

# Seaborn Grafik Türleri

Kütüphanelerin eklenmesi ve verinin okunması:

```
import seaborn as sns
from matplotlib import pyplot as plt
import pandas as pd

[2] df = pd.read_csv("train.csv", sep=",")
```

df.info() fonksiyonu ile birlikte toplam satır ve sütun sayısına bakıp, her değişkenin veri tiplerinin integer mi, float mı, kategorik mi (object olarak yazar) olup olmadığına bakıyoruz. 891 tane satır, 12 tane sütun. Sütun değerleri değişkenlerdir.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype  
---  -
 0   PassengerId   891 non-null    int64  
 1   Survived      891 non-null    int64  
 2   Pclass        891 non-null    int64  
 3   Name          891 non-null    object  
 4   Sex           891 non-null    object  
 5   Age           714 non-null    float64 
 6   SibSp         891 non-null    int64  
 7   Parch         891 non-null    int64  
 8   Ticket        891 non-null    object  
 9   Fare          891 non-null    float64 
10   Cabin         204 non-null    object  
11   Embarked      889 non-null    object  
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

dtypes ile her değişkenin sadece veri tiplerini görüp kullanabiliriz.

```
df.dtypes
```

PassengerId	int64
Survived	int64
Pclass	int64
Name	object
Sex	object
Age	float64
SibSp	int64
Parch	int64
Ticket	object
Fare	float64
Cabin	object
Embarked	object
dtype:	object

Size komutu ile toplam veri sayımızı,

Shape komutu ile satır ve sütun sayımızı olduğunu analiz ediyoruz.

```
[6] df.size
```

10692

```
df.shape
```

(891, 12)

Eksik verileri temizleme:

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
<b>PassengerId</b>	891.0	446.000000	257.353842	1.00	223.5000	446.0000	668.5	891.0000
<b>Survived</b>	891.0	0.383838	0.486592	0.00	0.0000	0.0000	1.0	1.0000
<b>Pclass</b>	891.0	2.308642	0.836071	1.00	2.0000	3.0000	3.0	3.0000
<b>Age</b>	714.0	29.699118	14.526497	0.42	20.1250	28.0000	38.0	80.0000
<b>SibSp</b>	891.0	0.523008	1.102743	0.00	0.0000	0.0000	1.0	8.0000
<b>Parch</b>	891.0	0.381594	0.806057	0.00	0.0000	0.0000	0.0	6.0000
<b>Fare</b>	891.0	32.204208	49.693429	0.00	7.9104	14.4542	31.0	512.3292

```
df.isnull().sum()
```

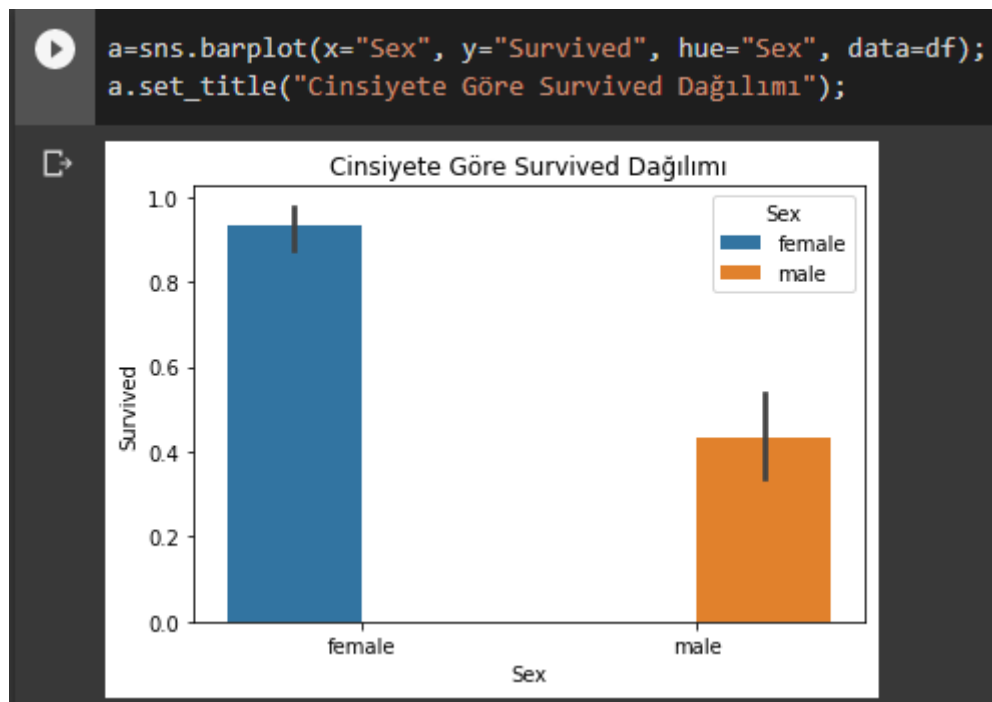
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2
dtype:	int64

```
[10] df = df.dropna()
```

```
df.isnull().sum()
```

PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	0
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	0
Embarked	0
dtype:	int64

Seaborn ile veri görselleştirme: barplot

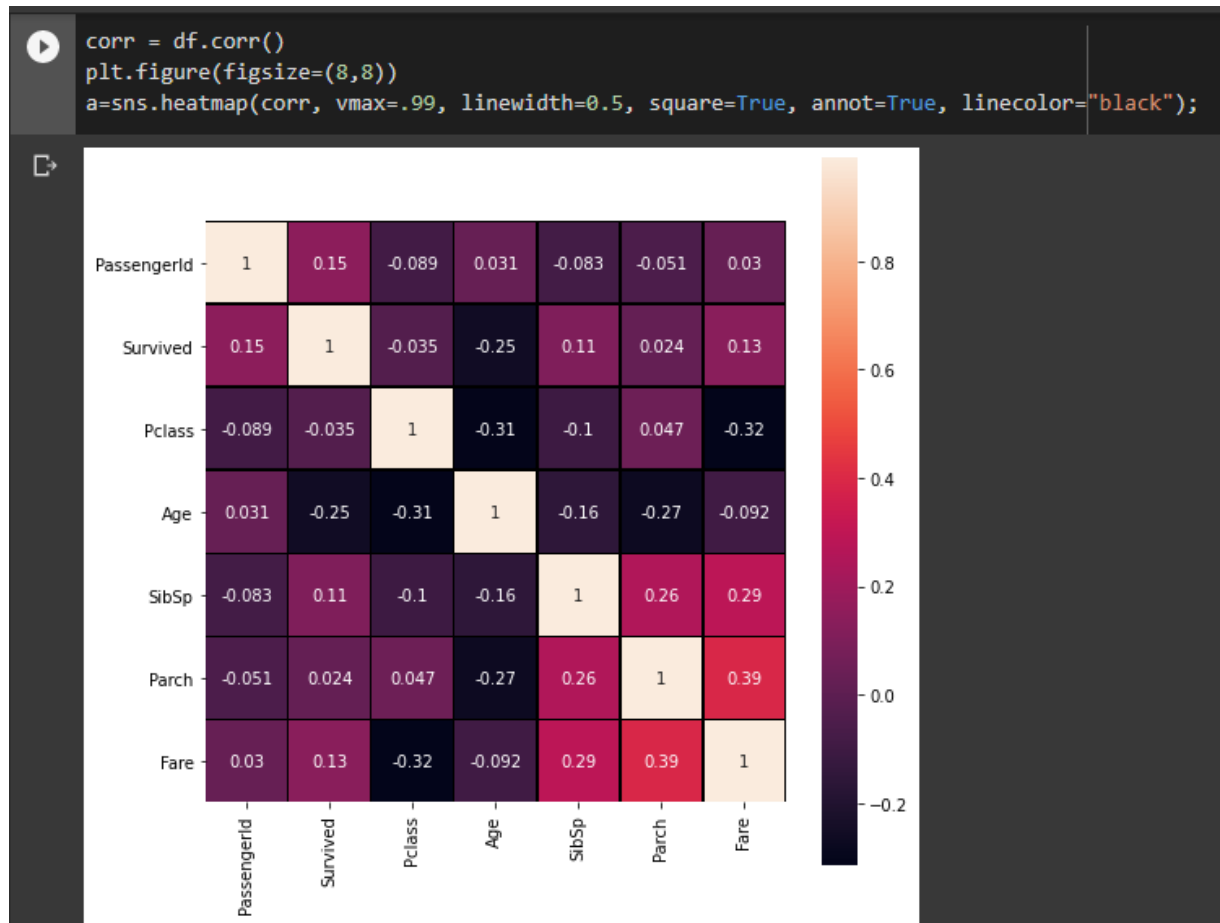


Cinsiyet değişkeninin sınıf frekanslarına baktığımızda kadınların daha çok hayatta kaldığını görselden analiz edebiliyoruz.

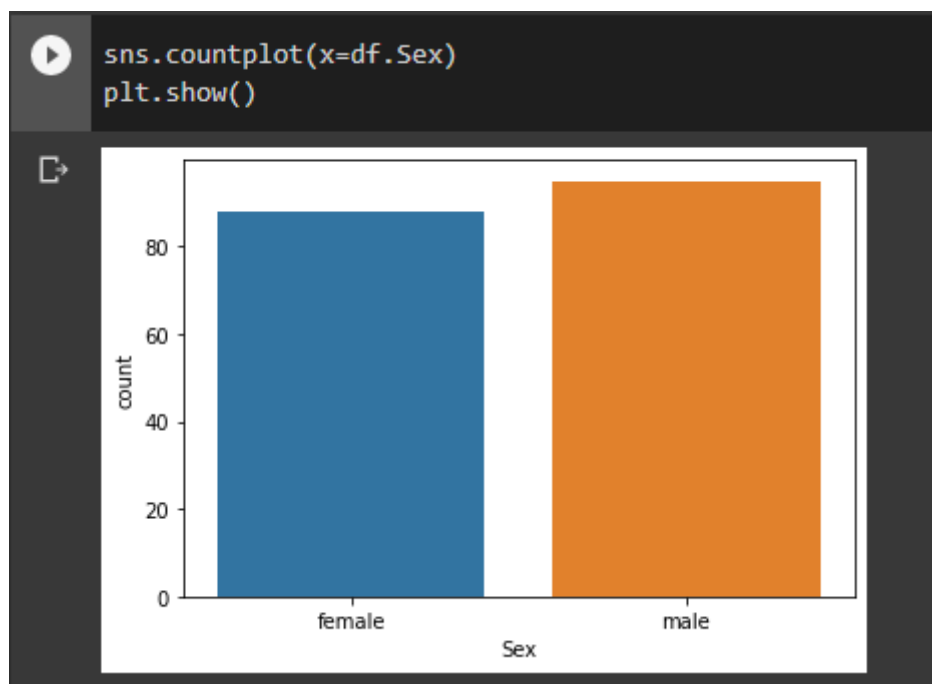
Pairplot, tüm veri setinde sayısal değişkenler arasında çift yönlü ilişkiler çiziyor.



Heatmap grafiđi deđiřkenlerin arasında iliřkinin nasıl olduđunu ve target deđiřkeni ile bađımsız deđiřkenler arasındaki iliřkiyi sayısal olarak ok net bir řekilde analiz edebiliyoruz. Korelasyon analizi iin bu grafik tercih ediliyor.



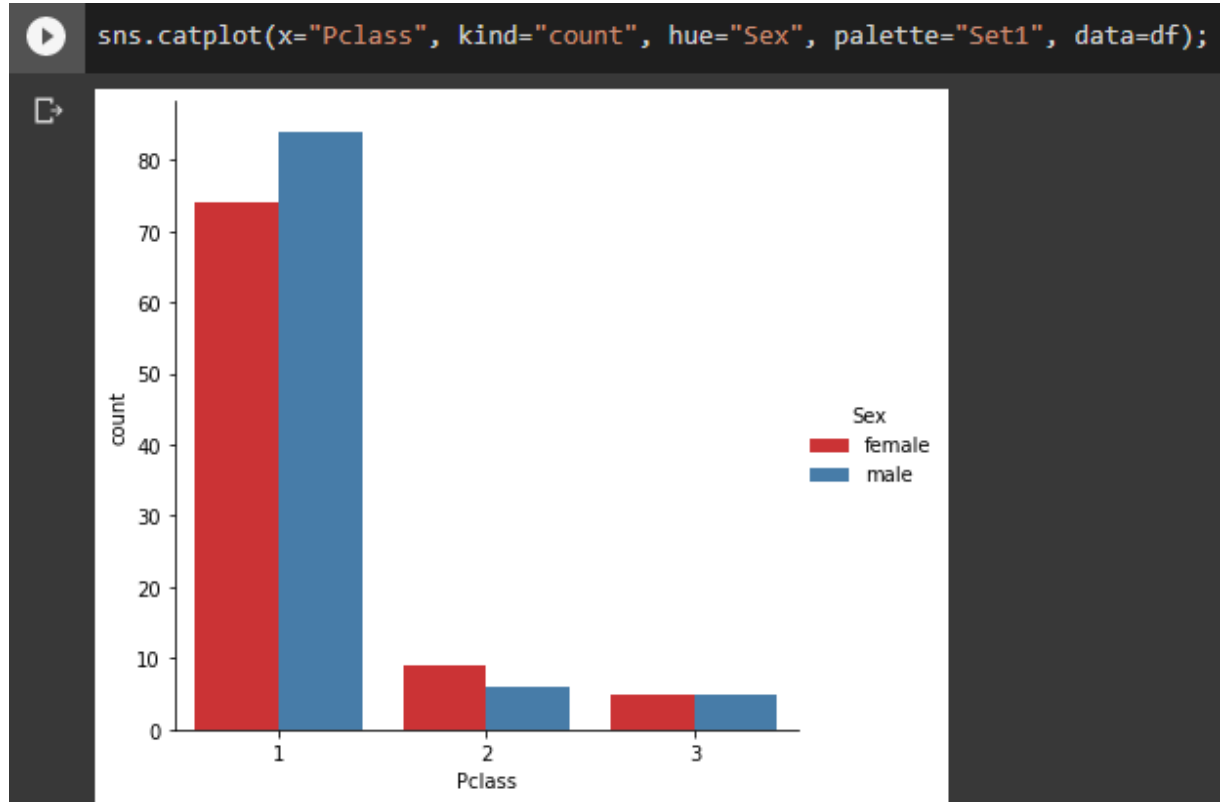
Stn grafiđi countplot: Kategorik deđiřkenleri grselleřtirme de kullanılıyor.



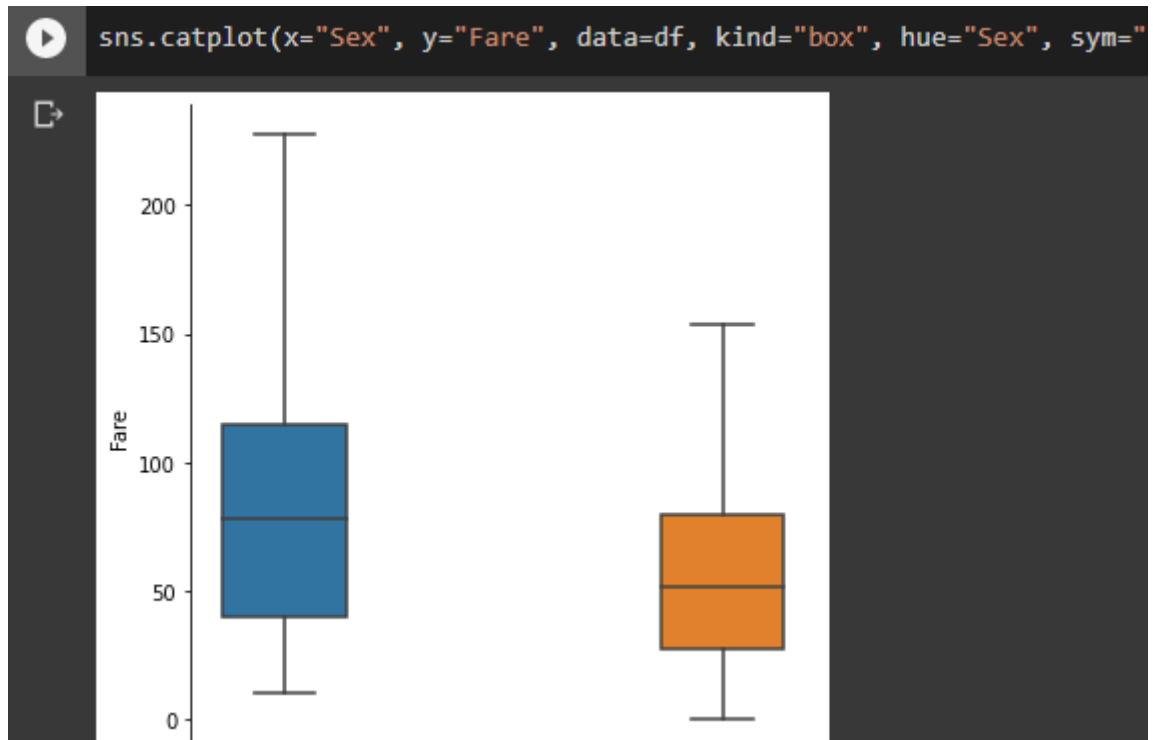
Bu dataset için erkeklerin kadınlardan daha fazla olduğunu analiz etmiş oluyoruz.

Catplot: Kategorik veriler için kullanılır.

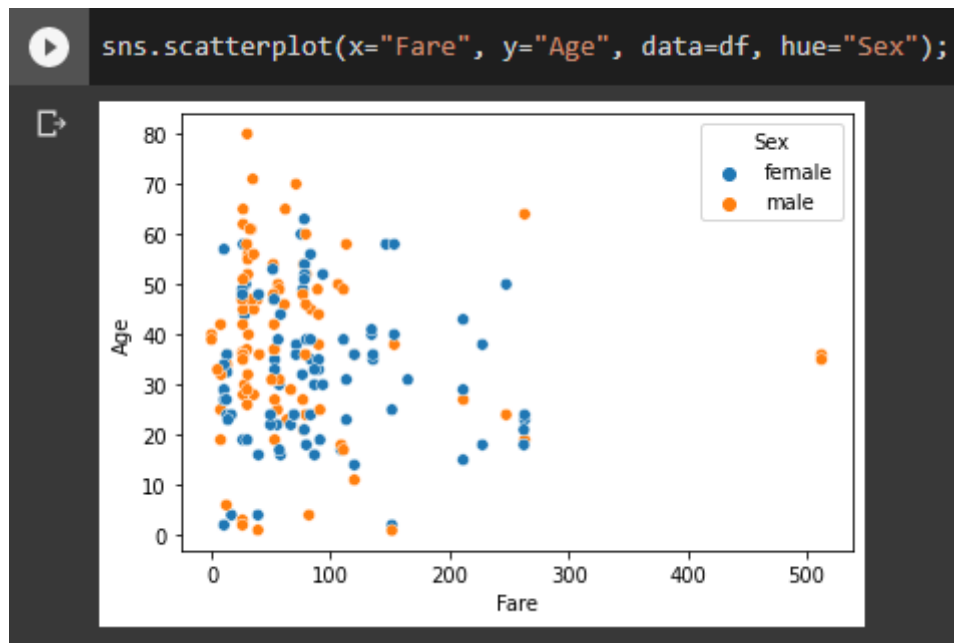
Catplot grafiğı sayesinde kind parametresinde istediğınız count, bar violin, boxplot gibi grafik türlerini elde edebiliriz.



Boxplot grafiğı: Outlierları belirlemek için medyan değęerlerine bakıyoruz, quartilere bakıyoruz ve en son minimum ve maksimum eşit değęerlerini belirleyip onların dıřında kalanların outlier olduėunu fark edebilmemizi saėlayan bir plot. Kısaca outlier değęerlerine bakarak yorum yapmamızı saėlar.



Scatterplot grafiği iki farklı değişkenin arasındaki ilişkiyi belirlemek için kullanılır.



# KAYNAKÇA

Bilgeiř “Herkes iin Yapay Zekâ I” eđitimi.

**KODLUYORUZ**  
geleceđi kodluyoruz >\_

 **EMpower**  
Enriching young lives in emerging markets