G-18 Univariate Analysis For Sentinel AOD Variable

November 21, 2020

1 G-18 Analysis and Visualization of Pollution using UV Aerosol Index UVAI as a proxy variable (Notebook 1 of 2)

Additional Packages required to run the code:

- conda update –all
- conda install geopandas
- conda install descartes

1.1 Table Of Contents

1.2 Done By: -

Name	Roll Number
Vimal Rajesh	B180336CS
Puchakayala Dheeraj Reddy	B180902CS
P Arjun	B180454CS
Alok Raj	B180411CS
Kunal Ravikumar Jagtap	B180921CS

Introduction

1. Our assigned dataset is Sentinel-5P NRTI AER AI: Near Real-Time UV Aerosol Index

2. Our variable/band of interest is Absorbing Aerosol Index (AAI)

3. The Absorbing Aerosol Index (AAI) can be used to determine the presence of UV-absorbing aerosols, such as dust and smoke.

4. Positive values indicate the presence of these pollutants.

5. Availability of our dataset is from July 10, 2018 to present(till October 13, 2020)

Atmospheric aerosols from human activity influence climate.

Atmospheric aerosols are suspensions of liquid, solid, or mixed particles with highly variable chemical composition and size distribution (Putaud et al. 2010). Their variability is due to the numerous sources and varying formation mechanisms (Figure 1). Aerosol particles are either emitted directly to the atmosphere (primary aerosols) or produced in the atmosphere from precursor gases (secondary aerosols).

The Indian subcontinent, apart from being a source region for aerosols, is bordered by densely populated and industrialized areas on the east and western sides from where different aerosol species are produced and transported and is one of the regional aerosol hot spots. The Indian landmass comprises coastal regions, inland plains, semiarid regions, mountains, and plateau regions and experiences tropical and subtropical climatic conditions resulting in extreme temperatures, rainfall, and relative humidity which modulate the aerosol characteristics

Absorbing Aerosol Index provides the measure of the prevalence of aerosols in the atmosphere, calculated by this equation on https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-5p/level-2/aerosol-index using the 354/388 wavelength pair.

2 Loading the data

We will load all the region data into a dictionary of dataframes. This way, we can access all of them.

[1]:
```
# importing required libraries
import glob
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import geopandas as gpd
import seaborn as sns
from scipy.stats import ttest_ind
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
cities1 = ["Ahmedabad", "Bangalore" , "Chennai", "Hyderabad", "Jaipur" ,
           "Pune", "Mumbai" ,"Bhopal" ,"Bhubaneshwar" , "Kolkata","Delhi"]
for x in cities1:
    print(x) # printing the cities of interest
```

```
Ahmedabad
Bangalore
Chennai
Hyderabad
Jaipur
Pune
Mumbai
Bhopal
Bhubaneshwar
Kolkata
Delhi
```

[2]:
```
# loading the csvs into a dictionary of dataframes
df = {}
for i in cities1:
    df[i] = pd.read_csv(r'Cleaned CSVs/'+i+'/merged_csv'+i+".csv")
    df[i]= df[i].fillna(0)
    df[i]["date"]= pd.to_datetime(df[i].date)
# df
```

[3]:
```
# Viewing the cleaned data with an example city
df["Delhi"].head(10)
```

[3]:

	Ward_No	date	mean
0	0	2018-07-11	0.438815
1	150	2018-07-11	0.745821
2	176	2018-07-11	0.416006
3	129	2018-07-11	0.767066
4	173	2018-07-11	0.562144
5	140	2018-07-11	0.740244
6	206	2018-07-11	0.555880
7	145	2018-07-11	0.727814
8	179	2018-07-11	0.270564
9	3	2018-07-11	0.264471

3 Visualising For Individual Cities

Now as we have already loaded the dataframes, lets move on to Visualising and deriving Insights from them.

3.1 City Number 1: Delhi

3.1.1 Loading and Visualizing

[4]:
```
df["Delhi"]
```

[4]:

	Ward_No	date	mean
0	0	2018-07-11	0.438815

```
1          150 2018-07-11  0.745821
2          176 2018-07-11  0.416006
3          129 2018-07-11  0.767066
4          173 2018-07-11  0.562144
…          …          …       …
689887      35 2020-10-13 -0.700897
689888     219 2020-10-13 -0.764063
689889      26 2020-10-13 -0.678314
689890     187 2020-10-13 -0.756840
689891     115 2020-10-13 -0.792268

[689892 rows x 3 columns]
```

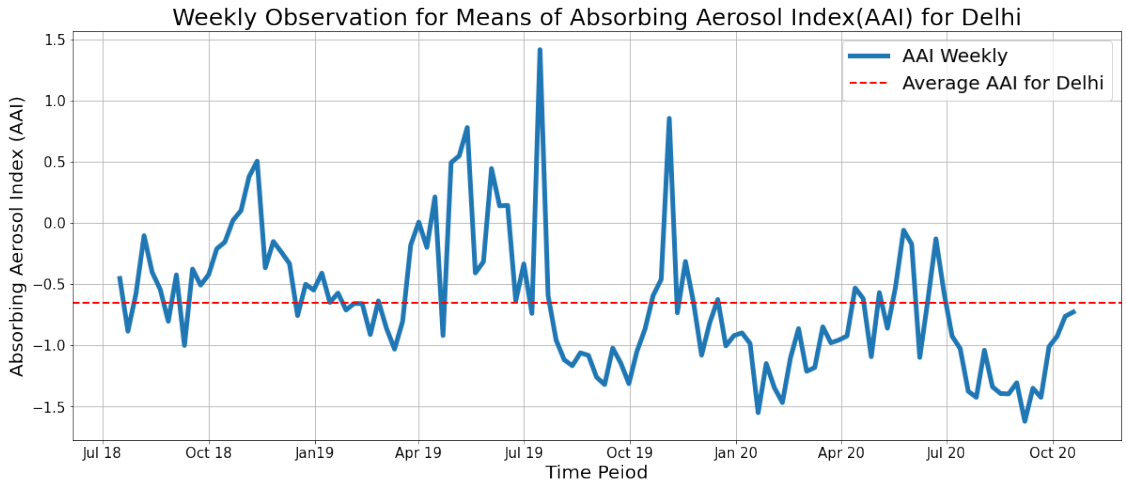
[5]:
```
# g = df1.groupby(pd.Grouper(freq="M"))
# g.mean()
Delhi = df["Delhi"].groupby(["date"]).mean()
Delhi["month"] = Delhi.index.month
Delhi["year"] = Delhi.index.year
Delhi
```

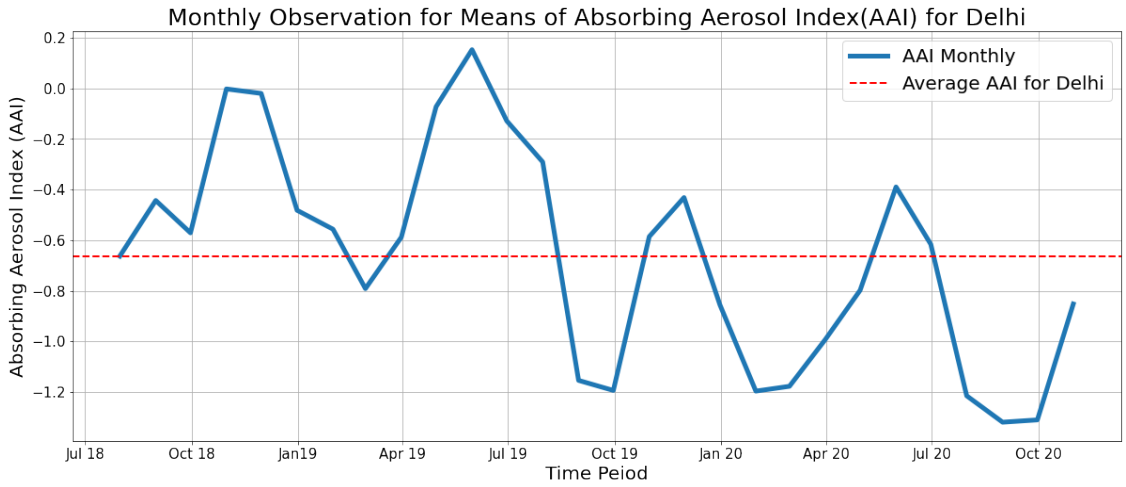
[5]:
```
                mean  month  year
date
2018-07-11  0.512202      7  2018
2018-07-12  0.032969      7  2018
2018-07-13 -0.495366      7  2018
2018-07-15 -1.645103      7  2018
2018-07-16 -0.672453      7  2018
…                …    …    …
2020-10-09 -0.199986     10  2020
2020-10-10 -0.842192     10  2020
2020-10-11 -0.850865     10  2020
2020-10-12 -0.508435     10  2020
2020-10-13 -0.727903     10  2020

[793 rows x 3 columns]
```

[6]:
```
# Aggregate statistics grouped by Weeks
Weekly_Delhi_mean = Delhi.groupby(pd.Grouper(freq="W-MON")).mean()
Weekly_Delhi_max = Delhi.groupby(pd.Grouper(freq="W-MON")).max()
Weekly_Delhi_min = Delhi.groupby(pd.Grouper(freq="W-MON")).min()

# Aggregate statistics grouped by Months
Monthly_Delhi_mean = Delhi.groupby(pd.Grouper(freq="M")).mean()
Monthly_Delhi_max = Delhi.groupby(pd.Grouper(freq="M")).max()
Monthly_Delhi_min = Delhi.groupby(pd.Grouper(freq="M")).min()

#Week Identifier
```

```
Weekly_Delhi_mean["week"] = Weekly_Delhi_mean.index.isocalendar().week
Weekly_Delhi_min["week"] = Weekly_Delhi_min.index.isocalendar().week
Weekly_Delhi_max["week"] = Weekly_Delhi_max.index.isocalendar().week

Monthly_Delhi_mean["week"] = Monthly_Delhi_mean.index.isocalendar().week
Monthly_Delhi_min["week"] = Monthly_Delhi_min.index.isocalendar().week
Monthly_Delhi_max["week"] = Monthly_Delhi_max.index.isocalendar().week
```

[7]:
```
# Visualizing by grouping into weeekly data
plt.figure(figsize = (20,8))
plt.plot( Weekly_Delhi_mean.index.tolist(),Weekly_Delhi_mean["mean"], linewidth␣
 ↪= 5, label = "AAI Weekly")
plt.gca().axhline(y=Weekly_Delhi_mean["mean"].mean(),color='r', linestyle='--',␣
 ↪lw=2, label = "Average AAI for Delhi")
plt.title("Weekly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Delhi",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()

# Visualizing by grouping into Monthly data
plt.figure(figsize = (20,8))
plt.plot( Monthly_Delhi_mean.index.tolist(),Monthly_Delhi_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Monthly")
plt.gca().axhline(y=Monthly_Delhi_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Delhi")
plt.title("Monthly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Delhi",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()
```



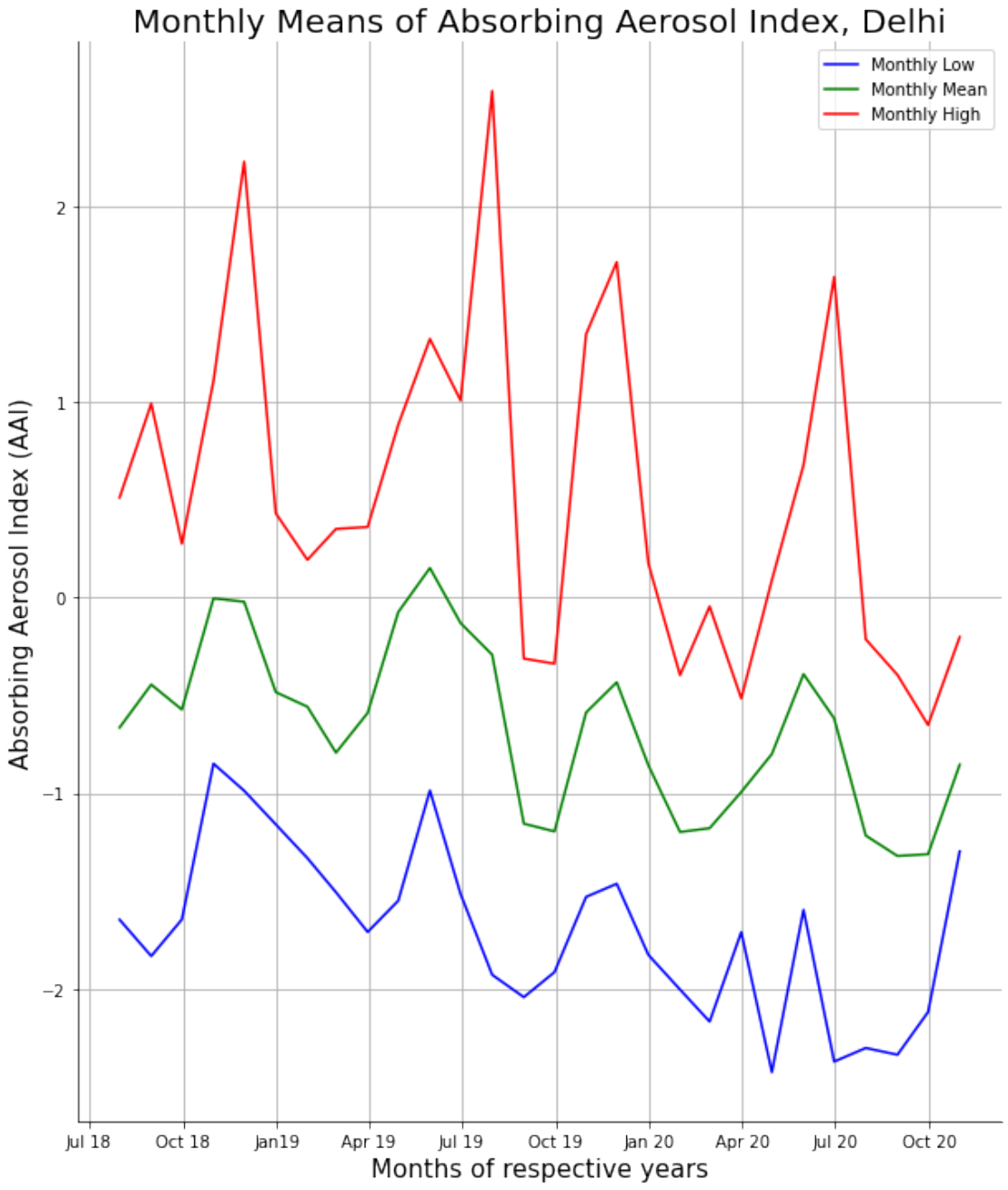


We can observe that Aerosol Optical Depth has decreased in the past year. The change in policies, Even Odd Rule and Coronavirus Pandemic seems to have shifted the curve down.

We will now plot some useful insights with Monthly Data

[8]:
```
# Plotting Monthly means
plt.figure(figsize=(10,12))
plt.plot(list(Monthly_Delhi_min.index),list(Monthly_Delhi_min["mean"]), label =␣
↪"Monthly Low", color = "blue" )
plt.plot(list(Monthly_Delhi_mean.index),list(Monthly_Delhi_mean["mean"]),␣
↪label= "Monthly Mean" , color ="green")
plt.plot(list(Monthly_Delhi_max.index),list(Monthly_Delhi_max["mean"]), label =␣
↪"Monthly High", color = "red" )
```

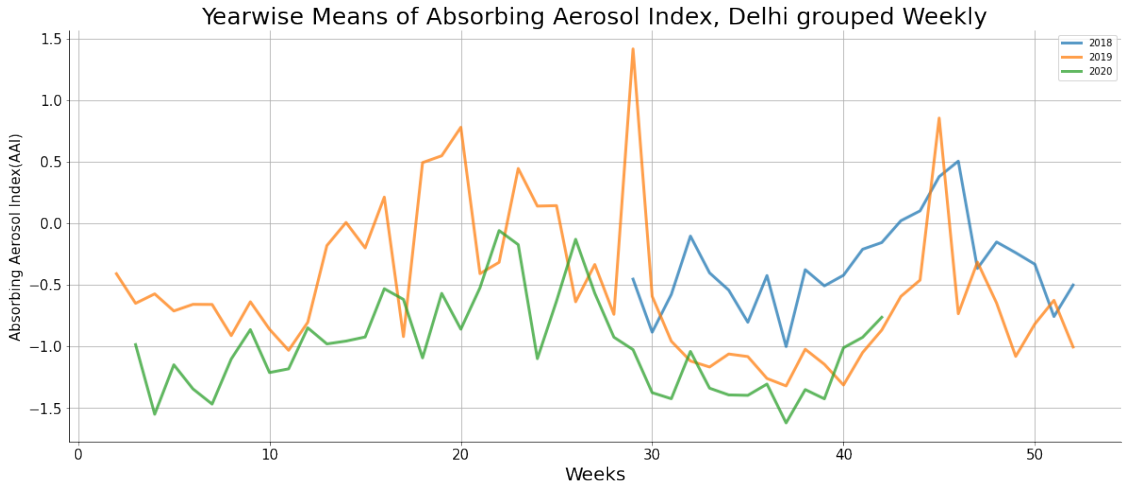
```
plt.xlabel("Months of respective years",fontsize= 15)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Monthly Means of Absorbing Aerosol Index, Delhi",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 15)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels)
plt.legend()
plt.grid()
```

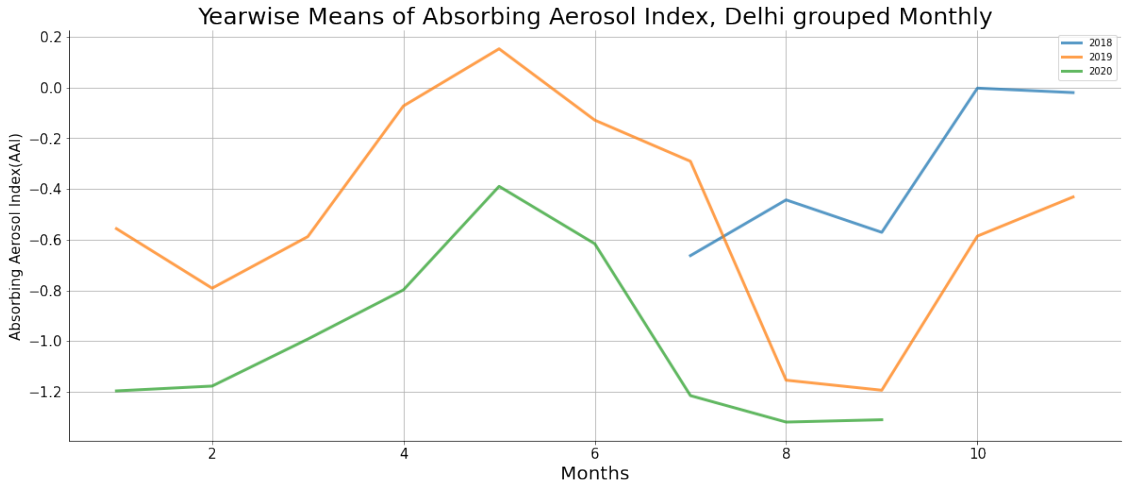


[9]:
```
# Plotting Yearwise means
x = [i for i in range(1,13)]
plt.figure(figsize = (20,8))
plt.plot(Weekly_Delhi_mean[Weekly_Delhi_mean["year"] == 2018]["week"].tolist()[:
 ↪-1],Weekly_Delhi_mean[Weekly_Delhi_mean["year"] == 2018]["mean"].tolist()[:
 ↪-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Weekly_Delhi_mean[Weekly_Delhi_mean["year"] == 2019]["week"].tolist()[:
 ↪-1],Weekly_Delhi_mean[Weekly_Delhi_mean["year"] == 2019]["mean"].tolist()[:
 ↪-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Weekly_Delhi_mean[Weekly_Delhi_mean["year"] == 2020]["week"].tolist()[:
 ↪-1],Weekly_Delhi_mean[Weekly_Delhi_mean["year"] == 2020]["mean"].tolist()[:
 ↪-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Weeks",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Yearwise Means of Absorbing Aerosol Index, Delhi grouped␣
 ↪Weekly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.grid()
plt.legend()

plt.figure(figsize = (20,8))
plt.plot(Monthly_Delhi_mean[Monthly_Delhi_mean["year"] == 2018]["month"].
 ↪tolist()[:-1],Monthly_Delhi_mean[Monthly_Delhi_mean["year"] == 2018]["mean"].
 ↪tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Monthly_Delhi_mean[Monthly_Delhi_mean["year"] == 2019]["month"].
 ↪tolist()[:-1],Monthly_Delhi_mean[Monthly_Delhi_mean["year"] == 2019]["mean"].
 ↪tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Monthly_Delhi_mean[Monthly_Delhi_mean["year"] == 2020]["month"].
 ↪tolist()[:-1],Monthly_Delhi_mean[Monthly_Delhi_mean["year"] == 2020]["mean"].
 ↪tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Months",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.title("Yearwise Means of Absorbing Aerosol Index, Delhi grouped␣
 ↪Monthly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.grid()
```

```
plt.legend()
```

[9]: <matplotlib.legend.Legend at 0x241bb1cdb20>





The observation from the above graph is that, the Absorbing Aerosol Index values have come down considerably in the last two years. From the graph it is evident that the Aerosol Index values are lesser in 2020. Therefore it can be concluded that the polution in Delhi has been lower in 2020 compared to the previous two years.

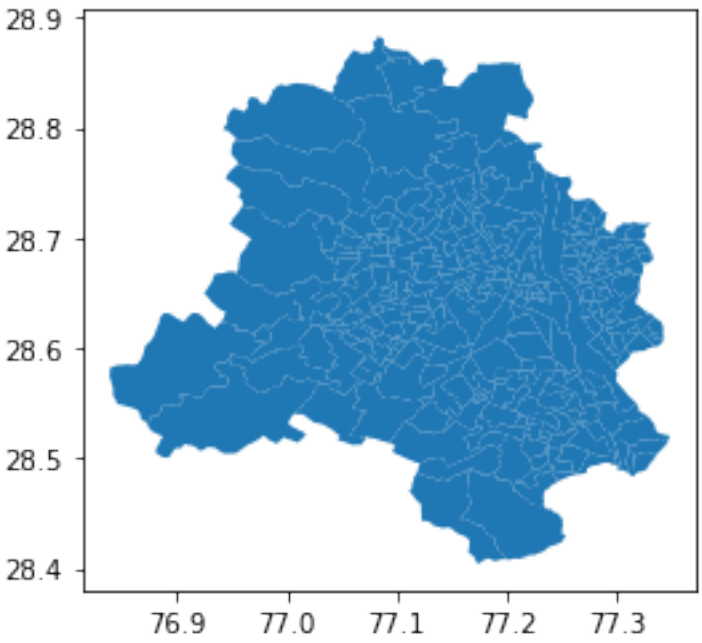
3.1.2 Plotting Maps For Insights

[10]:
```
# Grouping by Wards Data for further analysis
Delhi_wards = df["Delhi"].groupby(["Ward_No"]).mean()
Delhi_wards
```

[10]:
```
              mean
Ward_No
0        -0.680295
1        -0.731255
10       -0.689665
100      -0.647567
101      -0.625559
…              …
NDMC_5   -0.724424
NDMC_6   -0.726328
NDMC_7   -0.734284
NDMC_8   -0.746274
NDMC_9   -0.755857

[290 rows x 1 columns]
```

[11]:
```
# loading the shapefile
path = r'Cleaned CSVs/Delhi/Delhi_Wards-polygon.shp'
map_Delhi = gpd.read_file(path)
# Viewing the shapefile
map_Delhi.plot()
```

[11]:
```
<AxesSubplot:>
```

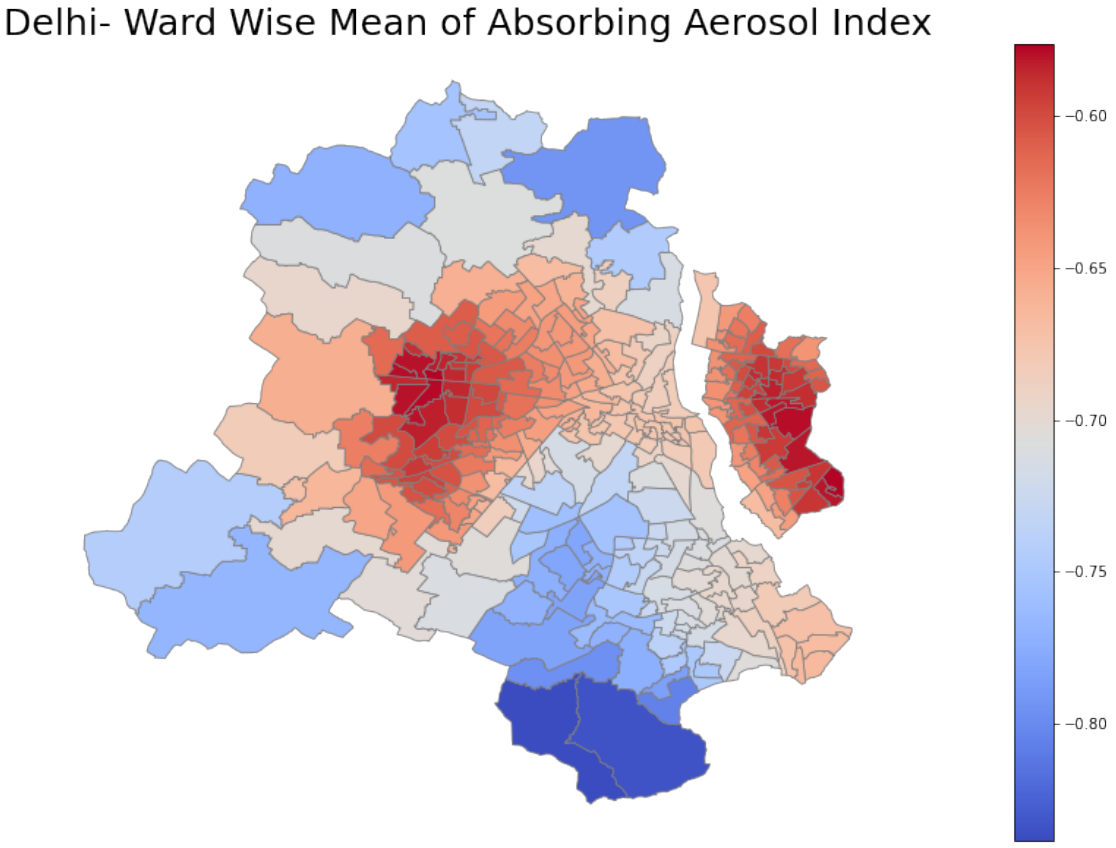


[12]:
```
# merging the shapefile with the mean data
merged_Delhi = map_Delhi.merge(Delhi_wards, left_on = "Ward_No", right_index =␣
 ↪True)
# merged_Delhi.head()
merged_Delhi
```

[12]:
```
                Ward_Name Ward_No  \
0    DELHI CANTT CHARGE 1  CANT_1
1    DELHI CANTT CHARGE 2  CANT_2
2    DELHI CANTT CHARGE 4  CANT_4
3    DELHI CANTT CHARGE 5  CANT_5
4    DELHI CANTT CHARGE 6  CANT_6
..                    …     …
285            PREM NAGAR      33
286         SHASTRI NAGAR      73
287         VISHNU GARDEN     107
288           TUKHMIR PUR     270
289             RAJ NAGAR     142

                                                  geometry      mean
0    POLYGON ((77.13228 28.63154, 77.13644 28.62062… -0.696738
1    POLYGON ((77.15429 28.62335, 77.15501 28.62228… -0.734493
2    POLYGON ((77.15755 28.57578, 77.15672 28.57564… -0.773336
3    POLYGON ((77.13480 28.57051, 77.13429 28.57048… -0.751757
4    POLYGON ((77.12157 28.59308, 77.12878 28.59029… -0.704756
..                                                 …       …
285  POLYGON ((77.06644 28.70141, 77.06574 28.70063… -0.587757
286  POLYGON ((77.19404 28.67288, 77.19196 28.66855… -0.664843
287  POLYGON ((77.09982 28.64496, 77.09947 28.64504… -0.611266
288  POLYGON ((77.25834 28.71855, 77.25858 28.71785… -0.637667
289  POLYGON ((77.08111 28.57601, 77.08554 28.57480… -0.661352

[289 rows x 4 columns]
```

[13]:
```
# Plotting choropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values
vmin, vmax = merged_Delhi["mean"].min(), merged_Delhi["mean"].max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))
# remove the axis
ax.axis('off')
# add a title and annotation
```

```
ax.set_title('Delhi- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))
# empty array for the data range
sm.set_array([])
# add the colorbar to the figure
fig.colorbar(sm)
# create map
merged_Delhi.plot(column=variable, cmap='coolwarm', linewidth=0.8, ax=ax,␣
 ↪edgecolor='0.5', figsize = (5,15))
plt.show()
```



[14]:
```
# Using T-Test to determine if climate has changed in the first 14 and last 14␣
 ↪months in Delhi
ttest_ind(Monthly_Delhi_mean.iloc[:14,:]["mean"].tolist(),Monthly_Delhi_mean.
 ↪iloc[14:,:]["mean"].tolist())
```

[14]: Ttest_indResult(statistic=4.03457856857236, pvalue=0.0004271712944067683)

We observe that Pollution levels has decreased in the last 14 months and the statistical test proves the same with a very small p-value for 95 % Confidence Interval

[ ]:

3.2 City Number 2: Mumbai

3.2.1 Loading and Visualizing

[15]:
```
df["Mumbai"]
```

[15]:

	date	Ward_No	mean
0	2018-07-11	12	-0.120585
1	2018-07-11	13	-0.166552
2	2018-07-11	14	-0.415738
3	2018-07-11	15	-0.414234
4	2018-07-11	16	-0.323975
…	…	…	…
57667	2020-10-13	10	-1.071737
57668	2020-10-13	11	-1.253872
57669	2020-10-13	22	-1.148774
57670	2020-10-13	23	-1.259724
57671	2020-10-13	24	-1.151767

[57672 rows x 3 columns]

[16]:
```
# g = df1.groupby(pd.Grouper(freq="M"))
# g.mean()
Mumbai = df["Mumbai"].groupby(["date"]).mean()
Mumbai = Mumbai[["mean"]]
Mumbai["month"] = Mumbai.index.month
Mumbai["year"] = Mumbai.index.year
Mumbai
```

[16]:

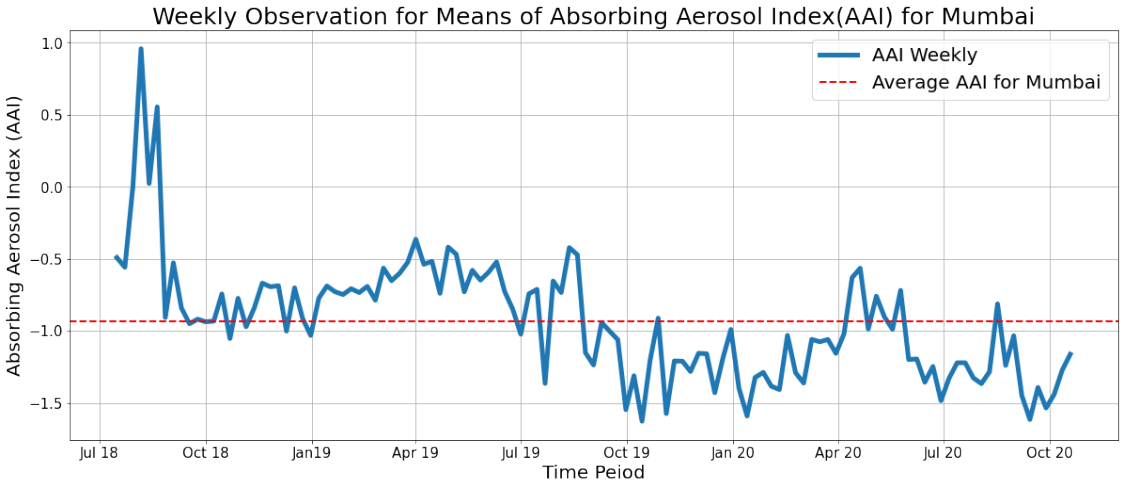
	mean	month	year
date			
2018-07-11	-0.291884	7	2018
2018-07-12	-0.842016	7	2018
2018-07-13	-0.540682	7	2018
2018-07-14	-0.703053	7	2018
2018-07-15	-0.356510	7	2018
…	…	…	…
2020-10-09	-1.677282	10	2020
2020-10-10	-1.181626	10	2020
2020-10-11	-1.465807	10	2020
2020-10-12	-1.601309	10	2020
2020-10-13	-1.162175	10	2020

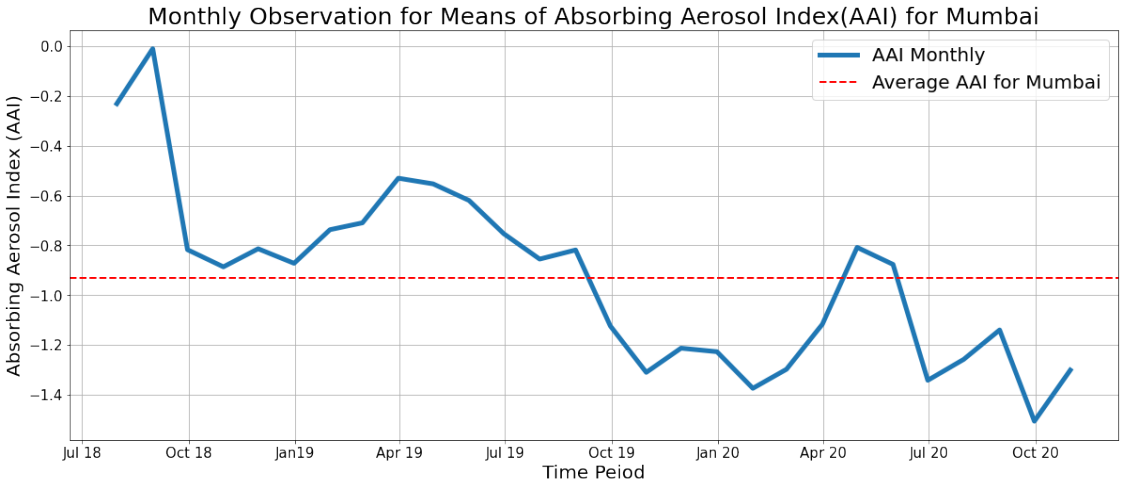
```
[801 rows x 3 columns]
```

[17]:
```
# Aggregate statistics grouped by Weeks
Weekly_Mumbai_mean = Mumbai.groupby(pd.Grouper(freq="W-MON")).mean()
Weekly_Mumbai_max = Mumbai.groupby(pd.Grouper(freq="W-MON")).max()
Weekly_Mumbai_min = Mumbai.groupby(pd.Grouper(freq="W-MON")).min()

# Aggregate statistics grouped by Months
Monthly_Mumbai_mean = Mumbai.groupby(pd.Grouper(freq="M")).mean()
Monthly_Mumbai_max = Mumbai.groupby(pd.Grouper(freq="M")).max()
Monthly_Mumbai_min = Mumbai.groupby(pd.Grouper(freq="M")).min()

#Week Identifier
Weekly_Mumbai_mean["week"] = Weekly_Mumbai_mean.index.isocalendar().week
Weekly_Mumbai_min["week"] = Weekly_Mumbai_min.index.isocalendar().week
Weekly_Mumbai_max["week"] = Weekly_Mumbai_max.index.isocalendar().week

Monthly_Mumbai_mean["week"] = Monthly_Mumbai_mean.index.isocalendar().week
Monthly_Mumbai_min["week"] = Monthly_Mumbai_min.index.isocalendar().week
Monthly_Mumbai_max["week"] = Monthly_Mumbai_max.index.isocalendar().week
```

[18]:
```
# Visualizing by grouping into weeekly data
plt.figure(figsize = (20,8))
plt.plot( Weekly_Mumbai_mean.index.tolist(),Weekly_Mumbai_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Weekly")
plt.gca().axhline(y=Weekly_Mumbai_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Mumbai")
plt.title("Weekly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Mumbai",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()

# Visualizing by grouping into Monthly data
plt.figure(figsize = (20,8))
plt.plot( Monthly_Mumbai_mean.index.tolist(),Monthly_Mumbai_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Monthly")
plt.gca().axhline(y=Monthly_Mumbai_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Mumbai")
plt.title("Monthly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Mumbai",fontsize = 25)
```

```
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()
```

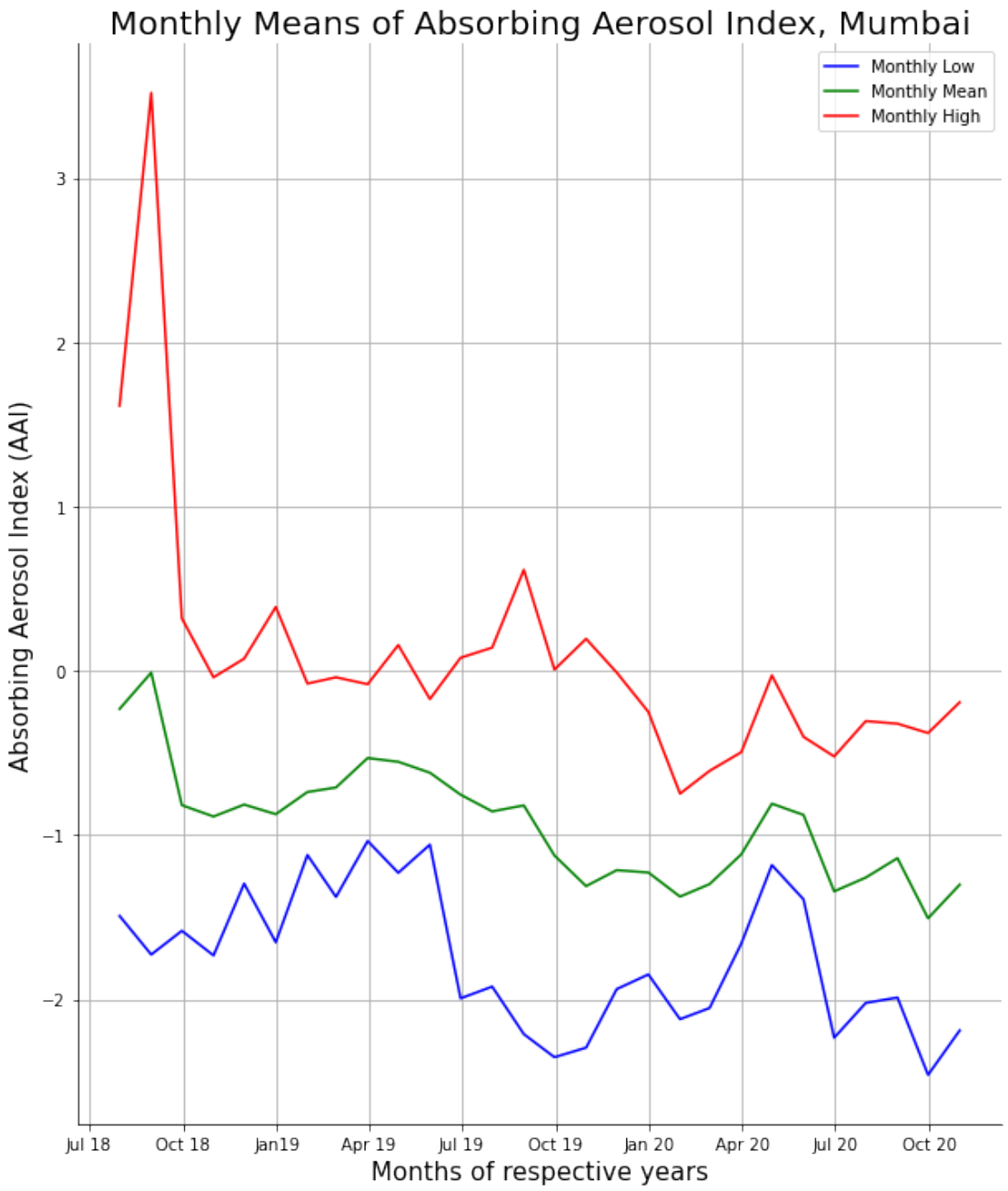




We will now plot some useful insights with Monthly Data

[19]:
```
# Plotting Monthly means
plt.figure(figsize=(10,12))
```

```
plt.plot(list(Monthly_Mumbai_min.index),list(Monthly_Mumbai_min["mean"]), label␣
 ↪= "Monthly Low", color = "blue" )
plt.plot(list(Monthly_Mumbai_mean.index),list(Monthly_Mumbai_mean["mean"]),␣
 ↪label= "Monthly Mean" , color ="green")
plt.plot(list(Monthly_Mumbai_max.index),list(Monthly_Mumbai_max["mean"]), label␣
 ↪= "Monthly High", color = "red" )

plt.xlabel("Months of respective years",fontsize= 15)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Monthly Means of Absorbing Aerosol Index, Mumbai",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 15)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels)
plt.legend()
plt.grid()
```

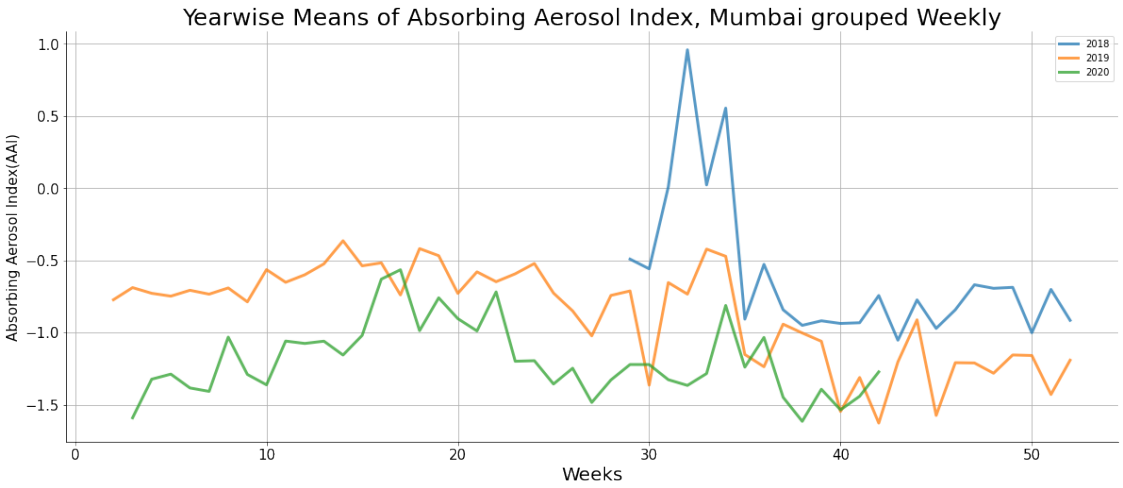


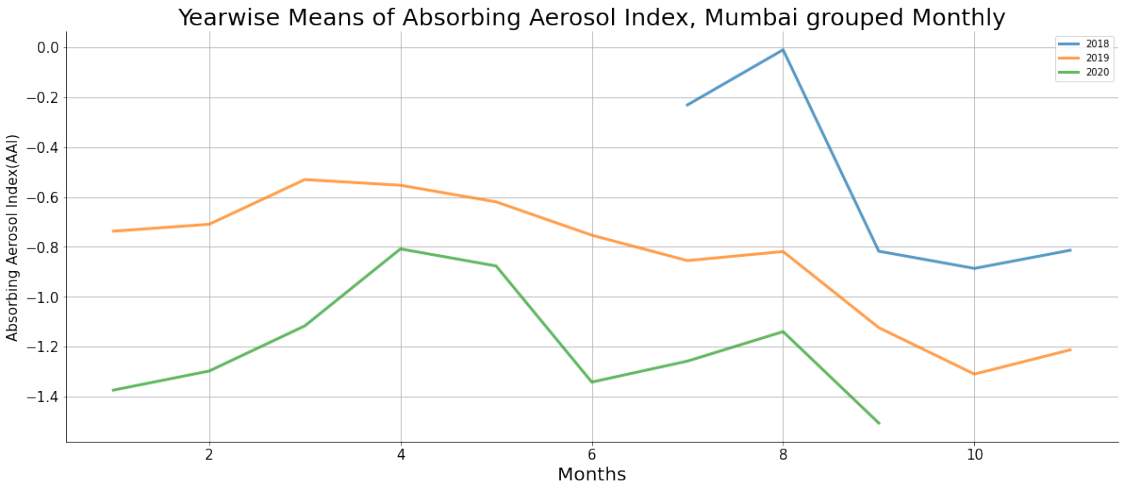
We can observe that Aerosol Optical Depth has decreased in the past year. The change in policies, Even Odd Rule and Coronavirus Pandemic seems to have shifted the curve down.

[20]:
```
# Plotting Yearwise means
x = [i for i in range(1,13)]
plt.figure(figsize = (20,8))
```

```
plt.plot(Weekly_Mumbai_mean[Weekly_Mumbai_mean["year"] == 2018]["week"].
 ↪tolist()[:-1],Weekly_Mumbai_mean[Weekly_Mumbai_mean["year"] == 2018]["mean"].
 ↪tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Weekly_Mumbai_mean[Weekly_Mumbai_mean["year"] == 2019]["week"].
 ↪tolist()[:-1],Weekly_Mumbai_mean[Weekly_Mumbai_mean["year"] == 2019]["mean"].
 ↪tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Weekly_Mumbai_mean[Weekly_Mumbai_mean["year"] == 2020]["week"].
 ↪tolist()[:-1],Weekly_Mumbai_mean[Weekly_Mumbai_mean["year"] == 2020]["mean"].
 ↪tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Weeks",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Yearwise Means of Absorbing Aerosol Index, Mumbai grouped␣
 ↪Weekly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.grid()
plt.legend()

plt.figure(figsize = (20,8))
plt.plot(Monthly_Mumbai_mean[Monthly_Mumbai_mean["year"] == 2018]["month"].
 ↪tolist()[:-1],Monthly_Mumbai_mean[Monthly_Mumbai_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Monthly_Mumbai_mean[Monthly_Mumbai_mean["year"] == 2019]["month"].
 ↪tolist()[:-1],Monthly_Mumbai_mean[Monthly_Mumbai_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Monthly_Mumbai_mean[Monthly_Mumbai_mean["year"] == 2020]["month"].
 ↪tolist()[:-1],Monthly_Mumbai_mean[Monthly_Mumbai_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Months",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.title("Yearwise Means of Absorbing Aerosol Index, Mumbai grouped␣
 ↪Monthly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.grid()
plt.legend()
```

[20]: <matplotlib.legend.Legend at 0x241bc061070>





The observation from the above graph is that, the Absorbing Aerosol Index values have come down considerably in the last two years. From the graph it is evident that the Aerosol Index values are lesser in 2020. Therefore it can be concluded that the polution in Delhi has been lower in 2020 compared to the previous two years.

3.2.2 Plotting Maps For Insights

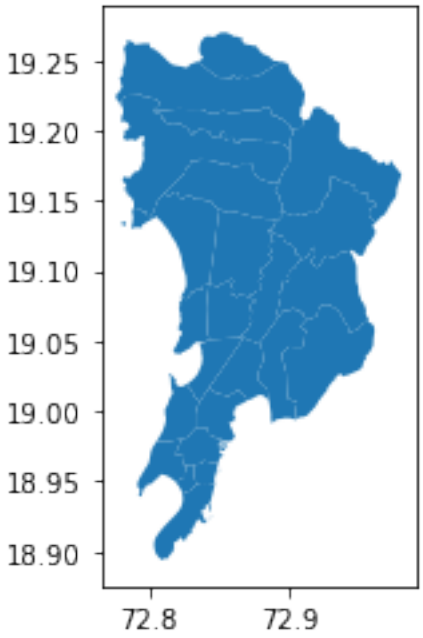
[21]:
```
# Grouping by Wards Data for further analysis
Mumbai_wards = df["Mumbai"].groupby(["Ward_No"]).mean()
Mumbai_wards
```

[21]:
```
              mean
Ward_No
1        -0.972569
```

```
2       -0.943850
3       -0.946406
4       -0.965683
5       -0.937518
6       -0.928589
7       -0.940254
8       -0.912056
9       -0.907992
10      -0.914665
11      -1.005084
12      -0.947979
13      -0.972217
14      -0.919700
15      -0.972282
16      -0.947716
17      -0.917153
18      -0.862986
19      -0.875402
20      -0.934214
21      -0.976226
22      -0.948118
23      -0.907944
24      -0.869281
```

[22]:
```
# loading the shapefile
path = r'Cleaned CSVs/Mumbai/BMC_Wards-polygon.shp'
map_Mumbai = gpd.read_file(path)
# Viewing the shapefile
map_Mumbai.plot()
```

[22]:
```
<AxesSubplot:>
```



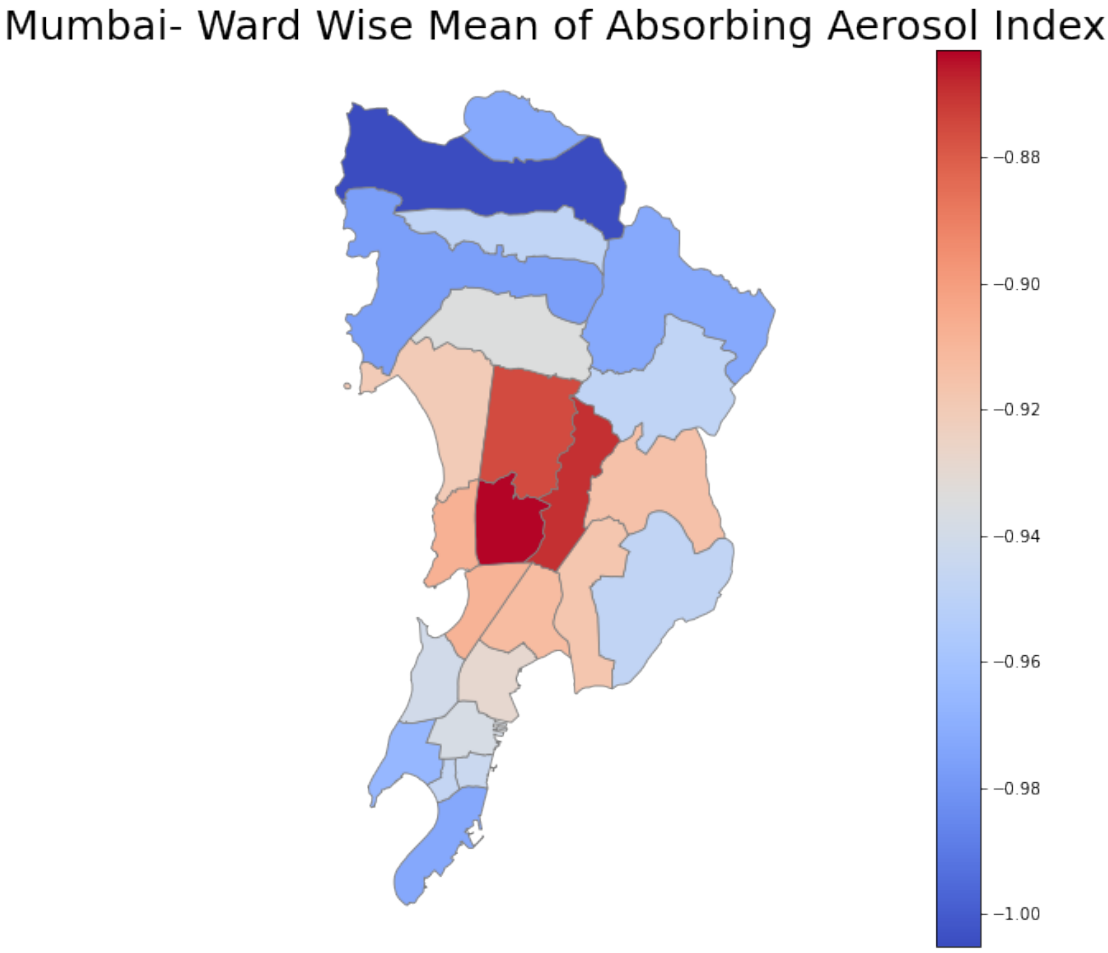
[23]:
```
# merging the shapefile with the mean data
merged_Mumbai = map_Mumbai.merge(Mumbai_wards, left_on = "gid", right_index =␣
↪True)
# merged_Mumbai.head()
merged_Mumbai
```

[23]:

	gid	name	geometry	mean
0	1	A	POLYGON ((72.84025 18.94881, 72.84030 18.94880…	-0.972569
1	2	B	POLYGON ((72.84456 18.96342, 72.84461 18.96342…	-0.943850
2	3	C	POLYGON ((72.83198 18.96174, 72.83197 18.96164…	-0.946406
3	4	D	POLYGON ((72.81873 18.96901, 72.81878 18.96900…	-0.965683
4	5	E	POLYGON ((72.84677 18.98183, 72.84658 18.98143…	-0.937518
5	6	F/S	POLYGON ((72.85625 19.01059, 72.85630 19.01053…	-0.928589
6	7	G/S	POLYGON ((72.82689 19.01942, 72.82691 19.01940…	-0.940254
7	8	F/N	POLYGON ((72.87091 19.05119, 72.87103 19.05113…	-0.912056
8	9	G/N	POLYGON ((72.86699 19.05237, 72.86738 19.05237…	-0.907992
9	10	N	POLYGON ((72.91670 19.10911, 72.91659 19.10893…	-0.914665
10	11	R/C	POLYGON ((72.78986 19.26147, 72.78993 19.26144…	-1.005084
11	12	S	POLYGON ((72.93438 19.16536, 72.93531 19.16487…	-0.947979
12	13	T	POLYGON ((72.94165 19.18778, 72.94194 19.18739…	-0.972217
13	14	K/W	MULTIPOLYGON (((72.78219 19.13248, 72.78170 19…	-0.919700
14	15	R/N	POLYGON ((72.85493 19.26995, 72.85503 19.26989…	-0.972282
15	16	M/E	POLYGON ((72.93774 19.06674, 72.93881 19.06464…	-0.947716
16	17	M/W	POLYGON ((72.90391 19.07285, 72.90416 19.07267…	-0.917153
17	18	H/E	POLYGON ((72.85932 19.08400, 72.85932 19.08397…	-0.862986

```
18   19  K/E  POLYGON ((72.85272 19.14252, 72.85274 19.14248… -0.875402
19   20  P/S  POLYGON ((72.83770 19.17933, 72.83789 19.17932… -0.934214
20   21  P/N  POLYGON ((72.79596 19.22451, 72.79629 19.22437… -0.976226
21   22  R/S  POLYGON ((72.88656 19.21429, 72.88660 19.21396… -0.948118
22   23  H/W  POLYGON ((72.84281 19.09042, 72.84265 19.08943… -0.907944
23   24    L  POLYGON ((72.90801 19.10814, 72.90733 19.10761… -0.869281
```

[24]:
```
# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values
vmin, vmax = merged_Mumbai["mean"].min(), merged_Mumbai["mean"].max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))
# remove the axis
ax.axis('off')
# add a title and annotation
ax.set_title('Mumbai- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))
# empty array for the data range
sm.set_array([])
# add the colorbar to the figure
fig.colorbar(sm)
# create map
merged_Mumbai.plot(column=variable, cmap='coolwarm', linewidth=0.8, ax=ax,␣
 ↪edgecolor='0.5', figsize = (5,15))
plt.show()
```



[25]:
```
# Using T-Test to determine if climate has changed in the first 14 and last 14 months in Mumbai
ttest_ind(Monthly_Mumbai_mean.iloc[:14,:]["mean"].tolist(),Monthly_Mumbai_mean.iloc[14:,:]["mean"].tolist())
```

[25]:
```
Ttest_indResult(statistic=6.458902618381275, pvalue=7.61736056158904e-07)
```

We observe that Pollution levels has decreased in the last 14 months and the statistical test proves the same with a very small p-value for 95 % Confidence Interval

[ ]:
```

```

3.3 City Number 3: Pune

3.3.1 Loading and Visualizing

[26]:
```
# g = df1.groupby(pd.Grouper(freq="M"))
# g.mean()
Pune = df["Pune"].groupby(["date"]).mean()
Pune["month"] = Pune.index.month
Pune["year"] = Pune.index.year
Pune
```

[26]:

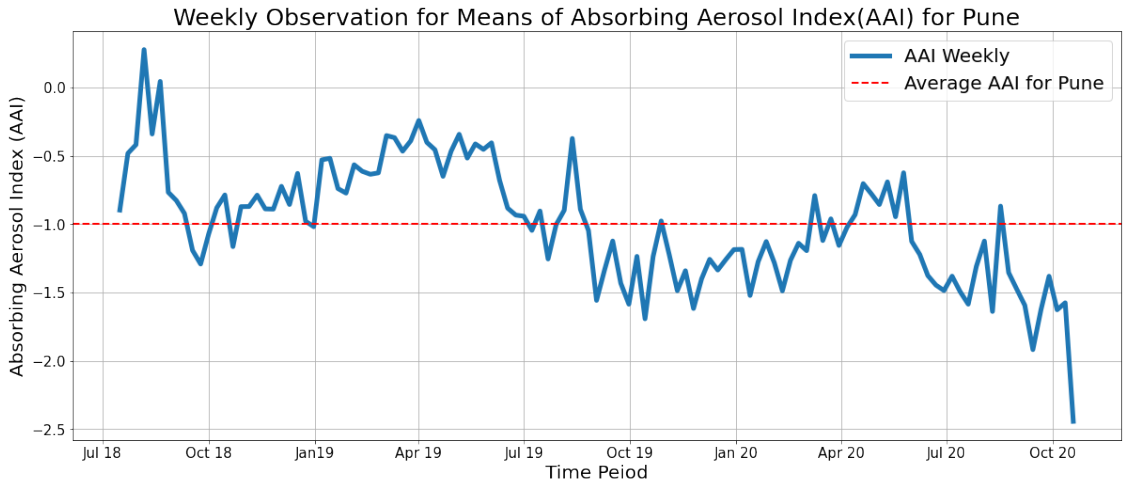
date	mean	Ward_No	month	year
2018-07-11	-0.611108	8	7	2018
2018-07-12	-0.883853	8	7	2018
2018-07-13	-0.807581	8	7	2018
2018-07-14	-1.678436	8	7	2018
2018-07-15	-1.032795	8	7	2018
…	…	…	…	…
2020-10-09	-0.907531	8	10	2020
2020-10-10	-1.887800	8	10	2020
2020-10-11	-1.128160	8	10	2020
2020-10-12	-1.927447	8	10	2020
2020-10-13	-2.441827	8	10	2020

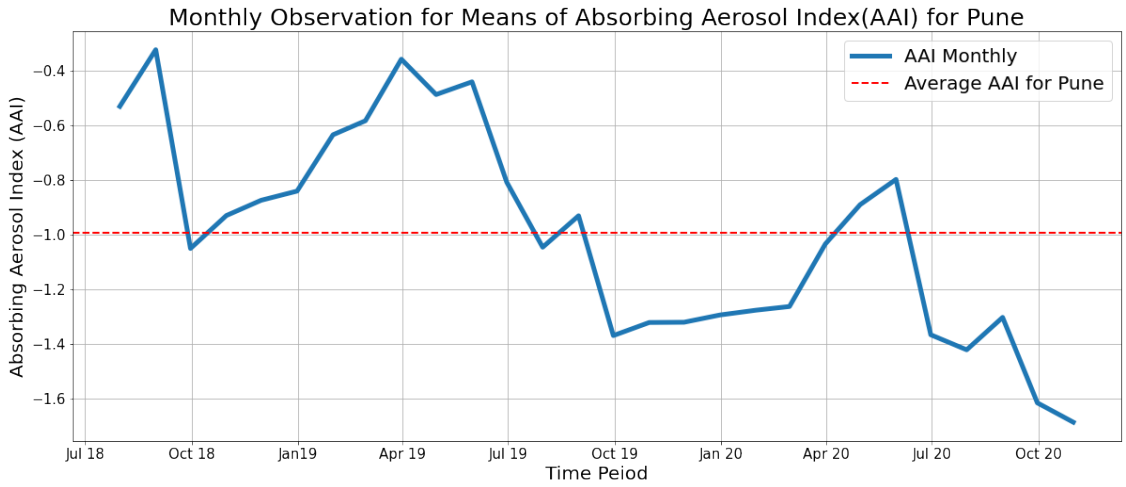
[801 rows x 4 columns]

[27]:
```
# Aggregate statistics grouped by Weeks
Weekly_Pune_mean = Pune.groupby(pd.Grouper(freq="W-MON")).mean()
Weekly_Pune_max = Pune.groupby(pd.Grouper(freq="W-MON")).max()
Weekly_Pune_min = Pune.groupby(pd.Grouper(freq="W-MON")).min()

# Aggregate statistics grouped by Months
Monthly_Pune_mean = Pune.groupby(pd.Grouper(freq="M")).mean()
Monthly_Pune_max = Pune.groupby(pd.Grouper(freq="M")).max()
Monthly_Pune_min = Pune.groupby(pd.Grouper(freq="M")).min()

#Week Identifier
Weekly_Pune_mean["week"] = Weekly_Pune_mean.index.isocalendar().week
Weekly_Pune_min["week"] = Weekly_Pune_min.index.isocalendar().week
Weekly_Pune_max["week"] = Weekly_Pune_max.index.isocalendar().week

Monthly_Pune_mean["week"] = Monthly_Pune_mean.index.isocalendar().week
Monthly_Pune_min["week"] = Monthly_Pune_min.index.isocalendar().week
Monthly_Pune_max["week"] = Monthly_Pune_max.index.isocalendar().week
```

[28]:
```
# Visualizing by grouping into weeekly data
plt.figure(figsize = (20,8))
```

```
plt.plot( Weekly_Pune_mean.index.tolist(),Weekly_Pune_mean["mean"], linewidth =␣
↪5, label = "AAI Weekly")
plt.gca().axhline(y=Weekly_Pune_mean["mean"].mean(),color='r', linestyle='--',␣
↪lw=2, label = "Average AAI for Pune")
plt.title("Weekly Observation for Means of Absorbing Aerosol Index(AAI) for␣
↪Pune",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()

# Visualizing by grouping into Monthly data
plt.figure(figsize = (20,8))
plt.plot( Monthly_Pune_mean.index.tolist(),Monthly_Pune_mean["mean"], linewidth␣
↪= 5, label = "AAI Monthly")
plt.gca().axhline(y=Monthly_Pune_mean["mean"].mean(),color='r', linestyle='--',␣
↪lw=2, label = "Average AAI for Pune")
plt.title("Monthly Observation for Means of Absorbing Aerosol Index(AAI) for␣
↪Pune",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()
```

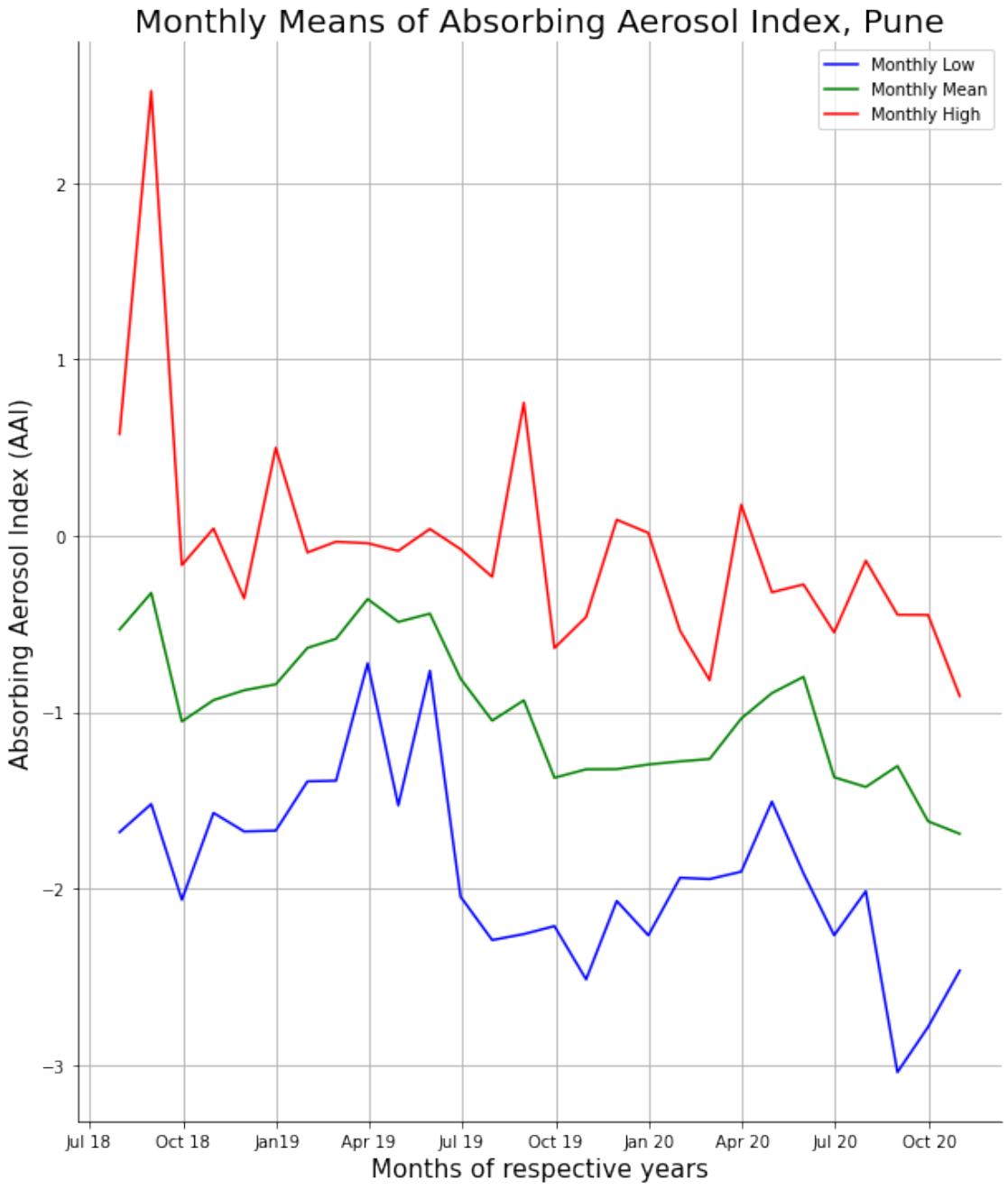




We will now plot some useful insights with Monthly Data

[29]:
```
# Plotting Monthly means
plt.figure(figsize=(10,12))
plt.plot(list(Monthly_Pune_min.index),list(Monthly_Pune_min["mean"]), label =␣
↪"Monthly Low", color = "blue" )
plt.plot(list(Monthly_Pune_mean.index),list(Monthly_Pune_mean["mean"]), label=␣
↪"Monthly Mean" , color ="green")
plt.plot(list(Monthly_Pune_max.index),list(Monthly_Pune_max["mean"]), label =␣
↪"Monthly High", color = "red" )

plt.xlabel("Months of respective years",fontsize= 15)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
```

```
plt.title("Monthly Means of Absorbing Aerosol Index, Pune",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 15)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels)
plt.legend()
plt.grid()
```

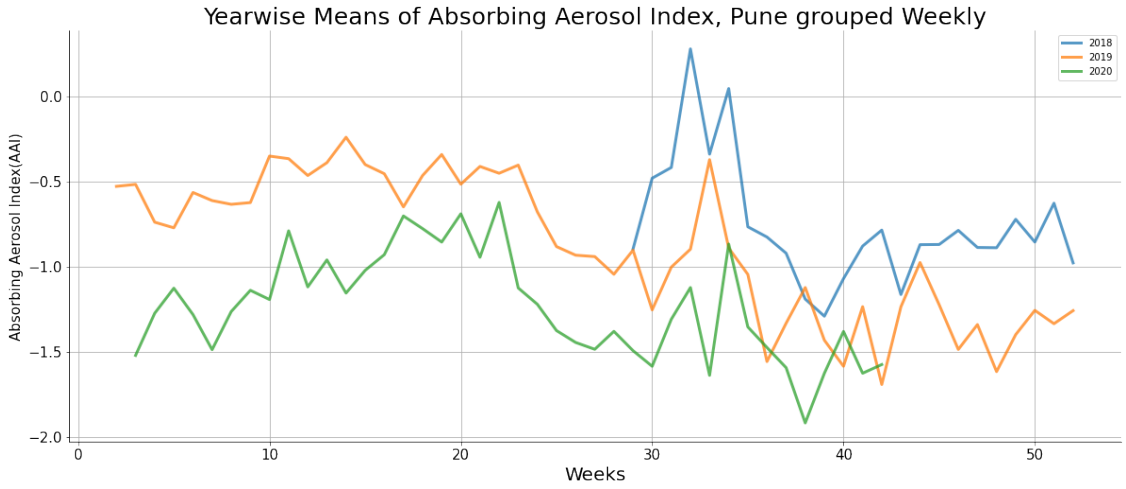


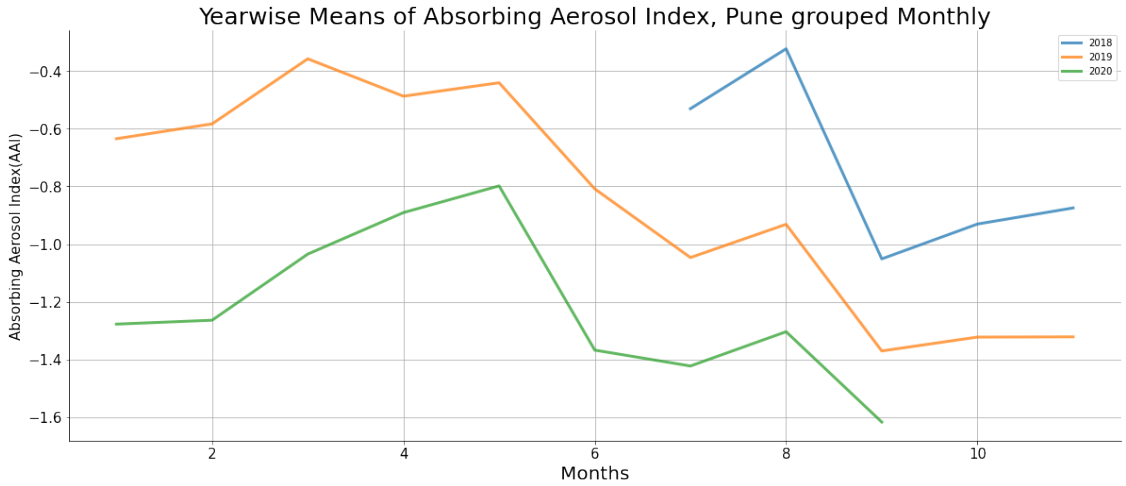
We can observe that Aerosol Optical Depth has decreased in the past year. The change in policies, Even Odd Rule and Coronavirus Pandemic seems to have shifted the curve down.

[30]:
```
# Plotting Yearwise means
x = [i for i in range(1,13)]
plt.figure(figsize = (20,8))
plt.plot(Weekly_Pune_mean[Weekly_Pune_mean["year"] == 2018]["week"].tolist()[:
 ↪-1],Weekly_Pune_mean[Weekly_Pune_mean["year"] == 2018]["mean"].tolist()[:-1]␣
 ↪, label = "2018", lw = 3, alpha = 0.75)
plt.plot(Weekly_Pune_mean[Weekly_Pune_mean["year"] == 2019]["week"].tolist()[:
 ↪-1],Weekly_Pune_mean[Weekly_Pune_mean["year"] == 2019]["mean"].tolist()[:-1]␣
 ↪, label = "2019" , lw =3, alpha = 0.75)
plt.plot(Weekly_Pune_mean[Weekly_Pune_mean["year"] == 2020]["week"].tolist()[:
 ↪-1],Weekly_Pune_mean[Weekly_Pune_mean["year"] == 2020]["mean"].tolist()[:-1]␣
 ↪ , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Weeks",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Yearwise Means of Absorbing Aerosol Index, Pune grouped␣
 ↪Weekly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.grid()
plt.legend()

plt.figure(figsize = (20,8))
plt.plot(Monthly_Pune_mean[Monthly_Pune_mean["year"] == 2018]["month"].
 ↪tolist()[:-1],Monthly_Pune_mean[Monthly_Pune_mean["year"] == 2018]["mean"].
 ↪tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Monthly_Pune_mean[Monthly_Pune_mean["year"] == 2019]["month"].
 ↪tolist()[:-1],Monthly_Pune_mean[Monthly_Pune_mean["year"] == 2019]["mean"].
 ↪tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Monthly_Pune_mean[Monthly_Pune_mean["year"] == 2020]["month"].
 ↪tolist()[:-1],Monthly_Pune_mean[Monthly_Pune_mean["year"] == 2020]["mean"].
 ↪tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Months",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.title("Yearwise Means of Absorbing Aerosol Index, Pune grouped␣
 ↪Monthly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.grid()
```

```
plt.legend()
```

[30]: <matplotlib.legend.Legend at 0x241bc565100>





The observation from the above graph is that, the Absorbing Aerosol Index values have come down considerably in the last two years. From the graph it is evident that the Aerosol Index values are lesser in 2020. Therefore it can be concluded that the polution in Delhi has been lower in 2020 compared to the previous two years.

3.3.2 Plotting Maps For Insights

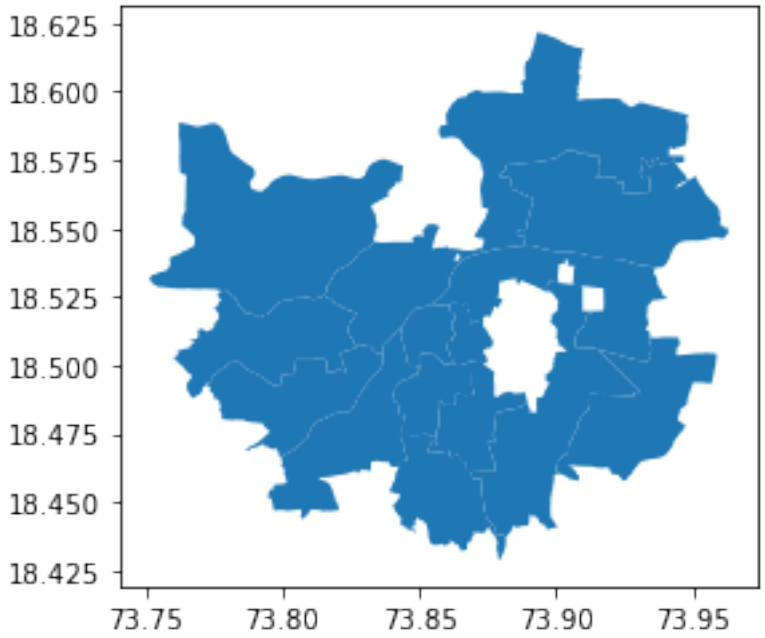
[31]:
```
# Grouping by Wards Data for further analysis
Pune_wards = df["Pune"].groupby(["Ward_No"]).mean()
Pune_wards
```

[31]:

Ward_No	mean
1	-1.015877
2	-1.011385
3	-1.021526
4	-0.992819
5	-0.983372
6	-0.984705
7	-0.964731
8	-0.984172
9	-0.972935
10	-0.957330
11	-0.958587
12	-0.981931
13	-0.988044
14	-0.977707
15	-0.979958

[32]:
```
# loading the shapefile
path = r'Cleaned CSVs/Pune/pune-admin-wards-polygon.shp'
map_Pune = gpd.read_file(path)
map_Pune["Ward_No"] = [i for i in range(1,16)]
# Viewing the shapefile
map_Pune.plot()
```

[32]: <AxesSubplot:>



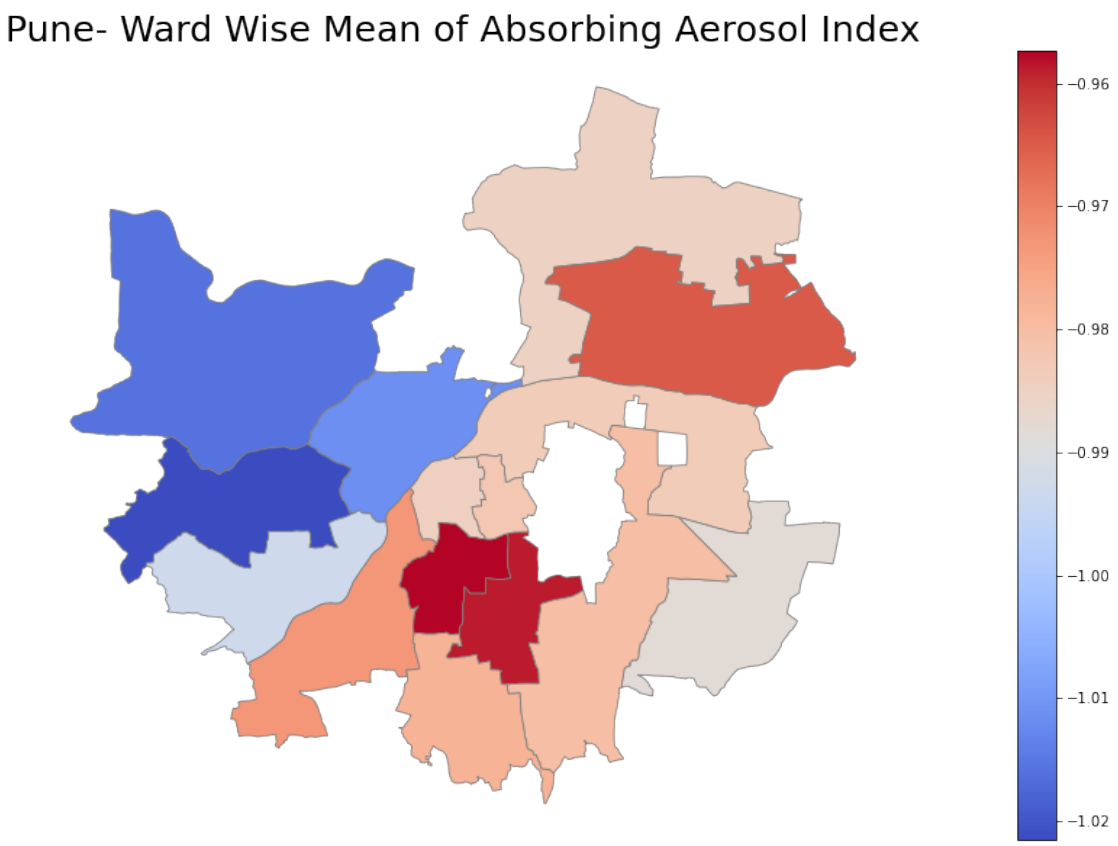
[33]:
```
# merging the shapefile with the mean data
merged_Pune = map_Pune.merge(Pune_wards, left_on = "Ward_No", right_index =␣
↪True)
# merged_Pune.head()
merged_Pune
```

[33]:

	name	geometry	Ward_No	mean
0	Admin Ward 01 Aundh	POLYGON ((73.82476 18.53713, 73.82464 18.53704…	1	-1.015877
1	Admin Ward 02 Ghole Road			
2	Admin Ward 03 Kothrud Karveroad			
3	Admin Ward 04 Warje Karvenagar			
4	Admin Ward 05 Dhole Patil Rd			
5	Admin Ward 06 Yerawda - Sangamwadi			
6	Admin Ward 07 Nagar Road			
7	Admin Ward 08 KasbaVishrambaugwada			
8	Admin Ward 09 Tilak Road			
9	Admin Ward 10 Sahakarnagar			
10	Admin Ward 11 Bibwewadi			
11	Admin Ward 12 Bhavani Peth			
12	Admin Ward 13 Hadapsar			
13	Admin Ward 14 Dhankawadi			
14	Admin Ward 15 Kondhwa Wanavdi			

```
1   POLYGON ((73.86190 18.53324, 73.86131 18.53207…        2 -1.011385
2   POLYGON ((73.77529 18.49650, 73.77416 18.49594…        3 -1.021526
3   POLYGON ((73.81155 18.47852, 73.81078 18.47732…        4 -0.992819
4   POLYGON ((73.88839 18.54376, 73.88864 18.54376…        5 -0.983372
5   POLYGON ((73.87439 18.55410, 73.87464 18.55532…        6 -0.984705
6   POLYGON ((73.88859 18.54538, 73.88872 18.54556…        7 -0.964731
7   POLYGON ((73.85081 18.50257, 73.85084 18.50156…        8 -0.984172
8   POLYGON ((73.82598 18.48445, 73.82600 18.48450…        9 -0.972935
9   POLYGON ((73.86339 18.48789, 73.86334 18.48622…       10 -0.957330
10  POLYGON ((73.85631 18.47448, 73.85626 18.47524…       11 -0.958587
11  POLYGON ((73.87321 18.50540, 73.87324 18.50537…       12 -0.981931
12  POLYGON ((73.93209 18.46513, 73.93170 18.46507…       13 -0.988044
13  POLYGON ((73.87959 18.42900, 73.87950 18.42891…       14 -0.977707
14  POLYGON ((73.89715 18.53003, 73.89735 18.53057…       15 -0.979958
```

[34]:
```
# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values
vmin, vmax = merged_Pune["mean"].min(), merged_Pune["mean"].max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))
# remove the axis
ax.axis('off')
# add a title and annotation
ax.set_title('Pune- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))
# empty array for the data range
sm.set_array([])
# add the colorbar to the figure
fig.colorbar(sm)
# create map
merged_Pune.plot(column=variable, cmap='coolwarm', linewidth=0.8, ax=ax,␣
 ↪edgecolor='0.5', figsize = (5,15))
plt.show()
```



[35]:
```
# Using T-Test to determine if climate has changed in the first 14 and last 14 months in Pune
ttest_ind(Monthly_Pune_mean.iloc[:14,:]["mean"].tolist(),Monthly_Pune_mean.iloc[14:,:]["mean"].tolist())
```

[35]:
```
Ttest_indResult(statistic=6.204768266156896, pvalue=1.4549075945828928e-06)
```

We observe that Pollution levels has decreased in the last 14 months and the statistical test proves the same with a very small p-value for 95 % Confidence Interval

[ ]:
```

```

3.4 City Number 4 Ahmedabad

3.4.1 Loading and Visualizing

[36]:
```
# g = df1.groupby(pd.Grouper(freq="M"))
# g.mean()
Ahmedabad = df["Ahmedabad"].groupby(["date"]).mean()
Ahmedabad["month"] = Ahmedabad.index.month
Ahmedabad["year"] = Ahmedabad.index.year
```

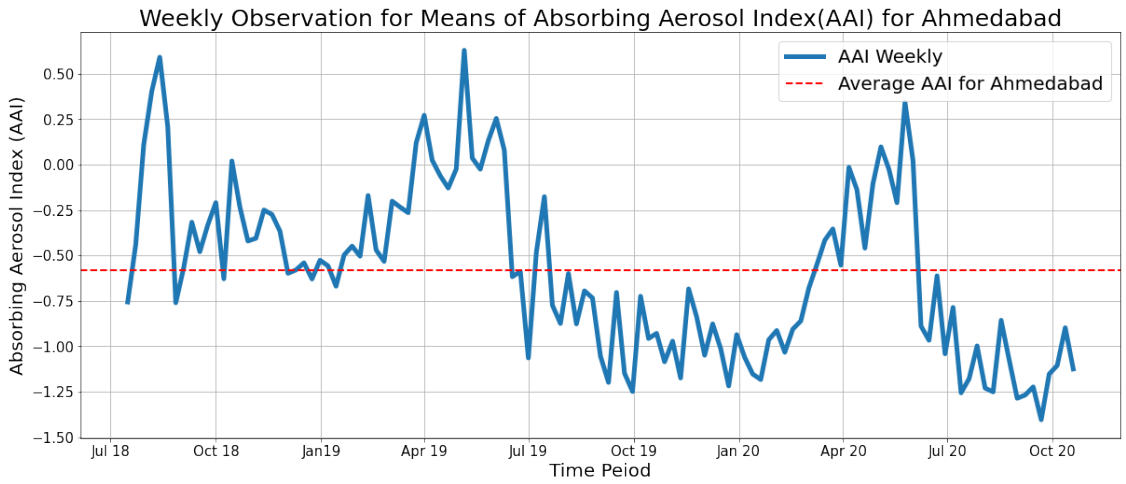
```
Ahmedabad
```

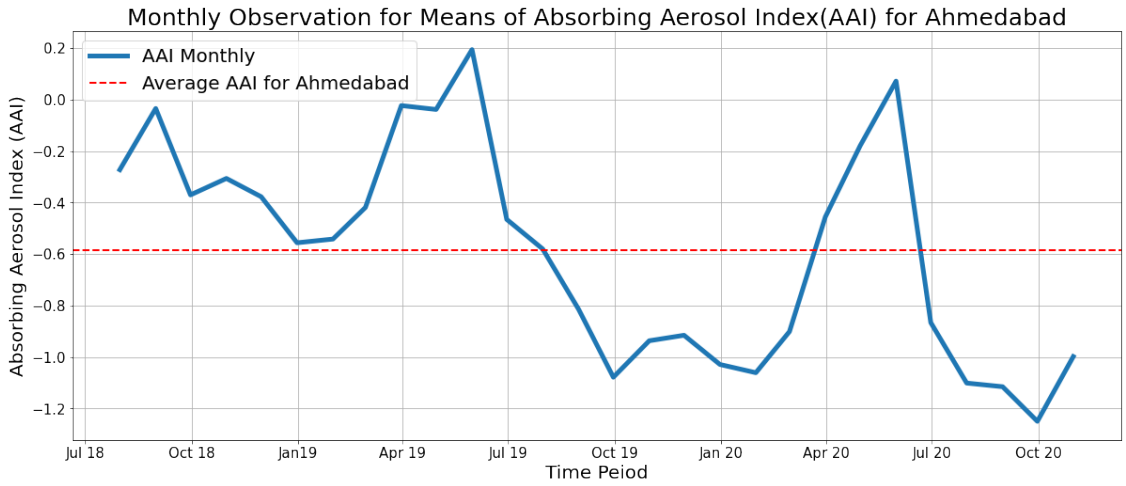
[36]:
```
                mean  Ward_No  month  year
date
2018-07-11 -0.286013     24.5      7  2018
2018-07-12 -0.496056     24.5      7  2018
2018-07-13 -1.052356     24.5      7  2018
2018-07-14 -1.225006     24.5      7  2018
2018-07-15 -0.997644     24.5      7  2018
…                 …      …    …     …
2020-10-09 -0.737113     24.5     10  2020
2020-10-10 -0.712917     24.5     10  2020
2020-10-11 -1.143606     24.5     10  2020
2020-10-12 -1.407086     24.5     10  2020
2020-10-13 -1.124830     24.5     10  2020

[799 rows x 4 columns]
```

[37]:
```
# Aggregate statistics grouped by Weeks
Weekly_Ahmedabad_mean = Ahmedabad.groupby(pd.Grouper(freq="W-MON")).mean()
Weekly_Ahmedabad_max = Ahmedabad.groupby(pd.Grouper(freq="W-MON")).max()
Weekly_Ahmedabad_min = Ahmedabad.groupby(pd.Grouper(freq="W-MON")).min()

# Aggregate statistics grouped by Months
Monthly_Ahmedabad_mean = Ahmedabad.groupby(pd.Grouper(freq="M")).mean()
Monthly_Ahmedabad_max = Ahmedabad.groupby(pd.Grouper(freq="M")).max()
Monthly_Ahmedabad_min = Ahmedabad.groupby(pd.Grouper(freq="M")).min()

#Week Identifier
Weekly_Ahmedabad_mean["week"] = Weekly_Ahmedabad_mean.index.isocalendar().week
Weekly_Ahmedabad_min["week"] = Weekly_Ahmedabad_min.index.isocalendar().week
Weekly_Ahmedabad_max["week"] = Weekly_Ahmedabad_max.index.isocalendar().week

Monthly_Ahmedabad_mean["week"] = Monthly_Ahmedabad_mean.index.isocalendar().week
Monthly_Ahmedabad_min["week"] = Monthly_Ahmedabad_min.index.isocalendar().week
Monthly_Ahmedabad_max["week"] = Monthly_Ahmedabad_max.index.isocalendar().week
```

[38]:
```
# Visualizing by grouping into weeekly data
plt.figure(figsize = (20,8))
plt.plot( Weekly_Ahmedabad_mean.index.tolist(),Weekly_Ahmedabad_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Weekly")
plt.gca().axhline(y=Weekly_Ahmedabad_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Ahmedabad")
plt.title("Weekly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Ahmedabad",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
```

```
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()

# Visualizing by grouping into Monthly data
plt.figure(figsize = (20,8))
plt.plot( Monthly_Ahmedabad_mean.index.tolist(),Monthly_Ahmedabad_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Monthly")
plt.gca().axhline(y=Monthly_Ahmedabad_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Ahmedabad")
plt.title("Monthly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Ahmedabad",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()
```





We will now plot some useful insights with Monthly Data

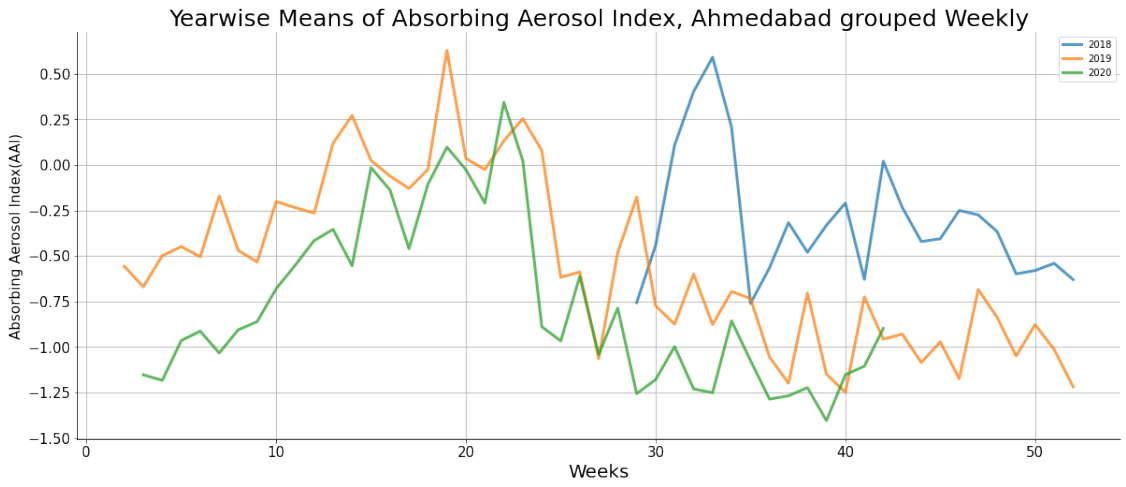
[39]:
```
# Plotting Monthly means
plt.figure(figsize=(10,12))
plt.plot(list(Monthly_Ahmedabad_min.index),list(Monthly_Ahmedabad_min["mean"]),␣
↪label = "Monthly Low", color = "blue" )
plt.plot(list(Monthly_Ahmedabad_mean.
↪index),list(Monthly_Ahmedabad_mean["mean"]), label= "Monthly Mean" , color␣
↪="green")
plt.plot(list(Monthly_Ahmedabad_max.index),list(Monthly_Ahmedabad_max["mean"]),␣
↪label = "Monthly High", color = "red" )

plt.xlabel("Months of respective years",fontsize= 15)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Monthly Means of Absorbing Aerosol Index, Ahmedabad",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 15)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels)
plt.legend()
plt.grid()
```

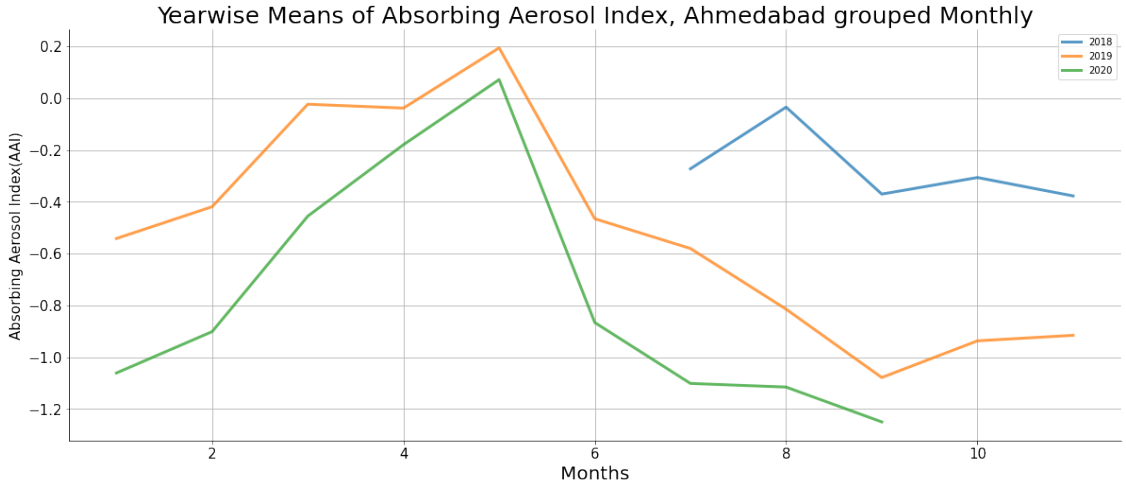
We can observe that Aerosol Optical Depth has decreased in the past year. The change in policies, Even Odd Rule and Coronavirus Pandemic seems to have shifted the curve down.

[40]:
```
# Plotting Yearwise means
x = [i for i in range(1,13)]
plt.figure(figsize = (20,8))
```

```
plt.plot(Weekly_Ahmedabad_mean[Weekly_Ahmedabad_mean["year"] == 2018]["week"].
 ↪tolist()[:-1],Weekly_Ahmedabad_mean[Weekly_Ahmedabad_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Weekly_Ahmedabad_mean[Weekly_Ahmedabad_mean["year"] == 2019]["week"].
 ↪tolist()[:-1],Weekly_Ahmedabad_mean[Weekly_Ahmedabad_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Weekly_Ahmedabad_mean[Weekly_Ahmedabad_mean["year"] == 2020]["week"].
 ↪tolist()[:-1],Weekly_Ahmedabad_mean[Weekly_Ahmedabad_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Weeks",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Yearwise Means of Absorbing Aerosol Index, Ahmedabad grouped␣
 ↪Weekly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.grid()
plt.legend()

plt.figure(figsize = (20,8))
plt.plot(Monthly_Ahmedabad_mean[Monthly_Ahmedabad_mean["year"] ==␣
 ↪2018]["month"].tolist()[:
 ↪-1],Monthly_Ahmedabad_mean[Monthly_Ahmedabad_mean["year"] == 2018]["mean"].
 ↪tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Monthly_Ahmedabad_mean[Monthly_Ahmedabad_mean["year"] ==␣
 ↪2019]["month"].tolist()[:
 ↪-1],Monthly_Ahmedabad_mean[Monthly_Ahmedabad_mean["year"] == 2019]["mean"].
 ↪tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Monthly_Ahmedabad_mean[Monthly_Ahmedabad_mean["year"] ==␣
 ↪2020]["month"].tolist()[:
 ↪-1],Monthly_Ahmedabad_mean[Monthly_Ahmedabad_mean["year"] == 2020]["mean"].
 ↪tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Months",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.title("Yearwise Means of Absorbing Aerosol Index, Ahmedabad grouped␣
 ↪Monthly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.grid()
plt.legend()
```

[40]: <matplotlib.legend.Legend at 0x241b88ed0a0>





The observation from the above graph is that, the Absorbing Aerosol Index values have come down considerably in the last two years. From the graph it is evident that the Aerosol Index values are lesser in 2020. Therefore it can be concluded that the polution in Delhi has been lower in 2020 compared to the previous two years.

3.4.2 Plotting Maps For Insights

[41]:
```
# Grouping by Wards Data for further analysis
Ahmedabad_wards = df["Ahmedabad"].groupby(["Ward_No"]).mean()
Ahmedabad_wards
```

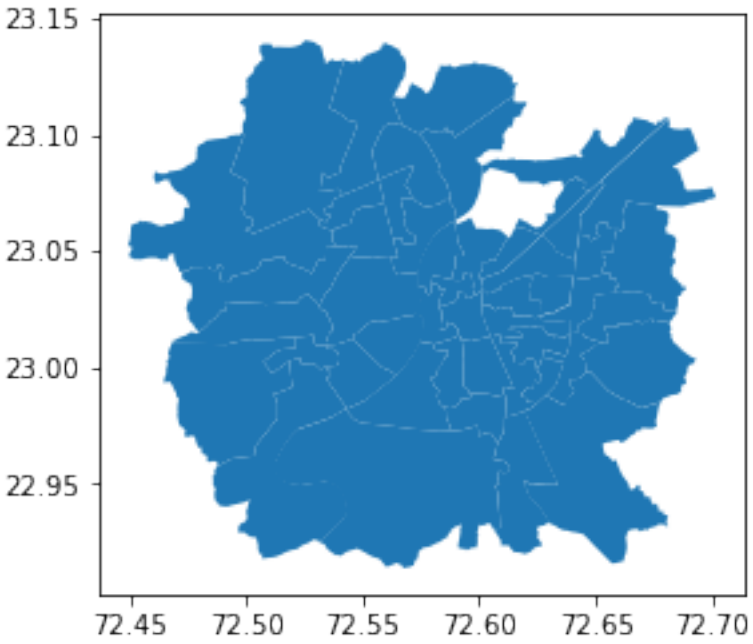
[41]:

Ward_No	mean
1	-0.709633
2	-0.666840
3	-0.734087
4	-0.656032
5	-0.577231
6	-0.520955
7	-0.513250
8	-0.701475
9	-0.495739
10	-0.528073
11	-0.689321
12	-0.705965
13	-0.545090
14	-0.583522
15	-0.540409
16	-0.545686
17	-0.531741
18	-0.513521
19	-0.602611
20	-0.602529
21	-0.524465
22	-0.512847
23	-0.513863
24	-0.619141
25	-0.496569
26	-0.487520
27	-0.497117
28	-0.508904
29	-0.517268
30	-0.533651
31	-0.576172
32	-0.575772
33	-0.719837
34	-0.794942
35	-0.581155
36	-0.519846
37	-0.501464
38	-0.491208
39	-0.491315
40	-0.566312
41	-0.671548
42	-0.537350
43	-0.524201
44	-0.508749
45	-0.547657

```
46      -0.748029
47      -0.720290
48      -0.761592
```

[42]:
```
# loading the shapefile
path = r'Cleaned CSVs/Ahmedabad/Wards-polygon.shp'
map_Ahmedabad = gpd.read_file(path)
x = map_Ahmedabad["Name"].tolist()
map_Ahmedabad["Ward_No"] = [i[:2] for i in x]
map_Ahmedabad["Ward_No"] =(map_Ahmedabad["Ward_No"]).astype(int)
# Viewing the shapefile
map_Ahmedabad.plot()
```

[42]:
```
<AxesSubplot:>
```



[43]:
```
# merging the shapefile with the mean data
merged_Ahmedabad = map_Ahmedabad.merge(Ahmedabad_wards, left_on = "Ward_No",␣
 ↪right_index = True)
# merged_Ahmedabad.head()
merged_Ahmedabad
```

[43]:
```
                    Name                                           geometry  \
0        48 RAMOL HATHIJAN  POLYGON Z ((72.64795 22.99784 0.00000, 72.6500…
1                47 VATVA  POLYGON Z ((72.61962 22.98128 0.00000, 72.6211…
2               46 LAMBHA  POLYGON Z ((72.58829 22.97285 0.00000, 72.5909…
```

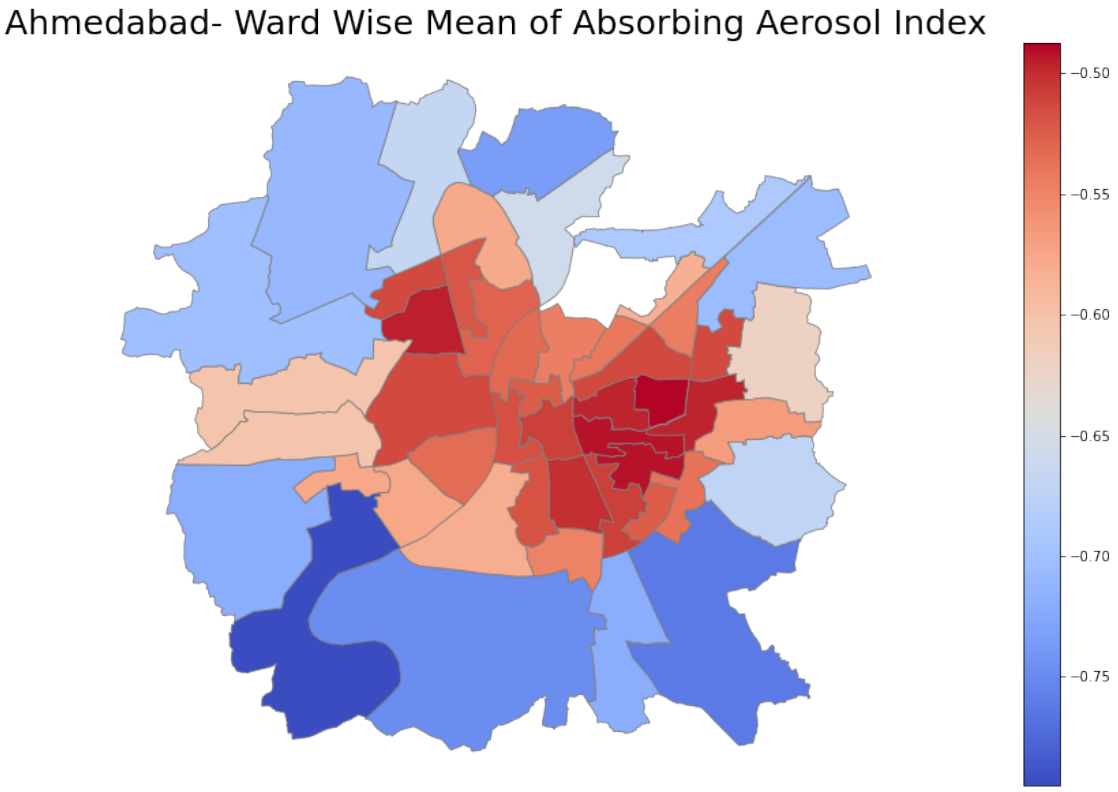
3	45	ISANPUR	POLYGON Z ((72.60390 22.98860 0.00000, 72.6092…
4	44	KHOKHRA	POLYGON Z ((72.60813 23.01394 0.00000, 72.6092…
5	43	BHAIPURA HATKESHWAR	POLYGON Z ((72.62793 23.00426 0.00000, 72.6284…
6	42	INDRAPURI	POLYGON Z ((72.64074 23.01490 0.00000, 72.6418…
7	41	VASTRAL	POLYGON Z ((72.67465 23.01994 0.00000, 72.6747…
8	40	ODHAV	POLYGON Z ((72.66042 23.03202 0.00000, 72.6610…
9	39	AMRAIWADI	POLYGON Z ((72.62154 23.01715 0.00000, 72.6218…
10	38	GOMTIPUR	POLYGON Z ((72.60590 23.02544 0.00000, 72.6065…
11	37	MANINAGAR	POLYGON Z ((72.60492 23.01313 0.00000, 72.6060…
12	36	DANILIMDA	POLYGON Z ((72.59167 23.01342 0.00000, 72.5919…
13	35	BAHERAMPURA	POLYGON Z ((72.58646 23.00754 0.00000, 72.5866…
14	34	MAKTAMPURA	POLYGON Z ((72.52318 23.00431 0.00000, 72.5234…
15	33	SARKHEJ	POLYGON Z ((72.47342 23.01025 0.00000, 72.4784…
16	32	VEJALPUR	POLYGON Z ((72.52873 23.01414 0.00000, 72.5289…
17	31	VASNA	POLYGON Z ((72.54647 23.01143 0.00000, 72.5468…
18	30	PALDI	POLYGON Z ((72.57342 23.02270 0.00000, 72.5760…
19	29	JAMALPUR	POLYGON Z ((72.57956 23.03572 0.00000, 72.5797…
20	28	KHADIA	POLYGON Z ((72.59494 23.02957 0.00000, 72.5950…
21	27	SARASPUR-RAKHIYAL	POLYGON Z ((72.62310 23.03961 0.00000, 72.6219…
22	26	BAPUNAGAR	POLYGON Z ((72.62996 23.03884 0.00000, 72.6304…
23	25	VIRATNAGAR	POLYGON Z ((72.65639 23.04071 0.00000, 72.6561…
24	24	NIKOL	POLYGON Z ((72.67657 23.06808 0.00000, 72.6769…
25	23	THAKKARBAPANAGAR	POLYGON Z ((72.65266 23.06241 0.00000, 72.6526…
26	22	INDIA COLONY	POLYGON Z ((72.62726 23.05679 0.00000, 72.6285…
27	21	DARIYAPUR	POLYGON Z ((72.59482 23.04045 0.00000, 72.5949…
28	20	JODHPUR	POLYGON Z ((72.49273 23.02824 0.00000, 72.4949…
29	19	BODAKDEV	POLYGON Z ((72.53629 23.05221 0.00000, 72.5385…
30	18	NAVRANGPURA	POLYGON Z ((72.55426 23.04696 0.00000, 72.5545…
31	17	SHAHPUR	POLYGON Z ((72.58978 23.06325 0.00000, 72.5902…
32	16	SHAHIBAG	POLYGON Z ((72.59705 23.06304 0.00000, 72.5972…
33	15	ASARWA	POLYGON Z ((72.61630 23.06023 0.00000, 72.6165…
34	14	KUBERNAGR	POLYGON Z ((72.64227 23.08174 0.00000, 72.6424…
35	13	SAIJPUR BOGHA	POLYGON Z ((72.65244 23.07938 0.00000, 72.6526…
36	12	NARODA	POLYGON Z ((72.68280 23.10264 0.00000, 72.6833…
37	11	SARDARNAGAR	POLYGON Z ((72.66653 23.10711 0.00000, 72.6682…
38	10	S.P.STADIUM	POLYGON Z ((72.57187 23.07250 0.00000, 72.5720…
39	09	NARANPURA	POLYGON Z ((72.55903 23.07093 0.00000, 72.5594…
40	08	THALTEJ	POLYGON Z ((72.49864 23.10063 0.00000, 72.4988…
41	07	GHATLODIA	POLYGON Z ((72.55681 23.06994 0.00000, 72.5566…
42	06	NEW WADAJ	POLYGON Z ((72.57210 23.08634 0.00000, 72.5717…
43	05	RANIP	POLYGON Z ((72.58045 23.09122 0.00000, 72.5831…
44	04	SABARMATI	POLYGON Z ((72.61467 23.10556 0.00000, 72.6138…
45	03	CHANDKHEDA	POLYGON Z ((72.58419 23.12945 0.00000, 72.5850…
46	02	CHANDLODIA	POLYGON Z ((72.55911 23.13933 0.00000, 72.5596…
47	01	GOTA	POLYGON Z ((72.50776 23.13886 0.00000, 72.5088…

Ward_No	mean

```
0        48 -0.761592
1        47 -0.720290
2        46 -0.748029
3        45 -0.547657
4        44 -0.508749
5        43 -0.524201
6        42 -0.537350
7        41 -0.671548
8        40 -0.566312
9        39 -0.491315
10       38 -0.491208
11       37 -0.501464
12       36 -0.519846
13       35 -0.581155
14       34 -0.794942
15       33 -0.719837
16       32 -0.575772
17       31 -0.576172
18       30 -0.533651
19       29 -0.517268
20       28 -0.508904
21       27 -0.497117
22       26 -0.487520
23       25 -0.496569
24       24 -0.619141
25       23 -0.513863
26       22 -0.512847
27       21 -0.524465
28       20 -0.602529
29       19 -0.602611
30       18 -0.513521
31       17 -0.531741
32       16 -0.545686
33       15 -0.540409
34       14 -0.583522
35       13 -0.545090
36       12 -0.705965
37       11 -0.689321
38       10 -0.528073
39        9 -0.495739
40        8 -0.701475
41        7 -0.513250
42        6 -0.520955
43        5 -0.577231
44        4 -0.656032
45        3 -0.734087
46        2 -0.666840
```

```
47        1 -0.709633
```

[44]:
```
# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values
vmin, vmax = merged_Ahmedabad["mean"].min(), merged_Ahmedabad["mean"].max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))
# remove the axis
ax.axis('off')
# add a title and annotation
ax.set_title('Ahmedabad- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))
# empty array for the data range
sm.set_array([])
# add the colorbar to the figure
fig.colorbar(sm)
# create map
merged_Ahmedabad.plot(column=variable, cmap='coolwarm', linewidth=0.8, ax=ax,␣
 ↪edgecolor='0.5', figsize = (5,15))
plt.show()
```



[45]:
```
# Using T-Test to determine if climate has changed in the first 14 and last 14␣
↪months in Ahmedabad
ttest_ind(Monthly_Ahmedabad_mean.iloc[:14,:]["mean"].
↪tolist(),Monthly_Ahmedabad_mean.iloc[14:,:]["mean"].tolist())
```

[45]:
```
Ttest_indResult(statistic=4.093274734894954, pvalue=0.0003664231460856578)
```

We observe that Pollution levels has decreased in the last 14 months and the statistical test proves the same with a very small p-value for 95 % Confidence Interval

[ ]:
```

```

3.5 City Number 5 Bangalore

3.5.1 Loading and Visualizing

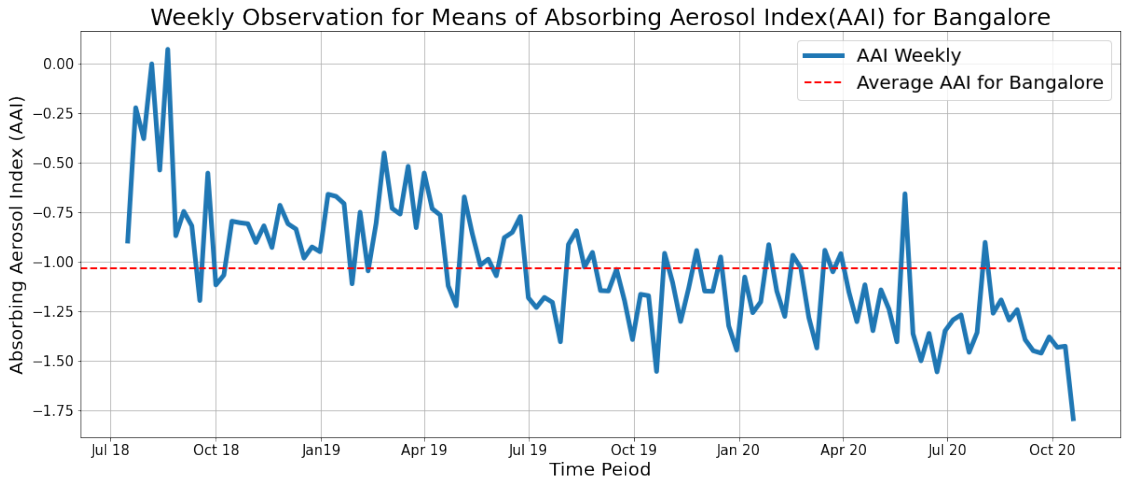
[46]:
```
# g = df1.groupby(pd.Grouper(freq="M"))
# g.mean()
Bangalore = df["Bangalore"].groupby(["date"]).mean()
Bangalore["month"] = Bangalore.index.month
Bangalore["year"] = Bangalore.index.year
Bangalore
```

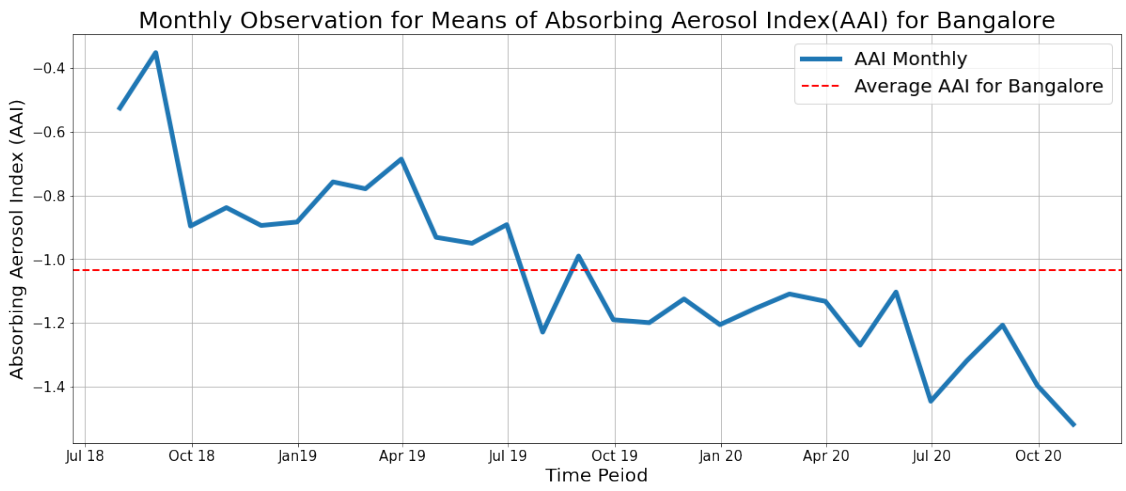
[46]:
```
            Ward_No      mean  month  year
date
2018-07-11     99.5 -0.456807      7  2018
2018-07-12     99.5 -0.425492      7  2018
2018-07-13     99.5 -0.235959      7  2018
2018-07-14     99.5 -1.893831      7  2018
2018-07-15     99.5 -0.984660      7  2018
…               …        …     …    …
2020-10-09     99.5 -0.970991     10  2020
2020-10-10     99.5 -0.937902     10  2020
2020-10-11     99.5 -1.491334     10  2020
2020-10-12     99.5 -1.461066     10  2020
2020-10-13     99.5 -1.793873     10  2020

[803 rows x 4 columns]
```

[47]:
```
# Aggregate statistics grouped by Weeks
Weekly_Bangalore_mean = Bangalore.groupby(pd.Grouper(freq="W-MON")).mean()
Weekly_Bangalore_max = Bangalore.groupby(pd.Grouper(freq="W-MON")).max()
Weekly_Bangalore_min = Bangalore.groupby(pd.Grouper(freq="W-MON")).min()

# Aggregate statistics grouped by Months
Monthly_Bangalore_mean = Bangalore.groupby(pd.Grouper(freq="M")).mean()
Monthly_Bangalore_max = Bangalore.groupby(pd.Grouper(freq="M")).max()
Monthly_Bangalore_min = Bangalore.groupby(pd.Grouper(freq="M")).min()

#Week Identifier
Weekly_Bangalore_mean["week"] = Weekly_Bangalore_mean.index.isocalendar().week
Weekly_Bangalore_min["week"] = Weekly_Bangalore_min.index.isocalendar().week
Weekly_Bangalore_max["week"] = Weekly_Bangalore_max.index.isocalendar().week

Monthly_Bangalore_mean["week"] = Monthly_Bangalore_mean.index.isocalendar().week
Monthly_Bangalore_min["week"] = Monthly_Bangalore_min.index.isocalendar().week
Monthly_Bangalore_max["week"] = Monthly_Bangalore_max.index.isocalendar().week
```

[48]:
```
# Visualizing by grouping into weeekly data
plt.figure(figsize = (20,8))
plt.plot( Weekly_Bangalore_mean.index.tolist(),Weekly_Bangalore_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Weekly")
plt.gca().axhline(y=Weekly_Bangalore_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Bangalore")
plt.title("Weekly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Bangalore",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
```

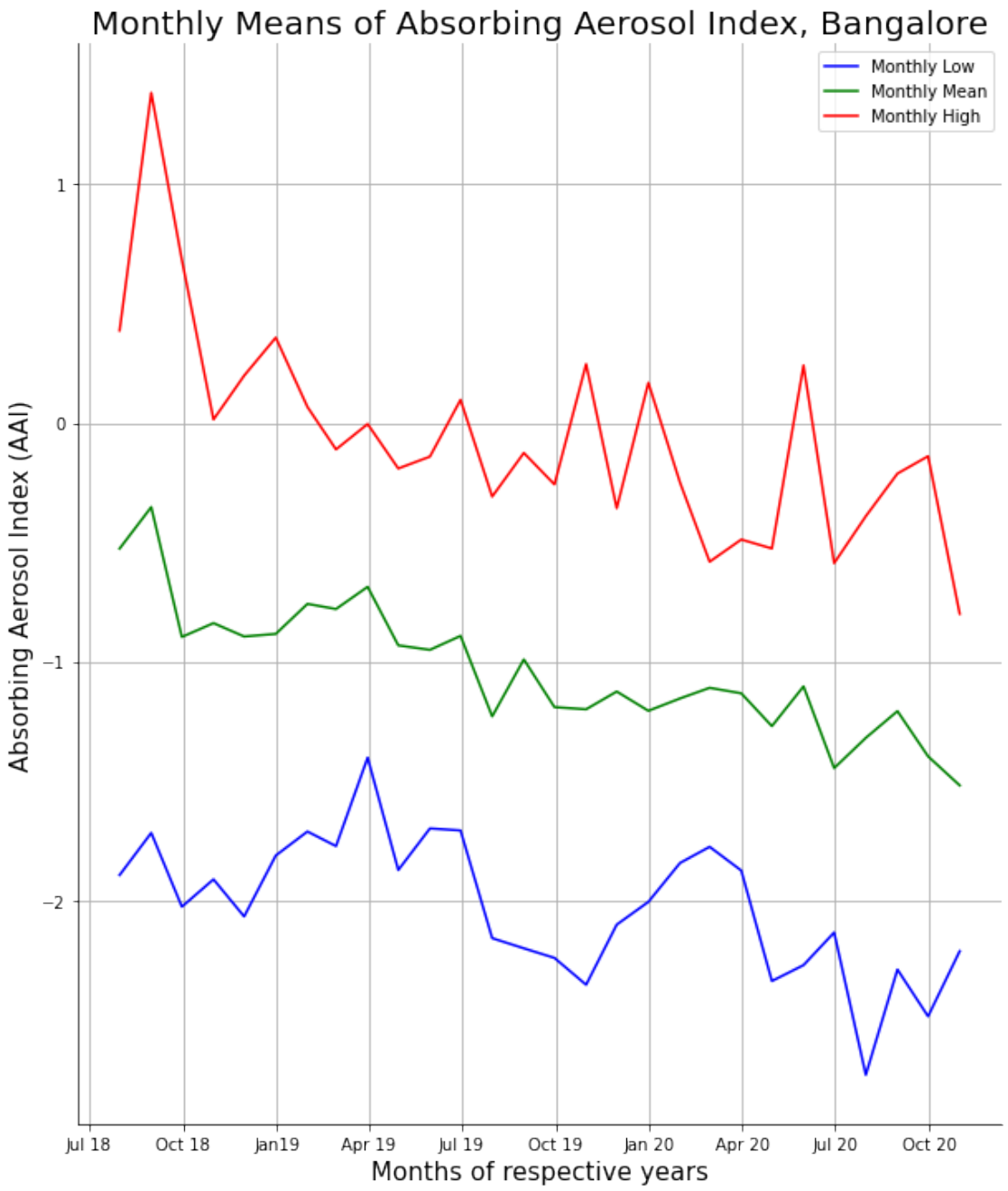
```
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()

# Visualizing by grouping into Monthly data
plt.figure(figsize = (20,8))
plt.plot( Monthly_Bangalore_mean.index.tolist(),Monthly_Bangalore_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Monthly")
plt.gca().axhline(y=Monthly_Bangalore_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Bangalore")
plt.title("Monthly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Bangalore",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()
```





We will now plot some useful insights with Monthly Data

[49]:
```
# Plotting Monthly means
plt.figure(figsize=(10,12))
plt.plot(list(Monthly_Bangalore_min.index),list(Monthly_Bangalore_min["mean"]),␣
↪label = "Monthly Low", color = "blue" )
plt.plot(list(Monthly_Bangalore_mean.
↪index),list(Monthly_Bangalore_mean["mean"]), label= "Monthly Mean" , color␣
↪="green")
plt.plot(list(Monthly_Bangalore_max.index),list(Monthly_Bangalore_max["mean"]),␣
↪label = "Monthly High", color = "red" )

plt.xlabel("Months of respective years",fontsize= 15)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Monthly Means of Absorbing Aerosol Index, Bangalore",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 15)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels)
plt.legend()
plt.grid()
```

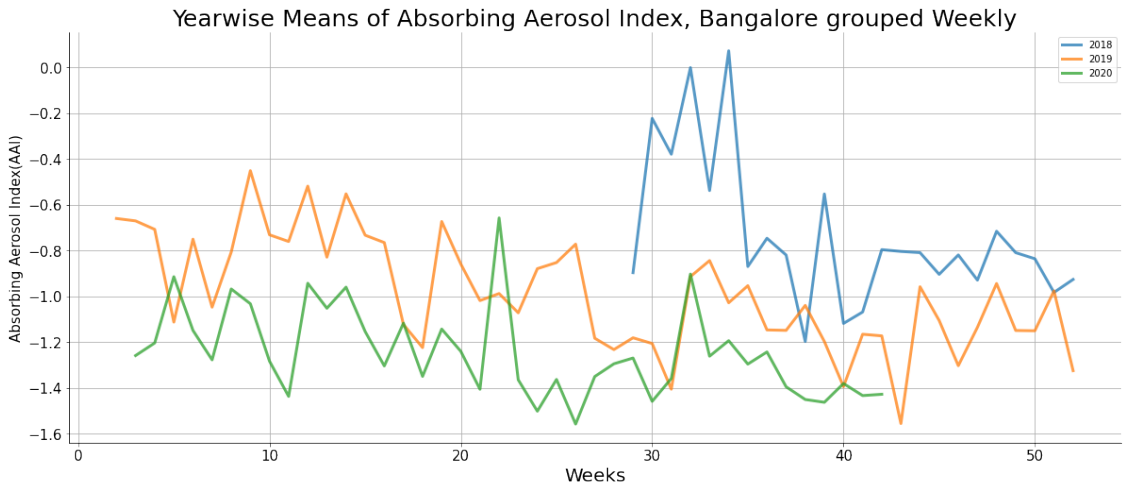


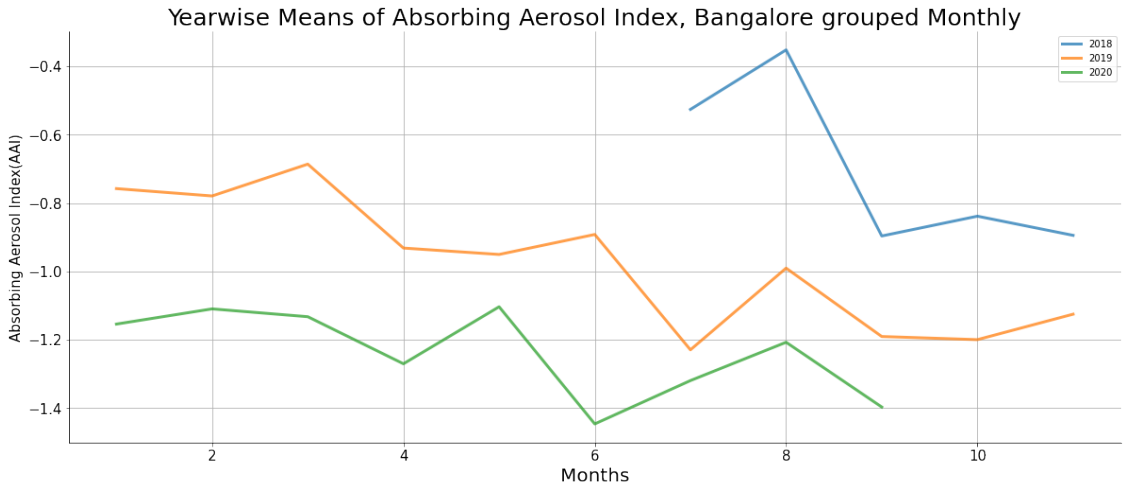
We can observe that Aerosol Optical Depth has decreased in the past year. The change in policies, Even Odd Rule and Coronavirus Pandemic seems to have shifted the curve down.

[50]:
```
# Plotting Yearwise means
x = [i for i in range(1,13)]
plt.figure(figsize = (20,8))
```

```
plt.plot(Weekly_Bangalore_mean[Weekly_Bangalore_mean["year"] == 2018]["week"].
 ↪tolist()[:-1],Weekly_Bangalore_mean[Weekly_Bangalore_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Weekly_Bangalore_mean[Weekly_Bangalore_mean["year"] == 2019]["week"].
 ↪tolist()[:-1],Weekly_Bangalore_mean[Weekly_Bangalore_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Weekly_Bangalore_mean[Weekly_Bangalore_mean["year"] == 2020]["week"].
 ↪tolist()[:-1],Weekly_Bangalore_mean[Weekly_Bangalore_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Weeks",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Yearwise Means of Absorbing Aerosol Index, Bangalore grouped␣
 ↪Weekly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.grid()
plt.legend()

plt.figure(figsize = (20,8))
plt.plot(Monthly_Bangalore_mean[Monthly_Bangalore_mean["year"] ==␣
 ↪2018]["month"].tolist()[:
 ↪-1],Monthly_Bangalore_mean[Monthly_Bangalore_mean["year"] == 2018]["mean"].
 ↪tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Monthly_Bangalore_mean[Monthly_Bangalore_mean["year"] ==␣
 ↪2019]["month"].tolist()[:
 ↪-1],Monthly_Bangalore_mean[Monthly_Bangalore_mean["year"] == 2019]["mean"].
 ↪tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Monthly_Bangalore_mean[Monthly_Bangalore_mean["year"] ==␣
 ↪2020]["month"].tolist()[:
 ↪-1],Monthly_Bangalore_mean[Monthly_Bangalore_mean["year"] == 2020]["mean"].
 ↪tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Months",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.title("Yearwise Means of Absorbing Aerosol Index, Bangalore grouped␣
 ↪Monthly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.grid()
plt.legend()
```

[50]: <matplotlib.legend.Legend at 0x241bb8c9be0>





The observation from the above graph is that, the Absorbing Aerosol Index values have come down considerably in the last two years. From the graph it is evident that the Aerosol Index values are lesser in 2020. Therefore it can be concluded that the polution in Delhi has been lower in 2020 compared to the previous two years.

3.5.2 Plotting Maps For Insights

[51]:
```
# Grouping by Wards Data for further analysis
Bangalore_wards = df["Bangalore"].groupby(["Ward_No"]).mean()
Bangalore_wards
```

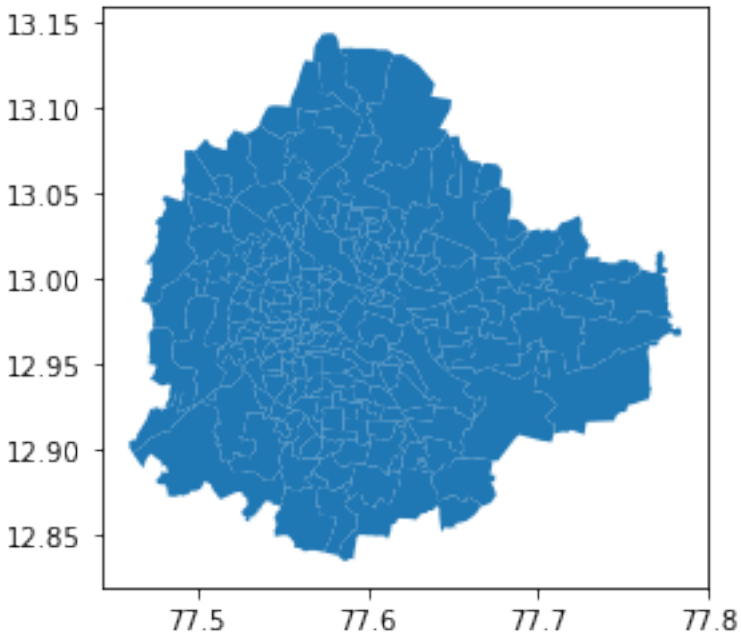
[51]:

Ward_No	mean
1	-1.159950
2	-1.173467
3	-1.165083
4	-1.139294
5	-1.133939
…	…
194	-1.095522
195	-1.043933
196	-1.118013
197	-1.044735
198	-1.122094

[198 rows x 1 columns]

[52]:
```
# loading the shapefile
path = r'Cleaned CSVs/Bangalore/BBMP-polygon.shp'
map_Bangalore = gpd.read_file(path)
# Viewing the shapefile
map_Bangalore.plot()
```

[52]: <AxesSubplot:>



[53]:
```
# merging the shapefile with the mean data
merged_Bangalore = map_Bangalore.merge(Bangalore_wards, left_on = "WARD_NO",␣
 ↪right_index = True)
# merged_Bangalore.head()
merged_Bangalore
```

[53]:

	OBJECTID	ASS_CONST_	ASS_CONST1	WARD_NO	WARD_NAME	\
0	1	150	Yelahanka	2.0	Chowdeswari Ward	
1	2	150	Yelahanka	3.0	Atturu	
2	3	150	Yelahanka	4.0	Yelahanka Satellite Town	
3	4	151	K.R. Puram	51.0	Vijnanapura	
4	5	151	K.R. Puram	53.0	Basavanapura	
..	…	…	…	…	…	
193	194	172	B.T.M. Layout	172.0	Madivala	
194	195	151	K.R. Puram	26.0	Ramamurthy Nagar	
195	196	151	K.R. Puram	25.0	Horamavu	
196	197	174	Mahadevapura	86.0	Marathahalli	
197	198	153	Yeshwantpura	198.0	Hemmigepura	

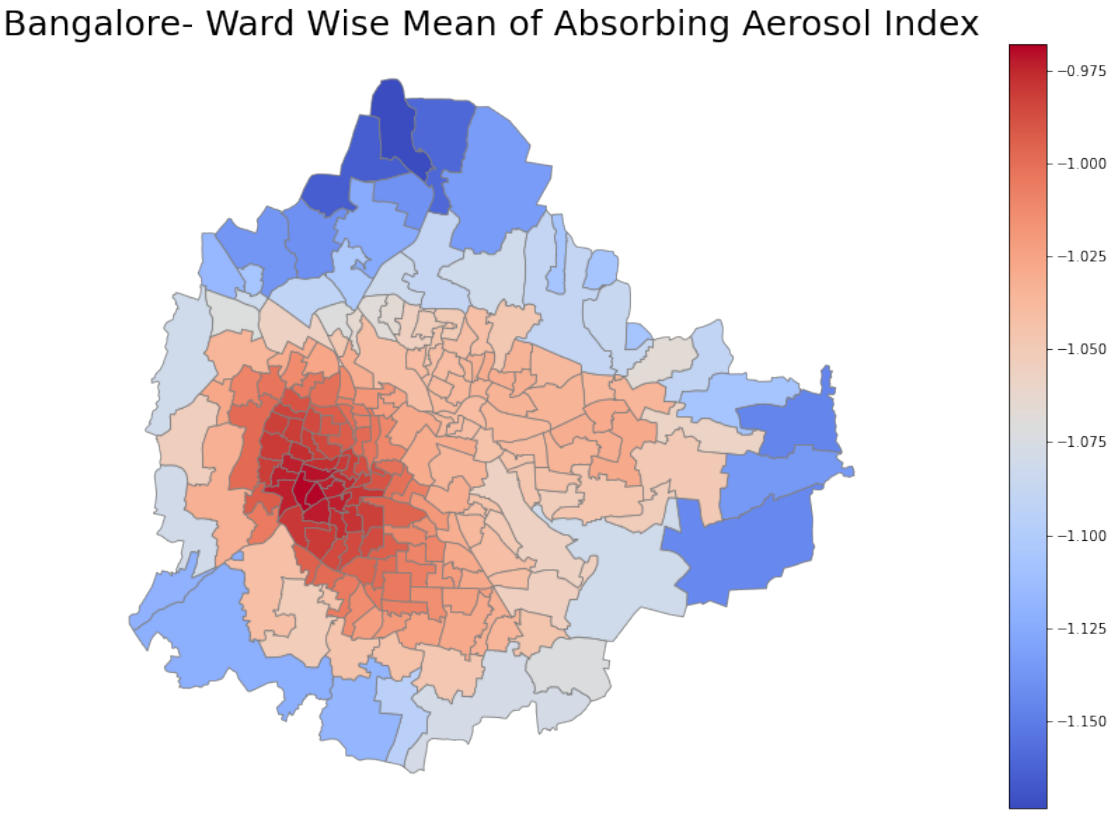
	POP_M	POP_F	POP_SC	POP_ST	POP_TOTAL	AREA_SQ_KM	LAT	\
0	10402.0	9224.0	2630.0	286.0	19626.0	7.06	13.121709	
1	13129.0	10891.0	2921.0	665.0	24020.0	10.15	13.102805	
2	13457.0	12325.0	3687.0	601.0	25782.0	4.90	13.090987	
3	18118.0	16969.0	6454.0	228.0	35087.0	2.05	13.006063	
4	11494.0	10518.0	4115.0	325.0	22012.0	6.28	13.016847	
..	…	…	…	…	…	…	…	
193	19121.0	16034.0	2658.0	555.0	35155.0	1.16	12.920018	
194	11330.0	10669.0	3489.0	247.0	21999.0	7.87	13.033613	
195	14696.0	13471.0	5677.0	666.0	28167.0	17.32	13.044561	
196	11967.0	10522.0	4126.0	274.0	22489.0	3.10	12.950743	
197	12926.0	11385.0	3997.0	319.0	24311.0	28.59	12.891903	

	LON	RESERVATIO	\
0	77.580422	General	
1	77.560038	General (Women)	
2	77.583925	Backward Category - A	
3	77.669565	Scheduled Caste	
4	77.715456	General	
..	…	…	
193	77.614418	Backward Category - A	
194	77.676539	Backward Category - A	
195	77.653271	General (Women)	
196	77.691495	General (Women)	
197	77.505013	Backward Category A (Women)	

	geometry	mean
0	POLYGON ((77.59229 13.09720, 77.59094 13.09842…	-1.173467

```
1    POLYGON ((77.56862 13.12705, 77.57064 13.12654… -1.165083
2    POLYGON ((77.59094 13.09842, 77.59229 13.09720… -1.139294
3    POLYGON ((77.67683 13.01147, 77.67695 13.01149… -1.036743
4    POLYGON ((77.72899 13.02061, 77.72994 13.01995… -1.090538
..                                                 …       …
193  POLYGON ((77.61399 12.92347, 77.61419 12.92344… -1.027605
194  POLYGON ((77.68336 13.05192, 77.68384 13.05067… -1.086044
195  POLYGON ((77.64931 13.07853, 77.64993 13.07701… -1.088657
196  POLYGON ((77.68549 12.94121, 77.68539 12.94129… -1.052418
197  MULTIPOLYGON (((77.49854 12.92574, 77.49854 12… -1.122094

[198 rows x 16 columns]
```

[54]:
```
# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values
vmin, vmax = merged_Bangalore["mean"].min(), merged_Bangalore["mean"].max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))
# remove the axis
ax.axis('off')
# add a title and annotation
ax.set_title('Bangalore- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))
# empty array for the data range
sm.set_array([])
# add the colorbar to the figure
fig.colorbar(sm)
# create map
merged_Bangalore.plot(column=variable, cmap='coolwarm', linewidth=0.8, ax=ax,␣
 ↪edgecolor='0.5', figsize = (5,15))
plt.show()
```



[55]:
```
# Using T-Test to determine if climate has changed in the first 14 and last 14 months in Bangalore
ttest_ind(Monthly_Bangalore_mean.iloc[:14,:]["mean"].tolist(),Monthly_Bangalore_mean.iloc[14:,:]["mean"].tolist())
```

[55]:
```
Ttest_indResult(statistic=6.214691795052053, pvalue=1.4184315032691761e-06)
```

We observe that Pollution levels has decreased in the last 14 months and the statistical test proves the same with a very small p-value for 95 % Confidence Interval

[ ]:
```

```

3.6 City Number 6: Bhopal

3.6.1 Loading and Visualizing

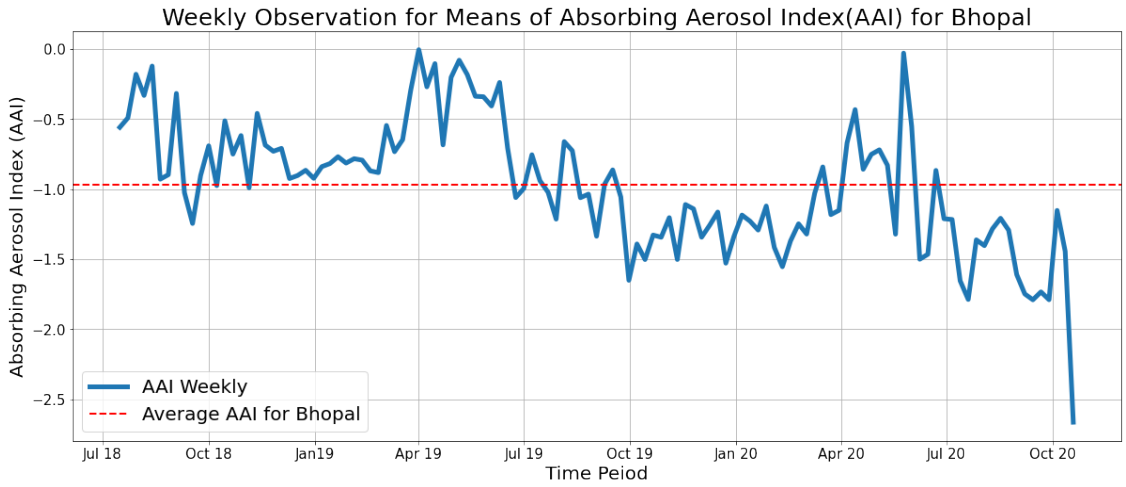
[56]:
```
# g = df1.groupby(pd.Grouper(freq="M"))
# g.mean()
Bhopal = df["Bhopal"].groupby(["date"]).mean()
Bhopal["month"] = Bhopal.index.month
Bhopal["year"] = Bhopal.index.year
Bhopal
```

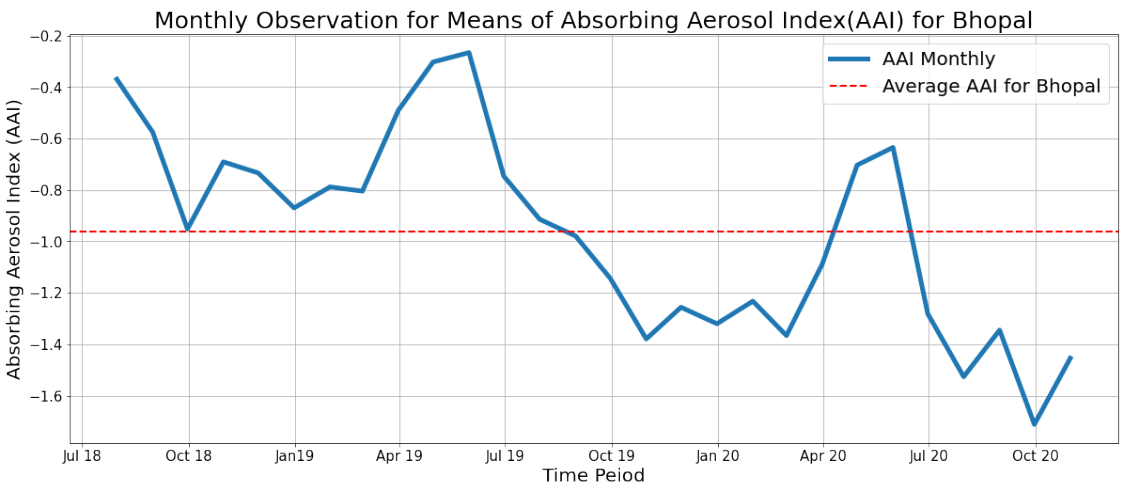
[56]:
```
              Ward_No      mean  month  year
date
2018-07-11  43.383721 -0.267739      7  2018
2018-07-12  43.383721 -0.004302      7  2018
2018-07-13  43.383721 -1.028058      7  2018
2018-07-14  43.383721 -1.125820      7  2018
2018-07-15  43.383721 -0.219812      7  2018
…                 …        …    …    …
2020-10-09  43.383721 -1.079642     10  2020
2020-10-10  43.383721 -1.055856     10  2020
2020-10-11  43.383721 -1.372294     10  2020
2020-10-12  43.383721 -1.831518     10  2020
2020-10-13  43.383721 -2.663199     10  2020

[797 rows x 4 columns]
```

[57]:
```
# Aggregate statistics grouped by Weeks
Weekly_Bhopal_mean = Bhopal.groupby(pd.Grouper(freq="W-MON")).mean()
Weekly_Bhopal_max = Bhopal.groupby(pd.Grouper(freq="W-MON")).max()
Weekly_Bhopal_min = Bhopal.groupby(pd.Grouper(freq="W-MON")).min()

# Aggregate statistics grouped by Months
Monthly_Bhopal_mean = Bhopal.groupby(pd.Grouper(freq="M")).mean()
Monthly_Bhopal_max = Bhopal.groupby(pd.Grouper(freq="M")).max()
Monthly_Bhopal_min = Bhopal.groupby(pd.Grouper(freq="M")).min()

#Week Identifier
Weekly_Bhopal_mean["week"] = Weekly_Bhopal_mean.index.isocalendar().week
Weekly_Bhopal_min["week"] = Weekly_Bhopal_min.index.isocalendar().week
Weekly_Bhopal_max["week"] = Weekly_Bhopal_max.index.isocalendar().week

Monthly_Bhopal_mean["week"] = Monthly_Bhopal_mean.index.isocalendar().week
Monthly_Bhopal_min["week"] = Monthly_Bhopal_min.index.isocalendar().week
Monthly_Bhopal_max["week"] = Monthly_Bhopal_max.index.isocalendar().week
```

[58]:
```
# Visualizing by grouping into weeekly data
plt.figure(figsize = (20,8))
plt.plot( Weekly_Bhopal_mean.index.tolist(),Weekly_Bhopal_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Weekly")
plt.gca().axhline(y=Weekly_Bhopal_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Bhopal")
plt.title("Weekly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Bhopal",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
```

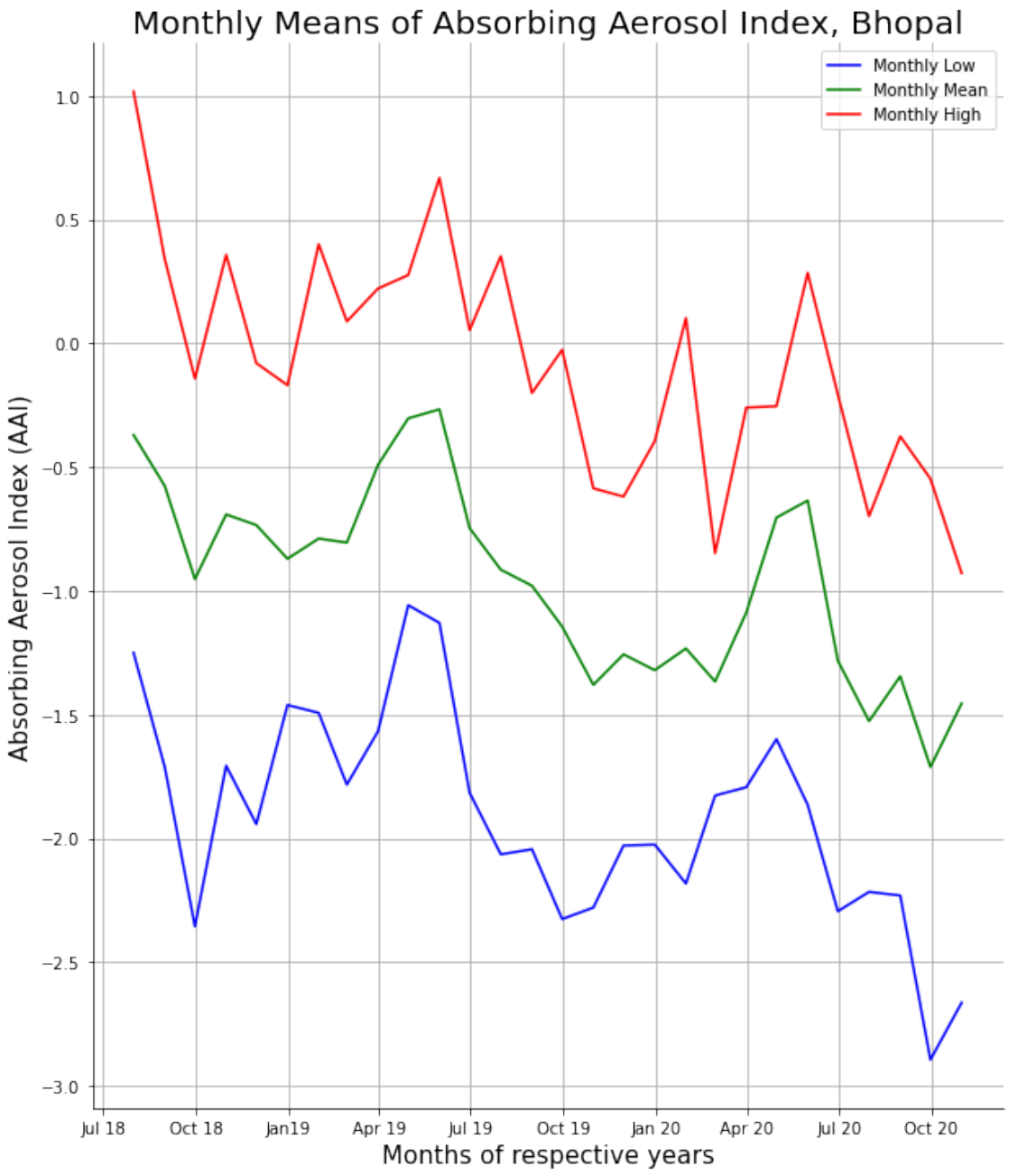
```
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()

# Visualizing by grouping into Monthly data
plt.figure(figsize = (20,8))
plt.plot( Monthly_Bhopal_mean.index.tolist(),Monthly_Bhopal_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Monthly")
plt.gca().axhline(y=Monthly_Bhopal_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Bhopal")
plt.title("Monthly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Bhopal",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()
```





We will now plot some useful insights with Monthly Data

[59]:
```
# Plotting Monthly means
plt.figure(figsize=(10,12))
plt.plot(list(Monthly_Bhopal_min.index),list(Monthly_Bhopal_min["mean"]), label␣
↪= "Monthly Low", color = "blue" )
plt.plot(list(Monthly_Bhopal_mean.index),list(Monthly_Bhopal_mean["mean"]),␣
↪label= "Monthly Mean" , color ="green")
plt.plot(list(Monthly_Bhopal_max.index),list(Monthly_Bhopal_max["mean"]), label␣
↪= "Monthly High", color = "red" )

plt.xlabel("Months of respective years",fontsize= 15)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Monthly Means of Absorbing Aerosol Index, Bhopal",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 15)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels)
plt.legend()
plt.grid()
```

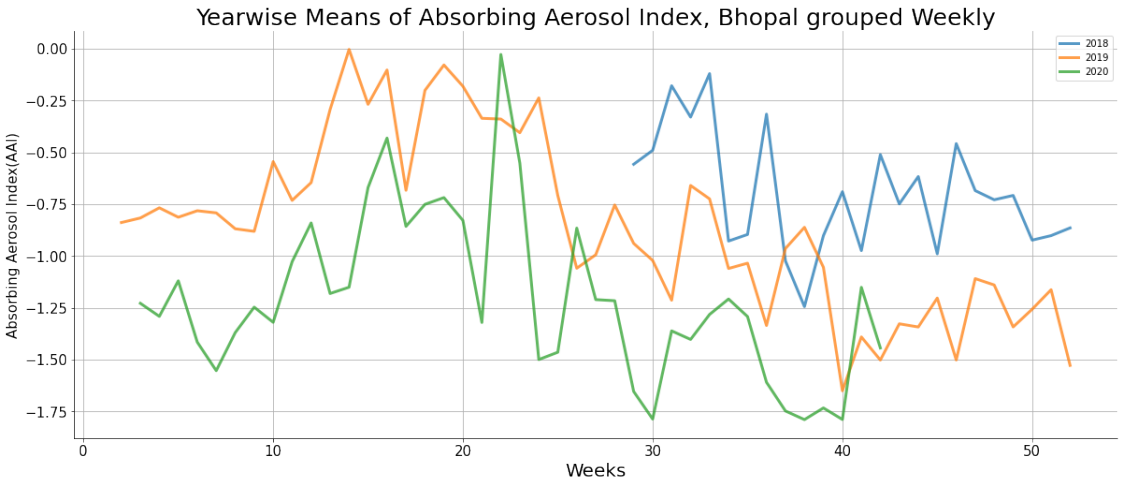


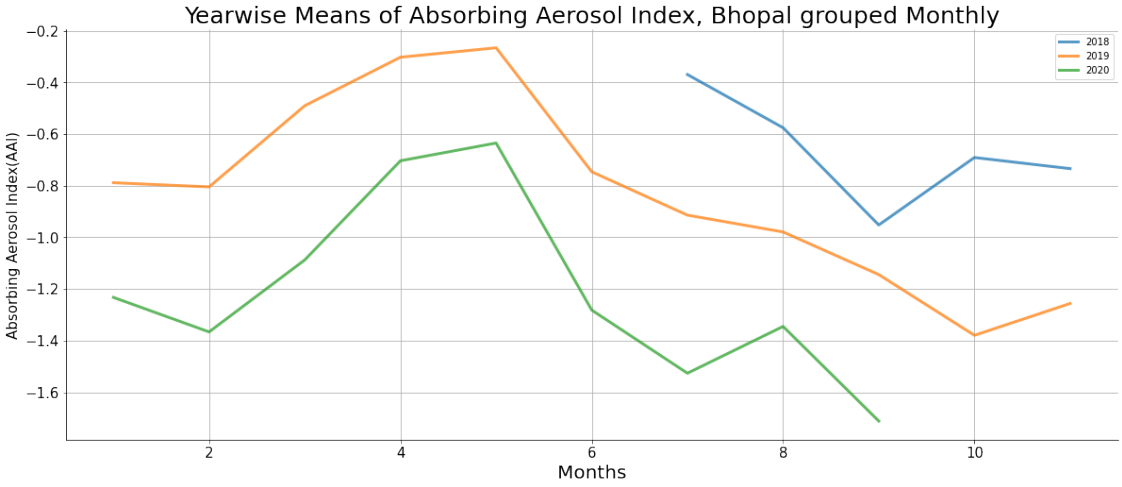
We can observe that Aerosol Optical Depth has decreased in the past year. The change in policies, Even Odd Rule and Coronavirus Pandemic seems to have shifted the curve down.

[60]:
```
# Plotting Yearwise means
x = [i for i in range(1,13)]
plt.figure(figsize = (20,8))
```

```
plt.plot(Weekly_Bhopal_mean[Weekly_Bhopal_mean["year"] == 2018]["week"].
 ↪tolist()[:-1],Weekly_Bhopal_mean[Weekly_Bhopal_mean["year"] == 2018]["mean"].
 ↪tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Weekly_Bhopal_mean[Weekly_Bhopal_mean["year"] == 2019]["week"].
 ↪tolist()[:-1],Weekly_Bhopal_mean[Weekly_Bhopal_mean["year"] == 2019]["mean"].
 ↪tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Weekly_Bhopal_mean[Weekly_Bhopal_mean["year"] == 2020]["week"].
 ↪tolist()[:-1],Weekly_Bhopal_mean[Weekly_Bhopal_mean["year"] == 2020]["mean"].
 ↪tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Weeks",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Yearwise Means of Absorbing Aerosol Index, Bhopal grouped␣
 ↪Weekly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.grid()
plt.legend()

plt.figure(figsize = (20,8))
plt.plot(Monthly_Bhopal_mean[Monthly_Bhopal_mean["year"] == 2018]["month"].
 ↪tolist()[:-1],Monthly_Bhopal_mean[Monthly_Bhopal_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Monthly_Bhopal_mean[Monthly_Bhopal_mean["year"] == 2019]["month"].
 ↪tolist()[:-1],Monthly_Bhopal_mean[Monthly_Bhopal_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Monthly_Bhopal_mean[Monthly_Bhopal_mean["year"] == 2020]["month"].
 ↪tolist()[:-1],Monthly_Bhopal_mean[Monthly_Bhopal_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Months",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.title("Yearwise Means of Absorbing Aerosol Index, Bhopal grouped␣
 ↪Monthly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.grid()
plt.legend()
```

[60]: <matplotlib.legend.Legend at 0x241bbf6abe0>





The observation from the above graph is that, the Absorbing Aerosol Index values have come down considerably in the last two years. From the graph it is evident that the Aerosol Index values are lesser in 2020. Therefore it can be concluded that the polution in Delhi has been lower in 2020 compared to the previous two years.

3.6.2 Plotting Maps For Insights

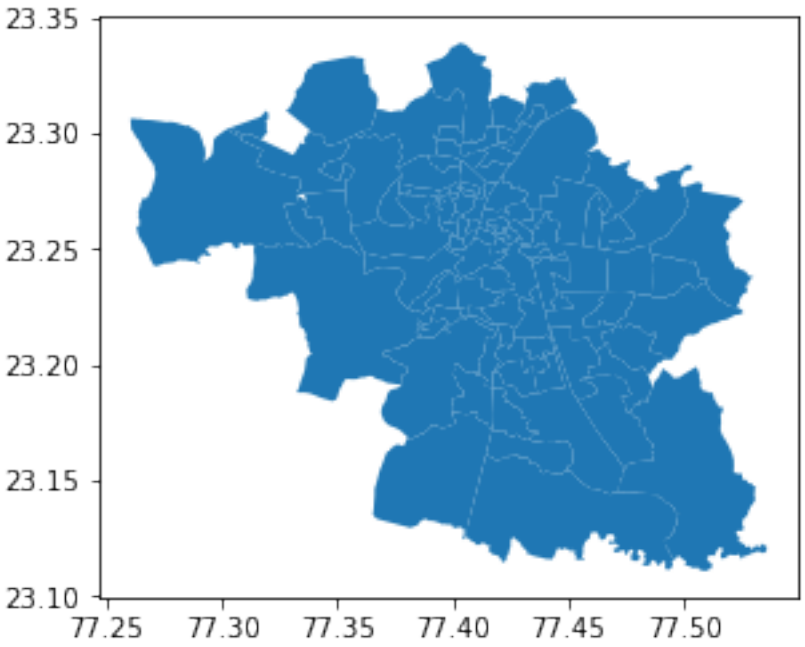
[61]:
```
# Grouping by Wards Data for further analysis
Bhopal_wards = df["Bhopal"].groupby(["Ward_No"]).mean()
Bhopal_wards
```

[61]:
```
              mean
Ward_No
1       -1.038455
```

```
2       -1.028628
3       -1.074727
4       -1.047196
5       -1.024474
…             …
81      -1.031333
82      -0.949440
83      -0.969825
84      -1.029477
85      -1.038108

[85 rows x 1 columns]
```

[62]:
```
# loading the shapefile
path = r'Cleaned CSVs/Bhopal/Bhopal_Wards-polygon.shp'
map_Bhopal = gpd.read_file(path)
map_Bhopal["Ward_Numbe"]= map_Bhopal["Ward_Numbe"].astype(int)
# Viewing the shapefile
map_Bhopal.plot()
```

[62]: <AxesSubplot:>



[63]:
```
# merging the shapefile with the mean data
merged_Bhopal = map_Bhopal.merge(Bhopal_wards, left_on = "Ward_Numbe",␣
 ↪right_index = True)
```

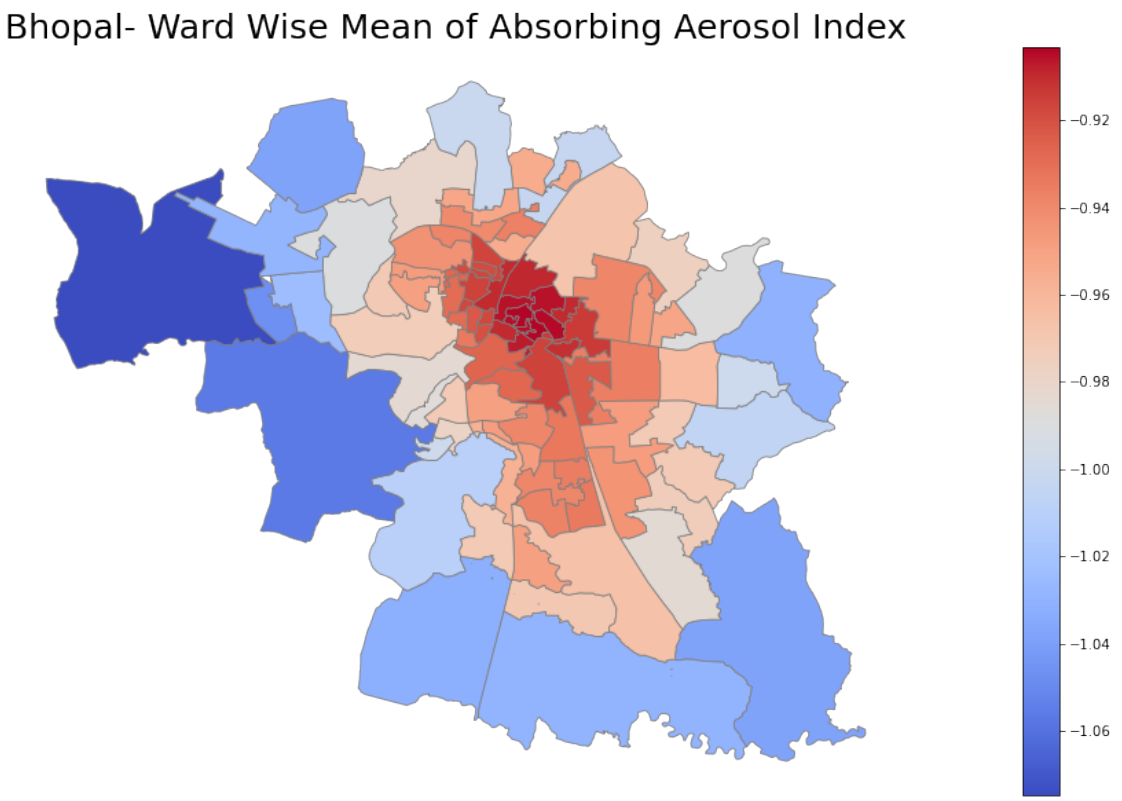
```
# merged_Bhopal.head()
merged_Bhopal
```

[63]:
```
    zone_offic                                              ward_offic  \
0   9424499906                  à¤¶à¥à¤°à¥  à¤ à¤¨à¥  à¤¸à¥à¤¨à¥
1   9424499906        à¤¶à¥à¤°à¥  à¤¬à¤¾à¤®à¤¨  à¤ à¤ à¤ à¤¾à¤²à¥
2   9424499918  à¤¶à¥à¤°à¥  à¤°à¤¾à¤ à¥à¤¶  à¤®à¤ à¥à¤°à¤¿à¤…
3   9424499907  à¤¶à¥à¤°à¥  à¤¬à¥à¤ à¤¤à¥à¤·à¤£  à¤¤à¤¿à¤µà¤…
4   9424499907           à¤¶à¥à¤°à¥  à¤à¤µà¤¾à¤¨à¥  à¤¶à¤ à¤ à¤°
..         …                                                    …
81  9424499915  à¤¶à¥à¤°à¥  à¤°à¤¾à¤ à¥à¤¶  à¤¶à¥à¤°à¥à¤°à¤µà¤…
82  9424499915              à¤¶à¥à¤°à¥  à¤ à¤¹à¤¸à¤¾à¤¨  à¤
à¤²à¥
83  9424499915  à¤¶à¥à¤°à¥  à¤¦à¥à¤µà¤¾à¤°à¥à¤ à¤¾à¤¾  à¤ªà¥à¤…
84  9424499902  à¤¶à¥à¤°à¥  à¤ à¥à¤ à¤¾à¤¾à¤¨à¤¦  à¤®à¤ à¤¤à¤…
85  9424499901              à¤¶à¥à¤°à¥  à¤®à¥à¤£à¤¾à¤²  à¤ à¤°à¥

                                                  zone_off_1  \
0            à¤¶à¥à¤°à¥  à¤¸à¥à¤à¤¾à¤·  à¤¬à¤¾à¤¥à¤®
1            à¤¶à¥à¤°à¥  à¤¸à¥à¤à¤¾à¤·  à¤¬à¤¾à¤¥à¤®
2   à¤¶à¥à¤°à¥  à¤¶à¥à¤²à¥à¤¨à¥à¤¦à¥  à¤¸à¥à¤°  à¤ à¥…
3   à¤¶à¥à¤°à¥  à¤
à¤°à¥à¤ à¥à¤¨  à¤®à¥à¤ à¤¾à¤…
4   à¤¶à¥à¤°à¥  à¤
à¤°à¥à¤ à¥à¤¨  à¤®à¥à¤ à¤¾à¤…
..                                                        …
81  à¤¶à¥à¤°à¥  à¤°à¤¾à¤ à¥à¤¶  à¤¶à¥à¤°à¥à¤°à¤µà¤…
82  à¤¶à¥à¤°à¥  à¤°à¤¾à¤ à¥à¤¶  à¤¶à¥à¤°à¥à¤°à¤µà¤…
83  à¤¶à¥à¤°à¥  à¤°à¤¾à¤ à¥à¤¶  à¤¶à¥à¤°à¥à¤°à¤µà¤…
84  à¤¶à¥à¤°à¥  à¤ à¤®.à¤ªà¥ . à¤¶à¤¾à¤ à¤¦à¤¿à¤²à¥…
85       à¤¶à¥à¤°à¥  à¤®à¤¤à¥à¤¨  à¤ à¥à¤°à¥à¤°à¥à¤¶à¥

                                                corporator zone  Ward_Numbe  \
0   à¤¶à¥à¤°à¥  à¤ªà¥à¤°à¤¦à¥à¤ª  à¤®à¥à¤¨à¥  à…    6          27
1   à¤¶à¥à¤°à¥à¤®à¤¤à¥  à¤²à¤ à¥à¤·à¥à¤®à¥  à¤…    6          28
2     à¤¶à¥à¤°à¥  à¤¸à¤ à¤¤à¥à¤· à¤ à¤ à¤¸à¤¾à¤¨à¤¾   18            29
3   à¤¶à¥à¤°à¥à¤®à¤¤à¥  à¤¸à¥à¤®à¤¾ à¤¸à¥à¤ à¥à¤…    7          30
4           à¤¶à¥à¤°à¥  à¤
à¤®à¤¿à¤¤ à¤¶à¥à¤°à¥à¤®à¤¾    7           31
..                                                      …  …          …
81  à¤¶à¥à¤°à¥  à¤¸à¥à¤°à¥à¤¨à¥à¤¦à¥  à¤¬à¥à¤°  15          64
82  à¤¶à¥à¤°à¥à¤®à¤¤à¥  à¤¦à¤¯à¤¾à¤¨à¤¤à¥  à¤ à¥…   15          63
83  à¤¶à¥à¤°à¥à¤®à¤¤à¥  à¤¡à¤¾à¤²à¥  à¤®à¤¨à¥à¤…   15          62
84       à¤¶à¥à¤°à¥à¤®à¤¤à¥  à¤¨à¥à¤¤à¥  à¤ à¤°à¥     2           6
85  à¤¶à¥à¤°à¥  à¤ à¥à¤·à¥à¤£à¤®à¥à¤¹à¤¨à¥  à¤¸à¥…    1           2

                                                    Name  corporat_1 population  \
```

```
0    à¤ à¥¤à¥ à¤µà¤¾à¤®à¥  à¤¤à¥ à¤²à¤¸à¥ à¤¦à¤¾à¤¸  9406903127       24989
1                 à¤°à¤¾à¤¨à¥  à¤
à¤µà¤ à¤¤à¥ à¤¬à¤¾à¤   9406903128       24155
2    à¤®à¥ à¤²à¤¾à¤¨à¤¾ à¤
à¤¬à¥ à¤²  à¤ à¤²à¤¾à¤® à…  9406903129       25529
3                 à¤ à¥ à¤¶à¤¾à¤à¤¾à¤  à¤ à¤¾à¤ à¤°à¥   9406903130       23007
4             à¤ à¤¤à¥ à¤°à¤ªà¤¤à¤¿ à¤¶à¤¿à¤µà¤¾à¤ à¥   9406903131       21171
..                                                  …         …        …
81                              à¤¸à¥ à¤¨à¤¾à¤ à¤¿à¤°à¥   9406903164       25964
82                           à¤ à¥ à¤¤à¤®à¤¬à¥ à¤¦à¥ à¤§  9406903163       25180
83                           à¤¹à¤¥à¤¾à¤ à¤ à¥ à¤¡à¤½à¤¾  9406903162       19669
84                        à¤®à¤¹à¤¾à¤µà¥ à¤°à¤ à¤¿à¤°à¥   9406903106       21267
85                              à¤ à¤¯à¤°à¤ªà¥ à¤°à¥ à¤   9406903102       23173

    ward_off_1                                              geometry      mean
0   9425601327  POLYGON Z ((77.39051 23.21976 0.00000, 77.3923… -0.998266
1   9425601328  POLYGON Z ((77.40383 23.22100 0.00000, 77.4026… -0.978653
2   9425601329  POLYGON Z ((77.38854 23.21192 0.00000, 77.3899… -1.009352
3   9425601330  POLYGON Z ((77.40744 23.22113 0.00000, 77.4062… -0.956975
4   9425601331  POLYGON Z ((77.40185 23.22866 0.00000, 77.4016… -0.949502
..         …                                                 …       …
81  9425601364  POLYGON Z ((77.46710 23.24980 0.00000, 77.4674… -0.988863
82  9425601363  POLYGON Z ((77.46610 23.24995 0.00000, 77.4671… -0.962341
83  9425601362  POLYGON Z ((77.49969 23.27661 0.00000, 77.4998… -1.029873
84  9425601306  POLYGON Z ((77.35384 23.27022 0.00000, 77.3536… -0.990073
85  9425601302  POLYGON Z ((77.33804 23.30147 0.00000, 77.3410… -1.028628

[86 rows x 12 columns]
```

[64]:
```
# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values
vmin, vmax = merged_Bhopal["mean"].min(), merged_Bhopal["mean"].max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))
# remove the axis
ax.axis('off')
# add a title and annotation
ax.set_title('Bhopal- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))
# empty array for the data range
sm.set_array([])
# add the colorbar to the figure
```

```
fig.colorbar(sm)
# create map
merged_Bhopal.plot(column=variable, cmap='coolwarm', linewidth=0.8, ax=ax,␣
 ↪edgecolor='0.5', figsize = (5,15))
plt.show()
```



[65]:
```
# Using T-Test to determine if climate has changed in the first 14 and last 14␣
 ↪months in Bhopal
ttest_ind(Monthly_Bhopal_mean.iloc[:14,:]["mean"].tolist(),Monthly_Bhopal_mean.
 ↪iloc[14:,:]["mean"].tolist())
```

[65]:
```
Ttest_indResult(statistic=5.659151468144071, pvalue=5.957401679702306e-06)
```

[ ]:
```

```

3.7 City Number 7: Bhubaneshwar

3.7.1 Loading and Visualizing

[66]:
```
# g = df1.groupby(pd.Grouper(freq="M"))
# g.mean()
Bhubaneshwar = df["Bhubaneshwar"].groupby(["date"]).mean()
Bhubaneshwar["month"] = Bhubaneshwar.index.month
Bhubaneshwar["year"] = Bhubaneshwar.index.year
Bhubaneshwar
```

[66]:

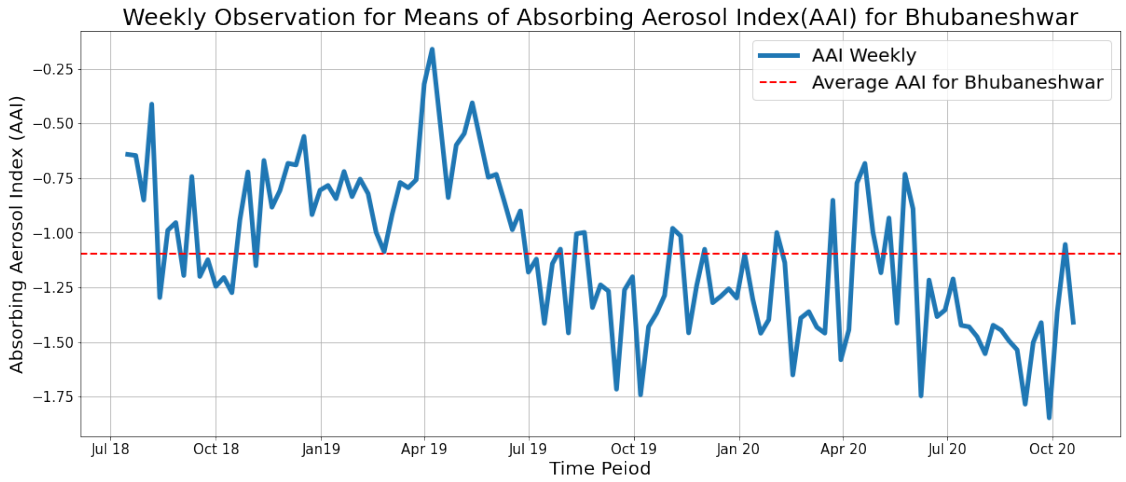
date	mean	Ward_No	month	year
2018-07-11	-1.124388	34	7	2018
2018-07-12	-1.593273	34	7	2018
2018-07-13	-0.079946	34	7	2018
2018-07-14	0.077493	34	7	2018
2018-07-15	-0.489921	34	7	2018
…	…	…	…	…
2020-10-09	-2.188640	34	10	2020
2020-10-10	-1.801918	34	10	2020
2020-10-11	-0.393510	34	10	2020
2020-10-12	-1.041583	34	10	2020
2020-10-13	-1.410126	34	10	2020

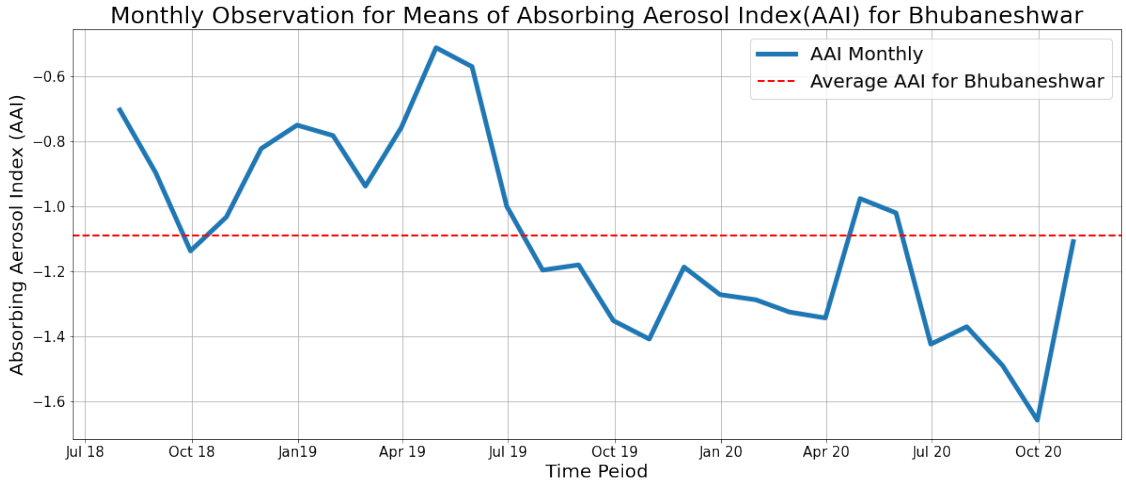
[792 rows x 4 columns]

[67]:
```
# Aggregate statistics grouped by Weeks
Weekly_Bhubaneshwar_mean = Bhubaneshwar.groupby(pd.Grouper(freq="W-MON")).mean()
Weekly_Bhubaneshwar_max = Bhubaneshwar.groupby(pd.Grouper(freq="W-MON")).max()
Weekly_Bhubaneshwar_min = Bhubaneshwar.groupby(pd.Grouper(freq="W-MON")).min()

# Aggregate statistics grouped by Months
Monthly_Bhubaneshwar_mean = Bhubaneshwar.groupby(pd.Grouper(freq="M")).mean()
Monthly_Bhubaneshwar_max = Bhubaneshwar.groupby(pd.Grouper(freq="M")).max()
Monthly_Bhubaneshwar_min = Bhubaneshwar.groupby(pd.Grouper(freq="M")).min()

#Week Identifier
Weekly_Bhubaneshwar_mean["week"] = Weekly_Bhubaneshwar_mean.index.isocalendar().
 ↪week
Weekly_Bhubaneshwar_min["week"] = Weekly_Bhubaneshwar_min.index.isocalendar().
 ↪week
Weekly_Bhubaneshwar_max["week"] = Weekly_Bhubaneshwar_max.index.isocalendar().
 ↪week

Monthly_Bhubaneshwar_mean["week"] = Monthly_Bhubaneshwar_mean.index.
 ↪isocalendar().week
```

```
Monthly_Bhubaneshwar_min["week"] = Monthly_Bhubaneshwar_min.index.isocalendar().
 ↪week
Monthly_Bhubaneshwar_max["week"] = Monthly_Bhubaneshwar_max.index.isocalendar().
 ↪week
```

[68]:
```
# Visualizing by grouping into weeekly data
plt.figure(figsize = (20,8))
plt.plot( Weekly_Bhubaneshwar_mean.index.
 ↪tolist(),Weekly_Bhubaneshwar_mean["mean"], linewidth = 5, label = "AAI␣
 ↪Weekly")
plt.gca().axhline(y=Weekly_Bhubaneshwar_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Bhubaneshwar")
plt.title("Weekly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Bhubaneshwar",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()

# Visualizing by grouping into Monthly data
plt.figure(figsize = (20,8))
plt.plot( Monthly_Bhubaneshwar_mean.index.
 ↪tolist(),Monthly_Bhubaneshwar_mean["mean"], linewidth = 5, label = "AAI␣
 ↪Monthly")
plt.gca().axhline(y=Monthly_Bhubaneshwar_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Bhubaneshwar")
plt.title("Monthly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Bhubaneshwar",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()
```

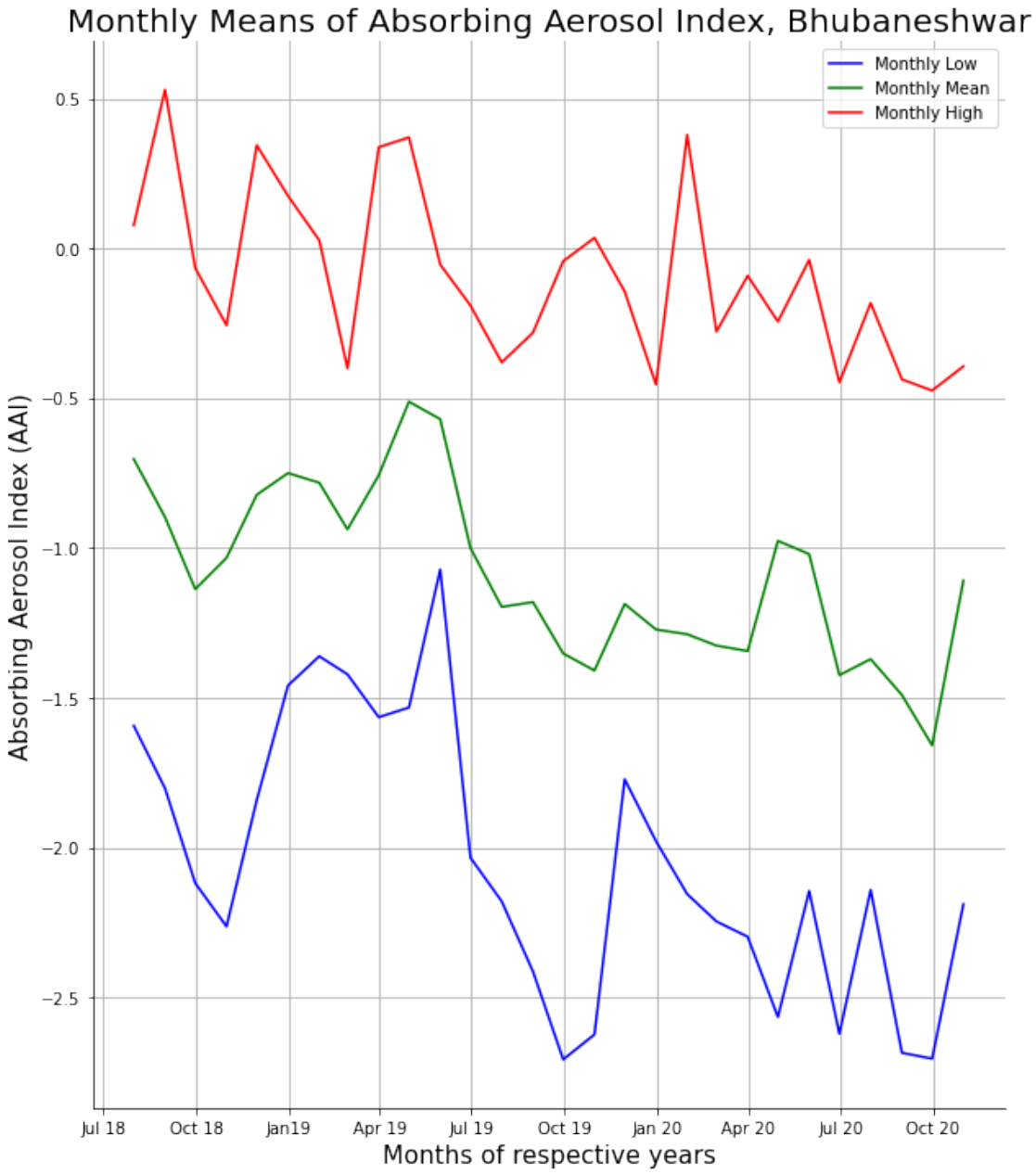




We will now plot some useful insights with Monthly Data

[69]:
```
# Plotting Monthly means
plt.figure(figsize=(10,12))
plt.plot(list(Monthly_Bhubaneshwar_min.
 ↪index),list(Monthly_Bhubaneshwar_min["mean"]), label = "Monthly Low", color␣
 ↪= "blue" )
plt.plot(list(Monthly_Bhubaneshwar_mean.
 ↪index),list(Monthly_Bhubaneshwar_mean["mean"]), label= "Monthly Mean" ,␣
 ↪color ="green")
plt.plot(list(Monthly_Bhubaneshwar_max.
 ↪index),list(Monthly_Bhubaneshwar_max["mean"]), label = "Monthly High", color␣
 ↪= "red" )
```

```
plt.xlabel("Months of respective years",fontsize= 15)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Monthly Means of Absorbing Aerosol Index, Bhubaneshwar",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 15)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels)
plt.legend()
plt.grid()
```

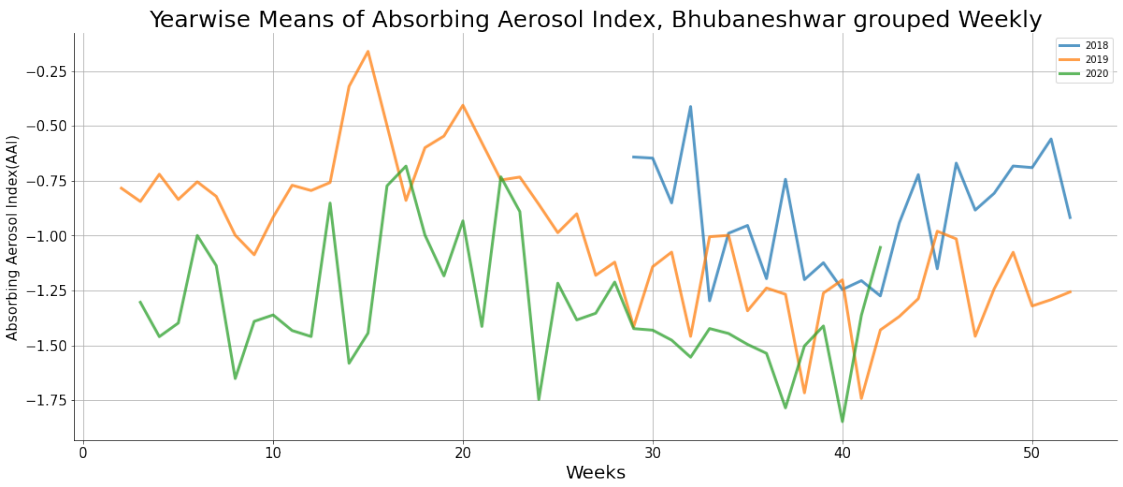


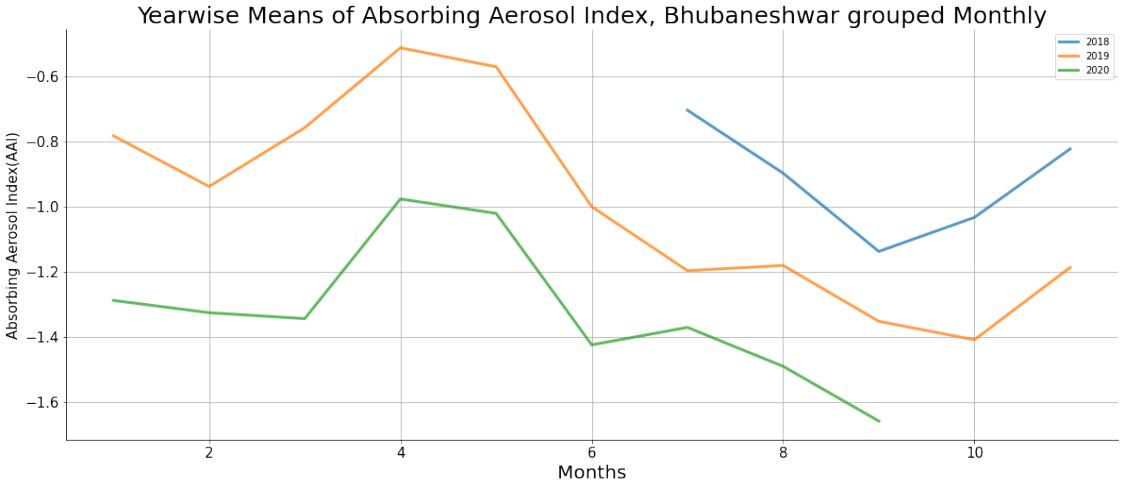
We can observe that Aerosol Optical Depth has decreased in the past year. The change in policies, Even Odd Rule and Coronavirus Pandemic seems to have shifted the curve down.

[70]:
```
# Plotting Yearwise means
x = [i for i in range(1,13)]
plt.figure(figsize = (20,8))
plt.plot(Weekly_Bhubaneshwar_mean[Weekly_Bhubaneshwar_mean["year"] ==␣
 ↪2018]["week"].tolist()[:
 ↪-1],Weekly_Bhubaneshwar_mean[Weekly_Bhubaneshwar_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Weekly_Bhubaneshwar_mean[Weekly_Bhubaneshwar_mean["year"] ==␣
 ↪2019]["week"].tolist()[:
 ↪-1],Weekly_Bhubaneshwar_mean[Weekly_Bhubaneshwar_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Weekly_Bhubaneshwar_mean[Weekly_Bhubaneshwar_mean["year"] ==␣
 ↪2020]["week"].tolist()[:
 ↪-1],Weekly_Bhubaneshwar_mean[Weekly_Bhubaneshwar_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Weeks",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Yearwise Means of Absorbing Aerosol Index, Bhubaneshwar grouped␣
 ↪Weekly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.grid()
plt.legend()

plt.figure(figsize = (20,8))
plt.plot(Monthly_Bhubaneshwar_mean[Monthly_Bhubaneshwar_mean["year"] ==␣
 ↪2018]["month"].tolist()[:
 ↪-1],Monthly_Bhubaneshwar_mean[Monthly_Bhubaneshwar_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Monthly_Bhubaneshwar_mean[Monthly_Bhubaneshwar_mean["year"] ==␣
 ↪2019]["month"].tolist()[:
 ↪-1],Monthly_Bhubaneshwar_mean[Monthly_Bhubaneshwar_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Monthly_Bhubaneshwar_mean[Monthly_Bhubaneshwar_mean["year"] ==␣
 ↪2020]["month"].tolist()[:
 ↪-1],Monthly_Bhubaneshwar_mean[Monthly_Bhubaneshwar_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)
```

```
plt.xlabel("Months",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.title("Yearwise Means of Absorbing Aerosol Index, Bhubaneshwar grouped␣
 ↪Monthly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.grid()
plt.legend()
```

[70]: <matplotlib.legend.Legend at 0x241bdfdda60>





The observation from the above graph is that, the Absorbing Aerosol Index values have come down

considerably in the last two years. From the graph it is evident that the Aerosol Index values are lesser in 2020. Therefore it can be concluded that the polution in Delhi has been lower in 2020 compared to the previous two years.

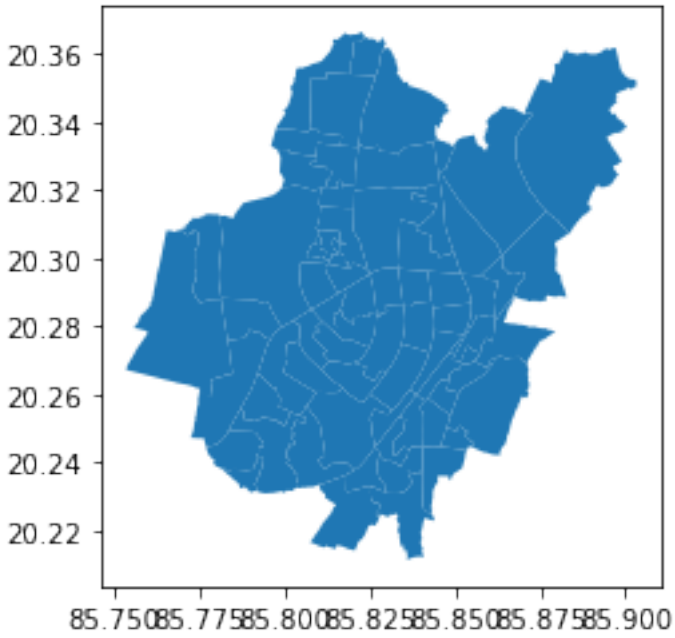
3.7.2 Plotting Maps For Insights

[71]:
```
# Grouping by Wards Data for further analysis
Bhubaneshwar_wards = df["Bhubaneshwar"].groupby(["Ward_No"]).mean()
Bhubaneshwar_wards
```

[71]:
```
              mean
Ward_No
1       -1.097257
2       -1.089468
3       -1.076829
4       -1.087910
5       -1.124838
…             …
63      -1.095557
64      -1.092055
65      -1.106248
66      -1.084328
67      -1.080156

[67 rows x 1 columns]
```

[72]:
```
# loading the shapefile
path = r'Cleaned CSVs/Bhubaneshwar/Bhubaneshwar.shp'
map_Bhubaneshwar = gpd.read_file(path)
# Viewing the shapefile
map_Bhubaneshwar.plot()
```

[72]:
```
<AxesSubplot:>
```



[73]:
```
# merging the shapefile with the mean data
merged_Bhubaneshwar = map_Bhubaneshwar.merge(Bhubaneshwar_wards, left_on =␣
 ↪"objectid", right_index = True)
# merged_Bhubaneshwar.head()
merged_Bhubaneshwar
```

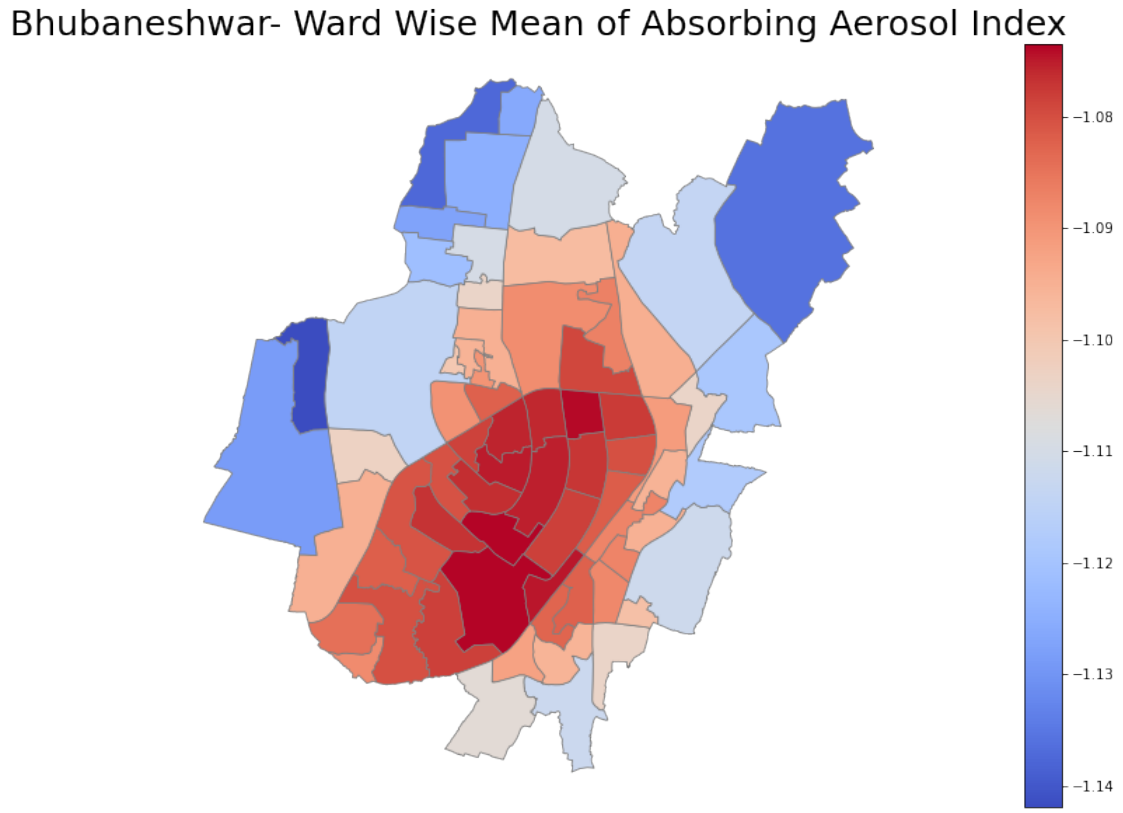
[73]:

	objectid	name	folderpath	symbolid	altmode	base	clamped	\
0	1	Placemark	Document/Wards	0	0	0	-1	
1	2	Placemark	Document/Wards	0	0	0	-1	
2	3	Placemark	Document/Wards	0	0	0	-1	
3	4	Placemark	Document/Wards	0	0	0	-1	
4	5	Placemark	Document/Wards	0	0	0	-1	
..	…	…	…	…	…	…	…	
62	63	Placemark	Document/Wards	0	0	0	-1	
63	64	Placemark	Document/Wards	0	0	0	-1	
64	65	Placemark	Document/Wards	0	0	0	-1	
65	66	Placemark	Document/Wards	0	0	0	-1	
66	67	Placemark	Document/Wards	0	0	0	-1	

	extruded	snippet	popupinfo	\
0	0	None	None	
1	0	None	None	
2	0	None	None	
3	0	None	None	
4	0	None	None	

```
..      …     …       …
62        0    None      None
63        0    None      None
64        0    None      None
65        0    None      None
66        0    None      None

                                                geometry      mean
0   POLYGON ((85.82041 20.32118, 85.82042 20.32141… -1.097257
1   POLYGON ((85.80412 20.29752, 85.80412 20.29796… -1.089468
2   POLYGON ((85.80375 20.27573, 85.80381 20.27569… -1.076829
3   POLYGON ((85.84112 20.25814, 85.84117 20.25819… -1.087910
4   POLYGON ((85.81945 20.35411, 85.81946 20.35403… -1.124838
..                                                    …       …
62  POLYGON ((85.82770 20.24244, 85.82772 20.24236… -1.095557
63  POLYGON ((85.81808 20.23642, 85.81821 20.23649… -1.092055
64  POLYGON ((85.81808 20.23642, 85.81800 20.23520… -1.106248
65  POLYGON ((85.77677 20.24464, 85.77683 20.24464… -1.084328
66  POLYGON ((85.79200 20.24172, 85.79201 20.24179… -1.080156

[67 rows x 12 columns]
```

[74]:
```
# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values
vmin, vmax = merged_Bhubaneshwar["mean"].min(), merged_Bhubaneshwar["mean"].
 ↪max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))
# remove the axis
ax.axis('off')
# add a title and annotation
ax.set_title('Bhubaneshwar- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))
# empty array for the data range
sm.set_array([])
# add the colorbar to the figure
fig.colorbar(sm)
# create map
merged_Bhubaneshwar.plot(column=variable, cmap='coolwarm', linewidth=0.8,␣
 ↪ax=ax, edgecolor='0.5', figsize = (5,15))
plt.show()
```



[75]:
```
# Using T-Test to determine if climate has changed in the first 14 and last 14 months in Bhubaneshwar
ttest_ind(Monthly_Bhubaneshwar_mean.iloc[:14,:]["mean"].tolist(),Monthly_Bhubaneshwar_mean.iloc[14:,:]["mean"].tolist())
```

[75]:
```
Ttest_indResult(statistic=5.615521933063699, pvalue=6.675152930440566e-06)
```

We observe that Pollution levels has decreased in the last 14 months and the statistical test proves the same with a very small p-value for 95 % Confidence Interval

[ ]:
```

```

3.8 City Number 8: Chennai

3.8.1 Loading and Visualizing

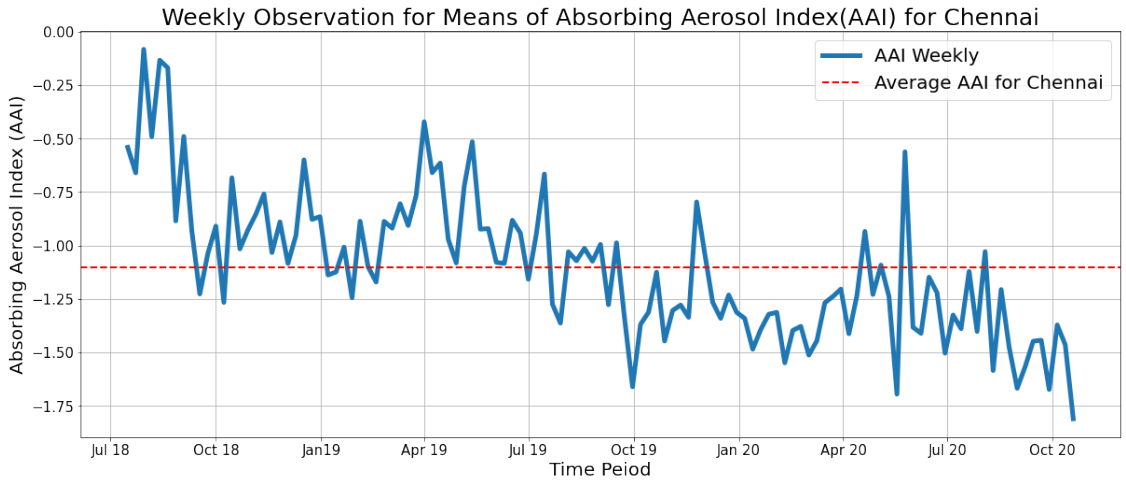
[76]:
```
# g = df1.groupby(pd.Grouper(freq="M"))
# g.mean()
Chennai = df["Chennai"].groupby(["date"]).mean()
Chennai["month"] = Chennai.index.month
Chennai["year"] = Chennai.index.year
Chennai
```

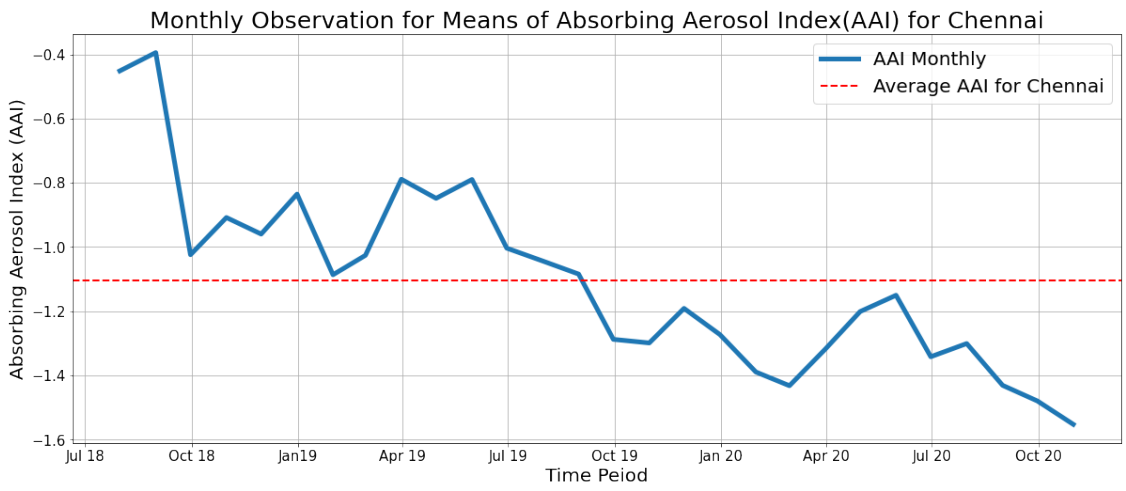
[76]:
```
            Ward_No      mean  month  year
date
2018-07-11    100.0 -0.119198      7  2018
2018-07-12    100.0 -0.380352      7  2018
2018-07-13    100.0 -0.760456      7  2018
2018-07-14    100.0 -0.772295      7  2018
2018-07-15    100.0 -1.264108      7  2018
…               …        …     …    …
2020-10-09    100.0 -1.030521     10  2020
2020-10-10    100.0 -0.795878     10  2020
2020-10-11    100.0 -1.247523     10  2020
2020-10-12    100.0 -2.314720     10  2020
2020-10-13    100.0 -1.810164     10  2020

[802 rows x 4 columns]
```

[77]:
```
# Aggregate statistics grouped by Weeks
Weekly_Chennai_mean = Chennai.groupby(pd.Grouper(freq="W-MON")).mean()
Weekly_Chennai_max = Chennai.groupby(pd.Grouper(freq="W-MON")).max()
Weekly_Chennai_min = Chennai.groupby(pd.Grouper(freq="W-MON")).min()

# Aggregate statistics grouped by Months
Monthly_Chennai_mean = Chennai.groupby(pd.Grouper(freq="M")).mean()
Monthly_Chennai_max = Chennai.groupby(pd.Grouper(freq="M")).max()
Monthly_Chennai_min = Chennai.groupby(pd.Grouper(freq="M")).min()

#Week Identifier
Weekly_Chennai_mean["week"] = Weekly_Chennai_mean.index.isocalendar().week
Weekly_Chennai_min["week"] = Weekly_Chennai_min.index.isocalendar().week
Weekly_Chennai_max["week"] = Weekly_Chennai_max.index.isocalendar().week

Monthly_Chennai_mean["week"] = Monthly_Chennai_mean.index.isocalendar().week
Monthly_Chennai_min["week"] = Monthly_Chennai_min.index.isocalendar().week
Monthly_Chennai_max["week"] = Monthly_Chennai_max.index.isocalendar().week
```

[78]:
```
# Visualizing by grouping into weeekly data
plt.figure(figsize = (20,8))
plt.plot( Weekly_Chennai_mean.index.tolist(),Weekly_Chennai_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Weekly")
plt.gca().axhline(y=Weekly_Chennai_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Chennai")
plt.title("Weekly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Chennai",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
```

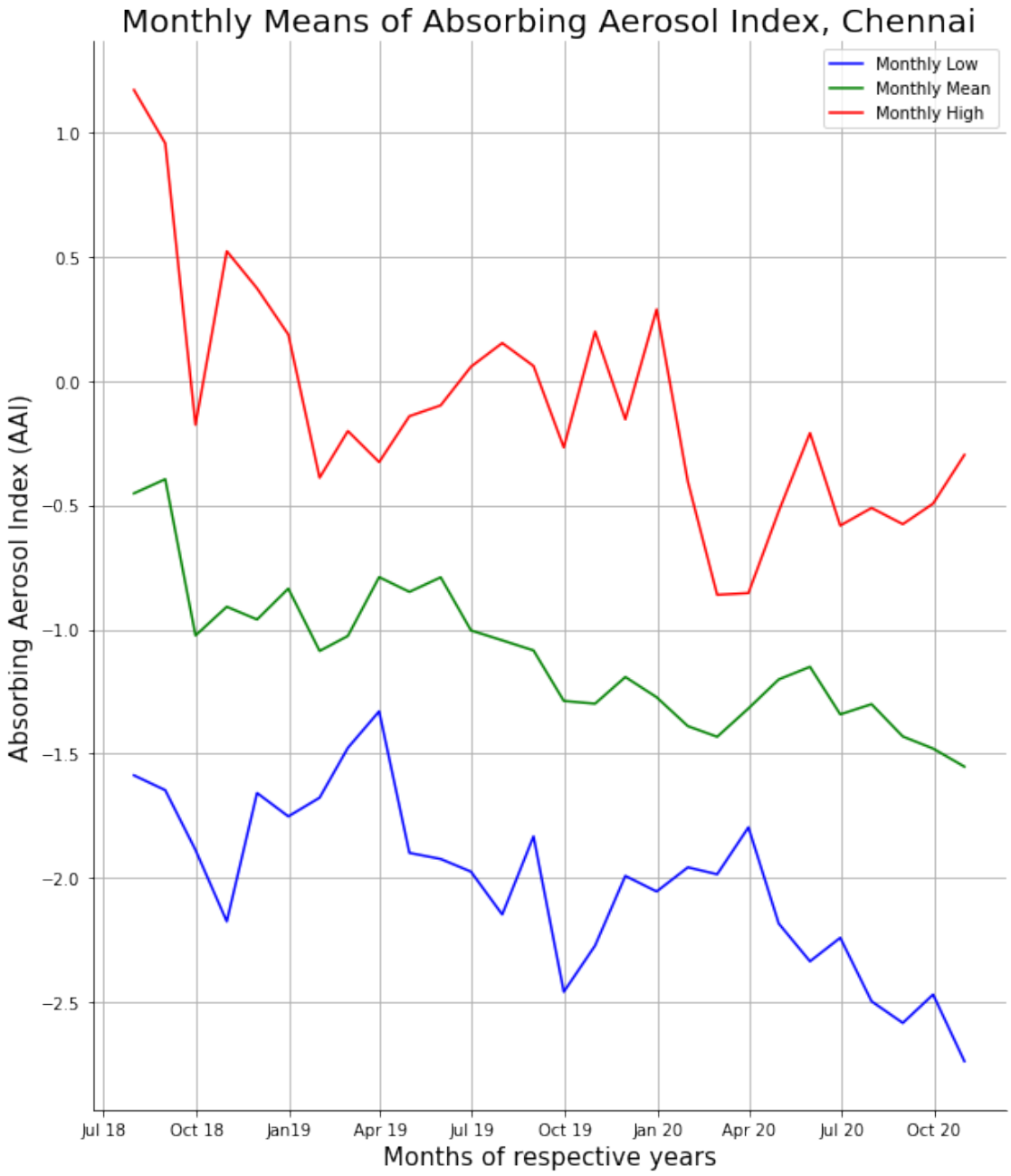
```
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()

# Visualizing by grouping into Monthly data
plt.figure(figsize = (20,8))
plt.plot( Monthly_Chennai_mean.index.tolist(),Monthly_Chennai_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Monthly")
plt.gca().axhline(y=Monthly_Chennai_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Chennai")
plt.title("Monthly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Chennai",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()
```





We will now plot some useful insights with Monthly Data

[79]:
```
# Plotting Monthly means
plt.figure(figsize=(10,12))
plt.plot(list(Monthly_Chennai_min.index),list(Monthly_Chennai_min["mean"]),␣
↪label = "Monthly Low", color = "blue" )
plt.plot(list(Monthly_Chennai_mean.index),list(Monthly_Chennai_mean["mean"]),␣
↪label= "Monthly Mean" , color ="green")
plt.plot(list(Monthly_Chennai_max.index),list(Monthly_Chennai_max["mean"]),␣
↪label = "Monthly High", color = "red" )

plt.xlabel("Months of respective years",fontsize= 15)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Monthly Means of Absorbing Aerosol Index, Chennai",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 15)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels)
plt.legend()
plt.grid()
```

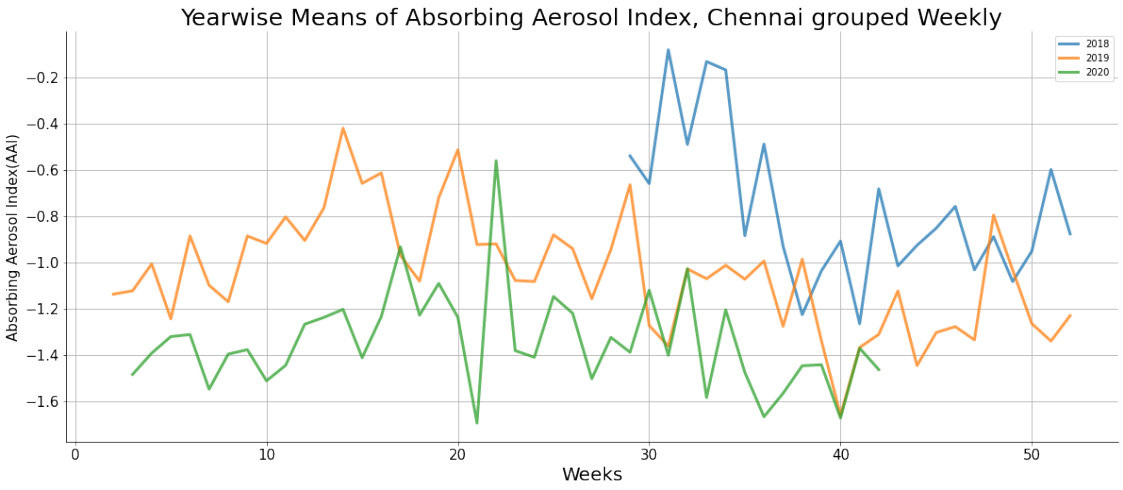


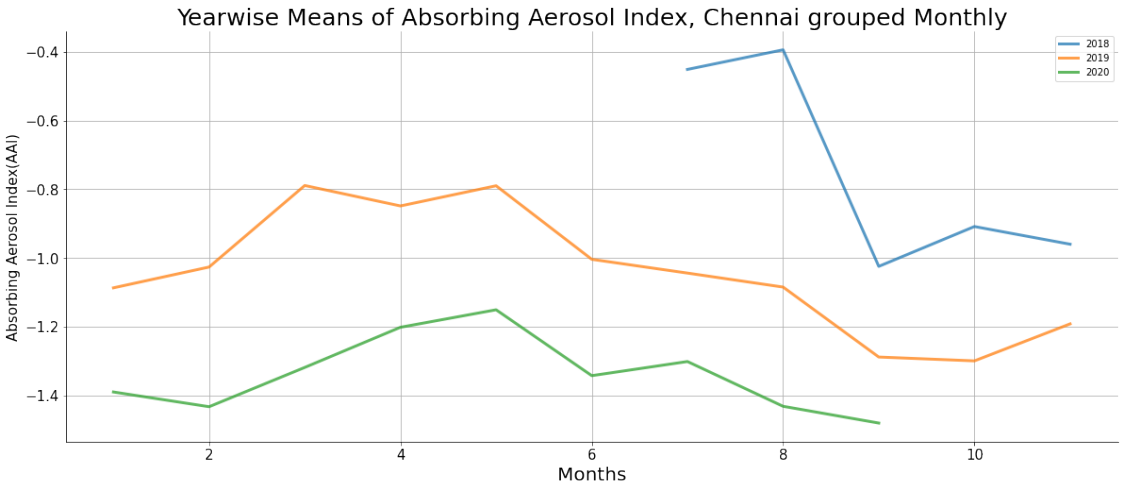
We can observe that Aerosol Optical Depth has decreased in the past year. The change in policies, Even Odd Rule and Coronavirus Pandemic seems to have shifted the curve down.

[80]:
```
# Plotting Yearwise means
x = [i for i in range(1,13)]
plt.figure(figsize = (20,8))
```

```
plt.plot(Weekly_Chennai_mean[Weekly_Chennai_mean["year"] == 2018]["week"].
 ↪tolist()[:-1],Weekly_Chennai_mean[Weekly_Chennai_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Weekly_Chennai_mean[Weekly_Chennai_mean["year"] == 2019]["week"].
 ↪tolist()[:-1],Weekly_Chennai_mean[Weekly_Chennai_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Weekly_Chennai_mean[Weekly_Chennai_mean["year"] == 2020]["week"].
 ↪tolist()[:-1],Weekly_Chennai_mean[Weekly_Chennai_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Weeks",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Yearwise Means of Absorbing Aerosol Index, Chennai grouped␣
 ↪Weekly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.grid()
plt.legend()

plt.figure(figsize = (20,8))
plt.plot(Monthly_Chennai_mean[Monthly_Chennai_mean["year"] == 2018]["month"].
 ↪tolist()[:-1],Monthly_Chennai_mean[Monthly_Chennai_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Monthly_Chennai_mean[Monthly_Chennai_mean["year"] == 2019]["month"].
 ↪tolist()[:-1],Monthly_Chennai_mean[Monthly_Chennai_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Monthly_Chennai_mean[Monthly_Chennai_mean["year"] == 2020]["month"].
 ↪tolist()[:-1],Monthly_Chennai_mean[Monthly_Chennai_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Months",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.title("Yearwise Means of Absorbing Aerosol Index, Chennai grouped␣
 ↪Monthly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.grid()
plt.legend()
```

[80]: <matplotlib.legend.Legend at 0x241bb34b7f0>





The observation from the above graph is that, the Absorbing Aerosol Index values have come down considerably in the last two years. From the graph it is evident that the Aerosol Index values are lesser in 2020. Therefore it can be concluded that the polution in Delhi has been lower in 2020 compared to the previous two years.

3.8.2 Plotting Maps For Insights

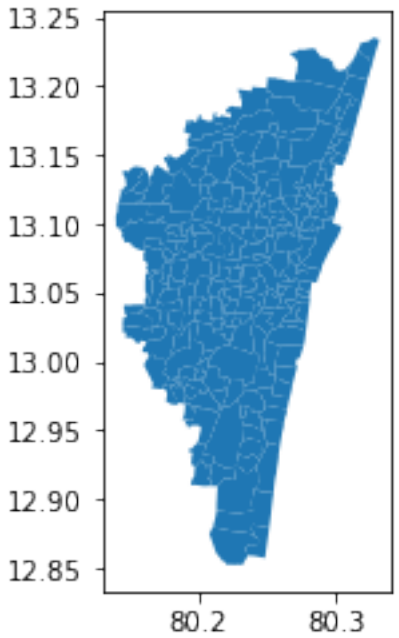
[81]:
```
# Grouping by Wards Data for further analysis
Chennai_wards = df["Chennai"].groupby(["Ward_No"]).mean()
Chennai_wards
```

[81]:
```
              mean
Ward_No
0       -1.036939
```

```
1       -1.251051
2       -1.255380
3       -1.261816
4       -1.245673
…             …
196     -1.239086
197     -1.201059
198     -1.243007
199     -1.278456
200     -1.238262

[201 rows x 1 columns]
```

[82]:
```
# loading the shapefile
path = r'Cleaned CSVs/Chennai/Wards-polygon.shp'
map_Chennai = gpd.read_file(path)
# Viewing the shapefile
map_Chennai.plot()
```

[82]: <AxesSubplot:>



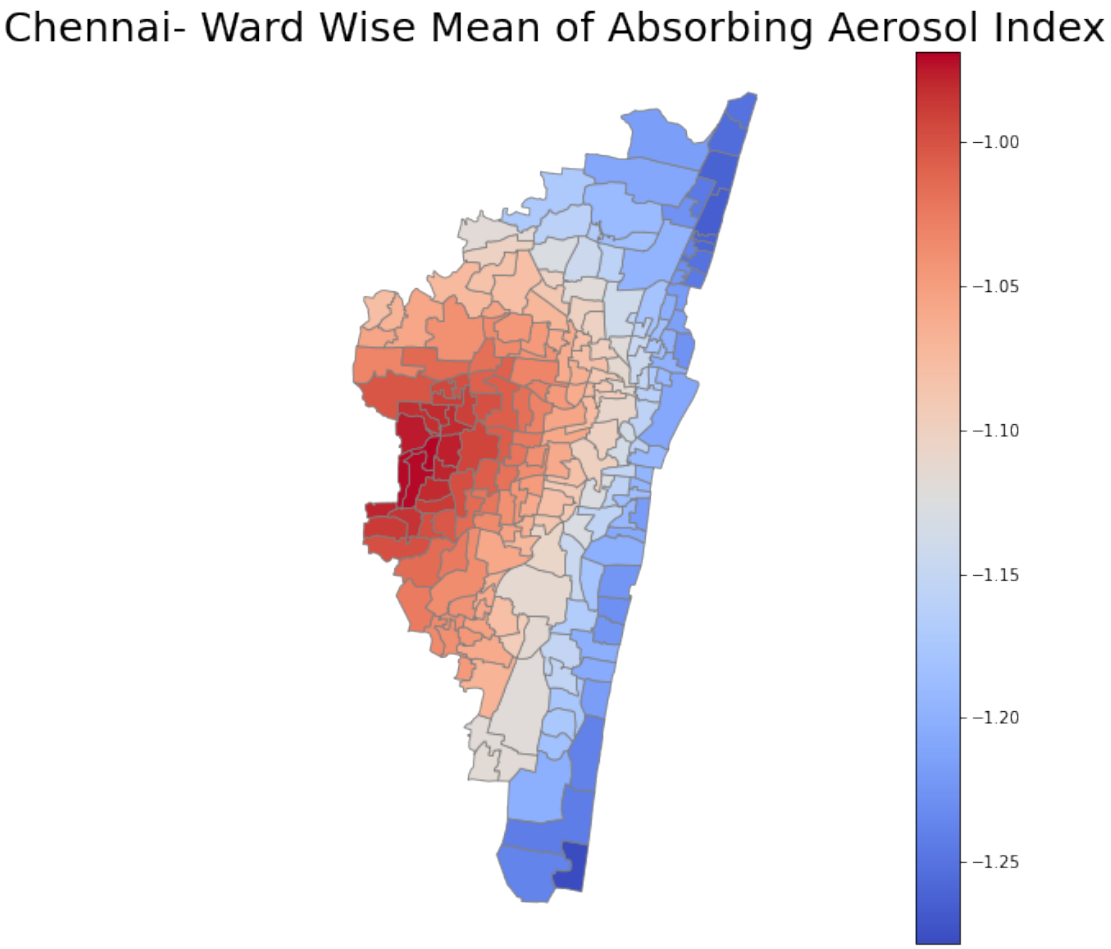
[83]:
```
# merging the shapefile with the mean data
merged_Chennai = map_Chennai.merge(Chennai_wards, left_on = "Ward_No",␣
 ↪right_index = True)
# merged_Chennai.head()
```

```
merged_Chennai
```

[83]:
```
    Zone_No  Ward_No      Zone_Name      AREA  PERIMETER  \
0        IX      119      TEYNAMPET  0.000115   0.061956
1         I        7  THIRUVOTTIYUR  0.000645   0.124303
2       III       26     MADHAVARAM  0.000401   0.096859
3      XIII      174          ADYAR  0.000695   0.121556
4         X      132    KODAMBAKKAM  0.000117   0.051055
..      …      …            …        …         …
196       X      134    KODAMBAKKAM  0.000087   0.042240
197       X      135    KODAMBAKKAM  0.000078   0.039567
198       X      133    KODAMBAKKAM  0.000085   0.063298
199       X      131    KODAMBAKKAM  0.000098   0.063492
200       X      130    KODAMBAKKAM  0.000098   0.047812

                                                geometry      mean
0    POLYGON ((80.27090 13.04753, 80.27062 13.04759… -1.156878
1    POLYGON ((80.29467 13.15558, 80.29291 13.15137… -1.196864
2    POLYGON ((80.22978 13.14719, 80.22842 13.14495… -1.079512
3    POLYGON ((80.24043 13.00684, 80.24130 13.00659… -1.113540
4    POLYGON ((80.20803 13.03565, 80.20835 13.03708… -1.039459
..                                                 …       …
196  POLYGON ((80.22926 13.04390, 80.22259 13.04572… -1.047412
197  POLYGON ((80.22085 13.03439, 80.21885 13.03459… -1.060044
198  POLYGON ((80.21711 13.03137, 80.21664 13.03131… -1.044094
199  POLYGON ((80.20950 13.04646, 80.20940 13.04483… -1.020649
200  POLYGON ((80.20954 13.04698, 80.20845 13.04700… -1.019344

[201 rows x 7 columns]
```

[84]:
```
# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values
vmin, vmax = merged_Chennai["mean"].min(), merged_Chennai["mean"].max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))
# remove the axis
ax.axis('off')
# add a title and annotation
ax.set_title('Chennai- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))
# empty array for the data range
sm.set_array([])
```

```
# add the colorbar to the figure
fig.colorbar(sm)
# create map
merged_Chennai.plot(column=variable, cmap='coolwarm', linewidth=0.8, ax=ax,␣
 ↪edgecolor='0.5', figsize = (5,15))
plt.show()
```



[85]:
```
# Using T-Test to determine if climate has changed in the first 14 and last 14␣
 ↪months in Chennai
ttest_ind(Monthly_Chennai_mean.iloc[:14,:]["mean"].
 ↪tolist(),Monthly_Chennai_mean.iloc[14:,:]["mean"].tolist())
```

[85]:
```
Ttest_indResult(statistic=6.962436683330846, pvalue=2.157851807333305e-07)
```

We observe that Pollution levels has decreased in the last 14 months and the statistical test proves the same with a very small p-value for 95 % Confidence Interval

[ ]:

3.9 City Number 9: Hyderabad

3.9.1 Loading and Visualizing

[86]:
```
# g = df1.groupby(pd.Grouper(freq="M"))
# g.mean()
Hyderabad = df["Hyderabad"].groupby(["date"]).mean()
Hyderabad["month"] = Hyderabad.index.month
Hyderabad["year"] = Hyderabad.index.year
Hyderabad
```

[86]:

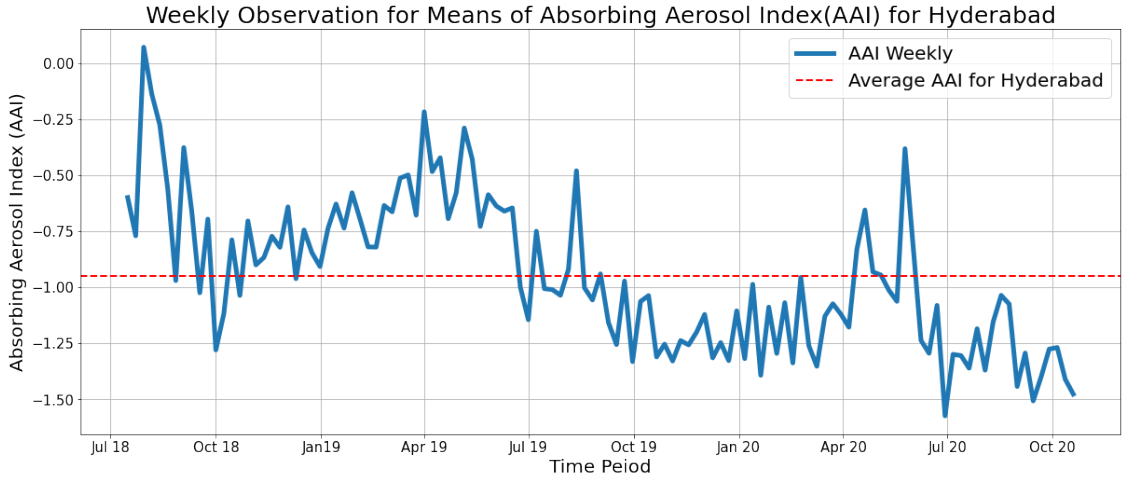
date	mean	Ward_No	month	year
2018-07-11	-0.225782	77.151724	7	2018
2018-07-12	-0.335623	77.151724	7	2018
2018-07-13	-0.820834	77.151724	7	2018
2018-07-14	-0.418168	77.151724	7	2018
2018-07-15	-1.409618	77.151724	7	2018
…	…	…	…	…
2020-10-09	-1.058768	77.151724	10	2020
2020-10-10	-0.643308	77.151724	10	2020
2020-10-11	-0.905575	77.151724	10	2020
2020-10-12	-1.914576	77.151724	10	2020
2020-10-13	-1.476454	77.151724	10	2020

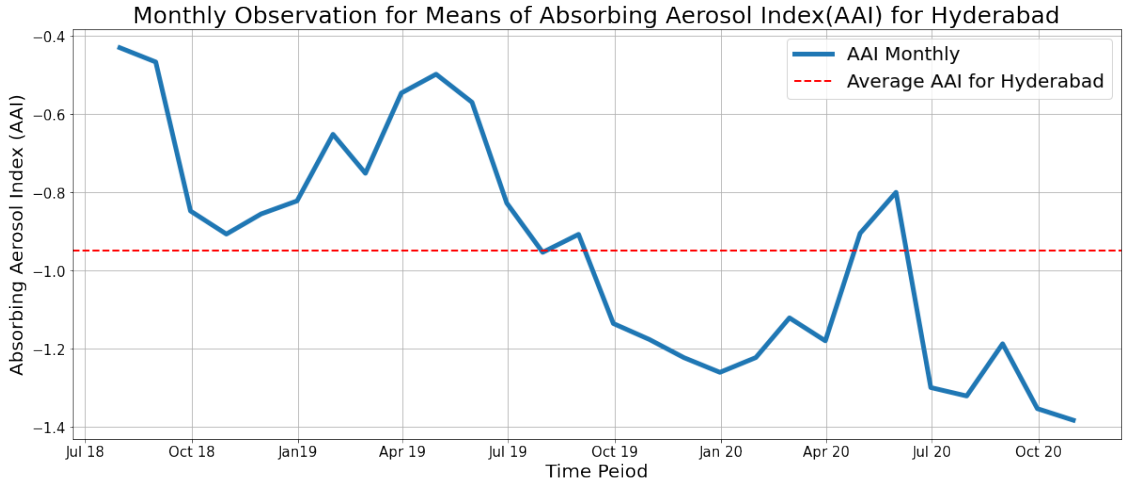
[800 rows x 4 columns]

[87]:
```
# Aggregate statistics grouped by Weeks
Weekly_Hyderabad_mean = Hyderabad.groupby(pd.Grouper(freq="W-MON")).mean()
Weekly_Hyderabad_max = Hyderabad.groupby(pd.Grouper(freq="W-MON")).max()
Weekly_Hyderabad_min = Hyderabad.groupby(pd.Grouper(freq="W-MON")).min()

# Aggregate statistics grouped by Months
Monthly_Hyderabad_mean = Hyderabad.groupby(pd.Grouper(freq="M")).mean()
Monthly_Hyderabad_max = Hyderabad.groupby(pd.Grouper(freq="M")).max()
Monthly_Hyderabad_min = Hyderabad.groupby(pd.Grouper(freq="M")).min()

#Week Identifier
Weekly_Hyderabad_mean["week"] = Weekly_Hyderabad_mean.index.isocalendar().week
Weekly_Hyderabad_min["week"] = Weekly_Hyderabad_min.index.isocalendar().week
Weekly_Hyderabad_max["week"] = Weekly_Hyderabad_max.index.isocalendar().week

Monthly_Hyderabad_mean["week"] = Monthly_Hyderabad_mean.index.isocalendar().week
Monthly_Hyderabad_min["week"] = Monthly_Hyderabad_min.index.isocalendar().week
Monthly_Hyderabad_max["week"] = Monthly_Hyderabad_max.index.isocalendar().week
```

[88]:
```
# Visualizing by grouping into weeekly data
plt.figure(figsize = (20,8))
```

```
plt.plot( Weekly_Hyderabad_mean.index.tolist(),Weekly_Hyderabad_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Weekly")
plt.gca().axhline(y=Weekly_Hyderabad_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Hyderabad")
plt.title("Weekly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Hyderabad",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()

# Visualizing by grouping into Monthly data
plt.figure(figsize = (20,8))
plt.plot( Monthly_Hyderabad_mean.index.tolist(),Monthly_Hyderabad_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Monthly")
plt.gca().axhline(y=Monthly_Hyderabad_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Hyderabad")
plt.title("Monthly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Hyderabad",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()
```





We will now plot some useful insights with Monthly Data

[89]:
```
# Plotting Monthly means
plt.figure(figsize=(10,12))
plt.plot(list(Monthly_Hyderabad_min.index),list(Monthly_Hyderabad_min["mean"]),␣
 ↪label = "Monthly Low", color = "blue" )
plt.plot(list(Monthly_Hyderabad_mean.
 ↪index),list(Monthly_Hyderabad_mean["mean"]), label= "Monthly Mean" , color␣
 ↪="green")
plt.plot(list(Monthly_Hyderabad_max.index),list(Monthly_Hyderabad_max["mean"]),␣
 ↪label = "Monthly High", color = "red" )

plt.xlabel("Months of respective years",fontsize= 15)
plt.gca().spines['top'].set_visible(False)
```

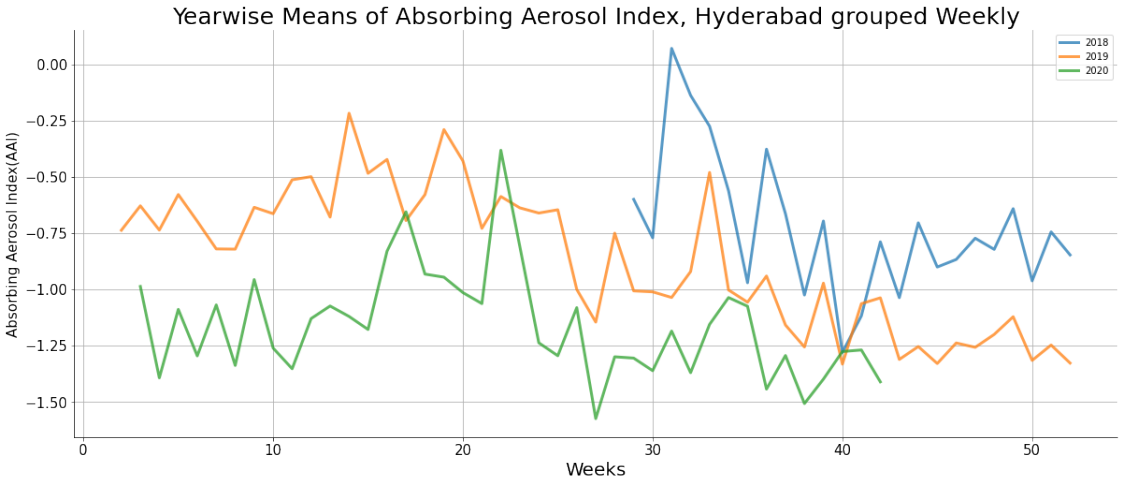
```
plt.gca().spines['right'].set_visible(False)
plt.title("Monthly Means of Absorbing Aerosol Index, Hyderabad",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 15)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels)
plt.legend()
plt.grid()
```

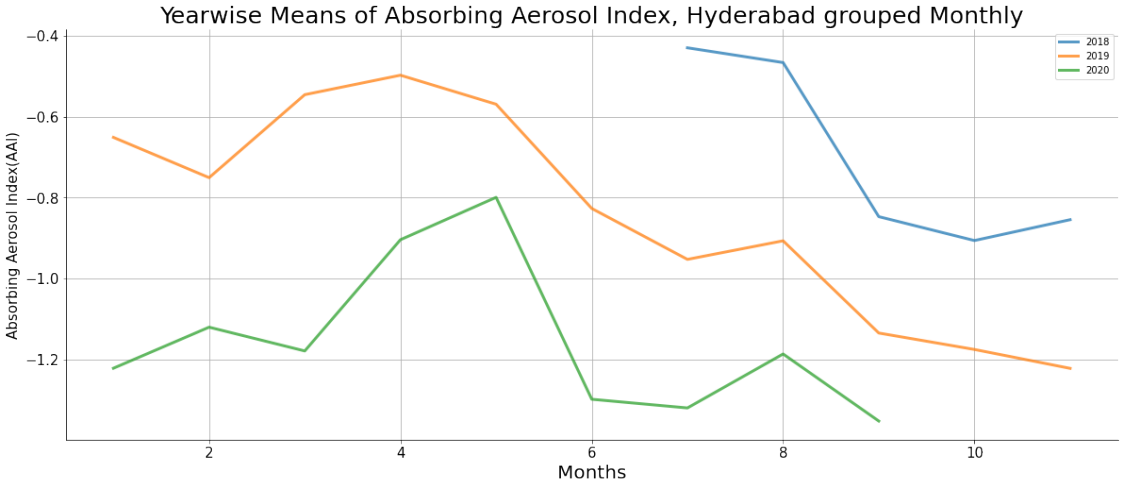
We can observe that Aerosol Optical Depth has decreased in the past year. The change in policies, Even Odd Rule and Coronavirus Pandemic seems to have shifted the curve down.

[90]:
```
# Plotting Yearwise means
x = [i for i in range(1,13)]
plt.figure(figsize = (20,8))
plt.plot(Weekly_Hyderabad_mean[Weekly_Hyderabad_mean["year"] == 2018]["week"].
 ↪tolist()[:-1],Weekly_Hyderabad_mean[Weekly_Hyderabad_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Weekly_Hyderabad_mean[Weekly_Hyderabad_mean["year"] == 2019]["week"].
 ↪tolist()[:-1],Weekly_Hyderabad_mean[Weekly_Hyderabad_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Weekly_Hyderabad_mean[Weekly_Hyderabad_mean["year"] == 2020]["week"].
 ↪tolist()[:-1],Weekly_Hyderabad_mean[Weekly_Hyderabad_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Weeks",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Yearwise Means of Absorbing Aerosol Index, Hyderabad grouped␣
 ↪Weekly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.grid()
plt.legend()

plt.figure(figsize = (20,8))
plt.plot(Monthly_Hyderabad_mean[Monthly_Hyderabad_mean["year"] ==␣
 ↪2018]["month"].tolist()[:
 ↪-1],Monthly_Hyderabad_mean[Monthly_Hyderabad_mean["year"] == 2018]["mean"].
 ↪tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Monthly_Hyderabad_mean[Monthly_Hyderabad_mean["year"] ==␣
 ↪2019]["month"].tolist()[:
 ↪-1],Monthly_Hyderabad_mean[Monthly_Hyderabad_mean["year"] == 2019]["mean"].
 ↪tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Monthly_Hyderabad_mean[Monthly_Hyderabad_mean["year"] ==␣
 ↪2020]["month"].tolist()[:
 ↪-1],Monthly_Hyderabad_mean[Monthly_Hyderabad_mean["year"] == 2020]["mean"].
 ↪tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Months",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
```

```
plt.title("Yearwise Means of Absorbing Aerosol Index, Hyderabad grouped␣
 ↪Monthly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.grid()
plt.legend()
```

[90]: <matplotlib.legend.Legend at 0x241be4971f0>





The observation from the above graph is that, the Absorbing Aerosol Index values have come down considerably in the last two years. From the graph it is evident that the Aerosol Index values are lesser in 2020. Therefore it can be concluded that the polution in Delhi has been lower in 2020 compared to the previous two years.

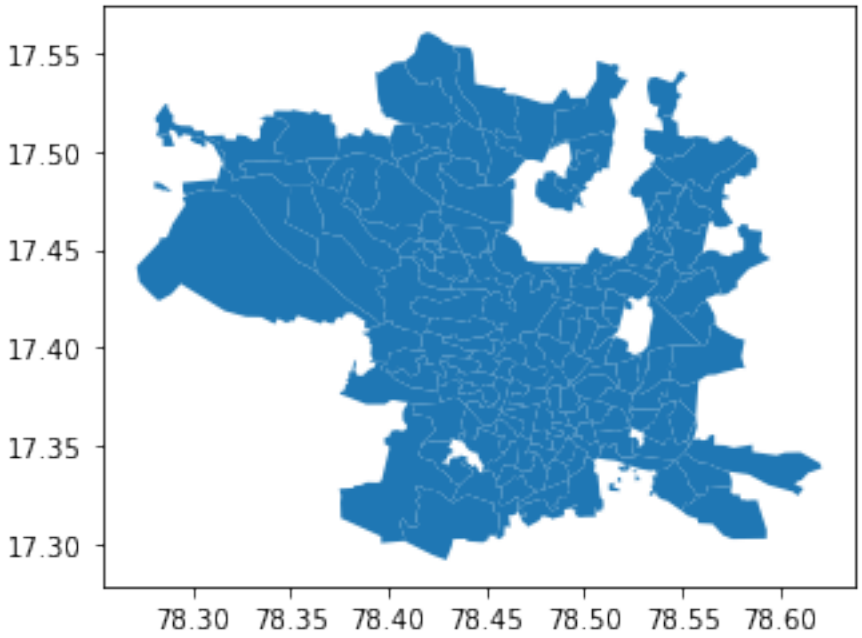
3.9.2 Plotting Maps For Insights

[91]:
```
# Grouping by Wards Data for further analysis
Hyderabad_wards = df["Hyderabad"].groupby(["Ward_No"]).mean()
Hyderabad_wards
```

[91]:
```
              mean
Ward_No
1       -1.023057
2       -0.998717
5       -0.991334
6       -0.973988
7       -0.976252
…             …
146     -0.939073
147     -0.939450
148     -0.948392
149     -0.966376
150     -0.955259

[144 rows x 1 columns]
```

[92]:
```
# loading the shapefile
path = r'Cleaned CSVs/Hyderabad/ghmc-wards-polygon.shp'
map_Hyderabad = gpd.read_file(path)
map_Hyderabad["Ward_No"] = [i.split()[1] for i in map_Hyderabad["name"].
 ↪tolist()]
map_Hyderabad["Ward_No"] = map_Hyderabad["Ward_No"].astype(int)
# Viewing the shapefile
map_Hyderabad.plot()
```

[92]:
```
<AxesSubplot:>
```



[93]:
```
# merging the shapefile with the mean data
merged_Hyderabad = map_Hyderabad.merge(Hyderabad_wards, left_on = "Ward_No",␣
↪right_index = True)
# merged_Hyderabad.head()
merged_Hyderabad
```

[93]:

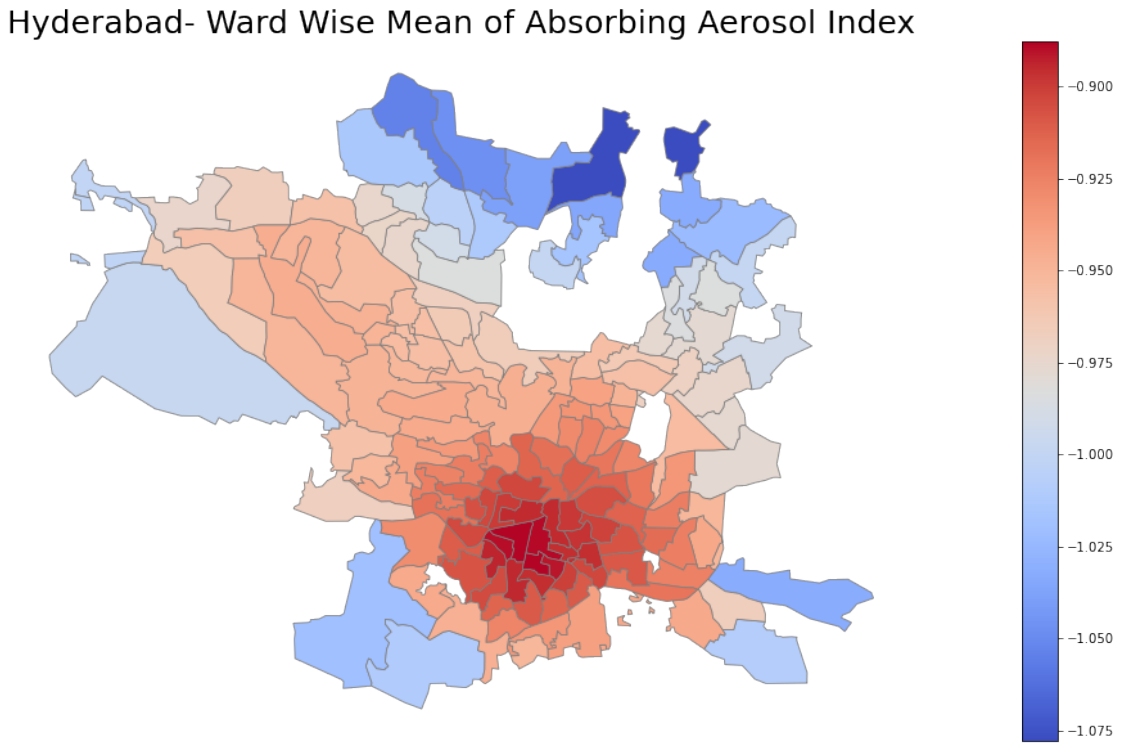
	id	@id	admin_leve	boundary	\
0	relation/7848799	relation/7848799	10	administrative	
1	relation/7848824	relation/7848824	10	administrative	
2	relation/7848844	relation/7848844	10	administrative	
3	relation/7848859	relation/7848859	10	administrative	
4	relation/7849315	relation/7849315	10	administrative	
..	…	…	…	…	
140	relation/7867250	relation/7867250	10	administrative	
141	relation/7867251	relation/7867251	10	administrative	
142	relation/7867252	relation/7867252	10	administrative	
143	relation/7867584	relation/7867584	10	administrative	
144	relation/7867585	relation/7867585	10	administrative	

	name	type	@timestamp	@version	@changeset	\
0	Ward 91 Khairatabad	boundary	2017-12-31	5	55064631	
1	Ward 105 Gachibowli	boundary	2018-01-01	4	55086287	
2	Ward 60 Rajendra Nagar	boundary	2018-01-02	4	55094174	
3	Ward 95 Jubilee Hills	boundary	2017-12-31	5	55064040	
4	Ward 6 Nacharam	boundary	2017-12-30	4	55028223	

```
..                          …       …          …      …         …
140  Ward 112 Ramachandrapuram  boundary  2018-01-03        2   55128738
141     Ward 111 Bharati Nagar  boundary  2018-01-03        2   55128738
142         Ward 30 Dabeerpura  boundary  2018-01-02        3   55100180
143             Ward 42 Barkas  boundary  2018-01-03        2   55128738
144        Ward 74 Ahmed Nagar  boundary  2018-01-02        1   55091137

          @user    @uid                                              geometry  \
0    adivik2000  250196  POLYGON ((78.45243 17.42676, 78.45292 17.42751…
1    adivik2000  250196  POLYGON ((78.31078 17.48009, 78.31962 17.47938…
2    adivik2000  250196  POLYGON ((78.41015 17.37208, 78.41221 17.37149…
3    adivik2000  250196  POLYGON ((78.43691 17.43014, 78.43647 17.42948…
4    adivik2000  250196  POLYGON ((78.54860 17.43451, 78.55180 17.43367…
..          …     …                                                   …
140  adivik2000  250196  POLYGON ((78.28733 17.52436, 78.28748 17.52315…
141  adivik2000  250196  MULTIPOLYGON (((78.28909 17.50801, 78.29091 17…
142  adivik2000  250196  POLYGON ((78.49242 17.36115, 78.49253 17.35985…
143  adivik2000  250196  POLYGON ((78.48316 17.31318, 78.47867 17.31432…
144  adivik2000  250196  POLYGON ((78.45301 17.40340, 78.45333 17.40291…

     Ward_No      mean
0         91 -0.946269
1        105 -0.996943
2         60 -1.019533
3         95 -0.948463
4          6 -0.973988
..       …       …
140      112 -1.000160
141      111 -1.002030
142       30 -0.894636
143       42 -0.950689
144       74 -0.935328

[145 rows x 14 columns]
```

[94]:
```
# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values
vmin, vmax = merged_Hyderabad["mean"].min(), merged_Hyderabad["mean"].max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))
# remove the axis
ax.axis('off')
# add a title and annotation
ax.set_title('Hyderabad- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
```

```
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))
# empty array for the data range
sm.set_array([])
# add the colorbar to the figure
fig.colorbar(sm)
# create map
merged_Hyderabad.plot(column=variable, cmap='coolwarm', linewidth=0.8, ax=ax,␣
 ↪edgecolor='0.5', figsize = (5,15))
plt.show()
```



[95]:
```
# Using T-Test to determine if climate has changed in the first 14 and last 14␣
 ↪months in Hyderabad
ttest_ind(Monthly_Hyderabad_mean.iloc[:14,:]["mean"].
 ↪tolist(),Monthly_Hyderabad_mean.iloc[14:,:]["mean"].tolist())
```

[95]:
```
Ttest_indResult(statistic=7.138772235499001, pvalue=1.3971545720645874e-07)
```

We observe that Pollution levels has decreased in the last 14 months and the statistical test proves the same with a very small p-value for 95 % Confidence Interval

[ ]:

3.10 City Number 10: Jaipur

3.10.1 Loading and Visualizing

[96]:
```
# g = df1.groupby(pd.Grouper(freq="M"))
# g.mean()
Jaipur = df["Jaipur"].groupby(["date"]).mean()
Jaipur["month"] = Jaipur.index.month
Jaipur["year"] = Jaipur.index.year
Jaipur
```

[96]:

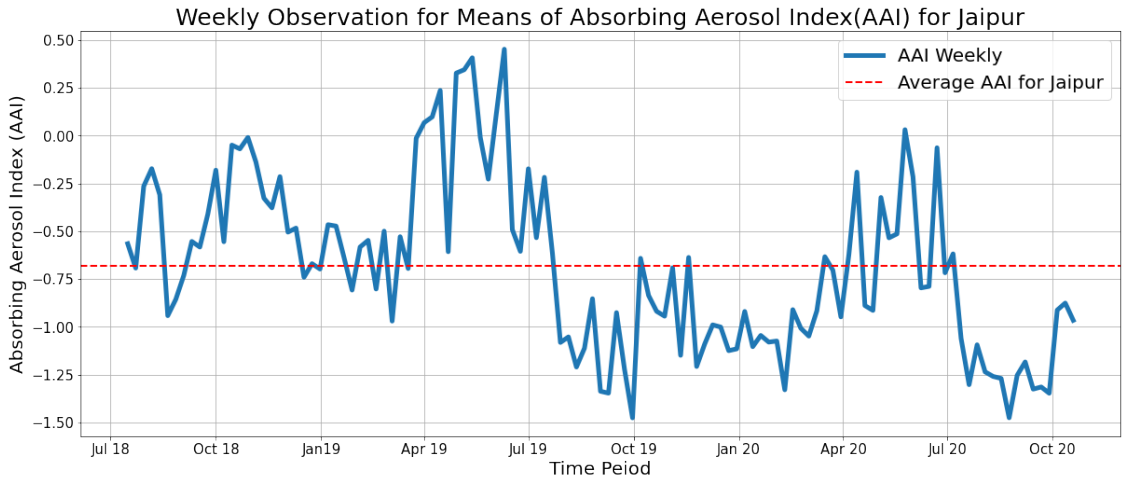
date	Ward_No	mean	month	year
2018-07-11	39.0	-0.091655	7	2018
2018-07-12	39.0	-0.349608	7	2018
2018-07-13	39.0	-0.133120	7	2018
2018-07-15	39.0	-1.357538	7	2018
2018-07-16	39.0	-0.895579	7	2018
…	…	…	…	…
2020-10-09	39.0	-0.805891	10	2020
2020-10-10	39.0	-0.819039	10	2020
2020-10-11	39.0	-0.731614	10	2020
2020-10-12	39.0	-0.864263	10	2020
2020-10-13	39.0	-0.966210	10	2020

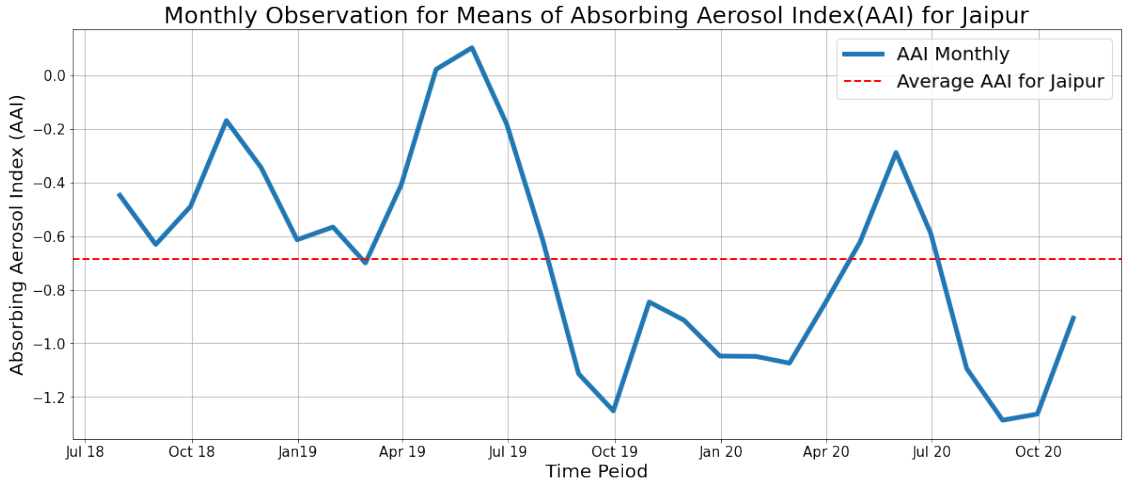
[798 rows x 4 columns]

[97]:
```
# Aggregate statistics grouped by Weeks
Weekly_Jaipur_mean = Jaipur.groupby(pd.Grouper(freq="W-MON")).mean()
Weekly_Jaipur_max = Jaipur.groupby(pd.Grouper(freq="W-MON")).max()
Weekly_Jaipur_min = Jaipur.groupby(pd.Grouper(freq="W-MON")).min()

# Aggregate statistics grouped by Months
Monthly_Jaipur_mean = Jaipur.groupby(pd.Grouper(freq="M")).mean()
Monthly_Jaipur_max = Jaipur.groupby(pd.Grouper(freq="M")).max()
Monthly_Jaipur_min = Jaipur.groupby(pd.Grouper(freq="M")).min()

#Week Identifier
Weekly_Jaipur_mean["week"] = Weekly_Jaipur_mean.index.isocalendar().week
Weekly_Jaipur_min["week"] = Weekly_Jaipur_min.index.isocalendar().week
Weekly_Jaipur_max["week"] = Weekly_Jaipur_max.index.isocalendar().week

Monthly_Jaipur_mean["week"] = Monthly_Jaipur_mean.index.isocalendar().week
Monthly_Jaipur_min["week"] = Monthly_Jaipur_min.index.isocalendar().week
Monthly_Jaipur_max["week"] = Monthly_Jaipur_max.index.isocalendar().week
```

[98]:
```
# Visualizing by grouping into weeekly data
plt.figure(figsize = (20,8))
```

```
plt.plot( Weekly_Jaipur_mean.index.tolist(),Weekly_Jaipur_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Weekly")
plt.gca().axhline(y=Weekly_Jaipur_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Jaipur")
plt.title("Weekly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Jaipur",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()

# Visualizing by grouping into Monthly data
plt.figure(figsize = (20,8))
plt.plot( Monthly_Jaipur_mean.index.tolist(),Monthly_Jaipur_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Monthly")
plt.gca().axhline(y=Monthly_Jaipur_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Jaipur")
plt.title("Monthly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Jaipur",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()
```

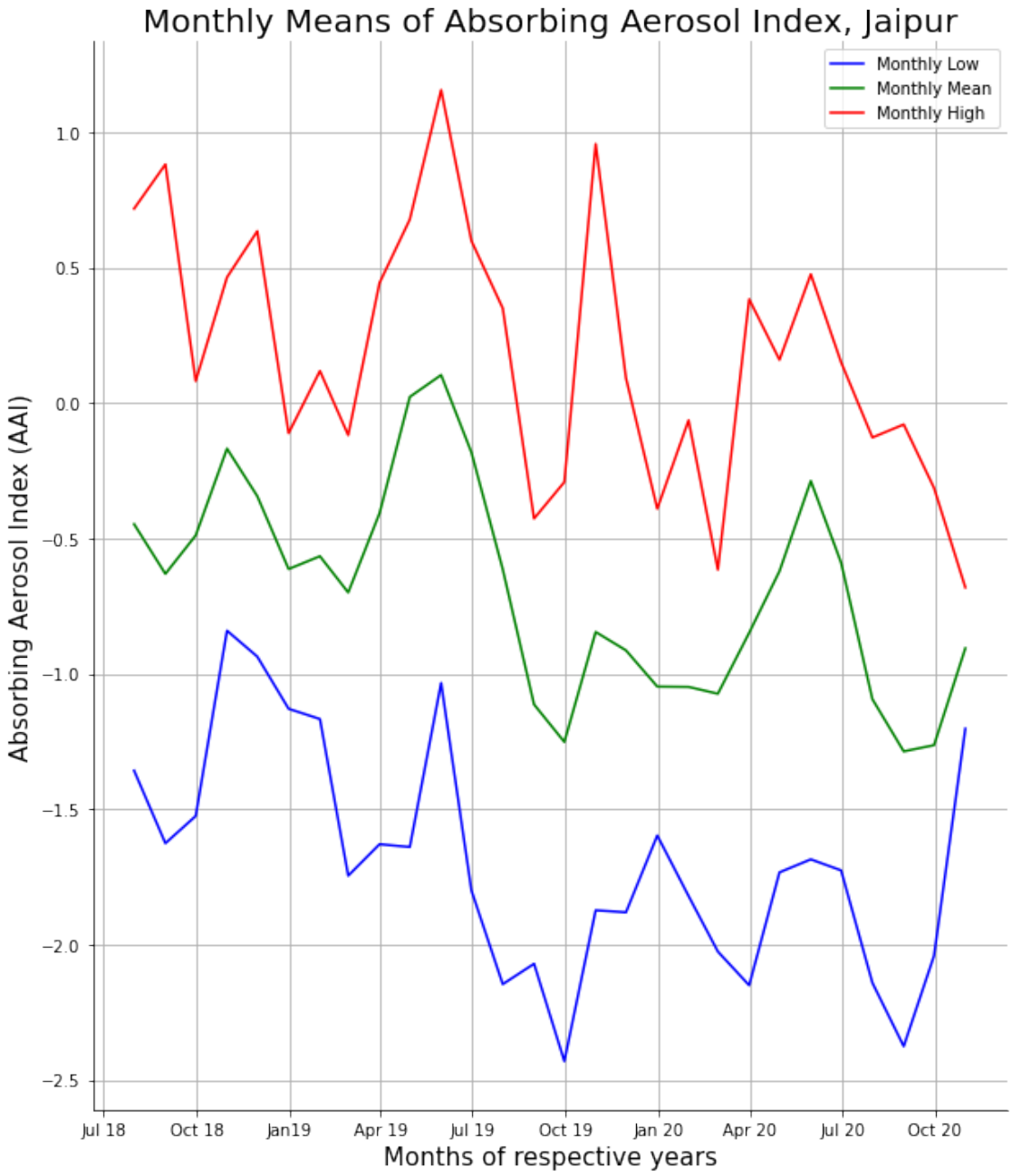




We will now plot some useful insights with Monthly Data

[99]:
```
# Plotting Monthly means
plt.figure(figsize=(10,12))
plt.plot(list(Monthly_Jaipur_min.index),list(Monthly_Jaipur_min["mean"]), label␣
 ↪= "Monthly Low", color = "blue" )
plt.plot(list(Monthly_Jaipur_mean.index),list(Monthly_Jaipur_mean["mean"]),␣
 ↪label= "Monthly Mean" , color ="green")
plt.plot(list(Monthly_Jaipur_max.index),list(Monthly_Jaipur_max["mean"]), label␣
 ↪= "Monthly High", color = "red" )

plt.xlabel("Months of respective years",fontsize= 15)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
```

```
plt.title("Monthly Means of Absorbing Aerosol Index, Jaipur",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 15)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels)
plt.legend()
plt.grid()
```

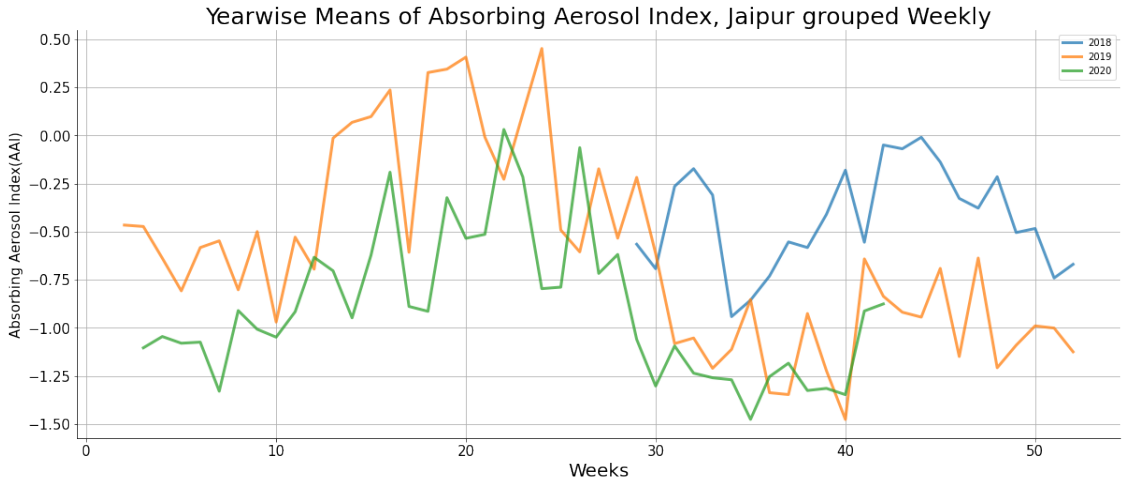


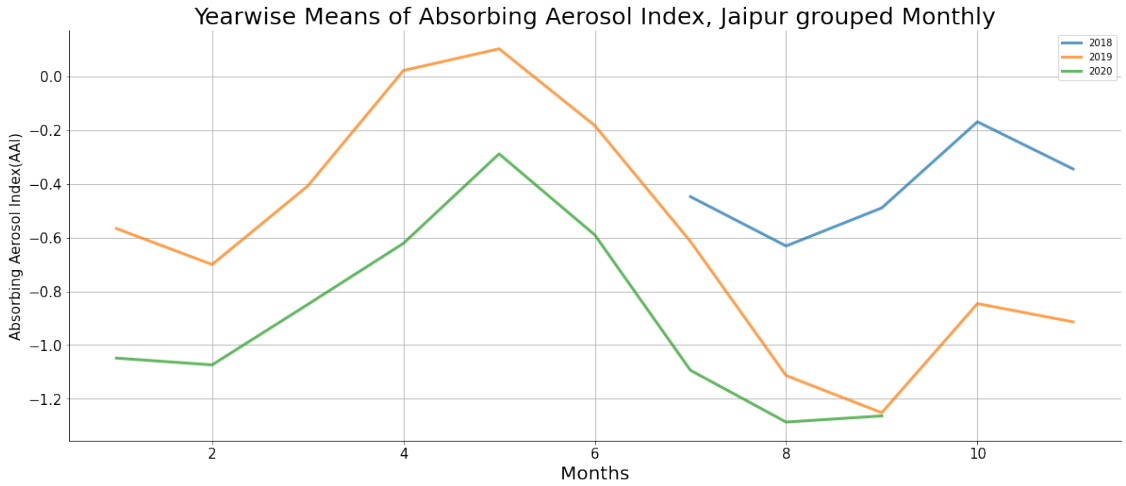
We can observe that Aerosol Optical Depth has decreased in the past year. The change in policies, Even Odd Rule and Coronavirus Pandemic seems to have shifted the curve down.

[100]:
```
# Plotting Yearwise means
x = [i for i in range(1,13)]
plt.figure(figsize = (20,8))
plt.plot(Weekly_Jaipur_mean[Weekly_Jaipur_mean["year"] == 2018]["week"].
 ↪tolist()[:-1],Weekly_Jaipur_mean[Weekly_Jaipur_mean["year"] == 2018]["mean"].
 ↪tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Weekly_Jaipur_mean[Weekly_Jaipur_mean["year"] == 2019]["week"].
 ↪tolist()[:-1],Weekly_Jaipur_mean[Weekly_Jaipur_mean["year"] == 2019]["mean"].
 ↪tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Weekly_Jaipur_mean[Weekly_Jaipur_mean["year"] == 2020]["week"].
 ↪tolist()[:-1],Weekly_Jaipur_mean[Weekly_Jaipur_mean["year"] == 2020]["mean"].
 ↪tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Weeks",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Yearwise Means of Absorbing Aerosol Index, Jaipur grouped␣
 ↪Weekly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.grid()
plt.legend()

plt.figure(figsize = (20,8))
plt.plot(Monthly_Jaipur_mean[Monthly_Jaipur_mean["year"] == 2018]["month"].
 ↪tolist()[:-1],Monthly_Jaipur_mean[Monthly_Jaipur_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Monthly_Jaipur_mean[Monthly_Jaipur_mean["year"] == 2019]["month"].
 ↪tolist()[:-1],Monthly_Jaipur_mean[Monthly_Jaipur_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Monthly_Jaipur_mean[Monthly_Jaipur_mean["year"] == 2020]["month"].
 ↪tolist()[:-1],Monthly_Jaipur_mean[Monthly_Jaipur_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Months",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.title("Yearwise Means of Absorbing Aerosol Index, Jaipur grouped␣
 ↪Monthly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.grid()
```

```
plt.legend()
```

[100]: <matplotlib.legend.Legend at 0x241bb214580>





The observation from the above graph is that, the Absorbing Aerosol Index values have come down considerably in the last two years. From the graph it is evident that the Aerosol Index values are lesser in 2020. Therefore it can be concluded that the polution in Delhi has been lower in 2020 compared to the previous two years.

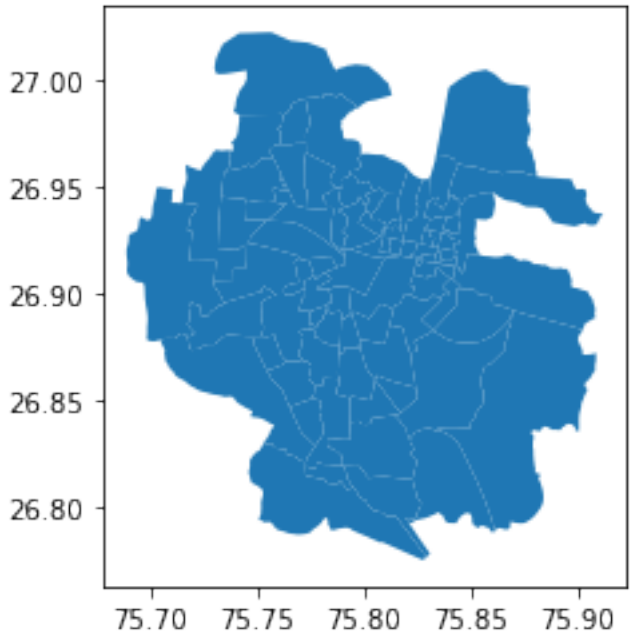
3.10.2 Plotting Maps For Insights

[101]:
```
# Grouping by Wards Data for further analysis
Jaipur_wards = df["Jaipur"].groupby(["Ward_No"]).mean()
Jaipur_wards
```

[101]:
```
              mean
Ward_No
1       -0.765325
2       -0.678072
3       -0.694134
4       -0.713368
5       -0.662916
…             …
73      -0.766756
74      -0.769348
75      -0.806649
76      -0.837085
77      -0.859528

[77 rows x 1 columns]
```

[102]:
```
# loading the shapefile
path = r'Cleaned CSVs/Jaipur/Jaipur_Wards-polygon.shp'
map_Jaipur = gpd.read_file(path)
map_Jaipur["WARD_NO"] = map_Jaipur["WARD_NO"].astype(int)
# Viewing the shapefile
map_Jaipur.plot()
```

[102]:
```
<AxesSubplot:>
```



[103]:
```
# merging the shapefile with the mean data
merged_Jaipur = map_Jaipur.merge(Jaipur_wards, left_on = "WARD_NO", right_index␣
↪= True)
# merged_Jaipur.head()
merged_Jaipur
```

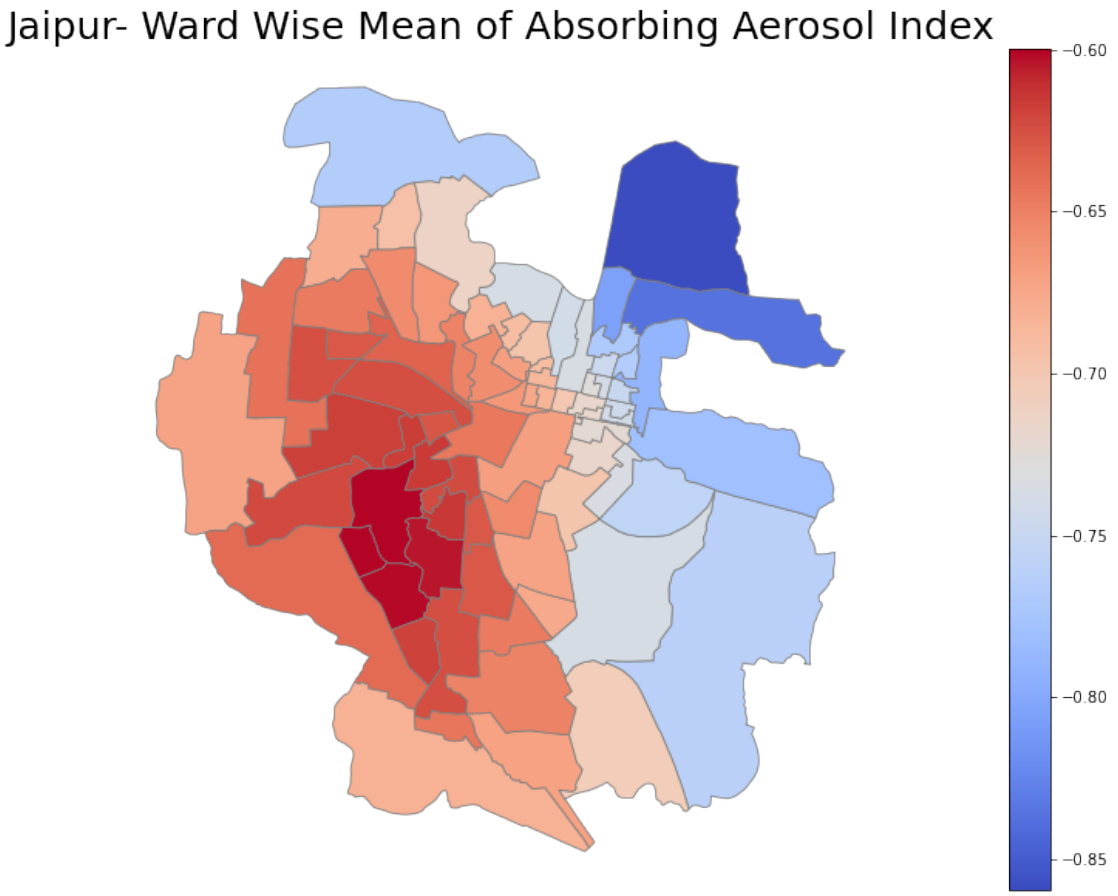
[103]:

	OBJECTID	WARD_NO	ZONE_NAME	AC_NO	AC	Shape_Leng	\
0	1	76	HAWA MAHAL	49	HAWA MAHAL	0.185122	
1	2	34	BAGRU ANSIK	56	BAGRU ANSIK	0.323362	
2	3	11	JHOTWARA ANSIK	46	JHOTWARA ANSIK	0.196200	
3	4	12	JHOTWARA ANSIK	46	JHOTWARA ANSIK	0.239431	
4	5	13	JHOTWARA ANSIK	46	JHOTWARA ANSIK	0.128021	
..	…	…	…	…	…	…	
72	73	6	VIDHYADHAR NAGAR	50	VIDHYADHAR NAGAR	0.085999	
73	74	7	VIDHYADHAR NAGAR	50	VIDHYADHAR NAGAR	0.094575	
74	75	9	VIDHYADHAR NAGAR	50	VIDHYADHAR NAGAR	0.052066	
75	76	10	VIDHYADHAR NAGAR	50	VIDHYADHAR NAGAR	0.086027	
76	77	8	VIDHYADHAR NAGAR	50	VIDHYADHAR NAGAR	0.138839	

	Shape_Area	geometry	mean
0	0.001005	POLYGON ((75.88071 26.95447, 75.88130 26.95433…	-0.837085
1	0.003600	POLYGON ((75.86130 26.78845, 75.86092 26.78949…	-0.760570
2	0.000891	POLYGON ((75.73039 26.96676, 75.73187 26.96650…	-0.641574
3	0.001858	POLYGON ((75.71488 26.94230, 75.71520 26.94221…	-0.671061
4	0.000630	POLYGON ((75.74591 26.92325, 75.74612 26.92299…	-0.617623

```
..          …                                                   …       …
72    0.000324  POLYGON ((75.77275 26.96986, 75.77280 26.96968… -0.656043
73    0.000436  POLYGON ((75.75743 26.96307, 75.75751 26.96297… -0.647106
74    0.000131  POLYGON ((75.75388 26.94220, 75.75474 26.94215… -0.628341
75    0.000348  POLYGON ((75.74407 26.93651, 75.74439 26.93378… -0.624840
76    0.000343  POLYGON ((75.76354 26.94992, 75.76496 26.94704… -0.633425

[77 rows x 9 columns]
```

[104]:
```
# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values
vmin, vmax = merged_Jaipur["mean"].min(), merged_Jaipur["mean"].max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))
# remove the axis
ax.axis('off')
# add a title and annotation
ax.set_title('Jaipur- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))
# empty array for the data range
sm.set_array([])
# add the colorbar to the figure
fig.colorbar(sm)
# create map
merged_Jaipur.plot(column=variable, cmap='coolwarm', linewidth=0.8, ax=ax,␣
 ↪edgecolor='0.5', figsize = (5,15))
plt.show()
```



[105]:
```
# Using T-Test to determine if climate has changed in the first 14 and last 14 months in Jaipur
ttest_ind(Monthly_Jaipur_mean.iloc[:14,:]["mean"].tolist(),Monthly_Jaipur_mean.iloc[14:,:]["mean"].tolist())
```

[105]:
```
Ttest_indResult(statistic=4.35364947430119, pvalue=0.00018499961790097988)
```

We observe that Pollution levels has decreased in the last 14 months and the statistical test proves the same with a very small p-value for 95 % Confidence Interval

[ ]:
```
```

3.11 City Number 11: Kolkata

[106]:
```
# g = df1.groupby(pd.Grouper(freq="M"))
# g.mean()
Kolkata = df["Kolkata"].groupby(["date"]).mean()
Kolkata["month"] = Kolkata.index.month
Kolkata["year"] = Kolkata.index.year
```

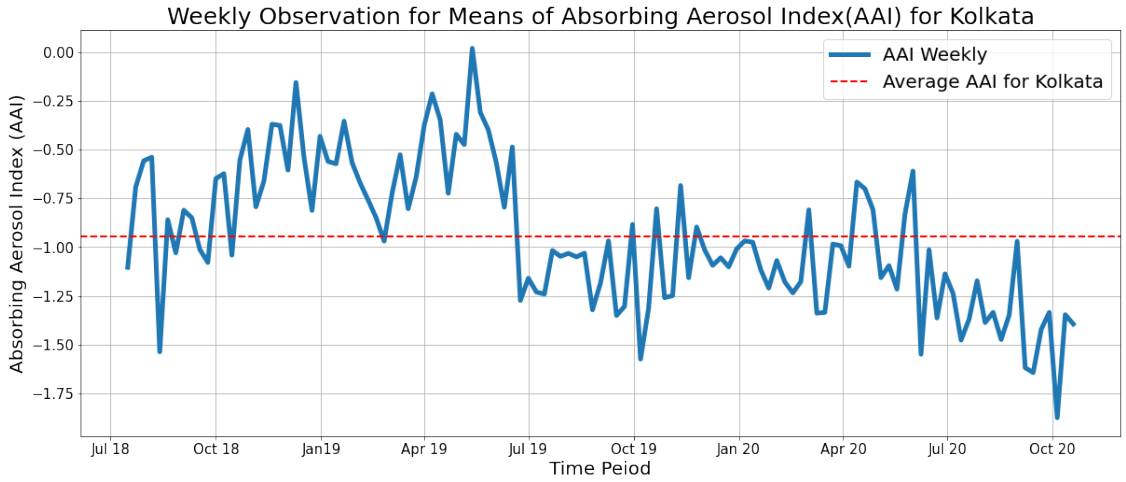
```
Kolkata
```

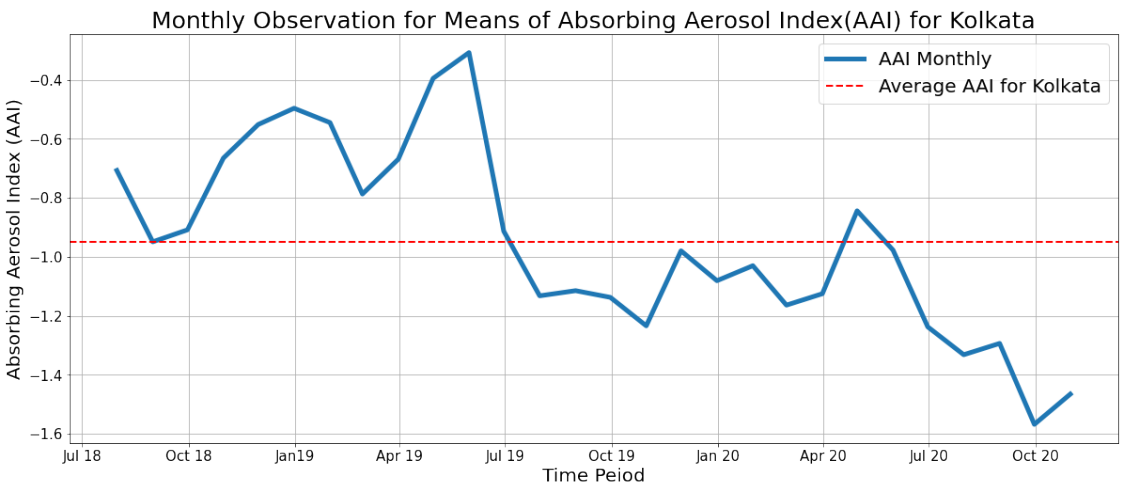
[106]:
```
            Ward_No      mean  month  year
date
2018-07-11       71 -1.469816      7  2018
2018-07-12       71 -1.446685      7  2018
2018-07-13       71 -0.944395      7  2018
2018-07-14       71 -0.602620      7  2018
2018-07-15       71 -1.058017      7  2018
…               …        …     …     …
2020-10-09       71 -1.950946     10  2020
2020-10-10       71 -1.609517     10  2020
2020-10-11       71 -1.321279     10  2020
2020-10-12       71 -1.076181     10  2020
2020-10-13       71 -1.394773     10  2020

[794 rows x 4 columns]
```

[107]:
```
# Aggregate statistics grouped by Weeks
Weekly_Kolkata_mean = Kolkata.groupby(pd.Grouper(freq="W-MON")).mean()
Weekly_Kolkata_max = Kolkata.groupby(pd.Grouper(freq="W-MON")).max()
Weekly_Kolkata_min = Kolkata.groupby(pd.Grouper(freq="W-MON")).min()

# Aggregate statistics grouped by Months
Monthly_Kolkata_mean = Kolkata.groupby(pd.Grouper(freq="M")).mean()
Monthly_Kolkata_max = Kolkata.groupby(pd.Grouper(freq="M")).max()
Monthly_Kolkata_min = Kolkata.groupby(pd.Grouper(freq="M")).min()

#Week Identifier
Weekly_Kolkata_mean["week"] = Weekly_Kolkata_mean.index.isocalendar().week
Weekly_Kolkata_min["week"] = Weekly_Kolkata_min.index.isocalendar().week
Weekly_Kolkata_max["week"] = Weekly_Kolkata_max.index.isocalendar().week

Monthly_Kolkata_mean["week"] = Monthly_Kolkata_mean.index.isocalendar().week
Monthly_Kolkata_min["week"] = Monthly_Kolkata_min.index.isocalendar().week
Monthly_Kolkata_max["week"] = Monthly_Kolkata_max.index.isocalendar().week
```

[108]:
```
# Visualizing by grouping into weeekly data
plt.figure(figsize = (20,8))
plt.plot( Weekly_Kolkata_mean.index.tolist(),Weekly_Kolkata_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Weekly")
plt.gca().axhline(y=Weekly_Kolkata_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Kolkata")
plt.title("Weekly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Kolkata",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
```

```
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()

# Visualizing by grouping into Monthly data
plt.figure(figsize = (20,8))
plt.plot( Monthly_Kolkata_mean.index.tolist(),Monthly_Kolkata_mean["mean"],␣
 ↪linewidth = 5, label = "AAI Monthly")
plt.gca().axhline(y=Monthly_Kolkata_mean["mean"].mean(),color='r',␣
 ↪linestyle='--', lw=2, label = "Average AAI for Kolkata")
plt.title("Monthly Observation for Means of Absorbing Aerosol Index(AAI) for␣
 ↪Kolkata",fontsize = 25)
plt.xlabel("Time Peiod",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 20)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
 ↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels, fontsize = 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
# plt.gca().set_yticklabels(plt.gca().get_yticklabels(),fontsize = 10)
plt.legend(fontsize = 20)
plt.grid()
```





We will now plot some useful insights with Monthly Data

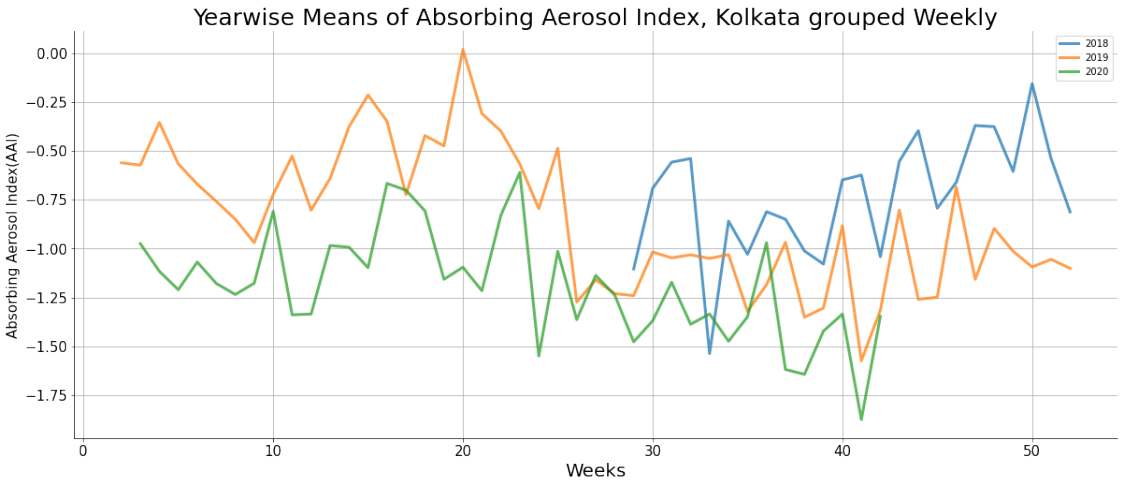
[109]:
```
# Plotting Monthly means
plt.figure(figsize=(10,12))
plt.plot(list(Monthly_Kolkata_min.index),list(Monthly_Kolkata_min["mean"]),␣
↪label = "Monthly Low", color = "blue" )
plt.plot(list(Monthly_Kolkata_mean.index),list(Monthly_Kolkata_mean["mean"]),␣
↪label= "Monthly Mean" , color ="green")
plt.plot(list(Monthly_Kolkata_max.index),list(Monthly_Kolkata_max["mean"]),␣
↪label = "Monthly High", color = "red" )

plt.xlabel("Months of respective years",fontsize= 15)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Monthly Means of Absorbing Aerosol Index, Kolkata",fontsize= 20)
plt.ylabel("Absorbing Aerosol Index (AAI)",fontsize= 15)
xticklabels = ['Jul 18','Oct 18','Jan19','Apr 19','Jul 19', 'Oct 19', 'Jan 20',␣
↪'Apr 20', 'Jul 20', 'Oct 20']
plt.gca().set_xticklabels(xticklabels)
plt.legend()
plt.grid()
```

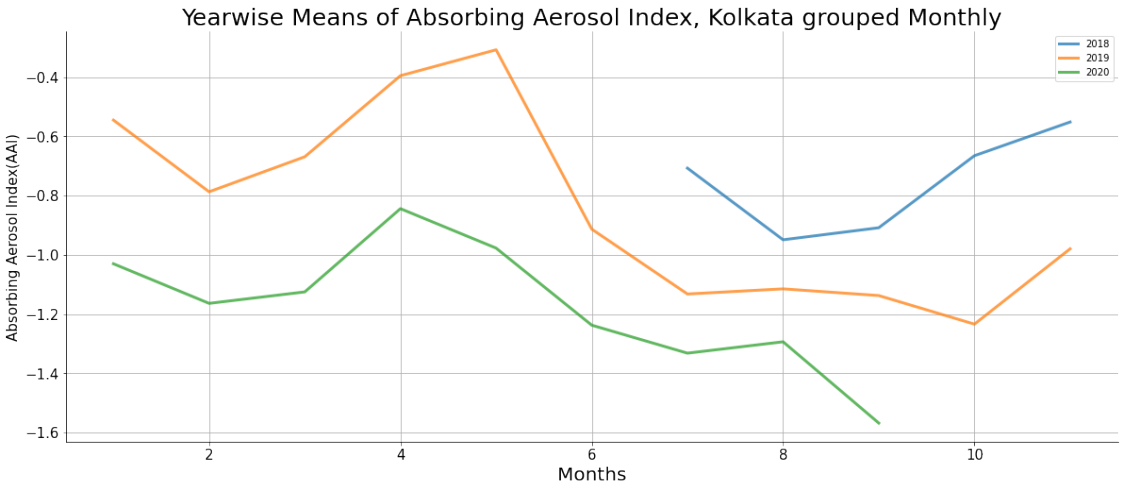
We can observe that Aerosol Optical Depth has decreased in the past year. The change in policies, Even Odd Rule and Coronavirus Pandemic seems to have shifted the curve down.

[110]:
```
# Plotting Yearwise means
x = [i for i in range(1,13)]
plt.figure(figsize = (20,8))
```

```
plt.plot(Weekly_Kolkata_mean[Weekly_Kolkata_mean["year"] == 2018]["week"].
 ↪tolist()[:-1],Weekly_Kolkata_mean[Weekly_Kolkata_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Weekly_Kolkata_mean[Weekly_Kolkata_mean["year"] == 2019]["week"].
 ↪tolist()[:-1],Weekly_Kolkata_mean[Weekly_Kolkata_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Weekly_Kolkata_mean[Weekly_Kolkata_mean["year"] == 2020]["week"].
 ↪tolist()[:-1],Weekly_Kolkata_mean[Weekly_Kolkata_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Weeks",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Yearwise Means of Absorbing Aerosol Index, Kolkata grouped␣
 ↪Weekly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.grid()
plt.legend()

plt.figure(figsize = (20,8))
plt.plot(Monthly_Kolkata_mean[Monthly_Kolkata_mean["year"] == 2018]["month"].
 ↪tolist()[:-1],Monthly_Kolkata_mean[Monthly_Kolkata_mean["year"] ==␣
 ↪2018]["mean"].tolist()[:-1] , label = "2018", lw = 3, alpha = 0.75)
plt.plot(Monthly_Kolkata_mean[Monthly_Kolkata_mean["year"] == 2019]["month"].
 ↪tolist()[:-1],Monthly_Kolkata_mean[Monthly_Kolkata_mean["year"] ==␣
 ↪2019]["mean"].tolist()[:-1] , label = "2019" , lw =3, alpha = 0.75)
plt.plot(Monthly_Kolkata_mean[Monthly_Kolkata_mean["year"] == 2020]["month"].
 ↪tolist()[:-1],Monthly_Kolkata_mean[Monthly_Kolkata_mean["year"] ==␣
 ↪2020]["mean"].tolist()[:-1]  , label = "2020", lw = 3, alpha = 0.75)

plt.xlabel("Months",fontsize= 20)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.title("Yearwise Means of Absorbing Aerosol Index, Kolkata grouped␣
 ↪Monthly",fontsize= 25)
plt.ylabel("Absorbing Aerosol Index(AAI)",fontsize= 15)
plt.grid()
plt.legend()
```

[110]: <matplotlib.legend.Legend at 0x241bb6c09a0>





The observation from the above graph is that, the Absorbing Aerosol Index values have come down considerably in the last two years. From the graph it is evident that the Aerosol Index values are lesser in 2020. Therefore it can be concluded that the polution in Delhi has been lower in 2020 compared to the previous two years.

3.11.1 Plotting Maps For Insights

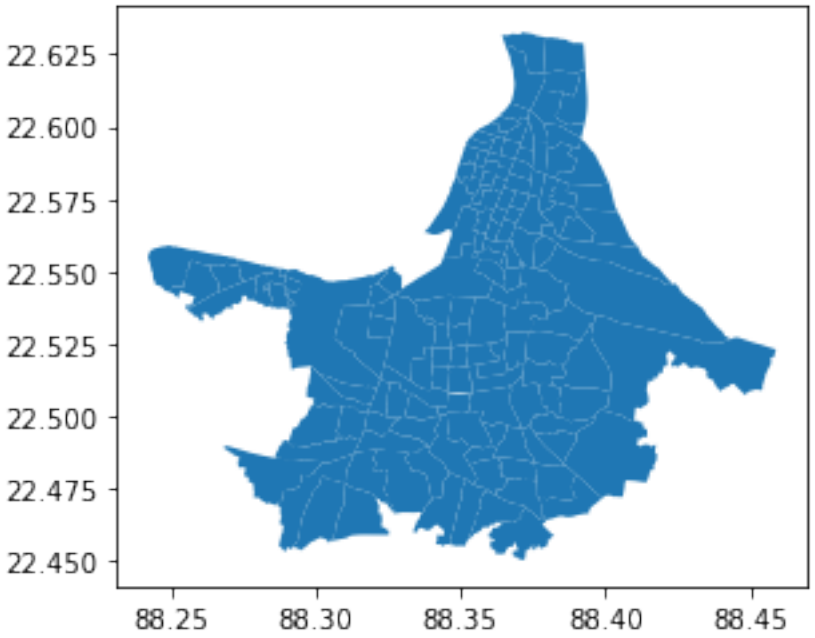
[111]:
```
# Grouping by Wards Data for further analysis
Kolkata_wards = df["Kolkata"].groupby(["Ward_No"]).mean()
Kolkata_wards
```

[111]:
```
              mean
Ward_No
1       -0.934003
```

```
2       -0.932568
3       -0.924106
4       -0.928921
5       -0.924812
…             …
137     -0.975984
138     -0.983189
139     -0.983234
140     -0.989563
141     -0.997091

[141 rows x 1 columns]
```

[112]:
```
# loading the shapefile
path = r'Cleaned CSVs/Kolkata/kolkata-polygon.shp'
map_Kolkata = gpd.read_file(path)
map_Kolkata["WARD"]= map_Kolkata["WARD"].astype(int)
# Viewing the shapefile
map_Kolkata.plot()
```

[112]: <AxesSubplot:>



[113]:
```
# merging the shapefile with the mean data
merged_Kolkata = map_Kolkata.merge(Kolkata_wards, left_on = "WARD", right_index␣
↪= True)
```

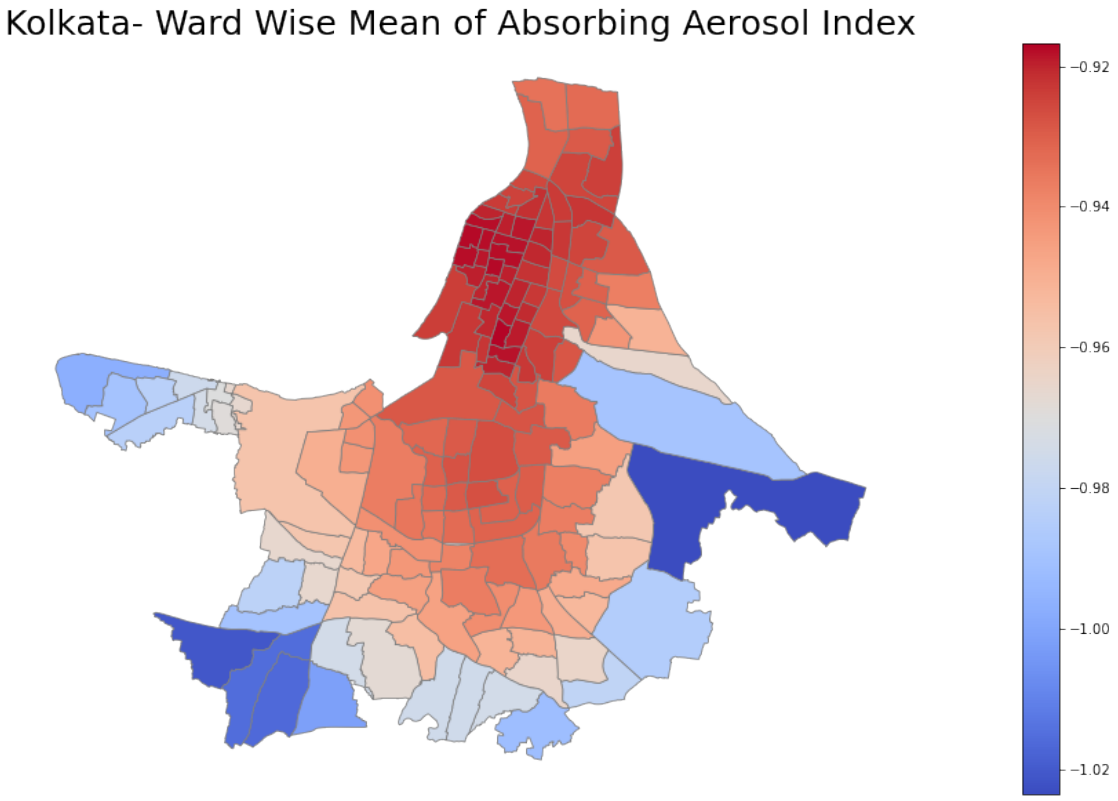
```
# merged_Kolkata.head()
merged_Kolkata
```

[113]:

```
     WARD                                               geometry      mean
0      93  POLYGON ((88.35296 22.50168, 88.35300 22.50204… -0.933401
1      61  POLYGON ((88.35655 22.55349, 88.35656 22.55349… -0.925042
2      86  POLYGON ((88.35978 22.52516, 88.35979 22.52516… -0.930224
3      90  POLYGON ((88.37161 22.51538, 88.37161 22.51535… -0.930679
4      26  POLYGON ((88.35878 22.58862, 88.35882 22.58873… -0.918075
..    …                                                  …       …
136    35  POLYGON ((88.37763 22.57034, 88.37826 22.57064… -0.930629
137   116  POLYGON ((88.32589 22.49510, 88.32589 22.49518… -0.945367
138    95  POLYGON ((88.35105 22.48683, 88.35108 22.48684… -0.940273
139    30  POLYGON ((88.38150 22.57798, 88.38152 22.57799… -0.930493
140    58  POLYGON ((88.38339 22.55975, 88.38346 22.55972… -0.989160

[141 rows x 3 columns]
```

[114]:

```
# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values
vmin, vmax = merged_Kolkata["mean"].min(), merged_Kolkata["mean"].max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))
# remove the axis
ax.axis('off')
# add a title and annotation
ax.set_title('Kolkata- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))
# empty array for the data range
sm.set_array([])
# add the colorbar to the figure
fig.colorbar(sm)
# create map
merged_Kolkata.plot(column=variable, cmap='coolwarm', linewidth=0.8, ax=ax,␣
 ↪edgecolor='0.5', figsize = (5,15))
plt.show()
```



[115]:
```
# Using T-Test to determine if climate has changed in the first 14 and last 14␣
↪months in Kolkata
ttest_ind(Monthly_Kolkata_mean.iloc[:14,:]["mean"].
↪tolist(),Monthly_Kolkata_mean.iloc[14:,:]["mean"].tolist())
```

[115]:
```
Ttest_indResult(statistic=5.244770969255621, pvalue=1.7633976410214782e-05)
```

We observe that Pollution levels has decreased in the last 14 months and the statistical test proves the same with a very small p-value for 95 % Confidence Interval

[ ]:
```

```

4 Visualising for India

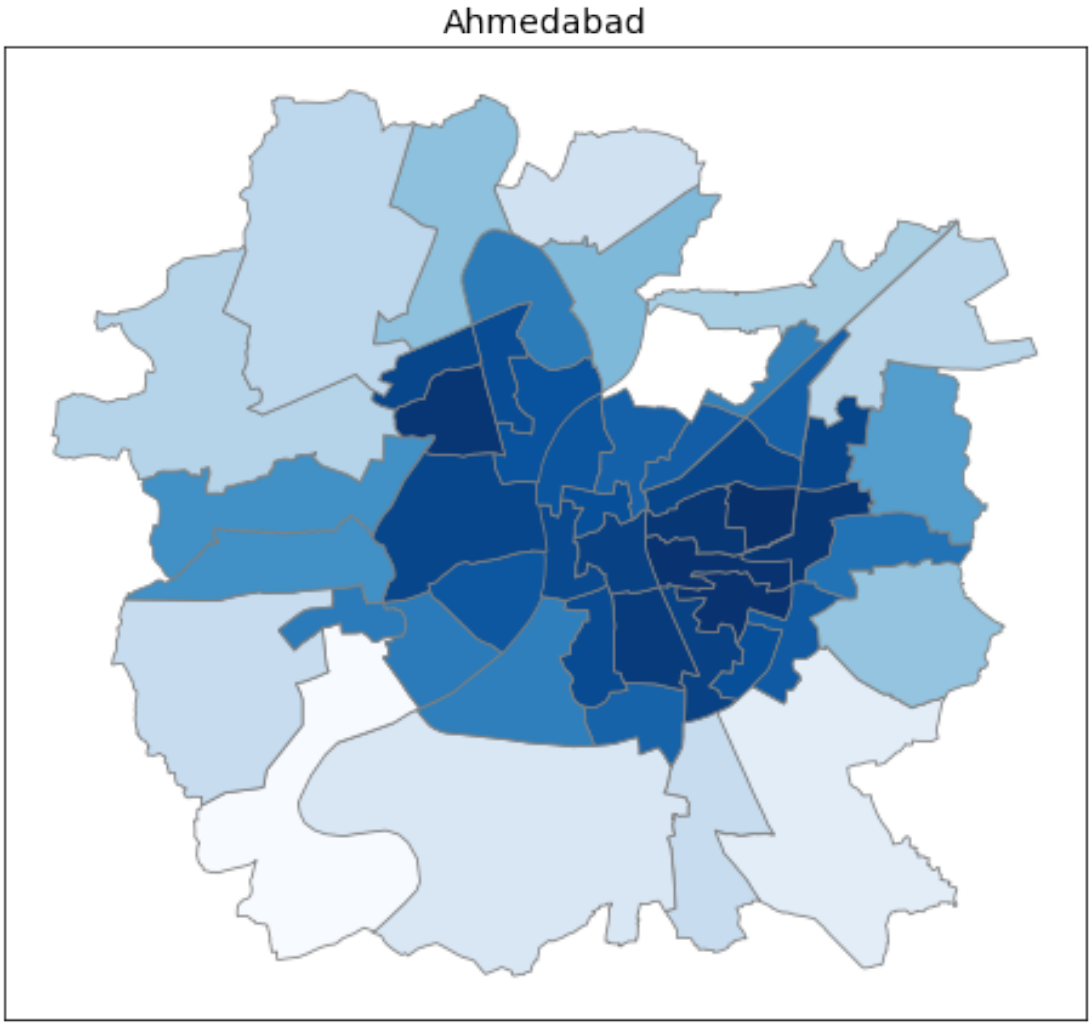
4.1 Loading and Visualizing

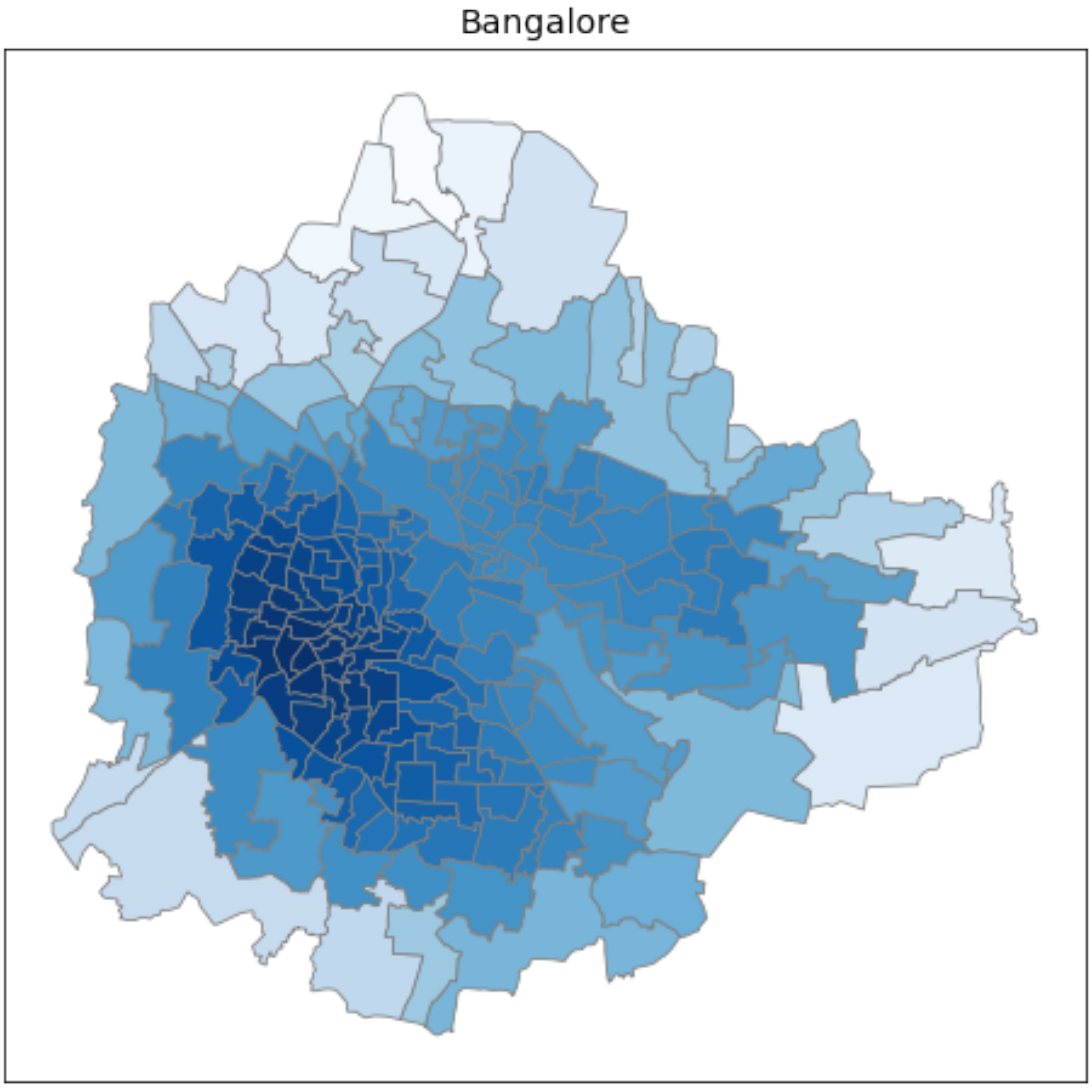
[116]:
```
%matplotlib inline

#Plotting all Maps with mean values

amd= merged_Ahmedabad.plot(column=variable, linewidth=0.8, edgecolor='0.5',␣
↪cmap = "Blues", figsize= (8,8))
```

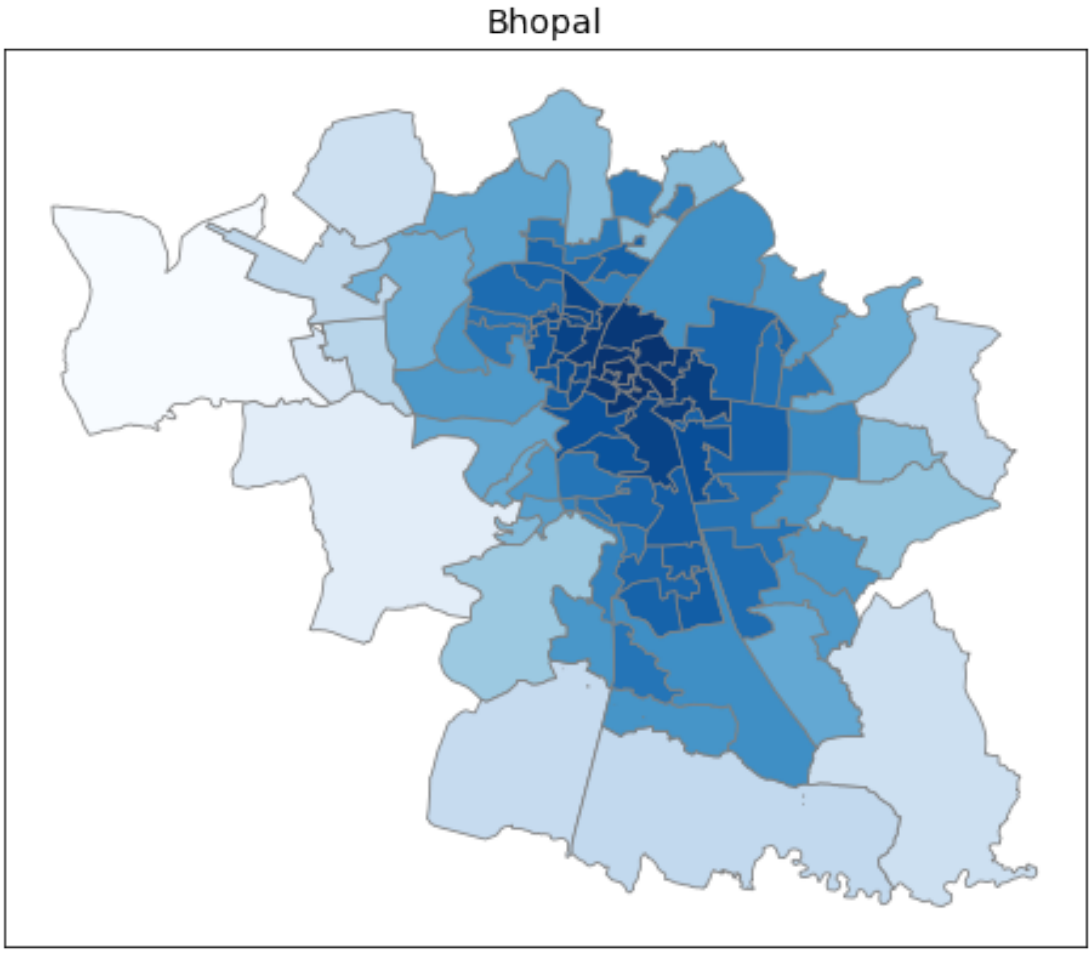
```
amd.xaxis.set_ticks([])
amd.yaxis.set_ticks([])
amd.set_title("Ahmedabad", fontsize = 14)

bglr=merged_Bangalore.plot(column=variable, linewidth=0.8, edgecolor='0.5',␣
↪cmap = "Blues", figsize= (8,8))
bglr.xaxis.set_ticks([])
bglr.yaxis.set_ticks([])
bglr.set_title("Bangalore", fontsize = 14)

bpl=merged_Bhopal.plot(column=variable, linewidth=0.8, edgecolor='0.5', cmap =␣
↪"Blues", figsize= (8,8))
bpl.xaxis.set_ticks([])
bpl.yaxis.set_ticks([])
bpl.set_title("Bhopal", fontsize = 14)

bbr=merged_Bhubaneshwar.plot(column=variable, linewidth=0.8, edgecolor='0.5',␣
↪cmap = "Blues", figsize= (8,8))
bbr.xaxis.set_ticks([])
bbr.yaxis.set_ticks([])
bbr.set_title("Bhubaneshwar", fontsize = 14)

csk=merged_Chennai.plot(column=variable, linewidth=0.8, edgecolor='0.5', cmap =␣
↪"Blues", figsize= (8,8))
csk.xaxis.set_ticks([])
csk.yaxis.set_ticks([])
csk.set_title("Chennai", fontsize = 14)

dc=merged_Delhi.plot(column=variable, linewidth=0.8, edgecolor='0.5', cmap =␣
↪"Blues", figsize= (8,8))
dc.xaxis.set_ticks([])
dc.yaxis.set_ticks([])
dc.set_title("Delhi", fontsize = 14)

srh=merged_Hyderabad.plot(column=variable, linewidth=0.8, edgecolor='0.5', cmap␣
↪= "Blues", figsize= (8,8))
srh.xaxis.set_ticks([])
srh.yaxis.set_ticks([])
srh.set_title("Hyderabad", fontsize = 14)

rr=merged_Jaipur.plot(column=variable, linewidth=0.8, edgecolor='0.5', cmap =␣
↪"Blues", figsize= (8,8))
rr.xaxis.set_ticks([])
rr.yaxis.set_ticks([])
rr.set_title("Jaipur", fontsize = 14)
```

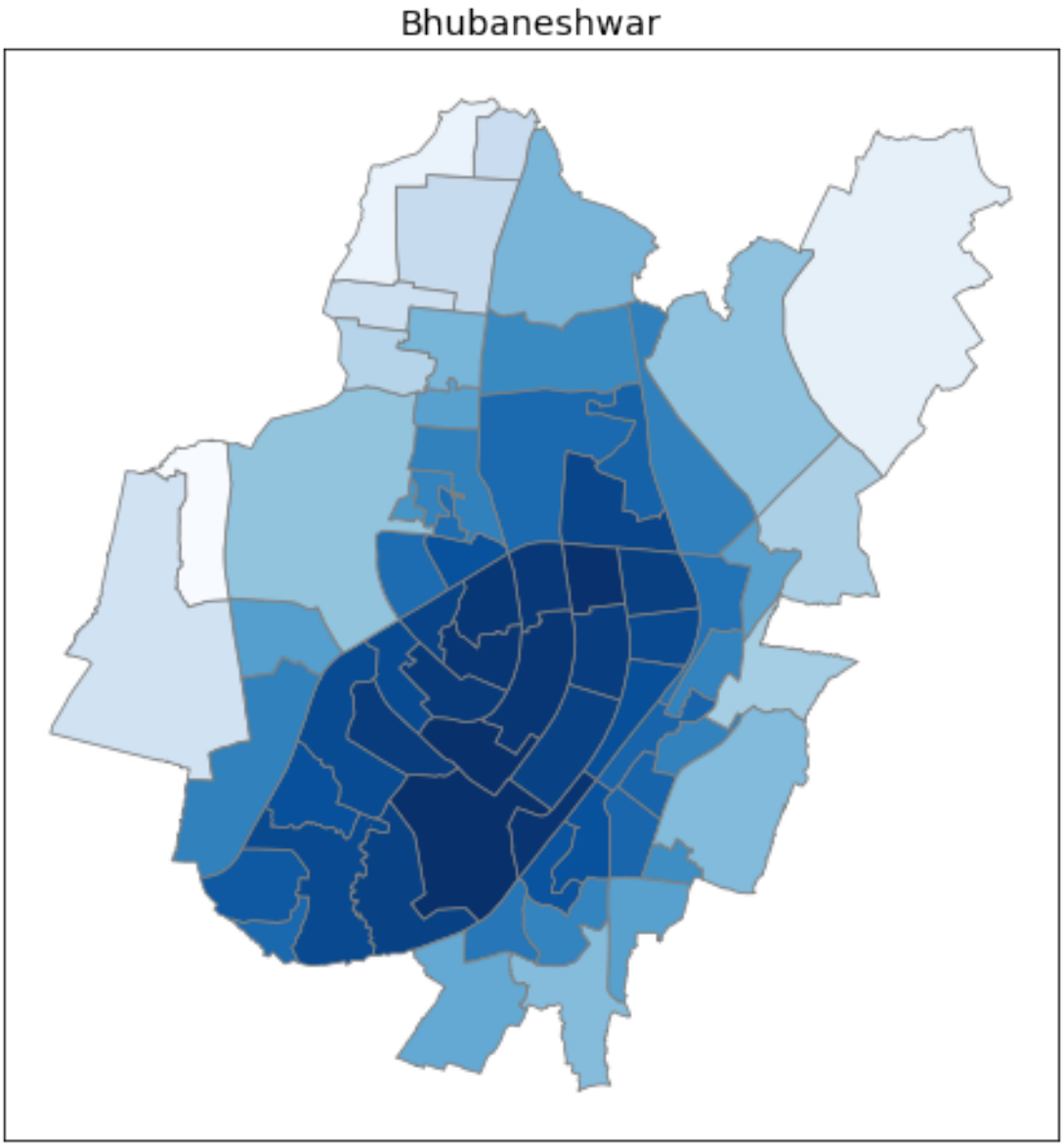
```
kkr=merged_Kolkata.plot(column=variable, linewidth=0.8, edgecolor='0.5', cmap =␣
 ↪"Blues", figsize= (8,8))
kkr.xaxis.set_ticks([])
kkr.yaxis.set_ticks([])
kkr.set_title("Kolkata", fontsize = 14)

MI=merged_Mumbai.plot(column=variable, linewidth=0.8, edgecolor='0.5', cmap =␣
 ↪"Blues", figsize= (8,8))
MI.xaxis.set_ticks([])
MI.yaxis.set_ticks([])
MI.set_title("Mumbai", fontsize = 14)

RPS=merged_Pune.plot(column=variable, linewidth=0.8, edgecolor='0.5', cmap =␣
 ↪"Blues", figsize= (8,8))
RPS.xaxis.set_ticks([])
RPS.yaxis.set_ticks([])
RPS.set_title("Pune", fontsize = 14)
```

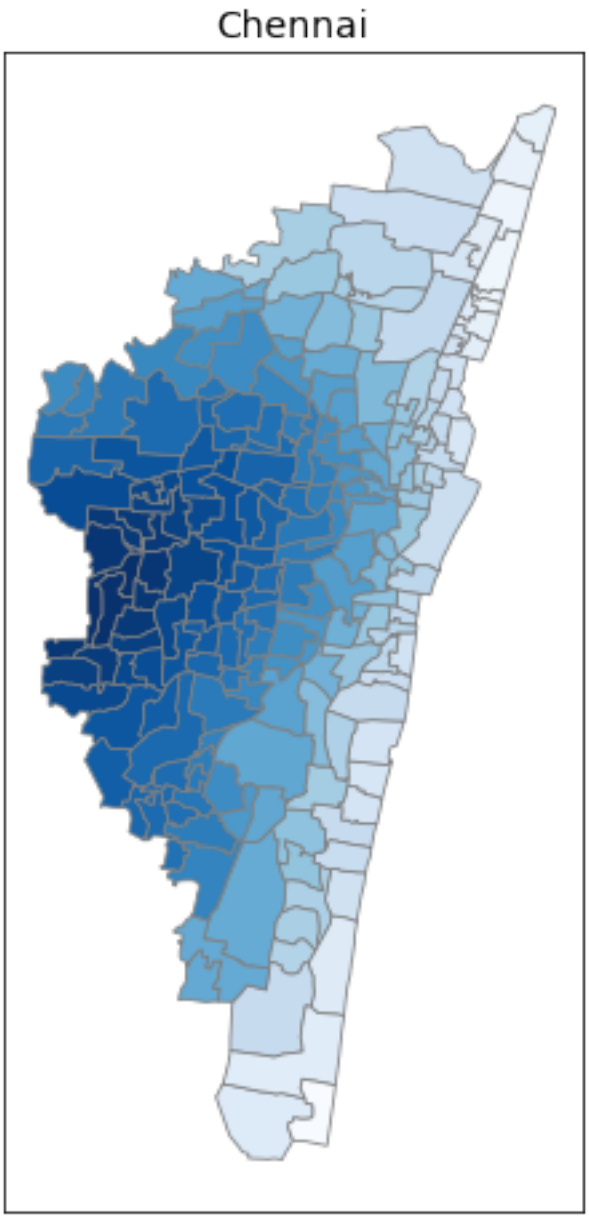
[116]: Text(0.5, 1.0, 'Pune')

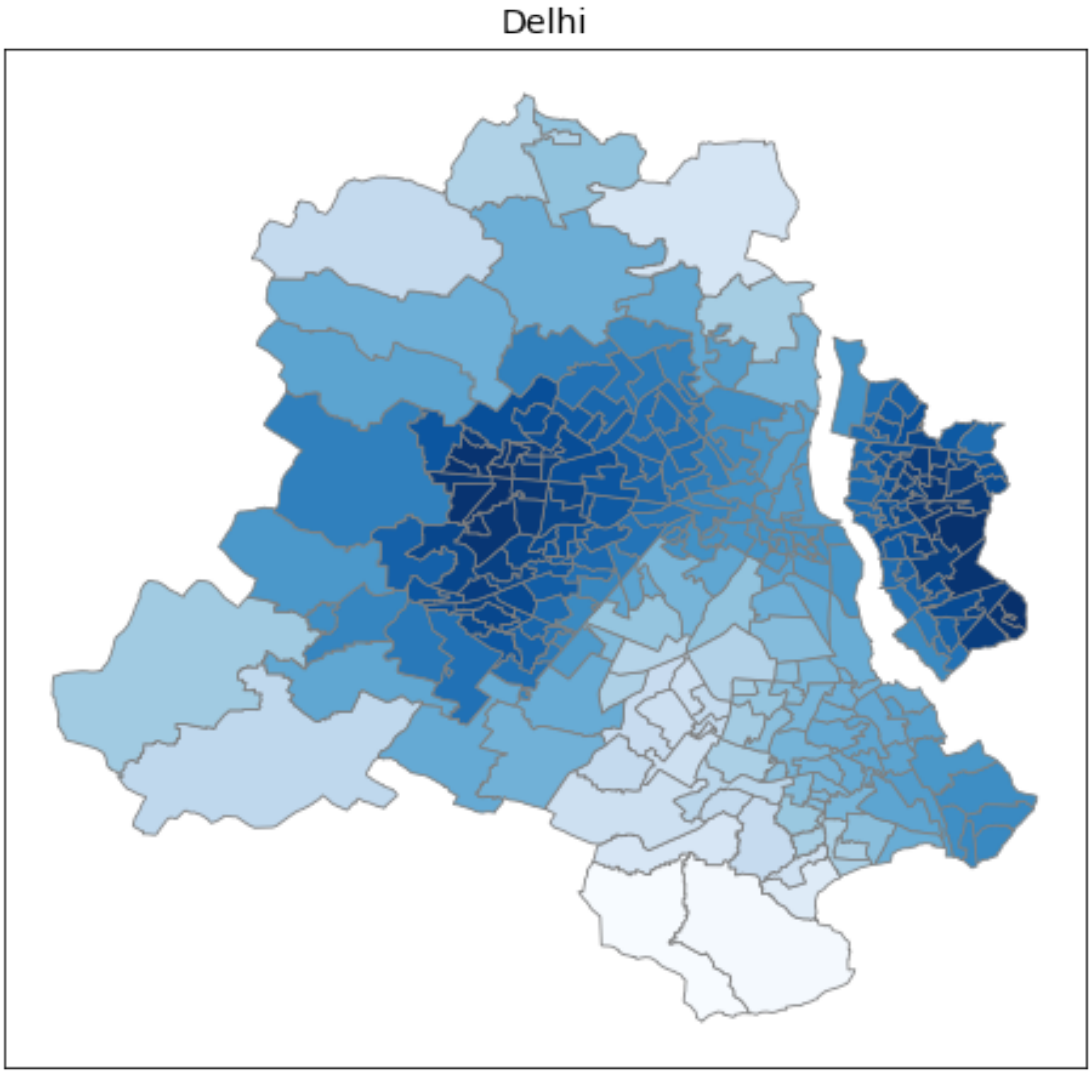

Ahmedabad

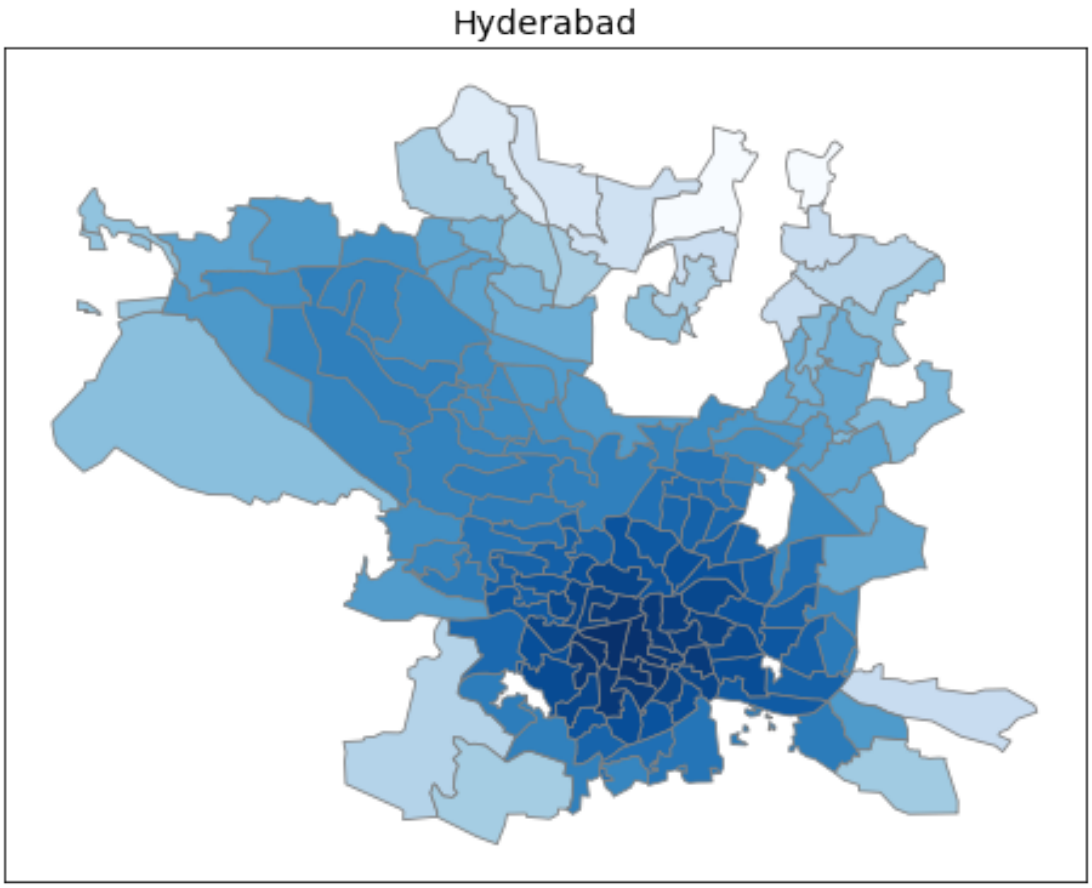

Bangalore

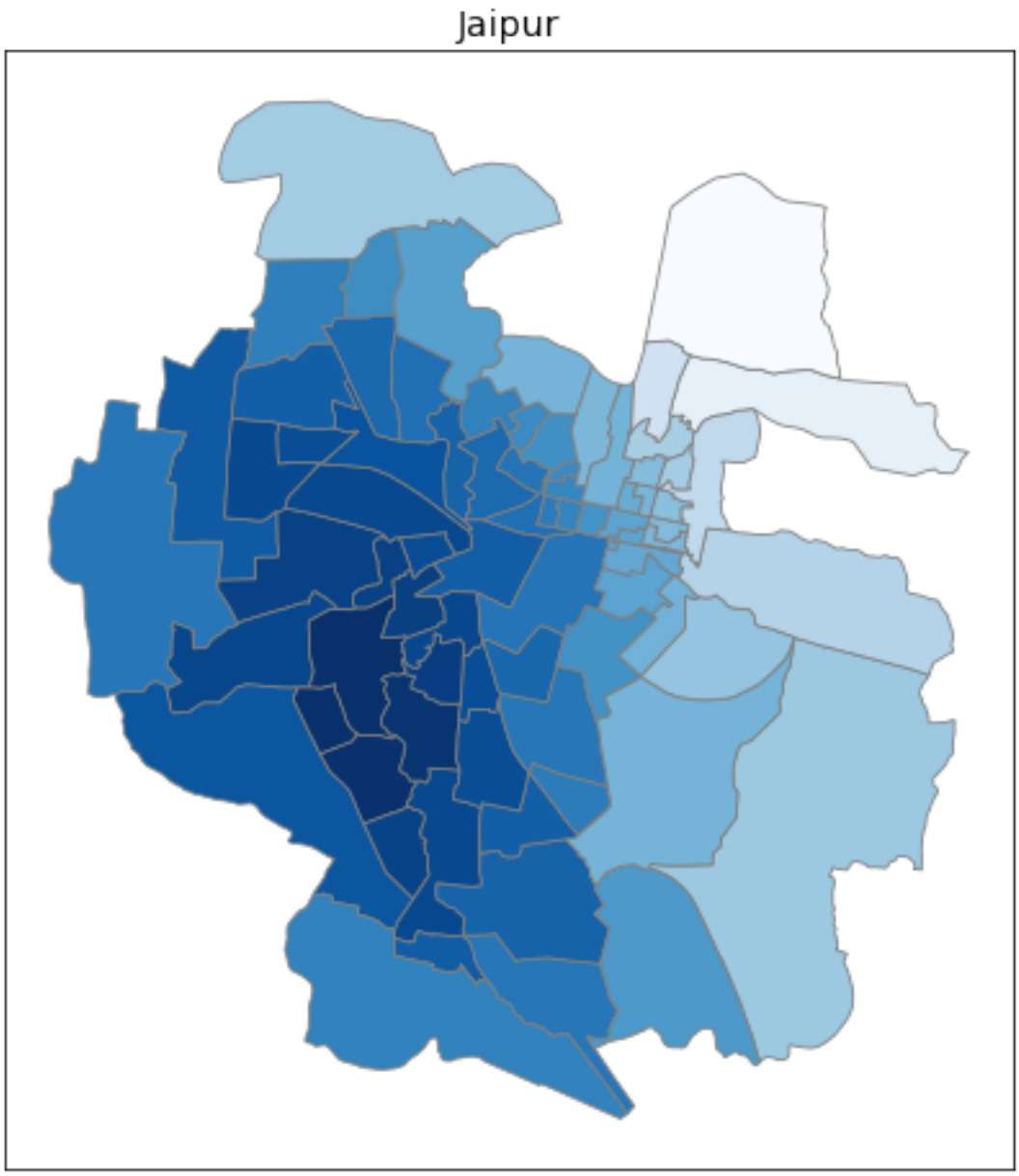

Bhopal

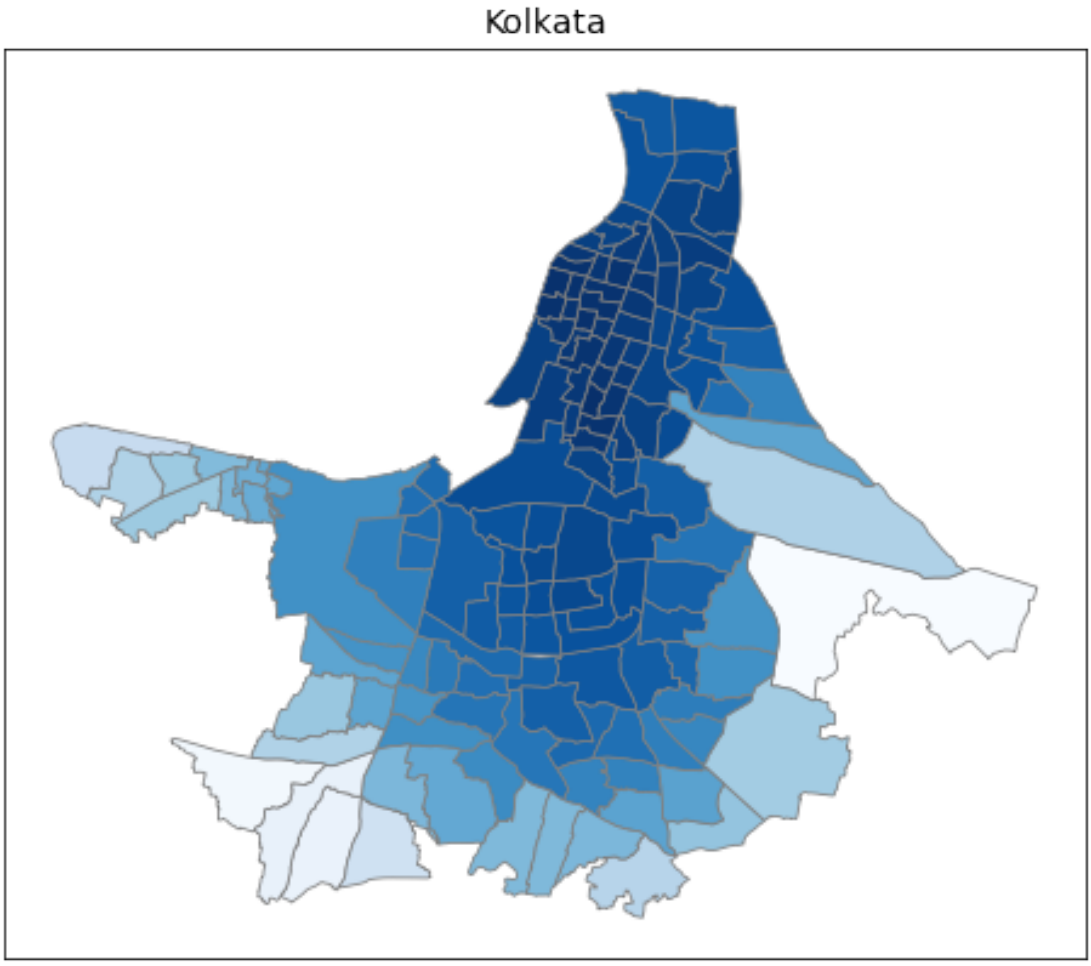

Bhubaneshwar

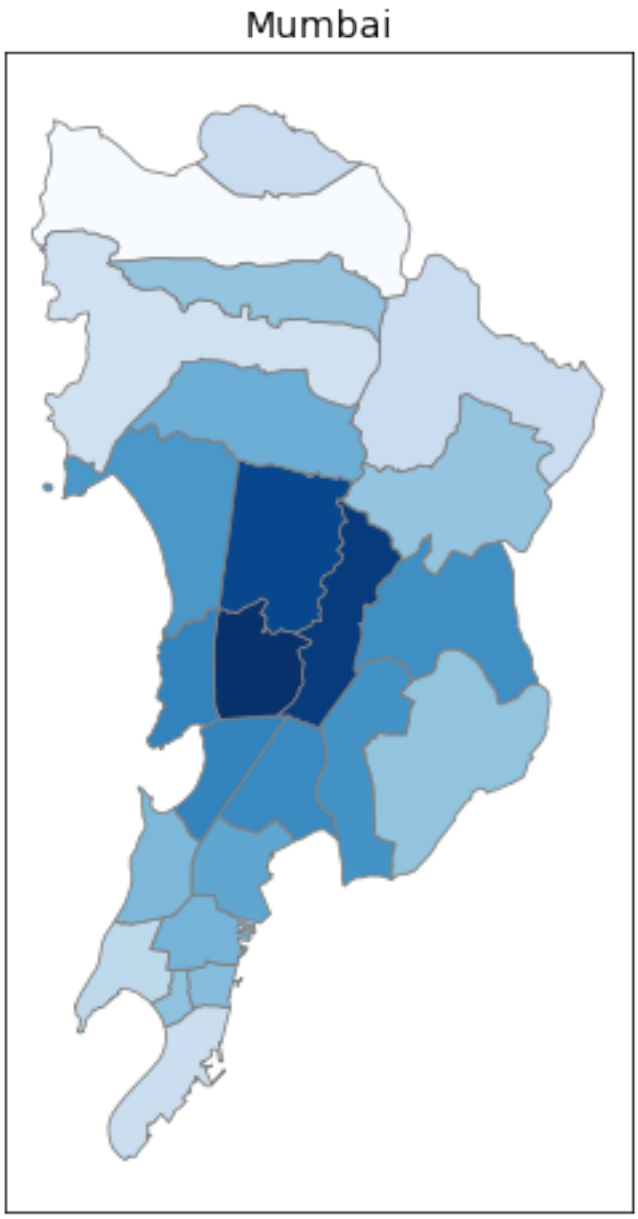

Chennai

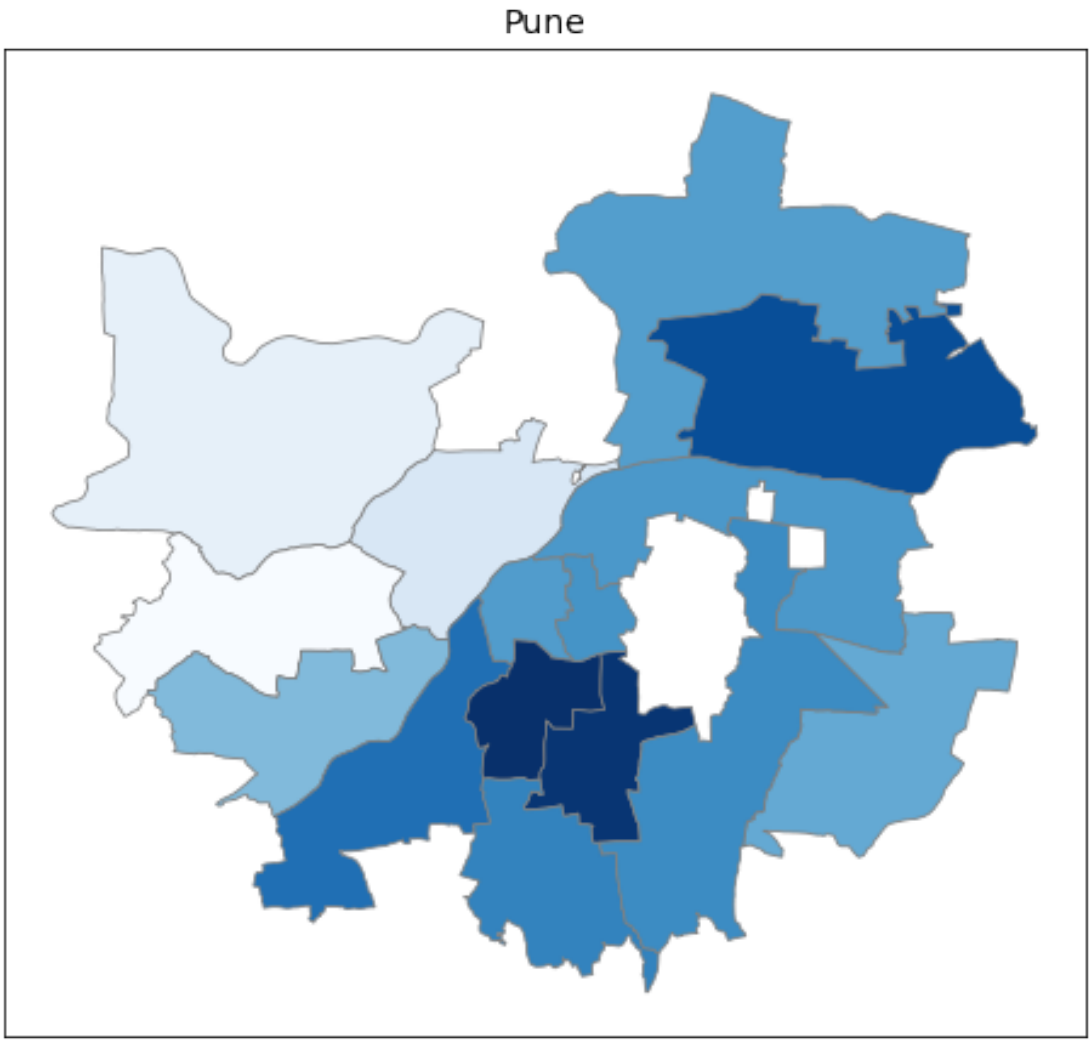

Delhi


Hyderabad


Jaipur


Kolkata


Mumbai



[117]:
```
# Loading India Boundary Shapefile

path = r'Cleaned CSVs/India/India_Boundary.shp'
map_India = gpd.read_file(path)
# map_India("Hyderabad", (map_Hyderabad["geometry"].geometry))
# map_India
map_India
```

[117]:
```
                                source  \
0  Survey of India State Map, Datameet

                                                 geometry
0  MULTIPOLYGON (((84.76858 19.08002, 84.76625 19…
```

[118]:
```
# Viewing India Map
map_India.plot()
```

[118]: <AxesSubplot:>

[119]:
```
net_mean = []
net_avg= 0.0
count = 1
for i in cities1:
    net_mean.append((count,i,df[i]["mean"].mean()))
    net_avg += df[i]["mean"].mean()
    count = count +1
net_mean.append((0,"India",-2))
df_boundaries = pd.DataFrame(columns = ["index","Name", "mean"])
# Cleaned Dataframe
df_boundaries = pd.DataFrame(net_mean, columns = ['index','Name', 'mean'])
df_boundaries.set_index("index",inplace = True)
df_boundaries # Boundaries and means of all Locations to be plotted
```

[119]:

index	Name	mean
1	Ahmedabad	-0.583594
2	Bangalore	-1.033135
3	Chennai	-1.103986
4	Hyderabad	-0.947310
5	Jaipur	-0.685767

```
6              Pune -0.985005
7            Mumbai -0.934412
8            Bhopal -0.956933
9      Bhubaneshwar -1.094891
10          Kolkata -0.942615
11            Delhi -0.660274
0             India -2.000000
```

[120]:
```
cities2 = ["ahmedabad", "bangalore" , "chennai", "hyderabad", "jaipur" ,␣
↪"pune", "mumbai" ,"bhopal" ,"bhubaneshwar" , "kolkata","delhi"]
boundary_shapefiles = {}
geo = gpd.GeoSeries(1)
# print(geo)
for i in cities2:
    path1 = r'Cleaned CSVs/Boundary files/'+ i + '.geojson'
    try:
        boundary_shapefiles[i] = gpd.read_file(path1)
        # boundary_shapefiles[i].plot()

    except:
        path2 = r'Cleaned CSVs/Boundary files/'+ i + '.shp'
        try:
            boundary_shapefiles[i] = gpd.read_file(path2)
            # boundary_shapefiles[i].plot()

        except:
            continue
    geo = geo.append(boundary_shapefiles[i]["geometry"])
geo.iloc[0]= map_India.iloc[0]["geometry"]
GeoDF = gpd.GeoDataFrame(geo)
GeoDF.columns = ["geometry"]
GeoDF["i"] = [i for i in range(0,12)]
GeoDF.set_index("i",inplace= True)
GeoDF
```

[120]:

i	geometry
0	(POLYGON ((84.7685819052308 19.08002272866605,…
1	POLYGON ((72.4451995 23.0420819, 72.4460981999…
2	POLYGON ((77.46210670000001 12.8975897, 77.460…
3	(POLYGON ((80.1403998 13.0978875, 80.1401875 1…
4	(POLYGON ((78.2807423 17.4845165, 78.2807852 1…
5	POLYGON ((75.78598737700008 27.00526424200007,…
6	POLYGON ((73.772335 18.4888891, 73.7749958 18…
7	(POLYGON ((72.7802918 19.217928, 72.7799814 19…
8	POLYGON ((77.39355784991608 23.42155154218633,…
9	POLYGON ((85.86773970122816 20.28862111513216,…

```
10  POLYGON ((88.2406237 22.5535318, 88.2417083999…
11  POLYGON ((77.33088732800013 28.63171820200012,…
```

[121]:
```
# Merges Means with the shapefile DF
merged_India = GeoDF.merge(df_boundaries,left_index = True, right_index = True)
merged_India["centroid"] =merged_India.geometry.centroid
meansss = [str("{:.4f}".format(round(i, 2))) for i in (merged_India["mean"])]
merged_India["data"] = (merged_India["Name"])+ " " + meansss
merged_India
```

[121]:

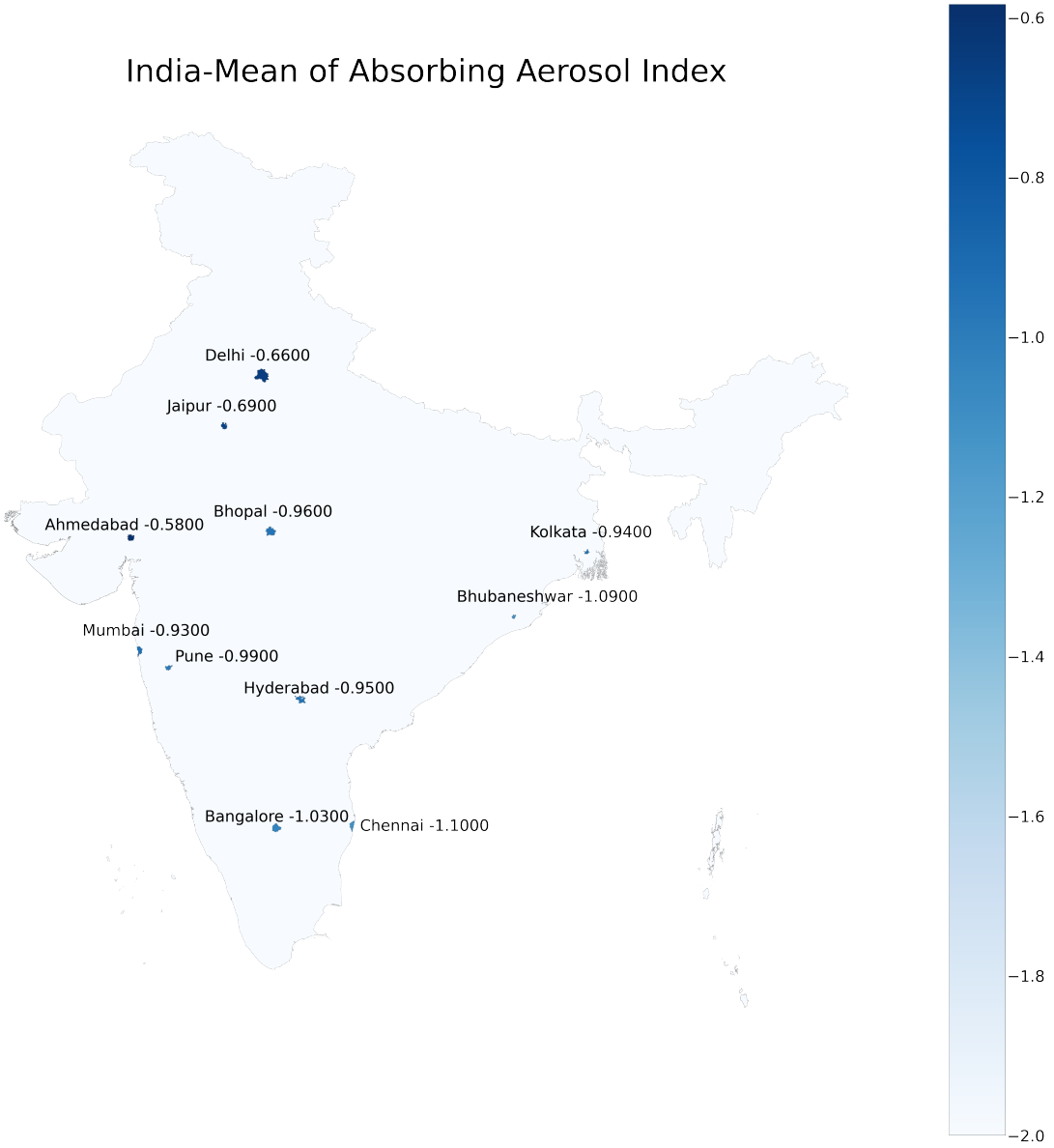
	geometry	Name	mean	centroid	data
0	MULTIPOLYGON (((84.76858 19.08002, 84.76625 19…	India	-2.000000	POINT (79.47926 23.37824)	India -2.0000
1	POLYGON ((72.44520 23.04208, 72.44610 23.04985…	Ahmedabad	-0.583594	POINT (72.57299 23.02695)	Ahmedabad -0.5800
2	POLYGON ((77.46211 12.89759, 77.46022 12.90045…	Bangalore	-1.033135	POINT (77.60356 12.97879)	Bangalore -1.0300
3	MULTIPOLYGON (((80.14040 13.09789, 80.14019 13…	Chennai	-1.103986	POINT (80.23080 13.06179)	Chennai -1.1000
4	MULTIPOLYGON (((78.28074 17.48452, 78.28079 17…	Hyderabad	-0.947310	POINT (78.45471 17.42677)	Hyderabad -0.9500
5	POLYGON ((75.78599 27.00526, 75.78696 27.00391…	Jaipur	-0.685767	POINT (75.80230 26.89589)	Jaipur -0.6900
6	POLYGON ((73.77233 18.48889, 73.77500 18.48970…	Pune	-0.985005	POINT (73.87636 18.52835)	Pune -0.9900
7	MULTIPOLYGON (((72.78029 19.21793, 72.77998 19…	Mumbai	-0.934412	POINT (72.87062 19.11833)	Mumbai -0.9300
8	POLYGON ((77.39356 23.42155, 77.39317 23.42189…	Bhopal	-0.956933	POINT (77.41044 23.24109)	Bhopal -0.9600
9	POLYGON ((85.86774 20.28862, 85.86776 20.28860…	Bhubaneshwar	-1.094891	POINT (85.82753 20.29198)	Bhubaneshwar -1.0900
10	POLYGON ((88.24062 22.55353, 88.24171 22.55423…	Kolkata	-0.942615	POINT (88.35547 22.52947)	Kolkata -0.9400
11	POLYGON ((77.33089 28.63172, 77.33225 28.62995…	Delhi	-0.660274	POINT (77.11586 28.64309)	Delhi -0.6600

4.2 Plotting On INDIA MAP

[122]:
```
# Viewing our variable on the India Map
# Chloropleth of India with The cities and mean of absorbing aerosol index␣
 ↪marked

%matplotlib inline
# set the value column that will be visualised
variable = 'mean'
```

```
# # set the range for the choropleth values
vmin, vmax = merged_India["mean"].min() , merged_India["mean"].max()
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(100, 100))
# remove the axis
ax.axis('off')
# add a title and annotation
ax.set_title('India-Mean of Absorbing Aerosol Index', fontdict={'fontsize':␣
↪'150', 'fontweight' : '3'})
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='Blues',norm=plt.Normalize(vmin=vmin,␣
↪vmax=vmax))
# empty array for the data range
sm.set_array([])
# add the colorbar to the figure
bx = fig.colorbar(sm)
bx.ax.tick_params(labelsize=75)

# create map
ax = merged_India.plot(column=variable, cmap='Blues', linewidth=0.8, ax=ax,␣
↪edgecolor='0.5', figsize = (150,450), label = merged_India.data)
 # ax = cities.plot()
counter = 0
for x, y, label in zip(merged_India.centroid.x, merged_India.centroid.y,␣
↪merged_India.data ):
    if(label == "India -2.0000"):
        continue
    if(label == "Ahmedabad -0.5800"):
            ax.annotate(label, xy=(x-3, y+0.25), xytext=(3, 3),␣
↪textcoords="offset points",fontsize = 75)
            continue
    if(label == "Pune -0.9900"):
            ax.annotate(label, xy=(x+0.2, y+0.2), xytext=(3, 3),␣
↪textcoords="offset points",fontsize = 75)
            continue
    if(label == "Hyderabad -0.9500"):
            ax.annotate(label, xy=(x-2, y+0.2), xytext=(3, 3),␣
↪textcoords="offset points",fontsize = 75)
            continue
    if(label == "Chennai -1.1000"):
            ax.annotate(label, xy=(x+0.25, y-0.2), xytext=(3, 3),␣
↪textcoords="offset points",fontsize = 75)
            continue
    if(label == "Bangalore -1.0300"):
            ax.annotate(label, xy=(x-2.5, y+0.2), xytext=(3, 3),␣
↪textcoords="offset points",fontsize = 75)
```

```
            continue
        ax.annotate(label, xy=(x-2, y+0.50), xytext=(3, 3), textcoords="offset␣
    ↪points",fontsize = 75)
# plt.legend()
plt.show()
```

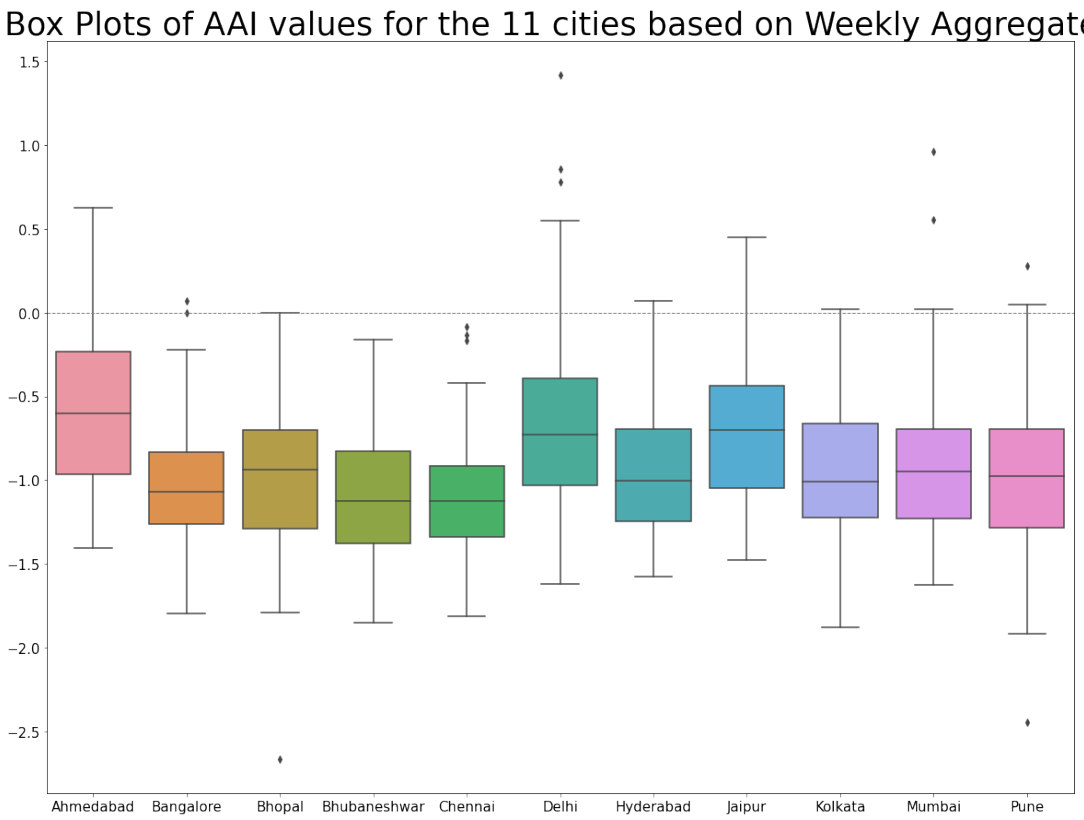


4.3 Plots For all Regions

[123]:
```
weekly_data_mean = pd.read_csv("Cleaned CSVs/WEEK_DATA/weekly_collab.csv")
weekly_data_mean.date = pd.to_datetime(weekly_data_mean.date)
weekly_data_mean.set_index("date",inplace = True)
weekly_data_mean
```

[123]:
```
            Ahmedabad  Bangalore    Bhopal  Bhubaneshwar   Chennai     Delhi  \
date
2018-07-16  -0.756585  -0.896302 -0.557584     -0.642007 -0.540671 -0.453550
2018-07-23  -0.441123  -0.222558 -0.490329     -0.647062 -0.660151 -0.884397
2018-07-30   0.108523  -0.378929 -0.178767     -0.851047 -0.082220 -0.577875
2018-08-06   0.401942  -0.000837 -0.329908     -0.412066 -0.491096 -0.103947
2018-08-13   0.590059  -0.537708 -0.120354     -1.297045 -0.133111 -0.402681
…               …         …         …             …         …       …
2020-09-21  -1.404141  -1.462358 -1.733017     -1.411822 -1.443073 -1.425032
2020-09-28  -1.152335  -1.380333 -1.788614     -1.847646 -1.673202 -1.010724
2020-10-05  -1.105371  -1.433105 -1.151164     -1.363438 -1.371587 -0.925760
2020-10-12  -0.897889  -1.427411 -1.443555     -1.054022 -1.463882 -0.763071
2020-10-19  -1.124830  -1.793873 -2.663199     -1.410126 -1.810164 -0.727903

            Hyderabad    Jaipur   Kolkata    Mumbai      Pune
date
2018-07-16  -0.600131 -0.565500 -1.104306 -0.491262 -0.896013
2018-07-23  -0.770834 -0.692978 -0.692227 -0.558497 -0.480637
2018-07-30   0.070294 -0.264672 -0.557875  0.009168 -0.417762
2018-08-06  -0.138146 -0.172926 -0.539648  0.958713  0.278316
2018-08-13  -0.275480 -0.309478 -1.537283  0.022972 -0.339340
…               …         …         …         …         …
2020-09-21  -1.398443 -1.314791 -1.422304 -1.392733 -1.625738
2020-09-28  -1.275906 -1.347430 -1.335335 -1.535179 -1.380134
2020-10-05  -1.268969 -0.912607 -1.875154 -1.441283 -1.625510
2020-10-12  -1.410603 -0.875882 -1.346682 -1.272204 -1.574471
2020-10-19  -1.476454 -0.966210 -1.394773 -1.162175 -2.441827

[119 rows x 11 columns]
```

[124]:
```
plt.figure(figsize = (20,15))
sns.boxplot(data= weekly_data_mean)
plt.axhline(0.0, color='gray', linestyle='dashed',
            linewidth=1, label = "Mean")
plt.title("Box Plots of AAI values for the 11 cities based on Weekly␣
 ↪Aggregates",fontsize = 35)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.show()
```



The box plots shows that some cities like Ahmedabad, Delhi, Jaipur have the Absorbing Aerosol Index values above 0 which is a sign of hazardous levels of pollution in these cities. In cities like Delhi, there are a few out liers which signifies that there a some weeks where there the smoke and dust level are very high and vidibility is quite low, justified by high AAI values. Mumbai has comparitively lesser weeks like this and it is less extreme.

[125]:
```
# Summary Statistics Weekly
weekly_data_mean.describe()
```

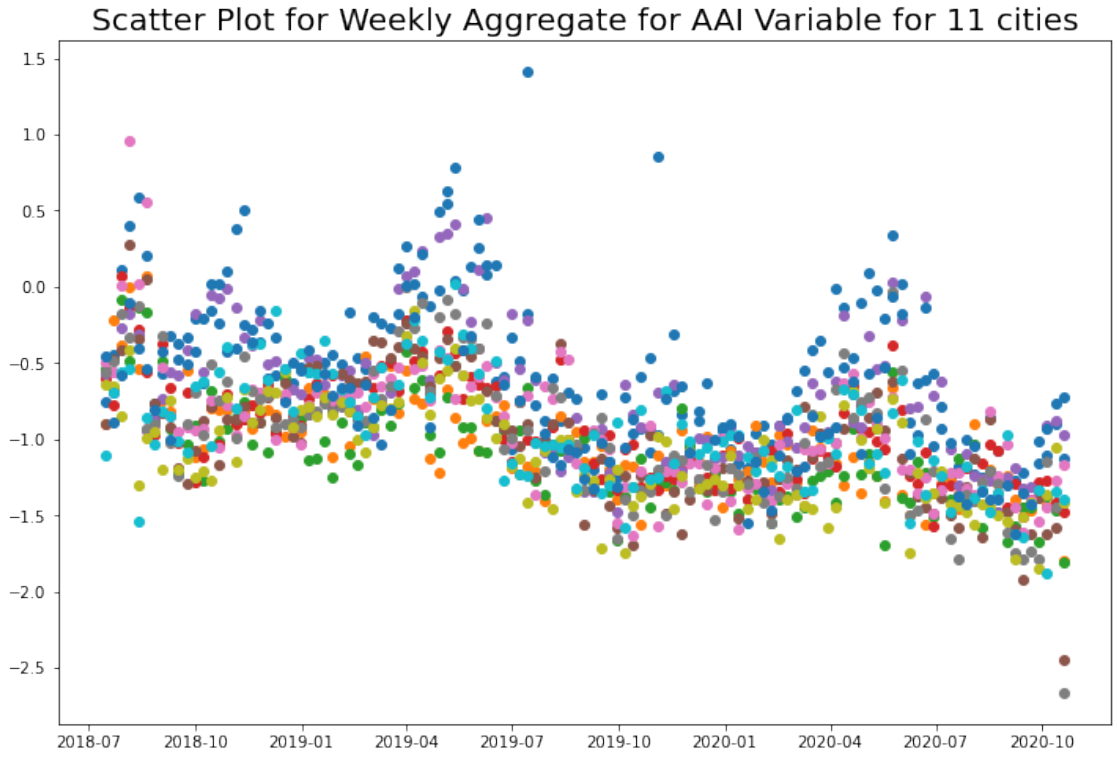
[125]:

	Ahmedabad	Bangalore	Bhopal	Bhubaneshwar	Chennai	\
count	119.000000	119.000000	119.000000	119.000000	119.000000	
mean	-0.582159	-1.034328	-0.964281	-1.095213	-1.100871	
std	0.467442	0.313817	0.457652	0.343291	0.332805	
min	-1.404141	-1.793873	-2.663199	-1.847646	-1.810164	
25%	-0.965512	-1.259664	-1.286647	-1.376484	-1.339684	
50%	-0.599787	-1.068108	-0.939337	-1.123672	-1.123548	
75%	-0.234328	-0.832001	-0.698949	-0.828347	-0.913961	
max	0.628108	0.072607	-0.003475	-0.160723	-0.082220	

	Delhi	Hyderabad	Jaipur	Kolkata	Mumbai	Pune
count	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000
mean	-0.654578	-0.948809	-0.683146	-0.945886	-0.932589	-0.993879

std	0.547450	0.330388	0.448705	0.365945	0.408459	0.430942
min	-1.620755	-1.573841	-1.476412	-1.875154	-1.626773	-2.441827
25%	-1.028708	-1.242476	-1.046962	-1.223273	-1.229079	-1.285855
50%	-0.727903	-1.002705	-0.697938	-1.009371	-0.950023	-0.975512
75%	-0.389772	-0.696824	-0.438290	-0.664147	-0.697365	-0.696135
max	1.415602	0.070294	0.451761	0.018691	0.958713	0.278316

[126]:
```
# sns.boxplot(data= weekly_data_mean)
plt.figure(figsize = (12,8))
for i in cities1:
    plt.scatter(x = weekly_data_mean.index, y = weekly_data_mean[i],label = i)
# plt.legend()
plt.title("Scatter Plot for Weekly Aggregate for AAI Variable for 11 cities",␣
 ↪fontsize= 20)
plt.setp(plt.gca().get_xticklabels(), fontsize=10)
plt.show()
```



The scatter plot of weekly aggregate shows a seasonal trend wehere there is are extreme values of AAI in summers and comparitively lower values in winter.

[127]:
```
monthly_data_mean = pd.read_csv("Cleaned CSVs/MONTH_DATA/monthly_collab.csv")
```

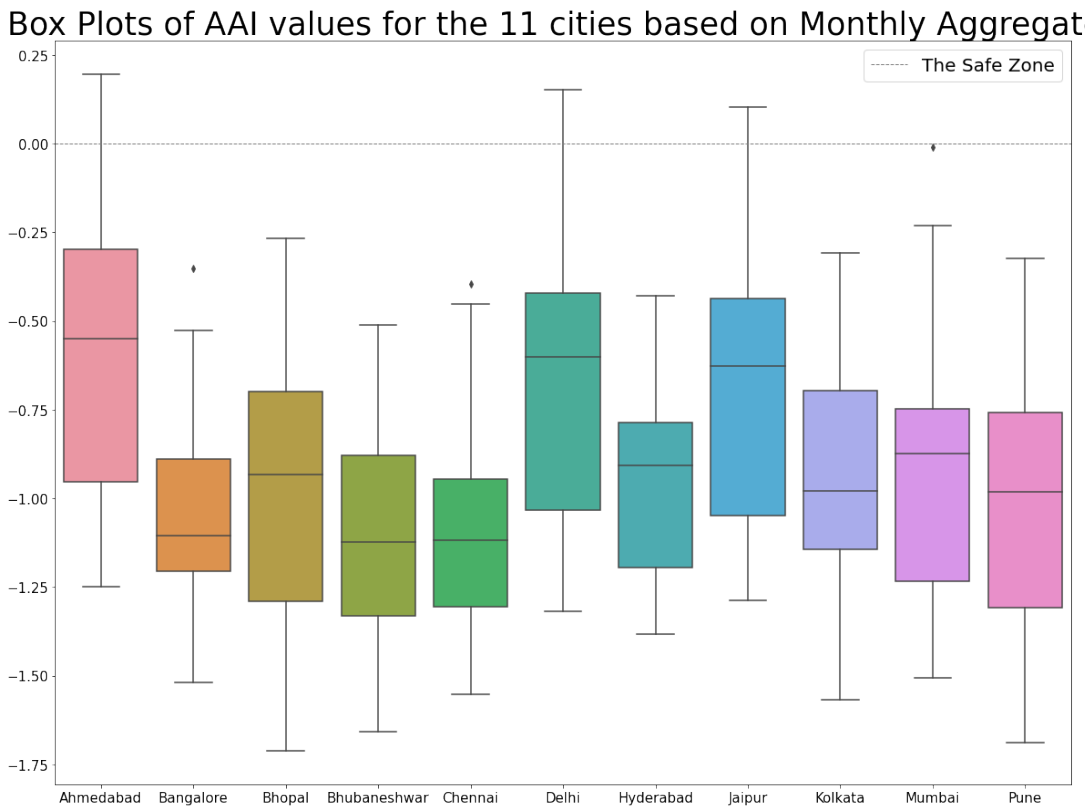
[128]:
```
# Summary Statistics Monthly
monthly_data_mean.describe()
```

[128]:

	Ahmedabad	Bangalore	Bhopal	Bhubaneshwar	Chennai	Delhi
count	28.000000	28.000000	28.000000	28.000000	28.000000	28.000000
mean	-0.586662	-1.035066	-0.961516	-1.089359	-1.103280	-0.662504
std	0.418977	0.271389	0.389578	0.292168	0.288606	0.429254
min	-1.250079	-1.518255	-1.711059	-1.658433	-1.552375	-1.319630
25%	-0.952383	-1.205974	-1.291184	-1.330089	-1.305032	-1.032945
50%	-0.549166	-1.106315	-0.932946	-1.122891	-1.118413	-0.602278
75%	-0.297980	-0.889730	-0.700275	-0.878198	-0.946842	-0.421228
max	0.194117	-0.352060	-0.266061	-0.511579	-0.394543	0.152676

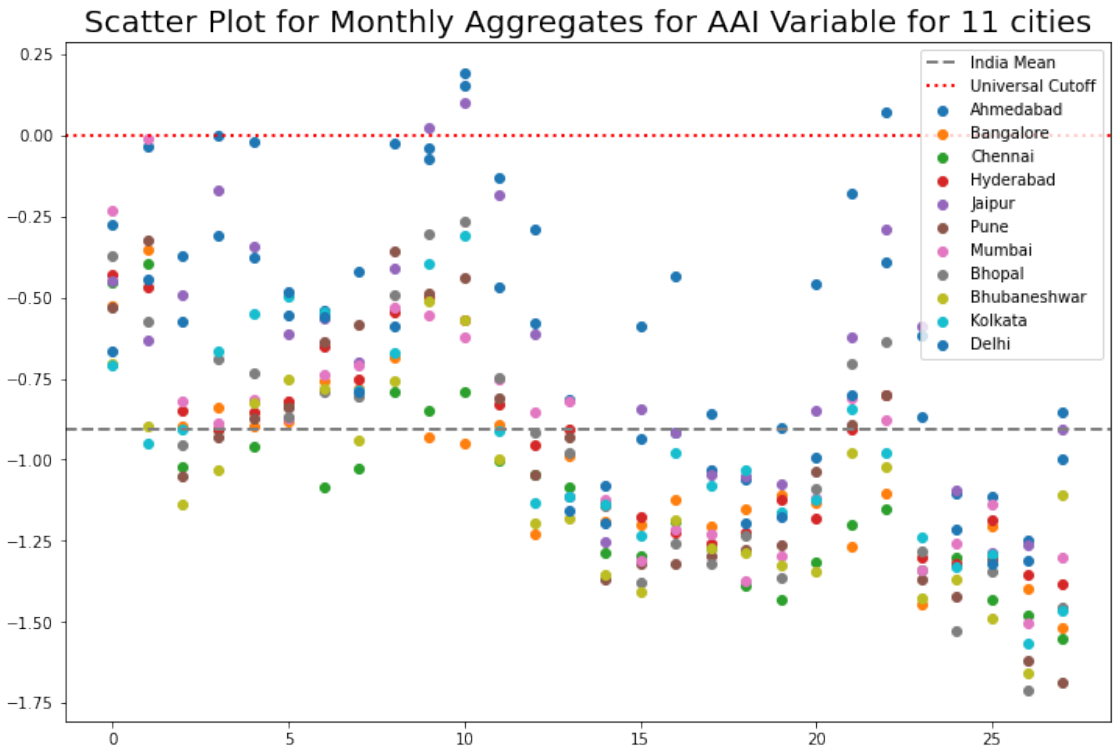
	Hyderabad	Jaipur	Kolkata	Mumbai	Pune
count	28.000000	28.000000	28.000000	28.000000	28.000000
mean	-0.949568	-0.686848	-0.950377	-0.932042	-0.993262
std	0.292071	0.387927	0.320910	0.356192	0.382903
min	-1.382430	-1.286629	-1.567887	-1.506054	-1.687852
25%	-1.195781	-1.047640	-1.144023	-1.234692	-1.308180
50%	-0.906686	-0.626147	-0.978432	-0.874155	-0.982953
75%	-0.787334	-0.437523	-0.697642	-0.749008	-0.757161
max	-0.429574	0.103095	-0.307695	-0.010012	-0.322667

[129]:
```
plt.figure(figsize = (20,15))
sns.boxplot(data= monthly_data_mean)
plt.axhline(0.0, color='gray', linestyle='dashed',
            linewidth=1, label = "The Safe Zone")
plt.title("Box Plots of AAI values for the 11 cities based on Monthly␣
 ↪Aggregates",fontsize = 35)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.legend(fontsize = 20)
plt.show()
```



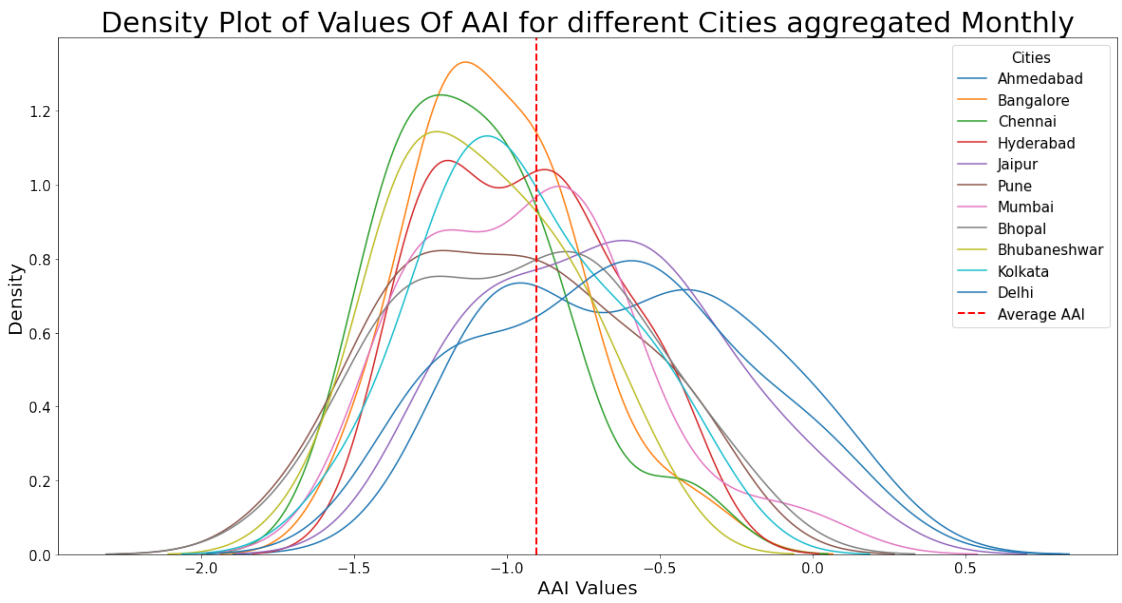
The box plots shows that some cities like Ahmedabad, Delhi, Jaipur have the Absorbing Aerosol Index values above 0 which is a sign of hazardous levels of pollution in these cities. In these cities, we can observe that there are some months where the smoke and dust levels are very high and vidibility is quite low, justified by high AAI values.

[130]:
```
plt.figure(figsize = (12,8))
mn = 0
for i in cities1:
    plt.scatter(x = monthly_data_mean.index, y = monthly_data_mean[i],label = i)
    mn += monthly_data_mean[i].mean()

plt.axhline(mn/11.0, color='gray', linestyle='dashed',
            linewidth=2, label = "India Mean")
plt.axhline(0.0, color='red', linestyle='dotted',
            linewidth=2, label = "Universal Cutoff")
plt.title("Scatter Plot for Monthly Aggregates for AAI Variable for 11 cities",␣
 ↪fontsize= 20)
plt.setp(plt.gca().get_xticklabels(), fontsize=10)
plt.legend()
plt.show()
```



The scatter plot of monthly aggregate of AAI, also show a similar trend in the AAI values. The value 0 on x-axis indicates July 2018 and the next values corresponding to their respective months. The Plot also signifies a downward shift in the overall values of AAI since 2018.

[131]:
```
plt.figure(figsize = (20,10))
for i in cities1:
    sns.kdeplot(monthly_data_mean[i], label = i)

plt.title("Density Plot of Values Of AAI for different Cities aggregated␣
 ↪Monthly",fontsize = 30)
plt.xlabel("AAI Values",fontsize = 20)
plt.ylabel("Density",fontsize = 20)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.gca().axvline(x=mn/11.0,color='r', linestyle='--', lw=2, label = "Average␣
 ↪AAI")
plt.legend(fontsize = 15, title ="Cities",title_fontsize = 15 )
plt.show()
```



Curves for most cities end near 0.25. But Cities like Ahmedabad, Delhi, Jaipur have a steeper curve and a lot of values of the Absorbing Aerosol Index above 0 which is a sign of hazardous levels of pollution in these cities. Delhi is the curve which ends the last signifying threatening levels of AAI Values on some occasions

4.4 Grouping Cities into Categories.

If mean of the AOD value for a city is greater than the overall mean of all cities, they are placed in Group 1. The cities not in Group 1 are placed in Group 2

[132]:
```
cities_describe = monthly_data_mean.describe()
grp1=[]
grp2=[]
for i in cities1:
    if(cities_describe[i]["mean"] > mn/11.0):
        grp1.append(i)
    else:
        grp2.append(i)
print("Group 1 of Cities: ", grp1)
print("Group 2 of Cities: ", grp2)
```

```
Group 1 of Cities:  ['Ahmedabad', 'Jaipur', 'Delhi']
Group 2 of Cities:  ['Bangalore', 'Chennai', 'Hyderabad', 'Pune', 'Mumbai',
'Bhopal', 'Bhubaneshwar', 'Kolkata']
```

[133]:
```
cities_describe
```

[133]:
	Ahmedabad	Bangalore	Bhopal	Bhubaneshwar	Chennai	Delhi
count	28.000000	28.000000	28.000000	28.000000	28.000000	28.000000
mean	-0.586662	-1.035066	-0.961516	-1.089359	-1.103280	-0.662504
std	0.418977	0.271389	0.389578	0.292168	0.288606	0.429254
min	-1.250079	-1.518255	-1.711059	-1.658433	-1.552375	-1.319630
25%	-0.952383	-1.205974	-1.291184	-1.330089	-1.305032	-1.032945
50%	-0.549166	-1.106315	-0.932946	-1.122891	-1.118413	-0.602278
75%	-0.297980	-0.889730	-0.700275	-0.878198	-0.946842	-0.421228
max	0.194117	-0.352060	-0.266061	-0.511579	-0.394543	0.152676

	Hyderabad	Jaipur	Kolkata	Mumbai	Pune
count	28.000000	28.000000	28.000000	28.000000	28.000000
mean	-0.949568	-0.686848	-0.950377	-0.932042	-0.993262
std	0.292071	0.387927	0.320910	0.356192	0.382903
min	-1.382430	-1.286629	-1.567887	-1.506054	-1.687852
25%	-1.195781	-1.047640	-1.144023	-1.234692	-1.308180
50%	-0.906686	-0.626147	-0.978432	-0.874155	-0.982953
75%	-0.787334	-0.437523	-0.697642	-0.749008	-0.757161
max	-0.429574	0.103095	-0.307695	-0.010012	-0.322667

[134]:
```
# color scheme
vmin, vmax = min(cities_describe.loc["50%"]),max(cities_describe.loc["50%"])
vmin,vmax
```

[134]:
```
(-1.1228910785, -0.5491664955)
```

We classified cities into 2 separate categories based on the spread of the values

Group 1 Contains cities whose median values lie above the overall mean of the cities. - Delhi - Ahmedabad - Jaipur

Group 2 contains the other cities - Mumbai - Bangalore
- Bhubaneshwar - Chennai - Hyderabad - Pune - Bhopal - Kolkata

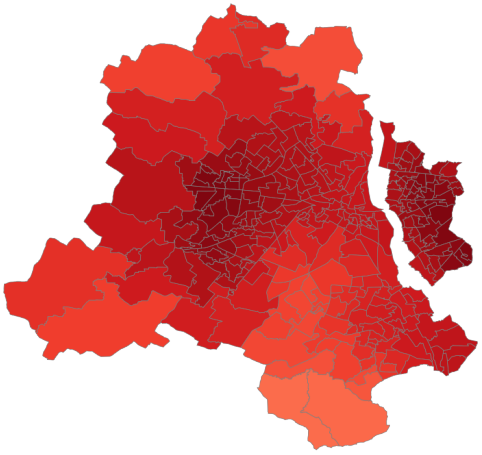
4.5 Comparison among Group 1 and Group 2

[135]:
```
# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values

# create figure and axes for Matplotlib
fig, axes = plt.subplots(nrows = 2,ncols = 2, figsize=(30, 30))
# remove the axis
axes[0][0].axis('off')
axes[0][1].axis('off')
# add a title and annotation
axes[0][0].set_title('Delhi- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
```

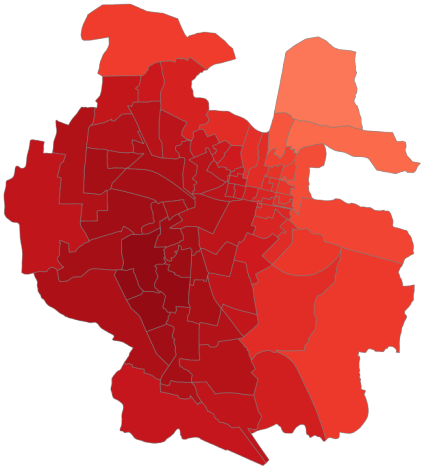
```
axes[0][1].set_title('Jaipur- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})


# sm = fig.cm.ScalarMappable(cmap='Reds',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))

# sm.set_array([])
# # # add the colorbar to the figure
# fig.colorbar(sm)



# remove the axis
axes[1][0].axis('off')
axes[1][1].axis('off')
# add a title and annotation
axes[1][0].set_title('Ahmedabad- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
axes[1][1].set_title('Mumbai- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})


# sm = fig.cm.ScalarMappable(cmap='Reds',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))

# sm.set_array([])
# # # add the colorbar to the figure
# fig.colorbar(sm)


# create map
merged_Ahmedabad.plot(column=variable, cmap='Reds',vmax = vmax,vmin = vmin,␣
 ↪linewidth=0.8, ax=axes[1][0], edgecolor='0.5', figsize = (5,15))
merged_Mumbai.plot(column=variable, cmap='Reds',vmax = vmax,vmin = vmin,␣
 ↪linewidth=0.8, ax=axes[1][1], edgecolor='0.5', figsize = (5,15))
# create map
merged_Delhi.plot(column=variable, cmap='Reds',vmax = vmax,vmin = vmin,␣
 ↪linewidth=0.8, ax=axes[0][0], edgecolor='0.5', figsize = (5,15))
merged_Jaipur.plot(column=variable, cmap='Reds',vmax = vmax,vmin = vmin,␣
 ↪linewidth=0.8, ax=axes[0][1], edgecolor='0.5', figsize = (5,15))
plt.show()

# Plotting chloropleth
# set the value column that will be visualised
variable = 'mean'
```

```
# # set the range for the choropleth values

# create figure and axes for Matplotlib
fig, axes = plt.subplots(nrows = 2,ncols = 2, figsize=(30, 30))
# remove the axis
axes[0][0].axis('off')
axes[0][1].axis('off')
# add a title and annotation
axes[0][0].set_title('Bangalore- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
axes[0][1].set_title('Kolkata- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})


# remove the axis
axes[1][0].axis('off')
axes[1][1].axis('off')
# add a title and annotation
axes[1][0].set_title('Hyderabad- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
axes[1][1].set_title('Chennai- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})



# create map
merged_Hyderabad.plot(column=variable, cmap='Reds',vmax = vmax,vmin = vmin,␣
 ↪linewidth=0.8, ax=axes[1][0], edgecolor='0.5', figsize = (5,15))
merged_Chennai.plot(column=variable, cmap='Reds',vmax = vmax,vmin = vmin,␣
 ↪linewidth=0.8, ax=axes[1][1], edgecolor='0.5', figsize = (5,15))
# create map
merged_Bangalore.plot(column=variable, cmap='Reds',vmax = vmax,vmin = vmin,␣
 ↪linewidth=0.8, ax=axes[0][0], edgecolor='0.5', figsize = (5,15))
merged_Kolkata.plot(column=variable, cmap='Reds',vmax = vmax,vmin = vmin,␣
 ↪linewidth=0.8, ax=axes[0][1], edgecolor='0.5', figsize = (5,15))
plt.show()
```

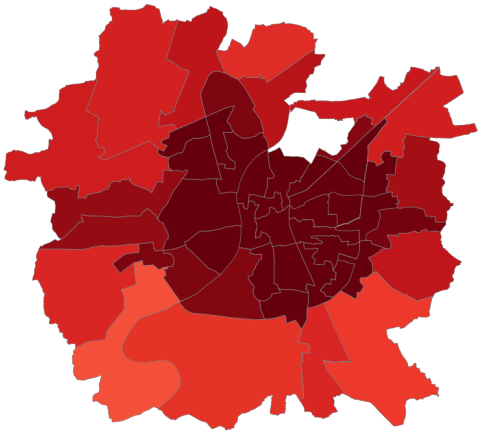
Delhi- Ward Wise Mean of Absorbing Aerosol Index



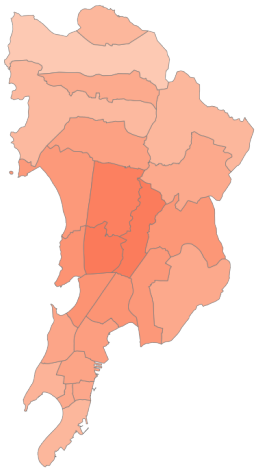
Jaipur- Ward Wise Mean of Absorbing Aerosol Index

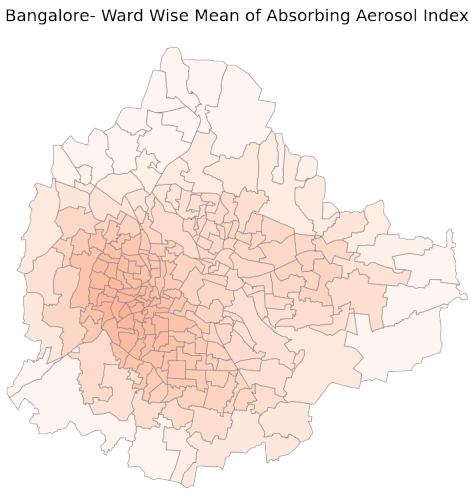


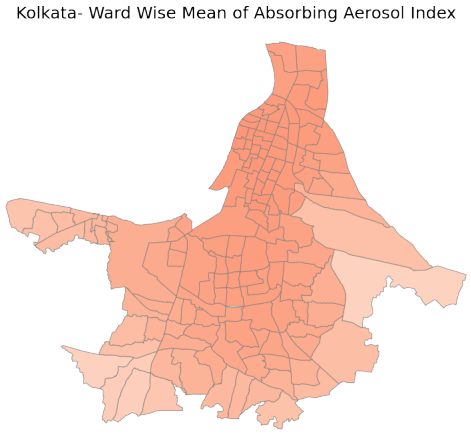
Ahmedabad- Ward Wise Mean of Absorbing Aerosol Index

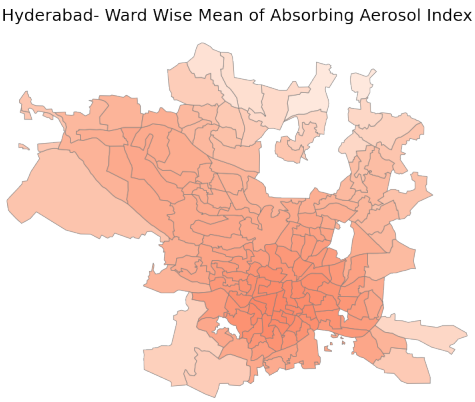


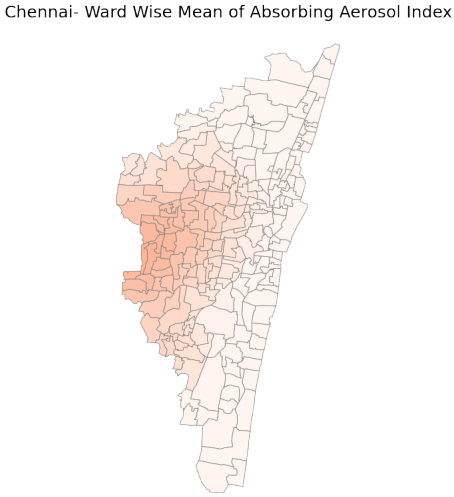
Mumbai- Ward Wise Mean of Absorbing Aerosol Index











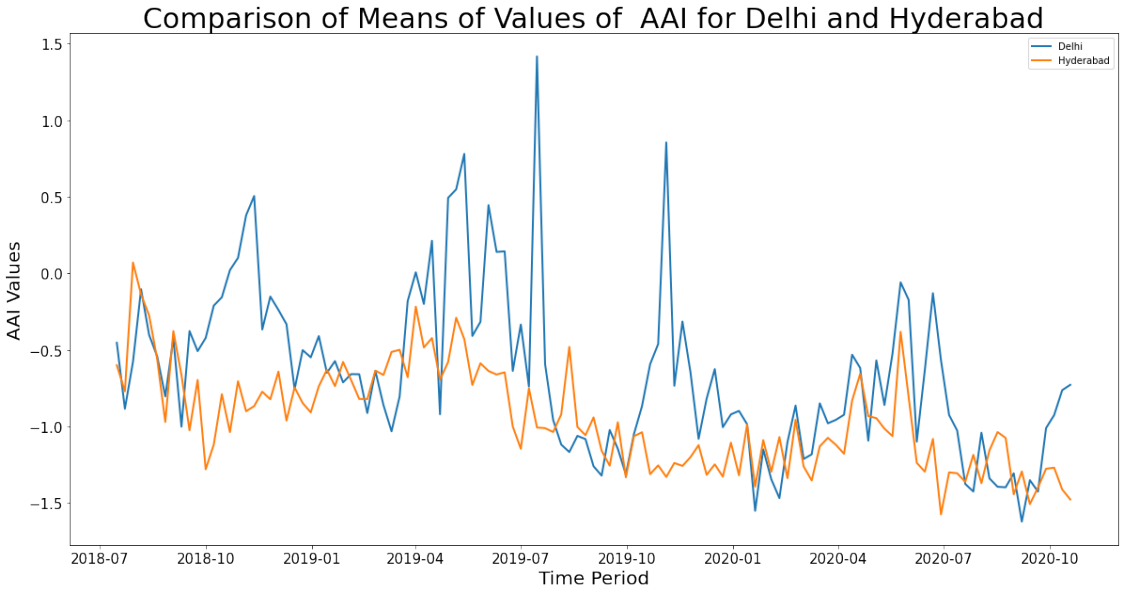
We can clearly observe that Group 1 cities have very high AAI values when compared to Group 2 cities. Now we will dive in to comparison of 2 specific cities picked randomly from Group 1 and Group 2 respectively.

4.6 Comparative Case Study Of Group 1 and Group 2 cities with Delhi And Hyderabad as Prime Cities on Absorbing Aerosol Index(AAI)

[136]:
```
plt.figure(figsize = (20,10))
plt.plot(Weekly_Delhi_mean.index, Weekly_Delhi_mean["mean"], label = "Delhi",␣
↪lw = 2)
plt.plot(Weekly_Hyderabad_mean.index, Weekly_Hyderabad_mean["mean"], label =␣
↪"Hyderabad", lw =2 )
plt.xlabel("Time Period",fontsize = 20)
```

```
plt.ylabel("AAI Values", fontsize = 20)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.title("Comparison of Means of Values of  AAI for Delhi and Hyderabad",␣
 ↪fontsize = 30)
plt.legend()
```

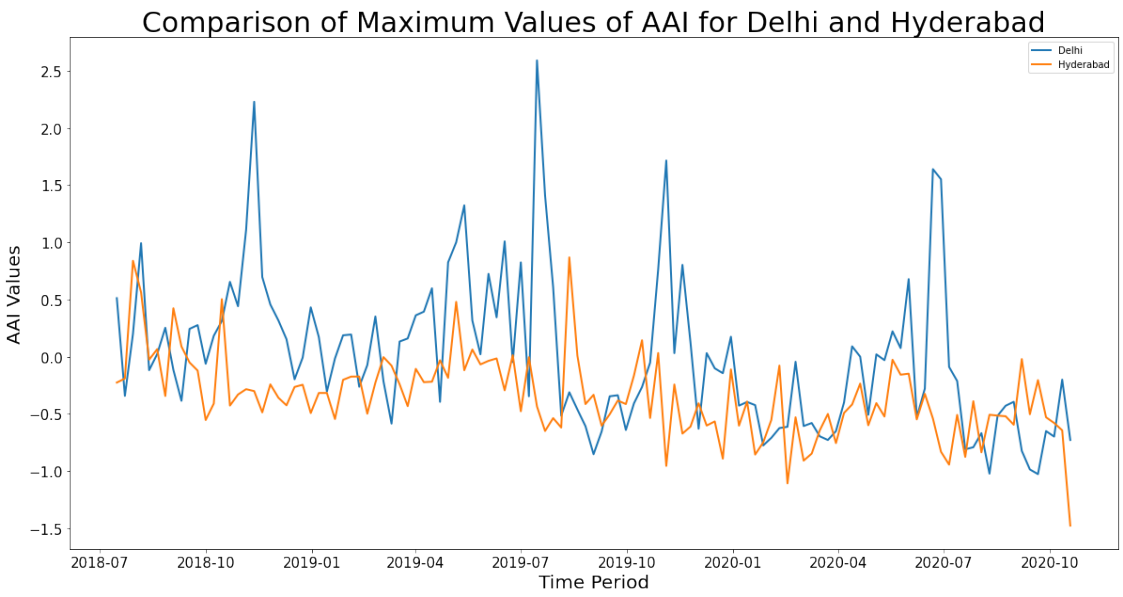
[136]: <matplotlib.legend.Legend at 0x241c52312e0>



We can observe that the curve of **Delhi (Group1)** is normally higher than **Hyderabad (Group 2)** at most points. Also **Delhi (Group1)** has larger peaks indicating its high mean AAI values.

[137]:
```
plt.figure(figsize = (20,10))
plt.plot(Weekly_Delhi_max.index, Weekly_Delhi_max["mean"], label = "Delhi", lw␣
 ↪= 2)
plt.plot(Weekly_Hyderabad_max.index, Weekly_Hyderabad_max["mean"], label =␣
 ↪"Hyderabad", lw =2 )
plt.xlabel("Time Period",fontsize = 20)
plt.ylabel("AAI Values", fontsize = 20)
plt.setp(plt.gca().get_xticklabels(), fontsize=15)
plt.setp(plt.gca().get_yticklabels(), fontsize=15)
plt.title("Comparison of Maximum Values of AAI for Delhi and Hyderabad",␣
 ↪fontsize = 30)
plt.legend()
```

[137]: <matplotlib.legend.Legend at 0x241bf9eb370>

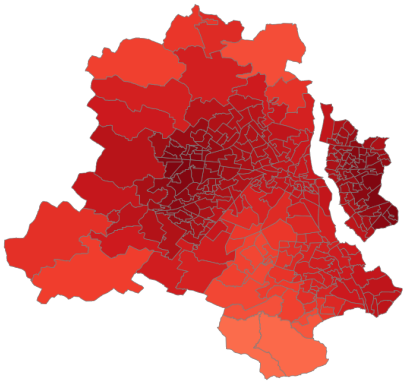


We can observe that the curve of **Delhi (Group1)** is mostly higher than **Hyderabad (Group 2)** at most points. Also **Delhi (Group1)** has larger peaks indicating its high max AAI values.

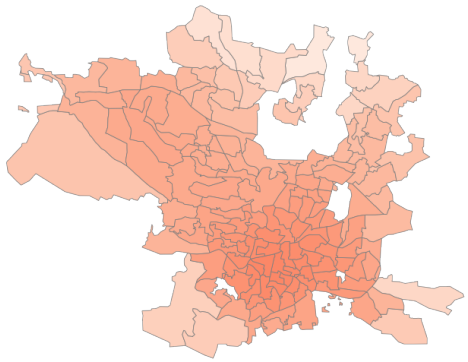
[138]:
```
# Plotting choropleth
# set the value column that will be visualised
variable = 'mean'
# # set the range for the choropleth values

# create figure and axes for Matplotlib
fig, axes = plt.subplots(ncols = 2, figsize=(30, 10))
fig.subplots_adjust(right=0.8)
# remove the axis
axes[0].axis('off')
axes[1].axis('off')
# add a title and annotation
axes[0].set_title('Delhi- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})
axes[1].set_title('Hyderabad- Ward Wise Mean of Absorbing Aerosol Index',␣
 ↪fontdict={'fontsize': '25', 'fontweight' : '3'})


# sm = fig.cm.ScalarMappable(cmap='coolwarm',norm=plt.Normalize(vmin=vmin,␣
 ↪vmax=vmax ))

# sm.set_array([])
# # # add the colorbar to the figure
# fig.colorbar(sm)
```

```
# create map
merged_Delhi.plot(column=variable, cmap='Reds',vmax = vmax,vmin = vmin,␣
 ↪linewidth=0.8, ax=axes[0], edgecolor='0.5', figsize = (5,15))
merged_Hyderabad.plot(column=variable, cmap='Reds',vmax = vmax,vmin = vmin,␣
 ↪linewidth=0.8, ax=axes[1], edgecolor='0.5', figsize = (5,15))
plt.show()
```

Delhi- Ward Wise Mean of Absorbing Aerosol Index



Hyderabad- Ward Wise Mean of Absorbing Aerosol Index



Conclusion

Delhi representing Group 1 have very high AAI Values depciting high presence of Aerosols. Where Hyderabad representing Group 2 is on a lighter shade.

Aerosols have a wide affect on the climatic conditions of the region. And due to presence of Aerosols which are produced by man made activities, it has further affected a region adversely. Initiatives have been taken by the Goverments of the Nations to curb this and move towards Sustainable Development. And we can see the results above where we had seen that climate had indeed gotten better than the previous year attributed to Corona Virus Pandemic where the working class was restricted to staying indoors. There is light at the end of this dark tunnel and we are currently heading towards it. Advancing in Technology to make use of Renewable Resources seems to be the next way forward.

Thank You