TRAINING PROJECT IN DATA SCIENCE

PROJECT_NAME - VISUALIZATION THE DATA

CSV_FILE - AIR POLLUTION IN THE WORLD

SUBMITTED BY- KRISHNA KUMAR SINGH

SUBMITTED TO - GURPREET MA'AM

REGISTRATION NO - 11701744

In [22]:

```
import numpy as np
import pandas as pd
```

In [73]:

```
df = pd.read_csv(r"C:\Users\Krishna Singh\Desktop\training materials\PROJECT\Air.csv")
df.head(15)
```

Out[73]:

	MeasureId	MeasureId.1	MeasureType	StratificationLevel	StateFips	StateName	CountyFips	CountyName	ReportYear	Value
0	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1027	Clay	1999	33.0
1	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1051	Elmore	1999	5.0
2	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1073	Jefferson	1999	39.0
3	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1079	Lawrence	1999	28.0
4	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1089	Madison	1999	31.0
5	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1097	Mobile	1999	32.0
6	83	Number of days with maximum 8-hour	Counts	State x County	1	Alabama	1101	Montgomery	1999	15.0

	MeasureId	MeasureId.1	MeasureType	StratificationLevel	StateFips	StateName	CountyFips	CountyName	ReportYear	Value
		average ozo...								
7	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1117	Shelby	1999	45.0
8	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1119	Sumter	1999	3.0
9	84	Number of person-days with maximum 8-hour aver...	Counts	State x County	1	Alabama	1027	Clay	1999	467742.0
10	84	Number of person-days with maximum 8-hour aver...	Counts	State x County	1	Alabama	1051	Elmore	1999	323340.0
11	84	Number of person-days with maximum 8-hour aver...	Counts	State x County	1	Alabama	1073	Jefferson	1999	25850955.0
12	84	Number of person-days with maximum 8-hour aver...	Counts	State x County	1	Alabama	1079	Lawrence	1999	967036.0
13	84	Number of person-days with maximum 8-hour aver...	Counts	State x County	1	Alabama	1089	Madison	1999	8515483.0
14	84	Number of person-days with maximum 8-hour aver...	Counts	State x County	1	Alabama	1097	Mobile	1999	12778336.0

In [74]:

```
df.tail(15)
```

Out[74]:

	MeasureId	MeasureId.1	MeasureType	StratificationLevel	StateFips	StateName	CountyFips	CountyName	ReportYear	
218620	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9013	Tolland	2008	10.53
218621	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9013	Tolland	2009	9.01
218622	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9013	Tolland	2010	8.42
218623	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9013	Tolland	2011	9.61
218624	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2001	12.88
		Annual								

	MeasureId	MeasureId.1	MeasureType	StratificationLevel	StateFips	StateName	CountyFips	CountyName	ReportYear	
218625	296	average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2002	11.98
218626	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2003	11.80
218627	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2004	11.18
218628	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2005	11.62
218629	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2006	10.49
218630	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2007	10.79
218631	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2008	10.08
218632	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2009	8.73
218633	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2010	8.08
218634	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2011	8.91

In [24]:

```
print("Information of total number of non-empty columns")
print("------------------------------------------------")
print(df.info(null_counts=True))
```

```
Information of total number of non-empty columns
------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 218635 entries, 0 to 218634
Data columns (total 14 columns):
MeasureId              218635 non-null int64
MeasureId.1            218635 non-null object
MeasureType            218635 non-null object
StratificationLevel    218635 non-null object
StateFips              218635 non-null int64
StateName              218635 non-null object
CountyFips             218635 non-null int64
CountyName             218635 non-null object
ReportYear             218635 non-null int64
Value                  218635 non-null float64
Unit                   218635 non-null object
UnitName               218635 non-null object
DataOrigin             218635 non-null object
MonitorOnly            218635 non-null int64
dtypes: float64(1), int64(5), object(8)
memory usage: 23.4+ MB
None
```

In [25]:

```
df.tail()
```

Out[25]:

	MeasureId	MeasureId.1	MeasureType	StratificationLevel	StateFips	StateName	CountyFips	CountyName	ReportYear	
218630	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2007	10.79
218631	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2008	10.08
218632	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2009	8.73
218633	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2010	8.08
218634	296	Annual average ambient concentrations of PM 2....	Average	State x County	9	Connecticut	9015	Windham	2011	8.91

In [26]:

```
print("Information of total number of non-empty columns")
print("------------------------------------------------")
print(df.info(null_counts=True))
```

```
Information of total number of non-empty columns
------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 218635 entries, 0 to 218634
Data columns (total 14 columns):
MeasureId              218635 non-null int64
MeasureId.1            218635 non-null object
MeasureType            218635 non-null object
StratificationLevel    218635 non-null object
StateFips              218635 non-null int64
StateName              218635 non-null object
CountyFips             218635 non-null int64
CountyName             218635 non-null object
ReportYear             218635 non-null int64
Value                  218635 non-null float64
Unit                   218635 non-null object
UnitName               218635 non-null object
DataOrigin             218635 non-null object
MonitorOnly            218635 non-null int64
dtypes: float64(1), int64(5), object(8)
memory usage: 23.4+ MB
None
```

In [27]:

```
print("Columns and their datatypes")
df.dtypes
# cleaning data
```

```
Columns and their datatypes
```

Out[27]:

```
MeasureId               int64
MeasureId.1            object
MeasureType            object
StratificationLevel    object
StateFips               int64
```

```
StateName               object
CountyFips               int64
CountyName              object
ReportYear               int64
Value                  float64
Unit                    object
UnitName                object
DataOrigin              object
MonitorOnly              int64
dtype: object
```

In [28]:

```
print("Columns and their datatypes")
df.dtypes
```

```
Columns and their datatypes
```

Out[28]:

```
MeasureId               int64
MeasureId.1            object
MeasureType            object
StratificationLevel    object
StateFips               int64
StateName              object
CountyFips              int64
CountyName             object
ReportYear              int64
Value                  float64
Unit                    object
UnitName                object
DataOrigin              object
MonitorOnly              int64
dtype: object
```

In [29]:

```
print("Frequency count of missing values")
df.apply(lambda X:sum(X.isnull()))
```

```
Frequency count of missing values
```

Out[29]:

```
MeasureId              0
MeasureId.1            0
MeasureType            0
StratificationLevel    0
StateFips              0
StateName              0
CountyFips             0
CountyName             0
ReportYear             0
Value                  0
Unit                   0
UnitName               0
DataOrigin             0
MonitorOnly            0
dtype: int64
```

In [30]:

```
%matplotlib inline
import matplotlib.pyplot as plt
from pandas import DataFrame as show
import seaborn as sns
```

In [31]:

```
plt.figure(figsize=(10,5)) #plt is the object of matplot lib and .figure() is used to show or change pr
operties of graphs sns.heatmap(df.isnull(),cmap='brg',yticklabels=False,cbar=False)#heatmaps are matrix
plots which can visualize data in 2D plt.show()
```

Out[31]:

```
<Figure size 720x360 with 0 Axes>
```

```
<Figure size 720x360 with 0 Axes>
```

In [32]:

```
df['MeasureType'].head(5)
```

Out[32]:

```
0    Counts
1    Counts
2    Counts
3    Counts
4    Counts
Name: MeasureType, dtype: object
```

In [33]:

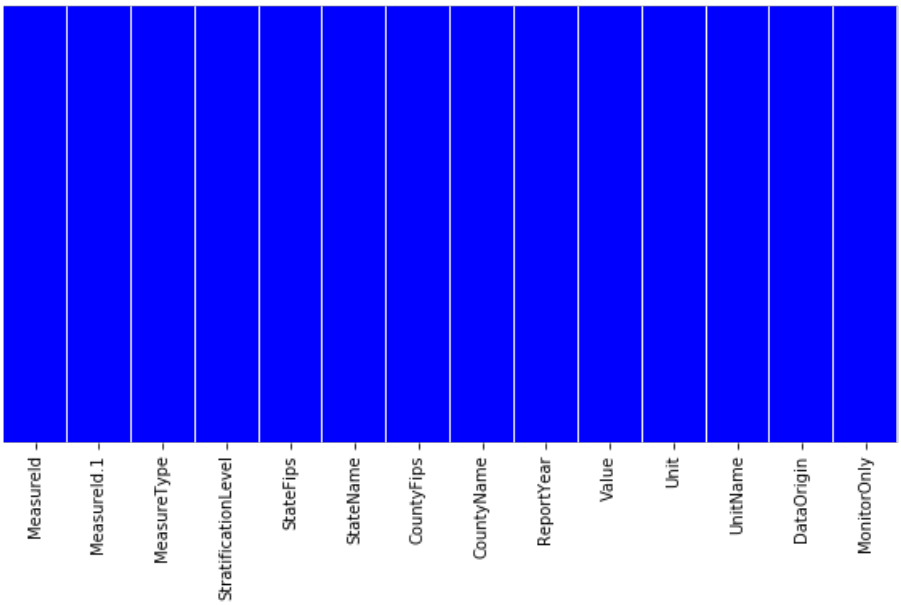
```
df['MeasureId'].head(4)
```

Out[33]:

```
0    83
1    83
2    83
3    83
Name: MeasureId, dtype: int64
```

In [48]:

```
plt.figure(figsize=(10,5))
sns.heatmap(df.isnull(),cmap='brg',yticklabels=False,cbar=False)
plt.show()
import matplotlib
matplotlib.rc('figure', figsize=[20,10])
```



In [35]:

```
df.head(10)
```

Out[35]:

	MeasureId	MeasureId.1	MeasureType	StratificationLevel	StateFips	StateName	CountyFips	CountyName	ReportYear	Value	U
0	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1027	Clay	1999	33.0	Ur
1	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1051	Elmore	1999	5.0	Ur
		Number of days with									

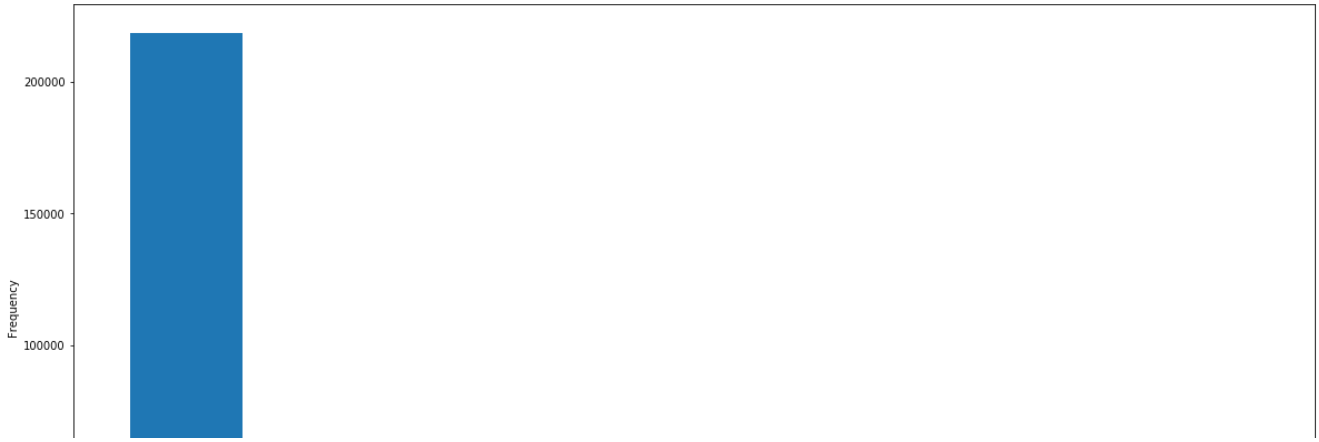
	MeasureId	MeasureName	MeasureType	StratificationLevel	StateFips	StateName	CountyFips	CountyName	ReportYear	Value	U
2	83	maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1073	Jefferson	1999	39.0	U
3	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1079	Lawrence	1999	28.0	Ur
4	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1089	Madison	1999	31.0	Ur
5	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1097	Mobile	1999	32.0	Ur
6	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1101	Montgomery	1999	15.0	Ur
7	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1117	Shelby	1999	45.0	Ur
8	83	Number of days with maximum 8-hour average ozo...	Counts	State x County	1	Alabama	1119	Sumter	1999	3.0	Ur
9	84	Number of person-days with maximum 8-hour aver...	Counts	State x County	1	Alabama	1027	Clay	1999	467742.0	Ur

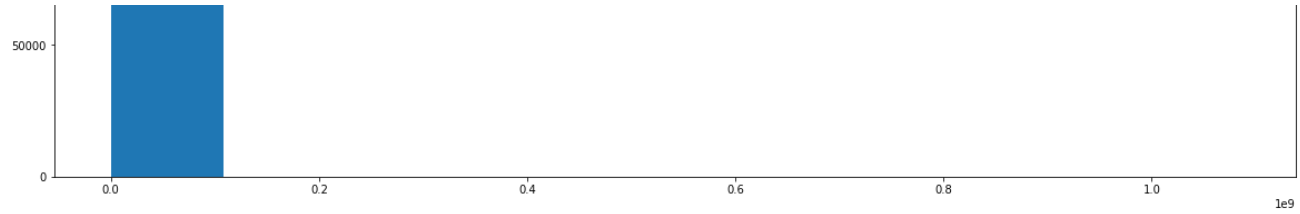
In [36]:

```
import numpy as np
import pandas as pd
%matplotlib inline
import matplotlib.pyplot as plt
from pandas import DataFrame as show
import seaborn as sns
```

In [47]:

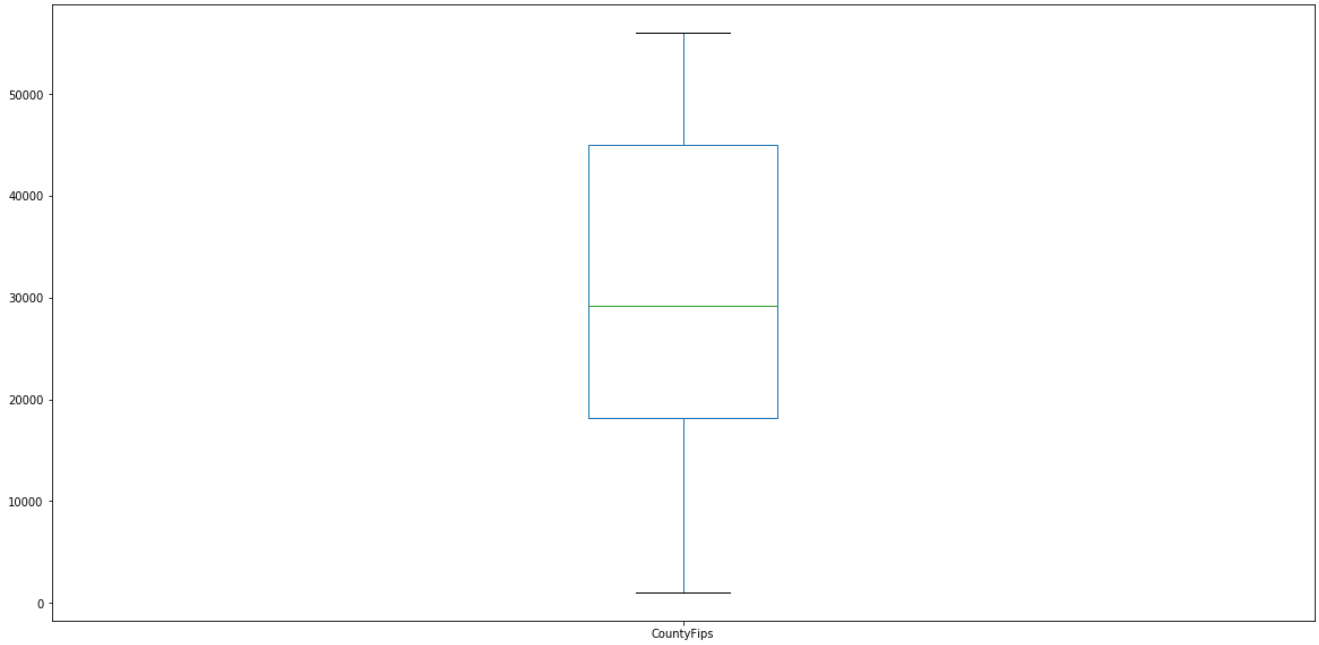
```
df = pd.read_csv(r"C:\Users\Krishna Singh\Desktop\training materials\PROJECT\Air.csv")
df['Value'].plot.hist()
import matplotlib
matplotlib.rc('figure', figsize=[20,10])
```





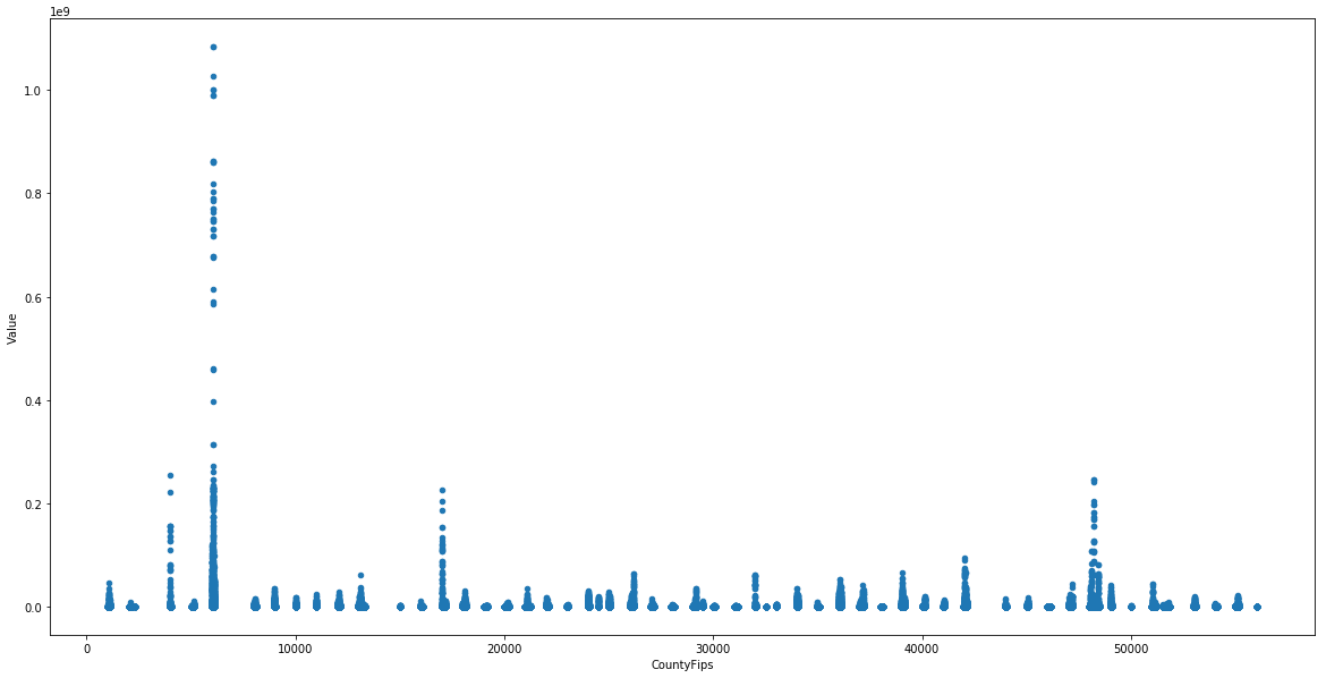
In [38]:

```
df['CountyFips'].plot.box()
import matplotlib
matplotlib.rc('figure', figsize=[20,10])
```



In [39]:

```
df.plot.scatter('CountyFips','Value')
import matplotlib
matplotlib.rc('figure', figsize=[20,10])
```



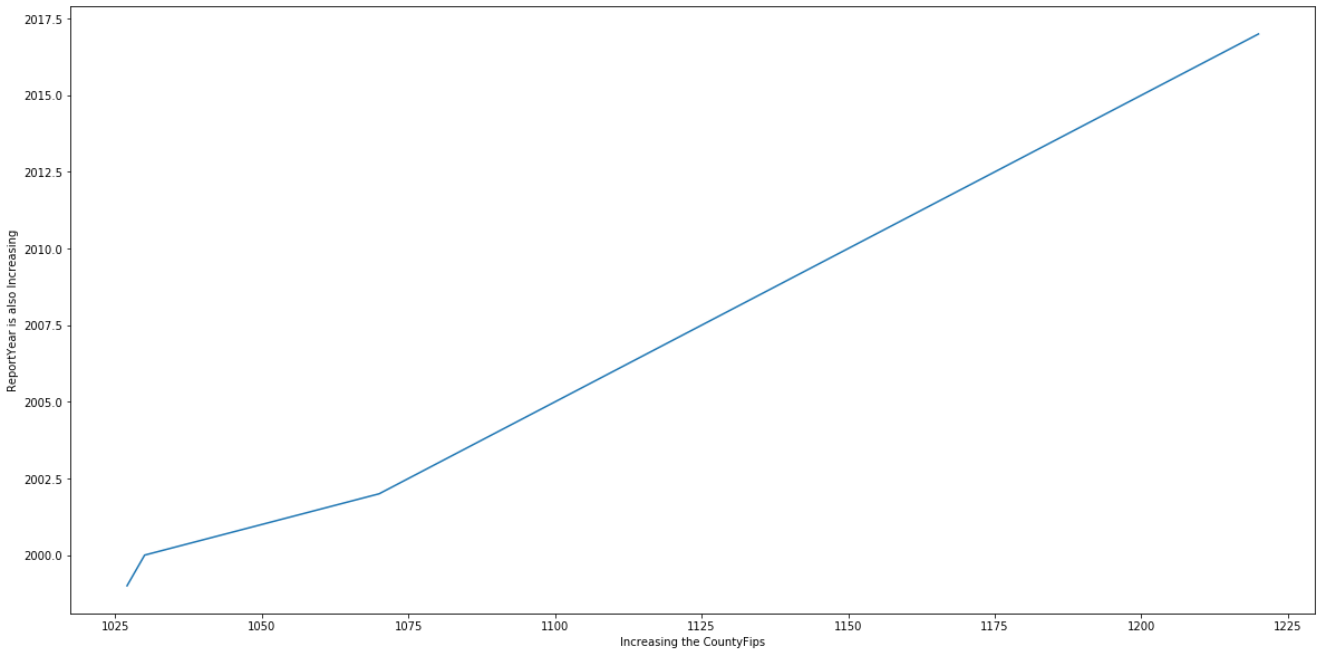
In [40]:

```
%matplotlib inline
import matplotlib.pyplot as plt
```

In [46]:

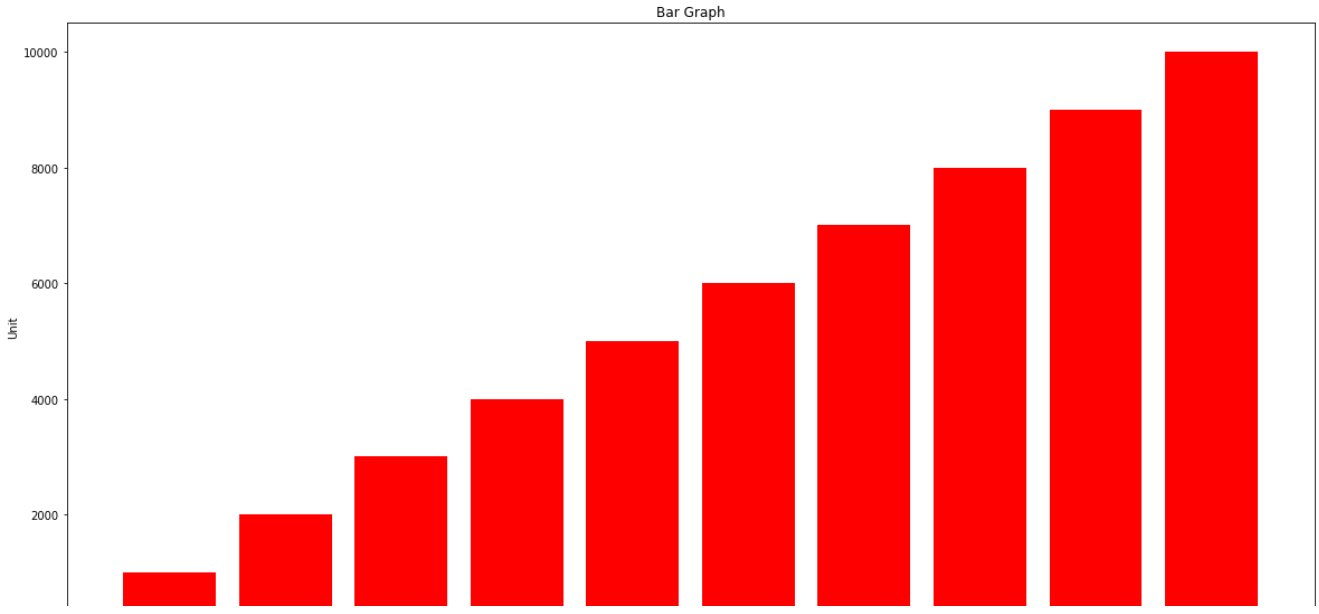
In [40]:

```
# In this we have to take some sample value of the that file And then perform it here for finding how i
t is increasing
CountyFips = [1027, 1030, 1050, 1070, 1080, 1090, 1100, 1110, 1120, 1130, 1140, 1150, 1160, 1170, 1180,
1190, 1200, 1210, 1220]
ReportYear = [1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007,2008, 2009, 2010, 2011, 2012, 2013,
2014, 2015, 2016, 2017]

plt.plot(CountyFips, ReportYear)
plt.xlabel('Increasing the CountyFips')
plt.ylabel('ReportYear is also Increasing')
import matplotlib
matplotlib.rc('figure', figsize=[20,10])
```



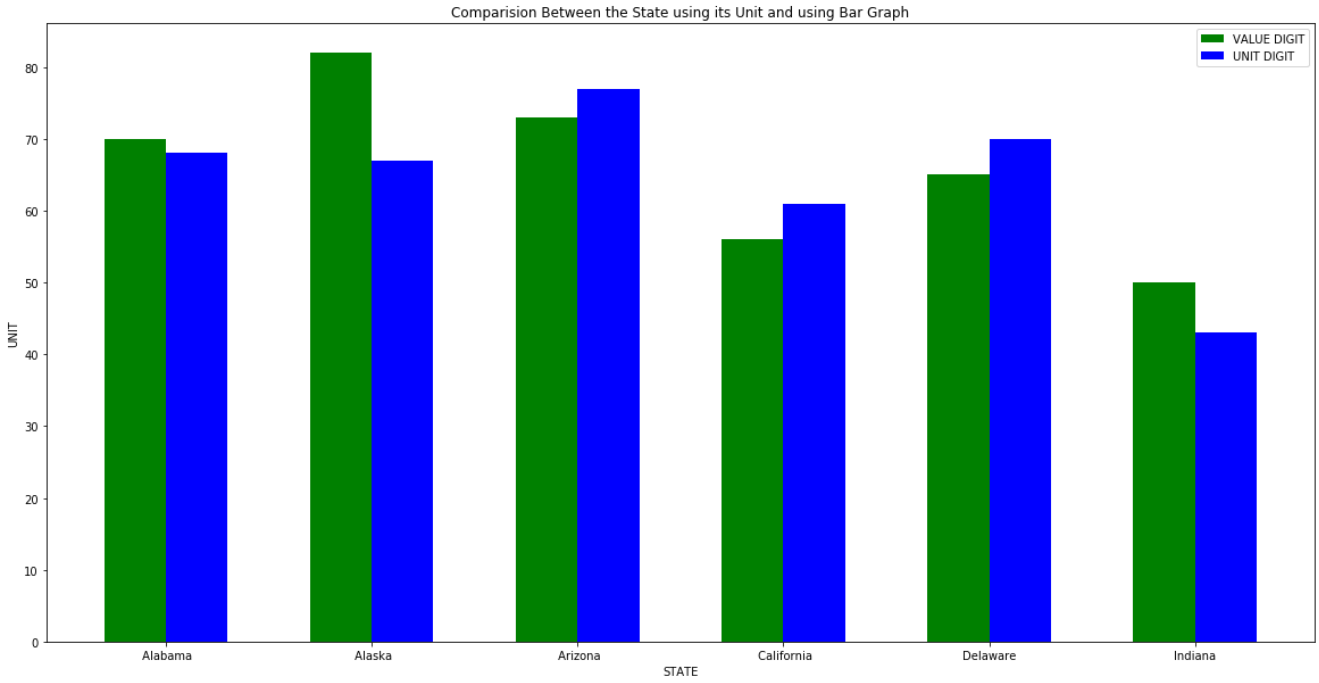
In [42]:

```
import matplotlib.pyplot as plt
import numpy as np

StateName = [" Alabama", " Alaska", " Arizona", " California", "Delaware", " Hawaii", " Indiana", " Iow
a", "New York", "New Mexico"]
Unit=[1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000]
plt.bar(StateName, Unit, color='red')
plt.title("Bar Graph")
plt.xlabel("StateName")
plt.ylabel("Unit")
plt.show()
import matplotlib
matplotlib.rc('figure', figsize=[20,10])
```



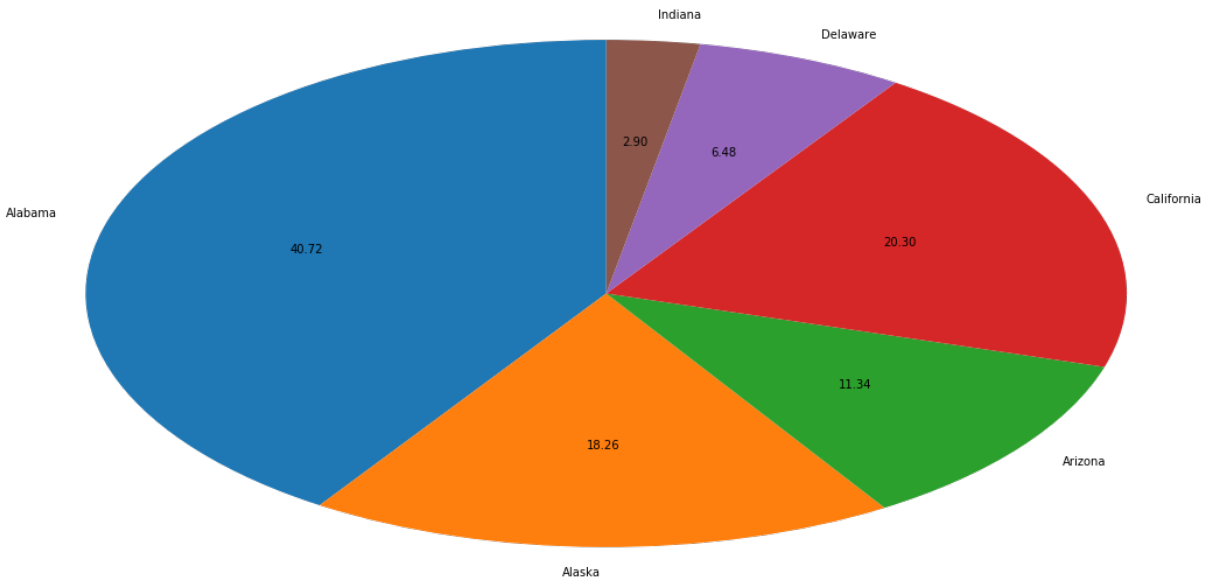
In [56]:

```
import matplotlib.pyplot as plt
import numpy as np

divisions = [" Alabama", " Alaska", " Arizona", " California", "Delaware","Indiana"]
u=[70, 82, 73, 56, 65, 50]
uu=[68, 67, 77, 61, 70, 43]
index= np.arange(6)
width=0.30

plt.bar(index, u, width, color='green', label="VALUE DIGIT")
plt.bar(index+width, uu, width, color='blue', label="UNIT DIGIT")

plt.title("Comparision Between the State using its Unit and using Bar Graph")
plt.xlabel("STATE")
plt.ylabel("UNIT")
plt.xticks(index+width/2, divisions)
plt.legend(loc='best')  # for upper right corner label of div and boys marks
plt.show()
import matplotlib
matplotlib.rc('figure', figsize=[20,10])
```



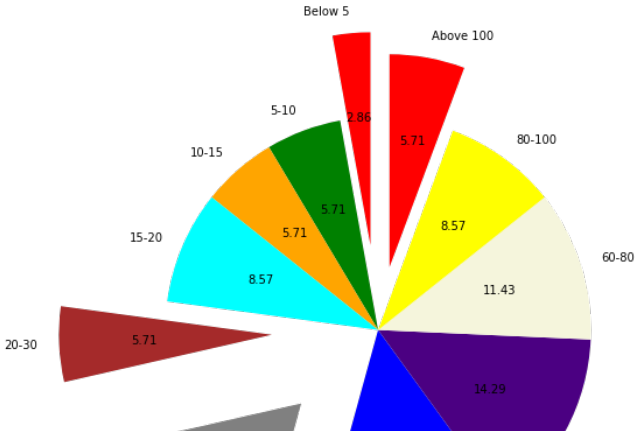
In [58]:

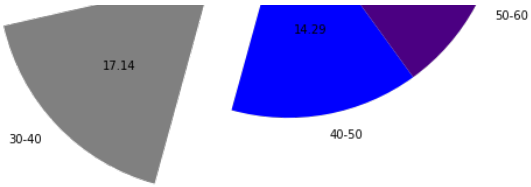
```
 import matplotlib.pyplot as plotter

StateName =  'Alabama', 'Alaska',  'Arizona',  'California', 'Delaware','Indiana'


# Polution data

AirPolution     = [35.69, 16, 9.94, 17.79, 5.68, 2.54]


plotter.pie(AirPolution,

        labels=StateName,

        autopct='%1.2f',  #is a string or function used to label the wedges with their numeric value.
                          #The label will be placed inside the wedge. If it is a format string,
                          #the label will be fmt%pct. If it is a function, it will be called.

        startangle=90)     #float, optional, default: None
                           #If not None, rotates the start of the pie chart by angle degrees counterclo
ckwise from the x-axis.

plotter.show()
```



In [59]:

```
import matplotlib.pyplot as plotter


Values = 'Below 5', '5-10', '10-15', '15-20', '20-30', '30-40', '40-50', '50-60', '60-80', '80-100', 'A
bove 100'


guestNumbers     = [5, 10, 10, 15, 10, 30, 25, 25, 20, 15, 10]

figureObject, axesObject = plotter.subplots()

explode = (0.4, 0.0, 0.0, 0.0, 0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 0.3)

colors  = ("red", "green", "orange", "cyan", "brown", "grey", "blue", "indigo", "beige", "yellow")

axesObject.pie(guestNumbers,

               explode      = explode,

               colors       = colors,

               labels       = ageGroupLabel,

               autopct      = '%1.2f',

               startangle   = 90)

axesObject.axis('equal')

plotter.show()
import matplotlib
matplotlib.rc('figure', figsize=[20,10])
```

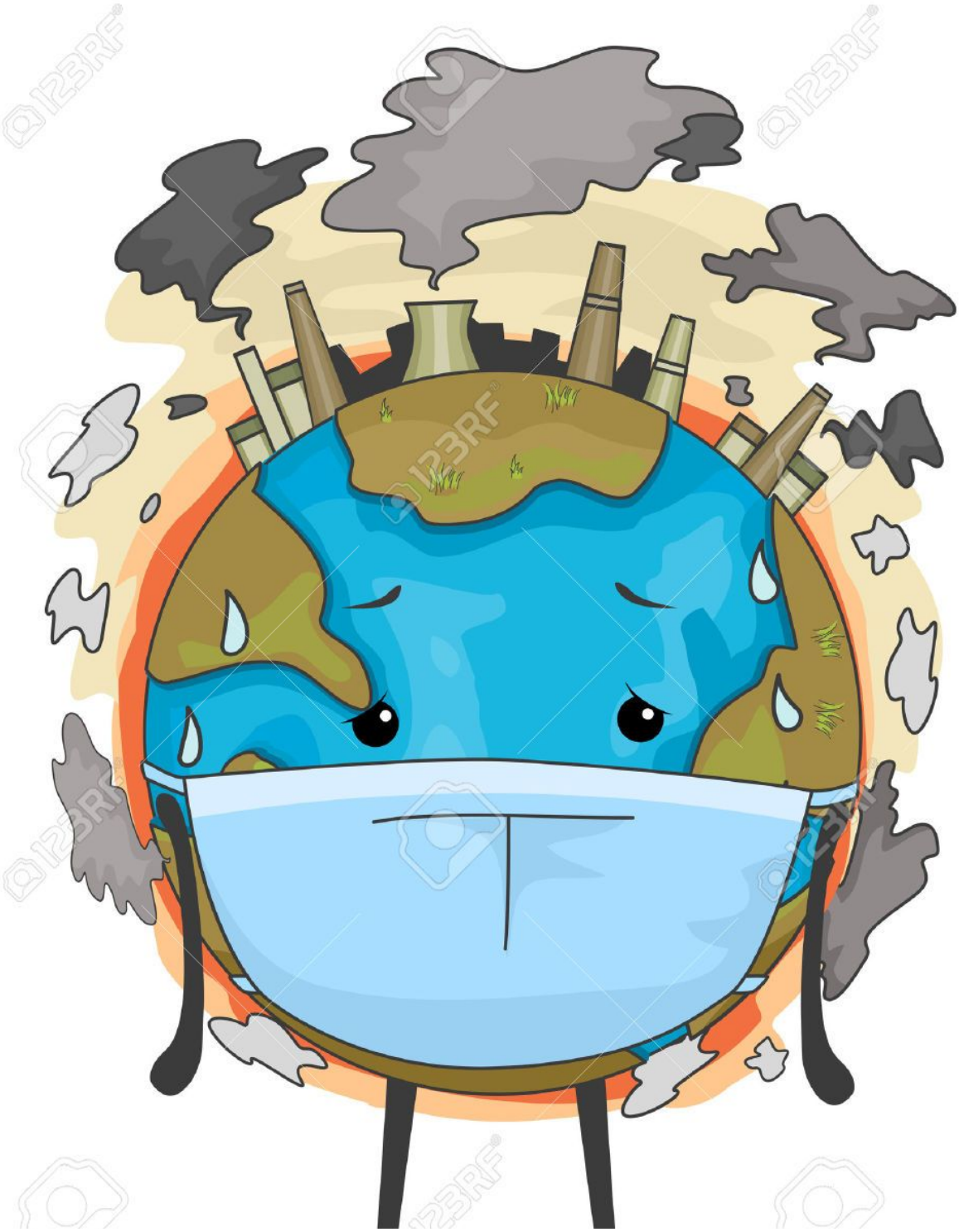




In [67]:

```
from IPython.display import Image
Image(filename='C:/Users/Krishna Singh/Desktop/training materials/PROJECT/img/sec.jpg',width=800, heigh
t=400)


# According to the Data In World 90% diseases comes from Air Pollution
# So please aware
# We don't want this types of world
```

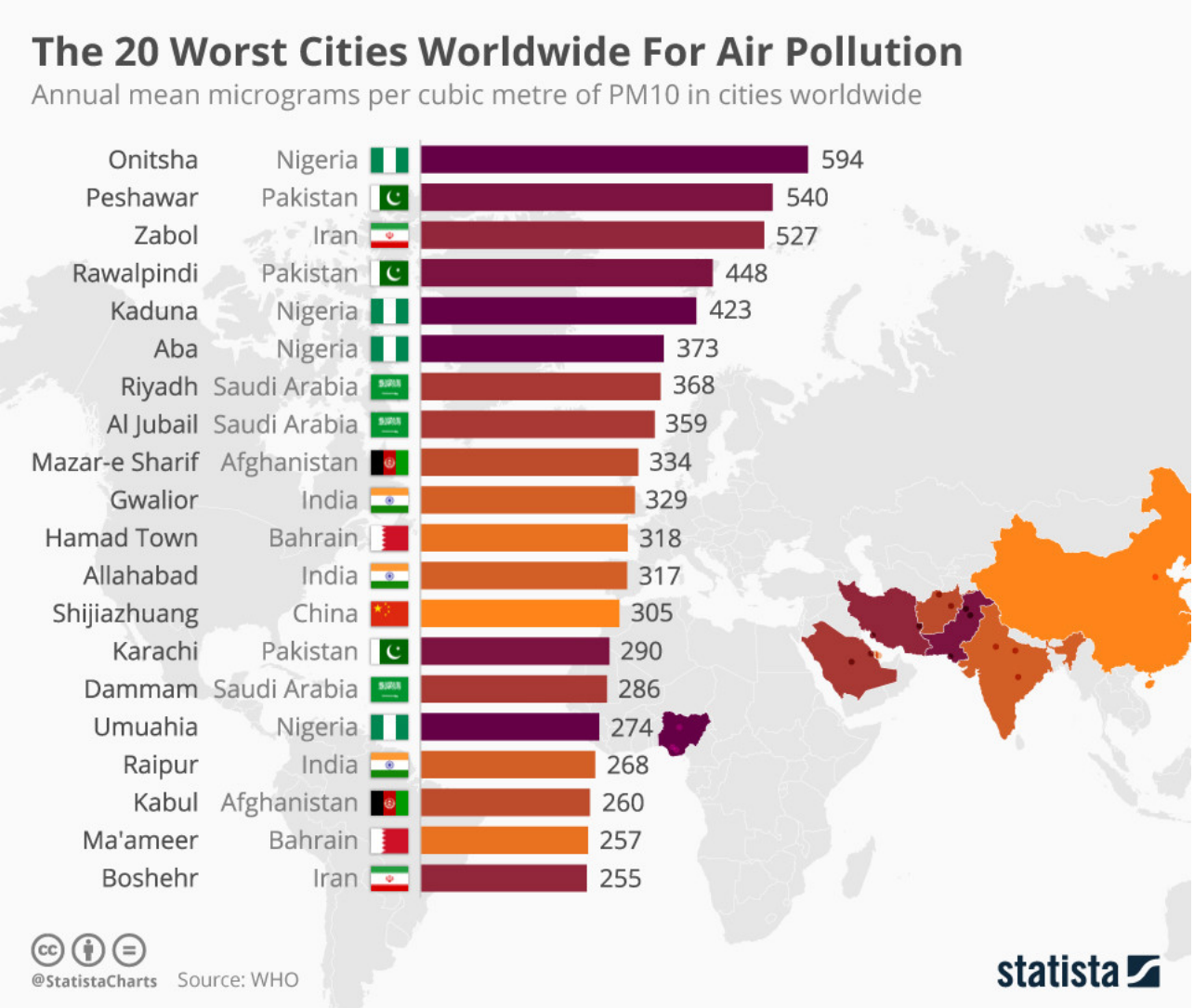
Out[67]:



In [68]:

```
from IPython.display import Image
Image(filename='C:/Users/Krishna Singh/Desktop/training materials/PROJECT/img/fifth.jpg',width=800, hei
ght=400)

#TOp Air Polluted Cities In The World
# According to the data we analysis the whole data the we find the real solution,,,,,,,,,,,,
```

Out[68]:



In [69]:

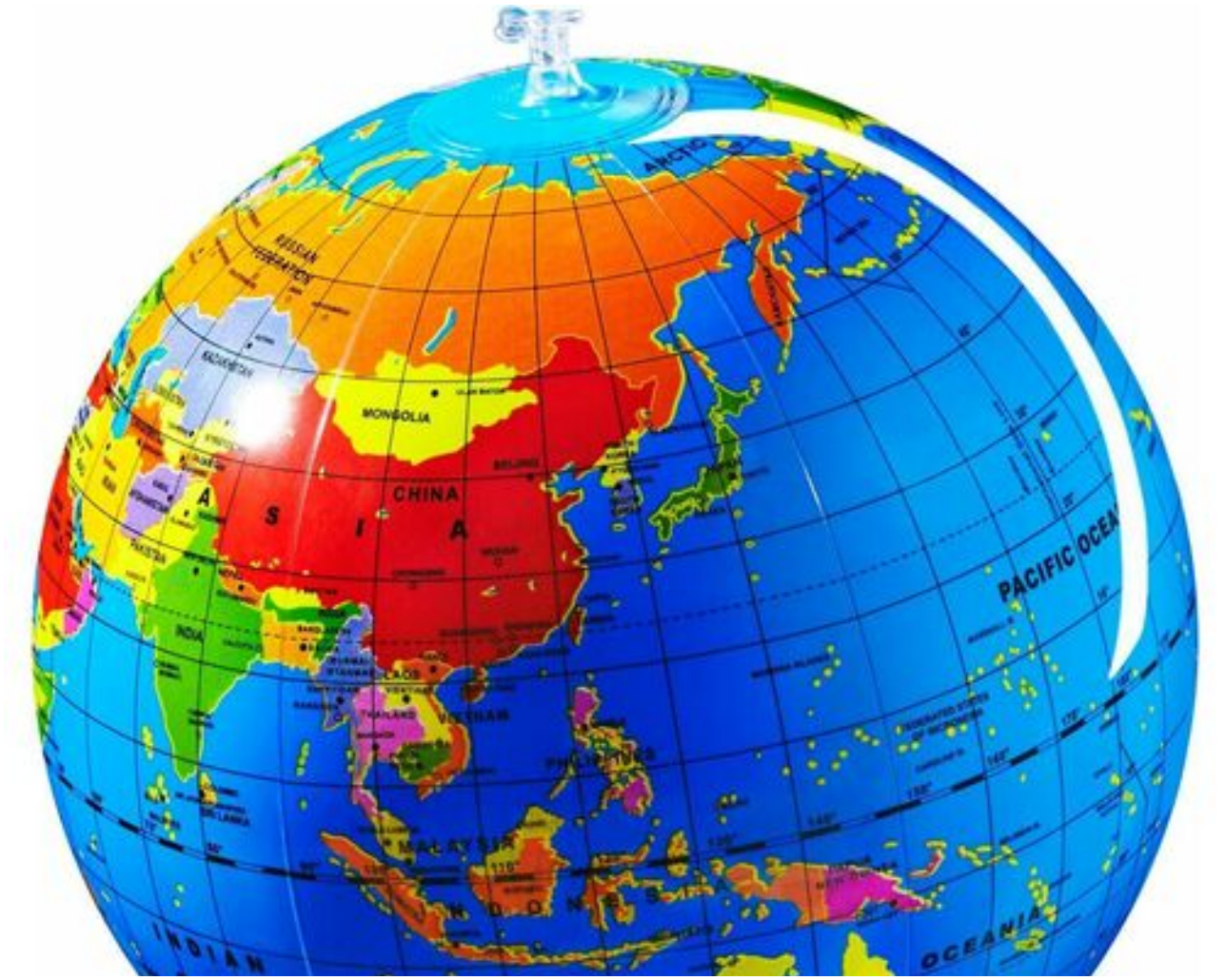
```
from IPython.display import Image
Image(filename='C:/Users/Krishna Singh/Desktop/training materials/PROJECT/img/three.jpg',width=800, hei
ght=400)
```

Out[69]:

In [70]:

```
from IPython.display import Image
Image(filename='C:/Users/Krishna Singh/Desktop/training materials/PROJECT/img/six.jpg',width=800, heigh
t=400)

# WE NEED THIS TYPE OF WORLD
#FOR THIS WE NEED STOP DOING THESE KIND OF THINGS
,,,,,,,,,,,,,,,,,,,,,,,,,,,,
#Stop eating meat (or at least reduce it). ...
#Stop eating dairy. ...
#Change your car driving habits. ...
#Notice how you use water. ...
#Reduce the amount of paper in your life. ...
#Use a refillable water bottle and reusable lunch containers. ...
#Be mindful of what you throw in the trash. ...
#Bag it yourself.
```

Out[70]:



In [ ]: