
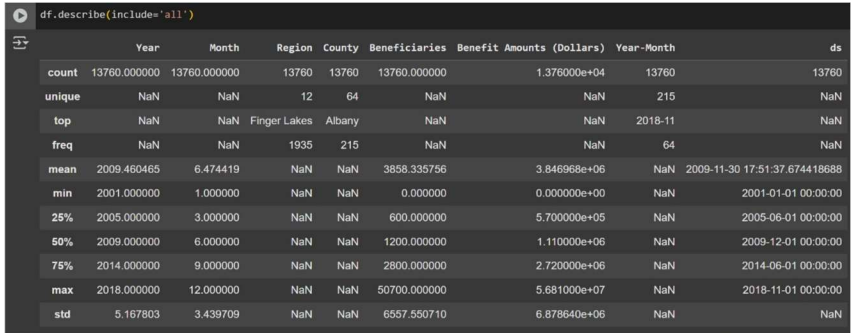


Data Collection and Preprocessing Phase

Date	15 June 2025
Team ID	SWTID1749896042
Project Title	Unemployed insurance beneficiary forecasting
Maximum Marks	6 Marks


Data Exploration and Preprocessing Report

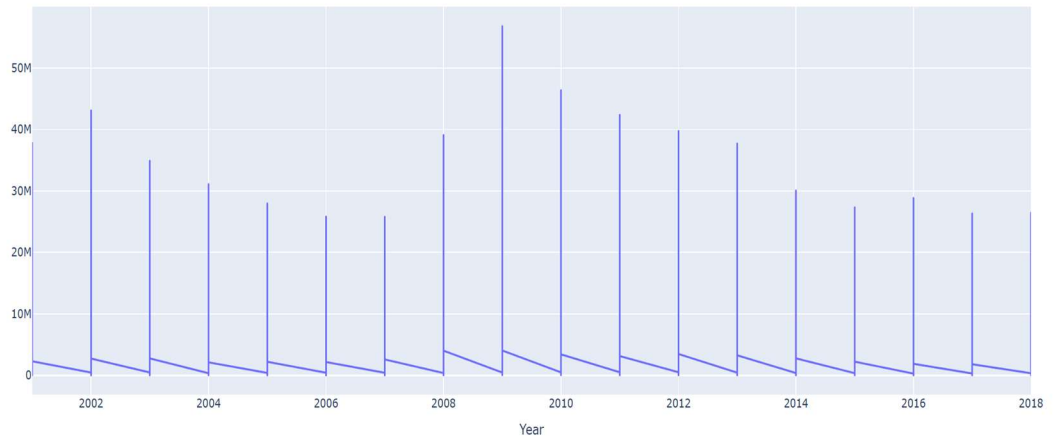
In this project, I explored and prepared data to understand how unemployment beneficiaries and benefit amounts change over time. I used Python to check for missing values and duplicates to keep the data clean. I also combined the Year and Month columns into a single date column and grouped the data by month to make it easier to analyse. These steps helped me create a clear and organised dataset that can be used for accurate time series forecasting and further analysis.

Section	Description
Data Overview	<p><u>Dimension:</u></p>  <p><u>Descriptive statistics:</u></p> 


Univariate Analysis

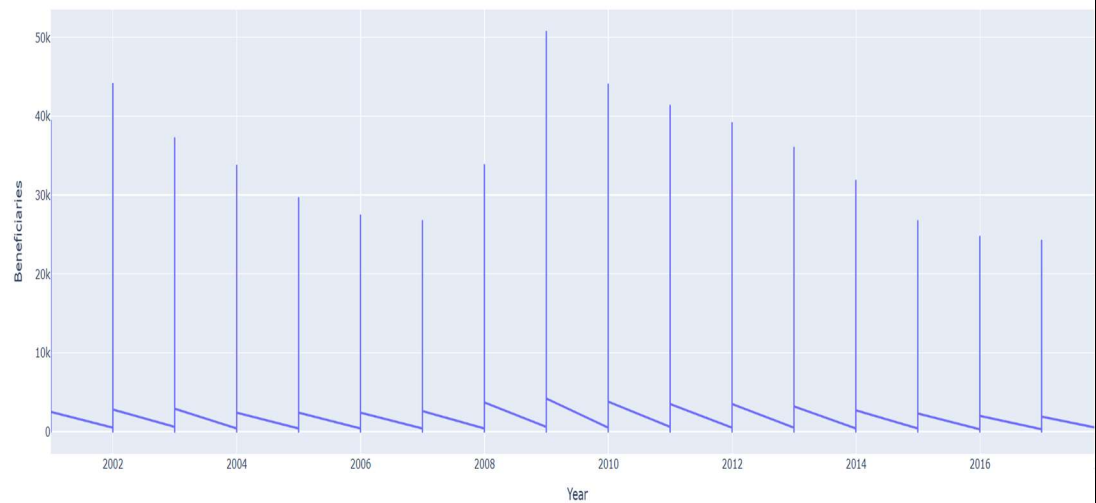
```

3s  import plotly.express as px
fig=px.line(df,x='Year',y='Benefit Amounts (Dollars)')
fig.show()
  
```



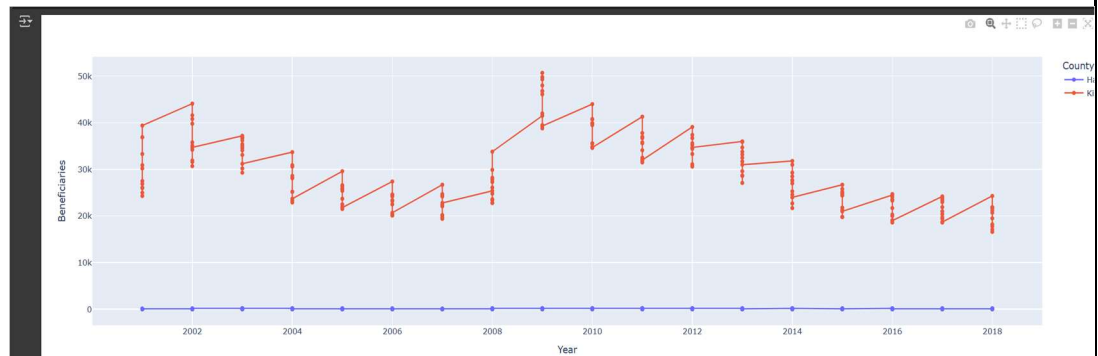
```

0s  fig=px.line(df,x='Year',y='Beneficiaries')
fig.show()
  
```

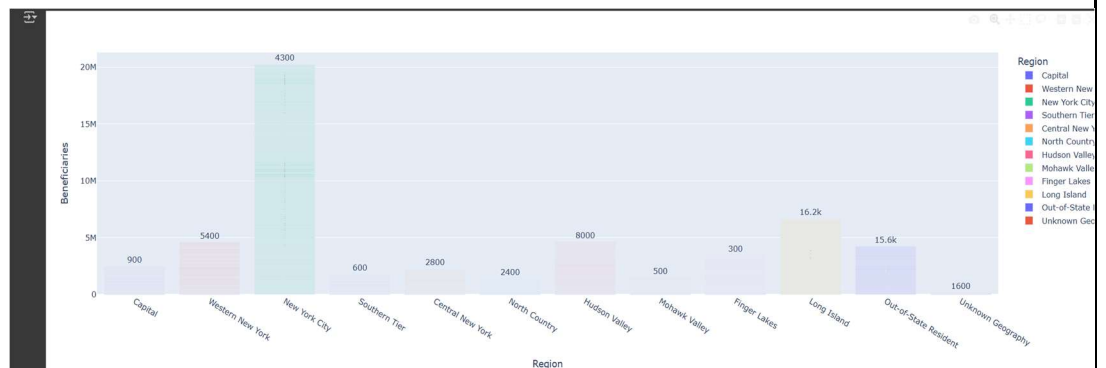


Bivariate Analysis

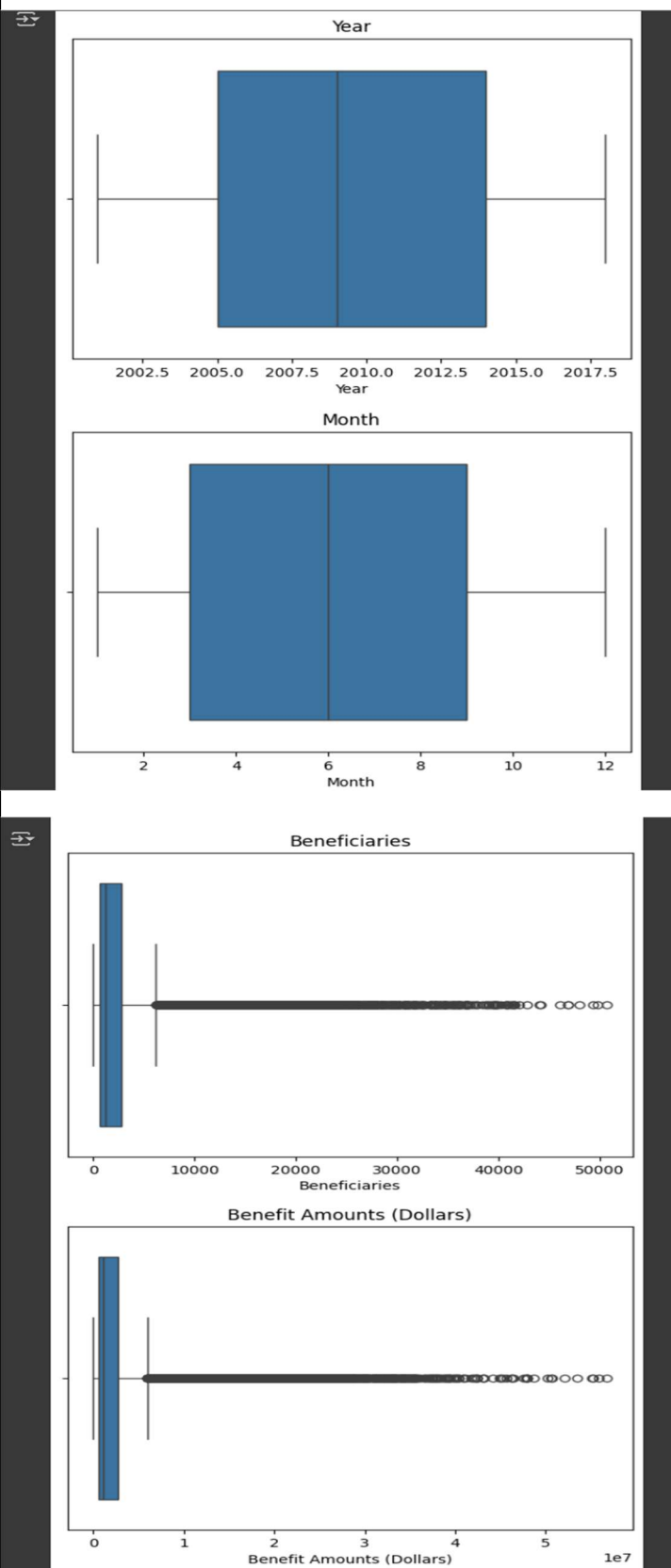
```
[19] df1=df.query("County in ['Hamilton','Kings']")
fig=px.line(df1,x='Year',y='Beneficiaries', color='County', markers=True)
fig.update_traces(textposition="bottom right")
fig.show()
```



```
[20] fig=px.bar(df,x='Region',y='Beneficiaries',color='Region',text_auto=True)
fig.show()
```



```
[22] for i in df.columns:
      if (df[i].dtype)=='int64':
          boxplot=sns.boxplot(x=df[i])
          plt.title(i)
          plt.show()
```



Multivariate
Analysis

Descriptive Analysis

```
df.describe(include='all')
```

	Year	Month	Region	County	Beneficiaries	Benefit Amounts (Dollars)	Year-Month	
count	13760.000000	13760.000000	13760	13760	13760.000000	1.376000e+04	13760	13760
unique	NaN	NaN	12	64	NaN	NaN	215	NaN
top	NaN	NaN	Finger Lakes	Albany	NaN	NaN	2018-11	NaN
freq	NaN	NaN	1935	215	NaN	NaN	64	NaN
mean	2009.460465	6.474419	NaN	NaN	3858.335756	3.846968e+06	NaN	2009-11-30 17:51:37.674418
min	2001.000000	1.000000	NaN	NaN	0.000000	0.000000e+00	NaN	2001-01-01 00:00
25%	2005.000000	3.000000	NaN	NaN	600.000000	5.700000e+05	NaN	2005-06-01 00:00
50%	2009.000000	6.000000	NaN	NaN	1200.000000	1.110000e+06	NaN	2009-12-01 00:00
75%	2014.000000	9.000000	NaN	NaN	2800.000000	2.720000e+06	NaN	2014-06-01 00:00
max	2018.000000	12.000000	NaN	NaN	50700.000000	5.681000e+07	NaN	2018-11-01 00:00
std	5.167803	3.439709	NaN	NaN	6557.550710	6.878640e+06	NaN	NaN

Data Preprocessing Code Screenshots

Loading Data

```
df=pd.read_csv(r"InsuranceUnemployedData.csv")
df.head()
```

	Year	Month	Region	County	Beneficiaries	Benefit Amounts (Dollars)	Year-Month
0	2018	11	Capital	Albany	1600	1570000	2018-11
1	2018	11	Western New York	Allegany	400	300000	2018-11
2	2018	11	New York City	Bronx	11600	11530000	2018-11
3	2018	11	Southern Tier	Broome	1400	1150000	2018-11
4	2018	11	Western New York	Cattaraugus	900	710000	2018-11

Checking for Missing Data

```
df.isna().sum()
```

	0
Year	0
Month	0
Region	0
County	0
Beneficiaries	0
Benefit Amounts (Dollars)	0
Year-Month	0

dtype: int64

Data Transformation

```
df['ds']=pd.to_datetime(df['Year'].astype(str)+'-'+df['Month'].astype(str).zfill(2))
df_monthly=df.groupby('ds').agg({'Beneficiaries':'sum','Benefit Amounts (Dollars)':'sum'}).reset_index()
df_monthly.describe()
```

	ds	Beneficiaries	Benefit Amounts (Dollars)
count	215	215.000000	2.150000e+02
mean	2009-11-30 17:51:37.674418688	246933.488372	2.462060e+08
min	2001-01-01 00:00:00	124900.000000	1.340700e+08
25%	2005-06-16 00:00:00	194350.000000	1.921550e+08
50%	2009-12-01 00:00:00	236300.000000	2.309200e+08
75%	2014-05-16 12:00:00	288050.000000	2.865300e+08
max	2018-11-01 00:00:00	456700.000000	5.354200e+08
std	NaN	67465.360405	7.187324e+07

Feature Engineering

```
df['ds']=pd.to_datetime(df['Year'].astype(str)+'-'+df['Month'].astype(str).zfill(2))

df_monthly=df.groupby('ds').agg({'Beneficiaries':'sum','Benefit Amounts (Dollars)':'sum'}).reset_index()
df_monthly.describe()
```

Feature Engineering

- Created a new datetime feature 'ds' by combining Year and Month.
- Aggregated Beneficiaries and Benefit Amounts monthly to prepare structured time series features for model training.