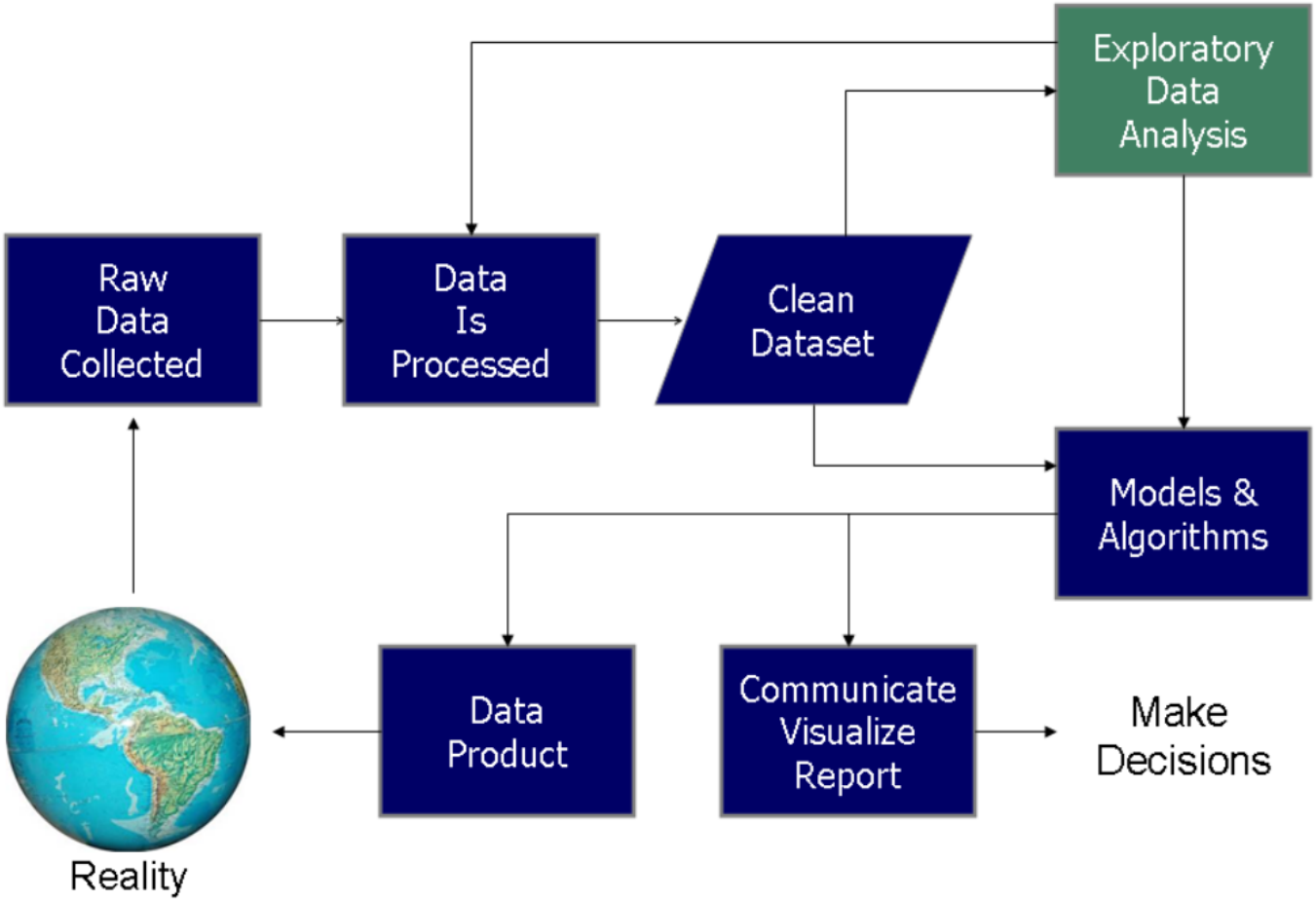
# Keşifçi Veri Analizi (EDA)

**Hafta 5**

Dr. Ögr. Üyesi Nihan ÖZBALTAN

# Data Science Process



# Keşifçi Veri Analizi (EDA)

Keşifçi veri analizi (**Exploratory Data Analysis**), faydalı bilgiler bulma, sonuçları bilgilendirme ve karar vermeyi destekleme amacıyla verileri inceleme ve **görselleştirme** gibi işlemleri kapsayan bir süreçtir.

Veri bilimi projesine başlamadan önce elimizdeki verilerin varyansından, yanlılığından, dağılım türlerinden, verilerin türlerinden, yapısal olup olmadığından emin olmak gerekir.

Bir lokantanın hesap ödeme esnasında müşterilerden edindiği verileri kullanarak keşifçi veri analizini tamamlayacağız.

# Tips Veri Çerçevesi Tanıtımı

Gerekli kütüphaneleri ve veri çerçevesini çekirdeğe dahil ettik.

**total_bill** : ödenen hesap miktarı (usd)
**tip** : bırakılan bahşiş (usd)
**sex** : hesabı ödeyen kişinin cinsiyeti
**smoker** : sigara kullanıp kullanmadığı
**day** : hangi gün ziyaret ettiği
**time** : hangi zaman dilimi hizmet aldığı
**size** : alınan servisin porsiyonları

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

df = sns.load_dataset('tips')
df.head()
```

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |
| **3** | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 |

# Keşifçi Veri Analizi

**244 gözlem** ve **7 değişkenden** oluşuyor.

**info( )** fonksiyonu ile veri çerçevesinin bellekte kapladığı alanı ve değişken tiplerini görüntülüyoruz.

Tüm değişkenlerde **244** adet eksik olmayan veri var.

```
df.shape
```

```
(244, 7)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
total_bill    244 non-null float64
tip           244 non-null float64
sex           244 non-null category
smoker        244 non-null category
day           244 non-null category
time          244 non-null category
size          244 non-null int64
dtypes: category(4), float64(2), int64(1)
memory usage: 7.3 KB
```

# Eksik Gözlemler ve Temel İstatistik

Teyit etmek amaçlı değişkenlerde eksik veri olup olmadığını **isna( ).sum( )** fonksiyonuyla görüntüledik.

**Dikkat**, eksik gözlem her zaman **NaN** formatında değildir, bazen boşluk bazen de karakterler girilebilir eksik gözlem için.

Bu durumda eksik verileri nasıl tespit edebiliriz?

```
df.isna().sum()
```

```
total_bill    0
tip           0
sex           0
smoker        0
day           0
time          0
size          0
dtype: int64
```

# Eksik Gözlem Bulmak Üzerine Alternatif Yöntem

Değişkenlerde hangi benzersiz değerlerin olduğunu görmek için **unique()** fonksiyonu kullanılabilir.

Değişkenlerde kaç adet benzersiz değer olduğunu görmek için **nunique()** fonksiyonu kullanılabilir.

Eksik veya anormal verileri tespit etmek için başka yöntemler de mevcuttur.

```
df["sex"].unique()                    # a
```

```
[Female, Male]
Categories (2, object): [Female, Male]
```

```
df["smoker"].unique()                 #
```

```
[No, Yes]
Categories (2, object): [No, Yes]
```

```
df["size"].unique()
```

```
array([2, 3, 4, 1, 6, 5], dtype=int64)
```

```
df["size"].nunique()
```

```
6
```

# Verileri Sorgulama

Hesaplar için ödenen ortalama değer 19.78 usd, standart sapması **8.9** usd.

Hesapların yanında ödenen bahşişlerin ortalaması **2.9** usd iken bahşişlerin medyan değeri de **2.9** usd.

Bahşişlerin standart sapması **1.38** usd.

```
df["total_bill"].mean()
```

```
19.78594262295082
```

```
df["total_bill"].std()
```

```
8.902411954856856
```

```
df["tip"].mean()   # bır
```

```
2.99827868852459
```

```
df["tip"].median()   # b
```

```
2.9
```

```
df["tip"].std()   # bıra
```

```
1.3836381890011822
```

# Verileri Sorgulama

**describe( )** fonksiyonu veri çerçevesindeki **sayısal değişkenler** için temel istatistik değerlerini gözlemlememize yardımcı olur.

**Sayısal değişkenlerin dağılımları** hakkında fikir sahibi olmak için idealdir. Daha okunabilir şekli için **.T** komutuyla **transpozu** alınmıştır

```
df.describe().T # sadece sayısal verileri açıklar.
```

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **total_bill** | 244.0 | 19.785943 | 8.902412 | 3.07 | 13.3475 | 17.795 | 24.1275 | 50.81 |
| **tip** | 244.0 | 2.998279 | 1.383638 | 1.00 | 2.0000 | 2.900 | 3.5625 | 10.00 |
| **size** | 244.0 | 2.569672 | 0.951100 | 1.00 | 2.0000 | 2.000 | 3.0000 | 6.00 |

# Verileri Sorgulama

Veri çerçevesini cinsiyetlere göre kırdırarak, **hangi cinsiyetlerin ne kadar hesap ödediği** hakkında temel istatistik bilgilerini sorgulamak istiyoruz.

Bu yüzden **groupby( )** fonksiyonunu ve **describe( )** fonksiyonlarını birlikte kullanabiliriz.

```
df.groupby(["sex"]).describe()["total_bill"]
```

| sex | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Male** | 157.0 | 20.744076 | 9.246469 | 7.25 | 14.00 | 18.35 | 24.71 | 50.81 |
| **Female** | 87.0 | 18.056897 | 8.009209 | 3.07 | 12.75 | 16.40 | 21.52 | 44.30 |

# Verileri Sorgulama

Lokantada satın alınan porsiyonlara göre bırakılan bahşiş miktarlarının ortalamasını almak istiyoruz.

**apply( )** ve **lambda** fonksiyonlarını kullanabiliriz.

Porsiyon arttıkça bahşiş ortalamasının arttığını düşünüyoruz, **korelasyon katsayısını** görüntüleyelim.

```
df.groupby('size')['tip'].apply(lambda x: np.mean(x))
```

```
size
1    1.437500
2    2.582308
3    3.393158
4    4.135405
5    4.028000
6    5.225000
Name: tip, dtype: float64
```

```
df.corr()["total_bill"]["size"]
```

```
0.5983151309049025
```

# Verileri Sorgulama

**Erkek müşterilerin 8$** üstünde bahşiş verenlerini filtrelemek istiyoruz. Sadece **2** müşteri varmış.

```
df[(df["sex"] == "Male") & (df["tip"] > 8)]
```

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **170** | 50.81 | 10.0 | Male | Yes | Sat | Dinner | 3 |
| **212** | 48.33 | 9.0 | Male | No | Sat | Dinner | 4 |

**5$** üstünde tip bırakan **kadın müşterileri** ödedikleri **hesap tutarına göre** sıralayalım.

```
df[(df['tip']>5) & (df['sex'] == "Female")].sort_values('total_bill', axis = 0, ascending=False)
```

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **85** | 34.83 | 5.17 | Female | No | Thur | Lunch | 4 |
| **52** | 34.81 | 5.20 | Female | No | Sun | Dinner | 4 |
| **155** | 29.85 | 5.14 | Female | No | Sun | Dinner | 5 |
| **214** | 28.17 | 6.50 | Female | Yes | Sat | Dinner | 3 |

# Verileri Sorgulama

Diğer bir yöntem **query( )** fonksiyonu ile **6$ dan daha çok bahşiş bırakan** ve aynı zamanda **5 porsiyondan daha küçük ürünler satın alan müşterilerin cinsiyetlerini**, **bahşişlerini** **ve** **porsiyon sayılarını** yazdıralım.

```
df_filtered = df.query('tip>6 & size<5')[["sex", "tip", "size"]]
df_filtered
```

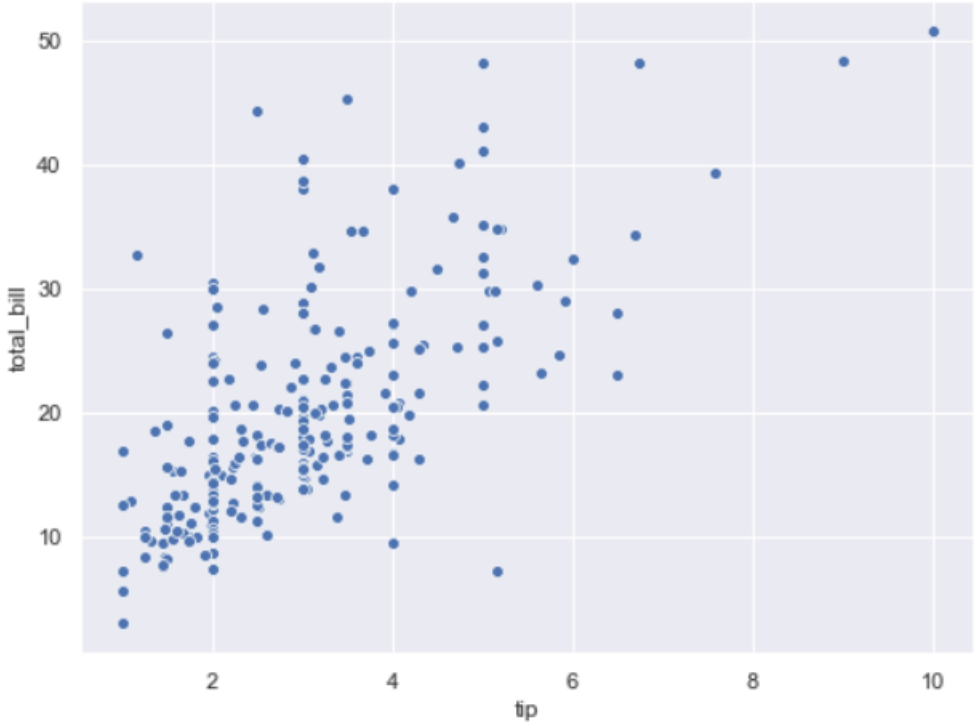
| | sex | tip | size |
|---|---|---|---|
| **23** | Male | 7.58 | 4 |
| **59** | Male | 6.73 | 4 |
| **170** | Male | 10.00 | 3 |
| **183** | Male | 6.50 | 4 |
| **212** | Male | 9.00 | 4 |
| **214** | Female | 6.50 | 3 |

# Korelasyon Görselleştirme

**Sürekli verileri** görselleştirmek için **dağılım grafiği** sıklıkla kullanılır.

Görüldüğü üzere bırakılan bahşiş ve ödenen hesap miktarı arasında **iyi** ve **pozitif** bir korelasyon vardır.

```
sns.scatterplot(x = "tip", y = "total_bill", data = df);
```

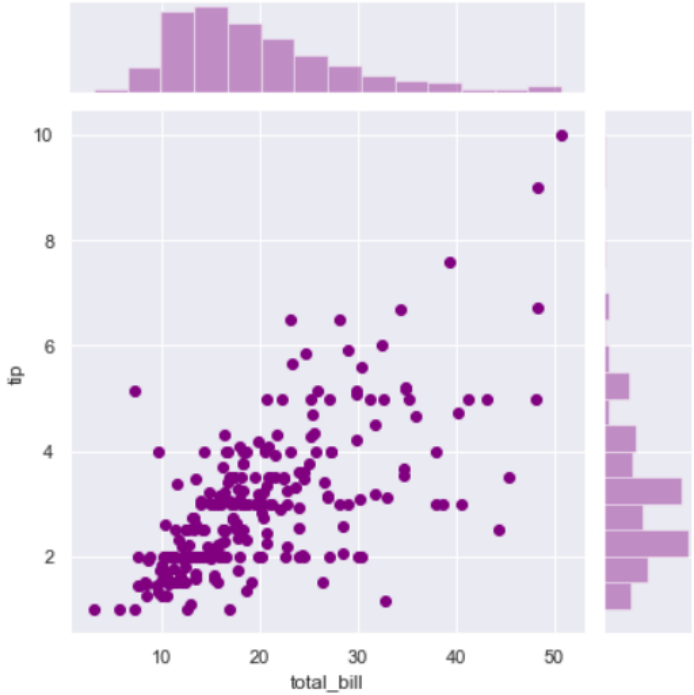


# Korelasyon Görselleştirme

Özellikle üst üste binen gözlemler okunabilirliği düşürebilir.

Bu yüzden **yoğunluklu dağılım grafiği** çizdirilerek gözlemlerin hangi aralıkta ne kadar yoğun olduğunu da gözlemleyebiliriz.

```
sns.jointplot(x = "total_bill", y = "tip", data = df, color="purple");
```



# Kovaryans ve Korelasyon Analizi

```
df.cov()
```

| | total_bill | tip | size |
|---|---|---|---|
| **total_bill** | 79.252939 | 8.323502 | 5.065983 |
| **tip** | 8.323502 | 1.914455 | 0.643906 |
| **size** | 5.065983 | 0.643906 | 0.904591 |

Kovaryans matrisi bize; ödenen hesap ile bahşiş arasında **pozitif**, porsiyon ile bahşiş arasında **pozitif** ve ödenen hesap ile porsiyon arasında **pozitif** bir ilişkinin var olduğunu göstermektedir.

<u>Ancak ilişkinin şiddeti hakkında bir yorum yapmak için korelasyon matrisine bakarız.</u>

```
df.corr()
```

| | total_bill | tip | size |
|---|---|---|---|
| **total_bill** | 1.000000 | 0.675734 | 0.598315 |
| **tip** | 0.675734 | 1.000000 | 0.489299 |
| **size** | 0.598315 | 0.489299 | 1.000000 |

Ödenen hesap ile bahşiş arasında **0.67** korelasyon vardır. **Güçlü ve pozitif bir ilişki.**

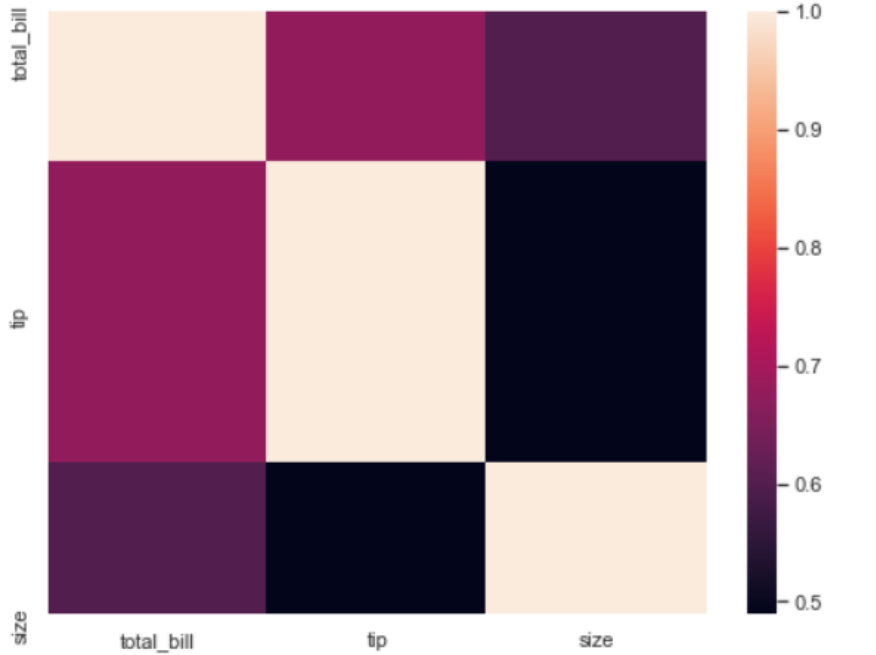
# Korelasyon ve Isı Haritası

Korelasyon matrisini anlayabilmek çokça değişkenin bulunduğu veri çerçevelerinde zorlaşabilir.

**Isı haritası** ile renklendirilmiş korelasyon matrisi de bize değişkenler arasındaki ilişki hakkında fikir verecektir.

Korelasyon matrisinin ısı haritası aynı bilgiyi görsel olarak ifade eder.

```
corr = df.corr()
sns.heatmap(corr,
            xticklabels=corr.columns.values,
            yticklabels=corr.columns.values);
```



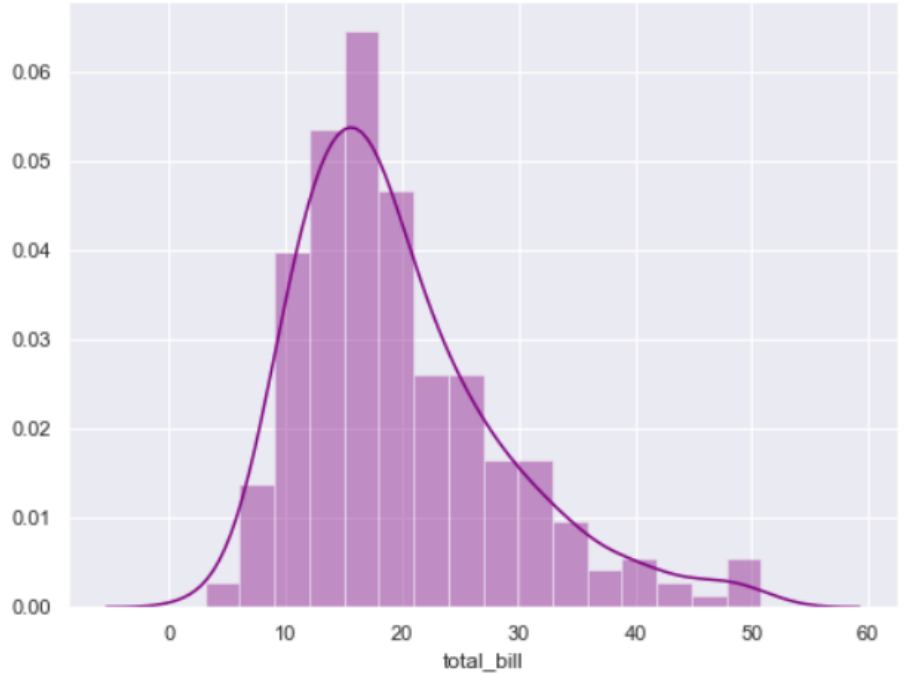
# Dağılım Görselleştirme

Ödenen hesapların dağılımını görmek istediğimizde **histogram** çizdirebiliriz.

Histogram bize dağılımın normal, sağa ve sola yaslanmış olduğu hakkında fikir verir.

Dağılımın nerede **yoğunlaştığı** hakkında bilgi ediniriz.

```
: sns.distplot(df["total_bill"], bins=16, color="purple");
```

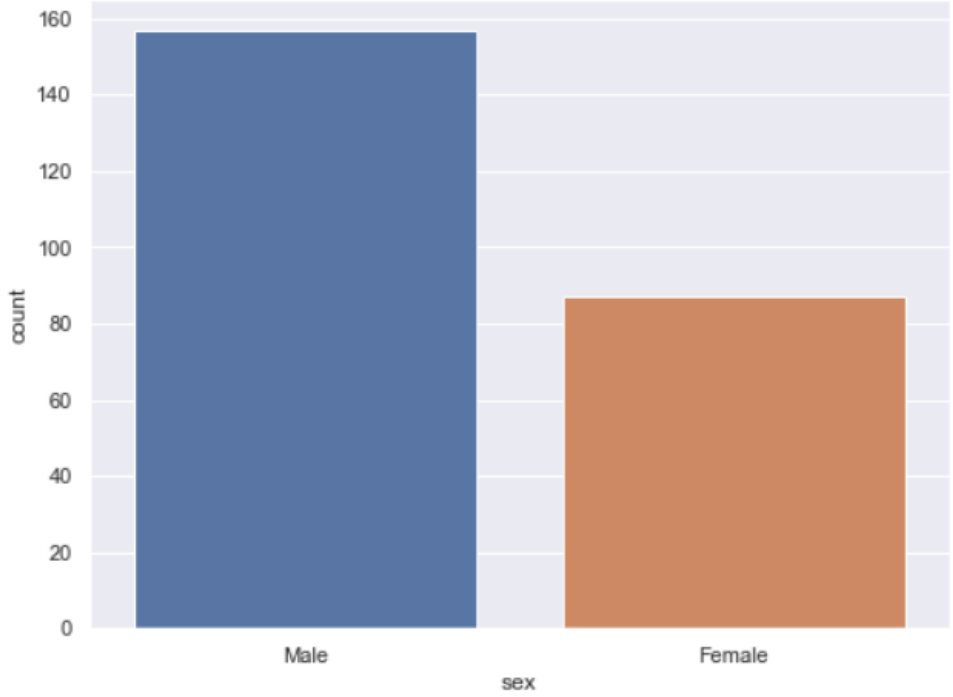


# Oran Görselleştirme

Hesap ödeyen müşterilerin kadın ve erkek oranlarını görüntülemek istersek **kutu grafiği** (**barplot**) çizdirebiliriz.

Grafikte görüldüğü üzere yaklaşık 155 erkek müşteri ve 85 kadın müşterinin kaydı tutulmuştur diyebiliriz.

```
sns.countplot(x = "sex", data = df);
```

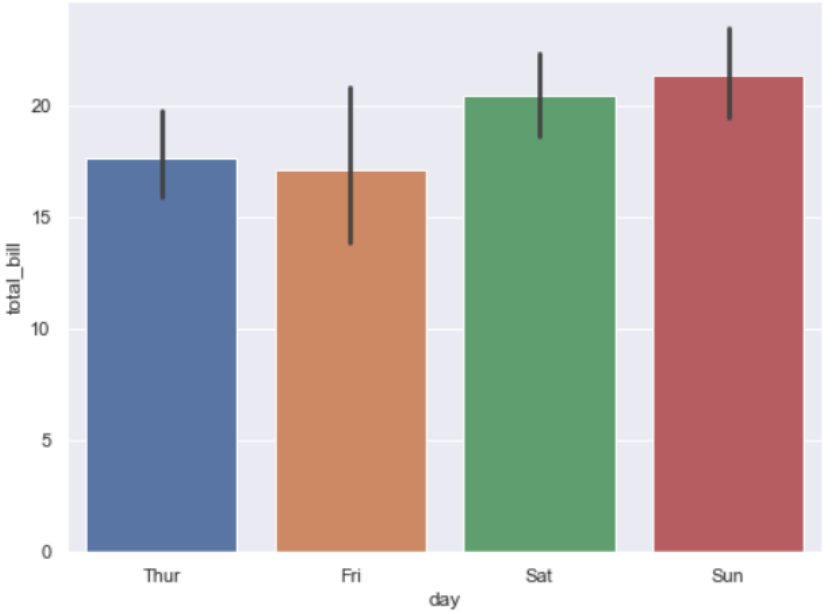


# Oran Görselleştirme

**Kutu grafiği** ile hangi günler yaklaşık olarak ödenen hesap miktarlarının hangi oranlarda dalgalandığını görebiliriz.

Ödenen hesapların maksimum olduğu günler hafta sonu gibi görünüyor.

```
sns.barplot(x = "day", y = "total_bill", data = df);
```

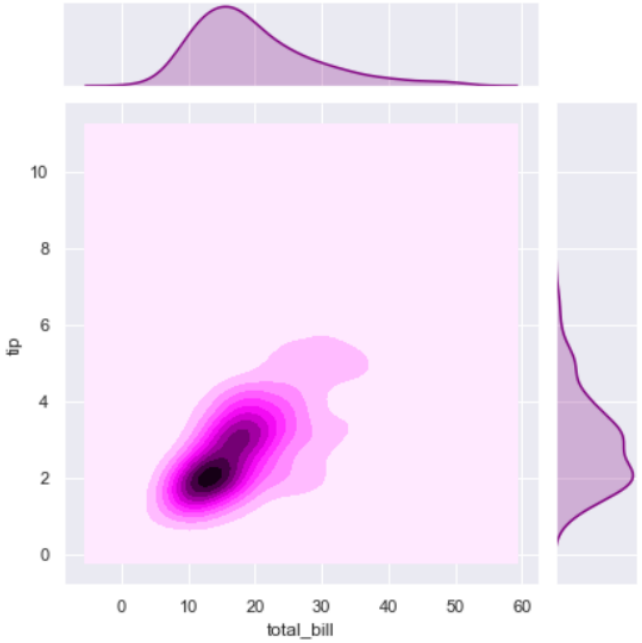


# Yoğunluk Görselleştirme

Bahşiş ve ödenen hesap miktarı arasındaki ilişkiyi **yoğunluk grafiği** çizdirerek de gözlemleyebiliriz.

Grafiğe bakarak bahşişin en sık gözlemlendiği miktar **2 usd** civarıyken ödenen hesap miktarının en sık gözlemlendiği miktar **18 usd** civarıdır çıkarımında bulunabiliriz.

```
sns.jointplot(x = df["total_bill"], y = df["tip"],kind = "kde", color="purple");
```



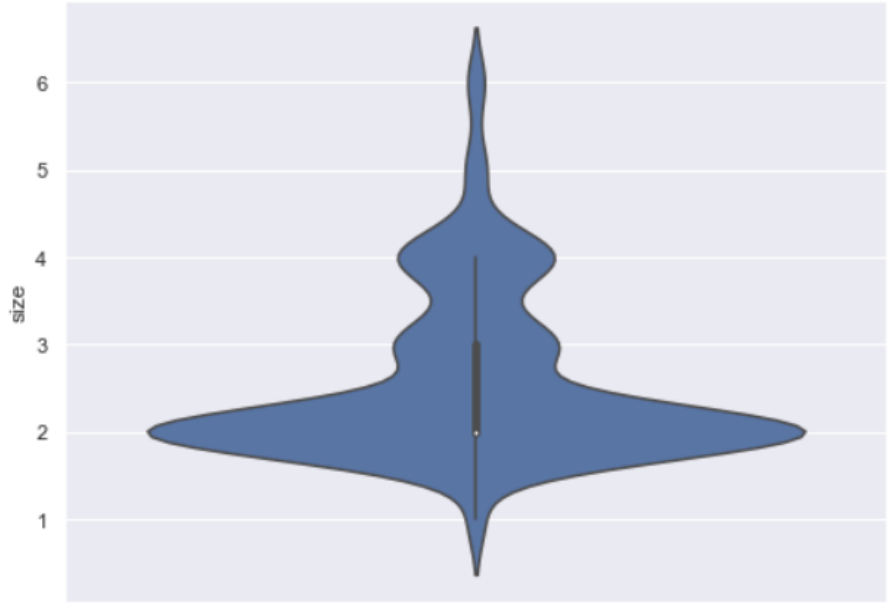
# Dağılım Görselleştirme

Tıpkı **dağılım grafiği** gibi sürekli verileri görselleştirmede ve dağılımları hakkında fikir sahibi olma konusunda **keman grafiği** bize yardımcı olacaktır.

Satın alınan porsiyonların **dağılımı** sağ tarafta gözlemlenmektedir.

Şişkin olan kısım en yoğunluklu olarak **2 porsiyonluk** siparişlerin ısmarlandığını temsil eder.

```
sns.violinplot(y = "size", data = df);
```



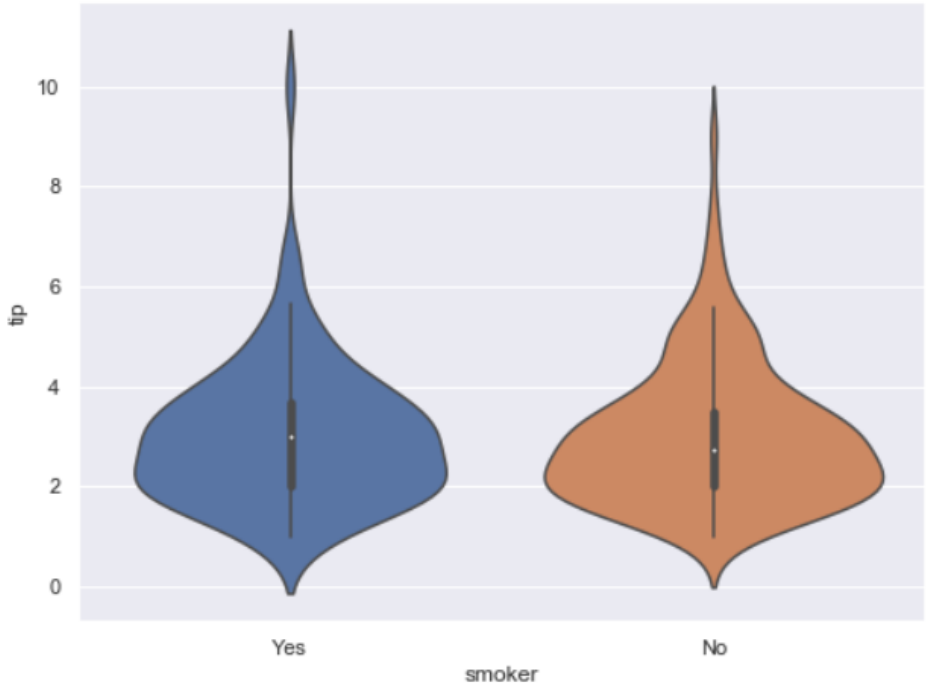
# Koşullu Dağılım Görselleştirme

Sigara içen ve içmeyen müşterilerin bıraktıkları bahşişlerin dağılımı hakkında fikir sahibi olmak için **keman grafiği** çizdirilebilir.

Aralarında neredeyse bir fark yok gibi görünmektedir.

*Bir fark olsa dahi bu farkın nedensellik (**causality**) bakımından incelenmesi ve **anlamlı olup olmadığının** araştırılması gerekir.*

```
sns.violinplot(x = "smoker", y = "tip", data = df);
```

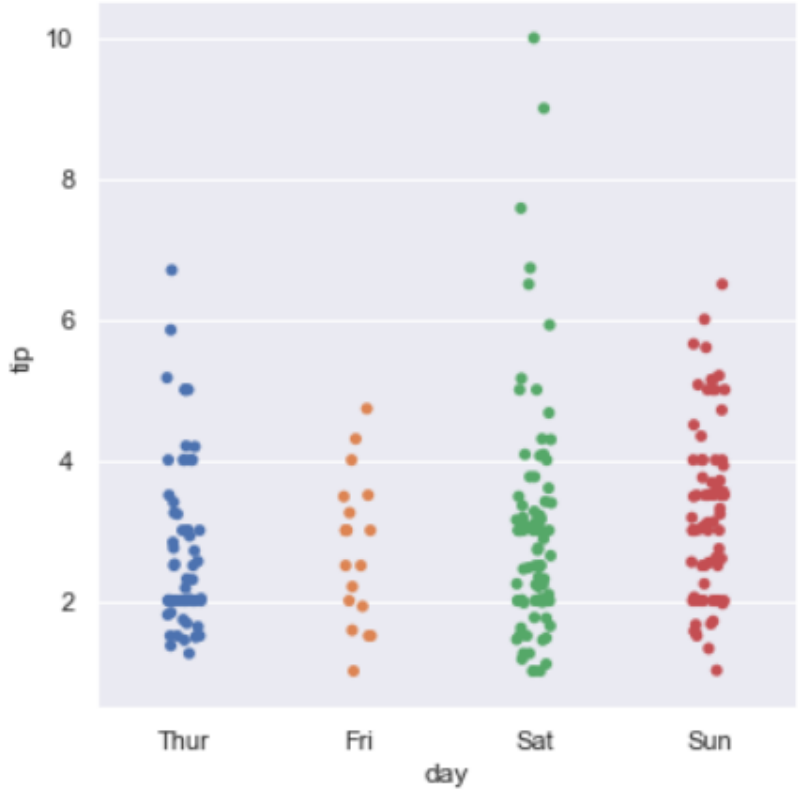


# Dağılım Görselleştirme

Hangi günler garsonların daha çok bahşiş aldığını görüntülemek istersek **CatPlot** çizdirebiliriz.

CatPlot, kategorik verileri, sürekli verilerle  görselleştirmek için ideal bir görselleştirme aracıdır.

```
sns.catplot(x = "day", y = "tip", data = df);
```



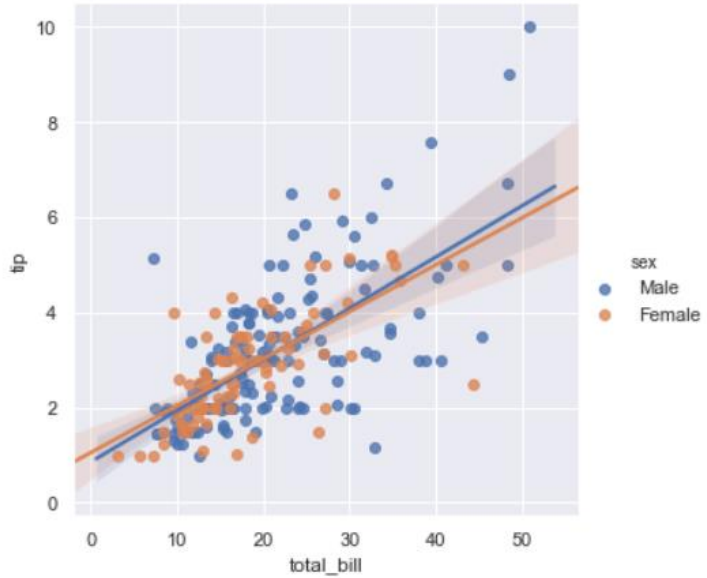
# Dağılım Görselleştirme

Hesap ödeyen erkeklerin ve kadınların bıraktığı bahşiş miktarları ve ödedikleri hesap miktarları arasındaki ilişkiye göre bir görselleştirme yapalım.

Kıyaslamak zor olduğu için bize yardımcı olması amacıyla "**doğrusal regresyon çizgisi**" çektik.

*Not : Çizginin etrafındaki soyut katman* ***güven aralığını*** *(Confidence Interval) temsil etmektedir.*

```
sns.lmplot(x = "total_bill", y = "tip", data = df, hue = "sex");
```



# Kaynakça


https://medium.com/@denizkilinc/python-ile-veri-tan%C4%B1maya-ve-temel-i%CC%87statisti%C4%9Fe-dal%C4%B1%C5%9F-7e1028270ac

https://medium.com/bili%C5%9Fim-hareketi/veri-bilimi-i%CC%87%C3%A7in-temel-python-k%C3%BCt%C3%BCphaneleri-2-pandas-dcc12ae01b7d

https://medium.com/bili%C5%9Fim-hareketi/veri-bilimi-i%CC%87%C3%A7in-temel-python-k%C3%BCt%C3%BCphaneleri-1-numpy-750429a0d8e5

https://medium.com/datarunner/veri%CC%87-bi%CC%87li%CC%87mi%CC%87-i%CC%87%C3%A7i%CC%87n-i%CC%87stati%CC%87sti%CC%87k-4c1c72c4158