Sahan Chowdhury Data Science project -

Investigating historical trends and future projections of UK GHG emissions

In [408…
```
#Importing required Libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
import numpy as np
import matplotlib.cm as cm
import seaborn as sns
from scipy.stats import zscore
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.svm import SVR
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

In [409…
```
#Dataset Link
#Reference[1]: www.ons.gov.uk, n.d.
#https://www.ons.gov.uk/economy/environmentalaccounts/datasets/ukenvironmentalaccountsatmosphericemissionsgreen
```

In [410…
```
#Importing the xlsx dataset and the correct sheet

dataset1 = 'provisionalatmoshpericemissionsghg.xlsx'
sheet_name = 2
#Sheet 2 refers to sheet 3 in the excel sheet
```

In [411…
```
#The sheet contains 2 tables
#First importing the first table from rows 7 to 42 and columsn A to Y
#Reference[2]: Suma Katabattuni, 2024
table1 = pd.read_excel(dataset1,sheet_name,header=0, usecols="A:Y",  skiprows=6,  nrows=36  )

# Display the full table
table1
```

Out[411]:

	SIC code	A	B	C	D	E	F	G	H	I	...	C
0	Industry name	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning s...	Water supply; sewerage, waste management and r...	Construction	Wholesale and retail trade; repair of motor ve...	Transport and storage	Accommodation and food services	...	Public administration and defence compulsory ..
1	1990	55690.3	50452.8	175212.7	217602.2	75316.3	8201.2	11560.8	66985.3	3015.3	...	12058.4
2	1991	55492.6	50608.3	176254.4	214317.5	76852.7	8072	11984.1	67438.6	3453.5	...	11381.2
3	1992	55149	51199.3	169388.2	202469.2	76959.1	8188.3	11789.9	68438.5	3163.4	...	11523.4
4	1993	54319.9	50807.1	164365	184997.2	77275.6	8275.1	12032.9	69735.2	3371.4	...	11710.9
5	1994	55129.7	44588.4	167285.9	180721.6	77637.9	8633.8	12403.1	70659.2	3342.3	...	11071.3
6	1995	55061.9	46451.4	165344.1	178162.0	78712	8695.3	12560.6	73508.2	3450.7	...	11079.5
7	1996	56343.6	46633.6	166622.3	177960.6	79209.4	8923.8	12825.8	79135.4	3636.1	...	10890.1
8	1997	55810	45761.2	165258.4	164554.2	77982.7	9263.6	12631.7	88159.2	3237.9	...	10462.7
9	1998	55950.8	44399.3	155653.0	170815.5	78691.1	9207.7	12926.4	93464.0	3581	...	10028.0
10	1999	56176.1	40501.6	140696.8	163207.5	73742.9	9246.5	13464.5	91941.5	3819.3	...	10260.4
11	2000	54133.6	38335.6	135627.4	176005.8	71060.5	9440.5	13774.1	94607.7	3913.4	...	10046.3
12	2001	51955.6	38210.6	128931.3	186969.6	69110.6	9566.7	13738.1	98397.8	4077.2	...	9921.5
13	2002	51495.5	36722.5	121456.3	182100.7	67450.9	9608.6	13544.7	95135.4	3457.7	...	9629.5
14	2003	51950.4	34092.4	124723.3	191163.6	63375.2	9791.5	14202.6	95101.3	3705	...	9518.5
15	2004	52813.4	32909.3	122868.9	191097.4	58712.5	10170.6	14461.8	97343.8	3673	...	9605.4
16	2005	52934	30738.2	121871.7	192574.6	56104.9	10152	14753.1	99137.9	3694.7	...	8493.2
17	2006	51899.8	28291.7	117269.7	200397.5	52739.5	10309.7	15013	90909.2	3518.8	...	8486.5
18	2007	51473.1	27770.4	116409.3	195947.3	49262.6	10785.7	15656.8	94883.2	3433.4	...	8360.2
19	2008	51240.7	27088.3	108134.2	191548.8	43934.9	10784.4	16297.4	90840.5	4178.1	...	7647
20	2009	50442.8	27056.8	90639.6	169738.9	39631	9504.6	16483.2	84039.5	3525.6	...	6938.5
21	2010	50502.1	26661.9	93988	177018.4	34835.1	9682.6	17024.2	84360.1	3773.8	...	6907.7
22	2011	49647.4	24205.7	90258	164156.6	32853.5	9235.3	16767.9	87918.1	3403.6	...	6299.3
23	2012	50368.8	22379.2	86759.9	178480.3	31469.9	9426.8	16946	84115	3794.9	...	6217.2
24	2013	49299.1	21096.9	89442.9	169292	28566.7	9033.9	16508.8	80876.9	3896.3	...	5696.7
25	2014	51149.2	21086.9	88797.2	148347.2	26627.8	9236	15865.5	85705.9	3444	...	5057
26	2015	51774.5	22402.3	86036.2	131922.4	26416.7	9971.5	15912.8	87193.9	4289.6	...	4693.1
27	2016	51397.4	21347.1	80201.2	109990	26154.7	10417.8	15949.4	89850.6	4486.3	...	4674.1
28	2017	51685.1	21287.7	80871.8	100019.5	27149.1	10474.8	15320.7	83486.7	4484.6	...	4685.5
29	2018	51227.2	21606.2	80321.5	95782.4	27404.6	10903.5	14993.8	88349.5	4685.3	...	4865.3
30	2019	50889.8	21858.1	80113	89388.3	26787.1	10734.7	14592.1	86998.4	4618	...	4739.5
31	2020	49836	20321.3	77060.3	80736.3	25434.7	9941	12705.2	63020.4	4213	...	4038.4
32	2021	50909.4	18346.2	77092.3	86311	25252.1	11180.6	13471.4	57961.1	4446.4	...	4494.4
33	2022	49682.9	17433.4	75449.5	82854.7	25109.3	10990.5	13084.3	80891	4203.9	...	4233.5
34	2023	49176.8	16415.6	73496.8	70134.5	24868.4	11243.8	12855.7	83875.1	4122.1	...	4380.0

35 rows × 25 columns

In [ ]:

In [412…

```
#table 2
dataset2 = 'provisionalatmoshpericemissionsghg.xlsx'
sheet_name = 2

#The second table starts from row 45 to row 80 (36 rows) and columns A up to EC
table2 = pd.read_excel(dataset2, sheet_name=sheet_name, header=44, usecols="A:EC", nrows=36)

# Print the second table
table2
```

Out[412]:

	SIC code	A	A.1	A.2	B	B.1	B.2	B.3	B.4	C	...	R.1	R.2	R.3
0	SIC(07) group	1	2	3	5	6	7	8	9	10.1	...	91	92	93
1	Industry name	Products of agriculture, hunting and related s...	Products of forestry, logging and related serv...	Fish and other fishing products; aquaculture p...	Mining of coal and lignite	Crude petroleum and natural gas	Mining of metal ores	Other mining and quarrying products	Mining support services	Processing and preserving of meat and producti...	...	Library, archive, museum and other cultural se...	Gambling and betting services	Sporting services and amusement and recreation...
2	1990	54715.1	27	948.2	25423.1	22983.5	9	1237.2	800.1	1033.2	...	182.8	267.8	1005.2
3	1991	54494.1	27.2	971.3	26059.1	22534.4	9	1244.9	760.8	1140	...	184.3	281.1	1078.9
4	1992	54119.9	28	1001	25814.5	23477.2	9.1	1203	695.5	1169.5	...	160.3	253.1	985.6
5	1993	53228.2	34.7	1057	23889.6	25204.8	9.2	1057.9	645.7	1180.2	...	164.2	262.3	1032.9
6	1994	54010.2	42.7	1076.8	15998.5	26873	9.6	997.5	709.8	1257.4	...	149.2	246.5	1003
7	1995	53906.4	33.2	1122.4	17362.9	27368.2	9.6	905	805.7	1391.4	...	144.6	242.3	1011.2
8	1996	55225.1	29.9	1088.6	16060	28798.2	7.5	913.2	854.5	1334.6	...	144	244.9	1057.6
9	1997	54706.5	36.4	1067.2	15418	28618.5	7.8	872.5	844.3	1316.3	...	127.5	220.2	964.7
10	1998	54809.8	33.7	1107.3	13103.9	29552.3	7.9	875.8	859.2	1264.2	...	157	255.2	1116.9
11	1999	55136.1	34.4	1005.6	10898.9	28494	7.2	806.2	295.3	1036.1	...	166.2	270.4	1200.7
12	2000	53193.8	34.5	905.3	9444.8	27420	7.6	1271.9	191.4	1297.3	...	170.6	282.5	1250.8
13	2001	51030	33.1	892.7	8478.8	28339.9	7.4	1230.5	154	1186.2	...	191.1	298.8	1326
14	2002	50626	32.7	836.8	8318.6	27352.9	7.3	837.5	206.2	1107	...	179.6	259.3	1212.5
15	2003	51128	32.8	789.6	6954.1	26056.7	7.4	879.6	194.7	1356.5	...	182.5	264.7	1266.4
16	2004	51972.8	32.6	807.9	6296.6	25590.8	8.2	826.4	187.3	1105.4	...	203.1	273.3	1310.1
17	2005	52096.1	37	801	4729.5	24878.7	8	949.9	172	1157.5	...	202.8	266.5	1316.6
18	2006	51117.5	35.7	746.6	4327.5	23023.6	8.3	815.6	116.7	1058.4	...	107.1	166.3	953.2
19	2007	50668.4	49	755.7	3616.2	23073.1	9	953.8	118.2	1055.3	...	95.9	152.3	911.5
20	2008	50477.6	32.9	730.2	3625.8	21982.8	7.3	1373.6	98.8	917.9	...	98.7	161.3	1020.6
21	2009	49706.2	30.6	706	3462.8	21993.5	5.7	1357.6	237.3	862.5	...	87.2	131	908
22	2010	49728.8	28.9	744.4	3291.8	21719.6	6.4	1519	125	905.8	...	85.1	133.4	918.6
23	2011	48891	29.1	727.3	3151.7	19873.4	6.5	1059.1	115.1	908.8	...	71	113.7	827.2
24	2012	49575.2	28.9	764.7	3100.7	18294.3	6.1	772.3	205.7	865.9	...	72.7	117.5	880.9
25	2013	48513.1	27.6	758.4	2226.9	17923.1	5.8	731.7	209.3	855.1	...	69.5	114.3	892.7
26	2014	50211.8	28.7	908.7	2231.7	17529.6	6.3	985.9	333.5	900.7	...	63.4	100.4	815.3
27	2015	50886.2	29.1	859.2	1890.2	19274.2	6.4	1061.2	170.4	880.7	...	42.5	80.5	862.2
28	2016	50522.8	30.4	844.2	939	18852.3	6.8	1234.1	314.8	818.1	...	44.9	81.7	902.5
29	2017	50792.9	29.6	862.7	918.6	18865.3	7.1	1392.3	104.3	854	...	46	81.8	913
30	2018	50378.6	28.7	819.9	890.7	18856.6	7.1	1744.1	107.7	912.8	...	48.1	83.8	930.3
31	2019	50136.5	28.8	724.6	870.3	18903.6	6.9	1952.6	124.7	959.1	...	46.8	81.3	915.7
32	2020	49101	23.5	711.6	858.1	17269.6	7.4	2164.8	21.4	937.7	...	38.8	74.1	856.5
33	2021	50147.1	26.1	736.1	806.9	15153.9	6.9	2312	66.4	915	...	44.5	77.1	893.2
34	2022	48951	27.3	704.6	766.1	14313.3	6.9	2314.2	32.8	896.2	...	41.4	71.9	855.3

35 rows × 133 columns

In [413…

```
# Check for missing values in each table
print(table1.isnull().sum())
print(table2.isnull().sum())
```

```
SIC code       0
A              0
B              0
C              0
D              0
E              0
F              0
G              0
H              0
I              0
J              0
K              0
L              0
M              0
N              0
O              0
P              0
Q              0
R              0
S              0
T              0
-              0
Unnamed: 22    0
Unnamed: 23    0
Unnamed: 24    0
dtype: int64
SIC code        0
A               0
A.1             0
A.2             0
B               0
               ..
S.2             0
T               0
Unnamed: 130    0
Unnamed: 131    0
Unnamed: 132    1
Length: 133, dtype: int64
```

In [ ]:

In [ ]:

In [ ]:

In [414…
```
##Reference:(Naveen Nelamali, 2022)
# Selecting the numerical data for the years skipping the first column and first row
numerical_data_table1 = table1.iloc[1:, 1:].apply(pd.to_numeric, errors='coerce')

#make industry names as headings
numerical_data_table1.columns = industry_names_table1
```

In [415…
```
#descriptive statistics
numerical_data_table1.describe()
```

Out[415]:

	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning supply	Water supply; sewerage, waste management and remediation activities	Construction	Wholesale and retail trade; repair of motor vehicles and motorcycles	Transport and storage	Accommodation and food services
count	34.000000	34.000000	34.000000	34.000000	34.000000	34.000000	34.000000	34.000000	34.000000
mean	52441.426471	32031.391176	117467.679412	159317.214706	50961.529412	9685.129412	14238.305882	83954.826471	3797.341176
std	2267.946845	11445.279839	35606.752905	42995.422940	22023.914957	901.871906	1673.486214	10880.706396	443.929933
min	49176.800000	16415.600000	73496.800000	70134.500000	24868.400000	8072.000000	11560.800000	57961.100000	3015.300000
25%	50894.700000	21669.175000	86217.125000	136028.600000	27212.975000	9214.600000	12833.275000	79570.775000	3451.400000
50%	51729.800000	28031.050000	116839.500000	176512.100000	51001.050000	9587.650000	13988.350000	86352.150000	3699.850000
75%	54876.400000	43424.875000	151913.950000	190065.450000	74922.950000	10390.775000	15813.325000	91683.425000	4164.100000
max	56343.600000	51199.300000	176254.400000	217602.200000	79209.400000	11243.800000	17024.200000	99137.900000	4685.300000

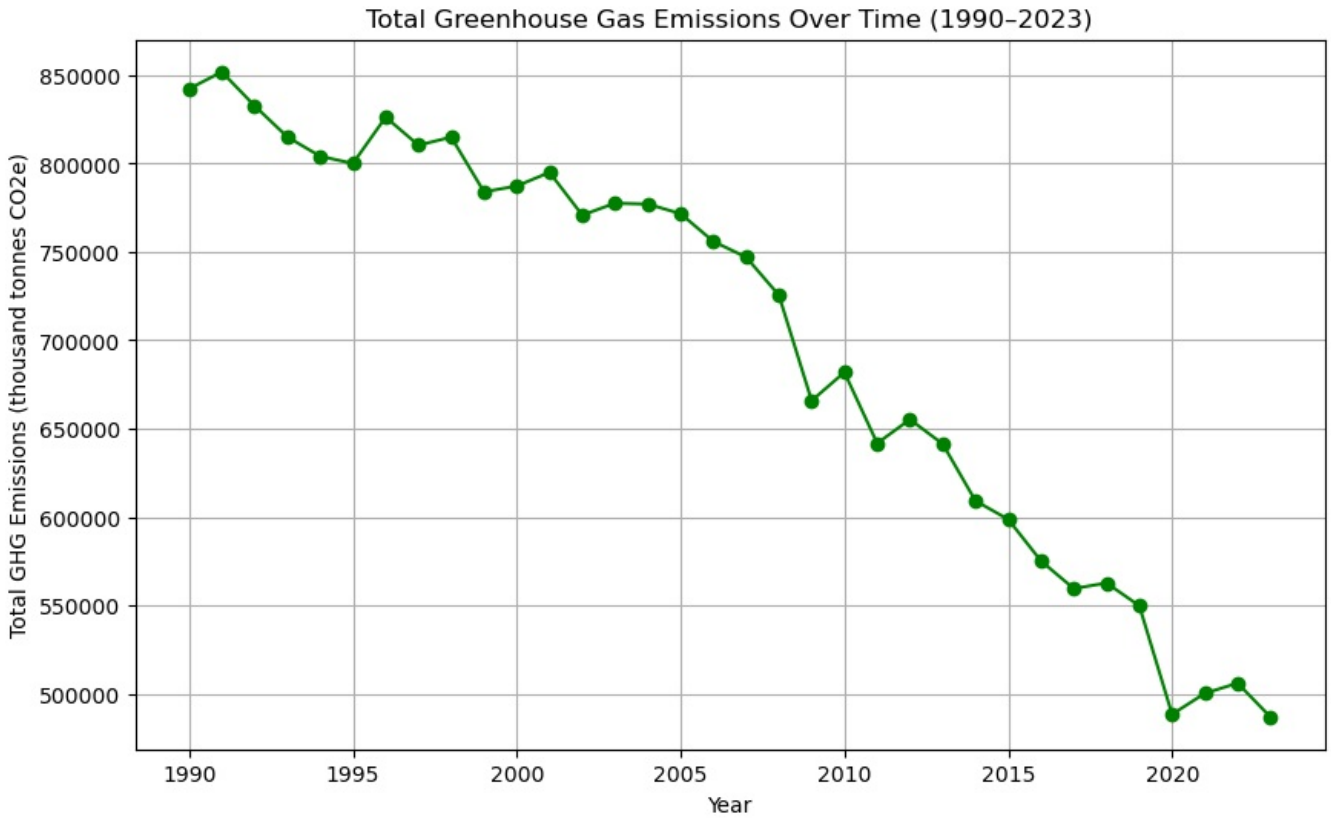
8 rows × 24 columns

In [ ]:

In [ ]:

In [ ]:
```
#Line graph
```

In [416…
```
#Line graph

# Extracting the years (first column starting from row 1)
years = table1.iloc[1:, 0].astype(int)

# Extracting the 'Total greenhouse gas emissions' column (last column)
total_ghg_emissions = table1.iloc[1:, -1].apply(pd.to_numeric, errors='coerce')

#sorting the df
emissions_data = pd.DataFrame({'Year': years, 'Total GHG Emissions': total_ghg_emissions})
emissions_data = emissions_data.sort_values(by='Year')

# Plot total greenhouse gas emissions over time
plt.figure(figsize=(10, 6))
plt.plot(emissions_data['Year'], emissions_data['Total GHG Emissions'], marker='o', linestyle='-', color='green

#labels
plt.xlabel("Year")
plt.ylabel("Total GHG Emissions (thousand tonnes CO2e)")
plt.title("Total Greenhouse Gas Emissions Over Time (1990–2023)")
plt.grid(True)

# Show the plot

plt.show()
```



In [ ]:

In [ ]:

In [ ]:

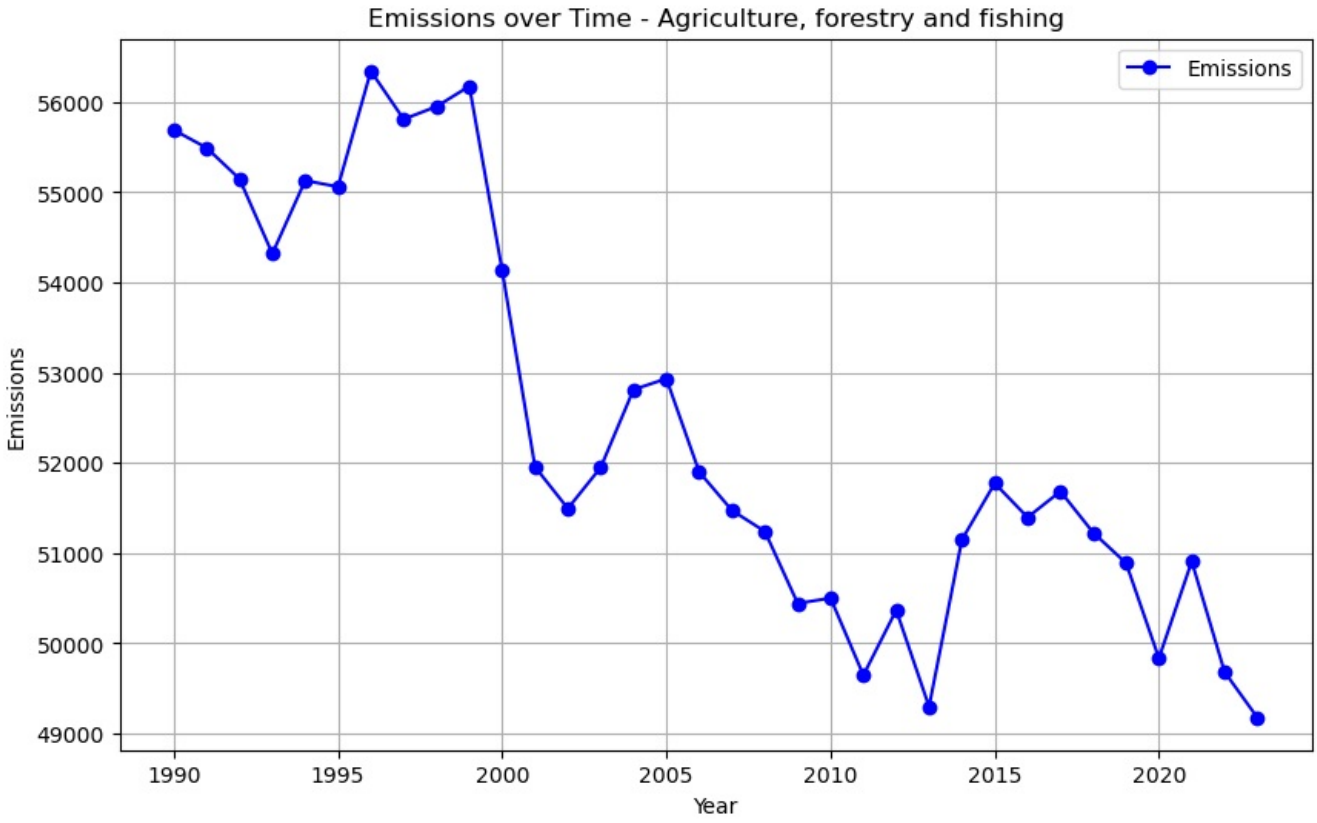
In [ ]:

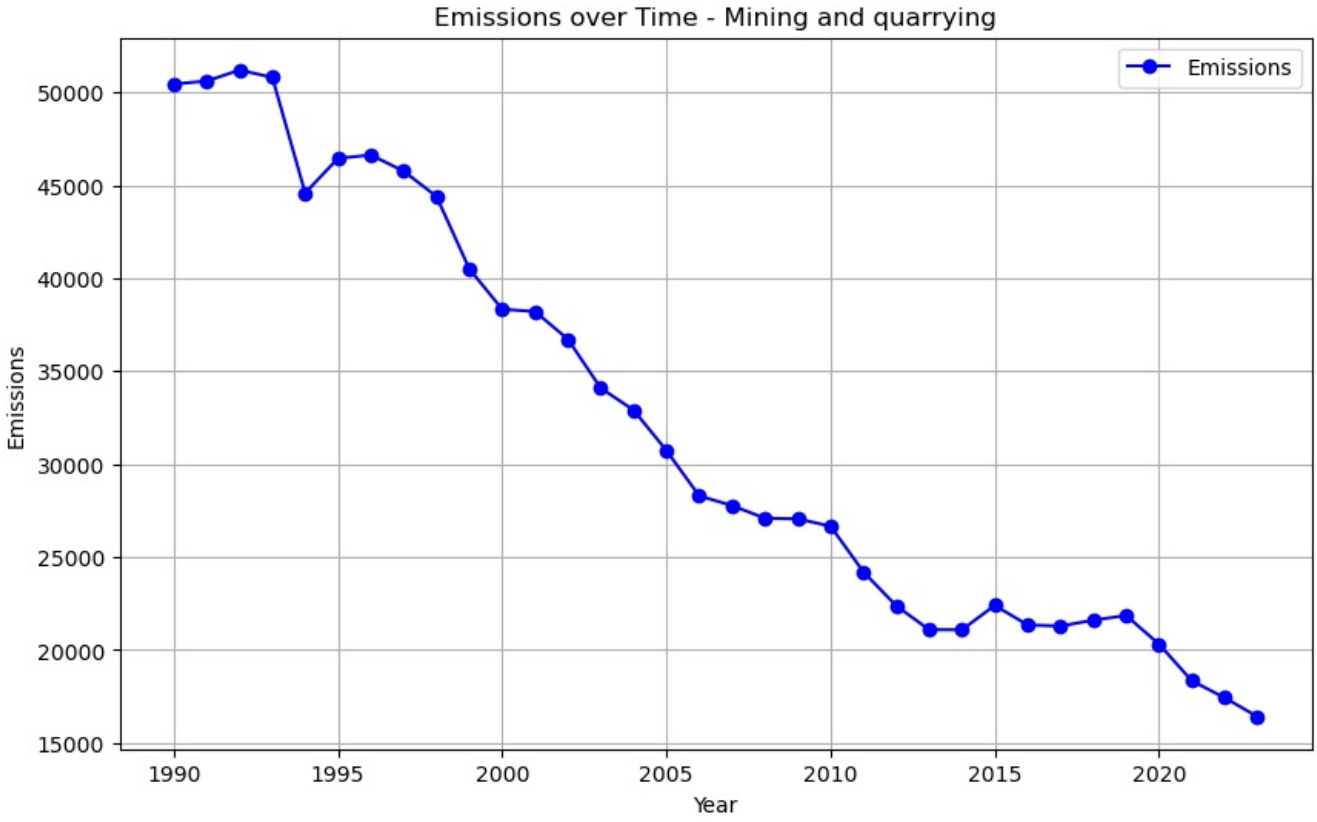
In [ ]:

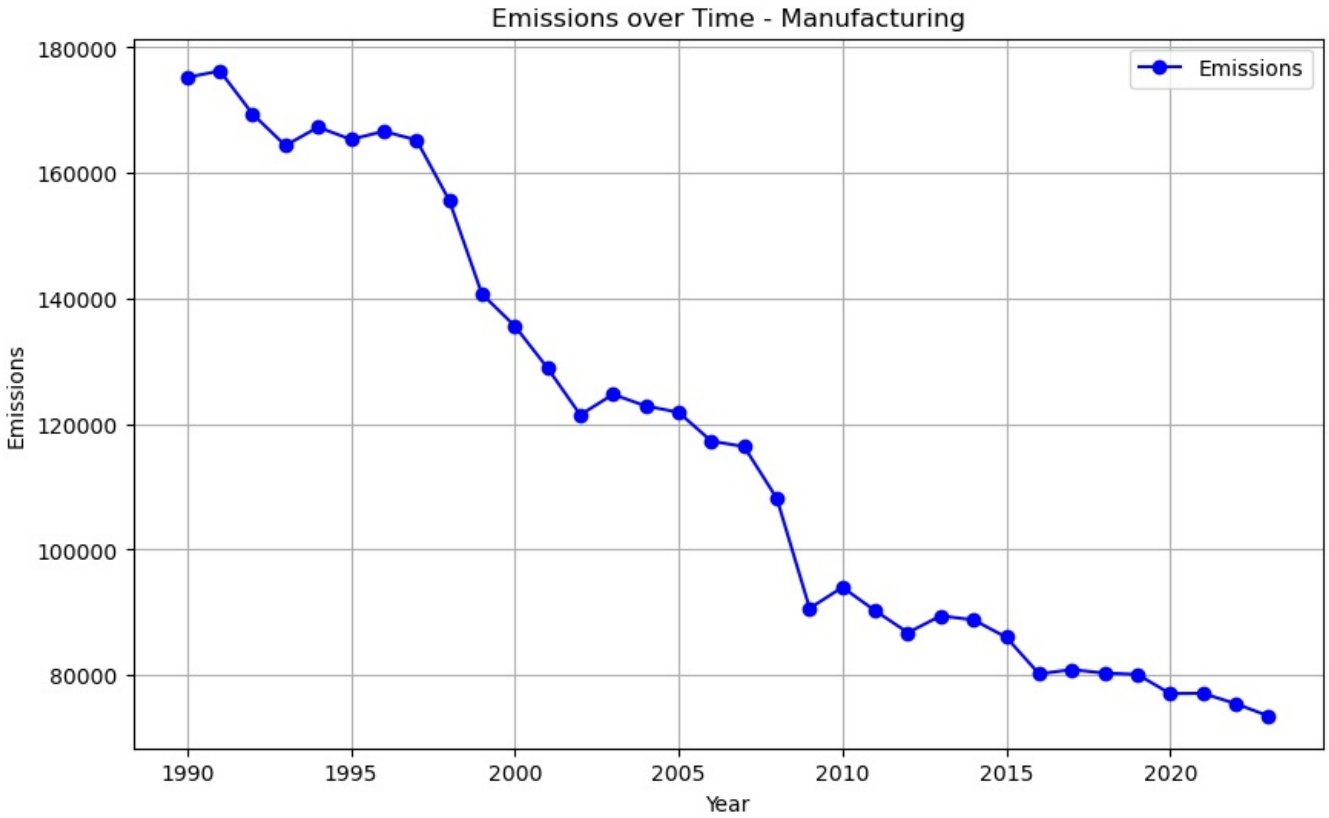
In [ ]:

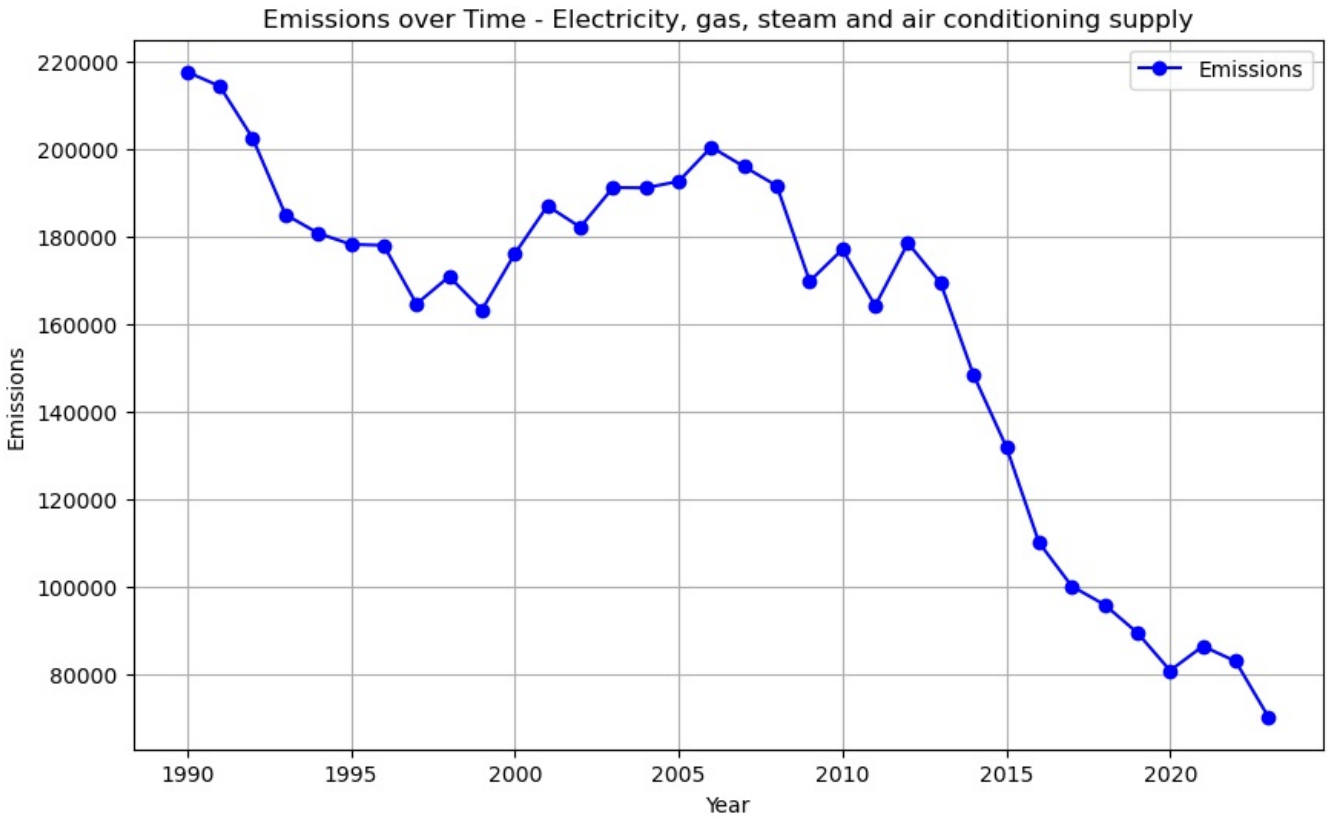
In [417…
```
#Line graph for each sector

# Extracting years (from column 1, starting from row 1 going down)
years = table1.iloc[1:, 0].values

# Extract the industry names (from column 2 onwards in row 0)
industry_names = table1.iloc[0, 1:].values

# Loop through each industry to plot the data
for industry_index, industry in enumerate(industry_names):
    # Extract emissions data for the current industry in the loop
    emissions_data = table1.iloc[1:, industry_index + 1].values
```

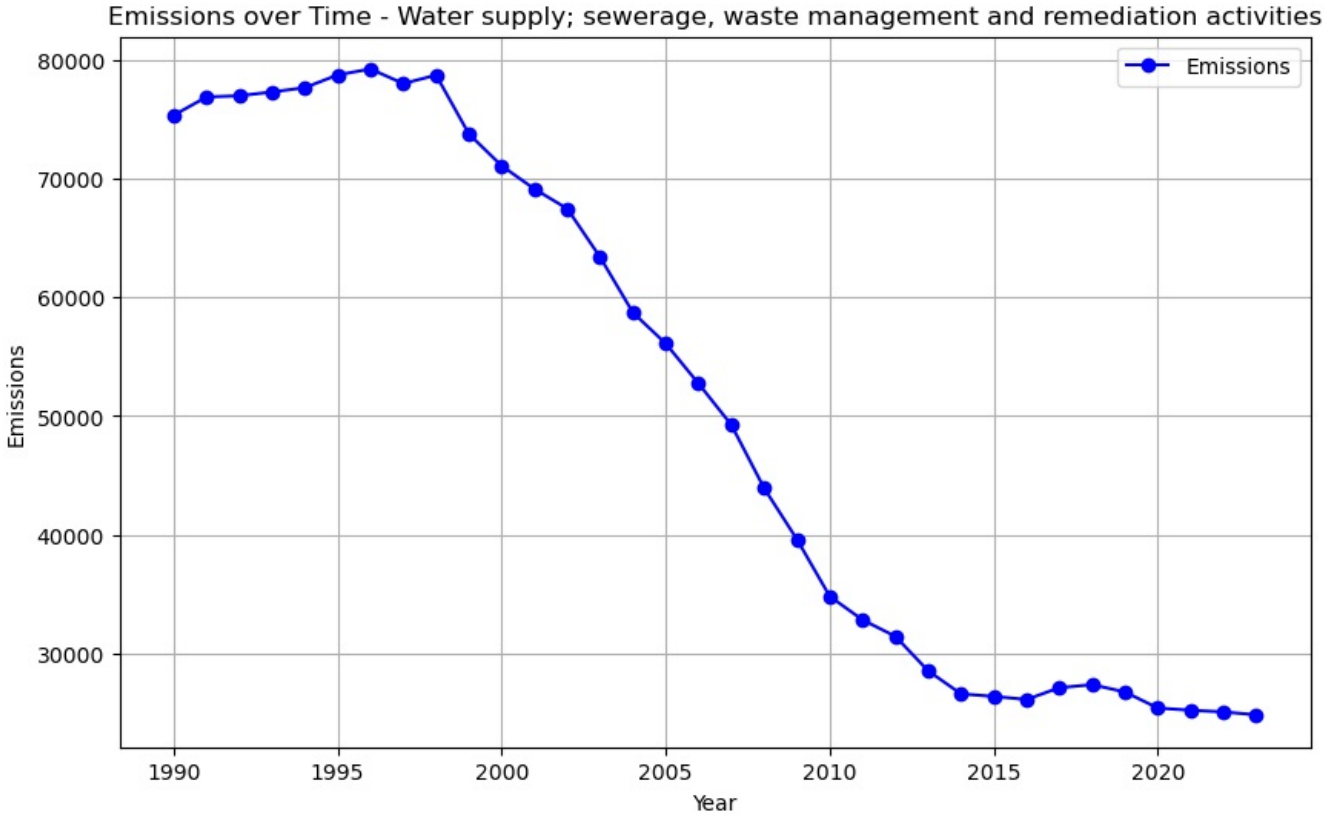
```
# Creating a figure and axis for each plot
plt.figure(figsize=(10, 6))
plt.plot(years, emissions_data, label='Emissions', marker='o', color='b')

#plot labels
plt.title(f'Emissions over Time - {industry}')
plt.xlabel('Year')
plt.ylabel('Emissions')
plt.legend()
plt.grid(True)

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

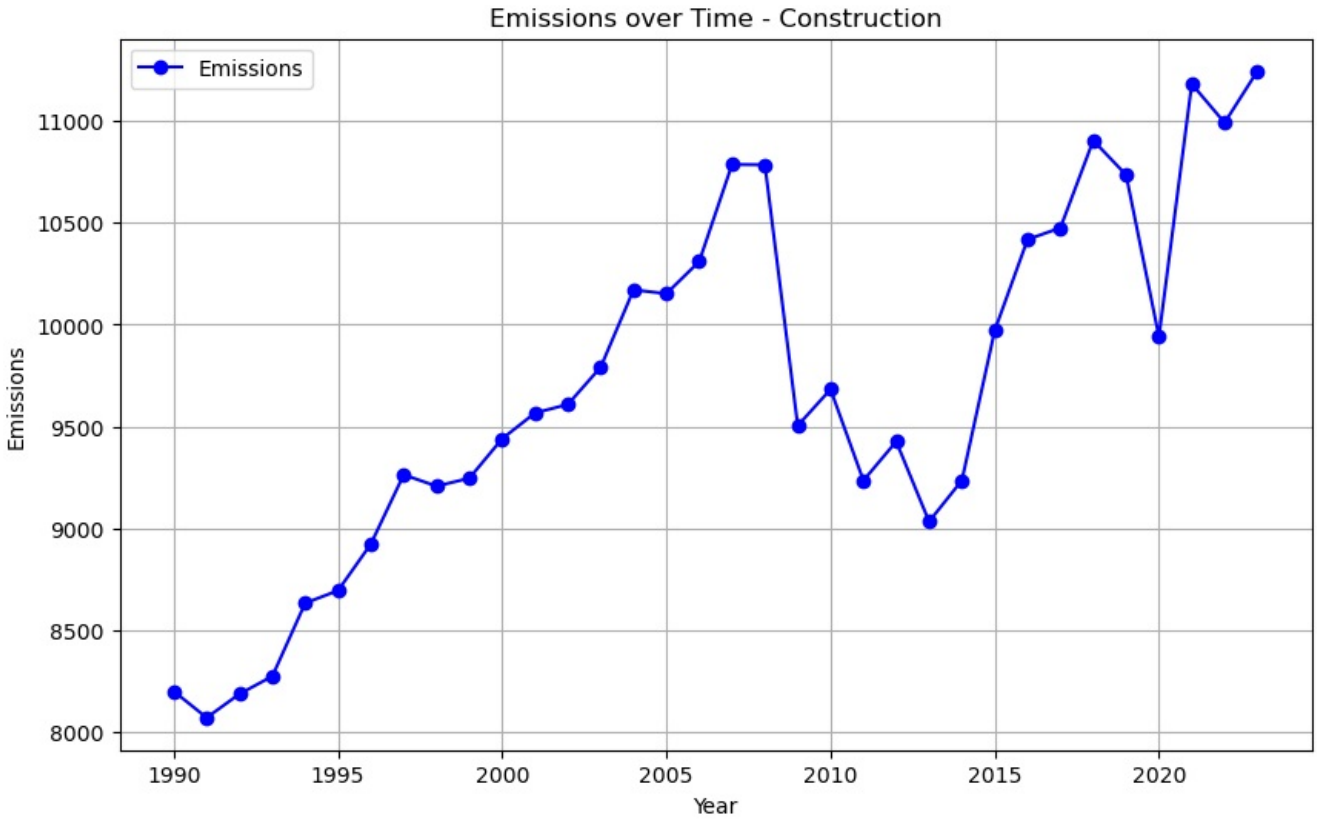




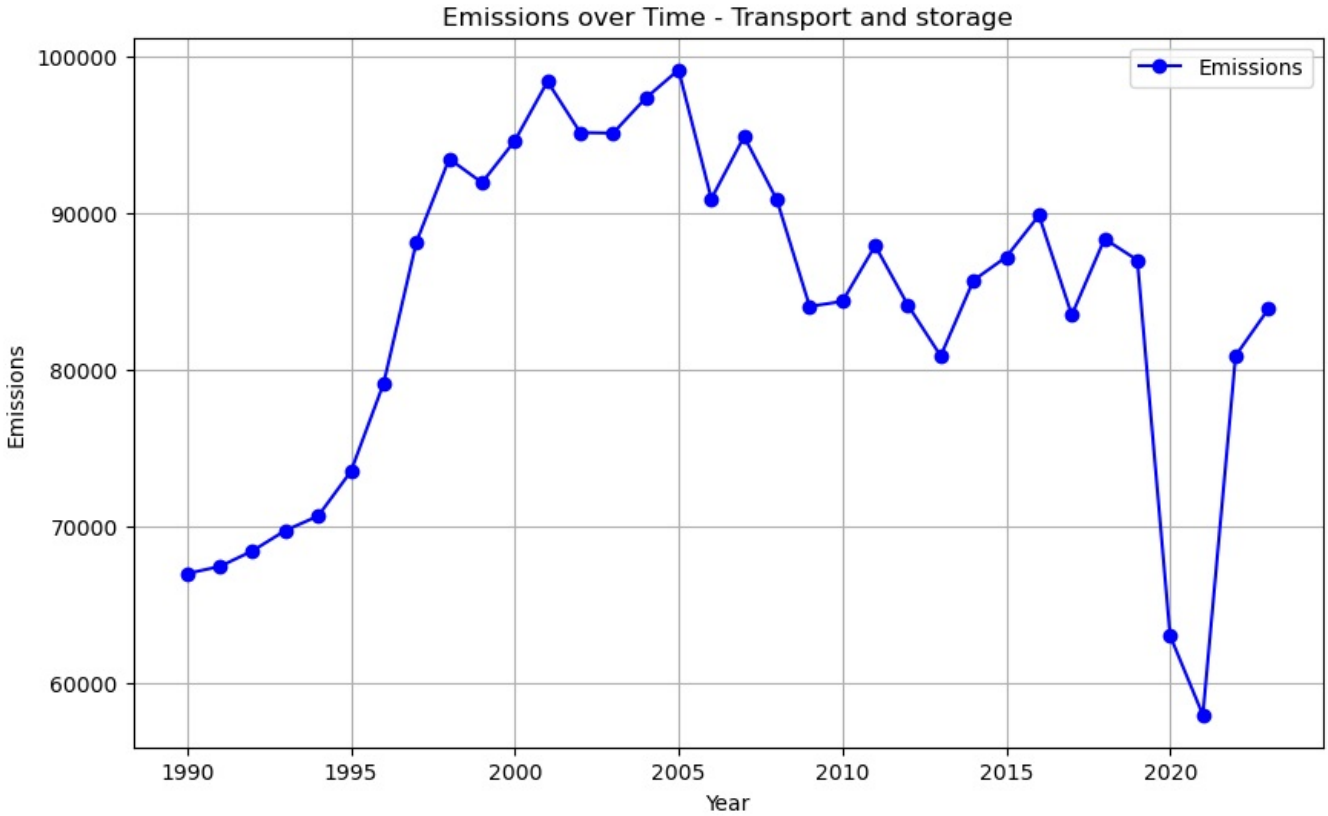

Emissions over Time - Manufacturing
Emissions
180000
160000
140000
120000
100000
80000
Emissions
1990
1995
2000
2005
2010
2015
2020
Year

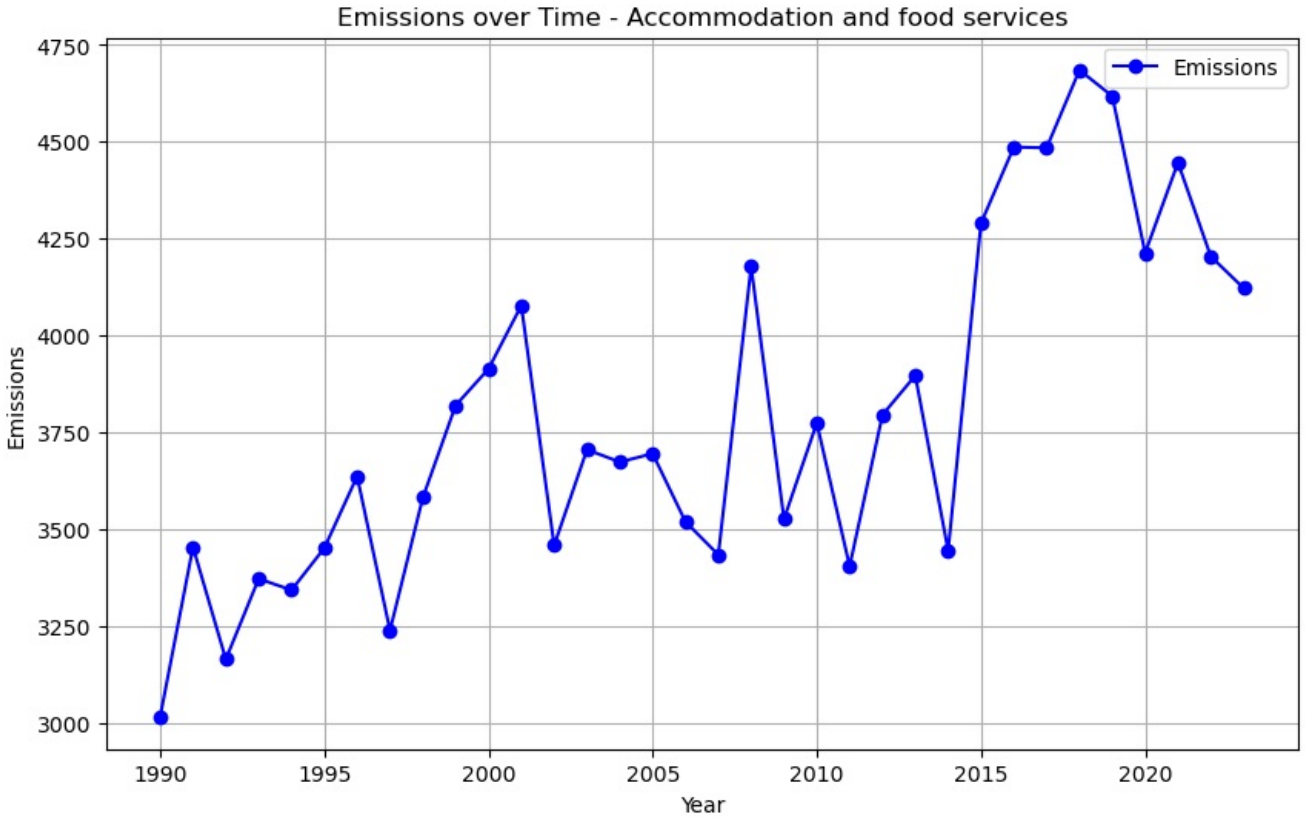

Emissions over Time - Electricity, gas, steam and air conditioning supply
Emissions
220000
200000
180000
160000
140000
120000
100000
80000
Emissions
1990
1995
2000
2005
2010
2015
2020
Year

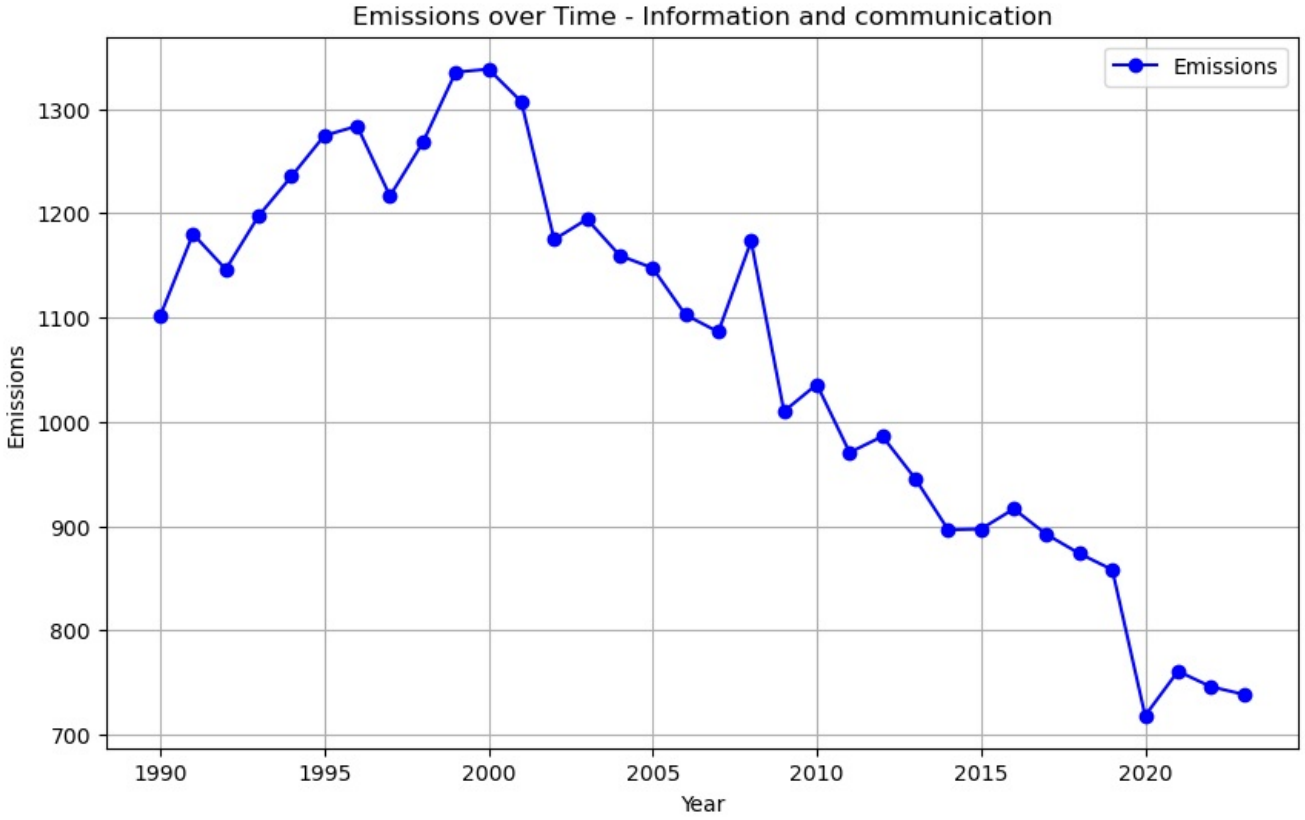

Emissions over Time - Water supply; sewerage, waste management and remediation activities
Emissions
80000
70000
60000
50000
40000
30000
Emissions
1990
1995
2000
2005
2010
2015
2020
Year

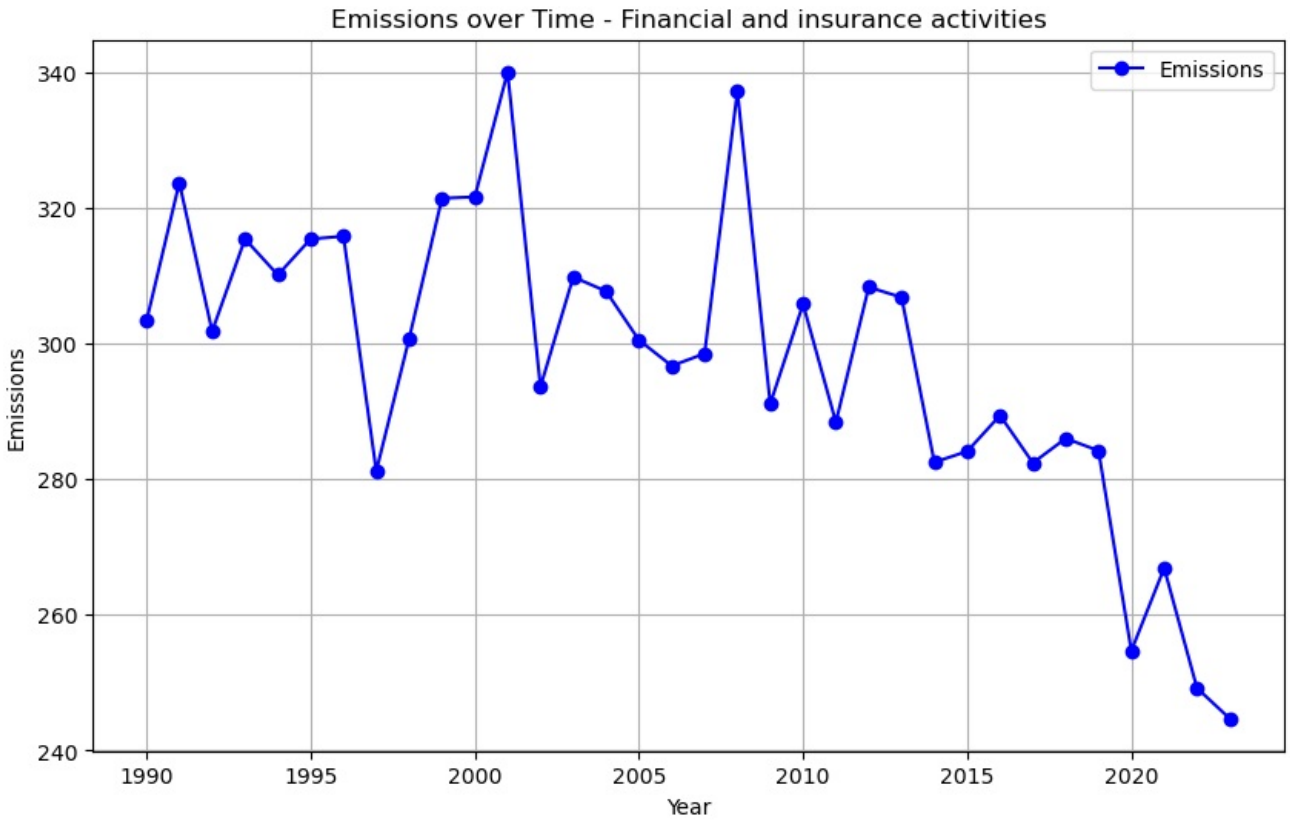

Emissions over Time - Construction
Emissions
11000
10500
10000
9500
9000
8500
8000
Emissions
1990
1995
2000
2005
2010
2015
2020
Year

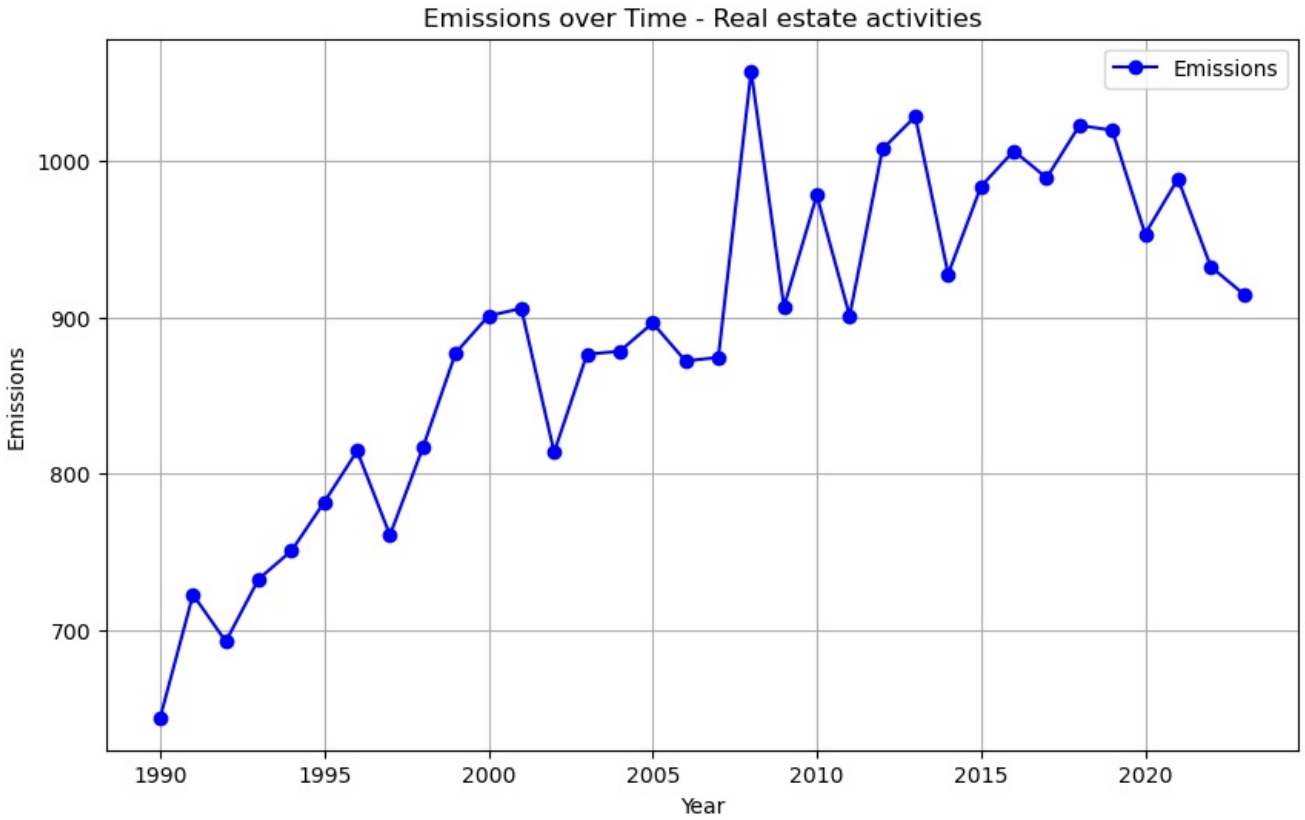

Emissions over Time - Wholesale and retail trade; repair of motor vehicles and motorcycles
Emissions
17000
16000
15000
14000
13000
12000
Emissions
1990
1995
2000
2005
2010
2015
2020
Year

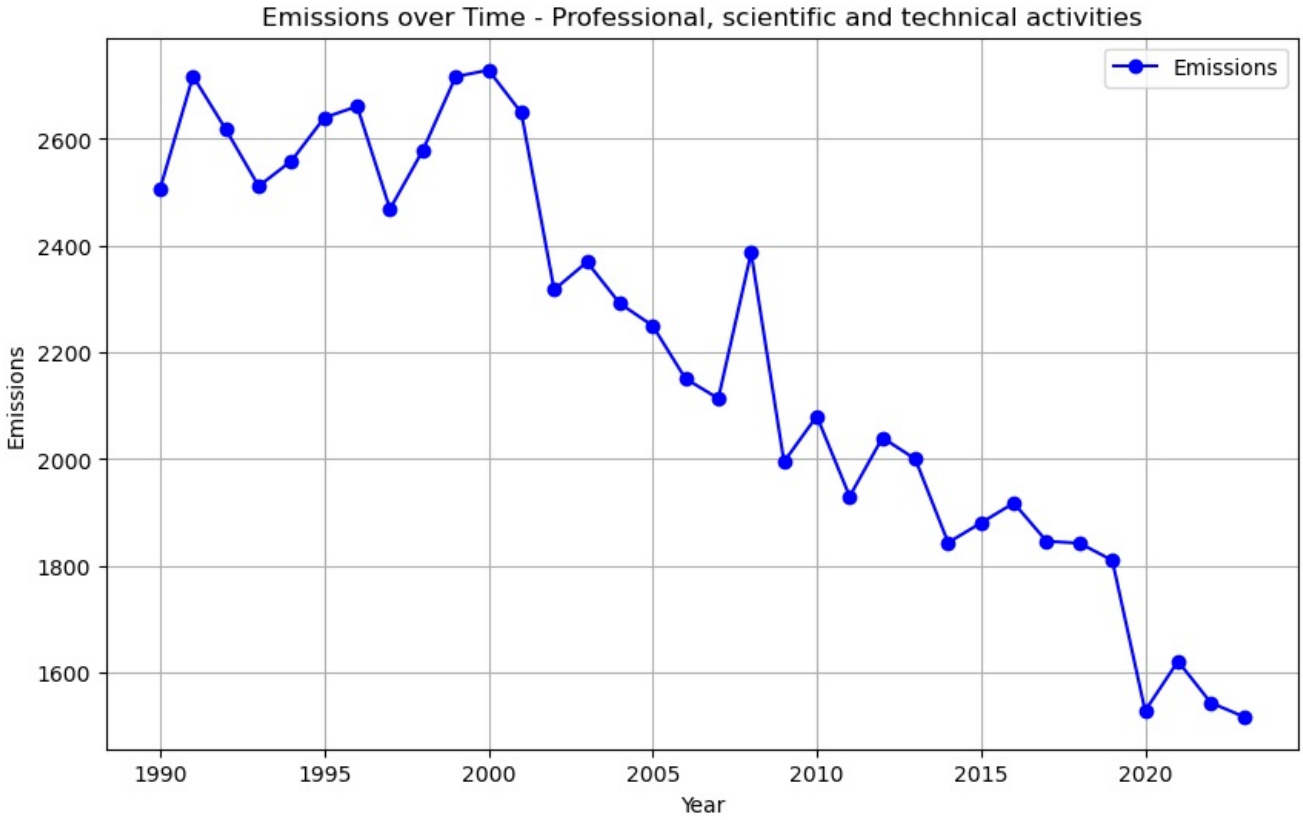

Emissions over Time - Transport and storage
Emissions
100000
90000
80000
70000
60000
Emissions
1990
1995
2000
2005
2010
2015
2020
Year

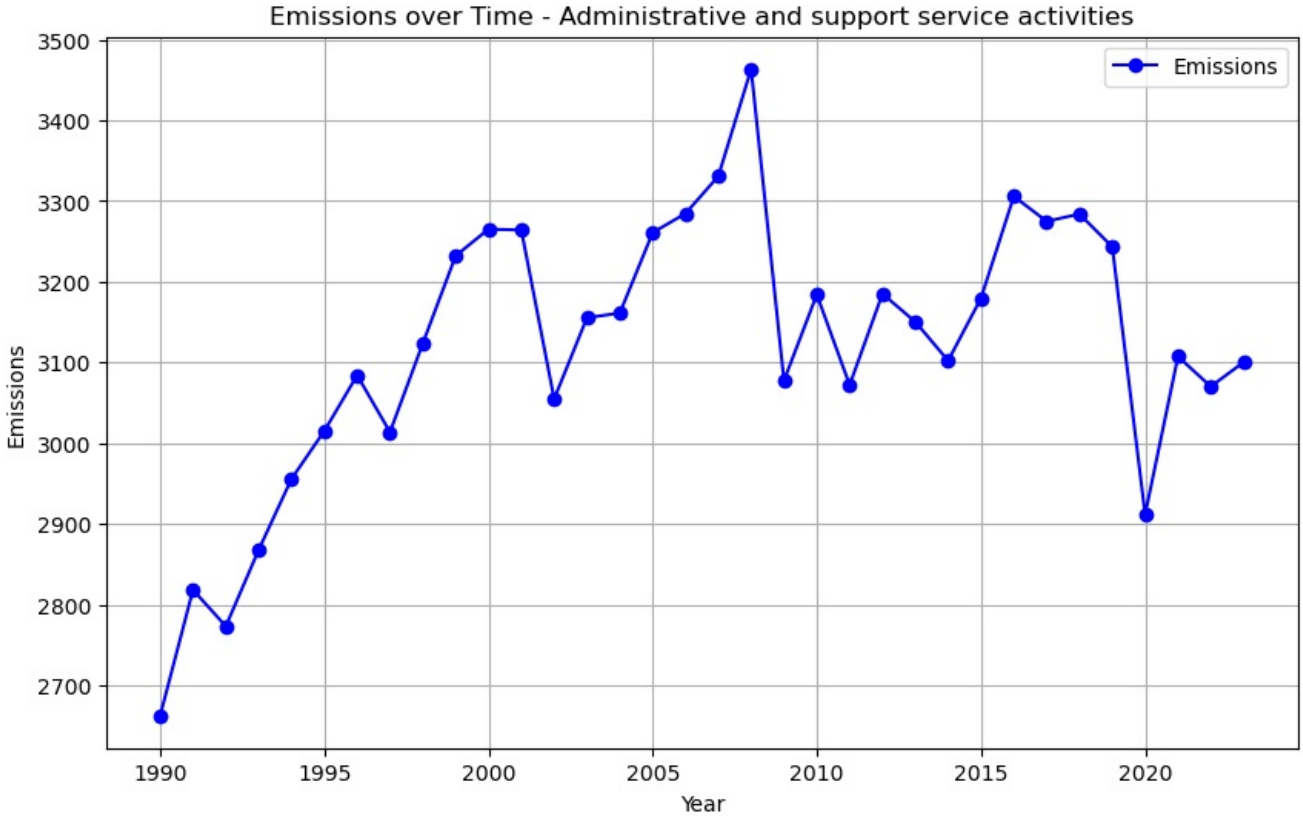

Emissions over Time - Accommodation and food services
Emissions
Year
4750
4500
4250
4000
3750
3500
3250
3000
1990
1995
2000
2005
2010
2015
2020

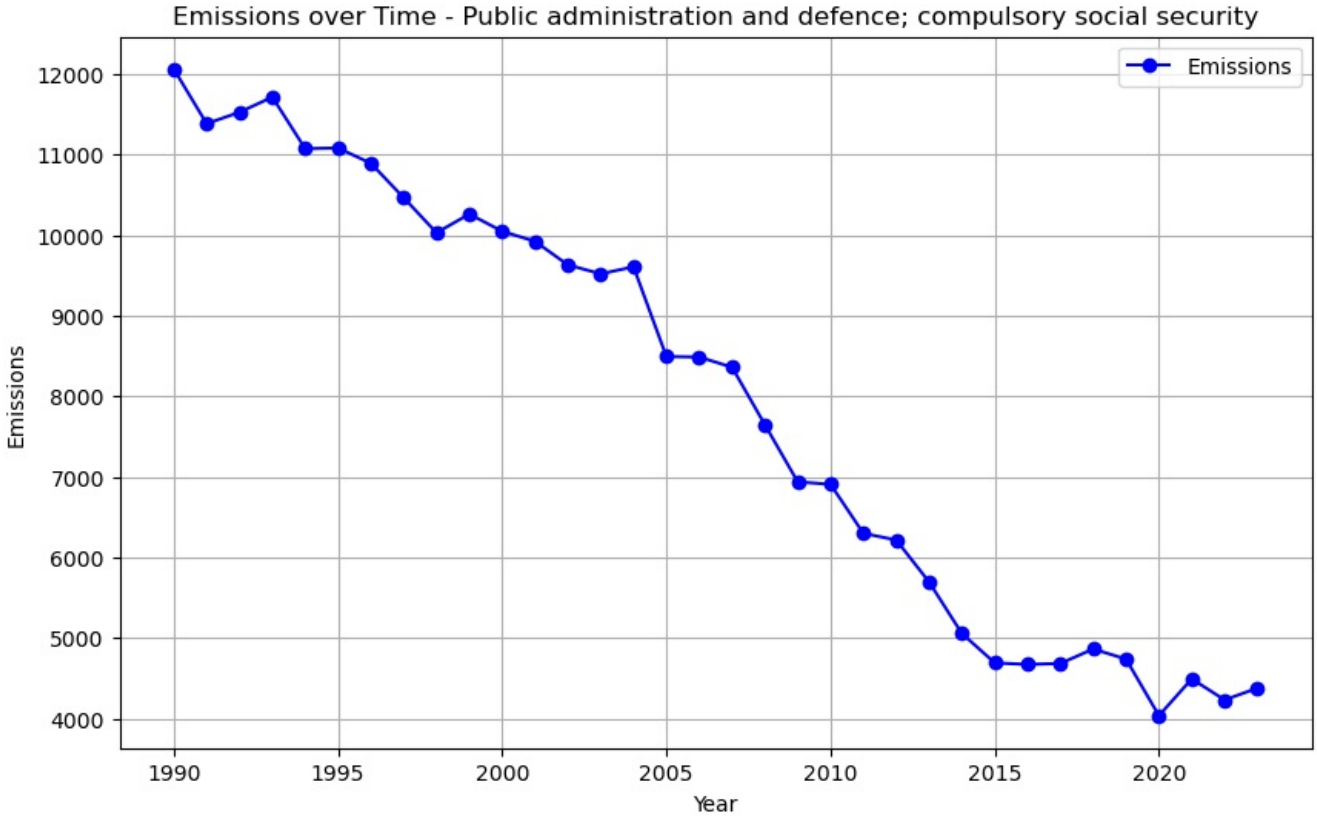

Emissions over Time - Information and communication
Emissions
Year
1300
1200
1100
1000
900
800
700
1990
1995
2000
2005
2010
2015
2020

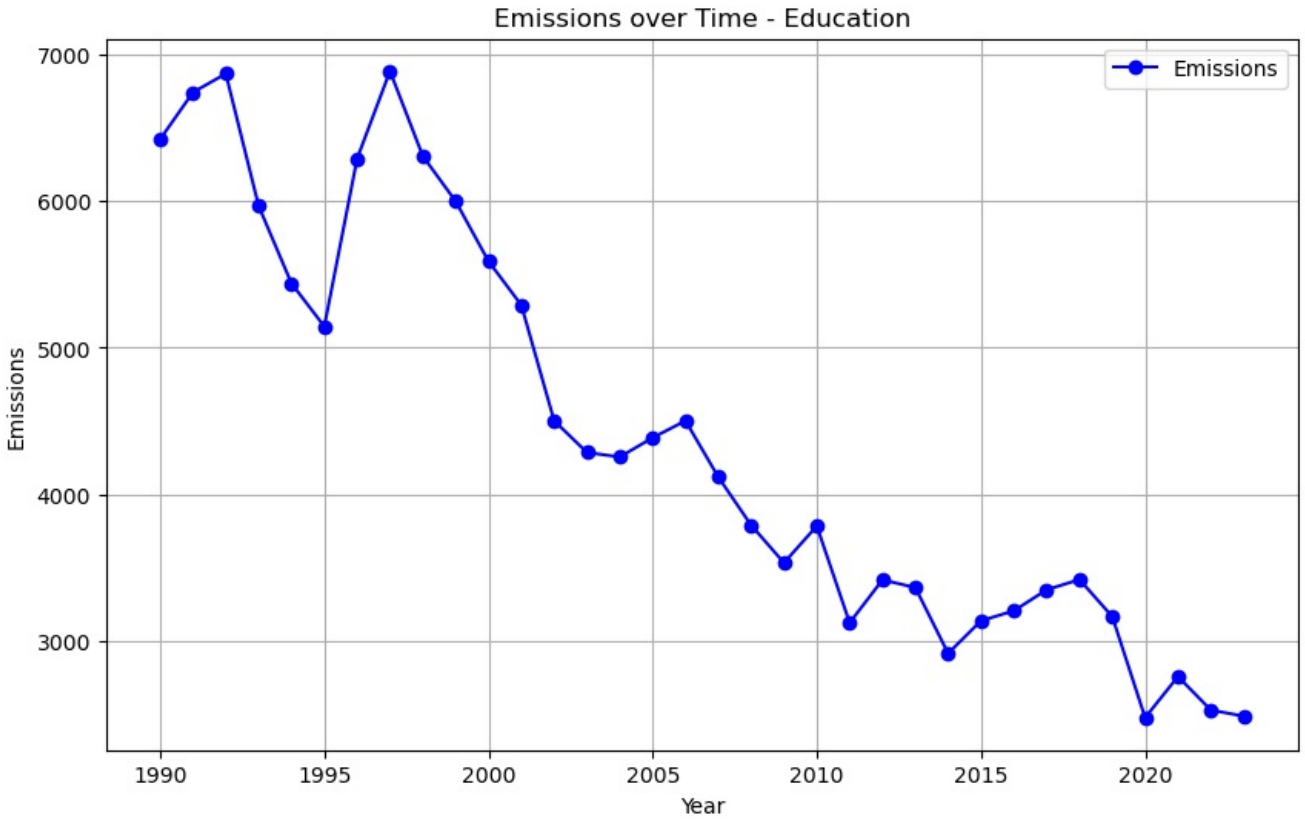

Emissions over Time - Financial and insurance activities
Emissions
340
320
300
280
260
240
Emissions
1990
1995
2000
2005
2010
2015
2020
Year

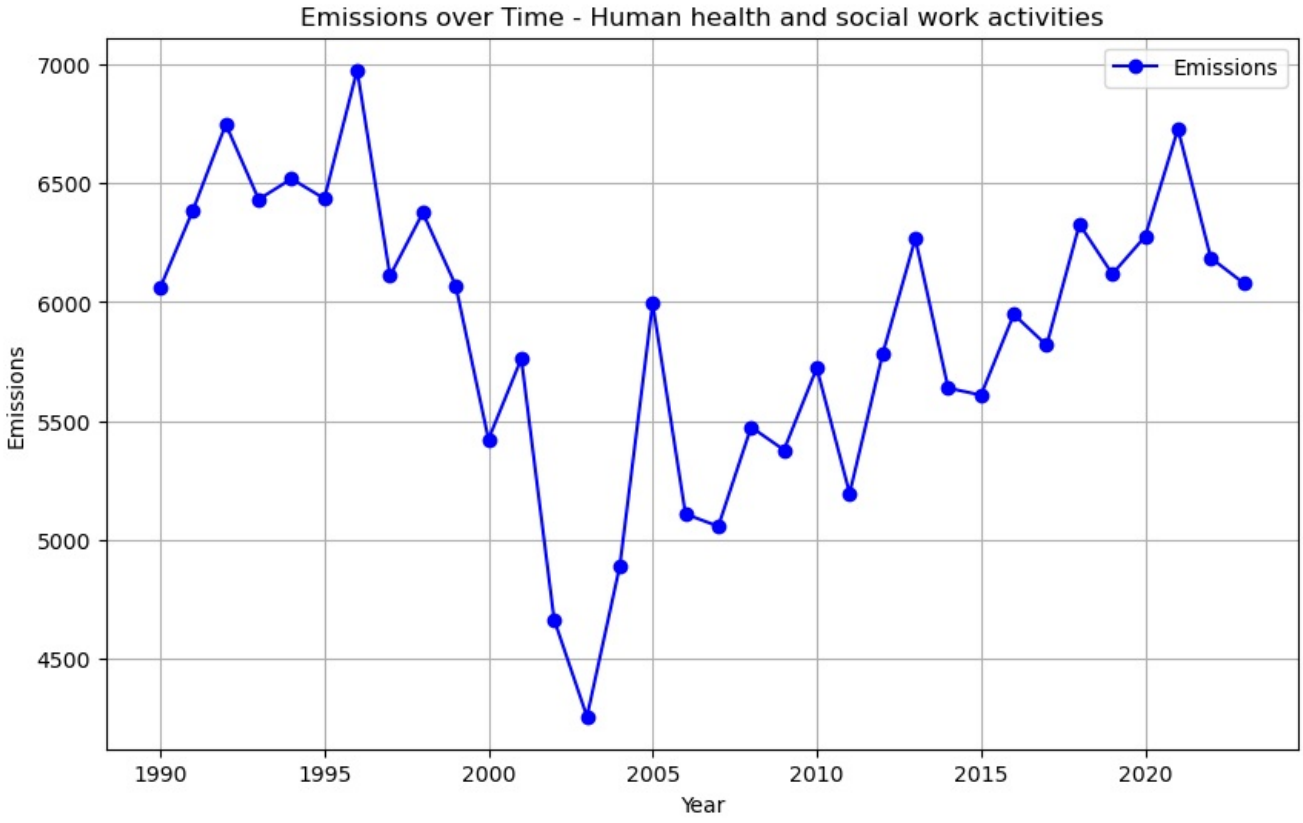

Emissions over Time - Real estate activities
Emissions
1000
900
800
700
Emissions
1990
1995
2000
2005
2010
2015
2020
Year

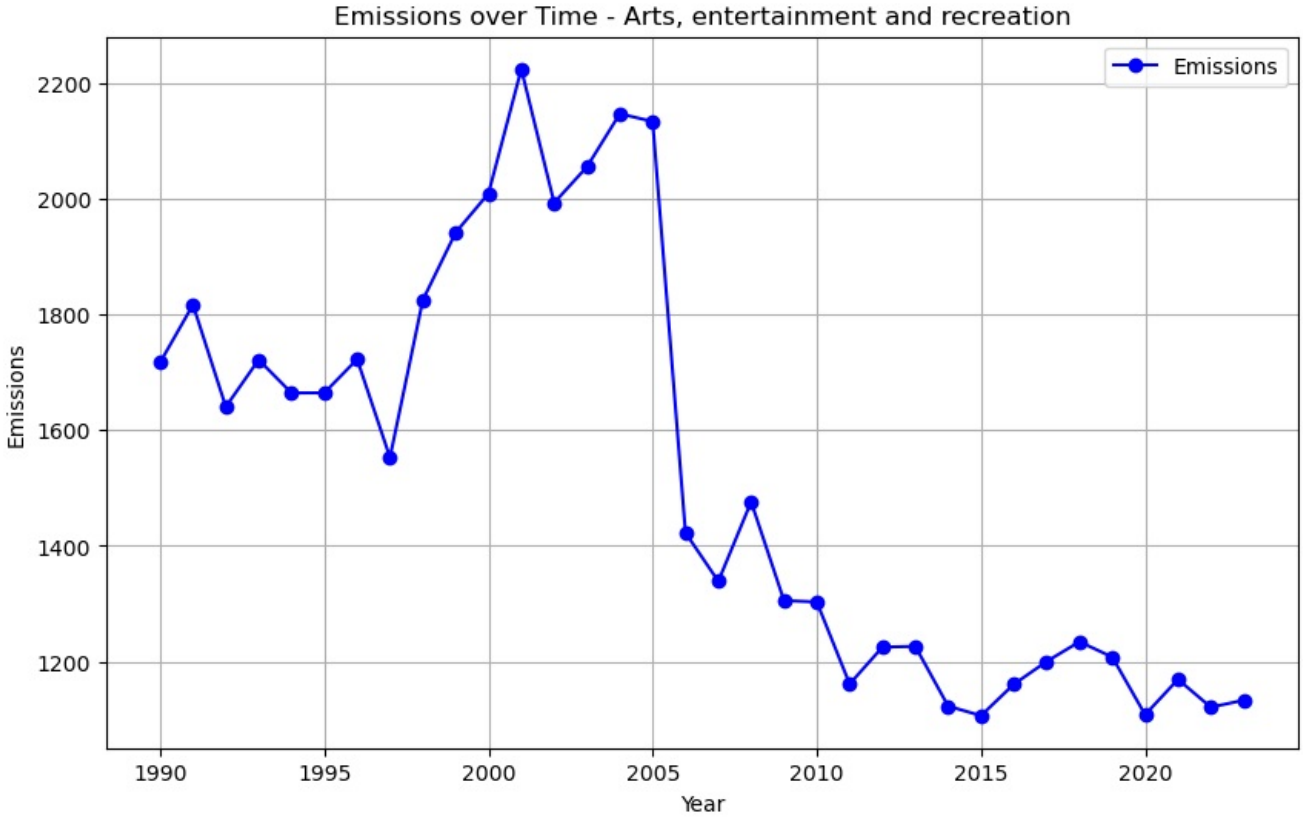

Emissions over Time - Professional, scientific and technical activities
Emissions
Emissions
Year
1600
1800
2000
2200
2400
2600
1990
1995
2000
2005
2010
2015
2020

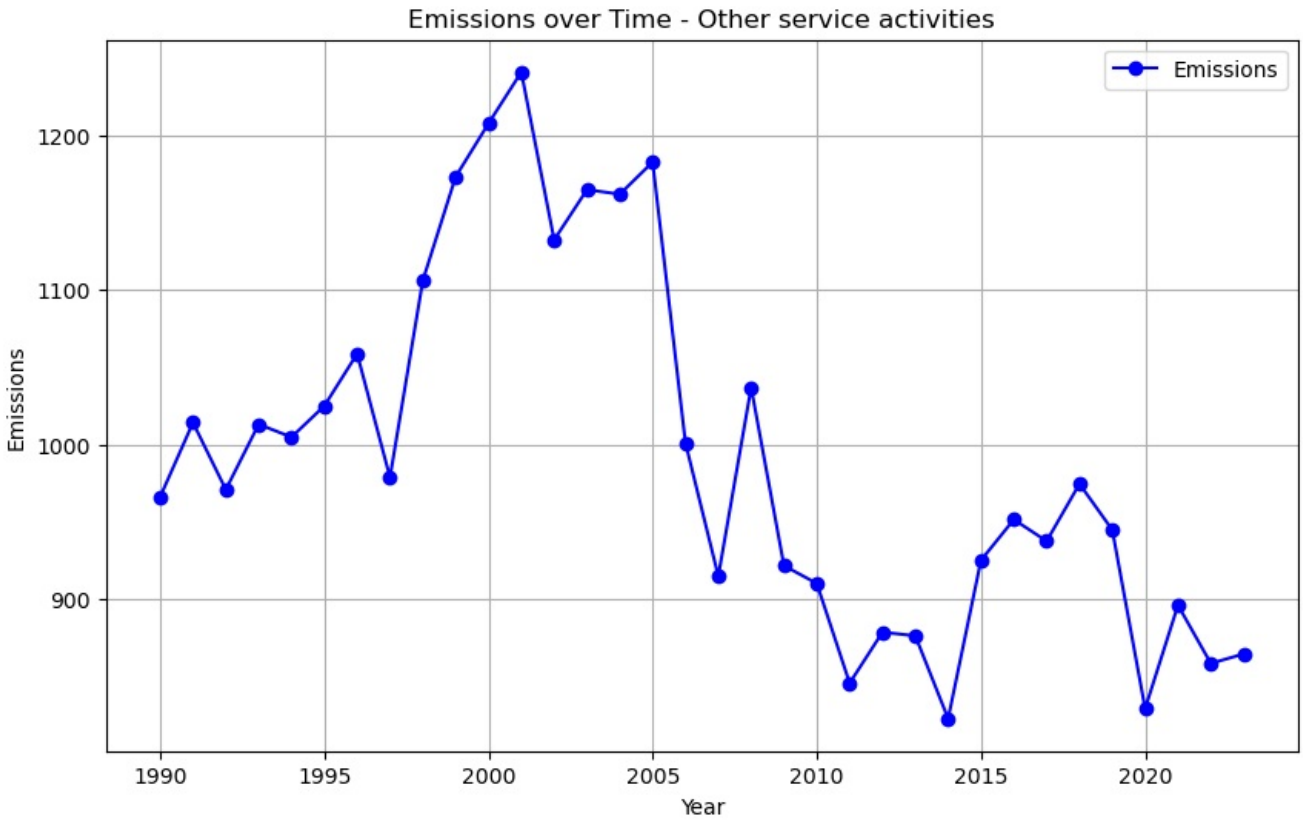

Emissions over Time - Administrative and support service activities
Emissions
Emissions
Year
2700
2800
2900
3000
3100
3200
3300
3400
3500
1990
1995
2000
2005
2010
2015
2020


Emissions over Time - Public administration and defence; compulsory social security
Emissions
Emissions
Year
12000
11000
10000
9000
8000
7000
6000
5000
4000
1990
1995
2000
2005
2010
2015
2020

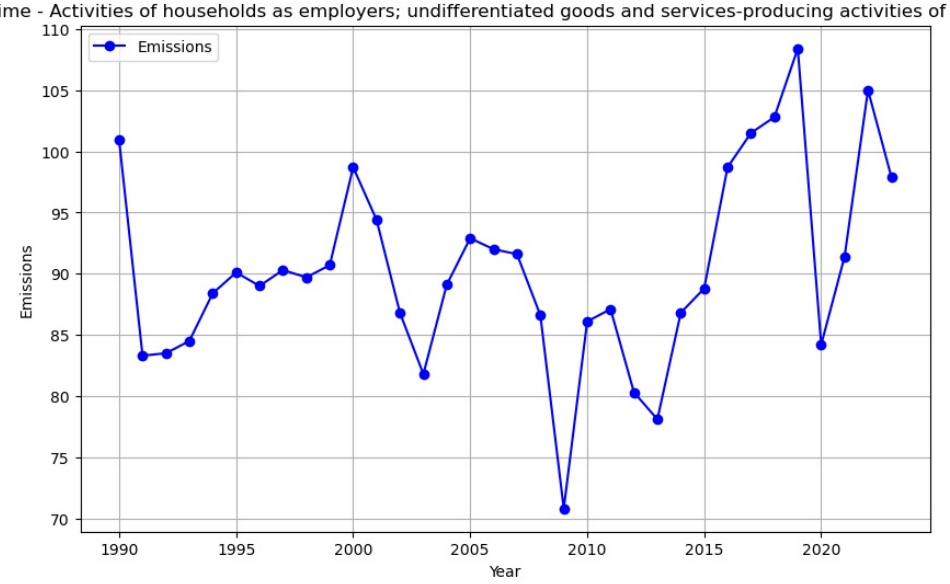

Emissions over Time - Education
Emissions
Emissions
Year
7000
6000
5000
4000
3000
1990
1995
2000
2005
2010
2015
2020

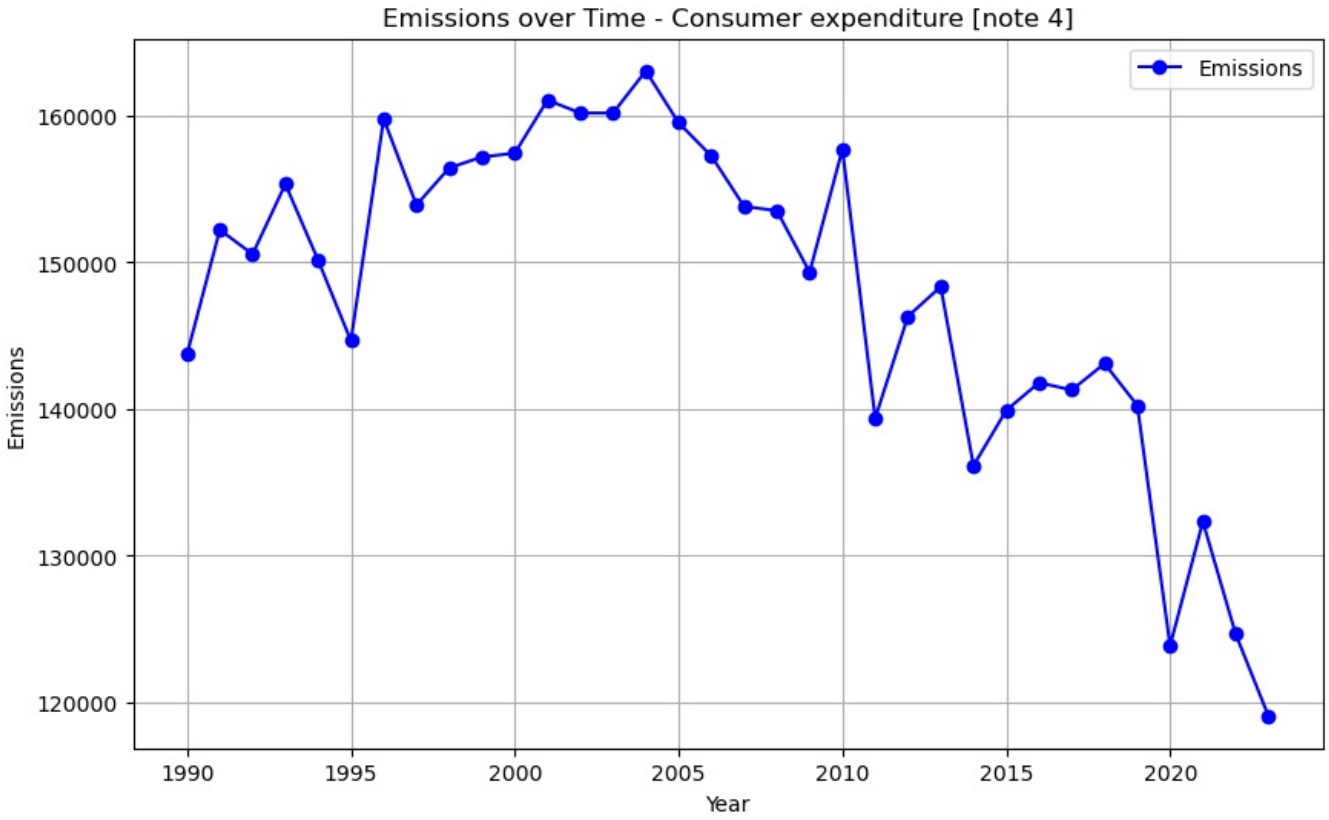

Emissions over Time - Human health and social work activities
Emissions
Year
7000
6500
6000
5500
5000
4500
1990
1995
2000
2005
2010
2015
2020

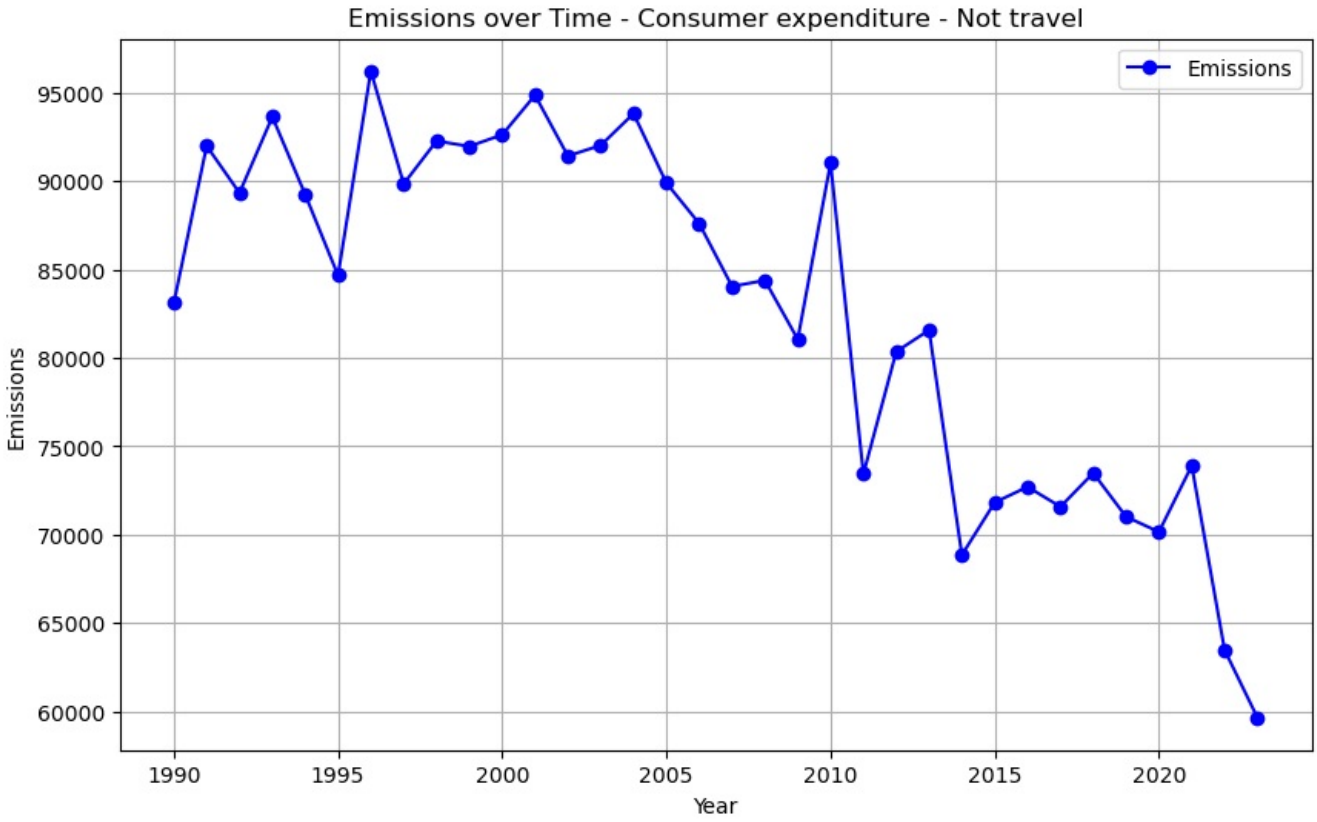

Emissions over Time - Arts, entertainment and recreation
Emissions
Year
2200
2000
1800
1600
1400
1200
1990
1995
2000
2005
2010
2015
2020

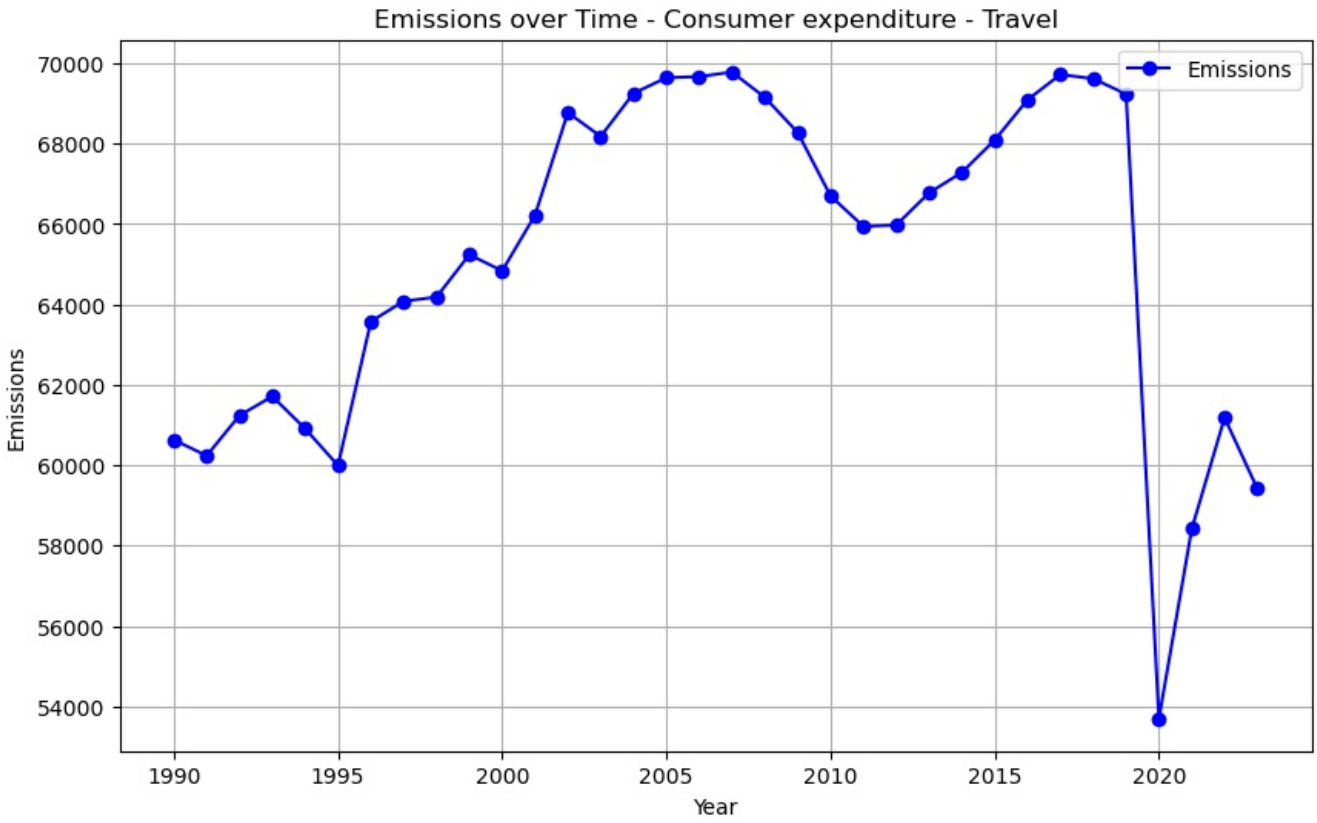


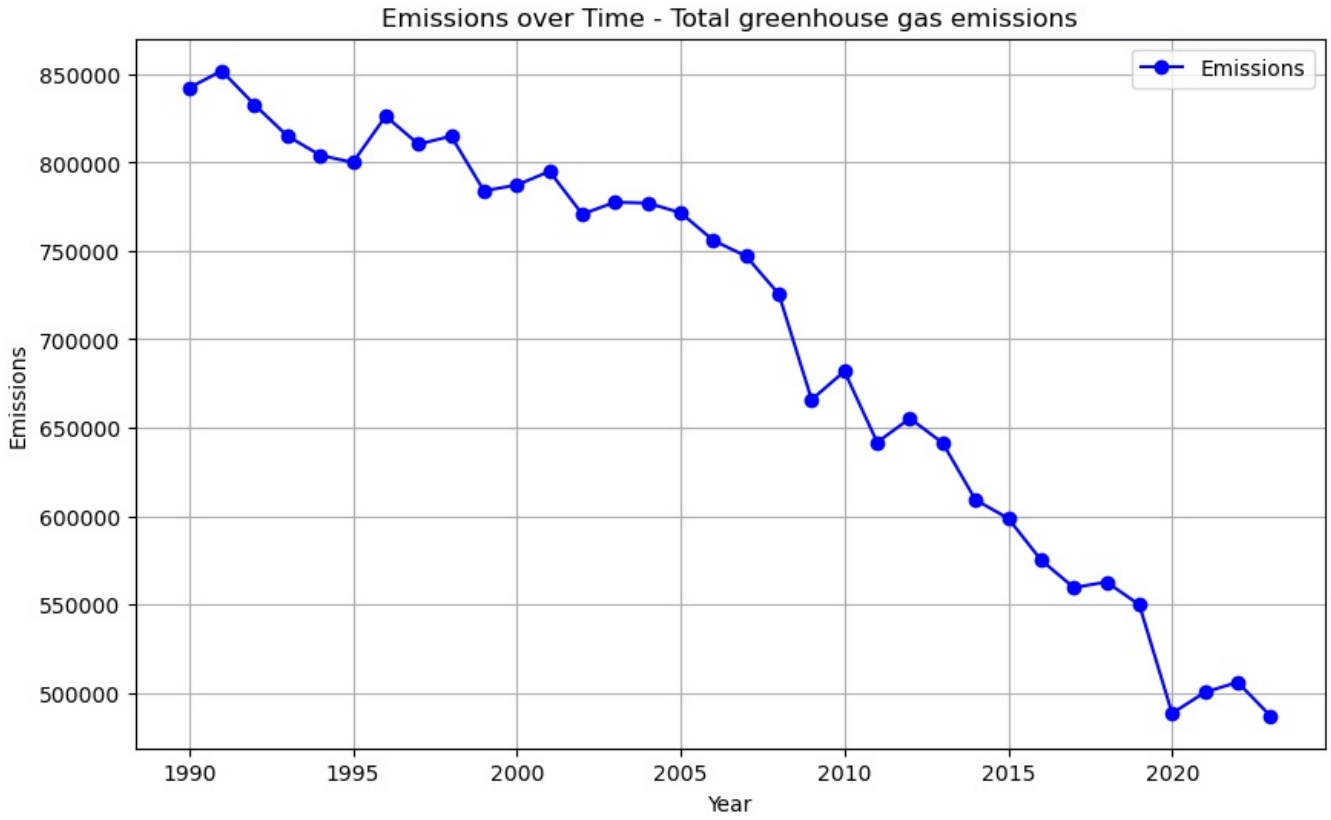
Emissions over Time - Activities of households as employers; undifferentiated goods and services-producing activities of households for own use




Emissions over Time - Consumer expenditure [note 4]
Emissions
160000
150000
140000
130000
120000
Emissions
1990
1995
2000
2005
2010
2015
2020
Year


Emissions over Time - Consumer expenditure - Not travel
Emissions
95000
90000
85000
80000
75000
70000
65000
60000
Emissions
1990
1995
2000
2005
2010
2015
2020
Year


Emissions over Time - Consumer expenditure - Travel
Emissions
70000
68000
66000
64000
62000
60000
58000
56000
54000
Emissions
1990
1995
2000
2005
2010
2015
2020
Year

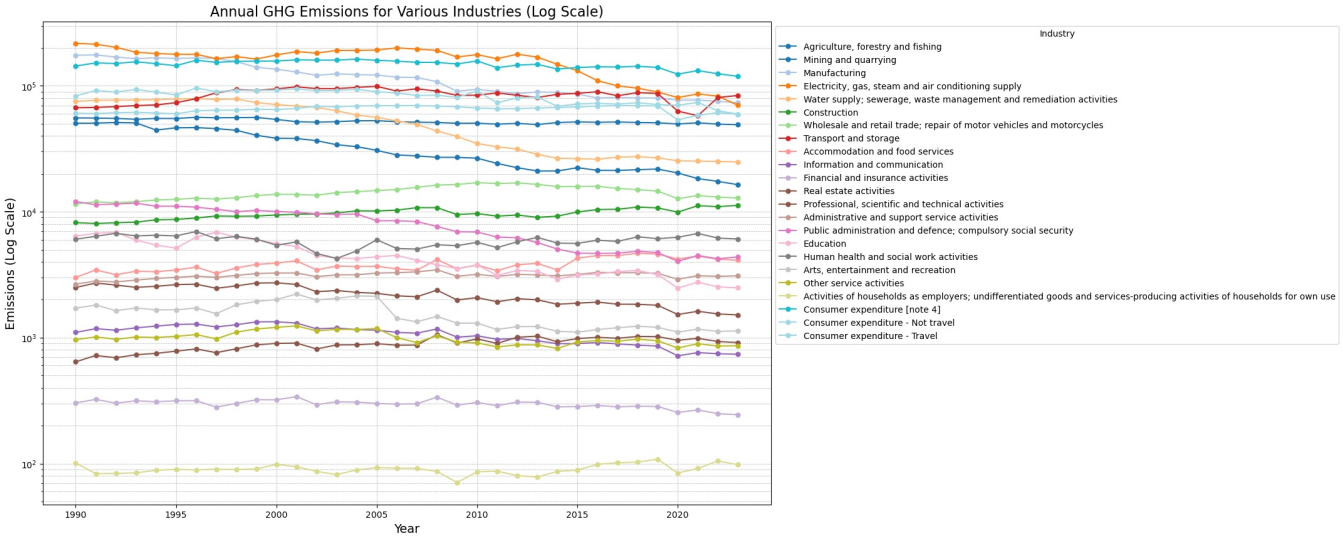


In [418…

```
#Log line graph

years = table1.iloc[1:, 0].values
industry_names = table1.iloc[0, 1:-1].values


plt.figure(figsize=(15, 10))

#tab20 for disntict coloours
colormap = cm.get_cmap('tab20', len(industry_names))

# Loop through each industry to plot the data
for industry_index, industry in enumerate(industry_names):
    emissions_data = table1.iloc[1:, industry_index + 1].values

    # Plot the emissions data for the industry with a log scale on the y-axis
    plt.plot(years, emissions_data, label=industry, marker='o', linestyle='-', markersize=5, color=colormap(ind

#log scale for y-axis
plt.yscale('log')

plt.title('Annual GHG Emissions for Various Industries (Log Scale)', fontsize=16)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Emissions (Log Scale)', fontsize=14)
plt.grid(True, which="both", linestyle='--', linewidth=0.5)
```

```
plt.legend(loc='upper left', bbox_to_anchor=(1, 1), title="Industry", fontsize=10)

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

```
/var/folders/qx/yygks_tn4_32lm3xlq53x1pc0000gn/T/ipykernel_38548/2882286533.py:10: MatplotlibDeprecationWarning
: The get_cmap function was deprecated in Matplotlib 3.7 and will be removed two minor releases later. Use ``ma
tplotlib.colormaps[name]`` or ``matplotlib.colormaps.get_cmap(obj)`` instead.
  colormap = cm.get_cmap('tab20', len(industry_names))
```



In [ ]:

In [419…

```
table1
```

Out[419]:

	SIC code	A	B	C	D	E	F	G	H	I	...	C
0	Industry name	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning s...	Water supply; sewerage, waste management and r...	Construction	Wholesale and retail trade; repair of motor ve...	Transport and storage	Accommodation and food services	...	Public administration and defence compulsory ..
1	1990	55690.3	50452.8	175212.7	217602.2	75316.3	8201.2	11560.8	66985.3	3015.3	...	12058.4
2	1991	55492.6	50608.3	176254.4	214317.5	76852.7	8072	11984.1	67438.6	3453.5	...	11381.2
3	1992	55149	51199.3	169388.2	202469.2	76959.1	8188.3	11789.9	68438.5	3163.4	...	11523.4
4	1993	54319.9	50807.1	164365	184997.2	77275.6	8275.1	12032.9	69735.2	3371.4	...	11710.9
5	1994	55129.7	44588.4	167285.9	180721.6	77637.9	8633.8	12403.1	70659.2	3342.3	...	11071.3
6	1995	55061.9	46451.4	165344.1	178162.0	78712	8695.3	12560.6	73508.2	3450.7	...	11079.5
7	1996	56343.6	46633.6	166622.3	177960.6	79209.4	8923.8	12825.8	79135.4	3636.1	...	10890.1
8	1997	55810	45761.2	165258.4	164554.2	77982.7	9263.6	12631.7	88159.2	3237.9	...	10462.7
9	1998	55950.8	44399.3	155653.0	170815.5	78691.1	9207.7	12926.4	93464.0	3581	...	10028.0
10	1999	56176.1	40501.6	140696.8	163207.5	73742.9	9246.5	13464.5	91941.5	3819.3	...	10260.4
11	2000	54133.6	38335.6	135627.4	176005.8	71060.5	9440.5	13774.1	94607.7	3913.4	...	10046.3
12	2001	51955.6	38210.6	128931.3	186969.6	69110.6	9566.7	13738.1	98397.8	4077.2	...	9921.5
13	2002	51495.5	36722.5	121456.3	182100.7	67450.9	9608.6	13544.7	95135.4	3457.7	...	9629.5
14	2003	51950.4	34092.4	124723.3	191163.6	63375.2	9791.5	14202.6	95101.3	3705	...	9518.5
15	2004	52813.4	32909.3	122868.9	191097.4	58712.5	10170.6	14461.8	97343.8	3673	...	9605.4
16	2005	52934	30738.2	121871.7	192574.6	56104.9	10152	14753.1	99137.9	3694.7	...	8493.2
17	2006	51899.8	28291.7	117269.7	200397.5	52739.5	10309.7	15013	90909.2	3518.8	...	8486.5
18	2007	51473.1	27770.4	116409.3	195947.3	49262.6	10785.7	15656.8	94883.2	3433.4	...	8360.2
19	2008	51240.7	27088.3	108134.2	191548.8	43934.9	10784.4	16297.4	90840.5	4178.1	...	7647
20	2009	50442.8	27056.8	90639.6	169738.9	39631	9504.6	16483.2	84039.5	3525.6	...	6938.5
21	2010	50502.1	26661.9	93988	177018.4	34835.1	9682.6	17024.2	84360.1	3773.8	...	6907.7
22	2011	49647.4	24205.7	90258	164156.6	32853.5	9235.3	16767.9	87918.1	3403.6	...	6299.3
23	2012	50368.8	22379.2	86759.9	178480.3	31469.9	9426.8	16946	84115	3794.9	...	6217.2
24	2013	49299.1	21096.9	89442.9	169292	28566.7	9033.9	16508.8	80876.9	3896.3	...	5696.7
25	2014	51149.2	21086.9	88797.2	148347.2	26627.8	9236	15865.5	85705.9	3444	...	5057
26	2015	51774.5	22402.3	86036.2	131922.4	26416.7	9971.5	15912.8	87193.9	4289.6	...	4693.1
27	2016	51397.4	21347.1	80201.2	109990	26154.7	10417.8	15949.4	89850.6	4486.3	...	4674.1
28	2017	51685.1	21287.7	80871.8	100019.5	27149.1	10474.8	15320.7	83486.7	4484.6	...	4685.5
29	2018	51227.2	21606.2	80321.5	95782.4	27404.6	10903.5	14993.8	88349.5	4685.3	...	4865.3
30	2019	50889.8	21858.1	80113	89388.3	26787.1	10734.7	14592.1	86998.4	4618	...	4739.5
31	2020	49836	20321.3	77060.3	80736.3	25434.7	9941	12705.2	63020.4	4213	...	4038.4
32	2021	50909.4	18346.2	77092.3	86311	25252.1	11180.6	13471.4	57961.1	4446.4	...	4494.4
33	2022	49682.9	17433.4	75449.5	82854.7	25109.3	10990.5	13084.3	80891	4203.9	...	4233.5
34	2023	49176.8	16415.6	73496.8	70134.5	24868.4	11243.8	12855.7	83875.1	4122.1	...	4380.0

35 rows × 25 columns

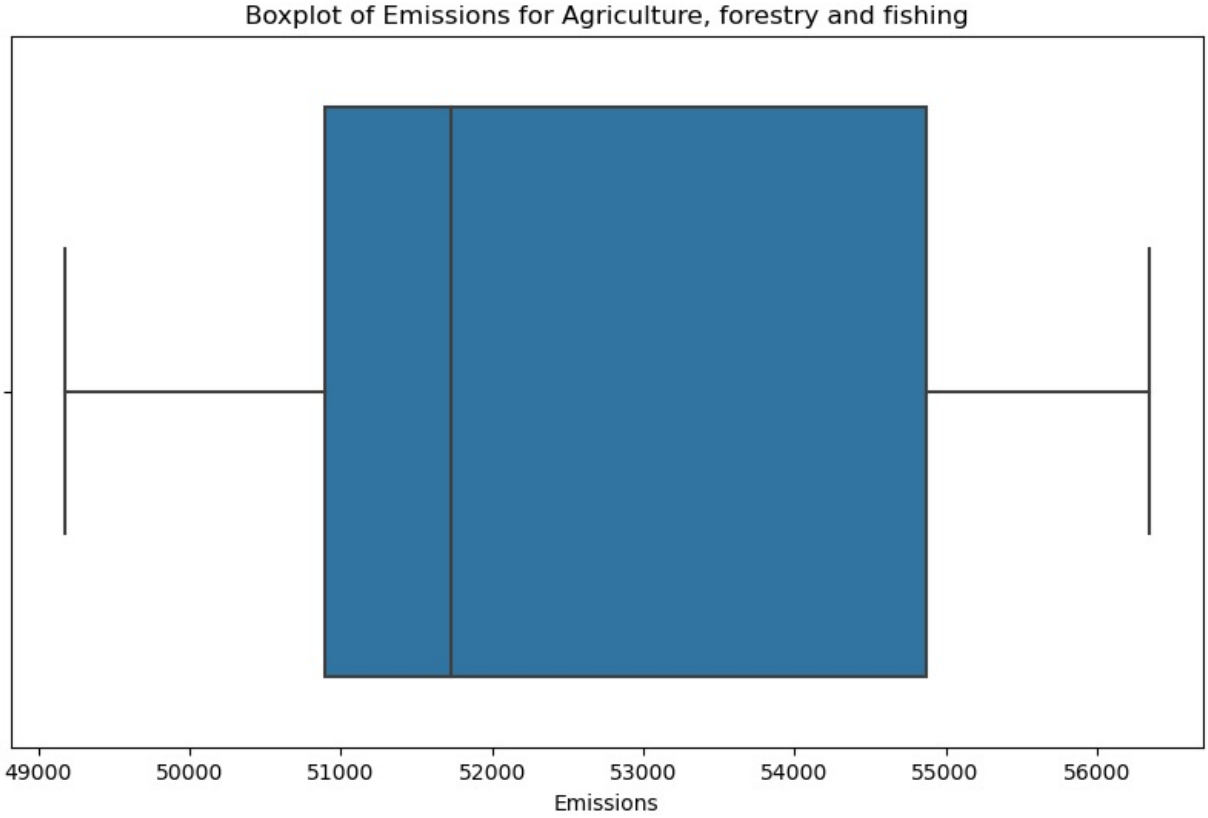
In [ ]:

In [420…

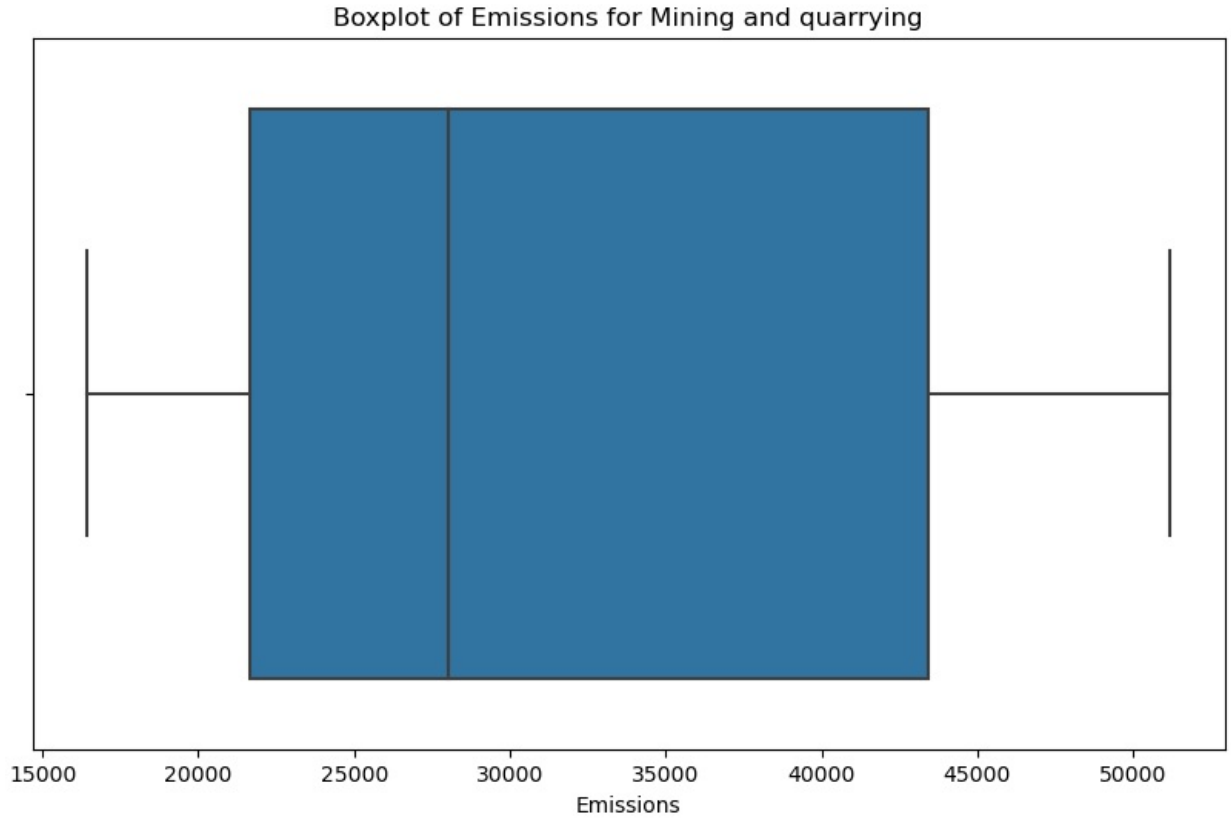
```
#Boxplots
#Reference: (Mondal, 2024)
years = table1.iloc[1:, 0]

# Setting a threshold for outliers = 2
outlier_threshold = 2

# Loop through all industries
for col in table1.columns[1:]:
    industry_name = table1.iloc[0, table1.columns.get_loc(col)]
    emissions_data = table1.iloc[1:, table1.columns.get_loc(col)].astype(float)

    # Calculating Z-scores for the emissions data
    z_scores = zscore(emissions_data)

    # Finding outliers based on Z-scores
    outliers = []
```

```
    for year, emission, z in zip(years, emissions_data, z_scores):
        if abs(z) > outlier_threshold:
            outliers.append((year, emission, z))

    # Displaying outliers
    if outliers:
        print(f"Outliers in {industry_name}:")
        for year, emission, z in outliers:
            print(f"Year: {year}, Emissions: {emission}, Z-score: {z:.2f}")
    else:
        print(f"No outliers found for {industry_name}.")

    # Plotting a box plot for emissions data of the industry
    plt.figure(figsize=(10, 6))
    sns.boxplot(x=emissions_data)
    plt.title(f"Boxplot of Emissions for {industry_name}")
    plt.xlabel("Emissions")
    plt.show()
```

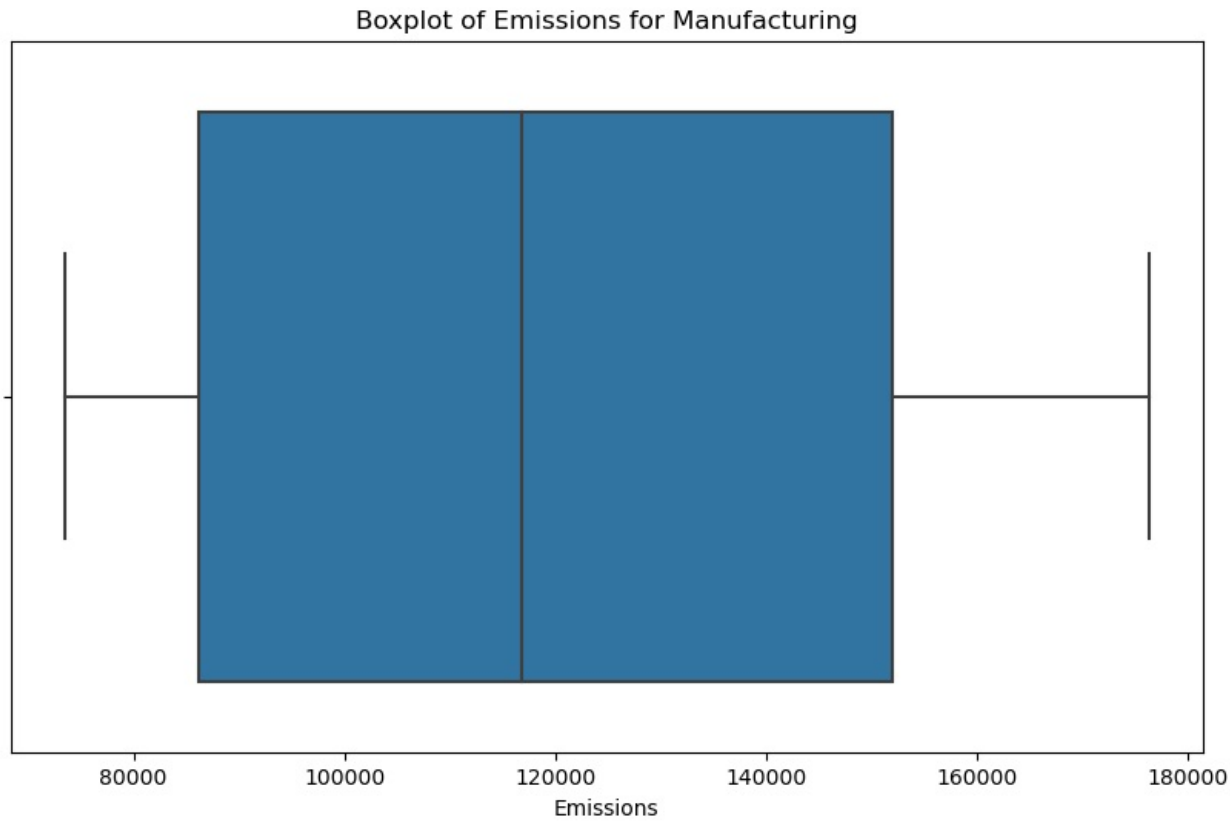
```
No outliers found for Agriculture, forestry and fishing.
```



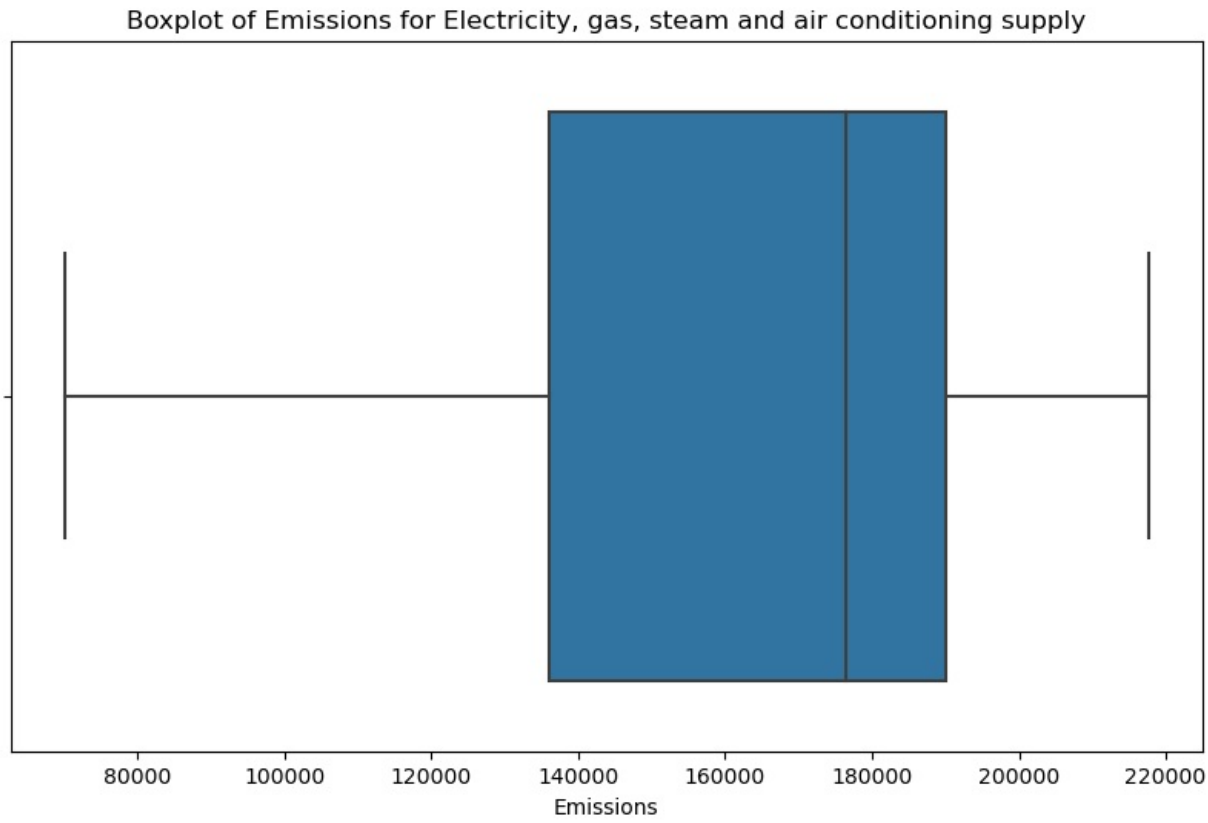

```
No outliers found for Mining and quarrying.
```



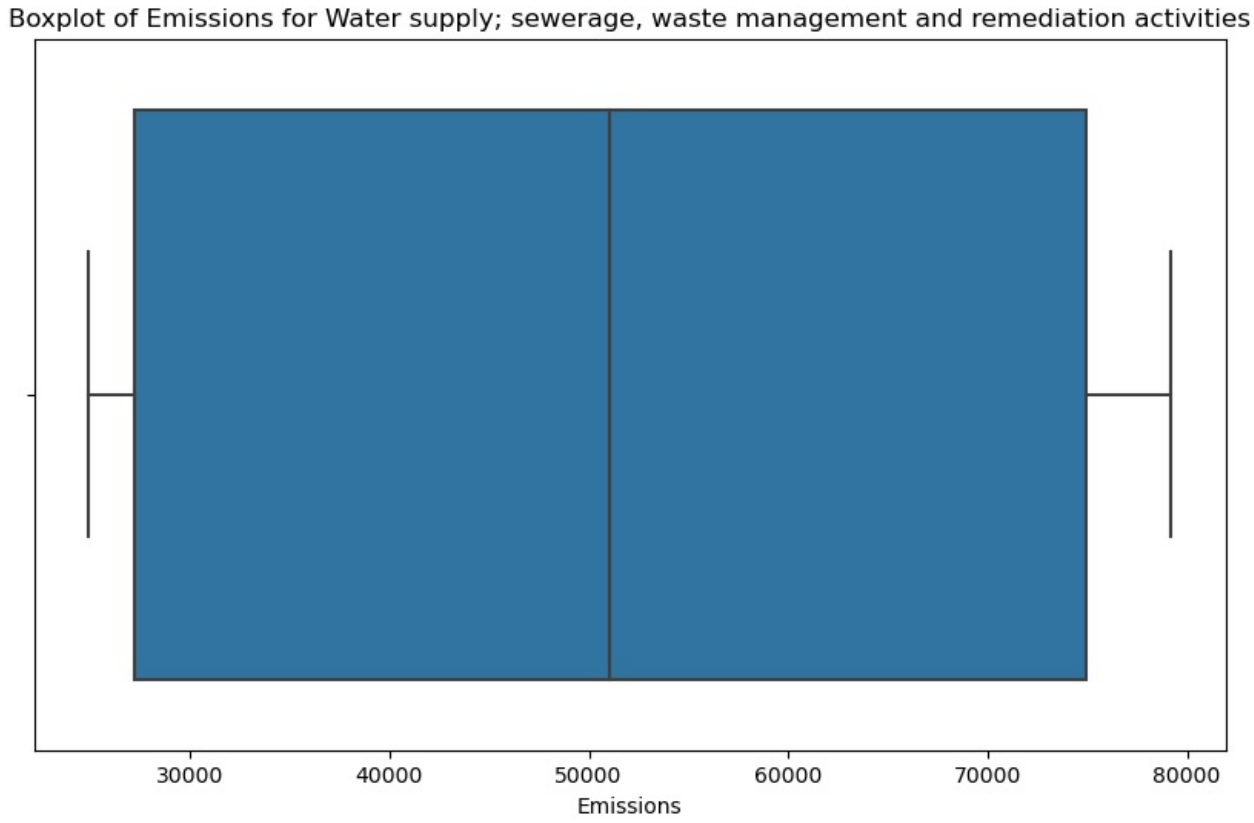
No outliers found for Manufacturing.



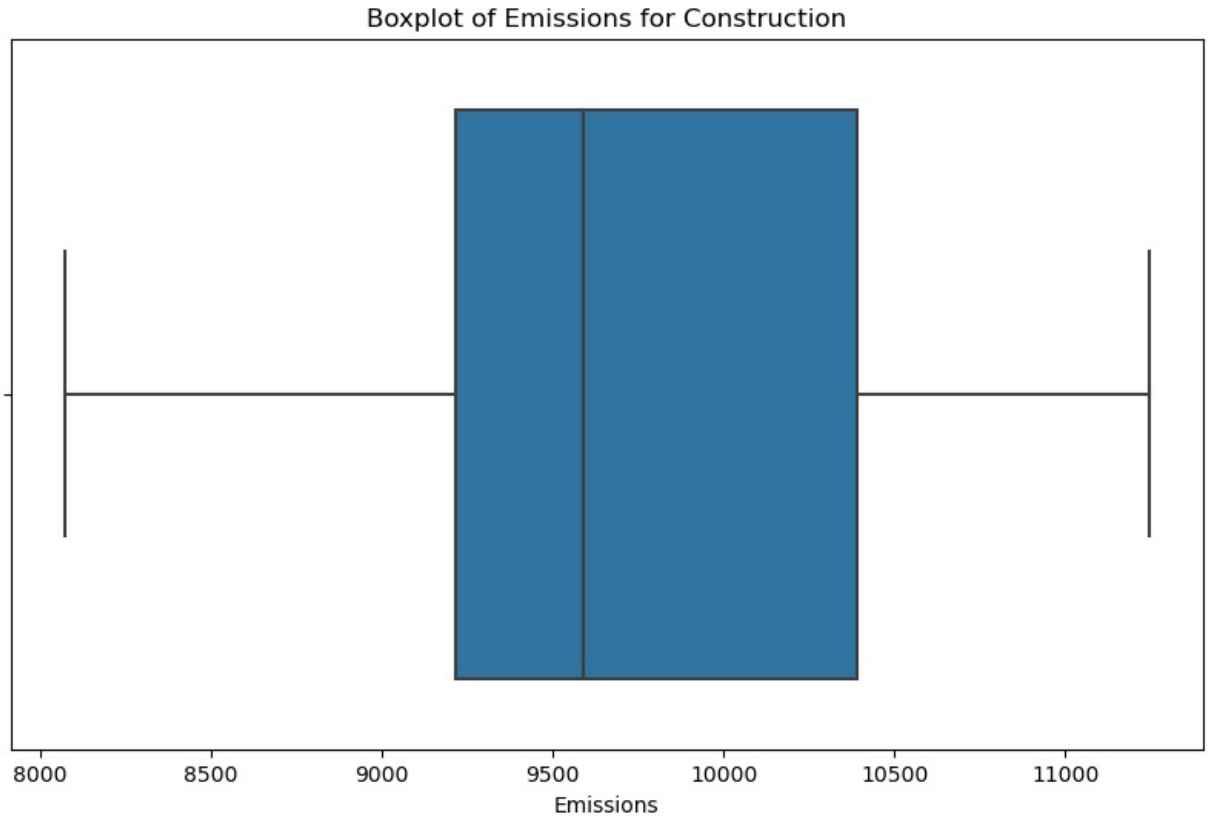
Outliers in Electricity, gas, steam and air conditioning supply:
Year: 2023, Emissions: 70134.5, Z-score: -2.11



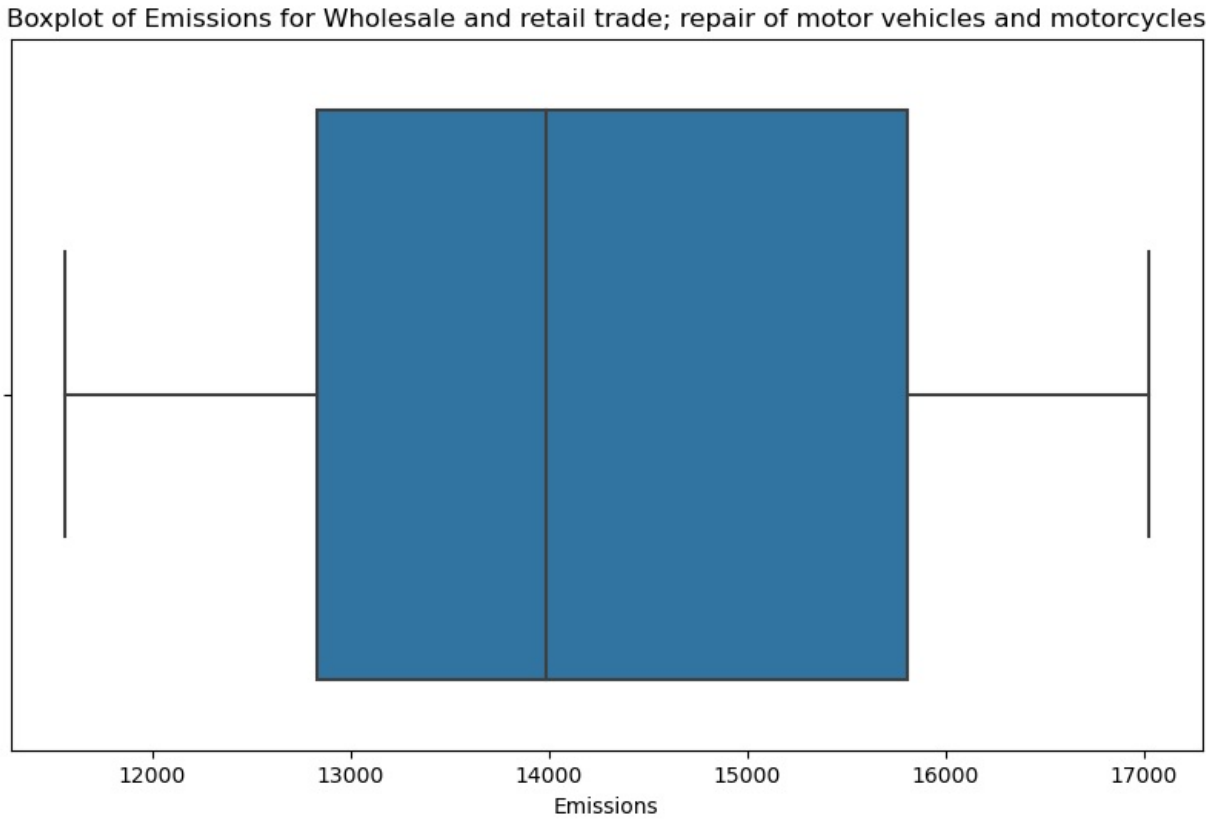
No outliers found for Water supply; sewerage, waste management and remediation activities.



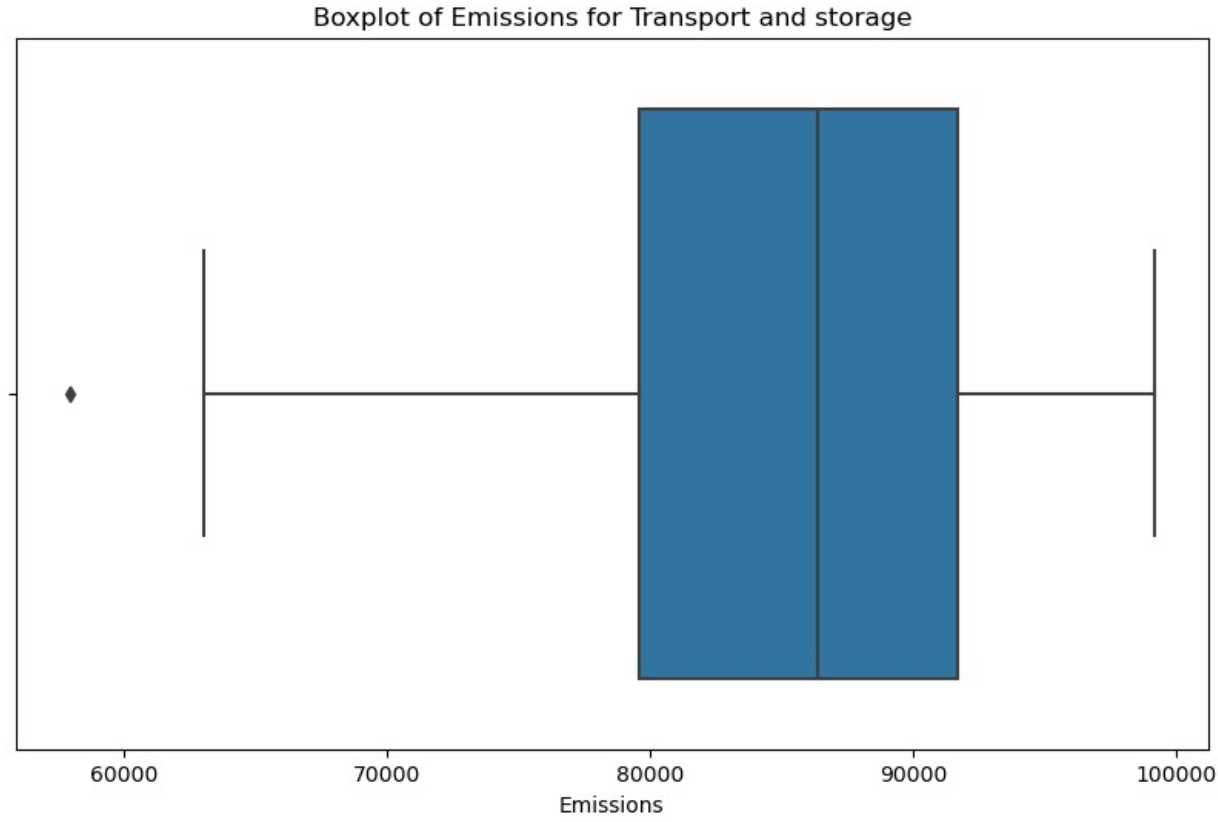
No outliers found for Construction.



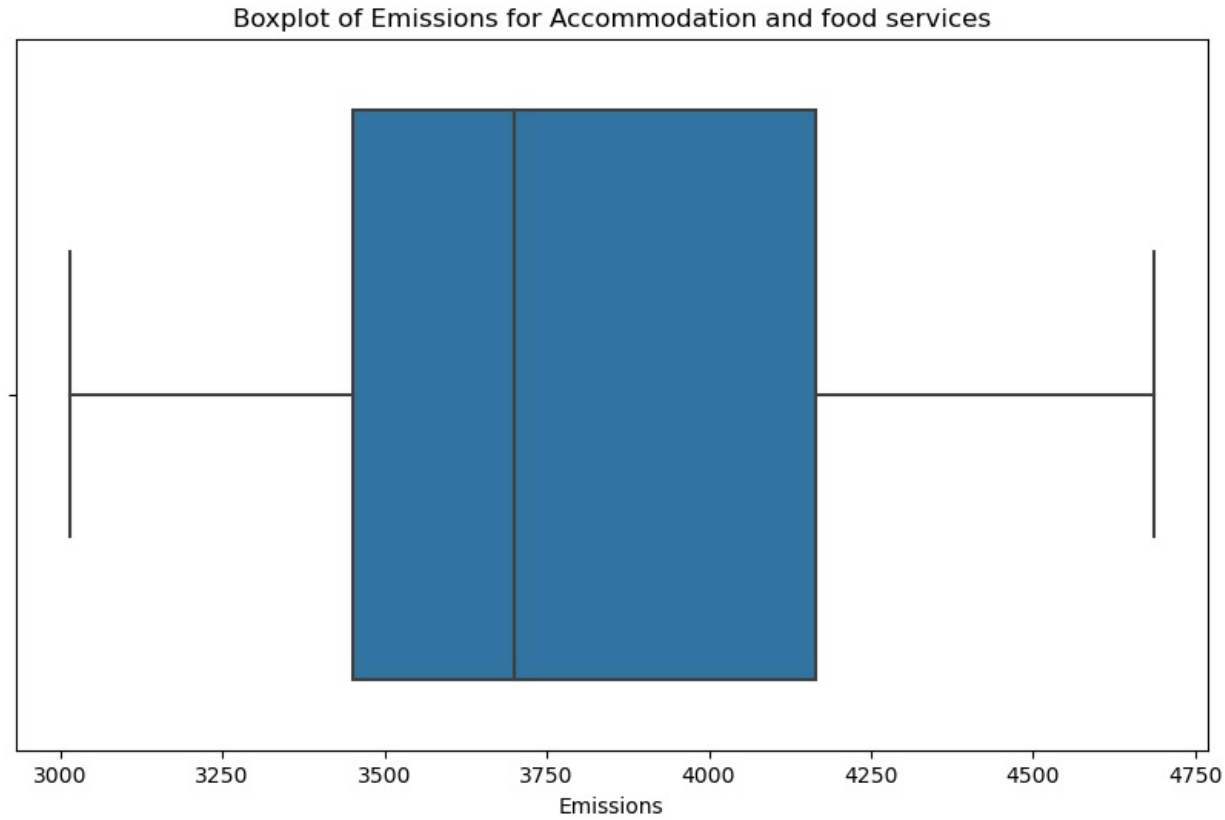
No outliers found for Wholesale and retail trade; repair of motor vehicles and motorcycles.



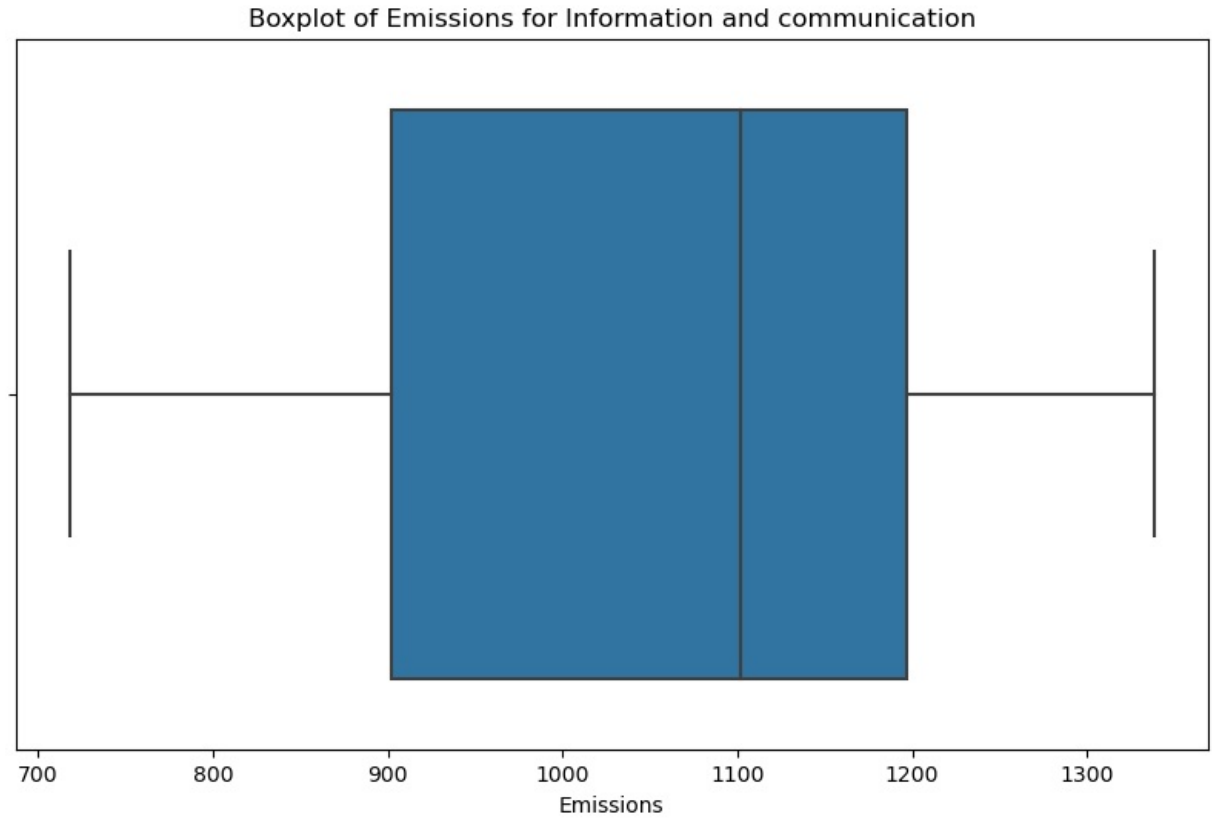
Outliers in Transport and storage:
Year: 2021, Emissions: 57961.09999999999, Z-score: -2.42



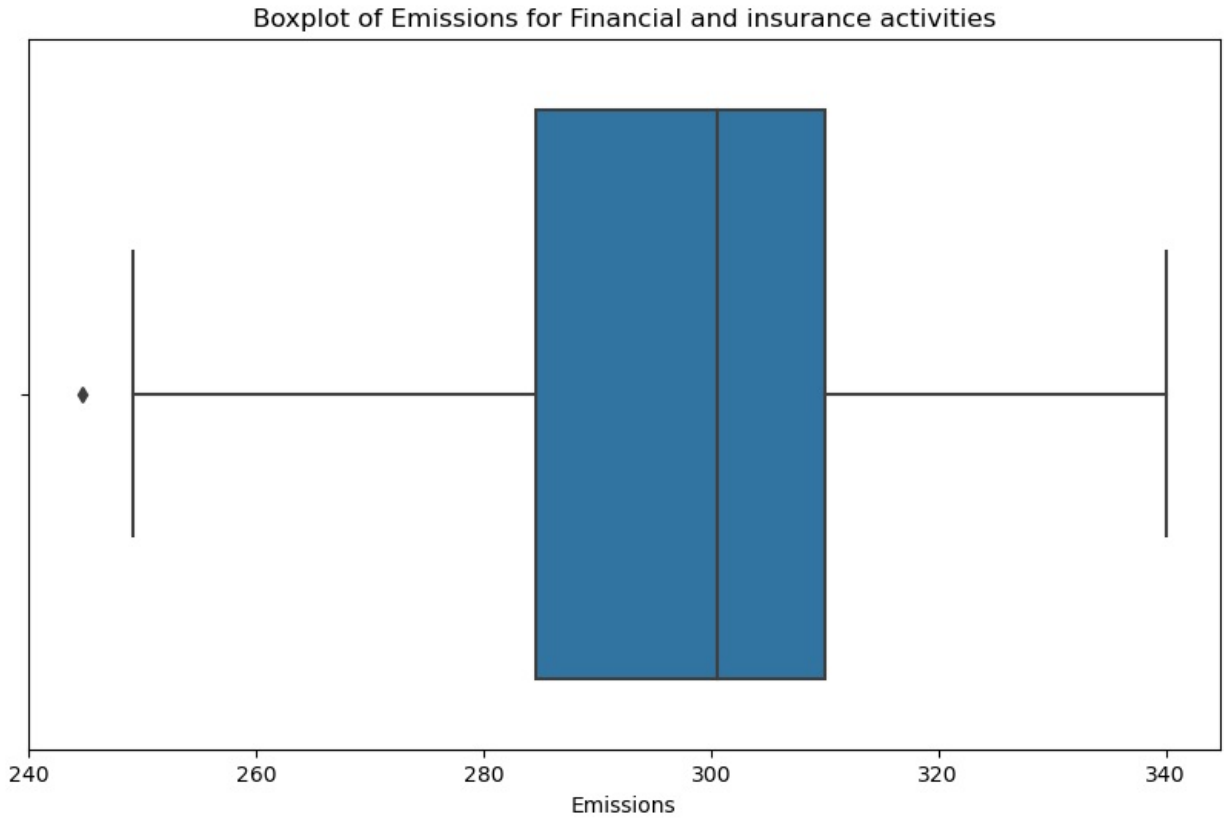

```
Outliers in Accommodation and food services:
Year: 2018, Emissions: 4685.300000000001, Z-score: 2.03
```



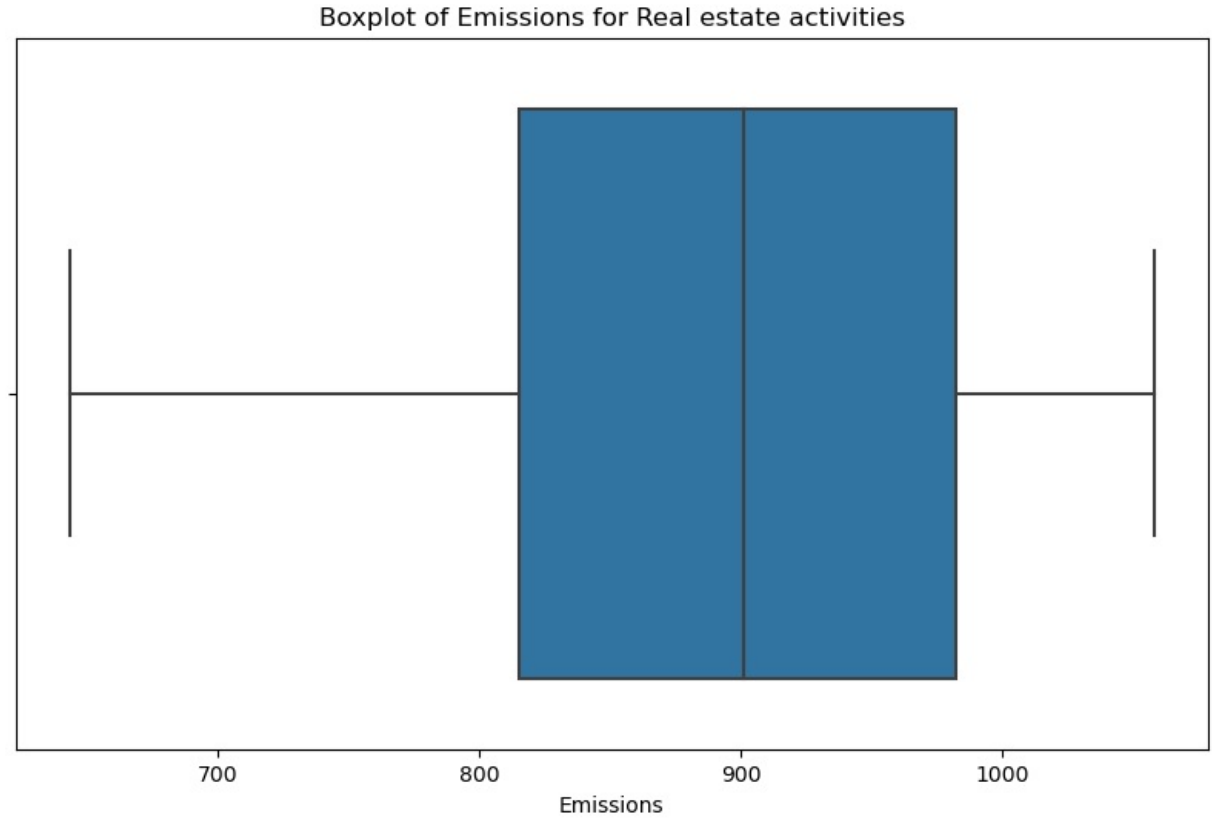

```
No outliers found for Information and communication.
```



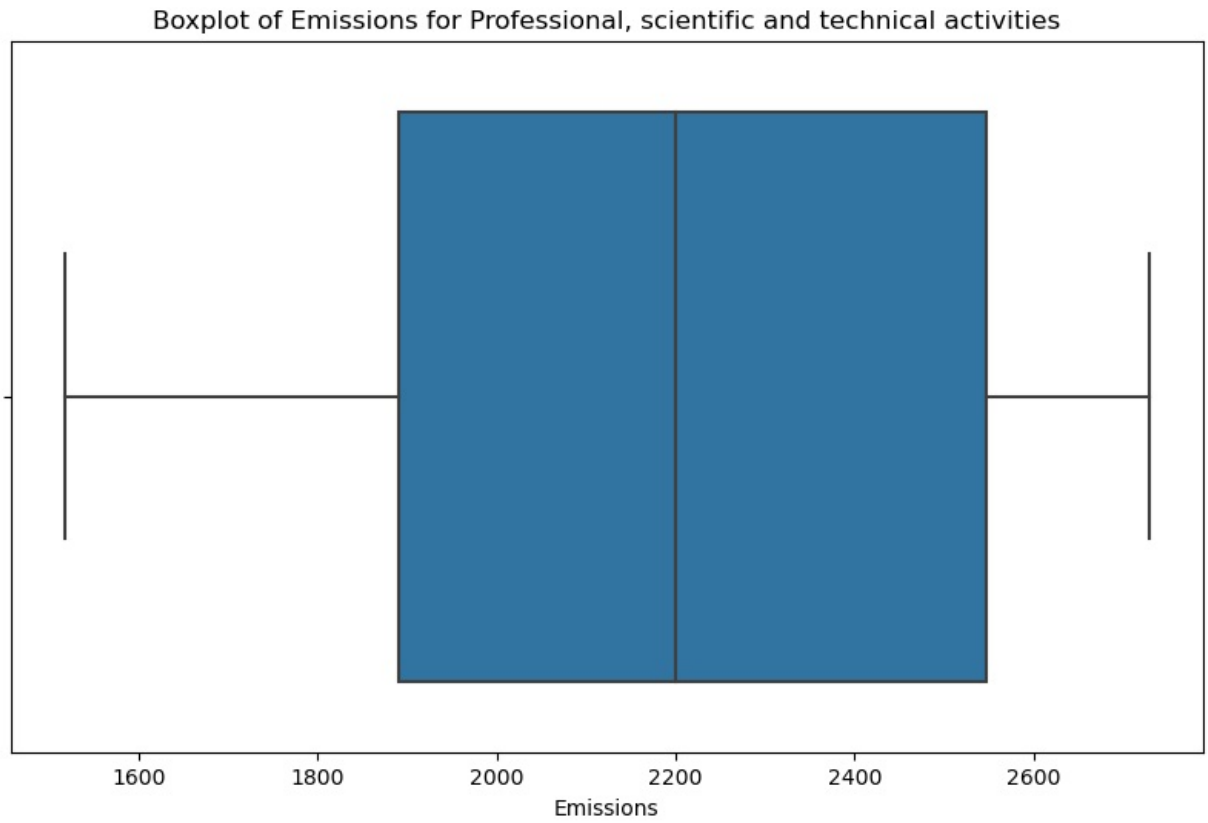

```
Outliers in Financial and insurance activities:
Year: 2022, Emissions: 249.20000000000002, Z-score: -2.21
Year: 2023, Emissions: 244.70000000000002, Z-score: -2.42
```



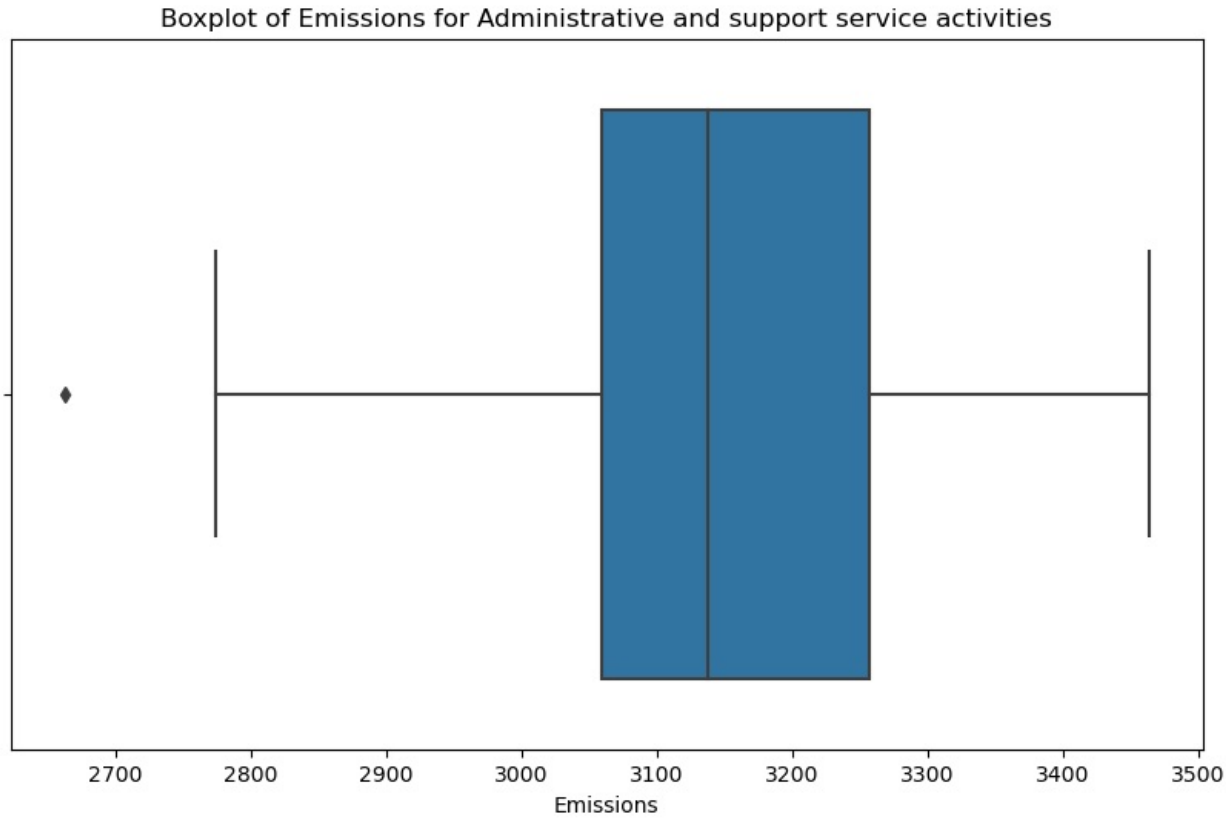

```
Outliers in Real estate activities:
Year: 1990, Emissions: 643.7, Z-score: -2.34
```



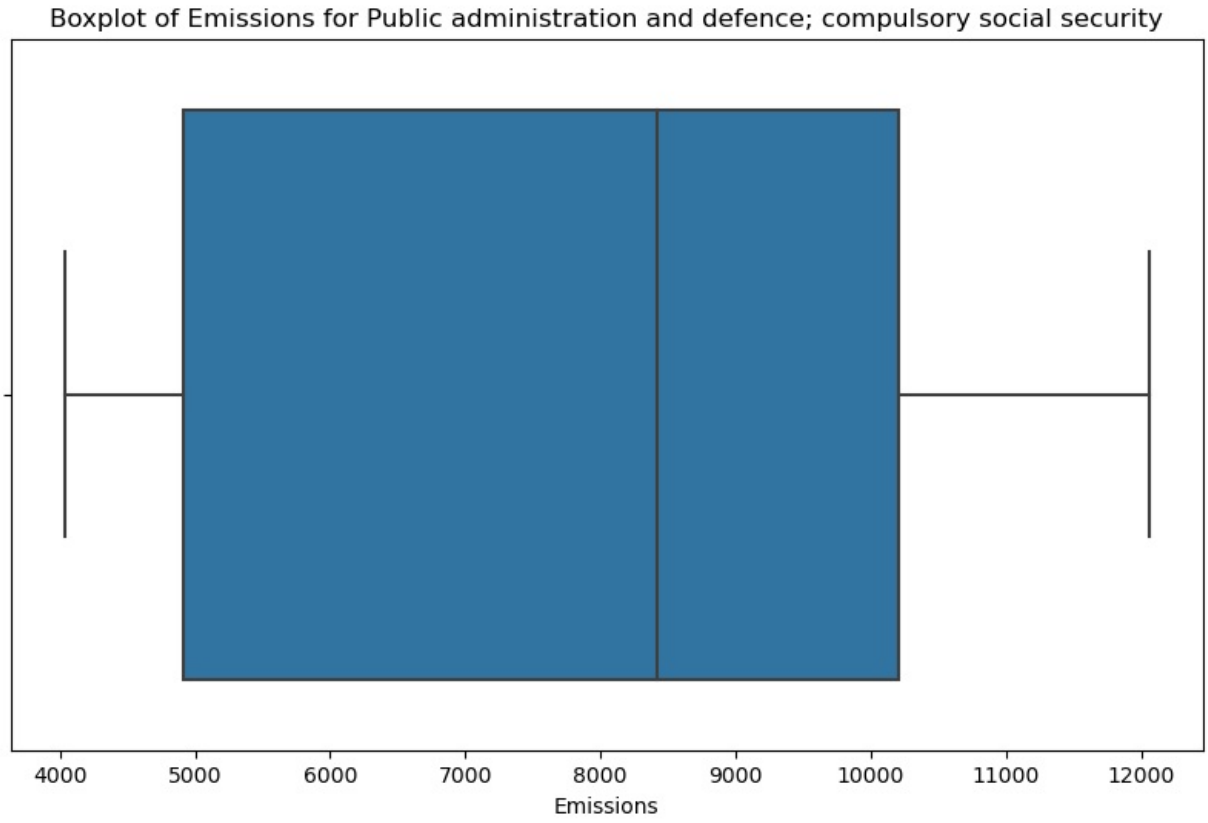
No outliers found for Professional, scientific and technical activities.



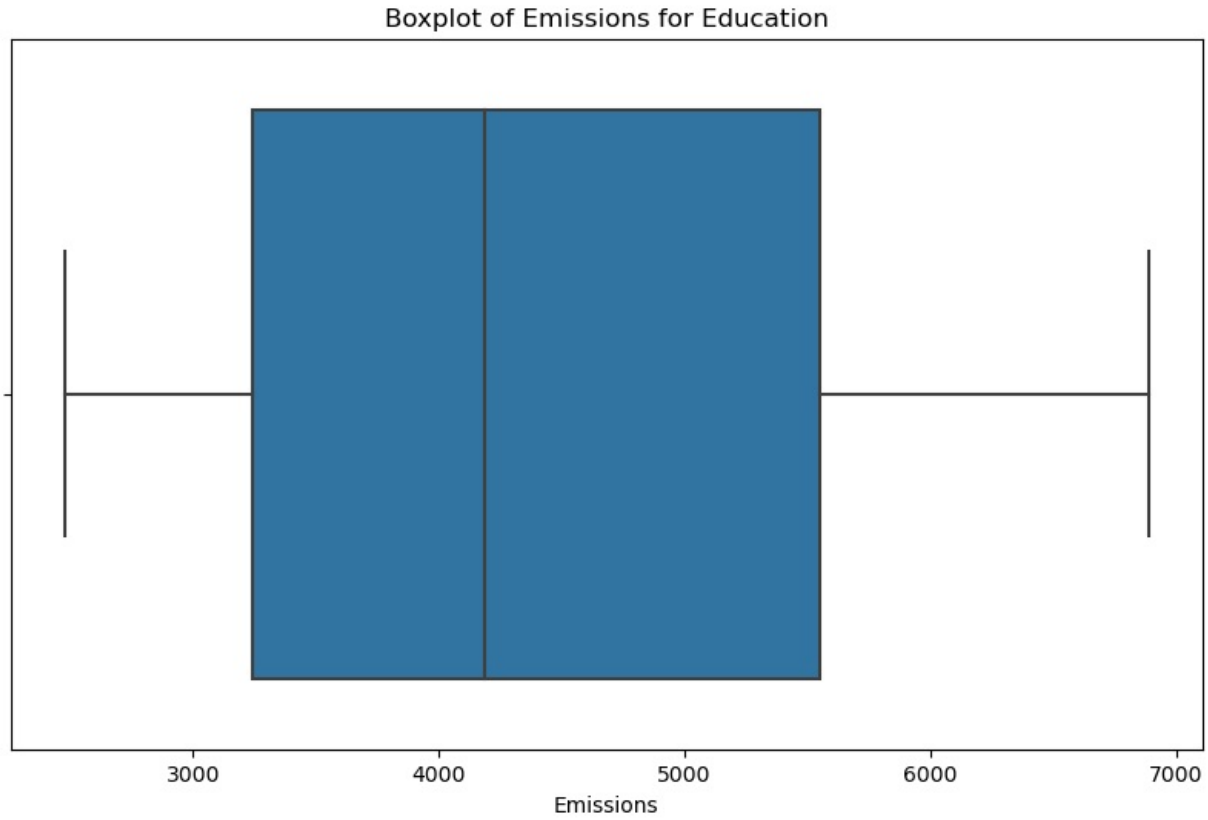

```
Outliers in Administrative and support service activities:
Year: 1990, Emissions: 2662.3, Z-score: -2.70
Year: 1992, Emissions: 2773.1, Z-score: -2.05
Year: 2008, Emissions: 3463.6, Z-score: 2.04
```



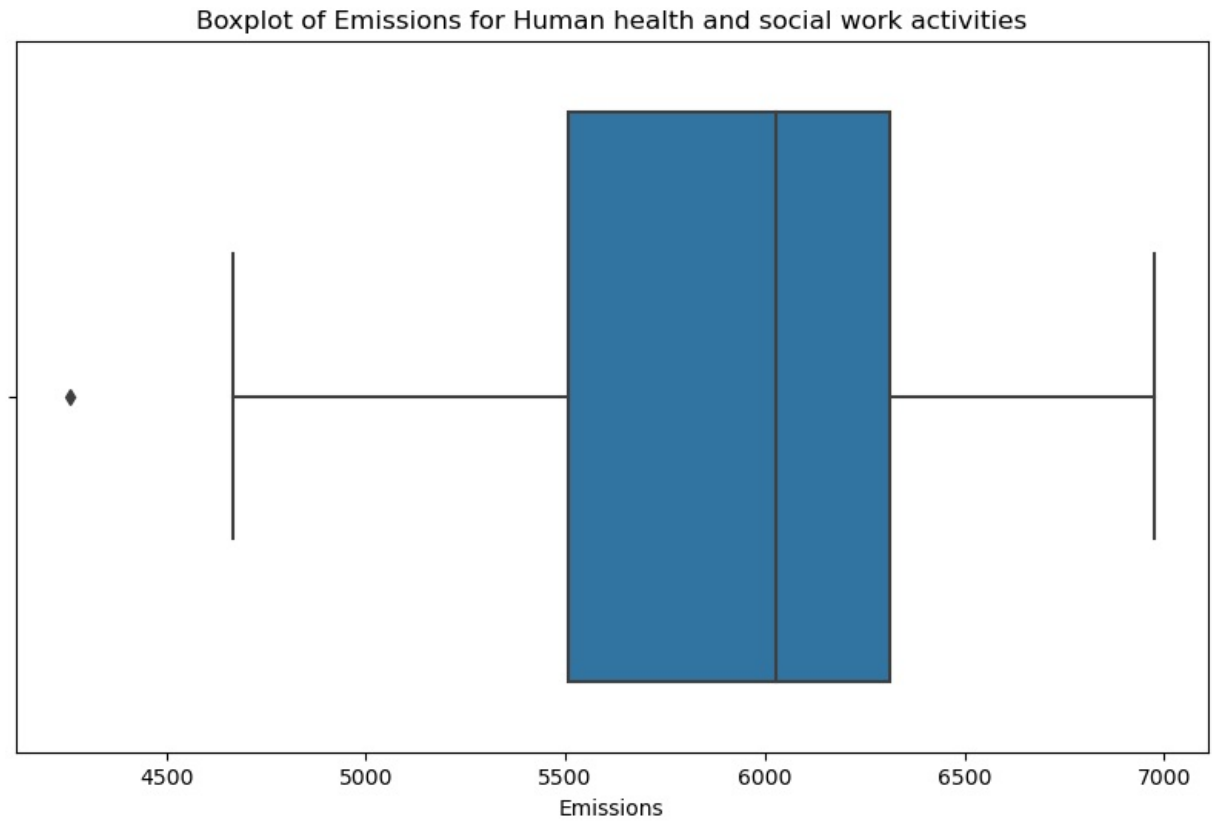
No outliers found for Public administration and defence; compulsory social security.



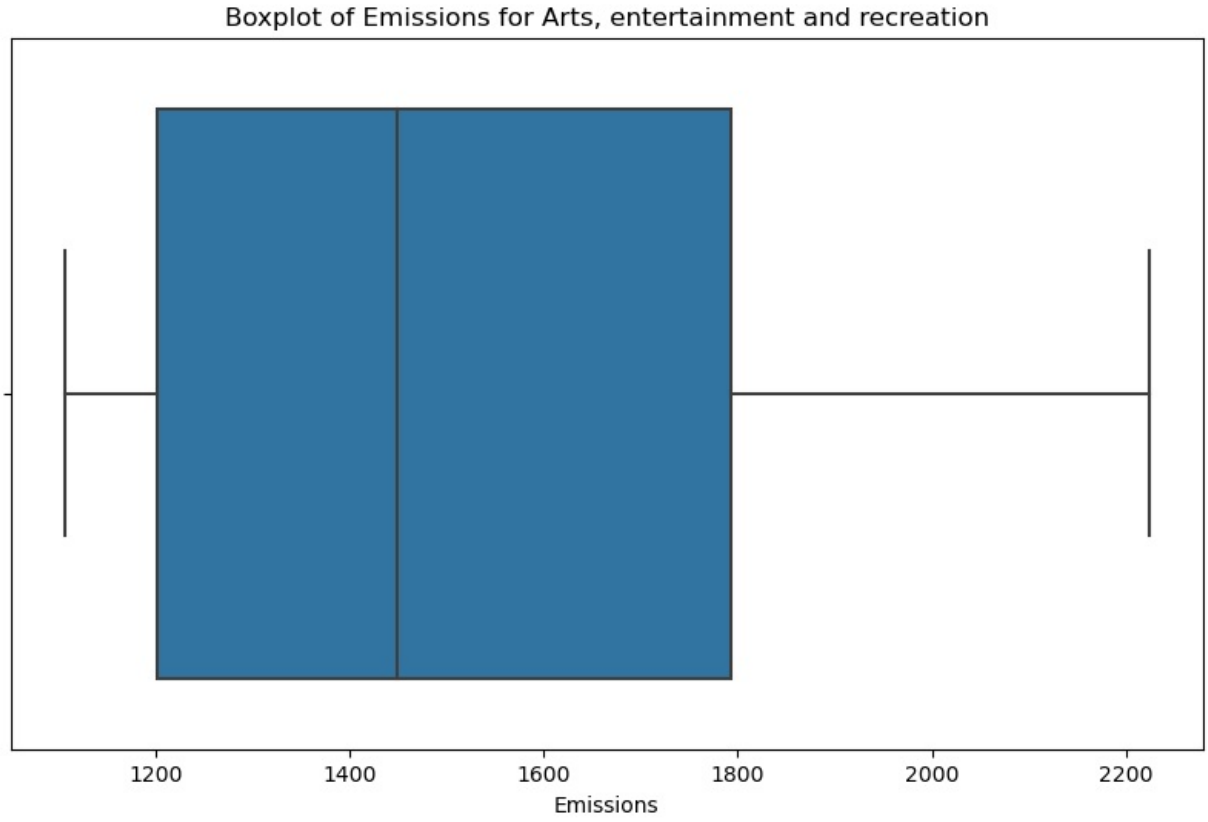
No outliers found for Education.



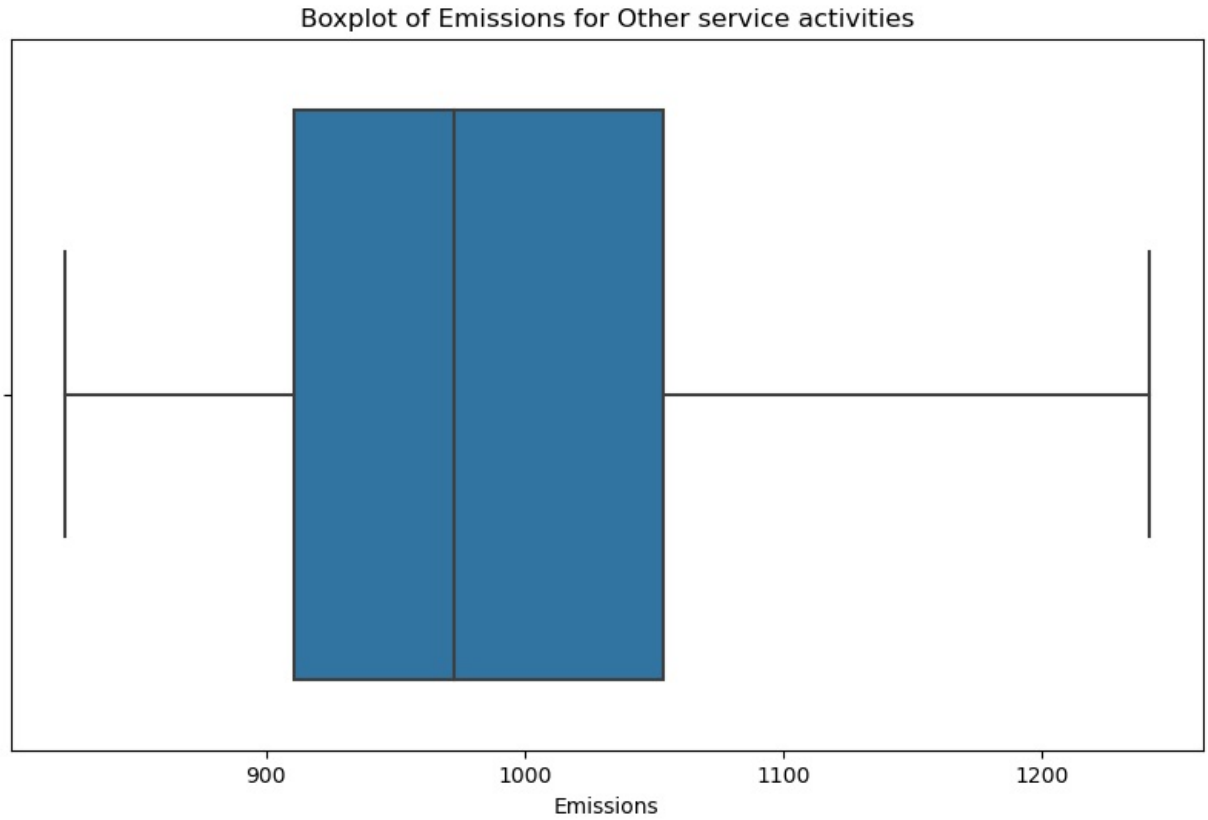

```
Outliers in Human health and social work activities:
Year: 2003, Emissions: 4258.400000000001, Z-score: -2.65
```



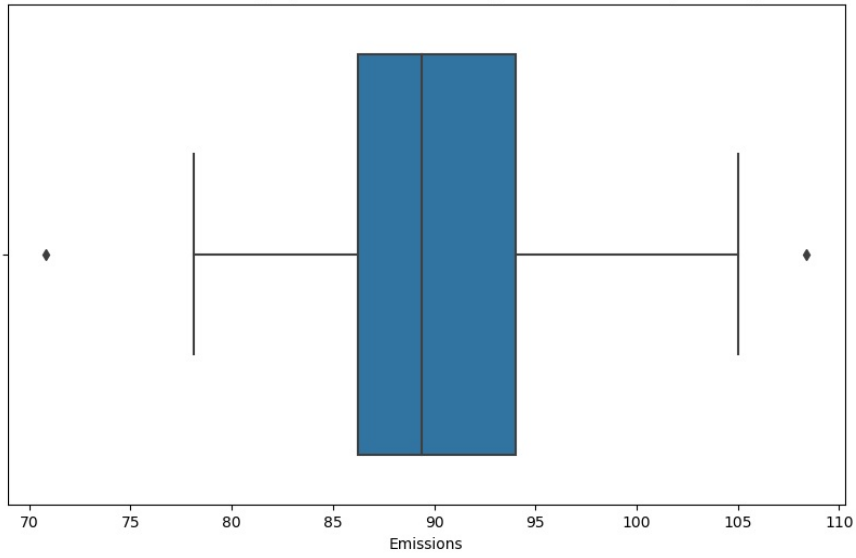
No outliers found for Arts, entertainment and recreation.



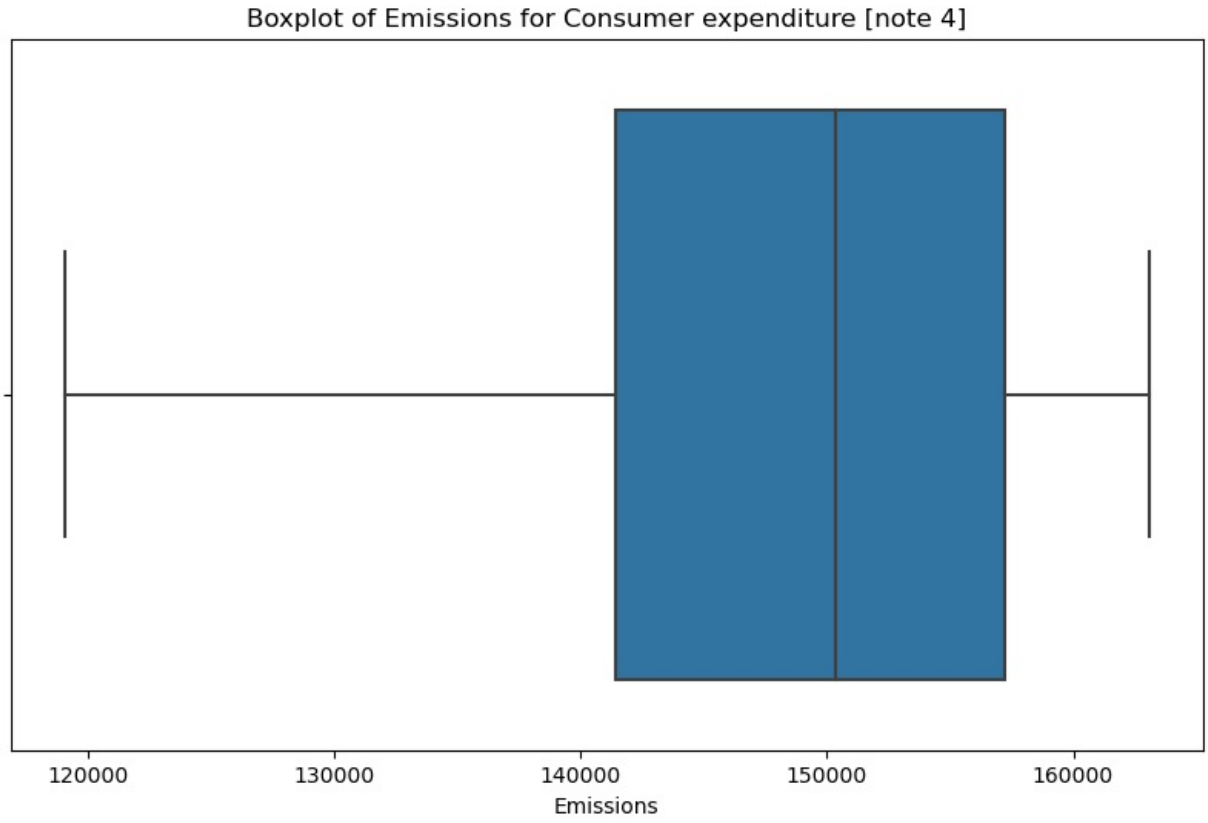

```
Outliers in Other service activities:
Year: 2001, Emissions: 1241.2, Z-score: 2.14
```



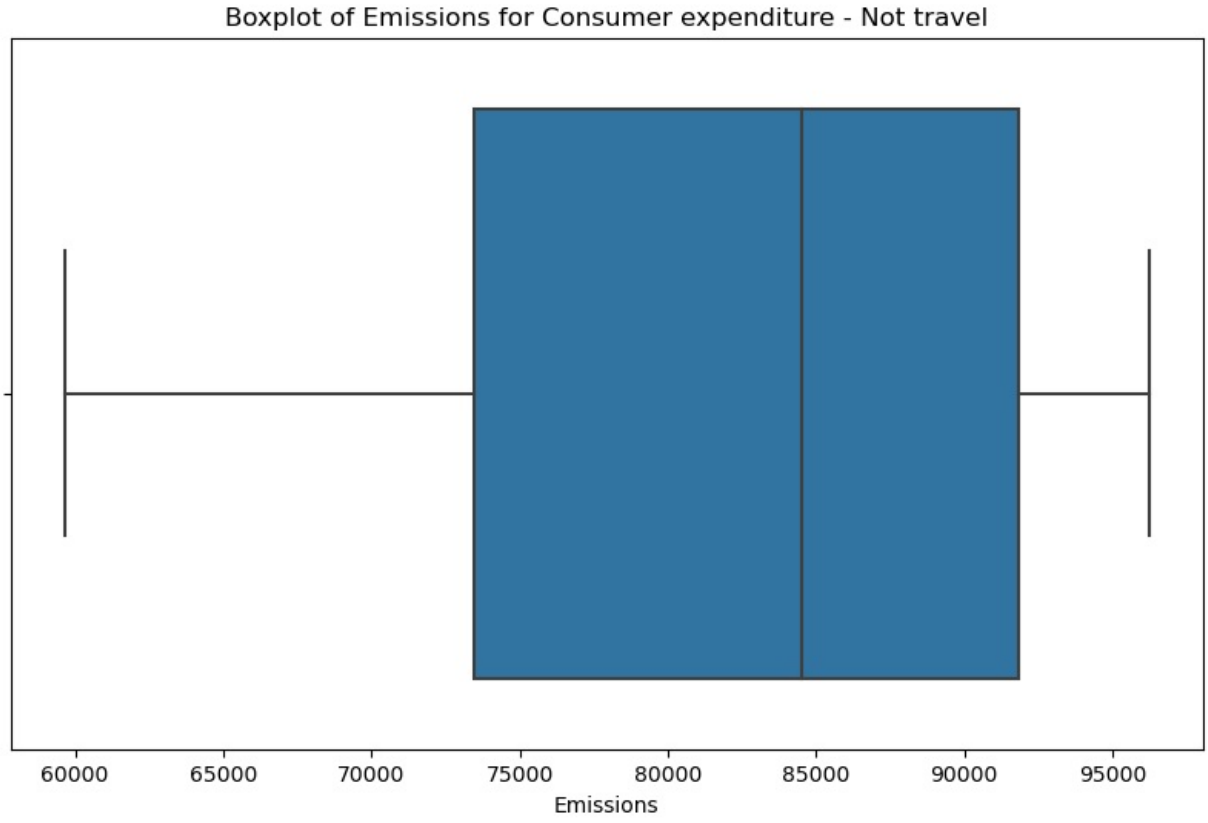
Outliers in Activities of households as employers; undifferentiated goods and services-producing activities of households for own use:
Year: 2009, Emissions: 70.8, Z-score: -2.48
Year: 2019, Emissions: 108.4, Z-score: 2.29



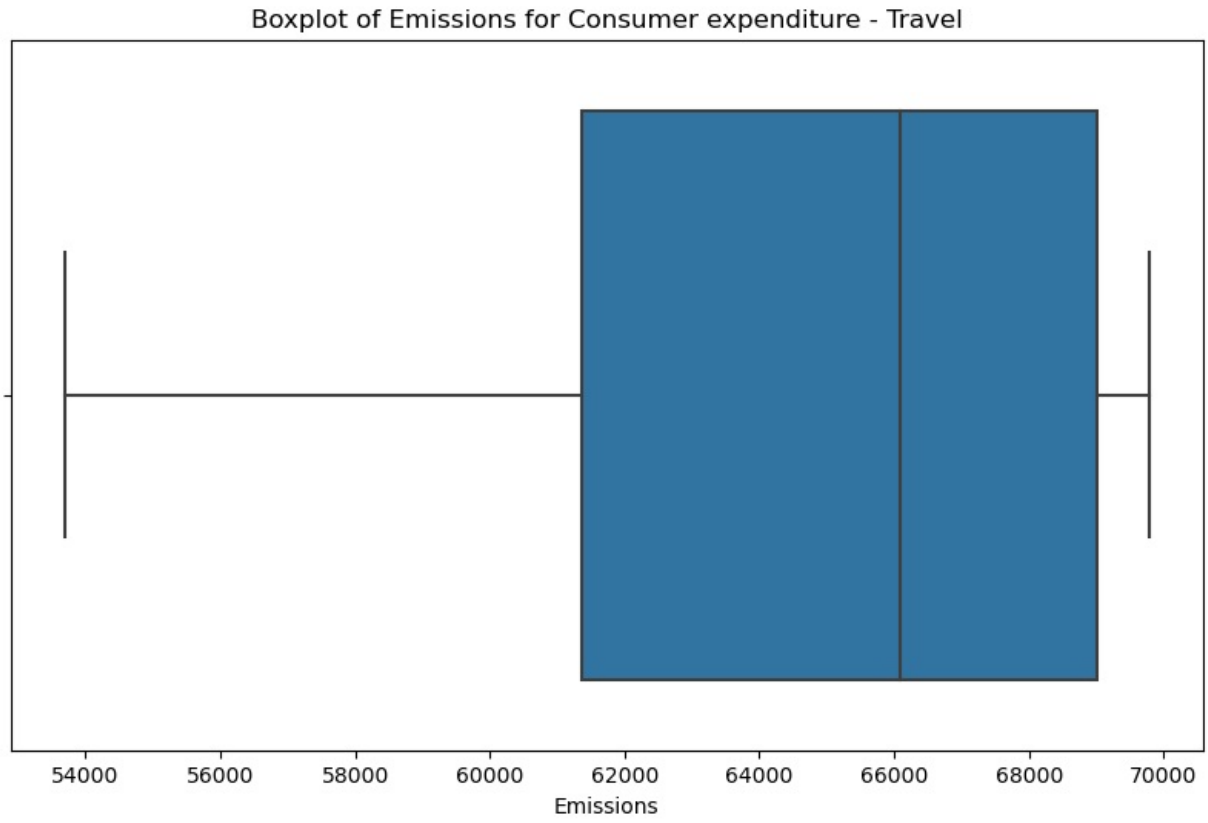
Outliers in Consumer expenditure [note 4]:
Year: 2020, Emissions: 123858.10000000002, Z-score: -2.16
Year: 2022, Emissions: 124667.7, Z-score: -2.09
Year: 2023, Emissions: 119071.5, Z-score: -2.59



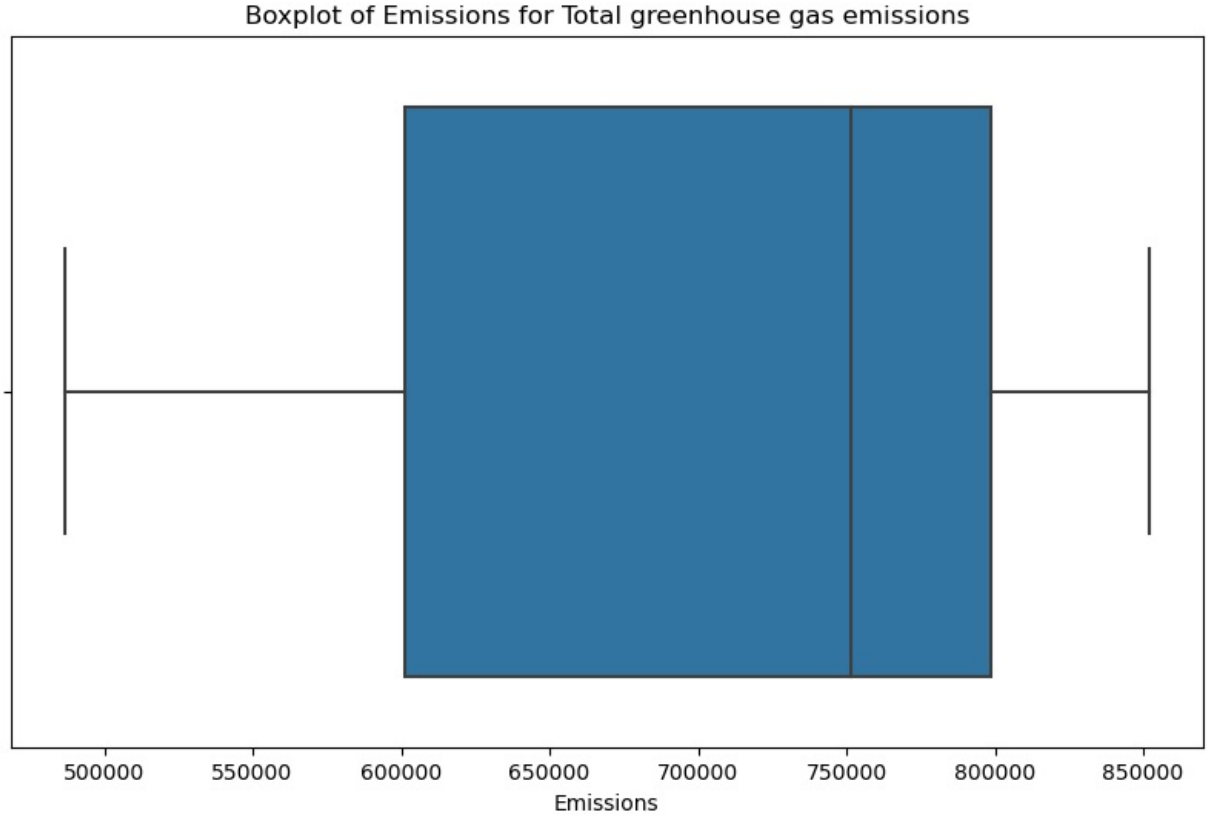
Outliers in Consumer expenditure - Not travel:
Year: 2023, Emissions: 59659.7, Z-score: -2.33



Outliers in Consumer expenditure - Travel:
Year: 2020, Emissions: 53696.3, Z-score: -2.82




```
No outliers found for Total greenhouse gas emissions.
```




```
In [421… table2
```

Out[421]:

	SIC code	A	A.1	A.2	B	B.1	B.2	B.3	B.4	C	...	R.1	R.2	R.3
0	SIC(07) group	1	2	3	5	6	7	8	9	10.1	...	91	92	93
1	Industry name	Products of agriculture, hunting and related s...	Products of forestry, logging and related serv...	Fish and other fishing products; aquaculture p...	Mining of coal and lignite	Crude petroleum and natural gas	Mining of metal ores	Other mining and quarrying products	Mining support services	Processing and preserving of meat and producti...	...	Library, archive, museum and other cultural se...	Gambling and betting services	Sporting services and amusement and recreation...
2	1990	54715.1	27	948.2	25423.1	22983.5	9	1237.2	800.1	1033.2	...	182.8	267.8	1005.2
3	1991	54494.1	27.2	971.3	26059.1	22534.4	9	1244.9	760.8	1140	...	184.3	281.1	1078.9
4	1992	54119.9	28	1001	25814.5	23477.2	9.1	1203	695.5	1169.5	...	160.3	253.1	985.6
5	1993	53228.2	34.7	1057	23889.6	25204.8	9.2	1057.9	645.7	1180.2	...	164.2	262.3	1032.9
6	1994	54010.2	42.7	1076.8	15998.5	26873	9.6	997.5	709.8	1257.4	...	149.2	246.5	1003
7	1995	53906.4	33.2	1122.4	17362.9	27368.2	9.6	905	805.7	1391.4	...	144.6	242.3	1011.2
8	1996	55225.1	29.9	1088.6	16060	28798.2	7.5	913.2	854.5	1334.6	...	144	244.9	1057.6
9	1997	54706.5	36.4	1067.2	15418	28618.5	7.8	872.5	844.3	1316.3	...	127.5	220.2	964.7
10	1998	54809.8	33.7	1107.3	13103.9	29552.3	7.9	875.8	859.2	1264.2	...	157	255.2	1116.9
11	1999	55136.1	34.4	1005.6	10898.9	28494	7.2	806.2	295.3	1036.1	...	166.2	270.4	1200.7
12	2000	53193.8	34.5	905.3	9444.8	27420	7.6	1271.9	191.4	1297.3	...	170.6	282.5	1250.8
13	2001	51030	33.1	892.7	8478.8	28339.9	7.4	1230.5	154	1186.2	...	191.1	298.8	1326
14	2002	50626	32.7	836.8	8318.6	27352.9	7.3	837.5	206.2	1107	...	179.6	259.3	1212.5
15	2003	51128	32.8	789.6	6954.1	26056.7	7.4	879.6	194.7	1356.5	...	182.5	264.7	1266.4
16	2004	51972.8	32.6	807.9	6296.6	25590.8	8.2	826.4	187.3	1105.4	...	203.1	273.3	1310.1
17	2005	52096.1	37	801	4729.5	24878.7	8	949.9	172	1157.5	...	202.8	266.5	1316.6
18	2006	51117.5	35.7	746.6	4327.5	23023.6	8.3	815.6	116.7	1058.4	...	107.1	166.3	953.2
19	2007	50668.4	49	755.7	3616.2	23073.1	9	953.8	118.2	1055.3	...	95.9	152.3	911.5
20	2008	50477.6	32.9	730.2	3625.8	21982.8	7.3	1373.6	98.8	917.9	...	98.7	161.3	1020.6
21	2009	49706.2	30.6	706	3462.8	21993.5	5.7	1357.6	237.3	862.5	...	87.2	131	908
22	2010	49728.8	28.9	744.4	3291.8	21719.6	6.4	1519	125	905.8	...	85.1	133.4	918.6
23	2011	48891	29.1	727.3	3151.7	19873.4	6.5	1059.1	115.1	908.8	...	71	113.7	827.2
24	2012	49575.2	28.9	764.7	3100.7	18294.3	6.1	772.3	205.7	865.9	...	72.7	117.5	880.9
25	2013	48513.1	27.6	758.4	2226.9	17923.1	5.8	731.7	209.3	855.1	...	69.5	114.3	892.7
26	2014	50211.8	28.7	908.7	2231.7	17529.6	6.3	985.9	333.5	900.7	...	63.4	100.4	815.3
27	2015	50886.2	29.1	859.2	1890.2	19274.2	6.4	1061.2	170.4	880.7	...	42.5	80.5	862.2
28	2016	50522.8	30.4	844.2	939	18852.3	6.8	1234.1	314.8	818.1	...	44.9	81.7	902.5
29	2017	50792.9	29.6	862.7	918.6	18865.3	7.1	1392.3	104.3	854	...	46	81.8	913
30	2018	50378.6	28.7	819.9	890.7	18856.6	7.1	1744.1	107.7	912.8	...	48.1	83.8	930.3
31	2019	50136.5	28.8	724.6	870.3	18903.6	6.9	1952.6	124.7	959.1	...	46.8	81.3	915.7
32	2020	49101	23.5	711.6	858.1	17269.6	7.4	2164.8	21.4	937.7	...	38.8	74.1	856.5
33	2021	50147.1	26.1	736.1	806.9	15153.9	6.9	2312	66.4	915	...	44.5	77.1	893.2
34	2022	48951	27.3	704.6	766.1	14313.3	6.9	2314.2	32.8	896.2	...	41.4	71.9	855.3

35 rows × 133 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [422…

```
#Formatting table 2
# Drop the first row (index 0), which contains the 'SIC(07) group'
table2 = table2.drop(0).reset_index(drop=True)

# Set the second row (index 1) as the column names and drop that row
table2.columns = table2.iloc[0]
table2 = table2.drop(1).reset_index(drop=True)

# Check the dataframe to ensure the changes were applied
table2.head()
```

Out[422]:

	Industry name	Products of agriculture, hunting and related services	Products of forestry, logging and related services	Fish and other fishing products; aquaculture products; support services to fishing	Mining of coal and lignite	Crude petroleum and natural gas	Mining of metal ores	Other mining and quarrying products	Mining support services	Processing and preserving of meat and production of meat products	...	Library, archive, museum and other cultural services	Gambling and betting services	Sp se amus recr se
0	Industry name	Products of agriculture, hunting and related s...	Products of forestry, logging and related serv...	Fish and other fishing products; aquaculture p...	Mining of coal and lignite	Crude petroleum and natural gas	Mining of metal ores	Other mining and quarrying products	Mining support services	Processing and preserving of meat and producti...	...	Library, archive, museum and other cultural se...	Gambling and betting services	Sp se amus recrea
1	1991	54494.1	27.2	971.3	26059.1	22534.4	9	1244.9	760.8	1140	...	184.3	281.1	1
2	1992	54119.9	28	1001	25814.5	23477.2	9.1	1203	695.5	1169.5	...	160.3	253.1	
3	1993	53228.2	34.7	1057	23889.6	25204.8	9.2	1057.9	645.7	1180.2	...	164.2	262.3	1
4	1994	54010.2	42.7	1076.8	15998.5	26873	9.6	997.5	709.8	1257.4	...	149.2	246.5	

5 rows × 133 columns

In [423…

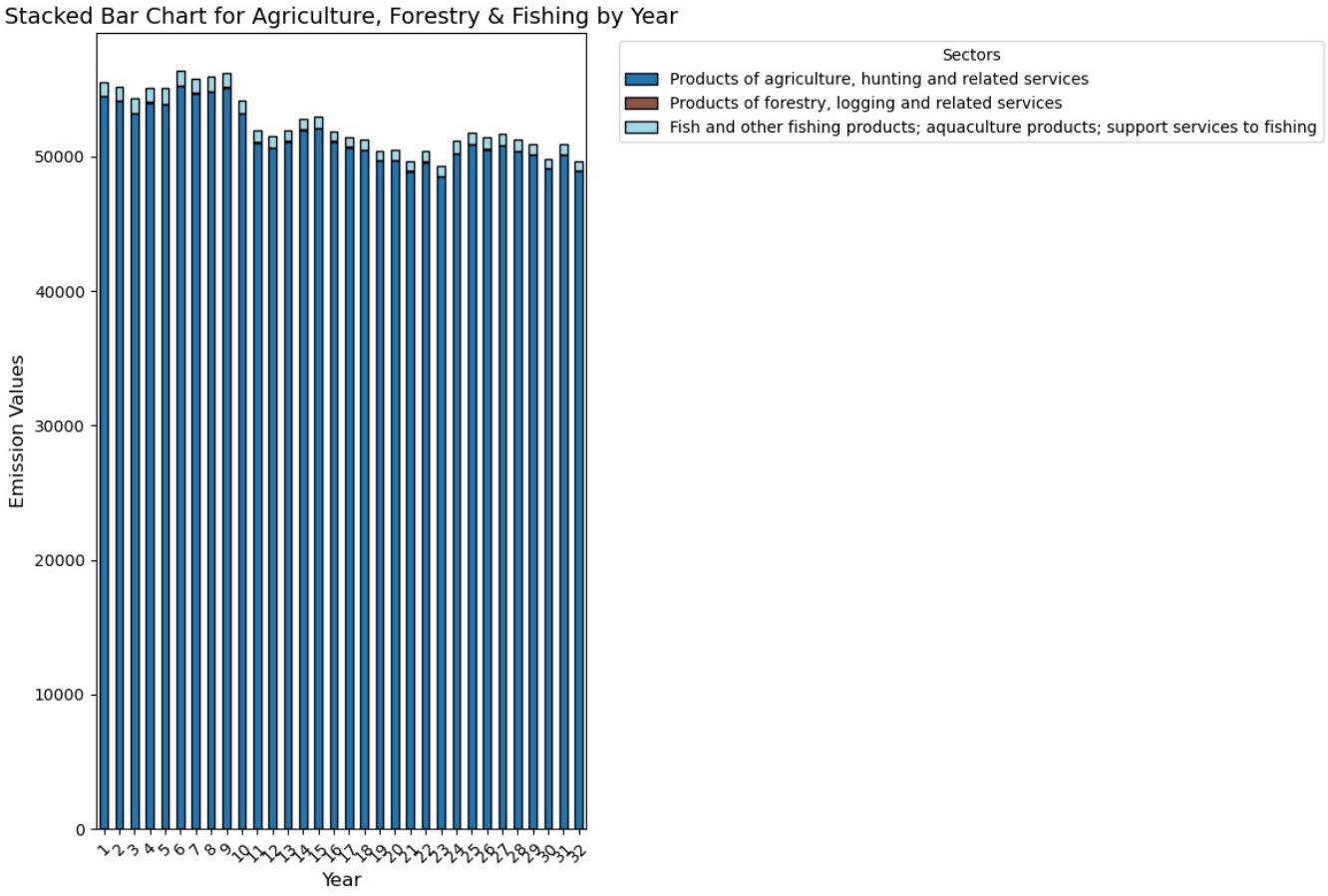
```
table2 = table2.drop(0)

table2.columns = ['Year'] + list(table2.columns[1:])

table2['Year'] = table2['Year'].astype(int)
```

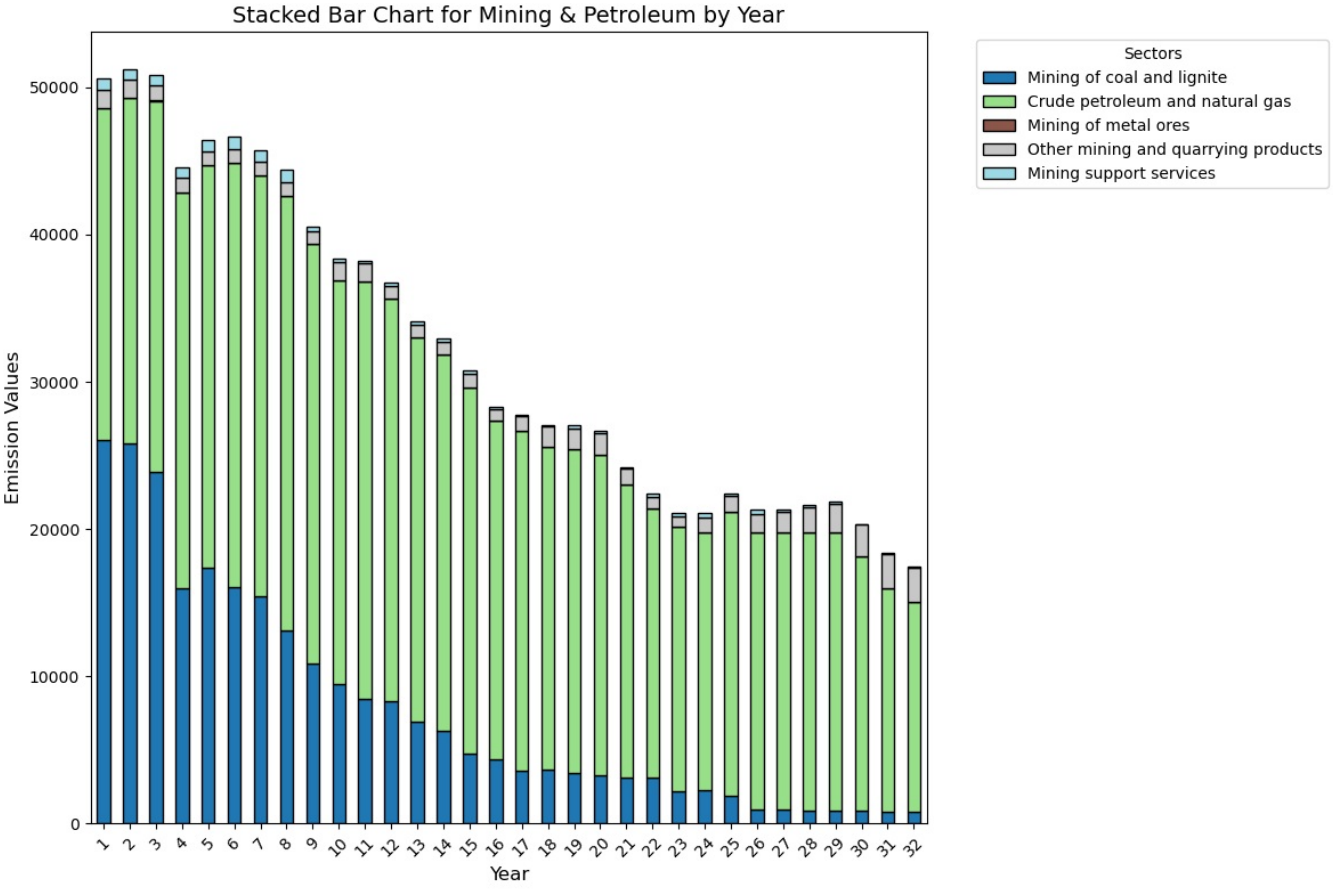
In [424…

```
#Mnaually assinging groups
group_1 = [
    'Products of agriculture, hunting and related services',
    'Products of forestry, logging and related services',
    'Fish and other fishing products; aquaculture products; support services to fishing'
]
group_2 = [
    'Mining of coal and lignite',
    'Crude petroleum and natural gas',
    'Mining of metal ores',
    'Other mining and quarrying products',
    'Mining support services'
]
group_3 = [
    'Processing and preserving of meat and production of meat products',
    'Processing and preserving of fish, crustaceans, molluscs, fruit and vegetables',
    'Manufacture of vegetable and animal oils and fats',
    'Manufacture of dairy products',
    'Manufacture of grain mill products, starches and starch products',
    'Manufacture of bakery and farinaceous products',
    'Manufacture of other food products',
    'Manufacture of prepared animal feeds',
    'Manufacture of alcoholic beverages, including spirits, wine, cider, beer and malt',
    'Manufacture of soft drinks: production of mineral waters and other bottled waters',
    'Tobacco products',
    'Textiles',
    'Wearing apparel',
    'Leather and related products',
    'Wood and of products of wood and cork, except furniture; articles of straw and plaiting materials',
    'Paper and paper products',
    'Printing and recording services',
    'Manufacture of coke oven products',
    'Manufacture of refined petroleum products',
    'Manufacture of industrial gases and non-nitrogen-based inorganic chemicals',
    'Fertilisers',
    'Other nitrogen compounds',
    'Manufacture of petrochemicals',
    'Manufacture of dyestuffs, agro-chemicals',
    'Manufacture of paints, varnishes & ink',
    'Manufacture of cleaning & toilet preparations',
    'Manufacture of other chemical products',
    'Basic pharmaceutical products and pharmaceutical preparations',
    'Rubber products',
    'Plastics products',
```

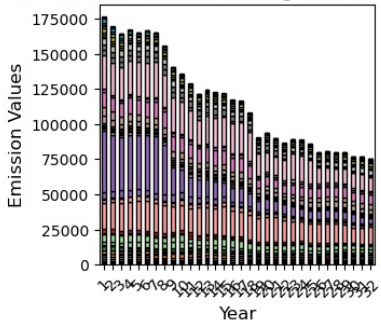
```
    'Manufacture of glass, refractory, clay, other porcelain and ceramic products, Stone, & abrasive products',
    'Manufacture of cement',
    'Manufacture of lime',
    'Manufacture of plaster',
    'Manufacture of articles of concrete, cement and plaster',
    'Manufacture of basic Iron & Steel',
    'Manufacture of other basic metals & casting (excl. Nuclear fuel & Aluminium)',
    'Aluminium production',
    'Processing of nuclear fuel',
    'Fabricated metal products, except machinery and equipment, excluding weapons and ammunition',
    'Manufacture of weapons and ammunition',
    'Computer, electronic, communication and optical products',
    'Electrical equipment',
    'Machinery and equipment n.e.c.',
    'Motor vehicles, trailers and semi-trailers',
    'Building of ships and boats',
    'Manufacture of air and spacecraft and related machinery',
    'Manufacture of other transport equipment, excluding ships, boats, air and spacecraft',
    'Furniture',
    'Other manufactured goods',
    'Repair & maintenance of ships',
    'Rest of repair; Installation'
]
group_4 = [
    'Electricity production - gas',
    'Electricity production - coal',
    'Electricity production - nuclear',
    'Electricity production - oil',
    'Electricity production - other',
    'Manufacture of gas; distribution of gaseous fuels through mains and steam and air conditioning supply'
]
group_5 = [
    'Natural water; water treatment and supply services',
    'Sewerage services; sewage sludge',
    'Waste collection, treatment and disposal services; materials recovery services',
    'Remediation services and other waste management services'
]
group_6 = [
    'Buildings and building construction works',
    'Constructions and construction works for civil engineering',
    'Specialised construction works'
]
group_7 = [
    'Wholesale and retail trade and repair services of motor vehicles and motorcycles',
    'Wholesale trade services, except of motor vehicles and motorcycles',
    'Retail trade services, except of motor vehicles and motorcycles'
]
group_8 = [
    'Rail transport',
    'Buses, coaches, trams and similar public urban transport n.e.c',
    'Underground, metro other non interurban rail services',
    'Taxis and other renting of private cars with driver',
    'Freight transport by road and removal services',
    'Transport via pipeline',
    'Water transport services',
    'Air transport services',
    'Warehousing and support services for transportation',
    'Postal and courier services'
]
group_9 = [
    'Accommodation services',
    'Food and beverage serving services'
]
group_10 = [
    'Publishing services',
    'Motion picture, video and television programme production services, sound recording and music publishing',
    'Programming and broadcasting services',
    'Telecommunications services',
    'Computer programming, consultancy and related services',
    'Information services'
]
group_11 = [
    'Financial services, except insurance and pension funding',
    'Insurance & Reinsurance',
    'Pension funding',
    'Services auxiliary to financial services and insurance services'
]
group_12 = [
    'Buying and selling of own real estate: renting and operating of own or leased real estate, excluding imput
    'Real estate activities on a fee or contract basis'
]
group_13 = [
    'Legal activities',
    'Accounting, bookkeeping and auditing activities: tax consultancy',
    'Services of head offices; management consulting services',
    'Architectural and engineering services; technical testing and analysis services',
    'Scientific research and development services',
    'Advertising and market research services',
    'Other professional, scientific and technical services',
```

```
    'Veterinary services'
]
group_14 = [
    'Rental and leasing services',
    'Employment services',
    'Travel agency, tour operator and other reservation services and related services',
    'Security and investigation services',
    'Services to buildings and landscape',
    'Office administrative, office support and other business support services'
]
group_15 = [
    'Public administration; compulsory social security services',
    'Public defence services'
]
group_16 = [
    'Education services'
]
group_17 = [
    'Human health services',
    'Residential care services',
    'Social work services without accommodation'
]
group_18 = [
    'Creative, arts and entertainment services',
    'Library, archive, museum and other cultural services',
    'Gambling and betting services',
    'Sporting services and amusement and recreation services'
]
group_19 = [
    'Services furnished by membership organisations',
    'Repair services of computers and personal and household goods',
    'Other personal services'
]
group_20 = [
    'Services of households as employers of domestic personnel'
]
group_21 = [
    'Consumer expenditure - not travel',
    'Consumer expenditure - travel'
]

# Function to plot stacked bar chart for each group
def plot_stacked_bar_chart(group, group_name):
    # Select the relevant data for the group from the DataFrame
    data_group = table2[group]

    # Plot the stacked bar chart
    ax = data_group.plot(kind='bar', stacked=True, figsize=(12, 8), colormap='tab20', edgecolor='black')

    # Title and axis labels
    ax.set_title(f'Stacked Bar Chart for {group_name} by Year', fontsize=14)
    ax.set_xlabel('Year', fontsize=12)
    ax.set_ylabel('Emission Values', fontsize=12)
    plt.xticks(rotation=45)

    # Adding legend and tight layout for neatness
    plt.legend(title='Sectors', bbox_to_anchor=(1.05, 1), loc='upper left')
    plt.tight_layout()
    plt.show()

# Plot stacked bar charts for each group
plot_stacked_bar_chart(group_1, "Agriculture, Forestry & Fishing")
plot_stacked_bar_chart(group_2, "Mining & Petroleum")
plot_stacked_bar_chart(group_3, "Manufacturing & Other Products")
plot_stacked_bar_chart(group_4, "Electricity & Gas")
plot_stacked_bar_chart(group_5, "Water & Waste Services")
plot_stacked_bar_chart(group_6, "Construction")
plot_stacked_bar_chart(group_7, "Trade & Retail Services")
plot_stacked_bar_chart(group_8, "Transport & Postal Services")
plot_stacked_bar_chart(group_9, "Accommodation & Food Services")
plot_stacked_bar_chart(group_10, "Publishing & Telecom Services")
plot_stacked_bar_chart(group_11, "Financial & Insurance Services")
plot_stacked_bar_chart(group_12, "Real Estate Services")
plot_stacked_bar_chart(group_13, "Professional Services")
plot_stacked_bar_chart(group_14, "Business Support Services")
plot_stacked_bar_chart(group_15, "Public Services")
plot_stacked_bar_chart(group_16, "Education Services")
plot_stacked_bar_chart(group_17, "Health & Social Services")
plot_stacked_bar_chart(group_18, "Arts, Sports & Recreation")
plot_stacked_bar_chart(group_19, "Other Personal Services")
plot_stacked_bar_chart(group_20, "Domestic Personnel Services")
plot_stacked_bar_chart(group_21, "Consumer Expenditure")
```


Stacked Bar Chart for Agriculture, Forestry & Fishing by Year
Sectors
Products of agriculture, hunting and related services
Products of forestry, logging and related services
Fish and other fishing products; aquaculture products; support services to fishing
Emission Values
Year

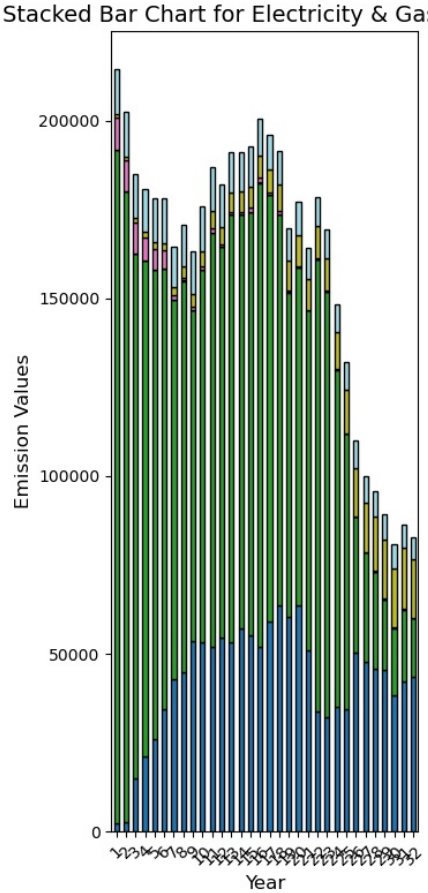

Stacked Bar Chart for Mining & Petroleum by Year
Sectors
Mining of coal and lignite
Crude petroleum and natural gas
Mining of metal ores
Other mining and quarrying products
Mining support services
Emission Values
Year

Stacked Bar Chart for Manufacturing & Other Products by Year

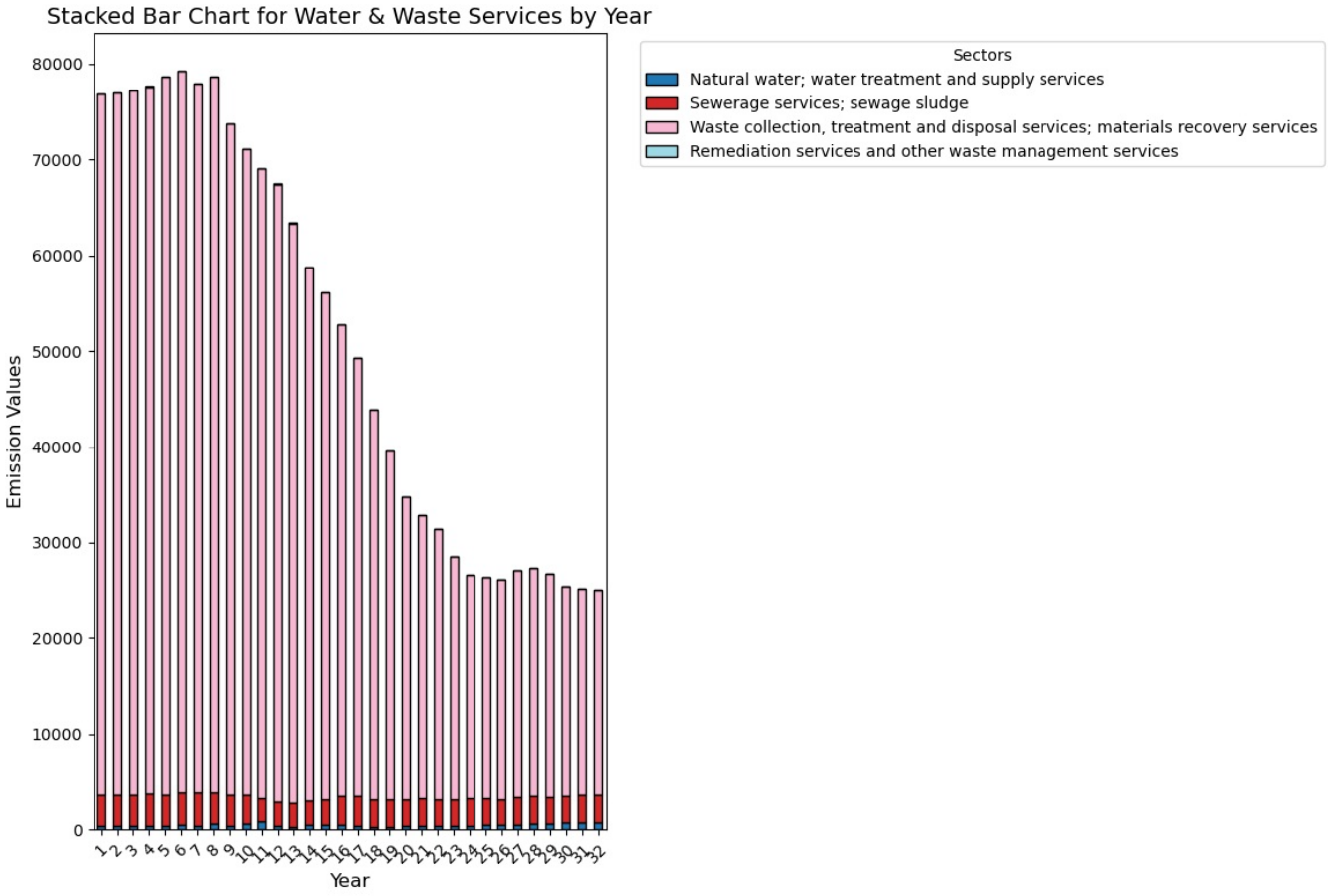


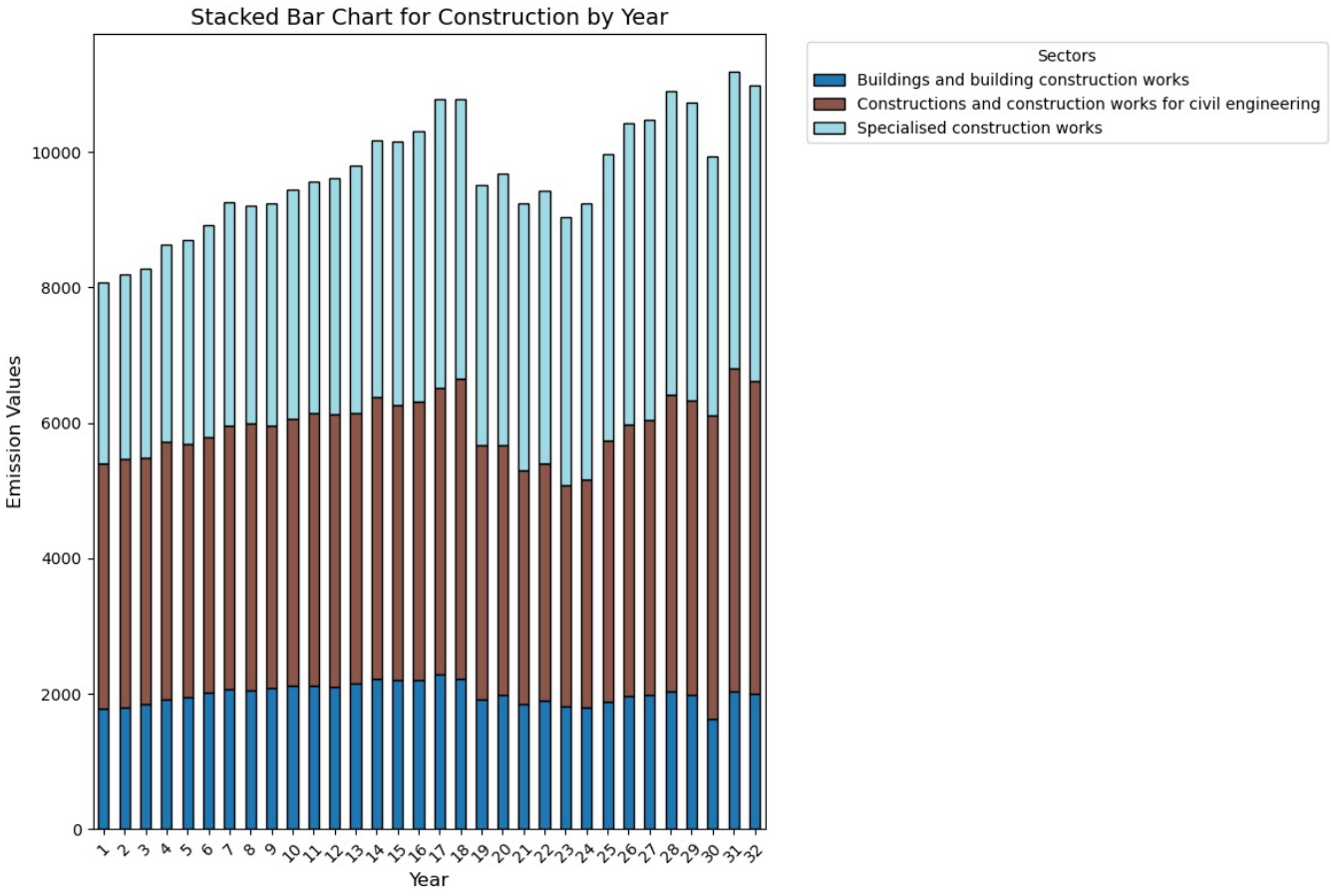
Sectors
- Processing and preserving of meat and production of meat products
- Processing and preserving of fish, crustaceans, molluscs, fruit and vegetables
- Manufacture of vegetable and animal oils and fats
- Manufacture of dairy products
- Manufacture of grain mill products, starches and starch products
- Manufacture of bakery and farinaceous products
- Manufacture of other food products
- Manufacture of prepared animal feeds
- Manufacture of alcoholic beverages, including spirits, wine, cider, beer and malt
- Manufacture of soft drinks: production of mineral waters and other bottled waters
- Tobacco products
- Textiles
- Wearing apparel
- Leather and related products
- Wood and of products of wood and cork, except furniture; articles of straw and plaiting materials
- Paper and paper products
- Printing and recording services
- Manufacture of coke oven products
- Manufacture of refined petroleum products
- Manufacture of industrial gases and non-nitrogen-based inorganic chemicals
- Fertilisers
- Other nitrogen compounds
- Manufacture of petrochemicals
- Manufacture of dyestuffs, agro-chemicals
- Manufacture of paints, varnishes & ink
- Manufacture of cleaning & toilet preparations
- Manufacture of other chemical products
- Basic pharmaceutical products and pharmaceutical preparations
- Rubber products
- Plastics products
- Manufacture of glass, refractory, clay, other porcelain and ceramic products, Stone, & abrasive products
- Manufacture of cement
- Manufacture of lime
- Manufacture of plaster
- Manufacture of articles of concrete, cement and plaster
- Manufacture of basic Iron & Steel
- Manufacture of other basic metals & casting (excl. Nuclear fuel & Aluminium)
- Aluminium production
- Processing of nuclear fuel
- Fabricated metal products, except machinery and equipment, excluding weapons and ammunition
- Manufacture of weapons and ammunition
- Computer, electronic, communication and optical products
- Electrical equipment
- Machinery and equipment n.e.c.
- Motor vehicles, trailers and semi-trailers
- Building of ships and boats
- Manufacture of air and spacecraft and related machinery
- Manufacture of other transport equipment, excluding ships, boats, air and spacecraft
- Furniture
- Other manufactured goods
- Repair & maintenance of ships
- Rest of repair; Installation

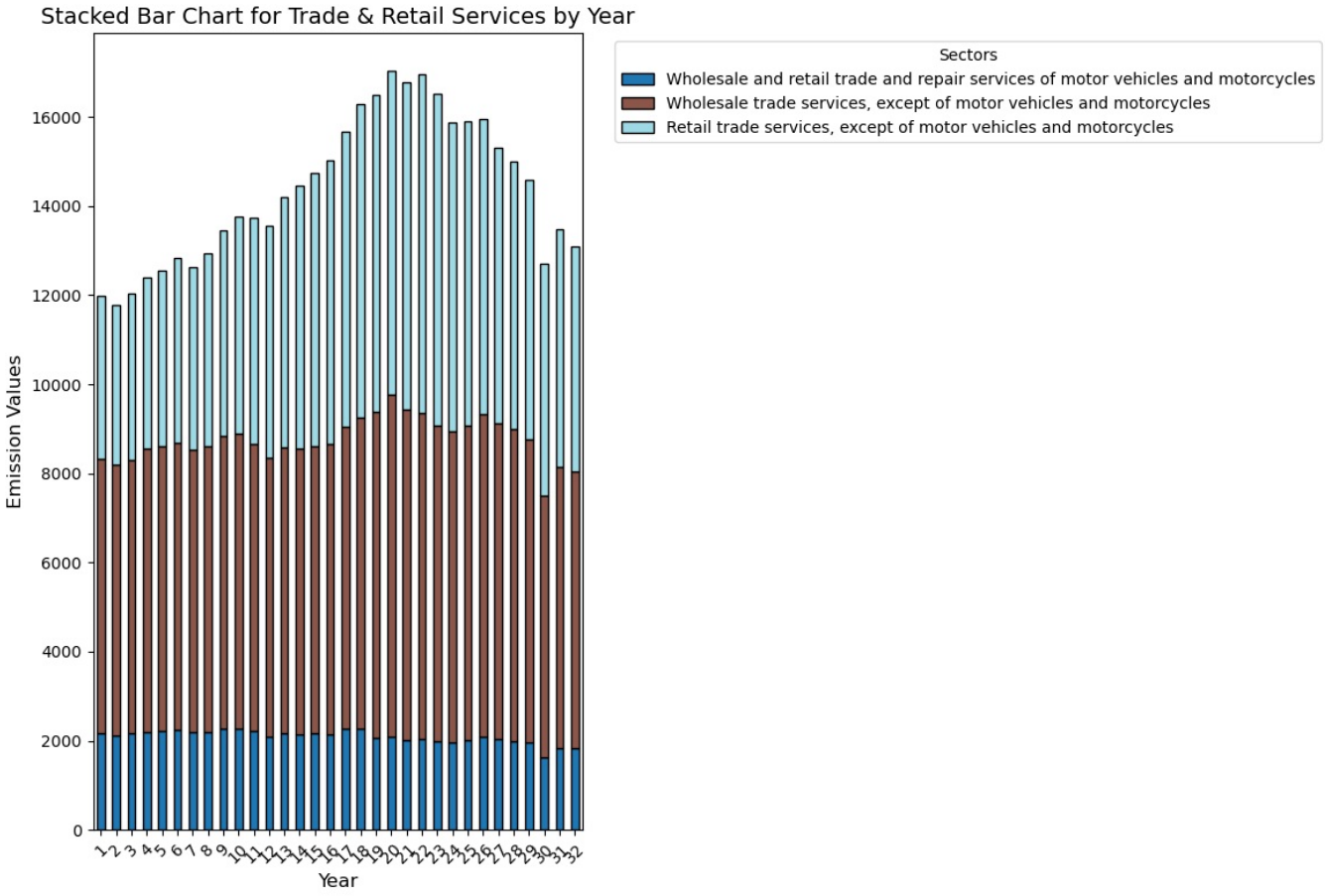
Stacked Bar Chart for Electricity & Gas by Year

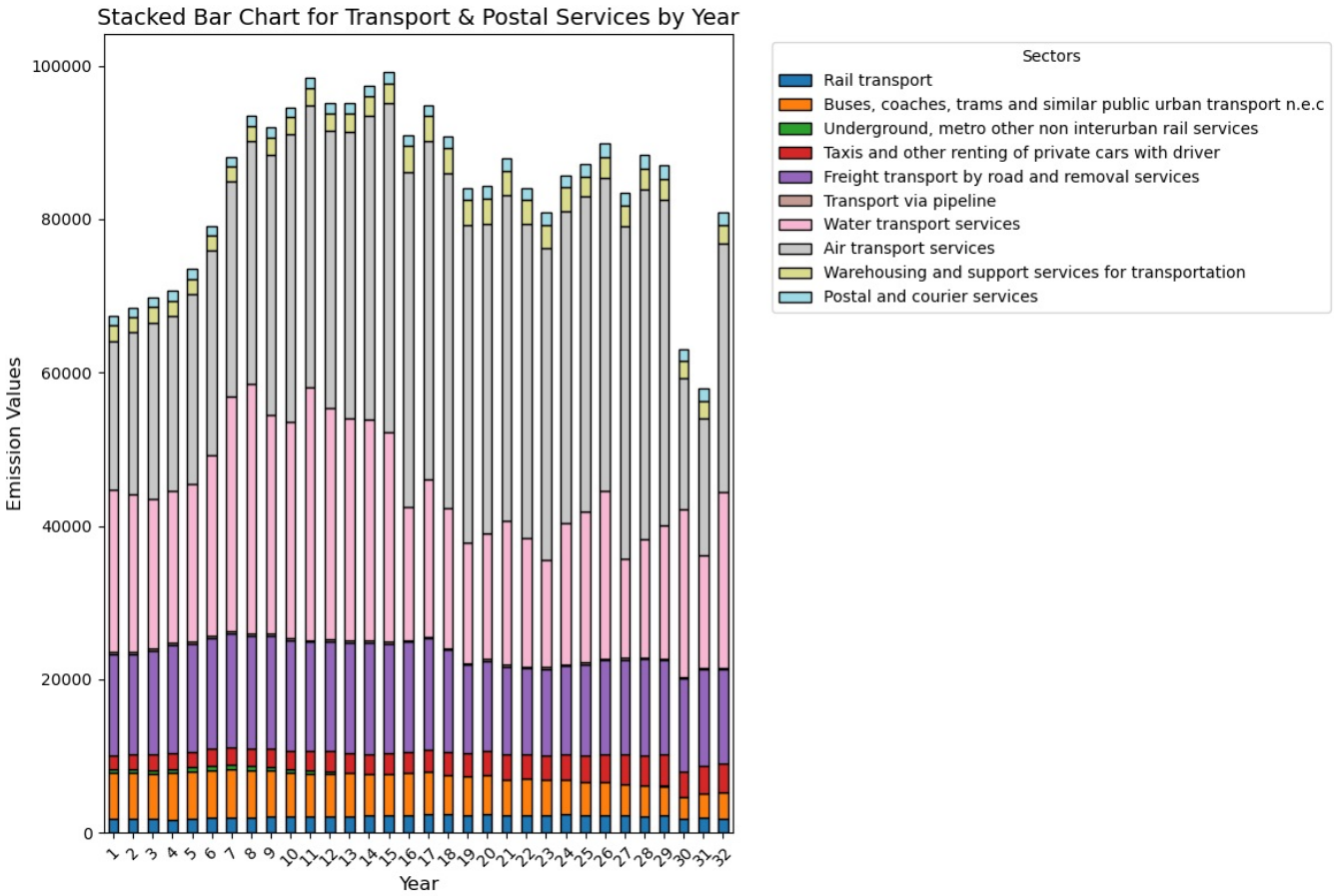


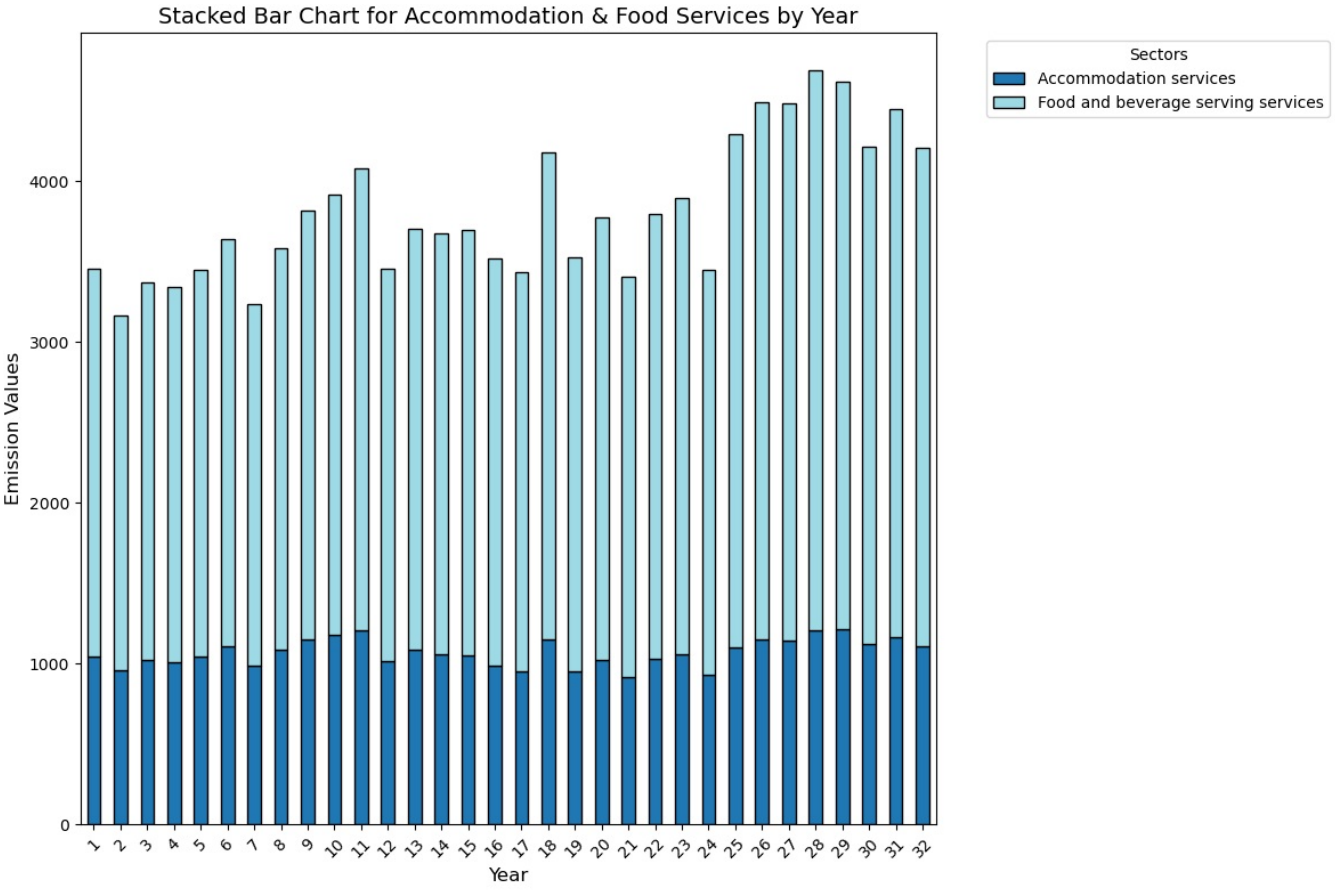
Sectors
- Electricity production - gas
- Electricity production - coal
- Electricity production - nuclear
- Electricity production - oil
- Electricity production - other
- Manufacture of gas; distribution of gaseous fuels through mains and steam and air conditioning supply

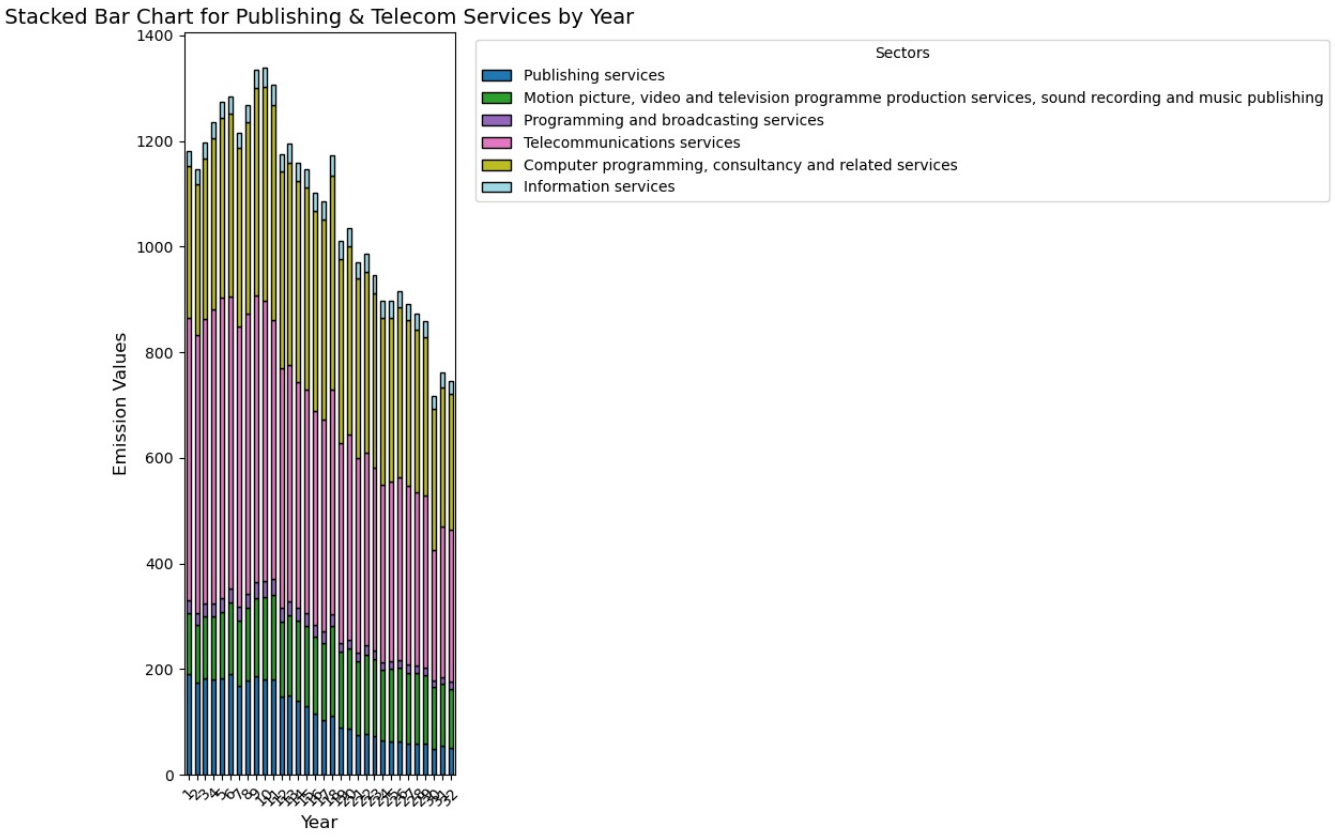

Stacked Bar Chart for Water & Waste Services by Year
Emission Values
Year
Sectors
Natural water; water treatment and supply services
Sewerage services; sewage sludge
Waste collection, treatment and disposal services; materials recovery services
Remediation services and other waste management services

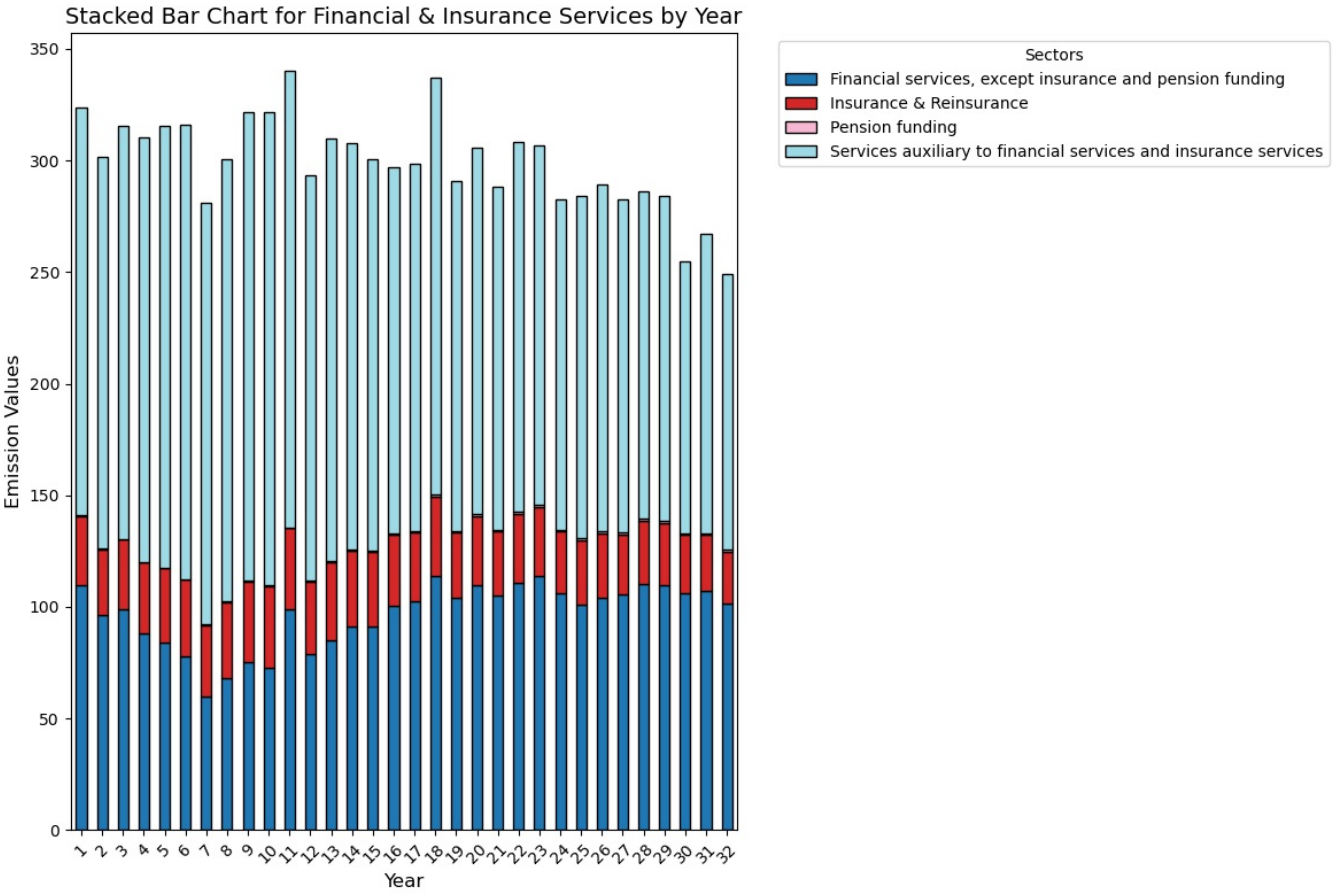

Stacked Bar Chart for Construction by Year
Emission Values
Year
Sectors
Buildings and building construction works
Constructions and construction works for civil engineering
Specialised construction works

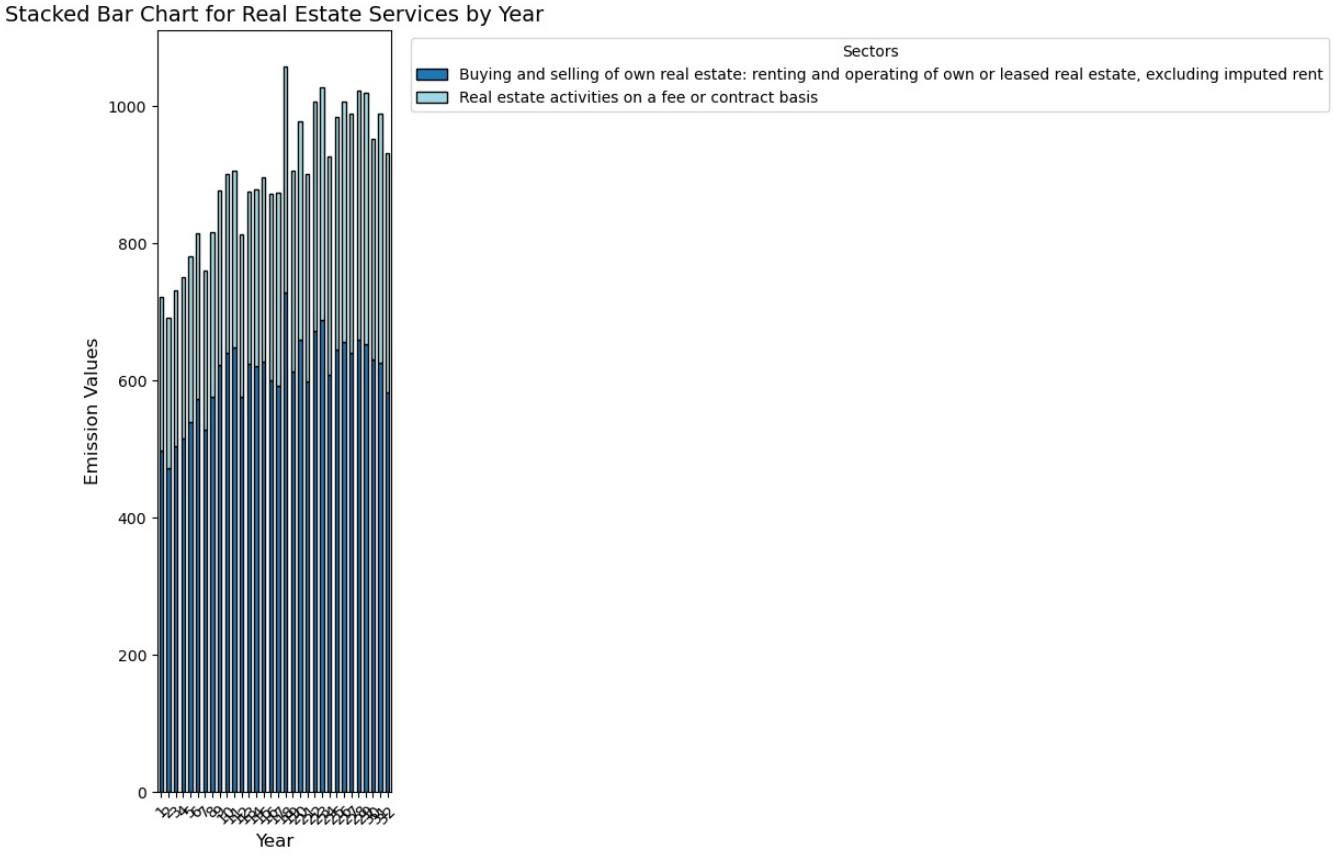

Stacked Bar Chart for Trade & Retail Services by Year
Sectors
Wholesale and retail trade and repair services of motor vehicles and motorcycles
Wholesale trade services, except of motor vehicles and motorcycles
Retail trade services, except of motor vehicles and motorcycles
Emission Values
Year

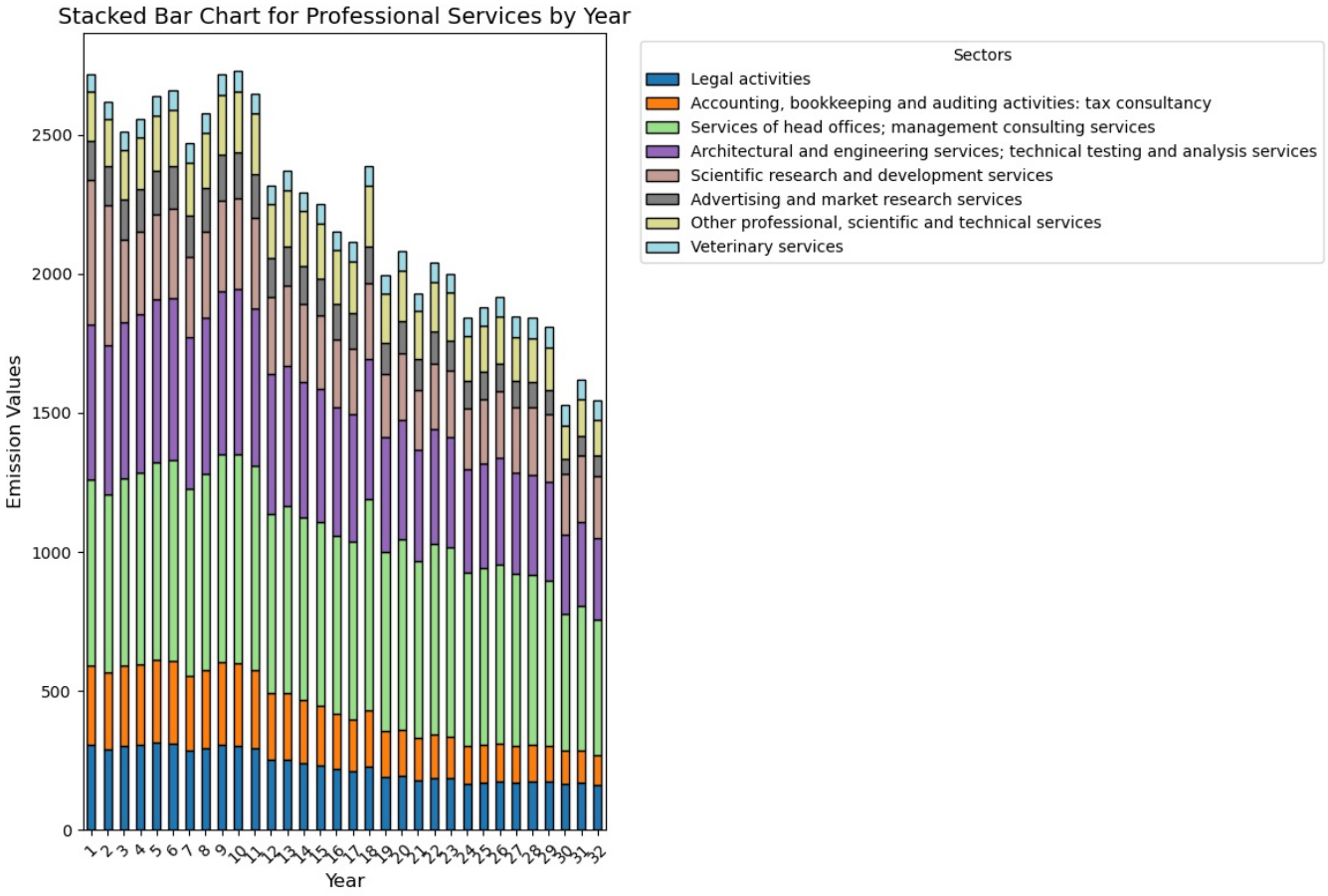

Stacked Bar Chart for Transport & Postal Services by Year
Sectors
Rail transport
Buses, coaches, trams and similar public urban transport n.e.c
Underground, metro other non interurban rail services
Taxis and other renting of private cars with driver
Freight transport by road and removal services
Transport via pipeline
Water transport services
Air transport services
Warehousing and support services for transportation
Postal and courier services
Emission Values
Year

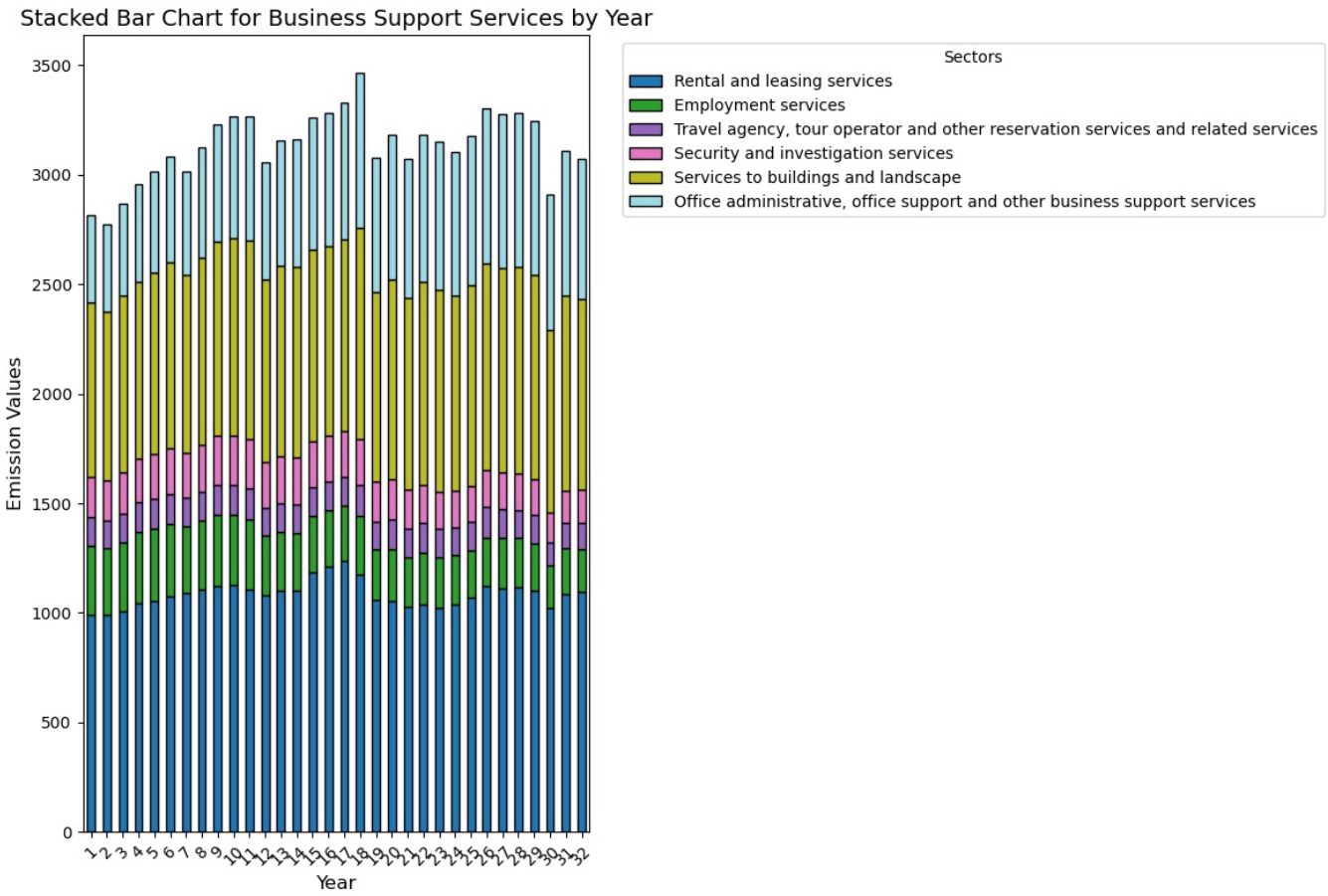

Stacked Bar Chart for Accommodation & Food Services by Year
Sectors
Accommodation services
Food and beverage serving services
Emission Values
0
1000
2000
3000
4000
Year
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

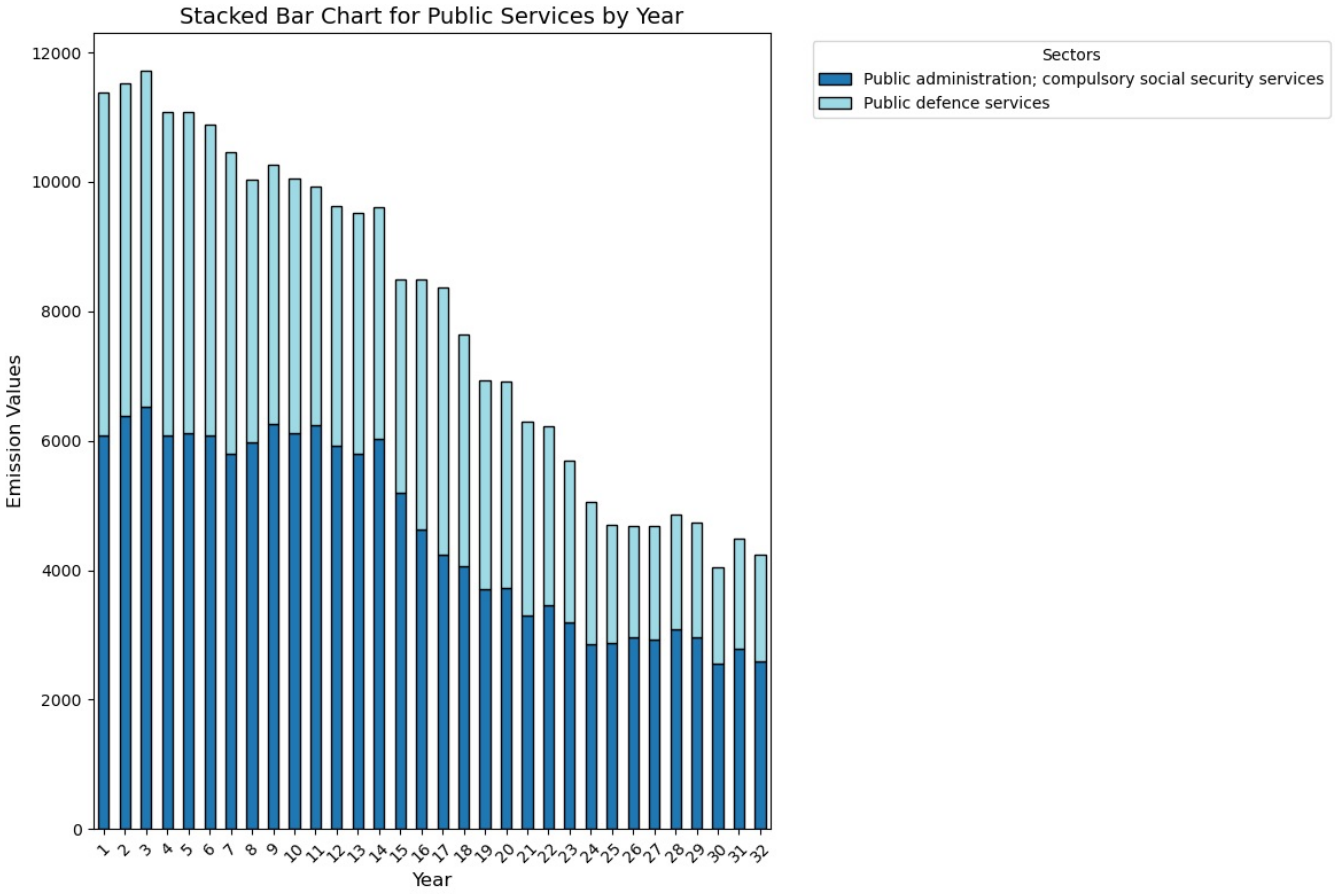

Stacked Bar Chart for Publishing & Telecom Services by Year
Sectors
Publishing services
Motion picture, video and television programme production services, sound recording and music publishing
Programming and broadcasting services
Telecommunications services
Computer programming, consultancy and related services
Information services
Emission Values
0
200
400
600
800
1000
1200
1400
Year

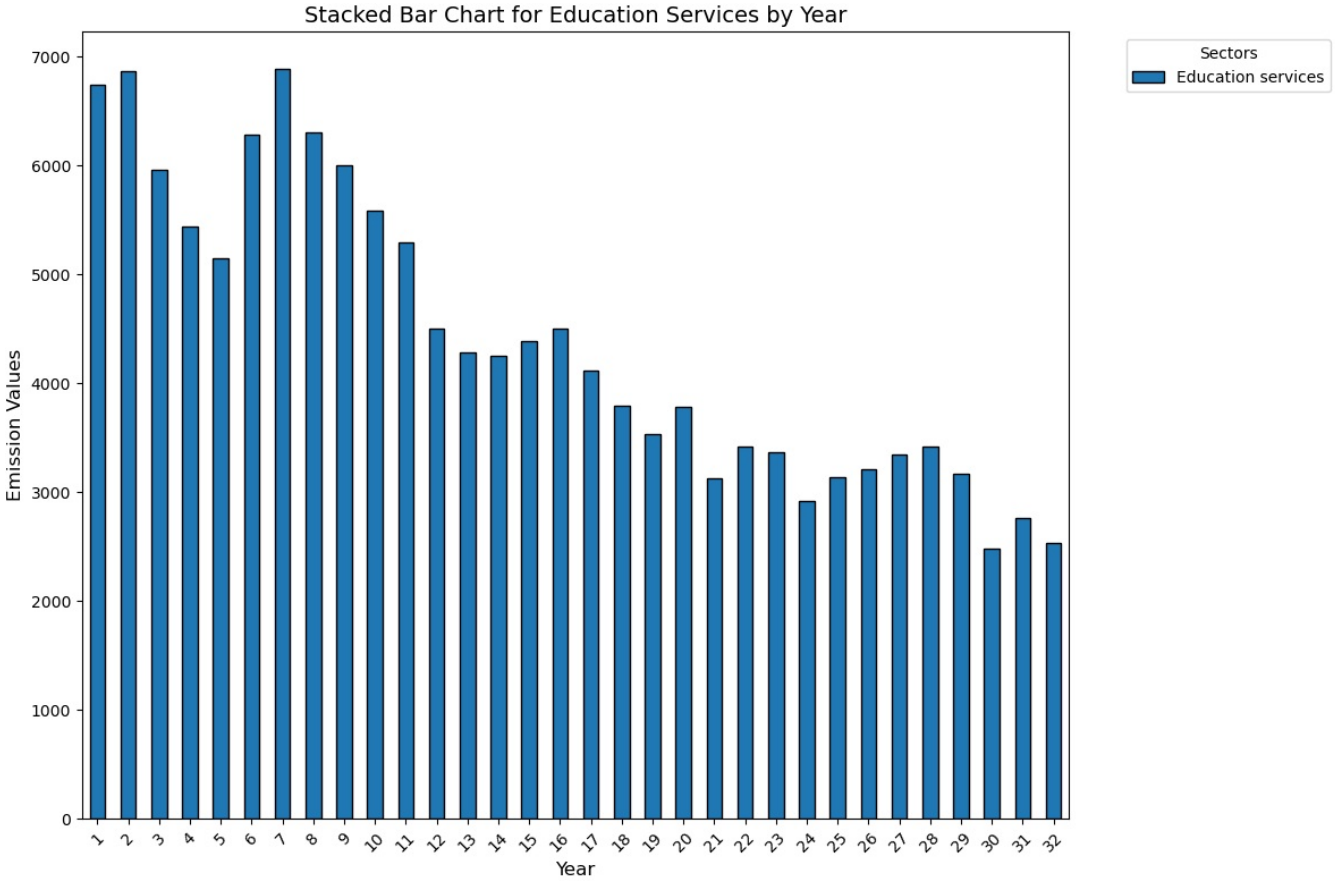

Stacked Bar Chart for Financial & Insurance Services by Year
Emission Values
Year
Sectors
Financial services, except insurance and pension funding
Insurance & Reinsurance
Pension funding
Services auxiliary to financial services and insurance services

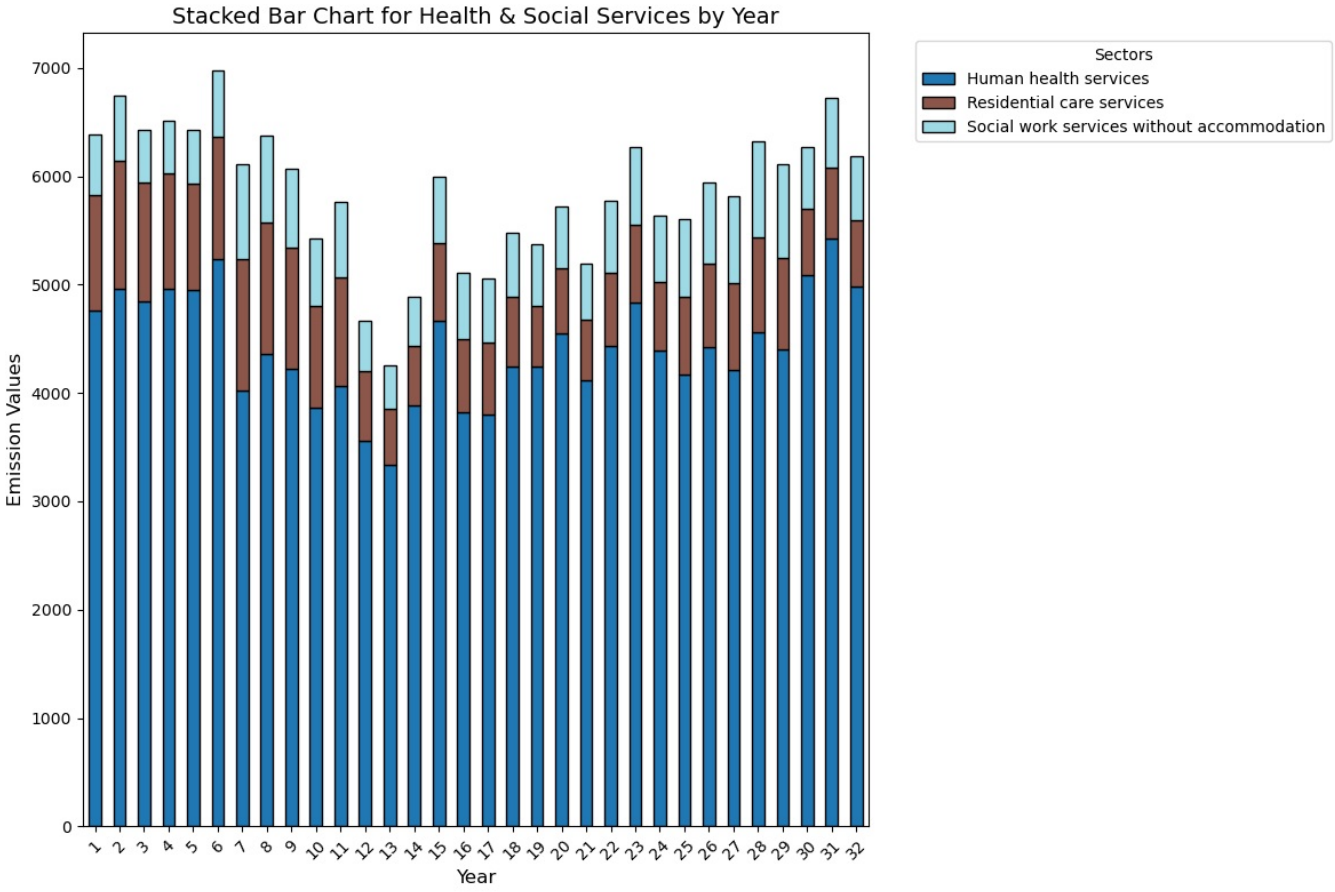

Stacked Bar Chart for Real Estate Services by Year
Emission Values
Year
Sectors
Buying and selling of own real estate: renting and operating of own or leased real estate, excluding imputed rent
Real estate activities on a fee or contract basis

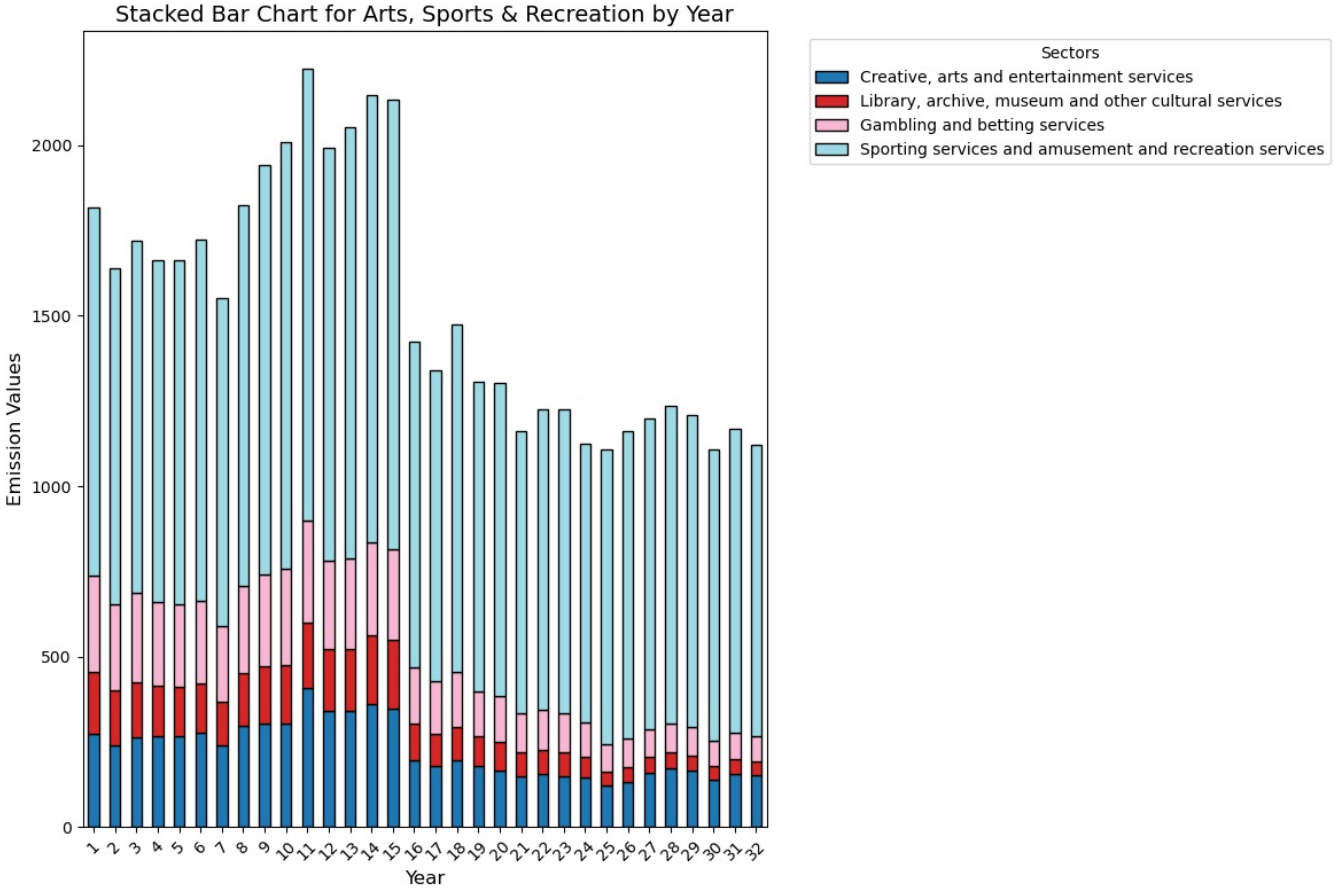

Stacked Bar Chart for Professional Services by Year
Emission Values
Year
Sectors
Legal activities
Accounting, bookkeeping and auditing activities: tax consultancy
Services of head offices; management consulting services
Architectural and engineering services; technical testing and analysis services
Scientific research and development services
Advertising and market research services
Other professional, scientific and technical services
Veterinary services

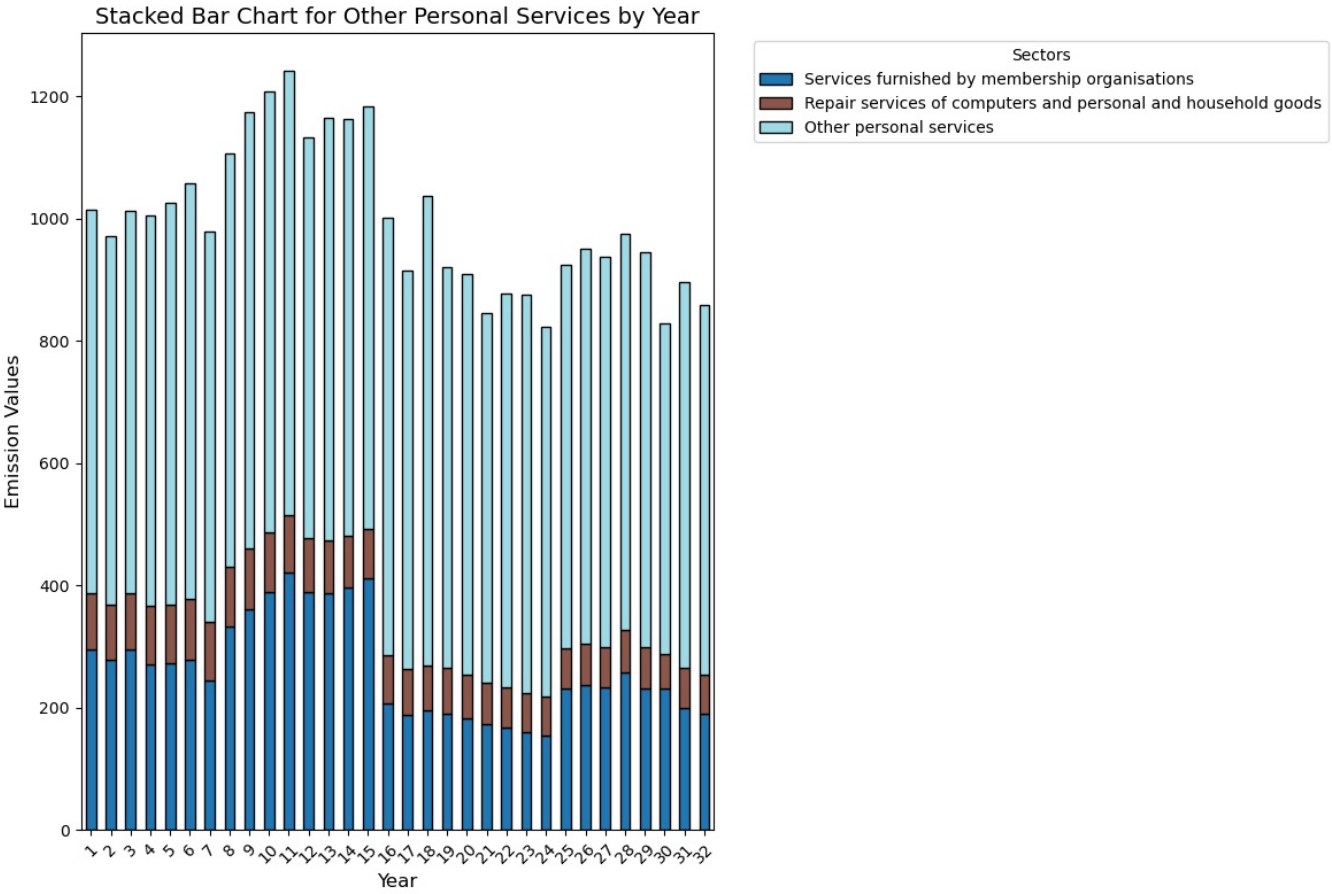

Stacked Bar Chart for Business Support Services by Year
Emission Values
Year
Sectors
Rental and leasing services
Employment services
Travel agency, tour operator and other reservation services and related services
Security and investigation services
Services to buildings and landscape
Office administrative, office support and other business support services

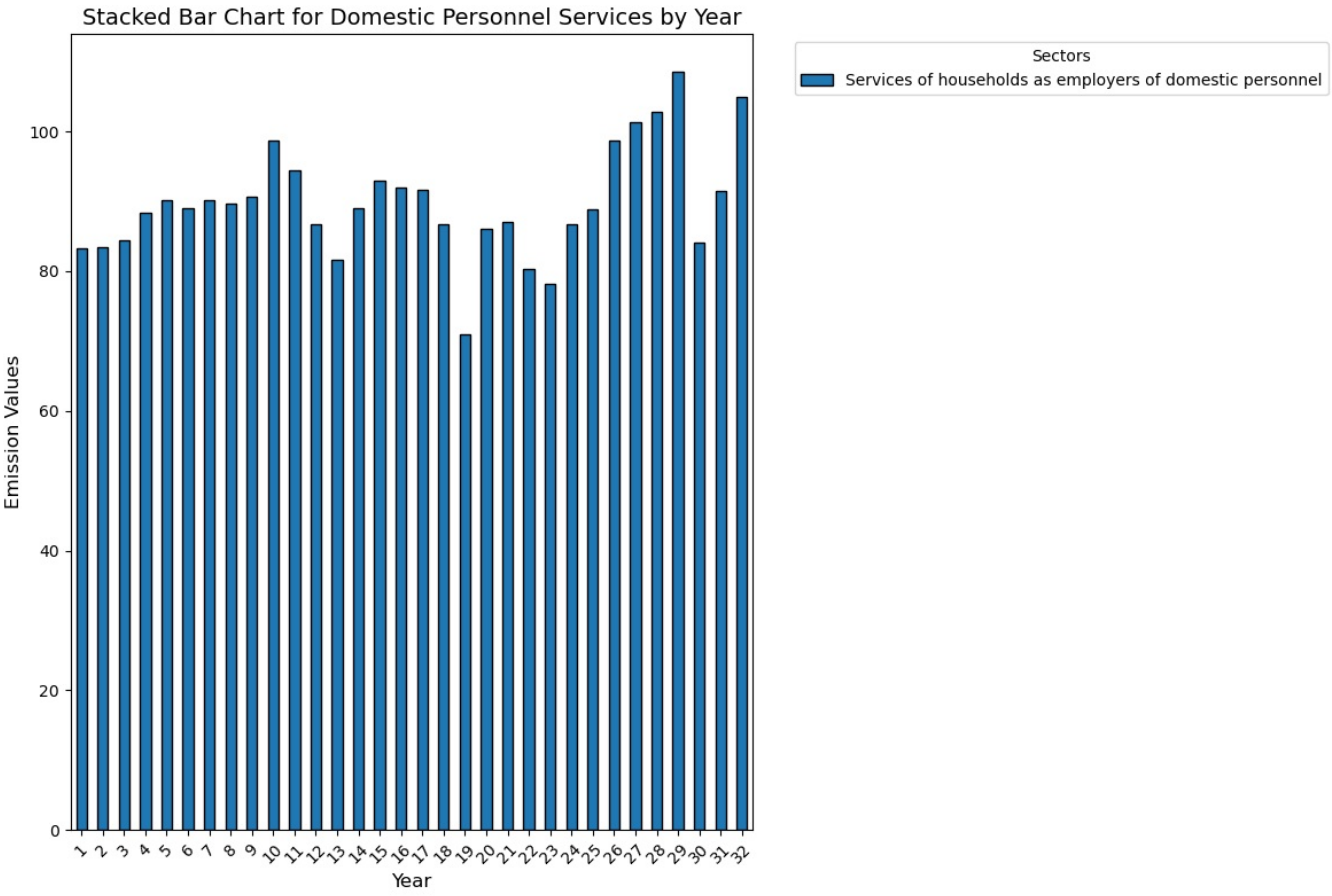

Stacked Bar Chart for Public Services by Year
Sectors
Public administration; compulsory social security services
Public defence services
Emission Values
Year

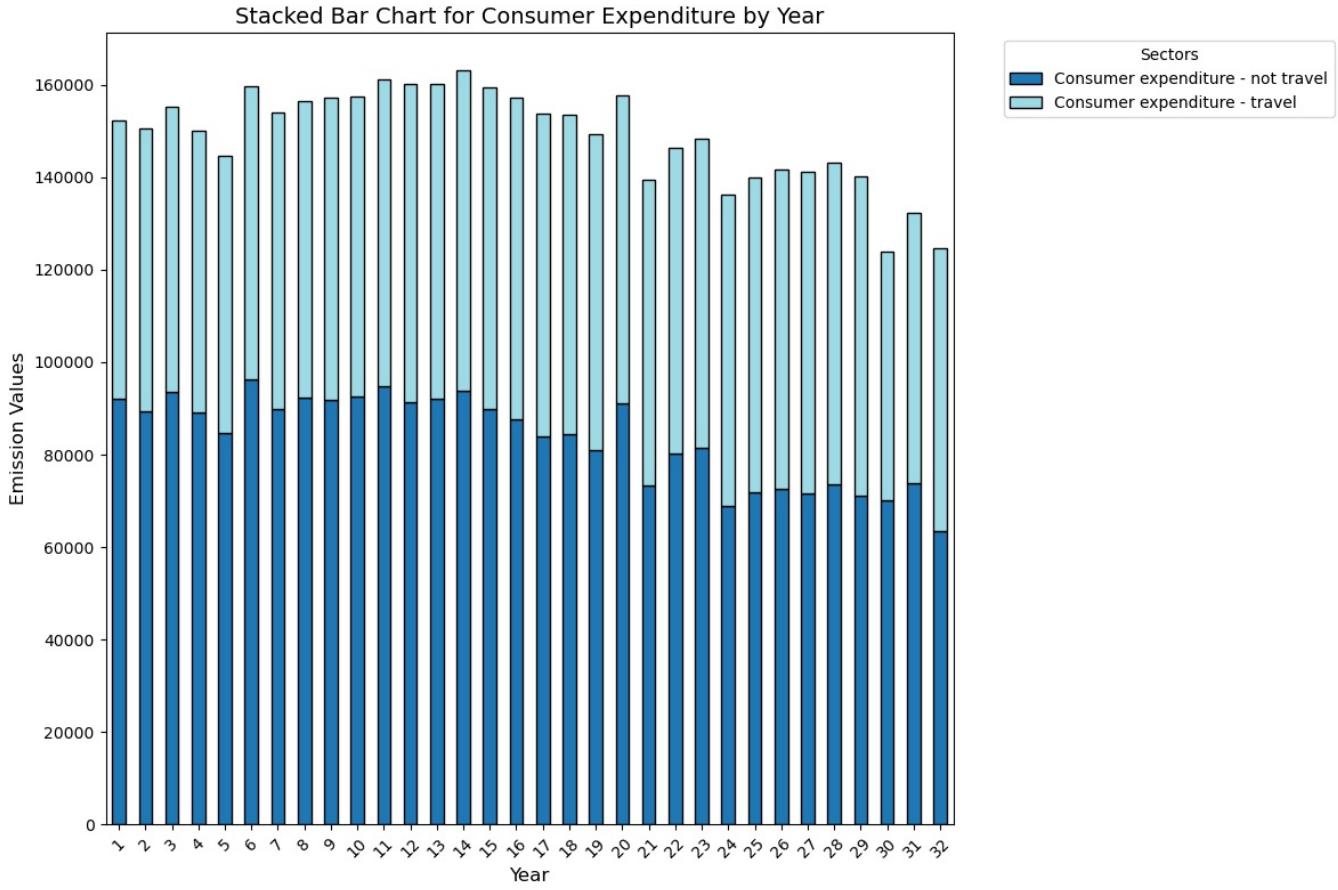

Stacked Bar Chart for Education Services by Year
Sectors
Education services
Emission Values
Year


Stacked Bar Chart for Health & Social Services by Year
Sectors
Human health services
Residential care services
Social work services without accommodation
Emission Values
Year


Stacked Bar Chart for Arts, Sports & Recreation by Year
Sectors
Creative, arts and entertainment services
Library, archive, museum and other cultural services
Gambling and betting services
Sporting services and amusement and recreation services
Emission Values
Year


Stacked Bar Chart for Other Personal Services by Year
Sectors
Services furnished by membership organisations
Repair services of computers and personal and household goods
Other personal services
Emission Values
Year


Stacked Bar Chart for Domestic Personnel Services by Year
Sectors
Services of households as employers of domestic personnel
Emission Values
Year



In [ ]:

In [425…

```
table1
```

Out[425]:

	SIC code	A	B	C	D	E	F	G	H	I	...	C
0	Industry name	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning s...	Water supply; sewerage, waste management and r...	Construction	Wholesale and retail trade; repair of motor ve...	Transport and storage	Accommodation and food services	...	Public administration and defence compulsory ..
1	1990	55690.3	50452.8	175212.7	217602.2	75316.3	8201.2	11560.8	66985.3	3015.3	...	12058.4
2	1991	55492.6	50608.3	176254.4	214317.5	76852.7	8072	11984.1	67438.6	3453.5	...	11381.2
3	1992	55149	51199.3	169388.2	202469.2	76959.1	8188.3	11789.9	68438.5	3163.4	...	11523.4
4	1993	54319.9	50807.1	164365	184997.2	77275.6	8275.1	12032.9	69735.2	3371.4	...	11710.9
5	1994	55129.7	44588.4	167285.9	180721.6	77637.9	8633.8	12403.1	70659.2	3342.3	...	11071.3
6	1995	55061.9	46451.4	165344.1	178162.0	78712	8695.3	12560.6	73508.2	3450.7	...	11079.5
7	1996	56343.6	46633.6	166622.3	177960.6	79209.4	8923.8	12825.8	79135.4	3636.1	...	10890.1
8	1997	55810	45761.2	165258.4	164554.2	77982.7	9263.6	12631.7	88159.2	3237.9	...	10462.7
9	1998	55950.8	44399.3	155653.0	170815.5	78691.1	9207.7	12926.4	93464.0	3581	...	10028.0
10	1999	56176.1	40501.6	140696.8	163207.5	73742.9	9246.5	13464.5	91941.5	3819.3	...	10260.4
11	2000	54133.6	38335.6	135627.4	176005.8	71060.5	9440.5	13774.1	94607.7	3913.4	...	10046.3
12	2001	51955.6	38210.6	128931.3	186969.6	69110.6	9566.7	13738.1	98397.8	4077.2	...	9921.5
13	2002	51495.5	36722.5	121456.3	182100.7	67450.9	9608.6	13544.7	95135.4	3457.7	...	9629.5
14	2003	51950.4	34092.4	124723.3	191163.6	63375.2	9791.5	14202.6	95101.3	3705	...	9518.5
15	2004	52813.4	32909.3	122868.9	191097.4	58712.5	10170.6	14461.8	97343.8	3673	...	9605.4
16	2005	52934	30738.2	121871.7	192574.6	56104.9	10152	14753.1	99137.9	3694.7	...	8493.2
17	2006	51899.8	28291.7	117269.7	200397.5	52739.5	10309.7	15013	90909.2	3518.8	...	8486.5
18	2007	51473.1	27770.4	116409.3	195947.3	49262.6	10785.7	15656.8	94883.2	3433.4	...	8360.2
19	2008	51240.7	27088.3	108134.2	191548.8	43934.9	10784.4	16297.4	90840.5	4178.1	...	7647
20	2009	50442.8	27056.8	90639.6	169738.9	39631	9504.6	16483.2	84039.5	3525.6	...	6938.5
21	2010	50502.1	26661.9	93988	177018.4	34835.1	9682.6	17024.2	84360.1	3773.8	...	6907.7
22	2011	49647.4	24205.7	90258	164156.6	32853.5	9235.3	16767.9	87918.1	3403.6	...	6299.3
23	2012	50368.8	22379.2	86759.9	178480.3	31469.9	9426.8	16946	84115	3794.9	...	6217.2
24	2013	49299.1	21096.9	89442.9	169292	28566.7	9033.9	16508.8	80876.9	3896.3	...	5696.7
25	2014	51149.2	21086.9	88797.2	148347.2	26627.8	9236	15865.5	85705.9	3444	...	5057
26	2015	51774.5	22402.3	86036.2	131922.4	26416.7	9971.5	15912.8	87193.9	4289.6	...	4693.1
27	2016	51397.4	21347.1	80201.2	109990	26154.7	10417.8	15949.4	89850.6	4486.3	...	4674.1
28	2017	51685.1	21287.7	80871.8	100019.5	27149.1	10474.8	15320.7	83486.7	4484.6	...	4685.5
29	2018	51227.2	21606.2	80321.5	95782.4	27404.6	10903.5	14993.8	88349.5	4685.3	...	4865.3
30	2019	50889.8	21858.1	80113	89388.3	26787.1	10734.7	14592.1	86998.4	4618	...	4739.5
31	2020	49836	20321.3	77060.3	80736.3	25434.7	9941	12705.2	63020.4	4213	...	4038.4
32	2021	50909.4	18346.2	77092.3	86311	25252.1	11180.6	13471.4	57961.1	4446.4	...	4494.4
33	2022	49682.9	17433.4	75449.5	82854.7	25109.3	10990.5	13084.3	80891	4203.9	...	4233.5
34	2023	49176.8	16415.6	73496.8	70134.5	24868.4	11243.8	12855.7	83875.1	4122.1	...	4380.0

35 rows × 25 columns

In [ ]:

In [426…

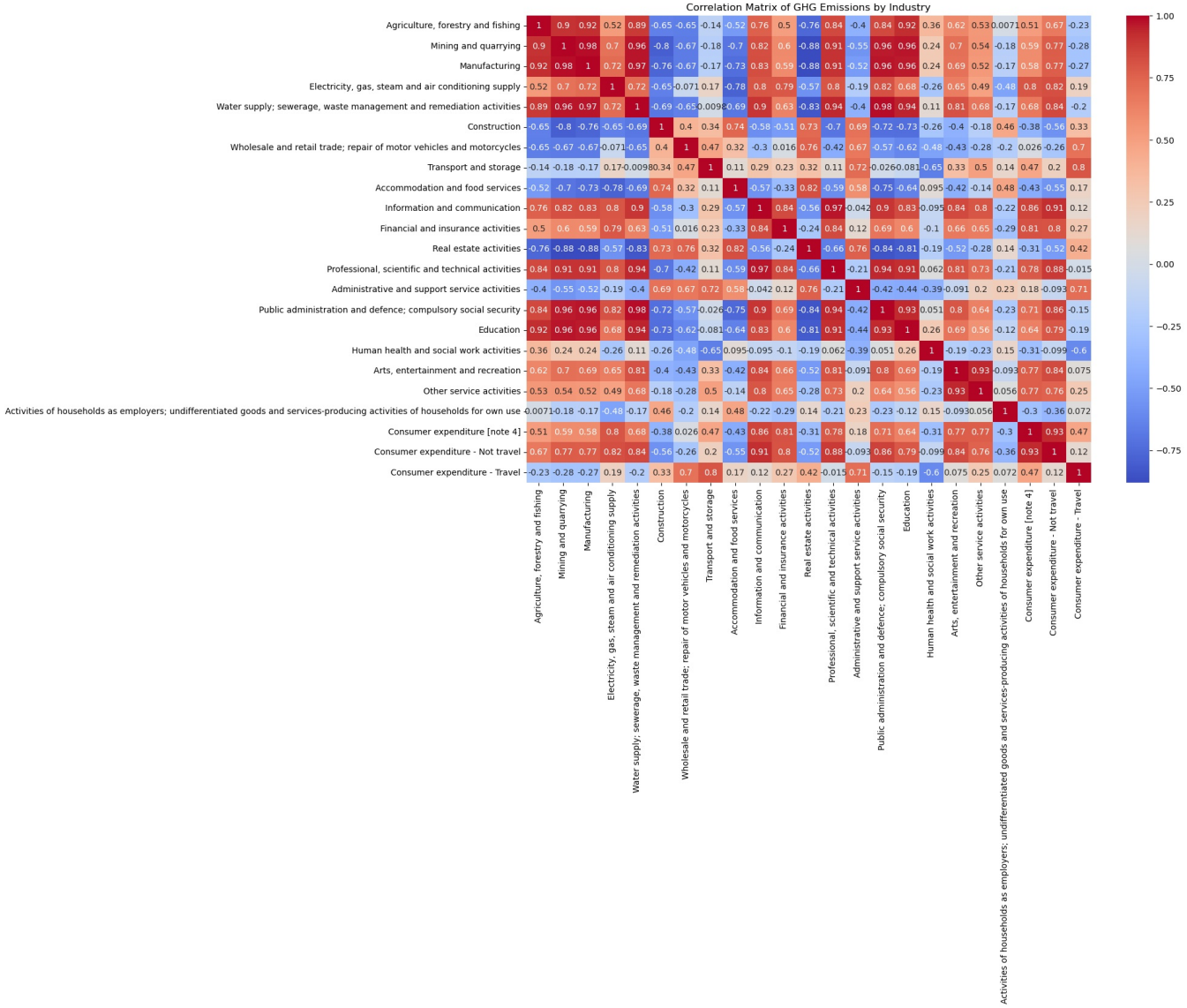
```python
#percentage increase calculator

years = table1.iloc[1:, 0].values
industry_names = table1.iloc[0, 1:-1].values

#empty array to store percentages
total_percentage_changes = {}

# Loop through each industry to calculate the total percentage change
for industry_index, industry in enumerate(industry_names):
    emissions_data = table1.iloc[1:, industry_index + 1].values

    #total percentage change from 1990 to 2023
    start_value = emissions_data[0]  # Emissions in 1990
    end_value = emissions_data[-1]  # Emissions in 2023
    total_percentage_change = ((end_value - start_value) / start_value) * 100
```

```
    # Store the total percentage change in the dictionary
    total_percentage_changes[industry] = total_percentage_change

# Display the results
print("Total percentage change in emissions from 1990 to 2023 for each industry:")
for industry, percentage_change in total_percentage_changes.items():
    print(f"{industry}: {percentage_change:.2f}%")
```

```
Total percentage change in emissions from 1990 to 2023 for each industry:
Agriculture, forestry and fishing: -11.70%
Mining and quarrying: -67.46%
Manufacturing: -58.05%
Electricity, gas, steam and air conditioning supply: -67.77%
Water supply; sewerage, waste management and remediation activities: -66.98%
Construction: 37.10%
Wholesale and retail trade; repair of motor vehicles and motorcycles: 11.20%
Transport and storage: 25.21%
Accommodation and food services: 36.71%
Information and communication: -32.93%
Financial and insurance activities: -19.32%
Real estate activities: 42.15%
Professional, scientific and technical activities: -39.47%
Administrative and support service activities: 16.46%
Public administration and defence; compulsory social security: -63.68%
Education: -61.21%
Human health and social work activities: 0.33%
Arts, entertainment and recreation: -34.00%
Other service activities: -10.54%
Activities of households as employers; undifferentiated goods and services-producing activities of households f
or own use: -2.97%
Consumer expenditure [note 4]: -17.16%
Consumer expenditure - Not travel: -28.22%
Consumer expenditure - Travel: -2.01%
```

In [427…

```
#Correlation matrix

emissions_data = table1.iloc[1:, 1:-1]

emissions_data = emissions_data.apply(pd.to_numeric, errors='coerce')

# Calculate the correlation matrix
correlation_matrix = np.corrcoef(emissions_data.T)

# Create the heatmap
plt.figure(figsize=(15, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', xticklabels=industry_names, yticklabels=industry_n
plt.title('Correlation Matrix of GHG Emissions by Industry')

plt.show()
```



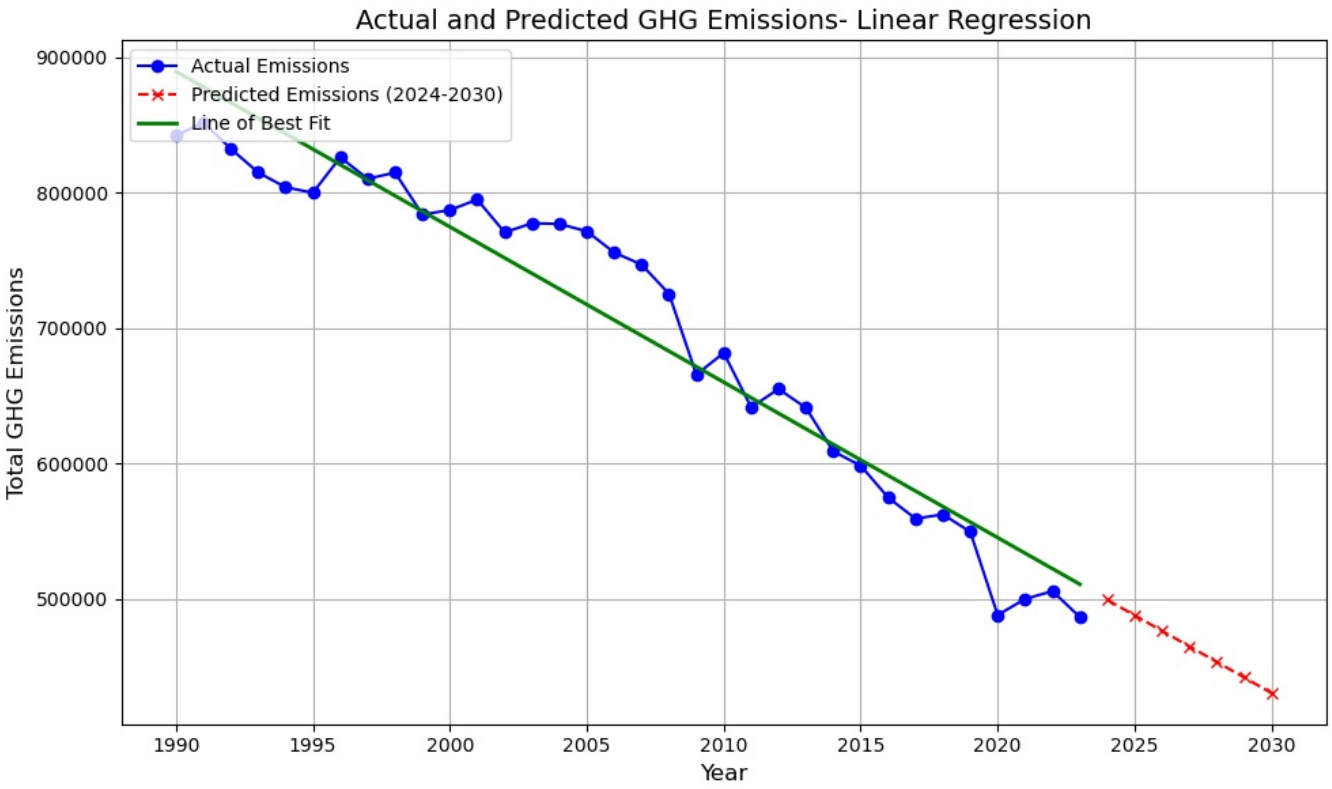
Models

In [ ]:

In [428…

```
#Linear regression model
#Reference[5]:(freshhotdata, 2023)


years = table1.iloc[1:, 0].values.reshape(-1, 1)
total_emissions = table1.iloc[1:, -1].values

# Train the linear regression model on availble data
model = LinearRegression()
model.fit(years, total_emissions)

# Predict the emissions for the years 2024 to 2030
future_years = np.array([2024, 2025, 2026, 2027, 2028, 2029, 2030]).reshape(-1, 1)
predicted_emissions = model.predict(future_years)

# Create a line of best fit for the historical data (line of best fit for actual data)
line_of_best_fit = model.predict(years)

# Plotting the results
plt.figure(figsize=(10, 6))

# Plot actual emissions
plt.plot(years, total_emissions, label='Actual Emissions', color='blue', marker='o')

# Plot predicted emissions for 2024–2030 as a dotted line
plt.plot(future_years, predicted_emissions, label='Predicted Emissions (2024-2030)', color='red', linestyle='--

# Plot the line of best fit for the historical data
plt.plot(years, line_of_best_fit, label='Line of Best Fit', color='green', linestyle='-', linewidth=2)

# Adding labels and title
plt.title('Actual and Predicted GHG Emissions- Linear Regression', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Total GHG Emissions', fontsize=12)
```

```
plt.legend(loc='upper left', fontsize=10)
plt.grid(True)

# Show plot
plt.tight_layout()
plt.show()

#statistical measures for the model's performance
r_squared = model.score(years, total_emissions)
mae = mean_absolute_error(total_emissions, line_of_best_fit)
mse = mean_squared_error(total_emissions, line_of_best_fit)

# Print the statistical measures
print(f"R-squared: {r_squared:.4f}")
print(f"Mean Absolute Error (MAE): {mae:.4f}")
print(f"Mean Squared Error (MSE): {mse:.4f}")

# Print predicted values for 2024-2030
for i, year in enumerate(future_years):
    print(f"Predicted Total GHG Emissions for {year[0]}: {predicted_emissions[i]:.2f}")
```




```
R-squared: 0.9329
Mean Absolute Error (MAE): 25031.9173
Mean Squared Error (MSE): 911108081.9840
Predicted Total GHG Emissions for 2024: 499563.25
Predicted Total GHG Emissions for 2025: 488090.44
Predicted Total GHG Emissions for 2026: 476617.63
Predicted Total GHG Emissions for 2027: 465144.82
Predicted Total GHG Emissions for 2028: 453672.01
Predicted Total GHG Emissions for 2029: 442199.20
Predicted Total GHG Emissions for 2030: 430726.39
```

In [ ]:

In [ ]:

In [ ]:

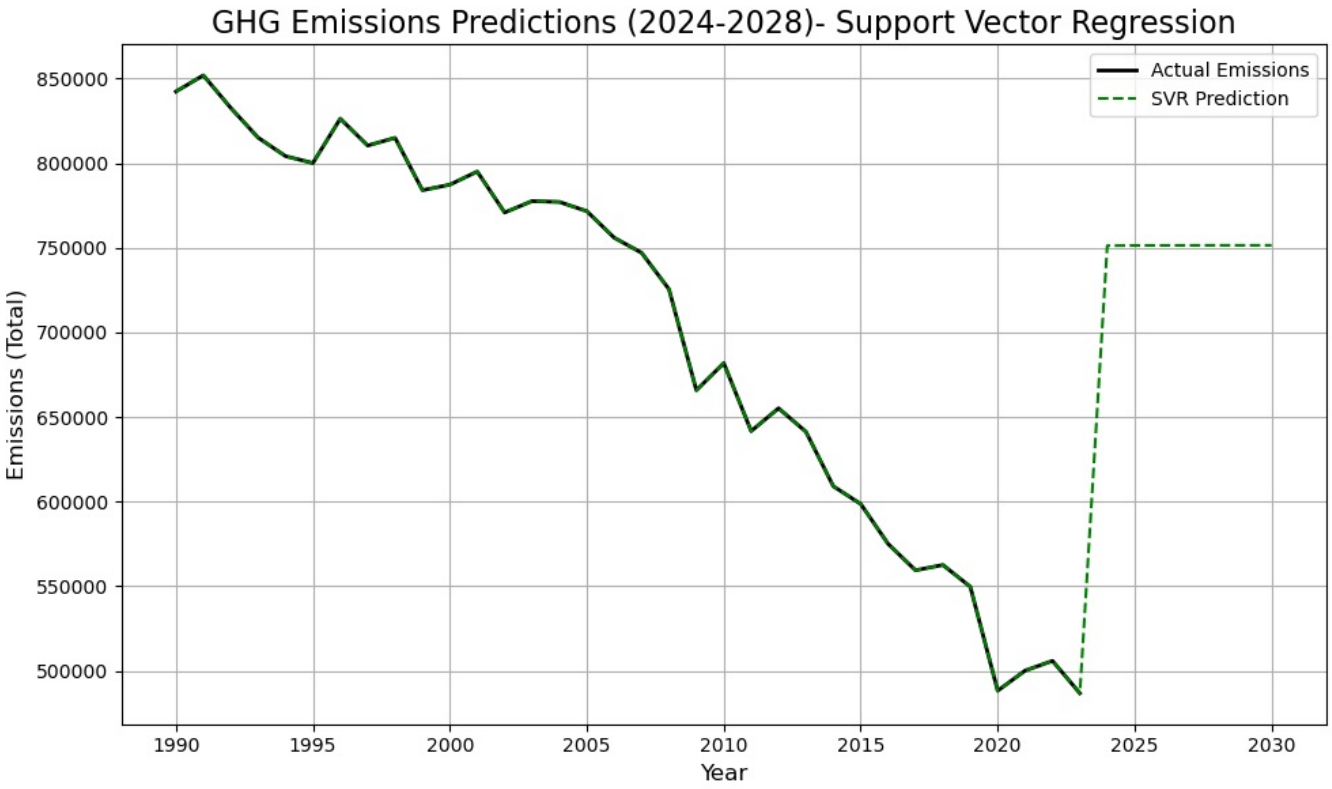
In [ ]:

In [ ]:

In [429…

```
#Support Vector Regression
#Reference[6]: (Muz, 2016)
years = table1.iloc[1:, 0].values
total_emissions = table1.iloc[1:, -1].values


years_reshaped = years.reshape(-1, 1)

#Using rbf kernel; and setting parameters
svr_model = SVR(kernel='rbf', C=100, gamma=0.1, epsilon=0.1)

# Train the SVR model
```

```
svr_model.fit(years_reshaped, total_emissions)

# Predict emissions for the next 5 years (2024 to 2028)
years_future = np.arange(2024, 2031).reshape(-1, 1)
svr_predictions = svr_model.predict(years_future)

#predictions
plt.figure(figsize=(10, 6))

# Plot the actual emissions data
plt.plot(years, total_emissions, label='Actual Emissions', color='black', linewidth=2)

# Plot Support Vector Regression predictions (with dotted line)
plt.plot(np.concatenate((years, years_future.flatten())), np.concatenate((total_emissions, svr_predictions)), l
plt.title('GHG Emissions Predictions (2024-2028)- Support Vector Regression', fontsize=16)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Emissions (Total)', fontsize=12)
plt.legend()
plt.grid(True)

# Show the plot
plt.tight_layout()
plt.show()

# ---Evaluation Metrics
svr_r2 = r2_score(total_emissions, svr_model.predict(years_reshaped))
svr_mae = mean_absolute_error(total_emissions, svr_model.predict(years_reshaped))
svr_mse = mean_squared_error(total_emissions, svr_model.predict(years_reshaped))

print("Support Vector Regression (SVR):")
print(f"R-squared: {svr_r2:.4f}")
print(f"Mean Absolute Error: {svr_mae:.4f}")
print(f"Mean Squared Error: {svr_mse:.4f}")

# Predict total emissions for the years 2024 to 2030
years_extended = np.arange(2024, 2031).reshape(-1, 1)
extended_predictions = svr_model.predict(years_extended)

#predictions for 2024-2030
print("\nPredictions for 2024 to 2030:")
for year, prediction in zip(years_extended.flatten(), extended_predictions):
    print(f"Year {year}: {prediction:.2f}")
```

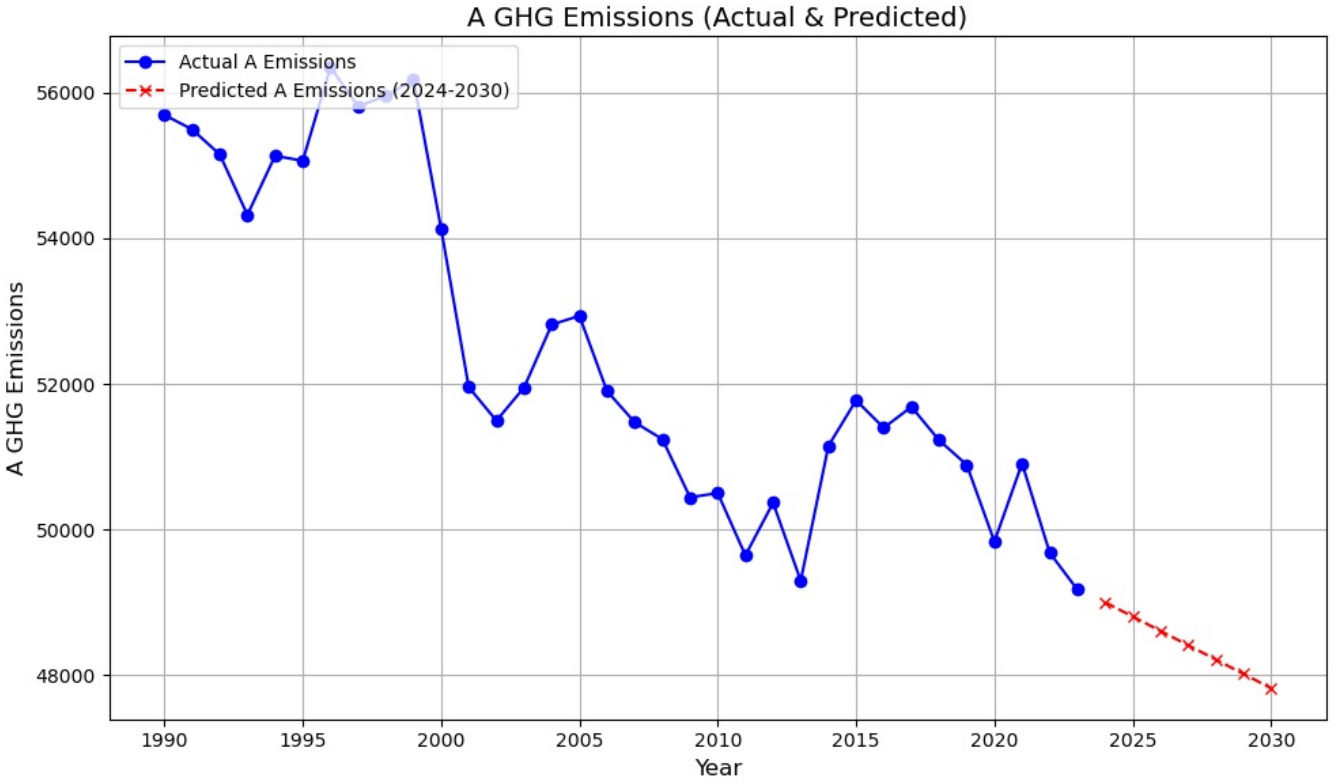



```
Support Vector Regression (SVR):
R-squared: -0.1853
Mean Absolute Error: 100198.8564
Mean Squared Error: 16095826593.6651

Predictions for 2024 to 2030:
Year 2024: 751252.75
Year 2025: 751343.23
Year 2026: 751410.27
Year 2027: 751450.92
Year 2028: 751471.11
Year 2029: 751479.32
Year 2030: 751482.05
```

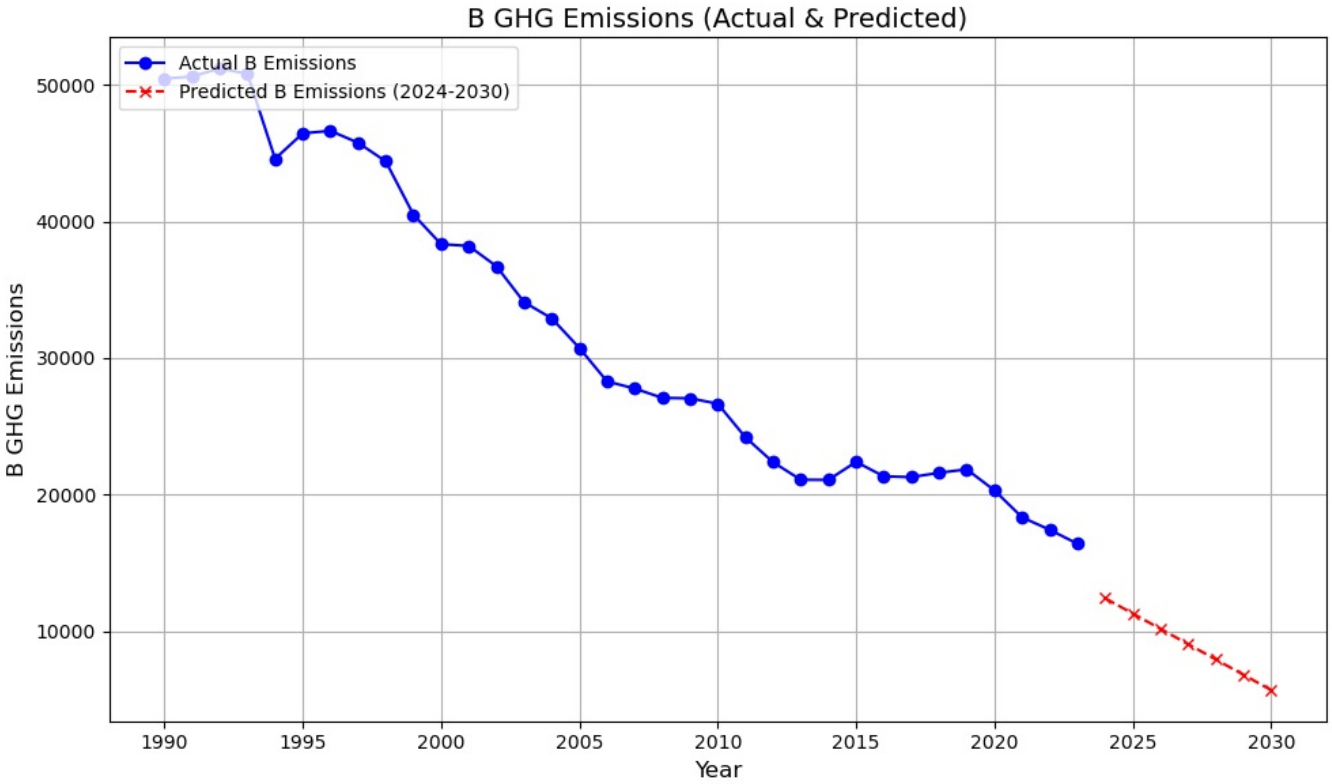
In [ ]:

In [430…

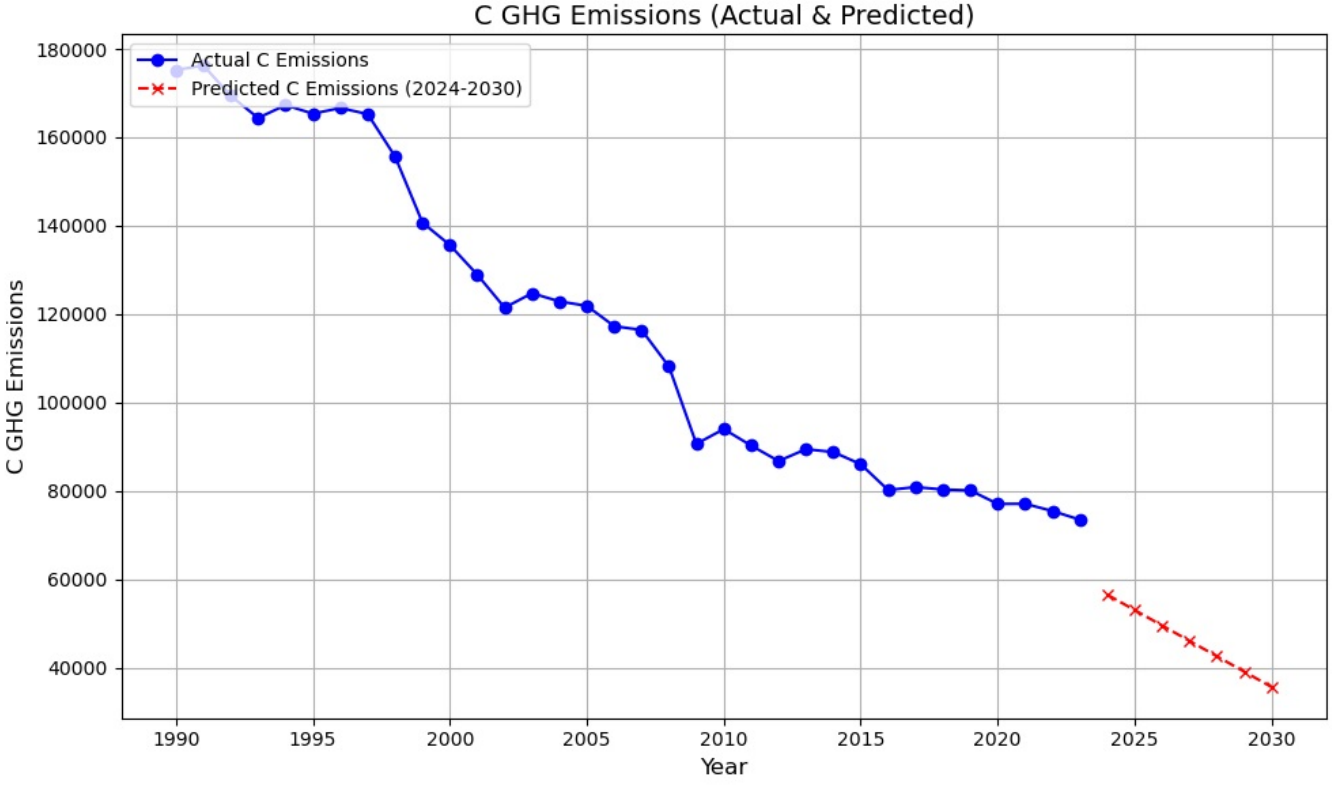
```
#Linear regression on industries

years = table1.iloc[1:, 0].values.reshape(-1, 1)

# Loop through each industry and apply linear regression
industry_names = table1.columns[1:-1]

predictions = {}  # Array to store predictions for each industry

for industry in industry_names:
    emissions = table1[industry].iloc[1:].values

    # Train the linear regression model
    model = LinearRegression()
    model.fit(years, emissions)

    # Predict the emissions for the years 2024 to 2030
    future_years = np.array([2024, 2025, 2026, 2027, 2028, 2029, 2030]).reshape(-1, 1)
    predicted_emissions = model.predict(future_years)

    # Store predictions
    predictions[industry] = predicted_emissions

    # Plotting the results for current data
    plt.figure(figsize=(10, 6))

    # Plot actual emissions
    plt.plot(years, emissions, label=f'Actual {industry} Emissions', color='blue', marker='o')

    # Plot predicted emissions for 2024–2030 as a dotted line
    plt.plot(future_years, predicted_emissions, label=f'Predicted {industry} Emissions (2024-2030)', color='red

    # Adding labels and title
    plt.title(f'{industry} GHG Emissions (Actual & Predicted)', fontsize=14)
    plt.xlabel('Year', fontsize=12)
    plt.ylabel(f'{industry} GHG Emissions', fontsize=12)
    plt.legend(loc='upper left', fontsize=10)
    plt.grid(True)

    # Show plot
    plt.tight_layout()
    plt.show()

    # Print predicted values for 2024-2030
    print(f"Predicted {industry} GHG Emissions for 2024-2030:")
    for i, year in enumerate(future_years):
        print(f"  {year[0]}: {predicted_emissions[i]:.2f}")
```



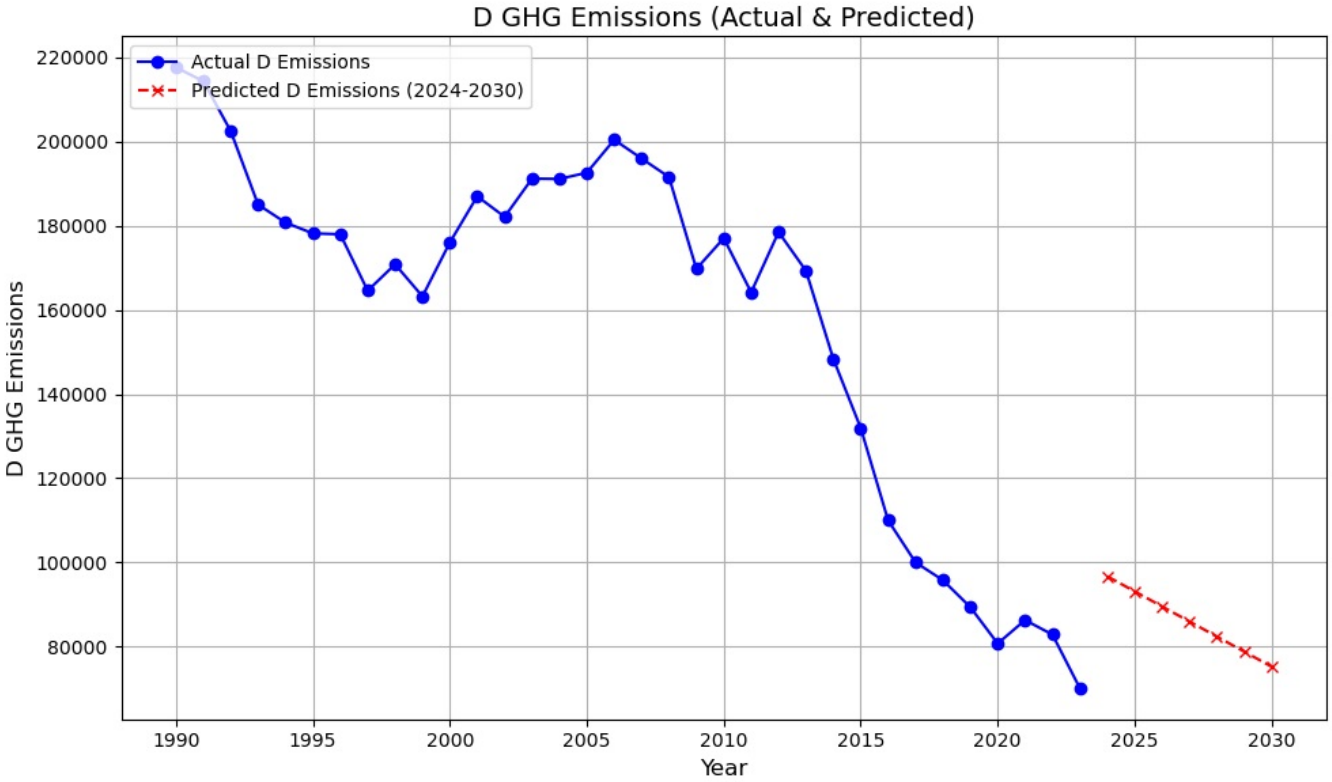
```
Predicted A GHG Emissions for 2024-2030:
  2024: 49003.57
  2025: 48807.12
  2026: 48610.67
  2027: 48414.22

  2028: 48217.78
  2029: 48021.33
  2030: 47824.88
```



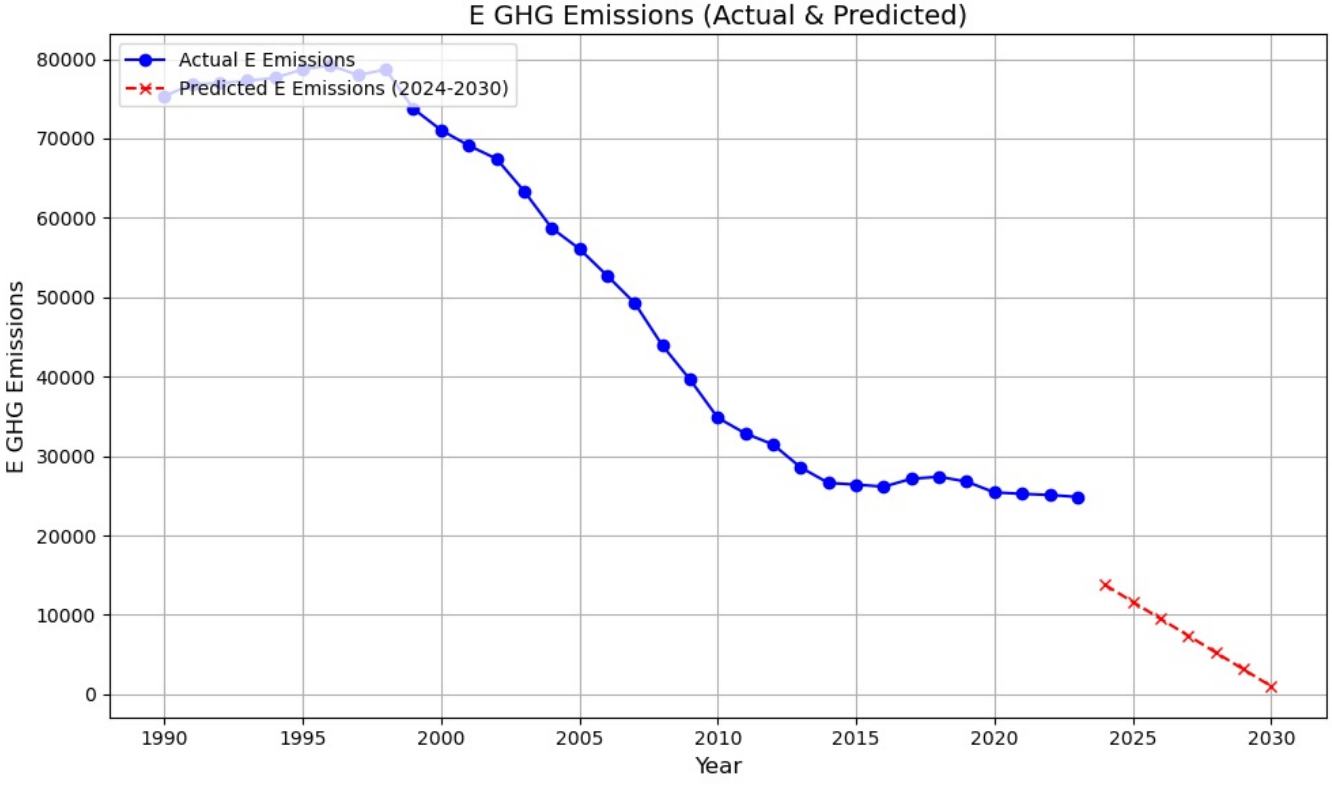

```
Predicted B GHG Emissions for 2024-2030:
  2024: 12426.71
  2025: 11306.44
  2026: 10186.17
  2027: 9065.91
  2028: 7945.64
  2029: 6825.37
  2030: 5705.10
```



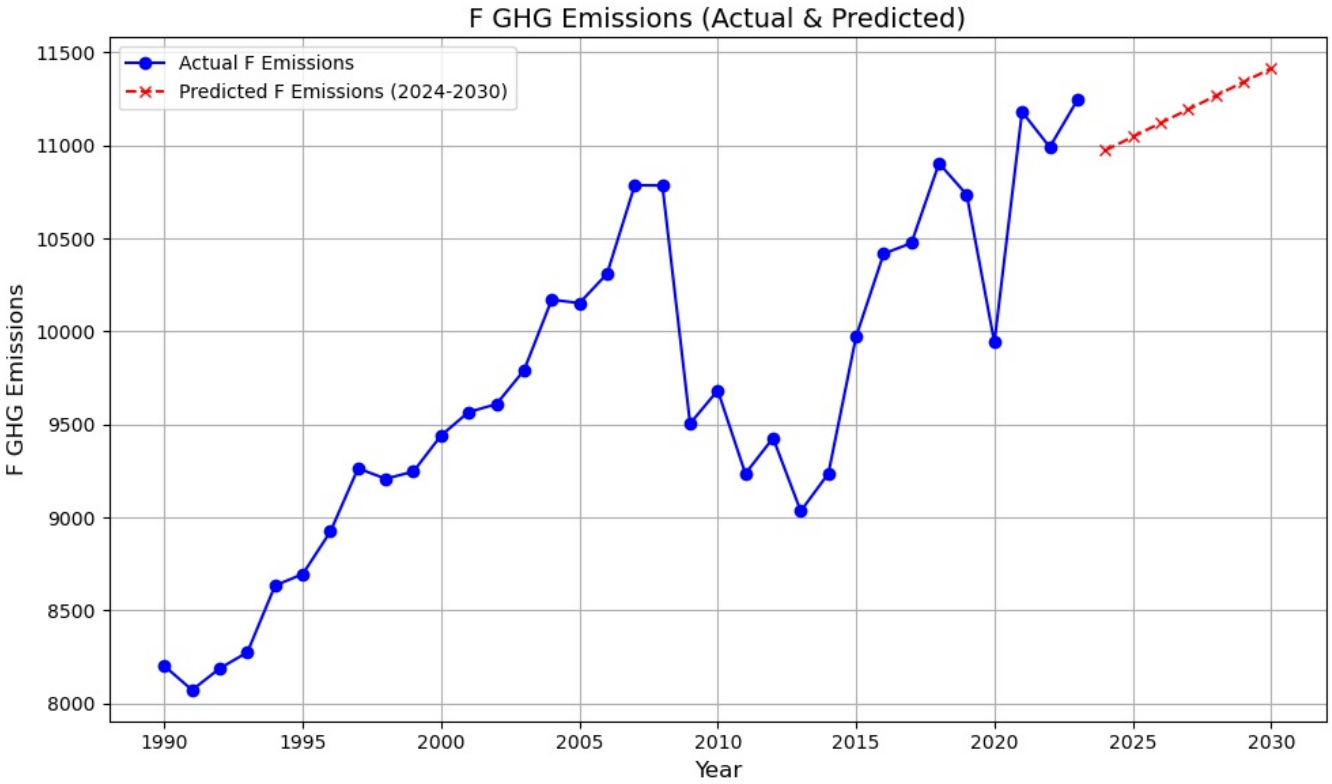
```
Predicted C GHG Emissions for 2024-2030:
  2024: 56525.20
  2025: 53042.77
  2026: 49560.34
  2027: 46077.91
  2028: 42595.49

  2029: 39113.06
  2030: 35630.63
```



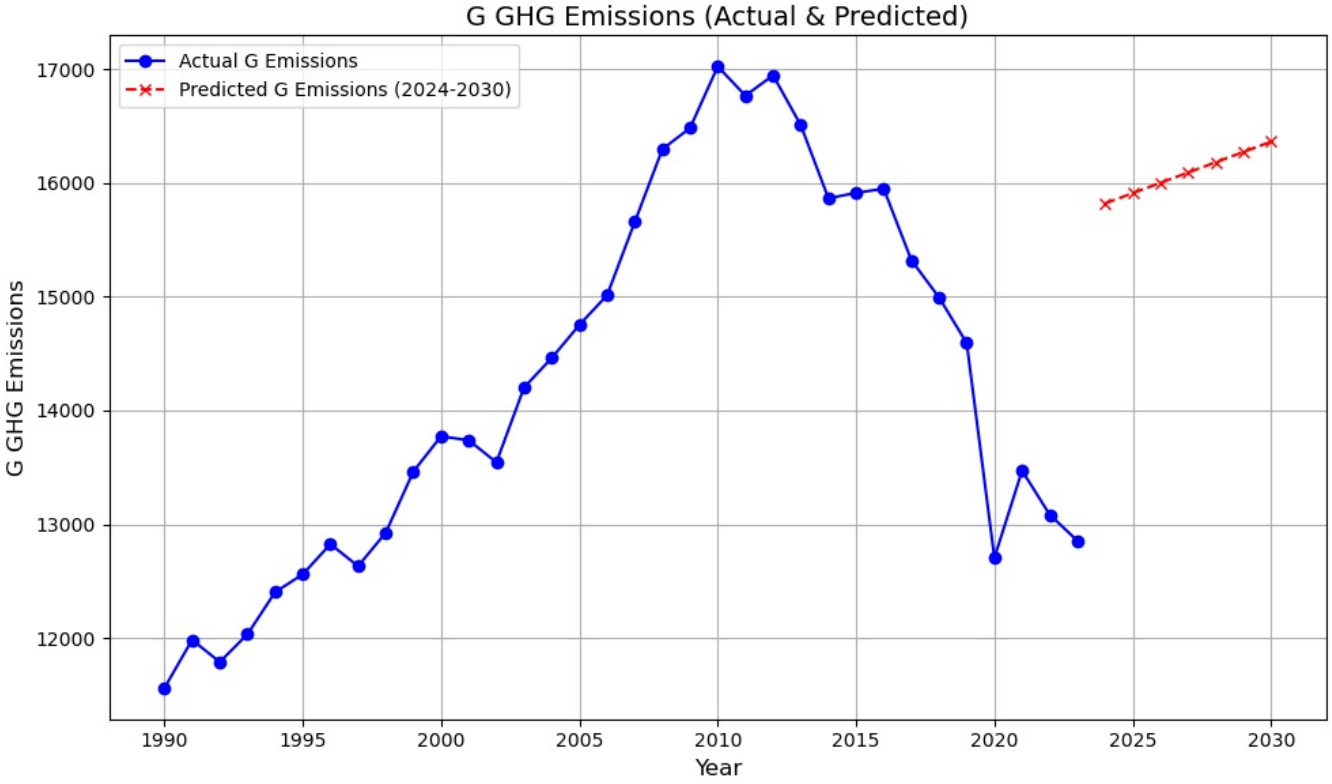

```
Predicted D GHG Emissions for 2024-2030:
  2024: 96706.14
  2025: 93128.37
  2026: 89550.59
  2027: 85972.81
  2028: 82395.04
  2029: 78817.26
  2030: 75239.49
```



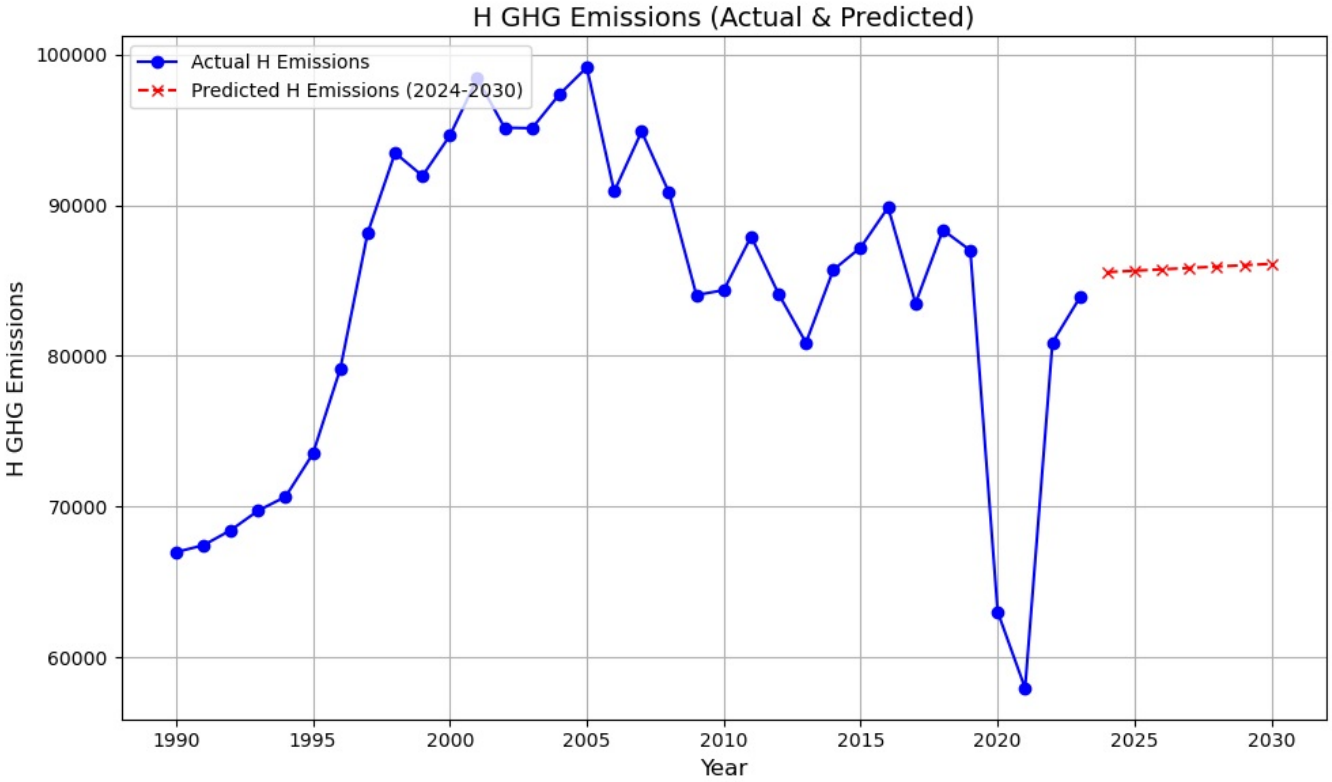
```
Predicted E GHG Emissions for 2024-2030:
  2024: 13780.29
  2025: 11655.65
  2026: 9531.01
  2027: 7406.37
  2028: 5281.73
  2029: 3157.08
  2030: 1032.44
```



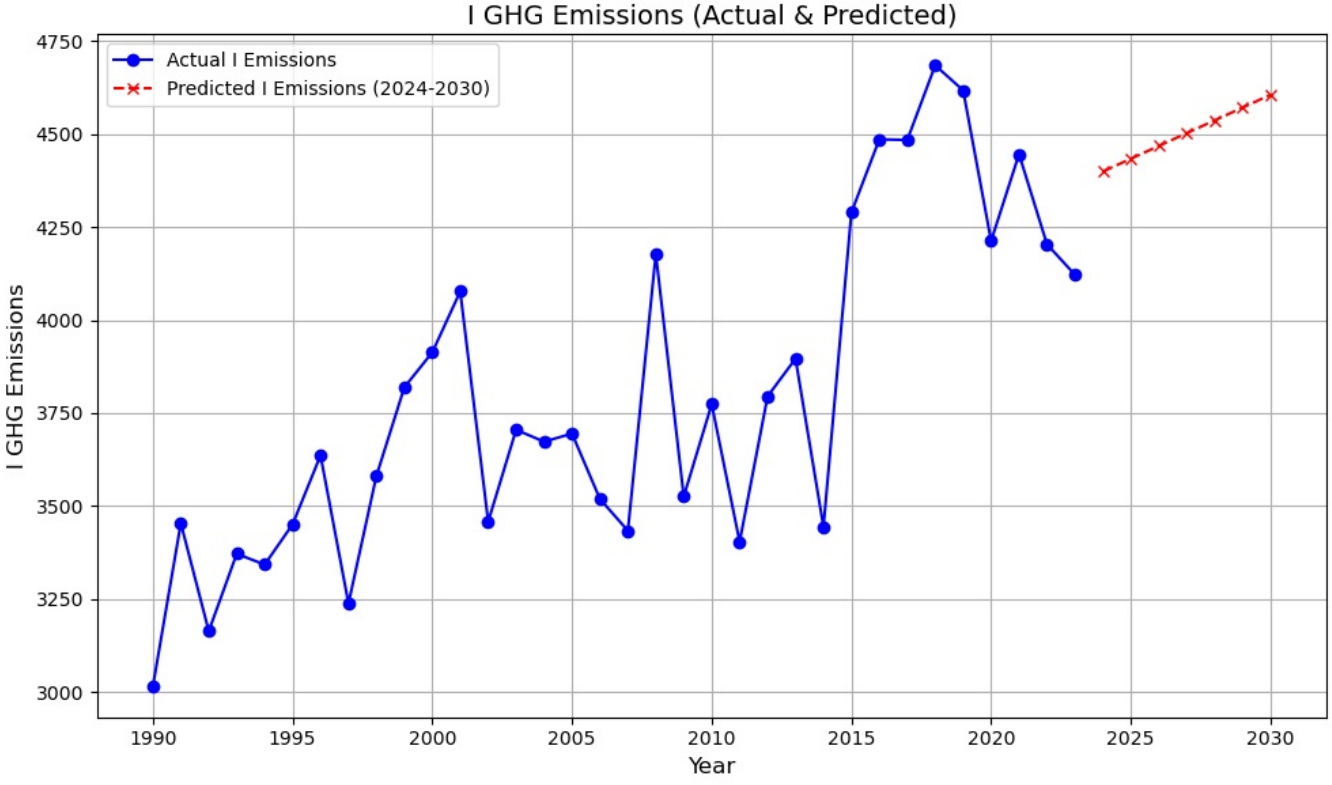

```
Predicted F GHG Emissions for 2024-2030:
  2024: 10972.22
  2025: 11045.76
  2026: 11119.31
  2027: 11192.86
  2028: 11266.41
  2029: 11339.96
  2030: 11413.50
```



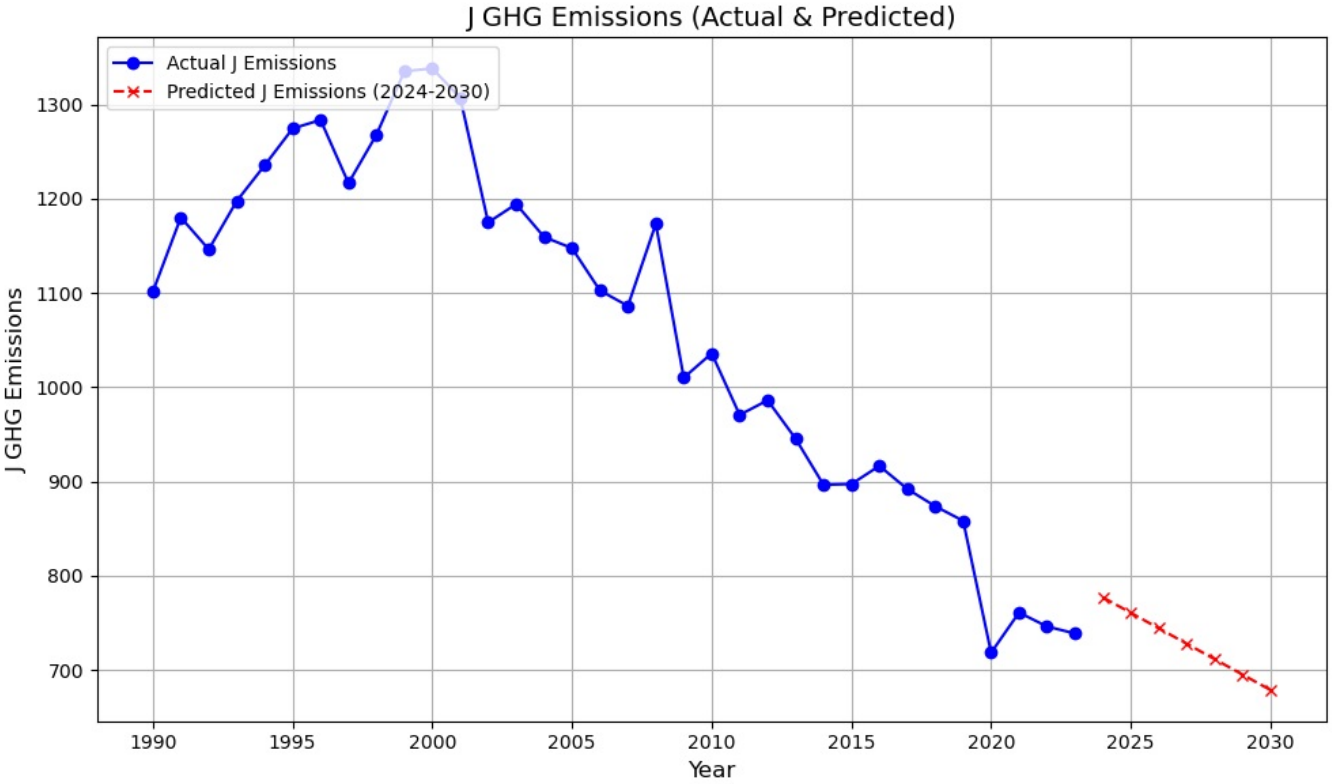
```
Predicted G GHG Emissions for 2024-2030:
  2024: 15820.22
  2025: 15910.62
  2026: 16001.01
  2027: 16091.41
  2028: 16181.80

  2029: 16272.20
  2030: 16362.59
```



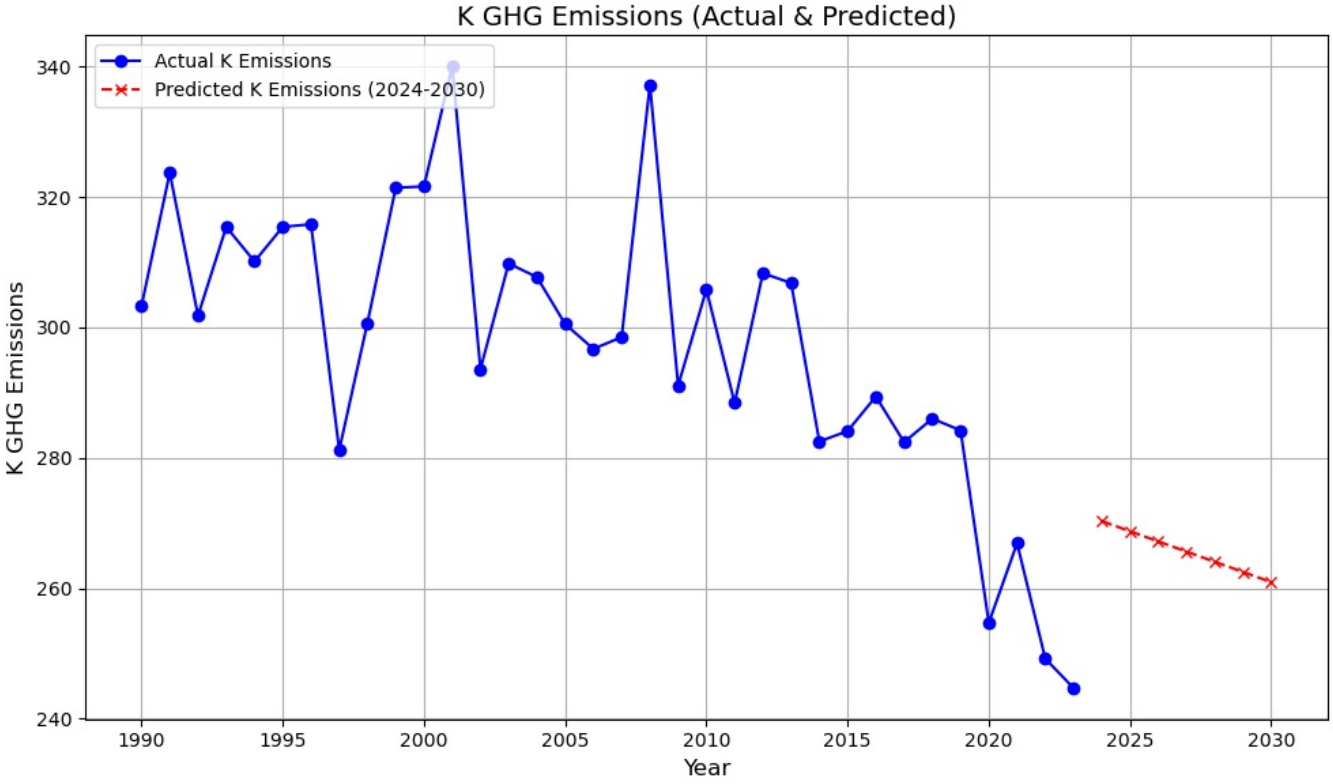

```
Predicted H GHG Emissions for 2024-2030:
  2024: 85557.14
  2025: 85648.70
  2026: 85740.27
  2027: 85831.83
  2028: 85923.39
  2029: 86014.95
  2030: 86106.51
```



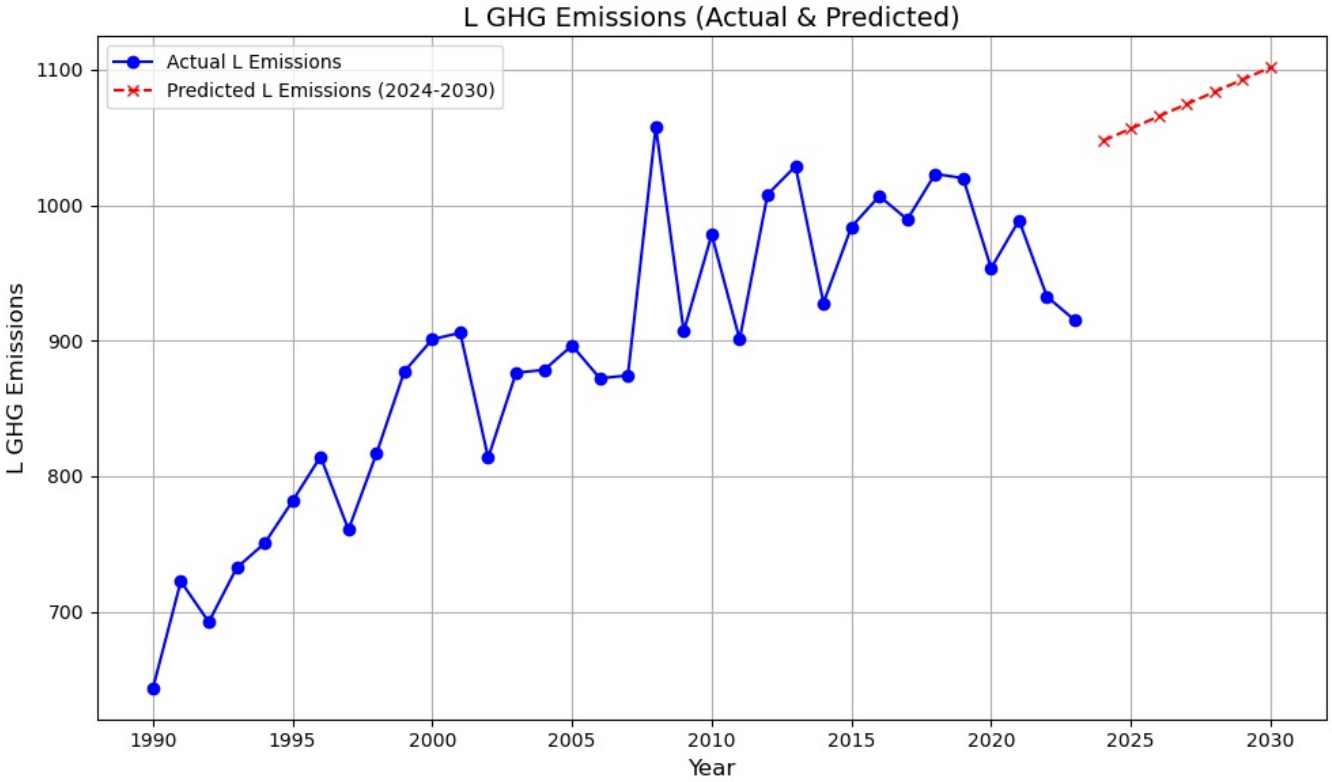
```
Predicted I GHG Emissions for 2024-2030:
  2024: 4399.63
  2025: 4434.04
  2026: 4468.46
  2027: 4502.88
  2028: 4537.29

  2029: 4571.71
  2030: 4606.13
```



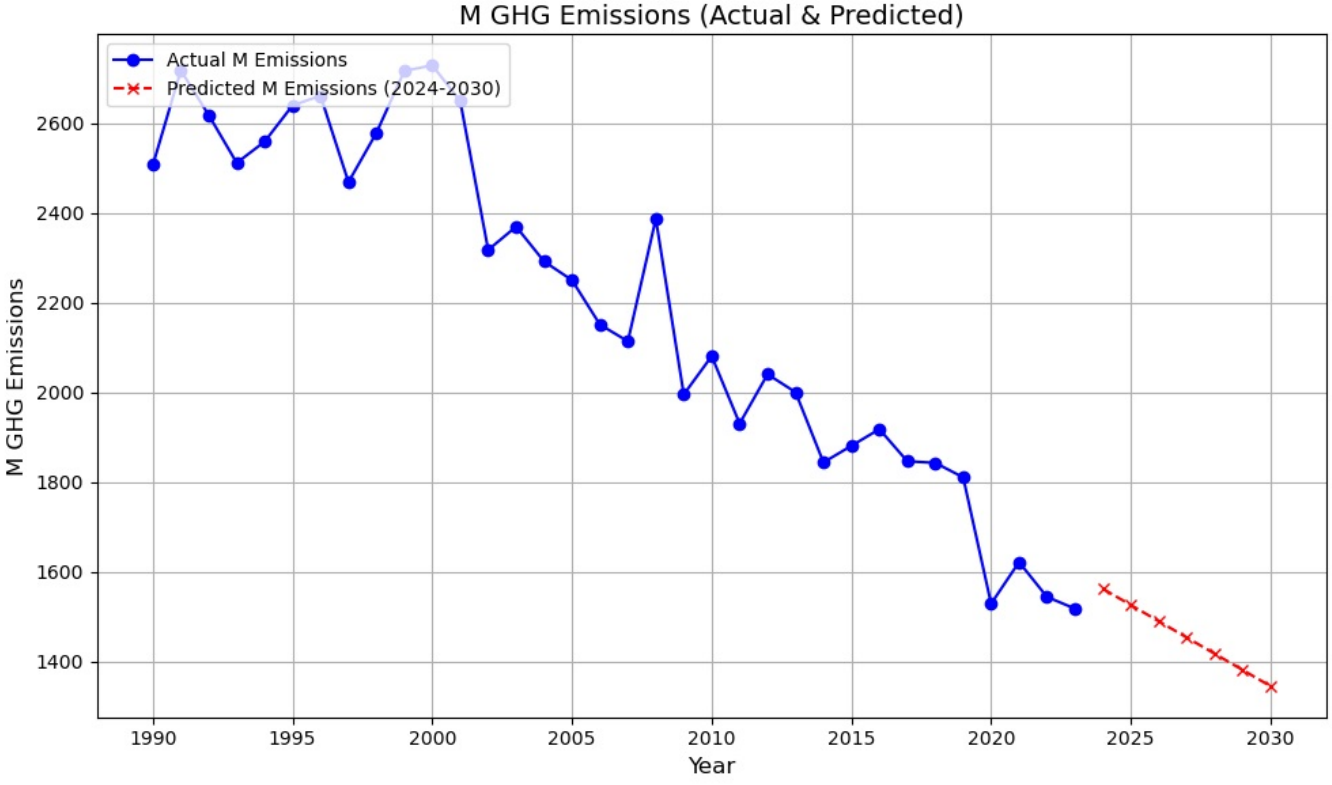

```
Predicted J GHG Emissions for 2024-2030:
  2024: 776.98
  2025: 760.60
  2026: 744.21
  2027: 727.82
  2028: 711.44
  2029: 695.05
  2030: 678.66
```



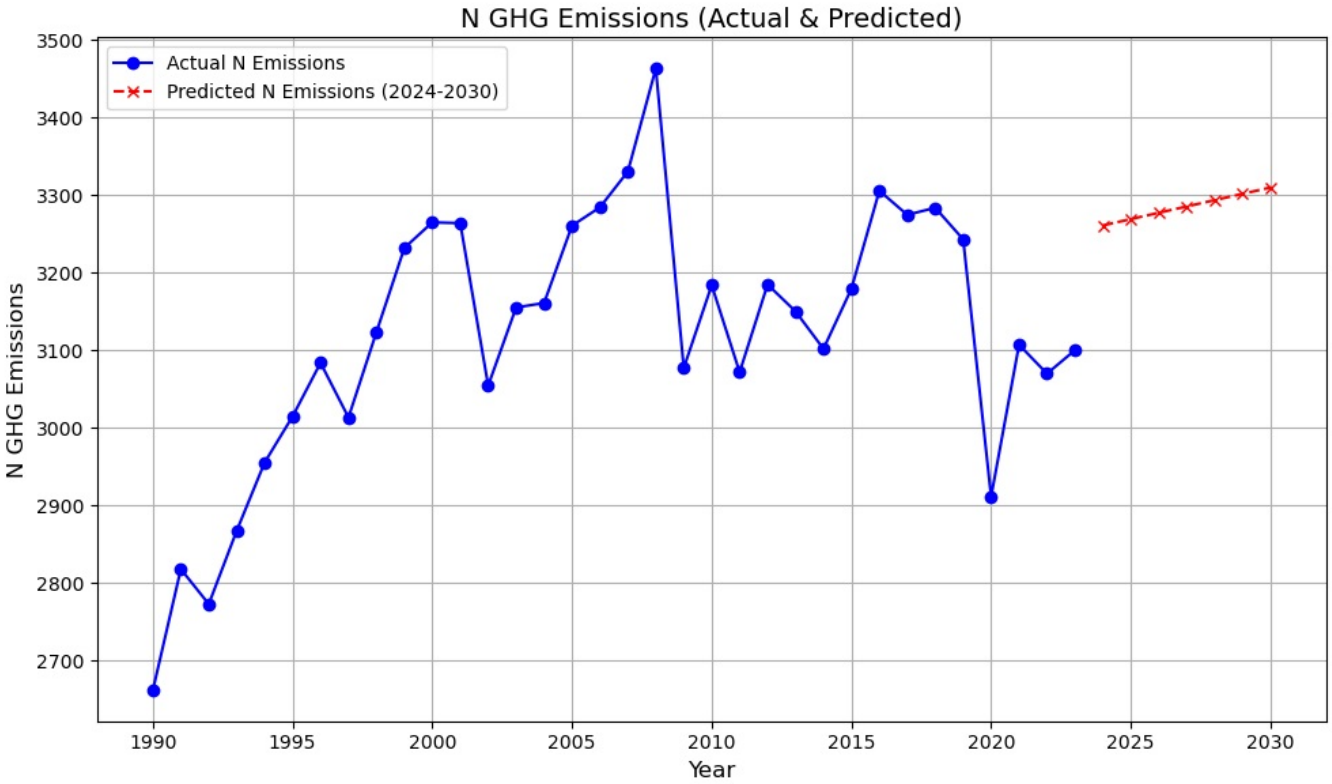
```
Predicted K GHG Emissions for 2024-2030:
  2024: 270.30
  2025: 268.74
  2026: 267.18
  2027: 265.62
  2028: 264.05

  2029: 262.49
  2030: 260.93
```



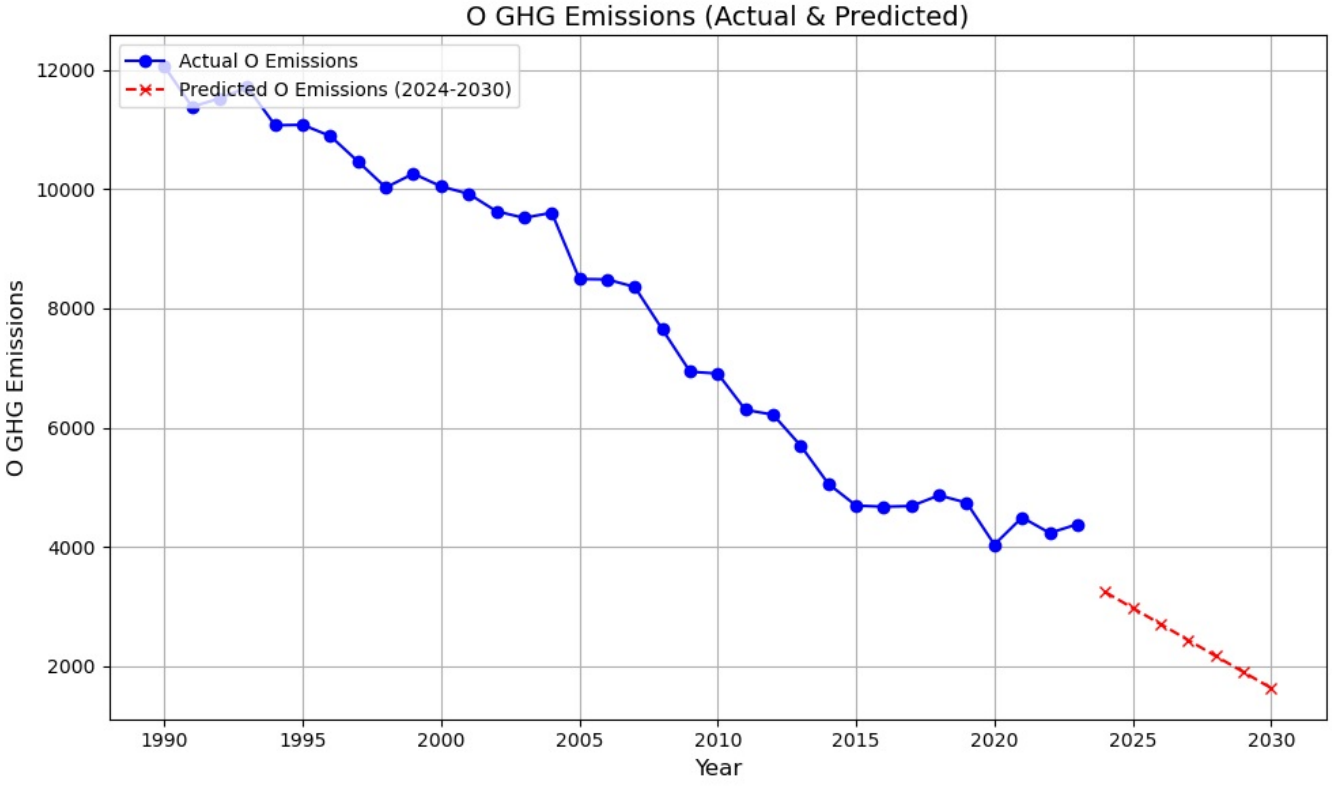

```
Predicted L GHG Emissions for 2024-2030:
  2024: 1047.36
  2025: 1056.40
  2026: 1065.44
  2027: 1074.48
  2028: 1083.52
  2029: 1092.56
  2030: 1101.60
```



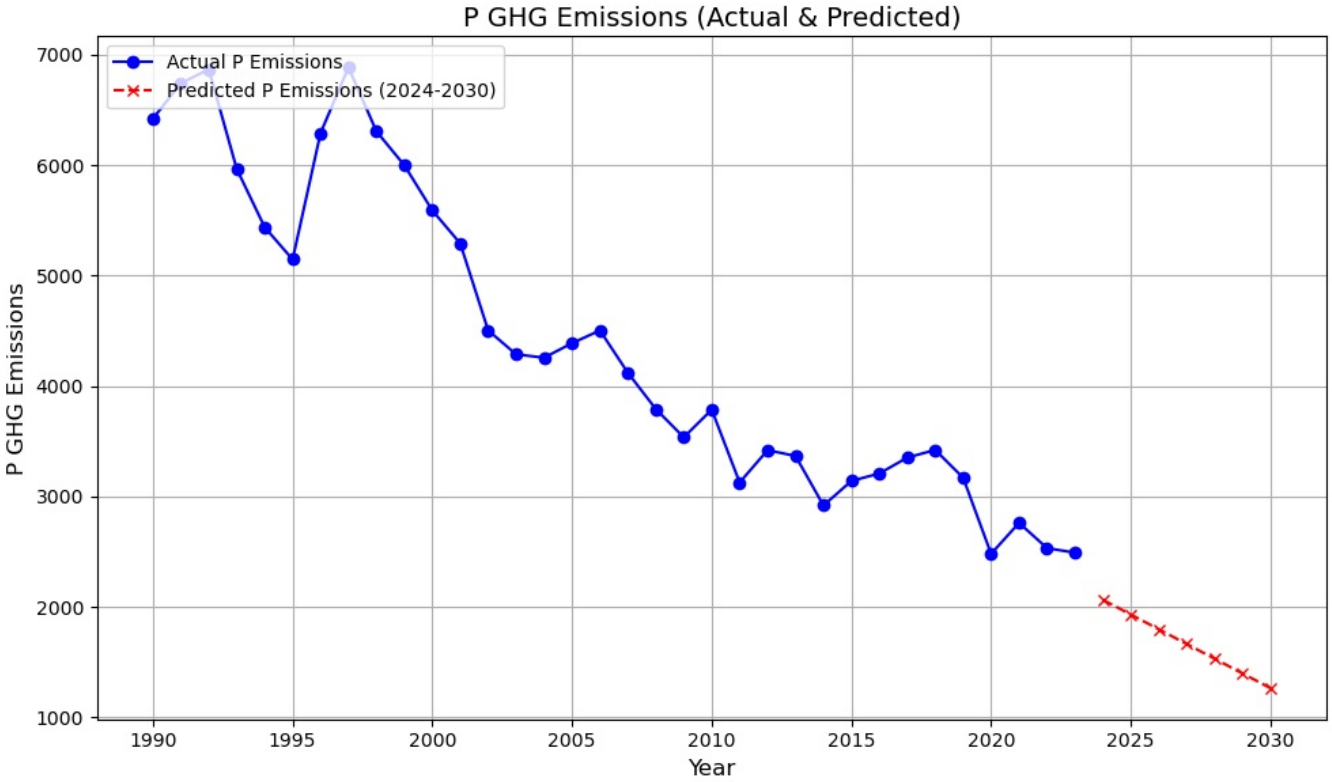
```
Predicted M GHG Emissions for 2024-2030:
  2024: 1561.75
  2025: 1525.56
  2026: 1489.37
  2027: 1453.18
  2028: 1416.99

  2029: 1380.80
  2030: 1344.61
```



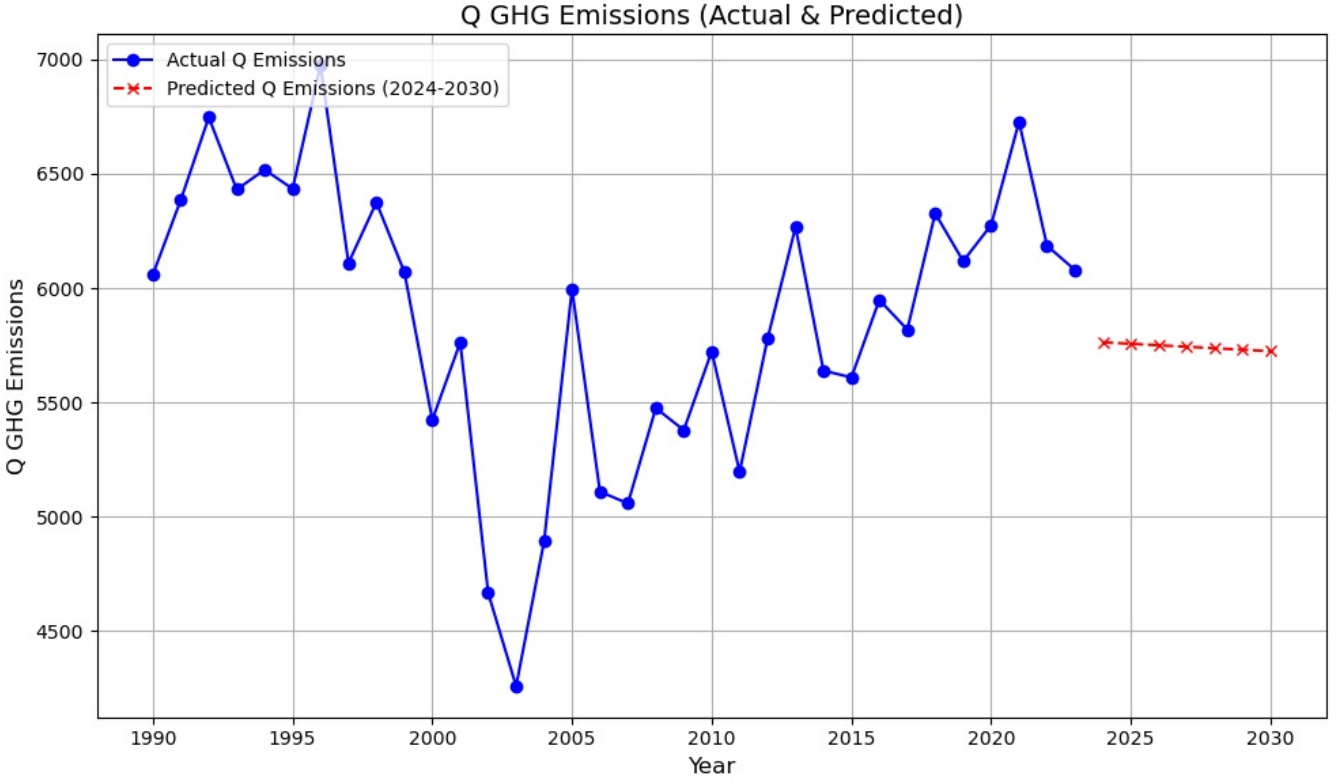

```
Predicted N GHG Emissions for 2024-2030:
  2024: 3261.24
  2025: 3269.38
  2026: 3277.53
  2027: 3285.67
  2028: 3293.82
  2029: 3301.96
  2030: 3310.11
```



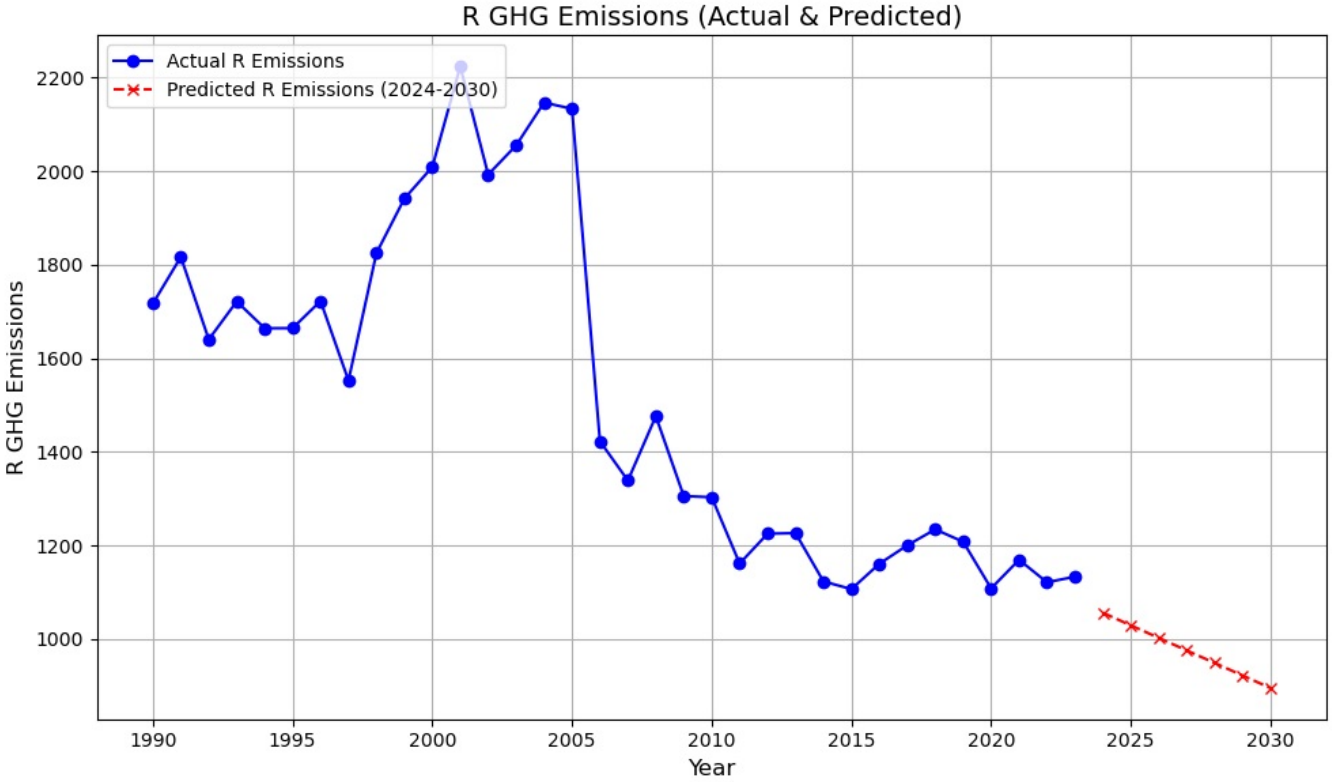
```
Predicted O GHG Emissions for 2024-2030:
  2024: 3244.08
  2025: 2975.52
  2026: 2706.96
  2027: 2438.39
  2028: 2169.83

  2029: 1901.26
  2030: 1632.70
```



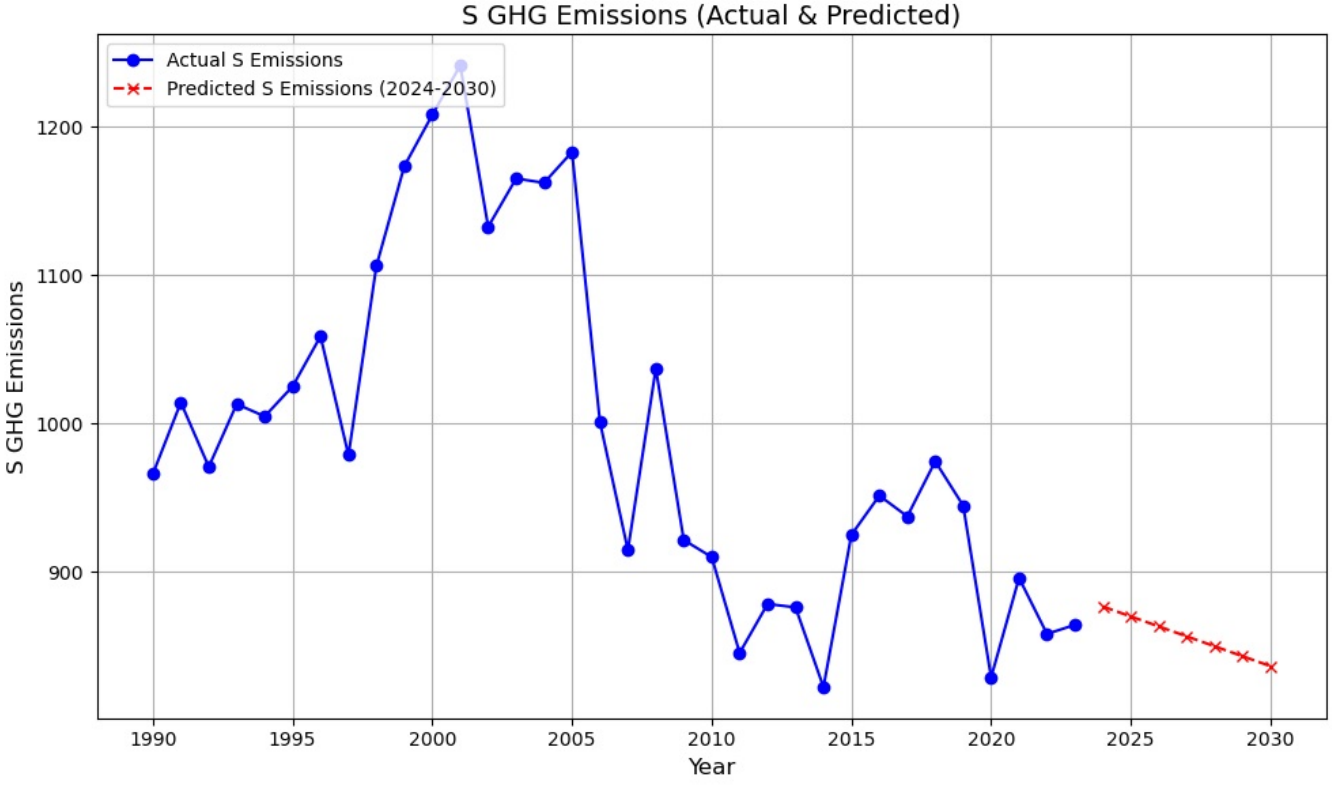

```
Predicted P GHG Emissions for 2024-2030:
  2024: 2063.89
  2025: 1930.58
  2026: 1797.26
  2027: 1663.94
  2028: 1530.63
  2029: 1397.31
  2030: 1264.00
```



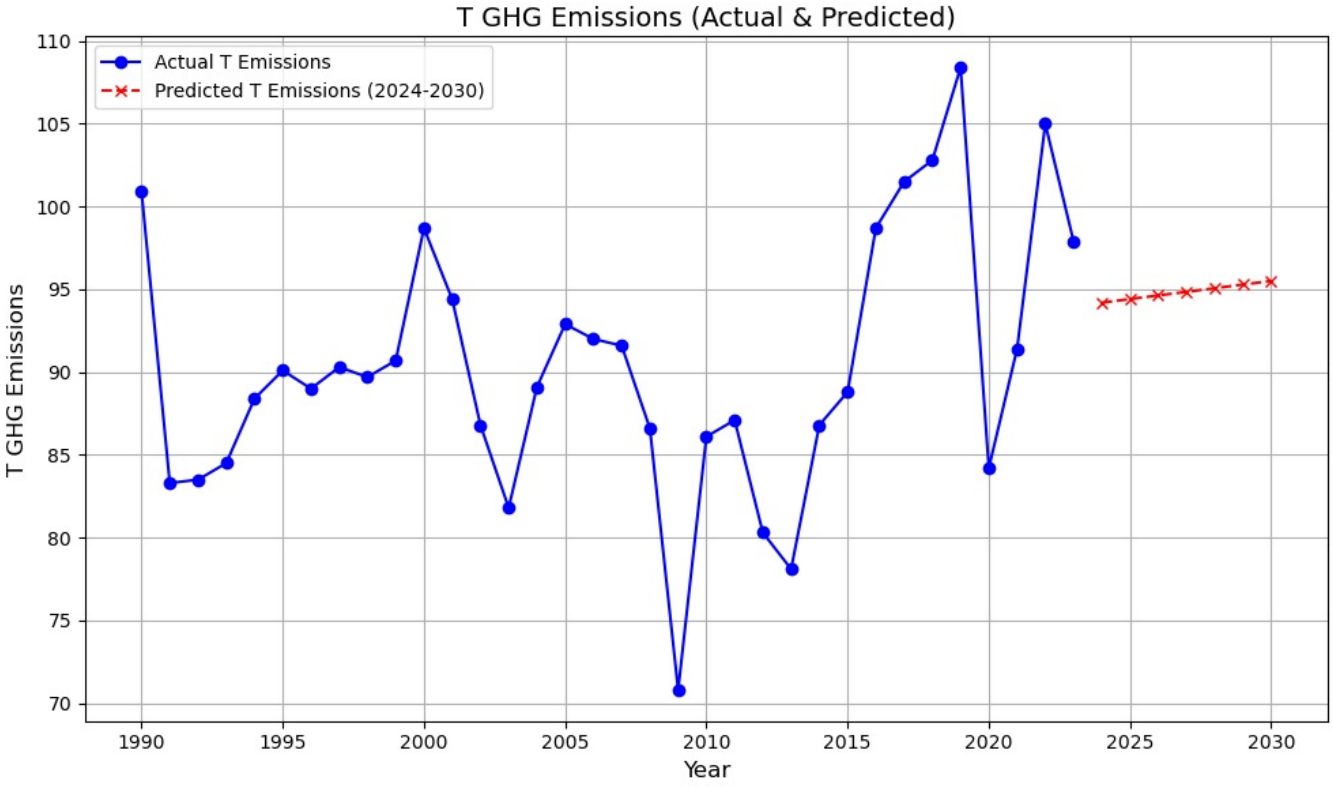
```
Predicted Q GHG Emissions for 2024-2030:
  2024: 5762.97
  2025: 5756.46
  2026: 5749.94
  2027: 5743.42
  2028: 5736.91

  2029: 5730.39
  2030: 5723.87
```



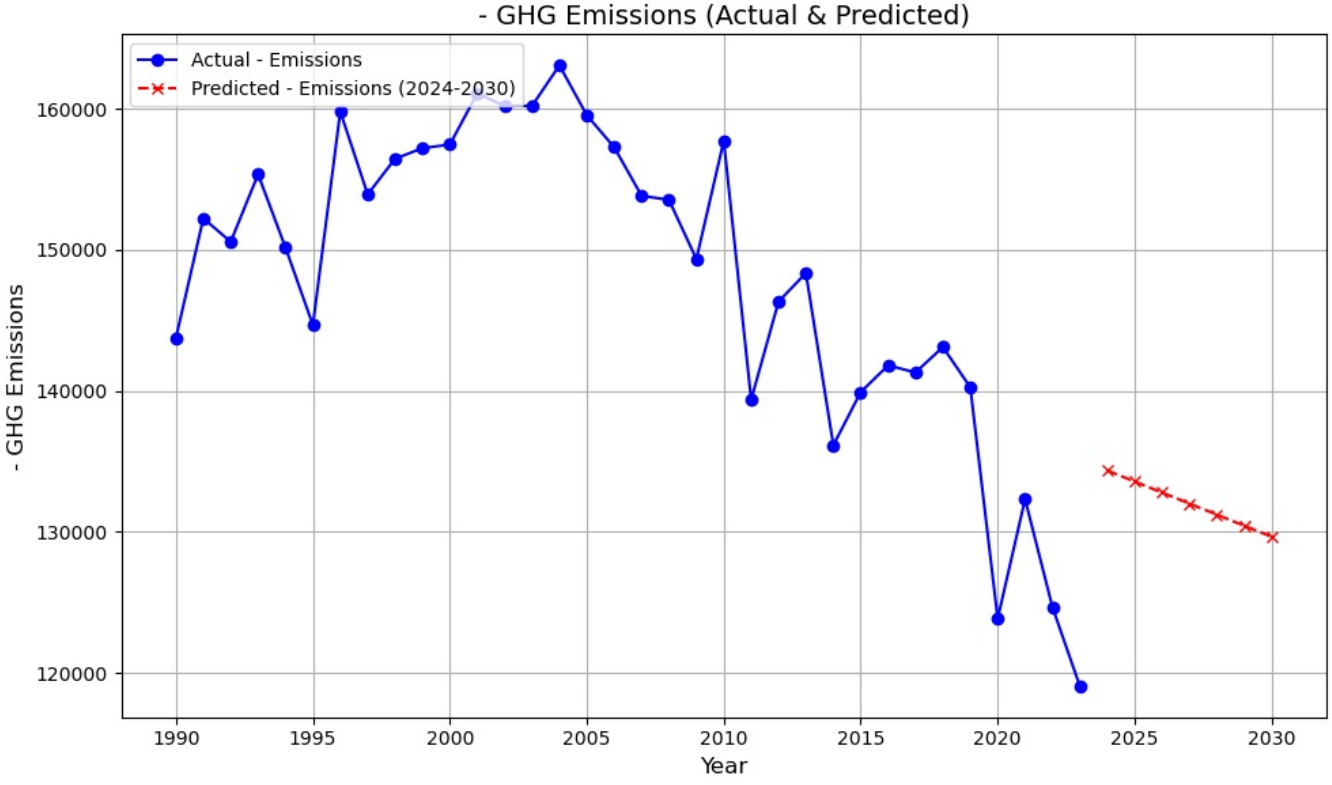

```
Predicted R GHG Emissions for 2024-2030:
  2024: 1056.09
  2025: 1029.30
  2026: 1002.51
  2027: 975.71
  2028: 948.92
  2029: 922.12
  2030: 895.33
```



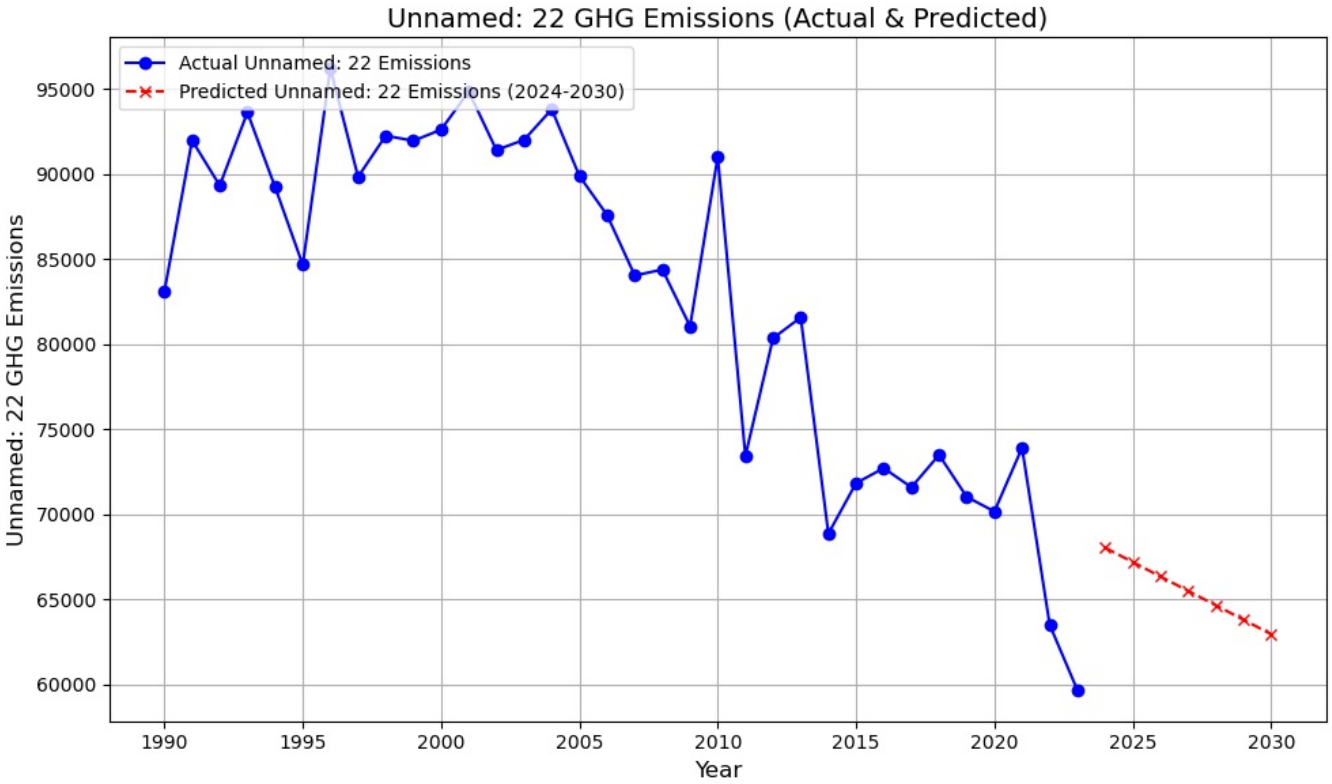
```
Predicted S GHG Emissions for 2024-2030:
  2024: 876.55
  2025: 869.85
  2026: 863.16
  2027: 856.46
  2028: 849.76

  2029: 843.07
  2030: 836.37
```



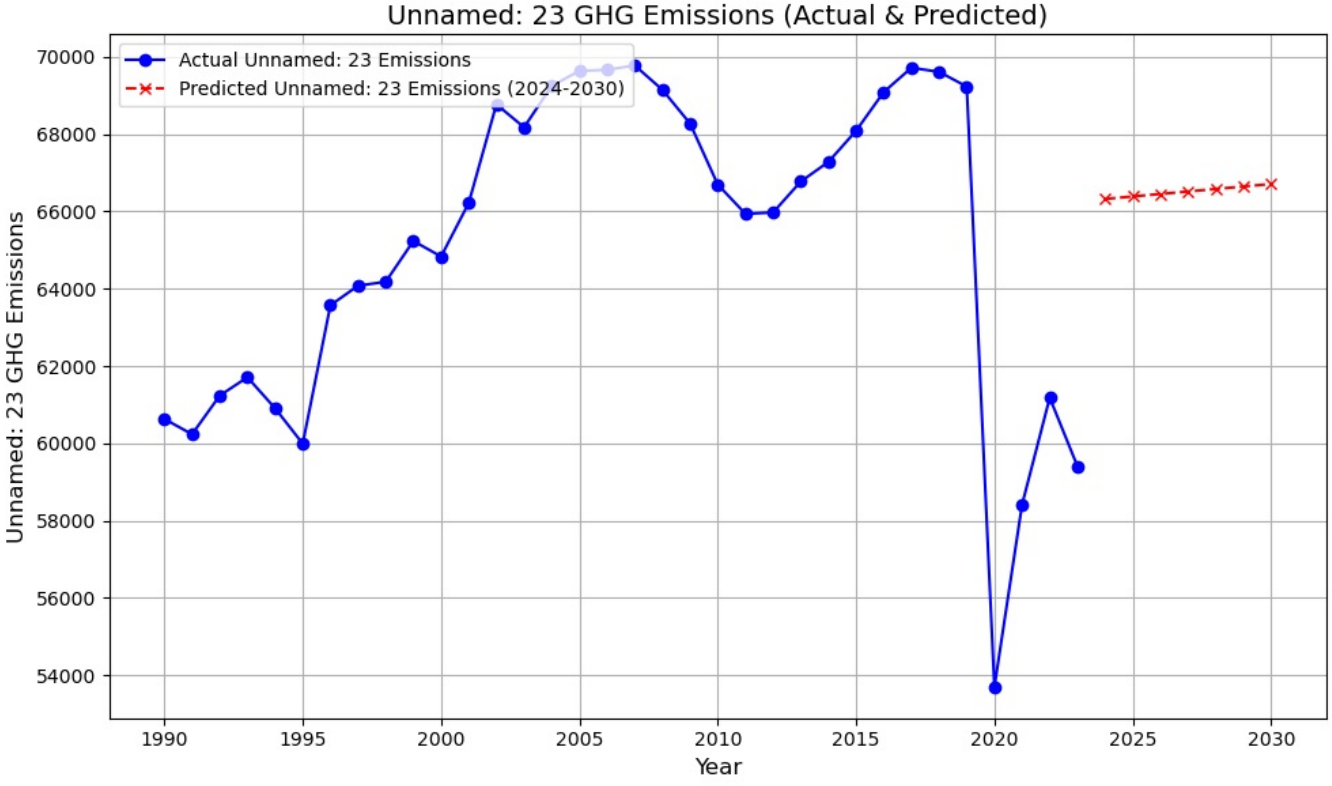

```
Predicted T GHG Emissions for 2024-2030:
  2024: 94.19
  2025: 94.41
  2026: 94.63
  2027: 94.85
  2028: 95.07
  2029: 95.29
  2030: 95.51
```



```
Predicted - GHG Emissions for 2024-2030:
  2024: 134356.72
  2025: 133574.17
  2026: 132791.63
  2027: 132009.08
  2028: 131226.53

  2029: 130443.99
  2030: 129661.44
```




```
Predicted Unnamed: 22 GHG Emissions for 2024-2030:
  2024: 68032.63
  2025: 67185.61
  2026: 66338.59
  2027: 65491.57
  2028: 64644.55
  2029: 63797.53
  2030: 62950.51
```



```
Predicted Unnamed: 23 GHG Emissions for 2024-2030:
  2024: 66324.13
  2025: 66388.61
  2026: 66453.08
  2027: 66517.56
  2028: 66582.04

  2029: 66646.51
  2030: 66710.99
```

In [ ]:

In [431…

```
#References
#[1]www.ons.gov.uk. (n.d.). Atmospheric emissions: greenhouse gases by industry and gas - Office for National S
#[2]Suma Katabattuni (2024). Simple pandas read_csv tricks to boost speed up to 250x. [online] Medium. Availabl
#[3]Naveen Nelamali (2022). Pandas iloc[] Usage with Examples. [online] Spark By {Examples}. Available at: http
#[4]Mondal, R. (2024). Z-Score to identify and remove outliers | Exploratory Data Analysis. [online] Medium. Av
#[5]freshhotdata (2023). Predicting Greenhouse Gas Emissions with Machine Learning. [online] Medium. Available
#[6]Muz (2016). I get a linear regression using the SVR by python scikit-learn when the data is not linear. [on
```

In [ ]: