Name : Ayush Panchal

Roll Number : P24DS013

In [1]:
```
import pandas as pd
import numpy as np
```

1. Display how many unique areas' average temperature data is provided.

In [2]:
```
data = pd.read_csv("Surface_Temperature.csv")
data.head()
```

Out[2]:

	Date	AverageTemperature	AverageTemperatureUncertainty	City	Country	Latitude	Longitude
0	1849-01-01	26.704	1.435	Abidjan	Côte D'Ivoire	5.63N	3.23W
1	1849-02-01	27.434	1.362	Abidjan	Côte D'Ivoire	5.63N	3.23W
2	1849-03-01	NaN	NaN	Abidjan	Côte D'Ivoire	5.63N	3.23W
3	1849-04-01	26.140	1.387	Abidjan	Côte D'Ivoire	5.63N	3.23W
4	1849-05-01	25.427	1.200	Abidjan	Côte D'Ivoire	5.63N	3.23W

In [5]:
```
data.shape
```

Out[5]:
```
(219575, 7)
```

In [4]:
```
num_of_unique_areas = len(data["City"].unique())

f"Number of unique areas of which temperature data is provided : {num_of_unique_areas}"
```

Out[4]:
```
'Number of unique areas of which temperature data is provided : 100'
```

2.Encode the area identification fields with the Area ordering displayed as in Q. 1.

In [6]:
```
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()

data["Label_encoded_city"] = label_encoder.fit_transform(data["City"])

data.head()
```

Out[6]:

	Date	AverageTemperature	AverageTemperatureUncertainty	City	Country	Latitude	Longitude	Label_encoded_city
0	1849-01-01	26.704	1.435	Abidjan	Côte D'Ivoire	5.63N	3.23W	0
1	1849-02-01	27.434	1.362	Abidjan	Côte D'Ivoire	5.63N	3.23W	0
2	1849-03-01	NaN	NaN	Abidjan	Côte D'Ivoire	5.63N	3.23W	0
3	1849-04-01	26.140	1.387	Abidjan	Côte D'Ivoire	5.63N	3.23W	0
4	1849-05-01	25.427	1.200	Abidjan	Côte D'Ivoire	5.63N	3.23W	0

3. Display the number of unique rows with the same average temperature value, if any

In [10]:
```
avg_temp_col = data["AverageTemperature"]
avg_temp_col = avg_temp_col.dropna()
```

In [11]:
```
avg_temp_col.shape
```

Out[11]:
```
(191994,)
```

In [12]:
```
len(data["AverageTemperature"])
```

Out[12]: 219575

In [13]:
```
avg_temp_col = avg_temp_col.drop_duplicates()
avg_temp_col.shape
```

Out[13]: (49920,)

In [15]:
```
f"Number of unique rows with the same average temperature value : {avg_temp_col.shape[0]}"
```

Out[15]: 'Number of unique rows with the same average temperature value : 49920'

4. Display for each unique area, display the descriptive statistics.

In [16]:
```
unique_cities = data["City"].unique()
unique_cities
```

Out[16]:
```
array(['Abidjan', 'Addis Abeba', 'Ahmadabad', 'Aleppo', 'Alexandria',
       'Ankara', 'Baghdad', 'Bangalore', 'Bangkok', 'Belo Horizonte',
       'Berlin', 'Bogotá', 'Bombay', 'Brasília', 'Cairo', 'Calcutta',
       'Cali', 'Cape Town', 'Casablanca', 'Changchun', 'Chengdu',
       'Chicago', 'Chongqing', 'Dakar', 'Dalian', 'Dar Es Salaam',
       'Delhi', 'Dhaka', 'Durban', 'Faisalabad', 'Fortaleza', 'Gizeh',
       'Guangzhou', 'Harare', 'Harbin', 'Ho Chi Minh City', 'Hyderabad',
       'Ibadan', 'Istanbul', 'Izmir', 'Jaipur', 'Jakarta', 'Jiddah',
       'Jinan', 'Kabul', 'Kano', 'Kanpur', 'Karachi', 'Kiev', 'Kinshasa',
       'Lagos', 'Lahore', 'Lakhnau', 'Lima', 'London', 'Los Angeles',
       'Luanda', 'Madras', 'Madrid', 'Manila', 'Mashhad', 'Melbourne',
       'Mexico', 'Mogadishu', 'Montreal', 'Moscow', 'Nagoya', 'Nagpur',
       'Nairobi', 'Nanjing', 'New Delhi', 'New York', 'Paris', 'Peking',
       'Pune', 'Rangoon', 'Rio De Janeiro', 'Riyadh', 'Rome',
       'Saint Petersburg', 'Salvador', 'Santiago', 'Santo Domingo',
       'Seoul', 'Shanghai', 'Shenyang', 'Singapore', 'Surabaya', 'Surat',
       'Sydney', 'São Paulo', 'Taipei', 'Taiyuan', 'Tangshan', 'Tianjin',
       'Tokyo', 'Toronto', 'Umm Durman', 'Wuhan', 'Xian'], dtype=object)
```

In [18]:
```
data.groupby(["City"]).describe().T
```

Out[18]:

	City	Abidjan	Addis Abeba	Ahmadabad	Aleppo	Alexandria	Ankara	Baghdad	Ban
AverageTemperature	**count**	1596.000000	1498.000000	2060.000000	2107.000000	2133.000000	2146.000000	2081.000000	2092.0
	mean	26.163751	17.519688	26.547694	17.431182	20.347205	10.398004	22.600975	24.8
	std	1.399779	1.215365	4.247763	8.512489	4.558519	8.142198	9.189700	1.8
	min	22.363000	14.528000	17.041000	1.086000	11.253000	-6.195000	4.236000	20.2
	25%	25.108250	16.568000	22.934250	9.420500	15.988000	3.073000	14.007000	23.5
	50%	26.262500	17.288500	27.253000	17.628000	20.628000	10.602000	23.031000	24.5
	75%	27.182250	18.453500	29.585500	25.755500	24.651000	17.905750	31.613000	26.1
	max	29.923000	21.223000	35.419000	32.629000	28.806000	26.044000	38.283000	29.6
AverageTemperatureUncertainty	**count**	1596.000000	1498.000000	2060.000000	2107.000000	2133.000000	2146.000000	2081.000000	2092.0
	mean	0.678107	0.827874	0.847150	0.905100	0.825722	0.895382	1.016012	0.7
	std	0.479361	0.480745	0.672429	0.762853	0.707597	0.805486	0.717917	0.7
	min	0.110000	0.133000	0.106000	0.101000	0.102000	0.110000	0.098000	0.0
	25%	0.301000	0.461250	0.368000	0.364000	0.329000	0.355250	0.443000	0.3
	50%	0.503000	0.719000	0.545000	0.596000	0.489000	0.579500	0.749000	0.4
	75%	0.975750	1.095750	1.339000	1.298500	1.274000	1.187500	1.495000	1.0
	max	3.032000	3.841000	5.260000	5.450000	5.001000	6.146000	4.752000	6.2
Label_encoded_city	**count**	1968.000000	1956.000000	2352.000000	2352.000000	2352.000000	2352.000000	2328.000000	2352.0
	mean	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.0
	std	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
	min	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.0
	25%	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.0

City	Abidjan	Addis Abeba	Ahmadabad	Aleppo	Alexandria	Ankara	Baghdad	Ban
50%	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.0
75%	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.0
max	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.0

24 rows × 100 columns

5. For ALL each unique area, display their distances from each other.

In [ ]:

6. For ALL each unique area and each year, display the number of missing values, if any.

In [22]:
```
data["year"] = data["Date"].apply(lambda x: int(x[:4]))
data.head()
```

Out[22]:

	Date	AverageTemperature	AverageTemperatureUncertainty	City	Country	Latitude	Longitude	Label_encoded_city	year
0	1849-01-01	26.704	1.435	Abidjan	Côte D'Ivoire	5.63N	3.23W	0	1849
1	1849-02-01	27.434	1.362	Abidjan	Côte D'Ivoire	5.63N	3.23W	0	1849
2	1849-03-01	NaN	NaN	Abidjan	Côte D'Ivoire	5.63N	3.23W	0	1849
3	1849-04-01	26.140	1.387	Abidjan	Côte D'Ivoire	5.63N	3.23W	0	1849
4	1849-05-01	25.427	1.200	Abidjan	Côte D'Ivoire	5.63N	3.23W	0	1849

In [24]:
```
data.groupby(["year"]).count().nunique()
```

Out[24]:
```
Date                           25
AverageTemperature            106
AverageTemperatureUncertainty 106
City                           25
Country                        25
Latitude                       25
Longitude                      25
Label_encoded_city             25
dtype: int64
```

In [25]:
```
data.groupby(["City"]).count().nunique()
```

Out[25]:
```
Date                           22
AverageTemperature             84
AverageTemperatureUncertainty  84
Country                        22
Latitude                       22
Longitude                      22
Label_encoded_city             22
year                           22
dtype: int64
```

7. Display 2 different visualization plots of average temperature to compare them for years 2001 and 2002 for 2 unique areas with less distance.

In [44]:
```
# for year 2001, 2002

data_2001 = data[data["year"] == 2001]
data_2002 = data[data["year"] == 2002]
year_data_combined = data_2001.merge(data_2002, how="outer")
year_data_combined.shape
```

Out[44]: (2400, 9)

In [45]:
```
city_ahm = data[data["City"] == "Ahmadabad"]
city_delhi = data[data["City"] == "Delhi"]
city_merged = city_ahm.merge(city_delhi, how="outer")
city_merged.head()
```

Out[45]:

	Date	AverageTemperature	AverageTemperatureUncertainty	City	Country	Latitude	Longitude	Label_encoded_city	year
0	1817-01-01	13.439	3.860	Delhi	India	28.13N	77.27E	26	1817
1	1817-01-01	18.439	3.923	Ahmadabad	India	23.31N	72.52E	2	1817
2	1817-02-01	17.130	2.329	Delhi	India	28.13N	77.27E	26	1817
3	1817-02-01	21.720	2.481	Ahmadabad	India	23.31N	72.52E	2	1817
4	1817-03-01	21.991	2.105	Delhi	India	28.13N	77.27E	26	1817

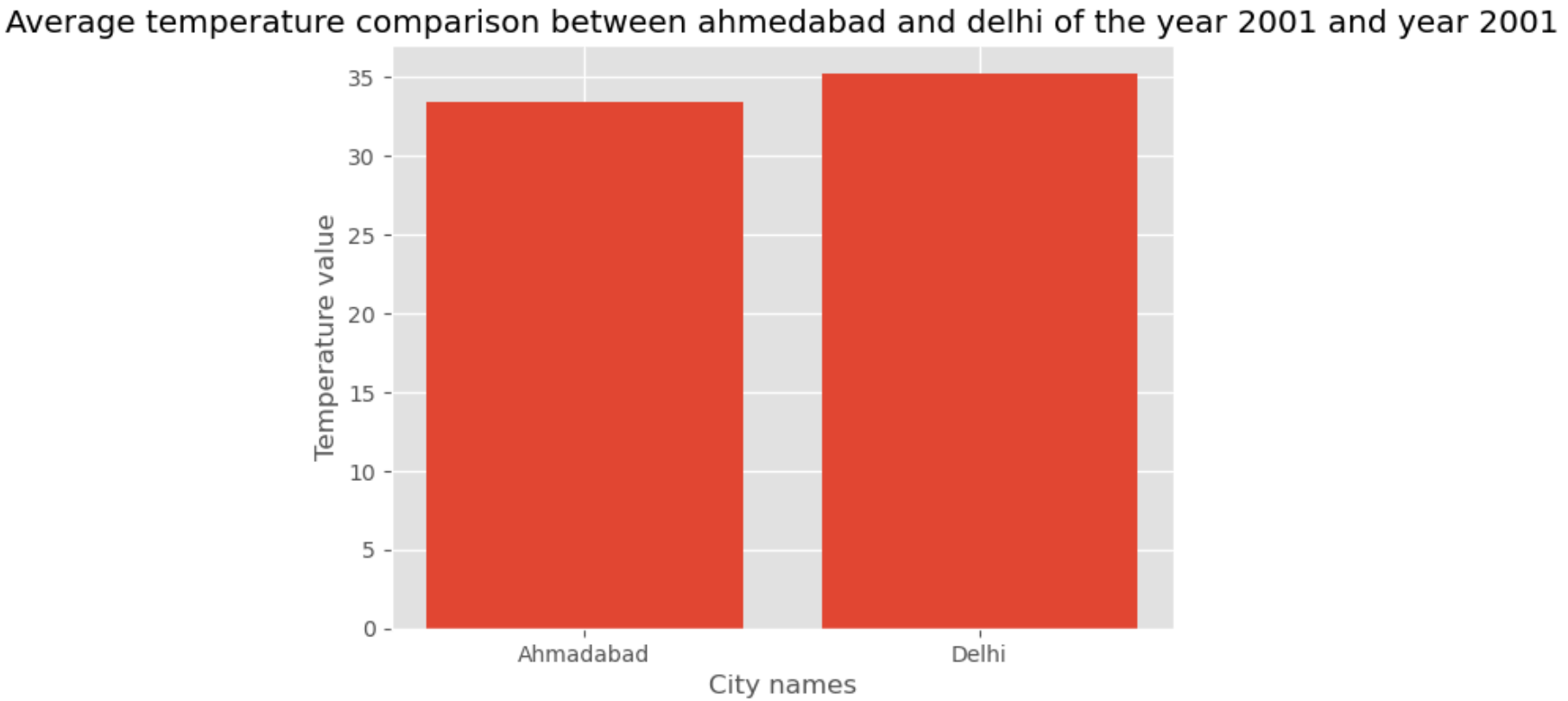
In [50]:
```
year_city_combined = year_data_combined.merge(city_merged, how="inner")
year_city_combined.head()
```

Out[50]:

	Date	AverageTemperature	AverageTemperatureUncertainty	City	Country	Latitude	Longitude	Label_encoded_city	year
0	2001-01-01	19.770	0.512	Ahmadabad	India	23.31N	72.52E	2	2001
1	2001-01-01	NaN	NaN	Delhi	India	28.13N	77.27E	26	2001
2	2001-02-01	18.882	0.612	Delhi	India	28.13N	77.27E	26	2001
3	2001-02-01	22.438	0.571	Ahmadabad	India	23.31N	72.52E	2	2001
4	2001-03-01	23.918	0.565	Delhi	India	28.13N	77.27E	26	2001

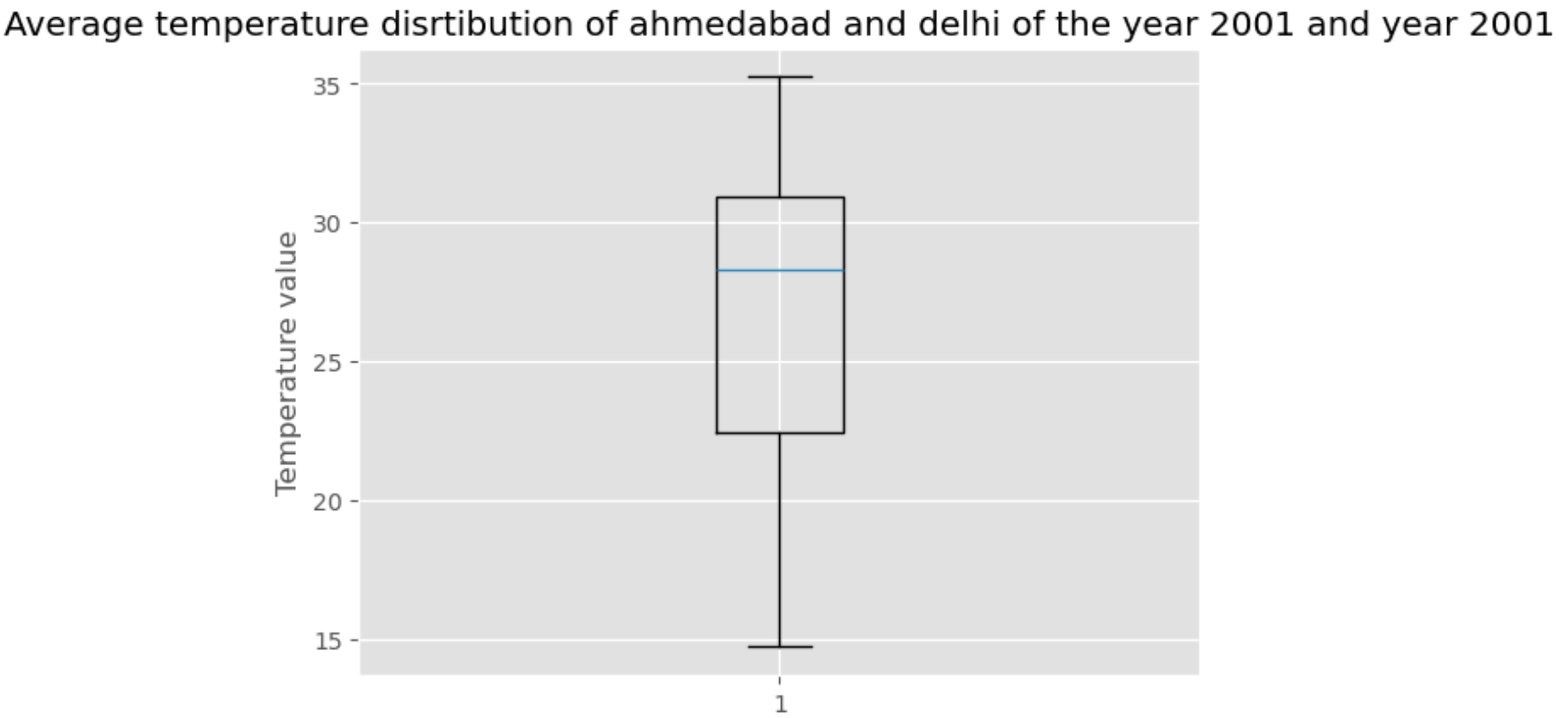
In [56]:
```
import matplotlib.pyplot as plt

plt.style.use("ggplot")
plt.title("Average temperature comparison between ahmedabad and delhi of the year 2001 and year 2001")
plt.xlabel("City names")
plt.ylabel("Temperature value")
plt.bar( year_city_combined["City"],year_city_combined["AverageTemperature"])
plt.show()
```



In [66]:
```
plt.title("Average temperature disrtibution of ahmedabad and delhi of the year 2001 and year 2001")

plt.ylabel("Temperature value")
plt.boxplot(year_city_combined["AverageTemperature"].dropna());
```


Average temperature disrtibution of ahmedabad and delhi of the year 2001 and year 2001
Temperature value
35
30
25
20
15
1

In [ ]: