In [3]:

```
sns.__version__
```

Out[3]:

```
'0.10.0'
```

In [4]:

```
import pandas as pd
```

In [5]:

```
pd.__version__
```

Out[5]:

```
'1.0.1'
```

In [27]:

```
pollution_data = pd.read_csv('./Downloads/UFOPOLLUTANTS.csv')
```

In [28]:

```
pollution_data.head()
```

Out[28]:

	State.Code	city	state	day	month	year	hour	NO2.Mean	NO2.1st.Max.Value	NO2.
0	4	kingman (desert&#44 out of)	az	7	7	2000	21	7.375000	15.0	
1	4	tacna (approx)	az	7	7	2000	21	7.375000	15.0	
2	4	apache junction	az	6	9	2001	21	7.708333	19.0	
3	4	sedona	az	5	6	2005	21	7.916667	17.0	
4	4	tucson	az	5	10	2007	2	8.625000	20.0	

5 rows × 24 columns

In [29]:

```
len(pollution_data.index)
```

Out[29]:

```
63173
```

In [30]:

```
poluution_data = pollution_data.head(1000)
```

In [31]:

```
pollution_data = pollution_data.head(1000)
```

In [32]:

```
len(pollution_data.index)
```

Out[32]:

```
1000
```

In [33]:

```
pollution_data.describe()
```

Out[33]:

	State.Code	day	month	year	hour	NO2.Mean	NO2.1st.M
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000	1000.000000	100
mean	10.933000	9.963000	4.759000	2002.697000	14.02400	22.794976	4
std	13.365371	7.372651	3.471739	2.871835	7.92971	14.666838	2
min	4.000000	1.000000	1.000000	2000.000000	0.00000	0.043478	
25%	4.000000	5.000000	2.000000	2000.000000	6.00000	10.076087	2
50%	6.000000	8.000000	3.000000	2002.000000	17.50000	20.043478	3
75%	6.000000	12.000000	7.000000	2005.000000	21.00000	33.775198	5
max	51.000000	31.000000	12.000000	2008.000000	23.00000	73.285714	12

8 rows × 22 columns

In [34]:

```
import warnings
```

In [35]:

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

In [36]:

```
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [39]:

```
pollution_data = pd.read_csv('./Downloads/UFOPOLLUTANTS.csv')
pollution_data = pollution_data.head(1000)
```

In [40]:

```
pollution_data.head()
```

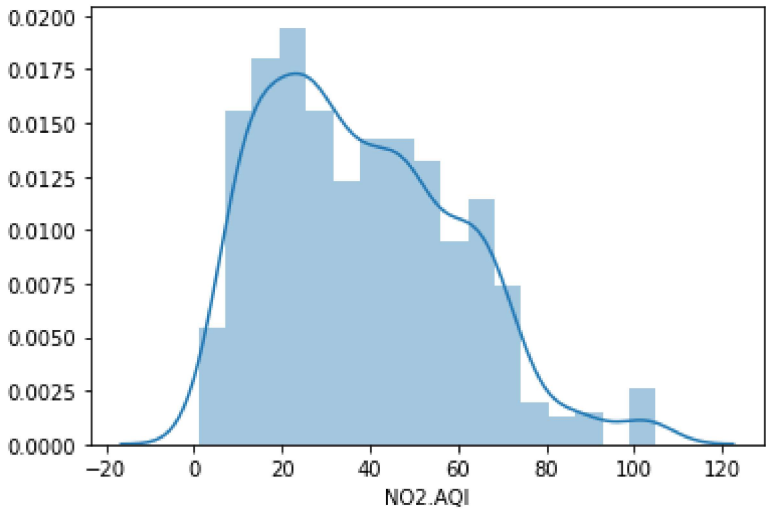
Out[40]:

	State.Code	city	state	day	month	year	hour	NO2.Mean	NO2.1st.Max.Value	NO2.
0	4	kingman (desert&#44 out of)	az	7	7	2000	21	7.375000	15.0	
1	4	tacna (approx)	az	7	7	2000	21	7.375000	15.0	
2	4	apache junction	az	6	9	2001	21	7.708333	19.0	
3	4	sedona	az	5	6	2005	21	7.916667	17.0	
4	4	tucson	az	5	10	2007	2	8.625000	20.0	

5 rows × 24 columns

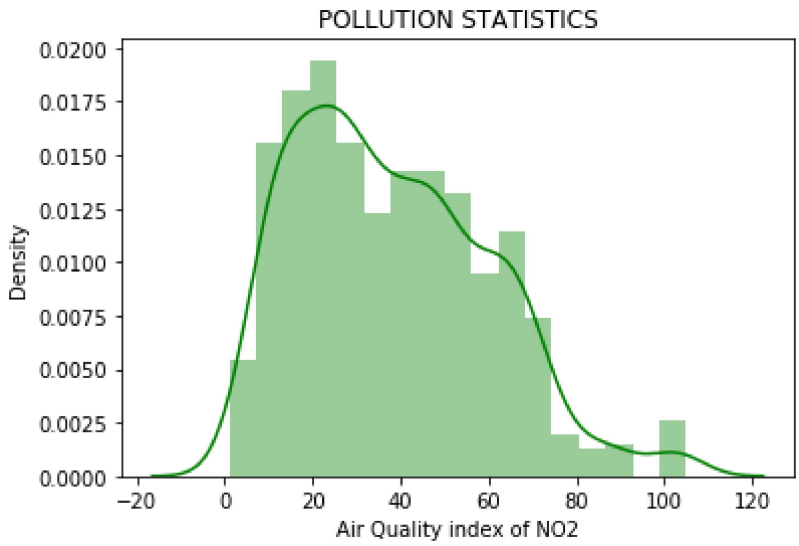
In [41]:

```
sns.distplot(pollution_data['NO2.AQI'])
plt.show()
```



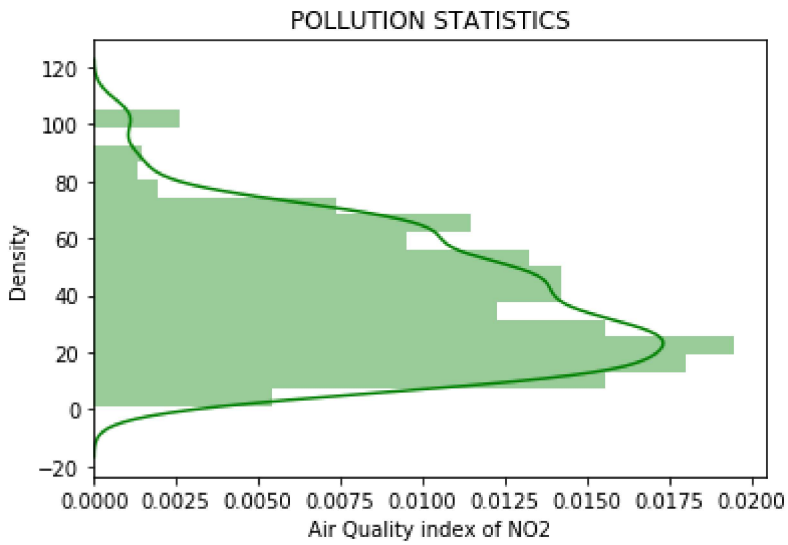
In [42]:

```
dist = sns.distplot(pollution_data["NO2.AQI"],
                    color = "Green")
dist.set_title("POLLUTION STATISTICS")
dist.set_xlabel("Air Quality index of NO2")
dist.set_ylabel("Density")
plt.show()
```



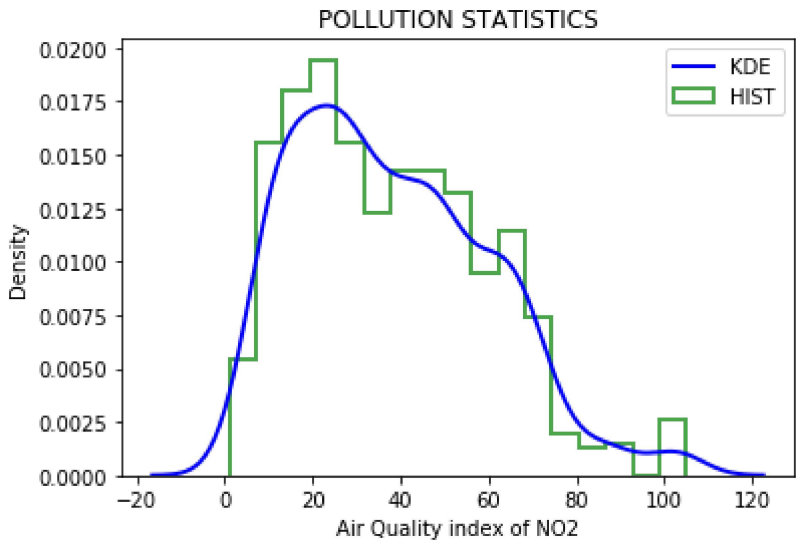
In [43]:

```
dist = sns.distplot(pollution_data["NO2.AQI"],
                    color = "Green",
                    vertical = True)
dist.set_title("POLLUTION STATISTICS")
dist.set_xlabel("Air Quality index of NO2")
dist.set_ylabel("Density")
plt.show()
```



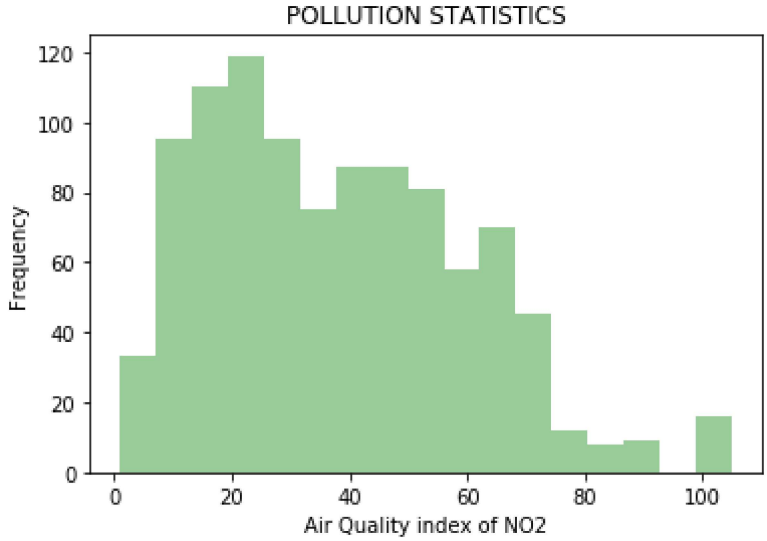
In [45]:

```
dist = sns.distplot(pollution_data["NO2.AQI"],
                    kde_kws ={"color": "B",
                              "lw":2,
                              "label": 'KDE'},
                    hist_kws = {"histtype": "step",
                                "linewidth": 2,
                                "alpha": 0.7,
                                "color": "g",
                                "label":"HIST"})

dist.set_title("POLLUTION STATISTICS")
dist.set_xlabel("Air Quality index of NO2")
dist.set_ylabel("Density")

plt.show()
```



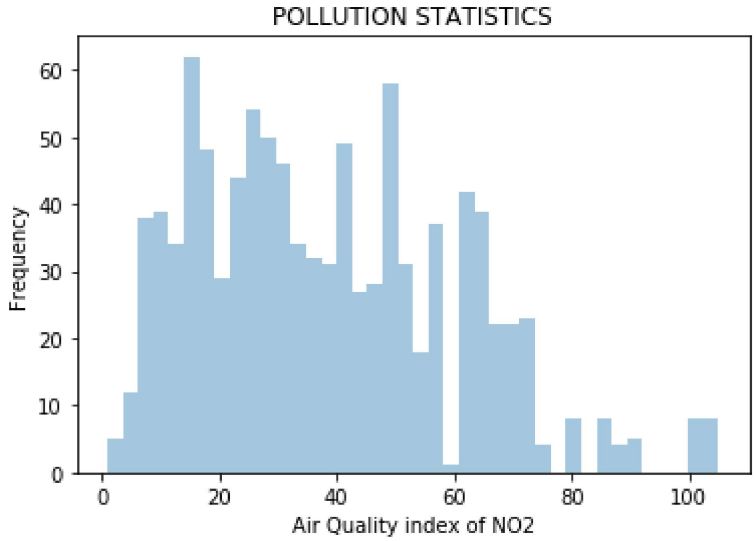
In [46]:

```
dist = sns.distplot(pollution_data["NO2.AQI"],
                    color = "Green",
                     kde = False)

dist.set_title("POLLUTION STATISTICS")
dist.set_xlabel("Air Quality index of NO2")
dist.set_ylabel("Frequency")
plt.show()
```



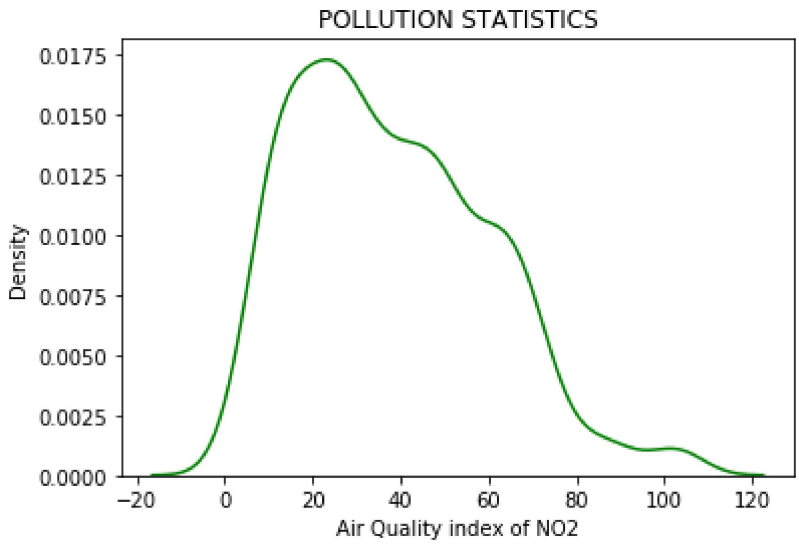
In [47]:

```
dist = sns.distplot(pollution_data["NO2.AQI"],
                    bins = 40,
                    kde = False)

dist.set_title("POLLUTION STATISTICS")
dist.set_xlabel("Air Quality index of NO2")
dist.set_ylabel("Frequency")

plt.show()
```



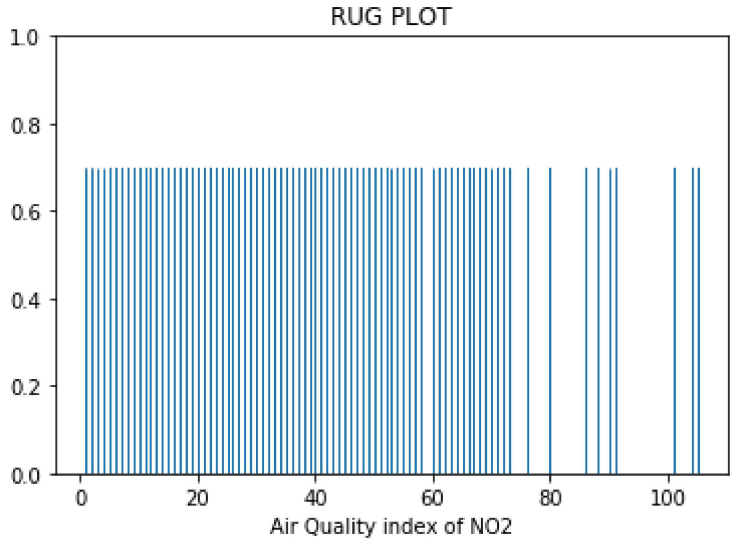
In [49]:

```
dist = sns.distplot(pollution_data["NO2.AQI"],
                    color = "Green",
                    hist = False)

dist.set_title("POLLUTION STATISTICS")
dist.set_xlabel("Air Quality index of NO2")
dist.set_ylabel("Density")

plt.show()
```



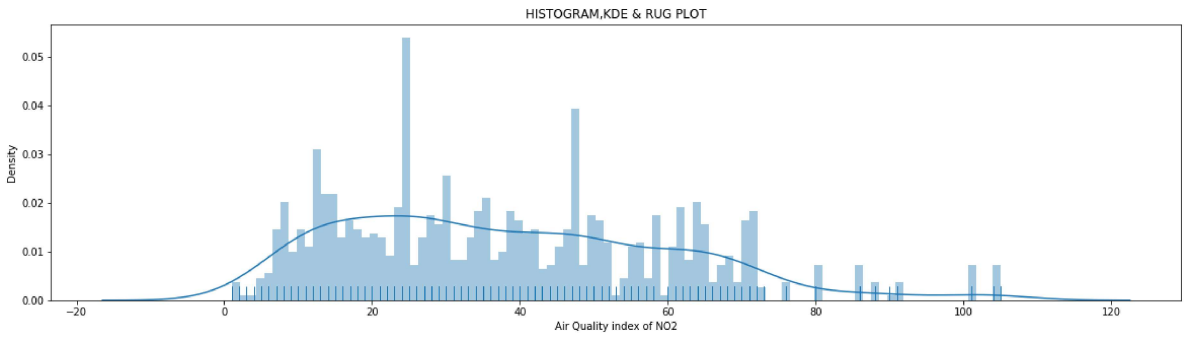
In [51]:

```
rug  = sns.rugplot(pollution_data["NO2.AQI"],
                   height = 0.7,
                   axis = "x")

rug.set_title("RUG PLOT")
rug.set_xlabel("Air Quality index of NO2")

plt.show()
```



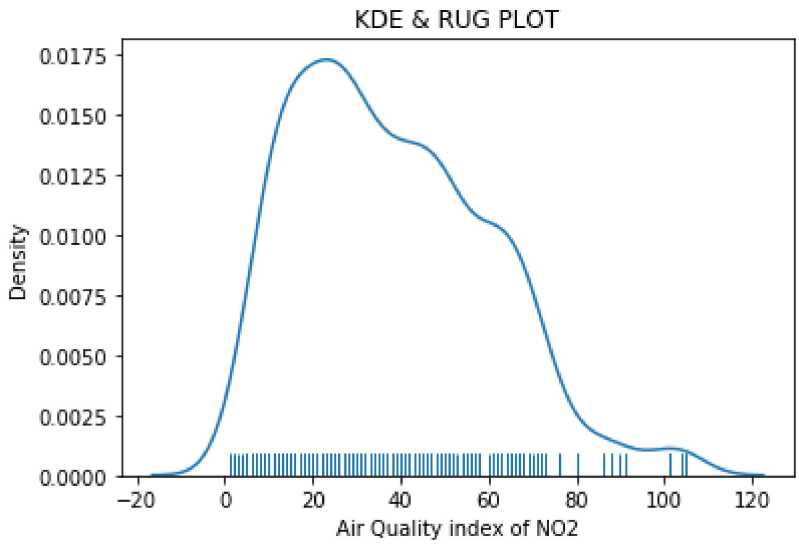
In [55]:

```
plt.subplots(figsize = (20,5))
dist = sns.distplot(pollution_data["NO2.AQI"],
                    bins = 95,
                    rug = True)

dist.set_title("HISTOGRAM,KDE & RUG PLOT")
dist.set_xlabel("Air Quality index of NO2")
dist.set_ylabel("Density")

plt.show()
```



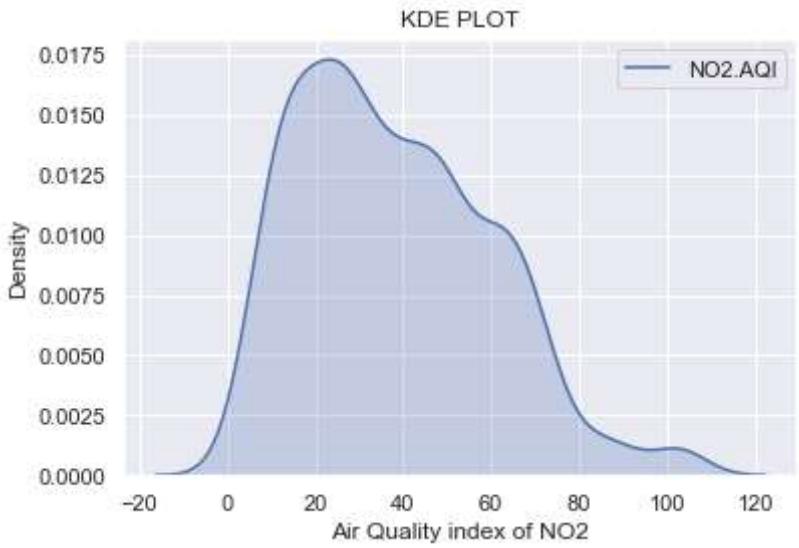
In [56]:

```
dist = sns.distplot(pollution_data["NO2.AQI"],
                    hist = False,
                    rug = True)

dist.set_title("KDE & RUG PLOT")
dist.set_xlabel("Air Quality index of NO2")
dist.set_ylabel("Density")

plt.show()
```



In [57]:

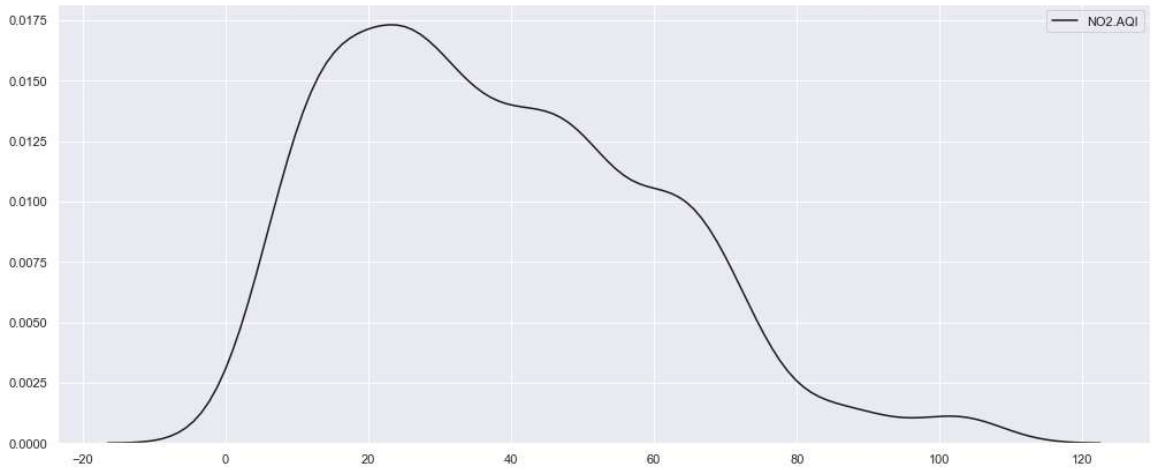
```
sns.set(color_codes = True)

plot = sns.kdeplot(pollution_data["NO2.AQI"],
                   shade = True,
                   color = "b")

plot.set_title("KDE PLOT")
plot.set_xlabel("Air Quality index of NO2")
plot.set_ylabel("Density")

plt.show()
```



In [58]:

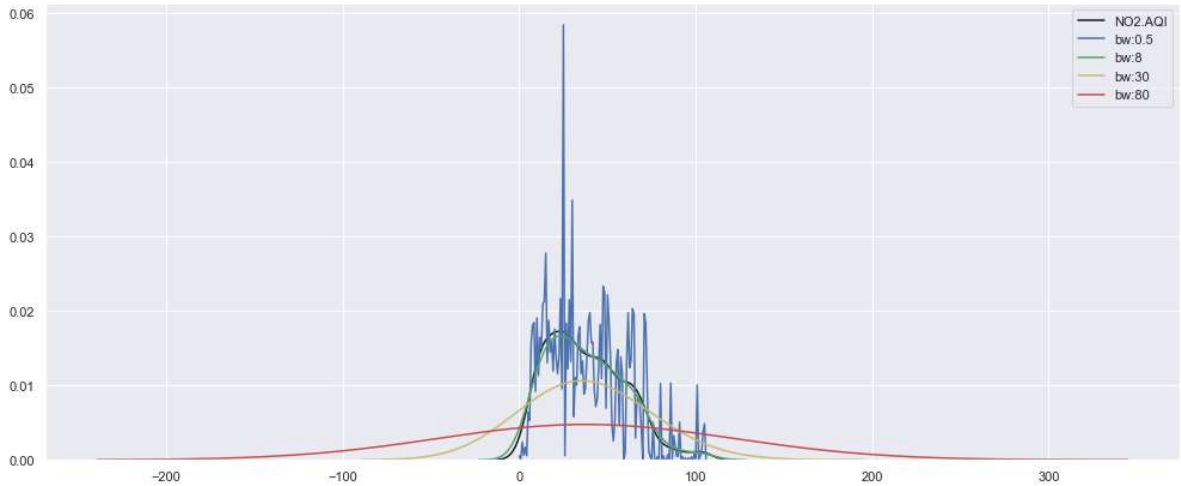
```
plt.subplots(figsize = (17,7))

sns.kdeplot(pollution_data["NO2.AQI"],
            color = "k")

sns.kdeplot(pollution_data["NO2.AQI"],
            BW = 0.3,
            label = "bw:0.5",
            color = "b")

sns.kdeplot(pollution_data["NO2.AQI"],
            BW = 8,
            label = "bw:8",
            color = "g")

sns.kdeplot(pollution_data["NO2.AQI"],
            BW = 30,
            label = "bw:30",
            color = "y")

sns.kdeplot(pollution_data["NO2.AQI"],
            BW = 80,
            label = "bw:80",
            color = "r")

plt.show()
```

```
    971                 return func(v)
    972

AttributeError: 'Line2D' object has no property 'bw'
```



In [59]:

```
plt.subplots(figsize = (17,7))

sns.kdeplot(pollution_data["NO2.AQI"],
           color = "k")

sns.kdeplot(pollution_data["NO2.AQI"],
           bw = 0.3,
           label = "bw:0.5",
           color = "b")

sns.kdeplot(pollution_data["NO2.AQI"],
           bw = 8,
           label = "bw:8",
           color = "g")

sns.kdeplot(pollution_data["NO2.AQI"],
           bw = 30,
           label = "bw:30",
           color = "y")

sns.kdeplot(pollution_data["NO2.AQI"],
           bw = 80,
           label = "bw:80",
           color = "r")

plt.show()
```



In [2]:

```
import seaborn as sns
import pandas as pd
```

In [4]:

```
pollution_data = pd.read_csv('./Downloads/UFOPOLLUTANTS.csv')
pollution_data = pollution_data.head(1000)
```

In [5]:

```
pollution_data.head()
```

Out[5]:

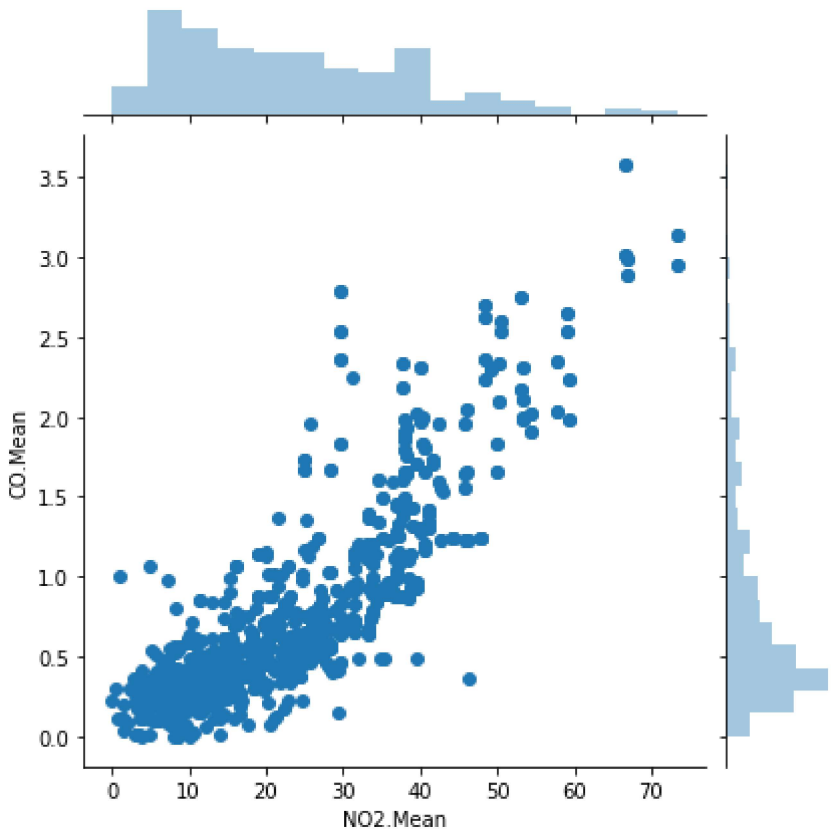
	State.Code	city	state	day	month	year	hour	NO2.Mean	NO2.1st.Max.Value	NO2.
0	4	kingman (desert&#44 out of)	az	7	7	2000	21	7.375000	15.0	
1	4	tacna (approx)	az	7	7	2000	21	7.375000	15.0	
2	4	apache junction	az	6	9	2001	21	7.708333	19.0	
3	4	sedona	az	5	6	2005	21	7.916667	17.0	
4	4	tucson	az	5	10	2007	2	8.625000	20.0	

5 rows × 24 columns

In [6]:

```
joint_plot = sns.jointplot("NO2.Mean",
                           "CO.Mean",
                           data = pollution_data)

plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-6-5f3143f7c7ed> in <module>
      3                           data = pollution_data)
      4
----> 5 plt.show()

NameError: name 'plt' is not defined
```

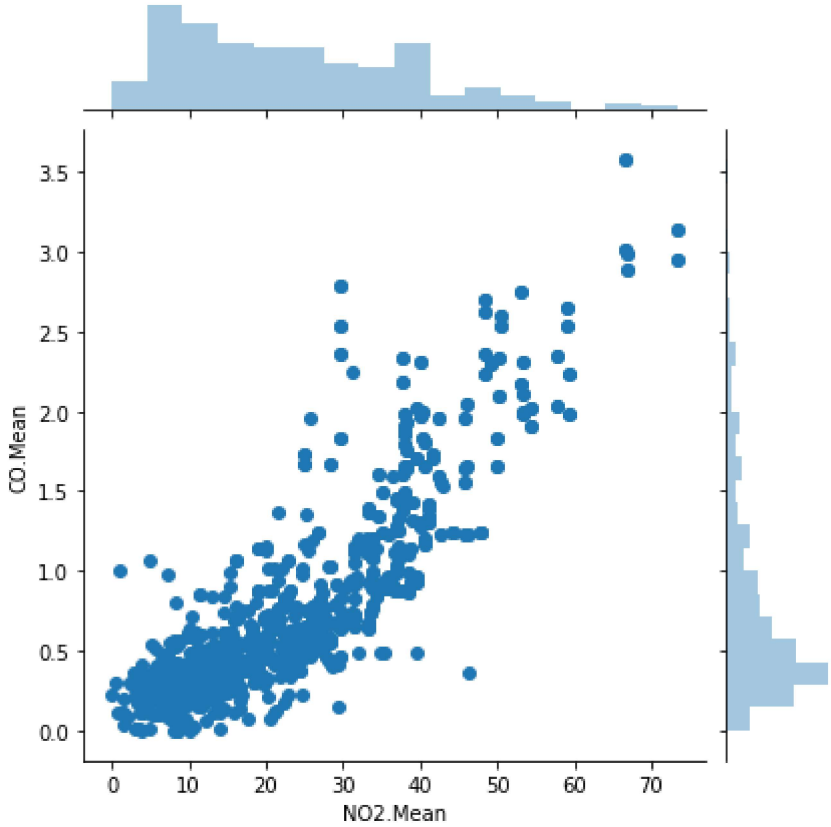


In [7]:

```
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

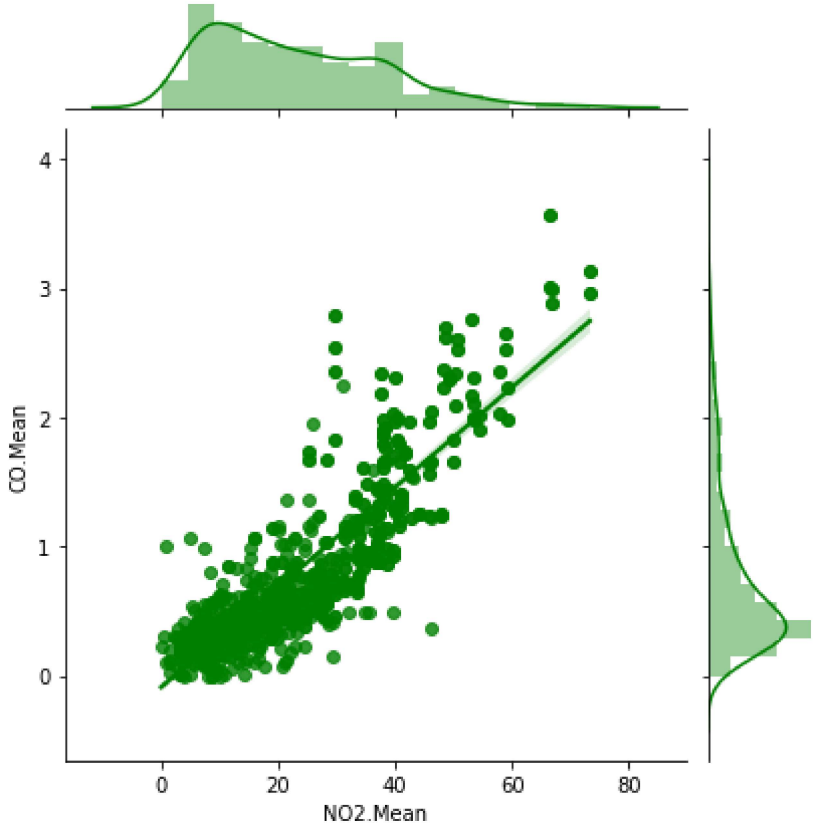
In [8]:

```
joint_plot = sns.jointplot("NO2.Mean",
                           "CO.Mean",
                           data = pollution_data)

plt.show()
```



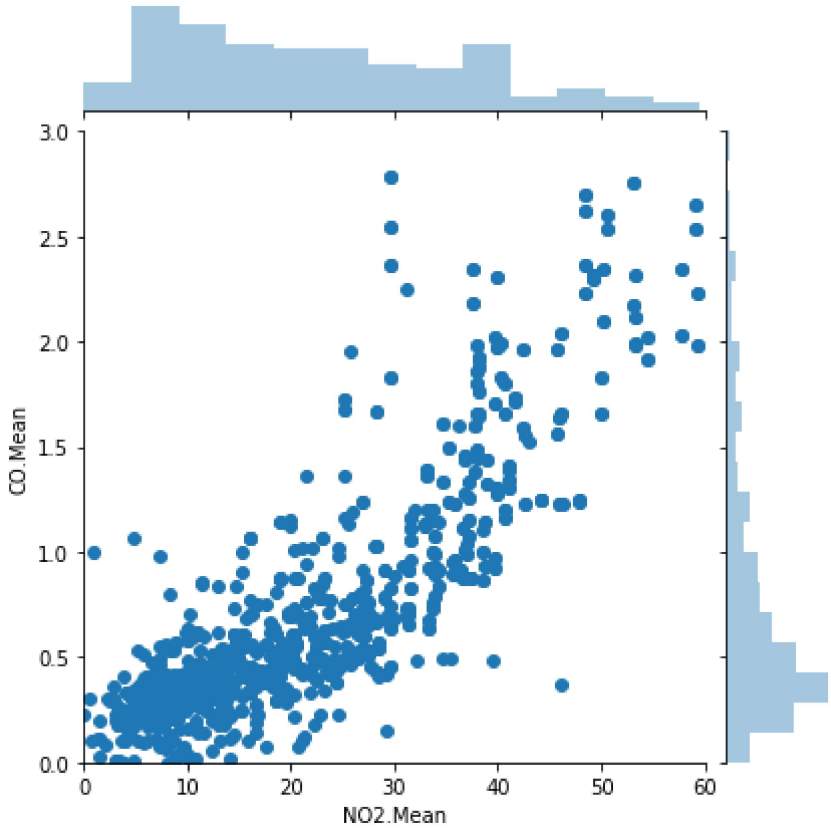
In [9]:

```
joint_plot = sns.jointplot("NO2.Mean",
                           "CO.Mean",
                            data = pollution_data,
                            kind = "reg",
                            color ="g")


plt.show()
```



In [10]:

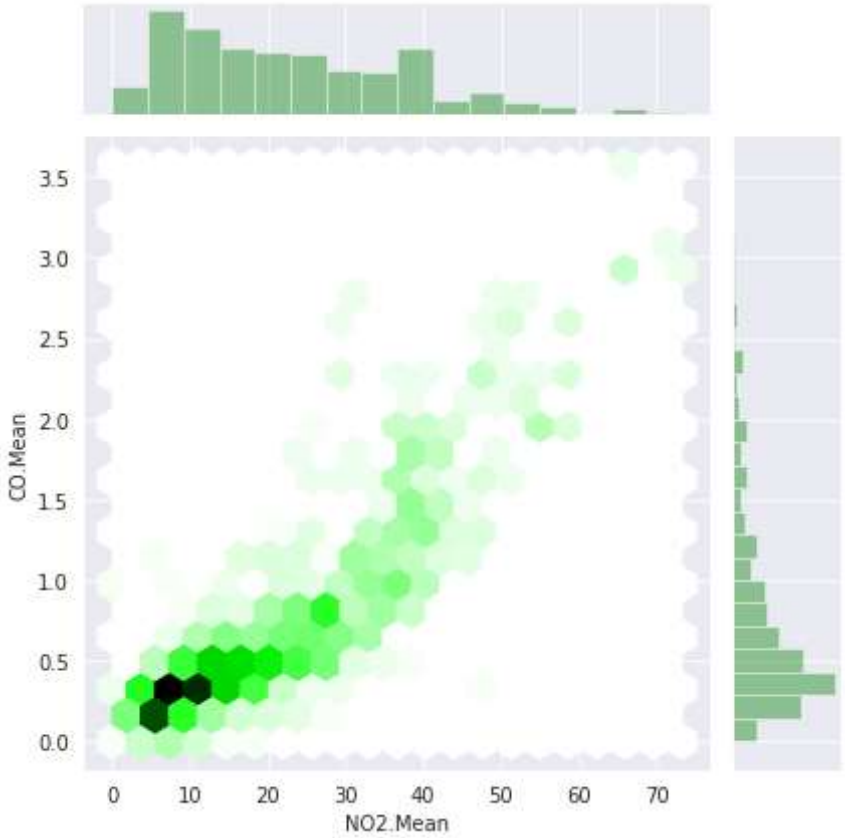
```
joint_plot = sns.jointplot("NO2.Mean",
                           "CO.Mean",
                            data = pollution_data,
                            xlim = {0,60},
                            ylim = {0,3})


plt.show()
```



In [11]:

```
with sns.axes_style("darkgrid"):

    sns.jointplot("NO2.Mean",
                  "CO.Mean",
                  data = pollution_data,
                  kind = "hex",
                  color = "G")
plt.show()
```

```
C:\Users\lenovo\anaconda3\lib\site-packages\seaborn\axisgrid.py:2283: Matplo
tlibDeprecationWarning: Support for uppercase single-letter colors is deprec
ated since Matplotlib 3.1 and will be removed in 3.3; please use lowercase i
nstead.
  color_rgb = mpl.colors.colorConverter.to_rgb(color)
```

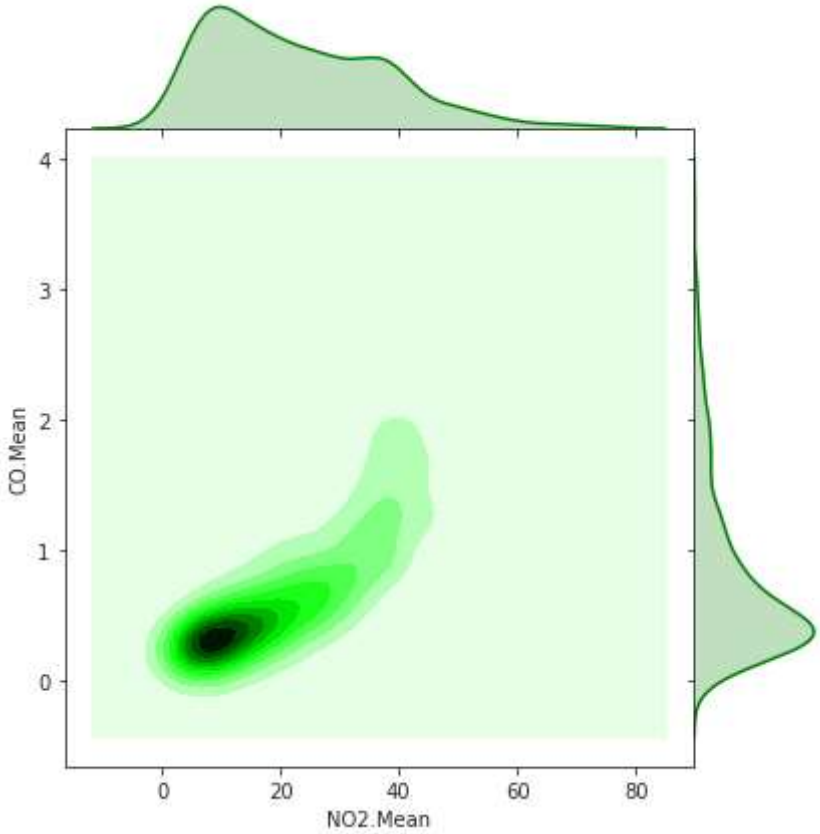


In [12]:

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

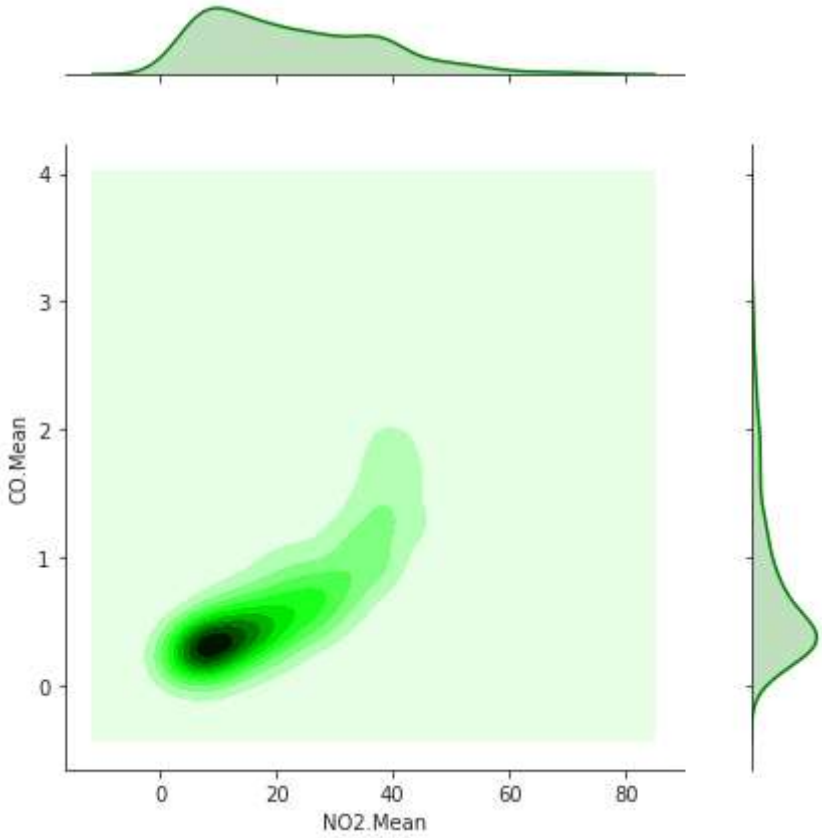
In [14]:

```
with sns.axes_style("ticks"):
    sns.jointplot("NO2.Mean",
                  "CO.Mean",
                  data = pollution_data,
                  kind = "kde",
                  space = 0,
                  color = "G")

plt.show()
```



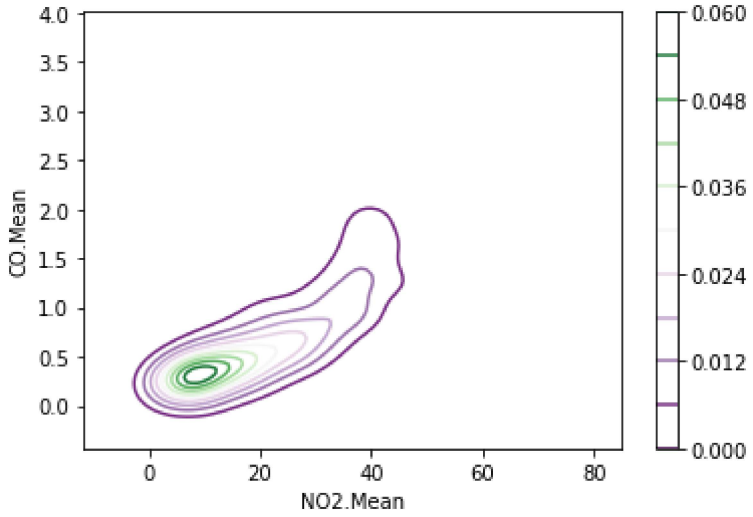
In [15]:

```
with sns.axes_style("ticks"):

    sns.jointplot("NO2.Mean",
                  "CO.Mean",
                  data = pollution_data,
                  kind = "kde",
                  space = 1,
                  color = "G")

plt.show()
```



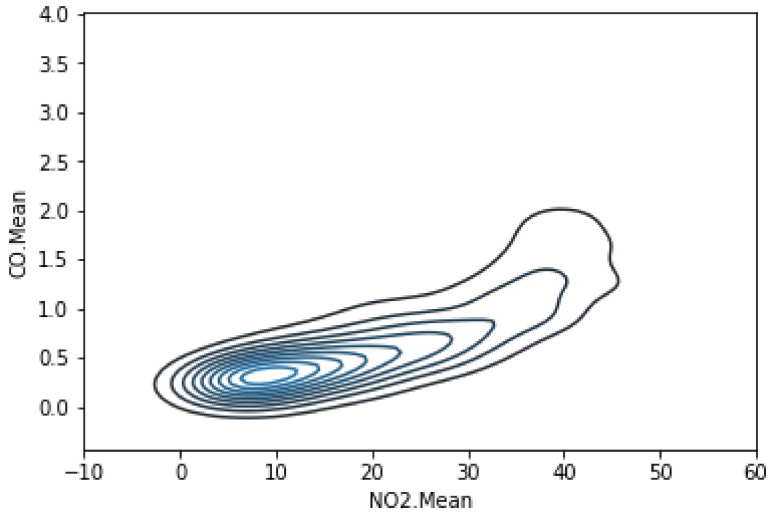
In [16]:

```
sns.kdeplot(pollution_data["NO2.Mean"],
           pollution_data["CO.Mean"],
           cmap = "PRGn",
           cbar = True)
plt.show()
```



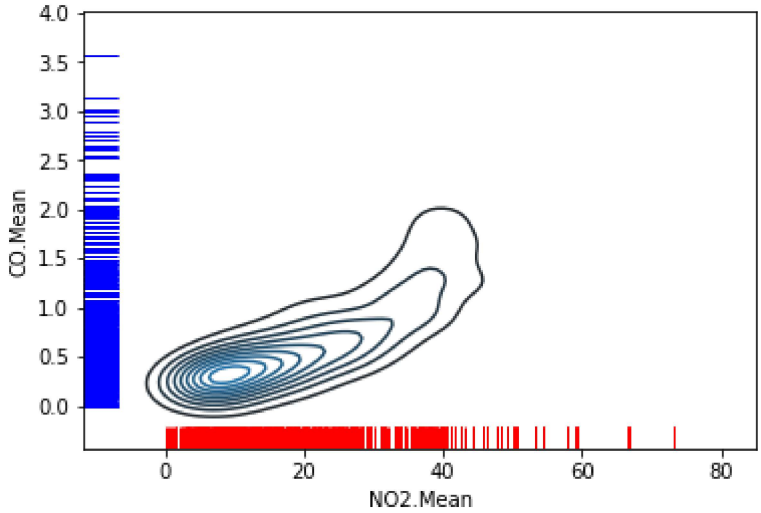
In [17]:

```
sns.kdeplot(pollution_data["NO2.Mean"],
           pollution_data["CO.Mean"],
           clip = (-10,60))

plt.show()
```



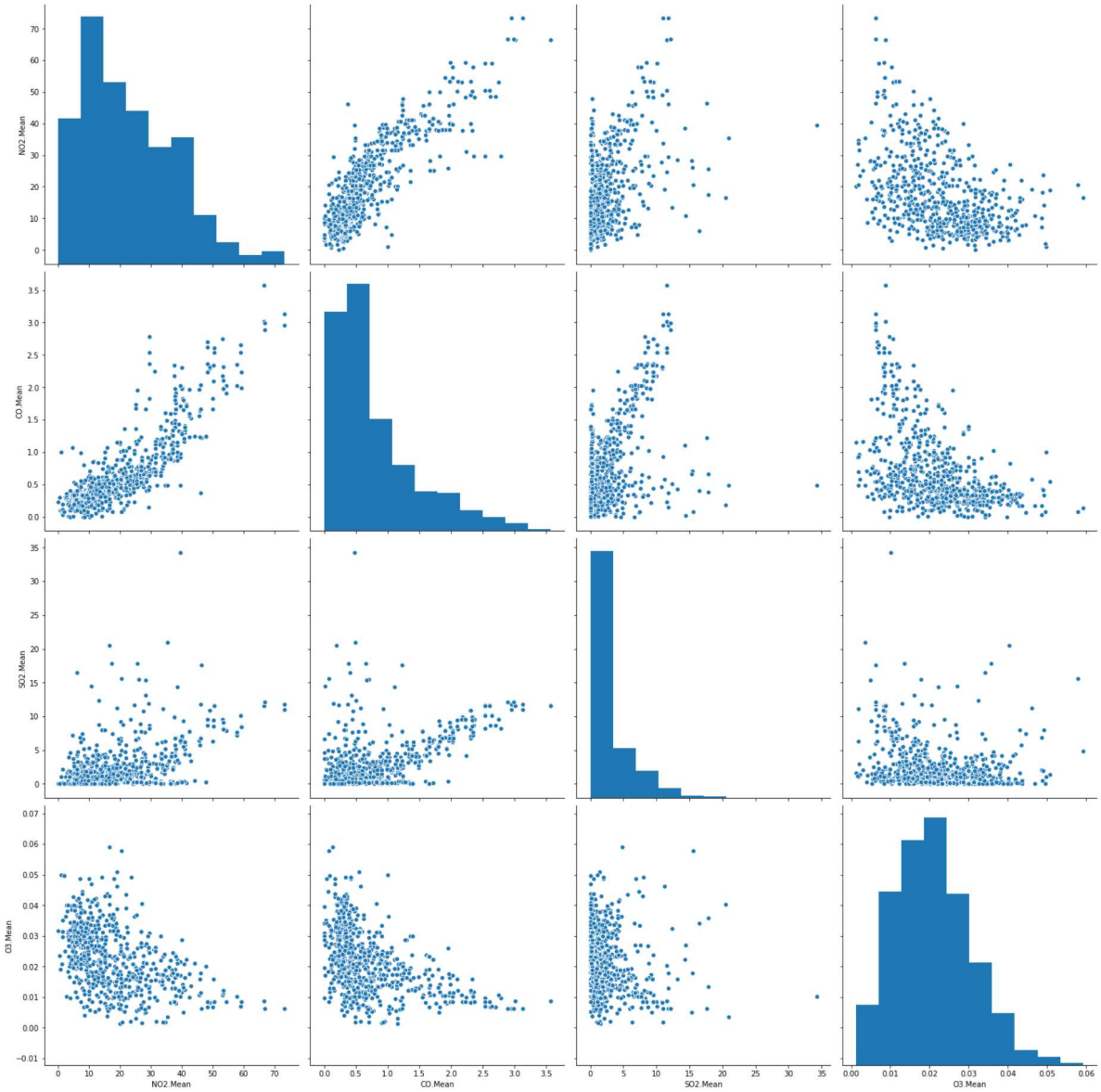
In [18]:

```
sns.kdeplot(pollution_data["NO2.Mean"],
           pollution_data["CO.Mean"])

sns.rugplot(pollution_data["NO2.Mean"],
           color = "r",
           height = .05)

sns.rugplot(pollution_data["CO.Mean"],
            color = "b",
            vertical = True,
            height = .05)

plt.show()
```



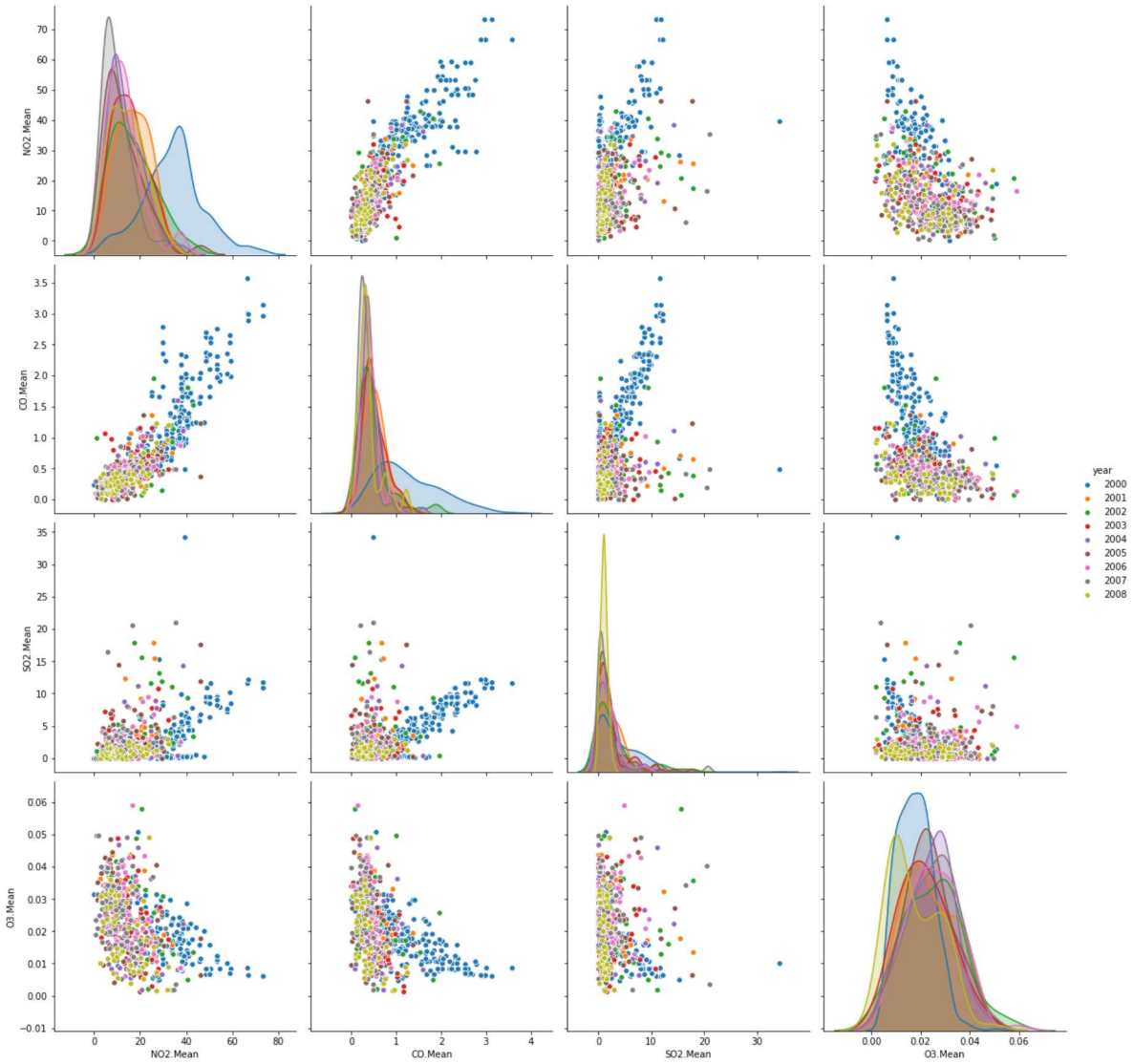
In [19]:

```
sns.pairplot(pollution_data,
             height = 5,
             vars = ["NO2.Mean",
                     "CO.Mean",
                     "SO2.Mean",
                     "O3.Mean"])

plt.show()
```



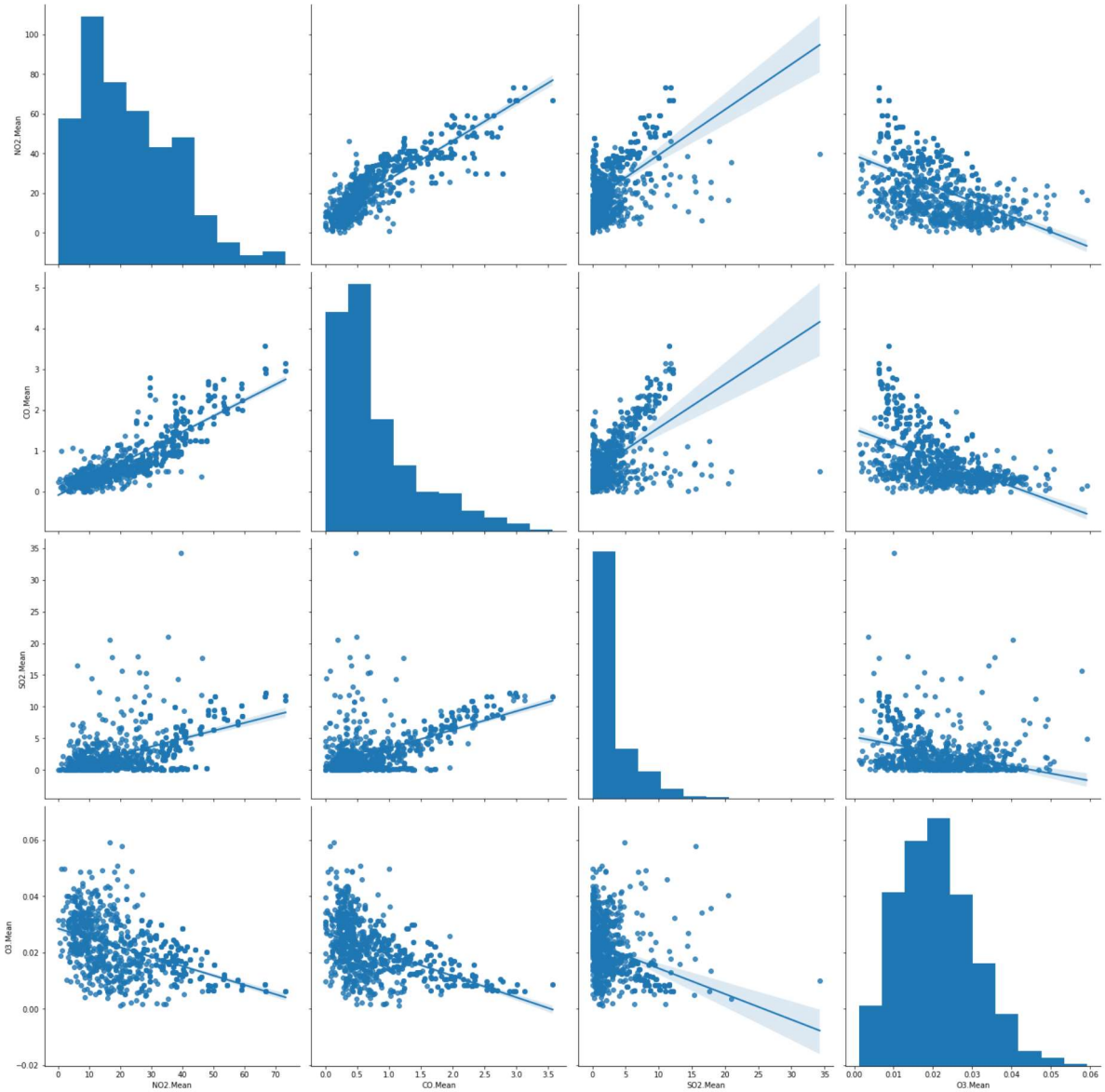
In [20]:

```
sns.pairplot(pollution_data,
             height = 4,
             vars = ["NO2.Mean",
                     "CO.Mean",
                     "SO2.Mean",
                     "O3.Mean"],
             hue = "year")

plt.show()
```



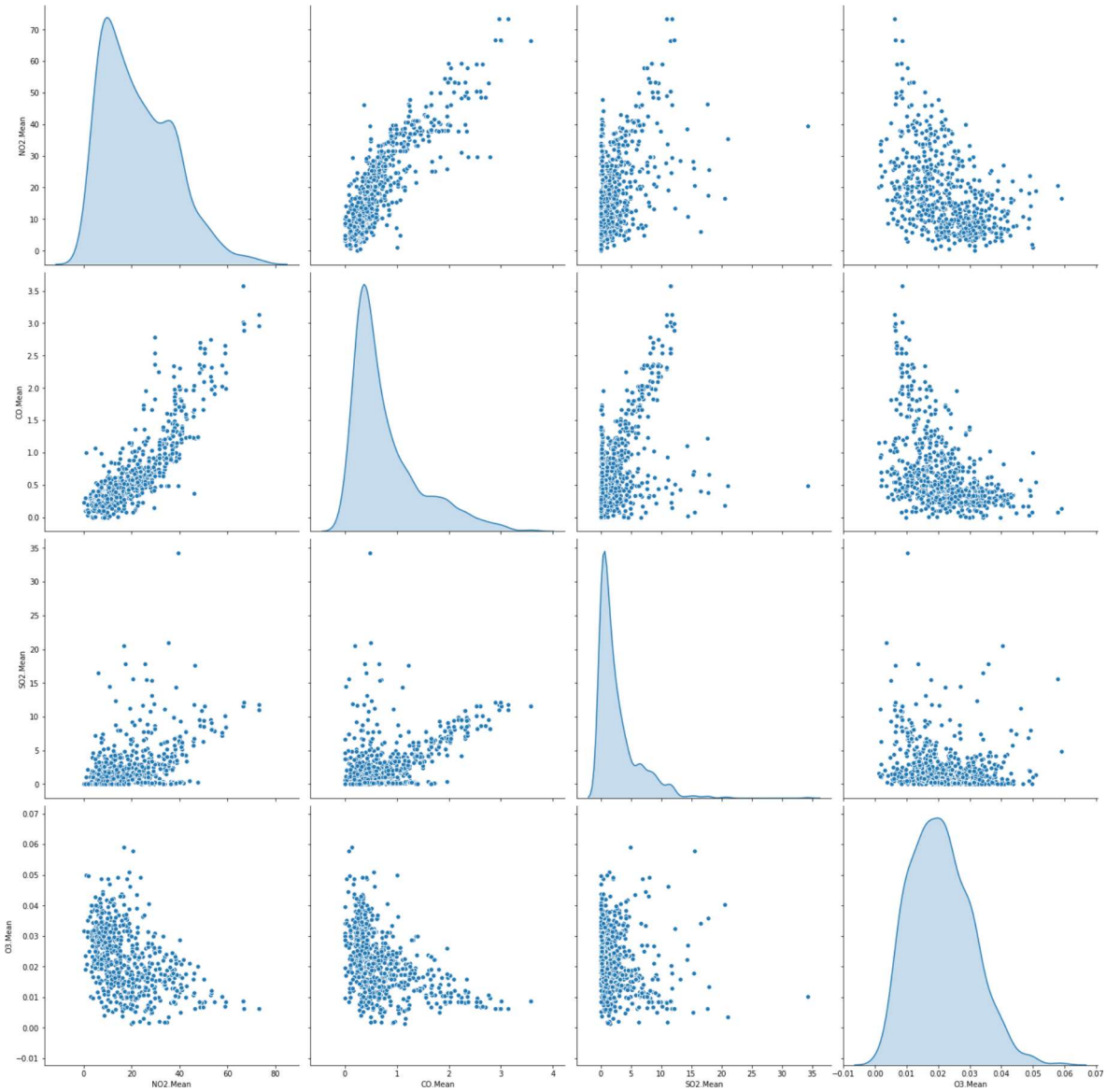
In [21]:

```
sns.pairplot(pollution_data,
             height = 5,
             vars = ["NO2.Mean",
                     "CO.Mean",
                     "SO2.Mean",
                     "O3.Mean"],
              kind = "reg")

plt.show()
```



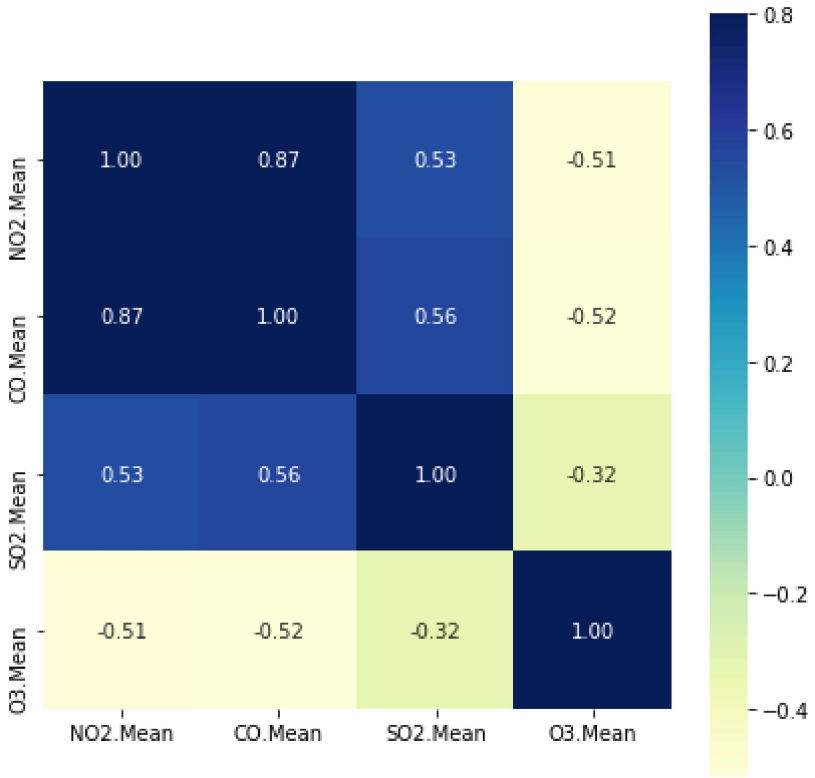
In [23]:

```
sns.pairplot(pollution_data,
             height = 5,
             vars = ["NO2.Mean",
                     "CO.Mean",
                     "SO2.Mean",
                     "O3.Mean"],
             diag_kind = "kde")

plt.show()
```



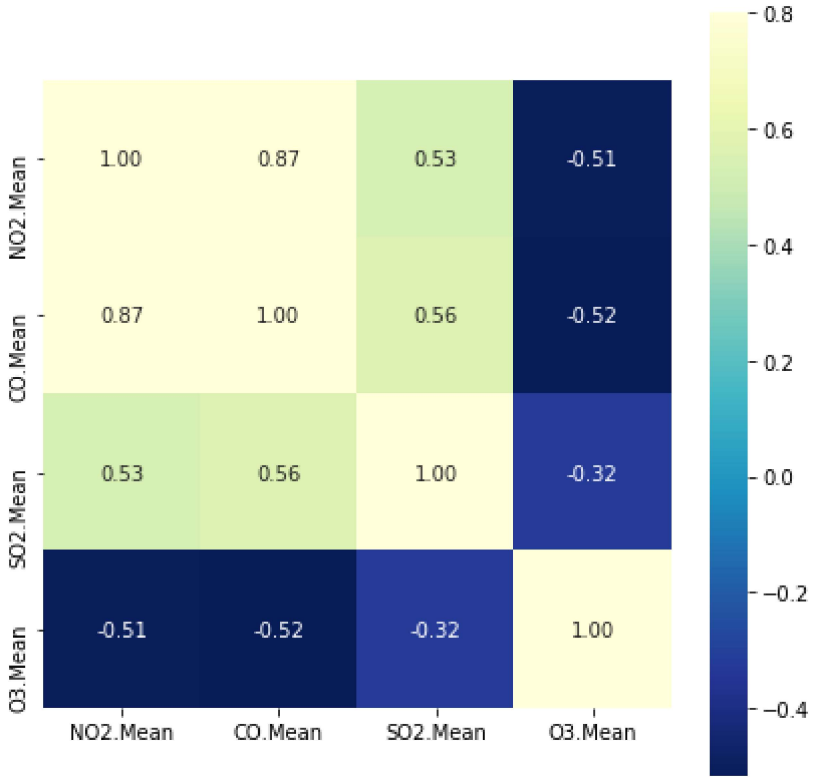
In [29]:

```
corrmatrix = pollutants_mean.corr()
plt.subplots(figsize = (7,7))

sns.heatmap(corrmatrix ,
            vmax =.8,
            square = True,
            annot = True,
            fmt = ".2f",
            cmap = "YlGnBu")

plt.show()
```



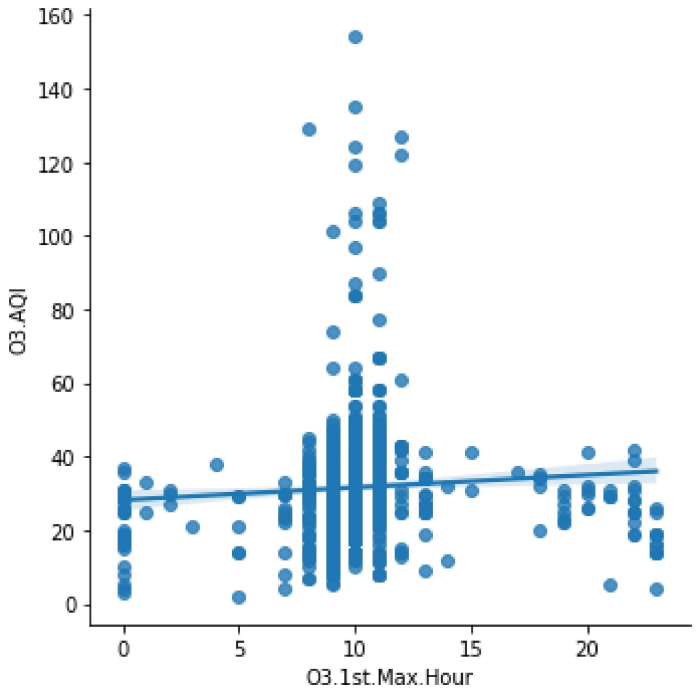
In [30]:

```
corrmatrix = pollutants_mean.corr()
plt.subplots(figsize = (7,7))

sns.heatmap(corrmatrix ,
            vmax =.8,
            square = True,
            annot = True,
            fmt = ".2f",
            cmap = "YlGnBu_r")

plt.show()
```



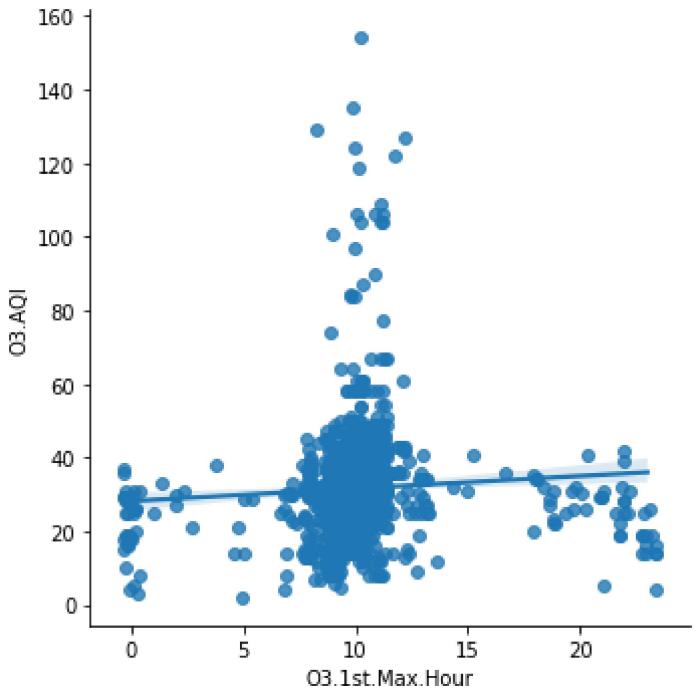
In [41]:

```
sns.lmplot("O3.1st.Max.Hour",
           "O3.AQI",
           data = pollution_data)

plt.show()
```



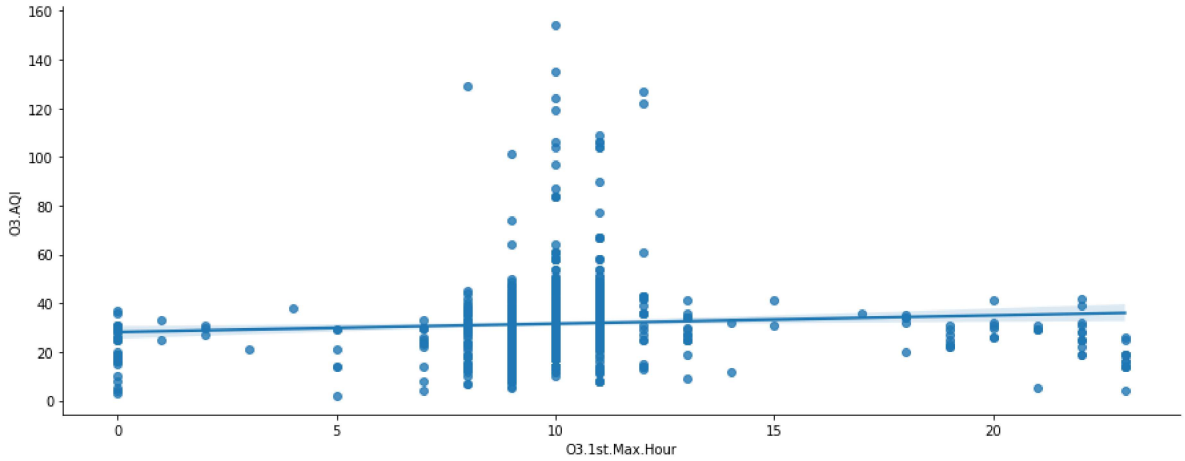
In [42]:

```
sns.lmplot("O3.1st.Max.Hour",
          "O3.AQI",
           x_jitter = .4,
           y_jitter = 0.1,
          data = pollution_data)

plt.show()
```



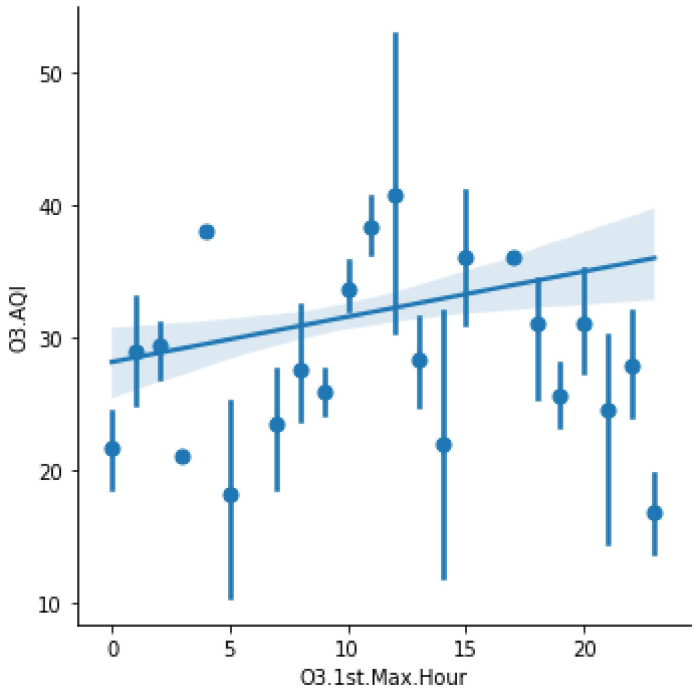
In [44]:

```
sns.lmplot("O3.1st.Max.Hour",
          "O3.AQI",
          data = pollution_data,
           height =5,
           aspect = 2.5)


plt.show()
```



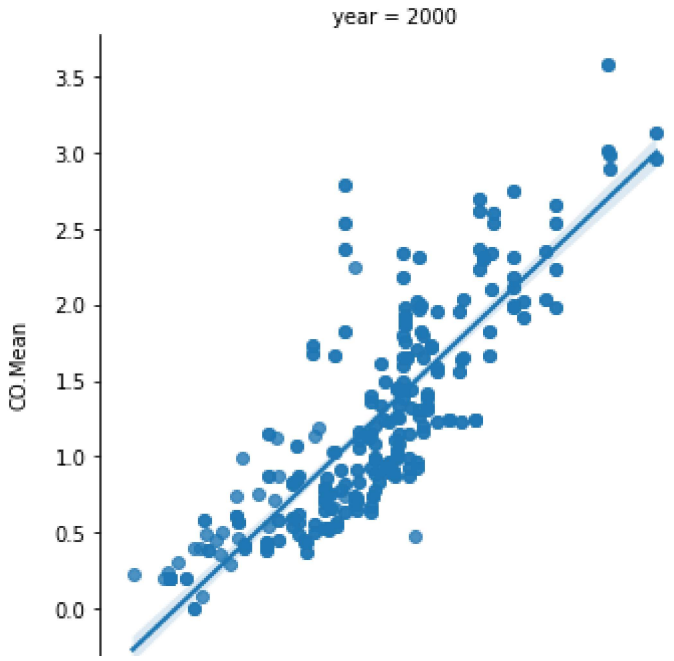
In [45]:

```
sns.lmplot("O3.1st.Max.Hour",
          "O3.AQI",
          data = pollution_data,
           x_estimator = np.mean)


plt.show()
```



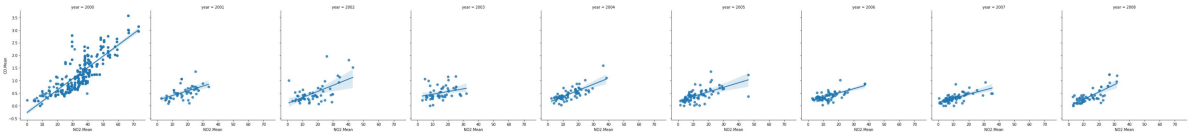
In [46]:

```
sns.lmplot("NO2.Mean",
           "CO.Mean",
           data = pollution_data,
           row = "year")
plt.show()
```



In [48]:

```
sns.lmplot("NO2.Mean",
           "CO.Mean",
           data = pollution_data,
           col = "year")
plt.show()
```



In [49]:

```
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [50]:

```
pollution_data = pd.read_csv('./Downloads/UFOPOLLUTANTS.csv')
pollution_data = pollution_data[:1000]

pollution_data.head()
```

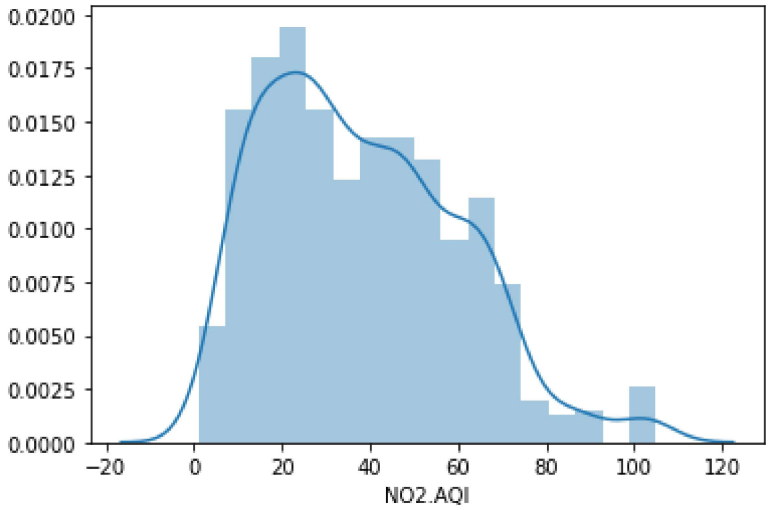
Out[50]:

	State.Code	city	state	day	month	year	hour	NO2.Mean	NO2.1st.Max.Value	NO2.
0	4	kingman (desert&#44 out of)	az	7	7	2000	21	7.375000	15.0	
1	4	tacna (approx)	az	7	7	2000	21	7.375000	15.0	
2	4	apache junction	az	6	9	2001	21	7.708333	19.0	
3	4	sedona	az	5	6	2005	21	7.916667	17.0	
4	4	tucson	az	5	10	2007	2	8.625000	20.0	

5 rows × 24 columns

In [51]:

```
sns.distplot(pollution_data["NO2.AQI"])

plt.show()
```

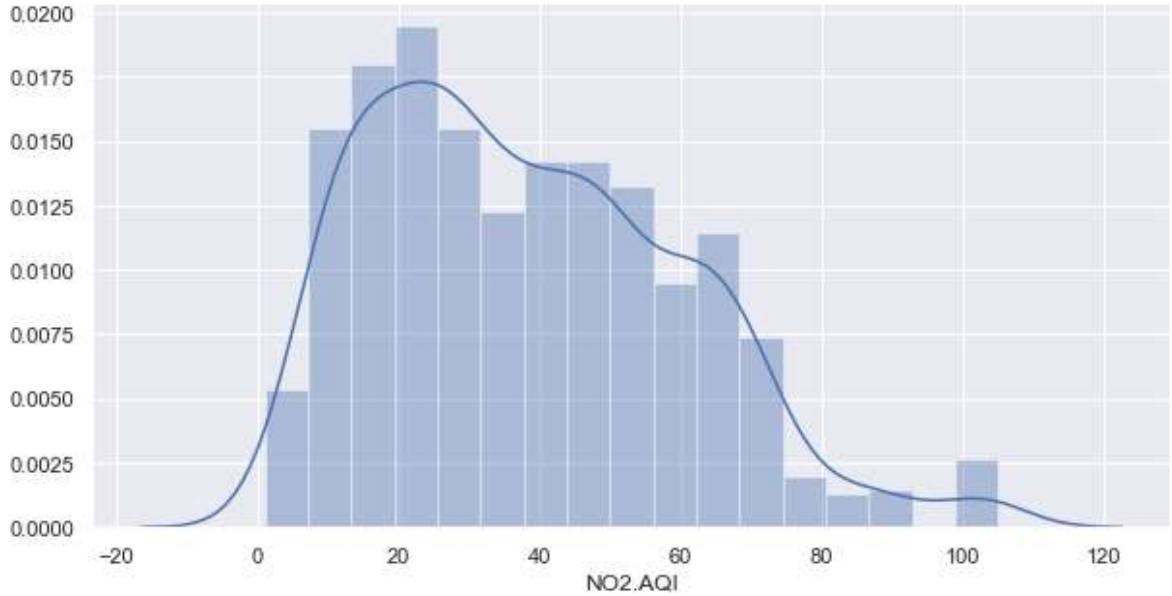


In [52]:

```
sns.set()
```

In [53]:

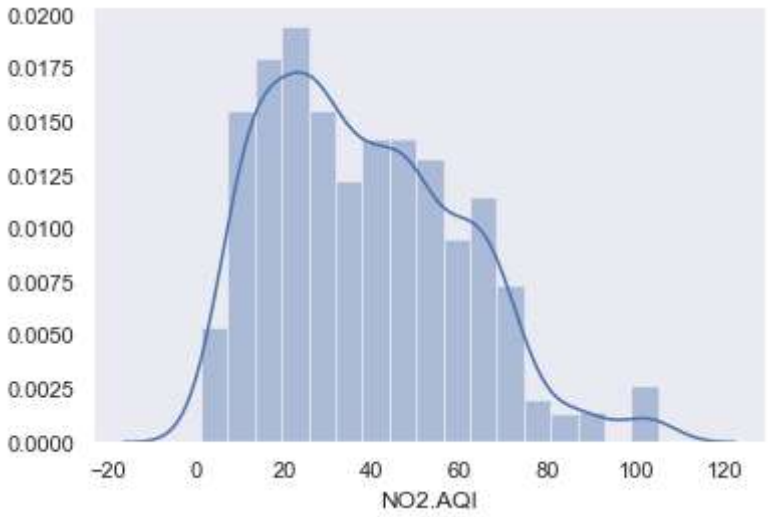
```
fig,axis = plt.subplots(figsize = (10,5))
sns.distplot(pollution_data["NO2.AQI"])

plt.show()
```



In [54]:

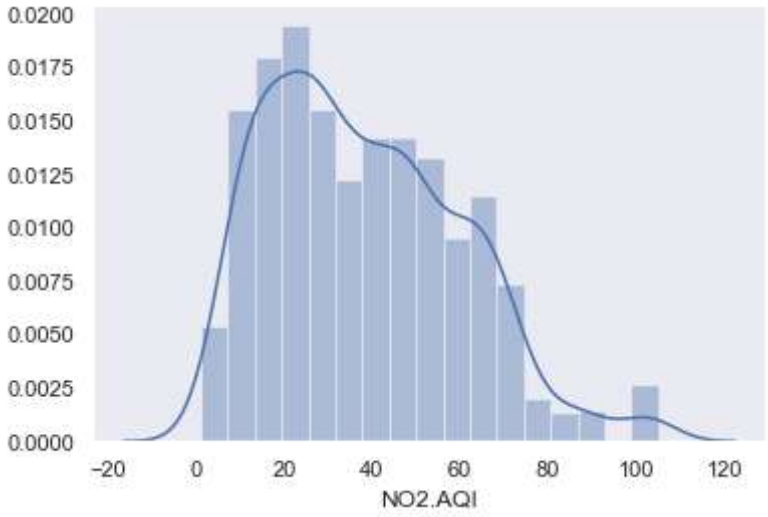
```
sns.set_style("dark")
sns.distplot(pollution_data["NO2.AQI"])

Plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-54-95e9045a3c7b> in <module>
      2 sns.distplot(pollution_data["NO2.AQI"])
      3
----> 4 Plt.show()

NameError: name 'Plt' is not defined
```



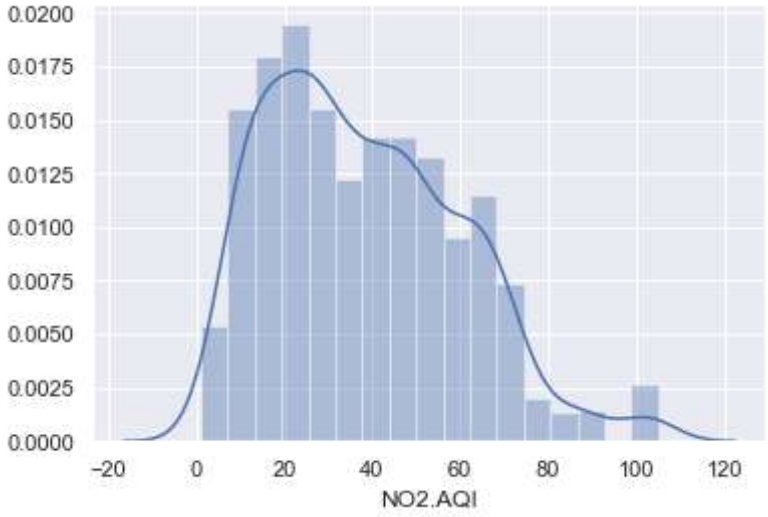
In [55]:

```
sns.set_style("dark")
sns.distplot(pollution_data["NO2.AQI"])

plt.show()
```



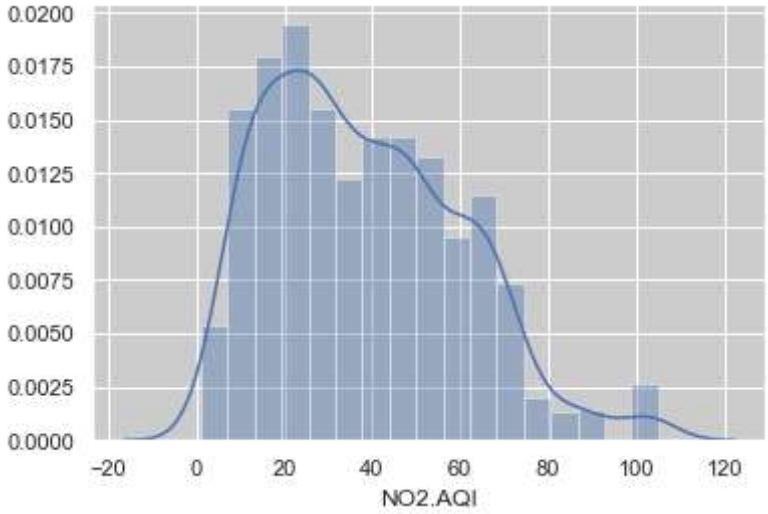
In [56]:

```
sns.set_style("darkgrid")
sns.distplot(pollution_data["NO2.AQI"])

plt.show()
```



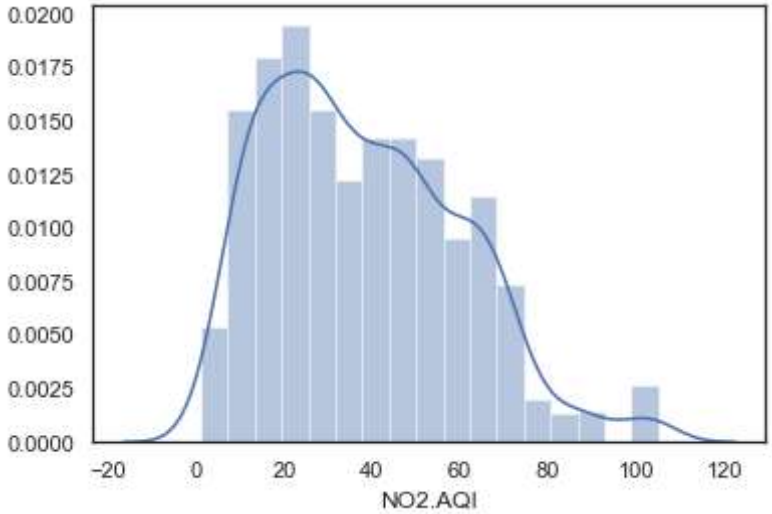
In [57]:

```
sns.set_style("darkgrid",
              {"axes.facecolor": "0.8"})

sns.distplot(pollution_data["NO2.AQI"])

plt.show()
```



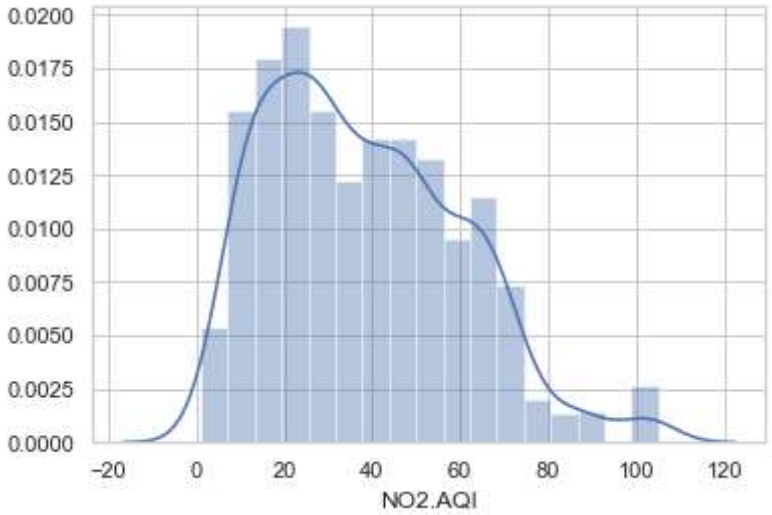
In [58]:

```
sns.set_style("white")
sns.distplot(pollution_data["NO2.AQI"])

plt.show()
```



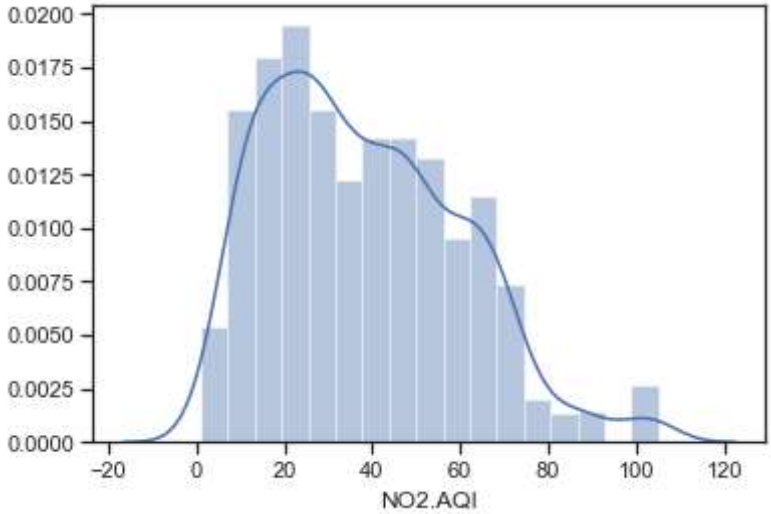
In [59]:

```
sns.set_style("whitegrid")
sns.distplot(pollution_data["NO2.AQI"])

plt.show()
```



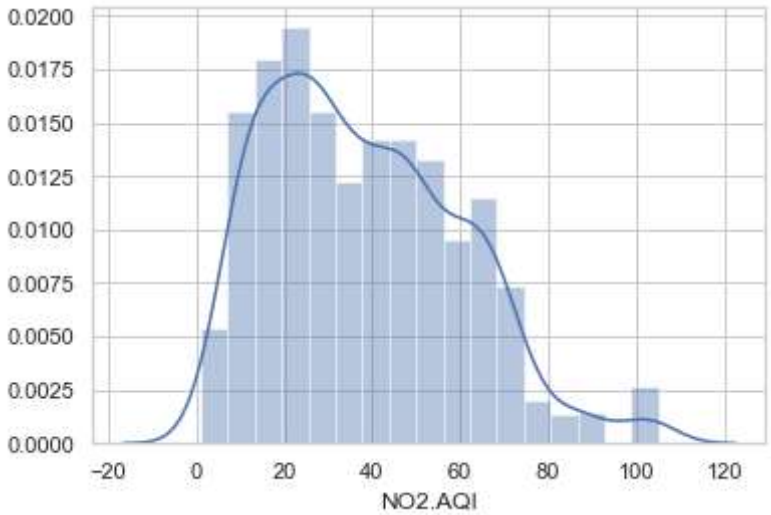
In [60]:

```
sns.set_style("ticks")
sns.distplot(pollution_data["NO2.AQI"])

plt.show()
```



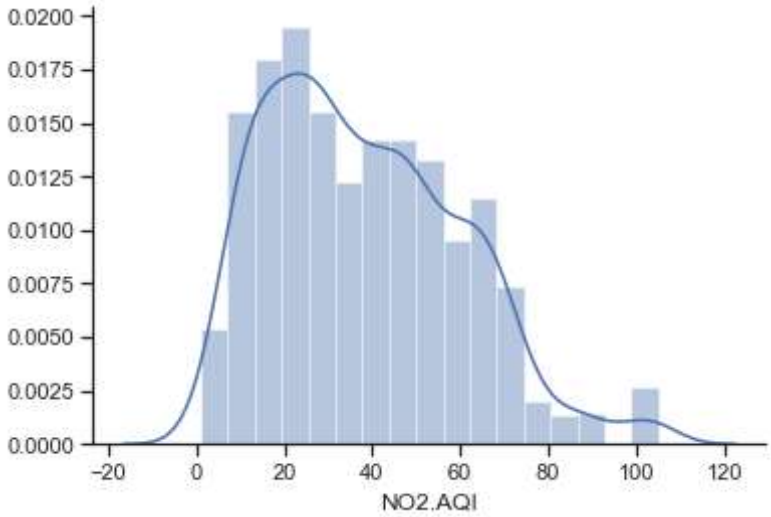
In [61]:

```
with sns.axes_style("whitegrid"):

    sns.distplot(pollution_data["NO2.AQI"])

plt.show()
```



In [62]:

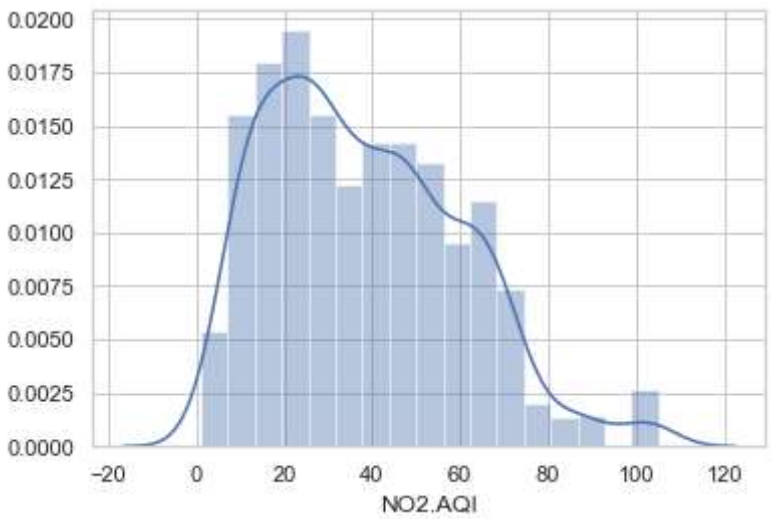
```
sns.distplot(pollution_data["NO2.AQI"])

sns.despine()
sns.set_style("whitegrid")
plt.show()
```



In [63]:

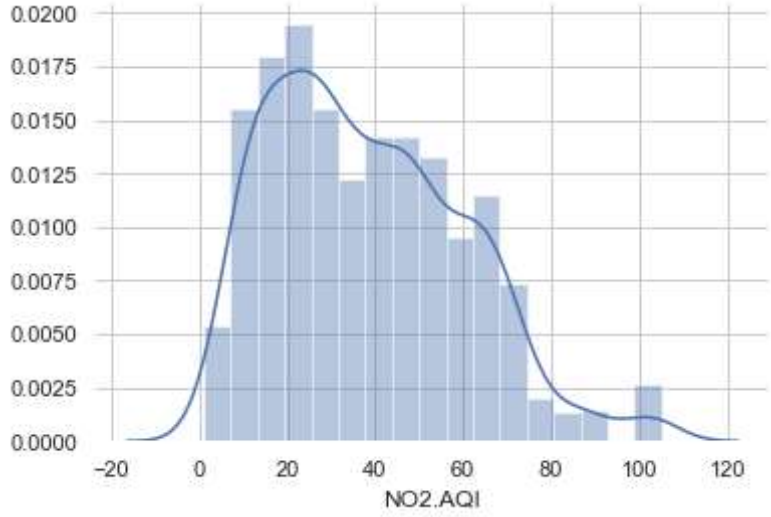
```
sns.distplot(pollution_data["NO2.AQI"])

sns.despine(left = True,
           right= True,
           bottom=True,
           top=true)

plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-63-13b2ec6da11c> in <module>
      4            right= True,
      5            bottom=True,
----> 6            top=true)
      7
      8 plt.show()

NameError: name 'true' is not defined
```



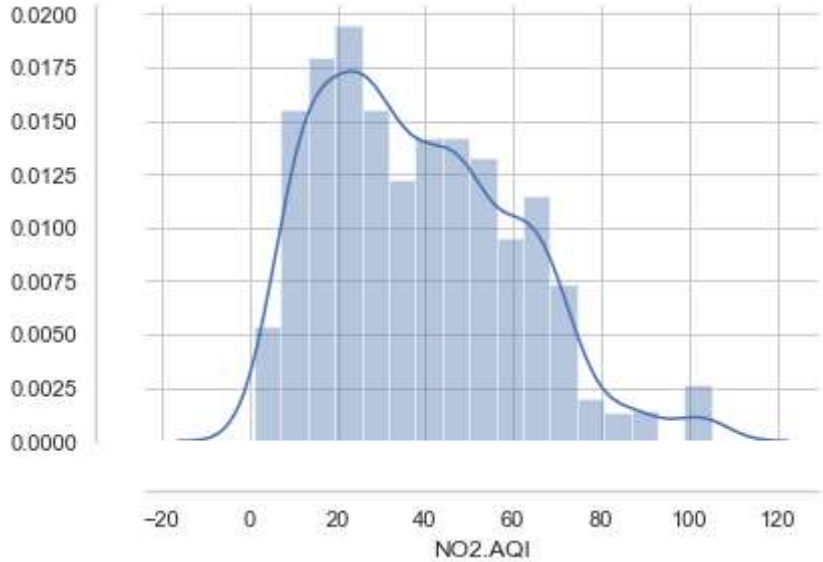
In [64]:

```
sns.distplot(pollution_data["NO2.AQI"])

sns.despine(left = True,
            right= True,
            bottom=True,
            top=True)

plt.show()
```



In [65]:

```
sns.distplot(pollution_data["NO2.AQI"])

sns.despine(offset=25,
            trim=False)

plt.show()
```



In [1]:

```
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [6]:

```
zomato_data = pd.read_csv('./Downloads/zomato.csv')

zomato_data.head
```

Out[6]:

```
<bound method NDFrame.head of       Restaurant ID           Restaurant Name
Country Code              City  \
0           6317637          Le Petit Souffle           162       Makati Cit
y
1           6304287          Izakaya Kikufuji           162       Makati Cit
y
2           6300002    Heat - Edsa Shangri-La           162  Mandaluyong Cit
y
3           6318506                      Ooma           162  Mandaluyong Cit
y
4           6314302               Sambo Kojin           162  Mandaluyong Cit
y
...             ...                       ...           ...
...
9546        5915730               Naml' Gurme           208         ××stanb
ul
9547        5908749              Ceviz A××ac'           208         ××sta
nbul
9548        5915807                     Huqqa           208         ××stanb
ul
9549        5916112               A×××k Kahve           208         ××sta
nbul
9550        5927402  Walter's Coffee Roastery           208         ××stanb
ul

                                                Address  \
0     Third Floor, Century City Mall, Kalayaan Avenu...
1     Little Tokyo, 2277 Chino Roces Avenue, Legaspi...
2     Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...
3     Third Floor, Mega Fashion Hall, SM Megamall, O...
4     Third Floor, Mega Atrium, SM Megamall, Ortigas...
...                                                 ...
9546  Kemanke×× Karamustafa Pa××a Mahallesi, R'ht'm ...
9547  Ko××uyolu Mahallesi, Muhittin ××st×_nda×× Cadd...
9548  Kuru×_e××me Mahallesi, Muallim Naci Caddesi, N...
9549  Kuru×_e××me Mahallesi, Muallim Naci Caddesi, N...
9550  Cafea××a Mahallesi, Bademalt' Sokak, No 21/B, ...

                                       Locality  \
0      Century City Mall, Poblacion, Makati City
1     Little Tokyo, Legaspi Village, Makati City
2     Edsa Shangri-La, Ortigas, Mandaluyong City
3         SM Megamall, Ortigas, Mandaluyong City
4         SM Megamall, Ortigas, Mandaluyong City
...                                          ...
9546                                  Karak×_y
9547                                 Ko××uyolu
9548                               Kuru×_e××me
9549                               Kuru×_e××me
9550                                      Moda

                                     Locality Verbose   Longitude  \
0     Century City Mall, Poblacion, Makati City, Mak...  121.027535
```

```
1     Little Tokyo, Legaspi Village, Makati City, Ma...  121.014101
2     Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...  121.056831
3     SM Megamall, Ortigas, Mandaluyong City, Mandal...  121.056475
4     SM Megamall, Ortigas, Mandaluyong City, Mandal...  121.057508
...                                                 ...         ...
9546                            Karak►_y, ►►stanbul   28.977392
9547                           Ko►►uyolu, ►►stanbul   29.041297
9548                         Kuru►_e►►me, ►►stanbul   29.034640
9549                         Kuru►_e►►me, ►►stanbul   29.036019
9550                                Moda, ►►stanbul   29.026016

       Latitude                          Cuisines  ...          Currency  \
0     14.565443        French, Japanese, Desserts  ...  Botswana Pula(P)
1     14.553708                          Japanese  ...  Botswana Pula(P)
2     14.581404  Seafood, Asian, Filipino, Indian  ...  Botswana Pula(P)
3     14.585318                   Japanese, Sushi  ...  Botswana Pula(P)
4     14.584450                  Japanese, Korean  ...  Botswana Pula(P)
...         ...                               ...  ...               ...
9546  41.022793                           Turkish  ...  Turkish Lira(TL)
9547  41.009847   World Cuisine, Patisserie, Cafe  ...  Turkish Lira(TL)
9548  41.055817            Italian, World Cuisine  ...  Turkish Lira(TL)
9549  41.057979                   Restaurant Cafe  ...  Turkish Lira(TL)
9550  40.984776                              Cafe  ...  Turkish Lira(TL)

     Has Table booking Has Online delivery Is delivering now  \
0                  Yes                  No                No
1                  Yes                  No                No
2                  Yes                  No                No
3                   No                  No                No
4                  Yes                  No                No
...                ...                 ...               ...
9546                No                  No                No
9547                No                  No                No
9548                No                  No                No
9549                No                  No                No
9550                No                  No                No

     Switch to order menu Price range  Aggregate rating  Rating color  \
0                      No           3               4.8    Dark Green
1                      No           3               4.5    Dark Green
2                      No           4               4.4         Green
3                      No           4               4.9    Dark Green
4                      No           4               4.8    Dark Green
...                   ...         ...               ...           ...
9546                   No           3               4.1         Green
9547                   No           3               4.2         Green
9548                   No           4               3.7        Yellow
9549                   No           4               4.0         Green
9550                   No           2               4.0         Green

     Rating text Votes
0      Excellent   314
1      Excellent   591
2      Very Good   270
3      Excellent   365
4      Excellent   229
...          ...   ...
9546   Very Good   788
9547   Very Good  1034
9548        Good   661
9549   Very Good   901
```

```
9550   Very Good   591

[9551 rows x 21 columns]>
```

In [7]:

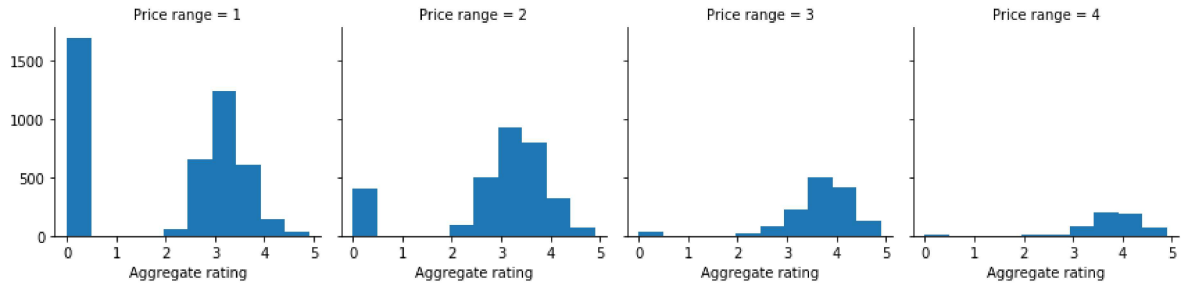
```
zomato_data.describe()
```

Out[7]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Ag
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4

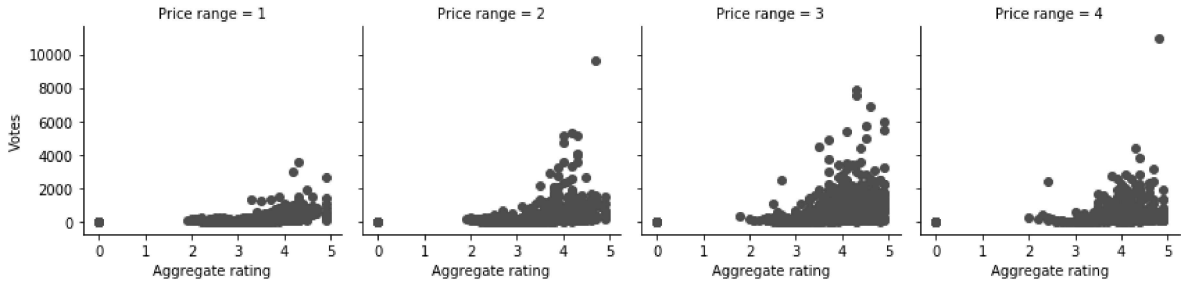
In [11]:

```
facet =sns.FacetGrid(zomato_data,
                     col = "Price range")

facet.map(plt.hist,
          "Aggregate rating")

plt.show()
```



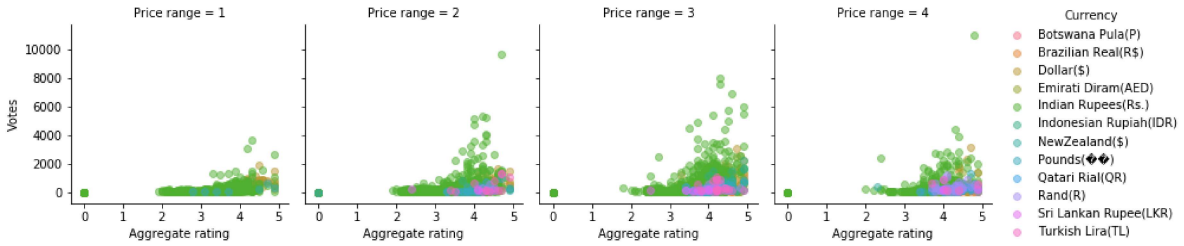
In [13]:

```
facet =sns.FacetGrid(zomato_data,
                     col = "Price range")

facet.map(plt.scatter ,
          "Aggregate rating",
          "Votes",
          color = "0.3")

plt.show()
```



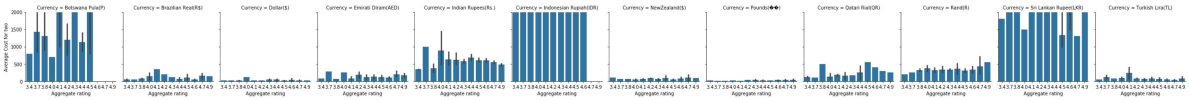
In [14]:

```
facet =sns.FacetGrid(zomato_data,
                     col = "Price range",
                     hue = "Currency")

facet.map(plt.scatter ,
          "Aggregate rating",
          "Votes",
          alpha = 0.5)

facet.add_legend()

plt.show()
```



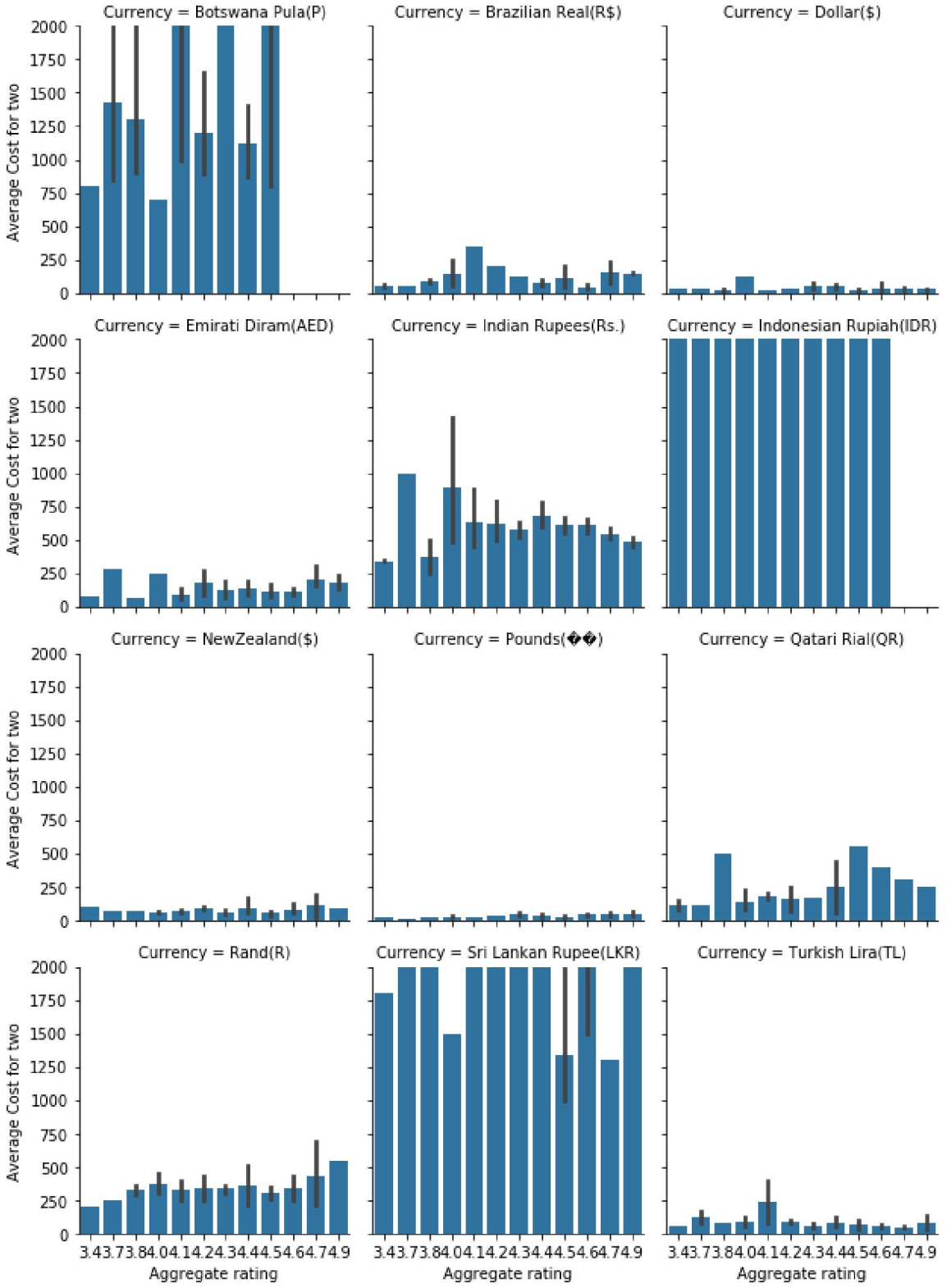
In [19]:

```
facet =sns.FacetGrid(zomato_data,
                     col = "Currency")



facet.map(sns.barplot ,
         "Aggregate rating",
         "Average Cost for two")

plt.ylim(0,2000)

plt.show()
```



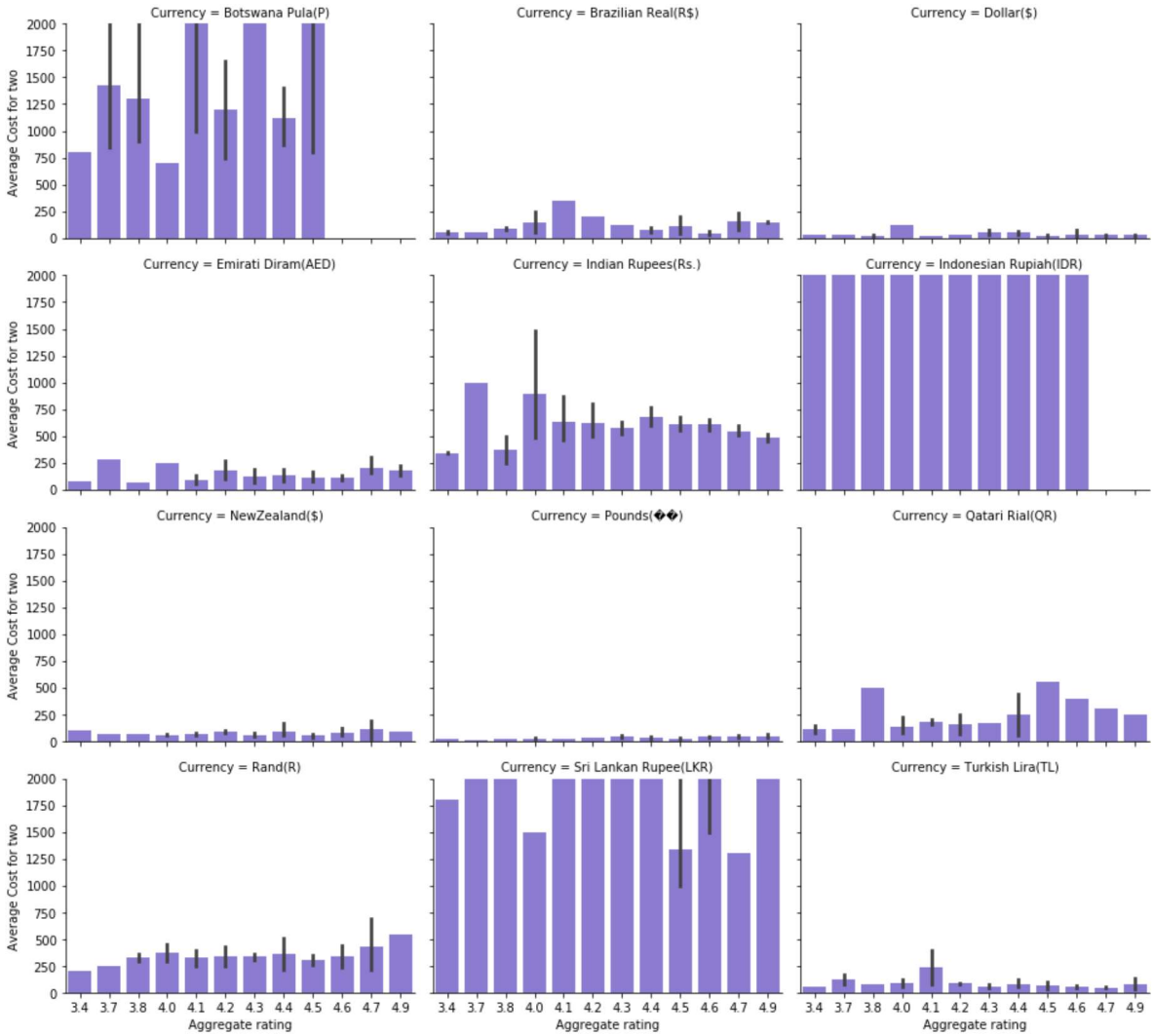
In [20]:

```
facet =sns.FacetGrid(zomato_data,
                     col = "Currency",
                     col_wrap = 3)



facet.map(sns.barplot ,
         "Aggregate rating",
         "Average Cost for two")

plt.ylim(0,2000)

plt.show()
```


Currency = Botswana Pula(P)
Currency = Brazilian Real(R$)
Currency = Dollar($)
Currency = Emirati Diram(AED)
Currency = Indian Rupees(Rs.)
Currency = Indonesian Rupiah(IDR)
Currency = NewZealand($)
Currency = Pounds(►►)
Currency = Qatari Rial(QR)
Currency = Rand(R)
Currency = Sri Lankan Rupee(LKR)
Currency = Turkish Lira(TL)
Average Cost for two
0
250
500
750
1000
1250
1500
1750
2000
3.4 3.7 3.8 4.0 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.9
Aggregate rating

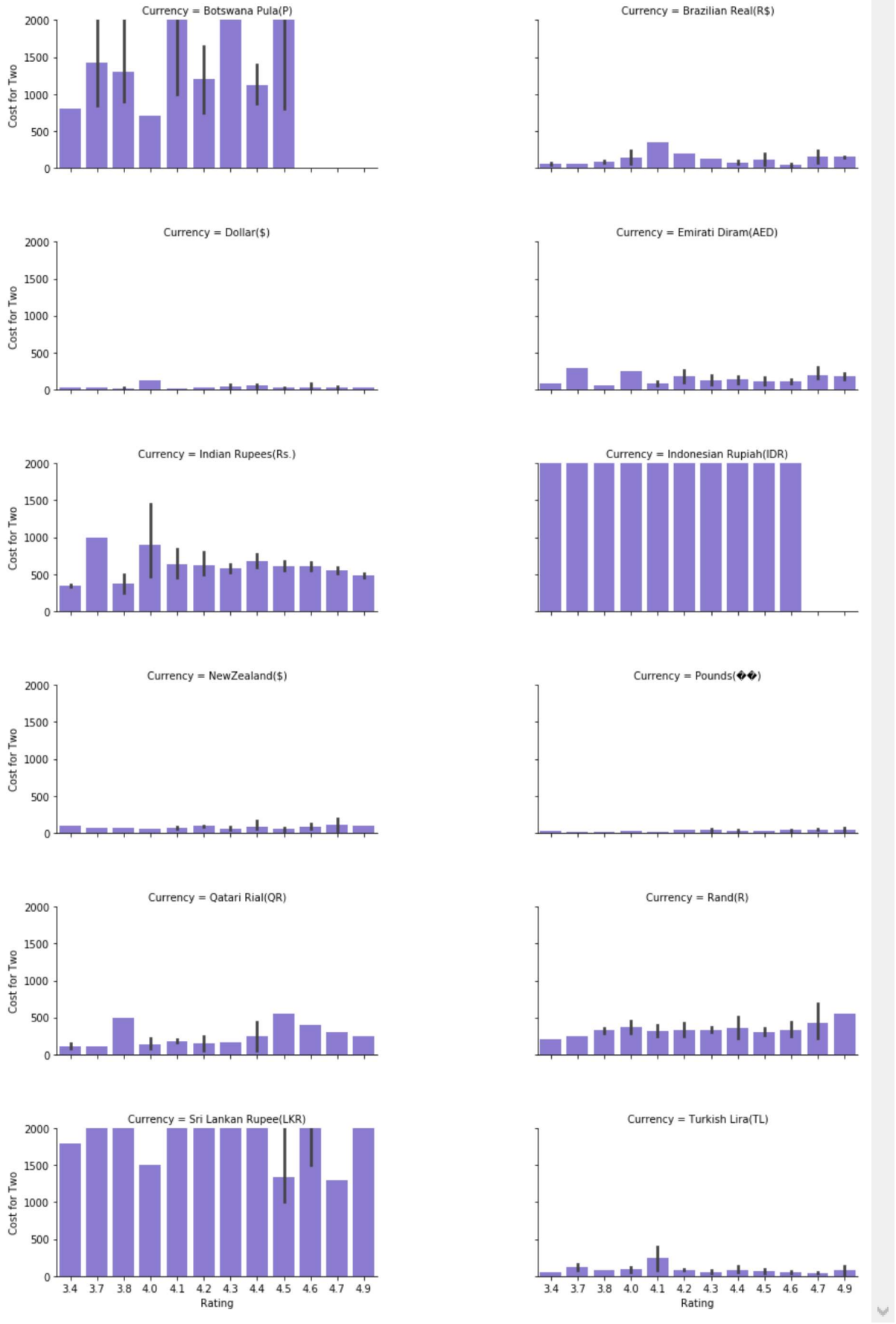
In [21]:

```
facet =sns.FacetGrid(zomato_data,
                     col = "Currency",
                     col_wrap = 3,
                     aspect = 1.5)



facet.map(sns.barplot ,
         "Aggregate rating",
         "Average Cost for two",
         color = '#826EDF')

plt.ylim(0,2000)

plt.show()
```



In [22]:

```
facet =sns.FacetGrid(zomato_data,
                     col = "Currency",
                     col_wrap = 2,
                     aspect = 2)



facet.map(sns.barplot ,
         "Aggregate rating",
         "Average Cost for two",
         color = '#826EDF')

facet.fig.subplots_adjust(wspace = 0.5,
                          hspace = 0.5)

facet.set_axis_labels("Rating",
                      "Cost for Two")

plt.ylim(0,2000)

plt.show()
```


Currency = Botswana Pula(P)
Currency = Brazilian Real(R$)
Currency = Dollar($)
Currency = Emirati Diram(AED)
Currency = Indian Rupees(Rs.)
Currency = Indonesian Rupiah(IDR)
Currency = NewZealand($)
Currency = Pounds(►►)
Currency = Qatari Rial(QR)
Currency = Rand(R)
Currency = Sri Lankan Rupee(LKR)
Currency = Turkish Lira(TL)
Cost for Two
0
500
1000
1500
2000
3.4 3.7 3.8 4.0 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.9
Rating

In [23]:

```
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [25]:

```
pollution_data = pd.read_csv('./Downloads/UFOPOLLUTANTS.csv')

pollution_data = pollution_data[:1000]

pollution_data.head()
```

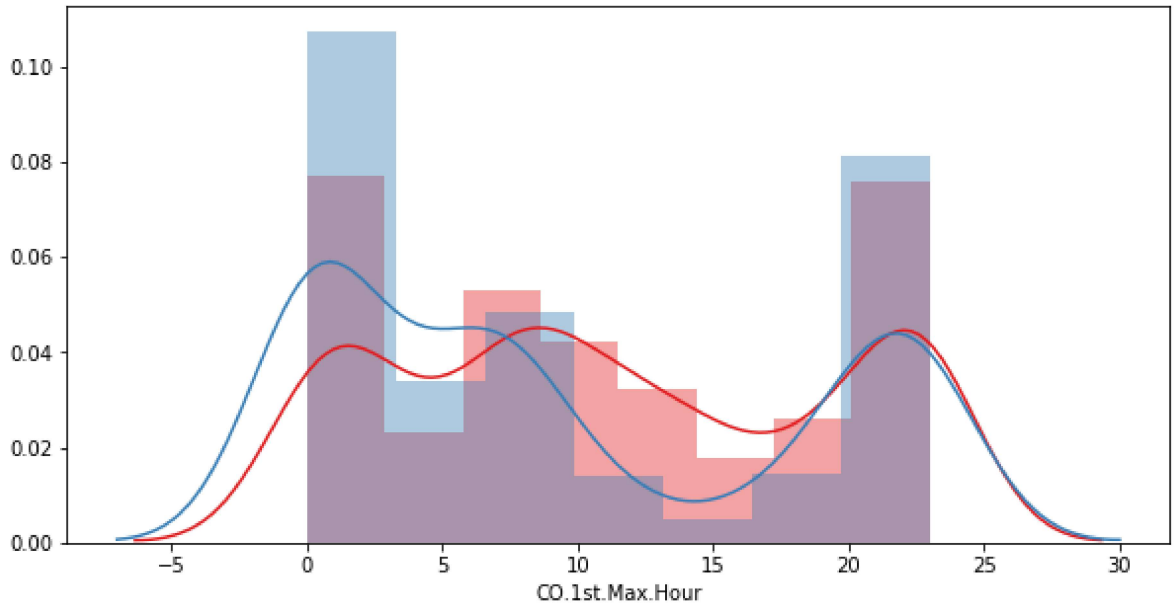
Out[25]:

	State.Code	city	state	day	month	year	hour	NO2.Mean	NO2.1st.Max.Value	NO2.
0	4	kingman (desert&#44 out of)	az	7	7	2000	21	7.375000	15.0	
1	4	tacna (approx)	az	7	7	2000	21	7.375000	15.0	
2	4	apache junction	az	6	9	2001	21	7.708333	19.0	
3	4	sedona	az	5	6	2005	21	7.916667	17.0	
4	4	tucson	az	5	10	2007	2	8.625000	20.0	

5 rows × 24 columns

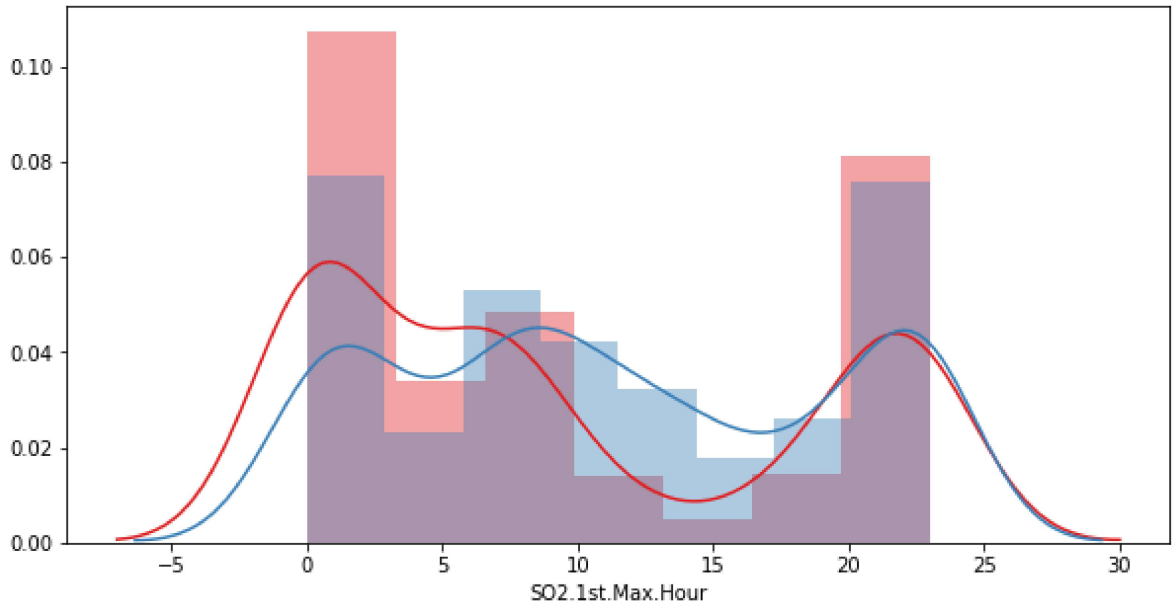
In [26]:

```
sns.set_palette("Set1")

fig, axis = plt.subplots(figsize = (10,5))

sns.distplot(pollution_data["SO2.1st.Max.Hour"])

sns.distplot(pollution_data["CO.1st.Max.Hour"])

plt.show()
```



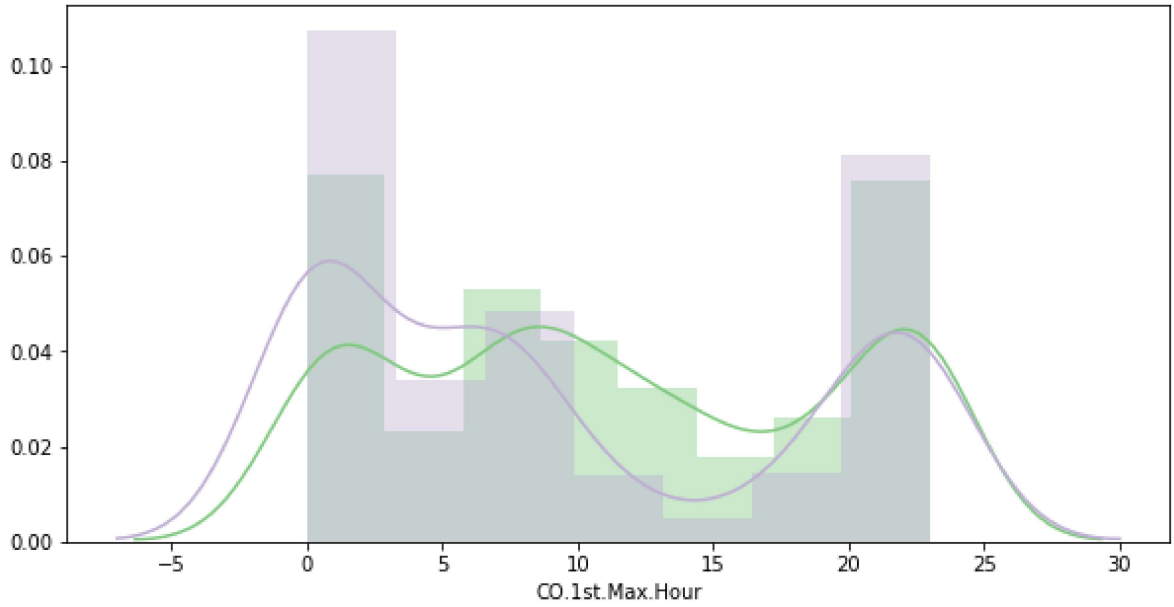
In [27]:

```
fig, axis = plt.subplots(figsize = (10,5))

sns.distplot(pollution_data["CO.1st.Max.Hour"])

sns.distplot(pollution_data["SO2.1st.Max.Hour"])

plt.show()
```



In [28]:

```
sns.set_palette("Accent")

fig, axis = plt.subplots(figsize = (10,5))

sns.distplot(pollution_data["SO2.1st.Max.Hour"])

sns.distplot(pollution_data["CO.1st.Max.Hour"])

plt.show()
```



In [29]:

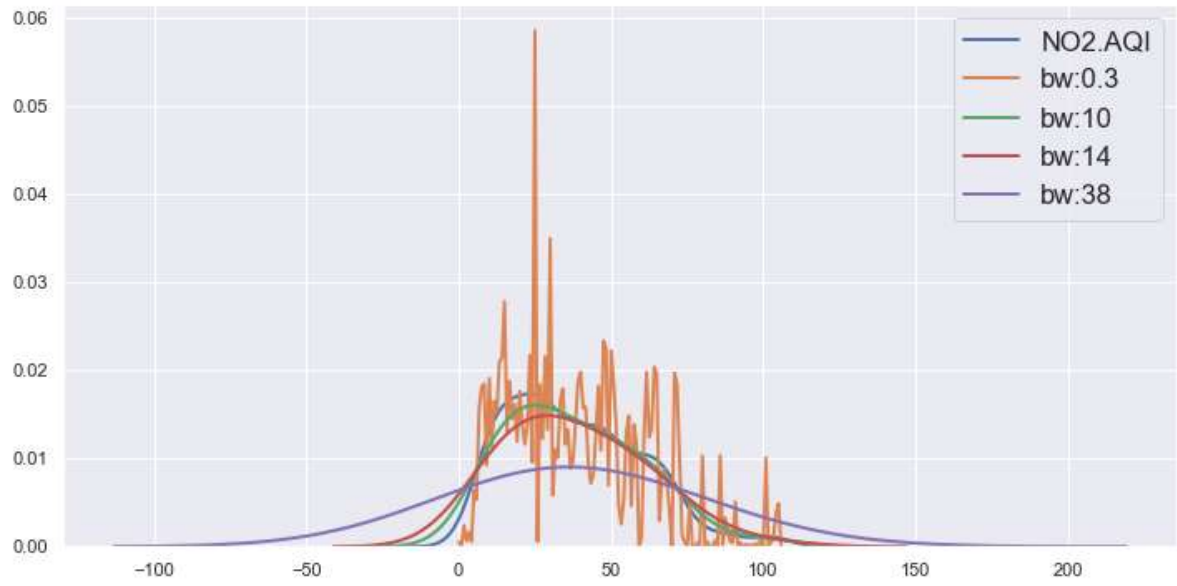
```
sns.set()
```

In [32]:

```
current_palette = sns.color_palette()

sns.palplot(current_palette)

plt.show()
```

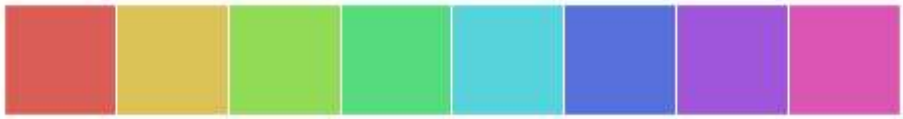
In [36]:

```
hist = plt.subplots(figsize = (12, 6))

with plt.rc_context(sns.plotting_context("notebook",
                                         font_scale = 1.5 ,
                                         rc = {"lines.linewidth": 2})):

    sns.kdeplot(pollution_data["NO2.AQI"])


    sns.kdeplot(pollution_data["NO2.AQI"],
                bw = 0.3 ,
                label = "bw:0.3")


    sns.kdeplot(pollution_data["NO2.AQI"],
                bw = 10,
                label = "bw:10")

    sns.kdeplot(pollution_data["NO2.AQI"],
                bw = 14,
                label = "bw:14")

    sns.kdeplot(pollution_data["NO2.AQI"],
                bw = 38,
                label = "bw:38")

plt.show()
```



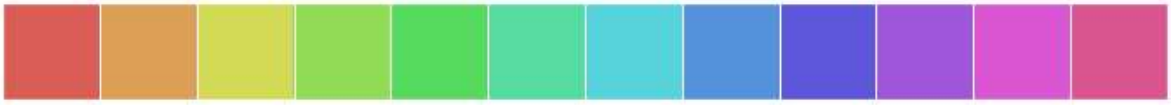
In [37]:

```
spec_num_color_palette = sns.palplot(sns.color_palette("hls", 8))

plt.show()
```



In [38]:

```
spec_num_color_palette = sns.palplot(sns.color_palette("hls", 12))

plt.show()
```



In [39]:

```
sns.palplot(sns.hls_palette(8,
                            l = 0.3,
                            s = 0.8))

plt.show()
```

In [40]:

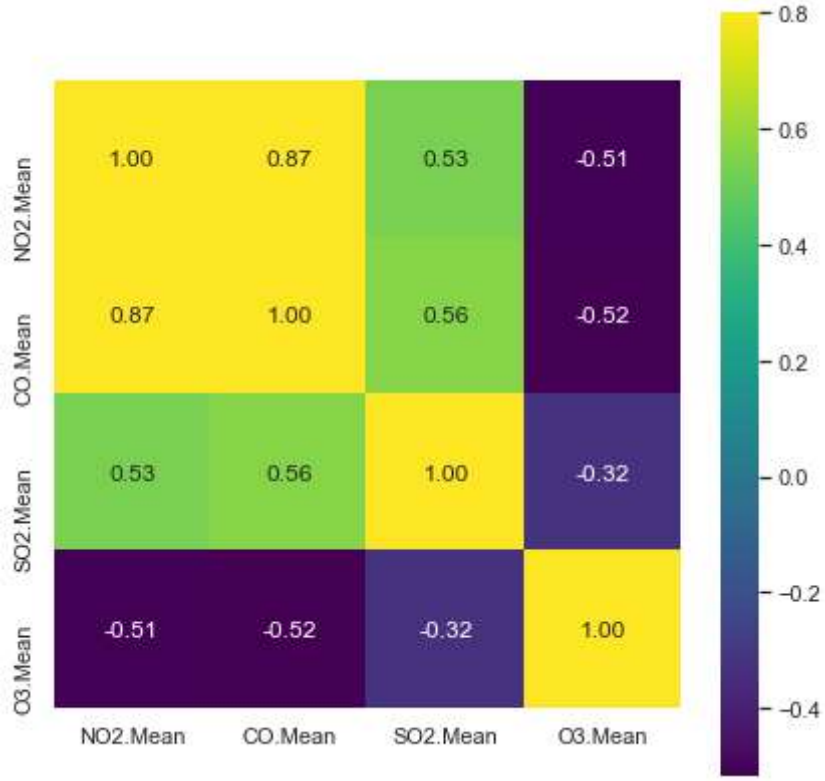
```
sns.palplot(sns.color_palette("viridis"))

plt.show()
```

In [41]:

```
pollutants_mean = pollution_data[['NO2.Mean',
                                  'CO.Mean',
                                  'SO2.Mean',
                                  'O3.Mean']]
```

In [42]:

```
corrmatrix = pollutants_mean.corr()

plt.subplots(figsize = (7,7))

sns.heatmap(corrmatrix ,
            vmax = .8,
            square = True,
            annot = True,
            fmt = ".2f",
            cmap = "viridis")

plt.show()
```



In [43]:

```
sns.palplot(sns.color_palette(["#5b59c6",
                               "#1918D1",
                               "#D32BB9",
                               "#D30E10",
                               "#4AD122"]))

plt.show()
```