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

from google.colab import drive
drive.mount('/content/drive')
```

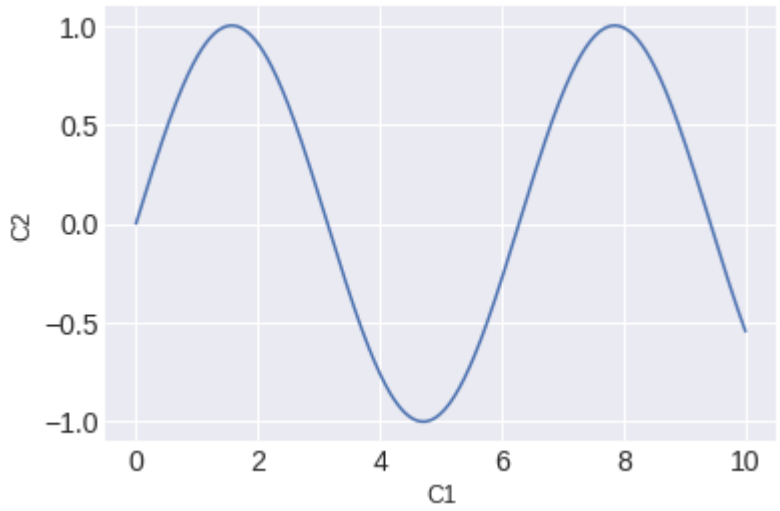
```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remou
```

Line Charts

```
col1 = np.linspace(0, 10, 1000)
col2 = np.sin(col1)
df = pd.DataFrame({"C1" : col1 , "C2" :col2})
df.head(10)
```

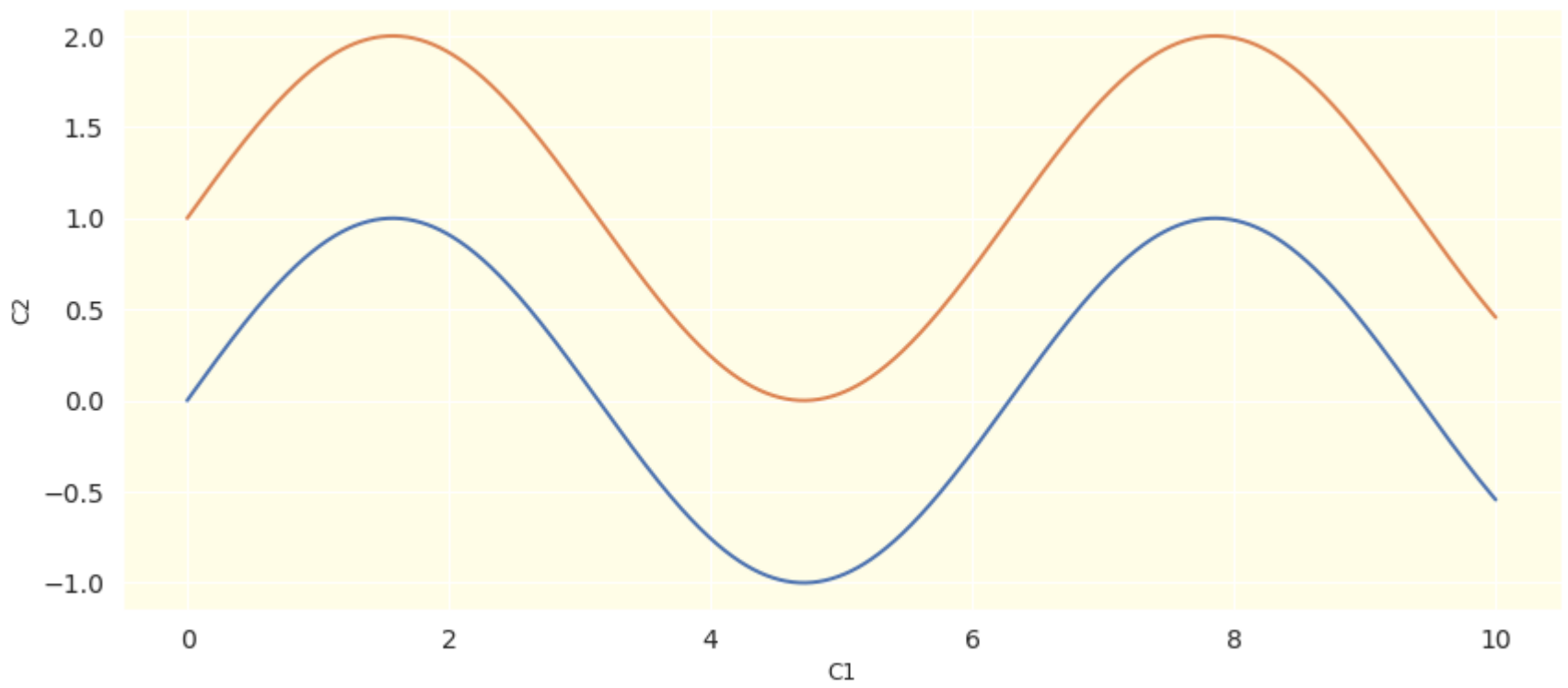
	C1	C2
0	0.00000	0.000000
1	0.01001	0.010010
2	0.02002	0.020019

```
# Plotting lineplot using sns.lineplot()
plt.style.use('seaborn-darkgrid')
%matplotlib inline
sns.lineplot(x=df.C1,y=df.C2,data=df)
plt.show()
```

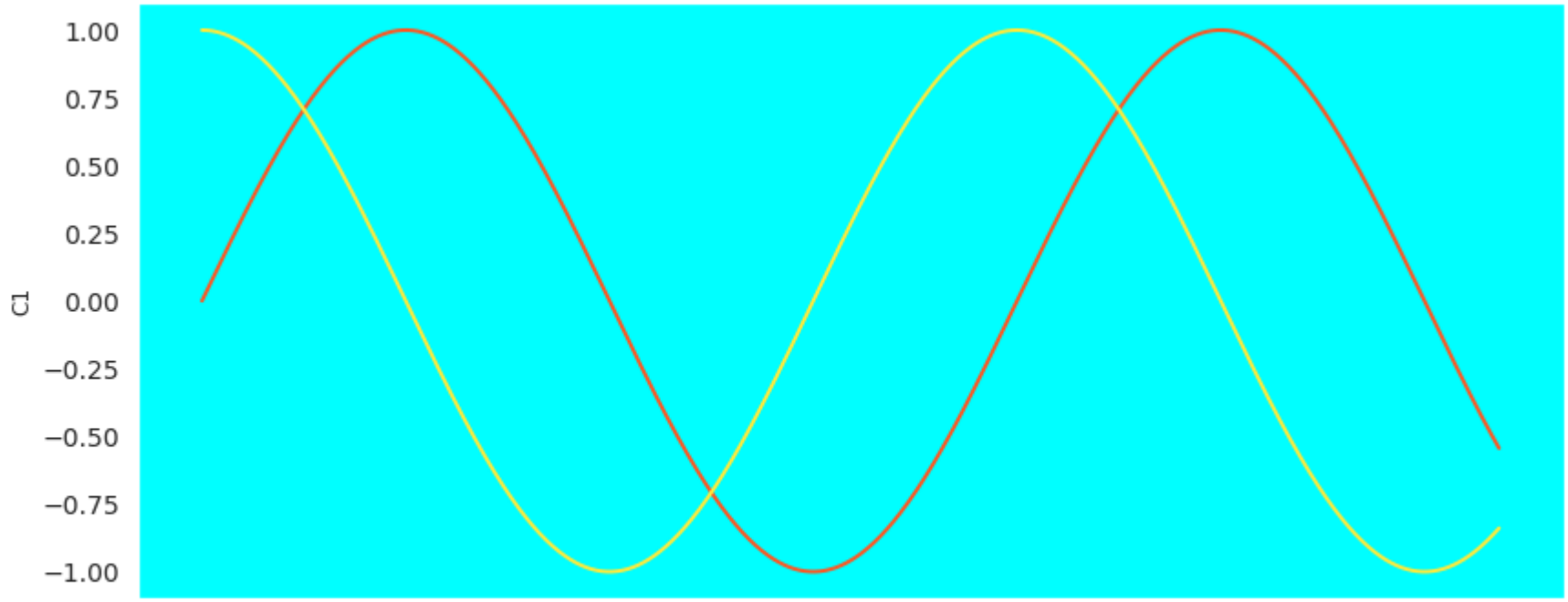



```
""" - Adjusting background color using axes.facecolor
    - Changing label size using xtick.labelsize , ytick.labelsize """

plt.figure(figsize=(14,6))
sns.set(rc={"axes.facecolor":"#FFFDE7", "axes.grid":True,'xtick.labelsize':14,'ytick.la
sns.lineplot(x=df.C1,y=df.C2,data=df , linewidth = 2)
sns.lineplot(x=df.C1,y=df.C2+1,data=df , linewidth = 2)
```

```
sns.lineplot(x=df.C1,y=df.C2+1,data=df , linewidth = 2)
plt.show()
```



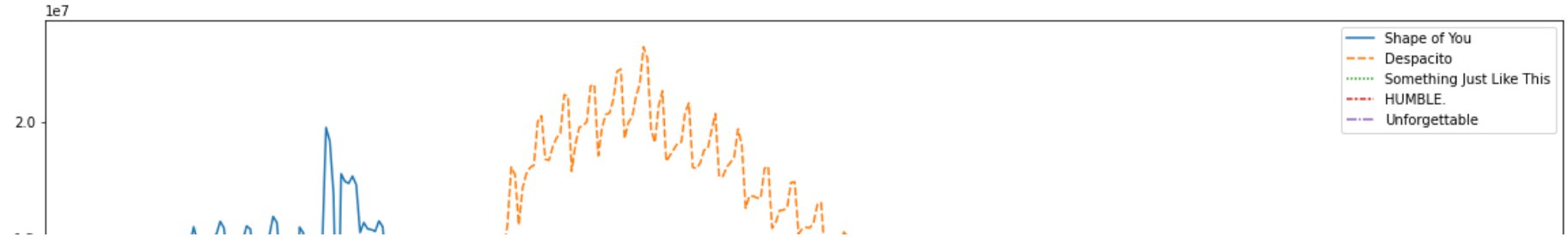

```
""" - Adjusting background color using axes.facecolor
    - Changing label size using xtick.labelsize , ytick.labelsize """

plt.figure(figsize=(14,6))
sns.set(rc={"axes.facecolor":"yellow", "axes.grid":False,'xtick.labelsize':14,'ytick.la
sns.lineplot(x=df.C1,y=df.C2,data=df , color = "#FF5722" , linewidth = 2 )
sns.lineplot(x=df.C1,y=np.cos(df.C1),data=df , color = "#FFEB3B" , linewidth = 2)
plt.show()
```




```
# Recover default matplotlib settings
mpl.rcParams.update(mpl.rcParamsDefault)
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")

spotify = pd.read_csv("/content/drive/My Drive/Python DataScience/Visualization/Seaborn
spotify.head(10)
```

Date	Shape of You	Despacito	Something Just Like This	HUMBLE.	Unforgettable
2017-01-06	12287078	NaN	NaN	NaN	NaN
2017-01-07	13190270	NaN	NaN	NaN	NaN
2017-01-08	13099919	NaN	NaN	NaN	NaN
2017-01-09	14506351	NaN	NaN	NaN	NaN
2017-01-10	14275628	NaN	NaN	NaN	NaN
2017-01-11	14372699	NaN	NaN	NaN	NaN
2017-01-12	14148108	NaN	NaN	NaN	NaN

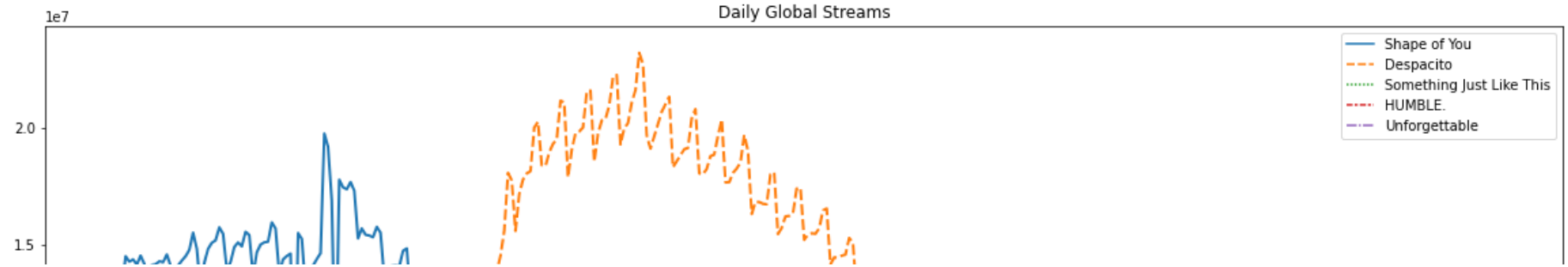
```
plt.figure(figsize=(20,8))
sns.lineplot(data=spotify)
plt.show()
```



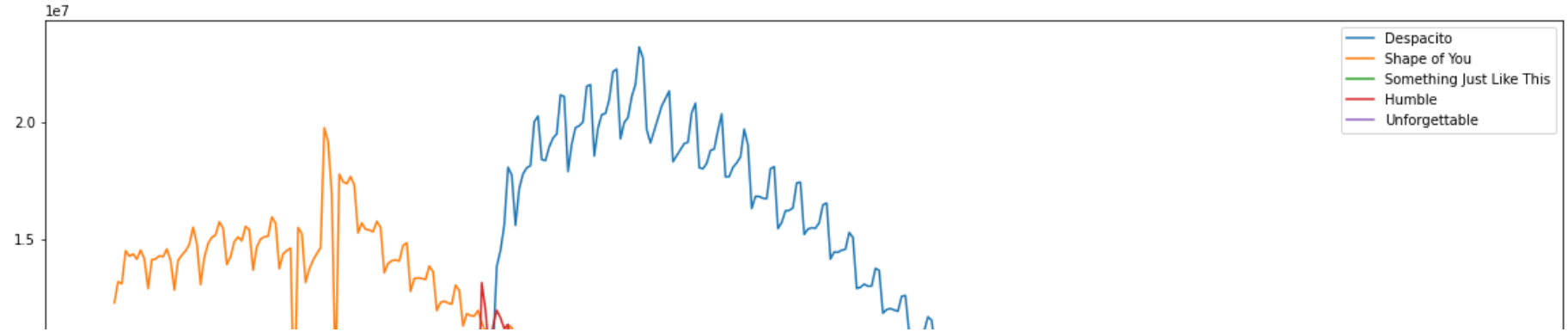
Look at the X-Asis. It is clealy not able to interpret the Index as Date. So the next step will be to convert the Index to Datetime.

```
spotify.index = pd.to_datetime(spotify.index) # Converting datatype of index to Datetim
```

```
plt.figure(figsize=(20,8))
sns.lineplot(data=spotify,linewidth = 2)
plt.title("Daily Global Streams")
plt.show()
```

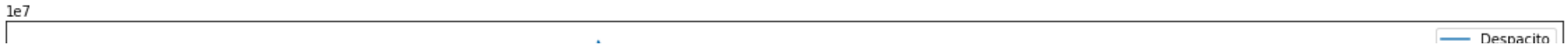



```
# Using Matplotlib for same visualization
import matplotlib as mpl
plt.figure(figsize=(20,8))
plt.plot(spotify['Despacito'] , label="Despacito")
plt.plot(spotify['Shape of You'] , label="Shape of You")
plt.plot(spotify['Something Just Like This'] , label="Something Just Like This")
plt.plot(spotify['HUMBLE.'] , label="Humble")
plt.plot(spotify['Unforgettable'] , label="Unforgettable")
plt.legend()
plt.show()
```



So we can see using Seabon we can save many lines of code.

```
plt.figure(figsize=(20,6))
sns.lineplot(data=spotify['Despacito'],linewidth = 1.5 , label = 'Despacito')
sns.lineplot(data=spotify['HUMBLE.'],linewidth = 1.5 , label = 'Humble')
plt.show()
```




```
canada = pd.read_csv("/content/drive/My Drive/Python DataScience/Visualization/Seaborn/
canada.head()
```

	Type	Coverage	OdName	AREA	AreaName	REG	RegName	DEV	DevName	1980	1981	1982	1983	1984	1985
0	Immigrants	Foreigners	Afghanistan	935	Asia	5501	Southern Asia	902	Developing regions	16	39	39	47	71	340
1	Immigrants	Foreigners	Albania	908	Europe	925	Southern Europe	901	Developed regions	1	0	0	0	0	0
2	Immigrants	Foreigners	Algeria	903	Africa	912	Northern Africa	902	Developing regions	80	67	71	69	63	44
3	Immigrants	Foreigners	American Samoa	909	Oceania	957	Polynesia	902	Developing regions	0	1	0	0	0	0
4	Immigrants	Foreigners	Andorra	908	Europe	925	Southern Europe	901	Developed regions	0	0	0	0	0	0

```
canada.columns
```

```
Index(['Type', 'Coverage', 'OdName', 'AREA', 'AreaName', 'REG', 'RegName',
       'DEV', 'DevName', '1980', '1981', '1982', '1983', '1984', '1985',
       '1986', '1987', '1988', '1989', '1990', '1991', '1992', '1993', '1994',
       '1995', '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003',
       '2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012',
       '2013'],
      dtype='object')
```

```
canada.drop(columns=['AREA' , 'DEV', 'DevName' , 'REG', 'Type', 'Coverage' , 'AreaName'
canada.head()
```

	OdName	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	199
0	Afghanistan	16	39	39	47	71	340	496	741	828	1076	1028	1378	1170	713	858	1537	2212	255
1	Albania	1	0	0	0	0	0	1	2	2	3	3	21	56	96	71	63	113	30
2	Algeria	80	67	71	69	63	44	69	132	242	434	491	872	795	717	595	1106	2054	184
3	American Samoa	0	1	0	0	0	0	0	1	0	1	2	0	0	0	0	0	0	
4	Andorra	0	0	0	0	0	0	2	0	0	0	3	0	1	0	0	0	0	

```
canada.rename(columns={'OdName':'Country'} , inplace=True)
canada.set_index(canada.Country,inplace=True)
canada.head()
```

	Country	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995
Country																	
Afghanistan	Afghanistan	16	39	39	47	71	340	496	741	828	1076	1028	1378	1170	713	858	1537
Albania	Albania	1	0	0	0	0	0	1	2	2	3	3	21	56	96	71	63
Algeria	Algeria	80	67	71	69	63	44	69	132	242	434	491	872	795	717	595	1106
American Samoa	American Samoa	0	1	0	0	0	0	0	1	0	1	2	0	0	0	0	0
Andorra	Andorra	0	0	0	0	0	0	2	0	0	0	3	0	1	0	0	0

```
canada.index.name=None
canada.head()
```

	Country	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995
Afghanistan	Afghanistan	16	39	39	47	71	340	496	741	828	1076	1028	1378	1170	713	858	1537
Albania	Albania	1	0	0	0	0	0	1	2	2	3	3	21	56	96	71	63
Algeria	Algeria	80	67	71	69	63	44	69	132	242	434	491	872	795	717	595	1106
American Samoa	American Samoa	0	1	0	0	0	0	0	1	0	1	2	0	0	0	0	0

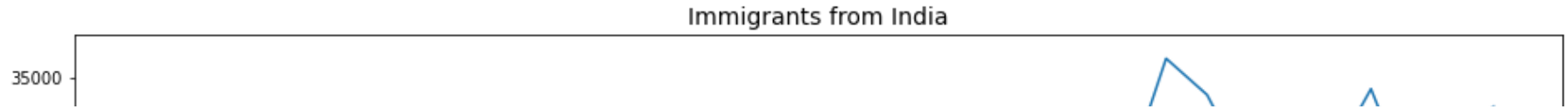
```
del canada['Country']
canada.head()
```

	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997
Afghanistan	16	39	39	47	71	340	496	741	828	1076	1028	1378	1170	713	858	1537	2212	2555
Albania	1	0	0	0	0	0	1	2	2	3	3	21	56	96	71	63	113	307
Algeria	80	67	71	69	63	44	69	132	242	434	491	872	795	717	595	1106	2054	1842
American Samoa	0	1	0	0	0	0	0	1	0	1	2	0	0	0	0	0	0	0
Andorra	0	0	0	0	0	0	2	0	0	0	3	0	1	0	0	0	0	0

```
canada = canada.transpose()

canada.head()
```

	Afghanistan	Albania	Algeria	American Samoa	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	Azerba
1980	16	1	80	0	0	1	0	368	0	702	234	
1981	39	0	67	1	0	3	0	426	0	639	238	
1982	39	0	71	0	0	6	0	626	0	484	201	

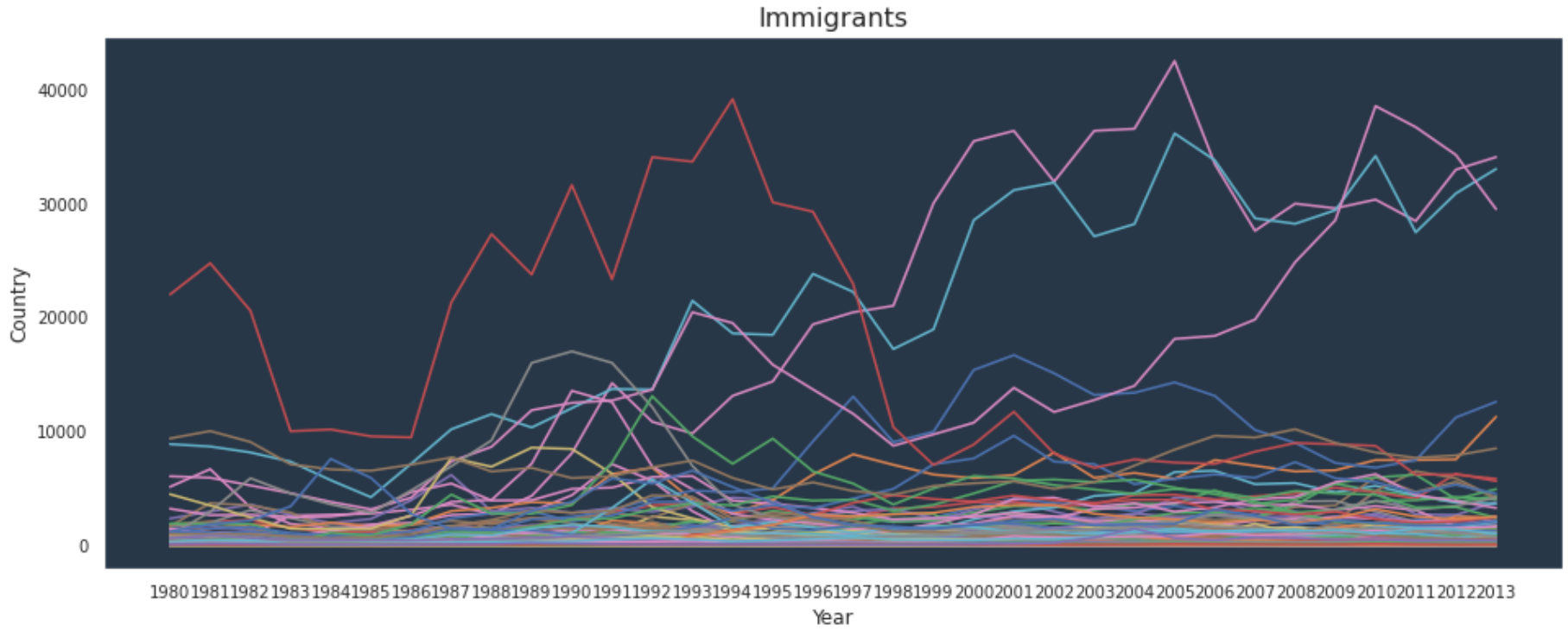
```
plt.figure(figsize=(16,6))
plt.title("Immigrants from India",fontsize = 14)
sns.set(rc={"axes.facecolor":"#283747", "axes.grid":False,'xtick.labelsize':10,'ytick.l
plt.xticks(rotation=45) # Rotating X tickts by 45 degrees
sns.lineplot(x = canada.index.values, y = canada['India'])
plt.show()
```




```
# Plotting multiple sets of data (E.g Immigration data of multiple countries in one plo
plt.figure(figsize=(16,6))
sns.set(rc={"axes.facecolor":"#283747", "axes.grid":False,'xtick.labelsize':10,'ytick.l
plt.title("Immigrants from India , Pakistan & Bangladesh",fontsize = 14)
plt.xticks(rotation=45) # Rotating X tickts by 45 degrees
sns.lineplot(x = canada.index.values, y = canada['India'] , color = '#ff9900' , label=
sns.lineplot(x = canada.index.values, y = canada['Pakistan'] , color = '#4586ff' , labe
sns.lineplot(x = canada.index.values, y = canada['Bangladesh'] , color = '#a2ef44' , la
plt.legend(facecolor= 'grey' , fontsize='large' , edgecolor = 'black' ,shadow=True) # L
plt.show()
```

Immigrants from India , Pakistan & Bangladesh

```
# Plotting multiple sets of data using for loop (E.g Immigration data of multiple count
plt.figure(figsize=(16,6))
plt.title("Immigrants",fontsize = 16)
for i in canada.columns:
    if canada[i].name != 'Total' and canada[i].name != 'Unknown':
        x=canada.index.values
        y=canada[i]
        sns.lineplot(x,y)
plt.xlabel ('Year')
plt.ylabel ('Country')
plt.show()
```



```
# Recover default matplotlib settings
mpl.rcParams.update(mpl.rcParamsDefault)
%matplotlib inline
```

```
employment = pd.read_excel("/content/drive/My Drive/Python DataScience/Visualization/Se
employment.head(10)
```

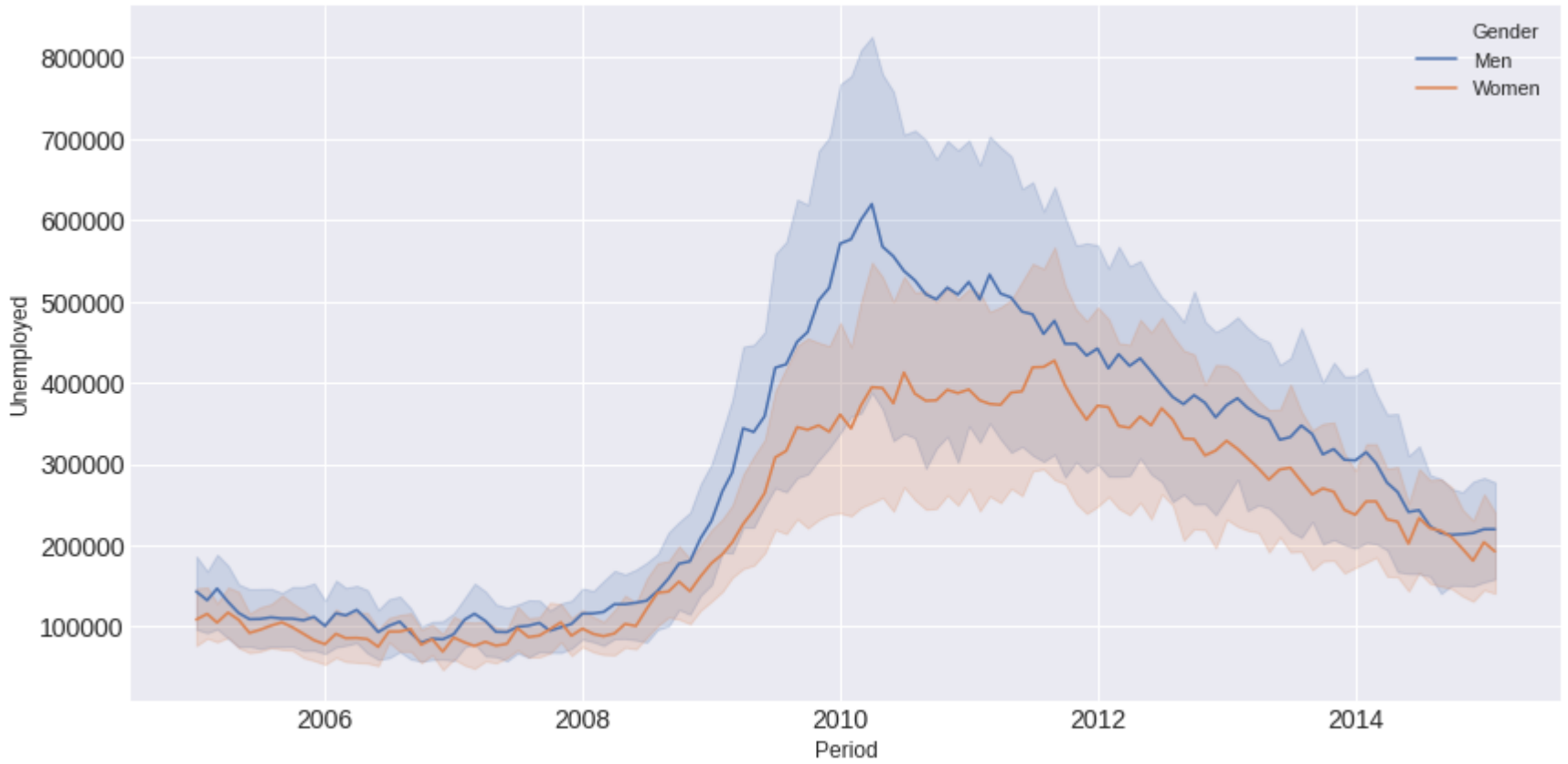
	Age	Gender	Period	Unemployed
0	16 to 19 years	Men	2005-01-01	91000
1	20 to 24 years	Men	2005-01-01	175000
2	25 to 34 years	Men	2005-01-01	194000
3	35 to 44 years	Men	2005-01-01	201000
4	45 to 54 years	Men	2005-01-01	207000
5	55 to 64 years	Men	2005-01-01	101000
6	65 years and over	Men	2005-01-01	33000
7	16 to 19 years	Women	2005-01-01	38000
8	20 to 24 years	Women	2005-01-01	90000
9	25 to 34 years	Women	2005-01-01	142000

```
employment.dtypes
```

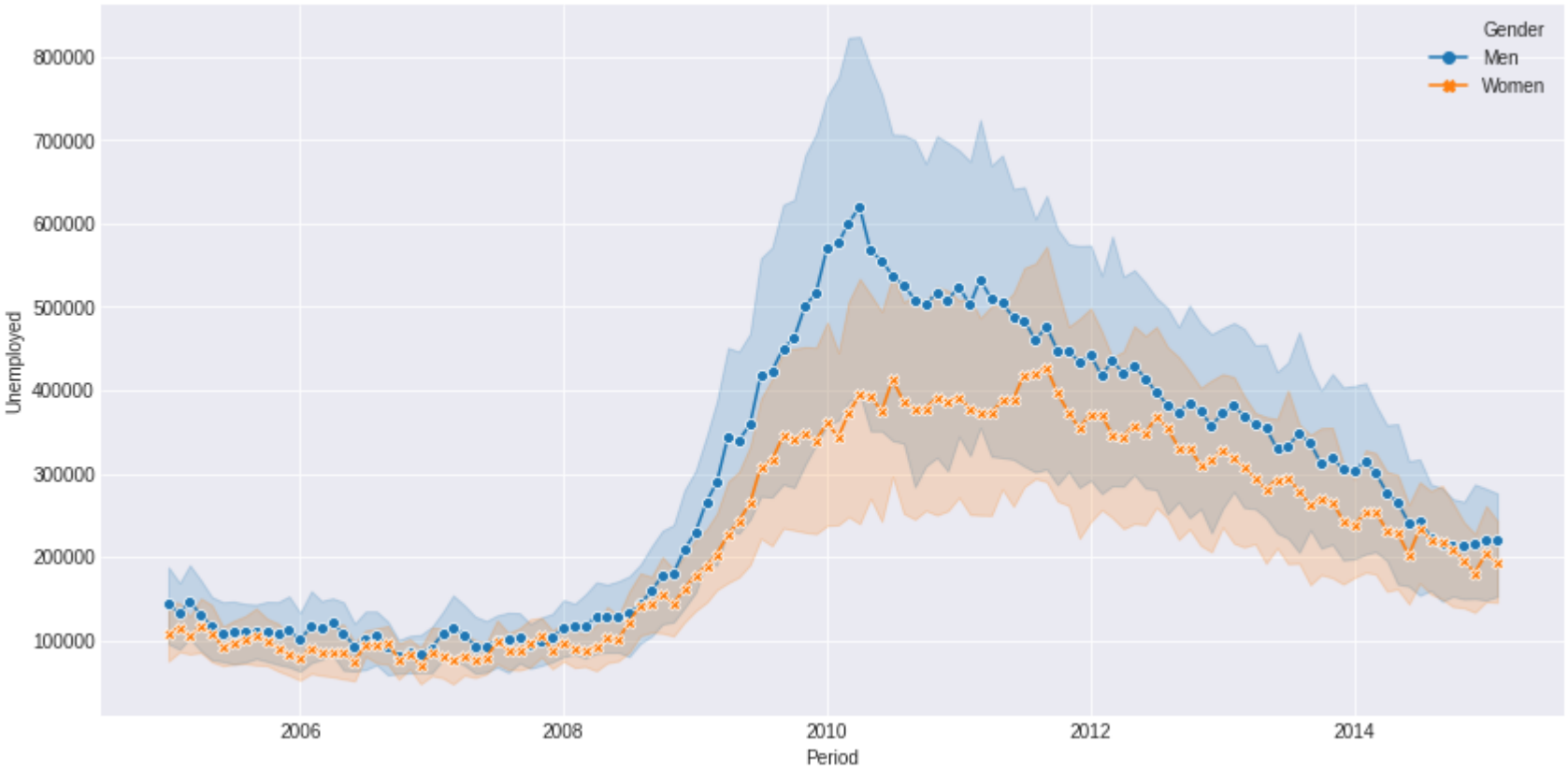
```
Age                   object
Gender                object
Period        datetime64[ns]
Unemployed             int64
dtype: object
```

```
# Show groups with different colors using "hue"
```

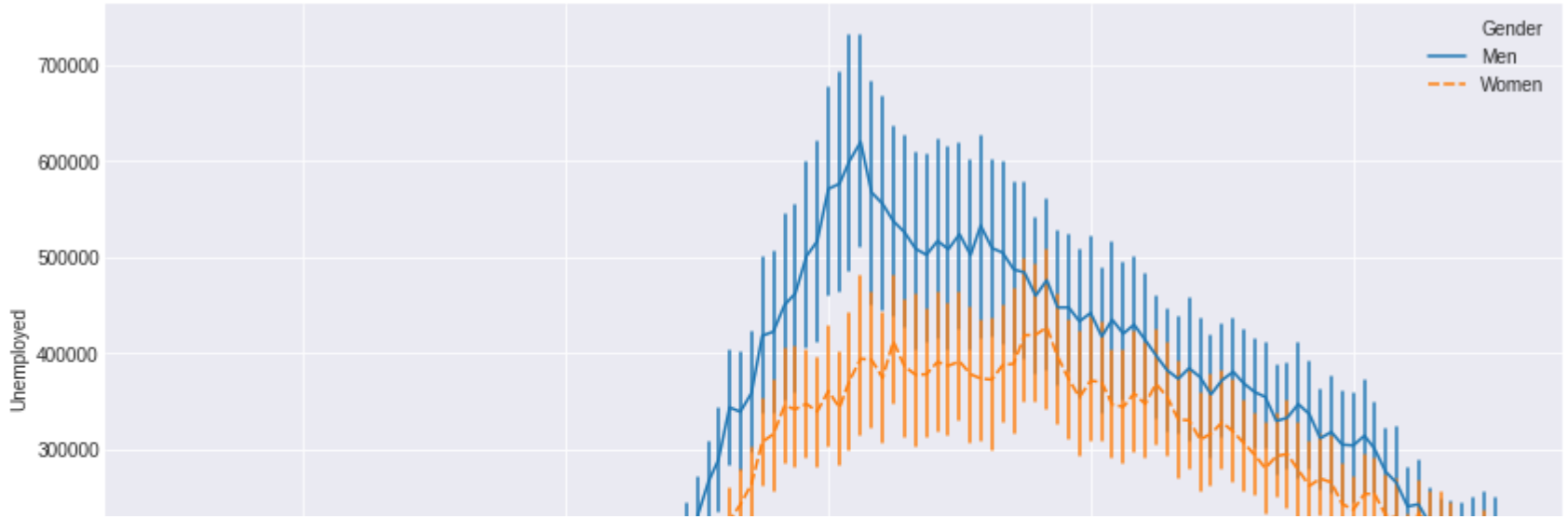
```
plt.figure(figsize=(14,7))
plt.style.use('seaborn-darkgrid')
# Group variable using "hue" that will produce lines with different colors
sns.lineplot(x="Period" , y="Unemployed" , hue="Gender" , data=employment)   # ,ci=None
plt.show()
```



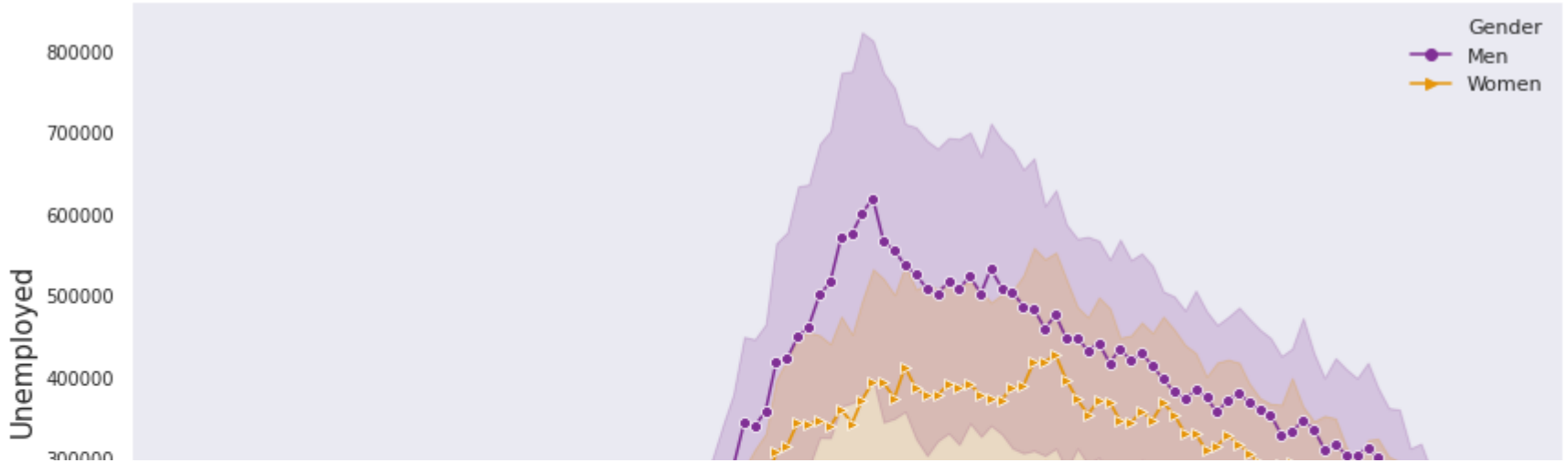

```
# Using markers to identify groups
plt.figure(figsize=(14,7))
plt.style.use('seaborn-darkgrid')
sns.lineplot(x="Period" , y="Unemployed" , hue = "Gender" ,style="Gender" , markers=Tru
plt.show()
```



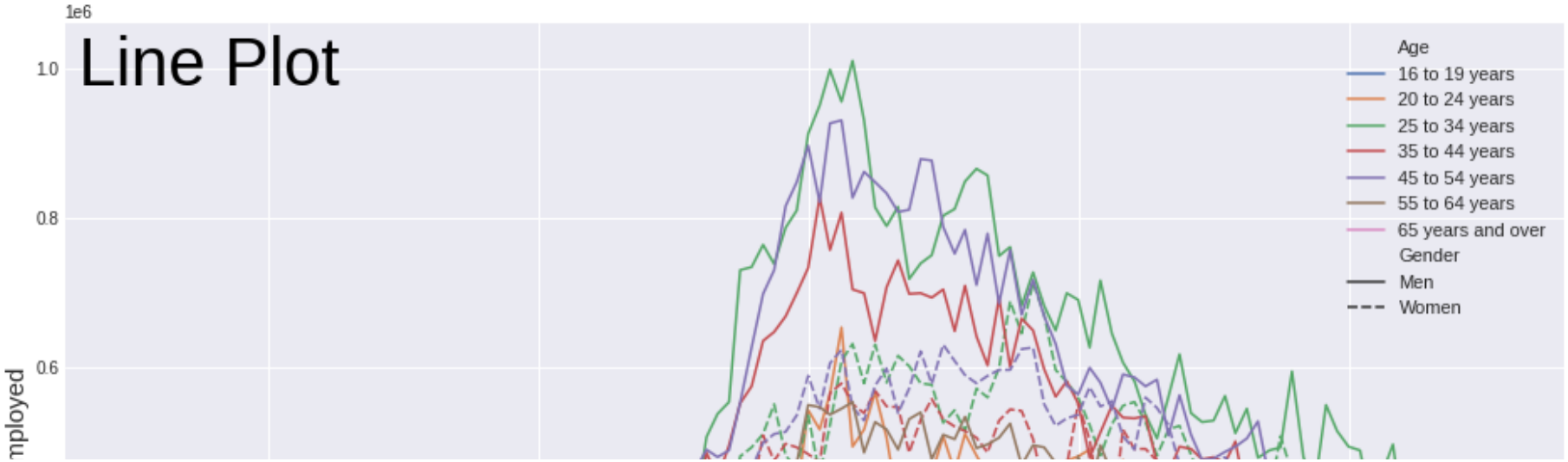

```
plt.figure(figsize=(14,7))
plt.style.use('seaborn-darkgrid')
sns.lineplot(x="Period" , y="Unemployed" , hue = "Gender" ,style="Gender" ,  err_style=
plt.show()
```




```
plt.figure(figsize=(14,7))
sns.set(rc={'xtick.labelsize':17,'ytick.labelsize':10,'axes.labelsize':15 , "axes.grid"

# Use "Pallete" to specify the colors to be used for different levels of the hue
sns.lineplot(x="Period" , y="Unemployed", data = employment, hue = "Gender",
             style = "Gender", dashes = False, palette = 'CMRmap' , markers = ["o", ">"]
#https://matplotlib.org/3.1.1/gallery/color/colormap_reference.html
#https://matplotlib.org/3.1.1/tutorials/colors/colormaps.html
plt.show()
```



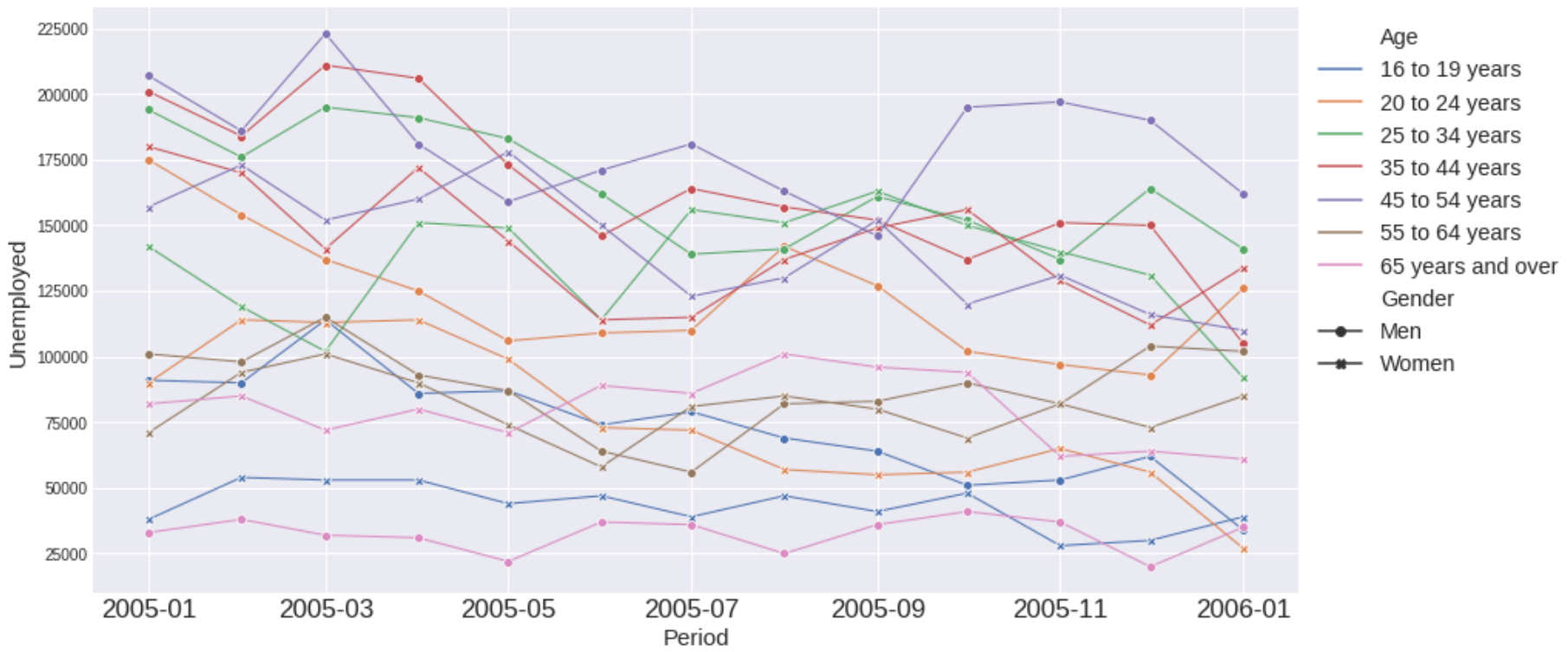

```
# Color and line dashing to represent 2 different grouping variables using "hue" & "sty
plt.figure(figsize=(16,9))
plt.style.use('seaborn-darkgrid')
plt.gcf().text(.2, .84, "Line Plot", fontsize = 40, color='Black' ,ha='center', va='cen
sns.lineplot(x="Period" , y="Unemployed" , hue="Age" , style="Gender",data=employment)
plt.show()
```




```
emp = employment[employment.Period.between('2005-01-01', '2006-01-01' , inclusive = Tru
emp.tail()
```

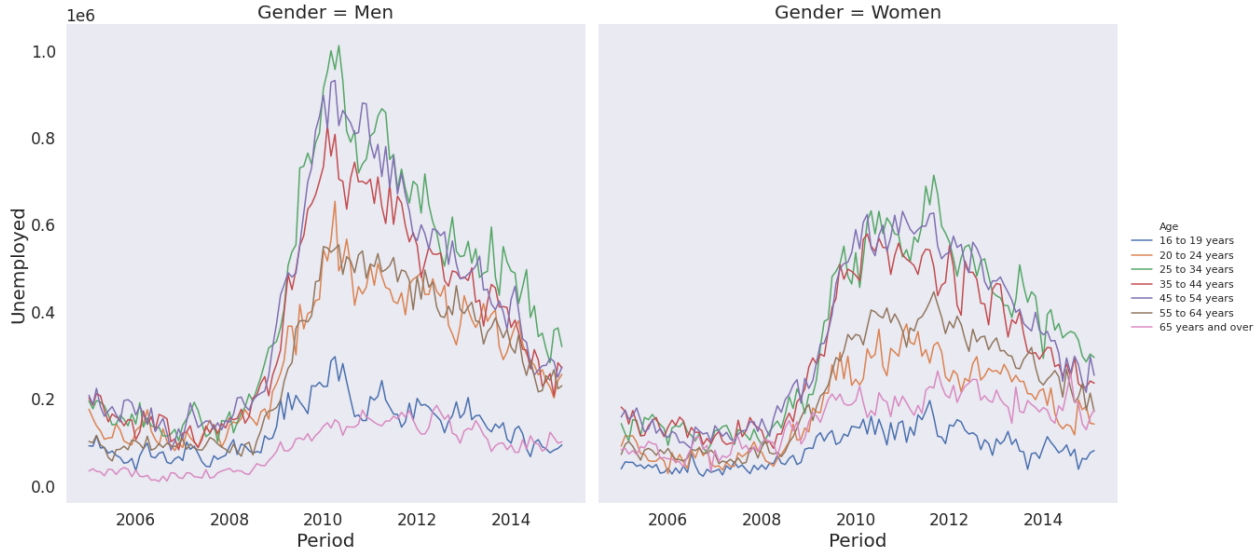
	Age	Gender	Period	Unemployed
177	25 to 34 years	Women	2006-01-01	92000
178	35 to 44 years	Women	2006-01-01	134000
179	45 to 54 years	Women	2006-01-01	110000
180	55 to 64 years	Women	2006-01-01	85000
181	65 years and over	Women	2006-01-01	61000

```
#Showing all experiments instead of Aggregate using "units" and "estimator"
plt.figure(figsize=(14,7))
plt.style.use('seaborn-darkgrid')
sns.lineplot(x="Period" , y="Unemployed" , hue = "Age" ,style="Gender" ,
 units="Age" ,markers=True , dashes=False ,estimator=None, lw=1,data=emp)
plt.legend(bbox_to_anchor=(1.0, 1.0) , shadow=True, fontsize='large')
plt.show()
```




```
# Combining lineplots using relplot
plt.figure(figsize=(10,10))
sns.set(rc={'xtick.labelsize':17,'ytick.labelsize':17,'axes.labelsize':20  , "axes.grid
sns.relplot(x="Period" , y="Unemployed" , hue="Age" , col="Gender",kind='line', height=
plt.show()
```

```
<Figure size 720x720 with 0 Axes>
```



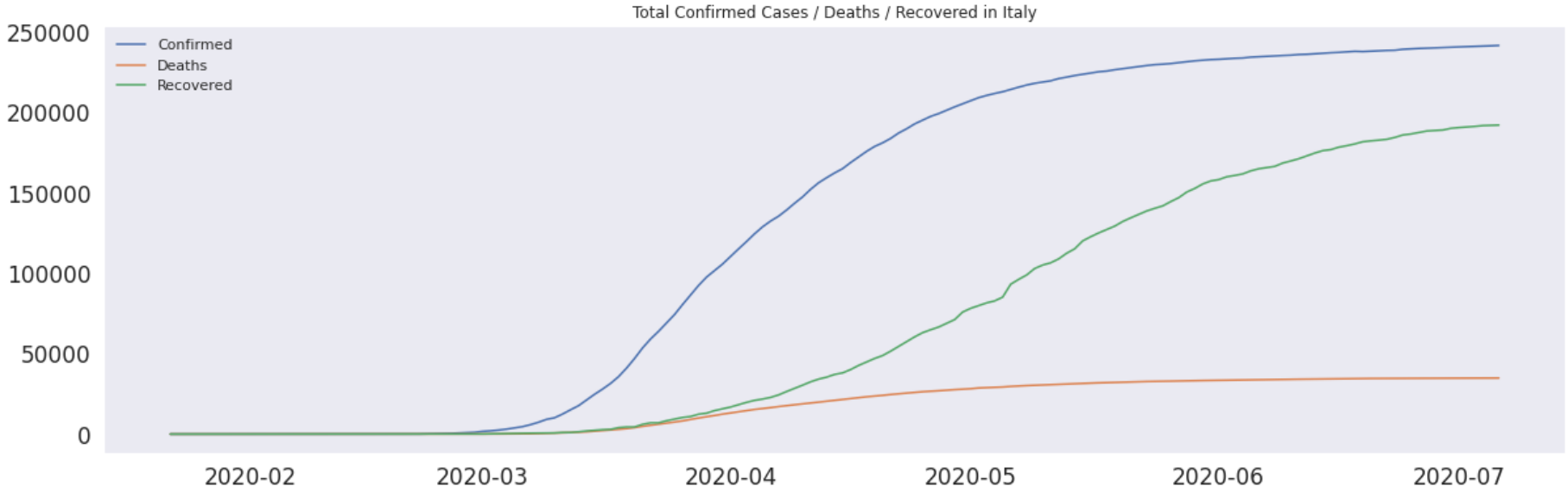

```
corona = pd.read_csv('/content/drive/My Drive/Python DataScience/Visualization/Seaborn/
                      index_col='Date' , parse_dates=True)
corona.head(10)
```

Date	Country	Confirmed	Recovered	Deaths
2020-01-22	Afghanistan	0	0	0
2020-01-22	Albania	0	0	0
2020-01-22	Algeria	0	0	0
2020-01-22	Andorra	0	0	0
2020-01-22	Angola	0	0	0
2020-01-22	Antigua and Barbuda	0	0	0

```
corona.head()
```

Date	Country	Confirmed	Recovered	Deaths
2020-01-22	Afghanistan	0	0	0
2020-01-22	Albania	0	0	0
2020-01-22	Algeria	0	0	0
2020-01-22	Andorra	0	0	0
2020-01-22	Angola	0	0	0

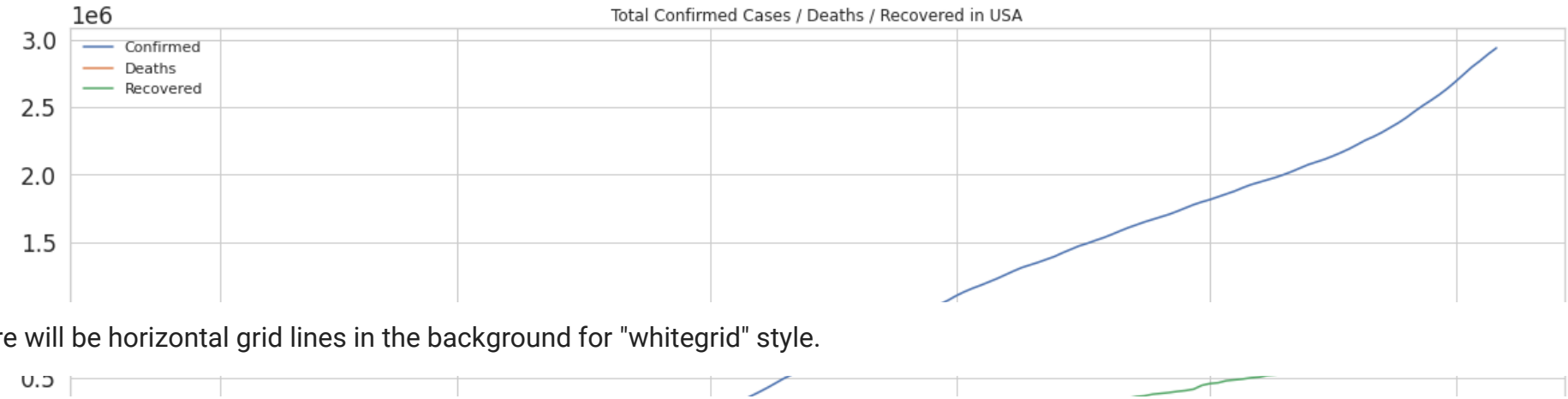
```
plt.figure(figsize=(20,6))
sns.lineplot(data=corona[corona['Country'] == 'Italy']['Confirmed'] , label = "Confirme
sns.lineplot(data=corona[corona['Country'] == 'Italy']['Deaths'] , label = "Deaths")
sns.lineplot(data=corona[corona['Country'] == 'Italy']['Recovered'], label = "Recovered
plt.title("Total Confirmed Cases / Deaths / Recovered in Italy")
plt.show()
```




```
plt.figure(figsize=(20,6))
sns.lineplot(data=corona[corona['Country'] == 'US']['Confirmed'] , label = "Confirmed")
sns.lineplot(data=corona[corona['Country'] == 'US']['Deaths'] , label = "Deaths")
sns.lineplot(data=corona[corona['Country'] == 'US']['Recovered'], label = "Recovered")
plt.title("Total Confirmed Cases / Deaths / Recovered in USA")
plt.show()
```

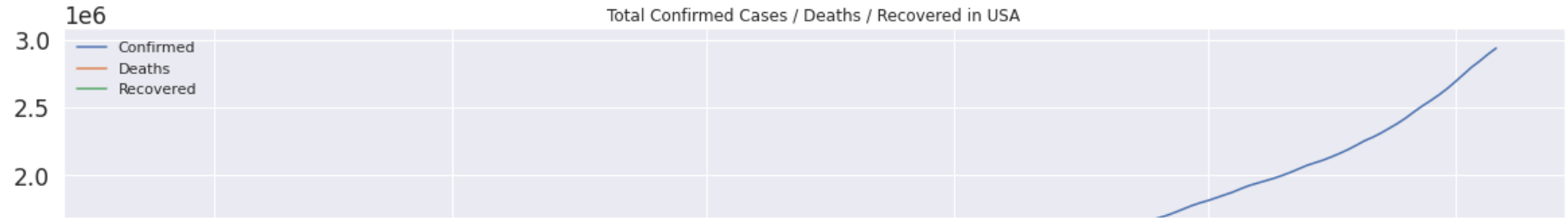
Plot Styling

```
# Enabling whitegrid style
sns.set_style("whitegrid")
plt.figure(figsize=(20,6))
sns.lineplot(data=corona[corona['Country'] == 'US']['Confirmed'] , label = "Confirmed")
sns.lineplot(data=corona[corona['Country'] == 'US']['Deaths'] , label = "Deaths")
sns.lineplot(data=corona[corona['Country'] == 'US']['Recovered'], label = "Recovered")
plt.title("Total Confirmed Cases / Deaths / Recovered in USA")
plt.show()
```



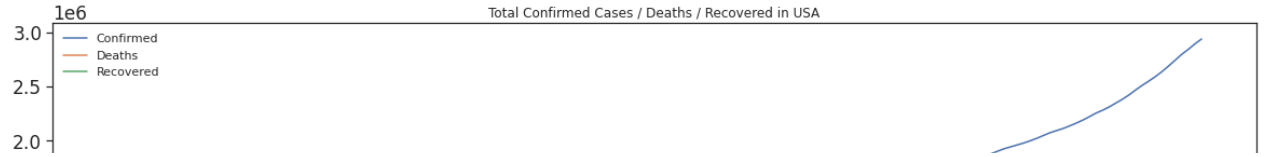
There will be horizontal grid lines in the background for "whitegrid" style.

```
# Enabling darkgrid style
sns.set_style("darkgrid")
plt.figure(figsize=(20,6))
sns.lineplot(data=corona[corona['Country'] == 'US']['Confirmed'] , label = "Confirmed")
sns.lineplot(data=corona[corona['Country'] == 'US']['Deaths'] , label = "Deaths")
sns.lineplot(data=corona[corona['Country'] == 'US']['Recovered'], label = "Recovered")
plt.title("Total Confirmed Cases / Deaths / Recovered in USA")
plt.show()
```



There will be horizontal grid lines in the background for "darkgrid" style. Also we will see a dark Background.

```
# Ticks Style
sns.set_style("ticks")
plt.figure(figsize=(20,6))
sns.lineplot(data=corona[corona['Country'] == 'US']['Confirmed'] , label = "Confirmed")
sns.lineplot(data=corona[corona['Country'] == 'US']['Deaths'] , label = "Deaths")
sns.lineplot(data=corona[corona['Country'] == 'US']['Recovered'], label = "Recovered")
plt.title("Total Confirmed Cases / Deaths / Recovered in USA")
plt.show()
```



There will be no horizontal grid lines in the background for "ticks" style. We will only see ticks in X & Y axis

Revert to default styling

[ ] *1 cell hidden*