STATS 3DA3

Homework Assignment 2

Neerat Grewal (400323214)

2025-01-24

For all the questions, use Python 3.11.5 and virtual environment. Then, install the required libraries for text mining and Shiny visualization.

Question 1: Word Cloud Analysis

Let's explore the article **"Data Science and Engineering With Human in the Loop, Behind the Loop, and Above the Loop"** by Xiao-Li Meng (2023). Follow the steps below to create and analyze a word cloud for pages 2–5 of the article.

(1) Add the article **"Data Science and Engineering With Human in the Loop, Behind the Loop, and Above the Loop"** by Xiao-Li Meng (2023) to your reference list.

References

Meng, X.-L. (2023). Data Science and Engineering With Human in the Loop, Behind the Loop, and Above the Loop. Harvard Data Science Review, 5(2). https://doi.org/10.1162/99608f92.68a012eb

(2) Download the PDF of the article.

Hint:

- Access the article via `https://doi.org/10.1162/99608f92.68a012eb`.
- Click the **Download** button in the top-right corner, and choose the **PDF** format.
- Move the downloaded file to your working folder and rename it as `paper.pdf`.

(3) Use `pdfplumber.open()` to open the PDF.

```
import numpy as numpy
import pandas as pd
import re
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import nltk
import pytz

import matplotlib.pyplot as plt
from wordcloud import WordCloud

import pdfplumber
```

```
# this NLTK data download is called only one time
# word_tokenize uses punkt tokenizer model
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Error loading punkt: <urlopen error [SSL:
[nltk_data]     CERTIFICATE_VERIFY_FAILED] certificate verify failed:
[nltk_data]     unable to get local issuer certificate (_ssl.c:1018)>
[nltk_data] Error loading stopwords: <urlopen error [SSL:
[nltk_data]     CERTIFICATE_VERIFY_FAILED] certificate verify failed:
[nltk_data]     unable to get local issuer certificate (_ssl.c:1018)>

False
```

```
with pdfplumber.open("/Users/jaspreetbrar/Desktop/Classes/Stats3DA3/Assignment2/paper.pdf") as
    # Extract text from pages 1 to 9
    text = ""
    for i in range(1, 9):
```

```
        page = pdf.pages[i]
        text += page.extract_text()
    print(text)
```

Harvard Data Science Review • Issue 5.2, Spring 2023 Data Science and Engineering With Human in
Loop, and Above the Loop

The term human-in-the-loop (HITL) generally refers to the need for human interaction, intervent
judgment to control or change the outcome of a process, and it is a practice that is being inc
emphasized in machine learning, generative AI, and the like. For readers who have not enjoyed m
assisted learning or who are comfortable with their human intelligences, it would not surprise
you would wonder, upon hearing the term HITL, ‘Wait, when did we take humans out of the loop?’
diversity of human minds guarantees that such a question would generate reactions ranging from
to being despised. Regardless, however, the arrival of ChatGPT and other AI chatbots has made t
academic debate of machine intelligence versus human intelligence into a household conversatio
therefore an apt time to turn such a rhetorical question into retrospection and reflection.

With my human thinking in the loop, I probably can take any issue of HDSR to remind ourselves o
we humans play in the data science (DS) ecosystem. Indeed, as I wrote in the inaugural editoria
Science: An Artificial Ecosystem," "the term ‘artificial’ highlights both the fact that DS is a
and that it depends critically on computing advances" (Meng, 2019). (The same can and should be
data engineering (DE), which can advance faster than DS, as we witnessed recently with the prog
GPT-n to GPT-(n+1), n<4.) However, the current issue is particularly fitting for this contempla
the first HDSR issue with a special theme dedicated to a beautiful mind that came with a beauti
that yet to be applied to any machine. As such, I ask for readers’ indulgence-
and my fellow data scientists’
forgiveness-for creating sibling rivalries to HITL.

Human in the Loop: From Developers to Deployers

As cliché as this may sound, humans are always in the loop for advancing data science or engine
fronts, literally and figuratively. Indeed, for those who don’t mind the label of anthropocentri
paraphrasing ‘DS/DE are of the people, by the people, and for the people’ might be a more arous
this editorial. Data are typically digital representations of some people’s mental activities o
endeavors, and they are collected because some (other) people see the benefit to doing so. The
for the data subjects, the data collectors, or other people, which often include those yet to b

and optimize the benefit, yet more groups of people would be required for developing methods, i
algorithms, interpreting and deploying the results, and so on, in order to reap the perceived b
For the Recreations in Randomness column, Kavya Mehul Shah and Ammaar Ahmed Saeed's (2023) acc
on "Innovating a Centuries-Old Sport: How Emerging Data Analytics Tools Are Redefining Cricket"
a concrete illustration of such human effort. For example, cricket highlights are short video c
collected because they can benefit players, coaches, and organizations. Traditionally the creat
highlights "requires sports broadcasters and video editors to manually parse through hours of i
identify exciting moments"-clearly a not very exciting process to say the least. Automated high
generation is then a very welcome advance that requires both data science (e.g., semantic segme
2Harvard Data Science Review • Issue 5.2, Spring 2023 Data Science and Engineering With Human i
Loop, and Above the Loop
identify "exciting moments") and data engineering (e.g., effective software for routine applica
progress has been made in the last 20 years, as reviewed in the article. Yet the 2 decades of p
indicate the amount of human effort required to automate the creation process, not to mention h
needed in translating highlights into the desired benefit.
As another indication of the human effort, when I initially opened the document file for this c
prepare for this editorial, I was a bit surprised to see that the word count was about 4,600-
far exceeding the
normally expected 2,500 words for a column. In case a reader wonders why I was not aware of the
submission until the end stages, I am grateful to the interim Co-Editor-in-Chiefs, Francesca Do
David Parkes, whose dedication allowed me to take a real sabbatical (which incidentally gave me
appreciate the benefit of sports). It turns out that the article has 65 references (which are n
the 2,500 limit), which is an unusually extensive list even for a regular article. But it is fi
since the list provides a chronological picture of human effort and progress in reshaping a hal
sport via data science and engineering. Indeed, it is almost certain that the list is still onl
sample.
… and From Software Engineers to Data Science Managers
The sabbatical year also afforded me the opportunity to join an international collaboration on
conditions globally in general, and in Africa in particular, by combining ground-level survey d
image data via deep learning. The data involved are treasures not only for the mission of the c
also for data science and engineering education in general. They exhibit almost the full spectr
and defects as routinely encountered in practice, including selection bias, measurement errors

information due to privacy protection, mismatches in spatial, temporal, and contextual resoluti
incomplete documentation of data provenance, just to provide a partial list. The collaboration
multi-faceted, involving multiple disciplines (e.g., computer science, engineering, sociology,
members in all career stages from undergraduate to senior researchers with a variety of working
are driven by different interests or incentives. This collaborative experience, which is still
my appreciation of two kinds of humans in the loop, so to speak.

One kind of profession is software engineers, whose technical skills and teamwork (or lack ther
govern the pace for progress. For complex projects, many methodological ideas need to be (stres
These tests can take weeks or minutes, all depending on the engineers' experiences and skills.
am involved in, a clever implementation circumventing a map computation with mismatched resolut
up the test by several weeks. With those experiences still fresh, I literally thought I was day
saw the article "Software Engineering Practices in Academia: Promoting the 3Rs-
Readability, Resilience, and
Reuse" by Andrew Connolly, Joseph Hellerstein, Naomi Alterman, David Beck, Rob Fatland, Ed Lazo
Vani Mandava, and Sarah Stone (2023). The first section title says it all: "Good Data Science I
Quality Software Engineering." Its diagnosis of the major issues in academia is spot on, as aca

"address their software development needs by hiring graduate students, often without any formal
software engineering," and the vast majority of the software projects are created in an often-i
setting: single developer, single user, and single study. For any academics who are interested
quality of their data science projects and of their life by reducing work inefficiency, the tim
digesting this article is likely to be negligible compared to the time saved moving forward, be
provides real-life successes, reasons, and recommendations-with specific recipes-
for adopting the '3R'
principle.

The other kind of profession is data science managers. With a team of diverse expertise, interes
habits and schedules, and the nature of the project being multi-phased, having an all-around da
manager is another key for both the quality and speed of the project. For the collaboration I a
things changed substantially once we had such a data science manager in place. A data science m
more like a chief of staff to the intellectual leaders of a data science collaboration, which m
manager needs to have sufficient technical, personal, and organizational skills to be effective

a data science manager is to help to identify, establish, and sustain a collaborative platform that is conducive to making the whole more than the sum of its parts, or at least not less. The happen (and did happen even in my very limited experience) when there are miscommunications, misunderstandings, and mismatched expectations among different team members or groups. The mana needs to be a strategic thinker with sufficient know-how to recognize or anticipate weak links or preventive measures accordingly, and ultimately creative an incentivizing infrastructure to efficiency and team members' desire to further the collaboration.

With these reflections from my recent experiences, I was instantly attracted by another article "Managing Embedded Data Science Teams for Success: How Managers Can Navigate the Advantages and Challenges of Distributed Data Science," by Marta Stelmaszak and Kelsey Kline (2023). Through a the authors summarize three advantages for the embedded structures, "1) agility in delivering c better tailored analytics, 2) concentrated and cultivated data science expertise, and 3) emergi solution ideas," as well as three disadvantages: "1) growing away from the enterprise infrastru duplicating data science efforts, and 3) weakening links with strategic impact." Although the s rather different from academic collaborations, the main conclusion of the article is the data s the critical HITL to reap these advantages and curtail the disadvantages. To do so, the article science managers need to have aptitudes for all three managing directions: within a team (e.g. balance workload for members with varying expertise), vertically in an organization (e.g., inte to maintain team visibility at all levels of the organization), and horizontally across the org capability to develop networks with other data science teams).

The vital role of managing is furthered demonstrated in "To Deploy Machine Learning, You Must M Operational Change-Here Is How UPS Got It Right" by Eric Siegel (2023). The article starts with

Loop, and Above the Loop

story of how a trash collection company dumped a university generated route-optimizing plan tha number of trash tracks by 50%. I won't spoil the rest of the plot, which could easily be adapte a Hollywood movie on machine learning or AI (perhaps with the help of GPT-n), but only to repea metaphor Siegel used. The "rocket science" of machine learning for commercial use is easy. The getting the end product or process launched, which requires changing humans' habits. That's not science; that's beyond rocket science.

Human Behind the Loop: With Beautiful Minds and Beautiful

Hearts

Much of the research of and discussions about HITL focus on the interaction between humans and
But as we just discussed, translating and transforming the advances in data science and enginee
value for human societies is a matter of interactions among humans. Arguably another kind of hu
interaction is even more important for the data science enterprise, namely the professional and
interactions among members of the data science ecosystem. A shining example is provided by the
theme in this issue celebrating the life and work of Sir David Cox, a giant in statistics and i
In his long and extremely distinguished career, David served as a global model in everything he
extremely deep, broad, and influential scholar, a much-loved advisor of over 60 doctorial stude
exceedingly effective editor for Biometrika for 25 years, and a well-respected academic and pro
(e.g., serving as Warden of Nuffield College, and as president of multiple professional societi
articles in the special theme document many interactions with David in a myriad of ways and acr
decades (and countries). Yet they reflect a common theme: the lasting impression and impact of
interactions. I am deeply grateful to the co-editors of this special theme, Sylvia Richardson a
Wermuth, for organizing and introducing (Richardson & Wermuth, 2023) such a rich and heartwarmi
collection, and to all contributors for sharing their personal stories with all of us, to which
My first interaction with David took place when I was a PhD student in late 1980s, where I co-a
article that was submitted to Biometrika. At that time, the review processes for most statistic
notoriously slow; it was rather common for the review process to take more than 6 months for th
Biometrika was an exception, despite having the smallest editorial board among all top-tier jou
that expectation, I was rather surprised when we heard from David within 2 weeks of submission
not a quick rejection letter. Rather, David had provided detailed comments and asked us to do a
he sent it out for a full review. I do not recall the details of David's comments, but I remem
being very impressed by the promptness and insightfulness of the comments, which led to a much-
revision, and ultimately my first publication in the tier-one journals in statistics.
Years later I had an opportunity to meet with David in person, when I could no longer hold my c
hence asked how he could handle so many articles personally, and yet with such efficiency. The

gave can be found in A C. Davison's (2023) reminisces: David read every submission on the daily
Tube ride. But of course, that is only half of the story since reading is not the same as comin
insightful and persuasive comments. For that, few could compete with David's encyclopedic knowl
insights, and rich intuition. Like Davison (2023) who tried to "emulate his quick turnaround on

and constructive suggestions to authors," I too tried to follow David's model in my own editori
find how challenging that is-it is certainly not nearly as easy a ride as David's daily Tube ri
In the world of leading scholars, there must be many minds that are as beautiful as David's int
borrow the Hollywoodized phrase 'beautiful mind.' But having a beautiful mind does not necessar
a great educator, and indeed in some academic circles, a negative correlation between the two h
suggested. As reported by my colleague Joe Blitzstein (2023) in the Bits and Bites column of th
David's beautiful mind for research came with a beautiful heart for education. About a decade a
co-taught a course on reading Cox's many contributions, and we invited David to come to provide
opportunity to students to meet an intellectual giant in person. He agreed very promptly, but u
issue (and his doctors) prevented him from making the international trip. He kindly offered to
with students in writing, which the students of course were thrilled to have the chance to dire
the world's most preeminent statistician alive then. They hence asked many tough questions, whi
summarized and sent to David. We of course only expected him to answer some of the questions, h
busy he was-he was ageless in terms of professional activities, and consequently in extremely h
To our great surprise, David provided detailed responses to essentially all questions. I was th
when I was reminded by Joe that he had kept this precious document. I immediately invited Joe t
so we could share with the world this hidden set of treasures of Cox's insights and wisdom. It
testimony to David's beautiful heart for education and his care for future generations, a repea
also vivid in the 20 articles celebrating Sir David Cox, a true model for all (data) scientists
Human Above the Loop: Oversight and Insight
A key public concern of ChatGPT and the like is that AI will replace humans in many ways, espec
regarding jobs. There are indeed jobs that will be done by AI-I certainly have used ChatGPT inc
for checking my Chinglish instead of sending my draft to a human editor. However, there is one
function that will never be replaced by machine or AI because of its very nature: human's overs
over machines, processes, or people themselves. Pavle Avramović's (2023) "Digital Transformatio
Financial Regulators and the Emergence of Supervisory Technologies (SupTech): A Case Study of t
Financial Conduct Authority" provides ample reasons why human oversight is always needed for fi
markets, especially with their adoption of advanced technologies. For example, human oversight
to ensure the functionality of the technologies as intended, curtail their unintended consequen
their abusive and malicious use by some people. To do this well, human oversight itself should
technologies, as long as we understand how these technologies fulfill our supervision and regul
6Harvard Data Science Review • Issue 5.2, Spring 2023 Data Science and Engineering With Human i

Loop, and Above the Loop

Avramović's article provides a rather revealing case study to demonstrate how this can be done through a theoretical lens by examining institutional factors and dynamic capabilities. It also overview of the emerging field of SupTech, as well as an outlook for the field based on the les research, regulators, and industry.

Indeed, learning from lessons is another human endeavor that cannot be replaced by machine or A simple reason: humans need to internalize the lessons for themselves in order to actualize the learning, which is very different from its meaning as in the phrase 'machine learning.' Machine algorithmic process of revealing patterns and organizing knowledge (hopefully). Learning from a requires a changing of behavior together with a changing of the tendency of having such behavio that either change is sustainable. The need of both changes to actualize the learning is perhap by the joke or reality for some: 'Quitting smoking is easy-I have done that many times.' Or in ChatGPT, it will apologize profusely when told that its answer is wrong, but that does not nece that it would change its answer. Even when it does correct its answer in a particular case, it kind mistakes when prompted with a similar but differently worded instruction, as demonstrated persistent ability to 'hallucinate' (e.g., by making up many plausible-looking but hopelessly w biosketches.)

Granted, sometimes we humans do no better than ChatGPT in this regard, or do not want to do bet comes to changing behaviors because of negative consequences. But even in that regard, humans a step ahead of ChatGPT, at least for now, as we often rationalize our behaviors by changing our metrics, perhaps subconsciously. However, when our humanity and humility are at their best, we actualize the lessons learned, we turn them into insights to generate principles for success. T "Frontline Data Science: Lessons Learned From a Pandemic" by Emily A. Beck, Hannah Tavalire, an Searcy (2023) does exactly that. The COVID-19 pandemic highlights both the importance and chall using data science for disaster response, with lack of accessibility, training, and flexibility problematic. Beck et al.'s article outlines the authors' experience building a COVID-19 Monitor Assessment Program and provides lessons learned and principles for success for future data scie to contribute to emergency response and disaster recovery efforts. It provides a particularly u because neither Beck nor Searcy started with the pertinent experience or training in epidemiolo care, one being a geneticist and the other a particle physicist. Like many data scientists, we strong desire to do something, as we know we can and should contribute to the fight against the we didn't or don't know how. The article's delineation from getting started to informing future

therefore is especially welcome and helpful.

In particular, the lessons from the article have broader implications than just dealing with pa
disasters. For example, for anyone who truly cares about data science done right and has gotte
dirty, the main lesson of the article will resonate extremely well (replacing "testing" with ‘s

Loop, and Above the Loop

"The most important lesson learned … was that the data science team needed to be involved in th
every aspect of testing for ensuring accuracy and quality assurance of data pipelines" (Beck et
article makes it clear that this necessity is mainly created by humans. Human errors are unavoi
typos on patients’ data to system glitches preventing prompt reporting to state agencies. But "
rapidly craft quality assurance and compliance solutions to catch and fix errors." Together, th
experience and their four major lessons and five principles for success provide an exceedingly
concrete demonstration that data science is of the people, by the people, and for the people.

Disclosure Statement

Xiao-Li Meng has no financial or non-financial disclosures to share for this editorial.

References

Avramović, P. (2023). Digital transformation of financial regulators and the emergence of super
technologies (SupTech): A case study of the U.K. financial conduct authority. Harvard Data Scie
5(2). https://doi.org/10.1162/99608f92.7a329be7

Beck, E. A., Tavalire, H., & Searcy, J. (2023). Frontline data science: Lessons learned from a
Harvard Data Science Review, 5(2). https://doi.org/10.1162/99608f92.eea782bf

Connolly, A., Hellerstein, J., Alterman, N., Beck, D., Fatland, R., Lazowska, E., Mandava, V.,
(2023). Software engineering practices in academia: Promoting the 3Rs-readability, resilience,
Harvard Data Science Review, 5(2). https://doi.org/10.1162/99608f92.018bf012

Davison, A. C. (2023). Some reminiscences of David Cox. Harvard Data Science Review, 5(2).
https://doi.org/10.1162/99608f92.85038493

Meng, X.-L. (2019). Data Science: An artificial ecosystem. Harvard Data Science Review, 1(1).
https://doi.org/10.1162/99608f92.ba20f892

Richardson, S., & Wermuth, N. (2023). Remembering David Cox. Harvard Data Science Review, 5(2)
https://doi.org/10.1162/99608f92.35281d57

Siegel, E. (2023). To deploy machine learning, you must manage operational change-
Here is how UPS got it

right. Harvard Data Science Review, 5(2). https://doi.org/10.1162/99608f92.73fc83a3

Shah, K. M., & Saeed, A. A. (2023). Innovating a centuries-old sport: How emerging data analyti redefining cricket. Harvard Data Science Review, 5(2). https://doi.org/10.1162/99608f92.1ea134

8Harvard Data Science Review • Issue 5.2, Spring 2023 Data Science and Engineering With Human i Loop, and Above the Loop

Stelmaszak, M., & Kline, K. (n.d.). Managing embedded data science teams for success: How manag navigate the advantages and challenges of distributed data science. Harvard Data Science Review https://doi.org/10.1162/99608f92.1f068331

References

Avramović, P. (2023). Digital Transformation of Financial Regulators and the Emergence of Super Technologies (SupTech): A Case Study of the U.K. Financial Conduct Authority. Harvard Data Scie Review, 5(2). https://doi.org/10.1162/99608f92.7a329be7

Beck, E. A., Tavalire, H., & Searcy, J. (2023). Frontline Data Science: Lessons Learned From a Harvard Data Science Review, 5(2). https://doi.org/10.1162/99608f92.eea782bf

Blitzstein, J. K. (2023). Sir David R. Cox: A Beautiful Mind With a Beautiful Heart. Harvard Da Review, 5(2). https://doi.org/10.1162/99608f92.74fb2783

Connolly, A., Hellerstein, J., Alterman, N., Beck, D., Fatland, R., Lazowska, E., Mandava, V., (2023). Software Engineering Practices in Academia: Promoting the 3Rs-Readability, Resilience, Reuse . Harvard Data Science Review, 5(2). https://doi.org/10.1162/99608f92.018bf012

Davison, A. C. (2023). Some Reminiscences of David Cox. Harvard Data Science Review, 5(2). https://doi.org/10.1162/99608f92.85038493

Meng, X.-L. (2019). Data Science: An Artificial Ecosystem. Harvard Data Science Review, 1(1). https://doi.org/10.1162/99608f92.ba20f892

Richardson, S., & Wermuth, N. (2023). Remembering David Cox. Harvard Data Science Review, 5(2) https://doi.org/10.1162/99608f92.35281d57

Shah, K. M., & Saeed, A. A. (2023). Innovating a Centuries-Old Sport: How Emerging Data Analyti Tools Are Redefining Cricket. Harvard Data Science Review, 5(2). https://doi.org/10.1162/99608f92.1ea13423

Siegel, E. (2023). To Deploy Machine Learning, You Must Manage Operational Change-

Here Is How
UPS Got It Right . Harvard Data Science Review, 5(2). https://doi.org/10.1162/99608f92.73fc83a3
Stelmaszak, M., & Kline, K. (2023). Managing Embedded Data Science Teams for Success: How Manag
Can Navigate the Advantages and Challenges of Distributed Data Science. Harvard Data Science Re
5(2). https://doi.org/10.1162/99608f92.1f068331
9

(4) Extract the text from pages 2 to 5.

```
with pdfplumber.open("/Users/jaspreetbrar/Desktop/Classes/Stats3DA3/Assignment2/paper.pdf") as
    # Extract text from pages 2 to 5
    text = ""
    for i in range(1, 5):
        page = pdf.pages[i]
        text += page.extract_text()
    print(text)
```

Harvard Data Science Review • Issue 5.2, Spring 2023 Data Science and Engineering With Human in
Loop, and Above the Loop
The term human-in-the-loop (HITL) generally refers to the need for human interaction, intervent
judgment to control or change the outcome of a process, and it is a practice that is being inc
emphasized in machine learning, generative AI, and the like. For readers who have not enjoyed m
assisted learning or who are comfortable with their human intelligences, it would not surprise
you would wonder, upon hearing the term HITL, ‘Wait, when did we take humans out of the loop?’
diversity of human minds guarantees that such a question would generate reactions ranging from
to being despised. Regardless, however, the arrival of ChatGPT and other AI chatbots has made t
academic debate of machine intelligence versus human intelligence into a household conversation
therefore an apt time to turn such a rhetorical question into retrospection and reflection.
With my human thinking in the loop, I probably can take any issue of HDSR to remind ourselves o
we humans play in the data science (DS) ecosystem. Indeed, as I wrote in the inaugural editoria
Science: An Artificial Ecosystem," "the term ‘artificial’ highlights both the fact that DS is a
and that it depends critically on computing advances" (Meng, 2019). (The same can and should be
data engineering (DE), which can advance faster than DS, as we witnessed recently with the prog
GPT-n to GPT-(n+1), n<4.) However, the current issue is particularly fitting for this contempla
the first HDSR issue with a special theme dedicated to a beautiful mind that came with a beauti
that yet to be applied to any machine. As such, I ask for readers’ indulgence-
and my fellow data scientists’
forgiveness-for creating sibling rivalries to HITL.
Human in the Loop: From Developers to Deployers

As cliché as this may sound, humans are always in the loop for advancing data science or engine
fronts, literally and figuratively. Indeed, for those who don't mind the label of anthropocentr
paraphrasing 'DS/DE are of the people, by the people, and for the people' might be a more arous
this editorial. Data are typically digital representations of some people's mental activities o
endeavors, and they are collected because some (other) people see the benefit to doing so. The
for the data subjects, the data collectors, or other people, which often include those yet to b
and optimize the benefit, yet more groups of people would be required for developing methods, i
algorithms, interpreting and deploying the results, and so on, in order to reap the perceived b
For the Recreations in Randomness column, Kavya Mehul Shah and Ammaar Ahmed Saeed's (2023) acc
on "Innovating a Centuries-Old Sport: How Emerging Data Analytics Tools Are Redefining Cricket"
a concrete illustration of such human effort. For example, cricket highlights are short video c
collected because they can benefit players, coaches, and organizations. Traditionally the creat
highlights "requires sports broadcasters and video editors to manually parse through hours of i
identify exciting moments"-clearly a not very exciting process to say the least. Automated high
generation is then a very welcome advance that requires both data science (e.g., semantic segme

identify "exciting moments") and data engineering (e.g., effective software for routine applica
progress has been made in the last 20 years, as reviewed in the article. Yet the 2 decades of p
indicate the amount of human effort required to automate the creation process, not to mention h
needed in translating highlights into the desired benefit.
As another indication of the human effort, when I initially opened the document file for this c
prepare for this editorial, I was a bit surprised to see that the word count was about 4,600-
far exceeding the
normally expected 2,500 words for a column. In case a reader wonders why I was not aware of the
submission until the end stages, I am grateful to the interim Co-Editor-in-Chiefs, Francesca Do
David Parkes, whose dedication allowed me to take a real sabbatical (which incidentally gave me
appreciate the benefit of sports). It turns out that the article has 65 references (which are n
the 2,500 limit), which is an unusually extensive list even for a regular article. But it is fi
since the list provides a chronological picture of human effort and progress in reshaping a hal
sport via data science and engineering. Indeed, it is almost certain that the list is still onl
sample.

… and From Software Engineers to Data Science Managers

The sabbatical year also afforded me the opportunity to join an international collaboration on conditions globally in general, and in Africa in particular, by combining ground-level survey image data via deep learning. The data involved are treasures not only for the mission of the also for data science and engineering education in general. They exhibit almost the full specti and defects as routinely encountered in practice, including selection bias, measurement errors information due to privacy protection, mismatches in spatial, temporal, and contextual resoluti incomplete documentation of data provenance, just to provide a partial list. The collaboration multi-faceted, involving multiple disciplines (e.g., computer science, engineering, sociology, members in all career stages from undergraduate to senior researchers with a variety of working are driven by different interests or incentives. This collaborative experience, which is still my appreciation of two kinds of humans in the loop, so to speak.

One kind of profession is software engineers, whose technical skills and teamwork (or lack ther govern the pace for progress. For complex projects, many methodological ideas need to be (stres These tests can take weeks or minutes, all depending on the engineers' experiences and skills. am involved in, a clever implementation circumventing a map computation with mismatched resolut up the test by several weeks. With those experiences still fresh, I literally thought I was day saw the article "Software Engineering Practices in Academia: Promoting the 3Rs-

Readability, Resilience, and

Reuse" by Andrew Connolly, Joseph Hellerstein, Naomi Alterman, David Beck, Rob Fatland, Ed Lazo Vani Mandava, and Sarah Stone (2023). The first section title says it all: "Good Data Science I Quality Software Engineering." Its diagnosis of the major issues in academia is spot on, as aca

"address their software development needs by hiring graduate students, often without any formal software engineering," and the vast majority of the software projects are created in an often-i setting: single developer, single user, and single study. For any academics who are interested quality of their data science projects and of their life by reducing work inefficiency, the tim digesting this article is likely to be negligible compared to the time saved moving forward, be provides real-life successes, reasons, and recommendations-with specific recipes-

for adopting the '3R'

principle.

The other kind of profession is data science managers. With a team of diverse expertise, intere
habits and schedules, and the nature of the project being multi-phased, having an all-around da
manager is another key for both the quality and speed of the project. For the collaboration I a
things changed substantially once we had such a data science manager in place. A data science m
more like a chief of staff to the intellectual leaders of a data science collaboration, which m
manager needs to have sufficient technical, personal, and organizational skills to be effective
a data science manager is to help to identify, establish, and sustain a collaborative platform
that is conducive to making the whole more than the sum of its parts, or at least not less. The
happen (and did happen even in my very limited experience) when there are miscommunications,
misunderstandings, and mismatched expectations among different team members or groups. The mana
needs to be a strategic thinker with sufficient know-how to recognize or anticipate weak links
or preventive measures accordingly, and ultimately creative an incentivizing infrastructure to
efficiency and team members' desire to further the collaboration.

With these reflections from my recent experiences, I was instantly attracted by another article
"Managing Embedded Data Science Teams for Success: How Managers Can Navigate the Advantages and
Challenges of Distributed Data Science," by Marta Stelmaszak and Kelsey Kline (2023). Through a
the authors summarize three advantages for the embedded structures, "1) agility in delivering c
better tailored analytics, 2) concentrated and cultivated data science expertise, and 3) emergi
solution ideas," as well as three disadvantages: "1) growing away from the enterprise infrastru
duplicating data science efforts, and 3) weakening links with strategic impact." Although the s
rather different from academic collaborations, the main conclusion of the article is the data s
the critical HITL to reap these advantages and curtail the disadvantages. To do so, the article
science managers need to have aptitudes for all three managing directions: within a team (e.g.
balance workload for members with varying expertise), vertically in an organization (e.g., inte
to maintain team visibility at all levels of the organization), and horizontally across the org
capability to develop networks with other data science teams).

The vital role of managing is furthered demonstrated in "To Deploy Machine Learning, You Must M
Operational Change-Here Is How UPS Got It Right" by Eric Siegel (2023). The article starts with

story of how a trash collection company dumped a university generated route-optimizing plan tha
number of trash tracks by 50%. I won't spoil the rest of the plot, which could easily be adapte

a Hollywood movie on machine learning or AI (perhaps with the help of GPT-n), but only to repea
metaphor Siegel used. The "rocket science" of machine learning for commercial use is easy. The
getting the end product or process launched, which requires changing humans' habits. That's not
science; that's beyond rocket science.

Human Behind the Loop: With Beautiful Minds and Beautiful Hearts

Much of the research of and discussions about HITL focus on the interaction between humans and
But as we just discussed, translating and transforming the advances in data science and enginee
value for human societies is a matter of interactions among humans. Arguably another kind of hu
interaction is even more important for the data science enterprise, namely the professional and
interactions among members of the data science ecosystem. A shining example is provided by the
theme in this issue celebrating the life and work of Sir David Cox, a giant in statistics and i
In his long and extremely distinguished career, David served as a global model in everything he
extremely deep, broad, and influential scholar, a much-loved advisor of over 60 doctorial stude
exceedingly effective editor for Biometrika for 25 years, and a well-respected academic and pro
(e.g., serving as Warden of Nuffield College, and as president of multiple professional societi
articles in the special theme document many interactions with David in a myriad of ways and acr
decades (and countries). Yet they reflect a common theme: the lasting impression and impact of
interactions. I am deeply grateful to the co-editors of this special theme, Sylvia Richardson a
Wermuth, for organizing and introducing (Richardson & Wermuth, 2023) such a rich and heartwarmi
collection, and to all contributors for sharing their personal stories with all of us, to which
My first interaction with David took place when I was a PhD student in late 1980s, where I co-a
article that was submitted to Biometrika. At that time, the review processes for most statistic
notoriously slow; it was rather common for the review process to take more than 6 months for th
Biometrika was an exception, despite having the smallest editorial board among all top-tier jou
that expectation, I was rather surprised when we heard from David within 2 weeks of submission
not a quick rejection letter. Rather, David had provided detailed comments and asked us to do a
he sent it out for a full review. I do not recall the details of David's comments, but I remem
being very impressed by the promptness and insightfulness of the comments, which led to a much-
revision, and ultimately my first publication in the tier-one journals in statistics.
Years later I had an opportunity to meet with David in person, when I could no longer hold my c
hence asked how he could handle so many articles personally, and yet with such efficiency. The

5

(5) Combine the text from these pages into a single string.

```
# Combine the text from pages 2 to 5 into a single string
combined_text = text.replace('\n', ' ')
print(combined_text)
```

```
Harvard Data Science Review • Issue 5.2, Spring 2023 Data Science and Engineering With Human in
and my fellow data scientists' forgiveness-for creating sibling rivalries to HITL. Human in the
clearly a not very exciting process to say the least. Automated highlights generation is then a
far exceeding the normally expected 2,500 words for a column. In case a reader wonders why I wa
Readability, Resilience, and Reuse" by Andrew Connolly, Joseph Hellerstein, Naomi Alterman, Dav
with specific recipes-for adopting the '3R' principle. The other kind of profession is data sci
Here Is How UPS Got It Right" by Eric Siegel (2023). The article starts with a telling 4Harvard
```

(6) Split the string by lines using **\n**.

```
lines = combined_text.split('\n')
print(lines)
```

```
['Harvard Data Science Review • Issue 5.2, Spring 2023 Data Science and Engineering With Human
and my fellow data scientists' forgiveness-for creating sibling rivalries to HITL. Human in the
clearly a not very exciting process to say the least. Automated highlights generation is then a
far exceeding the normally expected 2,500 words for a column. In case a reader wonders why I wa
Readability, Resilience, and Reuse" by Andrew Connolly, Joseph Hellerstein, Naomi Alterman, Dav
with specific recipes-for adopting the '3R' principle. The other kind of profession is data sci
Here Is How UPS Got It Right" by Eric Siegel (2023). The article starts with a telling 4Harvard
```

(7) Create a pandas data frame named `df` with a column labeled `line` containing the split lines.

```
df = pd.DataFrame({'line': lines})
print(df)
```

```
                                                line
0  Harvard Data Science Review • Issue 5.2, Sprin...
```

(8) Break each line into individual words.

```
df['words'] = df['line'].apply(lambda x: re.findall(r'\b\w+\b', x.lower()))
df = df.explode('words')
print(df)
```

```
                                                line       words
0   Harvard Data Science Review • Issue 5.2, Sprin...     harvard
0   Harvard Data Science Review • Issue 5.2, Sprin...        data
0   Harvard Data Science Review • Issue 5.2, Sprin...     science
0   Harvard Data Science Review • Issue 5.2, Sprin...      review
0   Harvard Data Science Review • Issue 5.2, Sprin...       issue
..                                                ...         ...
0   Harvard Data Science Review • Issue 5.2, Sprin...  efficiency
0   Harvard Data Science Review • Issue 5.2, Sprin...         the
0   Harvard Data Science Review • Issue 5.2, Sprin...      answer
0   Harvard Data Science Review • Issue 5.2, Sprin...          he
0   Harvard Data Science Review • Issue 5.2, Sprin...           5

[2366 rows x 2 columns]
```

(9) Convert each word into a separate row in the data frame.

```
df = df.explode('words')
print(df)
```

```
                                                line       words
0   Harvard Data Science Review • Issue 5.2, Sprin...     harvard
0   Harvard Data Science Review • Issue 5.2, Sprin...        data
0   Harvard Data Science Review • Issue 5.2, Sprin...     science
0   Harvard Data Science Review • Issue 5.2, Sprin...      review
0   Harvard Data Science Review • Issue 5.2, Sprin...       issue
..                                                ...         ...
0   Harvard Data Science Review • Issue 5.2, Sprin...  efficiency
0   Harvard Data Science Review • Issue 5.2, Sprin...         the
0   Harvard Data Science Review • Issue 5.2, Sprin...      answer
0   Harvard Data Science Review • Issue 5.2, Sprin...          he
0   Harvard Data Science Review • Issue 5.2, Sprin...           5

[2366 rows x 2 columns]
```

(10) Convert all words to lowercase.

```
df['words'] = df['words'].str.lower()
print(df)
```

```
                                                line       words
0   Harvard Data Science Review • Issue 5.2, Sprin...     harvard
0   Harvard Data Science Review • Issue 5.2, Sprin...        data
0   Harvard Data Science Review • Issue 5.2, Sprin...     science
0   Harvard Data Science Review • Issue 5.2, Sprin...      review
0   Harvard Data Science Review • Issue 5.2, Sprin...       issue
..                                                ...         ...
0   Harvard Data Science Review • Issue 5.2, Sprin...  efficiency
0   Harvard Data Science Review • Issue 5.2, Sprin...         the
0   Harvard Data Science Review • Issue 5.2, Sprin...      answer
0   Harvard Data Science Review • Issue 5.2, Sprin...          he
0   Harvard Data Science Review • Issue 5.2, Sprin...           5

[2366 rows x 2 columns]
```

(11) Remove stop words.

```
# Define the stop words
stop_words = set(stopwords.words('english'))


# Remove stop words from the 'words' column
df = df[~df['words'].isin(stop_words)]
print(df)
```

```
                                                line       words
0   Harvard Data Science Review • Issue 5.2, Sprin...     harvard
0   Harvard Data Science Review • Issue 5.2, Sprin...        data
0   Harvard Data Science Review • Issue 5.2, Sprin...     science
0   Harvard Data Science Review • Issue 5.2, Sprin...      review
0   Harvard Data Science Review • Issue 5.2, Sprin...       issue
..                                                ...         ...
0   Harvard Data Science Review • Issue 5.2, Sprin...  personally
0   Harvard Data Science Review • Issue 5.2, Sprin...         yet
0   Harvard Data Science Review • Issue 5.2, Sprin...  efficiency
0   Harvard Data Science Review • Issue 5.2, Sprin...      answer
0   Harvard Data Science Review • Issue 5.2, Sprin...           5

[1369 rows x 2 columns]
```

(12) Remove unsuitable words using the following steps:

Hint:

(i) Remove rows where the `word` column contains punctuation using

- `str.contains(r'[,.•‘"":’;\(\)\[\]]', regex=True)]`

(ii) Remove rows where the `word` column contains numbers using: - `str.contains(r'\d', regex=True)]`

(iii) Remove rows where the `word` column contains single letters using: - `str.contains(r'^[a-z]$', regex=True)]`

```
# Remove rows where the 'words' column contains punctuation
df = df[~df['words'].str.contains(r'[,.•‘"":’;\(\)\[\]]', regex=True)]

# Remove rows where the 'words' column contains numbers
df = df[~df['words'].str.contains(r'\d', regex=True)]

# Remove rows where the 'words' column contains single letters
df = df[~df['words'].str.contains(r'^[a-z]$', regex=True)]

print(df)
```

```
                                                line       words
0   Harvard Data Science Review • Issue 5.2, Sprin...     harvard
0   Harvard Data Science Review • Issue 5.2, Sprin...        data
0   Harvard Data Science Review • Issue 5.2, Sprin...     science
0   Harvard Data Science Review • Issue 5.2, Sprin...      review
0   Harvard Data Science Review • Issue 5.2, Sprin...       issue
..                                                ...         ...
0   Harvard Data Science Review • Issue 5.2, Sprin...    articles
0   Harvard Data Science Review • Issue 5.2, Sprin...  personally
0   Harvard Data Science Review • Issue 5.2, Sprin...         yet
0   Harvard Data Science Review • Issue 5.2, Sprin...  efficiency
```

```
0   Harvard Data Science Review • Issue 5.2, Sprin...      answer

[1302 rows x 2 columns]
```

(13) Create a term-frequency data frame.

Hint:

(i) Calculate the frequency of each unique word using: `value_counts().reset_index()`

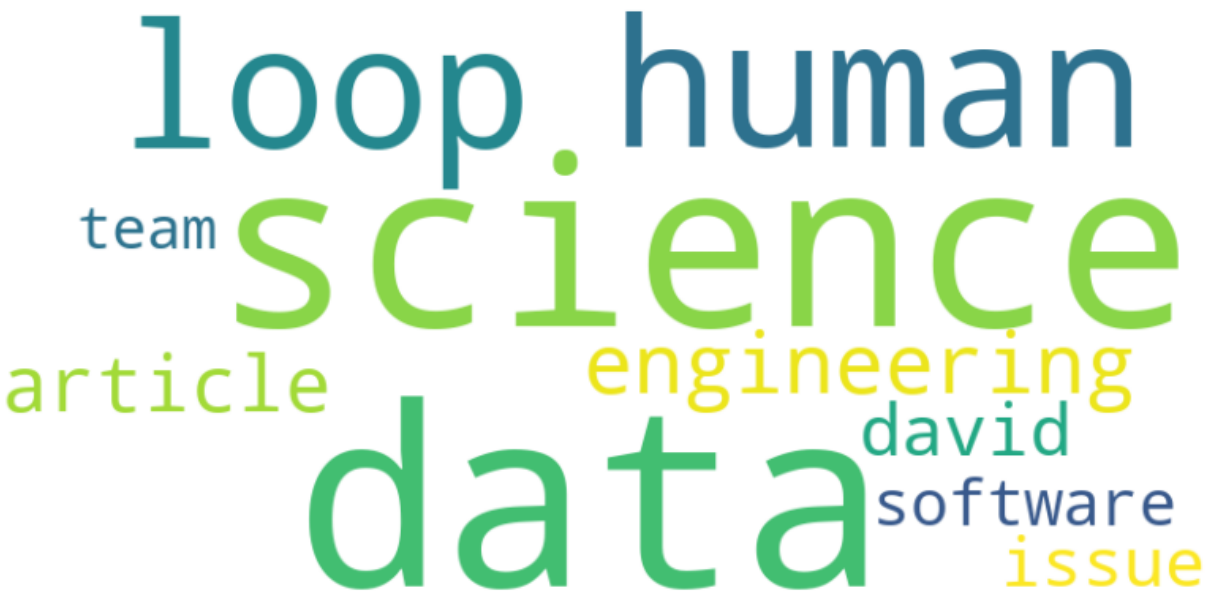
(ii) Save the result in a DataFrame called `freq`.

```
# Calculate the frequency of each unique word
freq = df['words'].value_counts().reset_index()

# Rename the columns for better readability
freq.columns = ['word', 'frequency']

print(freq)
```

```
            word  frequency
0           data         44
1        science         38
2          human         20
3           loop         19
4    engineering         14
..           ...        ...
783        image          1
784    treasures          1
785      mission          1
786    education          1
787       answer          1

[788 rows x 2 columns]
```

(14) Generate a word cloud for the most frequently occurring words (e.g., the top 10 words).

```
# Generate a word cloud for the top 10 most frequently occurring words
top_words = freq.head(10)
wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencie


# Plot the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```



(15) Write a summary paragraph (at least two statements) about your word cloud. The summary can include any limitations of your analysis and provide context based on the chosen text.

The word cloud generated shown that data science is the most prevalent word in the PDF as it is the largest. The second most common words found in the PDF is loop and human.

Question 2

Greenhouse gases (GHGs) play a significant role in global warming by capturing and retaining solar heat energy, leading to elevated global temperatures. In 2004, Canada launched the Greenhouse Gas Reporting Program (GHGRP) to monitor and record emissions from facilities that release 10 kilotonnes or more of greenhouse gases, measured in CO2-equivalent units. Facilities meeting this threshold are required to submit annual reports to Environment and Climate Change Canada. The dataset is publicly accessible through Canada's Open Government Portal: Greenhouse Gas Reporting Program (GHGRP) - Facility Greenhouse Gas (GHG) Data.

For Question 2, we have downloaded the dataset PDGES-GHGRP-GHGEmissionsGES-2004-Present.csv from the portal.

This analysis focuses on creating a Shiny App to explore trends in greenhouse gas emissions across Canada's provinces and territories, measured in CO2-equivalent units.

Data dictionary:

The dataset, spanning from 2004 to the present, includes emissions data (in tonnes and CO2-equivalent tonnes) for each facility, categorized by gas type, including carbon dioxide (CO2), methane (CH4), nitrous oxide (N2O), hydrofluorocarbons (HFCs), perfluorocarbons (PFCs), and sulphur hexafluoride (SF6). It also provides the province or territory where each facility is located. For further details, refer to the Greenhouse Gas Reporting Program (GHGRP) - Facility Greenhouse Gas (GHG) Data.

Pre-Processing Steps

To simplify the task of creating a Shiny App, we have pre-processed the data as follows:

We start by importing the necessary libraries for data transformation:

```
import numpy as np
import pandas as pd
import re
```

Next, we read the downloaded dataset in CSV format with the specified encoding (latin1):

```
df = pd.read_csv("/Users/jaspreetbrar/Desktop/Classes/Stats3DA3/Assignment2/PDGES-GHGRP-GHGEmis
```

The column names in the dataset are a mix of English and French. We use the `clean_column_names()` function to standardize the column names by removing French names, non-ASCII characters, and unnecessary symbols.

Here is the `clean_column_names()` function:

```
# clean_column_names function
def clean_column_names(column_names):
    cleaned_names = []
    # loop through each column name
    for name in column_names:
        # convert names to ASCII and remove non-ASCII characters
        name = name.encode('ascii', 'ignore').decode('ascii')
        # remove everything after '/' (French column name)
        name = re.sub(r'/.*', '', name)
        # remove parentheses
        name = re.sub(r'[()]', '', name)
        # remove extra whitespace
        name = ' '.join(name.split())
        cleaned_names.append(name)
        # return new column names
    return cleaned_names
```

We then apply this function to clean the column names in the DataFrame:

```
df.columns = clean_column_names(df.columns)
```

Next, we select the relevant columns for the analysis:

- `Reference Year` - the year GHG gas emission was recorded.

- `GHGRP ID No.` - the facility identity.
- `Facility Province or Territory` - province or territory of the facility.
- `CO2 tonnes` - emissions (in tonnes and tonnes of CO2 eq.) of carbon dioxide (CO2).
- `CH4 tonnes` - emissions (in tonnes and tonnes of CO2 eq.) of methane.
- `N2O tonnes` - emissions (in tonnes and tonnes of CO2 eq.) of nitrous oxide.
- `SF6 tonnes`- emissions (in tonnes and tonnes of CO2 eq.) of sulphur hexafluoride.
- `HFC Total tonnes CO2e` - emissions (in tonnes and tonnes of CO2 eq.) of hydrofluorocarbons.
- `PFC Total tonnes CO2e` - emissions (in tonnes and tonnes of CO2 eq.) of perfluorocarbons.

```
selected_cols = [
    "Reference Year", "GHGRP ID No.", "Facility Province or Territory",
    "CO2 tonnes", "CH4 tonnes", "N2O tonnes", "SF6 tonnes",
    "HFC Total tonnes CO2e", "PFC Total tonnes CO2e"
]

df= df[selected_cols]
```

We rename the columns to make them more concise and consistent:

```
df.rename(columns={
    "Reference Year": "Year",
    "GHGRP ID No.": "Facility_ID",
    "Facility Province or Territory": "Province_Territory",
    "CO2 tonnes": "CO2",
    "CH4 tonnes": "CH4",
    "N2O tonnes": "N2O",
    "SF6 tonnes": "SF6",
    "HFC Total tonnes CO2e": "HFC",
    "PFC Total tonnes CO2e": "PFC"
}, inplace=True)

print(df.head())
```

```
   Year Facility_ID Province_Territory           CO2          CH4        N2O  \
0  2022      G12721              Quebec  2.473800e+02   956.162100   0.001900
1  2022      G10001              Quebec  3.887707e+04     0.760073   0.675020
2  2022      G10003             Alberta  3.368673e+05   482.571900   8.587500
3  2022      G10004             Ontario  9.980300e+02     0.243572   0.025105
4  2022      G10006             Alberta  1.101755e+06  4295.113188  16.554313

   SF6  HFC  PFC
0  0.0  0.0  0.0
1  0.0  0.0  0.0
2  0.0  0.0  0.0
3  0.0  0.0  0.0
4  0.0  0.0  0.0
```

Finally, we save the pre-processed data to a new CSV file:

```
df.to_csv("cleaned_GHG_Emissions.csv", index=False)
```

The pre-processed dataset is now available for analysis and can be accessed at:

https://raw.githubusercontent.com/PratheepaJ/datasets/refs/heads/master/cleaned_GHG_Emissions.csv.

You will use this dataset for Question 2.

Next Steps

The following questions guide you through creating a Shiny App to explore trends in CO2, CH4, and N2O emissions across provinces and territories in Canada from 2004 to 2022.

(1) Read the pre-processed data from the provided link.

```
import requests
import io
import urllib3

# Suppress only the single warning from urllib3 needed.
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

url = "https://raw.githubusercontent.com/PratheepaJ/datasets/refs/heads/master/cleaned_GHG_Emis

response = requests.get(url, verify=False)
data = response.content.decode('latin1')
df = pd.read_csv(io.StringIO(data))
df.head()
```

	Year	Facility_ID	Province_Territory	CO2	CH4	N2O	SF6	HFC	PFC
0	2022	G12721	Quebec	2.473800e+02	956.162100	0.001900	0.0	0.0	0.0
1	2022	G10001	Quebec	3.887707e+04	0.760073	0.675020	0.0	0.0	0.0
2	2022	G10003	Alberta	3.368673e+05	482.571900	8.587500	0.0	0.0	0.0
3	2022	G10004	Ontario	9.980300e+02	0.243572	0.025105	0.0	0.0	0.0
4	2022	G10006	Alberta	1.101755e+06	4295.113188	16.554313	0.0	0.0	0.0

(2) Ensure that the year variable is in the correct format. If not, convert it to the date-time format and extract the year. Replace the original 'Year' variable with the extracted year.

Hint: Use the following command to convert the year:

```
# Ensure the 'Year' column is in the correct format (extract the year)
df.dtypes
```

```
Year                    int64
Facility_ID            object
Province_Territory     object
CO2                   float64
CH4                   float64
N2O                   float64
SF6                   float64
HFC                   float64
PFC                   float64
dtype: object
```

(3) Some territories may have no facilities reported in early years. Group the data by `Year` and `Province_Territory` to count distinct `Facility_ID` values. Find which territories are missing in 2004.

Hint: Use the following code to group the data and find missing territories:

```
# Group by Year and Province/Territory to count distinct facilities
df_facilities = df.groupby(['Year', 'Province_Territory']).agg(
    facilities=('Facility_ID', 'nunique')
).reset_index()



# List all territories from the dataset
all_territories = df['Province_Territory'].unique()

# Identify which territories reported facilities in 2004
facilities_2004 = df_facilities[df_facilities['Year'] == 2004]['Province_Territory'].unique()

# Find territories missing in 2004
missing_territories_2004 = set(all_territories) - set(facilities_2004)
print("Territories missing in 2004:", missing_territories_2004)
```

```
Territories missing in 2004: {nan, 'Yukon', 'Nunavut'}
```

(4) Find the earliest and latest year emissions were recorded.

```
min_year = df['Year'].min()
max_year = df['Year'].max()
print("Earliest year of emissions recorded:", min_year)
print("Latest year of emissions recorded:", max_year)
```

```
Earliest year of emissions recorded: 2004
Latest year of emissions recorded: 2022
```

(5) Group the data by `Year` and `Province_Territory` and sum the emissions of CO2, CH4, and N2O for each province.

Hint: Use the following code to calculate the total emissions:

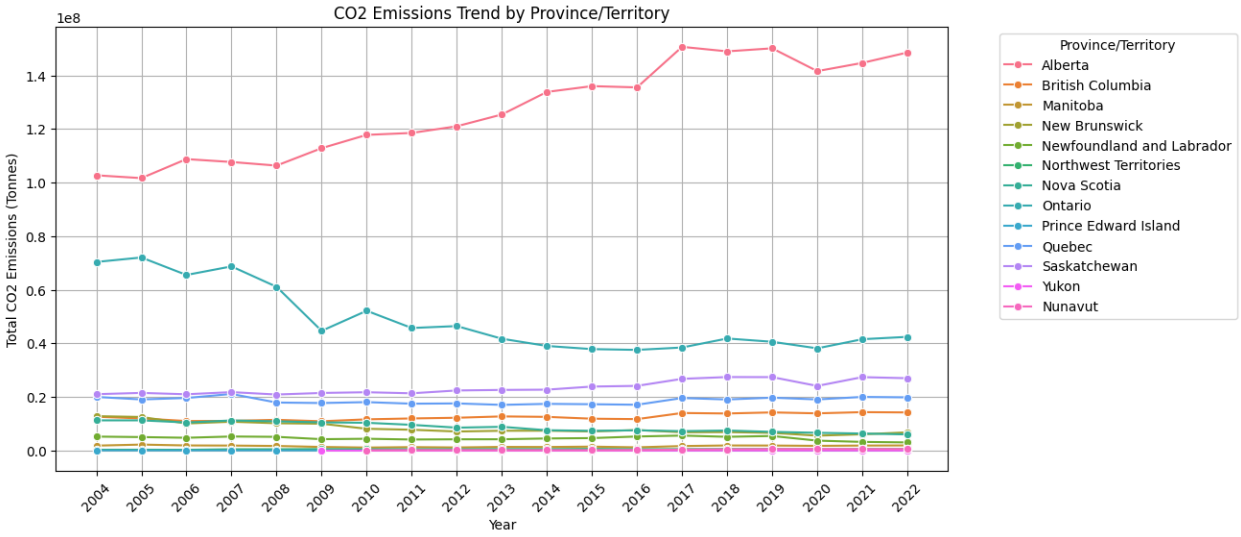
```
df_grouped = df.groupby(['Year', 'Province_Territory']).agg(
    CO2=('CO2', 'sum'),
    CH4=('CH4', 'sum'),
    N2O=('N2O', 'sum')
).reset_index()
print(df_grouped)
```

```
     Year         Province_Territory           CO2           CH4          N2O
0    2004                    Alberta  1.027572e+08  135337.195168  5120.460777
1    2004           British Columbia  1.263764e+07   34007.739560   673.717199
2    2004                   Manitoba  1.859467e+06   25508.017000   210.645008
3    2004               New Brunswick  1.286296e+07    1426.663022   270.524004
4    2004  Newfoundland and Labrador  5.233507e+06    5123.829998    88.760300
..    ...                         ...           ...            ...          ...
231  2022                    Ontario  4.245634e+07  168447.810213  1682.133227
232  2022       Prince Edward Island  5.014224e+04      15.055154     5.971497
233  2022                     Quebec  1.988334e+07   56517.375148   820.326410
234  2022               Saskatchewan  2.702207e+07   42829.795415   671.062091
235  2022                      Yukon  2.833794e+04       3.118111     0.340188

[236 rows x 5 columns]
```

(6) Plot the CO2 changes over the years for each province and territory, using colored lines to differentiate between them.

Note: you will use the dataset obtained in (5) for this plot.

```
import seaborn as sns
import matplotlib.pyplot as plt


plt.figure(figsize=(12, 6))
sns.lineplot(data=df_grouped, x='Year', y='CO2', hue='Province_Territory', marker='o')
plt.title('CO2 Emissions Trend by Province/Territory')
plt.xlabel('Year')
plt.ylabel('Total CO2 Emissions (Tonnes)')
plt.legend(title='Province/Territory', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(ticks=np.arange(min_year, max_year + 1, 1), rotation=45)
plt.grid(True)
plt.show()
```



(7) Provide a description of the CO2 emission trend across provinces and territories based on the plot in (6).

The plot reveals that CO2 emissions vary considerably across provinces and territories over time. Some regions show a steady increase in emissions, while others display fluctuations—likely reflecting differences in industrial activity and reporting practices.

(8) Develop a Shiny app that allows the user to input a start year (from 2004 to 2022), an end year (from 2004 to 2022), and select a gas type (CO2, CH4, N2O).

- Use `ui.input_select` to allow the user to specify the start year (between 2004 and 2022).
- Use `ui.input_select` to allow the user to specify the end year (between 2004 and 2022).
- Use `ui.input_select` to allow the user to select the gas type (CO2, CH4, or N2O).

You can start by using the following Shiny app template to structure your app. When writing the app in `app.py`, remove the template instructions and replace them with your implementation.

You will also need to copy-paste your `app.py` in your assignment answers, similar to the template provided here:

```
# Shiny App Development using Python's Shiny framework
from shiny import App, render, ui

app_ui = ui.page_fluid(
    ui.input_select(
        id='emissiontype',
        label='Choose emission type',
        choices=['CO2', 'CH4', 'N2O'],
        selected='CO2'
    ),
    ui.input_select(
        "start_year",
        "Start Year",
        [str(year) for year in range(2004, 2023)]
    ),
    ui.input_select(
        "end_year",
        "End Year",
        [str(year) for year in range(2004, 2023)]
    ),
```

```
    ui.output_plot('myplot')
)


def server(input, output, session):
    @output
    @render.plot
    def myplot():
        # Read the pre-processed GHG data
        df = pd.read_csv("https://raw.githubusercontent.com/PratheepaJ/datasets/refs/heads/mast
        # Format the 'Year' column
        df['Year'] = pd.to_datetime(df['Year'], format='%Y').dt.year
        # Filter data based on user-selected start and end years
        start_year = int(input.start_year())
        end_year = int(input.end_year())
        df_filtered = df[(df['Year'] >= start_year) & (df['Year'] <= end_year)]
        # Group data by Year and Province/Territory, summing emissions for each gas type
        df_grouped = df_filtered.groupby(['Year', 'Province_Territory']).agg(
            CO2=('CO2', 'sum'),
            CH4=('CH4', 'sum'),
            N2O=('N2O', 'sum')
        ).reset_index()
        # Determine the emission type to plot
        emission_type = input.emissiontype()
        plt.figure(figsize=(10, 6))
        sns.lineplot(data=df_grouped, x='Year', y=emission_type, hue='Province_Territory', mark
        plt.title(f'{emission_type} Emissions Trend')
        plt.xlabel('Year')
        plt.ylabel(f'Total {emission_type} Emissions (Tonnes)')
        plt.legend(title='Province/Territory', bbox_to_anchor=(1.05, 1), loc='upper left')
        plt.xticks(ticks=np.arange(df_grouped['Year'].min(), df_grouped['Year'].max() + 1, 1),
        plt.grid(True)
```

```
        return plt.gcf()


app = App(app_ui, server)
```

(9) Deploy your Shiny App at https://www.shinyapps.io/. Then, provide the link to the App.

For example, the link to my app is https://pratheepaj.shinyapps.io/my_app/.

The link to my app

(I used Github Copilot for assistance)