infectious_disease

April 1, 2019

Infectious Diseases Data Analysis
Cleaning Process
Following are the steps we followed for data analysis

1. Import the libraries

```
In [1]: #Import the Libraries
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt

        %matplotlib inline

        import seaborn as sns
        import warnings
        warnings.filterwarnings('ignore')
```

2.Load the dataset.

```
In [2]: dataset=pd.read_csv('data/data.csv')

In [3]: dataset.head(3)
```

Out[3]:

	Indicator Category	Indicator	Year	Sex	Race/Ethnicity	Value	Place
0	Behavioral Health/Substance Abuse	Opioid-Related Unintentional Drug Overdose Mor...	2010	Both	All	1.7	Washington, DC
1	Behavioral Health/Substance Abuse	Opioid-Related Unintentional Drug Overdose Mor...	2010	Both	All	2.2	Fort Worth (Tarrant County), TX
2	Behavioral Health/Substance Abuse	Opioid-Related Unintentional Drug Overdose Mor...	2010	Both	All	2.3	Oakland (Alameda County), CA

```
                           BCHC Requested Methodology  \
0  Age-Adjusted rate of opioid-related mortality ...
1  Age-adjusted rate of opioid-related mortality ...
2  Age-adjusted rate of opioid-related mortality ...

                                              Source  \
0  D.C. Department of Health, Center for Policy, ...
1              National Center for Health Statistics
2                                         CDC Wonder

                                             Methods  \
0                                                NaN
1                                                NaN
2  Age-adjusted rate of opioid-related mortality ...

                                               Notes  \
0  This indicator is not exclusive of other drugs...
1  This indicator is not exclusive of other drugs...
2  Data is for Alameda County. This indicator is ...

   90% Confidence Level - Low  90% Confidence Level - High  \
0                         NaN                          NaN
1                         NaN                          NaN
2                         NaN                          NaN

   95% Confidence Level - Low  95% Confidence Level - High
0                         NaN                          NaN
1                         1.5                          3.0
2                         1.6                          3.2
```

Above we saw the column names and we might need to fix the spaces in the column names. In order to change that we need to first know what are the actual names of the columns.

We do that using the pandas function columns to list all the columns

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

```
Out[4]: Index(['Indicator Category', 'Indicator', 'Year', 'Sex', 'Race/Ethnicity',
               'Value', 'Place', 'BCHC Requested Methodology', 'Source', 'Methods',
               'Notes', '90% Confidence Level - Low', '90% Confidence Level - High',
               '95% Confidence Level - Low', '95% Confidence Level - High'],
              dtype='object')
```

Now we rename the columns

```
In [5]: dataset.rename(columns={'Indicator Category':'indicator_category','Indicator':'indicato
               'Value':'value', 'Place':'place', 'BCHC Requested Methodology':'bchc_req_meth',
               'Notes':'notes', '90% Confidence Level - Low':'90pc_con_lvl-low', '90% Confidenc
               '95% Confidence Level - Low':'95pc_con_lvl-low','95% Confidence Level - High':'9
```

3.Now we need to filter the data according to the indicator category. We use one of the values "Cancer".

```
In [6]: infectious_ds = dataset.loc[dataset["indicator_category"] == "Cancer"]
```

4.And then we remove empty columns and unnecessary columns

```
In [7]: infectious_ds.drop(['indicator_category','bchc_req_meth','source','methods','notes','90
                           axis = 1, inplace= True)
```

5. Now we remove all the rows which has NaN or NA values

```
In [175]: infectious_ds.dropna(axis=0, how='any',inplace= True)

In [8]: infectious_ds.to_csv("data/infectious_diseases.csv")

In [9]: infectious_ds.head(3)
```

Out[9]:

	indicator	year	sex	race_ethnicity	value	place	95pc_con_lvl-low	95pc_con_lvl-high
1465	All Types of Cancer Mortality Rate (Age-Adjust...	2010	Both	All	88.5	Los Angeles, CA	NaN	NaN
1466	All Types of Cancer Mortality Rate (Age-Adjust...	2010	Both	All	98.0	Phoenix, AZ	NaN	NaN
1467	All Types of Cancer Mortality Rate (Age-Adjust...	2010	Both	All	140.1	Miami (Miami-Dade County), FL	NaN	NaN

Analysis

First we'll see how many patients have been reported for cancer in respective years from 2010 to 2016.

Following is the process to do the same

```
In [10]: c_year_2010=infectious_ds[infectious_ds['year']==2010]
         c_year_2010_count=c_year_2010['year'].count()

In [11]: c_year_2010.shape

Out[11]: (414, 8)

In [12]: c_year_2011=infectious_ds[infectious_ds['year']==2011]
         c_year_2011_count=c_year_2011['year'].count()

         c_year_2012=infectious_ds[infectious_ds['year']==2012]
         c_year_2012_count=c_year_2012['year'].count()
```

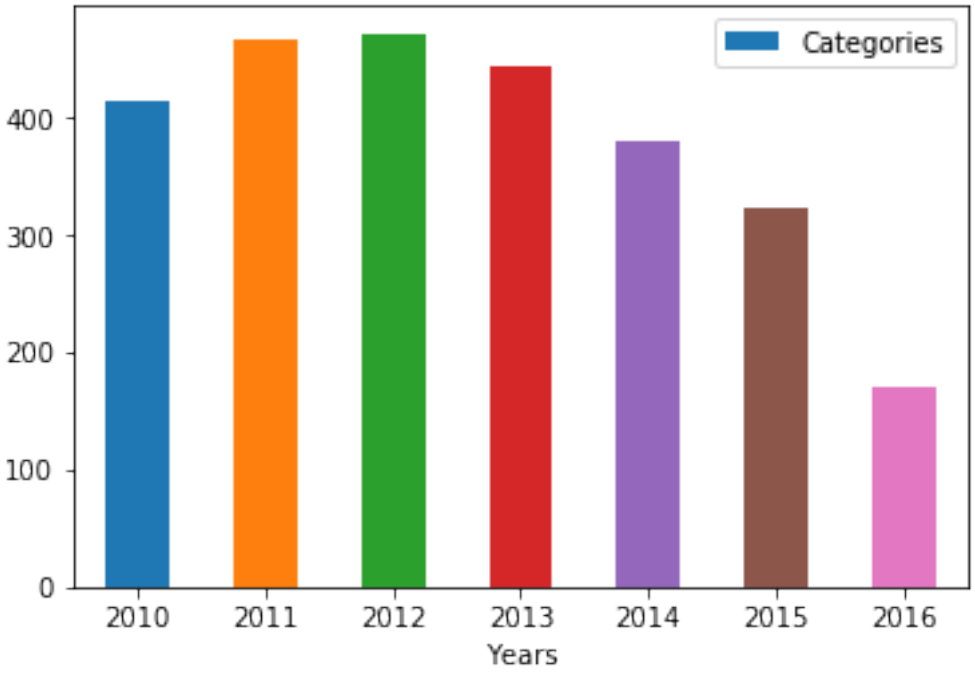
```
c_year_2013=infectious_ds[infectious_ds['year']==2013]
c_year_2013_count=c_year_2013['year'].count()

c_year_2014=infectious_ds[infectious_ds['year']==2014]
c_year_2014_count=c_year_2014['year'].count()

c_year_2015=infectious_ds[infectious_ds['year']==2015]
c_year_2015_count=c_year_2015['year'].count()

c_year_2016=infectious_ds[infectious_ds['year']==2016]
c_year_2016_count=c_year_2016['year'].count()
```

```
In [13]: fig1 = pd.DataFrame({'Years':['2010', '2011', '2012','2013','2014','2015','2016'], 'Ca
         ax = fig1.plot.bar(x='Years', y='Categories', rot=0)
```



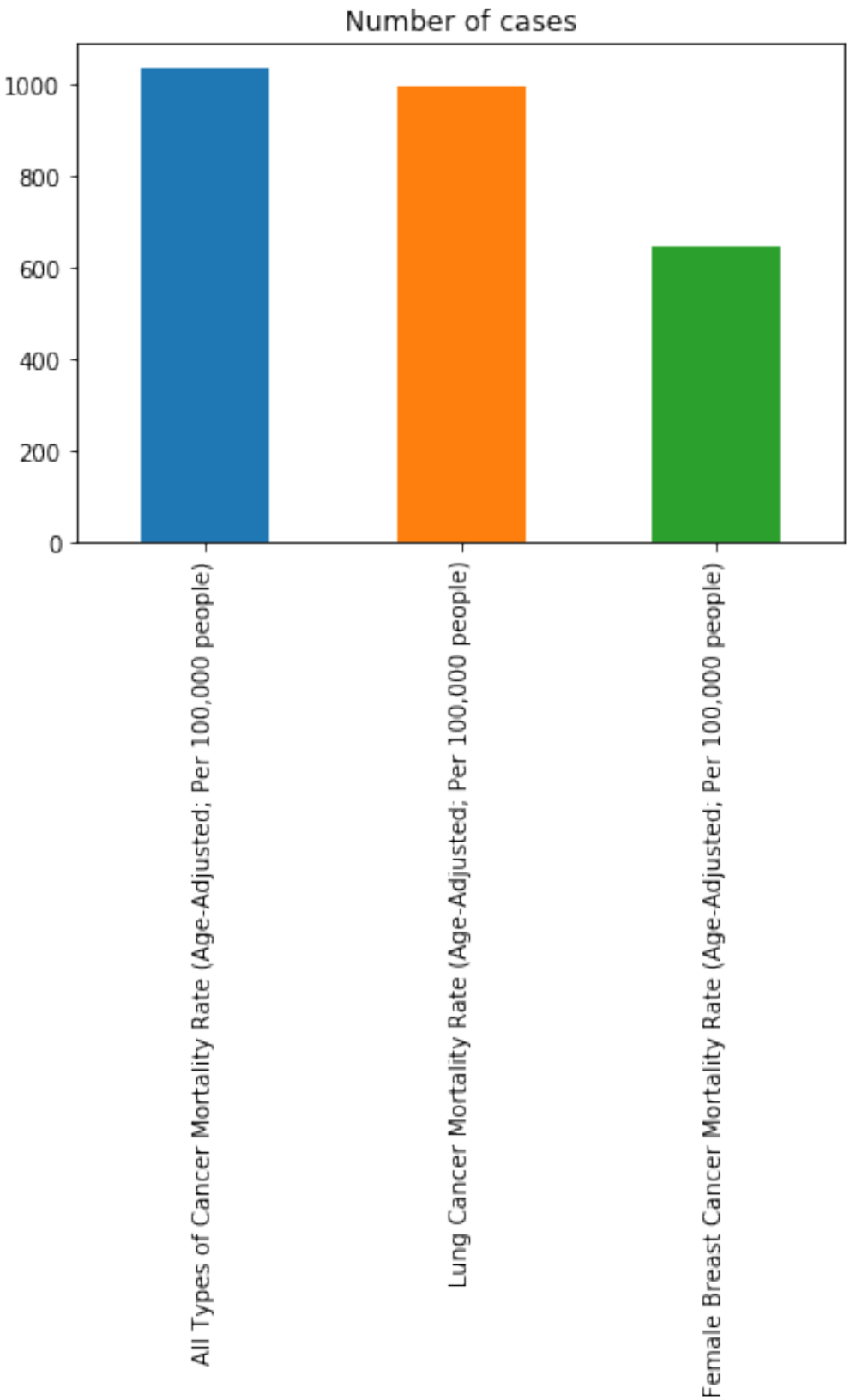
Now we calculate the number of cases for each type of cancer. In order to that we will group according to the indicator and take the count.

```
In [14]: sorted_infection = infectious_ds['indicator'].value_counts()
         sorted_infection
```

```
Out[14]: All Types of Cancer Mortality Rate (Age-Adjusted; Per 100,000 people)     1036
         Lung Cancer Mortality Rate (Age-Adjusted; Per 100,000 people)              992
         Female Breast Cancer Mortality Rate (Age-Adjusted; Per 100,000 people)     642
         Name: indicator, dtype: int64
```

And we plot a histogram to see.

```
In [15]: labels=list(infectious_ds.columns)
         sorted_infection = infectious_ds['indicator'].value_counts().plot(title='Number of cas
         plt.show()
         #label=list(group.columns)
```


Number of cases
1000
800
600
400
200
0
All Types of Cancer Mortality Rate (Age-Adjusted; Per 100,000 people)
Lung Cancer Mortality Rate (Age-Adjusted; Per 100,000 people)
Female Breast Cancer Mortality Rate (Age-Adjusted; Per 100,000 people)

Now we find out the distribution of cancer patients with respect to the race and ethnicity.

In [16]:
```
all=infectious_ds[infectious_ds['race_ethnicity']=="All"]
all_count=all.race_ethnicity.count()

asian=infectious_ds[infectious_ds['race_ethnicity']=="Asian/PI"]
asian_count=asian.race_ethnicity.count()

black=infectious_ds[infectious_ds['race_ethnicity']=="Black"]
black_count=black.race_ethnicity.count()

hispanic=infectious_ds[infectious_ds['race_ethnicity']=="Hispanic"]
hispanic_count=hispanic.race_ethnicity.count()

other=infectious_ds[infectious_ds['race_ethnicity']=="Other"]
other_count=other.race_ethnicity.count()

white=infectious_ds[infectious_ds['race_ethnicity']=="White"]
white_count=white.race_ethnicity.count()
```

In [17]:
```
fig2 = pd.DataFrame({'race_ethnicity':['All', 'Asian/PI', 'Black','Hispanic','Other',
ax = fig2.plot.bar(x='race_ethnicity', y='Count', rot=0)
```