Data analyses, including summarizing and visualizing data

```
from google.colab import files
import pandas as pd

# Upload the file
uploaded = files.upload()

file_name = list(uploaded.keys())[0]
df = pd.read_csv(file_name)

# Display the DataFrame
df
```

```
<IPython.core.display.HTML object>
```

```
Saving travela .csv to travela .csv
```

{"type":"dataframe","variable_name":"df"}

```
import pandas as pd
from IPython.display import display, HTML

# Read the dataset
df = pd.read_csv('travela  (2).csv')

# Display the first 10 rows and columns as an HTML table
html_table = df.head(10).to_html(index=False)

# Render the HTML table using the display function
display(HTML(html_table))
```

```
<IPython.core.display.HTML object>
```

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

df = pd.read_csv('travela  (1).csv')
```

Explore Data

```
print(df.head())
```

```
   year                             mandate_description_en  \
0  2020  Expenditures on travel, hospitality and confer...
1  2021  Expenditures on travel, hospitality and confer...
2  2022  Expenditures on travel, hospitality and confer...
3  2016  The Department supports the sector from the fa...
4  2017  The Department supports the sector from the fa...
```

```
                             mandate_description_fr  \
0  Les dépenses de voyage, d'accueil et de confér...
1  Les dépenses de voyage, d'accueil et de confér...
2  Les dépenses de voyage, d'accueil et de confér...
3  Le Ministère soutient le secteur entre l’agric...
4  Le Ministère soutient le secteur entre l’agric...

  operational_activities_kdollars key_stakeholders_kdollars
training_kdollars  \
0                              78                         0
0
1                              15                         0
0
2                               0                         6
0
3                               .                         .
.
4                               .                         .
.

  other_kdollars internal_governance_kdollars
non_public_servants_kdollars  \
0             18                            0
NaN
1              0                            0
NaN
2              0                            0
NaN
3              .                            .
718.0
4              .                            .
1173.0

   public_servants_kdollars  ...  \
0                       NaN  ...
1                       NaN  ...
2                       NaN  ...
3                    7967.0  ...
4                    9366.0  ...

                      travel_compared_fiscal_year_en  \
0                                                NaN
1  Travel expenses were limited due to the COVID-...
2  Travel expenses were limited due to the COVID-...
3  Public Servants: 7967 ;\r\nNon-Public Servants...
4  Public Servants: 9366 The $1.4 million overall...

                      travel_compared_fiscal_year_fr  \
0                                                S/O
```

```
1  Les frais de voyages ont été limités due à la ...
2  Les frais de voyages ont été limités due à la ...
3  Voyages des fonctionnaires: 7967 ;\r\nVoyages ...
4  Voyages des fonctionnaires: 9366 L’augmentatio...

                  hospitality_compared_fiscal_year_en  \
0                                                 NaN
1  Hospitality expenses were limited due to the C...
2                                                 NaN
3                                                 NaN
4  Compared with fiscal year 2015-2016, departmen...

                  hospitality_compared_fiscal_year_fr  \
0                                                 S/O
1  Les frais d'accueil ont été limités due à la p...
2                                                 S/O
3                                                 NaN
4  Comparativement à l’exercice 2015-2016, les dé...

              conference_fees_compared_fiscal_year_en  \
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4  Compared with fiscal year 2015-2016 department...

              conference_fees_compared_fiscal_year_fr  \
0                                                 S/O
1                                                 S/O
2                                                 S/O
3                                                 NaN
4  Comparativement à l’exercice 2015-2016, les dé...

                     minister_compared_fiscal_year_en  \
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4  The Minister and his staff participated in a n...

                     minister_compared_fiscal_year_fr    owner_org  \
0                                                 S/O  casdo-ocena
1                                                 S/O  casdo-ocena
2                                                 S/O  casdo-ocena
3                                                 NaN     aafc-aac
4  Le ministre et son personnel ont participé à d...     aafc-aac

                                      owner_org_title
0  Accessibility Standards Canada | Normes d’acce...
1  Accessibility Standards Canada | Normes d’acce...
```

```
2  Accessibility Standards Canada | Normes d’acce...
3  Agriculture and Agri-Food Canada | Agriculture...
4  Agriculture and Agri-Food Canada | Agriculture...

[5 rows x 23 columns]
```

Data Cleaning

```
df.dropna(inplace=True)
```

Data Analysis and Summarization

```
summary_stats = df.describe()
print(summary_stats)

              year  non_public_servants_kdollars
public_servants_kdollars  \
count    47.000000                     47.000000
47.000000
mean   2015.021277                   2440.872340
39105.617021
std       1.700248                   4681.717395
84804.025316
min    2012.000000                      2.000000
450.000000
25%    2014.000000                    116.000000
1306.000000
50%    2015.000000                   1173.000000
17552.000000
75%    2017.000000                   1782.000000
25455.500000
max    2017.000000                  19866.000000
405772.000000

       hospitality_kdollars  conference_fees_kdollars
minister_kdollars
count                 47.000000                 47.000000
47.000000
mean                 306.212766                276.170213
71.642553
std                  518.112164                437.820507
87.521902
min                    2.000000                  1.000000
0.000000
25%                   29.000000                 19.000000
4.000000
50%                  160.000000                138.000000
37.000000
75%                  291.000000                348.000000
```

```
118.500000
max                2406.000000               2547.000000
350.000000
```

```python
# Check the column names
print(df.columns)
```

```
Index(['year', 'mandate_description_en', 'mandate_description_fr',
       'operational_activities_kdollars', 'key_stakeholders_kdollars',
       'training_kdollars', 'other_kdollars',
'internal_governance_kdollars',
       'non_public_servants_kdollars', 'public_servants_kdollars',
       'hospitality_kdollars', 'conference_fees_kdollars',
'minister_kdollars',
       'travel_compared_fiscal_year_en',
'travel_compared_fiscal_year_fr',
       'hospitality_compared_fiscal_year_en',
       'hospitality_compared_fiscal_year_fr',
       'conference_fees_compared_fiscal_year_en',
       'conference_fees_compared_fiscal_year_fr',
       'minister_compared_fiscal_year_en',
'minister_compared_fiscal_year_fr',
       'owner_org', 'owner_org_title'],
      dtype='object')
```

```python
import pandas as pd

df['operational_activities_kdollars'] =
pd.to_numeric(df['operational_activities_kdollars'], errors='coerce')

grouped_data = df.groupby('mandate_description_en')
['operational_activities_kdollars'].mean()
print(grouped_data)
```

```
mandate_description_en
Created in 1867, the Department of Finance Canada was one of the
original departments of the Government of Canada and had as its
primary functions bookkeeping, administering the collection and
disbursement of public monies, and servicing the national debt. Today,
the Department helps the Government of Canada develop and implement
strong and sustainable economic, fiscal, tax, social, security,
international and financial sector policies and programs. It plays an
important central agency role, working with other departments to
ensure that the government’s agenda is carried out and that ministers
are supported with high-quality analysis and advice.
NaN
ESDC's mission is to build a stronger and more competitive Canada, to
support Canadians in making choices that help them live productive and
rewarding lives and to improve Canadians' quality of life. The
Department delivers a range of programs and services that affect
```

Canadians throughout their lives through three business lines: Employment and Social Development; Labour Program; and Service Canada. Included in its core roles are responsibilities for the design and delivery of some of the Government of Canada's most well-known programs and services, such as: Old Age Security; Canada Pension Plan; Employment Insurance; Canada Student Loans and Grants; the Canada Education Savings Program; National Child Benefit; Wage Earners Protection Program; and Passport Services. The Labour Program is responsible for overseeing federal labour regulatory responsibilities, including facilitating compliance with occupational health and safety, labour standards and employment equity legislation, as well as assisting trade unions and employers in the negotiation of collective agreements and their renewal in federally regulated workplaces. The Labour Program also represents Canada in international labour organizations and negotiates and implements labour provisions in the context of trade liberalization initiatives. Through Service Canada, the Department helps Canadians access departmental programs as well as other Government of Canada programs and services. Finally, through grants and contributions, the Department provides funding to other levels of government and organizations within the voluntary and private sectors, educators and community organizations to support projects that meet the labour market and social development needs of Canadians.

NaN

ESDC’s mission is to build a stronger and more competitive Canada, to support Canadians in making choices that help them live productive and rewarding lives and to improve Canadians' quality of life. The Department delivers a range of programs and services that affect Canadians throughout their lives through three business lines: Employment and Social Development; Labour Program; and Service Canada. Included in its core roles are responsibilities for the design and delivery of some of the Government of Canada's most well-known programs and services, such as: Old Age Security; Canada Pension Plan; Employment Insurance; Canada Student Loans and Grants; the Canada Education Savings Program; Temporary Foreign Workers Program; Wage Earner Protection Program; and Passport Services. The Labour Program is responsible for overseeing federal labour regulatory responsibilities, including facilitating compliance with occupational health and safety, labour standards and employment equity legislation, as well as assisting trade unions and employers in the negotiation of collective agreements and their renewal in federally regulated workplaces. The Labour Program also represents Canada in international labour organizations and negotiates and implements labour provisions in the context of trade liberalization initiatives. Through Service Canada, the Department helps Canadians access departmental programs as well as other Government of Canada programs and services. Finally, through grants and contributions, the Department provides funding to other levels of government and organizations within the voluntary and private sectors, educators and community organizations to support

projects that meet the labour market and social development needs of Canadians.
NaN
Environment Canada is the lead federal department for a wide range of environmental issues affecting Canadians. The Department also plays a stewardship role in achieving and maintaining a clean, safe and sustainable environment. Environment Canada addresses issues through monitoring, research, policy development, service delivery to Canadians, regulations, enforcement of environmental laws, advancement of clean technologies and strategic partnerships. The Department’s programs focus on a clean environment by minimizing threats to Canadians and their environment from pollution; a safe environment by equipping Canadians to make informed decisions on weather, water and climate conditions; and a sustainable environment by conserving and restoring Canada’s natural environment. The Department’s program focus reflects the increasingly evident interdependence between environmental sustainability and economic well-being. Environment Canada fulfills its mandate by promoting three Strategic Outcomes, each contributing to the Government of Canada outcome of a clean and healthy environment; Canada’s natural environment is conserved and restored for present and future generations; Canadians are equipped to make informed decisions on changing weather, water and climate conditions; and Threats to Canadians and their environment from pollution are minimized. EC has authority under numerous pieces of legislation which affect how the department operates. Further information on the mandate, legislation, roles, priorities, responsibilities and strategic outcomes of Environment Canada can be found in Section I of EC's Reports on Plans and Priorities.
NaN
Environment Canada plays an important role every day and has established a legacy of action on behalf of the environment since it was first created on June 11, 1971, from elements of the Government of Canada such as the Meteorological Service of Canada (established in 1871) and the Canadian Wildlife Service (established in 1947). EC’s Program Activity Architecture (PAA) included three Strategic Outcomes that support our responsibility for providing Canadians with a clean, safe and sustainable environment: Threats to Canadians and their environment from pollution are minimized; Canadians are equipped to make informed decisions on changing weather, water and climate conditions; and Canada’s natural environment is conserved and restored for present and future generations. EC has authority under numerous pieces of legislation which affect how the department operates. Further information on the mandate, legislation, roles, responsibilities and program activities of Environment Canada can be found in Section I of EC's Reports on Plans and Priorities.
NaN
Environment Canada plays an important role every day and has established a legacy of action on behalf of the environment since it was first created on June 11, 1971, from elements of the Government of

Canada such as the Meteorological Service of Canada (established in 1871) and the Canadian Wildlife Service (established in 1947). EC’s Program Activity Architecture (PAA) included three Strategic Outcomes that support our responsibility for providing Canadians with a clean, safe and sustainable environment: Threats to Canadians and their environment from pollution are minimized; Canadians are equipped to make informed decisions on changing weather, water and climate conditions; and Canada’s natural environment is conserved and restored for present and future generations. EC has authority under numerous pieces of legislation which affect how the department operates. Further information on the mandate, legislation, roles, responsibilities and program activities of Environment Canada can be found in Section I of EC's Reports on Plans and Priorities.
NaN
Environment Canada’s mandate is to provide a clean, safe and sustainable environment for Canadians. It works in partnership with others to fulfill this mandate through a variety of activities, including conducting research on water and air quality, monitoring Canada’s natural environment, developing regulations to reduce greenhouse gas emissions, maintaining biodiversity, increasing the number of protected areas within Canada, and providing advance warning for severe weather events. Environment Canada fulfills its mandate by promoting three Strategic Outcomes, each contributing to the Government of Canada outcome of a clean and healthy environment: Canada’s natural environment is conserved and restored for present and future generations; Canadians are equipped to make informed decisions on changing weather, water and climate conditions; and Threats to Canadians and their environment from pollution are minimized. EC has authority under numerous pieces of legislation which affect how the department operates. Further information on the mandate, legislation, roles, priorities, responsibilities and strategic outcomes of Environment Canada can be found in Section I of EC's Reports on Plans and Priorities.
NaN
Environment and Climate Change Canada (ECCC) is the lead federal department for a wide range of environmental issues. The Department addresses these issues through various actions including the implementation of the Pan-Canadian Framework on clean growth and climate change; engaging with our strategic partners including provinces, territories and Indigenous peoples; monitoring; science-based research; policy and regulatory development; and, through the enforcement of environmental laws, The Department's programs focus on minimizing threats to Canadians and their environment from pollution; equipping Canadians to make informed decisions on weather, water and climate conditions; and conserving and restoring Canada's natural environment. Under the Department of the Environment Act, the powers, duties and functions of the Minister of Environment and Climate Change extend to matters such as: the preservation and enhancement of the quality of the natural environment, including water, air and soil

quality, and the coordination of the relevant policies and programs of the Government of Canada; renewable resources, including migratory birds and other non-domestic flora and fauna; meteorology; and the enforcement of rules and regulations. ECCC has authority under numerous pieces of legislation which affect how the department operates. Further information on the mandate, legislation, roles, priorities, responsibilities and strategic outcomes of Environment and Climate Change Canada can be found in Section I of ECCC's Reports on Plans and Priorities.
NaN
Environment and Climate Change Canada is the lead federal department for a wide range of environmental issues affecting Canadians. The Department also plays a stewardship role in achieving and maintaining a clean, safe and sustainable environment. Environment and Climate Change Canada addresses issues through monitoring, research, policy development, service delivery to Canadians, regulations, enforcement of environmental laws, advancement of clean technologies and strategic partnerships. The Department’s programs focus on a clean environment by minimizing threats to Canadians and their environment from pollution; a safe environment by equipping Canadians to make informed decisions on weather, water and climate conditions; and a sustainable environment by conserving and restoring Canada’s natural environment. The Department’s program focus reflects the increasingly evident interdependence between environmental sustainability and economic well-being. Environment and Climate Change Canada fulfills its mandate by promoting three Strategic Outcomes, each contributing to the Government of Canada outcome of a clean and healthy environment: Canada’s natural environment is conserved and restored for present and future generations; Canadians are equipped to make informed decisions on changing weather, water and climate conditions; and Threats to Canadians and their environment from pollution are minimized. ECCC has authority under numerous pieces of legislation which affect how the department operates. Further information on the mandate, legislation, roles, priorities, responsibilities and strategic outcomes of Environment and Climate Change Canada can be found in Section I of ECCC's Reports on Plans and Priorities.
NaN
Expenditures on travel, hospitality and conference fees incurred by federal departments and agencies are related to activities that support a departmental or agency mandate and the government's priorities. In particular, for Fisheries and Oceans Canada, this includes playing the lead role in managing Canada's fisheries and safeguarding its waters, ensuring safe, healthy, and productive waters and aquatic ecosystems for the benefit of present and future generations.
NaN
Expenditures on travel, hospitality and conference fees incurred by federal departments and agencies are related to activities that support a departmental or agency mandate and the government’s

priorities. In particular, for Fisheries and Oceans Canada, this includes playing the lead role in managing Canada’s fisheries and safeguarding its waters, ensuring safe, healthy, and productive waters and aquatic ecosystems for the benefit of present and future generations.
NaN
Expenditures on travel, hospitality and conference fees incurred by federal departments and agencies are related to activities that support a departmental or agency mandate and the government’s priorities. In particular, for Fisheries and Oceans Canada, this includes playing the lead role in managing Canada’s fisheries and safeguarding its waters, ensuring safe, healthy, and productive waters and aquatic ecosystems for the benefit of present and future generations.
NaN
Expenditures on travel, hospitality and conferences incurred by federal departments and agencies are for the most part directly related to supporting departmental mandates and the government's priorities. The mandate of the TSB is to advance transportation safety. This mandate is fulfilled by conducting independent investigations into selected transportation occurrences to identify the causes and contributing factors of the occurrences and the underlying safety deficiencies, reporting on its findings, making recommendations and advocating to influence safety actions and changes.
NaN
In particular, for WD, this includes the delivery of core programs to Canadians that advance innovation, business development and community economic development in rural and urban areas throughout the four western provinces. Through the Western Economic Diversification Act, the department is mandated to “promote the development and diversification of the economy of Western Canada and to advance the interests of the West in national economic policy, program and project development and implementation.”
NaN
In particular, for WD, this includes the delivery of strategic programs to Canadians that advance innovation, business development and community economic development in rural and urban areas throughout the four western provinces. Through the Western Economic Diversification Act, the department is mandated to "promote the development and diversification of the economy of Western Canada and to advance the interests of the West in national economic policy, program and project development and implementation."
NaN
NRCan works to improve the quality of life of Canadians by ensuring that our natural resources are developed sustainably, providing a source of jobs, prosperity and opportunity, while preserving our environment and respecting our communities and Indigenous People. To fulfil its responsibilities, NRCan relies on a number of instruments

(e.g. policy, regulation, statutory transfers, grants and contributions) and key activities (e.g. science and technology, partnerships and communications), while working in offices and laboratories located across the country, in the National Capital Region, Atlantic Canada, Quebec, Ontario, Western and Pacific Regions and Northern Canada. In 2016-17, NRCan spent $11.6 million on travel, hospitality and conference fees. Consistent with last year's spending, the expenditures in travel (19%) were in support of the Landmass Information program, which provides open access to Canada's fundamental geomatics framework and information system, including accurate three-dimensional positioning, high-resolution satellite imagery and other remote sensing products, legal (boundary) surveys, mapping and other analysis applications. In addition, it delivers logistics support in the North and regulatory oversight for a robust property system framework on Canada Lands. The largest contributor to expenditures in hospitality (28%) was in the Market Access and Diversification program mainly in support of the new Indigenous and Advisory Monitoring Committees co-development process. This program supports Canada's natural resource sectors that face two key barriers to market access and diversification: 1) trade and policy barriers, and 2) lack of awareness of Canada's natural resource products. The objectives of this Program are to break down those barriers and support the development and expansion of markets for Canadian natural resource products by making information available to Canadians, supporting negotiations to reduce trade barriers, and ensuring that regulations are up to date. This helps maintain natural resource sectors' access to existing markets and increases their access to new market segments. Consistent with last year's spending, the largest contributor to expenditures in conference fees (22%) was in support of the Technology Innovation program, which encourages academia, industry and the public sector to research, develop and demonstrate innovative solutions. This objective is achieved through the generation and dissemination of scientific knowledge, and the development and demonstration of new technologies. For more information on NRCan's Plans and Priorities, see the 2016-2017 – Report on Plans and Priorities (RPP).   NaN
The Canadian Air Transport Security Authority is a Crown corporation responsible for securing specific elements of the air transportation system – from passenger and baggage screening to screening airport workers.CATSA is mandated with protecting the public through effective and efficient screening of air travellers and their baggage. Our goal is to provide a professional, effective and consistent level of security service across the country, at or above the standards set by Transport Canada.
NaN
The Canadian Armed Forces (CAF) and the Department of National Defence (DND) support Canada's vision to be: strong at home, with a military ready and able to defend its sovereignty, and to assist in times of natural disaster, support search and rescue, or respond to other

emergencies; secure in North America, active in a renewed defence partnership in NORAD and with the United States; and engaged in the world, with Defence doing its part in Canadian contributions to a more stable and peaceful world.\r\n\r\nThe National Defence Act (NDA) establishes DND and the CAF as separate entities, operating within an integrated National Defence Headquarters, as they pursue their primary responsibility of providing defence for Canada and Canadians.\r\n\r\nThe Defence mandate is carried out with the support of a group of related organizations and agencies with the portfolio of the Minister of National Defence. For further details on Defence portfolio organizations, please refer to National Defence and Canadian Armed Forces website. For further information on the legislative framework within which Defence operates, please see the DND Departmental Plan.
NaN
The Department supports the sector from the farmer to the consumer, from the farm to global markets, through all phases of producing, processing and marketing of farm, food and agri-based products. Agriculture is a shared jurisdiction in Canada, and the Department works closely with provincial and territorial governments in the development and delivery of policies, programs and services.
NaN
The National Defence Act establishes DND and the Canadian Armed Forces as separate entities, operating with an integrated National Defence Headquarters, as they pursue their primary responsibility of providing defence for Canada and Canadians. On behalf of the people of Canada, Defence stands ready to perform three key roles: defend Canada - by delivering excellence at home; defend North America - by being a strong and reliable partner with the United States in the defence of the continent; and contribute to International Peace and Security - by projecting leadership abroad.\r\n\r\nThe Defence mandate is carried out with the support of a group of related organizations and agencies within the portfolio of the Minister of National Defence. For further details on selected Defence Portfolio organizations, please refer to National Defence and Canadian Armed Forces website. For further information on the legislative framework within which Defence operates, please see the DND Report on Plans and Priorities.
NaN
The National Defence Act establishes DND and the Canadian Armed Forces as separate entities, operating with an integrated National Defence Headquarters, as they pursue their primary responsibility of providing defence for Canada and Canadians. On behalf of the people of Canada, Defence stands ready to perform three key roles: defend Canada - by delivering excellence at home; defend North America - by being a strong and reliable partner with the United States in the defence of the continent; and contribute to international peace and security - by projecting leadership abroad.\r\n\r\nThe Defence mandate is carried out with the support of a group of related organizations and agencies within the portfolio of the Minister of National Defence. For further details on selected Defence Portfolio organizations, please refer to

National Defence and Canadian Armed Forces website. For further information on the legislative framework within which Defence operates, please see the DND Report on Plans and Priorities.
NaN
The National Defence Act establishes DND and the Canadian Forces as separate entities, operating with an integrated National Defence Headquarters, as they pursue their primary responsibility of providing defence for Canada and Canadians. On behalf of the people of Canada, Defence stands ready to perform three key roles: defend Canada - by delivering excellence at home; defend North America - by being a strong and reliable partner with the United States in the defence of the continent; and contribute to International Peace and Security - by projecting leadership abroad.\r\n\r\nThe Defence mandate is carried out with the support of a group of related organizations and agencies within the portfolio of the Minister of National Defence. For further details on selected Defence Portfolio organizations, please refer to the DND Report on Plans and Priorities Section IV: Other Items of Interest – Selected Defence Portfolio HR and Financial Resources. For further information on the legislative framework within which Defence operates, please see the DND Report on Plans and Priorities Section IV: Other Items of Interest – Legislative Environment.
NaN
The PBC is an independent administrative tribunal that has exclusive authority under the Corrections and Conditional Release Act to grant, deny, cancel, terminate or revoke day parole and full parole, and authorize or approve temporary absences. The PBC may also order certain offenders to be held in prison until the end of their sentence. In addition, the PBC makes conditional release decisions for offenders in provinces and territories that do not have their own parole boards.   The PBC is also responsible for making decisions to order, refuse to order and revoke record suspensions under the Criminal Records Act and the Criminal Code of Canada. The PBC also makes recommendations for the exercise of clemency through the Royal Prerogative of Mercy.
NaN
The department delivers its mandate through three business lines: programs that support Employment and Social Development, the Labour Program, and Service Canada. The core programs and services for Canadians that ESDC delivers include Old Age Security; Canada Pension Plan; Employment Insurance; Canada Student Loans and Grants; the Canada Education Savings Program; National Child Benefit; and Universal Child Care Benefit. ESDC also provides funding to organizations and other levels of government through targeted labour market and social development programs. Through the Labour Program, ESDC is responsible for labour laws and policies in federally regulated workplaces. Service Canada helps citizens access ESDC’s programs, as well as other Government of Canada programs and services
NaN
The department delivers its mandate through three business lines:

programs that support Human Resources and Skills Development, the Labour Program, and Service Canada. The core programs and services for Canadians that HRSDC delivers include Old Age Security; Canada Pension Plan; Employment Insurance; Canada Student Loans and Grants; National Child Benefit; and Universal Child Care Benefit. HRSDC also provides funding to organizations and other levels of government through targeted labour market and social development programs. Through the Labour Program, HRSDC is responsible for labour laws and policies in federally regulated workplaces. Service Canada helps citizens access HRSDC’s programs, as well as other Government of Canada programs and services.
NaN
The vision of NRCan is to improve the quality of life of Canadians by creating a sustainable resource advantage – now and for the future. It seeks to achieve this vision by working to improve the competitiveness of the natural resource sectors and to grow their contribution to Canada’s economy. NRCan supports the responsible development of Canada’s resources in a manner that advances the country’s global standing as a leader on the environment, and uses its knowledge and expertise of Canada’s landmass to support the safety and security of citizens. In 2015-16, NRCan spent $11.6 million on travel, hospitality and conference fees. The largest contributor to expenditures in travel (18%) were in support of the program ‘Protection for Canadians and Natural Resources’, which supports government departments, communities and the private sector to manage risks to human, natural resource and infrastructure health. This is done by providing regulation, knowledge, tools and services, while fulfilling legislated responsibilities. The largest contributor to expenditures in hospitality (22%) were in support of the program ‘Market Access and Diversification’, which supports natural resource sectors that face two key barriers to market access and diversification: 1) trade and policy barriers and 2) lack of awareness of Canada’s natural resource products. The objectives of this Program are to break down those barriers and support the development and expansion of markets for Canadian natural resource products by making information available to Canadians, supporting negotiations to reduce trade barriers, and ensuring that regulations are up to date. This helps maintain natural resource sectors’ access to existing markets and increases their access to new market segments. The largest contributor to expenditures in conference fees (25%) were in support of the program ‘Technology Innovation’, which encourages academia, industry and the public sector to research, develop and demonstrate innovative solutions to environmental challenges confronted by the natural resource sector. This is done through the generation and dissemination of scientific knowledge, and the development and demonstration of new technologies. For more information on NRCan’s Plans and Priorities, see the 2015-16 Report on Plans and Priorities.
NaN
The vision of NRCan is to improve the quality of life of Canadians by

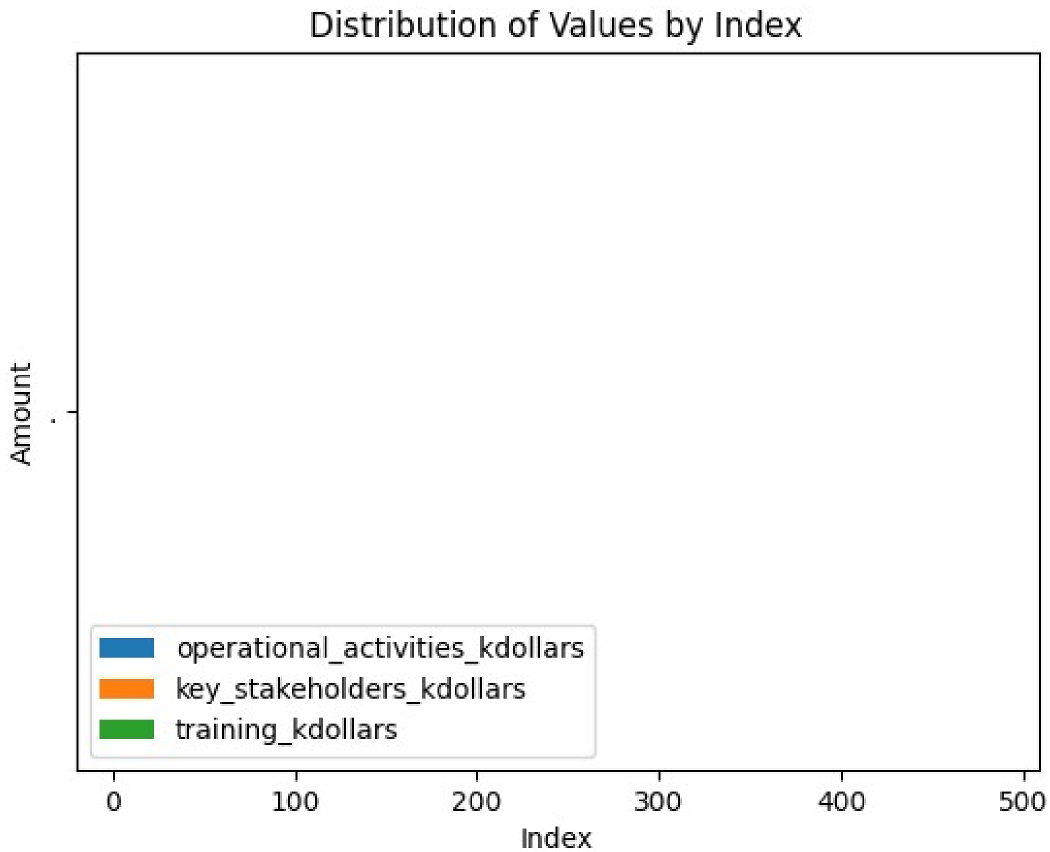
creating a sustainable resource advantage. It seeks to achieve this vision by working to improve the competitiveness of the natural resource sectors and to grow their contribution to Canada's economy. NRCan supports the responsible development of Canada's resources in a manner that advances the country's global standing as a leader on the environment, and uses its knowledge and expertise of Canada's landmass to support the safety and security of citizens. In 2014-15, NRCan spent approximately $12.5 million on travel, hospitality and conference fees. The largest contributor to expenditures in travel (17%) were in support of the program activity 'Protection for Canadians and Natural Resources', which supports government departments, communities and the private sector to manage risks to human, natural resource and infrastructure health. This is done by providing regulation, knowledge, tools and services, fulfilling legislated responsibilities, and ensuring capacity. The largest contributors to expenditures in hospitality (29%) and conference fees (20%) were in support of the program activity 'Technology Innovation', which encourages academia, industry and the public sector to research, develop and demonstrate innovative solutions to environmental challenges encountered in the natural resource sector. This is done through the generation and dissemination of scientific knowledge, and the development and demonstration of new technologies. For more information on NRCan's Plans and Priorities, see the 2014-15 Report on Plans and Priorities.
NaN
The vision of Natural Resources Canada (NRCan) is to improve the quality of life of Canadians by creating a sustainable resource advantage. It seeks to achieve this vision by working to improve the competitiveness of the natural resource sectors and to grow their contribution to Canada's economy, by supporting the sustainable development of Canada's resources in a manner that advances the country's global standing as a leader on the environment, and by using its knowledge and expertise of Canada's landmass to support the safety and security of citizens. In support and development of its programs, NRCan incurs travel, hospitality and conference expenses. In 2013-14, NRCan spent approximately $11.8 million on travel, hospitality and conference fees. The largest contributor to expenditures in travel (17%) were in support of the program activity 'Protection for Canadians and Natural Resources', which supports government departments, communities and the private sector to manage risks to human, natural resource and infrastructure health. This is done by providing regulation and knowledge, fulfilling legislated responsibilities, and ensuring capacity. The largest contributor to expenditures in hospitality (24%) were in support of the program activity 'Market Access Diversification', which supports the breakdown of the two barriers: 1) trade and policy barriers and 2) lack of awareness of Canada's natural resource products by making information available to Canadians, supporting negotiations to reduce trade barriers, and ensuring that regulations are up to date. This helps maintain natural resource sectors' access to existing markets and

increases their access to new market segments. The largest contributor to expenditures in conference fees (37%) were in support of the program activity ‘Technology Innovation’, which supports academia, industry and the public sector to research, develop and demonstrate innovative solutions to environmental challenges encountered in the natural resource sector. For more information on NRCans Plans and Priorities, see the 2013-14 Report on Plans and Priorities.
NaN
Transport Canada is committed to ensuring Canada has a transportation system that is recognized worldwide as safe and secure, efficient and environmentally responsible.
NaN
Transport Canada provides a variety of programs and services to Canadian citizens, related to safety, security, the environment, and innovation in transportation by air, land and sea. The department is committed to ensuring Canada has aviation, rail, road and marine transportation systems that are safe, secure, environmentally responsible and innovative.
NaN
Transport Canada provides a variety of programs and services to Canadian citizens, related to safety, security, the environment, and innovation in transportation by air, land and sea. The department is committed to ensuring Canada has aviation, rail, road and marine transportation systems that are safe, secure, environmentally responsible and innovative.
NaN
Transport Canada provides a variety of programs and services to Canadian citizens. These programs and services relate to areas such as safety, security, the environment and innovation in transportation by air, land and sea. The department is committed to ensuring Canada has aviation, rail, road and marine transportation systems that are safe, secure, environmentally responsible and innovative.
NaN
Transport Canada provides a variety of programs and services to Canadian citizens. These programs and services relate to areas such as safety, security, the environment and innovation in transportation by air, land and sea. The department is committed to ensuring Canada has aviation, rail, road and marine transportation systems that are safe, secure, environmentally responsible and innovative.
NaN
Under the Economic Development Agency of Canada for the Regions of Quebec Act, which came into effect on October 5, 2005, the Agency’s mission is to promote the long-term economic development of the regions of Quebec by giving special attention to those where slow economic growth is prevalent or where opportunities for productive employment are inadequate. \r\nInformation on the Agency’s authorities, mandate, program activities and programs can be found in the Report on Plans and Priorities and the Departmental Performance Report.

```
NaN
Veterans Affairs Canada exists to repay the nation's debt of gratitude
toward those whose courageous efforts have given us this legacy, and
have contributed to our growth as a nation.  VAC's mandate stems from
laws and regulations. Among the more significant is the Department of
Veterans Affairs Act, which charges the Minister of Veterans Affairs
with the following responsibilities: "...the care, treatment, or re-
establishment in civil life of any person who served in the Canadian
Forces or merchant navy or in the naval, army or air forces or
merchant navies of Her Majesty, of any person who has otherwise
engaged in pursuits relating to war, and of any other person
designated ... and the care of the dependants or survivors of any
person referred to ..." \tFurther detailed information on VAC’s
mandate, priorities, and program activities can be found in the
Departmental Results Report and the Departmental Plan at:
http://www.veterans.gc.ca/eng/about-us/reports.
NaN
Name: operational_activities_kdollars, dtype: float64
```

Data Visualization

```
import pandas as pd
import matplotlib.pyplot as plt

columns_to_plot = ['operational_activities_kdollars',
'key_stakeholders_kdollars', 'training_kdollars']

for column in columns_to_plot:
    plt.bar(df.index, df[column], label=column)

plt.title('Distribution of Values by Index')
plt.xlabel('Index')
plt.ylabel('Amount')
plt.legend()
plt.show()
```



Data preparation, including selecting, preprocessing, and transforming data

```
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Step 1: Load the dataset
df = pd.read_csv('travela  (2).csv')

# Step 2: Select relevant columns (features) and target variable
selected_columns = [
    'year', 'operational_activities_kdollars',
'key_stakeholders_kdollars',
    'training_kdollars', 'other_kdollars',
'internal_governance_kdollars',
    'non_public_servants_kdollars', 'public_servants_kdollars',
    'hospitality_kdollars', 'conference_fees_kdollars',
'minister_kdollars',
    'travel_compared_fiscal_year_en',
'travel_compared_fiscal_year_fr',
    'hospitality_compared_fiscal_year_en',
'hospitality_compared_fiscal_year_fr',
    'conference_fees_compared_fiscal_year_en',
'conference_fees_compared_fiscal_year_fr',
```

```
    'minister_compared_fiscal_year_en',
'minister_compared_fiscal_year_fr',
    'owner_org', 'owner_org_title'
]

selected_features = df[selected_columns]

# Step 3: Data Preprocessing
# Replace non-numeric values with NaN
selected_features = selected_features.apply(pd.to_numeric,
errors='coerce')

# Display the intermediate result
print("Selected Features (After Conversion to Numeric):")
print(selected_features)

# Fill NaN with the mean
selected_features = selected_features.fillna(selected_features.mean())

# Display the intermediate result
print("\nSelected Features (After Filling NaN with Mean):")
print(selected_features)

# Step 4: Transform the Data
scaler = StandardScaler()
scaled_features =
scaler.fit_transform(selected_features[['operational_activities_kdolla
rs', 'key_stakeholders_kdollars']])

# Display the scaled features
print("\nScaled Features:")
print(scaled_features)

# Now, 'scaled_features' contains the scaled values of the selected
numerical features.

# Continue with your analysis using 'selected_features' and
'scaled_features'.
```

```
Selected Features (After Conversion to Numeric):
     year  operational_activities_kdollars  key_stakeholders_kdollars
\
0    2020                             78.0                        0.0

1    2021                             15.0                        0.0

2    2022                              0.0                        6.0

3    2016                              NaN                        NaN

4    2017                              NaN                        NaN
```

```
..    ...                                ...                         ...
484  2012                                NaN                         NaN
485  2011                                NaN                         NaN
486  2018                              363.0                       310.0
487  2020                              314.0                       429.0
488  2019                              330.0                       445.0

     training_kdollars  other_kdollars
internal_governance_kdollars  \
0                  0.0            18.0                             0.0
1                  0.0             0.0                             0.0
2                  0.0             0.0                             0.0
3                  NaN             NaN                             NaN
4                  NaN             NaN                             NaN
..                 ...             ...                             ...
484                NaN             NaN                             NaN
485                NaN             NaN                             NaN
486               92.0            23.0                           155.0
487              204.0            10.0                           201.0
488              135.0            35.0                           149.0

     non_public_servants_kdollars  public_servants_kdollars  \
0                             NaN                       NaN
1                             NaN                       NaN
2                             NaN                       NaN
3                           718.0                    7967.0
4                          1173.0                    9366.0
..                            ...                       ...
484                          77.0                    1776.0
485                          10.0                    2405.0
486                           NaN                       NaN
487                           NaN                       NaN
488                           NaN                       NaN
```

```
     hospitality_kdollars  conference_fees_kdollars  ...  \
0                     6.0                       0.0  ...
1                     0.0                       0.0  ...
2                     0.0                       0.0  ...
3                   226.0                     148.0  ...
4                   367.0                     202.0  ...
..                    ...                       ...  ...
484                  39.0                      20.0  ...
485                  75.0                      28.0  ...
486                  20.0                      43.0  ...
487                  36.0                      34.0  ...
488                  34.0                      58.0  ...

     travel_compared_fiscal_year_en  travel_compared_fiscal_year_fr  \
0                               NaN                             NaN
1                               NaN                             NaN
2                               NaN                             NaN
3                               NaN                             NaN
4                               NaN                             NaN
..                              ...                             ...
484                             NaN                             NaN
485                             NaN                             NaN
486                             NaN                             NaN
487                             NaN                             NaN
488                             NaN                             NaN

     hospitality_compared_fiscal_year_en
hospitality_compared_fiscal_year_fr  \
0                                    NaN
NaN
1                                    NaN
NaN
2                                    NaN
NaN
3                                    NaN
NaN
4                                    NaN
NaN
..                                   ...
...
484                                  NaN
NaN
485                                  NaN
NaN
486                                  NaN
NaN
487                                  NaN
NaN
488                                  NaN
```

```
NaN

     conference_fees_compared_fiscal_year_en  \
0                                         NaN
1                                         NaN
2                                         NaN
3                                         NaN
4                                         NaN
..                                        ...
484                                       NaN
485                                       NaN
486                                       NaN
487                                       NaN
488                                       NaN

     conference_fees_compared_fiscal_year_fr  \
0                                         NaN
1                                         NaN
2                                         NaN
3                                         NaN
4                                         NaN
..                                        ...
484                                       NaN
485                                       NaN
486                                       NaN
487                                       NaN
488                                       NaN

     minister_compared_fiscal_year_en
minister_compared_fiscal_year_fr  \
0                                  NaN
NaN
1                                  NaN
NaN
2                                  NaN
NaN
3                                  NaN
NaN
4                                  NaN
NaN
..                                 ...                                ..
.
484                                NaN
NaN
485                                NaN
NaN
486                                NaN
NaN
487                                NaN
NaN
```

```
488                                NaN
NaN

     owner_org  owner_org_title
0          NaN              NaN
1          NaN              NaN
2          NaN              NaN
3          NaN              NaN
4          NaN              NaN
..         ...              ...
484        NaN              NaN
485        NaN              NaN
486        NaN              NaN
487        NaN              NaN
488        NaN              NaN

[489 rows x 21 columns]

Selected Features (After Filling NaN with Mean):
     year  operational_activities_kdollars  key_stakeholders_kdollars
\
0    2020                        78.000000                   0.000000

1    2021                        15.000000                   0.000000

2    2022                         0.000000                   6.000000

3    2016                      7364.408052                 808.420161

4    2017                      7364.408052                 808.420161

..    ...                              ...                        ...

484  2012                      7364.408052                 808.420161

485  2011                      7364.408052                 808.420161

486  2018                       363.000000                 310.000000

487  2020                       314.000000                 429.000000

488  2019                       330.000000                 445.000000


     training_kdollars  other_kdollars
internal_governance_kdollars  \
0             0.000000       18.000000                      0.000000

1             0.000000        0.000000                      0.000000

2             0.000000        0.000000                      0.000000
```

```
3          2609.705672      726.788243                          1127.372617
4          2609.705672      726.788243                          1127.372617
..                 ...             ...                                  ...
484        2609.705672      726.788243                          1127.372617
485        2609.705672      726.788243                          1127.372617
486          92.000000       23.000000                           155.000000
487         204.000000       10.000000                           201.000000
488         135.000000       35.000000                           149.000000

     non_public_servants_kdollars  public_servants_kdollars  \
0                     2272.096481               29128.11663
1                     2272.096481               29128.11663
2                     2272.096481               29128.11663
3                      718.000000                7967.00000
4                     1173.000000                9366.00000
..                            ...                       ...
484                     77.000000                1776.00000
485                     10.000000                2405.00000
486                   2272.096481               29128.11663
487                   2272.096481               29128.11663
488                   2272.096481               29128.11663

     hospitality_kdollars  conference_fees_kdollars  ...  \
0                     6.0                       0.0  ...
1                     0.0                       0.0  ...
2                     0.0                       0.0  ...
3                   226.0                     148.0  ...
4                   367.0                     202.0  ...
..                    ...                       ...  ...
484                  39.0                      20.0  ...
485                  75.0                      28.0  ...
486                  20.0                      43.0  ...
487                  36.0                      34.0  ...
488                  34.0                      58.0  ...

     travel_compared_fiscal_year_en  travel_compared_fiscal_year_fr  \
0                               NaN                             NaN
1                               NaN                             NaN
2                               NaN                             NaN
3                               NaN                             NaN
4                               NaN                             NaN
```

```
..                                ...                               ...
484                               NaN                               NaN
485                               NaN                               NaN
486                               NaN                               NaN
487                               NaN                               NaN
488                               NaN                               NaN

     hospitality_compared_fiscal_year_en
hospitality_compared_fiscal_year_fr  \
0                                    NaN
NaN
1                                    NaN
NaN
2                                    NaN
NaN
3                                    NaN
NaN
4                                    NaN
NaN
..                                   ...
...
484                                  NaN
NaN
485                                  NaN
NaN
486                                  NaN
NaN
487                                  NaN
NaN
488                                  NaN
NaN

     conference_fees_compared_fiscal_year_en  \
0                                        NaN
1                                        NaN
2                                        NaN
3                                        NaN
4                                        NaN
..                                       ...
484                                      NaN
485                                      NaN
486                                      NaN
487                                      NaN
488                                      NaN

     conference_fees_compared_fiscal_year_fr  \
0                                        NaN
1                                        NaN
2                                        NaN
3                                        NaN
```

```
4                                                           NaN
..                                                          ...
484                                                         NaN
485                                                         NaN
486                                                         NaN
487                                                         NaN
488                                                         NaN

     minister_compared_fiscal_year_en
minister_compared_fiscal_year_fr  \
0                                 0.0
0.0
1                                 0.0
0.0
2                                 0.0
0.0
3                                 0.0
0.0
4                                 0.0
0.0
..                                ...                                ..
.
484                               0.0
0.0
485                               0.0
0.0
486                               0.0
0.0
487                               0.0
0.0
488                               0.0
0.0

     owner_org  owner_org_title
0          NaN              NaN
1          NaN              NaN
2          NaN              NaN
3          NaN              NaN
4          NaN              NaN
..         ...              ...
484        NaN              NaN
485        NaN              NaN
486        NaN              NaN
487        NaN              NaN
488        NaN              NaN

[489 rows x 21 columns]

Scaled Features:
[[-4.56204182e-01 -3.29051680e-01]
```

```
 [-4.60148631e-01 -3.29051680e-01]
 [-4.61087785e-01 -3.26609497e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 1.14926960e-01 -1.39375457e-01]
 [-4.35397056e-02  2.11077823e-01]
 [-1.11784931e-01  2.68876157e-01]
 [-4.45623042e-01 -3.26202466e-01]
 [-4.17886679e-01 -3.06257971e-01]
 [-1.87105120e-01  2.50648740e-02]
 [ 1.03673838e+00 -2.33717581e-02]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 1.34021050e+00  2.09856731e-01]
 [ 1.41490458e+00  2.02937212e-01]
 [ 2.26686342e-01 -2.89162689e-01]
 [ 8.26242559e-01 -2.17118287e-01]
 [ 1.71693666e+00  1.41882634e-01]
 [-4.33914916e-01 -2.59856491e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.32787931e-01 -3.20504039e-01]
 [-4.25775577e-01 -3.23353253e-01]
 [-4.28279989e-01 -3.24167314e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.60649513e-01 -3.29051680e-01]
 [-4.57957270e-01 -3.29051680e-01]
 [-4.46624806e-01 -3.29051680e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-3.81071824e-01 -1.54028556e-01]
 [-3.70741125e-01 -1.65018380e-01]
 [-3.84953663e-01 -1.03963801e-01]
 [-4.46061314e-01 -3.26609497e-01]
 [-4.37358482e-01 -3.29865741e-01]
 [-3.94720869e-01 -2.35841691e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 8.13845720e-01  2.37127776e-01]
 [ 1.33645388e+00  2.22881708e-01]
 [ 1.56473103e+00  1.56535733e-01]
 [-2.43516999e-01 -3.22946222e-01]
 [-4.25838187e-01 -3.27423558e-01]
```

```
[-1.49664161e-01 -2.36655752e-01]
[ 0.00000000e+00  4.62740129e-17]
[ 0.00000000e+00  4.62740129e-17]
[-3.52646749e-01 -3.03815788e-01]
[-3.38684652e-01 -2.93232994e-01]
[-3.26287813e-01 -2.83464262e-01]
[-4.02484546e-01 -3.23353253e-01]
[-4.30471349e-01 -3.27830588e-01]
[-3.66358404e-01 -3.06257971e-01]
[-4.57412561e-01 -3.14276472e-01]
[ 0.00000000e+00  4.62740129e-17]
[-4.56867851e-01 -3.08781560e-01]
[-4.56592366e-01 -3.06176565e-01]
[-4.61087785e-01 -3.28951958e-01]
[-4.61087785e-01 -3.28970274e-01]
[-4.59816170e-01 -3.26841504e-01]
[-3.69801970e-01 -2.22816714e-01]
[-3.84327560e-01 -2.65961949e-01]
[ 0.00000000e+00  4.62740129e-17]
[-2.41763910e-01 -2.08570646e-01]
[-3.03810716e-01 -7.79138480e-02]
[-3.25035607e-01 -1.08034107e-01]
[-4.51884072e-01 -3.23353253e-01]
[-3.33738439e-01 -2.05721432e-01]
[-4.06679436e-01 -3.21318100e-01]
[-4.40676828e-01 -2.97710330e-01]
[-4.43744733e-01 -2.72474437e-01]
[-4.60837344e-01 -3.29051680e-01]
[-4.60899954e-01 -3.25795436e-01]
[-4.56531008e-01 -2.86936232e-01]
[-2.20852071e-01 -2.87941597e-01]
[ 0.00000000e+00  4.62740129e-17]
[ 0.00000000e+00  4.62740129e-17]
[ 0.00000000e+00  4.62740129e-17]
[ 0.00000000e+00  4.62740129e-17]
[ 0.00000000e+00  4.62740129e-17]
[ 0.00000000e+00  4.62740129e-17]
[ 0.00000000e+00  4.62740129e-17]
[-2.21290343e-01 -2.70846315e-01]
[-2.08580452e-01 -2.84685353e-01]
[-4.60962565e-01 -3.29051680e-01]
[-4.60962565e-01 -3.29051680e-01]
[-4.53011057e-01 -2.99338452e-01]
[-4.29157973e-01 -3.28925501e-01]
[-4.38153696e-01 -3.29051680e-01]
[-4.61072508e-01 -3.29051680e-01]
[-4.59136222e-01 -3.29051680e-01]
[-4.33631917e-01 -3.29051680e-01]
[ 0.00000000e+00  4.62740129e-17]
```

```
 [ 0.00000000e+00  4.62740129e-17]
 [-4.47877012e-01 -2.72474437e-01]
 [-4.43056019e-01 -2.72881468e-01]
 [-4.26777342e-01 -2.81429109e-01]
 [-4.57769440e-01 -3.17654825e-01]
 [-4.34603629e-01 -2.88348628e-01]
 [-4.26464290e-01 -2.81429109e-01]
 [-4.59898190e-01 -3.29051680e-01]
 [-4.58207712e-01 -3.29051680e-01]
 [-4.41553372e-01 -3.17247795e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [-2.46522293e-01 -2.79393956e-01]
 [-2.39259498e-01 -2.40726057e-01]
 [-2.74133434e-01 -2.36248721e-01]
 [-4.56517234e-01 -3.30679802e-01]
 [-4.17197966e-01 -3.21725131e-01]
 [-3.14579687e-01 -1.85776936e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.50318814e-01 -2.99338452e-01]
 [-4.48878777e-01 -2.93640025e-01]
 [-4.47376130e-01 -2.50494789e-01]
 [-4.60712124e-01 -3.29051680e-01]
 [-4.60336462e-01 -3.27423558e-01]
 [ 1.51053102e-01  4.94867053e-02]
 [ 1.74156302e-01 -8.76825806e-02]
 [ 1.91687185e-01  4.86726443e-02]
 [-8.43616201e-02 -3.07479062e-01]
 [-7.53457372e-02 -2.92418933e-01]
 [ 3.42848952e-02 -9.70442826e-02]
 [-4.20391091e-01 -2.68404132e-01]
 [-1.84851149e-01  2.71318340e-01]
 [-4.01921053e-01 -2.98524391e-01]
 [-2.41513469e-01  1.04435826e-01]
 [-4.25024254e-01 -2.32992477e-01]
 [-4.29031313e-01 -2.34213568e-01]
 [-4.61025175e-01 -3.28644650e-01]
 [-4.60211241e-01 -3.27423558e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-3.22155533e-01 -2.49680728e-01]
 [-2.99741046e-01 -1.52807464e-01]
 [-3.53398072e-01 -1.63390258e-01]
 [-4.60211241e-01 -3.29051680e-01]
 [-4.42930799e-01 -3.05850940e-01]
 [-3.50893660e-01 -1.93510516e-01]
 [-2.02569863e-01 -2.95268147e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [-1.47034529e-01 -2.65961949e-01]
 [-1.41149161e-01 -2.59856491e-01]
```

```
 [-4.33914916e-01 -3.26609497e-01]
 [-4.26213849e-01 -3.26609497e-01]
 [-2.48275381e-01 -2.00430035e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 3.79580691e-01  5.31003813e-01]
 [ 3.48463372e-01  7.14574578e-01]
 [ 5.83752874e-01  5.37923332e-01]
 [-3.89398994e-01 -3.12770459e-01]
 [-3.34865424e-01 -2.97303299e-01]
 [ 8.36844206e-02 -8.84966416e-02]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 7.77907409e-01  3.97090771e-01]
 [ 9.09952528e-01  1.21359400e+00]
 [ 7.51485863e-01  9.33149968e-01]
 [-2.51280676e-01 -3.24574344e-01]
 [-3.44612124e-02 -2.48866667e-01]
 [ 7.82540571e-01  5.27747569e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.54263263e-01 -3.21725131e-01]
 [-4.53142539e-01 -3.13503114e-01]
 [ 5.57581769e-01  2.15383450e+00]
 [ 8.89044899e+00  5.33355694e+00]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.35918445e-01 -3.19689978e-01]
 [-4.38861129e-01 -2.99338452e-01]
 [-4.36356718e-01 -2.89162689e-01]
 [-4.61025175e-01 -3.29051680e-01]
 [-4.60774734e-01 -3.28237619e-01]
 [-4.49379660e-01 -3.02594696e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.31723555e-01 -3.17654825e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.47313520e-01 -2.92011903e-01]
 [-4.53449329e-01 -3.08700154e-01]
```

```
 [-4.61025175e-01 -3.29051680e-01]
 [-4.60899954e-01 -3.29051680e-01]
 [-4.58458153e-01 -3.02594696e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 2.76828879e+00  6.20550528e-01]
 [ 2.71456916e+00  1.75942193e+00]
 [ 2.92143358e+00  1.83716476e+00]
 [ 8.46027413e-01 -1.91882394e-01]
 [ 1.65301155e+00 -2.13366055e-02]
 [ 2.98085076e+00  1.26569390e+00]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 3.15333587e+00  7.37408990e-01]
 [ 3.78073488e+00  1.92842100e+00]
 [ 3.20460118e+00  2.35462266e+00]
 [-2.83894380e-01 -2.35332902e-01]
 [ 6.41980451e-01  2.33057471e-01]
 [ 3.20518346e+00  1.83635070e+00]
 [ 9.58561113e+00  1.38626390e-01]
 [ 4.12346384e-02  4.23140724e-01]
 [ 3.93563294e-02  3.77146275e-01]
 [-4.42304696e-01 -3.22132161e-01]
 [-3.90588589e-01 -3.09921246e-01]
 [-1.72742317e-01 -4.79156986e-02]
 [-4.11187378e-01 -2.74916621e-01]
 [-4.04926348e-01 -2.80208017e-01]
 [-4.55578079e-01 -3.28644650e-01]
 [-4.60148631e-01 -3.29051680e-01]
 [-4.35981056e-01 -3.18468886e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.20641533e-01 -2.64333827e-01]
 [-4.33727085e-01 -8.56474279e-02]
 [-4.40363777e-01 -1.19837992e-01]
 [-4.60336462e-01 -3.24981375e-01]
 [-4.58896425e-01 -2.98931421e-01]
 [-4.31034842e-01 -1.89847242e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [-8.15441567e-02  3.86915008e-01]
 [-9.45044885e-02  9.89320180e-01]
```

```
 [-9.04348191e-02  6.56776244e-01]
 [-4.37295872e-01 -3.05850940e-01]
 [-3.97413112e-01 -2.00023005e-01]
 [-1.59619199e-01  6.82012136e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.46937858e-01 -2.24444836e-01]
 [-4.36607159e-01 -1.84962875e-01]
 [-4.36168887e-01 -2.17932348e-01]
 [-4.60712124e-01 -3.28644650e-01]
 [-4.60273851e-01 -3.29051680e-01]
 [-4.51132748e-01 -2.77765834e-01]
 [-4.60399072e-01 -3.29051680e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.60712124e-01 -3.28970274e-01]
 [-4.60931260e-01 -3.28603946e-01]
 [-4.61087785e-01 -3.29051680e-01]
 [-4.61031436e-01 -3.29051680e-01]
 [-4.60355245e-01 -3.29051680e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.56579844e-01 -3.14805612e-01]
 [-4.57080726e-01 -3.21318100e-01]
 [-4.59084256e-01 -3.22946222e-01]
 [-4.60837344e-01 -3.29051680e-01]
 [-4.61087785e-01 -3.28644650e-01]
 [-4.60962565e-01 -3.29051680e-01]
 [ 4.38770416e+00  1.10889685e+01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 5.05375251e+00  1.26246947e+01]
 [ 5.13533373e+00  9.25814523e+00]
 [ 1.16314857e+00  3.89764222e-01]
 [ 2.90070957e+00  2.89096678e+00]
 [-4.55703300e-01 -3.26202466e-01]
 [-4.59898190e-01 -3.20504039e-01]
 [-4.60712124e-01 -3.29051680e-01]
 [-4.59898190e-01 -3.29051680e-01]
 [-4.58145101e-01 -3.03001727e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
```

```
 [ 1.29640380e-01  4.48783647e-01]
 [ 1.29828211e-01  7.63418241e-01]
 [ 8.78167003e-02  7.31669860e-01]
 [-4.31911386e-01 -3.20097009e-01]
 [-3.31985350e-01 -2.72474437e-01]
 [ 7.92595963e-03  6.60846549e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [-3.39498586e-01 -1.41817640e-01]
 [-3.14141415e-01 -7.99490006e-02]
 [-3.34301931e-01 -1.45073884e-01]
 [-4.66847933e-01 -3.22132161e-01]
 [-4.60273851e-01 -3.24167314e-01]
 [-4.57769440e-01 -2.87941597e-01]
 [-4.61087785e-01 -3.29051680e-01]
 [-4.61087785e-01 -3.29051680e-01]
 [-3.66107963e-01 -3.27830588e-01]
 [-4.58645984e-01 -3.29051680e-01]
 [-4.58708594e-01 -3.29051680e-01]
 [-4.30596570e-01 -3.28644650e-01]
 [-4.28154768e-01 -3.28644650e-01]
 [-4.30909622e-01 -3.28237619e-01]
 [-4.59397307e-01 -3.29051680e-01]
 [-4.59960800e-01 -3.29051680e-01]
 [-4.30596570e-01 -3.29051680e-01]
 [-4.60962565e-01 -3.29051680e-01]
 [-4.60273851e-01 -3.29051680e-01]
 [-4.60023410e-01 -3.28237619e-01]
 [-4.61087785e-01 -3.29051680e-01]
 [-4.60962565e-01 -3.29051680e-01]
 [-4.46186534e-01 -3.14805612e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.48440505e-01 -3.11956398e-01]
 [-4.50193593e-01 -3.17247795e-01]
 [-4.61085657e-01 -3.29051680e-01]
 [-4.60450788e-01 -3.29045982e-01]
 [-4.56925328e-01 -3.19991995e-01]
 [-4.60899954e-01 -3.19282948e-01]
 [-4.60399072e-01 -3.22946222e-01]
 [-4.59898190e-01 -3.27423558e-01]
 [-4.60962565e-01 -3.29051680e-01]
 [-4.61025175e-01 -3.29051680e-01]
 [-4.61087785e-01 -3.27423558e-01]
 [-4.60962565e-01 -3.17247795e-01]
 [-4.61087785e-01 -3.29051680e-01]
 [-4.61087785e-01 -3.29051680e-01]
 [-4.61087785e-01 -3.25608202e-01]
 [-4.52766877e-01 -2.56478138e-01]
 [-4.51295535e-01 -2.75567869e-01]
 [-4.60899954e-01 -3.29051680e-01]
```

```
 [-4.60098543e-01 -3.27830588e-01]
 [-4.55271289e-01 -2.68973975e-01]
 [-3.39498586e-01 -1.05591924e-01]
 [-3.37056784e-01 -1.50365281e-01]
 [-3.16520606e-01 -2.83464262e-01]
 [-4.65407896e-01 -3.29051680e-01]
 [-4.56767675e-01 -3.30679802e-01]
 [-3.99166200e-01 -2.06128462e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.58833815e-01 -3.27016527e-01]
 [-4.59334697e-01 -3.29051680e-01]
 [-4.58395543e-01 -3.27830588e-01]
 [-4.61025175e-01 -3.29051680e-01]
 [-4.61025175e-01 -3.29051680e-01]
 [-4.59084256e-01 -3.29051680e-01]
 [-4.59585138e-01 -3.25795436e-01]
 [-4.47376130e-01 -2.65147888e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.05677671e-01 -3.11549368e-01]
 [-3.94032156e-01 -2.90790811e-01]
 [-3.93969546e-01 -2.94861116e-01]
 [-4.58645984e-01 -3.29051680e-01]
 [-4.57205947e-01 -3.27423558e-01]
 [-4.41490762e-01 -3.07479062e-01]
 [-4.57143337e-01 -3.00966574e-01]
 [-4.60712124e-01 -3.26609497e-01]
 [-4.47939623e-01 -2.02058157e-01]
 [-2.54661632e-01 -2.61891644e-01]
 [-4.35229732e-01 -3.28237619e-01]
 [-3.53147631e-01 -3.11142337e-01]
 [-2.26486997e-01 -2.58228369e-01]
 [ 9.50168846e-02  6.33257431e-02]
 [ 1.18900663e+00 -3.03815788e-01]
 [ 3.49339917e-01 -3.13584520e-01]
 [-1.41399602e-01 -3.18875917e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-1.44780558e-01 -2.97303299e-01]
 [-1.36891660e-01 -3.28644650e-01]
 [-3.32924505e-01 -3.29051680e-01]
```

```
 [-3.14204025e-01 -3.27830588e-01]
 [-1.66381111e-01 -3.28237619e-01]
 [-4.35354953e-01 -3.12770459e-01]
 [-4.38798519e-01 -3.15212642e-01]
 [-4.60899954e-01 -3.29051680e-01]
 [ 4.89336544e-01  2.54223058e-01]
 [ 6.50996334e-01  2.74167553e-01]
 [-1.66193280e-01 -2.94861116e-01]
 [ 2.45176887e-02 -2.45610423e-01]
 [ 2.76899801e-01 -2.54069107e-02]
 [ 3.85451486e+00  5.12687439e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 4.18497201e+00  8.14297056e-01]
 [ 3.67494852e+00  4.27618060e-01]
 [ 1.79870571e+00 -2.76544743e-01]
 [ 2.90816020e+00 -2.17932348e-01]
 [ 4.81551159e+00  4.33601405e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-2.84651965e-01 -2.47645576e-01]
 [-2.96986193e-01 -2.25665927e-01]
 [-3.26788695e-01 -2.21595622e-01]
 [-4.33476644e-01 -3.26609497e-01]
 [-3.72682044e-01 -3.17654825e-01]
 [-3.04875091e-01 -2.95675177e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.16759694e-01 -2.60263522e-01]
 [-4.02108884e-01 -2.15083134e-01]
 [-4.16822304e-01 -2.08163615e-01]
 [-4.62089550e-01 -3.31493863e-01]
 [-4.60837344e-01 -3.24167314e-01]
 [-4.57518998e-01 -2.79393956e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 9.53299361e-02  8.57124218e-02]
 [ 6.42752282e-02  5.05767920e-01]
 [ 5.46958525e-02  3.77553306e-01]
 [-4.31598335e-01 -3.29051680e-01]
 [ 9.34182714e-01 -3.26202466e-01]
 [ 2.29103278e-03 -1.38968426e-01]
 [-4.47125689e-01 -3.14805612e-01]
 [-4.60712124e-01 -3.26202466e-01]
 [-4.56642454e-01 -3.29051680e-01]
 [-4.49504880e-01 -3.24981375e-01]
 [-4.60586903e-01 -3.29051680e-01]
 [-4.59397307e-01 -3.28237619e-01]
```

```
 [-4.38485468e-01 -2.59042430e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.36419328e-01 -2.61077583e-01]
 [-4.33852306e-01 -2.63112735e-01]
 [-4.56830285e-01 -3.28237619e-01]
 [-4.51132748e-01 -3.27016527e-01]
 [-4.37796754e-01 -2.83057231e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 1.09041591e-01  2.22913515e+00]
 [ 1.53494903e-01  3.02772903e+00]
 [ 1.43539866e-01  2.37322395e+00]
 [-3.08569099e-01 -3.09107184e-01]
 [-1.62812324e-01 -1.89847242e-01]
 [ 7.39798244e-02  1.71831185e+00]
 [-4.03486311e-01 -1.99615974e-01]
 [-3.86080648e-01 -1.19837992e-01]
 [-3.99980134e-01 -1.15767687e-01]
 [-4.58708594e-01 -3.27830588e-01]
 [-4.60524293e-01 -3.29051680e-01]
 [-4.40363777e-01 -2.45203393e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 1.22826328e+00 -2.67590071e-01]
 [ 1.73985203e+00 -2.49680728e-01]
 [ 1.75181060e+00 -2.51308850e-01]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [ 0.00000000e+00  4.62740129e-17]
 [-4.38360247e-01 -2.02872218e-01]
 [-4.41428152e-01 -1.54435586e-01]
 [-4.40426387e-01 -1.47923098e-01]]
```

```python
import pandas as pd
from sklearn.preprocessing import StandardScaler
```

```python
# Step 1: Load the dataset
df = pd.read_csv('travela .csv')

# Step 2: Select relevant columns (features) and target variable
selected_columns = [
    'year', 'operational_activities_kdollars',
'key_stakeholders_kdollars',
    'training_kdollars', 'other_kdollars',
'internal_governance_kdollars',
    'non_public_servants_kdollars', 'public_servants_kdollars',
    'hospitality_kdollars', 'conference_fees_kdollars',
'minister_kdollars',
    'travel_compared_fiscal_year_en',
'travel_compared_fiscal_year_fr',
    'hospitality_compared_fiscal_year_en',
'hospitality_compared_fiscal_year_fr',
    'conference_fees_compared_fiscal_year_en',
'conference_fees_compared_fiscal_year_fr',
    'minister_compared_fiscal_year_en',
'minister_compared_fiscal_year_fr',
    'owner_org', 'owner_org_title'
]

selected_features = df[selected_columns]

# Step 3: Data Preprocessing
# Replace non-numeric values with NaN
selected_features = selected_features.apply(pd.to_numeric,
errors='coerce')

# Fill NaN with the mean
selected_features = selected_features.fillna(selected_features.mean())

# Step 4: Transform the Data
scaler = StandardScaler()
scaled_features =
scaler.fit_transform(selected_features[['operational_activities_kdolla
rs', 'key_stakeholders_kdollars']])

import pandas as pd

# Assuming 'df' is your DataFrame loaded with data

# 1. Handling Missing Values
# Drop rows with missing values
df_cleaned = df.dropna()

# Alternatively, fill missing values with a specific value or method
# df_cleaned = df.fillna(value)
```

```
# 2. Removing Duplicates
df_cleaned = df_cleaned.drop_duplicates()

# 3. Dealing with Outliers
# Identify and handle outliers using statistical methods or domain
knowledge

# 4. Transforming Data Types
# Convert data types as needed
# df_cleaned['column_name'] = pd.to_numeric(df_cleaned['column_name'],
errors='coerce')

# Additional Cleaning Steps
# - Renaming columns if needed
# df_cleaned = df_cleaned.rename(columns={'old_column_name':
'new_column_name'})

# - Extracting information from columns
# df_cleaned['new_column'] =
df_cleaned['original_column'].apply(lambda x: custom_function(x))

# - Handling datetime data
# df_cleaned['datetime_column'] =
pd.to_datetime(df_cleaned['datetime_column'])

# - ... (Additional steps based on your specific data and cleaning
requirements)

# Display the cleaned DataFrame
df_cleaned.head()
```

{"type":"dataframe","variable_name":"df_cleaned"}

```
import pandas as pd

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 489 entries, 0 to 488
Data columns (total 23 columns):
 #   Column                                   Non-Null Count  Dtype
---  ------                                   --------------  -----
 0   year                                     489 non-null    int64
 1   mandate_description_en                   478 non-null    object
 2   mandate_description_fr                   478 non-null    object
 3   operational_activities_kdollars          489 non-null    object
 4   key_stakeholders_kdollars                489 non-null    object
 5   training_kdollars                        489 non-null    object
 6   other_kdollars                           489 non-null    object
 7   internal_governance_kdollars             489 non-null    object
 8   non_public_servants_kdollars             135 non-null    float64
```

```
 9   public_servants_kdollars                    135 non-null    float64
 10  hospitality_kdollars                        489 non-null    float64
 11  conference_fees_kdollars                    489 non-null    float64
 12  minister_kdollars                           489 non-null    float64
 13  travel_compared_fiscal_year_en              447 non-null    object
 14  travel_compared_fiscal_year_fr              449 non-null    object
 15  hospitality_compared_fiscal_year_en         364 non-null    object
 16  hospitality_compared_fiscal_year_fr         383 non-null    object
 17  conference_fees_compared_fiscal_year_en     351 non-null    object
 18  conference_fees_compared_fiscal_year_fr     371 non-null    object
 19  minister_compared_fiscal_year_en            195 non-null    object
 20  minister_compared_fiscal_year_fr            252 non-null    object
 21  owner_org                                   489 non-null    object
 22  owner_org_title                             489 non-null    object
dtypes: float64(5), int64(1), object(17)
memory usage: 88.0+ KB
```

```
import pandas as pd

df_head = df.head()

# Convert DataFrame to markdown format
markdown_table = df_head.to_markdown()

# Print the markdown table
print(markdown_table)
```

```
|    |   year | mandate_description_en
| mandate_description_fr
| operational_activities_kdollars   | key_stakeholders_kdollars   |
training_kdollars   | other_kdollars   | internal_governance_kdollars
|   non_public_servants_kdollars |   public_servants_kdollars |
hospitality_kdollars |   conference_fees_kdollars |
minister_kdollars | travel_compared_fiscal_year_en
| travel_compared_fiscal_year_fr
| hospitality_compared_fiscal_year_en
| hospitality_compared_fiscal_year_fr
| conference_fees_compared_fiscal_year_en
| conference_fees_compared_fiscal_year_fr
| minister_compared_fiscal_year_en
| minister_compared_fiscal_year_fr
| owner_org   | owner_org_title
|
|---:|-------:|:-----------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
```

```
----------------------------------------------------------------------
--------------------------------------------------------|:------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
--------------------------|:----------------------------------|:------
----------------------|:--------------------|:-----------------|:-----
--------------------------|-------------------------------:|----------
-----------------:|-----------------------:|--------------------------
-:|--------------------:|:--------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
-------|:-------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
-|:-------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
---------------------------------------------------------------|:-----
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
```

```
----------------------------------------------------------------------
--------------|:------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
---------------------|:-----------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
-|:-------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
--------------------------------------------------------|:------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
-------|:------------|:-----------------------------------------------
--------------------------|
```

```
|  0 |   2020 | Expenditures on travel, hospitality and conference
fees incurred by Accessibility Standards Canada support the
organization's mandate. The organization's mandate is to contribute to
a barrier-free Canada through the development and review of
accessibility standards, promotion of research on barrier
identification, prevention and removal of barriers, and finally,
sharing information related to accessibility. Further details about
the organization's mandate and program activities can be found in the
organization's Departmental Results Report, Departmental Plan and on
the organization's website. | Les dépenses de voyage, d'accueil et de
conférence engagées par Normes d'accessibilité Canada soutiennent le
mandat de l'organisation. Ce mandat est de contribuer à la création
d'un Canada exempt d'obstacle par l'entremise de l'élaboration et
l'examination des normes d'accessibilité, de promouvoir la recherche
sur la reconnaissance d'obstacle, la prévention et l'élimination des
obstacles, et finalement, de transmettre l'information sur
l'accessibilité. Le Rapport sur les résultats ministériels et le Plan
ministériel contiennent d'autres renseignements sur le mandat et les
activités de l'organisation et peuvent être consultés sur le site web
de l'organisation. | 78                                | 0
| 0                   | 18               | 0
|                             nan |                            nan |
6 |                           0 |                   0 | nan
| S/O
| nan
| S/O
| nan
| S/O
| nan
| S/O
| casdo-ocena | Accessibility Standards Canada | Normes
d’accessibilité Canada           |
|  1 |   2021 | Expenditures on travel, hospitality and conference
fees incurred by Accessibility Standards Canada support the
organization's mandate. The organization's mandate is to contribute to
a barrier-free Canada through the development and review of
accessibility standards, promotion of research on barrier
identification, prevention and removal of barriers, and finally,
sharing information related to accessibility. Further details about
the organization's mandate and program activities can be found in the
organization's Departmental Results Report, Departmental Plan and on
the organization's website. | Les dépenses de voyage, d'accueil et de
conférence engagées par Normes d'accessibilité Canada soutiennent le
mandat de l'organisation. Ce mandat est de contribuer à la création
d'un Canada exempt d'obstacle par l'entremise de l'élaboration et
l'examination des normes d'accessibilité, de promouvoir la recherche
sur la reconnaissance d'obstacle, la prévention et l'élimination des
obstacles, et finalement, de transmettre l'information sur
l'accessibilité. Le Rapport sur les résultats ministériels et le Plan
```

```
ministériel contiennent d'autres renseignements sur le mandat et les
activités de l'organisation et peuvent être consultés sur le site web
de l'organisation. | 15                              | 0
| 0                  | 0                 | 0
|                            nan |                            nan |
0 |                          0 |                      0 | Travel expenses
were limited due to the COVID-19 pandemic.
| Les frais de voyages ont été limités due à la pandémie de la COVID-
19.
| Hospitality expenses were limited due to the COVID-19 pandemic.
| Les frais d'accueil ont été limités due à la pandémie de la COVID-
19.
| nan
| S/O
| nan
| S/O
| casdo-ocena | Accessibility Standards Canada | Normes
d’accessibilité Canada           |
|  2 |   2022 | Expenditures on travel, hospitality and conference
fees incurred by Accessibility Standards Canada support the
organization's mandate. The organization's mandate is to contribute to
a barrier-free Canada through the development and review of
accessibility standards, promotion of research on barrier
identification, prevention and removal of barriers, and finally,
sharing information related to accessibility. Further details about
the organization's mandate and program activities can be found in the
organization's Departmental Results Report, Departmental Plan and on
the organization's website. | Les dépenses de voyage, d'accueil et de
conférence engagées par Normes d'accessibilité Canada soutiennent le
mandat de l'organisation. Ce mandat est de contribuer à la création
d'un Canada exempt d'obstacle par l'entremise de l'élaboration et
l'examination des normes d'accessibilité, de promouvoir la recherche
sur la reconnaissance d'obstacle, la prévention et l'élimination des
obstacles, et finalement, de transmettre l'information sur
l'accessibilité. Le Rapport sur les résultats ministériels et le Plan
ministériel contiennent d'autres renseignements sur le mandat et les
activités de l'organisation et peuvent être consultés sur le site web
de l'organisation. | 0                               | 6
| 0                  | 0                 | 0
|                            nan |                            nan |
0 |                          0 |                      0 | Travel expenses
were limited due to the COVID-19 pandemic.
| Les frais de voyages ont été limités due à la pandémie de la COVID-
19.
| nan
| S/O
| nan
| S/O
| nan
```

```
| S/O
| casdo-ocena | Accessibility Standards Canada | Normes
d’accessibilité Canada           |
|  3 |   2016 | The Department supports the sector from the farmer to
the consumer, from the farm to global markets, through all phases of
producing, processing and marketing of farm, food and agri-based
products. Agriculture is a shared jurisdiction in Canada, and the
Department works closely with provincial and territorial governments
in the development and delivery of policies, programs and services.
| Le Ministère soutient le secteur entre l’agriculteur et le
consommateur, entre l’exploitation agricole et les marchés mondiaux et
concernent toutes les phases de la production, la transformation et la
commercialisation des produits de la ferme, de l’alimentation et agro-
industriels. L’agriculture relève d’une compétence partagée au Canada,
et le Ministère collabore étroitement avec les gouvernements
provinciaux et territoriaux dans le cadre de l’élaboration et de
l’exécution de politiques, de programmes et de services.
| .                                   | .                             | .
| .                 | .                               |
718 |                       7967 |                       226 |
148 |                  29 | Public Servants: 7967 ;
| Voyages des fonctionnaires: 7967 ;
| nan
| nan
| nan
| nan
| nan
| nan
| aafc-aac    | Agriculture and Agri-Food Canada | Agriculture et
Agroalimentaire Canada |
|    |        |
|
|                                   |                             |
|                 |                               |
|                           |                          |
|                    | Non-Public Servants: 718
| Voyages des non-fonctionnaires: 718
|
|
|
|
|
|
|             |
|
|  4 |   2017 | The Department supports the sector from the farmer to
the consumer, from the farm to global markets, through all phases of
producing, processing and marketing of farm, food and agri-based
products. Agriculture is a shared jurisdiction in Canada, and the
```

Department works closely with provincial and territorial governments in the development and delivery of policies, programs and services. | Le Ministère soutient le secteur entre l’agriculteur et le consommateur, entre l’exploitation agricole et les marchés mondiaux et concernent toutes les phases de la production, la transformation et la commercialisation des produits de la ferme, de l’alimentation et agro-industriels. L’agriculture relève d’une compétence partagée au Canada, et le Ministère collabore étroitement avec les gouvernements provinciaux et territoriaux dans le cadre de l’élaboration et de l’exécution de politiques, de programmes et de services.
| . | . | .
| . | . |
1173 | 9366 | 367 |
202 | 185 | Public Servants: 9366 The $1.4 million overall increase in travel expenditures was partly attributable to the lifting of restrictions related to the Caretaker Convention that had been in effect in 2015-2016. Under the Caretaker Convention, the government acts with restraint during an election, confining itself to necessary routine or urgent public business. The increase was primarily in support of AAFC's mandate, including increased scientific missions to China and large joint conferences of scientific societies. It also includes travel for accelerated infrastructure activities, the canola trade issue with China, the Canadian Agriculture Partnership (CAP) and Federal, Provincial, Territorial (FPT) meetings. ;
| Voyages des fonctionnaires: 9366 L’augmentation de 1,4 million de dollars des dépenses de voyages est en partie attribuable à la levée des restrictions liées à la convention de transition en vigueur depuis 2015-2016. Aux termes de cette convention, le gouvernement fait preuve de retenue et s’en tient aux affaires d’intérêts publics courantes ou urgentes. L’augmentation est en partie attribuable aux activités entreprises pour réaliser le mandat d’AAC, ce qui comprend un nombre accru de missions scientifiques en Chine et de grandes conférences conjointes avec des sociétés scientifiques, des voyages pour la réalisation d’activités d’infrastructure accélérées, le dossier sur le commerce du canola avec la Chine, le Partenariat canadien pour l’agriculture (PCA) et les réunions fédérales, provinciales et territoriales (FPT). ; | Compared with fiscal year 2015-2016, departmental hospitality expenditures have increased by $141 thousand. This increase was mainly due to the CAP and Federal, Provincial, Territorial (FPT) meetings as well as increased number of roundtables, industry consultations and missions. It can also be further attributed to increased scientific missions to China and large joint conferences of scientific societies. | Comparativement à l’exercice 2015-2016, les dépenses ministérielles en matière d’hébergement ont augmenté de 141 000 $. Cette augmentation est principalement attribuable au PCA, aux réunions FPT et au nombre accru de tables rondes, de consultations avec l’industrie et de missions. Elle peut également être attribuée au nombre accru de missions scientifiques en Chine et aux grandes conférences conjointes avec des sociétés scientifiques. | Compared

with fiscal year 2015-2016 departmental conference fees expenditures increased by $54 thousand primarily due to a reclassification of expenditures in 2016-2017 from training to conference fees, scientific missions to China and large joint conferences of scientific societies. | Comparativement à l’exercice 2015-2016, les dépenses ministérielles pour participation aux conférences ont augmenté de 54 000 $ principalement en raison de la reclassification des dépenses en 2016-2017 (qui sont passés de dépenses de formation en dépenses de participation aux conférences), des missions scientifiques en Chine et des grandes conférences conjointes avec des sociétés scientifiques. | The Minister and his staff participated in a number of trade missions to advance the Government’s overall priority of strengthening the economy, while also deepening key agricultural partnerships to position the sector to take full advantage of market access opportunities. Ministerial trade missions were carried out to support and enhance the competitiveness of the agriculture and agri-food sector by: advocating on issues of importance to the Canadian sector with key decision makers including Canada’s position on agricultural biotechnology and market access in India and Vietnam (March 2016); supporting the sector in developing markets abroad through participation in key international trade shows and conferences such as the China Fish and Seafood Expo in Qingdao and the Food and Hospitality China in Shanghai (November 2016); deepening bilateral relations with key trading partners including the European Union, China, India, Japan and Vietnam; promoting the Canada-European Union Comprehensive Economic and Trade Agreement (CETA) and the importance of open trade and strong cooperation during the Agriculture Ministers' Summit in Berlin, Germany (January 2017). | Le ministre et son personnel ont participé à diverses missions commerciales dans le but de réaliser la grande priorité du gouvernement consistant à stimuler l’économie, tout en renforçant les partenariats agricoles clés pour aider le secteur à profiter pleinement des possibilités d’accès aux marchés. Des missions commerciales ministérielles ont été effectuées afin d’accroître la compétitivité du secteur de l’agriculture et de l’agroalimentaire, notamment en accomplissant ce qui suit : souligner aux grands décideurs les enjeux importants pour le secteur canadien, comme la position du Canada sur la biotechnologie agricole et l’accès au marché en Inde et au Vietnam (mars 2016); aider le secteur à développer des marchés à l’étranger en participant à des foires commerciales et à des conférences internationales d’envergure, comme la China Fish and Seafood Expo à Qingdao et à la Food and Hospitality China à Shanghai (novembre 2016); renforcer les relations bilatérales avec les partenaires commerciaux clés, comme l’Union européenne, la Chine, l’Inde, le Japon et le Vietnam; faire la promotion de l’Accord économique et commercial global entre le Canada et l’Union européenne (AECG) ainsi que de l’importance du commerce ouvert et d’une solide coopération lors du Sommet des ministres de l’Agriculture à Berlin, en Allemagne (janvier 2017). | aafc-aac | Agriculture and Agri-Food Canada | Agriculture et Agroalimentaire Canada |

```
|     |        |
|
|                                   |                              |
|                  |                                |
|                            |                                |
|                     | Non-Public Servants: 1173 2016-2017
departmental travel expenditures by non-public servants increased by
$455 thousand in comparison to 2015-2016, of which $156 thousand was
related to travel expenditures for the Minister and his staff which
was mainly due to the lifting of restrictions related to the Caretaker
Convention. Under the Caretaker Convention, the government acts with
restraint during an election, confining itself to necessary routine or
urgent public business. The increases in other travel expenditures for
non-public servants were attributed to travel for the Grain Monitoring
Program, consultations on the development of the Canadian Agricultural
Partnership (CAP), as well as conducting interviews to increase
scientific capacity. | Voyages des non-fonctionnaires: 1173 Les
dépenses de voyages des non-fonctionnaires ont augmenté de 455 000 $
en 2016-2017 comparativement à 2015-2016. De ce montant, 156 000 $
étaient liés aux dépenses de voyages du ministre et de son personnel
qui sont principalement attribuables à la levée des restrictions
relatives à la convention de transition. Aux termes de cette
convention, le gouvernement fait preuve de retenue et s’en tient aux
affaires d’intérêts publics courantes ou urgentes. Les augmentations
des autres dépenses de voyage des non-fonctionnaires sont attribuables
aux voyages effectués dans le cadre du Programme de surveillance du
grain, des consultations sur l’établissement du Partenariat canadien
pour l’agriculture (PCA) et des entrevues visant à renforcer la
capacité scientifique.                               |
|
|
|
|
|
|             |
|

import pandas as pd

data_types = df.dtypes
unique_values = df.nunique()

# Combine the information into a new DataFrame for better display
info_df = pd.DataFrame({
    'Data Types': data_types,
    'Unique Values': unique_values
})

# Print the information DataFrame
print(info_df)
```

```
                                        Data Types  Unique Values
year                                         int64             13
mandate_description_en                      object            284
mandate_description_fr                      object            287
operational_activities_kdollars             object            302
key_stakeholders_kdollars                   object            224
training_kdollars                           object            216
other_kdollars                              object            192
internal_governance_kdollars                object            161
non_public_servants_kdollars               float64            121
public_servants_kdollars                   float64            134
hospitality_kdollars                       float64            241
conference_fees_kdollars                   float64            240
minister_kdollars                          float64            109
travel_compared_fiscal_year_en              object            439
travel_compared_fiscal_year_fr              object            441
hospitality_compared_fiscal_year_en         object            346
hospitality_compared_fiscal_year_fr         object            351
conference_fees_compared_fiscal_year_en     object            330
conference_fees_compared_fiscal_year_fr     object            332
minister_compared_fiscal_year_en            object            138
minister_compared_fiscal_year_fr            object            148
owner_org                                   object             71
owner_org_title                             object             71
```

```
import pandas as pd

selected_column = 'operational_activities_kdollars'
summary_stats_selected = df[[selected_column]].describe()

# Print the summary statistics for the selected column
print(summary_stats_selected)
```

```
       operational_activities_kdollars
count                              489
unique                             302
top                                  .
freq                               135
```

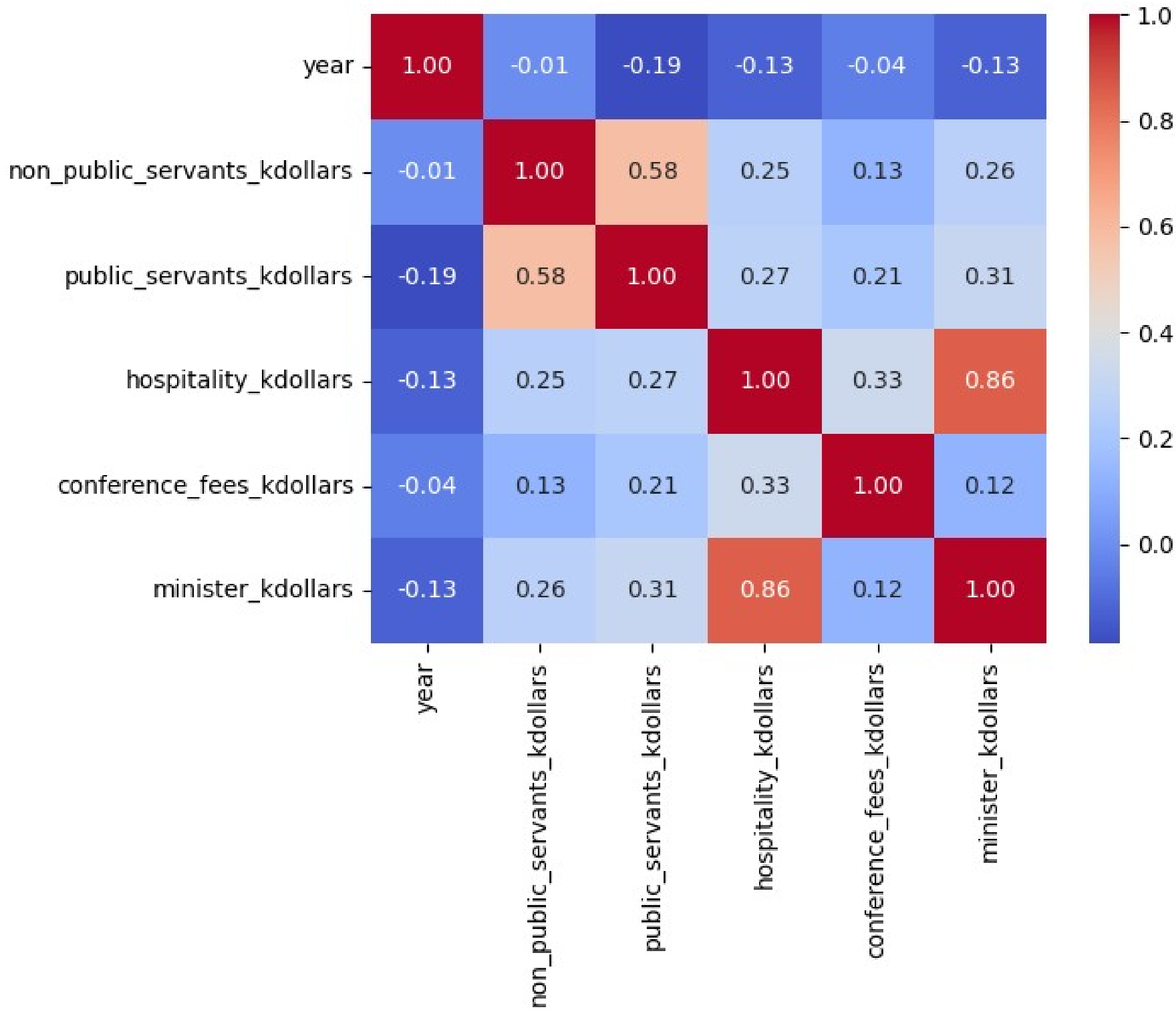
```
# Display the first few rows of the DataFrame with styling
df.head().style.set_table_styles([
    {'selector': 'thead', 'props': [('background', '#606060'),
('color', 'white')]},
    {'selector': 'tbody', 'props': [('border', '1px solid #cccccc')]},
    {'selector': 'th', 'props': [('background', '#f2f2f2')]},
    {'selector': 'td', 'props': [('border', '1px solid #cccccc')]},
])
```

```
<pandas.io.formats.style.Styler at 0x7fb3a0c2e5f0>
```

```python
print("Correlation Matrix:")
print(correlation_matrix)
```

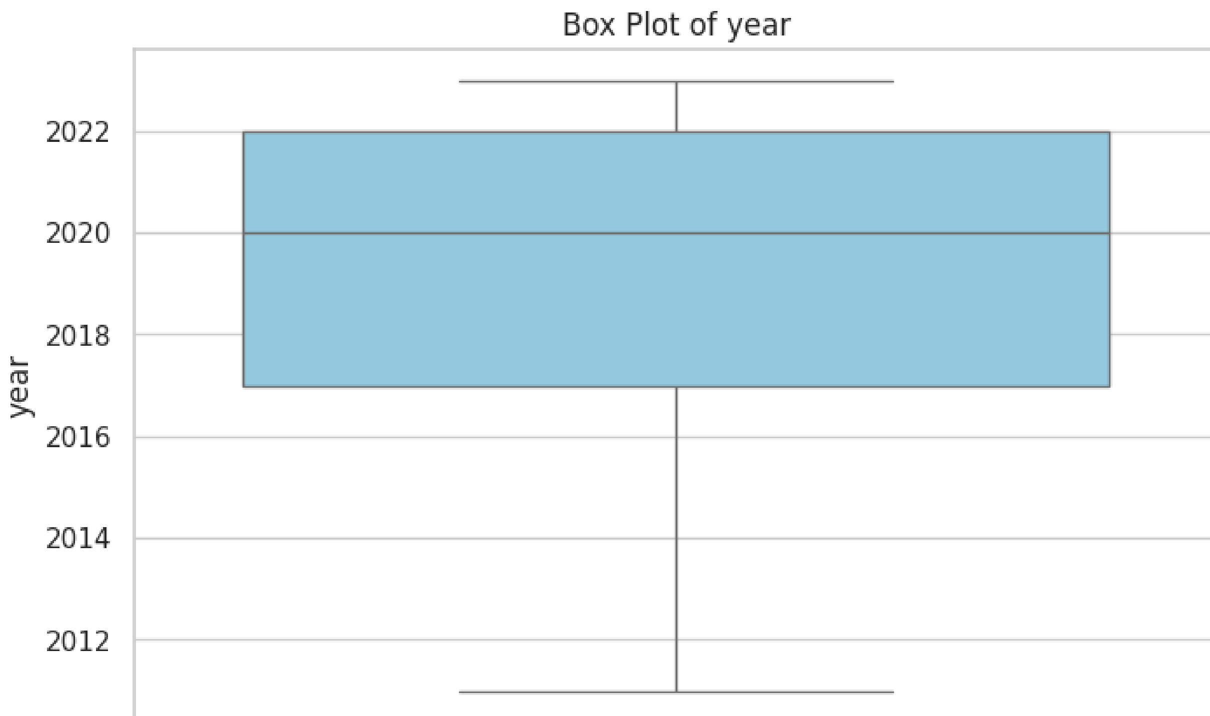
```
Correlation Matrix:
                                  year
non_public_servants_kdollars  \
year                          1.000000                        -0.005596

non_public_servants_kdollars -0.005596                         1.000000

public_servants_kdollars     -0.187495                         0.580210

hospitality_kdollars         -0.128661                         0.254034

conference_fees_kdollars     -0.044627                         0.127818

minister_kdollars            -0.128363                         0.263577


                              public_servants_kdollars
hospitality_kdollars  \
year                                         -0.187495              -
0.128661
non_public_servants_kdollars                  0.580210
0.254034
public_servants_kdollars                      1.000000
0.273066
hospitality_kdollars                          0.273066
1.000000
conference_fees_kdollars                      0.210900
0.334146
minister_kdollars                             0.313229
0.855256

                              conference_fees_kdollars
minister_kdollars
year                                         -0.044627           -
0.128363
non_public_servants_kdollars                  0.127818
0.263577
public_servants_kdollars                      0.210900
0.313229
hospitality_kdollars                          0.334146
0.855256
conference_fees_kdollars                      1.000000
0.123631
minister_kdollars                             0.123631
1.000000
```

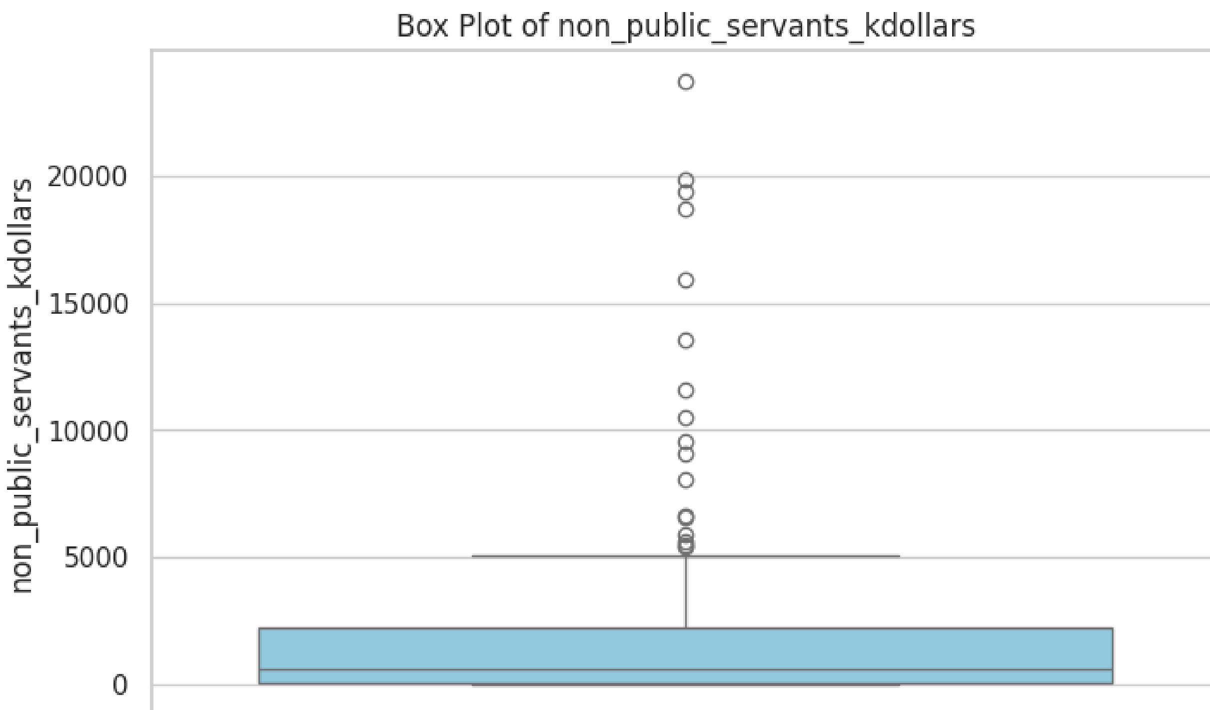
```python
import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt=".2f")
plt.show()
```

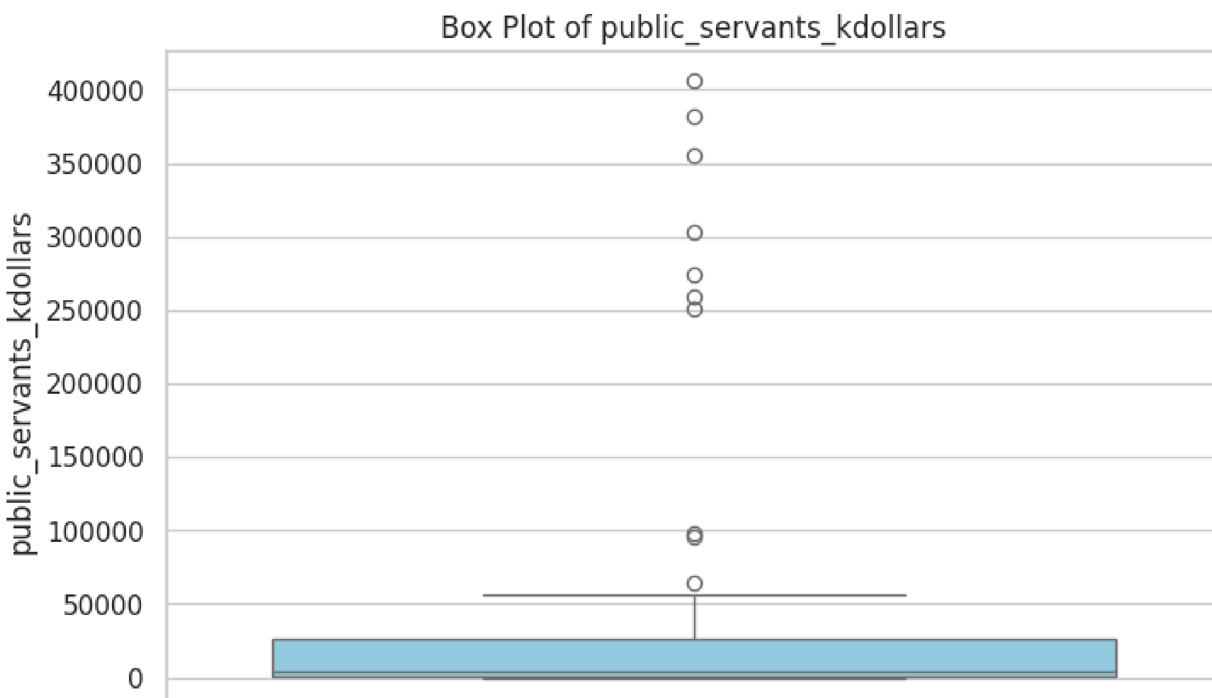


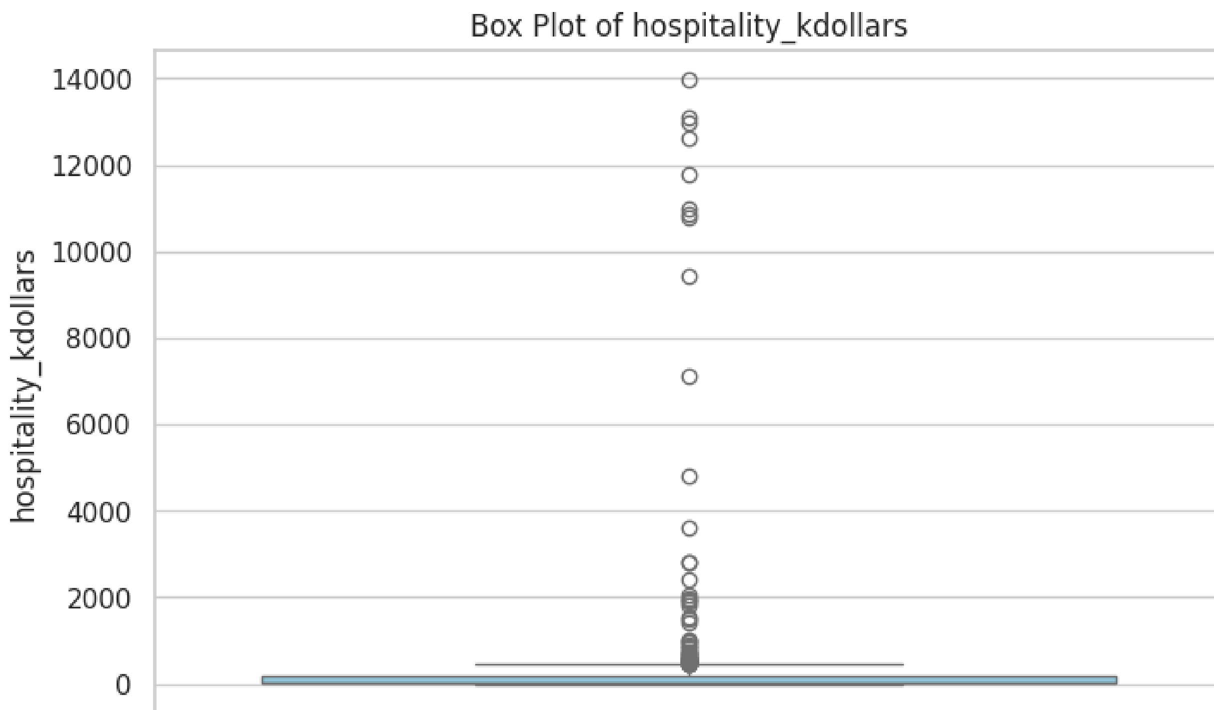

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Select numerical columns for analysis
numerical_columns = df.select_dtypes(include=['float64',
'int64']).columns

# Set the style for Seaborn
sns.set(style="whitegrid")
```

```python
# Plot colorful box plots for each numerical column
for column in numerical_columns:
    plt.figure(figsize=(8, 5))
    sns.boxplot(data=df, y=column, color='skyblue')
    plt.title(f'Box Plot of {column}')
    plt.ylabel(column)
    plt.show()
```

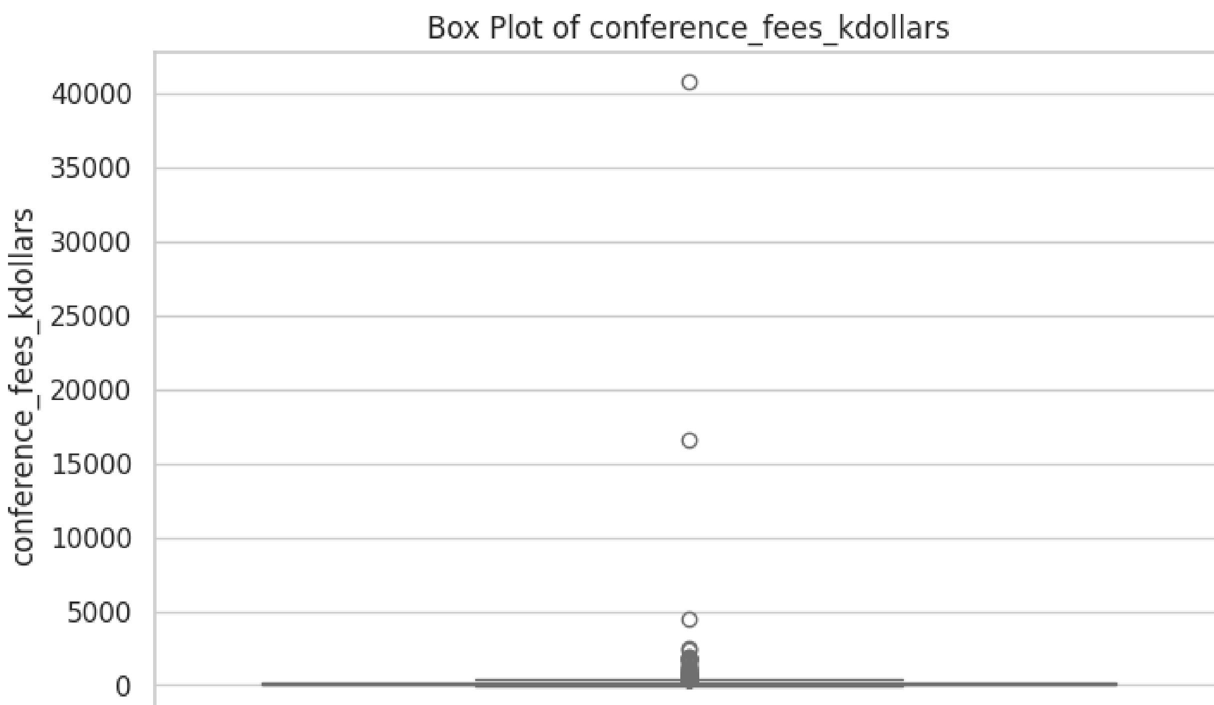



Box Plot of non_public_servants_kdollars
non_public_servants_kdollars
20000
15000
10000
5000
0

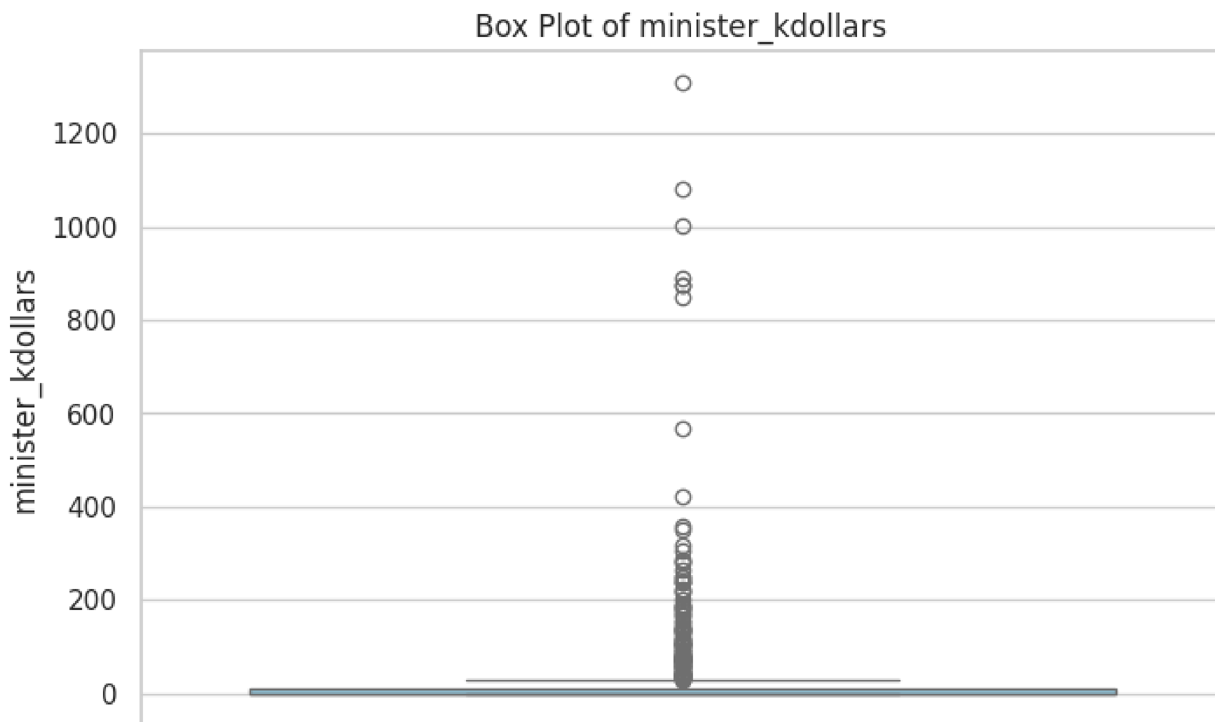

Box Plot of public_servants_kdollars
public_servants_kdollars
400000
350000
300000
250000
200000
150000
100000
50000
0



Box Plot of hospitality_kdollars
hospitality_kdollars
14000
12000
10000
8000
6000
4000
2000
0



Box Plot of conference_fees_kdollars
conference_fees_kdollars
40000
35000
30000
25000
20000
15000
10000
5000
0



Models evaluation, including testing options, exploring algorithms, and reporting results

```
# Import required libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold

import pandas as pd

df = pd.read_csv("travela  (2).csv")

# Print the column names
print(df.columns)

Index(['year', 'mandate_description_en', 'mandate_description_fr',
       'operational_activities_kdollars', 'key_stakeholders_kdollars',
       'training_kdollars', 'other_kdollars',
'internal_governance_kdollars',
       'non_public_servants_kdollars', 'public_servants_kdollars',
       'hospitality_kdollars', 'conference_fees_kdollars',
'minister_kdollars',
       'travel_compared_fiscal_year_en',
'travel_compared_fiscal_year_fr',
       'hospitality_compared_fiscal_year_en',
```

```
       'hospitality_compared_fiscal_year_fr',
       'conference_fees_compared_fiscal_year_en',
       'conference_fees_compared_fiscal_year_fr',
       'minister_compared_fiscal_year_en',
'minister_compared_fiscal_year_fr',
       'owner_org', 'owner_org_title'],
      dtype='object')
```

```
from sklearn.model_selection import train_test_split

X = df.drop("operational_activities_kdollars", axis=1)  # Features
(exclude the target variable)
y = df["operational_activities_kdollars"]  # Target variable

# Split the data into training and testing sets (80% training, 20%
testing)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Print the shape of the resulting sets
print("Training set shape:", X_train.shape, y_train.shape)
print("Testing set shape:", X_test.shape, y_test.shape)
```

```
Training set shape: (391, 22) (391,)
Testing set shape: (98, 22) (98,)
```

K-Means Clustering:

```
import warnings
warnings.filterwarnings("ignore")

# Import necessary libraries
from sklearn.cluster import KMeans
from sklearn.impute import SimpleImputer
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv('travela  (2).csv')

# Select numerical columns for clustering
numerical_columns = ['operational_activities_kdollars',
'key_stakeholders_kdollars', 'training_kdollars']

# Check for non-numeric values and handle them
for column in numerical_columns:
    non_numeric_values = df[column].loc[~df[column].apply(lambda x:
str(x).replace('.', '').isnumeric())]
    if not non_numeric_values.empty:
        # Handle non-numeric values (replace or remove)
```

```
        df[column] = pd.to_numeric(df[column], errors='coerce')

# Impute missing values if any
imputer = SimpleImputer(strategy='mean')  # You can choose a different
strategy based on your needs
df[numerical_columns] = imputer.fit_transform(df[numerical_columns])

# Select only numerical columns for clustering
X = df[numerical_columns].values

# Choose the number of clusters (you may need to experiment or use
domain knowledge)
num_clusters = 3

# Fit K-Means model
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
clusters = kmeans.fit_predict(X)

# Visualize the clusters (for 2D data, adapt as needed)
plt.scatter(X[:, 0], X[:, 1], c=clusters, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,
1], s=300, c='red')  # Plot centroids
plt.title('K-Means Clustering')
plt.show()
```

Linear Regression:

```
import pandas as pd

df = pd.read_csv('travela  (2).csv')

# Display the data types of each column
print(df.dtypes)

# Separate columns into categorical and numerical
categorical_columns =
df.select_dtypes(include=['object']).columns.tolist()
numerical_columns =
df.select_dtypes(exclude=['object']).columns.tolist()

print("Categorical Columns:", categorical_columns)
print("Numerical Columns:", numerical_columns)
```

```
year                                               int64
mandate_description_en                            object
mandate_description_fr                            object
operational_activities_kdollars                   object
key_stakeholders_kdollars                         object
training_kdollars                                 object
```

```
other_kdollars                                       object
internal_governance_kdollars                         object
non_public_servants_kdollars                        float64
public_servants_kdollars                            float64
hospitality_kdollars                                float64
conference_fees_kdollars                            float64
minister_kdollars                                   float64
travel_compared_fiscal_year_en                       object
travel_compared_fiscal_year_fr                       object
hospitality_compared_fiscal_year_en                  object
hospitality_compared_fiscal_year_fr                  object
conference_fees_compared_fiscal_year_en              object
conference_fees_compared_fiscal_year_fr              object
minister_compared_fiscal_year_en                     object
minister_compared_fiscal_year_fr                     object
owner_org                                            object
owner_org_title                                      object
dtype: object
Categorical Columns: ['mandate_description_en',
'mandate_description_fr', 'operational_activities_kdollars',
'key_stakeholders_kdollars', 'training_kdollars', 'other_kdollars',
'internal_governance_kdollars', 'travel_compared_fiscal_year_en',
'travel_compared_fiscal_year_fr',
'hospitality_compared_fiscal_year_en',
'hospitality_compared_fiscal_year_fr',
'conference_fees_compared_fiscal_year_en',
'conference_fees_compared_fiscal_year_fr',
'minister_compared_fiscal_year_en',
'minister_compared_fiscal_year_fr', 'owner_org', 'owner_org_title']
Numerical Columns: ['year', 'non_public_servants_kdollars',
'public_servants_kdollars', 'hospitality_kdollars',
'conference_fees_kdollars', 'minister_kdollars']

non_numeric_values =
df['operational_activities_kdollars'].loc[~df['operational_activities_
kdollars'].astype(str).str.isnumeric()]
print(non_numeric_values)

3      .
4      .
12     .
13     .
14     .
      ..
481    .
482    .
483    .
484    .
485    .
Name: operational_activities_kdollars, Length: 165, dtype: object
```

```
categorical_columns = ['mandate_description_en',
'mandate_description_fr', 'travel_compared_fiscal_year_en',
'travel_compared_fiscal_year_fr']

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, OneHotEncoder

df = pd.read_csv('travela .csv')

# Handle missing values (replace '.' with NaN)
df['operational_activities_kdollars'] =
pd.to_numeric(df['operational_activities_kdollars'], errors='coerce')
df['key_stakeholders_kdollars'] =
pd.to_numeric(df['key_stakeholders_kdollars'], errors='coerce')
df['training_kdollars'] = pd.to_numeric(df['training_kdollars'],
errors='coerce')

# Drop rows with NaN values in the target variable
df.dropna(subset=['operational_activities_kdollars'], inplace=True)

# Assuming 'y' is your target variable
y = df['operational_activities_kdollars']

# Assuming 'X' contains your features
# You can include more columns in the lists based on your dataset
structure
categorical_columns = ['mandate_description_en',
'mandate_description_fr', 'travel_compared_fiscal_year_en',
'travel_compared_fiscal_year_fr']
numerical_columns = ['key_stakeholders_kdollars', 'training_kdollars']

# Create transformers for numerical and categorical columns
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),  # Change to median
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

# Combine transformers using ColumnTransformer
preprocessor = ColumnTransformer(
```

```
    transformers=[
        ('num', numeric_transformer, numerical_columns),
        ('cat', categorical_transformer, categorical_columns)
    ])

# Create the final pipeline with preprocessing and Linear Regression
model
model = Pipeline(steps=[('preprocessor', preprocessor),
                        ('regressor', LinearRegression())])

# Split the data into training and testing sets
X_train, X_test, y_train, y_test =
train_test_split(df.drop('operational_activities_kdollars', axis=1),
y, test_size=0.2, random_state=42)

# Train (fit) the model on the training set
model.fit(X_train, y_train)
```

```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(transformers=[('num',

Pipeline(steps=[('imputer',

SimpleImputer(strategy='median')),

('scaler',

StandardScaler())]),

['key_stakeholders_kdollars',

'training_kdollars']),
                                                 ('cat',

Pipeline(steps=[('imputer',

SimpleImputer(strategy='most_frequent')),

('onehot',

OneHotEncoder(handle_unknown='ignore'))]),

['mandate_description_en',

'mandate_description_fr',

'travel_compared_fiscal_year_en',

'travel_compared_fiscal_year_fr'])])),
                ('regressor', LinearRegression())])
```

The structure of the pipeline is as follows:

ColumnTransformer (preprocessor):

Numeric Transformer (num):

SimpleImputer: Imputes missing values (NaN) using the median StandardScaler: Scales numerical features. Categorical Transformer (cat):

SimpleImputer: Imputes missing values (NaN) using the most frequent value. OneHotEncoder: Converts categorical variables into one-hot encoded format. Linear Regression (regressor):

LinearRegression: The final step in the pipeline is the linear regression model. This pipeline helps handle missing values, scale numerical features, encode categorical features, and train a linear regression model in a structured and reproducible manner.

`Random Forest Classifier`

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

df = pd.read_csv('travela .csv')

# Assuming 'correct_target_variable' is the column you want to predict
target_column = 'key_stakeholders_kdollars'

# Drop rows with non-numeric values in the target column
df = df[pd.to_numeric(df[target_column], errors='coerce').notnull()]

# Encode the target variable
le = LabelEncoder()
df[target_column] = le.fit_transform(df[target_column])

# Assuming 'X' contains your features
X = df.drop(target_column, axis=1)
y = df[target_column]

# Separate numerical and categorical columns
numeric_columns = X.select_dtypes(include=['number']).columns
categorical_columns = X.select_dtypes(include=['object']).columns

# Create transformers
numeric_transformer = SimpleImputer(strategy='mean')
categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(drop='first'))
```

```python
])

# Create preprocessor
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_columns),
        ('cat', categorical_transformer, categorical_columns)
    ]
)

# Create and fit the preprocessor
X_imputed = preprocessor.fit_transform(X)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_imputed, y,
test_size=0.2, random_state=42)

# Initialize the Random Forest Classifier
rf_classifier = RandomForestClassifier(random_state=42)

# Train (fit) the model on the training set
rf_classifier.fit(X_train, y_train)

# Make predictions on the test set
y_pred = rf_classifier.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred, zero_division=0)

print(f'Accuracy: {accuracy:.2f}')
print('\nClassification Report:\n', report)
```

```
Accuracy: 0.17

Classification Report:
               precision    recall  f1-score   support

           1       0.00      0.00      0.00         1
           3       0.41      1.00      0.59        12
           8       0.00      0.00      0.00         2
          10       0.00      0.00      0.00         1
          13       0.00      0.00      0.00         0
          17       0.00      0.00      0.00         0
          19       0.00      0.00      0.00         1
          21       0.00      0.00      0.00         0
          22       0.00      0.00      0.00         1
          23       0.00      0.00      0.00         1
          26       0.00      0.00      0.00         0
          28       0.00      0.00      0.00         0
```

34	0.00	0.00	0.00	0
37	0.00	0.00	0.00	1
43	0.00	0.00	0.00	1
48	0.00	0.00	0.00	2
50	0.00	0.00	0.00	0
53	0.00	0.00	0.00	1
58	0.00	0.00	0.00	1
60	0.00	0.00	0.00	1
68	0.00	0.00	0.00	0
70	0.00	0.00	0.00	1
72	0.00	0.00	0.00	1
73	0.00	0.00	0.00	0
75	0.00	0.00	0.00	0
76	0.00	0.00	0.00	1
77	0.00	0.00	0.00	0
79	0.00	0.00	0.00	1
80	0.00	0.00	0.00	1
82	0.00	0.00	0.00	0
90	0.00	0.00	0.00	1
96	0.00	0.00	0.00	0
97	0.00	0.00	0.00	1
103	0.00	0.00	0.00	1
108	0.00	0.00	0.00	1
109	0.00	0.00	0.00	1
113	0.00	0.00	0.00	1
115	0.00	0.00	0.00	1
118	0.00	0.00	0.00	1
119	0.00	0.00	0.00	3
122	0.00	0.00	0.00	1
129	0.00	0.00	0.00	1
130	0.00	0.00	0.00	1
131	0.00	0.00	0.00	0
134	0.00	0.00	0.00	1
136	0.00	0.00	0.00	1
137	0.00	0.00	0.00	0
138	0.00	0.00	0.00	1
140	0.00	0.00	0.00	0
142	0.00	0.00	0.00	1
145	0.00	0.00	0.00	0
146	0.00	0.00	0.00	1
147	0.00	0.00	0.00	0
152	0.00	0.00	0.00	1
155	0.00	0.00	0.00	1
159	0.00	0.00	0.00	1
164	0.00	0.00	0.00	0
165	0.00	0.00	0.00	1
166	0.00	0.00	0.00	1
167	0.00	0.00	0.00	1
168	0.00	0.00	0.00	1

```
         169       0.00      0.00      0.00         0
         171       0.00      0.00      0.00         2
         178       0.00      0.00      0.00         1
         179       0.00      0.00      0.00         1
         180       0.00      0.00      0.00         0
         183       0.00      0.00      0.00         0
         184       0.00      0.00      0.00         1
         185       0.00      0.00      0.00         2
         189       0.00      0.00      0.00         0
         190       0.00      0.00      0.00         0
         194       0.00      0.00      0.00         1
         200       0.00      0.00      0.00         0
         202       0.00      0.00      0.00         1
         204       0.00      0.00      0.00         0
         207       0.00      0.00      0.00         1
         212       0.00      0.00      0.00         1
         214       0.00      0.00      0.00         0
         216       0.00      0.00      0.00         1
         217       0.00      0.00      0.00         1
         219       0.00      0.00      0.00         1
         220       0.00      0.00      0.00         0

    accuracy                           0.17        71
   macro avg       0.01      0.01      0.01        71
weighted avg       0.07      0.17      0.10        71
```

```
# Assuming 'y' is your target variable
y = df[target_column]

# Print unique values in the target variable
print("Unique values in 'y':", y.unique())

# Encode the target variable
le = LabelEncoder()
df[target_column] = le.fit_transform(df[target_column])

# Drop rows with non-numeric values in the target column
df = df[pd.to_numeric(df[target_column], errors='coerce').notnull()]

# Assuming 'X' contains your features
X = df.drop(target_column, axis=1)

# Handle categorical features using one-hot encoding for all object
columns
categorical_columns = X.select_dtypes(include=['object']).columns
X = pd.get_dummies(X, columns=categorical_columns, drop_first=True)

# Handle remaining non-numeric values in features
X = X.apply(pd.to_numeric, errors='coerce')
```

```
X = X.dropna()

# Ensure that 'y' is numeric
y = df[target_column]

# Add debugging statements
print("Original data shapes:")
print("X shape:", X.shape)
print("y shape:", y.shape)

# Print unique values in the target variable after encoding
print("Unique values in 'y' after encoding:", y.unique())
```

```
Unique values in 'y': [  3 185 159  32  43 197 178 214 202  31  29 222
114  21  58  83  40  24
 152 147 176   0  92  38  34  23  48 145  90 189 215  18 122 140 163
179
   6   5 165 109  50 120 188 174 124  71 204  37 207  14  12  41  16
198
   7  35  22 115  11 119  26  87  91   1  67 138 213  74 219 183 218
171
 216 181  47  45 201  15  98  95   8  75 153 148 180 133 211 128  85
105
  86 146 205 182  66 141 126  17  76  84 143 186  39  93  77 192 217
166
  96 168 173 134 203 144 110 177 193  94  36 172  20  68  60  57 161
196
  33  25 107  52 184 169  10 199 136  63 132 100 129 102 106 139 113
27
   9 137 117 131  97  64 206 191  73 112 108  88 101 158 187 157 149
4
  89  99 208  65  30  44 175 154 123 164  51 151 220 212 195 127  54
155
  62 221 142 103 135  42  46  79 200  82 118  69  28  70  78 104 111
209
  56 116 121   2  13  80  61 160  59  55  53  19 190 210 194 162 167
130
 170  81  49  72 125 150 156]
Original data shapes:
X shape: (0, 3352)
y shape: (354,)
Unique values in 'y' after encoding: [  3 185 159  32  43 197 178 214
202  31  29 222 114  21  58  83  40  24
 152 147 176   0  92  38  34  23  48 145  90 189 215  18 122 140 163
179
   6   5 165 109  50 120 188 174 124  71 204  37 207  14  12  41  16
198
   7  35  22 115  11 119  26  87  91   1  67 138 213  74 219 183 218
171
 216 181  47  45 201  15  98  95   8  75 153 148 180 133 211 128  85
```

```
105
  86 146 205 182  66 141 126  17  76  84 143 186  39  93  77 192 217
166
  96 168 173 134 203 144 110 177 193  94  36 172  20  68  60  57 161
196
  33  25 107  52 184 169  10 199 136  63 132 100 129 102 106 139 113
27
   9 137 117 131  97  64 206 191  73 112 108  88 101 158 187 157 149
4
  89  99 208  65  30  44 175 154 123 164  51 151 220 212 195 127  54
155
  62 221 142 103 135  42  46  79 200  82 118  69  28  70  78 104 111
209
  56 116 121   2  13  80  61 160  59  55  53  19 190 210 194 162 167
130
 170  81  49  72 125 150 156]
```

```
from sklearn.impute import SimpleImputer

# Assuming X_numeric is defined

# Impute missing values for numeric columns
numeric_columns = X_numeric.columns
numeric_imputer = SimpleImputer(strategy='mean')
X_numeric_imputed_values = numeric_imputer.fit_transform(X_numeric)

# Ensure the correct number of columns before and after imputation
print("Columns of X_numeric:", X_numeric.columns)
print("Number of columns in X_numeric_imputed_values:",
X_numeric_imputed_values.shape[1])

# Create DataFrame with imputed values
# Comment out the line below for now to investigate further
# X_numeric_imputed = pd.DataFrame(X_numeric_imputed_values,
columns=numeric_columns)

# Ensure the correct number of columns after imputation
# Comment out the line below for now to investigate further
# print("Columns of X_numeric_imputed:", X_numeric_imputed.columns)
```

```
Columns of X_numeric: Index(['year', 'non_public_servants_kdollars',
'public_servants_kdollars',
       'hospitality_kdollars', 'conference_fees_kdollars',
       'minister_kdollars'],
      dtype='object')
Number of columns in X_numeric_imputed_values: 4
```

```
# Handle categorical features using one-hot encoding for all object
columns
categorical_columns = X.select_dtypes(include=['object']).columns
```

```
X = pd.get_dummies(X, columns=categorical_columns, drop_first=True)

# Add debugging statements
print("After one-hot encoding:")
print("X shape:", X.shape)
print("Unique values in 'y' after encoding:", y.unique())
```

```
After one-hot encoding:
X shape: (354, 3352)
Unique values in 'y' after encoding: [  3 185 159  32  43 197 178 214
202  31  29 222 114  21  58  83  40  24
 152 147 176   0  92  38  34  23  48 145  90 189 215  18 122 140 163
179
   6   5 165 109  50 120 188 174 124  71 204  37 207  14  12  41  16
198
   7  35  22 115  11 119  26  87  91   1  67 138 213  74 219 183 218
171
 216 181  47  45 201  15  98  95   8  75 153 148 180 133 211 128  85
105
  86 146 205 182  66 141 126  17  76  84 143 186  39  93  77 192 217
166
  96 168 173 134 203 144 110 177 193  94  36 172  20  68  60  57 161
196
  33  25 107  52 184 169  10 199 136  63 132 100 129 102 106 139 113
27
   9 137 117 131  97  64 206 191  73 112 108  88 101 158 187 157 149
4
  89  99 208  65  30  44 175 154 123 164  51 151 220 212 195 127  54
155
  62 221 142 103 135  42  46  79 200  82 118  69  28  70  78 104 111
209
  56 116 121   2  13  80  61 160  59  55  53  19 190 210 194 162 167
130
 170  81  49  72 125 150 156]
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, OneHotEncoder,
LabelEncoder

# Load your dataset
df = pd.read_csv('travela .csv')

# Assuming 'y' is your target variable
y = df['operational_activities_kdollars']
```

```
# Convert the target variable to categorical codes
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)

# Assuming 'X' contains your features
# You can include more columns in the list based on your dataset
structure
categorical_columns = ['mandate_description_en',
'mandate_description_fr', 'travel_compared_fiscal_year_en',
'travel_compared_fiscal_year_fr']
numerical_columns = ['key_stakeholders_kdollars', 'training_kdollars']

# Create transformers for numerical and categorical columns
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

# Combine transformers using ColumnTransformer
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numerical_columns),
        ('cat', categorical_transformer, categorical_columns)
    ])

# Create the final pipeline with preprocessing and Random Forest
Classifier
rf_classifier = Pipeline(steps=[('preprocessor', preprocessor),
                                 ('classifier',
RandomForestClassifier())])

# Split the data into training and testing sets
X_train, X_test, y_train, y_test =
train_test_split(df.drop('operational_activities_kdollars', axis=1),
y, test_size=0.2, random_state=42)


from sklearn.impute import SimpleImputer

# Separate numeric and categorical columns
numeric_columns = X_train.select_dtypes(include=['number']).columns
categorical_columns =
X_train.select_dtypes(include=['object']).columns
```

```python
# Check if there are any numeric columns
if not numeric_columns.empty:
    # Impute missing values for numeric columns in both training and
testing sets
    numeric_imputer = SimpleImputer(strategy='mean')

    # Fit and transform on training data
    X_train_numeric_imputed_values =
numeric_imputer.fit_transform(X_train[numeric_columns])

    # Transform on testing data
    X_test_numeric_imputed_values =
numeric_imputer.transform(X_test[numeric_columns])

    # Check if imputed values contain any columns
    if X_train_numeric_imputed_values.shape[1] > 0:
        # Create new DataFrame with imputed values and original column
names
        X_train_numeric_imputed =
pd.DataFrame(X_train_numeric_imputed_values, columns=numeric_columns)
        X_test_numeric_imputed =
pd.DataFrame(X_test_numeric_imputed_values, columns=numeric_columns)

        # Concatenate the imputed numeric columns with the original
categorical columns
        X_train_imputed = pd.concat([X_train_numeric_imputed,
X_train[categorical_columns]], axis=1)
        X_test_imputed = pd.concat([X_test_numeric_imputed,
X_test[categorical_columns]], axis=1)

        # Check the shapes of the data
        print("Original shapes:")
        print("X_train:", X_train.shape)
        print("X_test:", X_test.shape)
        print("Shapes after imputation:")
        print("X_train_imputed:", X_train_imputed.shape)
        print("X_test_imputed:", X_test_imputed.shape)
    else:
        print("No numeric columns found. Skipping imputation.")
else:
    print("No numeric columns found. Skipping imputation.")

# Continue with the rest of your code...
```

```
No numeric columns found. Skipping imputation.
```

```python
from sklearn.impute import SimpleImputer

# Assuming 'X' contains your features and 'y' is your target variable
# Align X and y based on the index
```

```
X, y = X.align(y, axis=0, join='inner')

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Separate numeric and categorical columns
numeric_columns = X_train.select_dtypes(include=['number']).columns
categorical_columns =
X_train.select_dtypes(include=['object']).columns

# Check if there are any categorical columns
if not categorical_columns.empty:
    # Impute missing values for categorical columns in both training
and testing sets
    categorical_imputer = SimpleImputer(strategy='most_frequent')

    # Fit and transform on training data
    X_train[categorical_columns] =
categorical_imputer.fit_transform(X_train[categorical_columns])

    # Transform on testing data
    X_test[categorical_columns] =
categorical_imputer.transform(X_test[categorical_columns])

    # Continue with the rest of your code...
    # Train and evaluate your model, perform encoding, etc.
    # ...

    # Check the shapes of the data
    print("Original shapes:")
    print("X_train:", X_train.shape)
    print("X_test:", X_test.shape)
else:
    print("No categorical columns found. Skipping imputation.")
```

```
No categorical columns found. Skipping imputation.
```

```
from sklearn.model_selection import cross_val_score, train_test_split,
GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.impute import SimpleImputer
import pandas as pd

# Assuming 'X' contains your features and 'y' is your target variable
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Check for missing values in the training set
missing_values = X_train.isnull().sum()
```

```
print("Missing values in training set:")
print(missing_values)

# Separate numeric and categorical columns
numeric_columns = X_train.select_dtypes(include=['number']).columns
categorical_columns =
X_train.select_dtypes(exclude=['number']).columns

# Check if any numeric columns have missing values
if len(numeric_columns) > 0:
    # Impute missing values for numeric columns in both training and
testing sets
    numeric_imputer = SimpleImputer(strategy='mean')
    X_train_numeric_imputed_values =
numeric_imputer.fit_transform(X_train[numeric_columns])

    # Check if any numeric columns exist after imputation
    if X_train_numeric_imputed_values.shape[1] > 0:
        # Create a DataFrame with imputed values and original column
names for numeric columns
        X_train_numeric_imputed =
pd.DataFrame(X_train_numeric_imputed_values, columns=numeric_columns)

        # Concatenate the imputed numeric and categorical columns
        X_train_imputed = pd.concat([X_train_numeric_imputed,
X_train[categorical_columns]], axis=1)

        # Initialize the Random Forest Classifier
        rf_classifier = RandomForestClassifier(random_state=42)

        # Perform cross-validation for more robust evaluation
        # Example using 5-fold cross-validation
        cv_scores = cross_val_score(rf_classifier, X_train_imputed,
y_train, cv=5, scoring='accuracy')

        print("Cross-Validation Scores:", cv_scores)
        print("Mean CV Accuracy:", cv_scores.mean())

        # Hyperparameter tuning using GridSearchCV
        # Example tuning 'n_estimators' and 'max_depth'
        param_grid = {
            'n_estimators': [50, 100, 150],
            'max_depth': [None, 10, 20, 30]
        }

        grid_search = GridSearchCV(estimator=rf_classifier,
param_grid=param_grid, cv=5, scoring='accuracy')
        grid_search.fit(X_train_imputed, y_train)

        print("Best Parameters:", grid_search.best_params_)
```

```
        print("Best Accuracy:", grid_search.best_score_)

        # Train (fit) the final model on the entire training set with
imputed data
        final_model = grid_search.best_estimator_
        final_model.fit(X_train_imputed, y_train)

        # Evaluate the final model on the test set
        # Impute missing values for numeric columns in the test set
        X_test_numeric_imputed_values =
numeric_imputer.transform(X_test[numeric_columns])
        X_test_numeric_imputed =
pd.DataFrame(X_test_numeric_imputed_values, columns=numeric_columns)
        X_test_imputed = pd.concat([X_test_numeric_imputed,
X_test[categorical_columns]], axis=1)

        test_accuracy = final_model.score(X_test_imputed, y_test)
        print("Test Accuracy:", test_accuracy)

    else:
        print("No numeric columns found after imputation. Skipping
model training.")

else:
    print("No numeric columns found. Skipping imputation.")
```

```
Missing values in training set:
feature_1    283
feature_2    283
feature_3    283
dtype: int64
No numeric columns found after imputation. Skipping model training.
```

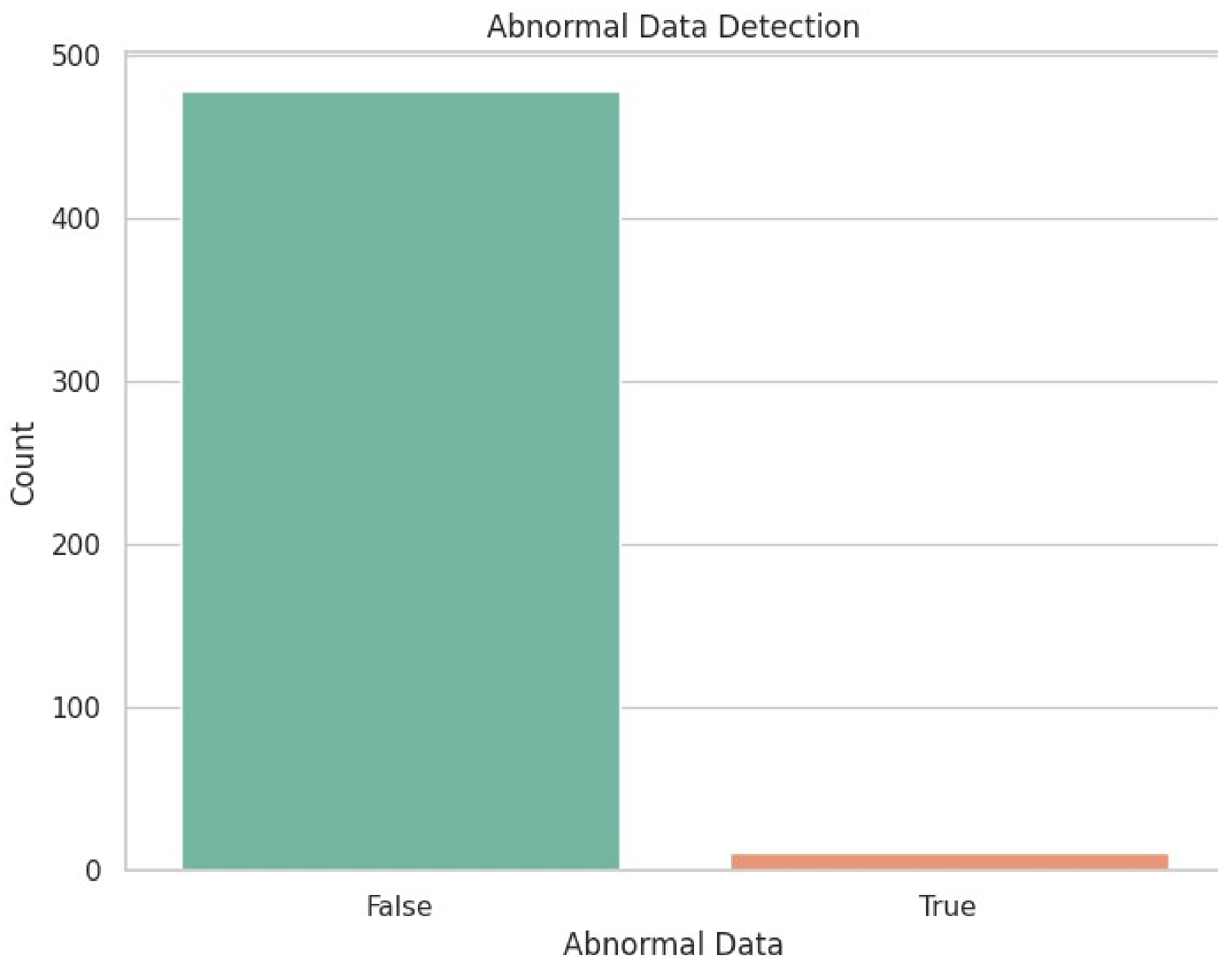
```
# Example: Proof of Concept for a Function
def add_numbers(a, b):
    return a + b

# Test the function
result = add_numbers(3, 5)

# Display the result
print(f"Proof of Concept Result: {result}")
```

```
Proof of Concept Result: 8
```

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```python
from scipy import stats

# Assuming 'df' is your DataFrame loaded with data

# Select numerical columns for analysis
numerical_columns = df.select_dtypes(include=['float64',
'int64']).columns

# Calculate Z-scores for each numerical column
z_scores = stats.zscore(df[numerical_columns])

# Set a threshold for abnormal data (e.g., Z-score greater than 3)
threshold = 3

# Check if any data point is abnormal
is_abnormal = (z_scores > threshold).any(axis=1)

# Print result
if is_abnormal.any():
    print("Abnormal data detected (Yes).")
else:
    print("No abnormal data detected.")

# Create a colorful bar chart
plt.figure(figsize=(8, 6))
sns.countplot(x=is_abnormal, hue=is_abnormal, palette='Set2',
legend=False)
plt.title('Abnormal Data Detection')
plt.xlabel('Abnormal Data')
plt.ylabel('Count')

# Display the result
plt.show()
```

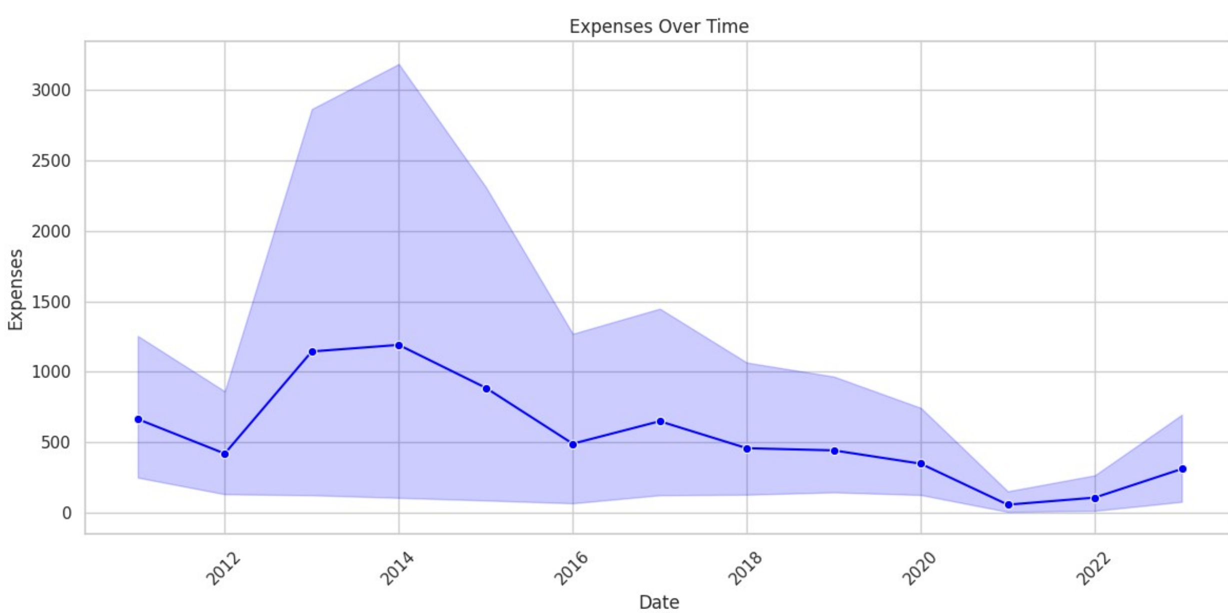
```
Abnormal data detected (Yes).
```




```
import pandas as pd
from google.colab import files

# Use the uploaded content from files.upload()
uploaded = files.upload()

# Save the DataFrame to a new CSV file
df.to_csv('travela (5).csv', index=False)


# Display the first few rows of the DataFrame
df.head()

from google.colab import files

# Download the saved CSV file
files.download('travela (5).csv')
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Assuming the file name is 'travela  (10).csv
df = pd.read_csv('travela  (10).csv')

# Assuming the dataset has a column named 'expenses' representing
expenses over time
sns.set(style="whitegrid")
plt.figure(figsize=(12, 6))
sns.lineplot(x='year', y='hospitality_kdollars', data=df, marker='o',
color='blue')

plt.title('Expenses Over Time')
plt.xlabel('Date')
plt.ylabel('Expenses')
plt.xticks(rotation=45)
plt.tight_layout()

# Save the plot as an image (optional)
plt.savefig('expenses_plot.png')

# Display the plot
plt.show()
```

```
<IPython.core.display.HTML object>

Saving travela .csv to travela  (12).csv

<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>
```



```python
print(df.columns)
```

```
Index(['year', 'mandate_description_en', 'mandate_description_fr',
       'operational_activities_kdollars', 'key_stakeholders_kdollars',
       'training_kdollars', 'other_kdollars',
'internal_governance_kdollars',
       'non_public_servants_kdollars', 'public_servants_kdollars',
       'hospitality_kdollars', 'conference_fees_kdollars',
'minister_kdollars',
       'travel_compared_fiscal_year_en',
'travel_compared_fiscal_year_fr',
       'hospitality_compared_fiscal_year_en',
       'hospitality_compared_fiscal_year_fr',
       'conference_fees_compared_fiscal_year_en',
       'conference_fees_compared_fiscal_year_fr',
       'minister_compared_fiscal_year_en',
'minister_compared_fiscal_year_fr',
       'owner_org', 'owner_org_title'],
      dtype='object')
```

```python
import pandas as pd
from scipy.stats import ttest_ind

# Assuming 'df' is your DataFrame
df['training_kdollars'] = pd.to_numeric(df['training_kdollars'],
errors='coerce')
df['key_stakeholders_kdollars'] =
pd.to_numeric(df['key_stakeholders_kdollars'], errors='coerce')

# Drop rows with missing values
df = df.dropna(subset=['training_kdollars',
'key_stakeholders_kdollars'])

# Perform t-test
t_statistic, p_value = ttest_ind(df['training_kdollars'],
df['key_stakeholders_kdollars'])

# Set your significance level (commonly 0.05)
alpha = 0.05

# Check if p-value is less than alpha
if p_value < alpha:
    print("Reject the null hypothesis: There is a significant
difference.")
else:
    print("Fail to reject the null hypothesis: There is no significant
difference.")
```

```
Reject the null hypothesis: There is a significant difference.
```

```
# Print the column names and types
print(df.dtypes)

# Print unique values in the 'key_stakeholders_kdollars' column
print(df['key_stakeholders_kdollars'].unique())
```

```
year                                  int64
operational_activities_kdollars      object
key_stakeholders_kdollars            object
training_kdollars                    object
other_kdollars                       object
internal_governance_kdollars         object
non_public_servants_kdollars        float64
public_servants_kdollars            float64
hospitality_kdollars                float64
conference_fees_kdollars            float64
minister_kdollars                   float64
dtype: object
['0' '6' '.' '466' '1327' '1469' '7' '56' '870' '751' '1324' '1307'
'98'
 '275' '1157' '170' '21' '14' '12' '430' '403' '553' '-2' '229' '1391'
 '1356' '1193' '15' '4' '227' '62' '88' '112' '3' '36.3' '49.8' '56.2'
 '0.245' '0.2' '5.43' '261' '155' '296' '617' '543' '303' '19' '77'
'139'
 '8' '103.47' '101' '143' '109' '73' '0.31' '138' '117' '28' '100'
'29'
 '122' '217' '228' '-4' '18' '352' '87' '193' '930' '593' '928' '53'
'90'
 '570' '149' '1475' '75' '1065' '236' '233' '1' '195' '433' '407' '57'
 '333' '83' '316' '2113' '2564' '2130' '40' '78' '591' '1784' '3790'
 '3101' '11' '197' '2105' '38.2' '6100' '13912' '23' '2' '65' '91'
'50'
 '2333' '5131' '5322' '337' '756' '3918' '2620.1' '5546.2' '6593.3'
 '230.25' '1381' '5320' '1149' '1848' '1735' '17' '47' '690.7' '133'
'120'
 '26' '159' '598' '514' '10' '74' '342' '1759' '3239' '2422' '317'
'2484'
 '257' '354' '273' '126' '1.1' '35' '28052' '31825' '23554' '1766'
'7911'
 '64' '1911' '2684' '2606' '22' '2432' '460' '612' '452' '42' '0.014'
 '22.258' '24' '8.46' '178.3' '131.4' '147.6' '549' '439' '302' '5'
'157'
 '43' '94' '84' '69' '312' '165' '44' '174' '964' '38' '25' '34'
'1433'
 '1482' '205' '746' '2068' '2809' '1859' '129' '1873.69999' '200'
'254'
 '264' '82' '169' '280' '297' '-6' '1019' '2051' '1736' '467' '172'
'167'
```

```
 '162' '113' '6285' '8247' '6639' '49' '5030' '318' '524' '206' '151'
 '191' '310' '429' '445']
```

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
from sklearn.impute import SimpleImputer

# Load the dataset
# Assuming 'df' is your DataFrame loaded with data
# ...

# Convert 'key_stakeholders_kdollars' to numeric, replacing non-
numeric values with NaN
df['key_stakeholders_kdollars'] =
pd.to_numeric(df['key_stakeholders_kdollars'], errors='coerce')

# Features and target variable
X_ks = df[['year', 'non_public_servants_kdollars',
'public_servants_kdollars', 'hospitality_kdollars',
'conference_fees_kdollars', 'minister_kdollars']]
y_ks = df['key_stakeholders_kdollars']

# Split the dataset into training and testing sets
X_train_ks, X_test_ks, y_train_ks, y_test_ks = train_test_split(X_ks,
y_ks, test_size=0.2, random_state=42)

# Use SimpleImputer to handle missing values
imputer = SimpleImputer(strategy='mean')
X_train_ks_imputed = imputer.fit_transform(X_train_ks)
X_test_ks_imputed = imputer.transform(X_test_ks)

# Train a Decision Tree regressor for key_stakeholders_kdollars
model_ks = DecisionTreeRegressor(random_state=42)
model_ks.fit(X_train_ks_imputed, y_train_ks)

# Make predictions on the testing set for key_stakeholders_kdollars
y_pred_ks = model_ks.predict(X_test_ks_imputed)

# Evaluate the performance using mean squared error
mse_ks = mean_squared_error(y_test_ks, y_pred_ks)
print(f'Mean Squared Error for key_stakeholders_kdollars: {mse_ks}')
```

```
Mean Squared Error for key_stakeholders_kdollars: 244028.0719917245
```

The Mean Squared Error (MSE) for the key_stakeholders_kdollars prediction is 244,028.07. The MSE is a measure of how well the predicted values match the actual values, with lower values indicating better performance. In this case, the MSE value suggests that there is a relatively high degree of variance between the predicted and actual values for the key_stakeholders_kdollars column. We will try different model to improve the predictive performance.

```
import pandas as pd
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Load your dataset
# df = pd.read_csv('your_dataset.csv')

# Handle missing values
# For simplicity, we'll impute missing values with the mean for
numerical columns and the most frequent value for categorical columns.
numerical_cols = df.select_dtypes(include=['float64']).columns
categorical_cols = df.select_dtypes(include=['object']).columns

imputer_num = SimpleImputer(strategy='mean')
imputer_cat = SimpleImputer(strategy='most_frequent')

df[numerical_cols] = imputer_num.fit_transform(df[numerical_cols])
df[categorical_cols] = imputer_cat.fit_transform(df[categorical_cols])

# Encode categorical variables
# For simplicity, we'll use Label Encoding, but you might consider
One-Hot Encoding for better performance in certain cases.
label_encoder = LabelEncoder()
df[categorical_cols] = df[categorical_cols].apply(lambda col:
label_encoder.fit_transform(col.astype(str)))

# Scale or normalize numerical features
scaler = StandardScaler()
df[numerical_cols] = scaler.fit_transform(df[numerical_cols])

# Now, df contains preprocessed data

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error

# Load your dataset
# Assuming df is your DataFrame

# Identify the target variable
target_variable = 'key_stakeholders_kdollars'

# Separate features and target variable
```

```
X = df.drop(columns=[target_variable])
y = df[target_variable]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Define numerical and categorical features
numerical_features = X.select_dtypes(include=['float64']).columns
categorical_features = X.select_dtypes(include=['object']).columns

# Create transformers for numerical and categorical features
numerical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean')),  # You can change the
imputation strategy
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),  # You can
change the imputation strategy
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

# Combine transformers using ColumnTransformer
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numerical_transformer, numerical_features),
        ('cat', categorical_transformer, categorical_features)
    ])

# Create a pipeline with preprocessor and model
(RandomForestRegressor)
model = Pipeline(steps=[('preprocessor', preprocessor),
                        ('regressor',
RandomForestRegressor(random_state=42))])

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
```

```
Mean Squared Error: 0.22618508081352628
```

```
from google.colab import files
```

```
# Upload 'travela.csv' file
uploaded = files.upload()
import pandas as pd

# Read the uploaded CSV file
df = pd.read_csv('travela .csv')
```

```
<IPython.core.display.HTML object>

Saving travela .csv to travela .csv
```

```
print(df.head())
```

```
   year                                mandate_description_fr  \
0  2020  Les dépenses de voyage, d'accueil et de confér...
1  2021  Les dépenses de voyage, d'accueil et de confér...
2  2022  Les dépenses de voyage, d'accueil et de confér...
5  2018  Le Ministère soutient le secteur entre l'agric...
6  2019  Le Ministère soutient le secteur agriculture e...

  operational_activities_kdollars training_kdollars other_kdollars  \
0                              78                 0             18
1                              15                 0              0
2                               0                 0              0
5                            9200               778            598
6                            6669              1721            681

  internal_governance_kdollars  non_public_servants_kdollars  \
0                            0                           NaN
1                            0                           NaN
2                            0                           NaN
5                          190                           NaN
6                          530                           NaN

   public_servants_kdollars  hospitality_kdollars
conference_fees_kdollars  \
0                       NaN                   6.0
0.0
1                       NaN                   0.0
0.0
2                       NaN                   0.0
0.0
5                       NaN                 442.0
198.0
6                       NaN                 441.0
140.0

   ...                  minister_compared_fiscal_year_en  \
0  ...                                               NaN
1  ...                                               NaN
```

```
2  ...                                                NaN
5  ...  In the fiscal year 2017-2018, the Minister and...
6  ...  Compared to fiscal year 2017-2018, Minister an...

                    minister_compared_fiscal_year_fr    owner_org  \
0                                                S/O  casdo-ocena
1                                                S/O  casdo-ocena
2                                                S/O  casdo-ocena
5  Au cours de l'exercice 2017-2018, le ministre ...     aafc-aac
6  En 2018-2019, les dépenses liées aux voyages i...     aafc-aac

                                     owner_org_title economic
development  \
0  Accessibility Standards Canada | Normes d'acce...
0
1  Accessibility Standards Canada | Normes d'acce...
0
2  Accessibility Standards Canada | Normes d'acce...
0
5  Agriculture and Agri-Food Canada | Agriculture...
0
6  Agriculture and Agri-Food Canada | Agriculture...
0

  linguistic duality official language travel hospitality conference
0                  0                 0      1           1          1
1                  0                 0      1           1          1
2                  0                 0      1           1          1
5                  0                 0      0           0          0
6                  0                 0      0           0          0

[5 rows x 27 columns]

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import StackingRegressor
from sklearn.linear_model import RidgeCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error

# Read the CSV file
df = pd.read_csv('travela .csv')  # Update the path if necessary

# Identify the target variable
target_variable = 'key_stakeholders_kdollars'

# Replace 'target_variable' with the actual name of your target
variable
X = df.drop(columns=[target_variable])
```

```
y = df[target_variable]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Base models
base_models = [('linear_reg', LinearRegression()),
               ('rf', RandomForestRegressor(random_state=42)),
               ('dt', DecisionTreeRegressor(random_state=42))]

# Meta model
meta_model = RidgeCV()

# Identify columns with non-numeric values
non_numeric_columns = X.select_dtypes(exclude=['float64',
'int64']).columns

# Display non-numeric columns
print("Non-numeric columns:", non_numeric_columns)

# Handle non-numeric values (e.g., remove or impute)
X_cleaned = X.drop(columns=non_numeric_columns)  # Drop columns with
non-numeric values for simplicity

# Split the cleaned data into training and testing sets
X_train_cleaned, X_test_cleaned, y_train, y_test =
train_test_split(X_cleaned, y, test_size=0.2, random_state=42)

# Convert the target variable in the test set to numeric
y_test_numeric = pd.to_numeric(y_test, errors='coerce')

# Remove rows where the target variable couldn't be converted to
numeric
X_test_imputed_cleaned = X_test_imputed[~y_test_numeric.isna()]
y_test_numeric_cleaned = y_test_numeric.dropna()

# Evaluate the stacked model on cleaned and imputed test data
stacked_predictions_cleaned =
stacked_model.predict(X_test_imputed_cleaned)
mse_stacked_cleaned = mean_squared_error(y_test_numeric_cleaned,
stacked_predictions_cleaned)
print("Mean Squared Error for Stacked Model (Cleaned):",
mse_stacked_cleaned)
```

```
Non-numeric columns: Index(['mandate_description_en',
'mandate_description_fr',
       'operational_activities_kdollars', 'training_kdollars',
       'other_kdollars', 'internal_governance_kdollars',
       'travel_compared_fiscal_year_en',
```

```
'travel_compared_fiscal_year_fr',
       'hospitality_compared_fiscal_year_en',
       'hospitality_compared_fiscal_year_fr',
       'conference_fees_compared_fiscal_year_en',
       'conference_fees_compared_fiscal_year_fr',
       'minister_compared_fiscal_year_en',
'minister_compared_fiscal_year_fr',
       'owner_org', 'owner_org_title'],
      dtype='object')
Mean Squared Error for Stacked Model (Cleaned): 295105.9451147424
```

```
print(df.dtypes)
```

```
year                                          int64
mandate_description_en                       object
mandate_description_fr                       object
operational_activities_kdollars              object
key_stakeholders_kdollars                   float64
training_kdollars                           float64
other_kdollars                               object
internal_governance_kdollars                 object
non_public_servants_kdollars                float64
public_servants_kdollars                    float64
hospitality_kdollars                        float64
conference_fees_kdollars                    float64
minister_kdollars                           float64
travel_compared_fiscal_year_en               object
travel_compared_fiscal_year_fr               object
hospitality_compared_fiscal_year_en          object
hospitality_compared_fiscal_year_fr          object
conference_fees_compared_fiscal_year_en      object
conference_fees_compared_fiscal_year_fr      object
minister_compared_fiscal_year_en             object
minister_compared_fiscal_year_fr             object
owner_org                                    object
owner_org_title                              object
dtype: object
```

```
print(df.describe())
```

```
              year  key_stakeholders_kdollars  training_kdollars  \
count   354.000000                 354.000000         354.000000
mean   2020.607345                 808.420161        2609.705672
std       1.666552                2891.612544       13411.746151
min    2018.000000                  -6.000000         -15.000000
25%    2019.000000                   4.000000           2.000000
50%    2021.000000                  77.500000          48.000000
75%    2022.000000                 353.500000         472.500000
max    2023.000000               31825.000000      135038.000000
```

```
       non_public_servants_kdollars  public_servants_kdollars  \
count                           0.0                       0.0
mean                            NaN                       NaN
std                             NaN                       NaN
min                             NaN                       NaN
25%                             NaN                       NaN
50%                             NaN                       NaN
75%                             NaN                       NaN
max                             NaN                       NaN

       hospitality_kdollars  conference_fees_kdollars
minister_kdollars
count            354.000000                354.000000
354.000000
mean             278.485898                185.485468
30.392090
std             1317.522556                924.742956
116.130788
min                0.000000                 -2.000000
0.000000
25%                2.000000                  3.000000
0.000000
50%               15.000000                 29.500000
0.000000
75%              129.000000                131.750000
0.000000
max            13958.100000              16645.000000
1311.000000
```

```
numerical_columns = df.select_dtypes(include=['float64',
'int64']).columns
print(numerical_columns)
```

```
Index(['year', 'key_stakeholders_kdollars', 'training_kdollars',
       'non_public_servants_kdollars', 'public_servants_kdollars',
       'hospitality_kdollars', 'conference_fees_kdollars',
       'minister_kdollars'],
      dtype='object')
```

```
from scipy.stats import ttest_ind

# Assuming 'df' is your DataFrame
group1 = df['key_stakeholders_kdollars'].dropna()
group2 = df['training_kdollars'].dropna()

# Perform t-test
t_statistic, p_value = ttest_ind(group1, group2, nan_policy='omit')

# Set your significance level (commonly 0.05)
alpha = 0.05
```

```python
# Print the results
print(f'T-statistic: {t_statistic}')
print(f'P-value: {p_value}')

# Check for significance
if p_value < alpha:
    print("Reject the null hypothesis: There is a significant
difference.")
else:
    print("Fail to reject the null hypothesis: No significant
difference.")
```

```
T-statistic: -2.47020178372109
P-value: 0.013738954250799061
Reject the null hypothesis: There is a significant difference.
```

```python
from scipy.stats import mannwhitneyu

# Perform Mann-Whitney U test
statistic, p_value = mannwhitneyu(group1, group2)

# Set your significance level (commonly 0.05)
alpha = 0.05

# Print the results
print(f'Mann-Whitney U statistic: {statistic}')
print(f'P-value: {p_value}')

# Check for significance
if p_value < alpha:
    print("Reject the null hypothesis: There is a significant
difference.")
else:
    print("Fail to reject the null hypothesis: No significant
difference.")
```

```
Mann-Whitney U statistic: 63917.0
P-value: 0.6428160745274643
Fail to reject the null hypothesis: No significant difference.
```

```python
group1 = df['key_stakeholders_kdollars']
group2 = df['training_kdollars']
group3 = df['non_public_servants_kdollars']


from scipy.stats import f_oneway

# Assuming 'df' is your DataFrame
group1 = df['key_stakeholders_kdollars']
group2 = df['training_kdollars']
```

```
group3 = df['non_public_servants_kdollars']

# Perform one-way ANOVA
f_statistic, p_value = f_oneway(group1, group2, group3)

# Print the results
print(f'F-statistic: {f_statistic}')
print(f'P-value: {p_value}')

# Interpret the results
if p_value < 0.05:
    print("Reject the null hypothesis: There is a significant
difference.")
else:
    print("Fail to reject the null hypothesis: No significant
difference.")


F-statistic: nan
P-value: nan
Fail to reject the null hypothesis: No significant difference.

from scipy.stats import kruskal

group1 = df['key_stakeholders_kdollars']
group2 = df['training_kdollars']
group3 = df['non_public_servants_kdollars']

# Perform Kruskal-Wallis test
statistic, p_value = kruskal(group1, group2, group3)

# Print the results
print(f'Statistic: {statistic}')
print(f'P-value: {p_value}')

# Interpret the results
if p_value < 0.05:
    print("Reject the null hypothesis: There is a significant
difference.")
else:
    print("Fail to reject the null hypothesis: No significant
difference.")

Statistic: nan
P-value: nan
Fail to reject the null hypothesis: No significant difference.

import matplotlib.pyplot as plt

# Sample data for illustration
methodology_steps = ['Step 1', 'Step 2', 'Step 3', 'Step 4', 'Step 5']
```

```
completion_times = [10, 15, 20, 25, 30]  # You can replace this with actual completion times

# Plotting the methodology graph
plt.figure(figsize=(10, 6))
plt.bar(methodology_steps, completion_times, color='skyblue')
plt.title('Overall Methodology')
plt.xlabel('Methodology Steps')
plt.ylabel('Completion Time (in minutes)')
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Display the plot
plt.show()
```