ((MAP_ratio * MAP_N_ratio)/MAP_P_ratio)
```

```
#tonnes
    totalNused = DAP_replaced_N+MAP_replaced_N #tonnes
    leftoverN = replaced_N - totalNused #tonnes

    Urea_replaced = leftoverN * Urea_ratio #tonnes
    Ammonia_replaced = leftoverN * Ammonia_ratio #tonnes
    UAN_replaced = leftoverN * UAN_ratio #tonnes

    total_fert_replaced = DAP_replaced_P + MAP_replaced_P + Urea_replaced
+ Ammonia_replaced + UAN_replaced

    return total_fert_replaced


def FERT_AP(P_total_used, N_total_used):
    replaced_P = P_total_used #tonnes, which is bascially the same as Mg
    replaced_N = N_total_used
    DAP_replaced_P = replaced_P * DAP_ratio #tonnes
    MAP_replaced_P = replaced_P * MAP_ratio #tonnes
    total_P = (DAP_replaced_P + MAP_replaced_P) * tonne2kg #convert from t
onnes to kg

    DAP_replaced_N = replaced_P * ((DAP_ratio * DAP_N_ratio)/DAP_P_ratio)
#tonnes
    MAP_replaced_N = replaced_P * ((MAP_ratio * MAP_N_ratio)/MAP_P_ratio)
#tonnes
    totalNused = DAP_replaced_N+MAP_replaced_N #tonnes
    leftoverN = replaced_N - totalNused #tonnes

    Urea_replaced = leftoverN * Urea_ratio #tonnes
    Ammonia_replaced = leftoverN * Ammonia_ratio #tonnes
    UAN_replaced = leftoverN * UAN_ratio #tonnes

    P2O5_mass_kg = (p2o5_molarmass/p_molarmass) * total_P #kg
    P2O5_mass = P2O5_mass_kg / kg2tonne #convert back to tonne (Mg)
    so2 = (fert_so2_EF * P2O5_mass) / kg2tonne #originally in kg, converte
d to tonnes
    pm_p = (fert_pm_EF * P2O5_mass) / kg2tonne
    hap = (fert_hap_EF * P2O5_mass) / kg2tonne

    pm_g_uan = ((UAN_replaced * uan_ef_try) * kg2g)
    pm_uan = (pm_g_uan / grams2tonnes) * UAN_scrubber

    nh3_g = ((UAN_replaced * uan_ef_try) * kg2g)
    nh3_t = (nh3_g / grams2tonnes) * UAN_scrubber
    nh3 = nh3_t / tonnes2ktonnes
```

```
    co_g = (ammonia_co_EF * Ammonia_replaced) * lbs2grams #lb, converted t
o grams
    co_t = co_g / grams2tonnes
    co = co_t / tonnes2ktonnes

    nox_g = (ammonia_nox_EF * Ammonia_replaced) * lbs2grams
    nox_t = nox_g / grams2tonnes
    nox = nox_t / tonnes2ktonnes

    pm_g_a = (ammonia_pm_EF * Ammonia_replaced) * lbs2grams
    pm_a = pm_g_a / grams2tonnes

    voc_g_a = (ammonia_voc_EF * Ammonia_replaced) * lbs2grams
    voc_a = voc_g_a / grams2tonnes

    sox_g = (ammonia_sox_EF * Ammonia_replaced) * lbs2grams
    sox_a = sox_g / grams2tonnes

    pm_g_u = (urea_pm_EF * Urea_replaced) * lbs2grams
    pm_u = pm_g_u / grams2tonnes

    voc_g_u = (urea_voc_EF * Urea_replaced) * lbs2grams
    voc_u = voc_g_u / grams2tonnes

    sox = (sox_a + so2) / tonnes2ktonnes
    pm = (pm_p + pm_a + pm_u + pm_uan) / tonnes2ktonnes
    voc = (voc_a + voc_u ) / tonnes2ktonnes


    return sox, pm, nh3, nox, co, voc


def STORAGE_AP(TOTAL_N):
    TOTAL_N_kg = TOTAL_N * kg2tonne #kg of N in feedstock
    NH3_all = TOTAL_N_kg * NH3_storage #kg of NH3 associated with that amo
unt of N in feedstock
    NH3_t = NH3_all / tonne2kg #converting kg of NH3 to tonnes
    NH3 = NH3_t / tonnes2ktonnes

    return NH3


def ANIMAL_UNIT(beef_manure, dairy_manure, hog_manure, broiler_manure):
    beef = beef_manure / beef_au #number of animals based on tonnes of man
ure
```

```
    dairy = (dairy_manure / dairy_au) / 0.7 #number of animal based on ton
nes of manure, then converted to EPA AU
    hog = (hog_manure / hog_au) / 2.5
    broiler = (broiler_manure / broiler_au) / 100

    return beef, dairy, hog, broiler

def MANURE_AP(beef_AU, dairy_AU, hog_AU, broiler_AU):
    beef_S = beef_AU * fMMS_SB
    beef_P = beef_AU * fMMS_PB
    dairy_L = dairy_AU * fMMS_LD
    dairy_S = dairy_AU * fMMS_SD
    dairy_P = dairy_AU * fMMS_PD
    hog_S = hog_AU * fMMS_SH
    hog_L = hog_AU * fMMS_LH
    broiler_S = broiler_AU * fMMS_LBr
    broiler_L = broiler_AU * fMMS_SBr

    beef_NH3_g = (beef_S * beef_S_NH3) + (beef_P * beef_P_NH3) #make sure
you've changed the emissions factors in g!!!!
    beef_PM_g = beef_S * beef_S_PM

    dairy_NH3_g = (dairy_S * dairy_S_NH3) + (dairy_L * dairy_L_NH3) + (dai
ry_P * dairy_P_NH3)
    dairy_PM_g = dairy_S * dairy_S_PM
    dairy_VOC_g = dairy_L * dairy_L_VOC

    hog_NH3_g = (hog_L * hog_L_NH3) + (hog_S * hog_S_NH3)
    hog_PM_g = hog_S * hog_S_PM
    hog_VOC_g = hog_L * hog_L_VOC

    broiler_NH3_g = (broiler_S * broiler_S_NH3) + (broiler_L * broiler_L_N
H3)
    broiler_VOC_g = broiler_L * broiler_L_VOC

    NH3_g = beef_NH3_g + dairy_NH3_g + hog_NH3_g + broiler_NH3_g
    PM_g = beef_PM_g + dairy_PM_g + hog_PM_g
    VOC_g = dairy_VOC_g + hog_VOC_g + broiler_VOC_g

    NH3_t = NH3_g / grams2tonnes
    PM_t = PM_g / grams2tonnes
    VOC_t = VOC_g / grams2tonnes

    NH3 = NH3_t / tonnes2ktonnes
    PM = PM_t / tonnes2ktonnes
    VOC = VOC_t / tonnes2ktonnes
```

```
    return NH3, PM, VOC

def MANURE_AP_GRIDDED(beef_AU, dairy_AU, hog_AU, broiler_AU):
    beef_S = beef_AU * fMMS_SB
    beef_P = beef_AU * fMMS_PB
    dairy_L = dairy_AU * fMMS_LD
    dairy_S = dairy_AU * fMMS_SD
    dairy_P = dairy_AU * fMMS_PD
    hog_S = hog_AU * fMMS_SH
    hog_L = hog_AU * fMMS_LH
    broiler_S = broiler_AU * fMMS_LBr
    broiler_L = broiler_AU * fMMS_SBr

    beef_NH3_g = (beef_S * beef_S_NH3) + (beef_P * beef_P_NH3) #make sure
you've changed the emissions factors in g!!!!
    beef_PM_g = beef_S * beef_S_PM

    beef_NH3 = beef_NH3_g / grams2tonnes
    beef_PM = beef_PM_g / grams2tonnes

    dairy_NH3_g = (dairy_S * dairy_S_NH3) + (dairy_L * dairy_L_NH3) + (dai
ry_P * dairy_P_NH3)
    dairy_PM_g = dairy_S * dairy_S_PM
    dairy_VOC_g = dairy_L * dairy_L_VOC

    dairy_NH3 = dairy_NH3_g / grams2tonnes
    dairy_PM = dairy_PM_g / grams2tonnes
    dairy_VOC = dairy_VOC_g / grams2tonnes

    hog_NH3_g = (hog_L * hog_L_NH3) + (hog_S * hog_S_NH3)
    hog_PM_g = hog_S * hog_S_PM
    hog_VOC_g = hog_L * hog_L_VOC

    hog_NH3 = hog_NH3_g / grams2tonnes
    hog_PM = hog_PM_g / grams2tonnes
    hog_VOC = hog_VOC_g / grams2tonnes

    broiler_NH3_g = (broiler_S * broiler_S_NH3) + (broiler_L * broiler_L_N
H3)
    broiler_VOC_g = broiler_L * broiler_L_VOC

    broiler_NH3 = broiler_NH3_g / grams2tonnes
    broiler_VOC = broiler_VOC_g / grams2tonnes
```

```
    return beef_NH3, beef_PM, dairy_NH3, dairy_PM, dairy_VOC, hog_NH3, hog
_PM, hog_VOC, broiler_NH3, broiler_VOC



def NATGAS_PROCESSING_AP(methane_output):
    m3 = (methane_output.sum()) * gj2m3
    million_m3 = m3 / m32millionm3
    so2_kg = so2_natgas_ef * million_m3
    so2_t = so2_kg / kg2tonne
    so2 = so2_t / tonnes2ktonnes

    return so2


#values for plotting
#TOTAL_BASELINE_MANURE = TOTAL_MANURE_N2O()+TOTAL_MANURE_CH4()
#TOTAL_BASELINE_FW = FW_METHANE(foodwaste_baseline_tonnes)

#ENERGY_SAVINGS_25 = ENERGY_SAVINGS(methane_output_25)
#TRANSPO_25 = BEEF_TRANSPO_25()+DAIRY_TRANSPO_25()+BROILER_TRANSPO_25()+HO
G_TRANSPO_25()
#UPGRADE_25 = UPGRADE(methane_output_25)
#TOTAL_25_MANURE = TOTAL_MANURE_N2O_25()+TOTAL_MANURE_CH4_25()
#TOTAL_25_FW = FW_METHANE(leftover_fw_25)
#PIPELINE_25 = PIPELINE(methane_output_25)

m3_50 = (methane_output_50.sum())*gj2m3

DAP_ratio, MAP_ratio , Urea_ratio, Ammonia_ratio , UAN_ratio

(0.4727272727272727,
 0.5272727272727272,
 0.5140924464487034,
 0.012401352874859075,
 0.47350620067643745)

m3_50 #should be the amount of methane in m3 from the 25% scenario?

434531095.20379895

test_ENERGY_SAVINGS_50 = ENERGY_SAVINGS(methane_output_50)

test_ENERGY_SAVINGS_50 #mtonnes of CO2 offset from the 25% scenario, conve
rt to kg

0.06472738170093456
```

```
tonne_50 = test_ENERGY_SAVINGS_50 * tonne2mtonne
kg_50 = tonne_50 * 1000 #convert to kg

#so to get kg CO2 / m3 natural gas, divide kg CO2 by m3....

energyoffset_test = kg_50 / m3_50

energyoffset_test

0.1489591479536737

29603/1000000

0.029603

uan_solid_nh3

14.625

TOTAL_N_25 = TOTAL_N(total_beef_manure_25, total_dairy_manure_25, total_br
oiler_manure_25, total_hog_manure_25)

TOTAL_N_25

13448.374795299525

TOTAL_P_25 = TOTAL_P(total_beef_manure_25, total_dairy_manure_25, total_br
oiler_manure_25, total_hog_manure_25)

TOTAL_N_50 = TOTAL_N(total_beef_manure_50, total_dairy_manure_50, total_br
oiler_manure_50, total_hog_manure_50)

TOTAL_P_50 = TOTAL_P(total_beef_manure_50, total_dairy_manure_50, total_br
oiler_manure_50, total_hog_manure_50)

TOTAL_N_75 = TOTAL_N(total_beef_manure_75, total_dairy_manure_75, total_br
oiler_manure_75, total_hog_manure_75)

TOTAL_P_75 = TOTAL_P(total_beef_manure_75, total_dairy_manure_75, total_br
oiler_manure_75, total_hog_manure_75)

TOTAL_N_100 = TOTAL_N(total_beef_manure_100, total_dairy_manure_100, total
_broiler_manure_100, total_hog_manure_100)

TOTAL_P_100 = TOTAL_P(total_beef_manure_100, total_dairy_manure_100, total
_broiler_manure_100, total_hog_manure_100)

TOTAL_P_50

17416.789151226374
```

```
P_total_used

---------------------------------------------------------------------------
-

NameError                                 Traceback (most recent call last
)

Cell In[472], line 1
----> 1 P_total_used


NameError: name 'P_total_used' is not defined
```

```
FERT_25 = TOTAL_FERT_USED(TOTAL_N_25, TOTAL_P_25)
```

```
FERT_25
```

```
-0.04679316875293088
```

```
FERT_50 = TOTAL_FERT_USED(TOTAL_N_50, TOTAL_P_50)
```

```
FERT_50
```

```
-0.3357545446452093
```

```
FERT_75 = TOTAL_FERT_USED(TOTAL_N_75, TOTAL_P_75)
```

```
FERT_75
```

```
-0.6684390734903892
```

```
FERT_100 = TOTAL_FERT_USED(TOTAL_N_100, TOTAL_P_100)
```

```
FERT_100
```

```
-1.0059723383396566
```

```
#Beef Manure CH4 Emissions
def TOTAL_BEEF():
    S1 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBL, fMMS_LB
, beef_baseline_manure_tonnes, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBS, fMMS_SB
, beef_baseline_manure_tonnes, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBP, fMMS_PB
, beef_baseline_manure_tonnes, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBO, fMMS_OB
, beef_baseline_manure_tonnes, tonne2kg, kg2tonne)
```

```
    return S1 + S2 + S3 + S4

def TOTAL_DAIRY():
    S1 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDL, fMMS_
LD, dairy_baseline_manure_tonnes, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDS, fMMS_
SD, dairy_baseline_manure_tonnes, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDP, fMMS_
PD, dairy_baseline_manure_tonnes, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDO, fMMS_
OD, dairy_baseline_manure_tonnes, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_BROILER():
    S1 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrL,
fMMS_LBr, broiler_baseline_manure_tonnes, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrS,
fMMS_SBr, broiler_baseline_manure_tonnes, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrP,
fMMS_PBr, broiler_baseline_manure_tonnes, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrO,
fMMS_OBr, broiler_baseline_manure_tonnes, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_HOG():
    S1 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCFHL, fMMS_LH,
hog_baseline_manure_tonnes, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCFHS, fMMS_SH,
hog_baseline_manure_tonnes, tonne2kg, kg2tonne)

    return S1 + S2

def TOTAL_MANURE_CH4():
    total_manure = TOTAL_BEEF()+TOTAL_DAIRY()+TOTAL_BROILER()+TOTAL_HOG()
    CO2e = total_manure*GWP_CH4
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

def BEEF_N2O():
    D1 = MANURE_N2O(beef_baseline_manure_tonnes, BN2O_direct)
    D2 = MANURE_N2O(beef_baseline_manure_tonnes, BN2O_indirect)

    return D1 + D2
```

```
def DAIRY_N2O():
    D1 = MANURE_N2O(dairy_baseline_manure_tonnes, DN2O_direct)
    D2 = MANURE_N2O(dairy_baseline_manure_tonnes, DN2O_indirect)

    return D1 + D2

def BROILER_N2O():
    D1 = MANURE_N2O(broiler_baseline_manure_tonnes, BrN2O_direct)
    D2 = MANURE_N2O(broiler_baseline_manure_tonnes, BrN2O_indirect)

    return D1 + D2

def HOG_N2O():
    D1 = MANURE_N2O(hog_baseline_manure_tonnes, HN2O_direct)
    D2 = MANURE_N2O(hog_baseline_manure_tonnes, HN2O_indirect)

    return D1 + D2

def TOTAL_MANURE_N2O():
    total_manure = BEEF_N2O()+DAIRY_N2O()+BROILER_N2O()+HOG_N2O() #kg
    tonnes = total_manure/kg2tonne
    CO2e = tonnes*GWP_N2O
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

TOTAL_BASELINE_MANURE = TOTAL_MANURE_N2O()+TOTAL_MANURE_CH4()

TOTAL_BASELINE_MANURE

2.257637291209932

TOTAL_BASELINE_FW = FW_METHANE(foodwaste_baseline_tonnes)

METHANE_BASELINE_FW = FW_JUST_METHANE(foodwaste_baseline_tonnes)
FW_AP_BASELINE = FW_AP(METHANE_BASELINE_FW)

METHANE_BASELINE_FW

6366.986792805918

FW_AP_BASELINE = (FW_AP_baseline(METHANE_BASELINE_FW))

FW_AP_BASELINE

#co nox pm

(0.10754777765046578, 0.005735881474691507, 0.0024377496267438906)
```

```
baseline_AU = ANIMAL_UNIT(beef_baseline_manure_tonnes, dairy_baseline_manu
re_tonnes, hog_baseline_manure_tonnes, broiler_baseline_manure_tonnes)

baseline_AU

(972081.6279069767, 939808.2298136646, 1415904.75, 507620.0)

baseline_AU

(972081.6279069767, 939808.2298136646, 1415904.75, 507620.0)

baseline_manure_AP = MANURE_AP(baseline_AU[0], baseline_AU[1], baseline_AU
[2], baseline_AU[3])

baseline_manure_AP

(32.68323944474921, 2.85052827733232, 3.570184330629538)
```

Chapter 625% scenario

```
#25% scenario, creating arrays

df_25 = pd.read_excel('/Users/savannahbindas/Documents/UW/Research/Process
ed_Results_MO.xlsx', sheet_name='25%')

methane_output_series_25 = df_25['Methane Output (GJ/y)']
methane_output_25 = methane_output_series_25.values

beef_moved_series_25 = df_25['Beef Moved (tonnes)']
beef_moved_all_25 = beef_moved_series_25.values
beef_moved_25 = beef_moved_all_25[~np.isnan(beef_moved_all_25)]

beef_distance_series_25 = df_25['Beef Distance (km)']
beef_distance_all_25 = beef_distance_series_25.values
beef_distance_25 = beef_distance_all_25[~np.isnan(beef_distance_all_25)]

broiler_moved_series_25 = df_25['Broiler Moved (tonnes)']
broiler_moved_all_25 = broiler_moved_series_25.values
broiler_moved_25 = broiler_moved_all_25[~np.isnan(broiler_moved_all_25)]

broiler_distance_series_25 = df_25['Broiler Distance (km)']
broiler_distance_all_25 = broiler_distance_series_25.values
broiler_distance_25 = broiler_distance_all_25[~np.isnan(broiler_distance_a
ll_25)]

dairy_moved_series_25 = df_25['Dairy Moved (tonnes)']
dairy_moved_all_25 = dairy_moved_series_25.values
dairy_moved_25 = dairy_moved_all_25[~np.isnan(dairy_moved_all_25)]

dairy_distance_series_25 = df_25['Dairy Distance (km)']
dairy_distance_all_25 = dairy_distance_series_25.values
dairy_distance_25 = dairy_distance_all_25[~np.isnan(dairy_distance_all_25)
]

hog_moved_series_25 = df_25['Hog Moved (tonnes)']
hog_moved_all_25 = hog_moved_series_25.values
hog_moved_25 = hog_moved_all_25[~np.isnan(hog_moved_all_25)]

hog_distance_series_25 = df_25['Hog Distance (km)']
hog_distance_all_25 = hog_distance_series_25.values
hog_distance_25 = hog_distance_all_25[~np.isnan(hog_distance_all_25)]

fw_moved_series_25 = df_25['FW Moved (tonnes)']
```

```
fw_moved_all_25 = fw_moved_series_25.values
fw_moved_25 = fw_moved_all_25[~np.isnan(fw_moved_all_25)]

fw_distance_series_25 = df_25['FW Distance (km)']
fw_distance_all_25 = fw_distance_series_25.values
fw_distance_25 = fw_distance_all_25[~np.isnan(fw_distance_all_25)]

baseline_AU = ANIMAL_UNIT(beef_baseline_manure_tonnes, dairy_baseline_manu
re_tonnes, hog_baseline_manure_tonnes, broiler_baseline_manure_tonnes)



median_value_fw = (np.median(fw_distance_25)), (np.median(fw_distance_50))
, (np.median(fw_distance_75)), (np.median(fw_distance_100))

median_value_beef = (np.median(beef_distance_25)), (np.median(beef_distanc
e_50)), (np.median(beef_distance_75)), (np.median(beef_distance_100))

median_value_dairy = (np.median(dairy_distance_25)), (np.median(dairy_dist
ance_50)), (np.median(dairy_distance_75)), (np.median(dairy_distance_100))


median_value_broiler = (np.median(broiler_distance_25)), (np.median(broile
r_distance_50)), (np.median(broiler_distance_75)), (np.median(broiler_dist
ance_100))


median_value_hog = (np.median(hog_distance_25)), (np.median(hog_distance_5
0)), (np.median(hog_distance_75)), (np.median(hog_distance_100))

median_value_fw, median_value_beef, median_value_dairy, median_value_broil
er, median_value_hog

((52.0, 27.0, 63.5, 25.0),
 (18.0, 9.0, 12.0, 13.0),
 (12.5, 8.0, 10.0, 9.0),
 (20.0, 10.0, 10.0, 9.0),
 (14.0, 9.0, 10.0, 9.0))

beef_moved_25.sum() / 22.7

17814.351041414273

(beef_moved_25 / 22.7).sum()

17814.351041414273

beef_distance_25.sum()
```

```
5787.0
```

```
methane_output_25.sum()
```

```
8090922.712616812
```

```
ENERGY_SAVINGS_25 = -ENERGY_SAVINGS(methane_output_25)
```

```
methane_output_25.sum() * gj2mj
```

```
8090922712.616813
```

```
NATGAS_AP_25 = NATGAS_PROCESSING_AP(methane_output_25)
```

```
NATGAS_AP_25
```

```
0.019344020765187497
```

```
def BEEF_TRANSPO_25():
    B1 = TRANSPO(beef_moved_25, CC_M, beef_distance_25, transpo_CH4_M, tra
nspo_N2O_M, transpo_CO2_M)

    return B1

def DAIRY_TRANSPO_25():
    B1 = TRANSPO(dairy_moved_25, CC_M, dairy_distance_25, transpo_CH4_M, t
ranspo_N2O_M, transpo_CO2_M)

    return B1

def BROILER_TRANSPO_25():
    B1 = TRANSPO(broiler_moved_25, CC_M, broiler_distance_25, transpo_CH4_
M, transpo_N2O_M, transpo_CO2_M)

    return B1

def HOG_TRANSPO_25():
    B1 = TRANSPO(hog_moved_25, CC_M, hog_distance_25, transpo_CH4_M, trans
po_N2O_M, transpo_CO2_M)

    return B1

def FW_TRANSPO_25():
    B1 = TRANSPO(fw_moved_25, CC_FW, fw_distance_25, transpo_CH4_FW, trans
po_N2O_FW, transpo_CO2_FW)

    return B1
```

```
TRANSPO_25 = BEEF_TRANSPO_25()+DAIRY_TRANSPO_25()+BROILER_TRANSPO_25()+HOG
_TRANSPO_25()+FW_TRANSPO_25()
```

```
TRANSPO_25
```

```
0.0075021820949069985
```

```
BEEF_TRANSPO_AP_25 = TRANSPO_AP(beef_moved_25, CC_M, beef_distance_25, tra
nspo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VOC_M)
DAIRY_TRANSPO_AP_25 = TRANSPO_AP(dairy_moved_25, CC_M, dairy_distance_25,
transpo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VOC_M
)
HOG_TRANSPO_AP_25 = TRANSPO_AP(hog_moved_25, CC_M, hog_distance_25, transp
o_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VOC_M)
BROILER_TRANSPO_AP_25 = TRANSPO_AP(broiler_moved_25, CC_M, broiler_distanc
e_25, transpo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo
_VOC_M)
FW_TRANSPO_AP_25 = TRANSPO_AP(fw_moved_25, CC_FW, fw_distance_25, transpo_
NH3_FW, transpo_NOx_FW, transpo_SO2_FW, transpo_PM25_FW, transpo_VOC_FW)
```

```
TOTAL_NH3_TRANSPO_25 = BEEF_TRANSPO_AP_25[0]+DAIRY_TRANSPO_AP_25[0]+HOG_TR
ANSPO_AP_25[0]+BROILER_TRANSPO_AP_25[0]+FW_TRANSPO_AP_25[0]
TOTAL_NOx_TRANSPO_25 = BEEF_TRANSPO_AP_25[1]+DAIRY_TRANSPO_AP_25[1]+HOG_TR
ANSPO_AP_25[1]+BROILER_TRANSPO_AP_25[1]+FW_TRANSPO_AP_25[1]
TOTAL_SO2_TRANSPO_25 = BEEF_TRANSPO_AP_25[2]+DAIRY_TRANSPO_AP_25[2]+HOG_TR
ANSPO_AP_25[2]+BROILER_TRANSPO_AP_25[2]+FW_TRANSPO_AP_25[2]
TOTAL_PM25_TRANSPO_25 = BEEF_TRANSPO_AP_25[3]+DAIRY_TRANSPO_AP_25[3]+HOG_T
RANSPO_AP_25[3]+BROILER_TRANSPO_AP_25[3]+FW_TRANSPO_AP_25[3]
TOTAL_VOC_TRANSPO_25 = BEEF_TRANSPO_AP_25[4]+DAIRY_TRANSPO_AP_25[4]+HOG_TR
ANSPO_AP_25[4]+BROILER_TRANSPO_AP_25[4]+FW_TRANSPO_AP_25[4]
```

```
TOTAL_NH3_TRANSPO_25, TOTAL_NOx_TRANSPO_25, TOTAL_SO2_TRANSPO_25, TOTAL_PM
25_TRANSPO_25, TOTAL_VOC_TRANSPO_25
```

```
(0.0001058603950155364,
 0.03628742098181989,
 6.313898929722315e-05,
 0.0011677010219627313,
 0.0014715371416265844)
```

```
UPGRADE_25 = UPGRADE(methane_output_25)
```

```
UPGRADE_25
```

```
0.044854059102742344
```

```
UPGRADE_AP_25 = UPGRADE_AP(methane_output_25)

UPGRADE_AP_25

(0.020870804176427653,
 0.01841541544978911,
 0.0024553887266385477,
 0.014732332359831284,
 0.0040923145443975805)

UPGRADE_AP_GRIDDED_25 = UPGRADE_AP_GRIDDED(methane_output_25)

NOx, CO, VOC, SOx, PM

(array([150.48700521,  83.48321671,   0.        ,   0.        ,
          0.        ,  -7.73657415]),
 array([ 0.        , 73.6616618 ,  0.        ,  0.        ,  0.        ,
        -0.19484705]),
 array([ 6.32222139,  9.82155491,  0.        ,  0.        ,  0.        ,
        -1.20503003]),
 array([  0.2649991 ,  58.92932944,   0.        ,   0.        ,
        -77.37608306,  -4.72290327]),
 array([   5.95279167,   16.36925818,    0.        ,    0.        ,
           0.        , -328.10257982]))

UPGRADE_NOx_25 = UPGRADE_AP_GRIDDED_25[0]
UPGRADE_CO_25 = UPGRADE_AP_GRIDDED_25[1]
UPGRADE_VOC_25 = UPGRADE_AP_GRIDDED_25[2]
UPGRADE_SOx_25 = UPGRADE_AP_GRIDDED_25[3]
UPGRADE_PM_25 = UPGRADE_AP_GRIDDED_25[4]

numpy.savetxt("upgrading_nox.csv", UPGRADE_NOx_25, delimiter=",")
numpy.savetxt("upgrading_co.csv", UPGRADE_CO_25, delimiter=",")
numpy.savetxt("upgrading_voc.csv", UPGRADE_VOC_25, delimiter=",")
numpy.savetxt("upgrading_sox.csv", UPGRADE_SOx_25, delimiter=",")
numpy.savetxt("upgrading_pm.csv", UPGRADE_PM_25, delimiter=",")

#leftover manure management emissions
#creating arrays

beef_manure_series_25 = df_25['Beef Manure (tonnes/y)']
beef_manure_25 =( beef_manure_series_25.values)

broiler_manure_series_25 = df_25['Broiler Manure (tonnes/y)']
broiler_manure_25 = (broiler_manure_series_25.values)

dairy_manure_series_25 = df_25['Dairy Manure (tonnes/y)']
dairy_manure_25 = (dairy_manure_series_25.values)
```

```
hog_manure_series_25 = df_25['Hog Manure (tonnes/y)']
hog_manure_25 = (hog_manure_series_25.values)

fw_series_25 = df_25['Food Waste (tonnes/y)']
fw_25 = fw_series_25.values


manure_AP_25 = MANURE_AP(AU_25[0], AU_25[1], AU_25[2], AU_25[3])

AU_25 = ANIMAL_UNIT(total_beef_manure_25, leftover_dairy_25, leftover_hog_
25, leftover_broiler_25)
manure_AP_25 = MANURE_AP(AU_25[0], AU_25[1], AU_25[2], AU_25[3])

manure_AP_25

(24.418411763619275, 0.42798842738639414, 3.028460869718345)

manure_AP_25

(24418.411763619275, 427.98842738639416, 3028.460869718345)

GRIDDED_AU_25_beef = GRIDDED_AU_25[0]
GRIDDED_AU_25_dairy = GRIDDED_AU_25[1]
GRIDDED_AU_25_hog = GRIDDED_AU_25[2]
GRIDDED_AU_25_broiler = GRIDDED_AU_25[3]

GRIDDED_MANURE_AP_25 = MANURE_AP_GRIDDED(GRIDDED_AU_25_beef, GRIDDED_AU_25
_dairy, GRIDDED_AU_25_hog, GRIDDED_AU_25_broiler)

beef_NH3, beef_PM, dairy_NH3, dairy_PM, dairy_VOC, hog_NH3, hog_PM, hog_VO
C, broiler_NH3, broiler_VOC

---------------------------------------------------------------------------
-

NameError                                 Traceback (most recent call last
)

Cell In[989], line 1
----> 1 beef_NH3, beef_PM, dairy_NH3, dairy_PM, dairy_VOC, hog_NH3, hog_PM
, hog_VOC, broiler_NH3, broiler_VOC


NameError: name 'beef_NH3' is not defined

GRIDDED_beef_nh3_25 = GRIDDED_MANURE_AP_25[0]
GRIDDED_beef_pm_25 = GRIDDED_MANURE_AP_25[1]
```

```
GRIDDED_dairy_nh3_25 = GRIDDED_MANURE_AP_25[2]
GRIDDED_dairy_pm_25 = GRIDDED_MANURE_AP_25[3]
GRIDDED_dairy_voc_25 = GRIDDED_MANURE_AP_25[4]
GRIDDED_hog_nh3_25 = GRIDDED_MANURE_AP_25[5]
GRIDDED_hog_pm_25 = GRIDDED_MANURE_AP_25[6]
GRIDDED_hog_voc_25 = GRIDDED_MANURE_AP_25[7]
GRIDDED_broiler_nh3_25 = GRIDDED_MANURE_AP_25[8]
GRIDDED_broiler_voc_25 = GRIDDED_MANURE_AP_25[9]

numpy.savetxt("beef_nh3.csv", GRIDDED_beef_nh3_25, delimiter=",")
numpy.savetxt("beef_pm.csv", GRIDDED_beef_pm_25, delimiter=",")
numpy.savetxt("dairy_nh3.csv", GRIDDED_dairy_nh3_25, delimiter=",")
numpy.savetxt("dairy_pm.csv", GRIDDED_dairy_pm_25, delimiter=",")
numpy.savetxt("dairy_voc.csv", GRIDDED_dairy_voc_25, delimiter=",")
numpy.savetxt("beef_nh3.csv", GRIDDED_beef_nh3_25, delimiter=",")
numpy.savetxt("beef_pm.csv", GRIDDED_beef_pm_25, delimiter=",")
numpy.savetxt("dairy_nh3.csv", GRIDDED_dairy_nh3_25, delimiter=",")
numpy.savetxt("dairy_pm.csv", GRIDDED_dairy_pm_25, delimiter=",")
numpy.savetxt("dairy_voc.csv", GRIDDED_dairy_voc_25, delimiter=",")

#calculating leftover manure

total_beef_manure_25 = beef_manure_25.sum()
total_broiler_manure_25 = broiler_manure_25.sum()
total_dairy_manure_25 = dairy_manure_25.sum()
total_hog_manure_25 = hog_manure_25.sum()
total_fw_25 = fw_25.sum()

leftover_beef_25 = beef_baseline_manure_tonnes - total_beef_manure_25
leftover_broiler_25 = broiler_baseline_manure_tonnes - total_broiler_manur
e_25
leftover_dairy_25 = dairy_baseline_manure_tonnes - total_dairy_manure_25
leftover_hog_25 = hog_baseline_manure_tonnes - total_hog_manure_25
leftover_fw_25 = foodwaste_baseline_tonnes - total_fw_25

total_beef_manure_25

404385.768640104

leftover_beef_25

7955516.231359896

def TOTAL_BEEF_25():
    S1 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBL, fMMS_LB
, leftover_beef_25, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBS, fMMS_SB
```

```
, leftover_beef_25, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBP, fMMS_PB
, leftover_beef_25, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBO, fMMS_OB
, leftover_beef_25, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_DAIRY_25():
    S1 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDL, fMMS_
LD, leftover_dairy_25, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDS, fMMS_
SD, leftover_dairy_25, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDP, fMMS_
PD, leftover_dairy_25, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDO, fMMS_
OD, leftover_dairy_25, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_BROILER_25():
    S1 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrL,
fMMS_LBr, leftover_broiler_25, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrS,
fMMS_SBr, leftover_broiler_25, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrP,
fMMS_PBr, leftover_broiler_25, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrO,
fMMS_OBr, leftover_broiler_25, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_HOG_25():
    S1 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCFHL, fMMS_LH,
leftover_hog_25, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCFHS, fMMS_SH,
leftover_hog_25, tonne2kg, kg2tonne)

    return S1 + S2

def TOTAL_MANURE_CH4_25():
    total_manure = TOTAL_BEEF_25()+TOTAL_DAIRY_25()+TOTAL_BROILER_25()+TOT
AL_HOG_25()
    CO2e = total_manure*GWP_CH4
    mtonnes = CO2e/tonne2mtonne

    return mtonnes
```

```
TOTAL_MANURE_25 = total_beef_manure_25+total_broiler_manure_25+total_dairy
_manure_25+total_hog_manure_25
BEEF_PER_25 = total_beef_manure_25 / TOTAL_MANURE_25
BROILER_PER_25 = total_broiler_manure_25 / TOTAL_MANURE_25
DAIRY_PER_25 = total_dairy_manure_25 / TOTAL_MANURE_25
HOG_PER_25 = total_hog_manure_25 / TOTAL_MANURE_25

def TOTAL_STORAGE_25():
    S1 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCF_storage, B
EEF_PER_25, TOTAL_MANURE_25, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCF_stor
age, BROILER_PER_25, TOTAL_MANURE_25, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCF_storage,
DAIRY_PER_25, TOTAL_MANURE_25, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCF_storage, HOG
_PER_25, TOTAL_MANURE_25, tonne2kg, kg2tonne)
    total_storage = S1+S2+S3+S4
    CO2e = total_storage*GWP_CH4
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

ts_25 = TOTAL_STORAGE_25()

def BEEF_N2O_25():
    D1 = MANURE_N2O(leftover_beef_25, BN2O_direct)
    D2 = MANURE_N2O(leftover_beef_25, BN2O_indirect)

    return D1 + D2

def DAIRY_N2O_25():
    D1 = MANURE_N2O(leftover_dairy_25, DN2O_direct)
    D2 = MANURE_N2O(leftover_dairy_25, DN2O_indirect)

    return D1 + D2

def BROILER_N2O_25():
    D1 = MANURE_N2O(leftover_broiler_25, BrN2O_direct)
    D2 = MANURE_N2O(leftover_broiler_25, BrN2O_indirect)

    return D1 + D2

def HOG_N2O_25():
    D1 = MANURE_N2O(leftover_hog_25, HN2O_direct)
    D2 = MANURE_N2O(leftover_hog_25, HN2O_indirect)

    return D1 + D2
```

```
def TOTAL_MANURE_N2O_25():
    total_manure = BEEF_N2O_25()+DAIRY_N2O_25()+BROILER_N2O_25()+HOG_N2O_2
5() #kg
    tonnes = total_manure/kg2tonne
    CO2e = tonnes*GWP_N2O
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

TOTAL_25_MANURE = TOTAL_MANURE_N2O_25()+TOTAL_MANURE_CH4_25()

TOTAL_25_FW = FW_METHANE(leftover_fw_25)



METHANE_25_FW = FW_JUST_METHANE(leftover_fw_25)
FW_AP_25 = FW_AP(METHANE_25_FW)

PIPELINE_25 = PIPELINE(methane_output_25)

PIPELINE_25_AP = PIPELINE_AP(methane_output_25)

PIPELINE_25_AP

(1415.9114747079423, 364.09152206775656, 12.945476340186902)

PIPELINE_25

0.013557370170358513

PIPELINE_25

0.013557370170358513

STORAGE_AP_25 = STORAGE_AP(TOTAL_N_25)

STORAGE_AP_25

0.012103537315769573

ENERGY_SAVINGS_AP_25 = ENERGY_SAVINGS_AP(methane_output_25)

ENERGY_SAVINGS_AP_25

(2.0881563944367083, 109.0481672650281)

TOTAL_P_25 = TOTAL_P(total_beef_manure_25, total_dairy_manure_25, total_br
oiler_manure_25, total_hog_manure_25)
```

```
TOTAL_N_25 = TOTAL_N(total_beef_manure_25, total_dairy_manure_25, total_br
oiler_manure_25, total_hog_manure_25)

TOTAL_N_25_GRIDDED = TOTAL_N(beef_manure_25, dairy_manure_25, broiler_manu
re_25, hog_manure_25)

STORAGE_AP_25_GRIDDED = STORAGE_AP(TOTAL_N_25_GRIDDED)

TOTAL_N_25

13448.374795299525

TOTAL_N_25_GRIDDED.sum()

13448.374795299527

numpy.savetxt("storage_nh3.csv", STORAGE_AP_25_GRIDDED, delimiter=",")

FERT_AP_25 = FERT_AP(TOTAL_P_25, TOTAL_N_25)

FERT_AP_25

(0.00024220001617297917,
 0.031161908040169824,
 0.018766943831107368,
 0.00035409286682257376,
 8.917894423679634e-06,
 5.5152646529334216e-05)

FERT_AP_25

(0.24220001617297918,
 31.161908040169823,
 18.76694383110737,
 0.35409286682257374,
 0.008917894423679634,
 0.055152646529334215)

fertilizer_replaced_25 = FERT_REPLACED(TOTAL_P_25, TOTAL_N_25)

fertilizer_replaced_25

14295.83140151632
```

Chapter 750% scenario

```
#50% scenario, creating arrays

df_50 = pd.read_excel('/Users/savannahbindas/Documents/UW/Research/Process
ed_Results_MO.xlsx', sheet_name='50%')

methane_output_series_50 = df_50['Methane Output (GJ/y)']
methane_output_50 = methane_output_series_50.values

beef_moved_series_50 = df_50['Beef Moved (tonnes)']
beef_moved_all_50 = beef_moved_series_50.values
beef_moved_50 = beef_moved_all_50[~np.isnan(beef_moved_all_50)]

beef_distance_series_50 = df_50['Beef Distance (km)']
beef_distance_all_50 = beef_distance_series_50.values
beef_distance_50 = beef_distance_all_50[~np.isnan(beef_distance_all_50)]

broiler_moved_series_50 = df_50['Broiler Moved (tonnes)']
broiler_moved_all_50 = broiler_moved_series_50.values
broiler_moved_50 = broiler_moved_all_50[~np.isnan(broiler_moved_all_50)]

broiler_distance_series_50 = df_50['Broiler Distance (km)']
broiler_distance_all_50 = broiler_distance_series_50.values
broiler_distance_50 = broiler_distance_all_50[~np.isnan(broiler_distance_a
ll_50)]

dairy_moved_series_50 = df_50['Dairy Moved (tonnes)']
dairy_moved_all_50 = dairy_moved_series_50.values
dairy_moved_50 = dairy_moved_all_50[~np.isnan(dairy_moved_all_50)]

dairy_distance_series_50 = df_50['Dairy Distance (km)']
dairy_distance_all_50 = dairy_distance_series_50.values
dairy_distance_50 = dairy_distance_all_50[~np.isnan(dairy_distance_all_50)
]

hog_moved_series_50 = df_50['Hog Moved (tonnes)']
hog_moved_all_50 = hog_moved_series_50.values
hog_moved_50 = hog_moved_all_50[~np.isnan(hog_moved_all_50)]

hog_distance_series_50 = df_50['Hog Distance (km)']
hog_distance_all_50 = hog_distance_series_50.values
hog_distance_50 = hog_distance_all_50[~np.isnan(hog_distance_all_50)]

fw_moved_series_50 = df_50['FW Moved (tonnes)']
```

```
fw_moved_all_50 = fw_moved_series_50.values
fw_moved_50 = fw_moved_all_50[~np.isnan(fw_moved_all_50)]

fw_distance_series_50 = df_50['FW Distance (km)']
fw_distance_all_50 = fw_distance_series_50.values
fw_distance_50 = fw_distance_all_50[~np.isnan(fw_distance_all_50)]

ENERGY_SAVINGS_50 = -ENERGY_SAVINGS(methane_output_50)

NATGAS_AP_50 = NATGAS_PROCESSING_AP(methane_output_50)

def BEEF_TRANSPO_50():
    B1 = TRANSPO(beef_moved_50, CC_M, beef_distance_50, transpo_CH4_M, tra
nspo_N2O_M, transpo_CO2_M)

    return B1

def DAIRY_TRANSPO_50():
    B1 = TRANSPO(dairy_moved_50, CC_M, dairy_distance_50, transpo_CH4_M, t
ranspo_N2O_M, transpo_CO2_M)

    return B1

def BROILER_TRANSPO_50():
    B1 = TRANSPO(broiler_moved_50, CC_M, broiler_distance_50, transpo_CH4_
M, transpo_N2O_M, transpo_CO2_M)

    return B1

def HOG_TRANSPO_50():
    B1 = TRANSPO(hog_moved_50, CC_M, hog_distance_50, transpo_CH4_M, trans
po_N2O_M, transpo_CO2_M)

    return B1

def FW_TRANSPO_50():
    B1 = TRANSPO(fw_moved_50, CC_FW, fw_distance_50, transpo_CH4_FW, trans
po_N2O_FW, transpo_CO2_FW)

    return B1

TRANSPO_50 = BEEF_TRANSPO_50()+DAIRY_TRANSPO_50()+BROILER_TRANSPO_50()+HOG
_TRANSPO_50()+FW_TRANSPO_50()

BEEF_TRANSPO_AP_50 = TRANSPO_AP(beef_moved_50, CC_M, beef_distance_50, tra
nspo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VOC_M)
DAIRY_TRANSPO_AP_50 = TRANSPO_AP(dairy_moved_50, CC_M, dairy_distance_50,
transpo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VOC_M
```

```
)
HOG_TRANSPO_AP_50 = TRANSPO_AP(hog_moved_50, CC_M, hog_distance_50, transp
o_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VOC_M)
BROILER_TRANSPO_AP_50 = TRANSPO_AP(broiler_moved_50, CC_M, broiler_distanc
e_50, transpo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo
_VOC_M)
FW_TRANSPO_AP_50 = TRANSPO_AP(fw_moved_50, CC_FW, fw_distance_50, transpo_
NH3_FW, transpo_NOx_FW, transpo_SO2_FW, transpo_PM25_FW, transpo_VOC_FW)

TOTAL_NH3_TRANSPO_50 = BEEF_TRANSPO_AP_50[0]+DAIRY_TRANSPO_AP_50[0]+HOG_TR
ANSPO_AP_50[0]+BROILER_TRANSPO_AP_50[0]+FW_TRANSPO_AP_50[0]
TOTAL_NOx_TRANSPO_50 = BEEF_TRANSPO_AP_50[1]+DAIRY_TRANSPO_AP_50[1]+HOG_TR
ANSPO_AP_50[1]+BROILER_TRANSPO_AP_50[1]+FW_TRANSPO_AP_50[1]
TOTAL_SO2_TRANSPO_50 = BEEF_TRANSPO_AP_50[2]+DAIRY_TRANSPO_AP_50[2]+HOG_TR
ANSPO_AP_50[2]+BROILER_TRANSPO_AP_50[2]+FW_TRANSPO_AP_50[2]
TOTAL_PM25_TRANSPO_50 = BEEF_TRANSPO_AP_50[3]+DAIRY_TRANSPO_AP_50[3]+HOG_T
RANSPO_AP_50[3]+BROILER_TRANSPO_AP_50[3]+FW_TRANSPO_AP_50[3]
TOTAL_VOC_TRANSPO_50 = BEEF_TRANSPO_AP_50[4]+DAIRY_TRANSPO_AP_50[4]+HOG_TR
ANSPO_AP_50[4]+BROILER_TRANSPO_AP_50[4]+FW_TRANSPO_AP_50[4]

UPGRADE_50 = UPGRADE(methane_output_50)

UPGRADE_AP_50 = UPGRADE_AP(methane_output_50)

#leftover manure management emissions
#creating arrays

beef_manure_series_50 = df_50['Beef Manure (tonnes/y)']
beef_manure_50 = beef_manure_series_50.values

broiler_manure_series_50 = df_50['Broiler Manure (tonnes/y)']
broiler_manure_50 = broiler_manure_series_50.values

dairy_manure_series_50 = df_50['Dairy Manure (tonnes/y)']
dairy_manure_50 = dairy_manure_series_50.values

hog_manure_series_50 = df_50['Hog Manure (tonnes/y)']
hog_manure_50 = hog_manure_series_50.values

fw_series_50 = df_50['Food Waste (tonnes/y)']
fw_50 = fw_series_50.values

#calculating leftover manure

total_beef_manure_50 = beef_manure_50.sum()
total_broiler_manure_50 = broiler_manure_50.sum()
total_dairy_manure_50 = dairy_manure_50.sum()
```

```
total_hog_manure_50 = hog_manure_50.sum()
total_fw_50 = fw_50.sum()

leftover_beef_50 = beef_baseline_manure_tonnes - total_beef_manure_50
leftover_broiler_50 = broiler_baseline_manure_tonnes - total_broiler_manur
e_50
leftover_dairy_50 = dairy_baseline_manure_tonnes - total_dairy_manure_50
leftover_hog_50 = hog_baseline_manure_tonnes - total_hog_manure_50
leftover_fw_50 = foodwaste_baseline_tonnes - total_fw_50

AU_50 = ANIMAL_UNIT(total_beef_manure_50, leftover_dairy_50, leftover_hog_
50, leftover_broiler_50)
manure_AP_50 = MANURE_AP(AU_50[0], AU_50[1], AU_50[2], AU_50[3])

manure_AP_50
```

```
(14.226186020463373, 0.8509343338003165, 1.5269740553832245)
```

```
def TOTAL_BEEF_50():
    S1 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBL, fMMS_LB
, leftover_beef_50, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBS, fMMS_SB
, leftover_beef_50, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBP, fMMS_PB
, leftover_beef_50, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBO, fMMS_OB
, leftover_beef_50, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_DAIRY_50():
    S1 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDL, fMMS_
LD, leftover_dairy_50, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDS, fMMS_
SD, leftover_dairy_50, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDP, fMMS_
PD, leftover_dairy_50, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDO, fMMS_
OD, leftover_dairy_50, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_BROILER_50():
    S1 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrL,
fMMS_LBr, leftover_broiler_50, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrS,
fMMS_SBr, leftover_broiler_50, tonne2kg, kg2tonne)
```

```
    S3 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrP,
fMMS_PBr, leftover_broiler_50, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrO,
fMMS_OBr, leftover_broiler_50, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_HOG_50():
    S1 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCFHL, fMMS_LH,
leftover_hog_50, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCFHS, fMMS_SH,
leftover_hog_50, tonne2kg, kg2tonne)

    return S1 + S2

def TOTAL_MANURE_CH4_50():
    total_manure = TOTAL_BEEF_50()+TOTAL_DAIRY_50()+TOTAL_BROILER_50()+TOT
AL_HOG_50()
    CO2e = total_manure*GWP_CH4
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

TOTAL_MANURE_50 = total_beef_manure_50+total_broiler_manure_50+total_dairy
_manure_50+total_hog_manure_50
BEEF_PER_50 = total_beef_manure_50 / TOTAL_MANURE_50
BROILER_PER_50 = total_broiler_manure_50 / TOTAL_MANURE_50
DAIRY_PER_50 = total_dairy_manure_50 / TOTAL_MANURE_50
HOG_PER_50 = total_hog_manure_50 / TOTAL_MANURE_50

def TOTAL_STORAGE_50():
    S1 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCF_storage, B
EEF_PER_50, TOTAL_MANURE_50, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCF_stor
age, BROILER_PER_50, TOTAL_MANURE_50, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCF_storage,
DAIRY_PER_50, TOTAL_MANURE_50, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCF_storage, HOG
_PER_50, TOTAL_MANURE_50, tonne2kg, kg2tonne)
    total_storage = S1+S2+S3+S4
    CO2e = total_storage*GWP_CH4
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

ts_50 = TOTAL_STORAGE_50()
```

```
def BEEF_N2O_50():
    D1 = MANURE_N2O(leftover_beef_50, BN2O_direct)
    D2 = MANURE_N2O(leftover_beef_50, BN2O_indirect)

    return D1 + D2

def DAIRY_N2O_50():
    D1 = MANURE_N2O(leftover_dairy_50, DN2O_direct)
    D2 = MANURE_N2O(leftover_dairy_50, DN2O_indirect)

    return D1 + D2

def BROILER_N2O_50():
    D1 = MANURE_N2O(leftover_broiler_50, BrN2O_direct)
    D2 = MANURE_N2O(leftover_broiler_50, BrN2O_indirect)

    return D1 + D2

def HOG_N2O_50():
    D1 = MANURE_N2O(leftover_hog_50, HN2O_direct)
    D2 = MANURE_N2O(leftover_hog_50, HN2O_indirect)

    return D1 + D2

def TOTAL_MANURE_N2O_50():
    total_manure = BEEF_N2O_50()+DAIRY_N2O_50()+BROILER_N2O_50()+HOG_N2O_5
0() #kg
    tonnes = total_manure/kg2tonne
    CO2e = tonnes*GWP_N2O
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

TOTAL_50_MANURE = TOTAL_MANURE_N2O_50()+TOTAL_MANURE_CH4_50()

TOTAL_50_FW = FW_METHANE(leftover_fw_50)

METHANE_50_FW = FW_JUST_METHANE(leftover_fw_50)
FW_AP_50 = FW_AP(METHANE_50_FW)

PIPELINE_50 = PIPELINE(methane_output_50)

PIPELINE_50_AP = PIPELINE_AP(methane_output_50)

STORAGE_AP_50 = STORAGE_AP(TOTAL_N_50)

ENERGY_SAVINGS_AP_50 = ENERGY_SAVINGS_AP(methane_output_50)

FERT_AP_50 = FERT_AP(TOTAL_P_50, TOTAL_N_50)
```

```
FERT_AP_50

(0.0015988141393622705,
 0.22534627026324808,
 0.13654919214036437,
 0.002576396846626842,
 6.488703169282418e-05,
 0.0004012933270200596)
```

Chapter 875% scenario

```
#75% scenario, creating arrays

df_75 = pd.read_excel('/Users/savannahbindas/Documents/UW/Research/Process
ed_Results_MO.xlsx', sheet_name='75%')

methane_output_series_75 = df_75['Methane Output (GJ/y)']
methane_output_75 = methane_output_series_75.values

beef_moved_series_75 = df_75['Beef Moved (tonnes)']
beef_moved_all_75 = beef_moved_series_75.values
beef_moved_75 = beef_moved_all_75[~np.isnan(beef_moved_all_75)]

beef_distance_series_75 = df_75['Beef Distance (km)']
beef_distance_all_75 = beef_distance_series_75.values
beef_distance_75 = beef_distance_all_75[~np.isnan(beef_distance_all_75)]

broiler_moved_series_75 = df_75['Broiler Moved (tonnes)']
broiler_moved_all_75 = broiler_moved_series_75.values
broiler_moved_75 = broiler_moved_all_75[~np.isnan(broiler_moved_all_75)]

broiler_distance_series_75 = df_75['Broiler Distance (km)']
broiler_distance_all_75 = broiler_distance_series_75.values
broiler_distance_75 = broiler_distance_all_75[~np.isnan(broiler_distance_a
ll_75)]

dairy_moved_series_75 = df_75['Dairy Moved (tonnes)']
dairy_moved_all_75 = dairy_moved_series_75.values
dairy_moved_75 = dairy_moved_all_75[~np.isnan(dairy_moved_all_75)]

dairy_distance_series_75 = df_75['Dairy Distance (km)']
dairy_distance_all_75 = dairy_distance_series_75.values
dairy_distance_75 = dairy_distance_all_75[~np.isnan(dairy_distance_all_75)
]

hog_moved_series_75 = df_75['Hog Moved (tonnes)']
hog_moved_all_75 = hog_moved_series_75.values
hog_moved_75 = hog_moved_all_75[~np.isnan(hog_moved_all_75)]

hog_distance_series_75 = df_75['Hog Distance (km)']
hog_distance_all_75 = hog_distance_series_75.values
hog_distance_75 = hog_distance_all_75[~np.isnan(hog_distance_all_75)]

fw_moved_series_75 = df_75['FW Moved (tonnes)']
```

```
fw_moved_all_75 = fw_moved_series_75.values
fw_moved_75 = fw_moved_all_75[~np.isnan(fw_moved_all_75)]

fw_distance_series_75 = df_75['FW Distance (km)']
fw_distance_all_75 = fw_distance_series_75.values
fw_distance_75 = fw_distance_all_75[~np.isnan(fw_distance_all_75)]

ENERGY_SAVINGS_75 = -ENERGY_SAVINGS(methane_output_75)

NATGAS_AP_75 = NATGAS_PROCESSING_AP(methane_output_75)

def BEEF_TRANSPO_75():
    B1 = TRANSPO(beef_moved_75, CC_M, beef_distance_75, transpo_CH4_M, tra
nspo_N2O_M, transpo_CO2_M)

    return B1

def DAIRY_TRANSPO_75():
    B1 = TRANSPO(dairy_moved_75, CC_M, dairy_distance_75, transpo_CH4_M, t
ranspo_N2O_M, transpo_CO2_M)

    return B1

def BROILER_TRANSPO_75():
    B1 = TRANSPO(broiler_moved_75, CC_M, broiler_distance_75, transpo_CH4_
M, transpo_N2O_M, transpo_CO2_M)

    return B1

def HOG_TRANSPO_75():
    B1 = TRANSPO(hog_moved_75, CC_M, hog_distance_75, transpo_CH4_M, trans
po_N2O_M, transpo_CO2_M)

    return B1

def FW_TRANSPO_75():
    B1 = TRANSPO(fw_moved_75, CC_FW, fw_distance_75, transpo_CH4_FW, trans
po_N2O_FW, transpo_CO2_FW)

    return B1

TRANSPO_75 = BEEF_TRANSPO_75()+DAIRY_TRANSPO_75()+BROILER_TRANSPO_75()+HOG
_TRANSPO_75()+FW_TRANSPO_75()

BEEF_TRANSPO_AP_75 = TRANSPO_AP(beef_moved_75, CC_M, beef_distance_75, tra
nspo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VOC_M)
DAIRY_TRANSPO_AP_75 = TRANSPO_AP(dairy_moved_75, CC_M, dairy_distance_75,
transpo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VOC_M
```

```
)
HOG_TRANSPO_AP_75 = TRANSPO_AP(hog_moved_75, CC_M, hog_distance_75, transp
o_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VOC_M)
BROILER_TRANSPO_AP_75 = TRANSPO_AP(broiler_moved_75, CC_M, broiler_distanc
e_75, transpo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo
_VOC_M)
FW_TRANSPO_AP_75 = TRANSPO_AP(fw_moved_75, CC_FW, fw_distance_75, transpo_
NH3_FW, transpo_NOx_FW, transpo_SO2_FW, transpo_PM25_FW, transpo_VOC_FW)

TOTAL_NH3_TRANSPO_75 = BEEF_TRANSPO_AP_75[0]+DAIRY_TRANSPO_AP_75[0]+HOG_TR
ANSPO_AP_75[0]+BROILER_TRANSPO_AP_75[0]+FW_TRANSPO_AP_75[0]
TOTAL_NOx_TRANSPO_75 = BEEF_TRANSPO_AP_75[1]+DAIRY_TRANSPO_AP_75[1]+HOG_TR
ANSPO_AP_75[1]+BROILER_TRANSPO_AP_75[1]+FW_TRANSPO_AP_75[1]
TOTAL_SO2_TRANSPO_75 = BEEF_TRANSPO_AP_75[2]+DAIRY_TRANSPO_AP_75[2]+HOG_TR
ANSPO_AP_75[2]+BROILER_TRANSPO_AP_75[2]+FW_TRANSPO_AP_75[2]
TOTAL_PM25_TRANSPO_75 = BEEF_TRANSPO_AP_75[3]+DAIRY_TRANSPO_AP_75[3]+HOG_T
RANSPO_AP_75[3]+BROILER_TRANSPO_AP_75[3]+FW_TRANSPO_AP_75[3]
TOTAL_VOC_TRANSPO_75 = BEEF_TRANSPO_AP_75[4]+DAIRY_TRANSPO_AP_75[4]+HOG_TR
ANSPO_AP_75[4]+BROILER_TRANSPO_AP_75[4]+FW_TRANSPO_AP_75[4]

UPGRADE_75 = UPGRADE(methane_output_75)

UPGRADE_AP_75 = UPGRADE_AP(methane_output_75)

#leftover manure management emissions
#creating arrays

beef_manure_series_75 = df_75['Beef Manure (tonnes/y)']
beef_manure_75 = beef_manure_series_75.values

broiler_manure_series_75 = df_75['Broiler Manure (tonnes/y)']
broiler_manure_75 = broiler_manure_series_75.values

dairy_manure_series_75 = df_75['Dairy Manure (tonnes/y)']
dairy_manure_75 = dairy_manure_series_75.values

hog_manure_series_75 = df_75['Hog Manure (tonnes/y)']
hog_manure_75 = hog_manure_series_75.values

fw_series_75 = df_75['Food Waste (tonnes/y)']
fw_75 = fw_series_75.values

#calculating leftover manure

total_beef_manure_75 = beef_manure_75.sum()
total_broiler_manure_75 = broiler_manure_75.sum()
total_dairy_manure_75 = dairy_manure_75.sum()
```

```
total_hog_manure_75 = hog_manure_75.sum()
total_fw_75 = fw_75.sum()

leftover_beef_75 = beef_baseline_manure_tonnes - total_beef_manure_75
leftover_broiler_75 = broiler_baseline_manure_tonnes - total_broiler_manur
e_75
leftover_dairy_75 = dairy_baseline_manure_tonnes - total_dairy_manure_75
leftover_hog_75 = hog_baseline_manure_tonnes - total_hog_manure_75
leftover_fw_75 = foodwaste_baseline_tonnes - total_fw_75

AU_75 = ANIMAL_UNIT(total_beef_manure_75, leftover_dairy_75, leftover_hog_
75, leftover_broiler_75)
manure_AP_75 = MANURE_AP(AU_75[0], AU_75[1], AU_75[2], AU_75[3])

def TOTAL_BEEF_75():
    S1 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBL, fMMS_LB
, leftover_beef_75, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBS, fMMS_SB
, leftover_beef_75, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBP, fMMS_PB
, leftover_beef_75, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBO, fMMS_OB
, leftover_beef_75, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_DAIRY_75():
    S1 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDL, fMMS_
LD, leftover_dairy_75, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDS, fMMS_
SD, leftover_dairy_75, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDP, fMMS_
PD, leftover_dairy_75, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDO, fMMS_
OD, leftover_dairy_75, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_BROILER_75():
    S1 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrL,
fMMS_LBr, leftover_broiler_75, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrS,
fMMS_SBr, leftover_broiler_75, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrP,
fMMS_PBr, leftover_broiler_75, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrO,
fMMS_OBr, leftover_broiler_75, tonne2kg, kg2tonne)
```

```
    return S1 + S2 + S3 + S4

def TOTAL_HOG_75():
    S1 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCFHL, fMMS_LH,
leftover_hog_75, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCFHS, fMMS_SH,
leftover_hog_75, tonne2kg, kg2tonne)

    return S1 + S2

def TOTAL_MANURE_CH4_75():
    total_manure = TOTAL_BEEF_75()+TOTAL_DAIRY_75()+TOTAL_BROILER_75()+TOT
AL_HOG_75()
    CO2e = total_manure*GWP_CH4
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

TOTAL_MANURE_75 = total_beef_manure_75+total_broiler_manure_75+total_dairy
_manure_75+total_hog_manure_75
BEEF_PER_75 = total_beef_manure_75 / TOTAL_MANURE_75
BROILER_PER_75 = total_broiler_manure_75 / TOTAL_MANURE_75
DAIRY_PER_75 = total_dairy_manure_75 / TOTAL_MANURE_75
HOG_PER_75 = total_hog_manure_75 / TOTAL_MANURE_75

def TOTAL_STORAGE_75():
    S1 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCF_storage, B
EEF_PER_75, TOTAL_MANURE_75, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCF_stor
age, BROILER_PER_75, TOTAL_MANURE_75, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCF_storage,
DAIRY_PER_75, TOTAL_MANURE_75, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCF_storage, HOG
_PER_75, TOTAL_MANURE_75, tonne2kg, kg2tonne)
    total_storage = S1+S2+S3+S4
    CO2e = total_storage*GWP_CH4
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

ts_75 = TOTAL_STORAGE_75()

def BEEF_N2O_75():
    D1 = MANURE_N2O(leftover_beef_75, BN2O_direct)
    D2 = MANURE_N2O(leftover_beef_75, BN2O_indirect)
```

```
    return D1 + D2

def DAIRY_N2O_75():
    D1 = MANURE_N2O(leftover_dairy_75, DN2O_direct)
    D2 = MANURE_N2O(leftover_dairy_75, DN2O_indirect)

    return D1 + D2

def BROILER_N2O_75():
    D1 = MANURE_N2O(leftover_broiler_75, BrN2O_direct)
    D2 = MANURE_N2O(leftover_broiler_75, BrN2O_indirect)

    return D1 + D2

def HOG_N2O_75():
    D1 = MANURE_N2O(leftover_hog_75, HN2O_direct)
    D2 = MANURE_N2O(leftover_hog_75, HN2O_indirect)

    return D1 + D2

def TOTAL_MANURE_N2O_75():
    total_manure = BEEF_N2O_75()+DAIRY_N2O_75()+BROILER_N2O_75()+HOG_N2O_7
5() #kg
    tonnes = total_manure/kg2tonne
    CO2e = tonnes*GWP_N2O
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

TOTAL_75_MANURE = TOTAL_MANURE_N2O_75()+TOTAL_MANURE_CH4_75()

TOTAL_75_FW = FW_METHANE(leftover_fw_75)

METHANE_75_FW = FW_JUST_METHANE(leftover_fw_75)
FW_AP_75 = FW_AP(METHANE_75_FW)

PIPELINE_75 = PIPELINE(methane_output_75)

PIPELINE_75_AP = PIPELINE_AP(methane_output_75)

STORAGE_AP_75 = STORAGE_AP(TOTAL_N_75)

ENERGY_SAVINGS_AP_75 = ENERGY_SAVINGS_AP(methane_output_75)

FERT_AP_75 = FERT_AP(TOTAL_P_75, TOTAL_N_75)

FERT_AP_75
```

```
(3.1416199715806226,
 449.1531557499123,
 272.41271563237075,
 5.139856564034753,
 0.1294482393905049,
 0.8005715981557161)
```

```
FERT_AP_75
```

```
(3.1416199715806226,
 449.1531557499123,
 272.41271563237075,
 5.139856564034753,
 0.1294482393905049,
 0.8005715981557161)
```

Chapter 9100% scenario

```
#100% scenario, creating arrays

df_100 = pd.read_excel('/Users/savannahbindas/Documents/UW/Research/Proces
sed_Results_MO.xlsx', sheet_name='100%')

methane_output_series_100 = df_100['Methane Output (GJ/y)']
methane_output_100 = methane_output_series_100.values

beef_moved_series_100 = df_100['Beef Moved (tonnes)']
beef_moved_all_100 = beef_moved_series_100.values
beef_moved_100 = beef_moved_all_100[~np.isnan(beef_moved_all_100)]

beef_distance_series_100 = df_100['Beef Distance (km)']
beef_distance_all_100 = beef_distance_series_100.values
beef_distance_100 = beef_distance_all_100[~np.isnan(beef_distance_all_100)
]

broiler_moved_series_100 = df_100['Broiler Moved (tonnes)']
broiler_moved_all_100 = broiler_moved_series_100.values
broiler_moved_100 = broiler_moved_all_100[~np.isnan(broiler_moved_all_100)
]

broiler_distance_series_100 = df_100['Broiler Distance (km)']
broiler_distance_all_100 = broiler_distance_series_100.values
broiler_distance_100 = broiler_distance_all_100[~np.isnan(broiler_distance
_all_100)]

dairy_moved_series_100 = df_100['Dairy Moved (tonnes)']
dairy_moved_all_100 = dairy_moved_series_100.values
dairy_moved_100 = dairy_moved_all_100[~np.isnan(dairy_moved_all_100)]

dairy_distance_series_100 = df_100['Dairy Distance (km)']
dairy_distance_all_100 = dairy_distance_series_100.values
dairy_distance_100 = dairy_distance_all_100[~np.isnan(dairy_distance_all_1
00)]

hog_moved_series_100 = df_100['Hog Moved (tonnes)']
hog_moved_all_100 = hog_moved_series_100.values
hog_moved_100 = hog_moved_all_100[~np.isnan(hog_moved_all_100)]

hog_distance_series_100 = df_100['Hog Distance (km)']
hog_distance_all_100 = hog_distance_series_100.values
hog_distance_100 = hog_distance_all_100[~np.isnan(hog_distance_all_100)]
```

```
fw_moved_series_100 = df_100['FW Moved (tonnes)']
fw_moved_all_100 = fw_moved_series_100.values
fw_moved_100 = fw_moved_all_100[~np.isnan(fw_moved_all_100)]

fw_distance_series_100 = df_100['FW Distance (km)']
fw_distance_all_100 = fw_distance_series_100.values
fw_distance_100 = fw_distance_all_100[~np.isnan(fw_distance_all_100)]

ENERGY_SAVINGS_100 = -ENERGY_SAVINGS(methane_output_100)

NATGAS_AP_100 = NATGAS_PROCESSING_AP(methane_output_100)

def BEEF_TRANSPO_100():
    B1 = TRANSPO(beef_moved_100, CC_M, beef_distance_100, transpo_CH4_M, t
ranspo_N2O_M, transpo_CO2_M)

    return B1

def DAIRY_TRANSPO_100():
    B1 = TRANSPO(dairy_moved_100, CC_M, dairy_distance_100, transpo_CH4_M,
transpo_N2O_M, transpo_CO2_M)

    return B1

def BROILER_TRANSPO_100():
    B1 = TRANSPO(broiler_moved_100, CC_M, broiler_distance_100, transpo_CH
4_M, transpo_N2O_M, transpo_CO2_M)

    return B1

def HOG_TRANSPO_100():
    B1 = TRANSPO(hog_moved_100, CC_M, hog_distance_100, transpo_CH4_M, tra
nspo_N2O_M, transpo_CO2_M)

    return B1

def FW_TRANSPO_100():
    B1 = TRANSPO(fw_moved_25, CC_FW, fw_distance_25, transpo_CH4_FW, trans
po_N2O_FW, transpo_CO2_FW)

    return B1

TRANSPO_100 = BEEF_TRANSPO_100()+DAIRY_TRANSPO_100()+BROILER_TRANSPO_100()
+HOG_TRANSPO_100()+FW_TRANSPO_100()

TRANSPO_100
```

```
0.02892876309441111

BEEF_TRANSPO_AP_100 = TRANSPO_AP(beef_moved_100, CC_M, beef_distance_100,
transpo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VOC_M
)
DAIRY_TRANSPO_AP_100 = TRANSPO_AP(dairy_moved_100, CC_M, dairy_distance_10
0, transpo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VO
C_M)
HOG_TRANSPO_AP_100 = TRANSPO_AP(hog_moved_100, CC_M, hog_distance_100, tra
nspo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, transpo_VOC_M)
BROILER_TRANSPO_AP_100 = TRANSPO_AP(broiler_moved_100, CC_M, broiler_dista
nce_100, transpo_NH3_M, transpo_NOx_M, transpo_SO2_M, transpo_PM25_M, tran
spo_VOC_M)
FW_TRANSPO_AP_100 = TRANSPO_AP(fw_moved_100, CC_FW, fw_distance_100, trans
po_NH3_FW, transpo_NOx_FW, transpo_SO2_FW, transpo_PM25_FW, transpo_VOC_FW
)

TOTAL_NH3_TRANSPO_100 = BEEF_TRANSPO_AP_100[0]+DAIRY_TRANSPO_AP_100[0]+HOG
_TRANSPO_AP_100[0]+BROILER_TRANSPO_AP_100[0]+FW_TRANSPO_AP_100[0]
TOTAL_NOx_TRANSPO_100 = BEEF_TRANSPO_AP_100[1]+DAIRY_TRANSPO_AP_100[1]+HOG
_TRANSPO_AP_100[1]+BROILER_TRANSPO_AP_100[1]+FW_TRANSPO_AP_100[1]
TOTAL_SO2_TRANSPO_100 = BEEF_TRANSPO_AP_100[2]+DAIRY_TRANSPO_AP_100[2]+HOG
_TRANSPO_AP_100[2]+BROILER_TRANSPO_AP_100[2]+FW_TRANSPO_AP_100[2]
TOTAL_PM25_TRANSPO_100 = BEEF_TRANSPO_AP_100[3]+DAIRY_TRANSPO_AP_100[3]+HO
G_TRANSPO_AP_100[3]+BROILER_TRANSPO_AP_100[3]+FW_TRANSPO_AP_100[3]
TOTAL_VOC_TRANSPO_100 = BEEF_TRANSPO_AP_100[4]+DAIRY_TRANSPO_AP_100[4]+HOG
_TRANSPO_AP_100[4]+BROILER_TRANSPO_AP_100[4]+FW_TRANSPO_AP_100[4]

UPGRADE_100 = UPGRADE(methane_output_100)

UPGRADE_100

0.17941623641096954

UPGRADE_AP_100 = UPGRADE_AP(methane_output_100)

#leftover manure management emissions
#creating arrays

beef_manure_series_100 = df_100['Beef Manure (tonnes/y)']
beef_manure_100 = beef_manure_series_100.values

broiler_manure_series_100 = df_100['Broiler Manure (tonnes/y)']
broiler_manure_100 = broiler_manure_series_100.values

dairy_manure_series_100 = df_100['Dairy Manure (tonnes/y)']
dairy_manure_100 = dairy_manure_series_100.values
```

```
hog_manure_series_100 = df_100['Hog Manure (tonnes/y)']
hog_manure_100 = hog_manure_series_100.values

fw_series_100 = df_100['Food Waste (tonnes/y)']
fw_100 = fw_series_100.values

#calculating leftover manure

total_beef_manure_100 = beef_manure_100.sum()
total_broiler_manure_100 = broiler_manure_100.sum()
total_dairy_manure_100 = dairy_manure_100.sum()
total_hog_manure_100 = hog_manure_100.sum()
total_fw_100 = fw_100.sum()

leftover_beef_manure_100 = beef_baseline_manure_tonnes - total_beef_manure
_100

leftover_beef_manure_100, leftover_dairy_100, leftover_hog_100, leftover_b
roiler_100

(0.006496826186776161, -3.725290298461914e-09, 0.0, 0.0)

AU_100 = ANIMAL_UNIT(leftover_beef_manure_100, leftover_dairy_100, leftove
r_hog_100, leftover_broiler_100)
manure_AP_100 = MANURE_AP(AU_100[0], AU_100[1], AU_100[2], AU_100[3])

def TOTAL_BEEF_100():
    S1 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBL, fMMS_LB
, leftover_beef_100, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBS, fMMS_SB
, leftover_beef_100, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBP, fMMS_PB
, leftover_beef_100, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCFBO, fMMS_OB
, leftover_beef_100, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_DAIRY_100():
    S1 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDL, fMMS_
LD, leftover_dairy_100, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDS, fMMS_
SD, leftover_dairy_100, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDP, fMMS_
PD, leftover_dairy_100, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCFDO, fMMS_
```

```
OD, leftover_dairy_100, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_BROILER_100():
    S1 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrL,
fMMS_LBr, leftover_broiler_100, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrS,
fMMS_SBr, leftover_broiler_100, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrP,
fMMS_PBr, leftover_broiler_100, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCFBrO,
fMMS_OBr, leftover_broiler_100, tonne2kg, kg2tonne)

    return S1 + S2 + S3 + S4

def TOTAL_HOG_100():
    S1 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCFHL, fMMS_LH,
leftover_hog_100, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCFHS, fMMS_SH,
leftover_hog_100, tonne2kg, kg2tonne)

    return S1 + S2

def TOTAL_MANURE_CH4_100():
    total_manure = TOTAL_BEEF_100()+TOTAL_DAIRY_100()+TOTAL_BROILER_100()+
TOTAL_HOG_100()
    CO2e = total_manure*GWP_CH4
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

TOTAL_MANURE_100 = total_beef_manure_100+total_broiler_manure_100+total_da
iry_manure_100+total_hog_manure_100
BEEF_PER_100 = total_beef_manure_100 / TOTAL_MANURE_100
BROILER_PER_100 = total_broiler_manure_100 / TOTAL_MANURE_100
DAIRY_PER_100 = total_dairy_manure_100 / TOTAL_MANURE_100
HOG_PER_100 = total_hog_manure_100 / TOTAL_MANURE_100

def TOTAL_STORAGE_100():
    S1 = MANURE_CH4_EMISSION(VS_beef, B0_beef, CH4_density, MCF_storage, B
EEF_PER_100, TOTAL_MANURE_100, tonne2kg, kg2tonne)
    S2 = MANURE_CH4_EMISSION(VS_broiler, B0_broiler, CH4_density, MCF_stor
age, BROILER_PER_100, TOTAL_MANURE_100, tonne2kg, kg2tonne)
    S3 = MANURE_CH4_EMISSION(VS_dairy, B0_dairy, CH4_density, MCF_storage,
DAIRY_PER_100, TOTAL_MANURE_100, tonne2kg, kg2tonne)
    S4 = MANURE_CH4_EMISSION(VS_hog, B0_hog, CH4_density, MCF_storage, HOG
```

```
_PER_100, TOTAL_MANURE_100, tonne2kg, kg2tonne)
    total_storage = S1+S2+S3+S4
    CO2e = total_storage*GWP_CH4
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

ts_100 = TOTAL_STORAGE_100()

def BEEF_N2O_100():
    D1 = MANURE_N2O(leftover_beef_100, BN2O_direct)
    D2 = MANURE_N2O(leftover_beef_100, BN2O_indirect)

    return D1 + D2

def DAIRY_N2O_100():
    D1 = MANURE_N2O(leftover_dairy_100, DN2O_direct)
    D2 = MANURE_N2O(leftover_dairy_100, DN2O_indirect)

    return D1 + D2

def BROILER_N2O_100():
    D1 = MANURE_N2O(leftover_broiler_100, BrN2O_direct)
    D2 = MANURE_N2O(leftover_broiler_100, BrN2O_indirect)

    return D1 + D2

def HOG_N2O_100():
    D1 = MANURE_N2O(leftover_hog_100, HN2O_direct)
    D2 = MANURE_N2O(leftover_hog_100, HN2O_indirect)

    return D1 + D2

def TOTAL_MANURE_N2O_100():
    total_manure = BEEF_N2O_100()+DAIRY_N2O_100()+BROILER_N2O_100()+HOG_N2
O_100() #kg
    tonnes = total_manure/kg2tonne
    CO2e = tonnes*GWP_N2O
    mtonnes = CO2e/tonne2mtonne

    return mtonnes

TOTAL_100_MANURE = TOTAL_MANURE_N2O_100()+TOTAL_MANURE_CH4_100()

TOTAL_100_FW = FW_METHANE(leftover_fw_100)

METHANE_100_FW = FW_JUST_METHANE(leftover_fw_100)
FW_AP_100 = FW_AP(METHANE_100_FW)
```

```
PIPELINE_100 = PIPELINE(methane_output_100)

PIPELINE_100_AP = PIPELINE_AP(methane_output_100)

STORAGE_AP_100 = STORAGE_AP(TOTAL_N_100)

ENERGY_SAVINGS_AP_100 = ENERGY_SAVINGS_AP(methane_output_100)

FERT_AP_100 = FERT_AP(TOTAL_P_100, TOTAL_N_100)

FERT_AP_100

(0.004722903267547444,
 0.6760205933706791,
 0.41003890813174126,
 0.007736574148450415,
 0.00019484705262764007,
 0.0012050300340313673)

fertilizer_replaced_100 = FERT_REPLACED(TOTAL_P_100, TOTAL_N_100)

fertilizer_replaced_100

306143.0791134294
```

Chapter 10Trying stacked bar chart now

```
import matplotlib.pyplot as plt
import numpy as np
%pylab inline
import pylab
import pandas as pd
import csv
```

```
%pylab is deprecated, use %matplotlib inline and import the required libra
ries.
Populating the interactive namespace from numpy and matplotlib


/Users/savannahbindas/anaconda3/lib/python3.10/site-packages/IPython/core/
magics/pylab.py:162: UserWarning: pylab import has clobbered these variabl
es: ['f', 'pylab', 'e']
`%matplotlib` prevents importing * from pylab and numpy
  warn("pylab import has clobbered these variables: %s"  % clobbered +
```

```
x = ['25%', '50%', '75%', '100%','Baseline']
fig, ax1 = plt.subplots()

Manure = np.array([TOTAL_25_MANURE, TOTAL_50_MANURE, TOTAL_75_MANURE, TOTA
L_100_MANURE, TOTAL_BASELINE_MANURE])
Pipeline_transfer = (PIPELINE_25, PIPELINE_50, PIPELINE_75, PIPELINE_100,0
)
FW = (TOTAL_25_FW, TOTAL_50_FW,TOTAL_75_FW,TOTAL_100_FW,TOTAL_BASELINE_FW)
Upgrading = (UPGRADE_25, UPGRADE_50, UPGRADE_75, UPGRADE_100,0)
Transpo = (TRANSPO_25, TRANSPO_50, TRANSPO_75, TRANSPO_100,0)
Storage = (ts_25, ts_50, ts_75, ts_100, 0)
Energy = (ENERGY_SAVINGS_25, ENERGY_SAVINGS_50, ENERGY_SAVINGS_75, ENERGY_
SAVINGS_100, 0)
Fertilizer = (FERT_25, FERT_50, FERT_75, FERT_100, 0)

ax1.bar(x, Manure, label = 'Manure', color="brown")
ax1.bar(x, Pipeline_transfer, bottom = Manure, label = 'Pipeline Transfer'
, color="gold")
ax1.bar(x, FW, bottom = Manure+Pipeline_transfer, label = 'Food Waste',  c
olor="b")
ax1.bar(x, Upgrading, bottom=Manure+Pipeline_transfer+FW, label = 'Uprgrad
ing', color='darkorange')
ax1.bar(x, Transpo, bottom=Manure+Pipeline_transfer+FW+Upgrading, label =
'Transportation', color="c")
ax1.bar(x, Storage, bottom=Manure+Pipeline_transfer+FW+Upgrading+Transpo,
label = 'Storage', color='purple')
```

```
#ax1.bar(x, Energy, bottom=Manure+Pipeline_transfer+FW+Upgrading+Transpo,
color="k")
ax1.bar(x, Energy, label = 'Energy Offset', color="forestgreen")
ax1.bar(x, Fertilizer, bottom=Energy, label = 'Fertilizer Production Offse
t', color="m")

net_value_baseline = Manure[0] + FW[0]
net_value_run2 = Upgrading[1] + Transpo[1] + Manure[1] + FW[1] + Storage[1
] - (Energy[1] + Fertilizer[1])

plt.xlabel("Scenarios")
ax1.set_ylabel("CO$_{2}$e Emissions (Mtonnes)")
#plt.legend(['Manure Management', 'Pipeline Transfer', 'Food Waste', 'Upgr
ading', 'Transportation','Storage', 'Energy', 'Fertilizer'])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("CO$_{2}$e Emissions (Mtonnes) by Scenario")
plt.savefig('GHG Emissions')
plt.show()
```

```
png
```

```
print((methane_output_series_25.sum()) / pipeline_CO2e_mtonnes_25)
print((methane_output_series_50.sum()) / pipeline_CO2e_mtonnes_50)
print((methane_output_series_75.sum()) / pipeline_CO2e_mtonnes_75)
print((methane_output_series_100.sum()) / pipeline_CO2e_mtonnes_100)
```

```
608974358.974359
608974358.974359
608974358.9743589
608974358.974359
```

```
FERT_100
```

```
-1.0059723383396566
```

```
x = ['Baseline','25%', '50%', '75%', '100%']
fig, ax1 = plt.subplots()

Manure = np.array([TOTAL_BASELINE_MANURE, TOTAL_25_MANURE, TOTAL_50_MANURE
, TOTAL_75_MANURE, TOTAL_100_MANURE])
#Pipeline_transfer = (PIPELINE_25, PIPELINE_50, PIPELINE_75, PIPELINE_100,
0)
FW = (TOTAL_BASELINE_FW, TOTAL_25_FW, TOTAL_50_FW,TOTAL_75_FW,TOTAL_100_FW
)
Upgrading = (0, UPGRADE_25, UPGRADE_50, UPGRADE_75, UPGRADE_100)
Transpo = (0, TRANSPO_25, TRANSPO_50, TRANSPO_75, TRANSPO_100)
Storage = (0, ts_25, ts_50, ts_75, ts_100)
```

```
Energy = (0, ENERGY_SAVINGS_25, ENERGY_SAVINGS_50, ENERGY_SAVINGS_75, ENER
GY_SAVINGS_100)
Fertilizer = (0, FERT_25, FERT_50, FERT_75, FERT_100)

ax1.bar(x, Manure, label = 'Manure', color="tan", width = 0.1)
#ax1.bar(x, Pipeline_transfer, bottom = Manure, label = 'Pipeline Transfer
', color="gold")
ax1.bar(x, FW, bottom = Manure, label = 'Food Waste',  color="navajowhite"
, width = 0.1)
ax1.bar(x, Upgrading, bottom=Manure+FW, label = 'Uprgrading', color='golde
nrod', width = 0.1)
ax1.bar(x, Transpo, bottom=Manure+FW+Upgrading, label = 'Transportation',
color="khaki", width = 0.1)
ax1.bar(x, Storage, bottom=Manure+FW+Upgrading+Transpo, label = 'Storage',
color='orange', width = 0.1)
#ax1.bar(x, Energy, bottom=Manure+Pipeline_transfer+FW+Upgrading+Transpo,
color="k")
ax1.bar(x, Energy, label = 'Natural Gas Production Offset', color="skyblue
", width = 0.1)
ax1.bar(x, Fertilizer, bottom=Energy, label = 'Fertilizer Production Offse
t', color="deepskyblue", width = 0.1)


net_value_baseline = Manure[0] + FW[0]
net_value_25 = Upgrading[1] + Transpo[1] + Manure[1] + FW[1] + Storage[1]
+ (Energy[1] + Fertilizer[1])
net_value_50 = Upgrading[2] + Transpo[2] + Manure[2] + FW[2] + Storage[2]
+ (Energy[2] + Fertilizer[2])
net_value_75 = Upgrading[3] + Transpo[3] + Manure[3] + FW[3] + Storage[3]
+ (Energy[3] + Fertilizer[3])
net_value_100 = Upgrading[4] + Transpo[4] + Manure[4] + FW[4] + Storage[4]
+ (Energy[4] + Fertilizer[4])

netvalues = net_value_baseline, net_value_25, net_value_50, net_value_75,
net_value_100


plt.xlabel("Model Run")
ax1.set_ylabel("CO$_{2}$e Emissions (Mtonnes)")
plt.ylim(-2, 3)
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.axvline(x = 0.5, color = 'black', linestyle = '--', alpha = 1)
plt.axhline(y = 0, color = 'black', alpha = 1)
plt.plot(x, netvalues, marker="o", alpha=1, color="red")
plt.title("CO$_{2}$e Emissions (Mtonnes) by Model Run")
```

```
fig.savefig('GHG emissions', bbox_inches='tight', transparent=True)
plt.show()

png

SOx_25 = (TOTAL_SO2_TRANSPO_25 + UPGRADE_AP_25[3]) - (NATGAS_AP_25 + FERT_
AP_25[0])
PM_25 = (TOTAL_PM25_TRANSPO_25 + UPGRADE_AP_25[4] + FW_AP_25[2] + manure_A
P_25[1]) - (FERT_AP_25[1])
NH3_25 = (TOTAL_NH3_TRANSPO_25 + STORAGE_AP_25 + manure_AP_25[0]) - (FERT_
AP_25[2])
NOx_25 = (TOTAL_NOx_TRANSPO_25 + UPGRADE_AP_25[0] + FW_AP_25[1]) - (FERT_A
P_25[3])
CO_25 = (UPGRADE_AP_25[1] + FW_AP_25[0]) - (FERT_AP_25[4])
VOC_25 = (TOTAL_VOC_TRANSPO_25 + UPGRADE_AP_25[2] + manure_AP_25[2]) - (FE
RT_AP_25[5])

SOx_50 = (TOTAL_SO2_TRANSPO_50 + UPGRADE_AP_50[3]) - (NATGAS_AP_50 + FERT_
AP_50[0])
PM_50 = (TOTAL_PM25_TRANSPO_50 + UPGRADE_AP_50[4] + FW_AP_50[2] + manure_A
P_50[1]) - (FERT_AP_50[1])
NH3_50 = (TOTAL_NH3_TRANSPO_50 + STORAGE_AP_50 + manure_AP_50[0]) - (FERT_
AP_50[2])
NOx_50 = (TOTAL_NOx_TRANSPO_50 + UPGRADE_AP_50[0] + FW_AP_50[1]) - (FERT_A
P_50[3])
CO_50 = (UPGRADE_AP_50[1] + FW_AP_50[0]) - (FERT_AP_50[4])
VOC_50 = (TOTAL_VOC_TRANSPO_50 + UPGRADE_AP_50[2] + manure_AP_50[2]) - (FE
RT_AP_50[5])

SOx_75 = (TOTAL_SO2_TRANSPO_75 + UPGRADE_AP_75[3]) - (NATGAS_AP_75 + FERT_
AP_75[0])
PM_75 = (TOTAL_PM25_TRANSPO_75 + UPGRADE_AP_75[4] + FW_AP_75[2] + manure_A
P_75[1]) - (FERT_AP_75[1])
NH3_75 = (TOTAL_NH3_TRANSPO_75 + STORAGE_AP_75 + manure_AP_75[0]) - (FERT_
AP_75[2])
NOx_75 = (TOTAL_NOx_TRANSPO_75 + UPGRADE_AP_75[0] + FW_AP_75[1]) - (FERT_A
P_75[3])
CO_75 = (UPGRADE_AP_75[1] + FW_AP_75[0]) - (FERT_AP_75[4])
VOC_75 = (TOTAL_VOC_TRANSPO_75 + UPGRADE_AP_75[2] + manure_AP_75[2]) - (FE
RT_AP_75[5])

SOx_100 = (TOTAL_SO2_TRANSPO_100 + UPGRADE_AP_100[3]) - (NATGAS_AP_100 + F
ERT_AP_100[0])
PM_100 = (TOTAL_PM25_TRANSPO_100 + UPGRADE_AP_100[4] + FW_AP_100[2] + manu
re_AP_100[1]) - (FERT_AP_100[1])
NH3_100 = (TOTAL_NH3_TRANSPO_100 + STORAGE_AP_100 + manure_AP_100[0]) - (F
```

```
ERT_AP_100[2])
NOx_100 = (TOTAL_NOx_TRANSPO_100 + UPGRADE_AP_100[0] + FW_AP_100[1]) - (FE
RT_AP_100[3])
CO_100 = (UPGRADE_AP_100[1] + FW_AP_100[0]) - (FERT_AP_100[4])
VOC_100 = (TOTAL_VOC_TRANSPO_100 + UPGRADE_AP_100[2] + manure_AP_100[2]) -
(FERT_AP_100[5])

#SOx = np.array([TOTAL_SO2_TRANSPO_25, UPGRADE_AP_25[3], 0, 0, -NATGAS_AP_
25, -FERT_AP_25[0]])
#PM = np.array([TOTAL_PM25_TRANSPO_25, UPGRADE_AP_25[4], 0, FW_AP_25[2], 0
, -FERT_AP_25[1]])
#NH3 = np.array([TOTAL_NH3_TRANSPO_25, 0, STORAGE_AP_25, 0, 0, -FERT_AP_25
[2]])
#NOx = np.array([TOTAL_NOx_TRANSPO_25, UPGRADE_AP_25[0], 0, FW_AP_25[1], 0
, -FERT_AP_25[3]])
#CO = np.array([0, UPGRADE_AP_25[1], 0, FW_AP_25[0], 0, -FERT_AP_25[4]])
#VOC = np.array([TOTAL_VOC_TRANSPO_25, UPGRADE_AP_25[2], 0, 0, 0, -FERT_AP
_25[5]])

baseline_PM = FW_AP_BASELINE[2] + baseline_manure_AP[1]

PM = np.array([FW_AP_BASELINE[2], baseline_manure_AP[1]])
NH3 = np.array([0, baseline_manure_AP[0]])
NOx = np.array([FW_AP_BASELINE[1], 0])
CO = np.array([FW_AP_BASELINE[0], 0])
VOC = np.array([0, baseline_manure_AP[2]])

x = ['Baseline','25%', '50%', '75%', '100%']
fig, ax1 = plt.subplots()

SOx = np.array([0, SOx_25, SOx_50, SOx_75, SOx_100])
PM = (baseline_PM, PM_25, PM_50,PM_75,PM_100)
NH3 = (baseline_manure_AP[0], NH3_25, NH3_50, NH3_75, NH3_100)
NOx = (FW_AP_BASELINE[1], NOx_25, NOx_50, NOx_75, NOx_100)
CO = (FW_AP_BASELINE[0], CO_25, CO_50, CO_75, CO_100)
VOC = (baseline_manure_AP[2],VOC_25, VOC_50, VOC_75, VOC_100)

ax1.bar(x, SOx, label = 'SOx', color="darkseagreen", width = 0.1)
ax1.bar(x, PM, bottom = SOx, label = 'PM2.5',  color="mediumpurple", width
= 0.1)
ax1.bar(x, NH3, bottom=SOx+PM, label = 'NH3', color='lightpink', width = 0
.1)
ax1.bar(x, NOx, bottom=SOx+PM+NH3, label = 'NOx', color="palevioletred", w
idth = 0.1)
ax1.bar(x, CO, bottom=SOx+PM+NH3+NOx, label = 'CO', color='salmon', width
= 0.1)
ax1.bar(x, VOC, bottom=SOx+PM+NH3+NOx+CO, label = 'VOC', color="forestgree
```

```
n", width = 0.1)


#net_value_baseline = Manure[0] + FW[0]
#net_value_25 = Upgrading[1] + Transpo[1] + Manure[1] + FW[1] + Storage[1]
+ (Energy[1] + Fertilizer[1])
#net_value_50 = Upgrading[2] + Transpo[2] + Manure[2] + FW[2] + Storage[2]
+ (Energy[2] + Fertilizer[2])
#net_value_75 = Upgrading[3] + Transpo[3] + Manure[3] + FW[3] + Storage[3]
+ (Energy[3] + Fertilizer[3])
#net_value_100 = Upgrading[4] + Transpo[4] + Manure[4] + FW[4] + Storage[4
] + (Energy[4] + Fertilizer[4])

#netvalues = net_value_baseline, net_value_25, net_value_50, net_value_75,
net_value_100


plt.xlabel("Model Run")
ax1.set_ylabel("Emissions (ktonnes)")
plt.ylim(-2, 45)
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.axvline(x = 0.5, color = 'black', linestyle = '--', alpha = 1)
plt.axhline(y = 0, color = 'black', alpha = 1)
#plt.plot(x, netvalues, marker="o", alpha=1, color="red")
plt.title("Net Emissions by Scenario (ktonnes)")
fig.savefig('AP_net_emissions', bbox_inches='tight', transparent=True)
plt.show()

png

net_value_100

-0.861770595776586

ts_100

0.06531150616142958

x = ['25%', '50%', '75%', '100%']
fig, ax1 = plt.subplots()

NH3 = np.array([TOTAL_NH3_TRANSPO_25, TOTAL_NH3_TRANSPO_50, TOTAL_NH3_TRAN
SPO_75, TOTAL_NH3_TRANSPO_100])
NOx = np.array([TOTAL_NOx_TRANSPO_25, TOTAL_NOx_TRANSPO_50, TOTAL_NOx_TRAN
SPO_75, TOTAL_NOx_TRANSPO_100])
SO2 = np.array([TOTAL_SO2_TRANSPO_25, TOTAL_SO2_TRANSPO_50, TOTAL_SO2_TRAN
SPO_75, TOTAL_SO2_TRANSPO_100])
```

```
PM25 = np.array([TOTAL_PM25_TRANSPO_25, TOTAL_PM25_TRANSPO_50, TOTAL_PM25_
TRANSPO_75, TOTAL_PM25_TRANSPO_100])
VOC = np.array([TOTAL_VOC_TRANSPO_25, TOTAL_VOC_TRANSPO_50, TOTAL_VOC_TRAN
SPO_75, TOTAL_VOC_TRANSPO_100])

ax1.bar(x, NH3, label = 'NH3', color="brown")
ax1.bar(x, NOx, bottom = NH3, label = 'NOx', color="gold")
ax1.bar(x, SO2, bottom = NH3+NOx, label = 'SO2',  color="blue")
ax1.bar(x, PM25, bottom=NH3+NOx+SO2, label = 'PM2.5', color='darkorange')
ax1.bar(x, VOC, bottom=NH3+NOx+SO2+PM25, label = 'VOCs', color="c")

plt.xlabel("Scenarios")
ax1.set_ylabel("Emissions (Tonnes)")
#plt.legend(['Manure Management', 'Pipeline Transfer', 'Food Waste', 'Upgr
ading', 'Transportation','Storage', 'Energy', 'Fertilizer'])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("Transportation Emissions (Tonnes) by Scenario")
#plt.savefig('GHG Emissions')
plt.show()
```

png

```
x = ['25%', '50%', '75%', '100%']
fig, ax1 = plt.subplots()

SOx = np.array([-FERT_AP_25[0], -FERT_AP_50[0], -FERT_AP_75[0], -FERT_AP_1
00[0]])
PM = np.array([-FERT_AP_25[1], -FERT_AP_50[1], -FERT_AP_75[1], -FERT_AP_10
0[1]])
NH3 = np.array([-FERT_AP_25[2], -FERT_AP_50[2], -FERT_AP_75[2], -FERT_AP_1
00[2]])
NOx = np.array([-FERT_AP_25[3], -FERT_AP_50[3], -FERT_AP_75[3], -FERT_AP_1
00[3]])
CO = np.array([-FERT_AP_25[4], -FERT_AP_50[4], -FERT_AP_75[4], -FERT_AP_10
0[4]])
VOC = np.array([-FERT_AP_25[5], -FERT_AP_50[5], -FERT_AP_75[5], -FERT_AP_1
00[5]])

ax1.bar(x, SOx, label = 'SOx', color="blue")
ax1.bar(x, PM, bottom = SOx, label = 'PM', color="darkorange")
ax1.bar(x, NH3, bottom = SOx+PM, label = 'NH3',  color="brown")
ax1.bar(x, NOx, bottom=SOx+PM+NH3, label = 'NOx', color='gold')
ax1.bar(x, CO, bottom=SOx+PM+NH3+NOx, label = 'CO', color="indigo")
ax1.bar(x, VOC, bottom=SOx+PM+NH3+NOx+CO, label = 'VOC', color="c")

plt.xlabel("Scenarios")
```

```
ax1.set_ylabel("Emissions (Tonnes)")
#plt.legend(['Manure Management', 'Pipeline Transfer', 'Food Waste', 'Upgr
ading', 'Transportation','Storage', 'Energy', 'Fertilizer'])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("Fertilizer Offset (Tonnes) by Scenario")
#plt.savefig('GHG Emissions')
plt.show()
```

```
png
```

```
x = ['25%', '50%', '75%', '100%']
fig, ax1 = plt.subplots()

NOx = np.array([UPGRADE_AP_25[0], UPGRADE_AP_50[0], UPGRADE_AP_75[0], UPGR
ADE_AP_100[0]])
CO = np.array([UPGRADE_AP_25[1], UPGRADE_AP_50[1], UPGRADE_AP_75[1], UPGRA
DE_AP_100[1]])
VOC = np.array([UPGRADE_AP_25[2], UPGRADE_AP_50[2], UPGRADE_AP_75[2], UPGR
ADE_AP_100[2]])
SOx = np.array([UPGRADE_AP_25[3], UPGRADE_AP_50[3], UPGRADE_AP_75[3], UPGR
ADE_AP_100[3]])
PM = np.array([UPGRADE_AP_25[4], UPGRADE_AP_50[4], UPGRADE_AP_75[4], UPGRA
DE_AP_100[4]])

ax1.bar(x, NOx, label = 'NOx', color="gold")
ax1.bar(x, CO, bottom = NOx, label = 'CO', color="indigo")
ax1.bar(x, VOC, bottom = NOx+CO, label = 'VOCs',  color="c")
ax1.bar(x, SOx, bottom=NOx+CO+VOC, label = 'SOx', color='blue')
ax1.bar(x, PM, bottom=NOx+CO+VOC+SOx, label = 'PM', color="darkorange")

plt.xlabel("Scenarios")
ax1.set_ylabel("Emissions (Tonnes)")
#plt.legend(['Manure Management', 'Pipeline Transfer', 'Food Waste', 'Upgr
ading', 'Transportation','Storage', 'Energy', 'Fertilizer'])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("Upgrading Emissions (Tonnes) by Scenario")
#plt.savefig('GHG Emissions')
plt.show()
```

```
png
```

```
x = ['25%', '50%', '75%', '100%', 'Baseline']
fig, ax1 = plt.subplots()

CO = np.array([FW_AP_25[0], FW_AP_50[0], FW_AP_75[0], FW_AP_100[0], FW_AP_
BASELINE[0]])
NOx = np.array([FW_AP_25[1], FW_AP_50[1], FW_AP_75[1], FW_AP_100[1], FW_AP
```

```
_BASELINE[1]])
PM25 = np.array([FW_AP_25[2], FW_AP_50[2], FW_AP_75[2], FW_AP_100[2], FW_A
P_BASELINE[2]])


ax1.bar(x, CO, label = 'CO', color="indigo")
ax1.bar(x, NOx, bottom = CO, label = 'NOx', color="gold")
ax1.bar(x, PM25, bottom = CO+NOx, label = 'PM2.5',  color="darkorange")


plt.xlabel("Scenarios")
ax1.set_ylabel("Emissions (Tonnes)")
#plt.legend(['Manure Management', 'Pipeline Transfer', 'Food Waste', 'Upgr
ading', 'Transportation','Storage', 'Energy', 'Fertilizer'])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("Food Waste Emissions (Tonnes) by Scenario")
#plt.savefig('GHG Emissions')
plt.show()
```

png

```
FW_AP_BASELINE[0]
```

```
3.0454110512184944
```

```
x = ['25%', '50%', '75%', '100%']
fig, ax1 = plt.subplots()

NOx = np.array([PIPELINE_25_AP[0], PIPELINE_50_AP[0], PIPELINE_75_AP[0], P
IPELINE_100_AP[0]])
CO = np.array([PIPELINE_25_AP[1], PIPELINE_50_AP[1], PIPELINE_75_AP[1], PI
PELINE_100_AP[1]])
VOC = np.array([PIPELINE_25_AP[2], PIPELINE_50_AP[2], PIPELINE_75_AP[2], P
IPELINE_100_AP[2]])

ax1.bar(x, NOx, label = 'NOx', color="gold")
ax1.bar(x, CO, bottom = NOx, label = 'CO', color="indigo")
ax1.bar(x, VOC, bottom = NOx+CO, label = 'VOCs',  color="c")

plt.xlabel("Scenarios")
ax1.set_ylabel("Emissions (Tonnes)")
#plt.legend(['Manure Management', 'Pipeline Transfer', 'Food Waste', 'Upgr
ading', 'Transportation','Storage', 'Energy', 'Fertilizer'])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("Pipeline Transfer Emissions (Tonnes) by Scenario")
#plt.savefig('GHG Emissions')
plt.show()
```

```
png
```

```
x = ['25%', '50%', '75%', '100%']
fig, ax1 = plt.subplots()

NH3 = np.array([STORAGE_AP_25, STORAGE_AP_50, STORAGE_AP_75, STORAGE_AP_10
0])

ax1.bar(x, NH3, label = 'NH3', color="brown")


plt.xlabel("Scenarios")
ax1.set_ylabel("Emissions (Tonnes)")
#plt.legend(['Manure Management', 'Pipeline Transfer', 'Food Waste', 'Upgr
ading', 'Transportation','Storage', 'Energy', 'Fertilizer'])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("Digestate Storage Emissions (Tonnes) by Scenario")
#plt.savefig('GHG Emissions')
plt.show()
```

```
png
```

```
x = ['25%', '50%', '75%', '100%']
fig, ax1 = plt.subplots()

SO2 = np.array([-NATGAS_AP_25, -NATGAS_AP_50, -NATGAS_AP_75, -NATGAS_AP_10
0])

ax1.bar(x, SO2, label = 'SO2', color="blue")


plt.xlabel("Scenarios")
ax1.set_ylabel("Emissions (Tonnes)")
#plt.legend(['Manure Management', 'Pipeline Transfer', 'Food Waste', 'Upgr
ading', 'Transportation','Storage', 'Energy', 'Fertilizer'])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("Natural Gas Production Offset Emissions (tonnes) by Scenario")
#plt.savefig('GHG Emissions')
plt.show()
```

```
png
```

```
x = ['25%', '50%', '75%', '100%']
fig, ax1 = plt.subplots()

SO2 = np.array([-ENERGY_SAVINGS_AP_25[0], -ENERGY_SAVINGS_AP_50[0], -ENERG
Y_SAVINGS_AP_75[0], -ENERGY_SAVINGS_AP_100[0]])
```

```
NOx = np.array([-ENERGY_SAVINGS_AP_25[1], -ENERGY_SAVINGS_AP_50[1], -ENERG
Y_SAVINGS_AP_75[1], -ENERGY_SAVINGS_AP_100[1]])

ax1.bar(x, SO2, label = 'SO2', color="blue")
ax1.bar(x, NOx, bottom = SO2, label = 'NOx', color="gold")

plt.xlabel("Scenarios")
ax1.set_ylabel("Emissions (Tonnes)")
#plt.legend(['Manure Management', 'Pipeline Transfer', 'Food Waste', 'Upgr
ading', 'Transportation','Storage', 'Energy', 'Fertilizer'])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("Energy Offset Emissions (Tonnes) by Scenario")
#plt.savefig('GHG Emissions')
plt.show()

png

import pandas as pd
import numpy as np
import matplotlib.pyplot as plot

#nox, nh3, sox, pm, voc, co

_data=[["Transporation", TOTAL_NOx_TRANSPO_25,TOTAL_NH3_TRANSPO_25, TOTAL_
SO2_TRANSPO_25, TOTAL_PM25_TRANSPO_25, TOTAL_VOC_TRANSPO_25, 0],
      ["Upgrading", UPGRADE_AP_25[0], 0, UPGRADE_AP_25[3], UPGRADE_AP_25[4
], UPGRADE_AP_25[2], UPGRADE_AP_25[1]],
      ["Food Waste", FW_AP_25[1], 0, 0, FW_AP_25[2], 0, FW_AP_25[0]],
      ["Digestate Storage", 0, STORAGE_AP_25, 0, 0, 0, 0],
      ["Natural Gas Production Offset", 0, 0, -NATGAS_AP_25, 0, 0, 0],
       ["Fertilizer Production Offset", -FERT_AP_25[0], -FERT_AP_25[1], -F
ERT_AP_25[2], -FERT_AP_25[3], -FERT_AP_25[4], -FERT_AP_25[5]]]

color = {"NOx":"gold",
        "CO":"indigo",
         "VOCs" : "c",
         "PM" : "darkorange",
         "SOx" : "blue",
         "NH3" : "brown"
        }

# Plotting multiple groups

_df=pd.DataFrame(_data, columns=["Emissions Source", "NOx", "NH3", "SOx",
"PM", "VOCs", "CO"])
_df.plot.bar(x="Emissions Source", y=["NOx", "NH3", "SOx", "PM", "VOCs", "
```

```
CO"],figsize=(8,5), color=color, width = 0.9,  ylabel = "Emissions (tonnes
)", title = "25% Run")
```

```
<Axes: title={'center': '25% Run'}, xlabel='Emissions Source', ylabel='Emi
ssions (tonnes)'>
```

```
png
```

```
FERT_AP_25
```

```
(5519.689362090402,
 0.24220001617297918,
 15.23815619947522,
 153.15390078169537,
 0.35409286682257374,
 0.008917894423679634,
 0.055152646529334215)
```

```
NH3_t, PM_t, VOC_t
```

```
import pandas as pd

x = ['Transporation', 'Upgrading', 'Digestate Storage', 'Food Waste','Natu
ral Gas Production Offset', 'Fertilizer Production Offset']
fig, ax1 = plt.subplots()

X_axis = np.arange(len(x))

SOx = np.array([TOTAL_SO2_TRANSPO_25, UPGRADE_AP_25[3], 0, 0, -NATGAS_AP_2
5, -FERT_AP_25[0]])
PM = np.array([TOTAL_PM25_TRANSPO_25, UPGRADE_AP_25[4], 0, FW_AP_25[2], 0,
-FERT_AP_25[1]])
NH3 = np.array([TOTAL_NH3_TRANSPO_25, 0, STORAGE_AP_25, 0, 0, -FERT_AP_25[
2]])
NOx = np.array([TOTAL_NOx_TRANSPO_25, UPGRADE_AP_25[0], 0, FW_AP_25[1], 0,
-FERT_AP_25[3]])
CO = np.array([0, UPGRADE_AP_25[1], 0, FW_AP_25[0], 0, -FERT_AP_25[4]])
VOC = np.array([TOTAL_VOC_TRANSPO_25, UPGRADE_AP_25[2], 0, 0, 0, -FERT_AP_
25[5]])


ax1.bar(x, SOx, label = 'SOx', color="darkseagreen")
ax1.bar(x, PM, bottom = SOx, label = 'PM', color="mediumpurple")
ax1.bar(x, NH3, bottom = SOx+PM, label = 'NH3',  color="lightpink")
ax1.bar(x, NOx, bottom=SOx+PM+NH3, label = 'NOx', color="palevioletred")
```

```
ax1.bar(x, CO, bottom=SOx+PM+NH3+NOx, label = 'CO', color="salmon")
ax1.bar(x, VOC, bottom=SOx+PM+NH3+NOx+CO, label = 'VOC', color="forestgree
n")

plt.xlabel("Model Runs")
ax1.set_ylabel("Emissions (tonnes)")
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.xticks(rotation=90)
plt.title("25% Run")
plt.axhline(y = 0, color = 'black', linestyle = '-')
fig.savefig('AP emissions, 25 run', bbox_inches='tight', transparent=True)
plt.show()
```

png

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Sample data
x = ['Manure Management']
fig, ax1 = plt.subplots()

X_axis = np.arange(len(x))

PM = np.array([baseline_manure_AP[1]])
NH3 = np.array([baseline_manure_AP[0]])
NOx = np.array([0])
CO = np.array([0])
VOC = np.array([baseline_manure_AP[2]])

ax1.bar(x, PM, label='PM', color="mediumpurple")
ax1.bar(x, NH3, bottom=PM, label='NH3', color="lightpink")
ax1.bar(x, NOx, bottom=PM+NH3, label='NOx', color="palevioletred")
ax1.bar(x, CO, bottom=PM+NH3+NOx, label='CO', color="salmon")
ax1.bar(x, VOC, bottom=PM+NH3+NOx+CO, label='VOC', color="forestgreen")



ax1.set_ylabel("Emissions (tonnes)", fontsize=14)
ax1.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0, fo
ntsize=14)
#plt.xticks(rotation=90)
plt.title("Baseline", fontsize=14)
ax1.set_xticklabels(x, fontsize=14)
ax1.axhline(y=0, color='black', linestyle='-')
```

```
#FW_AP_BASELINE, baseline_manure_AP
#CO, NOx, PM25
#NH3, PM, VOC


fig.tight_layout()
fig.savefig('AP emissions, MM baseline', bbox_inches='tight', transparent=
True)
plt.show()
```

```
/var/folders/nv/39y1lpjd48l8wwycln0ybvkh0000gn/T/ipykernel_1852/1016991518
.py:29: UserWarning: FixedFormatter should only be used together with Fixe
dLocator
  ax1.set_xticklabels(x, fontsize=14)
```

```
png
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Sample data
x = ['Food Waste']
fig, ax1 = plt.subplots()

#X_axis = np.arange(len(x))

PM = np.array([FW_AP_BASELINE[2]])
NH3 = np.array([0])
NOx = np.array([FW_AP_BASELINE[1]])
CO = np.array([FW_AP_BASELINE[0]])
VOC = np.array([0])

ax1.bar(x, PM, label='PM', color="mediumpurple")
ax1.bar(x, NH3, bottom=PM, label='NH3', color="lightpink")
ax1.bar(x, NOx, bottom=PM+NH3, label='NOx', color="palevioletred")
ax1.bar(x, CO, bottom=PM+NH3+NOx, label='CO', color="salmon")
ax1.bar(x, VOC, bottom=PM+NH3+NOx+CO, label='VOC', color="forestgreen")

#ax1.set_xlabel("Emissions Sources")
ax1.set_ylabel("Emissions (ktonnes)", fontsize=14)
ax1.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0, fo
ntsize=14)
#plt.xticks(rotation=90)
plt.title("Baseline", fontsize=14)
ax1.set_xticklabels(x, fontsize=14)
```

```
ax1.axhline(y=0, color='black', linestyle='-')
plt.ylim(0, 0.150)


#FW_AP_BASELINE, baseline_manure_AP
#CO, NOx, PM25
#NH3, PM, VOC


fig.tight_layout()
fig.savefig('FW baseline, AP', bbox_inches='tight', transparent=True)
plt.show()
```

```
/var/folders/nv/39y1lpjd48l8wwycln0ybvkh0000gn/T/ipykernel_1852/2638509203
.py:28: UserWarning: FixedFormatter should only be used together with Fixe
dLocator
  ax1.set_xticklabels(x, fontsize=14)
```

```
png
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Sample data for the primary y-axis
x = ['Food Waste', 'Manure']
fig, ax1 = plt.subplots()

X_axis = np.arange(len(x))

PM = np.array([FW_AP_BASELINE[2], baseline_manure_AP[1]])
NH3 = np.array([0, baseline_manure_AP[0]])
NOx = np.array([FW_AP_BASELINE[1], 0])
CO = np.array([FW_AP_BASELINE[0], 0])
VOC = np.array([0, baseline_manure_AP[2]])

ax1.bar(X_axis, PM, 0.4, label='PM', color="mediumpurple")
ax1.bar(X_axis, NH3, 0.4, bottom=PM, label='NH3', color="lightpink")
ax1.bar(X_axis, NOx, 0.4, bottom=PM+NH3, label='NOx', color="palevioletred
")
ax1.bar(X_axis, CO, 0.4, bottom=PM+NH3+NOx, label='CO', color="salmon")
ax1.bar(X_axis, VOC, 0.4, bottom=PM+NH3+NOx+CO, label='VOC', color="forest
green")

ax1.set_xlabel("Emission Sources")
ax1.set_ylabel("Emissions (ktonnes)")
ax1.legend(bbox_to_anchor=(1.5, 1), loc='upper left', borderaxespad=0)
```

```
plt.xticks(X_axis, x, rotation=90)
plt.title("Baseline")
ax1.axhline(y=0, color='black', linestyle='-')

# Creating a second y-axis
#ax2 = ax1.twinx()

# Data for the secondary y-axis
#PM_2 = manure_AP_25[1]
#NH3_2 = manure_AP_25[0]
#VOC_2 = manure_AP_25[2]

#secondary_x = 6

# Summing the values for the single bar
#secondary_values = np.array([PM_2, NH3_2, VOC_2])
#bottoms = np.zeros_like(secondary_values)
#bottoms[1:] = np.cumsum(secondary_values[:-1])

#ax2.bar(6, secondary_values[0], 0.4, label='PM_2', color='mediumpurple')
#ax2.bar(6, secondary_values[1], 0.4, bottom=secondary_values[0], label='N
H3_2', color='lightpink')
#ax2.bar(6, secondary_values[2], 0.4, bottom=secondary_values[0] + seconda
ry_values[1], label='VOC_2', color='forestgreen')

#ax2.set_ylabel('Emissions (ktonnes)')

# Setting limits for the secondary y-axis
#primary_y_min, primary_y_max = ax1.get_ylim()
#secondary_y_max = (PM_2 + NH3_2 + VOC_2) * 1.1
#secondary_y_min = -secondary_y_max * abs(primary_y_min / primary_y_max)

#ax2.set_ylim(secondary_y_min, secondary_y_max)

# Ensure ticks align at zero
#primary_ticks = ax1.get_yticks()
#secondary_ticks = np.linspace(secondary_y_min, secondary_y_max, len(prima
ry_ticks))
#ax2.set_yticks(secondary_ticks)

#plt.xticks(np.append(X_axis, secondary_x), x + ['Manure Management'], rot
ation=90)

#plt.axvline(x=5.5, color='black', linestyle='--')

fig.tight_layout()
```

```
fig.savefig('baseline_AP_1', bbox_inches='tight', transparent=True)
plt.show()
```

png

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Sample data for the primary y-axis
x = ['Transportation', 'Upgrading', 'Digestate Storage', 'Food Waste', 'Na
tural Gas Production Offset', 'Fertilizer Production Offset']
fig, ax1 = plt.subplots()

X_axis = np.arange(len(x))

SOx = np.array([TOTAL_SO2_TRANSPO_25, UPGRADE_AP_25[3], 0, 0, -NATGAS_AP_2
5, -FERT_AP_25[0]])
PM = np.array([TOTAL_PM25_TRANSPO_25, UPGRADE_AP_25[4], 0, FW_AP_25[2], 0,
-FERT_AP_25[1]])
NH3 = np.array([TOTAL_NH3_TRANSPO_25, 0, STORAGE_AP_25, 0, 0, -FERT_AP_25[
2]])
NOx = np.array([TOTAL_NOx_TRANSPO_25, UPGRADE_AP_25[0], 0, FW_AP_25[1], 0,
-FERT_AP_25[3]])
CO = np.array([0, UPGRADE_AP_25[1], 0, FW_AP_25[0], 0, -FERT_AP_25[4]])
VOC = np.array([TOTAL_VOC_TRANSPO_25, UPGRADE_AP_25[2], 0, 0, 0, -FERT_AP_
25[5]])

ax1.bar(X_axis, SOx, 0.4, label='SOx', color="darkseagreen")
ax1.bar(X_axis, PM, 0.4, bottom=SOx, label='PM', color="mediumpurple")
ax1.bar(X_axis, NH3, 0.4, bottom=SOx+PM, label='NH3', color="lightpink")
ax1.bar(X_axis, NOx, 0.4, bottom=SOx+PM+NH3, label='NOx', color="paleviole
tred")
ax1.bar(X_axis, CO, 0.4, bottom=SOx+PM+NH3+NOx, label='CO', color="salmon"
)
ax1.bar(X_axis, VOC, 0.4, bottom=SOx+PM+NH3+NOx+CO, label='VOC', color="fo
restgreen")

ax1.set_xlabel("Emission Sources")
ax1.set_ylabel("Emissions (tonnes)")
ax1.legend(bbox_to_anchor=(1.5, 1), loc='upper left', borderaxespad=0)
plt.xticks(X_axis, x, rotation=90)
plt.title("25% Run")
ax1.axhline(y=0, color='black', linestyle='-')

# Creating a second y-axis
```

```
ax2 = ax1.twinx()

# Data for the secondary y-axis
PM_2 = manure_AP_25[1]
NH3_2 = manure_AP_25[0]
VOC_2 = manure_AP_25[2]

secondary_x = 6

# Summing the values for the single bar
secondary_values = np.array([PM_2, NH3_2, VOC_2])
bottoms = np.zeros_like(secondary_values)
bottoms[1:] = np.cumsum(secondary_values[:-1])

ax2.bar(6, secondary_values[0], 0.4, label='PM_2', color='mediumpurple')
ax2.bar(6, secondary_values[1], 0.4, bottom=secondary_values[0], label='NH
3_2', color='lightpink')
ax2.bar(6, secondary_values[2], 0.4, bottom=secondary_values[0] + secondar
y_values[1], label='VOC_2', color='forestgreen')

ax2.set_ylabel('Emissions (ktonnes)')

# Setting limits for the secondary y-axis
primary_y_min, primary_y_max = ax1.get_ylim()
secondary_y_max = (PM_2 + NH3_2 + VOC_2) * 1.1
secondary_y_min = -secondary_y_max * abs(primary_y_min / primary_y_max)

ax2.set_ylim(secondary_y_min, secondary_y_max)

# Ensure ticks align at zero
primary_ticks = ax1.get_yticks()
secondary_ticks = np.linspace(secondary_y_min, secondary_y_max, len(primar
y_ticks))
ax2.set_yticks(secondary_ticks)

plt.xticks(np.append(X_axis, secondary_x), x + ['Manure Management'], rota
tion=90)

plt.axvline(x=5.5, color='black', linestyle='--')

fig.tight_layout()
fig.savefig('AP emissions, 25 run', bbox_inches='tight', transparent=True)
plt.show()

png
```

```
net_AP_25 = (TOTAL_SO2_TRANSPO_25+UPGRADE_AP_25[3]+TOTAL_PM25_TRANSPO_25+U
PGRADE_AP_25[4]+TOTAL_NH3_TRANSPO_25+STORAGE_AP_25+TOTAL_NOx_TRANSPO_25+UP
GRADE_AP_25[0]+UPGRADE_AP_25[1]+TOTAL_VOC_TRANSPO_25+UPGRADE_AP_25[2])-(FE
RT_AP_25[0]+NATGAS_AP_25+FERT_AP_25[1]+FERT_AP_25[2]+FERT_AP_25[3]+FERT_AP
_25[4]+FERT_AP_25[5])
```

```
net_AP_25
```

```
41.832215042162446
```

```
net_AP_50 = (TOTAL_SO2_TRANSPO_50+UPGRADE_AP_50[3]+TOTAL_PM25_TRANSPO_50+U
PGRADE_AP_50[4]+FW_AP_50[2]+TOTAL_NH3_TRANSPO_50+STORAGE_AP_50+TOTAL_NOx_T
RANSPO_50+UPGRADE_AP_50[0]+FW_AP_50[1]+UPGRADE_AP_50[1]+FW_AP_50[0]+TOTAL_
VOC_TRANSPO_50+UPGRADE_AP_50[2]+manure_AP_50[0]+manure_AP_50[1]+manure_AP_
50[2])-(FERT_AP_50[0]+NATGAS_AP_50+FERT_AP_50[1]+FERT_AP_50[2]+FERT_AP_50[
3]+FERT_AP_50[4]+FERT_AP_50[5])
```

```
net_AP_50
```

```
16458.444507867396
```

```
net_AP_75 = (TOTAL_SO2_TRANSPO_75+UPGRADE_AP_75[3]+TOTAL_PM25_TRANSPO_75+U
PGRADE_AP_75[4]+FW_AP_75[2]+TOTAL_NH3_TRANSPO_75+STORAGE_AP_75+TOTAL_NOx_T
RANSPO_75+UPGRADE_AP_75[0]+FW_AP_75[1]+UPGRADE_AP_75[1]+FW_AP_75[0]+TOTAL_
VOC_TRANSPO_75+UPGRADE_AP_75[2]+manure_AP_75[0]+manure_AP_75[1]+manure_AP_
75[2])-(FERT_AP_75[0]+NATGAS_AP_75+FERT_AP_75[1]+FERT_AP_75[2]+FERT_AP_75[
3]+FERT_AP_75[4]+FERT_AP_75[5])
```

```
net_AP_100 = (TOTAL_SO2_TRANSPO_100+UPGRADE_AP_100[3]+TOTAL_PM25_TRANSPO_1
00+UPGRADE_AP_100[4]+FW_AP_100[2]+TOTAL_NH3_TRANSPO_100+STORAGE_AP_100+TOT
AL_NOx_TRANSPO_100+UPGRADE_AP_100[0]+FW_AP_100[1]+UPGRADE_AP_100[1]+FW_AP_
100[0]+TOTAL_VOC_TRANSPO_100+UPGRADE_AP_100[2])-(FERT_AP_100[0]+NATGAS_AP_
100+FERT_AP_100[1]+FERT_AP_100[2]+FERT_AP_100[3]+FERT_AP_100[4]+FERT_AP_10
0[5])
```

```
net_AP_100
```

```
-510.9684956307939
```

```
FERT_AP_100[2]
```

```
410.03890813174127

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Sample data for the primary y-axis
x = ['Transportation', 'Upgrading', 'Digestate Storage', 'Food Waste', 'Na
tural Gas Production Offset', 'Fertilizer Production Offset']
fig, ax1 = plt.subplots()

X_axis = np.arange(len(x))

SOx = np.array([TOTAL_SO2_TRANSPO_50, UPGRADE_AP_50[3], 0, 0, -NATGAS_AP_5
0, -FERT_AP_50[0]])
PM = np.array([TOTAL_PM25_TRANSPO_50, UPGRADE_AP_50[4], 0, FW_AP_50[2], 0,
-FERT_AP_50[1]])
NH3 = np.array([TOTAL_NH3_TRANSPO_50, 0, STORAGE_AP_50, 0, 0, -FERT_AP_50[
2]])
NOx = np.array([TOTAL_NOx_TRANSPO_50, UPGRADE_AP_50[0], 0, FW_AP_50[1], 0,
-FERT_AP_50[3]])
CO = np.array([0, UPGRADE_AP_50[1], 0, FW_AP_50[0], 0, -FERT_AP_50[4]])
VOC = np.array([TOTAL_VOC_TRANSPO_50, UPGRADE_AP_50[2], 0, 0, 0, -FERT_AP_
50[5]])

ax1.bar(X_axis, SOx, 0.4, label='SOx', color="darkseagreen")
ax1.bar(X_axis, PM, 0.4, bottom=SOx, label='PM', color="mediumpurple")
ax1.bar(X_axis, NH3, 0.4, bottom=SOx+PM, label='NH3', color="lightpink")
ax1.bar(X_axis, NOx, 0.4, bottom=SOx+PM+NH3, label='NOx', color="paleviole
tred")
ax1.bar(X_axis, CO, 0.4, bottom=SOx+PM+NH3+NOx, label='CO', color="salmon"
)
ax1.bar(X_axis, VOC, 0.4, bottom=SOx+PM+NH3+NOx+CO, label='VOC', color="fo
restgreen")

ax1.set_xlabel("Emission Sources")
ax1.set_ylabel("Emissions (tonnes)")
ax1.legend(bbox_to_anchor=(1.5, 1), loc='upper left', borderaxespad=0)
plt.xticks(X_axis, x, rotation=90)
plt.title("50% Run")
ax1.axhline(y=0, color='black', linestyle='-')

# Creating a second y-axis
ax2 = ax1.twinx()

# Data for the secondary y-axis
```

```
PM_2 = manure_AP_50[1]
NH3_2 = manure_AP_50[0]
VOC_2 = manure_AP_50[2]

secondary_x = 6

# Summing the values for the single bar
secondary_values = np.array([PM_2, NH3_2, VOC_2])
bottoms = np.zeros_like(secondary_values)
bottoms[1:] = np.cumsum(secondary_values[:-1])

ax2.bar(6, secondary_values[0], 0.4, label='PM_2', color='mediumpurple')
ax2.bar(6, secondary_values[1], 0.4, bottom=secondary_values[0], label='NH
3_2', color='lightpink')
ax2.bar(6, secondary_values[2], 0.4, bottom=secondary_values[0] + secondar
y_values[1], label='VOC_2', color='forestgreen')

ax2.set_ylabel('Emissions (ktonnes)')

# Setting limits for the secondary y-axis
primary_y_min, primary_y_max = ax1.get_ylim()
secondary_y_max = (PM_2 + NH3_2 + VOC_2) * 1.1
secondary_y_min = -secondary_y_max * abs(primary_y_min / primary_y_max)

ax2.set_ylim(secondary_y_min, secondary_y_max)

# Ensure ticks align at zero
primary_ticks = ax1.get_yticks()
secondary_ticks = np.linspace(secondary_y_min, secondary_y_max, len(primar
y_ticks))
ax2.set_yticks(secondary_ticks)

plt.xticks(np.append(X_axis, secondary_x), x + ['Manure Management'], rota
tion=90)

plt.axvline(x=5.5, color='black', linestyle='--')

fig.tight_layout()
fig.savefig('AP emissions, 50 run', bbox_inches='tight', transparent=True)
plt.show()

png

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
# Sample data for the primary y-axis
x = ['Transportation', 'Upgrading', 'Digestate Storage', 'Food Waste', 'Na
tural Gas Production Offset', 'Fertilizer Production Offset']
fig, ax1 = plt.subplots()

X_axis = np.arange(len(x))

SOx = np.array([TOTAL_SO2_TRANSPO_75, UPGRADE_AP_75[3], 0, 0, -NATGAS_AP_7
5, -FERT_AP_75[0]])
PM = np.array([TOTAL_PM25_TRANSPO_75, UPGRADE_AP_75[4], 0, FW_AP_75[2], 0,
-FERT_AP_75[1]])
NH3 = np.array([TOTAL_NH3_TRANSPO_75, 0, STORAGE_AP_75, 0, 0, -FERT_AP_75[
2]])
NOx = np.array([TOTAL_NOx_TRANSPO_75, UPGRADE_AP_75[0], 0, FW_AP_75[1], 0,
-FERT_AP_75[3]])
CO = np.array([0, UPGRADE_AP_75[1], 0, FW_AP_75[0], 0, -FERT_AP_75[4]])
VOC = np.array([TOTAL_VOC_TRANSPO_75, UPGRADE_AP_75[2], 0, 0, 0, -FERT_AP_
75[5]])

ax1.bar(X_axis, SOx, 0.4, label='SOx', color="darkseagreen")
ax1.bar(X_axis, PM, 0.4, bottom=SOx, label='PM', color="mediumpurple")
ax1.bar(X_axis, NH3, 0.4, bottom=SOx+PM, label='NH3', color="lightpink")
ax1.bar(X_axis, NOx, 0.4, bottom=SOx+PM+NH3, label='NOx', color="paleviole
tred")
ax1.bar(X_axis, CO, 0.4, bottom=SOx+PM+NH3+NOx, label='CO', color="salmon"
)
ax1.bar(X_axis, VOC, 0.4, bottom=SOx+PM+NH3+NOx+CO, label='VOC', color="fo
restgreen")

ax1.set_xlabel("Emission Sources")
ax1.set_ylabel("Emissions (tonnes)")
ax1.legend(bbox_to_anchor=(1.5, 1), loc='upper left', borderaxespad=0)
plt.xticks(X_axis, x, rotation=90)
plt.title("75% Run")
ax1.axhline(y=0, color='black', linestyle='-')

# Creating a second y-axis
ax2 = ax1.twinx()

# Data for the secondary y-axis
PM_2 = manure_AP_75[1]
NH3_2 = manure_AP_75[0]
VOC_2 = manure_AP_75[2]

secondary_x = 6
```

```
# Summing the values for the single bar
secondary_values = np.array([PM_2, NH3_2, VOC_2])
bottoms = np.zeros_like(secondary_values)
bottoms[1:] = np.cumsum(secondary_values[:-1])

ax2.bar(6, secondary_values[0], 0.4, label='PM_2', color='mediumpurple')
ax2.bar(6, secondary_values[1], 0.4, bottom=secondary_values[0], label='NH
3_2', color='lightpink')
ax2.bar(6, secondary_values[2], 0.4, bottom=secondary_values[0] + secondar
y_values[1], label='VOC_2', color='forestgreen')

ax2.set_ylabel('Emissions (ktonnes)')

# Setting limits for the secondary y-axis
primary_y_min, primary_y_max = ax1.get_ylim()
secondary_y_max = (PM_2 + NH3_2 + VOC_2) * 1.1
secondary_y_min = -secondary_y_max * abs(primary_y_min / primary_y_max)

ax2.set_ylim(secondary_y_min, secondary_y_max)

# Ensure ticks align at zero
primary_ticks = ax1.get_yticks()
secondary_ticks = np.linspace(secondary_y_min, secondary_y_max, len(primar
y_ticks))
ax2.set_yticks(secondary_ticks)

plt.xticks(np.append(X_axis, secondary_x), x + ['Manure Management'], rota
tion=90)

plt.axvline(x=5.5, color='black', linestyle='--')

fig.tight_layout()
fig.savefig('AP emissions, 75 run', bbox_inches='tight', transparent=True)
plt.show()
```

png

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Sample data for the primary y-axis
x = ['Transportation', 'Upgrading']
fig, ax1 = plt.subplots()
```

```
X_axis = np.arange(len(x))

SOx = np.array([TOTAL_SO2_TRANSPO_100, UPGRADE_AP_100[3]])
PM = np.array([TOTAL_PM25_TRANSPO_100, UPGRADE_AP_100[4]])
NH3 = np.array([TOTAL_NH3_TRANSPO_100, 0])
NOx = np.array([TOTAL_NOx_TRANSPO_100, UPGRADE_AP_100[0]])
CO = np.array([0, UPGRADE_AP_100[1]])
VOC = np.array([TOTAL_VOC_TRANSPO_100, UPGRADE_AP_100[2]])

ax1.bar(X_axis, SOx, 0.4, label='SOx', color="darkseagreen")
ax1.bar(X_axis, PM, 0.4, bottom=SOx, label='PM', color="mediumpurple")
ax1.bar(X_axis, NH3, 0.4, bottom=SOx+PM, label='NH3', color="lightpink")
ax1.bar(X_axis, NOx, 0.4, bottom=SOx+PM+NH3, label='NOx', color="paleviole
tred")
ax1.bar(X_axis, CO, 0.4, bottom=SOx+PM+NH3+NOx, label='CO', color="salmon"
)
ax1.bar(X_axis, VOC, 0.4, bottom=SOx+PM+NH3+NOx+CO, label='VOC', color="fo
restgreen")

ax1.set_xlabel("Emission Sources")
ax1.set_ylabel("Emissions (tonnes)")
ax1.legend(bbox_to_anchor=(1.5, 1), loc='upper left', borderaxespad=0)
plt.xticks(X_axis, x, rotation=90)
plt.title("100% Run")
ax1.axhline(y=0, color='black', linestyle='-')

# Creating a second y-axis
#ax2 = ax1.twinx()

# Data for the secondary y-axis
PM_2 = 0
NH3_2 = 0
VOC_2 = 0

secondary_x = 6

# Summing the values for the single bar
#secondary_values = np.array([PM_2, NH3_2, VOC_2])
#bottoms = np.zeros_like(secondary_values)
#bottoms[1:] = np.cumsum(secondary_values[:-1])

#ax2.bar(6, secondary_values[0], 0.4, label='PM_2', color='mediumpurple')
#ax2.bar(6, secondary_values[1], 0.4, bottom=secondary_values[0], label='N
H3_2', color='lightpink')
#ax2.bar(6, secondary_values[2], 0.4, bottom=secondary_values[0] + seconda
ry_values[1], label='VOC_2', color='forestgreen')
```

```
#ax2.set_ylabel('Emissions (ktonnes)')

# Setting limits for the secondary y-axis
#ax2.set_ylim(-0.03, 0.002)

#plt.xticks(np.append(X_axis, secondary_x), x + ['Manure Management'], rot
ation=90)

#plt.axvline(x=5.5, color='black', linestyle='--')

fig.tight_layout()
fig.savefig('AP emissions, 100 run, transpo_up', bbox_inches='tight', tran
sparent=True)
plt.show()
```

png

```
FERT_AP_100[2]
```

```
1006.536111434761
```

```
TOTAL_SO2_TRANSPO_25, TOTAL_SO2_TRANSPO_50, TOTAL_SO2_TRANSPO_75, TOTAL_SO
2_TRANSPO_100, TOTAL_NOx_TRANSPO_25, TOTAL_NOx_TRANSPO_50, TOTAL_NOx_TRANS
PO_75, TOTAL_NOx_TRANSPO_100
```

```
(0.06313898929722316,
 0.08339827120641477,
 0.19681544691009764,
 0.264999104874038,
 36.287420981819885,
 47.46886688292719,
 111.93116454490055,
 150.48700521438604)
```

```
print((methane_output_series_25.sum()) / total_upgrade_CO2e_mtonnes_25)
print((methane_output_series_50.sum()) / total_upgrade_CO2e_mtonnes_50)
print((methane_output_series_75.sum()) / total_upgrade_CO2e_mtonnes_75)
print((methane_output_series_100.sum()) / total_upgrade_CO2e_mtonnes_100)
```

```
180380107.78544107
180380107.7854411
180380107.78544107
180380107.78544107
```

```
print(methane_output_series_25.sum())
print(methane_output_series_50.sum())
```

```
print(methane_output_series_75.sum())
print(methane_output_series_100.sum())
```

```
8090922.712616812
16181845.42523364
24272768.13785048
32363690.850467283
```

```
TOTAL_STORAGE_25
```

```
<function __main__.TOTAL_STORAGE_25()>
```

```
TOTAL_25_MANURE, TOTAL_50_MANURE, TOTAL_75_MANURE, TOTAL_100_MANURE, TOTAL
_BASELINE_MANURE
```

```
(2.029183713660765,
 1.1996286773750138,
 0.5602108119806156,
 2.981296058493005e-10,
 2.257637291209932)
```

```
PIPELINE_25, PIPELINE_50, PIPELINE_75, PIPELINE_100
```

```
(0.013557370170358513,
 0.027114740340717057,
 0.04067211051107561,
 0.05422948068143412)
```

```
TOTAL_25_FW, TOTAL_50_FW,TOTAL_75_FW,TOTAL_100_FW,TOTAL_BASELINE_FW
```

```
(0.0068917074796330784, 0.0, 0.0, 0.0, 0.15917466982014797)
```

```
UPGRADE_25, UPGRADE_50, UPGRADE_75, UPGRADE_100
```

```
(0.044854059102742344,
 0.08970811820548474,
 0.1345621773082272,
 0.17941623641096954)
```

```
TRANSPO_25, TRANSPO_50, TRANSPO_75, TRANSPO_100
```

```
(0.0075021820949069985,
 0.009910134735840924,
 0.02338753557565065,
 0.02892876309441111)
```

```
ts_25, ts_50, ts_75, ts_100
```

```
(0.0035864708835526664,
 0.01099696191883509,
 0.040689719803424416,
 0.06531150616142958)

ENERGY_SAVINGS_25, ENERGY_SAVINGS_50, ENERGY_SAVINGS_75, ENERGY_SAVINGS_10
0

(-0.44500074919392457,
 -0.8900014983878503,
 -1.3350022475817767,
 -1.7800029967757005)

FERT_25, FERT_50, FERT_75, FERT_100

(-0.04679316875293088,
 -0.3357545446452093,
 -0.6684390734903892,
 -1.0059723383396566)

x = ['Baseline','Biogas Scenario']
fig, ax1 = plt.subplots()

Manure = np.array([TOTAL_BASELINE_MANURE, TOTAL_100_MANURE])
Pipeline_transfer = (0, PIPELINE_100)
FW = (TOTAL_BASELINE_FW, TOTAL_100_FW)
Upgrading = (0, UPGRADE_100)
Transpo = (0, TRANSPO_100)
Storage = (0, ts_100)
Energy = (0, ENERGY_SAVINGS_100)
Fertilizer = (0, FERT_100)

ax1.bar(x, Manure, label = 'Manure', color="brown")
ax1.bar(x, Pipeline_transfer, bottom = Manure, label = 'Pipeline Transfer'
, color="gold")
ax1.bar(x, FW, bottom = Manure+Pipeline_transfer, label = 'Food Waste',  c
olor="b")
ax1.bar(x, Upgrading, bottom=Manure+Pipeline_transfer+FW, label = 'Uprgrad
ing', color='darkorange')
ax1.bar(x, Transpo, bottom=Manure+Pipeline_transfer+FW+Upgrading, label =
'Transportation', color="c")
ax1.bar(x, Storage, bottom=Manure+Pipeline_transfer+FW+Upgrading+Transpo,
label = 'Storage', color='purple')
#ax1.bar(x, Energy, bottom=Manure+Pipeline_transfer+FW+Upgrading+Transpo,
color="k")
ax1.bar(x, Energy, label = 'Energy Offset', color="forestgreen")
ax1.bar(x, Fertilizer, bottom=Energy, label = 'Fertilizer Production Offse
```

```
t', color="m")

# Calculating and adding symbols at net values for each bar
net_value_baseline = Manure[0] + FW[0]
net_value_run2 = Upgrading[1] + Transpo[1] + Manure[1] + FW[1] + Storage[1
] - (Energy[1] + Fertilizer[1])

plt.xlabel("Scenarios")
ax1.set_ylabel("CO$_{2}$e Emissions (Mtonnes)")
#plt.legend(['Manure Management', 'Pipeline Transfer', 'Food Waste', 'Upgr
ading', 'Transportation','Storage', 'Energy', 'Fertilizer'])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("CO$_{2}$e Emissions (Mtonnes) by Scenario")
plt.axhline(y = 0, color = 'black', linestyle = '-')
#plt.savefig('GHG Emissions')
plt.show()

png

beef_0 = 4.17372879934669, 3.61149916033265, 1.7448201296302, 1.3124667380
7024
beef_1 = 0.663478250858742, 21.3475080136933, 25.32678594, 27.36216629
beef_2 = 0, 1.803955946, 23.7770826, 49.22163381
beef_3 = 0, 0, 2.863019369, 20.21781393
beef_4 = 0, 0, 0.633927197, 0.986857312
beef_5 = 0, 0, 0.400110857, 0.899061926

broiler_0 = 33.2036371337185, 22.61574918, 12.61148323, 9.607603044
broiler_1 = 3.52929134714228, 53.1516457, 55.03344314, 49.03770738
broiler_2 = 0, 2.758090186, 27.64280158, 39.08794766
broiler_3 = 0, 0, 0.74572743, 2.162154022
broiler_4 = 0, 0, 0.03796684, 0.067695592
broiler_5 = 0, 0, 0.058741149, 0.036892307

dairy_0 = 1.39160872221024, 1.515794565, 1.207750435, 0.891226802
dairy_1 = 0.144079352021896, 19.89469124, 23.75238088, 21.79111915
dairy_2 = 0, 2.718122406, 33.85963127, 63.05792546
dairy_3 = 0, 0, 1.309560624, 13.81802026
dairy_4 = 0, 0, 0.107687846, 0.304967383
dairy_5 = 0, 0, 0.47744002, 0.136740941

hog_0 = 11.8300061179963, 9.251206036, 5.266425286, 4.08574756
hog_1 = 2.32389848787498, 59.38139253, 58.43863032, 51.74899622
hog_2 = 0, 3.709101746, 30.52366847, 42.1529384
hog_3 = 0, 0, 0.602415364, 1.8553687
hog_4 = 0, 0, 0.074228157, 0.09947703
```

```
hog_5 = 0, 0, 0.030669436, 0.057472086

fw_0 = 78.939705805276, 75.04738136, 49.17573583, 40.6206586
fw_1 = 16.7306433220593, 24.91492709, 44.07226684, 43.78914341
fw_2 = 0, 0.037691556, 6.231886215, 11.55053677
fw_3 = 0, 0, 0.520111114, 4.039661222
fw_4 = 0, 0, 0, 0
fw_5 = 0, 0, 0, 0

import pandas as pd

x = ['<20', '20-25', '25-30', '30-40', '40-50', '50-75']
fig, ax1 = plt.subplots()

X_axis = np.arange(len(x))

beef = np.array([beef_0[0], beef_1[0], beef_2[0], beef_3[0], beef_4[0], be
ef_5[0]])
broiler = np.array([broiler_0[0], broiler_1[0], broiler_2[0], broiler_3[0]
, broiler_4[0], broiler_5[0]])
dairy = np.array([dairy_0[0], dairy_1[0], dairy_2[0], dairy_3[0], dairy_4[
0], dairy_5[0]])
hog = np.array([hog_0[0], hog_1[0], hog_2[0], hog_3[0], hog_4[0], hog_5[0]
])
fw = np.array([fw_0[0], fw_1[0], fw_2[0], fw_3[0], fw_4[0], fw_5[0]])


ax1.bar(x, fw, label = 'Food Waste', color="maroon", width = 1)
ax1.bar(x, broiler, bottom = fw, label = 'Broiler', color="indianred", wid
th = 1)
ax1.bar(x, hog, bottom = fw+broiler, label = 'Hog',  color="red", width =
1)
ax1.bar(x, dairy, bottom=fw+broiler+hog, label = 'Dairy', color="coral", w
idth = 1)
ax1.bar(x, beef, bottom=fw+broiler+hog+dairy, label = 'Beef', color="misty
rose", width = 1)

plt.xlabel("Cost of RNG ($/GJ)")
ax1.set_ylabel("Share of available feedstock consumed (%)")
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("25% Run")
#plt.savefig('GHG Emissions')
plt.show()

png
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plot

_data=[["(0-25]", fw_0[0],broiler_0[0], hog_0[0], dairy_0[0], beef_0[0]],
      ["(20-25]", fw_1[0], broiler_1[0], hog_1[0], dairy_1[0], beef_1[0]],
      ["(25-30]", fw_2[0], broiler_2[0], hog_2[0], dairy_2[0], beef_2[0]],
      ["(30-40]", fw_3[0], broiler_3[0], hog_3[0], dairy_3[0], beef_3[0]],
       ["(40-50]",fw_4[0], broiler_4[0], hog_4[0], dairy_4[0], beef_4[0]],
      ["(50-75]", fw_5[0], broiler_5[0],hog_5[0], dairy_5[0], beef_5[0]] ]

color = {"Beef":"mistyrose",
        "Chicken":"indianred",
         "Dairy" : "coral",
         "Hog" : "red",
         "Food Waste" : "maroon",
        }

# Plotting multiple groups
_df=pd.DataFrame(_data, columns=["RNG production cost", "Food Waste", "Chi
cken", "Hog", "Dairy", "Beef"])
_df.plot.bar(x="RNG production cost", y=["Food Waste", "Chicken", "Hog", "
Dairy", "Beef"],figsize=(8,5), color=color, width = 0.9,  ylabel = "Share
of available feedstock consumed (%)", title = "25% Run")
```

```
<Axes: title={'center': '25% Run'}, xlabel='RNG production cost', ylabel='
Share of available feedstock consumed (%)'>
```

```
png
```

```
x = ['<20', '20-25', '25-30', '30-40', '40-50', '50-75']
fig, ax1 = plt.subplots()

beef = np.array([beef_0[1], beef_1[1], beef_2[1], beef_3[1], beef_4[1], be
ef_5[1]])
broiler = np.array([broiler_0[1], broiler_1[1], broiler_2[1], broiler_3[1]
, broiler_4[1], broiler_5[1]])
dairy = np.array([dairy_0[1], dairy_1[1], dairy_2[1], dairy_3[1], dairy_4[
1], dairy_5[1]])
hog = np.array([hog_0[1], hog_1[1], hog_2[1], hog_3[1], hog_4[1], hog_5[1]
])
fw = np.array([fw_0[1], fw_1[1], fw_2[1], fw_3[1], fw_4[1], fw_5[1]])

ax1.bar(x, fw, label = 'Food Waste', color="darkblue")
ax1.bar(x, broiler, bottom = fw, label = 'Broiler', color="blue")
```

```
ax1.bar(x, hog, bottom = fw+broiler, label = 'Hog',  color="cornflowerblue
")
ax1.bar(x, dairy, bottom=fw+broiler+hog, label = 'Dairy', color='skyblue')
ax1.bar(x, beef, bottom=fw+broiler+hog+dairy, label = 'Beef', color="alice
blue")

plt.xlabel("Cost of RNG ($/GJ)")
ax1.set_ylabel("Share of available feedstock consumed (%)")
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("50% Run")
#plt.savefig('GHG Emissions')
plt.show()
```

```
png
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plot

_data=[["(0-20]", fw_0[1],broiler_0[1], hog_0[1], dairy_0[1], beef_0[1]],
      ["(20-25]", fw_1[1], broiler_1[1], hog_1[1], dairy_1[1], beef_1[1]],
      ["(25-30]", fw_2[1], broiler_2[1], hog_2[1], dairy_2[1], beef_2[1]],
      ["(30-40]", fw_3[1], broiler_3[1], hog_3[1], dairy_3[1], beef_3[1]],
       ["(40-50]",fw_4[1], broiler_4[1], hog_4[1], dairy_4[1], beef_4[1]],
      ["(50-75]", fw_5[1], broiler_5[1],hog_5[1], dairy_5[1], beef_5[1]] ]

color = {"Beef":"aliceblue",
        "Chicken":"blue",
         "Dairy" : "skyblue",
         "Hog" : "cornflowerblue",
         "Food Waste" : "darkblue",
        }

# Plotting multiple groups
_df=pd.DataFrame(_data, columns=["RNG production cost", "Food Waste", "Chi
cken", "Hog", "Dairy", "Beef"])
_df.plot.bar(x="RNG production cost", y=["Food Waste", "Chicken", "Hog", "
Dairy", "Beef"],figsize=(8,5), color=color, width = 0.9,  ylabel = "Share
of available feedstock consumed (%)", title = "50% Run")
```

```
<Axes: title={'center': '50% Run'}, xlabel='RNG production cost', ylabel='
Share of available feedstock consumed (%)'>
```

```
png
```

```
x = ['<20', '20-25', '25-30', '30-40', '40-50', '50-75']
fig, ax1 = plt.subplots()
```

```
beef = np.array([beef_0[2], beef_1[2], beef_2[2], beef_3[2], beef_4[2], be
ef_5[2]])
broiler = np.array([broiler_0[2], broiler_1[2], broiler_2[2], broiler_3[2]
, broiler_4[2], broiler_5[2]])
dairy = np.array([dairy_0[2], dairy_1[2], dairy_2[2], dairy_3[2], dairy_4[
2], dairy_5[2]])
hog = np.array([hog_0[2], hog_1[2], hog_2[2], hog_3[2], hog_4[2], hog_5[2]
])
fw = np.array([fw_0[2], fw_1[2], fw_2[2], fw_3[2], fw_4[2], fw_5[2]])

ax1.bar(x, fw, label = 'Food Waste', color="darkgreen")
ax1.bar(x, broiler, bottom = fw, label = 'Broiler', color="green")
ax1.bar(x, hog, bottom = fw+broiler, label = 'Hog',  color="limegreen")
ax1.bar(x, dairy, bottom=fw+broiler+hog, label = 'Dairy', color='lime')
ax1.bar(x, beef, bottom=fw+broiler+hog+dairy, label = 'Beef', color="light
green")

plt.xlabel("Cost of RNG ($/GJ)")
ax1.set_ylabel("Share of available feedstock consumed (%)")
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("75% Run")
#plt.savefig('GHG Emissions')
plt.show()

png

import pandas as pd
import numpy as np
import matplotlib.pyplot as plot

_data=[["(0-20]", fw_0[2],broiler_0[2], hog_0[2], dairy_0[2], beef_0[2]],
      ["(20-25]", fw_1[2], broiler_1[2], hog_1[2], dairy_1[2], beef_1[2]],
      ["(25-30]", fw_2[2], broiler_2[2], hog_2[2], dairy_2[2], beef_2[2]],
      ["(30-40]", fw_3[2], broiler_3[2], hog_3[2], dairy_3[2], beef_3[2]],
       ["(40-50]",fw_4[2], broiler_4[2], hog_4[2], dairy_4[2], beef_4[2]],
      ["(50-75]", fw_5[2], broiler_5[2],hog_5[2], dairy_5[2], beef_5[2]] ]

color = {"Beef":"lightgreen",
        "Chicken":"green",
         "Dairy" : "lime",
         "Hog" : "limegreen",
         "Food Waste" : "darkgreen",
        }

# Plotting multiple groups
```

```
_df=pd.DataFrame(_data, columns=["RNG production cost", "Food Waste", "Chi
cken", "Hog", "Dairy", "Beef"])
_df.plot.bar(x="RNG production cost", y=["Food Waste", "Chicken", "Hog", "
Dairy", "Beef"],figsize=(8,5), color=color, width = 0.9,  ylabel = "Share
of available feedstock consumed (%)", title = "75% Run")
```

```
<Axes: title={'center': '75% Run'}, xlabel='RNG production cost', ylabel='
Share of available feedstock consumed (%)'>
```

png

```
x = ['<20', '20-25', '25-30', '30-40', '40-50', '50-75']
fig, ax1 = plt.subplots()

beef = np.array([beef_0[3], beef_1[3], beef_2[3], beef_3[3], beef_4[3], be
ef_5[3]])
broiler = np.array([broiler_0[3], broiler_1[3], broiler_2[3], broiler_3[3]
, broiler_4[3], broiler_5[3]])
dairy = np.array([dairy_0[3], dairy_1[3], dairy_2[3], dairy_3[3], dairy_4[
3], dairy_5[3]])
hog = np.array([hog_0[3], hog_1[3], hog_2[3], hog_3[3], hog_4[3], hog_5[3]
])
fw = np.array([fw_0[3], fw_1[3], fw_2[3], fw_3[3], fw_4[3], fw_5[3]])

ax1.bar(x, fw, label = 'Food Waste', color="indigo")
ax1.bar(x, broiler, bottom = fw, label = 'Broiler', color="blueviolet")
ax1.bar(x, hog, bottom = fw+broiler, label = 'Hog',  color="mediumpurple")
ax1.bar(x, dairy, bottom=fw+broiler+hog, label = 'Dairy', color='mediumorc
hid')
ax1.bar(x, beef, bottom=fw+broiler+hog+dairy, label = 'Beef', color="plum"
)

plt.xlabel("Cost of RNG ($/GJ)")
ax1.set_ylabel("Share of available feedstock consumed (%)")
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.title("100% Run")
#plt.savefig('GHG Emissions')
plt.show()
```

png

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plot

_data=[["(0-20]", fw_0[3],broiler_0[3], hog_0[3], dairy_0[3], beef_0[3]],
      ["(20-25]", fw_1[3], broiler_1[3], hog_1[3], dairy_1[3], beef_1[3]],
```

```
      ["(25-30]", fw_2[3], broiler_2[3], hog_2[3], dairy_2[3], beef_2[3]],
      ["(30-40]", fw_3[3], broiler_3[3], hog_3[3], dairy_3[3], beef_3[3]],
       ["(40-50]",fw_4[3], broiler_4[3], hog_4[3], dairy_4[3], beef_4[3]],
      ["(50-75]", fw_5[3], broiler_5[3],hog_5[3], dairy_5[3], beef_5[3]] ]

color = {"Beef":"plum",
        "Chicken":"blueviolet",
         "Dairy" : "mediumorchid",
         "Hog" : "mediumpurple",
         "Food Waste" : "indigo",
        }

# Plotting multiple groups
_df=pd.DataFrame(_data, columns=["RNG production cost", "Food Waste", "Chi
cken", "Hog", "Dairy", "Beef"])
_df.plot.bar(x="RNG production cost", y=["Food Waste", "Chicken", "Hog", "
Dairy", "Beef"],figsize=(8,5), color=color, width = 0.9,  ylabel = "Share
of available feedstock consumed (%)", title = "100% Run")
```

```
<Axes: title={'center': '100% Run'}, xlabel='RNG production cost', ylabel=
'Share of available feedstock consumed (%)'>
```

```
png
```

```
x = ['(0-20]', '(20-25]', '(25-30]', '(30-40]', '(40-50]', '(50-75]']

fig = pylab.figure()
ax  = fig.add_subplot(111)

percentages = [77, 23, 0, 0, 0, 0]

ax.bar(x, percentages, color = "maroon", width = 0.95)
ax.set_ylabel("Percentage of Plants")
ax.set_xlabel("RNG production cost")

plt.axvline(x = 0.2, color = 'white', linestyle = '--', alpha = 1)
```

```
<matplotlib.lines.Line2D at 0x7fe69328b640>
```

```
png
```

```
run25sys = 3117210735 / 161818454.3
run50sys = 6893678425 / 323636908.5
run75sys = 11620302291 / 485455362.8
run100sys = 16517941597 / 647273817

print(run25sys, run50sys, run75sys, run100sys)
```

```
19.26362940793459 21.30065590155024 23.936911982961 25.519248829742175

run25 = [77, 23, 0, 0, 0, 0]
run50 = [21, 72, 7, 0, 0, 0]
run75 = [7, 43, 44, 3, 1, 2]
run100 = [4, 27.75, 54, 13.5, 0.5, 0.25]

x = ['(0-20]', '(20-25]', '(25-30]', '(30-40]', '(40-50]', '(50-75]']

fig = pylab.figure()
ax  = fig.add_subplot(111)

percentages = [21, 72, 7, 0, 0, 0]

ax.bar(x, percentages, color = "darkblue", width = 0.95)
ax.set_ylabel("Percentage of Plants")
ax.set_xlabel("RNG production cost")

plt.axvline(x = 0.7, color = 'white', linestyle = '--', alpha = 1)

<matplotlib.lines.Line2D at 0x7fe6933563b0>

png

x = ['(0-20]', '(20-25]', '(25-30]', '(30-40]', '(40-50]', '(50-75]']

fig = pylab.figure()
ax  = fig.add_subplot(111)

percentages = [7, 43, 44, 3, 1, 2]

ax.bar(x, percentages, color = "darkgreen", width = 0.95)
ax.set_ylabel("Percentage of Plants")
ax.set_xlabel("RNG production cost")

plt.axvline(x = 1.2, color = 'white', linestyle = '--', alpha = 1)

<matplotlib.lines.Line2D at 0x7fe662eb3220>

png

x = ['(0-20]', '(20-25]', '(25-30]', '(30-40]', '(40-50]', '(50-75]']

fig = pylab.figure()
ax  = fig.add_subplot(111)

percentages = [4, 27.75, 54, 13.5, 0.5, 0.25]
```

```
ax.bar(x, percentages, color = "indigo", width = 0.95)
ax.set_ylabel("Percentage of Plants")
ax.set_xlabel("RNG production cost")

plt.axvline(x = 1.6, color = 'white', linestyle = '--', alpha = 1)
```

```
<matplotlib.lines.Line2D at 0x7fe692d60f40>
```

```
png
```

```
energy = [161, 323, 485, 647]
cost = [3, 6, 11, 16]

import numpy as np
import matplotlib.pyplot as plt


# Plotting the data
fig = plt.figure()
ax = fig.add_subplot(111)

colors = ['maroon', 'darkblue', 'darkgreen', 'indigo']
runlabels = ['$19.3/GJ', '$21.3/GJ', '$23.9/GJ', '$25.5/GJ']

for i in range(len(energy)):
    if i == 0:
        ax.plot(energy[i], cost[i], marker='o', markersize=8, color=colors
[i])
        ax.annotate(runlabels[i], (energy[i], cost[i]), textcoords="offset
points", xytext=(10,10), ha='center')
    else:
        ax.plot(energy[i], cost[i], marker='o', markersize=8, color=colors
[i])
        ax.annotate(runlabels[i], (energy[i], cost[i]), textcoords="offset
points", xytext=(-10,-15), ha='center')


ax.text(1.1, -0.05, 'Millions', transform=ax.transAxes, ha='right', va='bo
ttom')

ax.text(-0.1, 0.95, 'Billions', transform=ax.transAxes, ha='left', va='bot
tom', rotation=90)

# Adding labels and title
ax.set_xlabel('Energy produced over project lifetime (f$_E$, GJ)')
ax.set_ylabel('Present value of system cost (f$_c$, dollar)')
ax.set_title('System Level RNG Cost')
```

```
plt.show()
```

```
png
```

```
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, sharex = True)
fig.suptitle('Plant level RNG cost')

x = ['(0-20]', '(20-25]', '(25-30]', '(30-40]', '(40-50]', '(50-75]']

ax1.bar ( x, run25, color = "maroon", width = 0.95)

ax2.bar (x, run50, color = "darkblue", width = 0.95)

ax3.bar (x, run75, color = "darkgreen", width = 0.95)

ax4.bar (x, run100, color = "indigo", width = 0.95)

ax1.axvline (x = 0.2, color = 'white', linestyle = '--', alpha = 1)
ax2.axvline (x = 0.7, color = 'white', linestyle = '--', alpha = 1)
ax3.axvline (x = 1.2, color = 'white', linestyle = '--', alpha = 1)
ax4.axvline (x = 1.6, color = 'white', linestyle = '--', alpha = 1)


fig.text(0.5, 0.04, 'RNG production cost', ha='center', va='center')
fig.text(0.06, 0.5, 'Percentage of plants', ha='center', va='center', rota
tion='vertical')

Text(0.06, 0.5, 'Percentage of plants')

png
```

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.gridspec import GridSpec

# Given data for subplots
x = ['(0-20]', '(20-25]', '(25-30]', '(30-40]', '(40-50]', '(50-75]']
run25 = np.random.randint(1, 10, size=len(x))
run50 = np.random.randint(1, 10, size=len(x))
run75 = np.random.randint(1, 10, size=len(x))
run100 = np.random.randint(1, 10, size=len(x))
```

```
# Given data for the new plot
energy = [161, 323, 485, 647]
cost = [3, 6, 11, 16]
run_labels = ['run1', 'run2', 'run3', 'run4']  # Labels for each data poin
t

# Create figure and gridspec
fig = plt.figure(figsize=(12, 6))
gs = GridSpec(1, 2, width_ratios=[2, 1.5])  # Adjust width ratios for bett
er visualization

# Subplots
ax1 = fig.add_subplot(gs[0, 0])
ax1.bar(x, run25, color="maroon", width=0.95)
ax1.set_title('Subplot 1')

ax2 = fig.add_subplot(gs[0, 0])
ax2.bar(x, run50, color="darkblue", width=0.95)
ax2.set_title('Subplot 2')

ax3 = fig.add_subplot(gs[0, 0])
ax3.bar(x, run75, color="darkgreen", width=0.95)
ax3.set_title('Subplot 3')

ax4 = fig.add_subplot(gs[0, 0])
ax4.bar(x, run100, color="indigo", width=0.95)
ax4.set_title('Subplot 4')

# New plot
ax5 = fig.add_subplot(gs[0, 1])
colors = ['maroon', 'darkblue', 'darkgreen', 'indigo']
for i in range(len(energy)):
    if i == 0:
        ax5.plot(energy[i], cost[i], marker='o', markersize=8, color=color
s[i])
        ax5.annotate(run_labels[i], (energy[i], cost[i]), textcoords="offs
et points", xytext=(5, -15), ha='center')
    else:
        ax5.plot(energy[i], cost[i], marker='o', markersize=8, color=color
s[i])
        ax5.annotate(run_labels[i], (energy[i], cost[i]), textcoords="offs
et points", xytext=(5, 5), ha='center')

ax5.set_xlabel('Energy produced over project lifetime (f$_E$, GJ)')
ax5.set_ylabel('Present value of system cost (f$_c$, cost)')
ax5.set_title('System Level RNG Cost')
```

```
# Adding units for the axis labels
ax5.text(1.1, -0.05, 'Millions', transform=ax5.transAxes, ha='right', va='
bottom')
ax5.text(-0.1, 0.95, 'Billions', transform=ax5.transAxes, ha='left', va='b
ottom', rotation=90)

plt.tight_layout()
plt.show()
```

```
png
```

```
import geopandas as gpd

# Set filepath
fp = '/Users/savannahbindas/Documents/UW/Research/Feedstock_Data_All_Savan
nah/Feedstock_Data_All_Savannah.shp'

# Read file using gpd.read_file()
data = gpd.read_file(fp)

print(data.head(5))
```

```
        id          left           top         right        bottom   ICI
\
0  41030.0  1.294976e+06  2.554193e+06  1.299976e+06  2.549193e+06   0.0
1  41200.0  1.299976e+06  2.564193e+06  1.304976e+06  2.559193e+06   0.0
2  41204.0  1.299976e+06  2.544193e+06  1.304976e+06  2.539193e+06   2.7
3  41205.0  1.299976e+06  2.539193e+06  1.304976e+06  2.534193e+06  47.1
4  41373.0  1.304976e+06  2.559193e+06  1.309976e+06  2.554193e+06   0.0

   Scaled_Man   SSO  SSO_ICI      Beef  ...  Chicken_50  Dairy_50  Hog_50
\
0       429.8   0.0      0.0   429.836  ...           0         0       0
1      1102.7   0.0      0.0  1102.653  ...           0         0       0
2        21.3   1.4      4.1     0.000  ...           0         0       0
3         0.0  25.3     72.4     0.000  ...           0         0       0
4       816.3   0.0      0.0   816.340  ...           0         0       0

   FW_50  Beef_75  Chicken_75  Dairy_75  Hog_75  FW_75  \
0      0      430           0         0       0      0
1      0     1103           0         0       0      0
2      0        0           0         0       0      0
3      0        0           0         0       0      0
4      0      816           0         0       0      0

                                            geometry
```

```
0  POLYGON ((1294976.415 2554192.604, 1299976.415...
1  POLYGON ((1299976.415 2564192.604, 1304976.415...
2  POLYGON ((1299976.415 2544192.604, 1304976.415...
3  POLYGON ((1299976.415 2539192.604, 1304976.415...
4  POLYGON ((1304976.415 2559192.604, 1309976.415...

[5 rows x 29 columns]
```

```
import matplotlib.pyplot as plt

%matplotlib inline
data.plot()
```

```
<Axes: >
```

```
png
```

```
pd.set_option('display.max_columns',None)
print(data.head())
```

```
        id          left           top         right        bottom   ICI
\
0  41030.0  1.294976e+06  2.554193e+06  1.299976e+06  2.549193e+06   0.0
1  41200.0  1.299976e+06  2.564193e+06  1.304976e+06  2.559193e+06   0.0
2  41204.0  1.299976e+06  2.544193e+06  1.304976e+06  2.539193e+06   2.7
3  41205.0  1.299976e+06  2.539193e+06  1.304976e+06  2.534193e+06  47.1
4  41373.0  1.304976e+06  2.559193e+06  1.309976e+06  2.554193e+06   0.0

   Scaled_Man   SSO  SSO_ICI      Beef  Dairy  Broiler  Hog  Beef_25  \
0       429.8   0.0      0.0   429.836    0.0    0.000  0.0      430
1      1102.7   0.0      0.0  1102.653    0.0    0.000  0.0     1103
2        21.3   1.4      4.1     0.000    0.0   21.267  0.0        0
3         0.0  25.3     72.4     0.000    0.0    0.000  0.0        0
4       816.3   0.0      0.0   816.340    0.0    0.000  0.0      816

   Chicken_25  Dairy_25  Hog_25  FW_25  Beef_50  Chicken_50  Dairy_50  Hog
_50  \
0           0         0       0      0      430           0         0
0
1           0         0       0      0     1103           0         0
0
2           0         0       0      0        0           0         0
0
3           0         0       0      0        0           0         0
0
4           0         0       0      0      816           0         0
0
```

```
   FW_50  Beef_75  Chicken_75  Dairy_75  Hog_75  FW_75  \
0      0      430           0         0       0      0
1      0     1103           0         0       0      0
2      0        0           0         0       0      0
3      0        0           0         0       0      0
4      0      816           0         0       0      0

                                              geometry
0  POLYGON ((1294976.415 2554192.604, 1299976.415...
1  POLYGON ((1299976.415 2564192.604, 1304976.415...
2  POLYGON ((1299976.415 2544192.604, 1304976.415...
3  POLYGON ((1299976.415 2539192.604, 1304976.415...
4  POLYGON ((1304976.415 2559192.604, 1309976.415...
```

```
print(data['Beef'])
```

```
0        429.836
1       1102.653
2          0.000
3          0.000
4        816.340
          ...
3843     692.124
3844     329.430
3845     439.805
3846    1355.376
3847     597.945
Name: Beef, Length: 3848, dtype: float64
```

```
print(data.columns)
```

```
Index(['id', 'left', 'top', 'right', 'bottom', 'ICI', 'Scaled_Man', 'SSO',
       'SSO_ICI', 'Beef', 'Dairy', 'Broiler', 'Hog', 'Beef_25', 'Chicken_2
5',
       'Dairy_25', 'Hog_25', 'FW_25', 'Beef_50', 'Chicken_50', 'Dairy_50',
       'Hog_50', 'FW_50', 'Beef_75', 'Chicken_75', 'Dairy_75', 'Hog_75',
       'FW_75', 'geometry'],
      dtype='object')
```