GDP ANALYSIS USING DATA SCIENCE OPERATIONS

By: Sakshi Neeraj

This is an analysis made on the publicly available data for US, however the actual numerical values of the GDP can differ based upon the sources of obtaining data. Here, all the numerical values are in hundred thousand Dollars.

GDP = PERSONAL CONSUMPTION EXPENDITURES (PCE) + NET EXPORTS + GOVERNMENT CONSUMPTION EXPENDITURES AND GROSS INVESTMENT (GCE) + PRIVATE INVESTMENT

Datasets for practical-

https://www.quandl.com/data/BEA/T20805_M-Personal-Consumption-Expenditures-by-Major-Type-of-Product-Monthly-Monthly-data-from-1959M01-to-2018M05 (https://www.quandl.com/data/BEA/T20805_M-Personal-Consumption-Expenditures-by-Major-Type-of-Product-Monthly-Monthly-data-from-1959M01-to-2018M05)

https://www.quandl.com/data/BEA/T40205_A-Exports-and-Imports-of-Goods-and-Services-by-Type-of-Product-Annual-data-from-1967-to-2017 (https://www.quandl.com/data/BEA/T40205_A-Exports-and-Imports-of-Goods-and-Services-by-Type-of-Product-Annual-data-from-1967-to-2017)

https://www.quandl.com/data/BEA/T30905_A-Government-Consumption-Expenditures-and-Gross-Investment-Annual-data-from-1929-to-2017 (https://www.quandl.com/data/BEA/T30905_A-Government-Consumption-Expenditures-and-Gross-Investment-Annual-data-from-1929-to-2017)

https://www.quandl.com/data/BEA/GMP00998_REAL_PRIVATEINDUSTRIES_UNITEDSTATES-Real-GDP-Private-industries-United-States (https://www.quandl.com/data/BEA/GMP00998_REAL_PRIVATEINDUSTRIES_UNITEDSTATES-Real-GDP-Private-industries-United-States)

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
```

Importing PCE dataset for 2014 and calculting total PCE for 2014

In [3]:

```
PCE=pd.read_csv("personal consumption expenditure.csv") # This is the file path
df1=pd.DataFrame(PCE) # creaing a dataframe for storage of the imported data
# Printing only selective rows and columns by creating a new dataframe
df2=df1.loc[41:52,['Date','Personal consumption expenditures (PCE)']]
print(df2)
sum2014=df2['Personal consumption expenditures (PCE)'].sum() # storing the value of the tot
print("Total PCE of 2014= ",sum2014)
```

```
          Date  Personal consumption expenditures (PCE)
41  2014-12-31                                12095614.0
42  2014-11-30                                12098232.0
43  2014-10-31                                12050418.0
44  2014-09-30                                11975585.0
45  2014-08-31                                11966790.0
46  2014-07-31                                11881054.0
47  2014-06-30                                11840142.0
48  2014-05-31                                11787340.0
49  2014-04-30                                11748351.0
50  2014-03-31                                11715055.0
51  2014-02-28                                11629124.0
52  2014-01-31                                11576322.0
Total PCE of 2014=  142364027.0
```

Importing PCE dataset for 2013 and calculting total PCE for 2013

In [4]:

```
PCE=pd.read_csv("personal consumption expenditure.csv") # This is the file path
df3=pd.DataFrame(PCE) # making a dataframe for storage of the imported data
# Printing only selective rows and columns by creating a new dataframe
df4=df3.loc[53:64,['Date','Personal consumption expenditures (PCE)']]
print(df4)
sum2013=df4['Personal consumption expenditures (PCE)'].sum() # storing the value of the tot
print("Total PCE of 2013= ",sum2013)
```

```
          Date  Personal consumption expenditures (PCE)
53  2013-12-31                                11566012.0
54  2013-11-30                                11541950.0
55  2013-10-31                                11465233.0
56  2013-09-30                                11423949.0
57  2013-08-31                                11370236.0
58  2013-07-31                                11343123.0
59  2013-06-30                                11320199.0
60  2013-05-31                                11279494.0
61  2013-04-30                                11253931.0
62  2013-03-31                                11261641.0
63  2013-02-28                                11282963.0
64  2013-01-31                                11225360.0
Total PCE of 2013=  136334091.0
```

Importing PCE dataset for 2012 and calculting total PCE for 2012

In [5]:

```
PCE=pd.read_csv("personal consumption expenditure.csv") # This is the file path
df5=pd.DataFrame(PCE)
# Printing only selective rows and columns by creating a new dataframe
df6=df5.loc[65:76,['Date','Personal consumption expenditures (PCE)']]
print(df6)
sum2012=df6['Personal consumption expenditures (PCE)'].sum() # storing the value of the tot
print("Total PCE of 2012= ",sum2012)
```

```
          Date  Personal consumption expenditures (PCE)
65  2012-12-31                                11186525.0
66  2012-11-30                                11161726.0
67  2012-10-31                                11144735.0
68  2012-09-30                                11130983.0
69  2012-08-31                                11054923.0
70  2012-07-31                                11034902.0
71  2012-06-30                                10997390.0
72  2012-05-31                                11010291.0
73  2012-04-30                                11017359.0
74  2012-03-31                                10987202.0
75  2012-02-29                                10975925.0
76  2012-01-31                                10905563.0
Total PCE of 2012=  132607524.0
```

creating a new table for the PCE of the 3 years using dictionary

In [6]:

```
PCE_total={'Year':[2014,2013,2012],'Personal consumption expenditures (PCE)':[sum2014,sum20
df7=pd.DataFrame(PCE_total) # creaing a new dataframe from the previously formed table
print(df7)
```

```
   Year  Personal consumption expenditures (PCE)
0  2014                             142364027.0
1  2013                             136334091.0
2  2012                             132607524.0
```

Importing Exports, Imports dataset

In [7]:

```
EXPORTS_IMPORTS=pd.read_csv("export and import of goods and services.csv") # This is the fi
df8=pd.DataFrame(EXPORTS_IMPORTS) # Creating a data frame for storage of the imported data
# Printing only selective rows and columns by creating a new dataframe
df9=df8.loc[3:5,['Date','Exports of goods and services','::Imports of goods and services']]
print(df9)
```

```
         Date  Exports of goods and services  ::Imports of goods and service
s
3  2014-12-31                      2373648.0                         2883157.
0
4  2013-12-31                      2276608.0                         2768613.
0
5  2012-12-31                      2198182.0                         2763844.
0
```

Calculating net exports

In [8]:

```
df10=df9['Exports of goods and services']-df9['::Imports of goods and services'] # Creating
print(df10)
```

```
3   -509509.0
4   -492005.0
5   -565662.0
dtype: float64
```

creating a new dataframe for the net exports of the 3 years using dictionary

In [9]:

```
net_exports={'Year':[2014,2013,2012],'net exports':df10}
df11=pd.DataFrame(net_exports) # creating a dataframe from the table made previously
print(df11)
```

```
   Year  net exports
3  2014    -509509.0
4  2013    -492005.0
5  2012    -565662.0
```

Importing government consumption expenditures and gross investment dataset

In [10]:

```
government_consumption_expenditures_and_gross_investment=pd.read_csv("govt consumption expe
df12=pd.DataFrame(government_consumption_expenditures_and_gross_investment) # Creating a da
# Printing only selective rows and columns by creating a new dataframe
df13=df12.loc[3:5,['Date','Government consumption expenditures and gross investment']]
print(df13)
```

```
         Date  Government consumption expenditures and gross investment
3  2014-12-31                                          3157046.0
4  2013-12-31                                          3116051.0
5  2012-12-31                                          3158586.0
```

Importing private investment dataset

In [11]:

```
private_investment=pd.read_csv("private investments.csv") # This is the file path
df14=pd.DataFrame(private_investment) # Creating a data frame for storage of the imported d
# Printing only selective rows and columns by creating a new dataframe
df15=df14.loc[0:2,['Date','Value']]
print(df15)
```

```
         Date       Value
0  2014-12-31  12610597.0
1  2013-12-31  12283387.0
2  2012-12-31  12015103.0
```

Creating a new dataframe for the calculation of GDP of the 3 years

In [12]:

```
newlist_PCE=PCE_total['Personal consumption expenditures (PCE)'] # making a list from a tab
newlist_netexports=net_exports['net exports'] # making a list from a table
x=df13['Government consumption expenditures and gross investment'] # making a new dataframe
y=pd.Series(x) # making a series from a dataframe
GCE2014=y[3] # accessing elements of a series
GCE2013=y[4]
GCE2012=y[5]
newlist_government_consumption_expenditures_and_gross_investment=[GCE2014,GCE2013,GCE2012]
a=df15['Value'] # making a new dataframe from old dataframe
b=pd.Series(a) # making a series from a dataframe
PI2014=b[0] # accessing elements of a series
PI2013=b[1]
PI2012=b[2]
newlist_private_investment=[PI2014,PI2013,PI2012] # making a list
GDP={'year':[2014,2013,2012],'PCE':newlist_PCE,'net exports':newlist_netexports,
     'GCE':newlist_government_consumption_expenditures_and_gross_investment,
     'private investment':newlist_private_investment} # making a dictinary using lists
df16=pd.DataFrame(GDP) # making a dataframe using dictinary
print(df16)
```

```
   year          PCE  net exports        GCE  private investment
3  2014  142364027.0    -509509.0  3157046.0          12610597.0
4  2013  136334091.0    -492005.0  3116051.0          12283387.0
5  2012  132607524.0    -565662.0  3158586.0          12015103.0
```

Calculating the GDP

In [13]:

```
df16['GDP']=df16['PCE']+df16['net exports']+df16['GCE']+df16['private investment'] # adding
print(df16)
```

```
   year          PCE  net exports        GCE  private investment          GD
P
3  2014  142364027.0    -509509.0  3157046.0          12610597.0  157622161.
0
4  2013  136334091.0    -492005.0  3116051.0          12283387.0  151241524.
0
5  2012  132607524.0    -565662.0  3158586.0          12015103.0  147215551.
0
```

To find the coefficient of correlation of the 4 factors with GDP

In [14]:

```
GDP_dict={'GDP':df16['GDP']} # Creating a dictionary of GDP values
newlist_GDP=GDP_dict['GDP'] # Creating a list of GDP values from the previous dictionary
print(np.corrcoef(newlist_PCE,newlist_GDP)) # This gives correlation coefficient between 2
print(np.corrcoef(newlist_netexports,newlist_GDP)) # This gives correlation coefficient bet
print(np.corrcoef(newlist_government_consumption_expenditures_and_gross_investment,newlist_
print(np.corrcoef(newlist_private_investment,newlist_GDP)) # This gives correlation coeffic
```

```
[[1.         0.99998449]
 [0.99998449 1.        ]]
[[1.         0.63488205]
 [0.63488205 1.        ]]
[[1.         0.09782048]
 [0.09782048 1.        ]]
[[1.         0.99734902]
 [0.99734902 1.        ]]
```

Calculating rate of change

In [15]:

```
change1=(df16.loc[3,'GDP']-df16.loc[4,'GDP'])*100/df16.loc[3,'GDP'] # accessing GDP of 2 ye
print('rate of change between 2014 and 2013 is',round(change1,3),"%")
change2=(df16.loc[4,'GDP']-df16.loc[5,'GDP'])*100/df16.loc[4,'GDP'] # accessing GDP of 2 ye
print('rate of change between 2013 and 2012 is',round(change2,3),"%")
```

```
rate of change between 2014 and 2013 is 4.048 %
rate of change between 2013 and 2012 is 2.662 %
```

To find highest and lowest GDP year

In [16]:

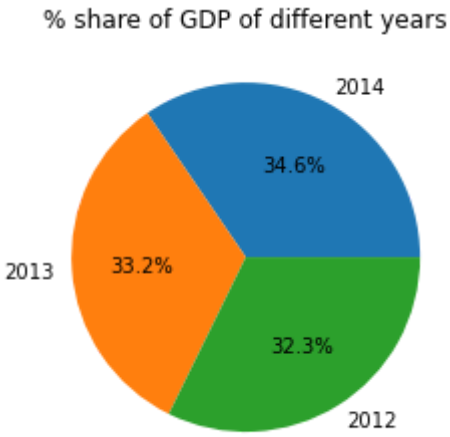
```
highest_GDP=df16['GDP'].max() # finding the value in the GDP column which is the highest
print('highest GDP=',df16['GDP'].max())
# comparing all the GDP values with the highest GDP to find the highest GDP index
print('index value of highest GDP is ',df16[df16['GDP']==highest_GDP].index.values) # print
print('year with highest GDP is ',df16.loc[3,'year']) # printing the year on the index of h
lowest_GDP=df16['GDP'].min() # finding the value in the GDP column which is the lowest
print('lowest GDP=',df16['GDP'].min())
# comparing all the GDP values with the lowest GDP to find the lowest GDP index
print('index value of lowest GDP is ',df16[df16['GDP']==lowest_GDP].index.values) # printin
print('year with lowest GDP is ',df16.loc[5,'year']) # printing the year on the index of lo
```

```
highest GDP= 157622161.0
index value of highest GDP is  [3]
year with highest GDP is  2014
lowest GDP= 147215551.0
index value of lowest GDP is  [5]
year with lowest GDP is  2012
```

To plot pie chart

In [17]:

```
year=[2014,2013,2012]
plt.pie(newlist_GDP,labels=year,autopct='%0.1f%%') # autopct='%0.1f%%' gives percentage sha
plt.title('% share of GDP of different years') # value for the pie chart is the GDP and the
plt.show() # print the pie chart
```



To forecast the GDP Here the data is not divided into training and testing dataset because lack of availability of data The whole data is taken for training the linear regression model

In [20]:

```
from sklearn.model_selection import train_test_split
x=df16[['PCE','net exports','GCE','private investment']]
y=df16.GDP
x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=0.2,random_state=3)
reg=linear_model.LinearRegression()
reg.fit(x_train,y_train)
print('regression coefficient=',reg.coef_) # Printing the regression coesfficients
print('regression intercept=',reg.intercept_) # Printing the intercept of the model
print('predicited GDP with the given factors is=',reg.predict([[1.2*(10**8),-723067,3003191
```

```
regression coefficient= [ 1.06263927e+00  6.11596011e-03 -1.67730639e-04  6.
48588241e-02]
regression intercept= 5526291.781786948
predicited GDP with the given factors is= [1.33792171e+08]
```