FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
Yazılım Mühendisliği Bölümü

YMH418 – Yaz. Müh. Güncel Konular

Doc.Dr. Fatih ÖZKAYNAK

Veri Bilimi

Rapor-2

15542507 – Neslihan KOLUKISA

Nisan – 2020

İçerik:

Eksik Değerler

- Eksik Değerlerin Bulunması
- Eksik Değerlerin Doldurulması

Eksik Değerlerin Bulunması

In [2]:

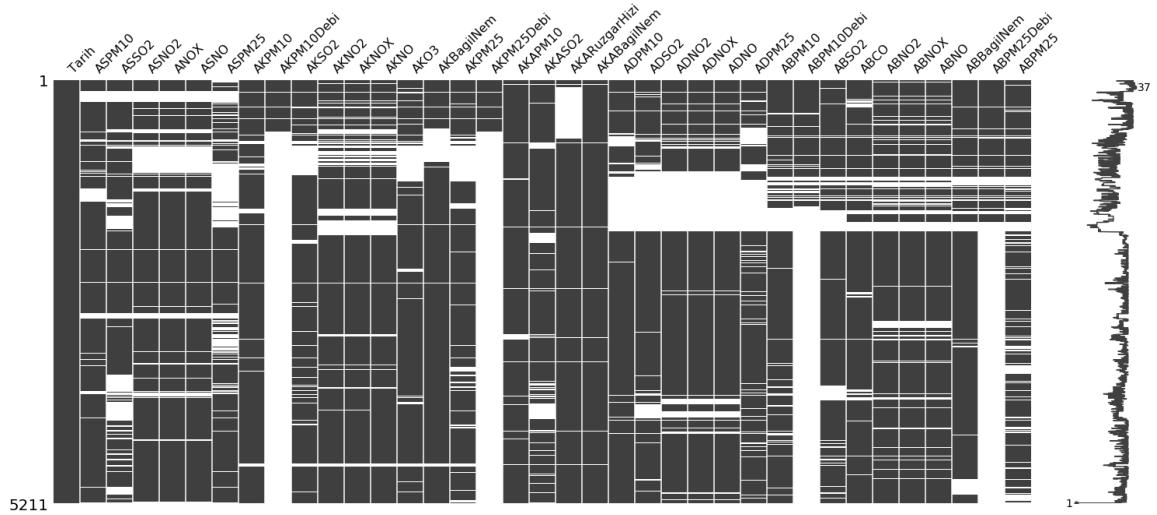
```
import pandas as pd
import numpy as np
kentsel_df = pd.read_csv("kentsel.csv")
kentsel_df.isnull().sum() #Parametrelerin ne kadar boş veri içerdiği görülmektedir.
```

Out[2]:

```
Tarih              0
ASPM10           553
ASSO2           1371
ASNO2            918
ANOX             919
ASNO             919
ASPM25          1654
AKPM10           327
AKPM10Debi      4615
AKSO2            727
AKNO2            779
AKNOX            780
AKNO             776
AKO3             766
AKBagilNem       515
AKPM25           948
AKPM25Debi      4617
AKAPM10          194
AKASO2           695
AKARuzgarHizi    706
AKABagilNem       71
ADPM10           923
ADSO2           1167
ADNO2           1122
ADNOX           1121
ADNO            1121
ADPM25          1366
ABPM10           808
ABPM10Debi      3935
ABSO2            821
ABCO             633
ABNO2            829
ABNOX            836
ABNO             836
ABBagilNem       650
ABPM25Debi      3773
ABPM25          1196
dtype: int64
```

In [112]:

```
import missingno as msno
data_missingno = pd.read_csv("kentsel.csv")
msno.matrix(data_missingno)
plt.show()
```



Yukarıda ki grafik boş veri içermeyen değerleri siyah bir bar ile göstermektedir. Eğer nan bir değer var ise beyaz bir bar ile göstermektedir. Yukarı da verilen sayısal verilerden ve ayrıca grafikten de görüldüğü üzere en çok Ankara Keçiören Sanatoryum istasyonunda ölçülen PM10 Debi değeri boş veri içermektedir.

Eksik Değerlerin Doldurulması

Zamandan bağımsız verilerde (zaman serisi olmayan), yaygın bir uygulama, boşlukları alanın ortalama veya medyan değeri ile doldurmaktır. Ancak, bu zaman serilerinde geçerli değildir. Nedeni anlamak için bir sıcaklık veri kümesini düşünelim. Şubat ayı sıcaklık değeri Temmuz ayı değerinden çok uzak. Bu durum, bazı sezonları yüksek satışları olan, düşük veya düzenli satışları olan satış veri kümeleri için de geçerlidir. Bu nedenle, empütasyon yöntemi zamana bağlı olmalıdır.
Bu varsayımı kanıtlamak için, veri seti üzerinde ortalama,medyan ve diğer zaman serilerine yönelik yöntemlerle boş veriler doldurulacak ardından en yüksek puana göre yöntem seçilecektir.

In [117]:

```
kentsel_df.head(1)
```

Out[117]:

	Tarih	ASPM10	ASSO2	ASNO2	ANOX	ASNO	ASPM25	AKPM10	AKPM10Debi	AKSO2
0	2019-12-31	34.37	5.07	43.07	65.83	22.76	NaN	12.71	0.83	5.88

1 rows × 37 columns

Ortalama / medyan ve hareketli ortalama / hareketli medyan değerleri kullanarak impute etmeye çalışmak

In [4]:

```
ASPM10_df_null = pd.DataFrame(kentsel_df['ASPM10'].tolist())
ASPM10_df_fill_mean = ASPM10_df_null.fillna(ASPM10_df_null.mean())
ASPM10_df_fill_median = ASPM10_df_null.fillna(ASPM10_df_null.median())
ASPM10_df_fill_rollingMean = ASPM10_df_null.fillna(ASPM10_df_null.rolling(24,min_period
s=1,).mean())
ASPM10_df_fill_rollingMedian = ASPM10_df_null.fillna(ASPM10_df_null.rolling(24,min_peri
ods=1,).median())
```

Farklı yöntemlerle enterpolasyon kullanarak impute etmeye çalışmak

In [5]:

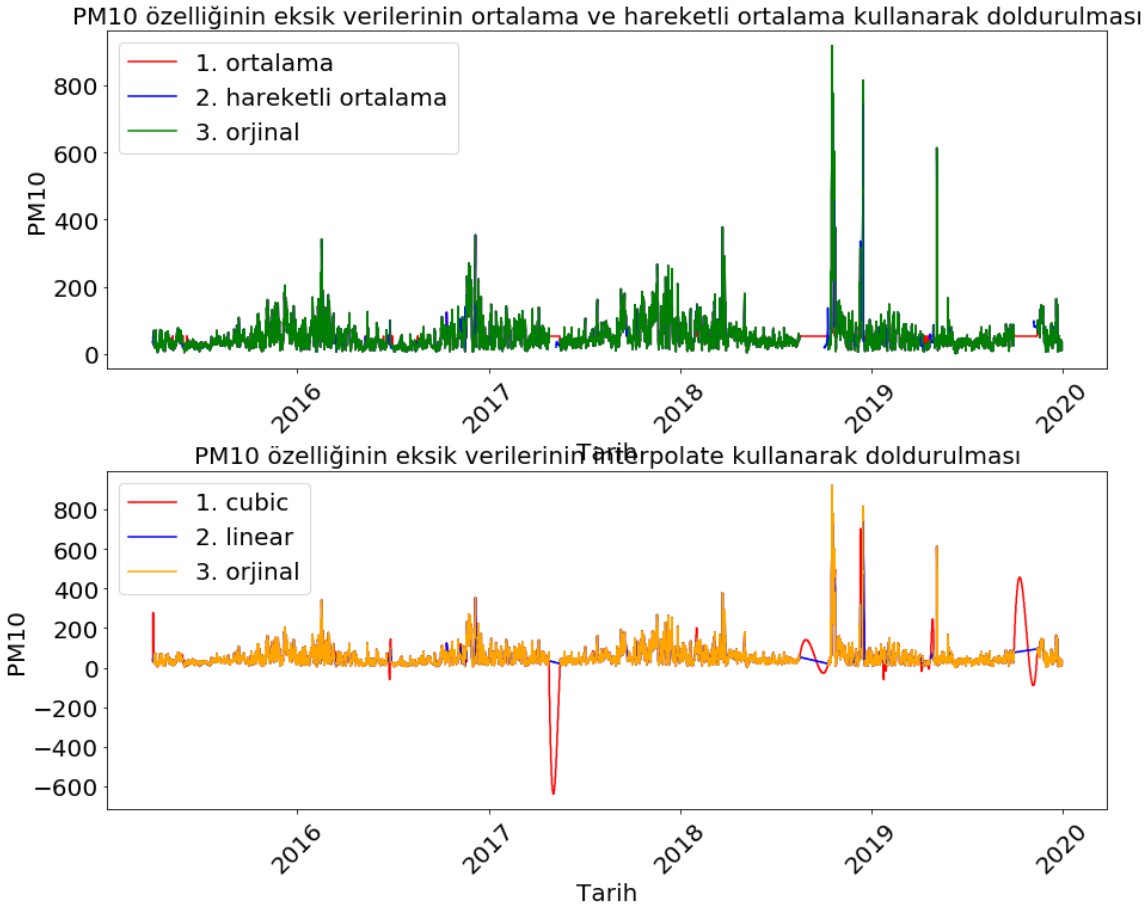
```
ASPM10_df_fill_linear = ASPM10_df_null.interpolate(method='linear')
ASPM10_df_fill_cubic = ASPM10_df_null.interpolate(method='cubic')
ASPM10_df_fill_quadratic = ASPM10_df_null.interpolate(method='quadratic')
ASPM10_df_fill_nearest = ASPM10_df_null.interpolate(method='nearest')
```

In [115]:

```
import matplotlib.pyplot as plt
import matplotlib
import scipy.interpolate
import warnings
plt.figure(figsize=(15,12))
kentsel_df.Tarih = pd.to_datetime(kentsel_df.Tarih, format='%d.%m.%Y')
ax = plt.gca()
ax.xaxis.set_major_locator(matplotlib.dates.YearLocator())
ax.xaxis.set_major_formatter(matplotlib.dates.DateFormatter('%Y'))
plt.subplots_adjust(hspace=.3)

plt.subplot(2,1,1)
plt.plot(kentsel_df.Tarih,ASPM10_df_fill_mean,color="red")
plt.plot(kentsel_df.Tarih,ASPM10_df_fill_rollingMean,color="blue")
plt.plot(kentsel_df.Tarih,ASPM10_df_null,color="green")
plt.legend(["1. ortalama","2. hareketli ortalama","3. orjinal"], fontsize=20)
plt.title("PM10 özelliğinin eksik verilerinin ortalama ve hareketli ortalama kullanarak
doldurulması", fontsize=20)
plt.xlabel("Tarih", fontsize=20)
plt.ylabel("PM10", fontsize=20)
plt.xticks(rotation=45, fontsize=20)
plt.yticks(fontsize=20)
plt.subplot(2,1,2)
plt.plot(kentsel_df.Tarih,ASPM10_df_fill_cubic,color="red")
plt.plot(kentsel_df.Tarih,ASPM10_df_fill_linear,color="blue")
plt.plot(kentsel_df.Tarih,ASPM10_df_null,color="orange")
plt.legend(["1. cubic","2. linear","3. orjinal"], fontsize=20)
plt.title("PM10 özelliğinin eksik verilerinin interpolate kullanarak doldurulması", fon
tsize=20)
plt.xlabel("Tarih", fontsize=20)
plt.ylabel("PM10", fontsize=20)
plt.xticks(rotation=45, fontsize=20)
plt.yticks(fontsize=20)
```

Out[115]:

```
(array([-800., -600., -400., -200.,    0.,  200.,  400.,  600.,  800.,
        1000.]), <a list of 10 Text yticklabel objects>)
```



Sonuçların değerlendirilmesi

Grafiklerden elde edilen sonuçlar:

1. Kullanılan veri seti zamana bağlı olduğundan dolayı eksik veriler doldurulurken interpolate methodları kullanılmalıdır.
2. İnterpolate methodu seçilirken, grafik kullanılarak orjinal değere en yakın olan method seçilmiştir. Grafikden de görüldüğü üzere linear methodu orjinale en yakın değerler üretmiştir. Diğer parametrin eksik verileride benzer olarak doldurulmuştur.

Verileri Görselleştirme

Verilere görselleştirme işlemleri yapılırken aşağıda ki hazırlanmış olan sorulardan faydalanılacaktır.

Sorular

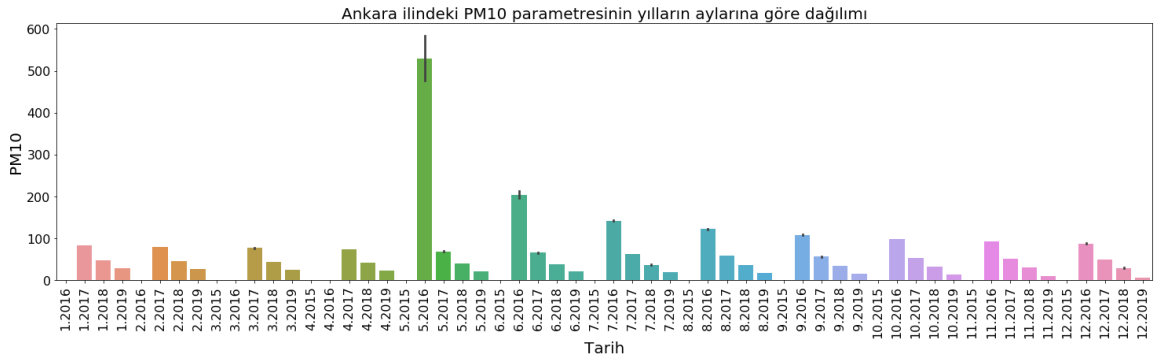
1. Her bir yıldaki madde oranı aylara göre nasıl değişiklik göstermektedir?
2. 2019 yılında farklı istasyonlarda ölçülen değerler nasıl değişiklik göstermektedir?

In [18]:

```
# Her bir yıldaki aylara göre madde oranı
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from matplotlib.dates import DateFormatter
from datetime import datetime, timedelta
import matplotlib

df = pd.read_csv("dataframee.csv")
def ciz(madde):
    plt.figure(figsize=(25,6))
    plt.subplots_adjust(hspace=.8)
    pm10_all = pd.DataFrame(df.groupby(by=df[madde]).mean())
    sns.barplot(df.Tarih.sort_values(ascending=False),pm10_all.index)
    titleA = 'Ankara ilindeki ' + madde + ' parametresinin yılların aylarına göre dağıl
ımı'
    plt.title(titleA, fontsize=20)
    plt.xlabel("Tarih", fontsize=20)
    plt.xticks(rotation=90, fontsize=16)
    plt.yticks(fontsize=16)
    plt.ylabel(madde, fontsize=20)
```

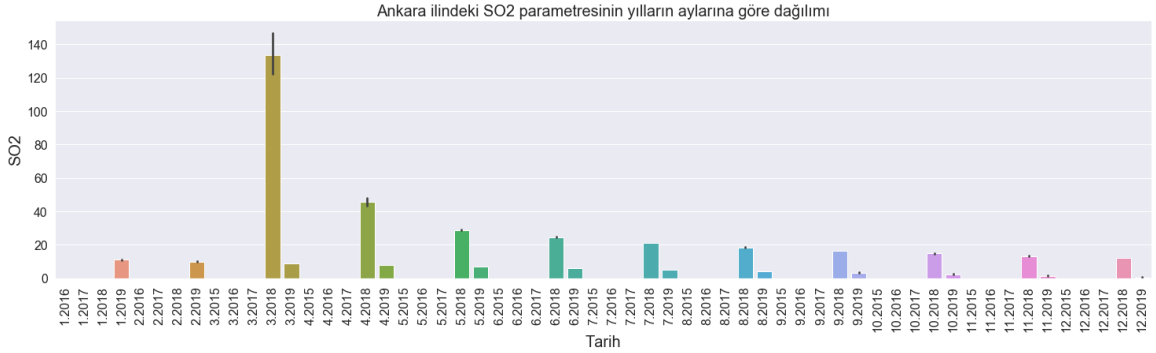
In [19]:

```
ciz("PM10")
```



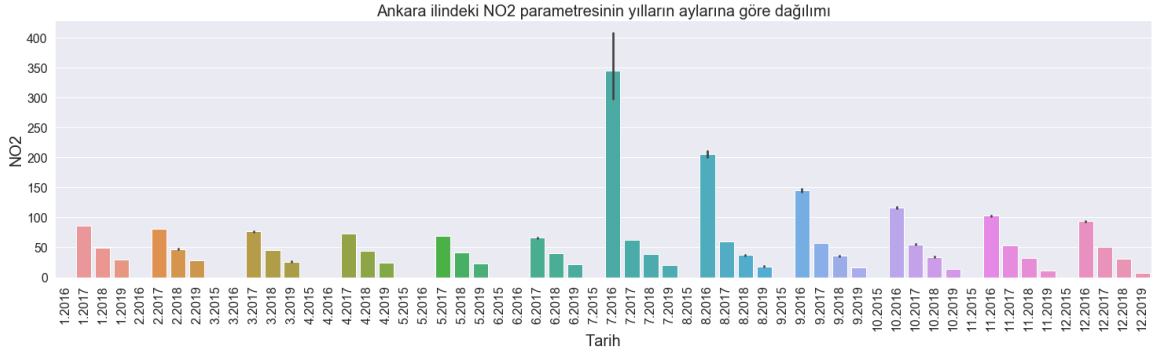
In [131]:

```
ciz("SO2")
```



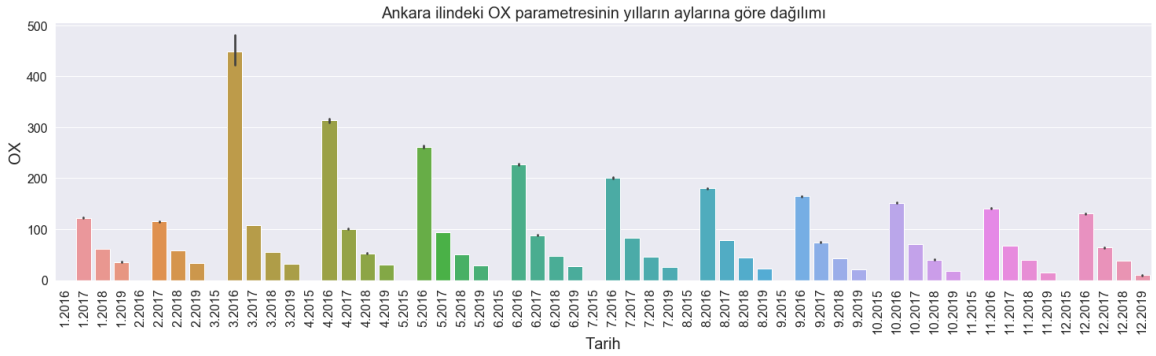
In [132]:

```
ciz("NO2")
```



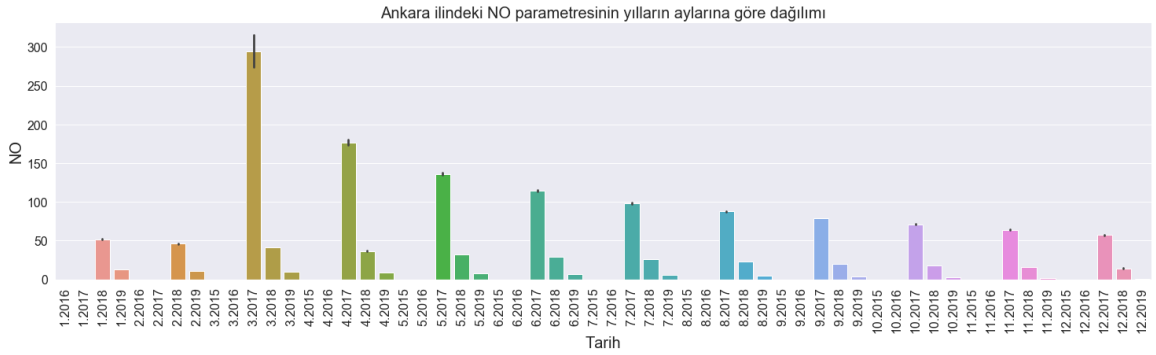
In [133]:

```
ciz("OX")
```



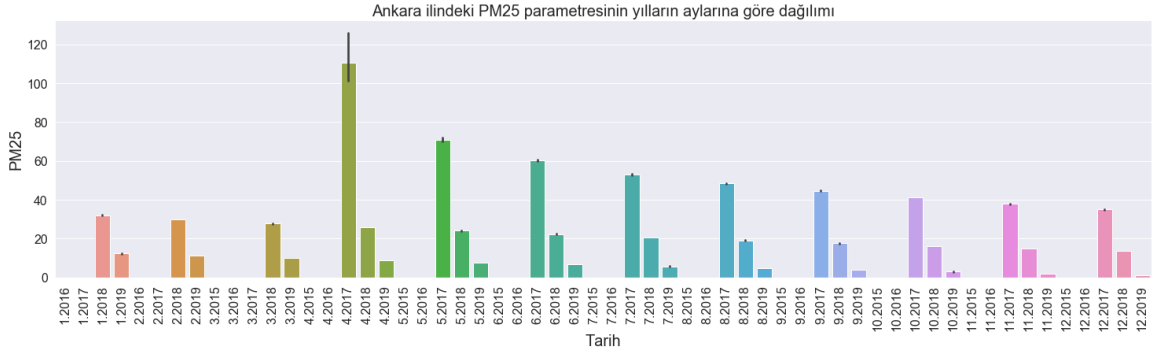
In [134]:

```
ciz("NO")
```



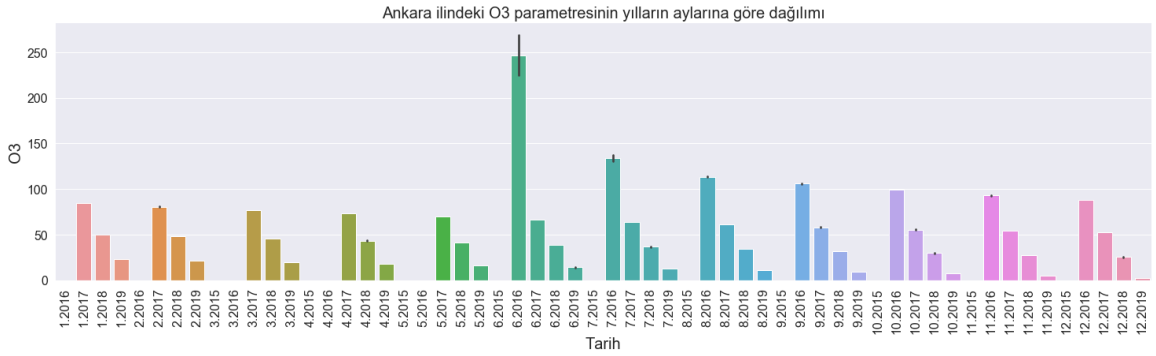
In [135]:

```
ciz("PM25")
```



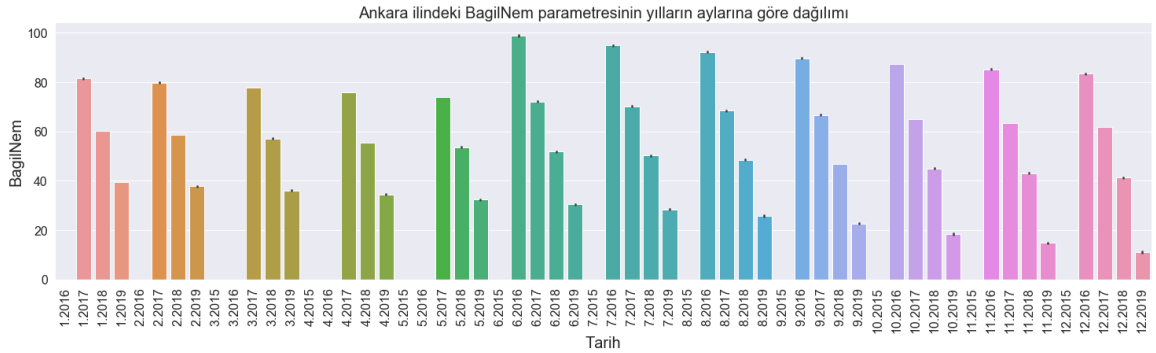
In [136]:

```
ciz("O3")
```



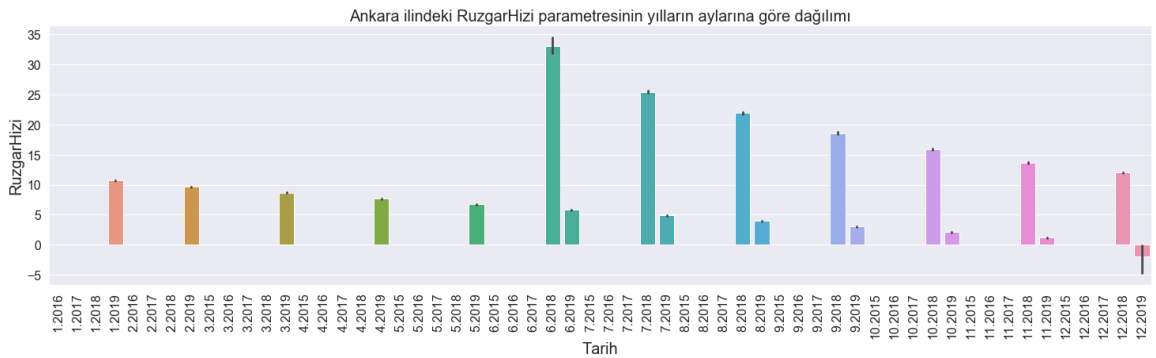
In [137]:

```
ciz("BagilNem")
```



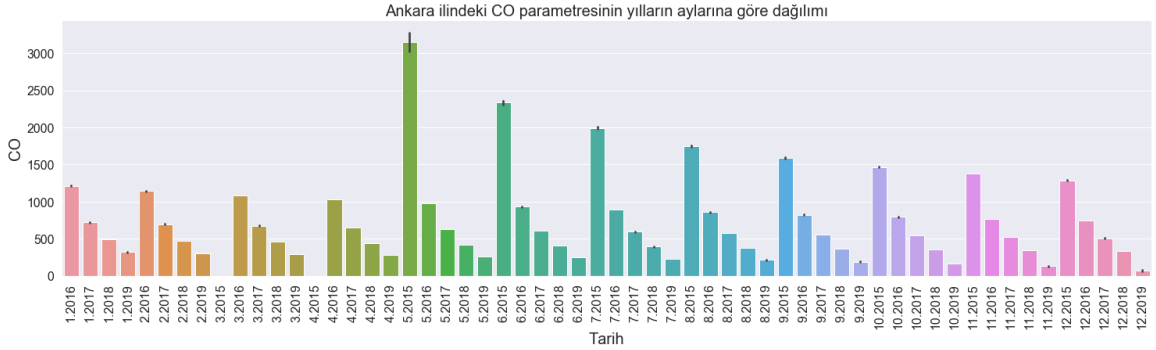
In [138]:

```
ciz("RuzgarHizi")
```



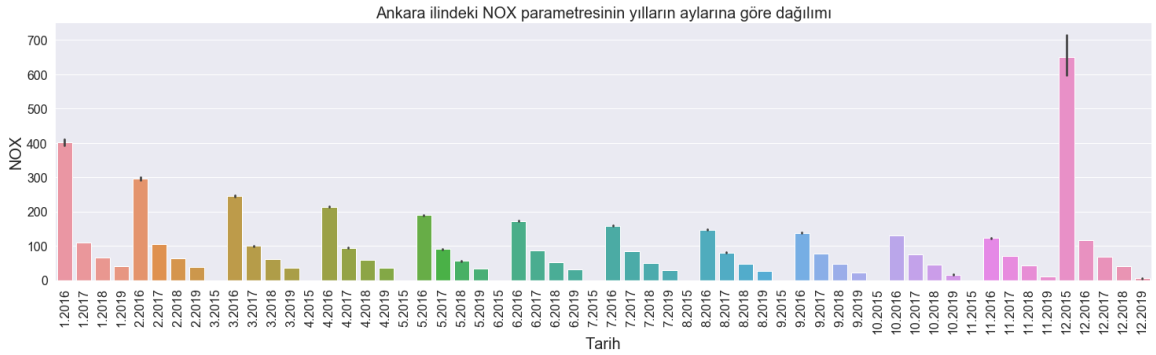
In [139]:

```
ciz("CO")
```



In [140]:

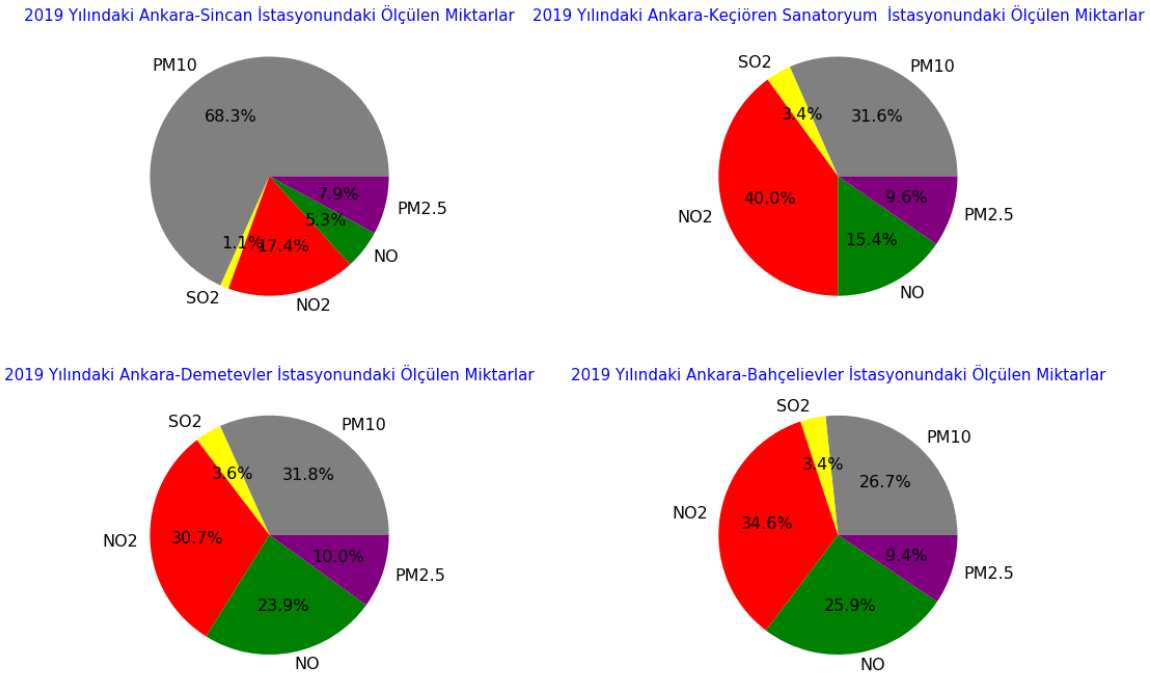
```
ciz("NOX")
```



In [133]:

```
#Farklı istasyonlarda ölçülen değerler
import matplotlib.pyplot as plt
import warnings

plt.figure(figsize=(18,18))
labels=["PM10","SO2","NO2","NO","PM2.5"]
colors=["grey","yellow","red","green","purple"]
explode=[0,0,0,0,0]
df2019PM10 = pd.read_csv("dataframe2019PM10.csv")
size[0]=df2019PM10.PM10.values.sum()
df2019SO2 = pd.read_csv("dataframe2019SO2.csv")
size[1]=df2019SO2.SO2.values.sum()
df2019NO2 = pd.read_csv("dataframe2019NO2.csv")
size[2]=df2019NO2.NO2.values.sum()
df2019NO = pd.read_csv("dataframe2019NO.csv")
size[3]=df2019NO.NO.values.sum()
df2019PM25 = pd.read_csv("dataframe2019PM25.csv")
size[4]=df2019PM25.PM25.values.sum()
plt.subplot(3, 2, 1)
plt.pie(size,explode,labels,colors,autopct="%1.1f%%", textprops={'fontsize': 16})
plt.title("2019 Yılındaki Ankara-Sincan İstasyonundaki Ölçülen Miktarlar",color="blue",
fontsize=15)

dfAK2019PM10 = pd.read_csv("dataframeAK2019PM10.csv")
size[0]=dfAK2019PM10.PM10.values.sum()
dfAK2019SO2 = pd.read_csv("dataframeAK2019SO2.csv")
size[1]=dfAK2019SO2.SO2.values.sum()
dfAK2019NO2 = pd.read_csv("dataframeAK2019NO2.csv")
size[2]=dfAK2019NO2.NO2.values.sum()
dfAK2019NO = pd.read_csv("dataframe2019NO.csv")
size[3]=dfAK2019NO.NO.values.sum()
dfAK2019PM25 = pd.read_csv("dataframeAK2019PM25.csv")
size[4]=dfAK2019PM25.PM25.values.sum()
plt.subplot(3, 2, 2)
plt.pie(size,explode,labels,colors,autopct="%1.1f%%", textprops={'fontsize': 16})
plt.title("2019 Yılındaki Ankara-Keçiören Sanatoryum  İstasyonundaki Ölçülen Miktarlar"
,color="blue",fontsize=15)

df2019ADPM10 = pd.read_csv("dataframeAD2019PM10.csv")
size[0]=df2019ADPM10.PM10.values.sum()
df2019ADSO2 = pd.read_csv("dataframeAD2019SO2.csv")
size[1]=df2019ADSO2.SO2.values.sum()
df2019ADNO2 = pd.read_csv("dataframeAD2019NO2.csv")
size[2]=df2019ADNO2.NO2.values.sum()
df2019ADNO = pd.read_csv("dataframeAD2019NO.csv")
size[3]=df2019ADNO.NO.values.sum()
df2019ADPM25 = pd.read_csv("dataframeAD2019PM25.csv")
size[4]=df2019ADPM25.PM25.values.sum()
plt.subplot(3, 2, 3)
plt.pie(size,explode,labels,colors,autopct="%1.1f%%", textprops={'fontsize': 16})
plt.title("2019 Yılındaki Ankara-Demetevler İstasyonundaki Ölçülen Miktarlar",color="bl
ue",fontsize=15)

df2019ABPM10 = pd.read_csv("dataframeAB2019PM10.csv")
size[0]=df2019ABPM10.PM10.values.sum()
df2019ABSO2 = pd.read_csv("dataframeAB2019SO2.csv")
size[1]=df2019ABSO2.SO2.values.sum()
df2019ABNO2 = pd.read_csv("dataframeAB2019NO2.csv")
size[2]=df2019ABNO2.NO2.values.sum()
```

```
df2019ABNO = pd.read_csv("dataframeAB2019NO.csv")
size[3]=df2019ABNO.NO.values.sum()
df2019ABPM25 = pd.read_csv("dataframeAB2019PM25.csv")
size[4]=df2019ABPM25.PM25.values.sum()
plt.subplot(3, 2, 4)
plt.pie(size,explode,labels,colors,autopct="%1.1f%%", textprops={'fontsize': 16})
plt.title("2019 Yılındaki Ankara-Bahçelievler İstasyonundaki Ölçülen Miktarlar",color=
"blue",fontsize=15)
plt.show()
```



Yukarıda ki grafiklerde Ankara ilinde geçtiğimiz yılın farklı istasyonlarındaki ölçülen ortak olan veriler değerlendirilmiştir. Elde edilen sonuçlar aşağıdaki gibidir.

- 2019 yılında Ankara-Sincan istasyonunda ki en yüksek değer PM10 olarak ölçülmüştür.
- 2019 yılında Ankara-Keçiören Sanatoryum istasyonunda ki en yüksek değer NO2 olarak ölçülmüştür.
- 2019 yılında Ankara-Demetevler istasyonunda ki en yüksek değer PM10 olarak ölçülmüştür.
- 2019 yılında Ankara-Bahçelievler istasyonunda ki en yüksek değer NO2 olarak ölçülmüştür.
- 2019 yılında Ankara-Sincan istasyonunda ki en düşük değer SO2 olarak ölçülmüştür.
- 2019 yılında Ankara-Keçiören Sanatoryum istasyonunda ki en düşük değer SO2 olarak ölçülmüştür.
- 2019 yılında Ankara-Demetevler istasyonunda ki en düşük değer SO2 olarak ölçülmüştür.
- 2019 yılında Ankara-Bahçelievler istasyonunda ki en düşük değer SO2 olarak ölçülmüştür.
- 2019 yılında tüm istasyonlarda ölçülen değerlere bakıldığında ortak olarak en düşük değer SO2 olarak belirlenmiştir.
- 2019 yılında tüm istasyonlarda ölçülen değerlere bakıldığında en yüksek değer maddelere göre farklılık oluşturmaktadır.

Kaynakça

[1] https://www.udemy.com/course/data-visualization-adan-zye-veri-gorsellestirme-3/learn/lecture/10835752#overview (https://www.udemy.com/course/data-visualization-adan-zye-veri-gorsellestirme-3/learn/lecture/10835752#overview), Erişim tarihi: 06.04.2020.

[2] https://stackoverflow.com/questions/30560198/resampling-non-time-series-data# (https://stackoverflow.com/questions/30560198/resampling-non-time-series-data#), Erişim tarihi: 06.04.2020.

[3] https://www.astro.umass.edu/~schloerb/ph281/Lectures/Interpolation/Interpolation.pdf (https://www.astro.umass.edu/~schloerb/ph281/Lectures/Interpolation/Interpolation.pdf), Erişim tarihi: 07.04.2020.

[4] https://github.com/pandas-dev/pandas/issues/8796 (https://github.com/pandas-dev/pandas/issues/8796), Erişim tarihi: 08.04.2020.

[5] https://medium.com/@drnesr/filling-gaps-of-a-time-series-using-python-d4bfddd8c460 (https://medium.com/@drnesr/filling-gaps-of-a-time-series-using-python-d4bfddd8c460), Erişim tarihi: 08.04.2020.

[7] https://stackoverflow.com/questions/12444716/how-do-i-set-the-figure-title-and-axes-labels-font-size-in-matplotlib (https://stackoverflow.com/questions/12444716/how-do-i-set-the-figure-title-and-axes-labels-font-size-in-matplotlib), Erişim tarihi: 08.04.2020.

[8] https://stackoverflow.com/questions/31255815/seaborn-tsplot-does-not-show-datetimes-on-x-axis-well (https://stackoverflow.com/questions/31255815/seaborn-tsplot-does-not-show-datetimes-on-x-axis-well), Erişim tarihi: 09.04.2020.

[9] https://www.earthdatascience.org/courses/use-data-open-source-python/use-time-series-data-in-python/date-time-types-in-pandas-python// (https://www.earthdatascience.org/courses/use-data-open-source-python/use-time-series-data-in-python/date-time-types-in-pandas-python//), Erişim tarihi: 09.04.2020.

[10] https://www.earthdatascience.org/courses/use-data-open-source-python/use-time-series-data-in-python/date-time-types-in-pandas-python/customize-dates-matplotlib-plots-python/ (https://www.earthdatascience.org/courses/use-data-open-source-python/use-time-series-data-in-python/date-time-types-in-pandas-python/customize-dates-matplotlib-plots-python/), Erişim tarihi: 09.04.2020.

[11] https://matplotlib.org/3.1.1/gallery/recipes/common_date_problems.html/ (https://matplotlib.org/3.1.1/gallery/recipes/common_date_problems.html/), Erişim tarihi: 09.04.2020.