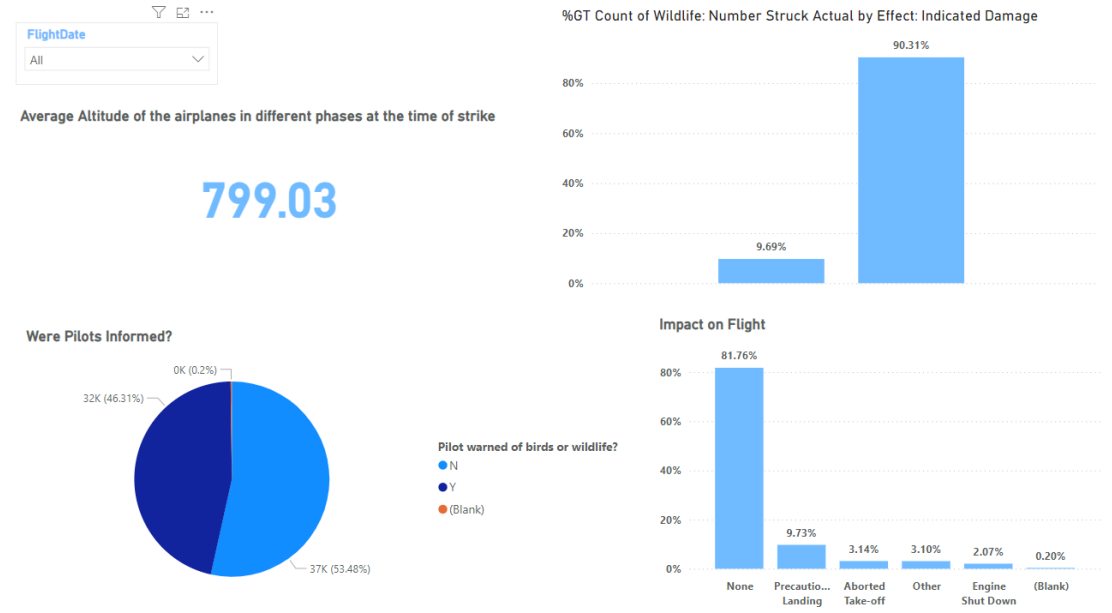
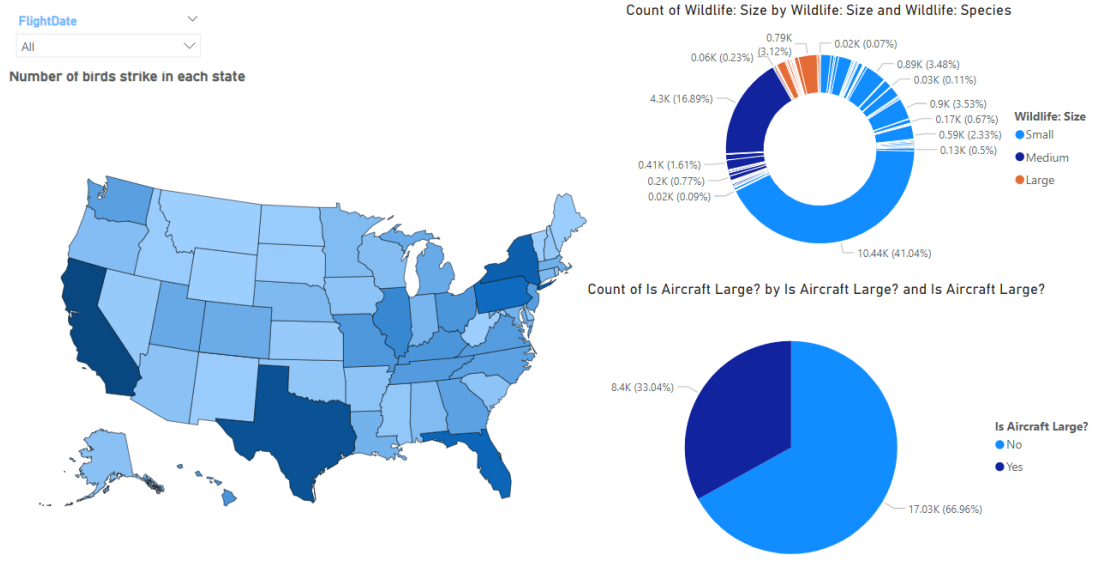
UNIFIED MENTOR

Name:S.SUMITH KUMAR (Business Analyst Intern)
UNID:UMIP19852
Time period : 20-8-24 to 20-10-24

Project 1:**Data Visualization of Bird Strikes between 2000 – 2011**

The dashboard is created using Power BI

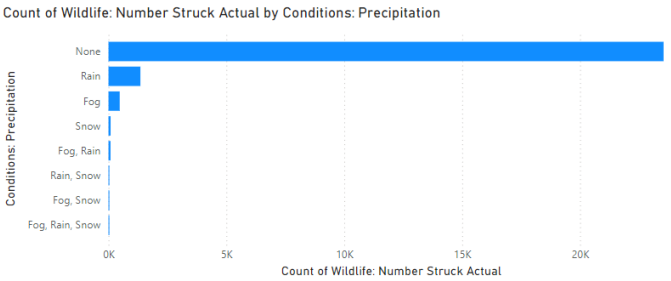


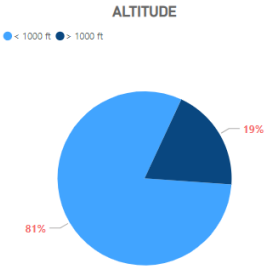


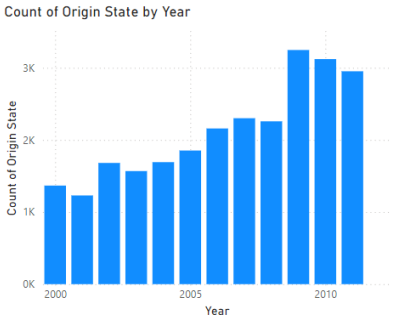
FlightDate

All

Airport: Name	Count of Wildlife: Number struck
ABERDEEN REGIONAL AR	17
ABILENE REGIONAL ARPT	9
ABRAHAM LINCOLN CAPITAL ARPT	30
ADAMS COUNTY- LEGION FIELD	1
ADAMS FIELD ARPT	67
ADDINGTON FIELD ARPT	1
ADDISON	32
ADIRONDAK REGIONAL ARPT	1
AIRBORNE AIRPARK	112
AKRON-CANTON MUNICIPAL	25
ALBANY INTL	96
ALBERT J ELLIS	11
ALBERT LEA MUNICIPAL ARPT	2
ALBERT WHITTED	1
ALBUQUERQUE INTL	82
ALEXANDRIA INTL	3
ALLEGHENY COUNTY ARPT	12
ALLENTOWN QUEEN CITY MUNICIPAL	3
ALLIANCE MUNICIPAL ARPT	2
ALPENA COUNTY REG	1
ALTOONA-BLAIR COUNTY	1
AMARILLO INTL	19
AMSTERDAM SCHIPHOL	2
ANDERSON REGIONAL	1
ANDREWS AIR FORCE BASE	1
ANIAK AIRPORT	1
Total	**25429**







Data Visualization of Bird Strikes between 2000 – 2011

FlightDate

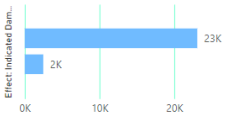
All

Count of Bird Strik

26K

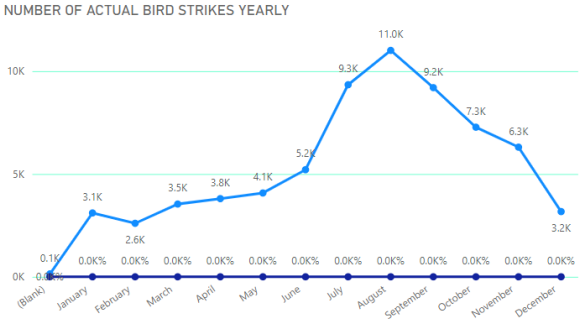
TOTAL COST IN REPAIRMENTS

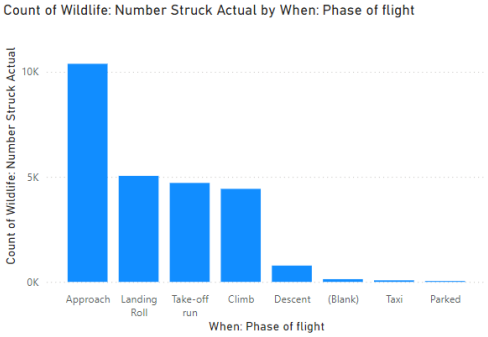
142M



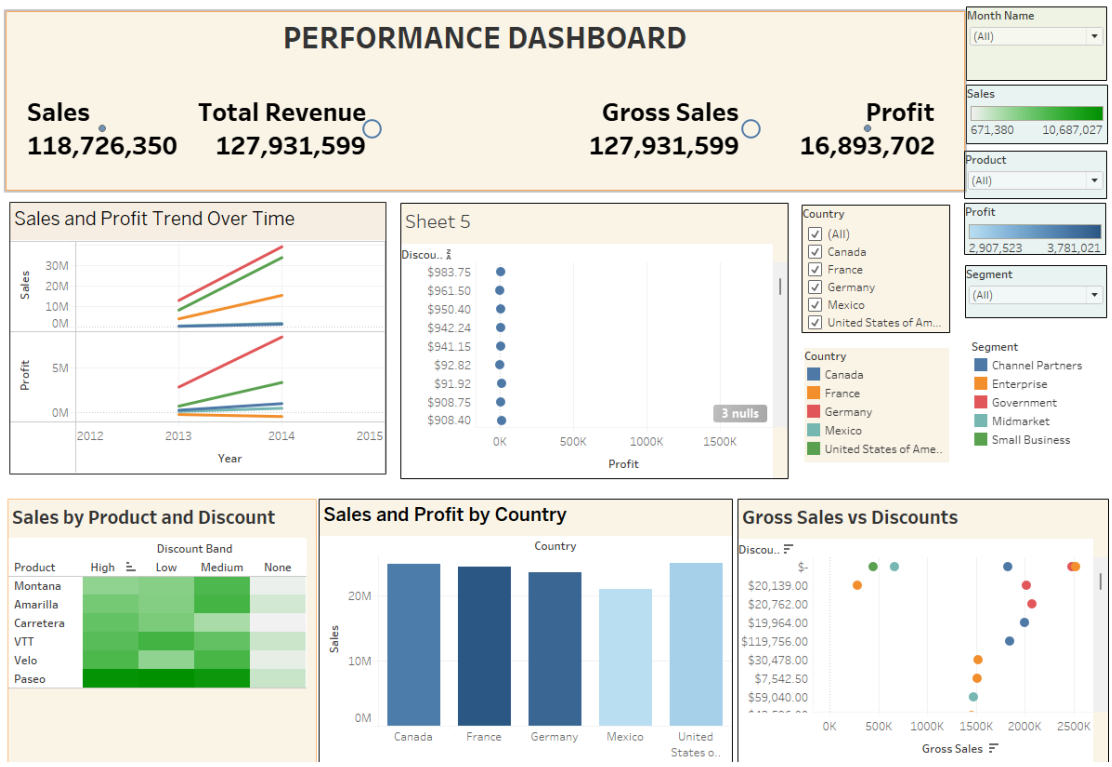
Count of Number of People Injured

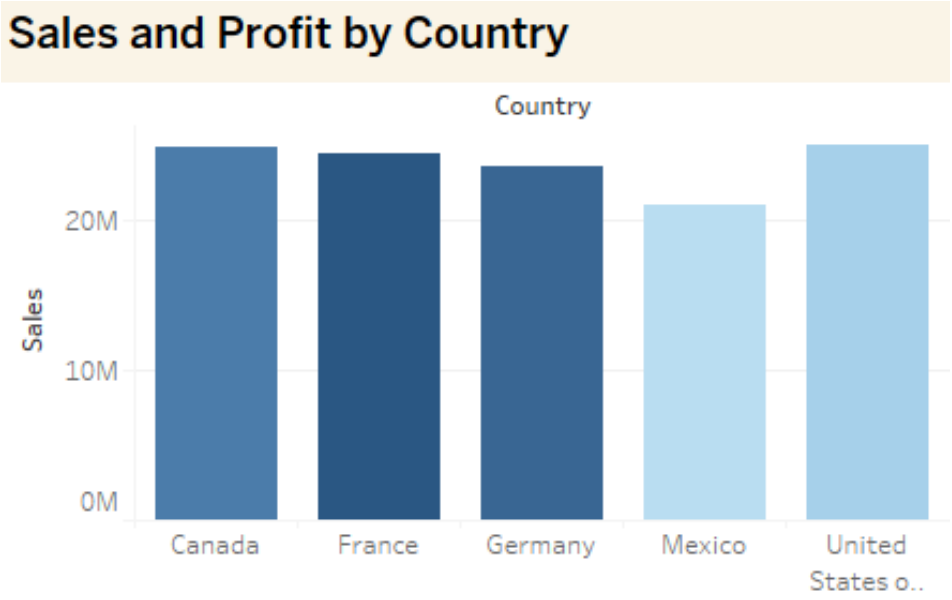
25.56K



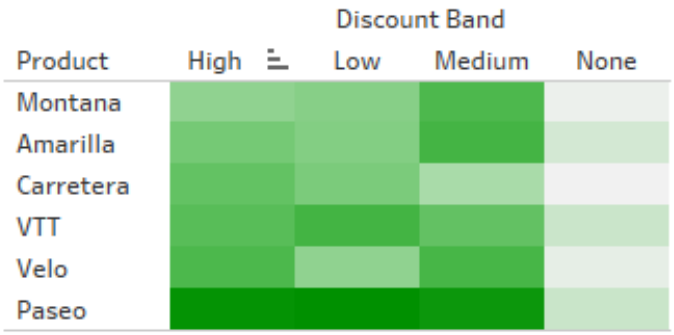


Project 2:**Financial Performance Dashboard**



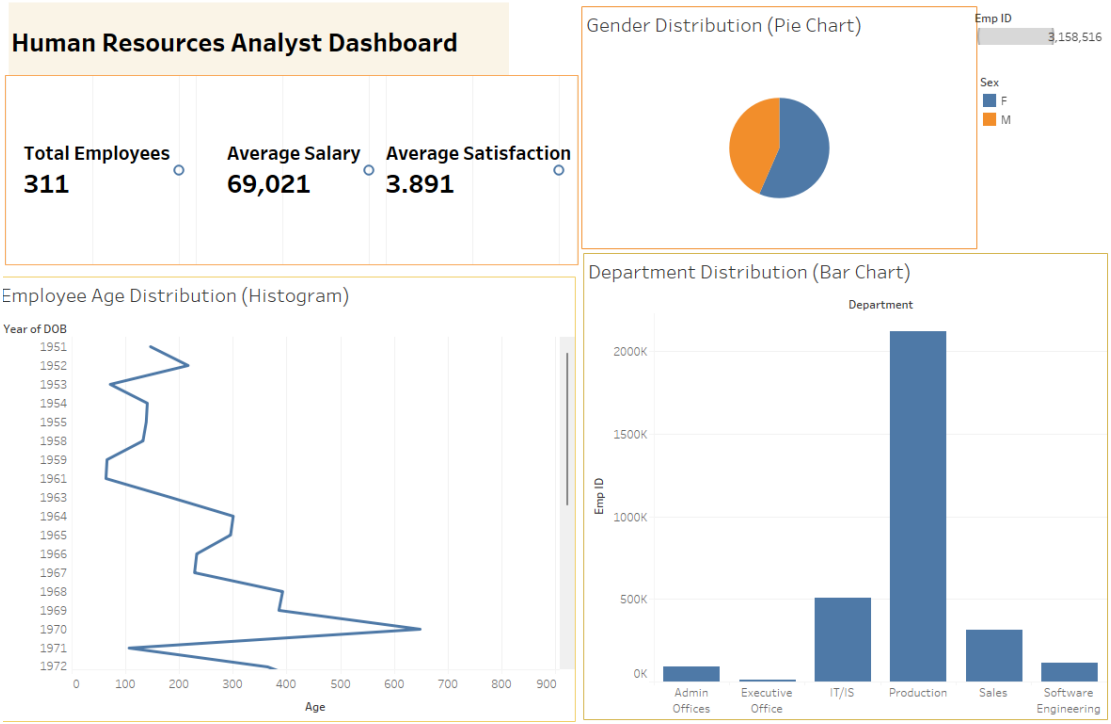


Sales by Product and Discount



Project 3:**Human Resources Analyst**

The dashboard is created using Tableau



Project 4(extra):**Data Governance and Security Dashboard**

```
[1]: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns # data visualization library
import matplotlib.pyplot as plt

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
[3]: ESGData = pd.read_csv('C:\\Users\\Sumit\\Downloads\\ESGData.csv')
ESGSeries = pd.read_csv('C:\\Users\\Sumit\\Downloads\\ESGSeries.csv')
ESGCountry = pd.read_csv('C:\\Users\\Sumit\\Downloads\\ESGCountry.csv')
ESGSeries_Time = pd.read_csv('C:\\Users\\Sumit\\Downloads\\ESGSeries-Time.csv')
ESGCountry_Series = pd.read_csv('C:\\Users\\Sumit\\Downloads\\ESGCountry-Series.csv')
ESGFootNote = pd.read_csv('C:\\Users\\Sumit\\Downloads\\ESGFootNote.csv')
```

```
[4]: ESGData.head()
```

[4]:

	Series Code	Topic	Indicator Name	Short definition	Long definition	Unit of measure	Periodicity	Base Period	Other notes	Aggregation method	...	Notes from original source	General comments	Source	S con meth
0	AG.LND.AGRI.ZS	Environment: Land use	Agricultural land (% of land area)	NaN	Agricultural land refers to the share of land ...	NaN	Annual	NaN	NaN	Weighted average	...	NaN	NaN	Food and Agriculture Organization, electronic ...	Agri stil r
1	AG.LND.FRST.ZS	Environment: Land use	Forest area (% of land area)	NaN	Forest area is land under natural or planted s...	NaN	Annual	NaN	NaN	Weighted average	...	NaN	NaN	Food and Agriculture Organization, electronic ...	de bo pr
2	AG.PRD.FOOD.XD	Environment: Agricultural production	Food production index (2004-2006 = 100)	NaN	Food production index covers food crops that a...	NaN	Annual	2004-06	NaN	Weighted average	...	NaN	NaN	Food and Agriculture Organization, electronic ...	ag pr pr
3	CC.EST	Governance: Stability & Rule of Law	Control of Corruption: Estimate	NaN	Control of Corruption captures perceptions of	NaN	Annual	NaN	NaN	NaN	...	NaN	NaN	Detailed documentation of the WGI, interactive...	

[5]:
```
col = ESGData.columns
print(col)
y = ESGData.loc[19]
```

```
Index(['Series Code', 'Topic', 'Indicator Name', 'Short definition',
       'Long definition', 'Unit of measure', 'Periodicity', 'Base Period',
       'Other notes', 'Aggregation method', 'Limitations and exceptions',
       'Notes from original source', 'General comments', 'Source',
       'Statistical concept and methodology', 'Development relevance',
       'Related source links', 'Other web links', 'Related indicators',
       'License Type', 'Unnamed: 20'],
      dtype='object')
```

[6]:
```
print(ESGData.loc[19])
```

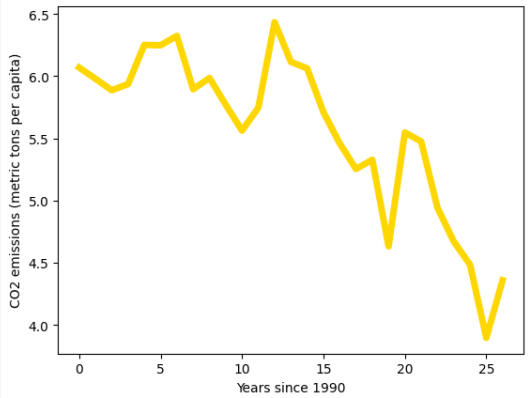
```
Series Code                                                  EN.CLC.HEAT.XD
Topic                                Environment: Environment/climate risk & resili...
Indicator Name                               Heat Index 35 (projected change in days)
Short definition                                                             NaN
Long definition                      Total count of days per year where the daily m...
Unit of measure                                                              NaN
Periodicity                                                                  NaN
Base Period                                                                  NaN
Other notes                          Refer to the Climate Knowledge Portal for impo...
Aggregation method                                                           NaN
Limitations and exceptions                                                   NaN
Notes from original source                                                   NaN
General comments                                                             NaN
Source                               World Bank, Climate Change Knowledge Portal (h...
Statistical concept and methodology  These days represent extremely uncomfortable c...
Development relevance                                                        NaN
Related source links                                                         NaN
Other web links                                                              NaN
Related indicators                                                           NaN
License Type                                                          CC BY-4.0
Unnamed: 20                                                                  NaN
Name: 19, dtype: object
```

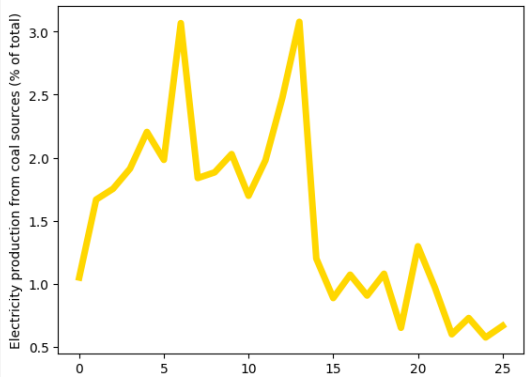
[8]:
```
print(ESGData.columns)
```

```
Index(['Series Code', 'Topic', 'Indicator Name', 'Short definition',
       'Long definition', 'Unit of measure', 'Periodicity', 'Base Period',
       'Other notes', 'Aggregation method', 'Limitations and exceptions',
       'Notes from original source', 'General comments', 'Source',
       'Statistical concept and methodology', 'Development relevance',
       'Related source links', 'Other web links', 'Related indicators',
       'License Type', 'Unnamed: 20'],
```

```
new = updated.columns
newlist = ['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code', '1960', '1961', '1962', '1963', '1964', '1965', '1966', '1967', '1968', '1
updated = ESGData.drop(newlist, axis = 1)
updated = updated.transpose()

# Top 3 ESG
sweden = updated[[14214, 14219, 14223, 14237, 14251]].reset_index(drop=True)
sweden = sweden.rename({14214: 'CO2 emissions (metric tons per capita)', 14219: 'Electricity production from coal sources (% of total)', 14223: 'Fertilit

finland = updated[[7045, 7050, 7054, 7068, 7082]].reset_index(drop=True)
finland = finland.rename({7045: 'CO2 emissions (metric tons per capita)', 7050: 'Electricity production from coal sources (% of total)', 7054: 'Fertility

norway = updated[[11735, 11740, 11744, 11758, 11772]].reset_index(drop=True)
norway = norway.rename({11735: 'CO2 emissions (metric tons per capita)', 11740: 'Electricity production from coal sources (% of total)', 11744: 'Fertilit

# Bottom 3 ESG
ssd = updated[[13678, 13683, 13687, 13701, 13715]].reset_index(drop=True)
ssd = ssd.rename({13678: 'CO2 emissions (metric tons per capita)', 13683: 'Electricity production from coal sources (% of total)', 13687: 'Fertility rate

car = updated[[5236, 5241, 5245, 5259, 5273]].reset_index(drop=True)
car = car.rename({5236: 'CO2 emissions (metric tons per capita)', 5241: 'Electricity production from coal sources (% of total)', 5245: 'Fertility rate, t

yemen = updated[[15822, 15827, 15831, 15845, 15859]].reset_index(drop=True)
yemen = yemen.rename({15822: 'CO2 emissions (metric tons per capita)', 15827: 'Electricity production from coal sources (% of total)', 15831: 'Fertility
```

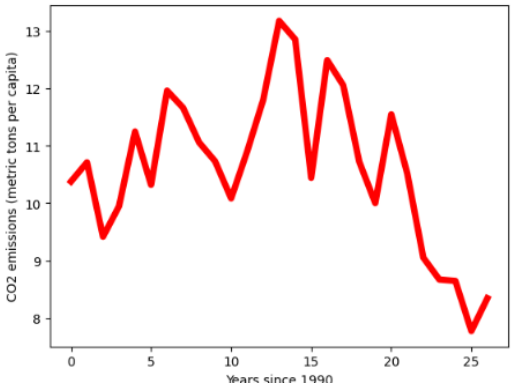
```
# Sweden Graphs
plt.plot(sweden.index, sweden['CO2 emissions (metric tons per capita)'], color = 'gold', linewidth = 5)
plt.ylabel('CO2 emissions (metric tons per capita)')
plt.xlabel('Years since 1990')
plt.show()

plt.plot(sweden.index, sweden['Electricity production from coal sources (% of total)'], color = 'gold', linewidth = 5)
plt.ylabel('Electricity production from coal sources (% of total)')
plt.xlabel('Years since 1990')
plt.show()

plt.plot(sweden.index, sweden['Fertility rate, total (births per woman)'], color = 'red', linewidth = 5)
plt.ylabel('Fertility rate, total (births per woman)')
plt.xlabel('Years since 1990')
plt.show()

plt.plot(sweden.index, sweden['Life expectancy at birth, total (years)'], color = 'green', linewidth = 5)
```



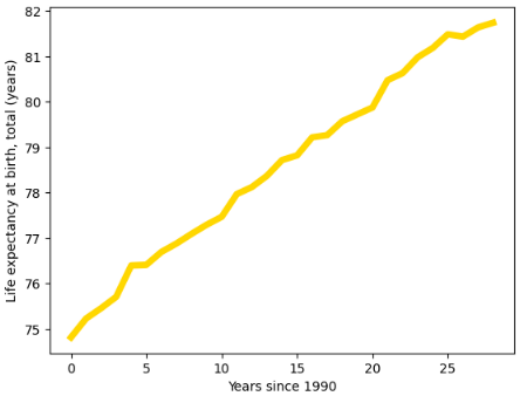


[20]:

```
# Finland Graphs
plt.plot(finland.index, finland['CO2 emissions (metric tons per capita)'], color = 'red', linewidth = 5)
plt.ylabel('CO2 emissions (metric tons per capita)')
plt.xlabel('Years since 1990')
plt.show()

plt.plot(finland.index, finland['Electricity production from coal sources (% of total)'], color = 'blue', linewidth = 5)
plt.ylabel('Electricity production from coal sources (% of total)')
plt.xlabel('Years since 1990')
plt.show()

plt.plot(finland.index, finland['Fertility rate, total (births per woman)'], color = 'blue', linewidth = 5)
plt.ylabel('Fertility rate, total (births per woman)')
plt.xlabel('Years since 1990')
plt.show()

plt.plot(finland.index, finland['Life expectancy at birth, total (years)'], color = 'gold', linewidth = 5)
plt.ylabel('Life expectancy at birth, total (years)')
plt.xlabel('Years since 1990')
plt.show()

plt.plot(finland.index, finland['Population ages 65 and above (% of total population)'], color = 'green', linewidth = 5)
plt.ylabel('Population ages 65 and above (% of total population)')
plt.xlabel('Years since 1990')
plt.show()
```




Fertility rate, total (births per woman)
1.8
1.7
1.6
1.5
1.4
0
5
10
15
20
25
Years since 1990


Life expectancy at birth, total (years)
82
81
80
79
78
77
76
75
0
5
10
15
20
25
Years since 1990

[20]:
```
# Finland Graphs
plt.plot(finland.index, finland['CO2 emissions (metric tons per capita)'], color = 'red', linewidth = 5)
plt.ylabel('CO2 emissions (metric tons per capita)')
plt.xlabel('Years since 1990')
plt.show()

plt.plot(finland.index, finland['Electricity production from coal sources (% of total)'], color = 'blue', linewidth = 5)
plt.ylabel('Electricity production from coal sources (% of total)')
plt.xlabel('Years since 1990')
plt.show()

plt.plot(finland.index, finland['Fertility rate, total (births per woman)'], color = 'blue', linewidth = 5)
plt.ylabel('Fertility rate, total (births per woman)')
plt.xlabel('Years since 1990')
plt.show()

plt.plot(finland.index, finland['Life expectancy at birth, total (years)'], color = 'gold', linewidth = 5)
plt.ylabel('Life expectancy at birth, total (years)')
plt.xlabel('Years since 1990')
plt.show()

plt.plot(finland.index, finland['Population ages 65 and above (% of total population)'], color = 'green', linewidth = 5)
plt.ylabel('Population ages 65 and above (% of total population)')
plt.xlabel('Years since 1990')
plt.show()
```