pandas example

November 8, 2019

[63]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

[13]:
```
world = {'Team':['west indies','west␣
 ↪indies','India','Australia','Pakistan','Sri␣
 ↪lanka','Australia','Australia','Australia','India','Australia'],
         'Ranks':[7,7,2,1,6,4,1,1,1,2,1],
          'year':[1975,1979,1983,1987,1992,1996,1999,2003,2007,2011,2015]}
```

[14]:
```
world
```

[14]:
```
{'Team': ['west indies',
  'west indies',
  'India',
  'Australia',
  'Pakistan',
  'Sri lanka',
  'Australia',
  'Australia',
  'Australia',
  'India',
  'Australia'],
 'Ranks': [7, 7, 2, 1, 6, 4, 1, 1, 1, 2, 1],
 'year': [1975, 1979, 1983, 1987, 1992, 1996, 1999, 2003, 2007, 2011, 2015]}
```

[15]:
```
df = pd.DataFrame(world)
df
```

[15]:

	Team	Ranks	year
0	west indies	7	1975
1	west indies	7	1979
2	India	2	1983
3	Australia	1	1987
4	Pakistan	6	1992
5	Sri lanka	4	1996
6	Australia	1	1999
7	Australia	1	2003
8	Australia	1	2007

```
9          India      2  2011
10     Australia      1  2015
```

[16]:
```
b = df.groupby('Team').groups
b
```

[16]:
```
{'Australia': Int64Index([3, 6, 7, 8, 10], dtype='int64'),
 'India': Int64Index([2, 9], dtype='int64'),
 'Pakistan': Int64Index([4], dtype='int64'),
 'Sri lanka': Int64Index([5], dtype='int64'),
 'west indies': Int64Index([0, 1], dtype='int64')}
```

[17]:
```
c = df.groupby('Team').mean()
c
```

[17]:
```
              Ranks    year
Team
Australia       1.0  2002.2
India           2.0  1997.0
Pakistan        6.0  1992.0
Sri lanka       4.0  1996.0
west indies     7.0  1977.0
```

[18]:
```
print(df.groupby(['Team', 'Ranks']).groups)
```

```
{('Australia', 1): Int64Index([3, 6, 7, 8, 10], dtype='int64'), ('India', 2):
Int64Index([2, 9], dtype='int64'), ('Pakistan', 6): Int64Index([4],
dtype='int64'), ('Sri lanka', 4): Int64Index([5], dtype='int64'), ('west
indies', 7): Int64Index([0, 1], dtype='int64')}
```

[19]:
```
df.groupby('Team').groups
```

[19]:
```
{'Australia': Int64Index([3, 6, 7, 8, 10], dtype='int64'),
 'India': Int64Index([2, 9], dtype='int64'),
 'Pakistan': Int64Index([4], dtype='int64'),
 'Sri lanka': Int64Index([5], dtype='int64'),
 'west indies': Int64Index([0, 1], dtype='int64')}
```

[20]:
```
df.groupby('Team').count()
```

[20]:
```
              Ranks  year
Team
Australia         5     5
India             2     2
Pakistan          1     1
Sri lanka         1     1
west indies       2     2
```

[ ]:
```
# Grouping for India

Team1 = df.groupby('Team')
Team1
```

```
print(Team1.get_group('India'))
```

[ ]:
```
##loc method to retrieve the particular column
df.loc[(df['Team'] =='India')]
```

[ ]:
```
d = { 'odd' :np.arange(1,100,2),
      'even': np.arange(0,100,2)}
d
```

[ ]:
```
d['odd']
```

[ ]:
```
d['even']
```

[ ]:
```
d.keys()
```

[ ]:
```
d.values()
```

[ ]:
```
df1 = pd.DataFrame(d)
df1.head(5)
```

[ ]:
```
print(df1.groupby('odd').groups)
```

[ ]:
```
df = pd.DataFrame(np.random.randn(5,4), columns=['col1','col2','col3','col4'])
df
```

[ ]:
```
# Concantation


chockers = {
    'Team' : ['South Africa','New Zealand','Zimbabwe'],
    'Rank' : [1,5,9],
    'points':[895,764,656]
}
```

[ ]:
```
worldcric = {'Team':['west indies','west␣
 ↪indies','India','Australia','Pakistan'],
         'Rank':[7,7,2,1,6],
         'year':[1975,1979,1983,1987,1992],
         'points':[895,764,656,673,844]}
```

[ ]:
```
df11 = pd.DataFrame(chockers)
df22 = pd.DataFrame(worldcric)
```

[ ]:
```
df11
```

[ ]:
```
df22
```

[ ]:
```
print(pd.concat([df11,df22]))
```

[ ]:
```
# Merging all data in the data frames
pd.merge(df11, df22, on= "Team", how= "outer")
```

[ ]:
```
# Includes all data in the data frame on left
pd.merge(df11, df22, on= "Team", how= "left")
```

[ ]:
```
 # Includes all data in the data frame on  right
pd.merge(df11, df22, on= "Team", how= "right")
```

[ ]:
```
 pd.merge(df11, df22, on= "Team", how= "inner")
```

[ ]:
```
left = pd.DataFrame({'Key':['k0','K1','K2','K3'],
                     'A'  :['A0','A1','A2','A3'],
                     'B'   :['B0','B1','B2','B3']})
left
```

[ ]:
```
right = pd.DataFrame({'Key':['k0','K1','K2','K3'],
                      'C':['C0','C1','C2','C3'],
                      'D': ['D0','D1','D2','D3',]})
right
```

[ ]:
```
print(pd.concat([left,right], axis=1))
```

[ ]:
```
print(pd.concat([left,right],axis=0))
```

[83]:
```
country = pd.read_csv(r'Desktop/Countries.csv')
country.head(5)
```

[83]:

	Country	Region	Population	\
0	Afghanistan	ASIA (EX. NEAR EAST)	31056997	
1	Albania	EASTERN EUROPE	3581655	
2	Algeria	NORTHERN AFRICA	32930091	
3	American Samoa	OCEANIA	57794	
4	Andorra	WESTERN EUROPE	71201	

	Area (sq. mi.)	Pop. Density (per sq. mi.)	Coastline (coast/area ratio)	\
0	647500	48,0	0,00	
1	28748	124,6	1,26	
2	2381740	13,8	0,04	
3	199	290,4	58,29	
4	468	152,1	0,00	

	Net migration	Infant mortality (per 1000 births)	GDP ($ per capita)	\
0	23,06	163,07	700.0	
1	-4,93	21,52	4500.0	
2	-0,39	31	6000.0	
3	-20,71	9,27	8000.0	
4	6,6	4,05	19000.0	

	Literacy (%)	Phones (per 1000)	Arable (%)	Crops (%)	Other (%)	Climate	\
0	36,0	3,2	12,13	0,22	87,65	1	
1	86,5	71,2	21,09	4,42	74,49	3	
2	70,0	78,1	3,22	0,25	96,53	1	
3	97,0	259,5	10	15	75	2	
4	100,0	497,2	2,22	0	97,78	3	

```
  Birthrate Deathrate Agriculture Industry Service
0      46,6     20,34        0,38     0,24    0,38
1     15,11      5,22       0,232    0,188   0,579
2     17,14      4,61       0,101      0,6   0,298
3     22,46      3,27         NaN      NaN     NaN
4      8,71      6,25         NaN      NaN     NaN
```

[ ]:
```
country.shape
```

[ ]:
```
country.dtypes
```

[ ]:
```
country.columns
```

[ ]:
```
country.head(5)
```

[ ]:
```
# Describing the columns with dtype as int and float

country.describe()
```

[ ]:
```
a =country.dtypes[country.dtypes == 'object'].index
country[a].head(5)
```

[ ]:
```
# Describing the objects

country[a].describe()
```

[ ]:
```
country['Population'].head(5)
```

[ ]:
```
# Displaying all values Whose population are greater than 500000
df1 =country.loc[country['Population']>500000]
df1.head(5)
```

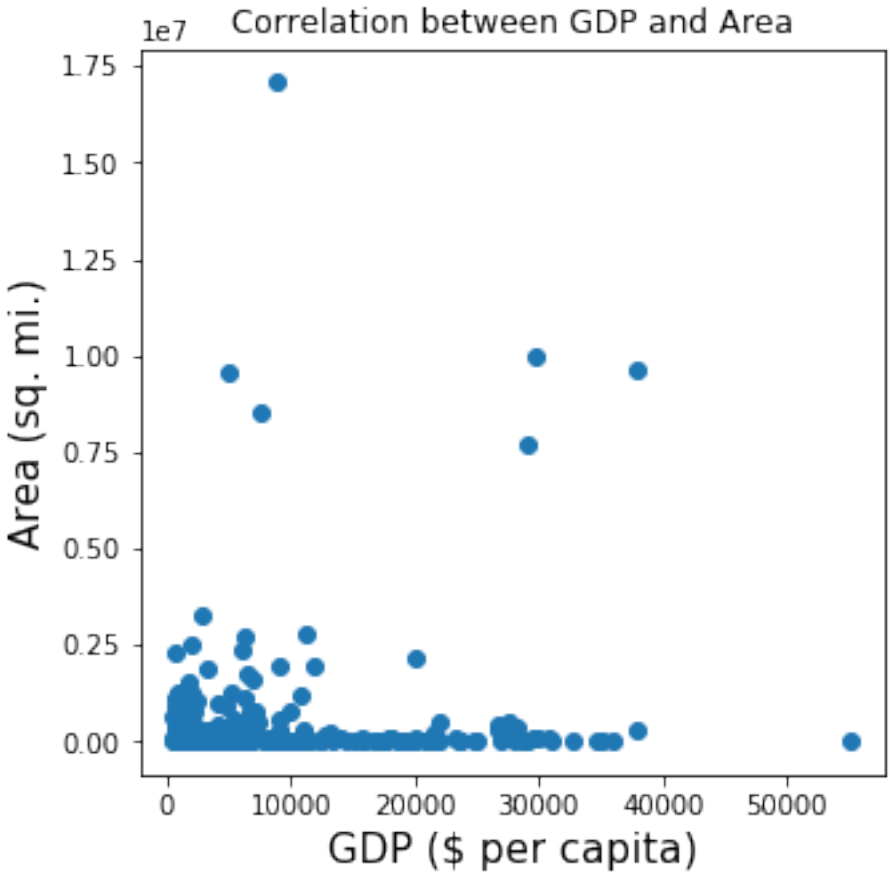
[ ]:
```
df1.shape
```

[ ]:
```
# Filtering Countries, whose population is >500000

df2=country.loc[country['Population']>500000,['Country','Population']]

df2.head(10)
```

[ ]:
```
country.isnull()
```

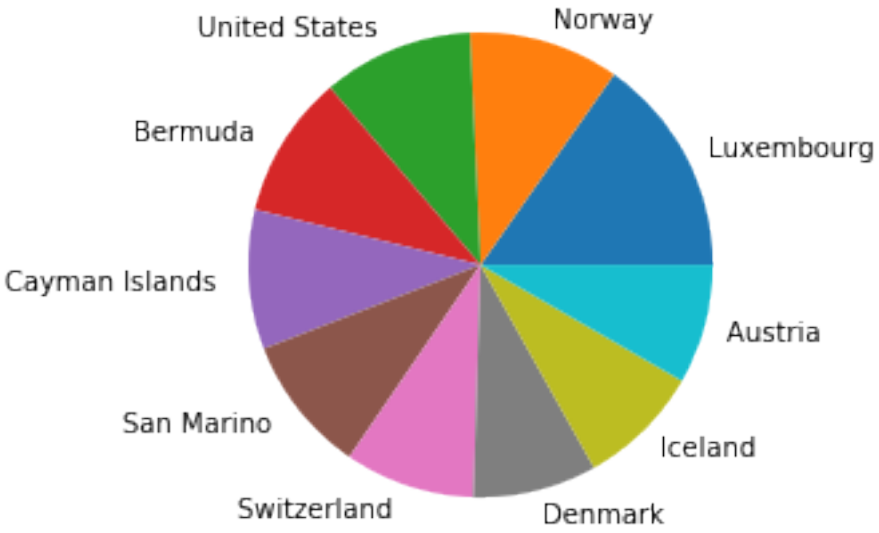
[37]:
```
# Correlation between GDP and Area
import matplotlib.pyplot as plt


data = country.loc[:,['Country','GDP ($ per capita)','Area (sq. mi.)']]
plt.figure(figsize=(5,5))
x = np.array(data['GDP ($ per capita)'])
y = np.array(data['Area (sq. mi.)'])
plt.title("Correlation between GDP and Area");
plt.xlabel('GDP ($ per capita)', fontsize=15)
plt.ylabel('Area (sq. mi.)', fontsize=15)
```

```
plt.scatter(x,y)

plt.show()
```



[118]:
```
# 10 Richest country in the world

df3=country.sort_values('GDP ($ per capita)',ascending = False)
df4=df3.loc[:,['Country','GDP ($ per capita)']]
df5 = df4.head(10)
```

[119]:
```
plt.pie(df5['GDP ($ per capita)'], labels=df5['Country'])
plt.figure(figsize=(500,500))
plt.show()
```



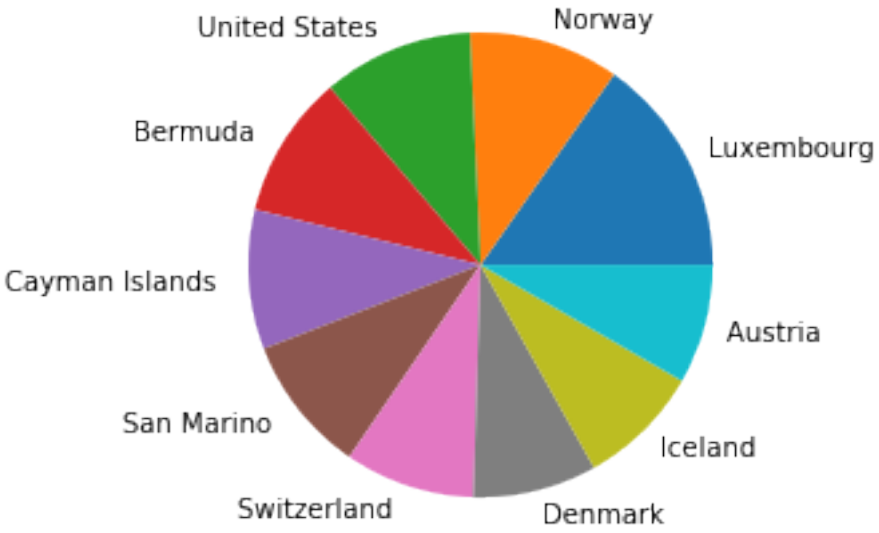

```
<Figure size 36000x36000 with 0 Axes>
```

[109]:
```
# Another way

select = country.loc[:,['GDP ($ per capita)','Country']]
sort1 = select.sort_values('GDP ($ per capita)', ascending = False)
top10 = sort1.iloc[:10]
top10
```

[109]:

	GDP ($ per capita)	Country
121	55100.0	Luxembourg
154	37800.0	Norway
214	37800.0	United States
22	36000.0	Bermuda
38	35000.0	Cayman Islands
177	34600.0	San Marino
196	32700.0	Switzerland
54	31100.0	Denmark
93	30900.0	Iceland
12	30000.0	Austria

[110]:
```
plt.pie(top10['GDP ($ per capita)'], labels=top10['Country'])
plt.figure(figsize=(500,500))
plt.show()
```




```
<Figure size 36000x36000 with 0 Axes>
```

[ ]: