Homework 03

Any Questions:
--Google!
--Discuss with peers, post questions on the class Piazza (https://piazza.com/class/j6o5l788o874i (https://piazza.com/class/j6o5l788o874i))
--Come to Office Hours on Tuesday 11am to 12 pm in Etcheverry 4176B.

Submission:
Submit on bcourses as directed in the assignment instructions.

In [1]:
```
# Load required modules
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Reading File

1) Read in a CSV file called 'data3.csv' into a dataframe called df.

Data description

- Data source: http://www.fao.org/nr/water/aquastat/data/query/index.html?* (http://www.fao.org/nr/water/aquastat/data/query/index.html?*) lang=en
- Data, units:
- GDP, current USD (CPI adjusted)
- NRI, mm/yr
- Population density, inhab/km^2
- Total area of the country, 1000 ha = 10km^2
- Total Population, unit 1000 inhabitants

2.1) Display the first 10 lines of the dataframe

2.1) Display the column names.

In [18]:
```
df = pd.read_csv('data3.csv')
print(df.head(10))
print()
print('column names')
print(list(df))
```

```
        Area  Area Id              Variable Name  Variable Id    Year  \
0  Argentina      9.0  Total area of the country       4100.0  1962.0
1  Argentina      9.0  Total area of the country       4100.0  1967.0
2  Argentina      9.0  Total area of the country       4100.0  1972.0
3  Argentina      9.0  Total area of the country       4100.0  1977.0
4  Argentina      9.0  Total area of the country       4100.0  1982.0
5  Argentina      9.0  Total area of the country       4100.0  1987.0
6  Argentina      9.0  Total area of the country       4100.0  1992.0
7  Argentina      9.0  Total area of the country       4100.0  1997.0
8  Argentina      9.0  Total area of the country       4100.0  2002.0
9  Argentina      9.0  Total area of the country       4100.0  2007.0

      Value Symbol  Other
0  278040.0      E    NaN
1  278040.0      E    NaN
2  278040.0      E    NaN
3  278040.0      E    NaN
4  278040.0      E    NaN
5  278040.0      E    NaN
6  278040.0      E    NaN
7  278040.0      E    NaN
8  278040.0      E    NaN
9  278040.0      E    NaN

column names
['Area', 'Area Id', 'Variable Name', 'Variable Id', 'Year', 'Value', 'Symbol',
 'Other']
```

Data Preprocessing

3.1) Create a mask of NAN values(i.e. apply .isnull on the dataframe). Inspect the mask for 'True' values, they denote NANs.

Hint: [You will notice that the last 8 rows and the last column ('Other') have NAN values.You can also use df.tail() to see the last lines.]

3.2) Now, we will try to get rid of the NaN valued rows and columns. Remove the bottom 8 rows from the dataframe. Also remove the column 'Other'.

In [19]:
```
mask = df.isnull()
df = df[:-8]
df = df.drop(df[['Other']], axis=1)
df.head(5)
```

Out[19]:

	Area	Area Id	Variable Name	Variable Id	Year	Value	Symbol
0	Argentina	9.0	Total area of the country	4100.0	1962.0	278040.0	E
1	Argentina	9.0	Total area of the country	4100.0	1967.0	278040.0	E
2	Argentina	9.0	Total area of the country	4100.0	1972.0	278040.0	E
3	Argentina	9.0	Total area of the country	4100.0	1977.0	278040.0	E
4	Argentina	9.0	Total area of the country	4100.0	1982.0	278040.0	E

4.1) For our analysis we do not want all the columns in our dataframe. Lets drop all the redundant columns/ features.

Drop columns: Area Id, Variable Id, Symbol. Save the new dataframe as df1.

In [23]:
```
df1 = df.drop(df[['Area Id', 'Variable Id', 'Symbol']], axis=1)
df1.head(5)
```

Out[23]:

	Area	Variable Name	Year	Value
0	Argentina	Total area of the country	1962.0	278040.0
1	Argentina	Total area of the country	1967.0	278040.0
2	Argentina	Total area of the country	1972.0	278040.0
3	Argentina	Total area of the country	1977.0	278040.0
4	Argentina	Total area of the country	1982.0	278040.0

4.2) Display all the unique values in your new dataframe for column: Area, Variable Name, Year.

Note the Countries and the Metrics (ie.recorded variables) represented in your dataset.

Hint: Use .unique( ) method.

In [30]:
```
print('unique area')
print(df1['Area'].unique())
print()
print('unique name')
print(df1['Variable Name'].unique())
print()
print('Unique year')
print(df1['Year'].unique())
```

```
unique area
['Argentina' 'Australia' 'Germany' 'Iceland' 'Ireland' 'Sweden'
 'United States of America']

unique name
['Total area of the country' 'Total population' 'Population density'
 'Gross Domestic Product (GDP)' 'National Rainfall Index (NRI)']

Unique year
[ 1962.  1967.  1972.  1977.  1982.  1987.  1992.  1997.  2002.  2007.
  2012.  2014.  2015.  1963.  1970.  1974.  1978.  1984.  1990.  1964.
  1981.  1985.  1996.  2001.  1969.  1973.  1979.  1993.  1971.  1975.
  1986.  1991.  1998.  2000.  1965.  1983.  1988.  1995.]
```

5) Convert the year column to pandas datetime.

Convert the 'Year' column string values to pandas datetime objects, where only the year is specified.
Hint:
df1['Year'] = pd.to_datetime(pd.Series(df1['Year']).astype(int),format='%Y').dt.year

Run df1.tail() to see if you get what you expect

In [33]:
```
df1['Year'] = pd.to_datetime(pd.Series(df1['Year']).astype(int),format='%Y').dt.
df1.tail(3)
```

Out[33]:

	Area	Variable Name	Year	Value
387	United States of America	National Rainfall Index (NRI)	1992	1020.0
388	United States of America	National Rainfall Index (NRI)	1996	1005.0
389	United States of America	National Rainfall Index (NRI)	2002	938.7

Extract specific statistics from the preprocessed data:

6) Create a dataframe 'dftemp' to store rows where Area is Iceland.

In [34]:
```
dftemp = df1[df1.Area=='Iceland']
print(dftemp)
```

```
        Area                  Variable Name  Year         Value
166  Iceland      Total area of the country  1962  1.030000e+04
167  Iceland      Total area of the country  1967  1.030000e+04
168  Iceland      Total area of the country  1972  1.030000e+04
169  Iceland      Total area of the country  1977  1.030000e+04
170  Iceland      Total area of the country  1982  1.030000e+04
171  Iceland      Total area of the country  1987  1.030000e+04
172  Iceland      Total area of the country  1992  1.030000e+04
173  Iceland      Total area of the country  1997  1.030000e+04
174  Iceland      Total area of the country  2002  1.030000e+04
175  Iceland      Total area of the country  2007  1.030000e+04
176  Iceland      Total area of the country  2012  1.030000e+04
177  Iceland      Total area of the country  2014  1.030000e+04
178  Iceland               Total population  1962  1.826000e+02
179  Iceland               Total population  1967  1.974000e+02
180  Iceland               Total population  1972  2.099000e+02
181  Iceland               Total population  1977  2.221000e+02
182  Iceland               Total population  1982  2.331000e+02
183  Iceland               Total population  1987  2.469000e+02
184  Iceland               Total population  1992  2.599000e+02
185  Iceland               Total population  1997  2.728000e+02
186  Iceland               Total population  2002  2.869000e+02
187  Iceland               Total population  2007  3.054000e+02
188  Iceland               Total population  2012  3.234000e+02
189  Iceland               Total population  2015  3.294000e+02
190  Iceland            Population density  1962  1.773000e+00
191  Iceland            Population density  1967  1.917000e+00
192  Iceland            Population density  1972  2.038000e+00
193  Iceland            Population density  1977  2.156000e+00
194  Iceland            Population density  1982  2.263000e+00
195  Iceland            Population density  1987  2.397000e+00
196  Iceland            Population density  1992  2.523000e+00
197  Iceland            Population density  1997  2.649000e+00
198  Iceland            Population density  2002  2.785000e+00
199  Iceland            Population density  2007  2.965000e+00
200  Iceland            Population density  2012  3.140000e+00
201  Iceland            Population density  2015  3.198000e+00
202  Iceland   Gross Domestic Product (GDP)  1962  2.849165e+08
203  Iceland   Gross Domestic Product (GDP)  1967  6.212260e+08
204  Iceland   Gross Domestic Product (GDP)  1972  8.465069e+08
205  Iceland   Gross Domestic Product (GDP)  1977  2.226539e+09
206  Iceland   Gross Domestic Product (GDP)  1982  3.232804e+09
207  Iceland   Gross Domestic Product (GDP)  1987  5.565384e+09
208  Iceland   Gross Domestic Product (GDP)  1992  7.138788e+09
209  Iceland   Gross Domestic Product (GDP)  1997  7.596126e+09
210  Iceland   Gross Domestic Product (GDP)  2002  9.161798e+09
211  Iceland   Gross Domestic Product (GDP)  2007  2.129384e+10
212  Iceland   Gross Domestic Product (GDP)  2012  1.419452e+10
213  Iceland   Gross Domestic Product (GDP)  2015  1.659849e+10
214  Iceland  National Rainfall Index (NRI)  1967  8.160000e+02
215  Iceland  National Rainfall Index (NRI)  1971  9.632000e+02
216  Iceland  National Rainfall Index (NRI)  1975  1.010000e+03
217  Iceland  National Rainfall Index (NRI)  1981  9.326000e+02
```

```
218  Iceland  National Rainfall Index (NRI)  1986  9.685000e+02
219  Iceland  National Rainfall Index (NRI)  1991  1.095000e+03
220  Iceland  National Rainfall Index (NRI)  1997  9.932000e+02
221  Iceland  National Rainfall Index (NRI)  1998  9.234000e+02
```

7) Print the years when the National Rainfall Index (NRI) was greater than 950 or less than 900 in Iceland. Use the dataframe you created in the previous question 'dftemp'.

In [38]:
```
print(dftemp['Year'][((df.Value<900)|(df.Value>950))].unique())
```

```
[1962 1967 1972 1977 1982 1987 1992 1997 2002 2007 2012 2014 2015 1971 1975
 1986 1991]
```

US statistics:

8) Get all the rows of df1 (preprocessed dataframe) area is United States of America

1) Create a new DataFrame called `df_usa` that only contains values where 'Area' is equal to 'United States of America'. Set the indices to be the 'Year' column (Use .set_index( ))

2) Pivot the DataFrame so that the unique 'Variable Name' entries becomes the column entries. The DataFrame values should be the ones in the the 'Value' column. Do this by running the three lines of code below:

```
df_usa=df_usa.pivot(columns='Variable Name',values='Value')
```

3) Display df_usa.head( ), rename new columns to ['GDP','NRI','PD','Area','Population']

In [79]:
```
df_usa = df1[df1.Area == 'United States of America'].set_index('Year')
df_usa=df_usa.pivot(columns='Variable Name',values='Value')
df_usa = df_usa.rename(index=str, columns = {'Gross Domestic Product (GDP)':'GDP
df_usa.head(4)
```

Out[79]:

Variable Name	GDP	NRI	PD	Area	Population
Year					
1962	6.050000e+11	NaN	19.93	962909.0	191861.0
1965	NaN	928.5	NaN	NaN	NaN
1967	8.620000e+11	NaN	21.16	962909.0	203713.0
1969	NaN	952.2	NaN	NaN	NaN

4) Find `df_usa.isnull().sum()`.This gives us the number of NAN values in each column. Replace NAN values by 0, using `df_usa=df_usa.fillna(0)`. Again check `df_usa.isnull().sum()`.'

In [80]:
```
df_usa.isnull().sum()
df_usa_nonNan=df_usa.fillna(0)
df_usa_nonNan.isnull().sum()
```

Out[80]:
```
Variable Name
GDP           0
NRI           0
PD            0
Area          0
Population    0
dtype: int64
```

5) Calculate and print all the column averages and the column standard deviations.

In [81]:
```
print('I calculate the mean and std of metric with NAN value instead of 0')
metric = df_usa.describe()
metric.loc[('mean','std'),:]
```

```
I calculate the mean and std of metric with NAN value instead of 0
```

Out[81]:

Variable Name	GDP	NRI	PD	Area	Population
mean	7.316417e+12	972.025000	26.444167	966331.666667	255729.583333
std	6.256868e+12	35.068861	4.425996	7857.100059	44281.029610

9) Use df_usa:

1: Multiply the Area by 10 (so instead of 1000 ha, the unit becomes 100 ha = 1km^2)

In [82]:
```
df_usa['Area'] = df_usa['Area']*10
df_usa.head(5)
```

Out[82]:

Variable Name	GDP	NRI	PD	Area	Population
Year					
1962	6.050000e+11	NaN	19.93	9629090.0	191861.0
1965	NaN	928.5	NaN	NaN	NaN
1967	8.620000e+11	NaN	21.16	9629090.0	203713.0
1969	NaN	952.2	NaN	NaN	NaN
1972	1.280000e+12	NaN	22.14	9629090.0	213220.0

2: Create a new column in df_us called 'GDP/capita' and populate it with the calculated GDP per capita. Round the results to two decimal points.

In [90]:
```
df_usa['GDP/capita'] = df_usa['GDP']/(df_usa['Population']*1000)
df_usa
```

Out[90]:

Variable Name	GDP	NRI	PD	Area	Population	GDP/capita	PD2
Year							
1962	6.050000e+11	NaN	19.93	9629090.0	191861.0	3153.324542	0.02
1965	NaN	928.5	NaN	NaN	NaN	NaN	NaN
1967	8.620000e+11	NaN	21.16	9629090.0	203713.0	4231.443256	0.02
1969	NaN	952.2	NaN	NaN	NaN	NaN	NaN
1972	1.280000e+12	NaN	22.14	9629090.0	213220.0	6003.189194	0.02
1974	NaN	1008.0	NaN	NaN	NaN	NaN	NaN
1977	2.090000e+12	NaN	23.17	9629090.0	223091.0	9368.374341	0.02
1981	NaN	949.2	NaN	NaN	NaN	NaN	NaN
1982	3.340000e+12	NaN	24.30	9629090.0	233954.0	14276.310728	0.02
1984	NaN	974.6	NaN	NaN	NaN	NaN	NaN
1987	4.870000e+12	NaN	25.49	9629090.0	245425.0	19843.129266	0.03
1992	6.540000e+12	1020.0	26.78	9629090.0	257908.0	25357.879554	0.03
1996	NaN	1005.0	NaN	NaN	NaN	NaN	NaN
1997	8.610000e+12	NaN	28.34	9629090.0	272883.0	31551.983817	0.03
2002	1.100000e+13	938.7	29.95	9632030.0	288471.0	38132.082601	0.03
2007	1.450000e+13	NaN	31.32	9632030.0	301656.0	48067.997984	0.03
2012	1.620000e+13	NaN	32.02	9831510.0	314799.0	51461.408708	0.03
2014	NaN	NaN	NaN	9831510.0	NaN	NaN	NaN
2015	1.790000e+13	NaN	32.73	NaN	321774.0	55629.106143	NaN

3: Create a new column called 'PD2' (i.e. Population density 2). Calculate the Population density. Note: the units should be inhab/km^2 (see Data description above). Round the reults to two decimal point.

In [91]:
```
df_usa['PD2']=(df_usa['Population']*1000/df_usa['Area']).round(2)
df_usa
```

Out[91]:

Variable Name	GDP	NRI	PD	Area	Population	GDP/capita	PD2
Year							
1962	6.050000e+11	NaN	19.93	9629090.0	191861.0	3153.324542	19.93
1965	NaN	928.5	NaN	NaN	NaN	NaN	NaN
1967	8.620000e+11	NaN	21.16	9629090.0	203713.0	4231.443256	21.16
1969	NaN	952.2	NaN	NaN	NaN	NaN	NaN
1972	1.280000e+12	NaN	22.14	9629090.0	213220.0	6003.189194	22.14
1974	NaN	1008.0	NaN	NaN	NaN	NaN	NaN
1977	2.090000e+12	NaN	23.17	9629090.0	223091.0	9368.374341	23.17
1981	NaN	949.2	NaN	NaN	NaN	NaN	NaN
1982	3.340000e+12	NaN	24.30	9629090.0	233954.0	14276.310728	24.30
1984	NaN	974.6	NaN	NaN	NaN	NaN	NaN
1987	4.870000e+12	NaN	25.49	9629090.0	245425.0	19843.129266	25.49
1992	6.540000e+12	1020.0	26.78	9629090.0	257908.0	25357.879554	26.78
1996	NaN	1005.0	NaN	NaN	NaN	NaN	NaN
1997	8.610000e+12	NaN	28.34	9629090.0	272883.0	31551.983817	28.34
2002	1.100000e+13	938.7	29.95	9632030.0	288471.0	38132.082601	29.95
2007	1.450000e+13	NaN	31.32	9632030.0	301656.0	48067.997984	31.32
2012	1.620000e+13	NaN	32.02	9831510.0	314799.0	51461.408708	32.02
2014	NaN	NaN	NaN	9831510.0	NaN	NaN	NaN
2015	1.790000e+13	NaN	32.73	NaN	321774.0	55629.106143	NaN

4: Find the maximum value and minimum value of the 'NRI' column in the US (using pandas methods). What years do the min and max values occur?

In [117]:
```
print('year maximum NRI is')
print(df_usa['NRI'].ix[df_usa[['NRI']].idxmax()])
print()
print('year minimum NRI is')
print(df_usa['NRI'].ix[df_usa[['NRI']].idxmin()])
```

```
year maximum NRI is
Year
1992    1020.0
Name: NRI, dtype: float64

year minimum NRI is
Year
1965    928.5
Name: NRI, dtype: float64
```

Now, lets read another CSV file.

See https://www.quantshare.com/sa-43-10-ways-to-download-historical-stock-quotes-data-for-free (https://www.quantshare.com/sa-43-10-ways-to-download-historical-stock-quotes-data-for-free)

10 a) Show a 3 x 3 correlation matrix for Nike, Apple, and Disney stock prices for the month of July, 2017

In [119]:
```
dfa = pd.read_csv('https://www.google.com/finance/historical?output=csv&q=aapl')
dfn = pd.read_csv('https://www.google.com/finance/historical?output=csv&q=nke')
dfd = pd.read_csv('https://www.google.com/finance/historical?output=csv&q=dis')
```

In [133]:
```
dfa = dfa.rename(columns = {'Close':'AAPL'})
dfd = dfd.rename(columns = {'Close':'DIS'})
dfn = dfn.rename(columns = {'Close':'NKE'})
df_full = dfa[['Date','AAPL']].merge(dfd[['Date','DIS']])
df_full = df_full.merge(dfn[['Date','NKE']])
corrl = df_full[32:52].corr()
print(corrl)
```

```
          AAPL       DIS       NKE
AAPL  1.000000  0.524912  0.417947
DIS   0.524912  1.000000  0.459045
NKE   0.417947  0.459045  1.000000
```

10b) Show the same correlation matrix but over different time periods,

i) the last 20 days ii) the last 80 days

In [134]:
```
print('corr the last 20 days')
print(df_full[:20].corr())
print()
print('corr the last 80 days')
print(df_full[:80].corr())
```

```
corr the last 20 days
          AAPL       DIS       NKE
AAPL  1.000000  0.237467 -0.547436
DIS   0.237467  1.000000  0.240004
NKE  -0.547436  0.240004  1.000000

corr the last 80 days
          AAPL       DIS       NKE
AAPL  1.000000 -0.507881 -0.078793
DIS  -0.507881  1.000000  0.272226
NKE  -0.078793  0.272226  1.000000
```

11) Change the code so that it accepts a list of any stock symbols, ie ['NKE', 'APPL', 'DIS', ...] and creates a correlation matrix for the time period of the past 100 days

In [135]:
```
# Insert list of companies here. Should be >=2 companies
l = ['AAPL', 'NKE', 'DIS', 'GOOG']
df1  = pd.read_csv('https://www.google.com/finance/historical?output=csv&q='+l[0
df2  = pd.read_csv('https://www.google.com/finance/historical?output=csv&q='+l[1
df1 = df1.rename(columns = {'Close':l[0]})
df2 = df2.rename(columns = {'Close':l[1]})

dff = df1[['Date',l[0]]].merge(df2[['Date',l[1]]])

for n in l[2:]:
    dfn = pd.read_csv('https://www.google.com/finance/historical?output=csv&q='+
    dfn = dfn.rename(columns = {'Close':n})
    dff = dff.merge(dfn[['Date', n]])

corr = dff[:100].corr()
print(corr)
```

```
          AAPL       NKE       DIS      GOOG
AAPL  1.000000 -0.068436 -0.529913  0.028501
NKE  -0.068436  1.000000  0.106217 -0.201377
DIS  -0.529913  0.106217  1.000000 -0.091609
GOOG  0.028501 -0.201377 -0.091609  1.000000
```