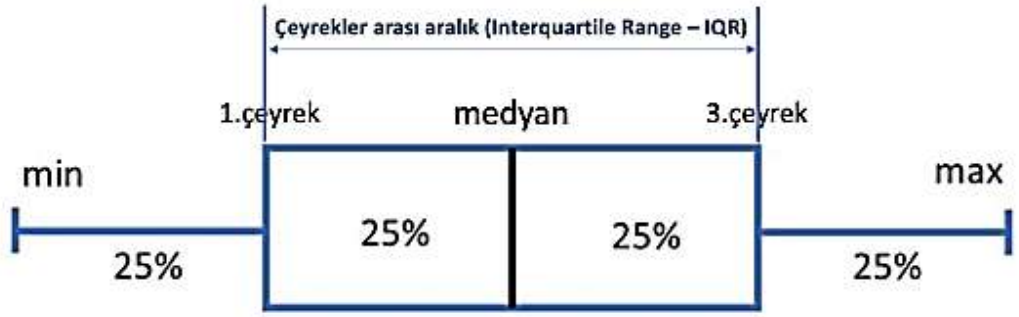
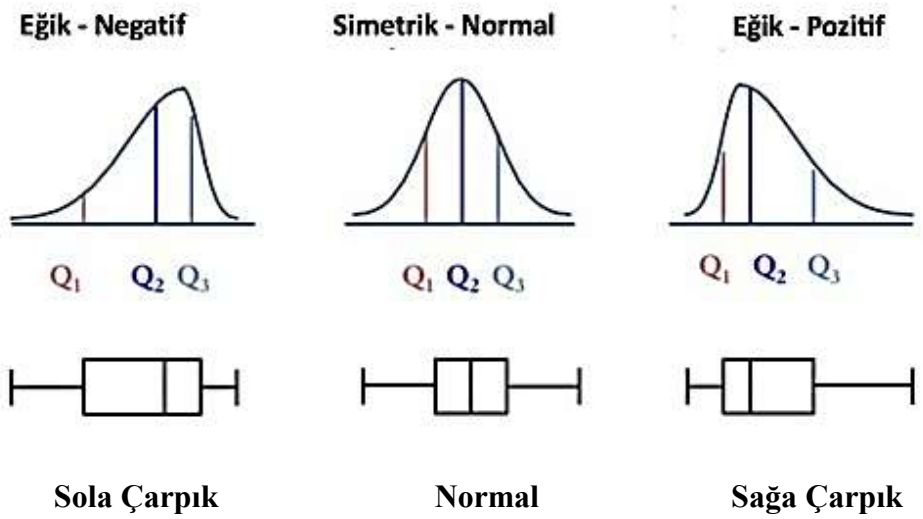
VERİ GÖRSELLEŞTİRME

- ***Boxplot / Kutu Grafiği (Kutu – Bıyık Grafiği)***

Kutu grafiği nicel değişkenleri betimlemek için kullanılır. Grafik bize verilerin merkezi konumunu, dağılışını, çarpıklığını ve basıklığı hakkında bilgiler vermektedir. Aynı zamanda veride **aykırı değerlerin tespit edilmesinde** de kullanılır. Grafiğin çizilmesi için değişken değerlerine ait minimum, maksimum, birinci çeyrek, medyan ve üçüncü çeyrek değerlerinin hesaplanması gerekmektedir.



Medyan verinin konumu hakkında bilgi verirken, **çeyrekler arası açıklık** verinin dağılışı hakkında bilgi vermektedir. Çeyrekler arası açıklığın büyümesi değişken değerlerinin daha geniş aralıkta yer aldığını, dağılışının daha büyük olduğunu göstermektedir.



Kutu grafiği basıklık – çarpıklık durumları hakkında da bilgiler vermektedir. Yukarıda bu durumlara ilişkin bilgiler bulunmaktadır. Eğer kutunun genişiliği çizginin genişliğine yaklaşırsa basık aksi durumda ise sivri olduğu yorumlanabilir.

AVANTAJLARI	DEZAVANTAJLARI
• Özet istatistiksel bilgileri gösterirken bize verinin merkezi hakkında bilgi vermektedir. • Aykırı değerlere duyarlı analizlerde çok tercih edilir. • Birden fazla değişken için yan ayana kutu grafikleri çizilerek verinin benzerliği hakkında bilgi vermektedir.	• Veri setinin küçük olduğu durumlarda yanıltıcı olabilmektedir. • Veri setinin tümünün özetidir. Derinlemesine analiz için diğer yöntemlerle kullanılmalıdır.

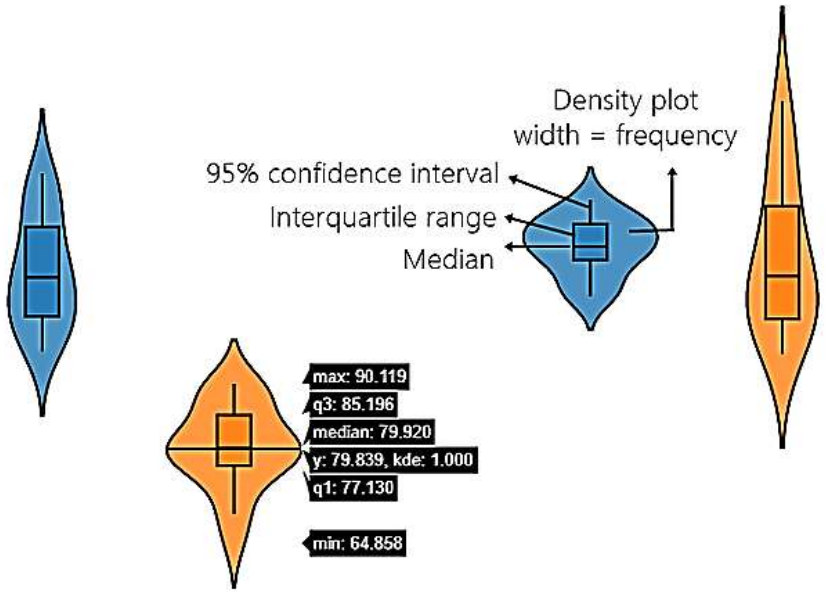
Kullanıldığı problem tipi;	Kullanıldığı veri tipi;
Dağılım analizlerinde Aykırı veri analizinde Gruplar arası veriyi karşılaştırmada Verinin durumunu inceleyip karar vermede Makine öğrenmesinde	Sürekli / sayısal veri tipi için uygundur.

- ***<u>Violin Plot / Violin Grafiği (Keman Grafiği)</u>***

  Farklı kategorilerde olan verilerin dağılımını tek grafikte göstererek bizlere hem kullanım kolaylığı hem de zarif bir şekilde analizi incelememizi sağlamaktadır. Keman grafiği, sayısal değerler için kullanılır. Bu sayede birden fazla istatistiksel analiz yapmamızı sağlar. Keman grafiği, kutu grafiği ve yoğunluk grafiklerinin karışımı olan bir grafiktir. Gruplar arasındaki sayısal değerlerin karşılaştırılmasında çok faydalıdır. Kutu grafikleri sadece özet istatistiksel bilgileri gösterirken keman grafikleri özet istatistiklerle birlikte her değişkenin yoğunluğunu da göstermektedir.

Keman grafiği çok modlu verilerde kullanılmaktadır. Bu sayede farklı tepelerin varlıklarını ve konumlarını göstermektedir. Kategoriler arasındaki yani gruplar arasındaki dağılımı karşılaştırmada kullanılır.

Kutu grafikleri verinin dağılımlarından etkilenmedikleri için yanıltıcı olduğu zaman keman grafikleri ile kutu grafiğinde ki tüm özet istatistiksel bilgileri barındırırken aynı zamanda verinin dağılım tuzağına düşmeyerek daha iyi bir sonuç vermektedir. Keman grafiğinde kalın, geniş kısım veri değerlerinin daha yüksek frekansa sahip olduğunu gösterirken ince, basık kısım ise verinin daha düşük frekansa ait olduğunu göstermektedir.

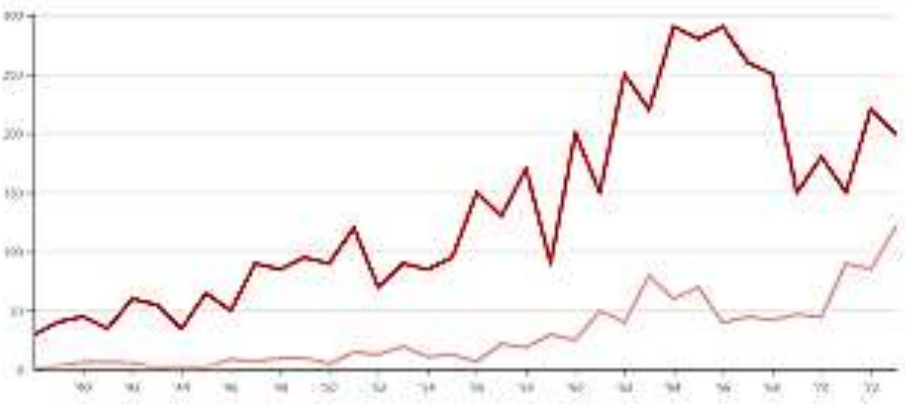


Keman grafiğinde ortada ki nokta medyan iken bıyıklar %95 güven aralığında veriyi temsil etmektedir. Her keman farklı veri kümelerini temsil etmektedir. veriler normal dağılıma uymasa bile keman grafikleri hem nicel hem de nitel verileri görselleştirmede çok başarılıdır.

AVANTAJLARI	DEZAVANTAJLARI
• Özet istatistiksel bilgileri gösterirken aynı zamanda verinin yoğunluğu hakkında da bilgi vermektedir. • Çok tepe noktası olan dağılımları göstermektedir. • Birden fazla değişken için gruplar arası karşılaştırma yaparken hem medyanı hem de dağılım şeklini yoğunluğunu da göstermektedir.	• Veri setinin küçük olduğu durumlarda yanıltıcı olabilmektedir. • Yorumlanması istatistikle arası iyi olmayanlar için bir tık daha zordur.

- Grupların dağılımı, dağılımın normalliği ve aykırı değerleri (verinin yığılmasını) incelemelerde kullanılır. Sürekli (sayısal) verilerde, gruplandırmada kategorik verilerde, nominal (sıralı) verilerde ve kesikli veri tipleriyle kullanılabilmektedir.

- ***Line Graph (Çizgi Grafiği)***



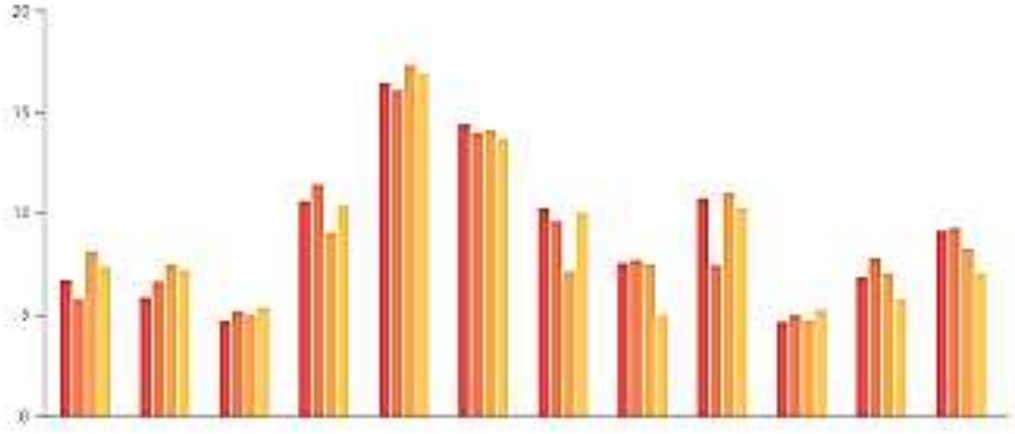
Çizgi grafikleri veriler arasındaki eğimi ve ilişkiyi göstermektedir. Elimizde zaman boyunca ölçtüğümüz veri seti varsa bu durumda değişkenin zaman boyunca nasıl gelişme gösterdiğini görmek için çizgi grafiği kullanılacaktır. Zaman boyunca çizilen değişken nicel bir değişken olmalıdır. Bazen birbirleriyle ilişkili iki veya daha fazla değişkenin bir arada zaman boyunca değişimini görmek içinde kullanılabilmektedir.

Çizgi grafiği oluşturabilmek için öncelikle koordinat sisteminde y ekseni nicel bir değere sahip olması gerekirken x ekseninin ölçülmek istenen değişken olması gerekmektedir. x ve y ekseninin ortak kesişimin de ki noktalar birleştirilerek çizgi grafiği oluşturulmaktadır.

AVANTAJLARI	DEZAVANTAJLARI
• Verinin zaman içinde ki eğilimini göstermektedir. • Sade ve anlaşılır bir grafiktir. • Birden fazla verinin zaman içinde ki durumunu karşılaştırabilmektedir. • Zaman serisi analizlerinde çok kullanılmakta ve en uygunudur.	• Verinin çok fazla olması bazen karmaşıklığa sebep olabilir. • Aykırı değer grafiği bozabileceği için verinin kontrol edilmesi gerekmektedir.

- Zaman serisi analizlerinde, trend takiplerinde, ve tahminleme gibi durumlarda kullanılması uygundur. X ekseni için sıralı (sürekli) veri tipi, y ekseni için de sürekli (sayısal) veri tipi kullanılarak çizgi grafikleri oluşturulabilir.

- ***Bar Graph / Bar Chart (Çubuk Grafiği)***



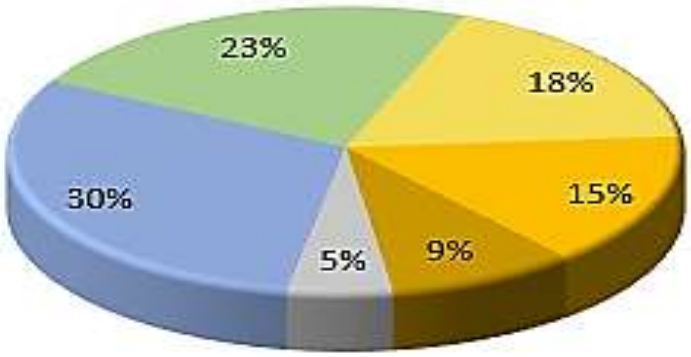
Kategorik değişkenleri özetlemek için çubuk grafikleri kullanılmaktadır. Histogram grafiği çizerken nicel değişkenler kullanılırken çubuk grafiğinde nitel değişkenler kullanılmaktadır. Grafiğin yatay x ekseninde sınıflama ölçme düzeyinde ki veri tipi yer alırken dikey y ekseninde bu sınıflara karşılık gelen frekanslar yer almaktadır.

Histogramda çubuklar arasında boşluk yer almaz iken çubuk grafiğinde yatay x ekseni kategorik değişken olduğu için çubuklar arasında genelde hafif boşluklar yer almaktadır. Çubuk grafikleri karşılaştırma amaçlı da kullanılabilir. Örneğin x eksende ki kategorilere iki ayrı yılın frekansları çizilerek incelenebilmektedir. Oransal frekans değerleri içinde oluşturulabilir.

AVANTAJLARI	DEZAVANTAJLARI
• Verilerin frekanslarını inceleyerek karşılaştırılması daha kolaydır. • Sade ve anlaşılır bir grafiktir. • Grafiğin okunması kolaydır.	• Çok fazla kategoride karmaşıklığa sebep olabilir. • Zaman içinde ki değişimi araştırmada uygun değildir.

- Frekans analizlerinde, dağılım analizlerinde ve kategori karşılaştırılmasında kullanılmaktadır. X ekseni kategorik veri tipi olmalıyken y ekseni nicel sayısal veriler içermelidir.

- ***Pie Chart (Daire - Pasta Grafiği)***



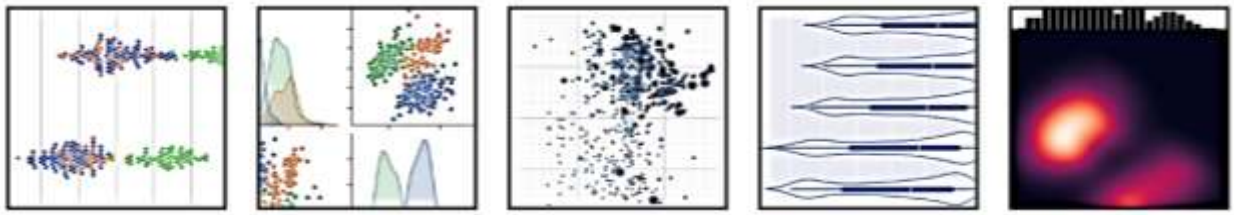
Daire grafiğinde her bir dilim bir kategoriye karşılık gelmek üzere dilimlere ayrılmıştır. Dilimlerin daire içinde ki büyüklüğü frekansları ile orantılıdır. Bu grafikler niteliksel değişkenlere ilişkin frekans dağılımlarında kullanılmaktadır. Daire grafiğindeki niteliksel değişkenlerin aldığı değerlerin her birinin sahip oldukları frekanslar, toplam frekans içinde ki ağırlığa göre dairenin toplam alanında ki belli bir paya sahip olmaktadır. Ayrıca bu alanların yüzde olarak ağırlıkları da grafik de belirtilmektedir.

AVANTAJLARI	DEZAVANTAJLARI
• Verilerin frekanslarına göre oluşarak oranları net bir şekilde aktarmaktadır. • Sade ve anlaşılır bir grafiktir. • Görsel olarak etkileyicidir.	• Çok fazla kategoride karmaşıklığa sebep olabilir. • Sayısal değişkenler kolay anlaşılmamaktadır.

- Yüzdesel dağılım analizlerinde, basit frekans karşılaştırılmalarında ve parça bütün ilişkilerinde kullanılmaktadır. Kategorik ve oransal sayısal veri türleriyle daire grafikleri oluşturulabilmektedir.

VERİ GÖRSELLEŞTİRME KÜTÜPHANELERİ

- ***“Seaborn” Kütüphanesi***



Matplotlib tabanlı bir python veri görselleştirme kütüphanelerinden birisidir. Python’da istatistiksel analizlerin görselleştirilmesidir. Pandas ile bütünleşmiştir.

Kütüphaneyi python’da yüklemek için,

```
import seaborn as sns
```

Yüklenen küyüphaneyi çalıştırmak için,

```
sns.set_theme()
```

kullanılır. Daha sonra çalışılmak istenen veri yüklenerek, görselleştirilmek istenen grafik türü ve özellikleri belirlenerek çalışmaya devam edilir.

Seaborn kütüphanesini kullanarak; dağılım grafikleri, grafik ile istatistiksel tahminler, kategorik veriler için özel grafikler (noktalar şeklinde dağılımını göstermek vb.), keman grafikleri vb. grafikler ile verilerimizi görselleştirip özetleyebilmekteyiz. Bir veri kümesi ve oluşturulmak istenilen grafik türü verildiğinde veri değerlerini renk, boyut ve stil gibi özelliklerle özelleştirip istatistiksel dönüşümleri hesaplayıp grafiği hem bilgilendirici hem de görsel bir sunum oluşturmaktadır. Veri kümesinde ki değişkenleri ya da alt kümelerini farklı eşleşmeler sağlayarak birden fazla şekil üreterek karşılaştırma yapabilmemize olanak sağlamaktadır. Veri görselleştirme, bilimsel sürecin vazgeçilmez bir parçasıdır. Etkili görselleştirmeler, bir bilim insanının hem kendi verilerini anlamasını hem de içgörülerini başkalarına iletmesini sağlar (Tukey, 1977).

Seaborn kütüphanesinde ki istatistiksel kütüphane grafikleri şu şekildedir:

1. Histogram grafiği
2. Kutu grafiği
3. Keman grafiği
4. Çubuk grafiği
5. Nokta grafiği
6. Çizgi grafiği
7. Korelasyon ve matrisler için ısı haritası
8. Regresyon dağılımın çizgisi için dağılım grafiğidir.

- ***"Plotly" Kütüphanesi***

R, Python ve Javascript dilleri ile uyumu çalışabilen çeşitli grafikler oluşturmaya sağlayan etkileşimli, güçlü ve açık kaynaklı bir kütüphane türüdür. Python da pansdas dataframe içerisinde istenilen veri manipülasyonu gerçekleştirilip istenilen görseller oluşturulabilmektedir. Gerçek zamanlı verilerle çalışarak estetik, şık grafik türleriyle görsel şölen oluşturmaktadır. Web tabanı ile uyumlu görsellerde oluşturabilmektedir. En önemli özelliği oluşturduğu grafiklerde kullanıcıdan zoom veya tıklama, yakınlaştırma, uzaklaştırma gibi eylemlerle kullanıcı etkileşimini desteklemektedir. Aynı zamanda kullanıcılarda veri noktalarıyla etkileşime girebilmektedir.

Kütüphaneyi python'da yüklemek için,

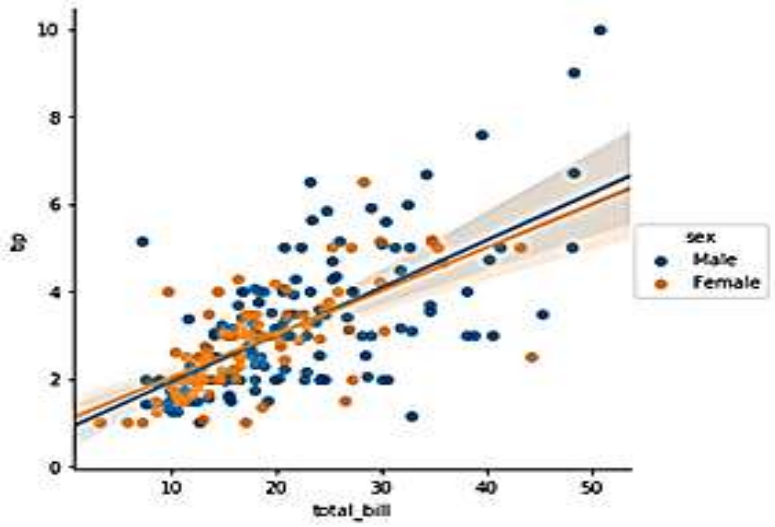
```
import plotly.express as px
```

kullanılır. Daha sonra çalışılmak istenen veri yüklenerek, görselleştirilmek istenen grafik türü ve özellikleri belirlenerek çalışmaya devam edilir.

Plotly kütüphanesinde ki istatistiksel kütüphane grafikleri şu şekildedir:

1. Çizgi grafiği
2. Çubuk grafiği
3. Pasta grafiği
4. Dağılım grafiği
5. Kutu grafiği
6. Harita ve ısı grafikleri
7. 3 boyutlu grafikler
8. Bilimsel, biyoinformatik ve yapay zeka / makine öğrenmesi gibi alanlarla da görseller oluşturarak bize veriler hakkında bilgiler sunmaktadır.

- ***“Matplotlib” Kütüphanesi***



Python'da animasyonlu ve etkileşimli görselleştirmeler oluşturarak farklı biçimde görseller oluşturmada kullanılan bir kütüphane çeşididir. 2 ve 3 boyutlu grafikler oluşturmayı sağlarken daha çok 2 boyutlu çizimlerde kullanılmaktadır.

Kütüphaneyi python’da yüklemek için,

```
import matplotlib.pyplot as plt
```

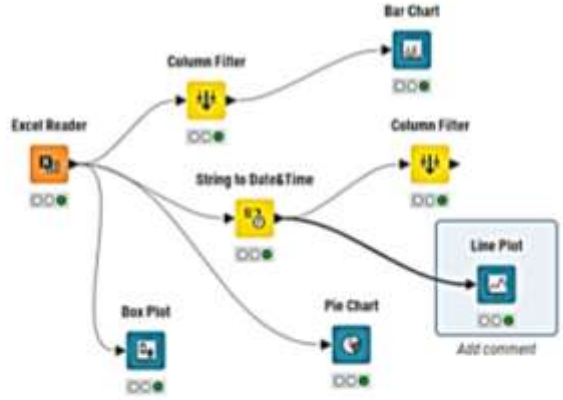
kullanılır. Daha sonra çalışılmak istenen veri yüklenerek, görselleştirilmek istenen grafik türü ve özellikleri belirlenerek çalışmaya devam edilir.

Matplotlib kütüphanesi ile python’da grafikleri tamamen özelleştirip istenilen renk, stil ve başlık gibi birçok özelliği ayarlayabilmekteyiz. Birden fazla alt grafik oluşturarak aynı zaman diliminde farklı grafikleri de incelememize olanak sağlamaktadır. Grafiklerin kalitesi yüksek kalitede olduğu için araştırma projelerinde, makalelerde ve sunumlarda tercih edilebilmektedir.

Matplotlib kütüphanesinde ki istatistiksel kütüphane grafikleri şu şekildedir:

1. Çizgi grafiği
2. Nokta grafiği
3. Pasta grafiği
4. Bar grafiği
5. Histogram grafiği
6. Serpilme grafikleri
7. Çubuk grafiği
8. Kutu grafiğidir.

KNIME İLE VERİLERİN GÖRSELLEŞTİRİLMESİ



- Ben verilerim üzerinde çalışırken Kaggle platformunda bulunan “Air Quality Day” verileri ile çalışmak istedim. Bu verileri indirip ilk olarak Python’da her kütüphaneye ait çalışmalarımı sonra da Knıme üzerinde denemeye çalıştım.

PYTHON İLE VERİLERİN GÖRSELLEŞTİRİLMESİ

Bu çalışma Kaggle da bulunan "Hava Kalitesi Verileri (AirQualityData)" ile çalışılmıştır.

https://www.kaggle.com/datasets/khushikyad001/air-quality-data

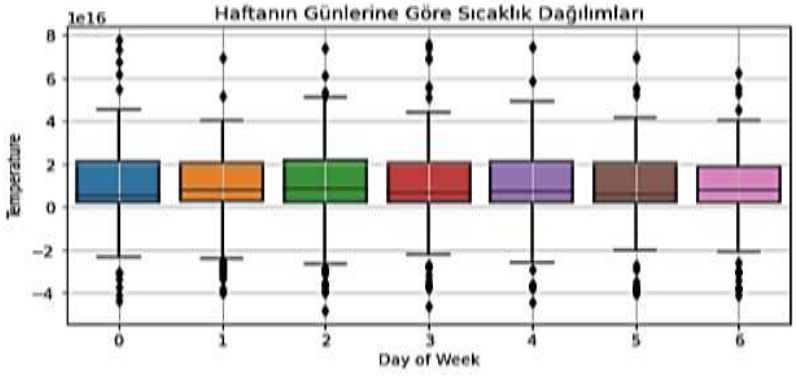
```
[29]: import pandas as pd
      import matplotlib.pyplot as plt

      # veriyi yükleyelim
      df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
      print(df.head())
```

```
        Date      Time        CO(GT)       NOx(GT)            NO2(GT)  \
0 2024-01-01  00:00:00  3.807947e+15  1.720268e+16   1443333168655310
1 2024-01-01  01:00:00  9.512072e+14  2.418243e+15   1377693178782120
2 2024-01-01  02:00:00  7.346740e+14  2.282881e+16  20055085534008100
3 2024-01-01  03:00:00  6.026719e+15  4.701607e+15   1845919085762540
4 2024-01-01  04:00:00  1.644585e+16  4.562559e+14   1141259682292070

         O3(GT)       SO2(GT)         PM2.5          PM10   Temperature  ...  \
0  1.181208e+16  1.215679e+16  1.473497e+16  2.088031e+16  2.856458e+16  ...
1  1.532583e+15  1.016178e+16  4.097984e+16  1.455956e+16  6.793192e+15  ...
2  4.437704e+16  2.414091e+15  7.259474e+14  2.615500e+16  2.443655e+15  ...
3  1.394886e+16  2.435392e+15  1.343397e+16  2.763679e+15  2.646395e+16  ...
4  9.563477e+15  4.875210e+15  9.900742e+15  2.942954e+16  1.053033e+16  ...
```

Burada verinin anlatmak istediğini, değişkenleri ve veri içeriğinin ilk 5 verisini inceleyerek verinin görselleştirmesi için bilgiler edinilmiştir.

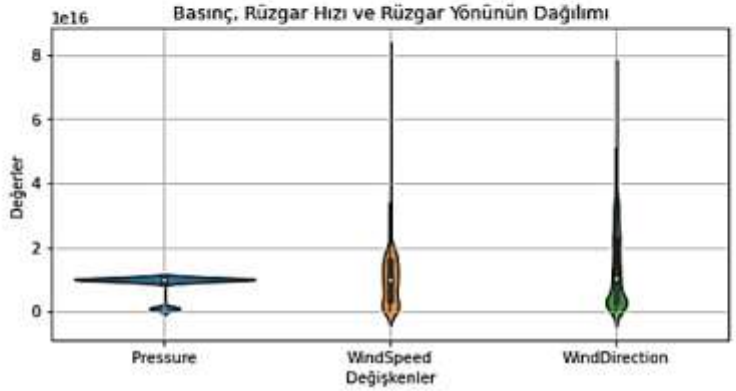
- **PYPLOT KÜTÜPHANESİ İLE VERİ GÖRSELLEŞTİRMESİ**

```
Kutu Grafiği Örneği

[1]: import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns

     df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")

     plt.figure(figsize=(8, 4))
     sns.boxplot(x="DayOfWeek", y="Temperature", data=df)
     plt.title("Haftanın Günlerine Göre Sıcaklık Dağılımları")
     plt.xlabel("Day of Week")
     plt.ylabel("Temperature")
     plt.grid(True)
     plt.show()
```



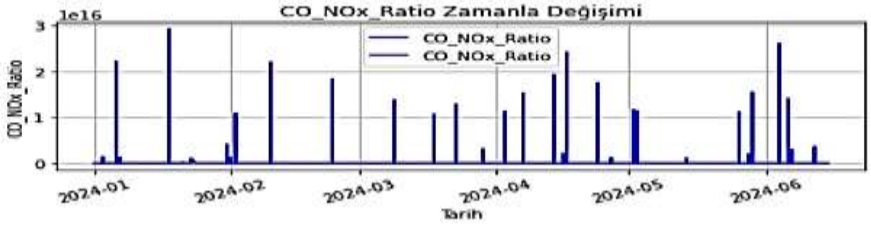
Keman grafiği örneği

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")

plt.figure(figsize=(8, 4))
sns.violinplot(data=df[["Pressure", "WindSpeed", "WindDirection"]])
plt.title("Basınç, Rüzgar Hızı ve Rüzgar Yönünün Dağılımı")
plt.xlabel("Değişkenler")
plt.ylabel("Değerler")
plt.grid(True)
plt.show()
```



Çizgi grafiği örneği

[4]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
plt.figure(figsize=(7, 3))
# tarihi dönüştürüyoruz
plt.plot(df['Date'], df['CO_NOx_Ratio'], color='b', label='CO_NOx_Ratio')
plt.plot(df['Date'], df['CO_NOx_Ratio'], color='b', label='CO_NOx_Ratio')
plt.title('CO_NOx_Ratio Zamanla Değişimi')
plt.xlabel('Tarih')
plt.ylabel('CO_NOx_Ratio')
plt.xticks(rotation=20)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

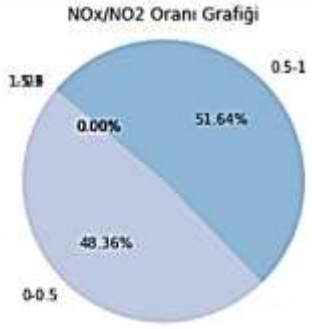


Çubuk grafiği örneği

[6]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
plt.figure(figsize=(7, 3))
# zamanı dönüştürüyoruz
df['Time'] = pd.to_datetime(df['Time'], format='%H:%M:%S')
df['Hour'] = df['Time'].dt.hour
plt.bar(df['Hour'], df['AirQualityIndex'], color='b', label='AirQualityIndex')
plt.title('Hava Kalitesi Endeksinin Zamanla Değişimi')
plt.xlabel('Hour of the Day')
plt.ylabel('AirQualityIndex')
plt.xticks(range(0, 24))  # 24 saatlik zaman diliminde ki değişimi görmek için
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

Pasta grafiği örneği

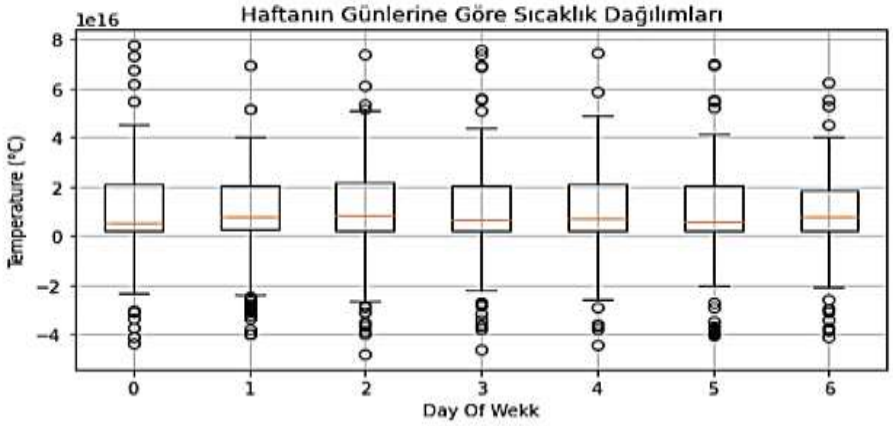
[7]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# NOx/NO2 oranını nicel olduğu için gruplandırıyoruz
bins = [0, 0.5, 1, 1.5, 2, 5]
labels = ['0-0.5', '0.5-1', '1-1.5', '1.5-2', '2+']
df['NOx_NO2_Group'] = pd.cut(df['NOx_NO2_Ratio'], bins=bins, labels=labels, include_lowest=True)
# oluşan gruplar sayılıyor
group_counts = df['NOx_NO2_Group'].value_counts().sort_index()
colors = [plt.cm.PuBuGn(i) for i in np.linspace(0.3, 0.8, len(group_counts))]
plt.figure(figsize=(4, 4))
plt.pie(group_counts, labels=group_counts.index, autopct='%2.2f%%', startangle=140, colors=colors)
plt.title("NOx/NO2 Oranı Grafiği")
plt.axis("equal")
plt.show()
```



- **MATPLOTLİB KÜTÜPHANESİ İLE VERİ GÖRSELLEŞTİRMESİ**

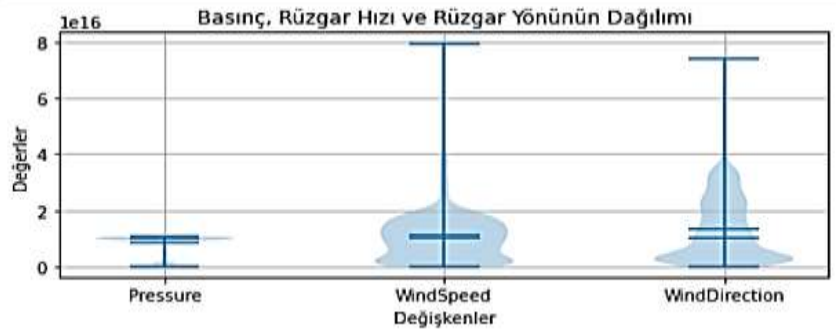
```
Kutu grafiği örneği
```

[8]:
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# günlere göre sıcaklık değerlerini gruplandırma
grouped = df.groupby("DayOfWeek")["Temperature"].apply(list)
plt.figure(figsize=(8, 4))
plt.boxplot(grouped, labels=grouped.index)
plt.title("Haftanın Günlerine Göre Sıcaklık Dağılımları")
plt.xlabel("Day Of Wekk")
plt.ylabel("Temperature (°C)")
plt.grid(True)
plt.show()
```



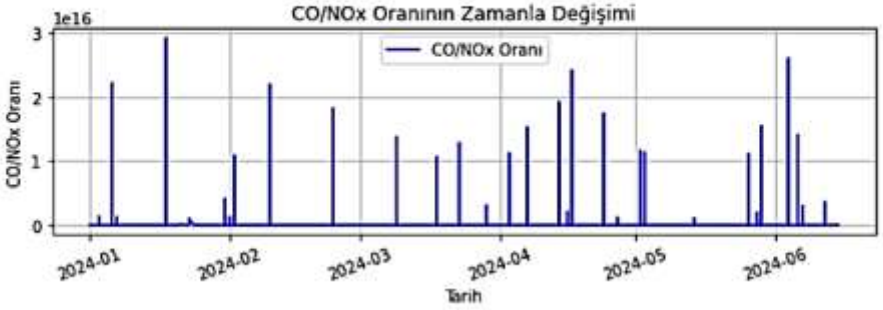

```
Keman grafiği örneği
```

[11]:
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
veri_düzeni = [
    df["Pressure"].dropna(),
    df["WindSpeed"].dropna(),
    df["WindDirection"].dropna()
]
plt.figure(figsize=(8, 3))
plt.violinplot(veri_düzeni, showmeans=True, showmedians=True)
plt.xticks([1, 2, 3], ["Pressure", "WindSpeed", "WindDirection"])
plt.title("Basınç, Rüzgar Hızı ve Rüzgar Yönünün Dağılımı")
plt.xlabel("Değişkenler")
plt.ylabel("Değerler")
plt.grid(True)
plt.show()
```



Cizgi grafiği örneği

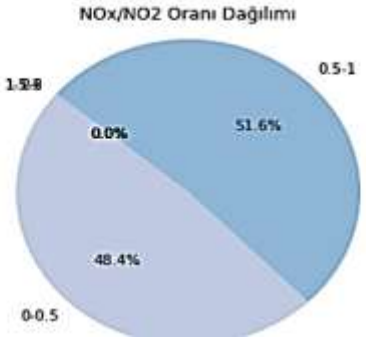
[13]:
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# tarihi dönüştürme
df['Date'] = pd.to_datetime(df['Date'])
plt.figure(figsize=(8, 3))
plt.plot(df['Date'], df['CO_NOx_Ratio'], color='b', label='CO/NOx Oranı')
plt.title('CO/NOx Oranının Zamanla Değişimi')
plt.xlabel('Tarih')
plt.ylabel('CO/NOx Oranı')
plt.xticks(rotation=20)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



Cubuk grafiği örneği

[15]:
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# zamanı datetime dönüştürme
df['Time'] = pd.to_datetime(df['Time'], format='%H:%M:%S', errors='coerce')
df['Hour'] = df['Time'].dt.hour
# saatlik ortalama hava kalitesini hesaplama
hourly_avg = df.groupby('Hour')['AirQualityIndex'].mean().reset_index()
plt.figure(figsize=(8, 3))
plt.bar(hourly_avg['Hour'], hourly_avg['AirQualityIndex'], color='skyblue', label='Ortalama Air Quality Index')
plt.title('Hava Kalitesi Endeksinin Gün İçindeki Değişimi (Saat Bazında)')
plt.xlabel('Saat (0-23)')
plt.ylabel('Ortalama AirQualityIndex')
plt.xticks(range(0, 24))
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend()
plt.tight_layout()
plt.show()
```

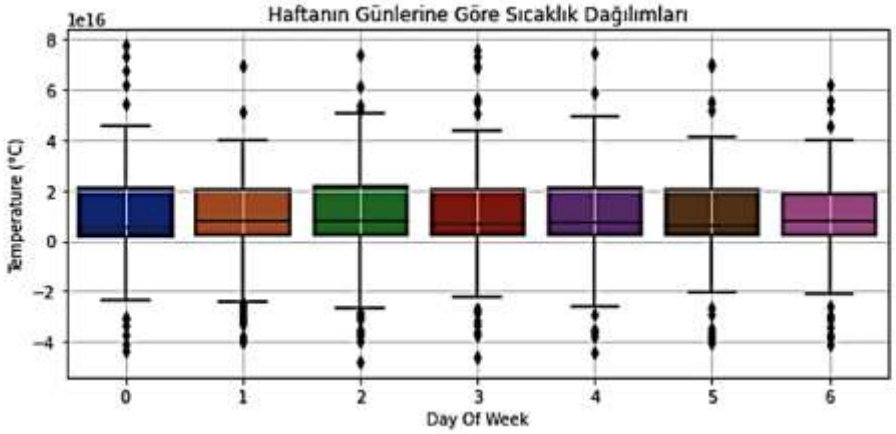
```
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# NOx/NO2 oran grupları oluşturma
bins = [0, 0.5, 1, 1.5, 2, 5]
labels = ['0-0.5', '0.5-1', '1-1.5', '1.5-2', '2+']
df['NOx_NO2_Group'] = pd.cut(df['NOx_NO2_Ratio'], bins=bins, labels=labels, include_lowest=True)
# grup sayıları
group_counts = df['NOx_NO2_Group'].value_counts().sort_index()
colors = [plt.cm.PuBuGn(i) for i in np.linspace(0.3, 0.8, len(group_counts))]
plt.figure(figsize=(4, 4))
plt.pie(group_counts,
        labels=group_counts.index,
        autopct='%1.1f%%',
        startangle=140,
        colors=colors)
plt.title("NOx/NO2 Oranı Dağılımı")
plt.axis("equal")
plt.tight_layout()
plt.show()
```



- **SEABORN KÜTÜPHANESİ İLE VERİ GÖRSELLEŞTİRMESİ**

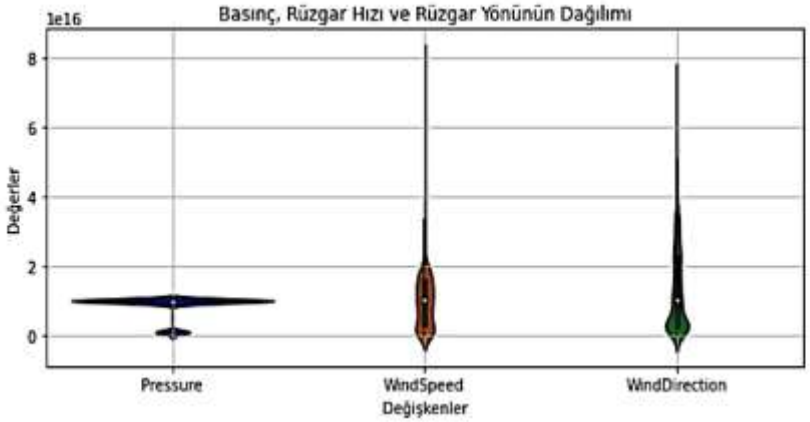
```
Kutu grafiği örneği
```

[17]:
```
import pandas as pd
import seaborn as sns
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
plt.figure(figsize=(8, 4))
sns.boxplot(x="DayOfWeek", y="Temperature", data=df, palette="dark")
plt.title("Haftanın Günlerine Göre Sıcaklık Dağılımları")
plt.xlabel("Day Of Week")
plt.ylabel("Temperature (°C)")
plt.grid(True)
plt.tight_layout()
plt.show()
```



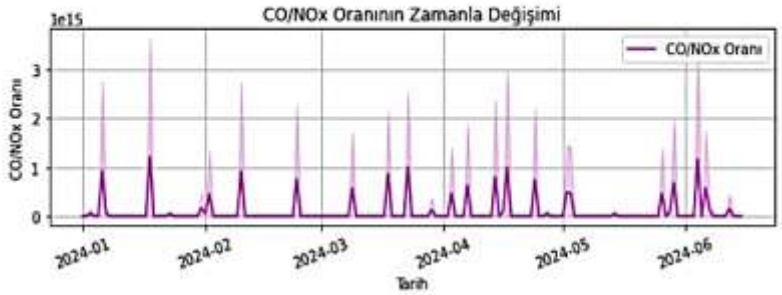
Keman grafiği örneği

[18]:
```
import pandas as pd
import seaborn as sns
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
df_violin = df[["Pressure", "WindSpeed", "WindDirection"]].melt(var_name="Değişken", value_name="Değer")
plt.figure(figsize=(8, 4))
sns.violinplot(x="Değişken", y="Değer", data=df_violin, palette="dark", inner="box")
plt.title("Basınç, Rüzgar Hızı ve Rüzgar Yönünün Dağılımı")
plt.xlabel("Değişkenler")
plt.ylabel("Değerler")
plt.grid(True)
plt.tight_layout()
plt.show()
```



Çizgi grafiği örneği

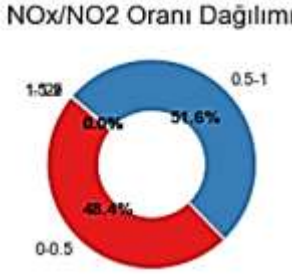
[20]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# tarihin dönüşümü
df['Date'] = pd.to_datetime(df['Date'])
plt.figure(figsize=(8, 3))
sns.lineplot(data=df, x='Date', y='CO_NOx_Ratio', color='m', label='CO/NOx Oranı') #color = m --> margaritanın baş harfi
plt.title('CO/NOx Oranının Zamanla Değişimi')
plt.xlabel('Tarih')
plt.ylabel('CO/NOx Oranı')
plt.xticks(rotation=20)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



Çubuk grafiği örneği

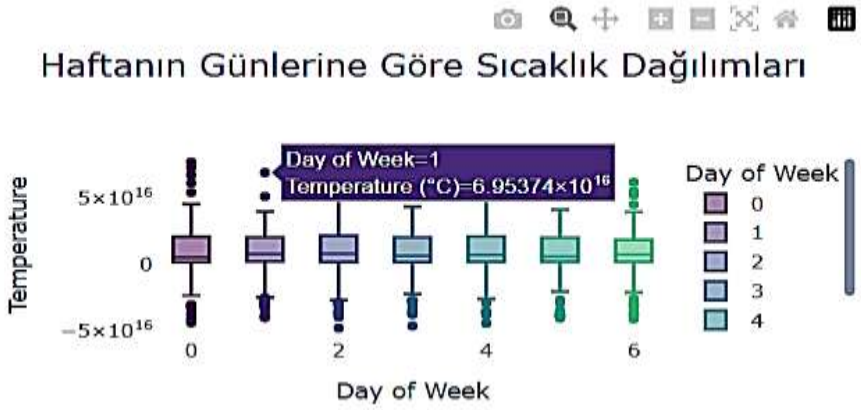
[22]:
```
import pandas as pd
import seaborn as sns
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# zamanı dönüştürüyoruz
df['Time'] = pd.to_datetime(df['Time'], format='%H:%M:%S', errors='coerce')
df['Hour'] = df['Time'].dt.hour
# ortalama hava kalitesinin hesabı (saatlik)
hourly_avg = df.groupby('Hour')['AirQualityIndex'].mean().reset_index()
# seaborn stili
sns.set(style="whitegrid")
plt.figure(figsize=(8, 3))
sns.barplot(x='Hour', y='AirQualityIndex', data=hourly_avg, palette='Spectral')
plt.title('Hava Kalitesi Endeksinin Gün İçindeki Değişimi (Saatlik)', fontsize=15)
plt.xlabel('Saat (0-23)')
plt.ylabel('Ortalama AirQualityIndex')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```


```
import numpy as np
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# NOx/NO2 oran grupları oluşturma
bins = [0, 0.5, 1, 1.5, 2, 5]
labels = ['0-0.5', '0.5-1', '1-1.5', '1.5-2', '2+']
df['NOx_NO2_Group'] = pd.cut(df['NOx_NO2_Ratio'], bins=bins, labels=labels, include_lowest=True)
# grup sayıları
group_counts = df['NOx_NO2_Group'].value_counts().sort_index()
colors = sns.color_palette("Set1", len(group_counts))
# donut grafiği oluşturma
plt.figure(figsize=(4, 3))
wedges, texts, autotexts = plt.pie(group_counts,
                                   labels=group_counts.index,
                                   autopct='%1.1f%%',
                                   startangle=140,
                                   colors=colors,
                                   wedgeprops=dict(width=0.5))  #donut efektinin çizimi
plt.setp(autotexts, size=12, weight="bold", color="black")
plt.title("NOx/NO2 Oranı Dağılımı", fontsize=17)
plt.tight_layout()
plt.show()
```



- **PLOTLY KÜTÜPHANESİ İLE VERİ GÖRSELLEŞTİRMESİ**

```
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")

box = px.box(df, x="DayOfWeek", y="Temperature",
             title="Haftanın Günlerine Göre Sıcaklık Dağılımları",
             labels={"DayOfWeek": "Day of Week", "Temperature": "Temperature (°C)"},
             color="DayOfWeek", # Her gün için farklı renkler
             color_discrete_sequence=px.colors.sequential.Viridis)  # Renk paleti
box.update_layout(
    xaxis_title="Day of Week",
    yaxis_title="Temperature",
    plot_bgcolor='rgba(0,0,0,0)',  #arka planın renksiz olması
    showlegend=True,
    title_font_size=20,
    title_font_color="darkblue",
    width=500,
    height=300 )
box.show()
```



Keman grafiği örneği

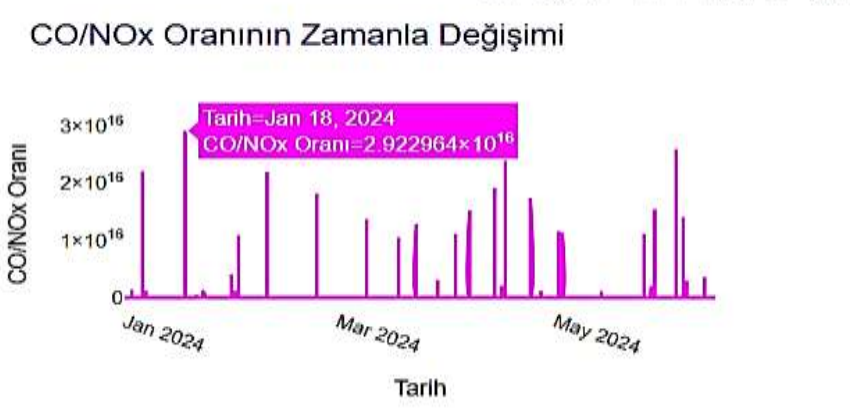
[31]:
```
import pandas as pd
import plotly.express as px
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# dönüştürme işlemi
df_violin = df[["Pressure", "WindSpeed", "WindDirection"]].melt(var_name="Değişken", value_name="Değer")

keman = px.violin(df_violin, x="Değişken", y="Değer", color="Değişken",
                  title="Basınç, Rüzgar Hızı ve Rüzgar Yönünün Dağılımı",
                  labels={"Değişken": "Değişkenler", "Değer": "Değerler"},
                  color_discrete_sequence=px.colors.qualitative.Dark24)
keman.update_layout(
    title_font_size=22,
    title_font_color="darkblue",
    plot_bgcolor='rgba(0,0,0,0)',
    xaxis_title="Değişkenler",
    yaxis_title="Değerler",
    showlegend=True,
    font=dict(family="Arial", size=12, color="black")
)
keman.update_layout(
    width=500,
    height=300
)
keman.show()
```

```
keman.show()
```


```
Çizgi grafiği örneği
```

[32]:
```
import pandas as pd
import plotly.express as px
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# tarihi dönüştürme
df['Date'] = pd.to_datetime(df['Date'])
çizgi = px.line(df, x='Date', y='CO_NOx_Ratio',
              title='CO/NOx Oranının Zamanla Değişimi',
              labels={"Date": "Tarih", "CO_NOx_Ratio": "CO/NOx Oranı"},
              line_shape='spline',
              color_discrete_sequence=['magenta'])
çizgi.update_layout(
    title_font_size=19,
    title_font_color="darkblue",
    xaxis_title="Tarih",
    yaxis_title="CO/NOx Oranı",
    xaxis=dict(tickangle=20),
    plot_bgcolor='rgba(0,0,0,0)',
    showlegend=True,
    font=dict(family="Arial", size=12, color="black"),
    width=500,
    height=300)
çizgi.show()
```

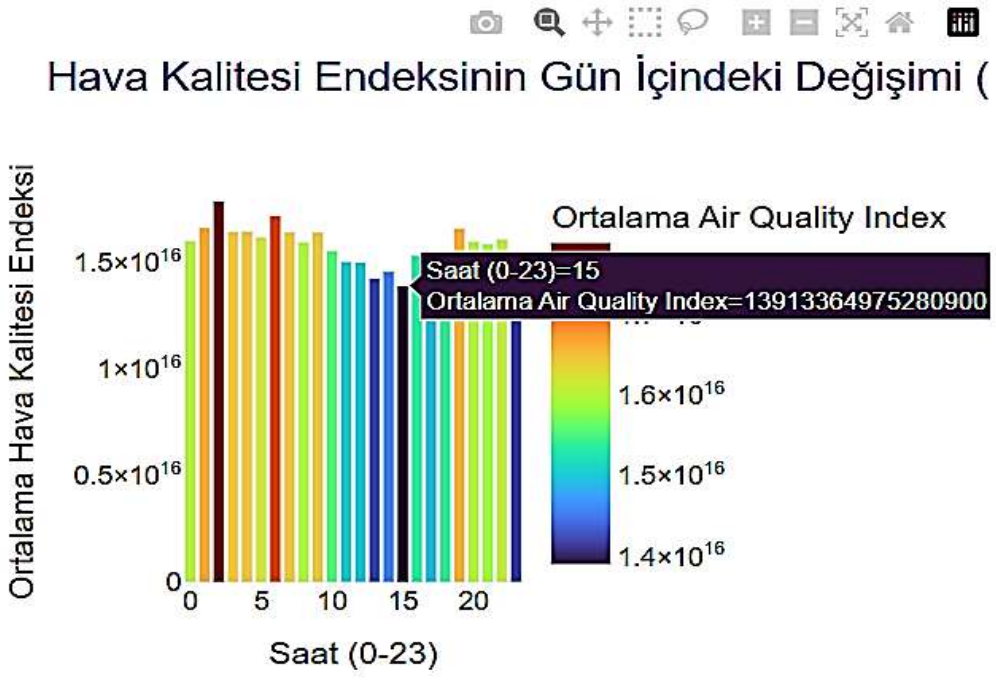
```
    height=300)
çizgi.show()
```



Çubuk grafiği örneği

[35]:
```
import pandas as pd
import plotly.express as px
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# zamanın dönüşümü
df['Time'] = pd.to_datetime(df['Time'], format='%H:%M:%S', errors='coerce')
df['Hour'] = df['Time'].dt.hour
# ortalama hava kalitesi endeksi (saatlik)
hourly_avg = df.groupby('Hour')['AirQualityIndex'].mean().reset_index()
çubuk = px.bar(hourly_avg,
               x='Hour',
               y='AirQualityIndex',
               color='AirQualityIndex',
               color_continuous_scale='Turbo',
               title='Hava Kalitesi Endeksinin Gün İçindeki Değişimi (Saatlik)',
               labels={'Hour': 'Saat (0-23)', 'AirQualityIndex': 'Ortalama Air Quality Index'})
çubuk.update_layout(
    title_font_size=22,
    title_font_color='darkblue',
    xaxis_title='Saat (0-23)',
    yaxis_title='Ortalama Hava Kalitesi Endeksi',
    plot_bgcolor='rgba(0,0,0,0)',
    font=dict(family='Arial', size=14, color='black'),
    width=500,
    height=400 )
çubuk.show()
```

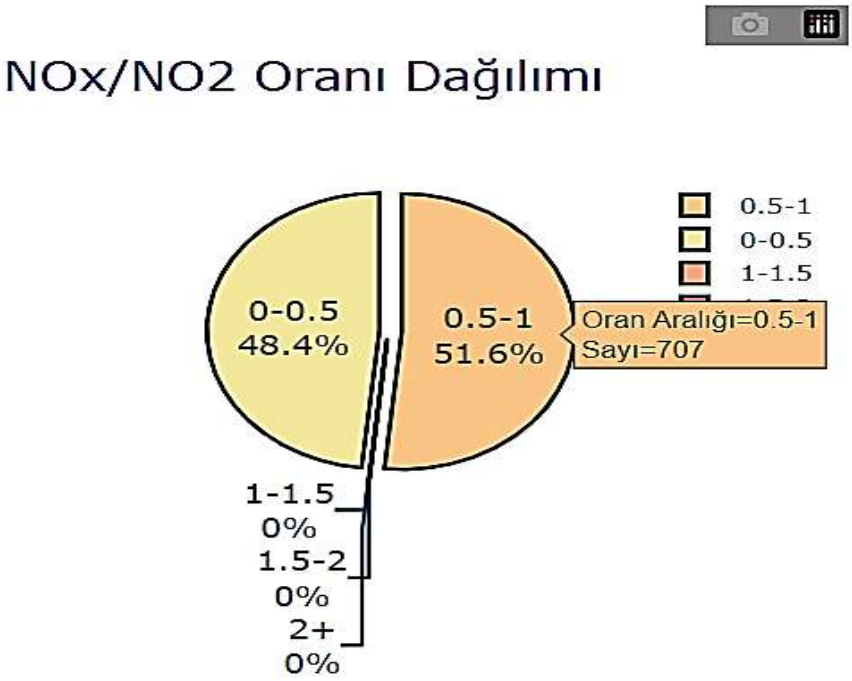
```
çubuk.show()
```



Pasta grafiği örneği

[38]:
```
import pandas as pd
import plotly.express as px
df = pd.read_excel(r"C:\Users\hadim\Desktop\AirQualityData.xlsx")
# NOx/NO2 oran grupları
bins = [0, 0.5, 1, 1.5, 2, 5]
labels = ['0-0.5', '0.5-1', '1-1.5', '1.5-2', '2+']
df['NOx_NO2_Group'] = pd.cut(df['NOx_NO2_Ratio'], bins=bins, labels=labels, include_lowest=True)
# grup sayılarını hesaplama
group_counts = df['NOx_NO2_Group'].value_counts().sort_index().reset_index()
group_counts.columns = ['Oran Aralığı', 'Sayı']
pasta = px.pie(group_counts,
             names='Oran Aralığı',
             values='Sayı',
             color='Oran Aralığı',
             color_discrete_sequence=px.colors.sequential.Sunset,
             hole=0,
             title="NOx/NO2 Oranı Dağılımı")
# dilimleri ayırma
pasta.update_traces(textinfo='percent+label',
                 pull=[0.05, 0.08, 0.06, 0.1, 0.07],
                 marker=dict(line=dict(color='black', width=2)),
                 textfont_size=15)
pasta.update_layout(
    title_font_size=22,
    title_font_color='darkblue',
    plot_bgcolor='rgba(0,0,0,0)',
    paper_bgcolor='rgba(0,0,0,0)',
    showlegend=True,
    width=400,
    height=400 )
```

```
    height=400 )
pasta.show()
```



Kaynakça

Verinin görselleştirmesi için araştırdığım tablo çeşitlerinin özellikleri ve kütüphane çeşitlerinin kullanım farklarını anlamak için yararlandığım kaynaklar şu şekildedir:

Seçkin yayınları – Temel İstatistik Yöntemler Kitabı / (Prof. Dr. Özkan Ünver,Prof. Dr. Hamza Gamgam,Prof. Dr. Bülent Altunkaynak)

Seçkin yayınları – Veri Madenciliği Yöntemleri ve R Uygulamaları Kitabı / (Prof. Dr. Bülent Altunkaynak) grafikleri araştırdım.

Kendi hocalarımın kaynaklarından yararlanırken aynı zamanda sizin ilettiğiniz,

https://seaborn.pydata.org/
https://plotly.com/python/
https://matplotlib.org/stable/gallery/index.html

Bu kaynakları da araştırarak verinin görselleştirmesi için grafik türlerini ve programlama da kullandığımız grafik kütüphanelerini raporladım.

Tüm bu işlemlerin ardından Knıme ve Python için görselleştirme çalışmalarımı tamamlayıp, raporlamış bulunmaktayım.

Kübra Nur Babacan