Homework 4 - Pandas

NAME: Abrar Syed

STUDENT ID: 3035529743

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

Pandas Introduction

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 (http://www.fao.org/nr/water/aquastat/data/query/index.html)
- 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]:
```
df = pd.read_csv('data3.csv')
```

1.2) Display the first 10 rows of the dataframe.

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

Out[3]:

	Area	Area Id	Variable Name	Variable Id	Year	Value	Symbol	Other
0	Argentina	9.0	Total area of the country	4100.0	1962.0	278040.0	E	NaN
1	Argentina	9.0	Total area of the country	4100.0	1967.0	278040.0	E	NaN
2	Argentina	9.0	Total area of the country	4100.0	1972.0	278040.0	E	NaN
3	Argentina	9.0	Total area of the country	4100.0	1977.0	278040.0	E	NaN
4	Argentina	9.0	Total area of the country	4100.0	1982.0	278040.0	E	NaN
5	Argentina	9.0	Total area of the country	4100.0	1987.0	278040.0	E	NaN
6	Argentina	9.0	Total area of the country	4100.0	1992.0	278040.0	E	NaN
7	Argentina	9.0	Total area of the country	4100.0	1997.0	278040.0	E	NaN
8	Argentina	9.0	Total area of the country	4100.0	2002.0	278040.0	E	NaN
9	Argentina	9.0	Total area of the country	4100.0	2007.0	278040.0	E	NaN

1.3) Display the column names.

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

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

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

In [5]:
```
df.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

Data Preprocessing

2.1) 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]:
```
df[df['Symbol'].isna()].head(5)
```

Out[6]:

	Area	Area Id	Variable Name	Variable Id	Year	Value	Symbol	Other
390	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
391	E - External data	NaN	NaN	NaN	NaN	NaN	NaN	NaN
392	I - AQUASTAT estimate	NaN	NaN	NaN	NaN	NaN	NaN	NaN
393	K - Aggregate data	NaN	NaN	NaN	NaN	NaN	NaN	NaN
394	L - Modelled data	NaN	NaN	NaN	NaN	NaN	NaN	NaN

2.2) 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'. Store the result in place. Then display the last 5 rows of the dataframe.

In [7]:
```
df.drop('Other', axis=1, inplace=True)
df.dropna(inplace=True)
df.tail(5)
```

Out[7]:

	Area	Area Id	Variable Name	Variable Id	Year	Value	Symbol
385	United States of America	231.0	National Rainfall Index (NRI)	4472.0	1981.0	949.2	E
386	United States of America	231.0	National Rainfall Index (NRI)	4472.0	1984.0	974.6	E
387	United States of America	231.0	National Rainfall Index (NRI)	4472.0	1992.0	1020.0	E
388	United States of America	231.0	National Rainfall Index (NRI)	4472.0	1996.0	1005.0	E
389	United States of America	231.0	National Rainfall Index (NRI)	4472.0	2002.0	938.7	E

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

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

2.4) Display all the unique values in your new dataframe for each of the columns: Area, Variable Name, Year.

In [9]:
```
print('Unique Areas = ', pd.unique(df1['Area']))
print('Unique Variable Names = ', pd.unique(df1['Variable Name']))
print('Unique Years = ', pd.unique(df1['Year']))
```

```
Unique Areas =  ['Argentina' 'Australia' 'Germany' 'Iceland' 'Ireland' 'Swe
den'
 'United States of America']
Unique Variable Names =  ['Total area of the country' 'Total population' 'P
opulation density'
 'Gross Domestic Product (GDP)' 'National Rainfall Index (NRI)']
Unique Years =  [1962. 1967. 1972. 1977. 1982. 1987. 1992. 1997. 2002. 200
7. 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.]
```

2.5) 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 the Year column after conversion.

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

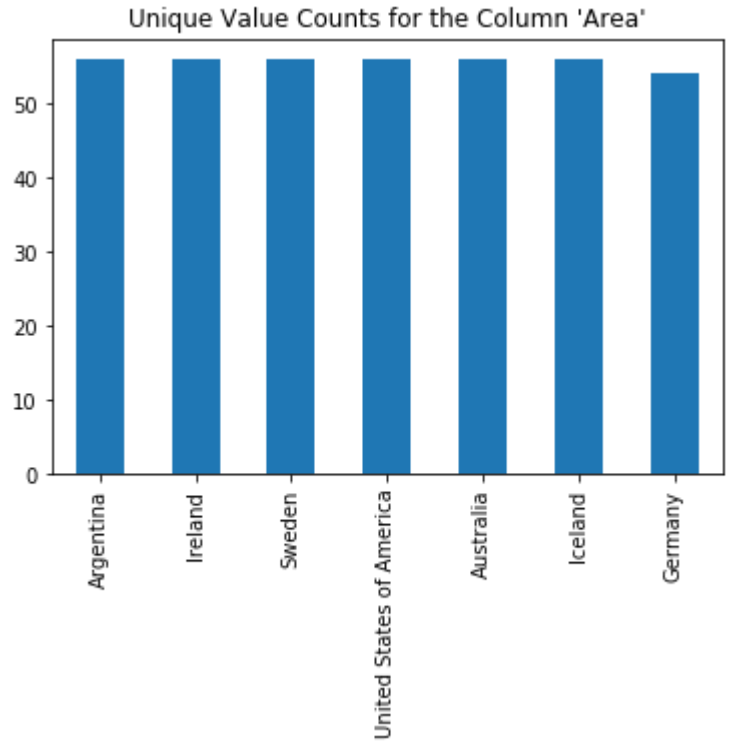
In [10]:
```
df1['Year'] = pd.to_datetime(df1['Year'], format = '%Y')
df1['Year'].head(5)
```

Out[10]:
```
0   1962-01-01
1   1967-01-01
2   1972-01-01
3   1977-01-01
4   1982-01-01
Name: Year, dtype: datetime64[ns]
```

Plot

3.1) Plot a bar graph showing the count for each unique value in the column 'Area'. Give it a title.

In [11]:
```
pd.value_counts(df1['Area']).plot.bar()
plt.title("Unique Value Counts for the Column 'Area'");
```



Extract specific statistics from the preprocessed data:

4.1) 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

	Area	Variable Name	Year	Value
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

4.2) Print the years (with the same format as 4.3) 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]:
```
rainfall = dftemp[(dftemp['Variable Name'] == 'National Rainfall Index (NRI)'
date_lst = pd.DatetimeIndex(rainfall[(rainfall['Value'] > 900) & (rainfall['V
for year in date_lst:
    print(str(year)[:-8])
```

```
1981-01-01
1998-01-01
```

US statistics:

5.1) 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( ), set inplace=True). Display the dataframe head.

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

Out[14]:

Year	Area	Variable Name	Value
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

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

Out[15]:

Year	Gross Domestic Product (GDP)	National Rainfall Index (NRI)	Population density	Total area of the country	Total population
1962-01-01	6.050000e+11	NaN	19.93	962909.0	191861.0
1965-01-01	NaN	928.5	NaN	NaN	NaN
1967-01-01	8.620000e+11	NaN	21.16	962909.0	203713.0
1969-01-01	NaN	952.2	NaN	NaN	NaN
1972-01-01	1.280000e+12	NaN	22.14	962909.0	213220.0

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

In [16]:
```
df_usa.rename(columns = {"Gross Domestic Product (GDP)": "GDP", "National Rai
                         "Population density": "PD", "Total area of the countr
                         "Total population": "Population"}, inplace = True)

df_usa.head()
```

Out[16]:

Variable Name	GDP	NRI	PD	Area	Population
Year					
1962-01-01	6.050000e+11	NaN	19.93	962909.0	191861.0
1965-01-01	NaN	928.5	NaN	NaN	NaN
1967-01-01	8.620000e+11	NaN	21.16	962909.0	203713.0
1969-01-01	NaN	952.2	NaN	NaN	NaN
1972-01-01	1.280000e+12	NaN	22.14	962909.0	213220.0

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

In [17]:
```
df_usa.fillna(0, inplace = True)
df_usa.head()
```

Out[17]:

Variable Name	GDP	NRI	PD	Area	Population
Year					
1962-01-01	6.050000e+11	0.0	19.93	962909.0	191861.0
1965-01-01	0.000000e+00	928.5	0.00	0.0	0.0
1967-01-01	8.620000e+11	0.0	21.16	962909.0	203713.0
1969-01-01	0.000000e+00	952.2	0.00	0.0	0.0
1972-01-01	1.280000e+12	0.0	22.14	962909.0	213220.0

Use df_usa:

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

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

Out[18]:

Variable Name	GDP	NRI	PD	Area	Population
Year					
1962-01-01	6.050000e+11	0.0	19.93	9629090.0	191861.0
1965-01-01	0.000000e+00	928.5	0.00	0.0	0.0
1967-01-01	8.620000e+11	0.0	21.16	9629090.0	203713.0
1969-01-01	0.000000e+00	952.2	0.00	0.0	0.0
1972-01-01	1.280000e+12	0.0	22.14	9629090.0	213220.0

6.2) 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]:
```
df_usa['GDP/capita'] = (df_usa['GDP']/df_usa['Population'])*1000
df_usa['GDP/capita'] = df_usa['GDP/capita'].round(decimals = 2)
df_usa.head()
```

Out[19]:

Variable Name	GDP	NRI	PD	Area	Population	GDP/capita
Year						
1962-01-01	6.050000e+11	0.0	19.93	9629090.0	191861.0	3.153325e+09
1965-01-01	0.000000e+00	928.5	0.00	0.0	0.0	NaN
1967-01-01	8.620000e+11	0.0	21.16	9629090.0	203713.0	4.231443e+09
1969-01-01	0.000000e+00	952.2	0.00	0.0	0.0	NaN
1972-01-01	1.280000e+12	0.0	22.14	9629090.0	213220.0	6.003189e+09

6.3) 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]:
```
desc_nri = df_usa.sort_values(by = ['NRI'], ascending = False)
max_nri = desc_nri['NRI'].values[0]
max_nri_year = str(desc_nri.index.values[0])[0:10]
print('Maximum NRI = ', max_nri)
print('Year of Maximum NRI = ', max_nri_year)
```

```
Maximum NRI =  1020.0
Year of Maximum NRI =  1992-01-01
```

Congratulations on completing hw4! Don't forget to click Kernel -> Restart & Run All, save your file, download or print as pdf, and submit pdf to Gradescope.