Deliverable 2

Dataset: Climate Change: Earth Surface Temperature Data
Source:
https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data

Libraries used:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from sklearn.metrics import mean_squared_error
from math import sqrt
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

Understanding the data:

```python
cities = pd.read_csv(s3_csv_path)
cities.head()
```

	dt	AverageTemperature	AverageTemperatureUncertainty	City	Country	Latitude	Longitude
0	1743-11-01	6.068	1.737	Århus	Denmark	57.05N	10.33E
1	1743-12-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
2	1744-01-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
3	1744-02-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
4	1744-03-01	NaN	NaN	Århus	Denmark	57.05N	10.33E

```python
cities.shape
```

```
(8599212, 7)
```

```python
cities.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8599212 entries, 0 to 8599211
Data columns (total 7 columns):
 #   Column                         Dtype
---  ------                         -----
 0   dt                             object
 1   AverageTemperature             float64
 2   AverageTemperatureUncertainty  float64
 3   City                           object
 4   Country                        object
 5   Latitude                       object
 6   Longitude                      object
dtypes: float64(2), object(5)
memory usage: 459.2+ MB
```

The data set has 8599212 rows and 7 columns. The datatype of these columns are displayed above.

```
cities.describe
```

```
<bound method NDFrame.describe of                dt  AverageTemperature  AverageTemperatureUncertainty  \
0          1743-11-01            6.068000                       1.737000
1          1743-12-01           16.727433                       1.028575
2          1744-01-01           16.727433                       1.028575
3          1744-02-01           16.727433                       1.028575
4          1744-03-01           16.727433                       1.028575
...               ...                 ...                            ...
8599207    2013-05-01           11.464000                       0.236000
8599208    2013-06-01           15.043000                       0.261000
8599209    2013-07-01           18.775000                       0.193000
8599210    2013-08-01           18.025000                       0.298000
8599211    2013-09-01           16.727433                       1.028575

            City      Country Latitude Longitude
0          Århus      Denmark   57.05N    10.33E
1          Århus      Denmark   57.05N    10.33E
2          Århus      Denmark   57.05N    10.33E
3          Århus      Denmark   57.05N    10.33E
4          Århus      Denmark   57.05N    10.33E
...          ...          ...      ...       ...
8599207   Zwolle  Netherlands   52.24N     5.26E
8599208   Zwolle  Netherlands   52.24N     5.26E
8599209   Zwolle  Netherlands   52.24N     5.26E
8599210   Zwolle  Netherlands   52.24N     5.26E
8599211   Zwolle  Netherlands   52.24N     5.26E

[8599212 rows x 7 columns]>
```

Checking for null values:

```
cities.isnull().sum()
```
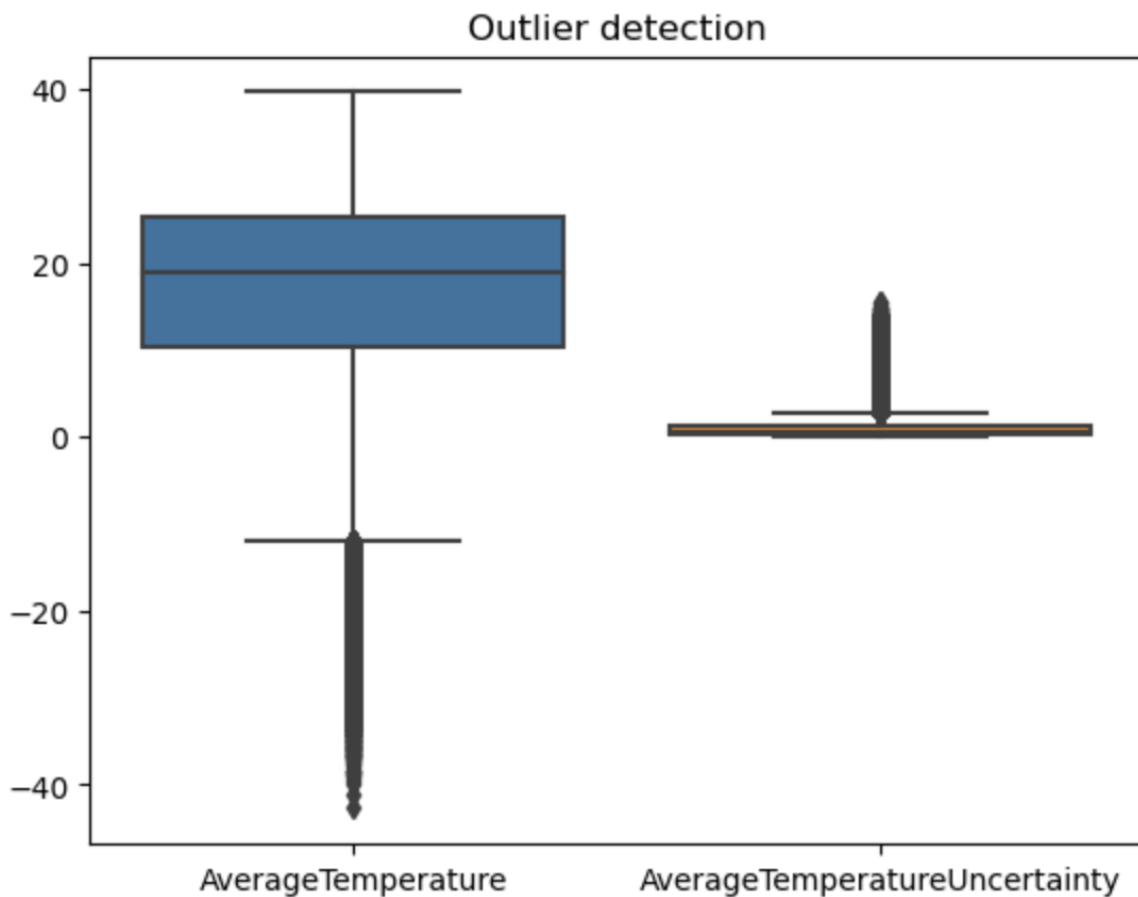
```
dt                                    0
AverageTemperature               364130
AverageTemperatureUncertainty    364130
City                                  0
Country                               0
Latitude                              0
Longitude                             0
dtype: int64
```

We found some null values which we will address in the data cleaning step.

Checking outliers in the dataset:

```python
sns.boxplot(data=cities)
plt.title("Outlier Detection")
plt.show()
```


Outlier detection

Data Cleaning:

Handling null values:

```python
cities['AverageTemperature'].fillna(cities['AverageTemperature'].mean(), inplace=True)
cities['AverageTemperatureUncertainty'].fillna(cities['AverageTemperatureUncertainty'].mean(), inplace=True)
```

```python
cities.isnull().sum()
```

```
dt                               0
AverageTemperature               0
AverageTemperatureUncertainty    0
City                             0
Country                          0
Latitude                         0
Longitude                        0
dtype: int64
```

Checking for duplicate rows:

```
duplicate = cities[cities.duplicated()]
print("Duplicate rows:", len(duplicate))
print(duplicate)
```

```
Duplicate rows: 0
Empty DataFrame
Columns: [dt, AverageTemperature, AverageTemperatureUncertainty, City, Country, Latitude, Longitude]
Index: []
```

We found no duplicate rows.

Feature correlations:

```
corr = cities.corr()
sns.heatmap(corr, cmap='coolwarm', annot=True)
plt.show()
```
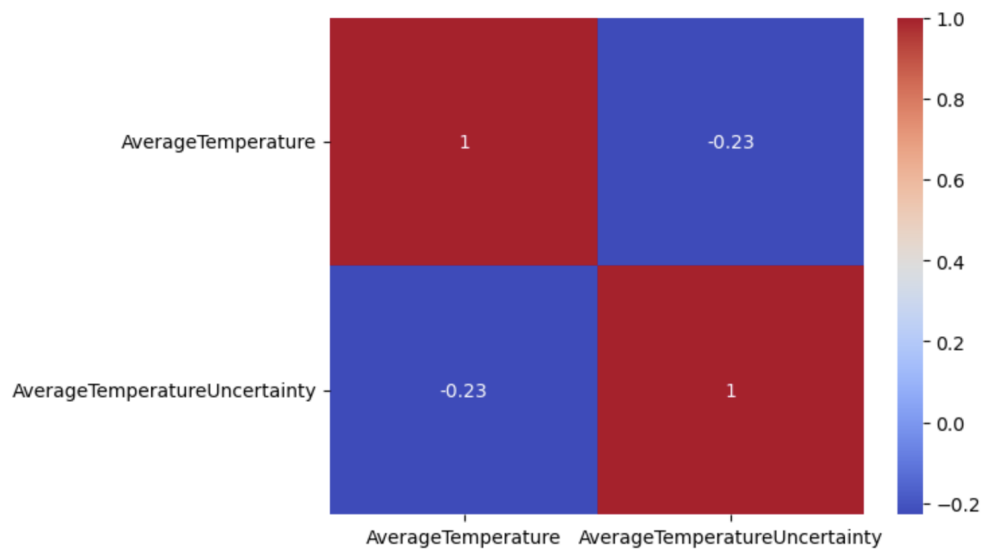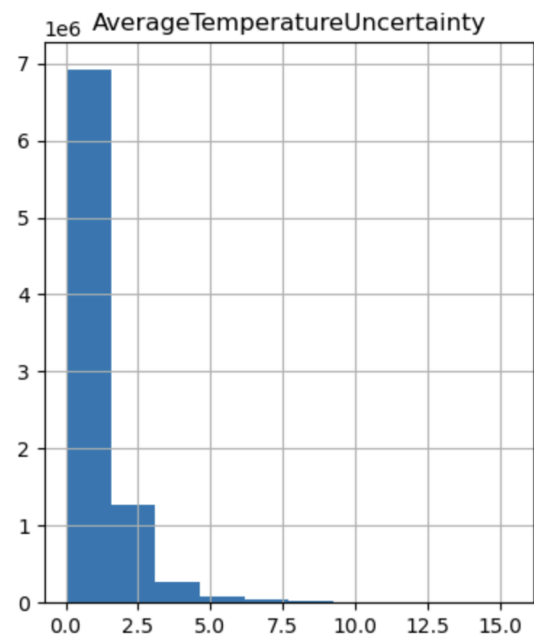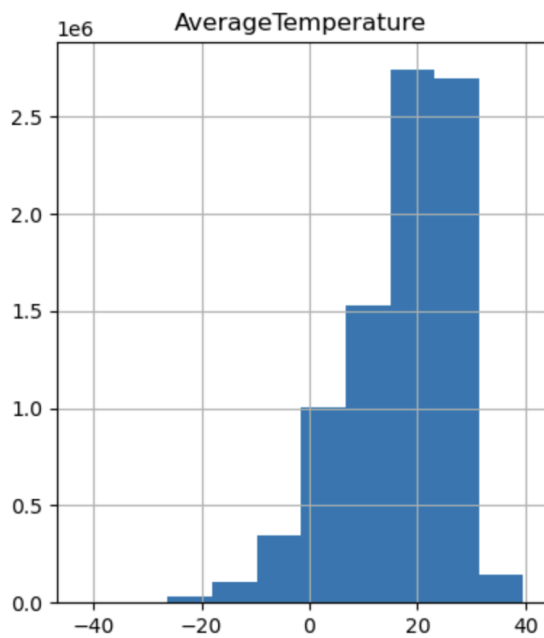
Data Visualizations

Histogram for AverageTemperature and AverageTemperatureUncertainity
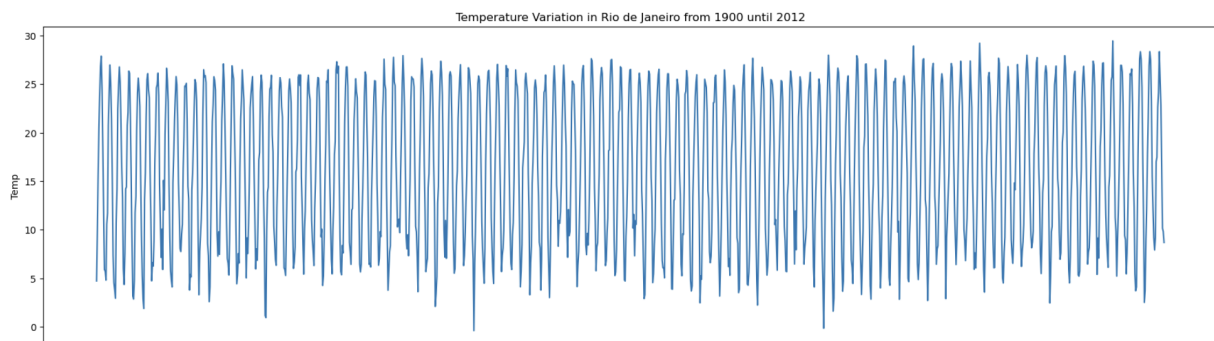
```
cities.hist(figsize=(10,5))
```

```
array([[<AxesSubplot: title={'center': 'AverageTemperature'}>,
        <AxesSubplot: title={'center': 'AverageTemperatureUncertainty'}>]],
      dtype=object)
```


Temperature Variation in Rio de Janeiro from 1900 until 2012

```
plt.figure(figsize=(22,6))
sns.lineplot(x=rio.index, y=rio['Temp'])
plt.title('Temperature Variation in Rio de Janeiro from 1900 until 2012')
plt.show()
```

Global Average Temperature over Time

```python
cities['dt'] = pd.to_datetime(cities['dt'])

# Group data by year and calculate average temperature
yearly_temps = cities.groupby(cities['dt'].dt.year)['AverageTemperature'].mean()

# Create a line plot of average temperature over time
plt.plot(yearly_temps.index, yearly_temps.values)
plt.xlabel('Year')
plt.ylabel('Average Temperature (°C)')
plt.title('Global Average Temperature over Time')
plt.show()
```
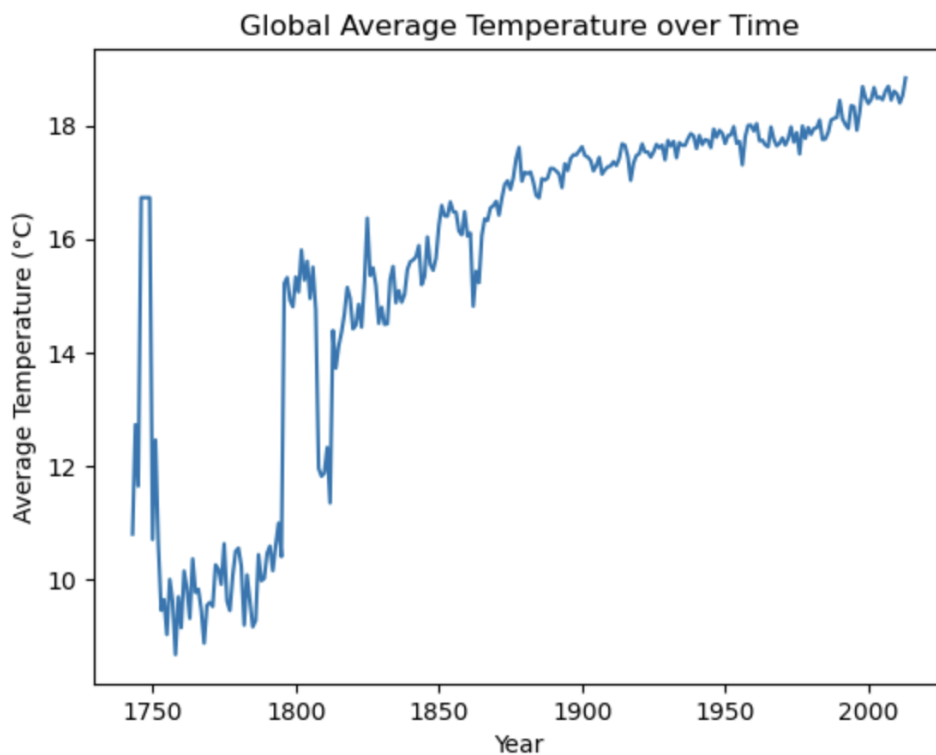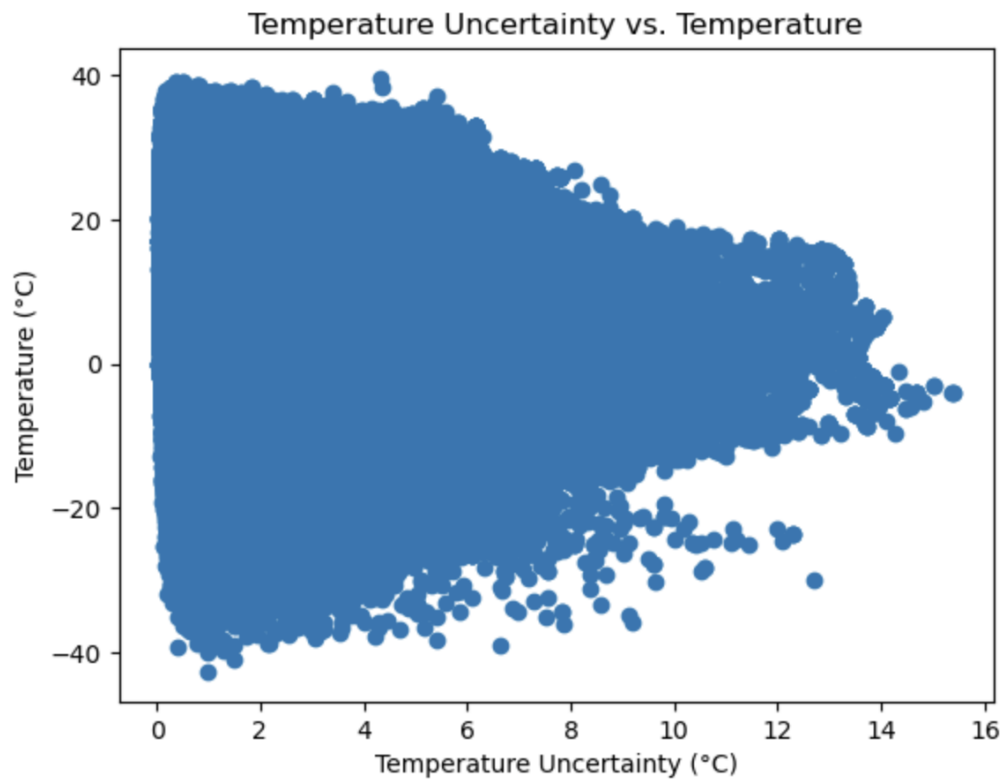
Temperature Uncertainty vs. Temperature

```python
plt.scatter(cities['AverageTemperatureUncertainty'], cities['AverageTemperature'])
plt.xlabel('Temperature Uncertainty (°C)')
plt.ylabel('Temperature (°C)')
plt.title('Temperature Uncertainty vs. Temperature')
plt.show()
```

Distribution of Temperature by City

```
sns.distplot(cities['AverageTemperature'])
plt.xlabel('Temperature (°C)')
plt.title('Distribution of Temperature by City')
plt.show()
```


Distribution of Temperature by City
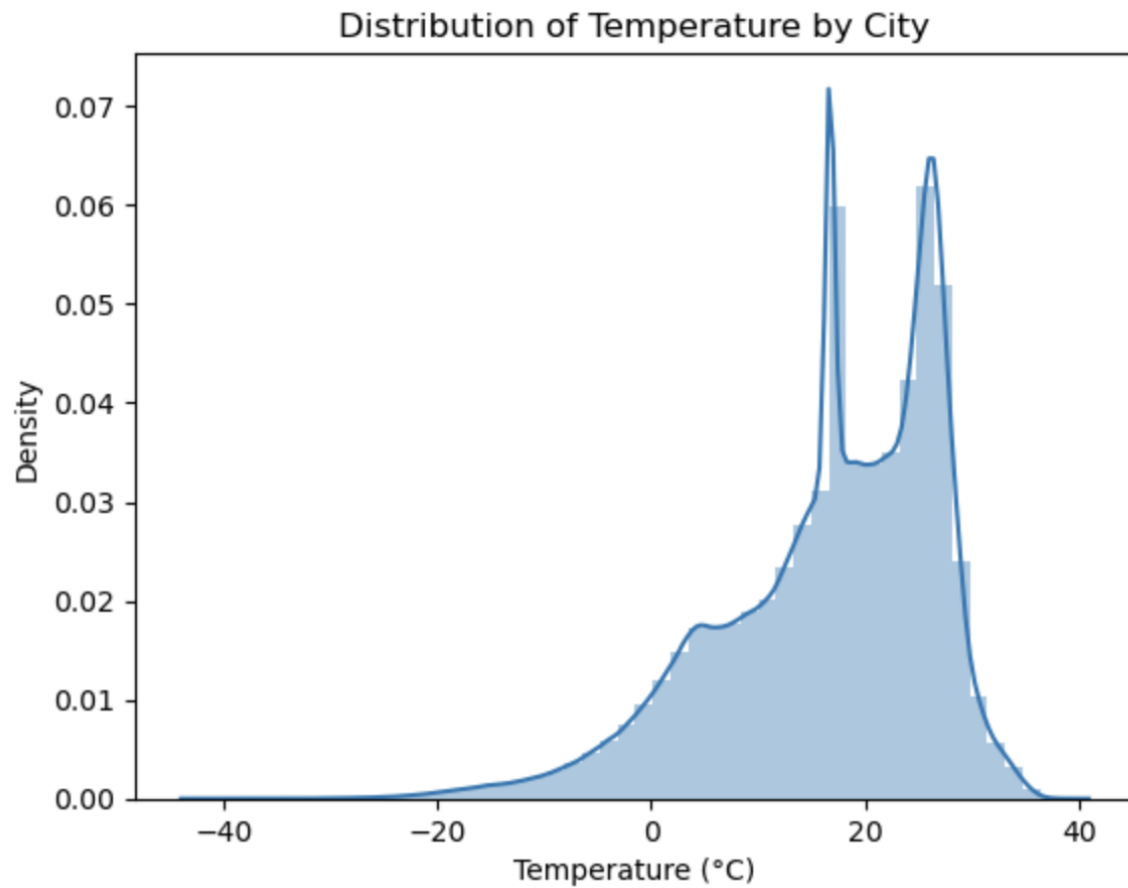
# Temperature Change Over Time in Charlotte vs. All Other Cities

```python
clt = cities[cities['City'] == 'Charlotte'].copy()
other_cities = cities[cities['City'] != 'Charlotte'].copy()

# Filter Charlotte data to include only the years after 1900
clt = clt[clt['dt'].str.startswith('19') | clt['dt'].str.startswith('20')].copy()
other_cities = other_cities[other_cities['dt'].str.startswith('19') | other_cities['dt'].str.startswith('20')].copy()

# Convert the date column to a datetime object
clt['dt'] = pd.to_datetime(clt['dt'])
other_cities['dt'] = pd.to_datetime(other_cities['dt'])

# Group the data by year and calculate the mean temperature for each year
clt_yearly_temp = clt.groupby(clt['dt'].dt.year)['AverageTemperature'].mean()
other_cities_yearly_temp = other_cities.groupby(other_cities['dt'].dt.year)['AverageTemperature'].mean()

# Create a line chart to compare the temperature changes over time in Charlotte and all other cities
plt.figure(figsize=(12,8))
sns.lineplot(x=clt_yearly_temp.index, y=clt_yearly_temp.values, label='Charlotte')
sns.lineplot(x=other_cities_yearly_temp.index, y=other_cities_yearly_temp.values, label='All Other Cities')
plt.title('Temperature Changes Over Time in Charlotte vs. All Other Cities')
plt.xlabel('Year')
plt.ylabel('Temperature (Celsius)')
plt.legend()
plt.show()
```
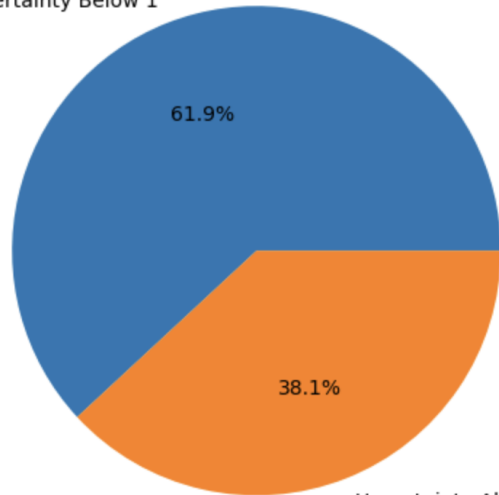
Proportion of Average Temperature Uncertainty Above and Below 1
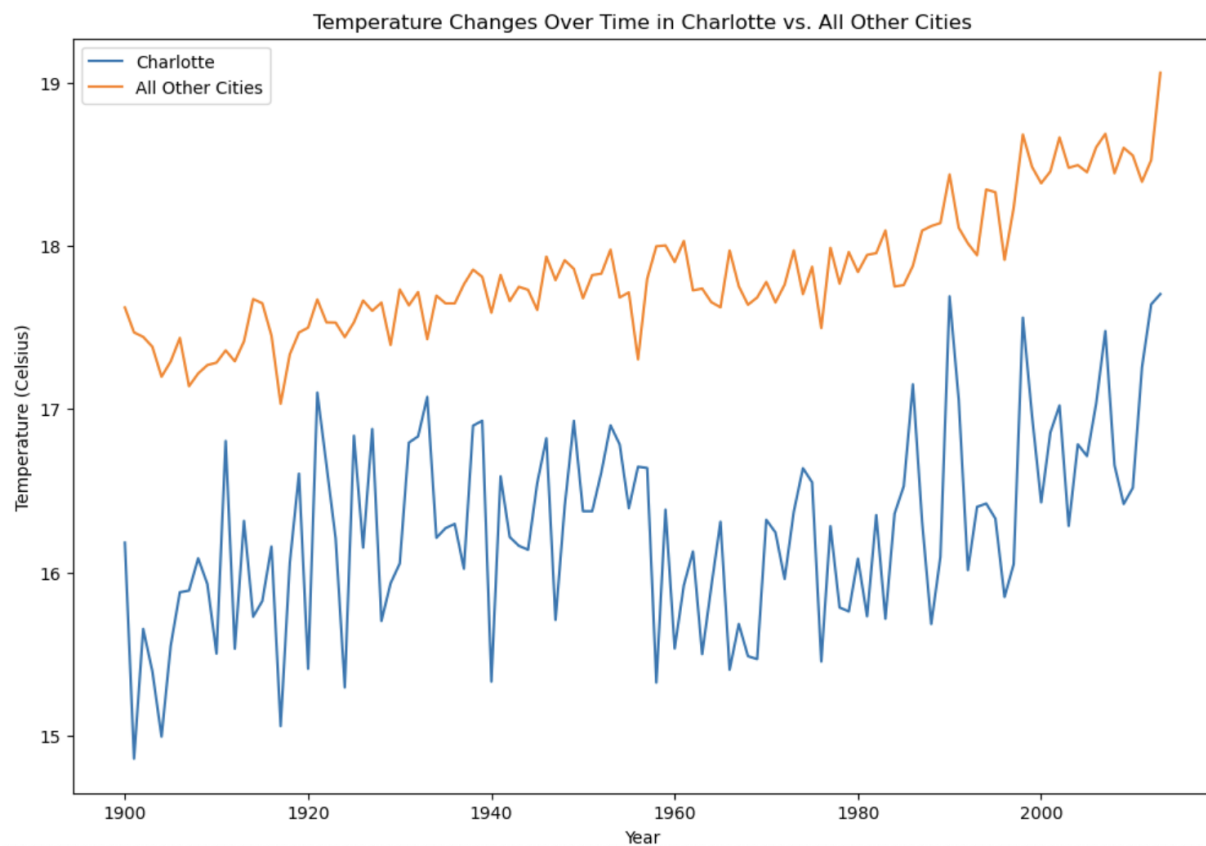
```python
plt.pie(uncertainty_counts, labels=['Uncertainty Below 1', 'Uncertainty Above 1'], autopct='%1.1f%%')
plt.title('Proportion of Average Temperature Uncertainty Above and Below 1')
plt.axis('equal')
plt.show()
```

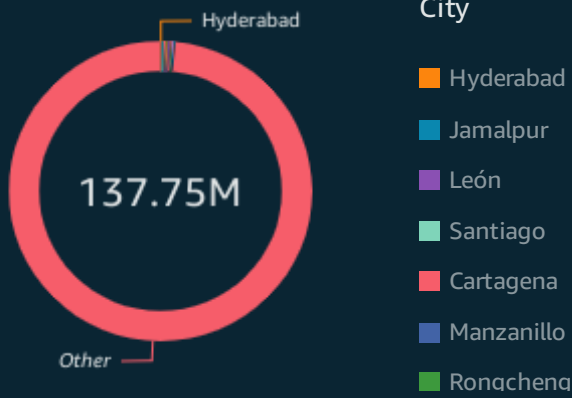Proportion of Average Temperature Uncertainty Above and Below 1

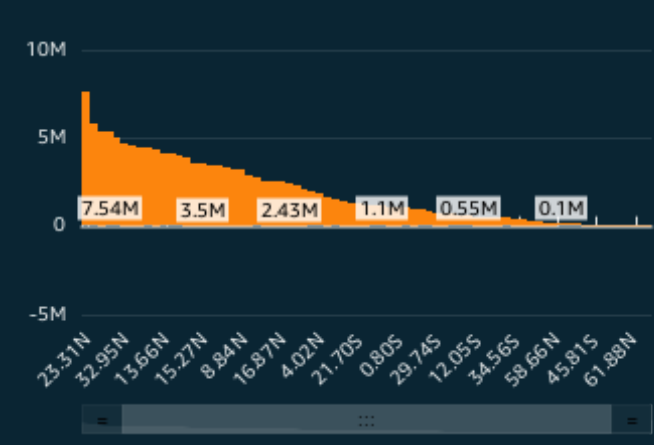## Sum of Averagetemperature by City

SHOWING TOP 20 IN CITY

137.75M

City
- Hyderabad
- Jamalpur
- León
- Santiago
- Cartagena
- Manzanillo
- Rongcheng

Sum of Averagetemperature by Latitude

SHOWING TOP 20 IN CITY

10M	
5M	
0	7.54M 3.5M 2.43M 1.1M 0.55M 0.1M
-5M	

23.31N 32.95N 13.66N 15.27N 8.84N 16.87N 4.02N 21.70S 0.80S 29.74S 12.05S 34.56S 58.66N 45.81S 61.88N

Sum of Averagetemperature by City and Dt

SHOWING TOP 200 IN CITY AND BOTTOM 25 IN DT

200M	
100M	131.44M
0	0.01M 0M 0M 0.03M 0.05M
-100M	

Ürümqi Zhum... Zhan... Zaan... Yining Yao Xinx... Wuxue Winds or Waw Other

Dt
- Nov 1, 1743
- Dec 1, 1743
- Jan 1, 1744
- Feb 1, 1744
- Mar 1, 1744
- Apr 1, 1744
- May 1, 1744
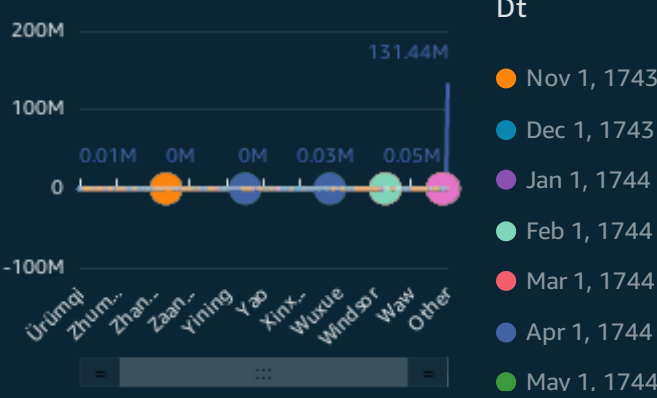
Distribution of AverageTemperature

3,00...	2,379,576
2,00...	1,526,511
1,00...	1,005,143 345,349 141,917
0	100 29,404

-42.7 -34.47 -26.23 -18 -9.76 -1.53 6.71 14.94 23.18 31.42 39.65
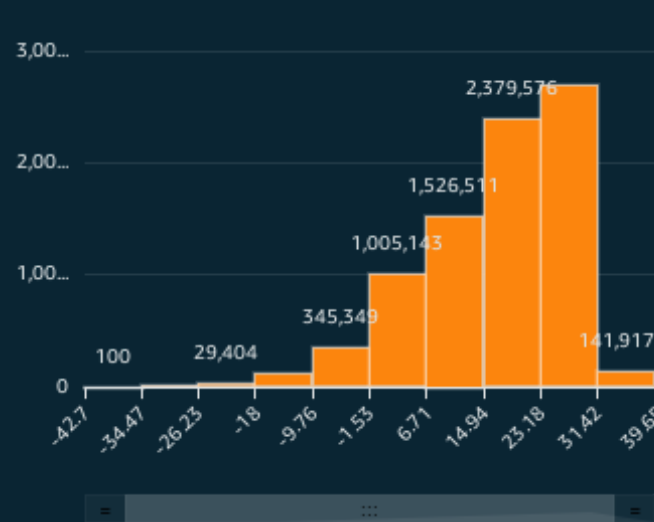
Sum of Averagetemperatureuncertainty by City an...

SHOWING BOTTOM 20 IN LATITUDE AND TOP 739 IN CITY

0.80N

3K	
2K	
1K	1.09K 1.32K 1.17K 1.16K 1.2K 1.29K
0	

Zinder Uruapan Tepic Seremban Saint Denis Puebla Palma Soriano Nairobi Maradi Long Xuyên Kismayo Jammagar Hadejia El Progreso Coacalco Campeche Bhubaneswar Baguio Abha
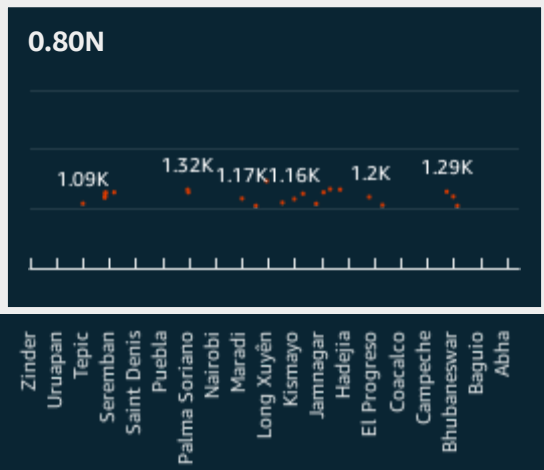
Sum of Avg temp for major cities

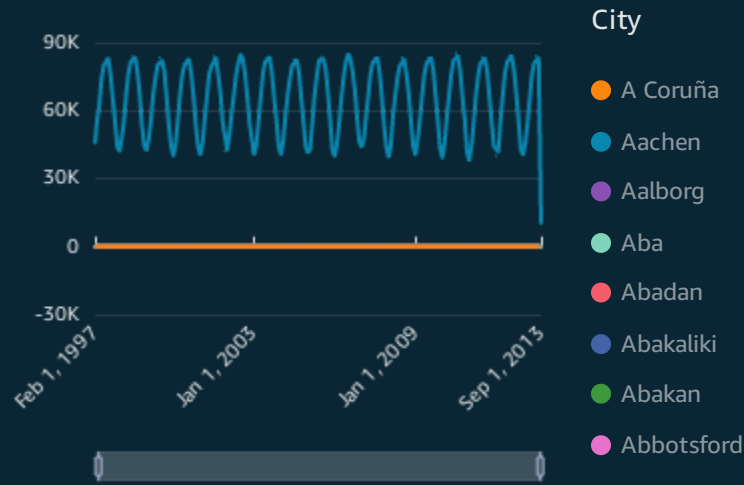| City	Value
Madras	
Rangoon	67.69K
Hyderabad	
Bombay	66.79K
Surat	
Calcutta	65.94K
Ahmadabad	
Dhaka	64.54K
Nagpur	
Bangalore	62.34K

0.01K 0 20K 40K 60K 80K
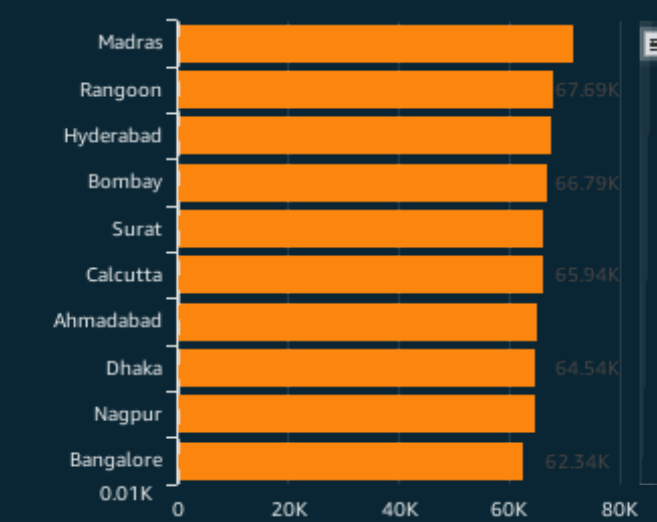
No data

There was no data found for the visual

Temperature trend over time

SHOWING TOP 200 IN DT AND BOTTOM 25 IN CITY

City
- 🟠 A Coruña
- 🔵 Aachen
- 🟣 Aalborg
- 🟢 Aba
- 🔴 Abadan
- 🔵 Abakaliki
- 🟢 Abakan
- 🟣 Abbotsford

Average temperature by latitude and longitude

SHOWING TOP 50 IN LATITUDE AND BOTTOM 50 IN LONGITUDE

Average Te...

1.62M

-0.03M
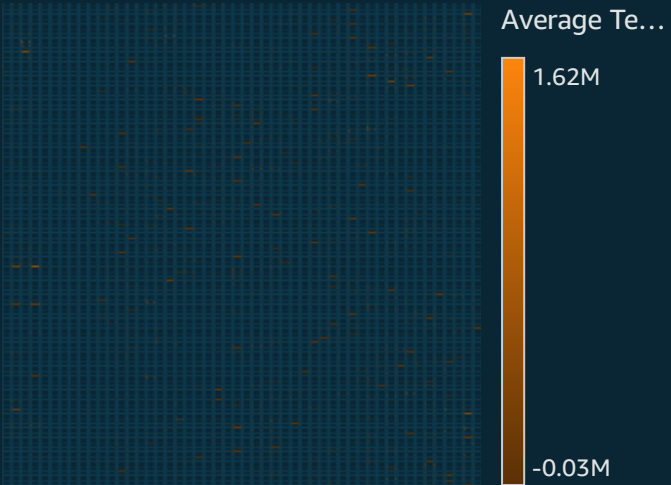
Relationship between average temperature and uncertainty

SHOWING TOP 2500 IN CITY

City
- 🟠 Springfield
- 🔵 Worcester
- 🟣 León
- 🟢 Rongcheng
- 🔴 Birmingham
- 🔵 Brest
- 🟢 Cambridge

April 25, 2023 12:40 AM (GMT)

☑ Powered by QuickSight