An Analysis And Visualization Of The World Crime Index Using Python

This notebook utilizes the world crime data set to discover some insights about world safety through data manipulation and visualization practices.

We will be using various funtions in the numpy, matplotlib and seaborn libraries we learned about in the Data Analysis with Python: Zero to Pandas(zerotopandas.com) course.

We will demonstrate the power of visualisation and how its beneficial in exploring the relationship between variables, and how visualisations simplify complex data. We will also explore very interesting funcctions that will help us achieve our goals and answer some questions. Lets jump in.

Downloading the Dataset

Let us download our data

```
!pip install jovian opendatasets --upgrade --quiet
```

Let's begin by downloading the data, and listing the files within the dataset.

```
dataset_url = 'https://www.kaggle.com/datasets/ahmadjalalmasood123/world-crime-index?se
```

```
import opendatasets as od
od.download(dataset_url)
```

```
Skipping, found downloaded files in "./world-crime-index" (use force=True to force
download)
```

The dataset has been downloaded and extracted.

```
data_dir = './world-crime-index'
```

```
import os
os.listdir(data_dir)
```

```
['World Crime Index .csv']
```

Now that we have created a directory for our data we can proceed.

Let us save and upload our work to Jovian before continuing.

```
project_name = "world-crime-index-analysis"
```

```
!pip install jovian --upgrade -q
```

```
!pip install plotly
```

```
Requirement already satisfied: plotly in /opt/conda/lib/python3.9/site-packages
(5.13.1)
Requirement already satisfied: tenacity>=6.2.0 in /opt/conda/lib/python3.9/site-
packages (from plotly) (8.2.2)
```

```
import jovian
```

```
jovian.commit(project=project_name)
```

```
[jovian] Updating notebook "edwardakuffoaddo/world-crime-index-analysis" on
https://jovian.com
[jovian] Committed successfully! https://jovian.com/edwardakuffoaddo/world-crime-index-
analysis
```

```
'https://jovian.com/edwardakuffoaddo/world-crime-index-analysis'
```

Data Preparation and Cleaning

Let's load the dataset into a data frame using Pandas.
Explore the number of rows & columns, ranges of values etc.
Handle missing, incorrect and invalid data and perform any additional steps (parsing dates, creating additional columns, merging multiple dataset etc.)

```
import pandas as pd
```

```
worldcrime_df = pd.read_csv('world-crime-index/World Crime Index .csv')
```

```
worldcrime_df
```

	Rank	City	Crime Index	Safety Index
0	1	Caracas, Venezuela	83.98	16.02
1	2	Pretoria, South Africa	81.98	18.02
2	3	Celaya, Mexico	81.80	18.20
3	4	San Pedro Sula, Honduras	80.87	19.13
4	5	Port Moresby, Papua New Guinea	80.71	19.29
...	...	...	...	...
448	449	Quebec City, Canada	15.14	84.86
449	450	Taipei, Taiwan	15.05	84.95
450	451	San Sebastian, Spain	14.86	85.14
451	452	Doha, Qatar	13.96	86.04
452	453	Abu Dhabi, United Arab Emirates	11.67	88.33

453 rows × 4 columns

First off, lets take a look at the basic information of the data. This includes the number of rows and columns, the number of non-null elements in each column and the data type. This basic information can let us know which area of the data to take a closer look at while cleaning.

```
worldcrime_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 453 entries, 0 to 452
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Rank          453 non-null    int64  
 1   City          453 non-null    object 
 2   Crime Index   453 non-null    float64
 3   Safety Index  453 non-null    float64
dtypes: float64(2), int64(1), object(1)
memory usage: 14.3+ KB
```

The data shows 453 entries so straight away we can tell which columns have null rows. In this case there are no non null values to account for.

```
worldcrime_df.shape
```

```
(453, 4)
```

The dot shape function simply returns the number of rows and columns in the data frame.

Now let us check for null data in each column just to demonstrate the specific use of the isnull function.

```
worldcrime_df.isnull().sum()
```

```
Rank            0
City            0
Crime Index     0
Safety Index    0
dtype: int64
```

We can see that there are in fact no null values to remove.

Let us try and split the city column into two separate city and country columns for easier analysis

```
citycountrysplit = worldcrime_df['City'].str.split(pat=',',expand=True)
citycountrysplit
```

	0	1	2
0	Caracas	Venezuela	None
1	Pretoria	South Africa	None
2	Celaya	Mexico	None

	0	1	2
3	San Pedro Sula	Honduras	None
4	Port Moresby	Papua New Guinea	None
...	...	...	...
448	Quebec City	Canada	None
449	Taipei	Taiwan	None
450	San Sebastian	Spain	None
451	Doha	Qatar	None
452	Abu Dhabi	United Arab Emirates	None

453 rows × 3 columns

Using the split function we are able to separate strings with a particular indicator. In this case our indicator is the comma separating the city and the country.

Now lets rename the columns for convenience

```
worldcrime_df['City'] = citycountrysplit[0]
worldcrime_df['Country'] = citycountrysplit[1]
worldcrime_df
```

	Rank	City	Crime Index	Safety Index	Country
0	1	Caracas	83.98	16.02	Venezuela
1	2	Pretoria	81.98	18.02	South Africa
2	3	Celaya	81.80	18.20	Mexico
3	4	San Pedro Sula	80.87	19.13	Honduras
4	5	Port Moresby	80.71	19.29	Papua New Guinea
...	...	...	...	...	...
448	449	Quebec City	15.14	84.86	Canada
449	450	Taipei	15.05	84.95	Taiwan
450	451	San Sebastian	14.86	85.14	Spain
451	452	Doha	13.96	86.04	Qatar
452	453	Abu Dhabi	11.67	88.33	United Arab Emirates

453 rows × 5 columns

Personally, I don't have any use for the rank column in this data set. So I'm going to go ahead and drop it using the drop function.

```
worldcrime_df = worldcrime_df.drop(['Rank'], axis=1)
```

```
worldcrime_df
```

	City	Crime Index	Safety Index	Country
0	Caracas	83.98	16.02	Venezuela

	City	Crime Index	Safety Index	Country
1	Pretoria	81.98	18.02	South Africa
2	Celaya	81.80	18.20	Mexico
3	San Pedro Sula	80.87	19.13	Honduras
4	Port Moresby	80.71	19.29	Papua New Guinea
...	...	...	...	...
448	Quebec City	15.14	84.86	Canada
449	Taipei	15.05	84.95	Taiwan
450	San Sebastian	14.86	85.14	Spain
451	Doha	13.96	86.04	Qatar
452	Abu Dhabi	11.67	88.33	United Arab Emirates

453 rows × 4 columns

This is a good time to check for duplicates as well. I'm going to check in the city column.

```
worldcrime_df['City'].duplicated().value_counts()
```

```
False    449
True       4
Name: City, dtype: int64
```

We have discovered four duplicated values, so let us go ahead and remove those. We must remember to use the keep argument to retain one of the duplicated values or else we will be removing the original value as well as the duplicates.

```
worldcrime_df = worldcrime_df.drop_duplicates(subset=['City'], keep='last')
worldcrime_df
```

	City	Crime Index	Safety Index	Country
0	Caracas	83.98	16.02	Venezuela
1	Pretoria	81.98	18.02	South Africa
2	Celaya	81.80	18.20	Mexico
3	San Pedro Sula	80.87	19.13	Honduras
4	Port Moresby	80.71	19.29	Papua New Guinea
...	...	...	...	...
448	Quebec City	15.14	84.86	Canada
449	Taipei	15.05	84.95	Taiwan
450	San Sebastian	14.86	85.14	Spain
451	Doha	13.96	86.04	Qatar
452	Abu Dhabi	11.67	88.33	United Arab Emirates

449 rows × 4 columns

Exploratory Analysis and Visualization

Compute the mean, sum, range and other interesting statistics for numeric columns
Explore distributions of numeric columns using histograms etc.
Explore relationship between columns using scatter plots, bar charts etc.
Make a note of interesting insights from the exploratory analysis

Let's begin by importing `matplotlib.pyplot and seaborn` .

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

sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9, 5)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

Let's display our data frame again for reference.

```
worldcrime_df
```

	City	Crime Index	Safety Index	Country
0	Caracas	83.98	16.02	Venezuela
1	Pretoria	81.98	18.02	South Africa
2	Celaya	81.80	18.20	Mexico
3	San Pedro Sula	80.87	19.13	Honduras
4	Port Moresby	80.71	19.29	Papua New Guinea
...	...	...	...	...
448	Quebec City	15.14	84.86	Canada
449	Taipei	15.05	84.95	Taiwan
450	San Sebastian	14.86	85.14	Spain
451	Doha	13.96	86.04	Qatar
452	Abu Dhabi	11.67	88.33	United Arab Emirates

449 rows × 4 columns

```
worldcrime_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 449 entries, 0 to 452
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   City          449 non-null    object
 1   Crime Index   449 non-null    float64
 2   Safety Index  449 non-null    float64
 3   Country        449 non-null    object
dtypes: float64(2), object(2)
memory usage: 17.5+ KB
```

Let us use the describe function to get the descriptive statistics for both data movie and TV Show data frames. This will be applied to the release year column. The pandas library applies this function to the the most applicable column which in this case is the release year column.

```
worldcrime_df.describe()
```

	Crime Index	Safety Index
count	449.000000	449.000000
mean	44.774944	55.225056
std	15.505000	15.505000
min	11.670000	16.020000
25%	32.920000	44.910000
50%	44.560000	55.440000
75%	55.090000	67.080000
max	83.980000	88.330000

The descriptive statistics tells us that there are 449 cities in the database. The mean indexes indicate a leaning towards safety overall. This could imply that global efforts towards peace and security are relatively successful.

Lets explore one or more columns by plotting a graph below, and add some explanation about it

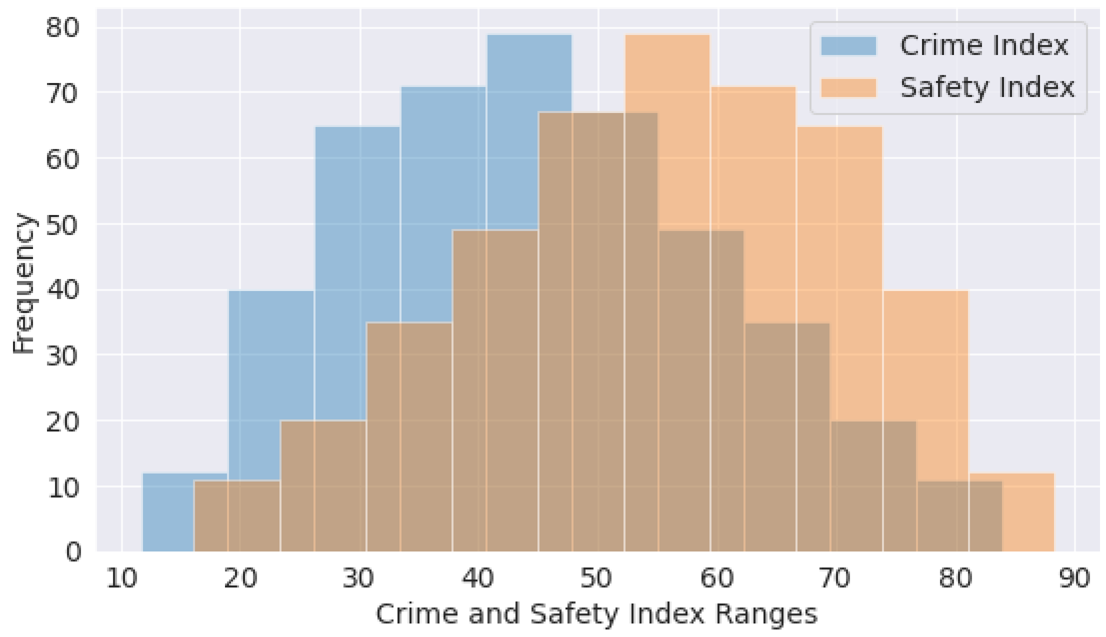
We can rule out the line plot since the information shown here is clearly inapplicable. This is because the data frame does not require a time series analysis. There are no date or time columns at all.

Explore one or more columns by plotting a graph below, and add some explanation about it

Now lets explore the data using a histogram.

First lets look at the Crime index column of the data frame

```
plt.hist(worldcrime_df['Crime Index'], alpha = 0.4)
plt.hist(worldcrime_df['Safety Index'], alpha = 0.4)
plt.xlabel('Crime and Safety Index Ranges')
plt.ylabel('Frequency')
plt.legend(['Crime Index', 'Safety Index']);
```

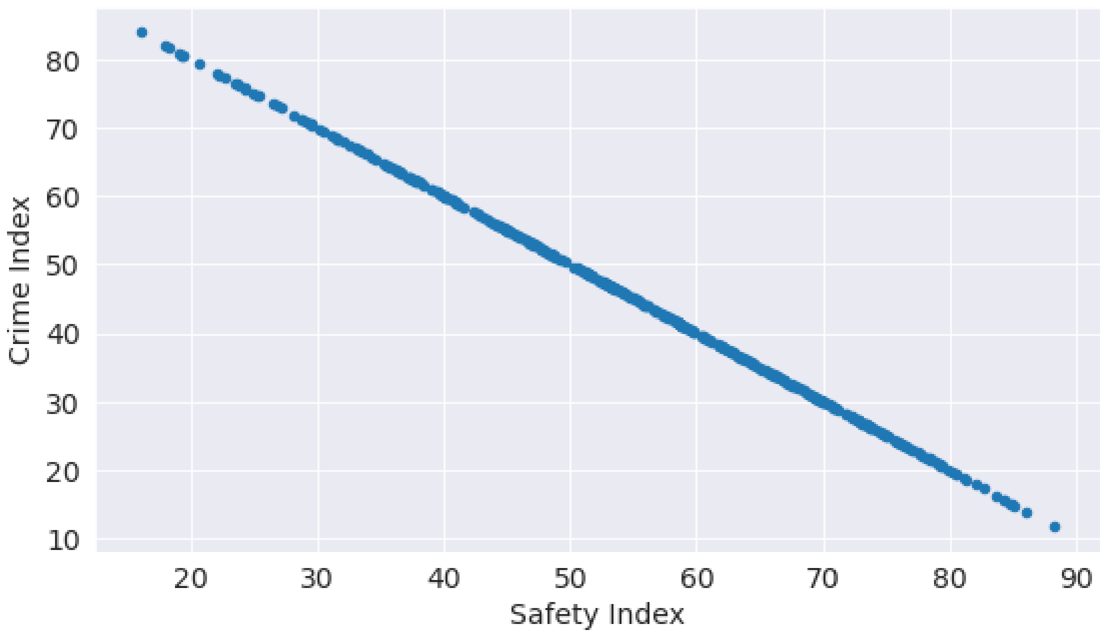


The histogram above shows a mirror image between the two columns. We can clearly absverve this image from the shape of the histograms overlapping eachother.
This may be attributed to the fact that the dataframe is based on indexes that compliment eachother and add up to 100 for each city recorded.

Explore one or more columns by plotting a graph below, and add some explanation about it

Let us explore this relationship further with a scatter plot. This graph explores the correlation between two data sets.

```
worldcrime_df.plot.scatter(x = 'Safety Index', y = 'Crime Index')
plt.xlabel('Safety Index')
plt.ylabel('Crime Index');
```



The scatter plot above resulted in a straight line. This is characteristic of a linear relationship between the data points. A relationship is linear if one variable increases by approximately the same rate as the other variables changes by one unit.

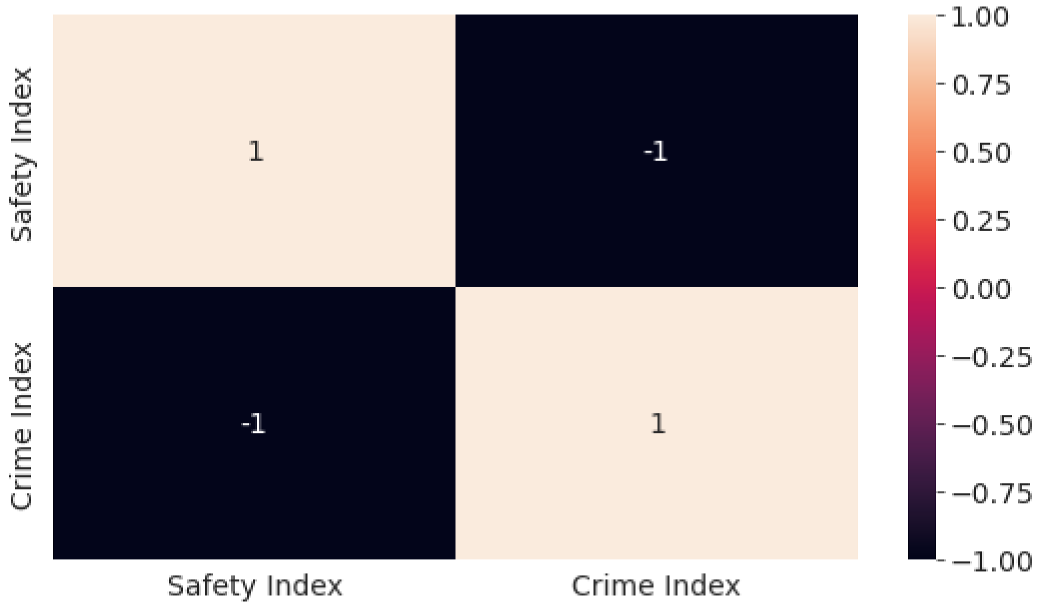
Explore one or more columns by plotting a graph below, and add some explanation about it

Lets us use the heatmap correlation graph to solidify this observation.

```
heatdf = pd.DataFrame(worldcrime_df, columns =['Safety Index', 'Crime Index'])

corr = heatdf.corr()
sns.heatmap(corr, annot = True)
```

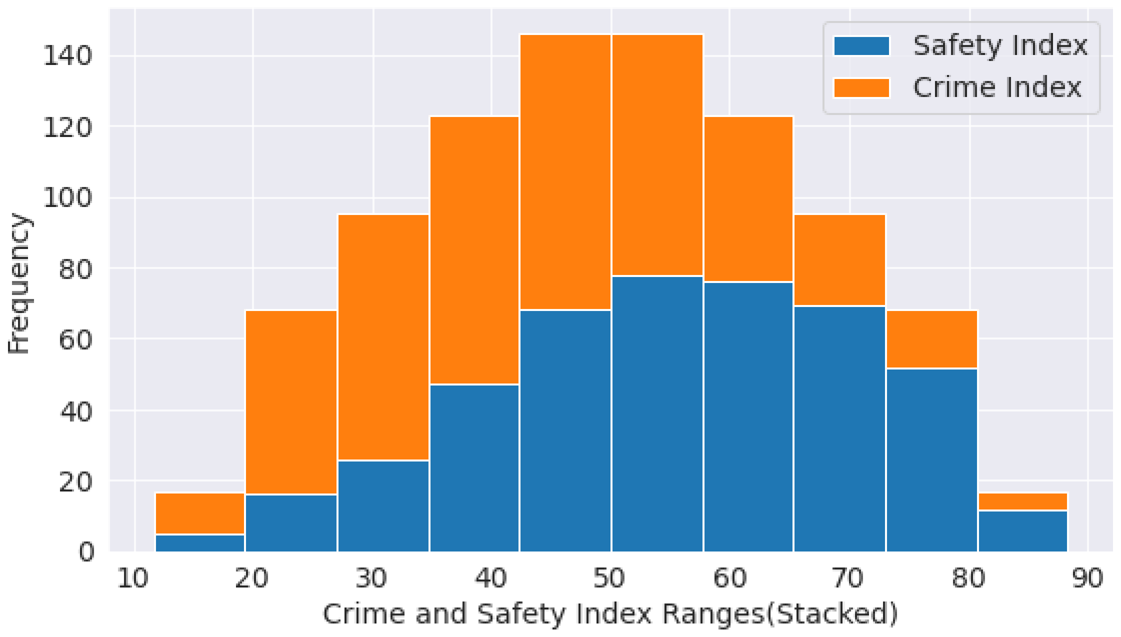
```
<AxesSubplot:>
```



our observation is confirmed.
The heatmap shows a minus one coefficient between the columns. indicating that an increase in one variable reliably predicts a decrease in the other one.

Let us try another type of histogram below.

```
plt.hist([worldcrime_df['Safety Index'], worldcrime_df['Crime Index']], stacked = True)
plt.xlabel('Crime and Safety Index Ranges(Stacked)')
plt.ylabel('Frequency')
plt.legend(['Safety Index', 'Crime Index']);
```



this histogram also allows us to see a direct comparison between the two columns within those ranges because they are stacked on top of each other. it allows easier visua comparison.

Let us save and upload our work to Jovian before continuing

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "edwardakuffoaddo/world-crime-index-analysis" on
https://jovian.com
[jovian] Committed successfully! https://jovian.com/edwardakuffoaddo/world-crime-index-
analysis
```

```
'https://jovian.com/edwardakuffoaddo/world-crime-index-analysis'
```

Asking and Answering Questions

Ask at least 5 interesting questions about your dataset
Answer the questions either by computing the results using Numpy/Pandas or by plotting graphs using Matplotlib/Seaborn
Create new columns, merge multiple dataset and perform grouping/aggregation wherever necessary
Wherever you're using a library function from Pandas/Numpy/Matplotlib etc. explain briefly what it does

Q1: What are the top 5 countries with the highest crime index

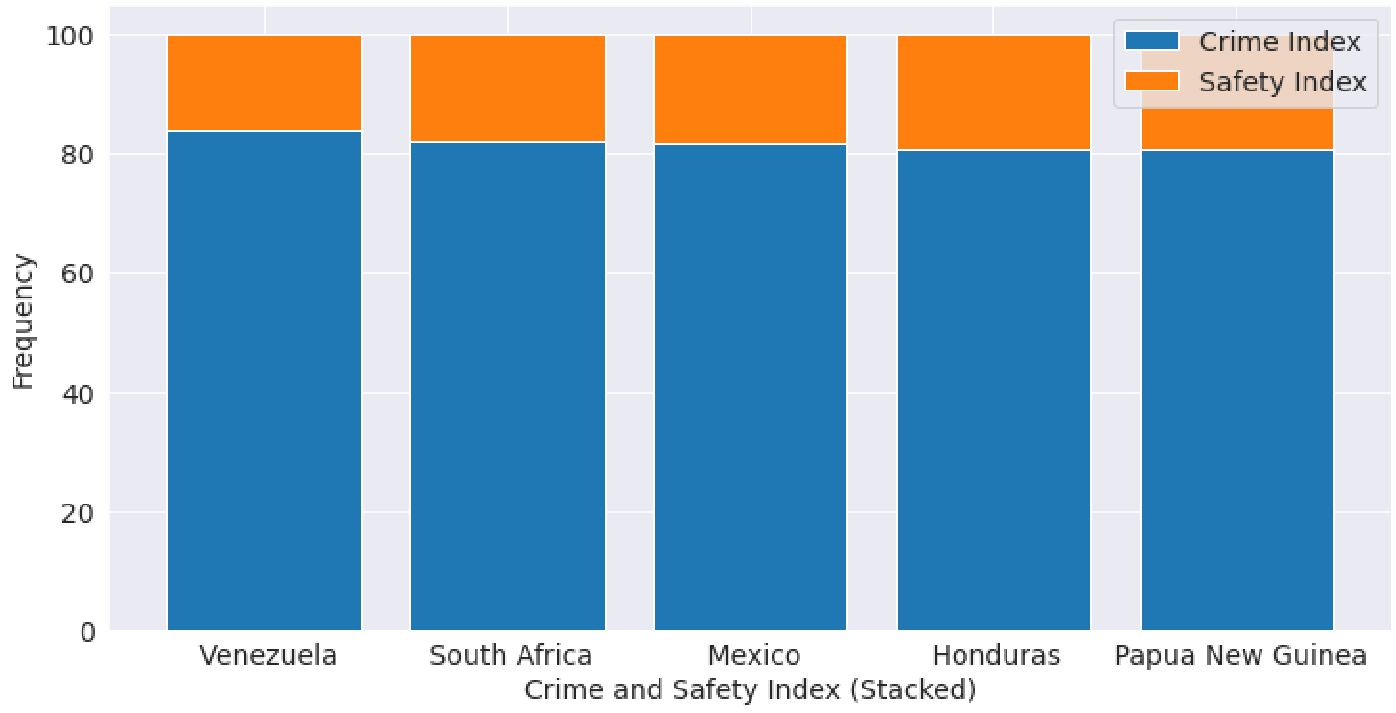
In order for us to find the top 5 countries with the highest crime index we need to utilise the head function.

```
top5temp = worldcrime_df.head(5)
top5temp
```

	City	Crime Index	Safety Index	Country
0	Caracas	83.98	16.02	Venezuela

	City	Crime Index	Safety Index	Country
1	Pretoria	81.98	18.02	South Africa
2	Celaya	81.80	18.20	Mexico
3	San Pedro Sula	80.87	19.13	Honduras
4	Port Moresby	80.71	19.29	Papua New Guinea

```
plt.figure(figsize=(12, 6))
plt.bar(top5temp['Country'],top5temp['Crime Index'])
plt.bar(top5temp['Country'], top5temp['Safety Index'], bottom = top5temp['Crime Index']
plt.xlabel('Crime and Safety Index (Stacked)')
plt.ylabel('Frequency')
plt.legend(['Crime Index', 'Safety Index']);
```



Since our data is already in ranking order we were just able to use the head function to isolated the top 5 rows of data as seen above. We can clearly see how high the countries' crime index is compared to the safety indexes at the top.

Q2: what countries in the top 10 safest countries have the highest safety index?

The data frame is sorted in descending order of world crime. This means that the safety column is in ascending order. This means we need to sort the data using the safety column and isolate the top 10

```
topsafetemp = worldcrime_df.sort_values('Safety Index', ascending=False)
topsafetemp
```

	City	Crime Index	Safety Index	Country
452	Abu Dhabi	11.67	88.33	United Arab Emirates
451	Doha	13.96	86.04	Qatar
450	San Sebastian	14.86	85.14	Spain
449	Taipei	15.05	84.95	Taiwan

	City	Crime Index	Safety Index	Country
448	Quebec City	15.14	84.86	Canada
...	...	...	...	...
4	Port Moresby	80.71	19.29	Papua New Guinea
3	San Pedro Sula	80.87	19.13	Honduras
2	Celaya	81.80	18.20	Mexico
1	Pretoria	81.98	18.02	South Africa
0	Caracas	83.98	16.02	Venezuela

449 rows × 4 columns

Now lets use the head function to isolate the top ten

```
top10safetemp =topsafetemp.head(10)
top10safetemp
```

	City	Crime Index	Safety Index	Country
452	Abu Dhabi	11.67	88.33	United Arab Emirates
451	Doha	13.96	86.04	Qatar
450	San Sebastian	14.86	85.14	Spain
449	Taipei	15.05	84.95	Taiwan
448	Quebec City	15.14	84.86	Canada
447	Ajman	15.64	84.36	United Arab Emirates
446	Sharjah	15.69	84.31	United Arab Emirates
445	Dubai	16.30	83.70	United Arab Emirates
444	Zurich	17.26	82.74	Switzerland
443	Bern	17.94	82.06	Switzerland

```
import plotly.express as px

fig = px.sunburst(top10safetemp, path=['Country'], values='Safety Index')
fig.show()
```

The sunburst chart above shows us a cummulative pie chart for the countries with multiple records of safety indexes. This allows us to get a sense of how much safer one country is from another by summing up all the safety indexes associated with said country and comparing the results. Here we can clearly see that The united arab emirates is leading.

Doing a cross sectional analysis requires that we find multiple angles to confirm a suspected or apparent result. So we will first look at a cummulative result as we have done above and then look at the number of cities that are associated with each country below.

```
jovian.commit()
```

```
[jovian] Updating notebook "edwardakuffoaddo/world-crime-index-analysis" on
https://jovian.com
[jovian] Committed successfully! https://jovian.com/edwardakuffoaddo/world-crime-index-
analysis
```

```
'https://jovian.com/edwardakuffoaddo/world-crime-index-analysis'
```

Q3: Which countries in the top 50 safest countries have the highest frequency of cities

First lets filter for the top 50 safest countries

```
top50safe = topsafetemp.head(50)
top50safe
```

	City	Crime Index	Safety Index	Country
452	Abu Dhabi	11.67	88.33	United Arab Emirates
451	Doha	13.96	86.04	Qatar
450	San Sebastian	14.86	85.14	Spain
449	Taipei	15.05	84.95	Taiwan
448	Quebec City	15.14	84.86	Canada
447	Ajman	15.64	84.36	United Arab Emirates
446	Sharjah	15.69	84.31	United Arab Emirates
445	Dubai	16.30	83.70	United Arab Emirates

	City	Crime Index	Safety Index	Country
444	Zurich	17.26	82.74	Switzerland
443	Bern	17.94	82.06	Switzerland
442	Munich	18.66	81.34	Germany
441	Eskisehir	18.86	81.14	Turkey
440	Trondheim	19.41	80.59	Norway
439	Lugano	19.48	80.52	Switzerland
438	Oradea	19.82	80.18	Romania
437	Basel	20.12	79.88	Switzerland
436	Muscat	20.54	79.46	Oman
435	Arhus	20.60	79.40	Denmark
434	Tartu	20.70	79.30	Estonia
433	Groningen	20.80	79.20	Netherlands
432	Irvine	20.90	79.10	CA
431	Manama	21.06	78.94	Bahrain
430	Zagreb	21.36	78.64	Croatia
429	Ljubljana	21.48	78.52	Slovenia
428	Hong Kong	21.62	78.38	Hong Kong
427	Yerevan	21.66	78.34	Armenia
426	Tampere	21.86	78.14	Finland
425	Cluj-Napoca	22.16	77.84	Romania
424	The Hague (Den Haag)	22.26	77.74	Netherlands
423	Canberra	22.31	77.69	Australia
422	Coquitlam	22.51	77.49	Canada
421	Amarillo	22.97	77.03	TX
420	Reykjavik	23.02	76.98	Iceland
419	Stavanger	23.32	76.68	Norway
418	Rijeka	23.43	76.57	Croatia
417	Tallinn	23.59	76.41	Estonia
416	Tokyo	23.79	76.21	Japan
415	Chiang Mai	23.91	76.09	Thailand
414	Timisoara	24.06	75.94	Romania
413	Prague	24.10	75.90	Czech Republic
412	Kigali	24.16	75.84	Rwanda
411	Eindhoven	24.32	75.68	Netherlands
410	Helsinki	24.92	75.08	Finland
409	Merida	24.96	75.04	Mexico
408	Mangalore	25.07	74.93	India
407	Markham	25.15	74.85	Canada
406	Tbilisi	25.16	74.84	Georgia

	City	Crime Index	Safety Index	Country
405	Seoul	25.38	74.62	South Korea
404	Lausanne	25.53	74.47	Switzerland
403	Aalborg	25.54	74.46	Denmark

Now lets use the value counts function to get a count of the number of times a country occurs in the list.

```
top50safe.Country.value_counts()
```

```
  Switzerland             5
  United Arab Emirates    4
  Romania                 3
  Netherlands             3
  Canada                  3
  Finland                 2
  Estonia                 2
  Denmark                 2
  Croatia                 2
  Norway                  2
  Australia               1
  Georgia                 1
  India                   1
  Mexico                  1
  Rwanda                  1
  Czech Republic          1
  Thailand                1
  Japan                   1
  Iceland                 1
  TX                      1
  Spain                   1
  Oman                    1
  Armenia                 1
  Hong Kong               1
  Slovenia                1
  Qatar                   1
  Bahrain                 1
  CA                      1
  Taiwan                  1
  Germany                 1
  Turkey                  1
  South Korea             1
 Name: Country, dtype: int64
```

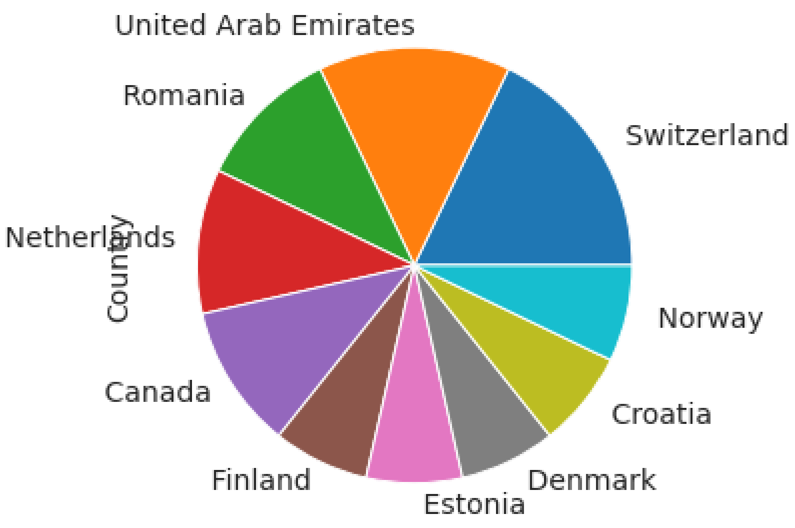
Now lets remove the negligible data by filtering for the top ten of the list.

```
temp50safe = top50safe.Country.value_counts().head(10)
```

```
temp50 = pd.DataFrame(temp50safe)
temp50
```

	Country
Switzerland	5
United Arab Emirates	4
Romania	3
Netherlands	3
Canada	3
Finland	2
Estonia	2
Denmark	2
Croatia	2
Norway	2

```
plot = temp50safe.plot.pie(y = 'Country' ,figsize=(5, 5))
```



Qatar took a substantial spot in the graph above but is not placed in this graph. this indicates that its safety index is not as substantial as compared to dubai and switzerland who took the lead in the previous graph as well as in the pie chart here. we get a stronger sense of meaningful safety.

This information tells us that Switzerland has the most cities with the highest safety indexes and this allows us to properly make assumptions about the level of safety. A country could have only one city with a high safety index and other cities with terrible safety indexes so this chart gives us a clear idea of safety in the country.

Q4: Which countries in the top 50 dangerous countries have the highest frequency of cities?

Let us filter the data frame for the top 50 countries using the head function.

```
top50crime = worldcrime_df.head(50)
top50crime
```

	City	Crime Index	Safety Index	Country
0	Caracas	83.98	16.02	Venezuela
1	Pretoria	81.98	18.02	South Africa
2	Celaya	81.80	18.20	Mexico
3	San Pedro Sula	80.87	19.13	Honduras
4	Port Moresby	80.71	19.29	Papua New Guinea
5	Durban	80.60	19.40	South Africa
6	Johannesburg	80.55	19.45	South Africa
7	Kabul	79.39	20.61	Afghanistan
8	Rio de Janeiro	77.93	22.07	Brazil
9	Natal	77.69	22.31	Brazil
10	Fortaleza	77.36	22.64	Brazil
11	Port Elizabeth	76.44	23.56	South Africa
12	Recife	76.42	23.58	Brazil
13	Port of Spain	76.21	23.79	Trinidad And Tobago
14	Baltimore	75.75	24.25	America
15	Salvador	75.69	24.31	Brazil
16	Rosario	75.11	24.89	Argentina
17	Memphis	74.76	25.24	America
18	Detroit	74.63	25.37	America
19	Rockhampton	73.51	26.49	Australia
20	Cape Town	73.13	26.87	South Africa
21	Porto Alegre	72.92	27.08	Brazil
22	Tijuana	71.87	28.13	Mexico
24	Bloemfontein	71.32	28.68	South Africa
25	Bradford	71.24	28.76	United Kingdom
26	Albuquerque	70.93	29.07	America
27	Lima	70.70	29.30	Peru
28	Guayaquil	70.59	29.41	Ecuador
29	Sao Paulo	70.49	29.51	Brazil
30	Saint Louis	70.46	29.54	America
31	San Salvador	69.79	30.21	El Salvador
32	Cali	69.50	30.50	Colombia
33	Mexico City	68.86	31.14	Mexico
34	Windhoek	68.64	31.36	Namibia
35	San Juan	68.55	31.45	Puerto Rico
36	Santo Domingo	68.37	31.63	Dominican Republic
37	Coventry	68.35	31.65	United Kingdom
38	Damascus	67.94	32.06	Syria
39	Luanda	67.45	32.55	Angola

	City	Crime Index	Safety Index	Country
40	New Orleans	67.05	32.95	America
41	Milwaukee	66.78	33.22	America
42	Campinas	66.74	33.26	Brazil
43	Oakland	66.54	33.46	America
44	Lagos	66.22	33.78	Nigeria
45	Chicago	66.10	33.90	America
46	Nantes	65.70	34.30	France
47	Bogota	65.42	34.58	Colombia
48	Manila	64.72	35.28	Philippines
49	Surrey	64.58	35.42	Canada
50	Cleveland	64.39	35.61	America

We noticed that the data entry recorded states in America instead of the country America. We need to fix that

We can fix this using the simple loc function to rename data at specific locations.

```
top50crime.at[14, 'Country'] = 'America'
top50crime.at[17, 'Country'] = 'America'
top50crime.at[18, 'Country'] = 'America'
top50crime.at[26, 'Country'] = 'America'
top50crime.at[30, 'Country'] = 'America'
top50crime.at[40, 'Country'] = 'America'
top50crime.at[41, 'Country'] = 'America'
top50crime.at[43, 'Country'] = 'America'
top50crime.at[45, 'Country'] = 'America'
top50crime.at[50, 'Country'] = 'America'
top50crime
```

	City	Crime Index	Safety Index	Country
0	Caracas	83.98	16.02	Venezuela
1	Pretoria	81.98	18.02	South Africa
2	Celaya	81.80	18.20	Mexico
3	San Pedro Sula	80.87	19.13	Honduras
4	Port Moresby	80.71	19.29	Papua New Guinea
5	Durban	80.60	19.40	South Africa
6	Johannesburg	80.55	19.45	South Africa
7	Kabul	79.39	20.61	Afghanistan
8	Rio de Janeiro	77.93	22.07	Brazil
9	Natal	77.69	22.31	Brazil
10	Fortaleza	77.36	22.64	Brazil
11	Port Elizabeth	76.44	23.56	South Africa
12	Recife	76.42	23.58	Brazil
13	Port of Spain	76.21	23.79	Trinidad And Tobago

	City	Crime Index	Safety Index	Country
14	Baltimore	75.75	24.25	America
15	Salvador	75.69	24.31	Brazil
16	Rosario	75.11	24.89	Argentina
17	Memphis	74.76	25.24	America
18	Detroit	74.63	25.37	America
19	Rockhampton	73.51	26.49	Australia
20	Cape Town	73.13	26.87	South Africa
21	Porto Alegre	72.92	27.08	Brazil
22	Tijuana	71.87	28.13	Mexico
24	Bloemfontein	71.32	28.68	South Africa
25	Bradford	71.24	28.76	United Kingdom
26	Albuquerque	70.93	29.07	America
27	Lima	70.70	29.30	Peru
28	Guayaquil	70.59	29.41	Ecuador
29	Sao Paulo	70.49	29.51	Brazil
30	Saint Louis	70.46	29.54	America
31	San Salvador	69.79	30.21	El Salvador
32	Cali	69.50	30.50	Colombia
33	Mexico City	68.86	31.14	Mexico
34	Windhoek	68.64	31.36	Namibia
35	San Juan	68.55	31.45	Puerto Rico
36	Santo Domingo	68.37	31.63	Dominican Republic
37	Coventry	68.35	31.65	United Kingdom
38	Damascus	67.94	32.06	Syria
39	Luanda	67.45	32.55	Angola
40	New Orleans	67.05	32.95	America
41	Milwaukee	66.78	33.22	America
42	Campinas	66.74	33.26	Brazil
43	Oakland	66.54	33.46	America
44	Lagos	66.22	33.78	Nigeria
45	Chicago	66.10	33.90	America
46	Nantes	65.70	34.30	France
47	Bogota	65.42	34.58	Colombia
48	Manila	64.72	35.28	Philippines
49	Surrey	64.58	35.42	Canada
50	Cleveland	64.39	35.61	America

Now that we have our data updated we can impliment the value counts function

```
top50crime.Country.value_counts()
```

```
America                 10
 Brazil                  8
 South Africa            6
 Mexico                  3
 Colombia                2
 United Kingdom          2
 Venezuela               1
 Philippines             1
 France                  1
 Nigeria                 1
 Angola                  1
 Syria                   1
 Dominican Republic      1
 Puerto Rico             1
 Namibia                 1
 Peru                    1
 El Salvador             1
 Ecuador                 1
 Australia               1
 Argentina               1
 Trinidad And Tobago     1
 Afghanistan             1
 Papua New Guinea        1
 Honduras                1
 Canada                  1
Name: Country, dtype: int64
```

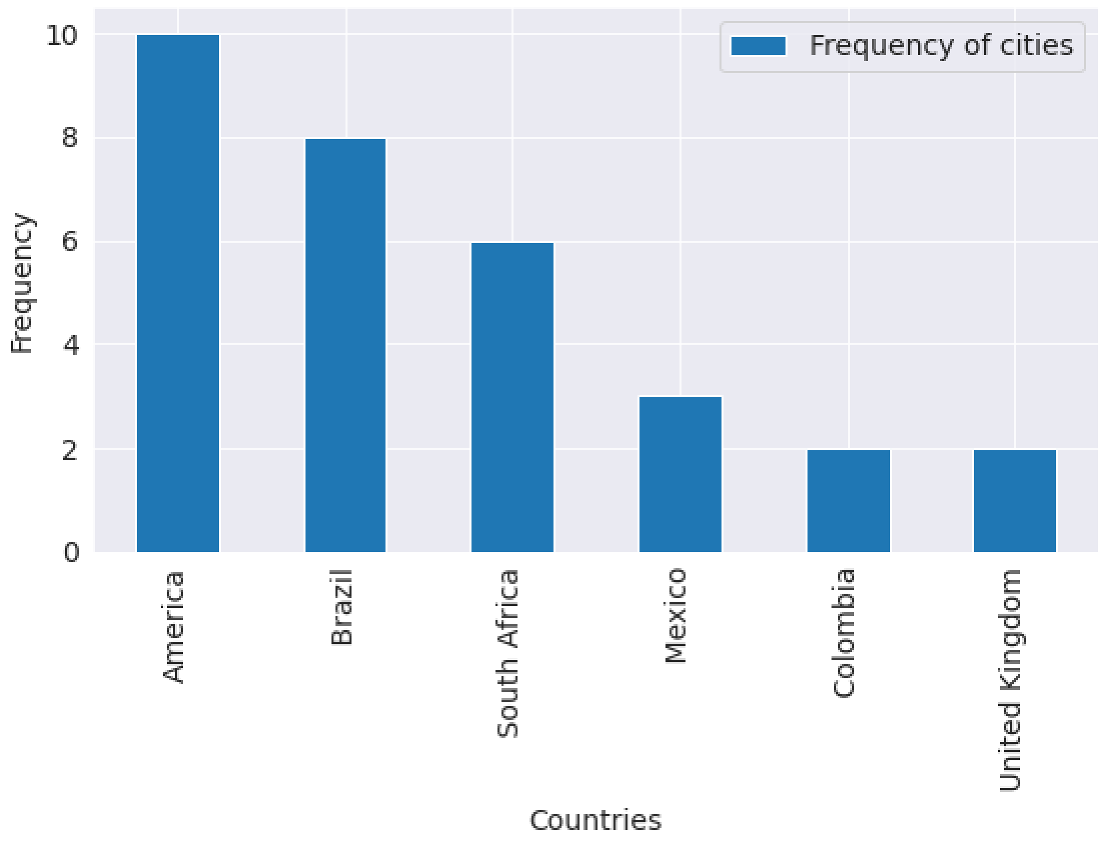
Finally lets filter for the countries that occur multiple times

```
top50crime.Country.value_counts().head(6)
```

```
America            10
 Brazil             8
 South Africa       6
 Mexico             3
 Colombia           2
 United Kingdom     2
Name: Country, dtype: int64
```

```
topworst = top50crime.Country.value_counts().head(6)
topworst =  {'Frequency': topworst}
topworst_df = pd.DataFrame(topworst)

topworst_df.plot(kind = 'bar')
plt.xlabel('Countries')
plt.ylabel('Frequency')
plt.legend(['Frequency of cities']);
```



As explained earlier, getting the countries with the most frequency of violent cities will allow us to get a clearer picture of the crime rate in the entire country. One city is not enough to define the safety or crime rate for an entire country. For a travel and tour company for example this insight is more useful. When you visit a country you want to know that majority of the country is safe not just a single city. This gives a stronger sense of safety.

```
jovian.commit()
```

```
[jovian] Updating notebook "edwardakuffoaddo/world-crime-index-analysis" on
https://jovian.com
[jovian] Committed successfully! https://jovian.com/edwardakuffoaddo/world-crime-index-
analysis
```

```
'https://jovian.com/edwardakuffoaddo/world-crime-index-analysis'
```

Q5: Does the data indicate more peace or more crime on a global scale?

The best way I can think of to answer this question is using the describe function.
The information provided includes the 75% row.

```
worldcrimedescribe = worldcrime_df.describe()
worldcrimedescribe
```

	Crime Index	Safety Index
count	449.000000	449.000000
mean	44.774944	55.225056
std	15.505000	15.505000
min	11.670000	16.020000

	Crime Index	Safety Index
25%	32.920000	44.910000
50%	44.560000	55.440000
75%	55.090000	67.080000
max	83.980000	88.330000

The describe function tells us the average index that contribute certain percentages of the distribution.

```
worldcrimedescribe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 8 entries, count to max
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Crime Index   8 non-null      float64
 1   Safety Index  8 non-null      float64
dtypes: float64(2)
memory usage: 192.0+ bytes
```

I used the info function here to just confirm that the output was a data frame class

```
Crime = worldcrimedescribe['Crime Index'][6]
print(f'the 75% statistic for world crime is {Crime}')
```

```
the 75% statistic for world crime is 55.09
```

```
Safety = worldcrimedescribe['Safety Index'][6]
print(f'the 75% statistic for world Safety is {Safety}')
```

```
the 75% statistic for world Safety is 67.08
```

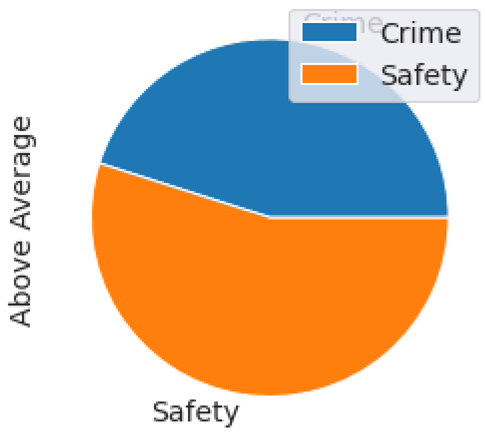
Lets create a new dataframe with this information that will make it easier for us to create a pie chart.

```
df = pd.DataFrame({'Above Average': [Crime, Safety]},
                  index=['Crime', 'Safety'])
df
```

	Above Average
Crime	55.09
Safety	67.08

Now we can use the plot function to plot a pie chart using the data above.

```
plot = df.plot.pie(y = 'Above Average' ,figsize=(4, 5))
```



The data in the table is very useful for identifying the safest country but with so much data it will be extremely difficult to say with any certainty what the real sense of the balance of global safety against crime is. Using this simple pie chart derived from the descriptive statistics we can see very clearly that the distribution favors world safety.

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "edwardakuffoaddo/world-crime-index-analysis" on
https://jovian.com
[jovian] Committed successfully! https://jovian.com/edwardakuffoaddo/world-crime-index-
analysis
```

```
'https://jovian.com/edwardakuffoaddo/world-crime-index-analysis'
```

Inferences and Conclusion

1. Top 5 countries with the highest crime indexes are developing countries from Africa and South america
2. The top countries with the highest safety indexes are located in Europe and United Arab Emirates.
3. Switzerland can be said to be the safest country in the world because it has one of the top 5 highest safety indexes and also has the highest number of cities within the top 50 safest cities in the world.
4. America has the highest number of cities with the highest crime indexes. This contradicts our developing countries theory. This could be attributed to the difference in population and land mass that America has but without additional information this is pure speculation.
   5.We can tell from the analysis we have conducted here, that world crime and world safety are in a bit of a tug of war. The frequencies of the countries with the highest crime safety ratios are outliers. 75% of the cities held a more balanced ratio with safety having a slight lead. This is an accurate representation of our reality. The world can be said to be relatively safe in todays day and age with select corners of the world experiencing spikes in crime. on a large scale crime is being actively fought against by a large part of humanity. The data seems to support this observation.

However the data has an obvious handicap. It can only reflect data of recorded crime. In areas of the world where crime is more difficult to record or document the data would be unavailable. This observation arises due to the fact that the country with the most cities with high crime indexes was America. This could be merely due to the fact that American crime is properly recorded where as in other countries crime may be properly recorded in capital cities and have other cities neglected.

In spite of these handicaps. I belive the data performed well.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "edwardakuffoaddo/world-crime-index-analysis" on
https://jovian.com
[jovian] Committed successfully! https://jovian.com/edwardakuffoaddo/world-crime-index-
analysis
```

```
'https://jovian.com/edwardakuffoaddo/world-crime-index-analysis'
```

References and Future Work.

The data set was relatively simple. The data entry was done quite well so the cleaning process was smooth. Having zero non null values and numerical columns that compliment each other quite directly made analysis more convenient. This data set showcases the important of proper data entry.

The information coming in the form of indexes was also an interesting situation. Figuring out that each row had values that added up to 100 allows for more robust visualizations using percentages.

The data set can be combined with age data relevant to the same date range to allow for possible inferences in the correlation between and safety in cities. It could also be combined with the data concerning education that can be used to assess a possible causation of the various safety and crime indexes in each country.

Finally the size of the countries and population size could be beneficial in having a more accurate picture of crime safety ratios.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "edwardakuffoaddo/world-crime-index-analysis" on
https://jovian.com
```

Here are some links that proved useful during the analysis:

https://www.askpython.com/python-modules/pandas/update-the-value-of-a-row-dataframe

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.pie.html

https://www.wikihow.com/Draw-a-Pie-Chart-from-Percentages

https://www.w3resource.com/pandas/dataframe/dataframe-rank.php#:~:text=The%20rank()%20function%20is,the%20ranks%20of%20those%20values.&text=Index%20to%2

https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy