Module 4: Cleaning data with pandas

Weekly Assignment 4B

Brandan Owens and Loan Pham

Q.1 Download the dataset "adult_income_data.csv"

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from functools import reduce
from fractions import Fraction
```

In [2]:
```
# (a) read in the dataset with pd.read_csv. Show the first 8 records, what's wrong with
income_data = pd.read_csv("../dataFiles/adult_income_data.csv")
income_data.head(8)
```

Out[2]:

	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	Male	2174	0	40	United-States	<50K
0	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	Male	0	0	13	United-States	<50K
1	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	Male	0	0	40	United-States	<50K
2	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Male	0	0	40	United-States	<50K
3	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Female	0	0	40	Cuba	<50K
4	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	Female	0	0	40	United-States	<50K
5	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Female	0	0	16	Jamaica	<50K
6	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	Male	0	0	45	United-States	>50K
7	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	Female	14084	0	50	United-States	>50K

In [3]:
```
# (b) read the dataset again and bear in mind that the dataset has no header (use heade
income_data = pd.read_csv("../dataFiles/adult_income_data.csv", header = None)
income_data.head()
```

Out[3]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	Male	2174	0	40	United-States	<=
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	Male	0	0	13	United-States	<=
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	Male	0	0	40	United-States	<=
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Male	0	0	40	United-States	<=
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Female	0	0	40	Cuba	<=

In [4]:
```
# (c) name the columns. Refer to the "adult_income_names.txt"
names=[]
with open("../dataFiles/adult_income_names.txt","r") as f:
    for line in f:
        f.readline()
        var = line.split(":")[0]
        names.append(var)
names
income_data.columns = names
income_data.head()
```

Out[4]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	sex	capital-gain
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	Male	2174
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	Male	0
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	Male	0
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Male	0

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	sex	capital-gain
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Female	0

In [5]:
```
# (d) get the statistical summary of the dataset.
income_data.describe()
```

Out[5]:

	age	fnlwgt	education-num	capital-gain	capital-loss	hours-per-week
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	1.897784e+05	10.080679	1077.648844	87.303830	40.437456
std	13.640433	1.055500e+05	2.572720	7385.292085	402.960219	12.347429
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178270e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.370510e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

In [6]:
```
# (e) Create a DataFrame with only age, education, and occupation

income_data_filtered = income_data[['age','education','occupation']]
income_data_filtered.head()
```

Out[6]:

	age	education	occupation
0	39	Bachelors	Adm-clerical
1	50	Bachelors	Exec-managerial
2	38	HS-grad	Handlers-cleaners
3	53	11th	Handlers-cleaners
4	28	Bachelors	Prof-specialty

In [7]:
```
# (g) Find the number of people who are aged between 30 and 50

income_data_filtered = income_data_filtered[income_data['age'].between(30, 50)]
len(income_data_filtered.index) #len(data.index) is fastest counter
```

Out[7]: 16390

In [8]:
```
# (h) Find the number of people who are aged between 30 and 50, and made positive capit
```

```
income_data_filtered = income_data[['age','education','occupation','capital-gain']]
income_data_filtered = income_data_filtered[(income_data['age'] >= 30) & (income_data_f
income_data_filtered.head()
```

Out[8]:

	age	education	occupation	capital-gain
0	39	Bachelors	Adm-clerical	2174
8	31	Masters	Prof-specialty	14084
9	42	Bachelors	Exec-managerial	5178
59	30	HS-grad	Machine-op-inspct	5013
60	30	Bachelors	Sales	2407

In [9]:

```
# (i)  group by "occupation", show the median age of each group.

grouped = income_data.groupby('occupation')['age'].median()
grouped
```

Out[9]:

```
occupation
 ?                  35
 Adm-clerical       35
 Armed-Forces       29
 Craft-repair       38
 Exec-managerial    41
 Farming-fishing    39
 Handlers-cleaners  29
 Machine-op-inspct  36
 Other-service      32
 Priv-house-serv    40
 Prof-specialty     40
 Protective-serv    36
 Sales              35
 Tech-support       36
 Transport-moving   39
Name: age, dtype: int64
```

In [10]:

```
# (j)  use the following codes to create two dataframes
df1 = income_data[['age', 'workclass', 'occupation']].sample(5,random_state=101)
df2 = income_data[['education','occupation']].sample(5,random_state=101)
# merge df1 with df2. The key is  'occupation'
occupation_data = pd.merge(df1,df2,on='occupation')
display(occupation_data)
```

	age	workclass	occupation	education
0	51	Private	Machine-op-inspct	HS-grad
1	19	Private	Sales	11th
2	40	Private	Exec-managerial	HS-grad
3	17	Private	Handlers-cleaners	10th
4	61	Private	Craft-repair	7th-8th

In [11]:
```
#Q.2.a Read in dataset 'Energy Indicators.xls'. Exclude the footer & header info, the f

header_names = ['toDelete','Country Name','Energy Supply','Energy Supply per Capita','%
energy_data = pd.read_excel("../dataFiles/Energy Indicators.xls", skipfooter=246, skipr
energy_data.drop(columns='toDelete', inplace=True)
display(energy_data)
```

Country Name	Energy Supply	Energy Supply per Capita	% Renewable
Afghanistan	321	10	78.669280
Albania	102	35	100.000000
Algeria	1959	51	0.551010
American Samoa	...	...	0.641026
Andorra	9	121	88.695650
Angola	642	27	70.909090
Anguilla	2	136	0.000000
Antigua and Barbuda	8	84	0.000000
Argentina	3378	79	24.064520
Armenia	143	48	28.236060
Aruba	12	120	14.870690
Australia1	5386	231	11.810810
Austria	1391	164	72.452820
Azerbaijan	567	60	6.384345
Bahamas	45	118	0.000000
Bahrain	574	425	0.000000
Bangladesh	1625	10	1.966329
Barbados	19	69	0.000000
Belarus	1142	120	0.463389

In [12]:
```
#Q.2.a.continue Rename the following countries ["Republic of Korea":"South Korea","Unit
replacement_names = {
    "Republic of Korea":"South Korea",
    "Australia1":"Australia",
    "United States of America":"United States",
    "United Kingdom of Great Britain and Northern Ireland":"United Kingdom",
    "China, Hong Kong Special Administrative Region":"Hong Kong",
    "Bolivia (Plurinational State of)":"Bolivia",
    "Switzerland17":"Switzerland"
    }
energy_data["Country Name"].replace(replacement_names, inplace=True)
display(energy_data)
```

Country Name	Energy Supply	Energy Supply per Capita	% Renewable
Afghanistan	321	10	78.669280

Country Name	Energy Supply	Energy Supply per Capita	% Renewable
Albania	102	35	100.000000
Algeria	1959	51	0.551010
American Samoa	...	...	0.641026
Andorra	9	121	88.695650
Angola	642	27	70.909090
Anguilla	2	136	0.000000
Antigua and Barbuda	8	84	0.000000
Argentina	3378	79	24.064520
Armenia	143	48	28.236060
Aruba	12	120	14.870690
Australia	5386	231	11.810810
Austria	1391	164	72.452820
Azerbaijan	567	60	6.384345
Bahamas	45	118	0.000000
Bahrain	574	425	0.000000
Bangladesh	1625	10	1.966329
Barbados	19	69	0.000000
Belarus	1142	120	0.463389

In [13]:

```
#Q.2.b Load the GDP Data from "world_bank.xlsx". Exclude the header, rename the followi
replacement_names = {
    "Korea, Rep.":"South Korea",
    "Iran, Islamic Rep.":"Iran",
    "Hong Kong SAR, China":"Hong Kong"
}
gdp_data = pd.read_excel("../dataFiles/world_bank.xlsx", skiprows=2, header=2, keep_def
gdp_data.rename
gdp_data["Country Name"].replace(replacement_names, inplace=True)
display(gdp_data)
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	
0	Aruba	ABW	GDP (current US$)	NY.GDP.MKTP.CD			
1	Afghanistan	AFG	GDP (current US$)	NY.GDP.MKTP.CD	537777811111111	548888895555556	54666667777
2	Angola	AGO	GDP (current US$)	NY.GDP.MKTP.CD			

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	
3	Albania	ALB	GDP (current US$)	NY.GDP.MKTP.CD			
4	Andorra	AND	GDP (current US$)	NY.GDP.MKTP.CD			
...	...	...	...	...	...	...	
259	Kosovo	XKX	GDP (current US$)	NY.GDP.MKTP.CD			
260	Yemen, Rep.	YEM	GDP (current US$)	NY.GDP.MKTP.CD			
261	South Africa	ZAF	GDP (current US$)	NY.GDP.MKTP.CD	75752484950301	797284054318914	84978300433
262	Zambia	ZMB	GDP (current US$)	NY.GDP.MKTP.CD	713000000	696285714285714	69314285714
263	Zimbabwe	ZWE	GDP (current US$)	NY.GDP.MKTP.CD	1052990400	1096646600	111760

264 rows × 62 columns

In [14]:
```
#Q.2.c Load the dataset scimagojr.xlsx
header_names=["Rank", "Country Name", "Documents", "Citable Documents", "Citations", "S
scimagojr = pd.read_excel("../dataFiles/scimagojr.xlsx", names=header_names)
display(scimagojr)
```

	Rank	Country Name	Documents	Citable Documents	Citations	Self-Citations	Citations per document	H index
0	1	China	147887	147512	856806	583858	5.79	162
1	2	United States	113579	111426	1085684	370574	9.56	259
2	3	Japan	34294	34054	275980	73491	8.05	145
3	4	United Kingdom	24328	23671	278694	52119	11.46	159
4	5	India	21450	21183	179494	54929	8.37	132
...	...	...	...	...	...	...	...	...
187	188	Reunion	1	1	4	1	4.00	1
188	189	Saint Helena	1	1	36	0	36.00	1

	Rank	Country Name	Documents	Citable Documents	Citations	Self-Citations	Citations per document	H index
189	190	American Samoa	1	1	0	0	0.00	0
190	191	Belize	1	1	9	0	9.00	1
191	192	British Indian Ocean Territory	1	1	45	0	45.00	1

192 rows × 8 columns

In [15]:
```
#Q.2.d Join the three datasets, retain only the last 10 years of GDP data and only the

data_final = pd.merge(scimagojr, energy_data, on="Country Name")
data_final = pd.merge(data_final, gdp_data, on="Country Name")
years = list(range(1960,2006,1))
data_final.drop(labels=years,axis=1,inplace=True)
display(data_final)
```

	Rank	Country Name	Documents	Citable Documents	Citations	Self-Citations	Citations per document	H index	Energy Supply	Energy Supply per Capita
0	14	Australia	10616	10496	129788	22759	12.23	123	5386	231
1	38	Austria	2133	2102	19946	2234	9.35	61	1391	164
2	40	Algeria	1645	1626	11257	2286	6.84	46	1959	51
3	43	Argentina	1389	1379	14060	2050	10.12	52	3378	79
4	63	Bangladesh	623	617	2895	309	4.65	26	1625	10
5	70	Azerbaijan	392	385	655	145	1.67	11	567	60
6	72	Belarus	340	334	1297	251	3.81	17	1142	120
7	83	Bahrain	154	150	1279	44	8.31	17	574	425
8	88	Armenia	105	105	648	250	6.17	13	143	48
9	104	Angola	56	56	140	2	2.50	7	642	27
10	120	Albania	27	27	217	14	8.04	7	102	35
11	148	Barbados	7	7	16	3	2.29	3	19	69
12	159	Afghanistan	5	5	5	0	1.00	1	321	10
13	169	Andorra	2	2	15	0	7.50	1	9	121
14	171	Antigua and Barbuda	2	2	0	0	0.00	0	8	84
15	190	American Samoa	1	1	0	0	0.00	0	...	...

16 rows × 26 columns

In [16]:
```
#Q.2.e What is the average GDP over the last 10 years for each country
columns = [2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017]
data_final['Average GDP over 10 years'] = round(data_final.mean(axis=1),1)
display(data_final)
```

	Rank	Country Name	Documents	Citable Documents	Citations	Self-Citations	Citations per document	H index	Energy Supply	Energy Supply per Capita
0	14	Australia	10616	10496	129788	22759	12.23	123	5386	231
1	38	Austria	2133	2102	19946	2234	9.35	61	1391	164
2	40	Algeria	1645	1626	11257	2286	6.84	46	1959	51
3	43	Argentina	1389	1379	14060	2050	10.12	52	3378	79
4	63	Bangladesh	623	617	2895	309	4.65	26	1625	10
5	70	Azerbaijan	392	385	655	145	1.67	11	567	60
6	72	Belarus	340	334	1297	251	3.81	17	1142	120
7	83	Bahrain	154	150	1279	44	8.31	17	574	425
8	88	Armenia	105	105	648	250	6.17	13	143	48
9	104	Angola	56	56	140	2	2.50	7	642	27
10	120	Albania	27	27	217	14	8.04	7	102	35
11	148	Barbados	7	7	16	3	2.29	3	19	69
12	159	Afghanistan	5	5	5	0	1.00	1	321	10
13	169	Andorra	2	2	15	0	7.50	1	9	121
14	171	Antigua and Barbuda	2	2	0	0	0.00	0	8	84
15	190	American Samoa	1	1	0	0	0.00	0	...	...

16 rows × 27 columns

In [17]:
```
#Q.2.f By how much had the GDP changed over the 10-year span for the country with the 6
gdp_difference = str(data_final.iloc[5][2016] - data_final.iloc[5][2006])
name = str(data_final.iloc[5]['Country Name'])
print("Country: " + name + " GDP Difference over 10 years: " + gdp_difference)
```

```
Country: Azerbaijan GDP Difference over 10 years: 168646958121543
```

In [18]:
```
#Q.2.g Create a new column that is the ration of Self-Citations to Total Citations. Wha
#self_citations = data_final["Self-Citations"]
```

```
#total_citations = data_final["Citations"] + self_citations
#zipper = zip(self_citations, total_citations)
#data_final["Citation Ratio"] = [str(Fraction(x,y)).replace('/',':') for x, y in zipper
data_final["Citation Ratio"] = data_final["Self-Citations"]/data_final["Citations"]
data_final.sort_values(by="Citation Ratio", ascending=False, inplace=True)
display(data_final)
```

	Rank	Country Name	Documents	Citable Documents	Citations	Self-Citations	Citations per document	H index	Energy Supply	Energy Supply per Capita
8	88	Armenia	105	105	648	250	6.17	13	143	48
5	70	Azerbaijan	392	385	655	145	1.67	11	567	60
2	40	Algeria	1645	1626	11257	2286	6.84	46	1959	51
6	72	Belarus	340	334	1297	251	3.81	17	1142	120
11	148	Barbados	7	7	16	3	2.29	3	19	69
0	14	Australia	10616	10496	129788	22759	12.23	123	5386	231
3	43	Argentina	1389	1379	14060	2050	10.12	52	3378	79
1	38	Austria	2133	2102	19946	2234	9.35	61	1391	164
4	63	Bangladesh	623	617	2895	309	4.65	26	1625	10
10	120	Albania	27	27	217	14	8.04	7	102	35
7	83	Bahrain	154	150	1279	44	8.31	17	574	425
9	104	Angola	56	56	140	2	2.50	7	642	27
12	159	Afghanistan	5	5	5	0	1.00	1	321	10
13	169	Andorra	2	2	15	0	7.50	1	9	121
14	171	Antigua and Barbuda	2	2	0	0	0.00	0	8	84
15	190	American Samoa	1	1	0	0	0.00	0	...	...

16 rows × 28 columns