

PROJECT REPORT

ANALYSIS OF HEALTHY LIFESTYLE ACROSS CITIES USING CLUSTERING

Submitted by

DEEPIKA S



ABSTRACT

Health as a synthetic indicator of the quality and standard of living. At the same time, in accordance with the presentation of the World Health Organization, the category of health includes the categories of physical, mental, spiritual and public health. Health is created and lived by people within the settings of their everyday life; where they learn, work, play, and love. A healthy city aims to create a health-supportive environment, achieve a good quality of life, provide basic sanitation and hygiene needs, and supply access to health care. Being a healthy lifestyle city depends not only on health infrastructures, but also on a commitment to improve a city's environment and a willingness to forgive the necessary connections in political, economic, and social areas. In past studies feature extraction using Supervised Learning has proposed manually, here we try to propose a methodology in which we will analysis healthiest cities across world and do Exploratory Data Analysis and we will use K-means Clustering algorithm in Unsupervised Learning. Because it consists of different features and continuous data so we use clustering to improve statistical analysis.

Dataset Link: <https://github.com/Deepikasak/CAPSTONE-PROJECT>

Source Code: <https://github.com/Deepikasak/CAPSTONE-PROJECT-SOURCE-CODE.git>

ACKNOWLEDGEMENTS

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of my capstone project. I am thankful for their aspiring guidance, invaluable constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

Further, I have fortunate to have Mr. PRASAD as my mentor. He has readily shared his immense knowledge in data analytics and guideme in a manner that the outcome resulted in enhancing my data skills.

I wish to thank all the faculties, as this project utilized knowledge gained from every course that formed the DSP program.

I certify that the work done by me for conceptualizing and completing this project is original and authentic.

Date: July 10, 2022

Name: DEEPIKA S

Certificate of Completion

I hereby certify that the project titled “ANALYSIS OF HEALTHY LIFESTYLE ACROSS CITIES USING CLUSTERING” was undertaken and completed under my supervision by Mr. Prasad from the batch of DSP (July 2022)

Mentor: Mr. Prasad

Date: July 10, 2022

Place: Karur

Table of contents

CHAPTER NO	TITLE	PAGE NO
	Abstract	2
	Acknowledgements	3
1	Introduction	7
1.1	Title & Objective of the study	7
1.2	Business or Enterprise under study	8
2	Data Collection & Data Preparation	9
2.1	Amsterdam	9
2.2	Sydney	10
2.3	Stockholm	10
3	EDA for Healthiest Cities	11
3.1	Top Five Healthiest Cities	11
3.2	Least Five Healthiest Cities	12
4	Analyze Asian Cities	13
4.1	Asian Cities	13
5	Fitting Models to Data	14
5.1	K-means clustering	14
5.2	Elbow Graph	15
5.3	Silhouette Score	16
5.4	Plotting the Custers	17
6	Conclusion	43
7	Reference	44

List of Figures

FIGURE NO	TITLE	PAGE NO
1.1	Healthy lifestyle cities	7
2.1	Data's in Amsterdam	9
2.2	Data's in Sydney	10
2.3	Data's in Stockholm	10
3.1.1	Code for EDA Process	11
3.1.2	Top Five Healthiest Cities	11
3.2	Code for EDA Process	12
4.1	Plotting of Asian Cities	13
5.1	Silhouette Visualizer	14
5.2	Elbow Graph	15
5.3	Silhouette Score	16
5.4.1	Cluster vs Happiness Level	17
5.4.2	Print the Clusters	17

CHAPTER 1

INTRODUCTION

1.1 TITLE & OBJECTIVE OF THE STUDY:

- ✓ Health is an indicator of the quality of life indicators of health and morbidity refer to mental groups of healthy and sick people. A healthy city aims to create a health-supportive environment, achieve a good quality of life, provide basic sanitation and hygiene needs, and supply access to health care. A healthier lifestyle is consistently associated with a longer life expectancy across various individual risks and irrespective of the presence of multiple long-term medical conditions. The major factor behind longer life expectancy is happiness. Higher levels of happiness leads to higher life expectancy rates.
- ✓ The main aim is to find relationships between various factors that lead to healthy or unhealthy lifestyle in cities and visualize them. To determine the healthy life of the people across the world using different feature which leads to the healthy or unhealthy life.



Figure:1.1 HEALTHY LIFESTYLE CITIES

1.2 BUSINESS OR ENTERPRISE UNDER STUDY

Relevant studies revealed that healthiest lifestyle associated with the a longer life expectancy. To our knowledge, no study to date has investigated this relationship in relation to the presence of multimorbidity. They investigated the association between healthy lifestyle and individual risk factors with life expectancy in relation to the presence of multimorbidity. And found that an overall healthy lifestyle largely counterbalances the negative association between multimorbidity and life expectancy.

According to a study conducted by YOUNGOV, the most common resolutions for 2020 we take care of themselves (45%), exercise more(43%), and lose weight(41%). It is no wonder that millions of people around the world are starting the new year with a good purpose to take care of themselves, whether it be joining the gym or eating a healthier diet. However, fitness and diet are not the only parameters that define being it. In fact, the environment that surrounds us and the choice of lifestyle are also important factors that determine our level of health.

Lenstore analyzed 44 international cities to find out which ones promote a healthy lifestyle. It was possible to evaluate each city according to 10 different parameters, including the levels of obesity and the rate of pollution.

In this paper, the model I used to analyze the data is K-means Clustering algorithm under unsupervised learning. It allow us to cluster the data into different groups in a convenient way to discover the categories of groups in the unlabeled dataset on its own, without the need for any training.

CHAPTER 2

DATA COLLECTION AND PREPARATION

Dataset for Healthy lifestyle across cities is found from various sources such as from Kaggle, web and browser searches. And the dataset consists of 12 columns such as City, Rank, Sunshine hours, Cost of a bottle of a water, Obesity levels, Life expectancy, Pollution, Annual avg hours, Happiness levels, Outdoor activities, Number of takeout places, Cost of a monthly gym membership.

Across the world I have taken only 44 cities which belongs to the healthy life styles. Now I have explain some cities, why I have choose these specific cities for my dataset.

2.1 AMSTERDAM

Amsterdam is the city that offers the healthiest lifestyle ever. Known around the world for its bicycles, it is no wonder that the Dutch capital is ranked first. Amsterdam offers more than 400 outdoor activities to keep you active and healthy, as well as ranking fifth among the happiest cities in the world. Amsterdam is the ideal city for all those who wish to lead a healthy and balanced lifestyle.

Sunshine hours	1858
Annual avg hours	1434
Happiness levels	7.44
Outdoor activities	422

Figure:2.1 DATA'S IN AMSTERDAM

2.2 SYDNEY

Sydney has 2,636 hours of sunshine per year. In addition to continuing to attract tourists from all over the world with its dream beaches. This city offers more than 406 activities to enjoy outdoors. Despite the relatively high average gym membership price 45.83 and a life expectancy of 82 years, Sydney ranks second among international cities offering the healthiest lifestyle.

Sunshine hours	2636
Annual avg hours	1712
Happiness levels	7.22
Outdoor activities	406

Figure:2.2 DATA'S IN SYDNEY

2.3 STOCKHOLM

Stockholm is among the best cities when it comes to maintaining a balance between personal and working life. When looking at the number of hours worked per year. Stockholm ranks fourth with an average of 1,452 working hours per year, giving Swedish workers the chance to earn for a more balance lifestyle. Further more, considering other parameters such as the very low pollution level, and the very high general happiness level.

Sunshine hours	1821
Annual avg hours	1452
Happiness levels	7.35
Outdoor activities	129

Figure: 3.2 DATA'S IN STOCKHOLM

CHAPTER 3

EDA FOR HEALTHIEST CITIES

3.1. TOP FIVE HEALTHIEST CITIES:

A dataset consists of 44 cities of healthy lifestyle, by using Exploratory data analysis (EDA), I have analyze top 5 healthiest cities based on the happiness level, outdoor activities and annual avg working hours.

Exploratory analysis of data for the 5 the healthiest cities.

```
In [27]: df.columns
Out[27]: Index(['City', 'Rank', 'Sunshine hours(City)', 'Cost of a bottle of water(City)', 'Obesity levels(Country)', 'Life expectancy(years) (Country)', 'Pollution(Index score) (City)', 'Annual avg. hours worked', 'Happiness levels(Country)', 'Outdoor activities(City)', 'Number of take out places(City)', 'Cost of a monthly gym membership(City)'],
dtype='object')

In [28]: columns=['Sunshine hours(City)', 'Cost of a bottle of water(City)', 'Obesity levels(Country)', 'Life expectancy(years) (Country)', 'Pollution(Index score) (City)', 'Annual avg. hours worked', 'Happiness levels(Country)', 'Outdoor activities(City)', 'Number of take out places(City)', 'Cost of a monthly gym membership(City)]

In [29]: import matplotlib.pyplot as plt
for column in columns:
    top10 = df.nlargest(5,column)
    plt.figure(figsize=(10,4))
    plt.bar(top10['City'], top10[column], color='blue', alpha=0.9)
    plt.ylabel(column, fontsize = 10)
    plt.xlabel("Country", fontsize = 10)
    plt.title("Top 5 countries per " + column, fontsize = 10)
```

Figure: 3.1.1 CODE FOR EDA PROCESS

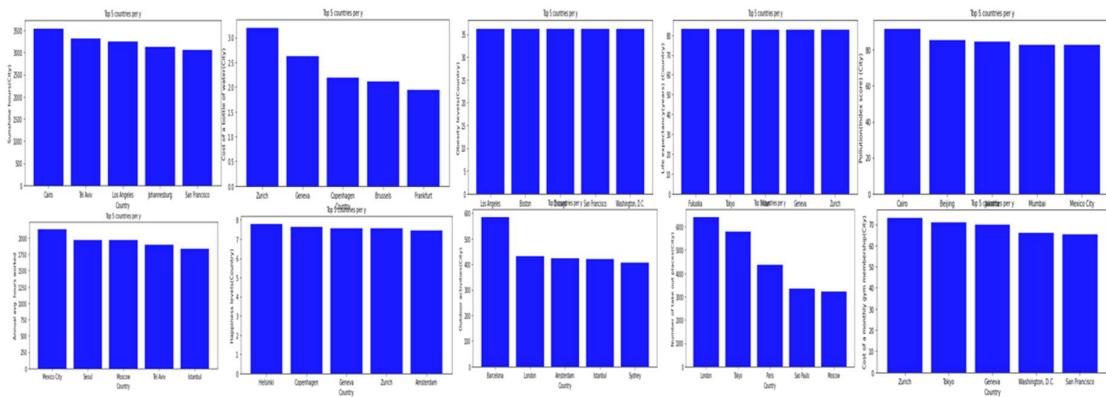


Figure:3.1.2 TOP 5 HEALTHIEST CITIES

3.2 LEAST FIVE HEALTHIEST CITIES:

A dataset consists of 44 cities of healthy lifestyle, by using Exploratory data analysis (EDA), I have analyze top 5 healthiest cities based on the happiness level, outdoor activities and annual avg hours working hours columns in the dataset.

```
In [30]: for column in columns:  
    top10 = df.nsmallest(5,column)  
    plt.figure(figsize=(10,4))  
    plt.bar(top10['city'], top10[column], color='green', alpha=0.5)  
    plt.ylabel(column, fontsize = 10)  
    plt.xlabel("Country", fontsize = 10)  
    plt.title("Least 5 countries per y", fontsize = 10)
```

Figure:3.2 CODE FOR EDA PROCESS

CHAPTER 4

ANALYZE ASIAN CITIES

4.1 ASIAN CITIES:

Asia is the most populated continent in the world. The Asian continent is one of the fastest-growing continents in the World, with increasing urbanization and a high growth rate for cities.

The dataset consists of around 44 cities in the world. In this dataset I have chosen only Asian cities for analyzing EDA process.

Fukuoka
Beijing
Bangkok
Jakarta
Seoul
Taipei
Tokyo
Hongkong
Shanghai

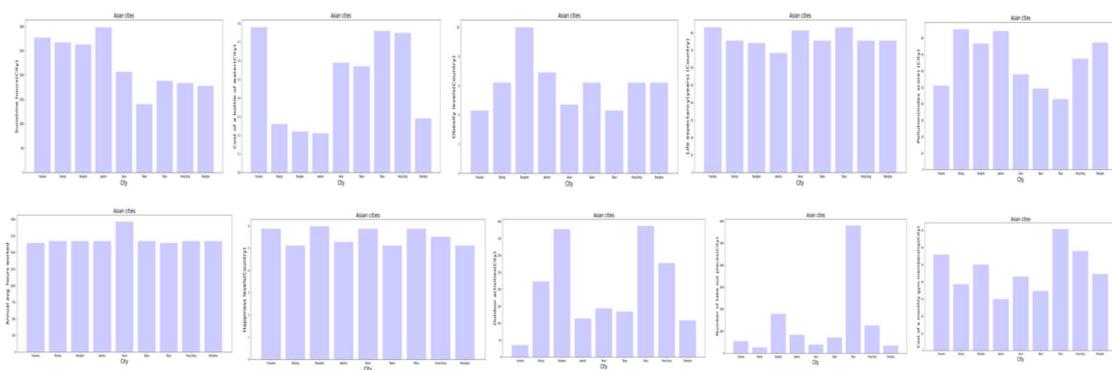


Figure:4.1 PLOTTING OF ASIAN CITIES

CHAPTER 5

FITTING MODELS TO DATA

5.1 K-MEANS CLUSTERING:

K-means clustering is an unsupervised learning algorithm. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own, without the need for any training.

Here we try to propose a methodology in which we will analysis healthiest cities across world and do Exploratory Data Analysis and we will use K-means Clustering algorithm in Unsupervised Learning. Because it consists of different features and continuous data so we use clustering to improve statistical analysis.

In this we create silhouette visualization instance with k-means instance and fit the visualizer. And we have plotted the elbow graph to find the number of clusters. Total we have seven clusters so we have calculated silhouette score for each clusters.

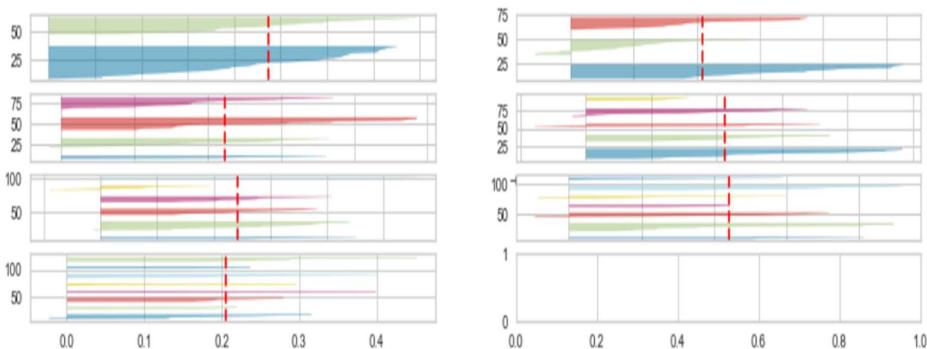


Figure:5.1 SILHOUETTE VISUALIZER

5.2 ELBOW GRAPH:

Elbow method is used to determine the most optimal value of K representing number of clusters in K-means clustering algorithm. Elbow method requires drawing a line plot between SSE (Within-clusters Sum of Squared errors) vs number of clusters. Elbow graph is used for finding the number of clustering in given dataset.

```
In [77]: from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
ssd = []
range_n_clusters = [2, 3, 4, 5, 6, 7, 8]
for num_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=num_clusters, max_iter=50)
    kmeans.fit(Healthy_Clust)

    ssd.append(kmeans.inertia_)

plt.plot(ssd)
plt.grid(linestyle='-', linewidth='1', color='green')
plt.show()
```

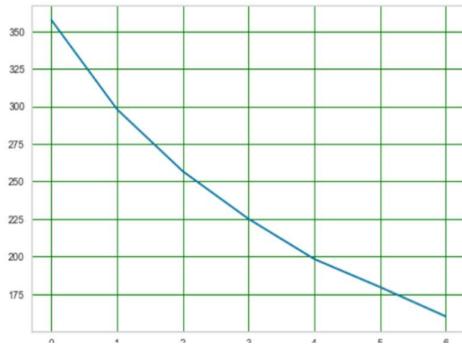


Figure:5.2 ELBOW GRAPH

5.3 SILHOUETTE SCORE:

Silhouette Score is a metric to evaluate the performance of clustering algorithm. It uses compactness of individual clusters and separation amongst clusters to measure an overall representative score of how well our clustering algorithm has performed. Using Silhouette score we find the number of cluster. The number of cluster is five.

```
In [74]: range_n_clusters = [2, 3, 4, 5, 6, 7, 8]
for num_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=num_clusters, max_iter=50)
    kmeans.fit(Healthy_Clust)
    cluster_labels = kmeans.labels_
    silhouette_avg = silhouette_score(Healthy_Clust, cluster_labels)
    print("For n_clusters={0}, the silhouette score is {1}".format(num_clusters, silhouette_avg))

For n_clusters=2, the silhouette score is 0.2365606023230474
For n_clusters=3, the silhouette score is 0.2132276901540575
For n_clusters=4, the silhouette score is 0.2236293384822821
For n_clusters=5, the silhouette score is 0.2149513666813524
For n_clusters=6, the silhouette score is 0.23076178000853914
For n_clusters=7, the silhouette score is 0.24601735205864064
For n_clusters=8, the silhouette score is 0.21935594689699284

In [51]: kmeans = KMeans(n_clusters=5,max_iter=100)
kmeans.fit(Healthy_Clust)

Out[51]: KMeans(max_iter=100, n_clusters=5)
```

Figure :5.3 SILHOUETTE SCORE

5.4 PLOTTING THE CLUSTERS:

After finding the number of clustering using silhouette score . The number of clusters we have found is five. So using this we have plot the clusters to visualize the clusters clearly. We have do clustering for all columns like happiness level, obesity level and so on.

```
In [53]: plt.figure(figsize=(10,5))
sns.barplot(y='Happiness levels(Country)',x='Cluster_ID',data=Healthy)
plt.title('Clusters vs Happiness Levels')
plt.show()
```

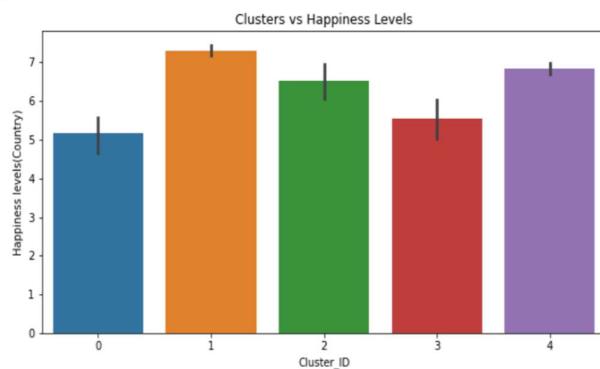


Figure:5.4.1 CLUSTER VS HAPPINESS LEVEL

```
print('Cluster 1:',list(Healthy[Healthy.Cluster_ID==0].index))
Cluster 1: ['Beijing', 'Fukuoka', 'Hong Kong', 'Jakarta', 'Mumbai', 'Seoul', 'Shanghai', 'Taipei']

print('Cluster 2 :',list(Healthy[Healthy.Cluster_ID==1].index))
Cluster 2 : ['Amsterdam', 'Berlin', 'Brussels', 'Copenhagen', 'Dublin', 'Frankfurt', 'Geneva', 'Helsinki', 'Melbourne', 'Stockholm', 'Sydney', 'Toronto', 'Vancouver', 'Vienna', 'Zurich']

print('Cluster 3 :',list(Healthy[Healthy.Cluster_ID==2].index))
Cluster 3 : ['London', 'Milan', 'Paris', 'Tokyo']

print('Cluster 4 :',list(Healthy[Healthy.Cluster_ID==3].index))
Cluster 4 : ['Bangkok', 'Barcelona', 'Buenos Aires', 'Cairo', 'Istanbul', 'Johannesburg', 'Moscow', 'Sao Paulo']

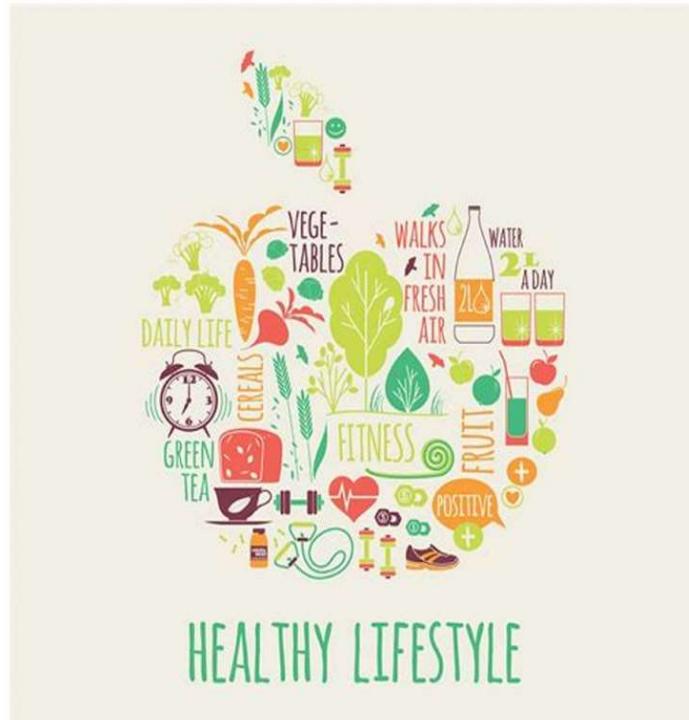
print('Cluster 5 :',list(Healthy[Healthy.Cluster_ID==4].index))
Cluster 5 : ['Boston', 'Chicago', 'Los Angeles', 'Madrid', 'Mexico City', 'New York', 'San Francisco', 'Tel Aviv', 'Washington, D.C.']}
```

Figure: 5.4.2 PRINT THE CLUSTERS

ANALYSIS OF HEALTHY LIFESTYLE ACROSS CITIES

Health as an indicator of the quality of life of the population Indicators of health and morbidity refer to mental groups of healthy and sick people. This obliges us to approach the characterization of a person's lifestyle not only from biological, but also from medical and social positions. An important quality of health as an element of quality is the level of self-determination of behavior, that is, a responsible attitude to the preservation and occurrence of people in their health. In this sense, health as a real human resource, amateurs can dispose of differently with different results.

The main aim of this is to find relationships between various factors that lead to healthy or unhealthy lifestyle in cities and visualize them. Based on given data set we need to cluster healthy life style city in 2021 into certain group. This data contain different feature and overall 44 healthy lifestyle cities in world. And to perform exploratory data analysis for top 5 healthiest cities as well as unhealthiest cities. And perform k-means clustering to improve statistical analysis



Import packages and load the csv file into DataFrame format

```
In [1]:  
import pandas as pd  
file=pd.read_csv("C:/Users/deepika sakthivel/Downloads/Healthy lifestyle across cities.csv")  
df=pd.DataFrame(file)  
df
```

Out[1]:

	City	Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)
0	Amsterdam	1	1858.0	£1.92	20.40%	81.2	30.93	1434.0	7.44	422	1048
1	Sydney	2	2636.0	£1.48	29.00%	82.1	26.86	1712.0	7.22	406	1103
2	Vienna	3	1884.0	£1.94	20.10%	81.0	17.33	1501.0	7.29	132	1008
3	Stockholm	4	1821.0	£1.72	20.60%	81.8	19.63	1452.0	7.35	129	598
4	Copenhagen	5	1630.0	£2.19	19.70%	79.8	21.24	1380.0	7.64	154	523
5	Helsinki	6	1662.0	£1.60	22.20%	80.4	13.08	1540.0	7.80	113	309
6	Fukuoka	7	2769.0	£0.78	4.30%	83.2	NaN	1644.0	5.87	35	539
7	Berlin	8	1626.0	£1.55	22.30%	80.6	39.41	1386.0	7.07	254	1729

	City	Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)
8	Barcelona	9	2591.0	£1.19	23.80%	82.2	65.19	1686.0	6.40	585	2344
9	Vancouver	10	1938.0	£1.08	29.40%	81.7	24.26	1670.0	7.23	218	788
10	Melbourne	11	2363.0	£1.57	29.00%	82.1	25.90	1712.0	7.22	243	813
11	Beijing	12	2671.0	£0.26	6.20%	75.4	85.43	NaN	5.12	223	261
12	Bangkok	13	2624.0	£0.22	10.00%	74.1	76.64	NaN	5.99	377	1796
13	Buenos Aires	14	2525.0	£0.57	28.30%	75.9	52.64	NaN	5.97	246	1435
14	Toronto	15	2066.0	£1.09	29.40%	81.7	37.83	1670.0	7.23	174	1656
15	Madrid	16	2769.0	£1.30	23.80%	82.2	52.68	1686.0	6.40	216	2491
16	Jakarta	17	2983.0	£0.21	6.90%	68.5	84.39	NaN	5.28	114	833
17	Seoul	18	2066.0	£0.59	4.70%	81.3	57.82	1967.0	5.87	144	389
18	Frankfurt	19	1662.0	£1.95	22.30%	80.6	37.78	1386.0	7.07	23	551
19	Geneva	20	NaN	£2.62	19.50%	82.6	27.25	1557.0	7.56	44	444
20	Tel Aviv	21	3311.0	£1.63	26.10%	81.9	47.28	1898.0	7.12	139	420
21	Istanbul	22	2218.0	£0.15	32.10%	74.7	69.49	1832.0	5.13	419	934
22	Cairo	23	3542.0	£0.16	32.00%	70.7	91.74	NaN	4.15	323	250
23	Taipei	24	1405.0	£0.57	6.20%	75.4	49.32	NaN	5.12	134	717
24	Los Angeles	25	3254.0	£1.52	36.20%	78.8	66.07	1779.0	6.94	223	1439
25	Mumbai	26	2584.0	£0.15	3.90%	67.3	82.84	NaN	3.57	187	1183
26	Boston	27	2634.0	£1.39	36.20%	78.8	27.03	1779.0	6.94	88	588
27	Dublin	28	1453.0	£1.40	25.30%	80.5	40.07	1772.0	7.09	159	659
28	Tokyo	29	1877.0	£0.76	4.30%	83.2	42.84	1644.0	5.87	387	5802
29	Chicago	30	2508.0	£1.20	36.20%	78.8	43.33	1779.0	6.94	171	1320
30	Hong Kong	31	1836.0	£0.75	6.20%	75.4	67.46	NaN	5.51	277	1257
31	Shanghai	32	1776.0	£0.29	6.20%	75.4	77.40	NaN	5.12	108	346
32	Brussels	33	1546.0	£2.11	22.10%	80.4	62.67	1583.0	6.86	55	988
33	San Francisco	34	3062.0	£1.60	36.20%	78.8	47.36	1779.0	6.94	242	1031
34	Paris	35	1662.0	£1.95	21.60%	81.8	65.10	1505.0	6.66	331	4363
35	Sao Paulo	36	2003.0	£0.44	22.10%	73.9	79.78	NaN	6.37	158	3355
36	Zurich	37	1566.0	£3.20	19.50%	82.6	17.31	1557.0	7.56	69	538
37	London	38	1633.0	£1.16	27.80%	80.4	58.91	1538.0	7.16	433	6417
38	Johannesburg	39	3124.0	£0.59	28.30%	56.3	61.83	NaN	4.81	194	492
39	Milan	40	1915.0	£1.15	19.90%	82.7	67.19	1718.0	6.38	110	2396
40	Washington, D.C.	41	2528.0	£1.45	36.20%	78.8	39.18	1779.0	6.94	83	744
41	New York	42	2535.0	£1.32	36.20%	78.8	57.36	1779.0	6.94	359	3081
42	Moscow	43	1901.0	£0.41	23.10%	69.5	57.63	1965.0	5.54	322	3206
43	Mexico City	44	2555.0	£0.45	28.90%	76.4	82.78	2137.0	6.46	192	1313

In [2]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44 entries, 0 to 43
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   City             44 non-null    object 
 1   Rank             44 non-null    int64  
 2   Sunshine hours(City) 43 non-null    float64
 3   Cost of a bottle of water(City) 44 non-null    object 
 4   Obesity levels(Country) 44 non-null    object 
 5   Life expectancy(years) (Country) 44 non-null    float64
 6   Pollution(Index score) (City)   43 non-null    float64
```

```

7 Annual avg. hours worked           33 non-null    float64
8 Happiness levels(Country)        44 non-null    float64
9 Outdoor activities(City)          44 non-null    int64
10 Number of take out places(City) 44 non-null    int64
11 Cost of a monthly gym membership(City) 44 non-null    object
dtypes: float64(5), int64(3), object(4)
memory usage: 4.2+ KB

In [3]: df.shape
Out[3]: (44, 12)

In [4]: df.head()

Out[4]:
   City  Rank  Sunshine hours(City)  Cost of a bottle of water(City)  Obesity levels(Country)  Life expectancy(years) (Country)  Pollution(Index score) (City)  Annual avg. hours worked  Happiness levels(Country)  Outdoor activities(City)  Number of take out places(City)  Co me
0  Amsterdam      1       1858.0        £1.92            20.40%             81.2                30.93            1434.0            7.44               422            1048
1  Sydney         2       2636.0        £1.48            29.00%             82.1                26.86            1712.0            7.22               406            1103
2  Vienna          3       1864.0        £1.94            20.10%             81.0                17.33            1501.0            7.29               132            1008
3  Stockholm        4       1821.0        £1.72            20.60%             81.8                19.63            1452.0            7.35               129            598
4  Copenhagen        5       1630.0        £2.19            19.70%             79.8                21.24            1380.0            7.64               154            523

```

we need to replace "£" and "%"

```

In [5]: df['Cost of a bottle of water(City)'] = df['Cost of a bottle of water(City)'].replace(to_replace = '£', value = '', regex = True)
df['Obesity levels(Country)'] = df['Obesity levels(Country)'].replace(to_replace = '%', value = '', regex = True)
df['Cost of a monthly gym membership(City)'] = df['Cost of a monthly gym membership(City)'].replace(to_replace = '€', value = '', regex = True)
df['Cost of a monthly gym membership(City)']

Out[5]:
0    34.90
1    41.66
2    25.74
3    37.31
4    32.53
5    35.23
6    55.87
7    26.11
8    37.80
9    31.84
10   36.89
11   38.62
12   50.83
13   22.45
14   32.64
15   34.54
16   29.94
17   43.03
18   39.81
19   70.00
20   58.31
21   16.97
22   23.25
23   34.76
24   32.00
25   19.54
26   46.27
27   37.35
28   78.82
29   41.14
30   57.95
31   44.68
32   25.34
33   65.13
34   35.93
35   16.87
36   73.11
37   42.71
38   24.28
39   53.49
40   65.99
41   64.66
42   31.40

```

```
43    41.99
Name: Cost of a monthly gym membership(City), dtype: object
```

Data preprocessing

```
In [6]: df['Sunshine hours(City)']=df['Sunshine hours(City)'].map(float)
df['Cost of a bottle of water(City)']=df['Cost of a bottle of water(City)'].map(float)
df['Obesity levels(Country)']=df['Obesity levels(Country)'].map(float)
df['Pollution(Index score) (City)']=df['Pollution(Index score) (City)'].map(float)
df['Annual avg. hours worked']=df['Annual avg. hours worked'].map(float)
df['Cost of a monthly gym membership(City)']=df['Cost of a monthly gym membership(City)'].map(float)

In [7]: df['Annual avg. hours worked'].unique()

Out[7]: array([1434., 1712., 1501., 1452., 1380., 1540., 1644., 1386., 1686.,
       1670., nan, 1967., 1557., 1898., 1832., 1779., 1772., 1583.,
       1505., 1538., 1718., 1965., 2137.])

In [8]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44 entries, 0 to 43
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   City             44 non-null      object 
 1   Rank             44 non-null      int64  
 2   Sunshine hours(City) 43 non-null      float64
 3   Cost of a bottle of water(City) 44 non-null      float64
 4   Obesity levels(Country) 44 non-null      float64
 5   Life expectancy(years) (Country) 44 non-null      float64
 6   Pollution(Index score) (City) 43 non-null      float64
 7   Annual avg. hours worked 33 non-null      float64
 8   Happiness levels(Country) 44 non-null      float64
 9   Outdoor activities(City) 44 non-null      int64  
 10  Number of take out places(City) 44 non-null      int64  
 11  Cost of a monthly gym membership(City) 44 non-null      float64
dtypes: float64(8), int64(3), object(1)
memory usage: 4.2+ KB
```

Let us look out missing values

```
In [9]: df.isna().sum()

Out[9]: City          0
Rank          0
Sunshine hours(City) 1
Cost of a bottle of water(City) 0
Obesity levels(Country) 0
Life expectancy(years) (Country) 0
Pollution(Index score) (City) 1
Annual avg. hours worked 11
Happiness levels(Country) 0
Outdoor activities(City) 0
Number of take out places(City) 0
Cost of a monthly gym membership(City) 0
dtype: int64
```

Fill the missing values

```
In [10]: x=df["Sunshine hours(City)"].mean()
df["Sunshine hours(City)"].fillna(x,inplace=True)

In [11]: x=df["Pollution(Index score) (City)"].mean()
df["Pollution(Index score) (City)"].fillna(x,inplace=True)

In [12]: x=df["Annual avg. hours worked"].mean()
df["Annual avg. hours worked"].fillna(x,inplace=True)

In [13]: df.isna().sum()

Out[13]: City          0
Rank          0
```

```

Sunshine hours(City)          0
Cost of a bottle of water(City) 0
Obesity levels(Country)       0
Life expectancy(years) (Country) 0
Pollution(Index score) (City)   0
Annual avg. hours worked      0
Happiness levels(Country)     0
Outdoor activities(City)       0
Number of take out places(City) 0
Cost of a monthly gym membership(City) 0
dtype: int64

```

Let us check for duplicates

```
In [14]: df.duplicated().sum()
# THERE IS NO DUPLICATED VALUES IN THE DATASET
Out[14]: 0
```

Let us look at the descriptive statistics of the dataset

```
In [15]: df.describe()
```

	Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)	Cc
count	44.000000	44.000000	44.000000	44.000000	44.000000	44.000000	44.000000	44.000000	44.000000	44.000000	44.000000
mean	22.500000	2245.860465	1.173409	21.92500	78.17500	51.122326	1672.909091	6.435000	213.977273	1443.113636	
std	12.845233	560.767190	0.718642	10.19567	5.30437	21.600553	154.957363	0.991202	127.190297	1388.803270	
min	1.000000	1405.000000	0.150000	3.90000	56.30000	13.080000	1380.000000	3.570000	23.000000	250.000000	
25%	11.750000	1809.750000	0.570000	19.50000	75.40000	36.067500	1576.500000	5.870000	125.250000	548.000000	
50%	22.500000	2142.000000	1.195000	22.30000	80.40000	51.881163	1672.909091	6.900000	189.500000	998.000000	
75%	33.250000	2626.500000	1.600000	29.00000	81.80000	66.350000	1773.750000	7.175000	288.250000	1674.250000	
max	44.000000	3542.000000	3.200000	36.20000	83.20000	91.740000	2137.000000	7.800000	585.000000	6417.000000	

General exploratory data analysis

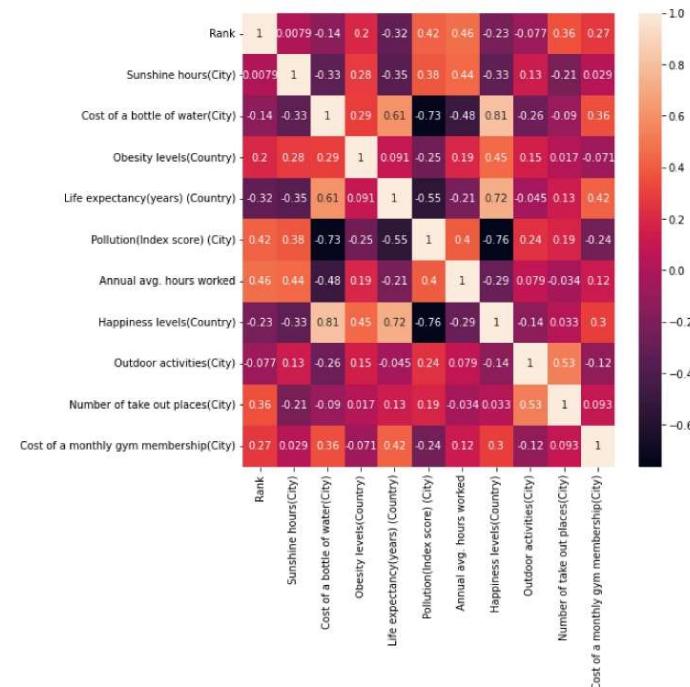
```
In [16]: import matplotlib.pyplot as plt
df.corr()
```

	Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)	Cc
Rank	1.000000	0.007937	-0.141420	0.197486	-0.320034	0.419829	0.463708	-0.231970	-0.076972	0.363	
Sunshine hours(City)	0.007937	1.000000	-0.331239	0.281115	-0.353388	0.383466	0.443596	-0.328979	0.126767	-0.210	
Cost of a bottle of water(City)	-0.141420	-0.331239	1.000000	0.293047	0.612382	-0.732659	-0.479196	0.813159	-0.260297	-0.089	
Obesity levels(Country)	0.197486	0.281115	0.293047	1.000000	0.091071	-0.246972	0.193710	0.446399	0.150171	0.016	
Life expectancy(years) (Country)	-0.320034	-0.353388	0.612382	0.091071	1.000000	-0.552715	-0.212952	0.724587	-0.044864	0.128	
Pollution(Index score) (City)	0.419829	0.383466	-0.732659	-0.246972	-0.552715	1.000000	0.395422	-0.762935	0.236837	0.186	
Annual avg. hours worked	0.463708	0.443596	-0.479196	0.193710	-0.212952	0.395422	1.000000	-0.290647	0.078753	-0.033	
Happiness levels(Country)	-0.231970	-0.328979	0.813159	0.446399	0.724587	-0.762935	-0.290647	1.000000	-0.137612	0.033	
Outdoor activities(City)	-0.076972	0.126767	-0.260297	0.150171	-0.044864	0.236837	0.078753	-0.137612	1.000000	0.528	

	Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number take places(C)
Number of take out places(City)	0.363058	-0.210539	-0.089963	0.016833	0.128163	0.186281	-0.033546	0.033116	0.528200	1.000
Cost of a monthly gym membership(City)	0.272521	0.029244	0.356461	-0.070904	0.417986	-0.241757	0.123009	0.297425	-0.115400	0.092

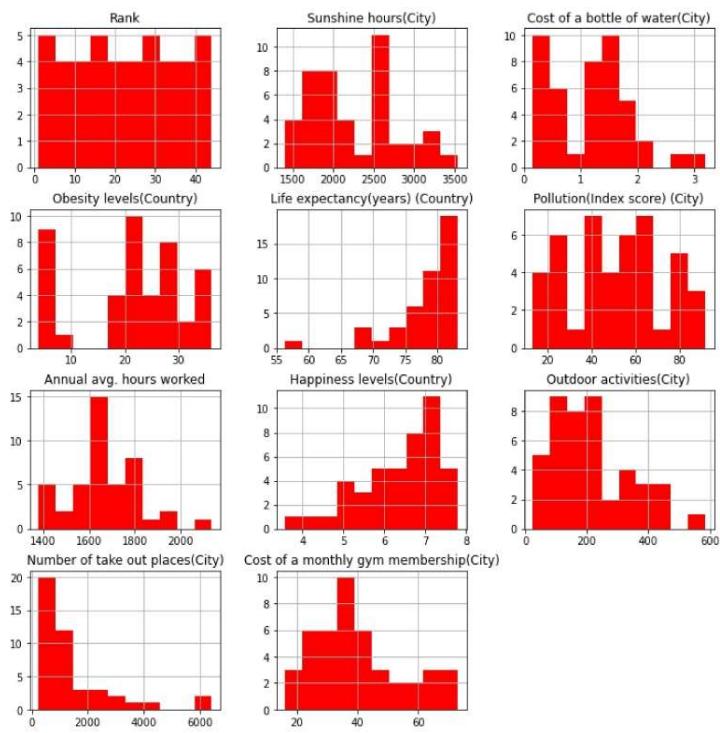
In [17]:

```
import seaborn as sns
plt.figure(figsize=(8, 8))
sns.heatmap(df.corr(), annot=True)
plt.show()
```



In [18]:

```
plotting=df.hist(figsize=(12,12),color="red")
```



In [19]:

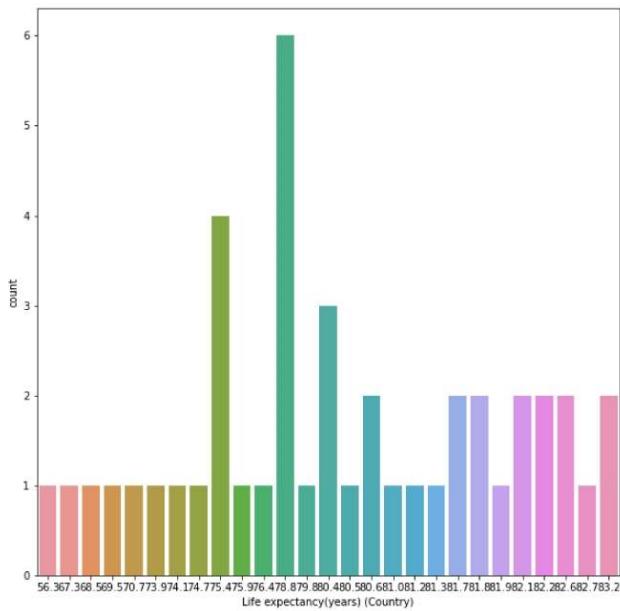
```
import warnings
warnings.filterwarnings("ignore")
```

In [20]:

```
plt.figure(figsize=(10, 10))
sns.countplot(df["Life expectancy(years) (Country)"])
```

Out[20]:

```
<AxesSubplot:xlabel='Life expectancy(years) (Country)', ylabel='count'>
```



Relationship between happiness level and life expectancy

```
In [21]: plt.scatter(df["Happiness levels(Country)"],df['Life expectancy(years) (Country)'],c="g",s=70)
plt.xlabel("Happiness levels");
plt.ylabel("Life expectancy");
plt.title("Relation between Happiness levels and Life expectancy")
```

```
Out[21]: Text(0.5, 1.0, 'Relation between Happiness levels and Life expectancy')
```



This clearly shows that happiness levels leads to the higher levels of expectancy. And clearly shows that all lead healthy life because of happiness level

Exploratory analysis of data for the 5 healthiest cities.

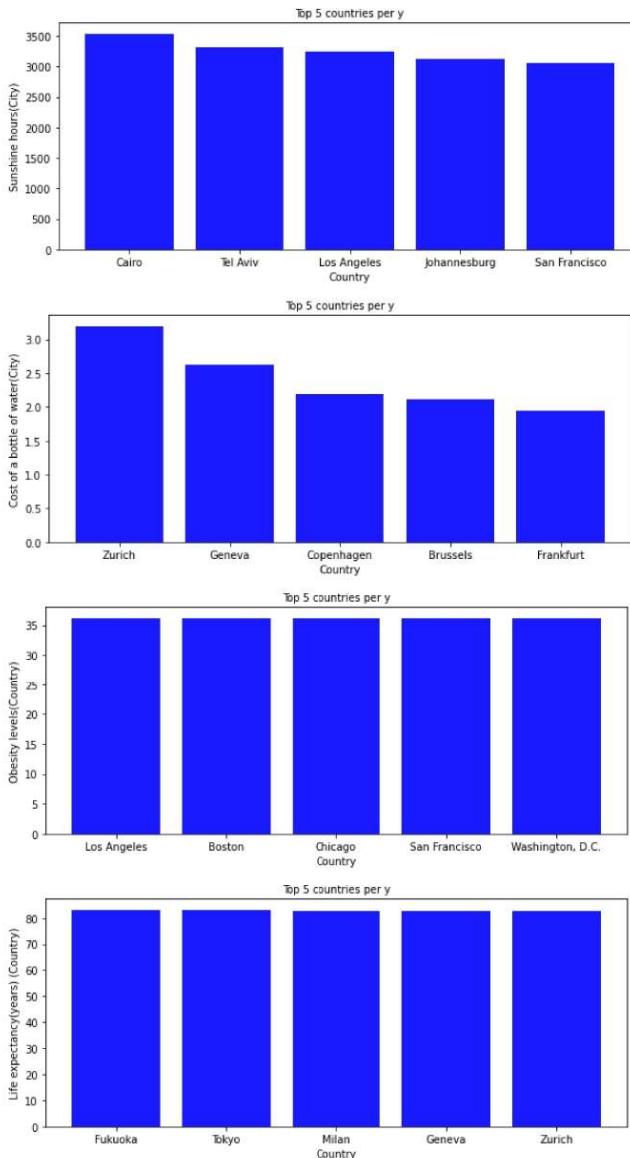
```
In [22]: df.columns
```

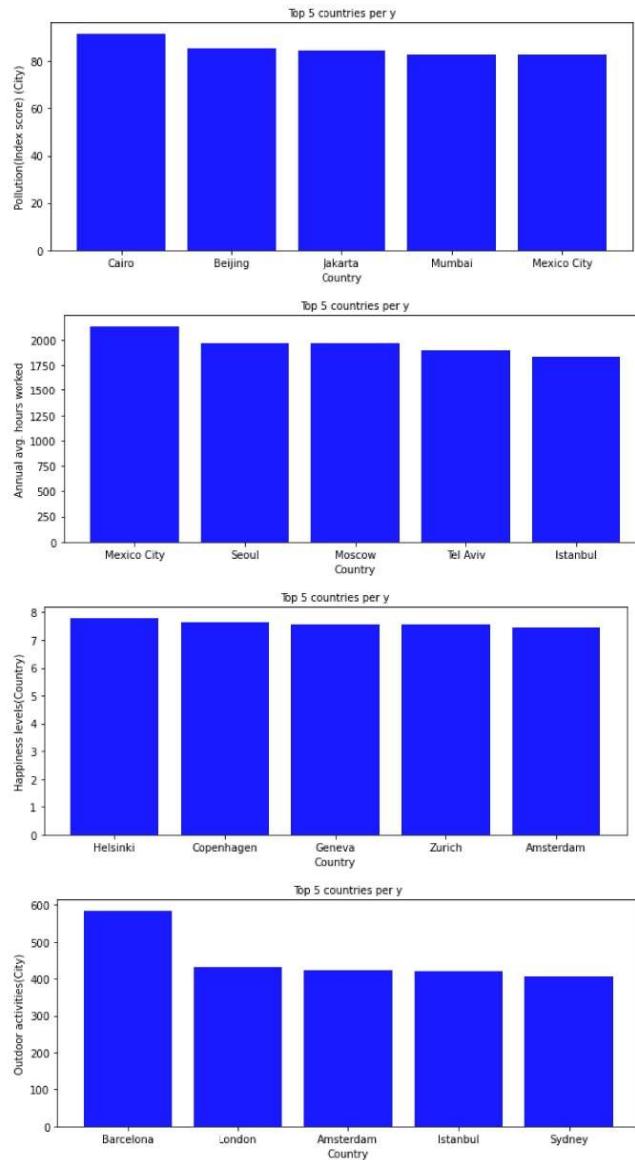
```
Out[22]: Index(['City', 'Rank', 'Sunshine hours(City)', 'Cost of a bottle of water(City)', 'Obesity levels(Country)', 'Life expectancy(years) (Country)', 'Pollution(Index score) (City)', 'Annual avg. hours worked', 'Happiness levels(Country)', 'Outdoor activities(City)', 'Number of take out places(City)', 'Cost of a monthly gym membership(City')], dtype='object')
```

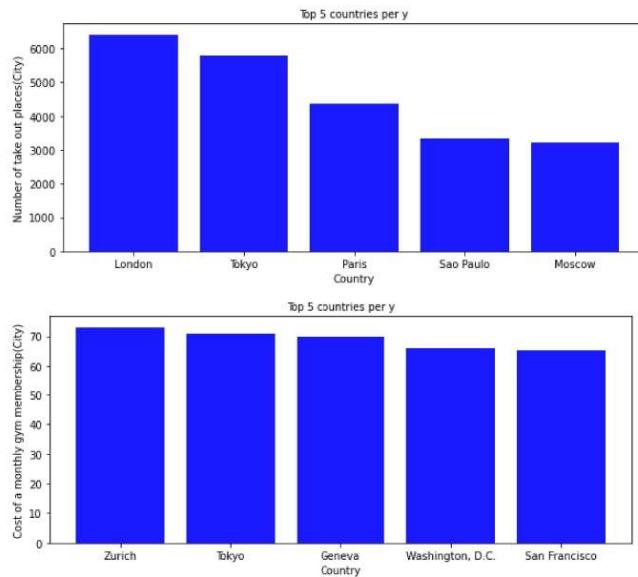
```
In [23]: columns=['Sunshine hours(City)', 'Cost of a bottle of water(City)', 'Obesity levels(Country)', 'Life expectancy(years) (Country)',
```

```
'Pollution(Index score) (City)', 'Annual avg. hours worked',
'Happiness levels(Country)', 'Outdoor activities(City)',
'Number of take out places(City)',
'Cost of a monthly gym membership(City)']
```

```
In [24]:  
import matplotlib.pyplot as plt  
for column in columns:  
    top10 = df.nlargest(5,column)  
    plt.figure(figsize=(10,4))  
    plt.bar(top10['City'], top10[column], color='blue', alpha=0.9)  
    plt.ylabel(column, fontsize = 10)  
    plt.xlabel("Country", fontsize = 10)  
    plt.title("Top 5 countries per " + column, fontsize = 10)
```



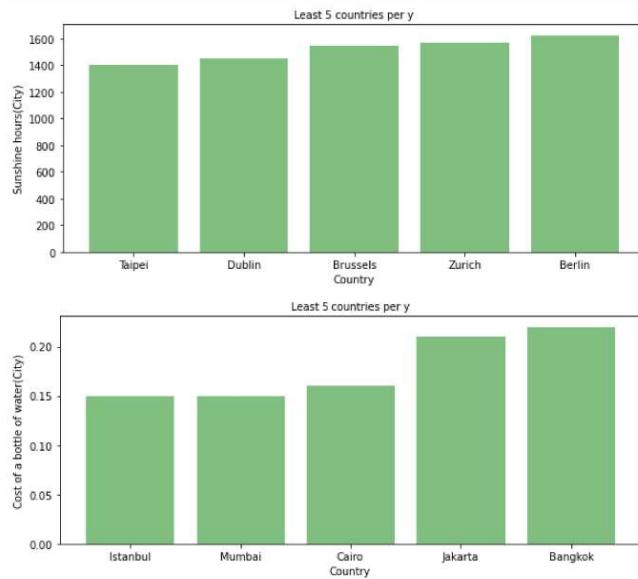


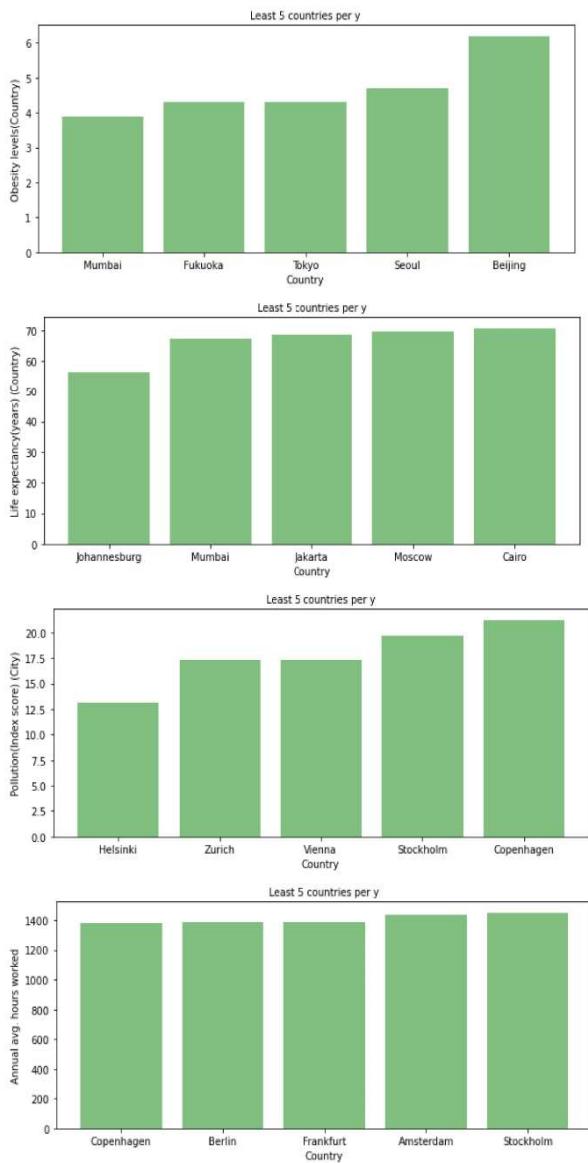


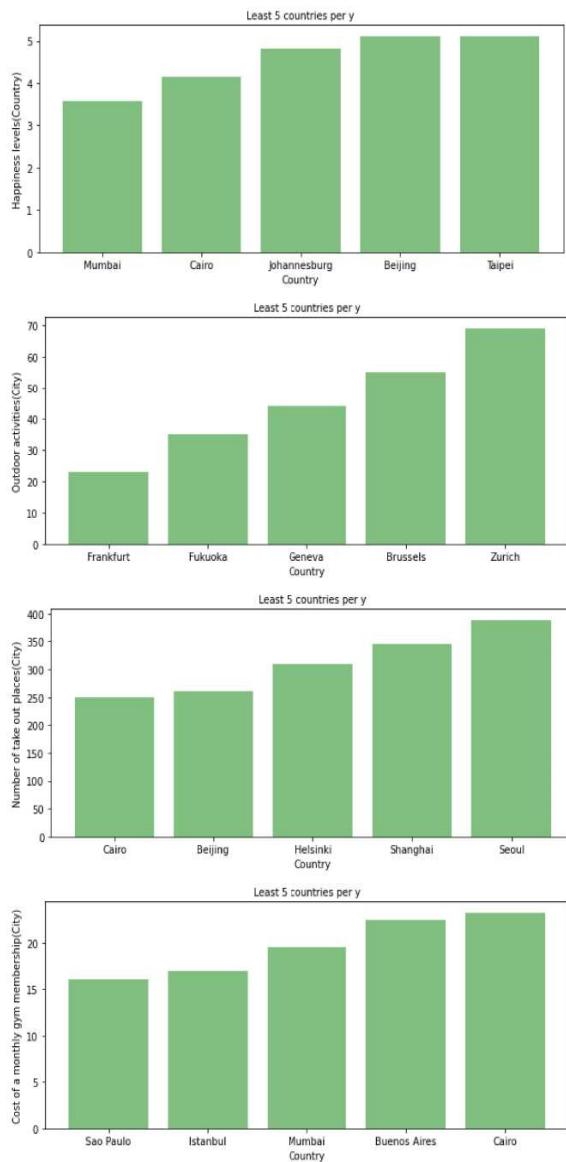
Now we have analyzed the top 5 healthiest cities based on various categories like - Sunshine hours,Cost of Bottle of water,Obesity levels,Life expectancy,pollution,Annual average hours,Happiness level,Outdoor activities,Number of take out places and Cost of monthly gym membership

Exploratory analysis of data for least 5 healthiest cities

```
In [25]: for column in columns:
    top10 = df.nsmallest(5,column)
    plt.figure(figsize=(10,4))
    plt.bar(top10['City'], top10[column], color='green', alpha=0.5)
    plt.ylabel(column, fontsize = 10)
    plt.xlabel("Country", fontsize = 10)
    plt.title("Least 5 countries per y", fontsize = 10)
```

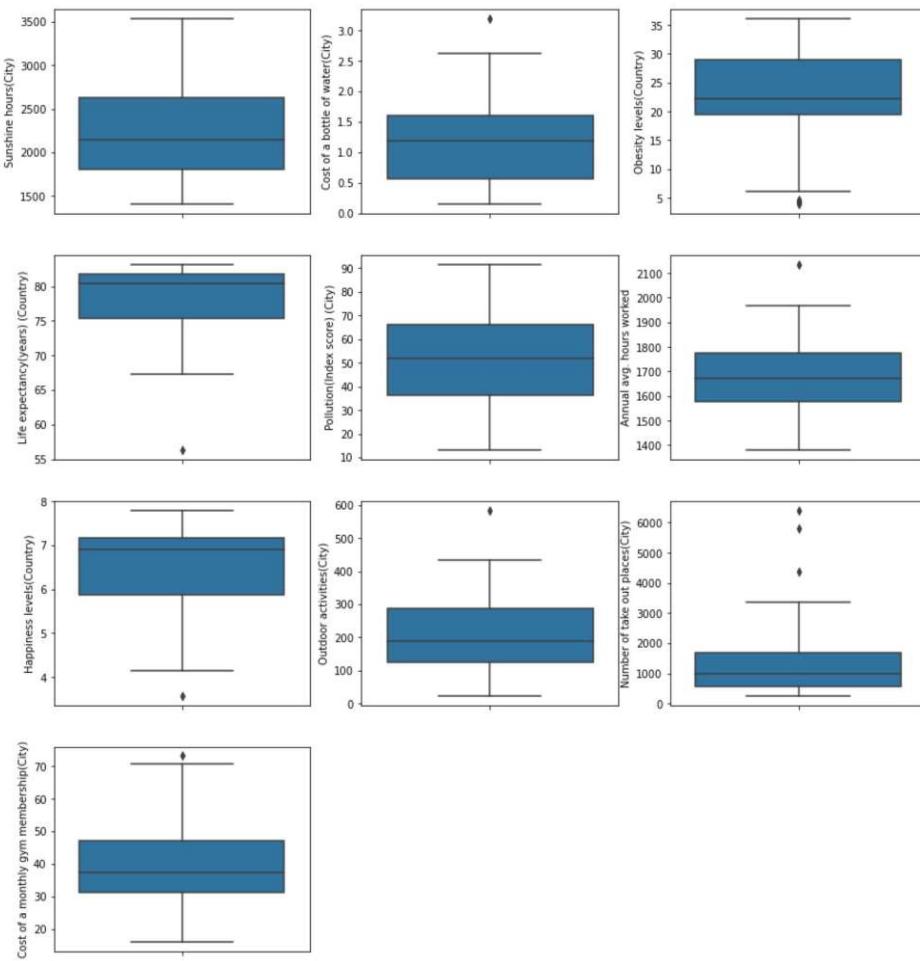






Lets look outliers in data

```
In [26]: import seaborn as sns
i=1
plt.figure(figsize=(15,25))
for feature in columns:
    plt.subplot(6,3,i)
    sns.boxplot(y=df[feature])
    i+=1
```



Let us analyze cities in asia

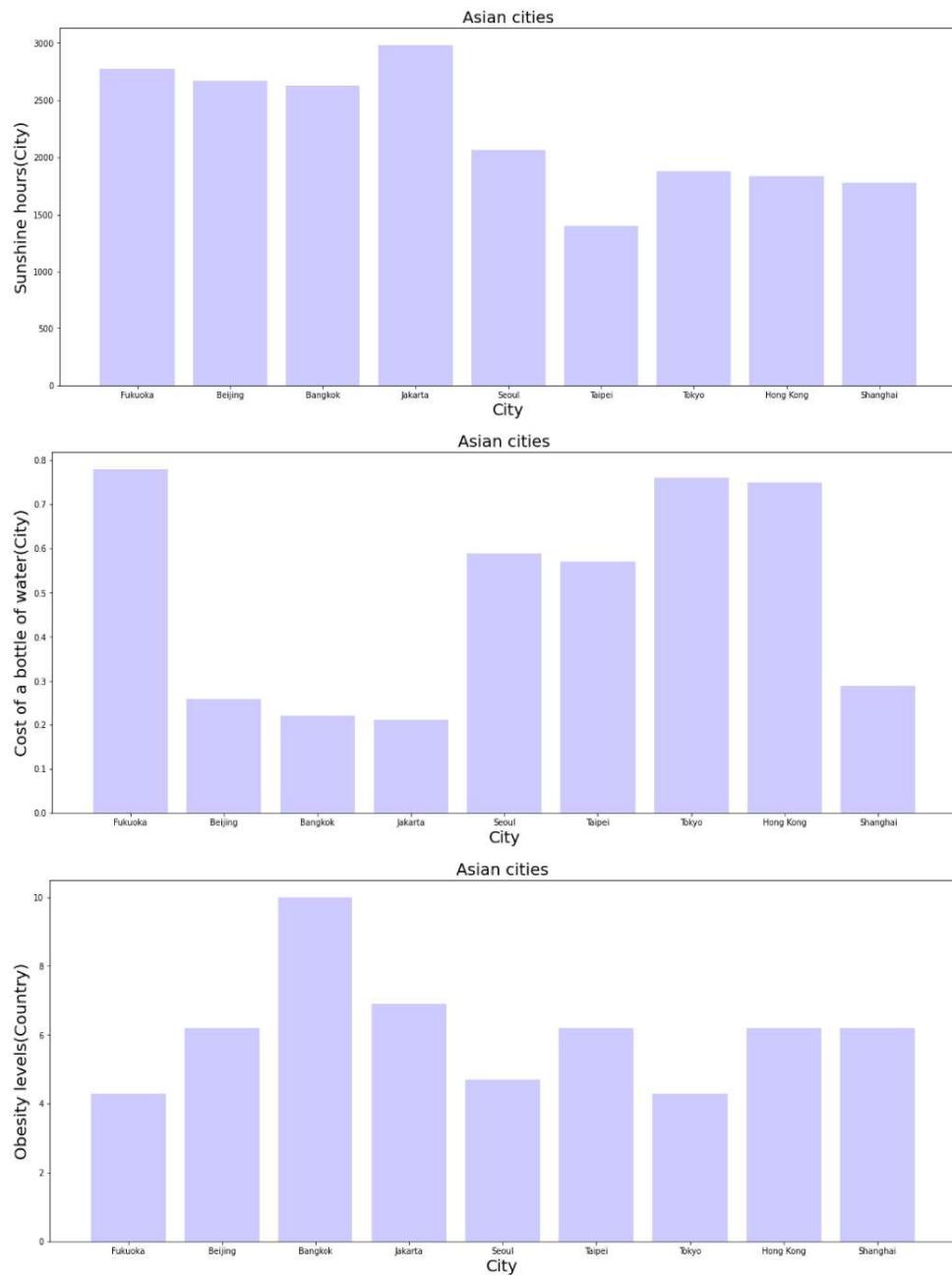
Asia - is Earth's largest and most populous continent, located primarily in the Eastern and Northern Hemispheres. It shares the continental landmass of Eurasia with the continent of Europe, and the continental landmass of Afro-Eurasia with Africa and Europe. Asia covers an area of 44,579,000 square kilometres (17,212,000 sq mi), about 30% of Earth's total land area and 8.7% of the Earth's total surface area. The continent, which has long been home to the majority of the human population, was the site of many of the first civilizations. Its 4.7 billion people constitutes roughly 60% of the world's population.

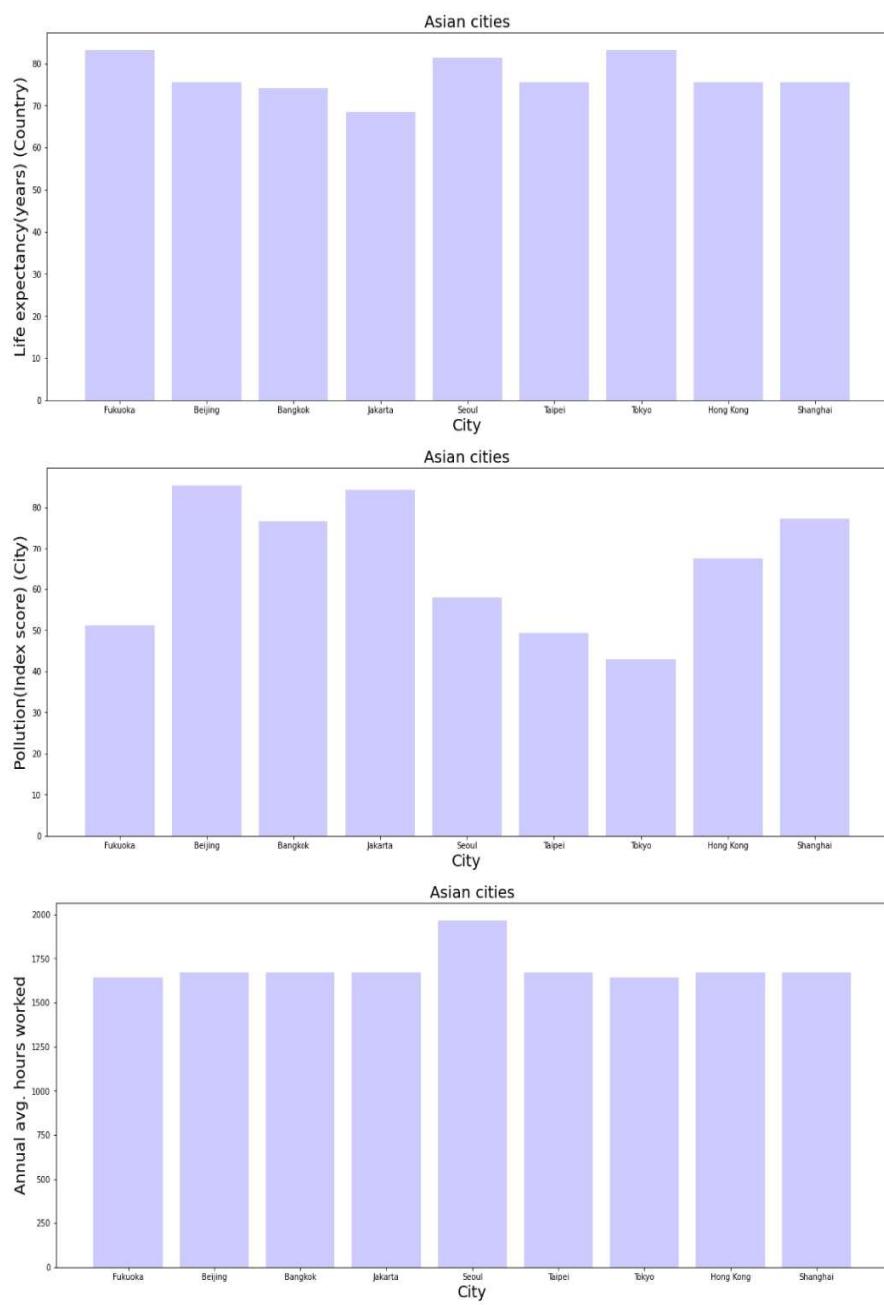


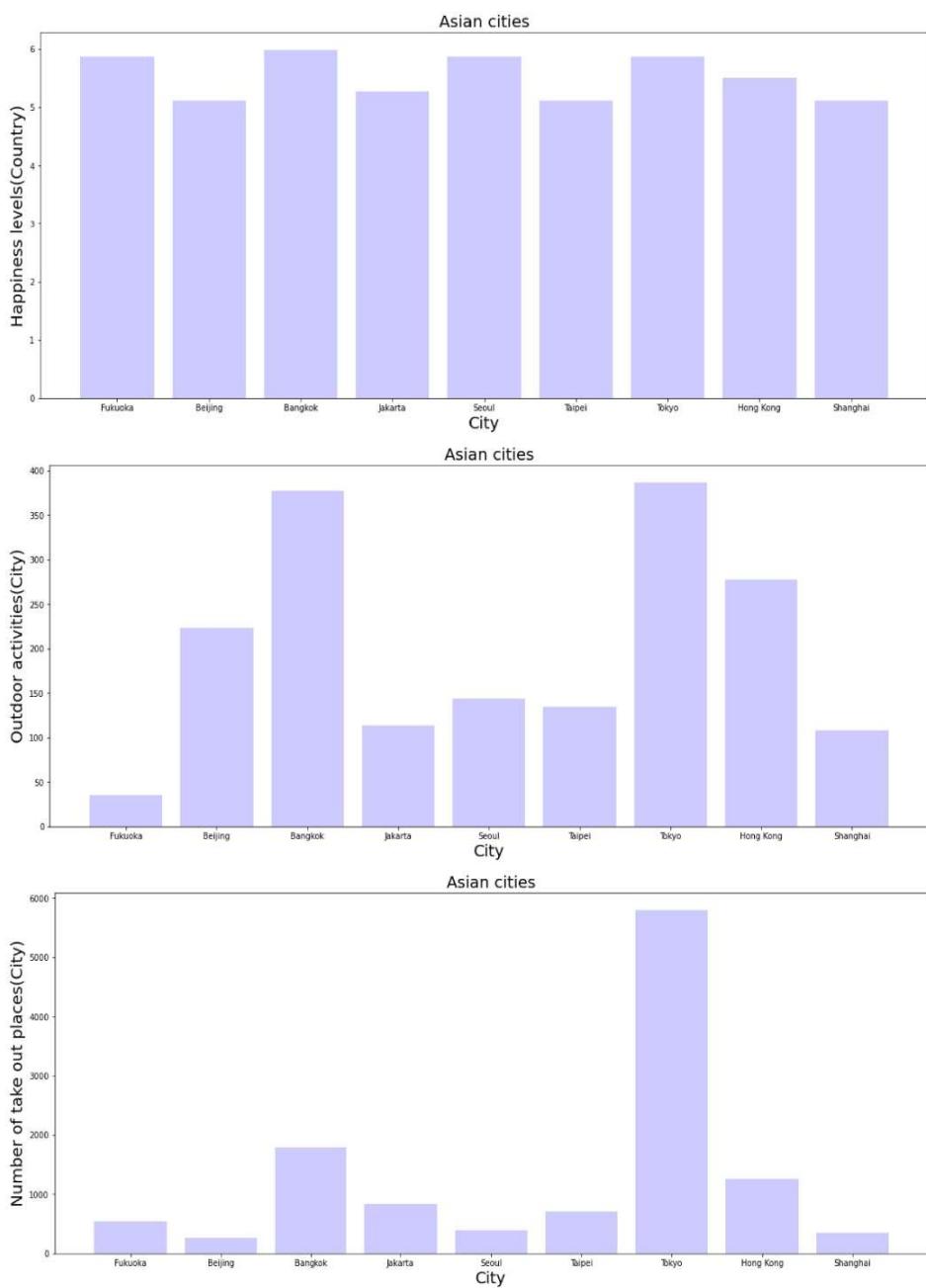
```
In [27]: Asia=df.loc[[6,11,12,16,17,23,28,30,31]]  
Asia
```

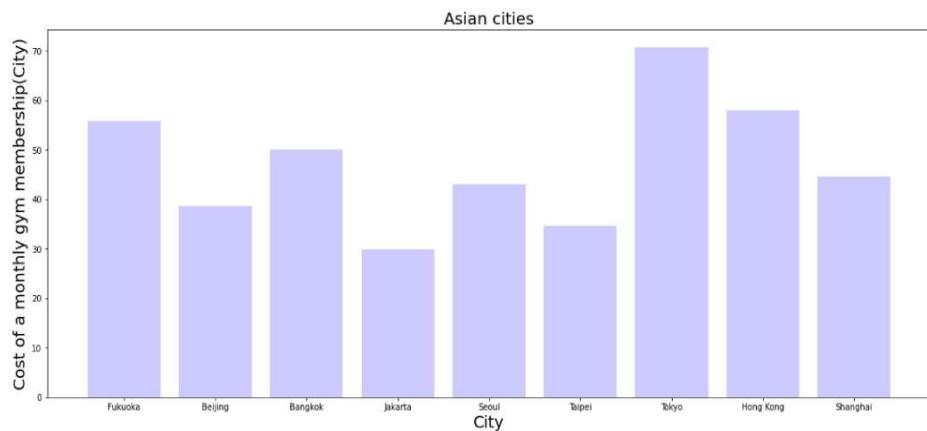
	City	Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)
6	Fukuoka	7	2769.0	0.78	4.3	83.2	51.122326	1644.000000	5.87	35	539
11	Beijing	12	2671.0	0.26	6.2	75.4	85.430000	1672.909091	5.12	223	261
12	Bangkok	13	2624.0	0.22	10.0	74.1	76.640000	1672.909091	5.99	377	1796
16	Jakarta	17	2963.0	0.21	6.9	68.5	84.390000	1672.909091	5.28	114	833
17	Seoul	18	2066.0	0.59	4.7	81.3	57.820000	1967.000000	5.87	144	389
23	Taipei	24	1405.0	0.57	6.2	75.4	49.320000	1672.909091	5.12	134	717
28	Tokyo	29	1877.0	0.76	4.3	83.2	42.840000	1644.000000	5.87	387	5802
30	Hong Kong	31	1836.0	0.75	6.2	75.4	67.460000	1672.909091	5.51	277	1257
31	Shanghai	32	1776.0	0.29	6.2	75.4	77.400000	1672.909091	5.12	108	346

```
In [28]: for column in columns:  
    plt.figure(figsize=(20,8))  
    plt.bar(Asia['City'], Asia[column], color='blue', alpha=0.2)  
    plt.ylabel(column, fontsize = 20)  
    plt.xlabel("City", fontsize = 20)  
    plt.title("Asian cities", fontsize = 20)
```









Clustering cities

```
In [29]: Healthy = df.groupby('City').mean()
Healthy.head()
```

```
Out[29]:
```

Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)	Co
City										
Amsterdam	1.0	1858.0	1.92	20.4	81.2	30.93	1434.000000	7.44	422.0	1048.0
Bangkok	13.0	2624.0	0.22	10.0	74.1	76.64	1672.909091	5.99	377.0	1796.0
Barcelona	9.0	2591.0	1.19	23.8	82.2	65.19	1686.000000	6.40	585.0	2344.0
Beijing	12.0	2671.0	0.26	6.2	75.4	85.43	1672.909091	5.12	223.0	261.0
Berlin	8.0	1626.0	1.55	22.3	80.6	39.41	1386.000000	7.07	254.0	1729.0

```
In [30]: Healthy.head()
```

```
Out[30]:
```

Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)	Co
City										
Amsterdam	1.0	1858.0	1.92	20.4	81.2	30.93	1434.000000	7.44	422.0	1048.0
Bangkok	13.0	2624.0	0.22	10.0	74.1	76.64	1672.909091	5.99	377.0	1796.0
Barcelona	9.0	2591.0	1.19	23.8	82.2	65.19	1686.000000	6.40	585.0	2344.0
Beijing	12.0	2671.0	0.26	6.2	75.4	85.43	1672.909091	5.12	223.0	261.0
Berlin	8.0	1626.0	1.55	22.3	80.6	39.41	1386.000000	7.07	254.0	1729.0

Drop the column rank

```
In [31]: x=df.drop(['Rank'],axis=1)
x
```

```
Out[31]:
```

	City	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)	Co
0	Amsterdam	1858.000000	1.92	20.4	81.2	30.930000	1434.000000	7.44	422	1048	
1	Sydney	2636.000000	1.48	29.0	82.1	26.860000	1712.000000	7.22	406	1103	

	City	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)	CC m
2	Vienna	1884.000000	1.94	20.1	81.0	17.330000	1501.000000	7.29	132	1008	
3	Stockholm	1821.000000	1.72	20.6	81.8	19.630000	1452.000000	7.35	129	598	
4	Copenhagen	1630.000000	2.19	19.7	79.8	21.240000	1380.000000	7.64	154	523	
5	Helsinki	1662.000000	1.60	22.2	80.4	13.080000	1540.000000	7.80	113	309	
6	Fukuoka	2769.000000	0.78	4.3	83.2	51.122326	1644.000000	5.87	35	539	
7	Berlin	1626.000000	1.55	22.3	80.6	39.410000	1386.000000	7.07	254	1729	
8	Barcelona	2591.000000	1.19	23.8	82.2	65.190000	1686.000000	6.40	585	2344	
9	Vancouver	1938.000000	1.08	29.4	81.7	24.260000	1670.000000	7.23	218	788	
10	Melbourne	2363.000000	1.57	29.0	82.1	25.900000	1712.000000	7.22	243	813	
11	Beijing	2671.000000	0.26	6.2	75.4	85.430000	1672.909091	5.12	223	261	
12	Bangkok	2624.000000	0.22	10.0	74.1	76.640000	1672.909091	5.99	377	1796	
13	Buenos Aires	2525.000000	0.57	28.3	75.9	52.640000	1672.909091	5.97	246	1435	
14	Toronto	2066.000000	1.09	29.4	81.7	37.830000	1670.000000	7.23	174	1656	
15	Madrid	2769.000000	1.30	23.8	82.2	52.680000	1686.000000	6.40	216	2491	
16	Jakarta	2983.000000	0.21	6.9	68.5	84.390000	1672.909091	5.28	114	833	
17	Seoul	2066.000000	0.59	4.7	81.3	57.820000	1967.000000	5.87	144	389	
18	Frankfurt	1662.000000	1.95	22.3	80.6	37.780000	1386.000000	7.07	23	551	
19	Geneva	2245.860465	2.62	19.5	82.6	27.250000	1557.000000	7.56	44	444	
20	Tel Aviv	3311.000000	1.63	26.1	81.9	47.280000	1898.000000	7.12	139	420	
21	Istanbul	2218.000000	0.15	32.1	74.7	69.490000	1832.000000	5.13	419	934	
22	Cairo	3542.000000	0.16	32.0	70.7	91.740000	1672.909091	4.15	323	250	
23	Taipei	1405.000000	0.57	6.2	75.4	49.320000	1672.909091	5.12	134	717	
24	Los Angeles	3254.000000	1.52	36.2	78.8	66.070000	1779.000000	6.94	223	1439	
25	Mumbai	2584.000000	0.15	3.9	67.3	82.840000	1672.909091	3.57	187	1183	
26	Boston	2634.000000	1.39	36.2	78.8	27.030000	1779.000000	6.94	88	588	
27	Dublin	1453.000000	1.40	25.3	80.5	40.070000	1772.000000	7.09	159	659	
28	Tokyo	1877.000000	0.76	4.3	83.2	42.840000	1644.000000	5.87	387	5802	
29	Chicago	2508.000000	1.20	36.2	78.8	43.330000	1779.000000	6.94	171	1320	
30	Hong Kong	1836.000000	0.75	6.2	75.4	67.460000	1672.909091	5.51	277	1257	
31	Shanghai	1776.000000	0.29	6.2	75.4	77.400000	1672.909091	5.12	108	346	
32	Brussels	1546.000000	2.11	22.1	80.4	62.670000	1583.000000	6.86	55	988	
33	San Francisco	3062.000000	1.60	36.2	78.8	47.360000	1779.000000	6.94	242	1031	
34	Paris	1662.000000	1.95	21.6	81.8	65.100000	1505.000000	6.66	331	4363	
35	Sao Paulo	2003.000000	0.44	22.1	73.9	79.780000	1672.909091	6.37	158	3355	
36	Zurich	1566.000000	3.20	19.5	82.6	17.310000	1557.000000	7.56	69	538	
37	London	1633.000000	1.16	27.8	80.4	58.910000	1538.000000	7.16	433	6417	
38	Johannesburg	3124.000000	0.59	28.3	56.3	61.830000	1672.909091	4.81	194	492	
39	Milan	1915.000000	1.15	19.9	82.7	67.190000	1718.000000	6.38	110	2396	
40	Washington, D.C.	2528.000000	1.45	36.2	78.8	39.180000	1779.000000	6.94	83	744	
41	New York	2535.000000	1.32	36.2	78.8	57.360000	1779.000000	6.94	359	3081	
42	Moscow	1901.000000	0.41	23.1	69.5	57.630000	1965.000000	5.54	322	3206	
43	Mexico City	2555.000000	0.45	28.9	76.4	82.780000	2137.000000	6.46	192	1313	

Perform K-Means Clustering algorithm to find different clusters

In [32]: `import sklearn`

```
from sklearn import datasets
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score,silhouette_samples
```

```
In [33]: from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
col = Healthy.columns
Healthy_Clust = scaler.fit_transform(Healthy)
```

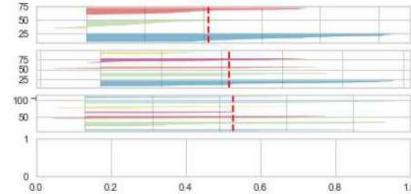
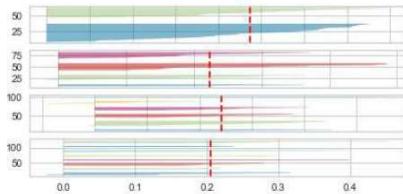
SilhouetteVisualizer

```
In [34]: import yellowbrick
import matplotlib.pyplot as plt
from yellowbrick.cluster import SilhouetteVisualizer

fig,ax=plt.subplots(4,2,figsize=(15,3))
for i in[2,3,4,5,6,7,8]:
    ...
    Create KMeans instance for different number of clusters
    ...
    km=kMeans(n_clusters=i,init='k-means++',n_init=10,max_iter=100,random_state=42)
    q,mod=divmod(i,2)

    #create silhouettevisualization instance with kmeans instance
    #fit the visualizer

    visualizer=SilhouetteVisualizer(km,colors='yellowbrick',ax=ax[q-1][mod])
    visualizer.fit(Healthy_Clust)
```

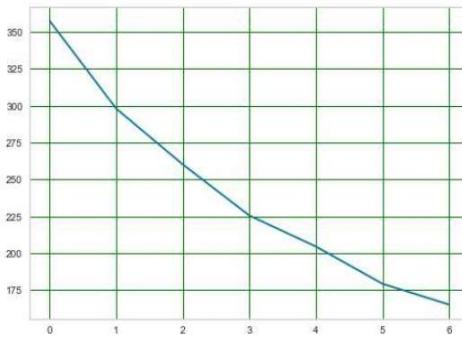


Elbow graph

```
In [35]: from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
ssd = []
range_n_clusters = [2, 3, 4, 5, 6, 7, 8]
for num_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=num_clusters, max_iter=50)
    kmeans.fit(Healthy_Clust)

    ssd.append(kmeans.inertia_)

plt.plot(ssd)
plt.grid(linestyle='-', linewidth='1', color='green')
plt.show()
```



Silhouette Score

```
In [36]: range_n_clusters = [2, 3, 4, 5, 6, 7, 8]
for num_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=num_clusters, max_iter=50)
    kmeans.fit(Healthy_Clust)
    cluster_labels = kmeans.labels_
    silhouette_avg = silhouette_score(Healthy_Clust, cluster_labels)
    print("For n_clusters={0}, the silhouette score is {1:.3f}.".format(num_clusters, silhouette_avg))

For n_clusters=2, the silhouette score is 0.2365000232304474
For n_clusters=3, the silhouette score is 0.2132276901540575
For n_clusters=4, the silhouette score is 0.22664721065505794
For n_clusters=5, the silhouette score is 0.20583280427627795
For n_clusters=6, the silhouette score is 0.21098734711124506
For n_clusters=7, the silhouette score is 0.24579948176404862
For n_clusters=8, the silhouette score is 0.21455601779284217
```

```
In [37]: kmeans = KMeans(n_clusters=5,max_iter=100)
kmeans.fit(Healthy_Clust)
```

```
Out[37]: KMeans
KMeans(max_iter=100, n_clusters=5)
```

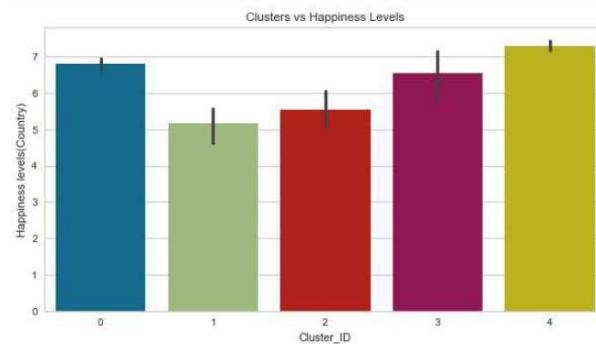
```
In [38]: Healthy['Cluster_ID'] = kmeans.labels_
Healthy.head()
```

```
Out[38]:
```

Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)	Co
City										
Amsterdam	1.0	1858.0	1.92	20.4	81.2	30.93	1434.000000	7.44	422.0	1048.0
Bangkok	13.0	2624.0	0.22	10.0	74.1	76.64	1672.909091	5.99	377.0	1796.0
Barcelona	9.0	2591.0	1.19	23.8	82.2	65.19	1686.000000	6.40	585.0	2344.0
Beijing	12.0	2671.0	0.26	6.2	75.4	85.43	1672.909091	5.12	223.0	261.0
Berlin	8.0	1626.0	1.55	22.3	80.6	39.41	1386.000000	7.07	254.0	1729.0

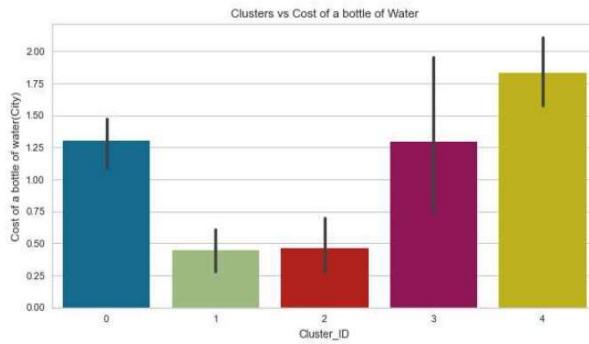
Plotting the clusters

```
In [39]: plt.figure(figsize=(10,5))
sns.barplot(y='Happiness levels(Country)',x='Cluster_ID',data=Healthy)
plt.title('Clusters vs Happiness Levels')
plt.show()
```

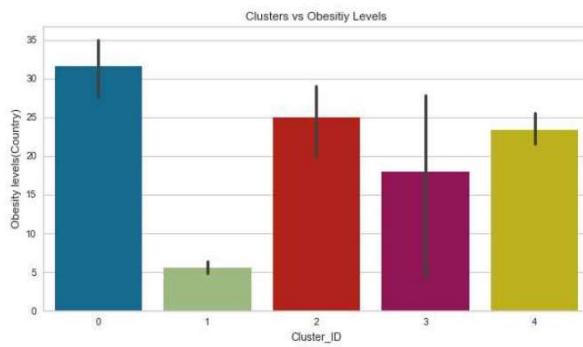


```
In [40]:
```

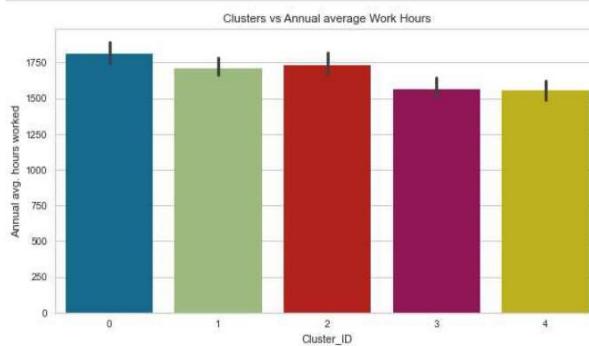
```
plt.figure(figsize=(10,5))
sns.barplot(y='Cost of a bottle of water(City)',x='Cluster_ID',data=Healthy)
plt.title('Clusters vs Cost of a bottle of Water')
plt.show()
```



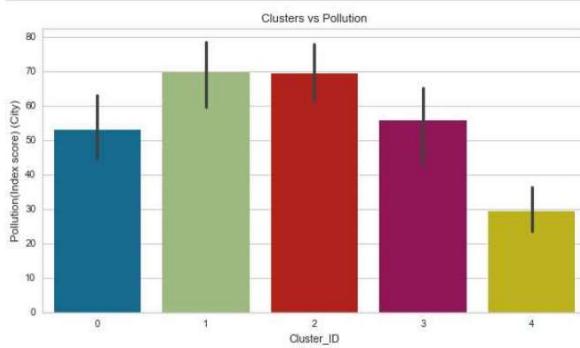
```
In [41]: plt.figure(figsize=(10,5))
sns.barplot(y='Obesity levels(Country)',x='Cluster_ID',data=Healthy)
plt.title('Clusters vs Obesity Levels')
plt.show()
```



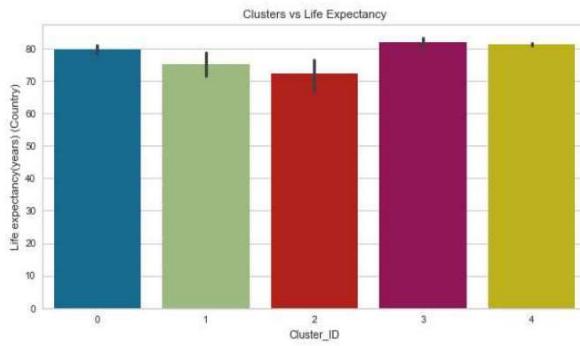
```
In [42]: plt.figure(figsize=(10,5))
sns.barplot(y='Annual avg. hours worked',x='Cluster_ID',data=Healthy)
plt.title('Clusters vs Annual average Work Hours')
plt.show()
```



```
In [43]: plt.figure(figsize=(10,5))
sns.barplot(y='Pollution(Index score) (City)',x='Cluster_ID',data=Healthy)
plt.title('Clusters vs Pollution')
plt.show()
```



```
In [44]: plt.figure(figsize=(10,5))
sns.barplot(y='Life expectancy(years) (Country)',x='Cluster_ID',data=Healthy)
plt.title('Clusters vs Life Expectancy')
plt.show()
```



Printing the clusters

```
In [45]: print('Cluster 1:',list(Healthy[Healthy.Cluster_ID==0].index))
Cluster 1: ['Boston', 'Chicago', 'Los Angeles', 'Madrid', 'Mexico City', 'Milan', 'New York', 'San Francisco', 'Tel Aviv', 'Washington, D.C.']

In [46]: print('Cluster 2 :',list(Healthy[Healthy.Cluster_ID==1].index))
Cluster 2 : ['Beijing', 'Fukuoka', 'Hong Kong', 'Jakarta', 'Mumbai', 'Seoul', 'Shanghai', 'Taipei']

In [47]: print('Cluster 3 :',list(Healthy[Healthy.Cluster_ID==2].index))
Cluster 3 : ['Bangkok', 'Barcelona', 'Buenos Aires', 'Cairo', 'Istanbul', 'Johannesburg', 'Moscow', 'Sao Paulo']

In [51]: print('Cluster 4 :',list(Healthy[Healthy.Cluster_ID==3].index))
Cluster 4 : ['London', 'Paris', 'Tokyo']

In [49]: print('Cluster 5 :',list(Healthy[Healthy.Cluster_ID==4].index))
Cluster 5 : ['Amsterdam', 'Berlin', 'Brussels', 'Copenhagen', 'Dublin', 'Frankfurt', 'Geneva', 'Helsinki', 'Melbourne', 'Stockholm', 'Sydney', 'Toronto', 'Vancouver', 'Vienna', 'Zurich']
```

From this project we have successfully clustered cities based on healthy lifestyle using k-means clustering algorithm. Because it consists of different features and continuous data so we use clustering to improve statistical analysis. The number of clusters we have found is five. A healthier lifestyle is consistently associated with a longer life expectancy across various individual risks and irrespective of the presence of multiple long-term medical conditions. The major factor behind longer life expectancy is happiness. Higher levels of happiness leads to higher life expectancy rates. The main aim

is to find relationships between various factors that lead to healthy or unhealthy lifestyle in cities and visualize them.

In []:

CHAPTER 6

CONCLUSION

I have evaluated the machine learning baselines using K-means algorithm, under unsupervised learning. It allow us to cluster the data into different groups in a convenient way to discover the categories of groups in the unlabeled dataset on its own, without the need for any training. In our dataset consists of continuous variable there is no categorical variable. So only we use k-means clustering algorithm, to find the clusters of cities in healthy lifestyle across cities. I have used elbow graph and silhouette score for finding the number of clusters. The clusters formed is five in healthy lifestyle dataset. Through this we have find the clusters for each columns like happiness level, obesity level and so on. This will clearly shows that healthy life style across cities dataset consists of these clusters. In future, there is no healthy lifestyle, because now a days the peoples are very lazy to do any kind of activities. So, it will be very useful for us to find the healthiest cities across the world.

CHAPTER 7

REFERENCE

- [1] Barnett K, Mercer SW, Norbury M, Watt G, Wyke S, Guthrie B. Epidemiology of multimorbidity and implications for health care, research, and medical education: a cross-sectional study. Lancet. 2012;380(9836):37–43.
- [2] Li K, Husing A, kaaks R. Lifestyle risk factors and residual life expectancy at age 40: a German cohort study. BMC Med. 2014;12(1):59.
- [3] Manuel DG, Perez R, Bennett C, Rosella L, Taljaard M, Roberts M, et al. Seven more years: the impact of smoking, alcohol, diet, physical activity and stress on health and life expectancy in Ontario: Toronto. Institute for Clinical Evaluative Sciences and Public Health Ontario. 2012. [cited 2020 Aug 31].
- [4] Pan XF, Li Y, Franco OH, Yuan JM, Pan A, Koh WP. Impact of combined lifestyle factors on all-cause and cause-specific mortality and life expectancy in Chinese: the Singapore Chinese Health Study. J Gerontol A Biol Sci Med Sci. 2019.
- [5] O'Doherty MG, Cairns K, O'Neill V, Lamrock F, Jørgensen T, Brenner H, et al. Effect of major lifestyle risk factors, independent and jointly, on life expectancy with and without cardiovascular disease: results from the Consortium on Health and Ageing Network of Cohorts in Europe and the United States (CHANCES). Eur J Epidemiol. 2016;31(5):455–68.