hw4

September 12, 2018

0.1 Homework 04

0.1.1 NAME: Louis Tilloy

0.2 STUDENT ID: 3034388270

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

0.3 Pandas Introduction

0.4 Reading File

1.1) Read the 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?* 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

```
In [2]: dataframe = pd.read_csv("data3.csv")
```

2.1) Display the first 10 rows of the dataframe

```
In [3]: dataframe.head(10)
```

```
Out[3]:         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
```

2.2) Display the column names.

```
In [4]: dataframe.columns
```

```
Out[4]: Index(['Area', 'Area Id', 'Variable Name', 'Variable Id', 'Year', 'Value',
               'Symbol', 'Other'],
              dtype='object')
```

2.3) Use iloc to display the first 3 rows and first 4 columns.

```
In [5]: dataframe.iloc[0:3, 0:4]
```

```
Out[5]:         Area  Area Id              Variable Name  Variable Id
        0  Argentina      9.0  Total area of the country       4100.0
        1  Argentina      9.0  Total area of the country       4100.0
        2  Argentina      9.0  Total area of the country       4100.0
```

0.5 Data Preprocessing

3.1) Find all the rows that have 'NaN' in the 'Symbol' column. Display first 5 rows.

Hint : You might have to use a mask

```
In [6]: nan_rows = dataframe[dataframe["Symbol"].isnull()]
        nan_rows.head(5)
```

```
Out[6]:                      Area  Area Id Variable Name  Variable Id  Year  Value  \
        390                   NaN      NaN           NaN          NaN   NaN    NaN
        391       E - External data      NaN           NaN          NaN   NaN    NaN
        392  I - AQUASTAT estimate      NaN           NaN          NaN   NaN    NaN
```

```
393     K - Aggregate data       NaN             NaN            NaN    NaN     NaN
394      L - Modelled data       NaN             NaN            NaN    NaN     NaN

    Symbol  Other
390    NaN    NaN
391    NaN    NaN
392    NaN    NaN
393    NaN    NaN
394    NaN    NaN
```

3.2) Now, we will try to get rid of the NaN valued rows and columns. First, drop the column 'Other' which only has 'NaN' values. Then drop all other rows that have any column with a value 'NaN'. Then display the last 5 rows of the dataframe.

```
In [7]: dataframe_2 = dataframe.drop(columns='Other')
        dataframe_2 = dataframe_2.dropna()
        dataframe_2.head()
```

```
Out[7]:         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

              Value Symbol
        0  278040.0      E
        1  278040.0      E
        2  278040.0      E
        3  278040.0      E
        4  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. Display the first 5 rows of the new dataframe.

```
In [8]: df1 = dataframe_2.drop(columns=["Area Id", "Variable Id", "Symbol"])
        df1.head()
```

```
Out[8]:         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 columns: Area, Variable Name, Year.

```
In [9]: print("Area: ", df1["Area"].unique())
        print()

        print("Variable Name: ", df1["Variable Name"].unique())
        print()

        print("Year: ", df1["Year"].unique())
```

```
Area:  ['Argentina' 'Australia' 'Germany' 'Iceland' 'Ireland' 'Sweden'
 'United States of America']

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

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.]
```

4.3) Convert the Year column to pandas datetime. Convert the 'Year' column float values to pandas datetime objects, where each year is represented as the first day of that year. Also display the first 5 values of column after conversion.

For eg: 1962.0 will be represented as 1962-01-01

```
In [10]: df1["Year"] = pd.to_datetime(df1["Year"].astype(int), format="%Y")
         df1.head()
```

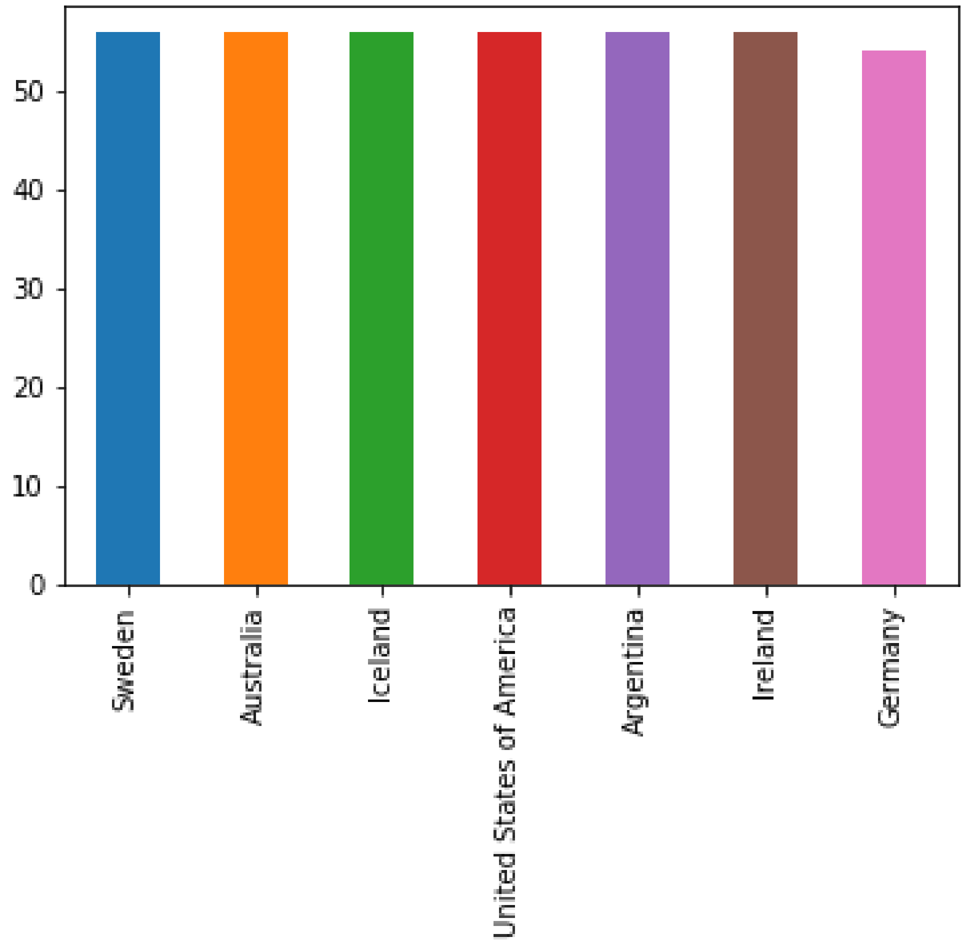
```
Out[10]:         Area              Variable Name       Year     Value
         0  Argentina  Total area of the country 1962-01-01  278040.0
         1  Argentina  Total area of the country 1967-01-01  278040.0
         2  Argentina  Total area of the country 1972-01-01  278040.0
         3  Argentina  Total area of the country 1977-01-01  278040.0
         4  Argentina  Total area of the country 1982-01-01  278040.0
```

0.6 Plot

5.1) Use pandas to plot a bar graph showing the count for each unique value in the column 'Area'. Give it a title.

```
In [11]: df1['Area'].value_counts().plot(kind='bar')
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x21d549d8ba8>
```



0.7 Extract specific statistics from the preprocessed data:

6.1) Create a dataframe 'dftemp' to store rows where Area is 'Iceland'. Display the dataframe.

```
In [12]: dftemp = df1[df1['Area'] == "Iceland"]
         dftemp
```

```
Out[12]:         Area                Variable Name       Year         Value
         166  Iceland      Total area of the country 1962-01-01  1.030000e+04
         167  Iceland      Total area of the country 1967-01-01  1.030000e+04
         168  Iceland      Total area of the country 1972-01-01  1.030000e+04
         169  Iceland      Total area of the country 1977-01-01  1.030000e+04
         170  Iceland      Total area of the country 1982-01-01  1.030000e+04
         171  Iceland      Total area of the country 1987-01-01  1.030000e+04
         172  Iceland      Total area of the country 1992-01-01  1.030000e+04
         173  Iceland      Total area of the country 1997-01-01  1.030000e+04
         174  Iceland      Total area of the country 2002-01-01  1.030000e+04
```

```
175  Iceland      Total area of the country 2007-01-01  1.030000e+04
176  Iceland      Total area of the country 2012-01-01  1.030000e+04
177  Iceland      Total area of the country 2014-01-01  1.030000e+04
178  Iceland               Total population 1962-01-01  1.826000e+02
179  Iceland               Total population 1967-01-01  1.974000e+02
180  Iceland               Total population 1972-01-01  2.099000e+02
181  Iceland               Total population 1977-01-01  2.221000e+02
182  Iceland               Total population 1982-01-01  2.331000e+02
183  Iceland               Total population 1987-01-01  2.469000e+02
184  Iceland               Total population 1992-01-01  2.599000e+02
185  Iceland               Total population 1997-01-01  2.728000e+02
186  Iceland               Total population 2002-01-01  2.869000e+02
187  Iceland               Total population 2007-01-01  3.054000e+02
188  Iceland               Total population 2012-01-01  3.234000e+02
189  Iceland               Total population 2015-01-01  3.294000e+02
190  Iceland             Population density 1962-01-01  1.773000e+00
191  Iceland             Population density 1967-01-01  1.917000e+00
192  Iceland             Population density 1972-01-01  2.038000e+00
193  Iceland             Population density 1977-01-01  2.156000e+00
194  Iceland             Population density 1982-01-01  2.263000e+00
195  Iceland             Population density 1987-01-01  2.397000e+00
196  Iceland             Population density 1992-01-01  2.523000e+00
197  Iceland             Population density 1997-01-01  2.649000e+00
198  Iceland             Population density 2002-01-01  2.785000e+00
199  Iceland             Population density 2007-01-01  2.965000e+00
200  Iceland             Population density 2012-01-01  3.140000e+00
201  Iceland             Population density 2015-01-01  3.198000e+00
202  Iceland   Gross Domestic Product (GDP) 1962-01-01  2.849165e+08
203  Iceland   Gross Domestic Product (GDP) 1967-01-01  6.212260e+08
204  Iceland   Gross Domestic Product (GDP) 1972-01-01  8.465069e+08
205  Iceland   Gross Domestic Product (GDP) 1977-01-01  2.226539e+09
206  Iceland   Gross Domestic Product (GDP) 1982-01-01  3.232804e+09
207  Iceland   Gross Domestic Product (GDP) 1987-01-01  5.565384e+09
208  Iceland   Gross Domestic Product (GDP) 1992-01-01  7.138788e+09
209  Iceland   Gross Domestic Product (GDP) 1997-01-01  7.596126e+09
210  Iceland   Gross Domestic Product (GDP) 2002-01-01  9.161798e+09
211  Iceland   Gross Domestic Product (GDP) 2007-01-01  2.129384e+10
212  Iceland   Gross Domestic Product (GDP) 2012-01-01  1.419452e+10
213  Iceland   Gross Domestic Product (GDP) 2015-01-01  1.659849e+10
214  Iceland  National Rainfall Index (NRI) 1967-01-01  8.160000e+02
215  Iceland  National Rainfall Index (NRI) 1971-01-01  9.632000e+02
216  Iceland  National Rainfall Index (NRI) 1975-01-01  1.010000e+03
217  Iceland  National Rainfall Index (NRI) 1981-01-01  9.326000e+02
218  Iceland  National Rainfall Index (NRI) 1986-01-01  9.685000e+02
219  Iceland  National Rainfall Index (NRI) 1991-01-01  1.095000e+03
220  Iceland  National Rainfall Index (NRI) 1997-01-01  9.932000e+02
221  Iceland  National Rainfall Index (NRI) 1998-01-01  9.234000e+02
```

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

```
In [13]: dftemp[(dftemp["Variable Name"] == "National Rainfall Index (NRI)") &
                (dftemp["Value"] < 950) &
                (dftemp["Value"] > 900)]
```

```
Out[13]:         Area                  Variable Name        Year  Value
         217  Iceland  National Rainfall Index (NRI) 1981-01-01  932.6
         221  Iceland  National Rainfall Index (NRI) 1998-01-01  923.4
```

0.8 US statistics:

8.1) 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( )). Display the dataframe head.

```
In [14]: df_usa = df1[df1["Area"] == "United States of America"].set_index("Year")
         df_usa.head()
```

```
Out[14]:                                Area            Variable Name     Value
         Year
         1962-01-01  United States of America  Total area of the country  962909.0
         1967-01-01  United States of America  Total area of the country  962909.0
         1972-01-01  United States of America  Total area of the country  962909.0
         1977-01-01  United States of America  Total area of the country  962909.0
         1982-01-01  United States of America  Total area of the country  962909.0
```

8.2) Pivot the DataFrame so that the unique values in the column 'Variable Name' becomes the columns. The DataFrame values should be the ones in the the 'Value' column. Save it in df_usa. Display the dataframe head.

```
In [15]: df_variables = df_usa.pivot(index=df_usa.index, columns='Variable Name')['Value']
         df_usa = df_usa.drop(columns=["Variable Name", "Value"])
         df_usa[df_variables.columns] = df_variables
         df_usa.head()
```

```
Out[15]:                                Area  Gross Domestic Product (GDP)  \
         Year
         1962-01-01  United States of America                  6.050000e+11
         1967-01-01  United States of America                  8.620000e+11
         1972-01-01  United States of America                  1.280000e+12
         1977-01-01  United States of America                  2.090000e+12
         1982-01-01  United States of America                  3.340000e+12

                     National Rainfall Index (NRI)  Population density  \
         Year
         1962-01-01                            NaN               19.93
         1967-01-01                            NaN               21.16
```

```
1972-01-01                            NaN             22.14
1977-01-01                            NaN             23.17
1982-01-01                            NaN             24.30

            Total area of the country  Total population
Year
1962-01-01                   962909.0          191861.0
1967-01-01                   962909.0          203713.0
1972-01-01                   962909.0          213220.0
1977-01-01                   962909.0          223091.0
1982-01-01                   962909.0          233954.0
```

8.3) Rename new columns to ['GDP','NRI','PD','Area','Population'] and display the head.

```
In [16]: # I chose to drop the original "Area" column since it is "United States of America"
         # for each row of the dataframe and since it is in conflict with the new "Area" colum
         # that we want to create
         df_usa = df_usa.drop(columns="Area")

         df_usa = df_usa.rename({
             "Gross Domestic Product (GDP)": "GDP",
             "National Rainfall Index (NRI)": "NRI",
             "Population density": "PD",
             "Total area of the country": "Area",
             "Total population": "Population"
         }, axis="columns")
         df_usa.head()
```

```
Out[16]:                      GDP  NRI     PD      Area  Population
         Year
         1962-01-01  6.050000e+11  NaN  19.93  962909.0    191861.0
         1967-01-01  8.620000e+11  NaN  21.16  962909.0    203713.0
         1972-01-01  1.280000e+12  NaN  22.14  962909.0    213220.0
         1977-01-01  2.090000e+12  NaN  23.17  962909.0    223091.0
         1982-01-01  3.340000e+12  NaN  24.30  962909.0    233954.0
```

8.4) Replace all 'Nan' values in df_usa with 0. Display the head of the dataframe.

```
In [17]: df_usa = df_usa.fillna(value=0)
         df_usa.head()
```

```
Out[17]:                      GDP  NRI     PD      Area  Population
         Year
         1962-01-01  6.050000e+11  0.0  19.93  962909.0    191861.0
         1967-01-01  8.620000e+11  0.0  21.16  962909.0    203713.0
         1972-01-01  1.280000e+12  0.0  22.14  962909.0    213220.0
         1977-01-01  2.090000e+12  0.0  23.17  962909.0    223091.0
         1982-01-01  3.340000e+12  0.0  24.30  962909.0    233954.0
```

0.9 Use df_usa:

9.1 Multiply the ‘Area’ column for all countries by 10 (so instead of 1000 ha, the unit becomes 100 ha = 1km^2). Display the dataframe head.

```
In [18]: df_usa["Area"] *= 10
         df_usa.head()
```

```
Out[18]:                      GDP  NRI     PD       Area  Population
         Year
         1962-01-01  6.050000e+11  0.0  19.93  9629090.0    191861.0
         1967-01-01  8.620000e+11  0.0  21.16  9629090.0    203713.0
         1972-01-01  1.280000e+12  0.0  22.14  9629090.0    213220.0
         1977-01-01  2.090000e+12  0.0  23.17  9629090.0    223091.0
         1982-01-01  3.340000e+12  0.0  24.30  9629090.0    233954.0
```

9.2: Create a new column in df_usa called ‘GDP/capita’ and populate it with the calculated GDP per capita. Round the results to two decimal points. Display the dataframe head. GDP per capita = (GDP / Population) * 1000

```
In [19]: # units is 1000 habitants => no 1000 scalar in the formula
         df_usa['GDP/capita'] = (df_usa["GDP"] / df_usa["Population"]).round(2)
         df_usa.head()
```

```
Out[19]:                      GDP  NRI     PD       Area  Population   GDP/capita
         Year
         1962-01-01  6.050000e+11  0.0  19.93  9629090.0    191861.0   3153324.54
         1967-01-01  8.620000e+11  0.0  21.16  9629090.0    203713.0   4231443.26
         1972-01-01  1.280000e+12  0.0  22.14  9629090.0    213220.0   6003189.19
         1977-01-01  2.090000e+12  0.0  23.17  9629090.0    223091.0   9368374.34
         1982-01-01  3.340000e+12  0.0  24.30  9629090.0    233954.0  14276310.73
```

9.3: Find the maximum value of the ‘NRI’ column in the US (using pandas methods). What year does the max value occur? Display the values.

```
In [20]: max_NRI = df_usa.max(axis=0)['NRI']
         max_index = df_usa.idxmax(axis=0)['NRI']

         print("Maximum value of the 'NRI' column: ", max_NRI)
         print("Year of the maximum value of the 'NRI' column: ", str(max_index)[0:4])
```

```
Maximum value of the 'NRI' column:  1020.0
Year of the maximum value of the 'NRI' column:  1992
```