İçindekiler

In [1]:

```
import numpy as np
import pandas as pd
```

```
import seaborn as sns
import scipy.stats as st
```

Veri Bilimi için İstatistik 101

Temel İstatistik

Veri Bilimi için İstatistiğin Önemi

- "İstatistiğe dayalı olmadan, bir veri bilimci laboratuvar asistanıdır." -Martyn Janes, Direktör @ Cambriano Energy
- İstatistik veri biliminin "bilim" makine öğrenmesinin "öğrenme" kısmıdır.

Örnek Teorisi

Örneklemin Önemi

- "Daha iyisi, daha fazlası değildir." -Greg Dixon, PhD

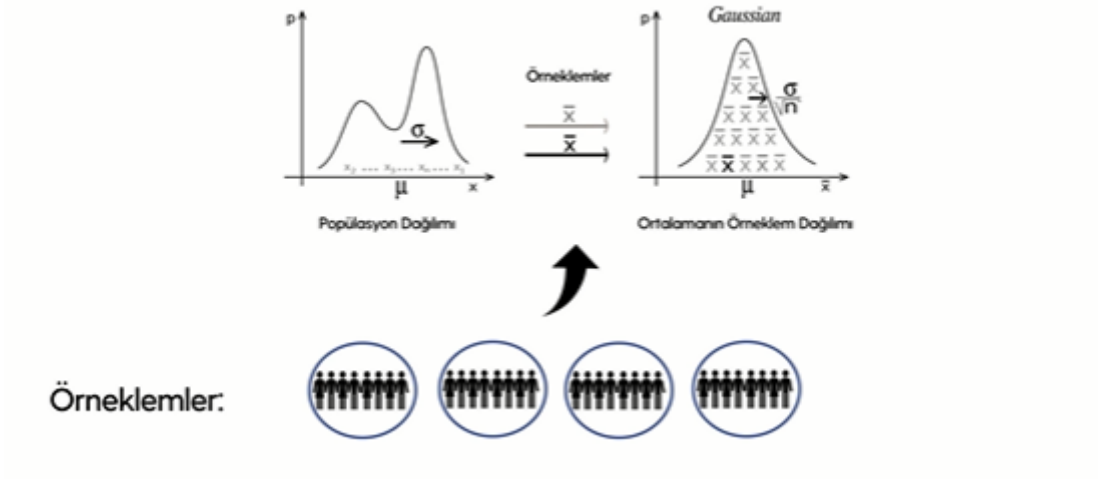
Örneklem Nedir?

- Popülasyonu temsil eden alt kümeye **örneklem** denir.
- Yansız olması gerekir.
- Olasılıksal ve olasılıksal olmayan şekilde ikiye ayrılır.

Bu eğitimde olasılıklı yani rastgele örneklemden bahsedeceğiz.

Örneklem Dağılımı

- Birden fazla örneklem çektiğimizde ve bunların dağılımı ile ilgilendiğimizde örneklem dağılımından bahsediyor oluruz.
- Yani 1 örneklem değil, birden fazla örneklem çekip onların dağılımıyla ilgileniyoruz.



Merkezi Limit Teoremi

- Bağımsız ve aynı dağılıma sahip rassal değişkenlerin toplamı ya da aritmetik ortalaması yaklaşık olarak normal dağılmaktadır.

Örnek Teorisi Uygulama

Senaryo: Bir ilçenin yaş ortalamasına ilişkin çıkarımda bulunmak istiyoruz. İlçede 10.000 kişi olduğunu varsayarsak tek tek gidip yaşlarını sormak mümkün değildir. O yüzden o kişileri temsil eden 100 kişiye gitmek(tahmini bir sayıdır) daha doğru olacaktır.

In [12]:
```
import numpy as np
```

In [13]:
```
populasyon = np.random.randint(0, 80, 10000)
```

Bunun popülasyon olduğu varsayalım(genel olarak bilinmez). 0-80 yaş arasında 10000 kişilik rastgele sayılar oluşturduk.

In [4]:
```
populasyon[0:10]
```

Out[4]:
```
array([ 1, 11,  1, 21, 76,  9, 59, 16, 66, 63])
```

Örneklem Çekimi

- `np.random.seed()` komutu yapılacak olan işlemlerin, her tekrar edildiğinde aynı sonuçları getirmesini garanti altına alan bir komuttur. İçerisinde herhangi bir sayı yazılabilir.
- Şimdi yapacağımız örnek çekme işlemini gerçekleştirdiğimizde bir yapıya/yönteme göre örnek çekecek, daha sonra bir daha çektiğimizde yine aynı yapıya göre çekecektir. Eğer bunu yazmazsak her seferinde farklı yapıya göre örnek çekecektir.

In [5]:
```
```

```
np.random.seed(10)
orneklem = np.random.choice(a = populasyon, size = 100) #Rastgele seçme komutu
orneklem[0:10]
```

Out[5]:
```
array([47, 70,  1, 38, 57, 14, 69, 26, 22, 26])
```

In [6]:
```
orneklem.mean()
```

Out[6]:
```
39.71
```

In [7]:
```
populasyon.mean()
```

Out[7]:
```
39.4567
```

Görüldüğü üzere çok yakın değerler elde ettik. İleride örneğin %95 güvenilirlik ile popülasyon ortalama şu değerle şu değer arasındadır diyeceğiz.

Örneklem Dağılımı

- Birden fazla örneklem çekerek dağılımların incelenmesi işlemine denir.
- Bu işlemin yapılmasının sebebi **popülasyona daha fazla yaklaşmaktır.**

In [10]:
```
np.random.seed(10)
orneklem1 = np.random.choice(a = populasyon, size = 100)
orneklem2 = np.random.choice(a = populasyon, size = 100)
orneklem3 = np.random.choice(a = populasyon, size = 100)
orneklem4 = np.random.choice(a = populasyon, size = 100)
orneklem5 = np.random.choice(a = populasyon, size = 100)
orneklem6 = np.random.choice(a = populasyon, size = 100)
orneklem7 = np.random.choice(a = populasyon, size = 100)
orneklem8 = np.random.choice(a = populasyon, size = 100)
orneklem9 = np.random.choice(a = populasyon, size = 100)
orneklem10 = np.random.choice(a = populasyon, size = 100)
```

In [11]:
```
(orneklem1.mean() + orneklem2.mean() + orneklem3.mean() +orneklem4.mean() + orneklem5.mean() +
orneklem6.mean() + orneklem7.mean() + orneklem8.mean() + orneklem9.mean() + orneklem10.mean()) / 10
```

Out[11]:
```
39.319
```

- Görüldüğü üzere 10 tane örneklemin ortalamaları da popülasyona yakın gelmiştir.
- Her bir örneklemin ortalaması birbirinden farklıdır fakat güzel çekildiğinde tek başına popülasyona yaklaşmış olurlar.

- Bunların da ortalamasını alındığında popülasyon ortalamasına daha çok yaklaşacaktır. Buna teorik olarak **merkezi limit teoremi** denir.

Betimsel İstatistikler

- Ortalama
- Medyan
- Mod
- Kartiller
- Değişim Aralığı
- Standart Sapma
- Kovaryans
- Korelasyon

İki değişken arasındaki ilişkinin değişkenlik ölçüsüdür.

$$\mathrm{cov}(X,Y)=\mathrm{E}[(X-\mathrm{E}[X])(Y-\mathrm{E}[Y])]$$

Kovaryans

Daha teknik bir ifadeyle, iki rastgele değişkenin kendi ortalamalarından olan sapmaların çarpımlarının beklenen değeridir. Böylece iki değişkenin birlikte ortaya çıkardığı değişim incelenmiş olur.

İki değişken arasındaki ilişkiyi, ilişkinin anlamlı olup olmadığını,
ilişkinin şiddetini ve yönünü ifade eden istatistiksel bir tekniktir.

$$r_{xy} = \frac{\sum x_i y_i - n\bar{x}\,\bar{y}}{\sqrt{(\sum x_i^2 - n\bar{x}^2)}\sqrt{(\sum y_i^2 - n\bar{y}^2)}}$$

Korelasyon

Betimsel İstatistikler Uygulama

In [15]:
```
import seaborn as sns
```

```
tips = sns.load_dataset("tips")
df = tips.copy()
df.head()
```

Out[15]:

	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

In [19]:

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

Out[19]:

	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

In [21]:

```
!pip install researchpy
import researchpy as rp
```

```
Collecting researchpy
  Downloading researchpy-0.3.2-py3-none-any.whl (15 kB)
Requirement already satisfied: pandas in c:\users\ertug\anaconda3\envs\tf\lib\site-packages (from researchpy) (1.1.5)
Requirement already satisfied: numpy in c:\users\ertug\anaconda3\envs\tf\lib\site-packages (from researchpy) (1.19.2)
Requirement already satisfied: scipy in c:\users\ertug\anaconda3\envs\tf\lib\site-packages (from researchpy) (1.5.2)
Requirement already satisfied: pytz>=2017.2 in c:\users\ertug\anaconda3\envs\tf\lib\site-packages (from pandas->researchpy) (2020.4)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\ertug\anaconda3\envs\tf\lib\site-packages (from pandas->researchpy) (2.8.1)
Requirement already satisfied: six>=1.5 in c:\users\ertug\anaconda3\envs\tf\lib\site-packages (from python-dateutil>=2.7.3->pandas->researchp
y) (1.15.0)
Collecting patsy
  Downloading patsy-0.5.1-py2.py3-none-any.whl (231 kB)
Collecting statsmodels
  Downloading statsmodels-0.12.2-cp38-none-win_amd64.whl (9.4 MB)
Installing collected packages: patsy, statsmodels, researchpy
Successfully installed patsy-0.5.1 researchpy-0.3.2 statsmodels-0.12.2
```

In [23]:

```
rp.summary_cont(df[["total_bill","tip","size"]])
```

Out[23]:

	Variable	N	Mean	SD	SE	95% Conf.	Interval
0	total_bill	244.0	19.7859	8.9024	0.5699	18.6633	20.9086
1	tip	244.0	2.9983	1.3836	0.0886	2.8238	3.1728
2	size	244.0	2.5697	0.9511	0.0609	2.4497	2.6896

- `summary.cont()` fonksiyonu **sayısal değişkenlerin** betimsel istatistiklerini(ort,ss vs.) görmek için kullanılır.
- Otomatize haliyle daha önce ele alındı.
- Daha anlamlı, okunabilir ve özet sonuçlar için bu kütüphane ve komut kullanılabilir.

In [25]:
```
rp.summary_cat(df[["sex","smoker","day"]])
```

Out[25]:

	Variable	Outcome	Count	Percent
0	sex	Male	157	64.34
1		Female	87	35.66
2	smoker	No	151	61.89
3		Yes	93	38.11
4	day	Sat	87	35.66
5		Sun	76	31.15
6		Thur	62	25.41
7		Fri	19	7.79

`summary.cat()` fonksiyonu ise **kategorik değişkenlerin** betimsel istatistiklerini(frekans vs.) görmek için için kullanılır.

In [26]:
```
df[["tip","total_bill"]].cov()
```

Out[26]:

	tip	total_bill
tip	1.914455	8.323502
total_bill	8.323502	79.252939

- Kovaryans hesaplayan komuttur.

- İki değişken arasındaki ilişkinin değişiminin ölçüsünü ele alan bir istatistiktir.

In [27]:
```
df[["tip","total_bill"]].corr()
```

Out[27]:

	tip	total_bill
tip	1.000000	0.675734
total_bill	0.675734	1.000000

- Korelasyon hesaplayan komuttur.
- İki değişken arasındaki ilişkinin anlamlı olup olmadığını, yönünü ve şiddetini belirten bir istatistiktir.

Görüldüğü üzere "tip" ile "total_bill" arasında pozitik yönlü orta şiddette bir ilişki vardır.

Güven Aralıkları

- Güven aralığı, anakütle parametresinin tahmini değerini kapsayabilecek iki sayıdan oluşan bir aralık bulunmasıdır. Burada anakütle parametresinin yerine örnek istatistiğinin iki sayı tarafından bir aralıkça ifade edilmesidir.
- Başka bir tanımı da ölçümün hassasiyetinin bir göstergesidir. Ayrıca yapmış olduğumuz tahminlerin ne kadar güvenilir olduğu ile ilgili bize bir değer sunar.

Örnek Soru:

- Ortalama ve standart sapmaya göre web sitesinde geçirilen süre **172-188** saniyeleri arasındadır.
- Başka bir yorum ise %95 güvenle yani **100 kişiden 95'i (bazen 90 bazen 99)** ortalama bu aralıkta web sitesinde zaman geçirecektir. Sadece 5 kişi bu aralığın dışında vakit geçirecektir.

Örnek Ortalaması için Güven Aralığının Hesaplanması

Adım 1: n, ortalama ve standart sapmayı bul

n = 100, ortalama = 180, standart sapma = 40

Adım 2: Güven aralığına karar ver: 95 mi 99 mu?

Z tablo değerini hesapla (1,96 – 2,57)

Adım 3: Yukarıdaki değerleri kullanarak güven aralığını hesapla:

$$\bar{x} \pm z \frac{s}{\sqrt{n}} = 180 \pm 1{,}96 \times \frac{40}{\sqrt{100}}$$

Sonuç: 180 ± 7,84 yani 172 ile 188 arasıdır.

İş Uygulaması: Fiyat Stratejisi Karar Destek

- Problem:

  CEO fiyat belirleme konusunda *bilimsel bir dayanak* ve *esneklik* isteniyor

- Detaylar:
  - Satıcı, alıcı ve bir ürün var.
  - Alıcılara ürüne ne kadar ücret öderdiniz diye soruluyor
  - Optimum fiyat bilimsel ve esnek olarak bulunmak isteniyor.

Not: Bu gerçek hayattan bir projedir.

In [1]:
```
fiyatlar = np.random.randint(10, 110, 1000)
```

In [2]:
```
fiyatlar.mean()
```

Out[2]:
```
57.404
```

In [3]:
```
import statsmodels.stats.api as sms
```

In [4]:
```
sms.DescrStatsW(fiyatlar).tconfint_mean()
```

Out[4]:
```
(55.59953088140909, 59.20846911859092)
```

Bu fonksiyon güven aralığı hesaplamak için kullanılır. (Varsayılan olarak %95'tir.)

Sonuca baktığımızda %95 güvenilirlik ile fiyat bu aralıkta olmalıdır.
Yalnızca %5'lik kısım bu fiyat aralığının dışında bir fiyat verebilirler.

Olasılığa Giriş

Olasılık Nedir?

- Veri biliminde belirsizlikle ilgili yorumlar yapabilmek için en sık başvurulan tekniklerden birisidir.
- Tanım olarak "olayların olabilirliğinin sayısal ifadesidir." -Özlem Özkılıç 2007
- Teknik tanımı ise bir olayın meydana gelmesi için uygun durum sayısının o konuda oluşabilecek tüm elverişli durum sayısına oranıdır.

Rassal Değişkenler ve Olasılık Dağılımları

- Değerlerini bir deneyin sonuçlarından alan değişkene rassal değişken denir.
- Bazı olasılık dağılımlarını kullanarak bazı olaylara ilişkin olasılıkları hesaplamak istediğimizde rassal değişkenleri kullanacağız.

Olasılık Dağılımları

Dağılım Nedir?

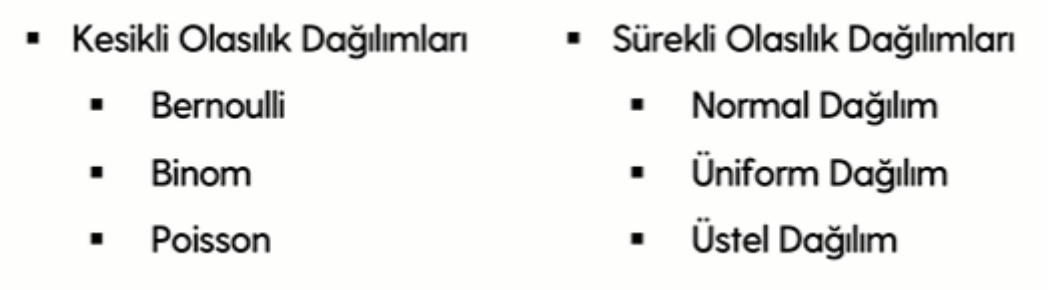
- Evrende gerçekleşen bazı olaylar ya da durumların sayısal karşılıklarının ortaya çıkardığı yapıya **dağılım** denir.

Olasılık Dağılımı Nedir?

- Bir rassal olaya ait değerler ve bu değerlerin gerçekleşme olasılıklarının bir arada ifade edilmesine **olasılık dağılımı** denir.

Olasılık Fonksiyonu Nedir?

- Bir değişkenin herhangi bir değeri alması olasılığını hesaplamaya yarayan fonksiyondur.



Kesikli ve Sürekli Olasılık Dağılımları

- Çok daha fazla vardır fakat bu eğitim kapsamında bu dağılımlar ele alınacaktır.

Bernoulli Dağılımı

Hatırlatma:

- Olasılık dağılımları, amacı itibariyle ve bize sunacakları faydalar açısından anlaşılması zor olabilen konulardandır.
- Temel amacımız, belirsizlik altında karar vermektir. Buna yönelik olarak bazı tekniklerle belirsizliği azaltmaya çalışıyoruz. Bu belirsizliği azaltmaya çalıştığımız yaklaşımlardan birisi de olasılık dağılımlarıdır.
- Olasılık dağılımları bize bazı değerlerin ne şekilde hangi olasılıklarla gerçekleşebileceğine yönelik fikir verir.

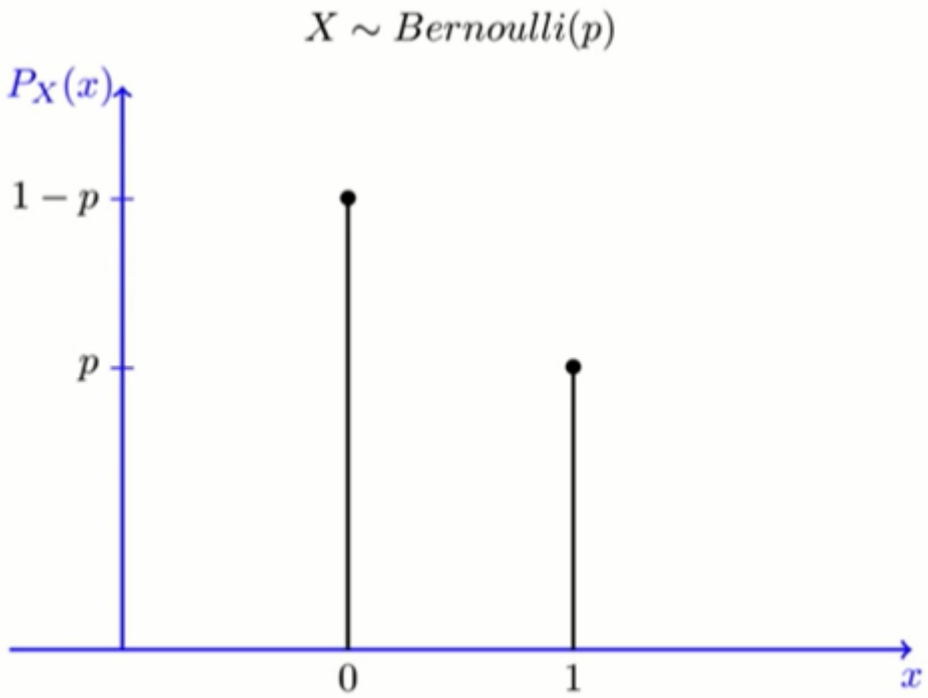
Bernoulli Dağılımı Nedir?

- Başarılı-başarısız, olumlu-olumsuz şeklindeki iki sonuçlu olaylar ile ilgilenildiğinde kullanılan kesikli olasılık dağılımıdır.

$$f(x;p) = p^x (1-p)^{1-x}, \qquad x \in \{0,1\}$$

$$E(X) = p \qquad Var(X) = pq = p(1-p)$$

- Burada p, olasılığı ifade eder. (yazı-tura deneyi için 0.50'i ifade eder.)
- x ise kesikli değişkenimizin alacağı değerlerdir. (yazı-tura deneyi için yazı: 1, tura: 0)
- Yani x burada rassal değişkenin alacağı değer demek; p ise ilgilendiğimiz bir olay için bir sınıfın gerçekleşme olasılığı demektir.
- Burada E(X) ifadesi beklenen değer demektir ve ortalamayı temsil eder.



Bernoulli Dağılımı Uygulama

${\displaystyle f(x;p)=p^{x}(1-p)^{1-x}\quad {\text{for }}x\in \{0,1\}}$

${\displaystyle \operatorname {E} \left(X\right)=p}$

$\operatorname {Var} (X)=pq=p(1-p)$

Yazı tura deneyi yapıyor olalım.

In [2]:
```
from scipy.stats import bernoulli
```

In [3]:
```
p = 0.6
```

p burada iki sonuçlu bir olayı ifade eder.
Başarı olarak tura ile ilgilenelim ve tura gelme olasılığının 0.6 olduğunu bildiğimizi varsayalım.

In [4]:
```
rv = bernoulli(p)
```

```
rv.pmf(k = 1)
```

Out[4]: 0.6

- `pmf` fonksiyonu **probability mass function**(olasılık kütle fonksiyonunun kısaltmasıdır.)
- **Kesikli dağımlar için bu fonksiyonu kullanılır.**
- **Sürekli dağılımlar için bir alan hesabı bulacağımızdan `cdf()` (cumulative density function: kümülatif yoğunluk fonksiyonu) fonksiyonunu kullanacağız.**
- Burada k=1 diyerek(yani rassal değişkenin alabileceği değer -> x=1 diyerek) başarı yani tura gelmesi olasılığını bulduk. Yani ilgilenmiş olduğumuz burada tura olması olasılığıdır -ki burada 0.6 olarak verdik-. k değerini girdiğimizde bize sonucu verecektir. (fonksiyona x diyemiyoruz çünkü hata veriyor)

In [5]:
```
rv = bernoulli(p)
rv.pmf(k = 0)
```

Out[5]: 0.4

Burada da 0 diyerek başarısızlık yani yazı gelmesi olasılığını bulmak istedik.

Büyük Sayılar Yasası

Bir rassal değişkenin uzun vadeli kararlılığını tanımlayan olasılık teoremidir.

Yazı-tura deneyini ele alalım ve 5T 1Y geldiğini varsayalım. Burada yazı gelme olasılığı %20'dir. Fakat bu deneyi çokça tekrar ettiğimizde bu olasılık %50'ye yaklaşacaktık. İşte bu uzun vadeli kararlılığa **büyük sayılar yasası** denir.
Örnek kod ile bunu gözlemleyelim:

In [4]:
```
rng = np.random.RandomState(123)
for i in np.arange(1,21):
    deney_sayisi = 2**i
    yazi_turalar = rng.randint(0, 2, size=deney_sayisi)
    yazi_olasiliklari = np.mean(yazi_turalar)
    print("Atış Sayısı:", deney_sayisi,"---",'Yazı Olasılığı: %.2f' % (yazi_olasiliklari*100))
```

```
Atış Sayısı: 2 --- Yazı Olasılığı: 50.00
Atış Sayısı: 4 --- Yazı Olasılığı: 0.00
Atış Sayısı: 8 --- Yazı Olasılığı: 62.50
Atış Sayısı: 16 --- Yazı Olasılığı: 43.75
Atış Sayısı: 32 --- Yazı Olasılığı: 46.88
Atış Sayısı: 64 --- Yazı Olasılığı: 56.25
Atış Sayısı: 128 --- Yazı Olasılığı: 50.78
Atış Sayısı: 256 --- Yazı Olasılığı: 52.73
Atış Sayısı: 512 --- Yazı Olasılığı: 52.93
Atış Sayısı: 1024 --- Yazı Olasılığı: 50.20
```

```
Atış Sayısı: 2048 --- Yazı Olasılığı: 48.58
Atış Sayısı: 4096 --- Yazı Olasılığı: 49.49
Atış Sayısı: 8192 --- Yazı Olasılığı: 49.58
Atış Sayısı: 16384 --- Yazı Olasılığı: 49.96
Atış Sayısı: 32768 --- Yazı Olasılığı: 50.00
Atış Sayısı: 65536 --- Yazı Olasılığı: 49.68
Atış Sayısı: 131072 --- Yazı Olasılığı: 49.97
Atış Sayısı: 262144 --- Yazı Olasılığı: 50.13
Atış Sayısı: 524288 --- Yazı Olasılığı: 50.01
Atış Sayısı: 1048576 --- Yazı Olasılığı: 50.09
```

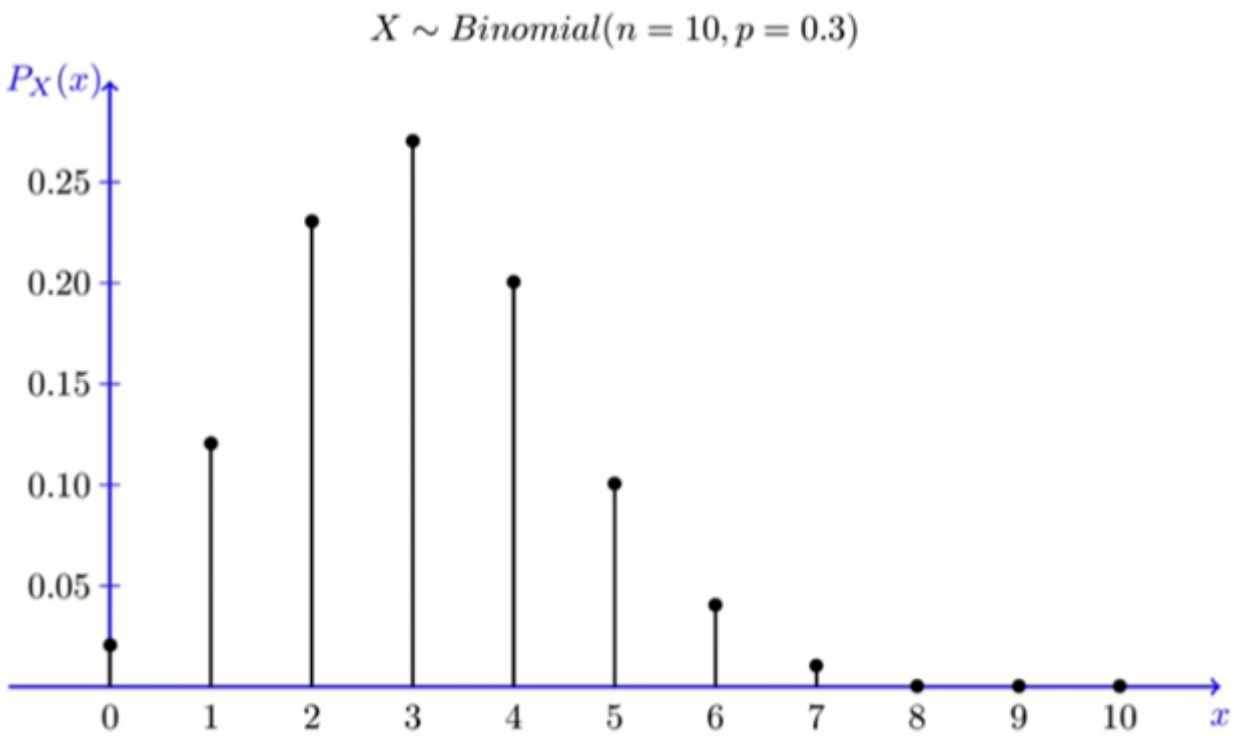
Görüldüğü üzere deney sayısı arttıkça olasılığın %50'ye yaklaştığını görüyoruz.

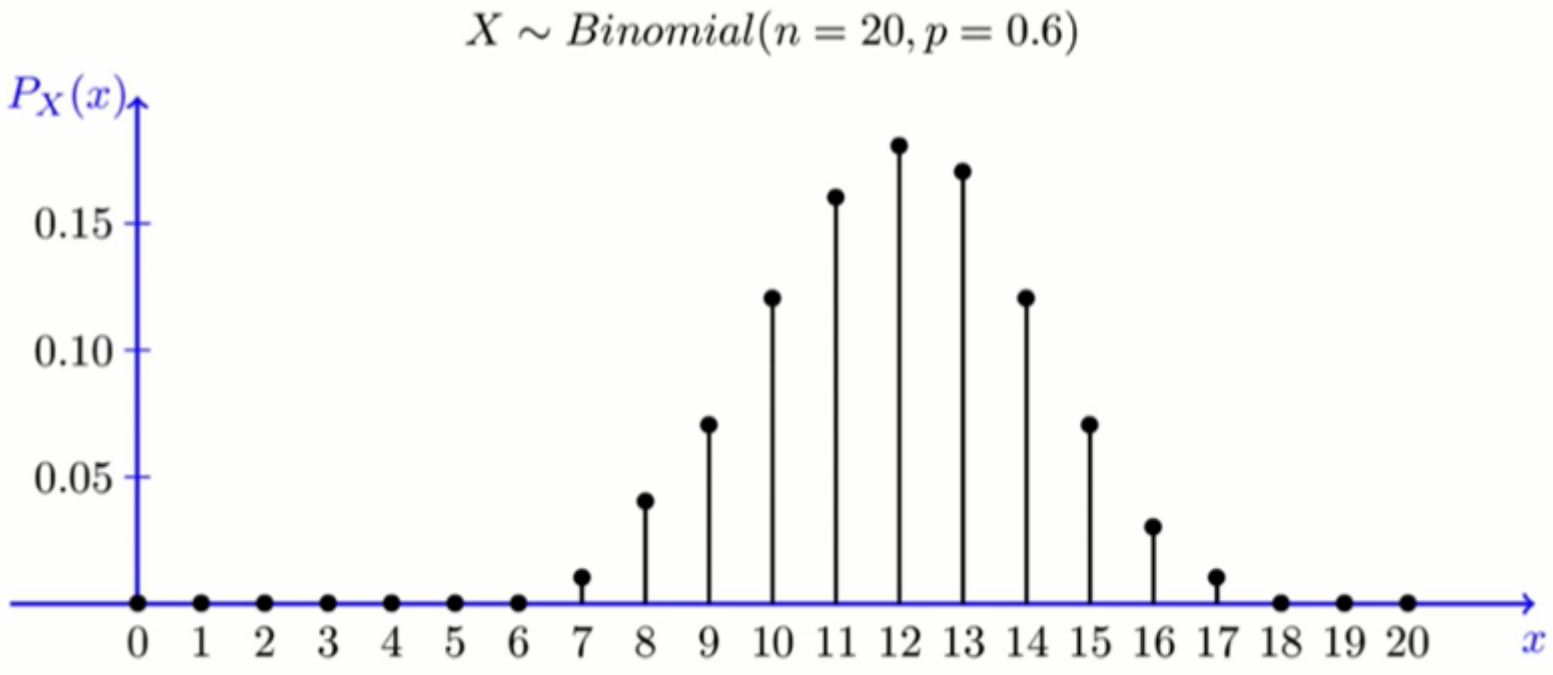
Binom Dağılımı

- Binom dağılımı, bağımsız n deneme sonucu k başarılı olma olasılığı ile ilgilenildiğinde kullanılan dağılımdır.
- Bernoulli dağılımının n defa gerçekleştirilmiş halidir.
- **Deneyler bağımsız ve aynı koşullar altında olmalıdır.**

$$f(x;n,p) = \binom{n}{x} p^x (1-p)^{n-x}, \qquad x = 0,1,2,\dots,n$$

$$E(X) = np \qquad Var(X) = np(1-p)$$


$X \sim Binomial(n = 10, p = 0.3)$
$P_X(x)$
0.25
0.20
0.15
0.10
0.05
0 1 2 3 4 5 6 7 8 9 10
x


$X \sim Binomial(n = 20, p = 0.6)$
$P_X(x)$
0.15
0.10
0.05
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
x

Bir madeni para 4 kere atılıyor. 2 kere yazı gelmesi olasılığı nedir?

$$f(x;n,p) = \binom{n}{x} p^x (1-p)^{n-x}, \qquad x = 0,1,2,\dots,n$$

$$f(2;4,0.50) = \binom{4}{2} 0.50^2 (1-0.50)^{4-2} = 0.375$$

İş Uygulaması: Reklam Harcaması Optimizasyonu

- Problem:

  Çeşitli mecralara reklam veriliyor, reklamların tıklanma ve geri dönüşüm oranları optimize edilmeye çalışılıyor. Buna yönelik olarak belirli bir mecrada çeşitli senaryolara göre reklama tıklama olasılıkları hesaplanmak isteniliyor.

- Detaylar:
  - Bir mecrada reklam verilecek
  - Dağılım ve reklama tıklama olasılığı biliniyor (0.01)
  - Soru: Reklamı 100 kişi gördüğünde 1,5,10 tıklanması olasılığı nedir?

Olasıkların Hesaplanması

$$f(1; 100, 0.01) = \binom{100}{1} 0.01^2 (1 - 0.01)^{100-1} = 0.37$$

$$f(5; 100, 0.01) = 0.00289779$$

$$f(10; 100, 0.01) = 0.00000007$$

Kod ile bunu yapalım:

In [1]:
```
from scipy.stats import binom
```

In [2]:
```
p = 0.01
n = 100
rv = binom(n, p)
print(rv.pmf(1))
print(rv.pmf(5))
print(rv.pmf(10))
```

```
0.36972963764971983
0.0028977871237616114
7.006035693977161e-08
```

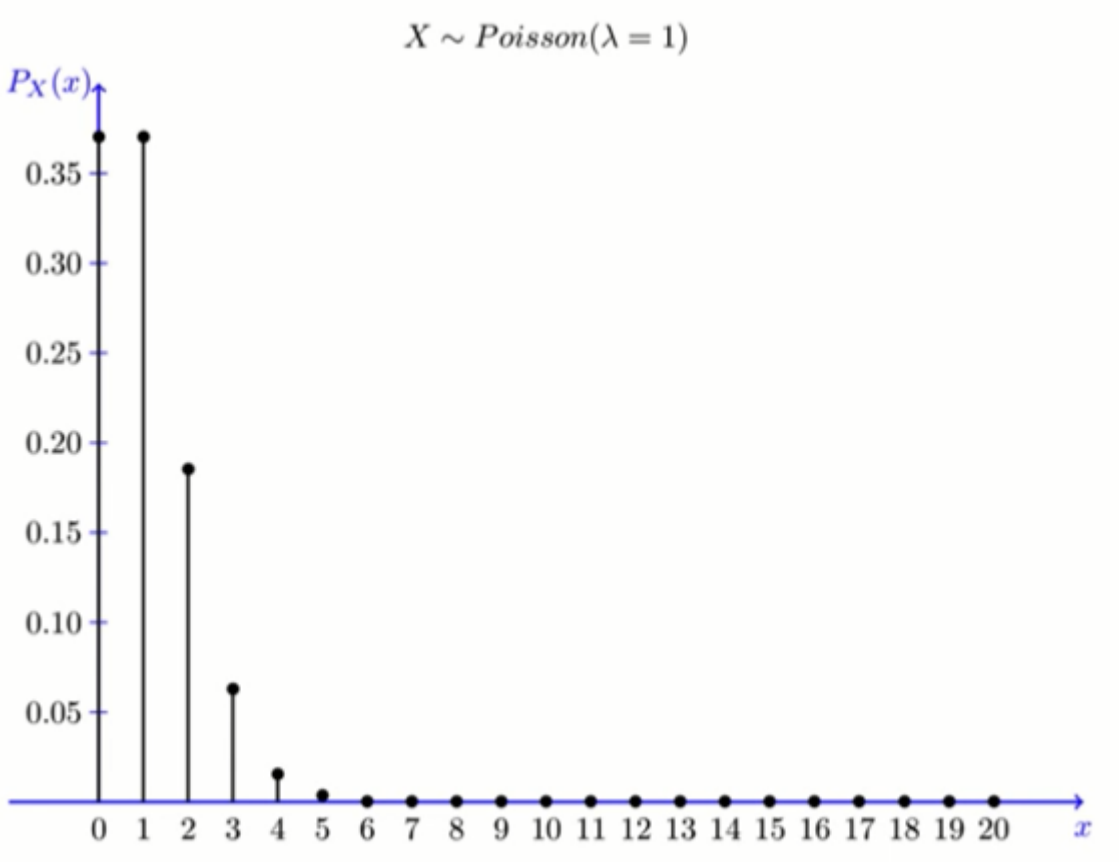
Poisson Dağılımı

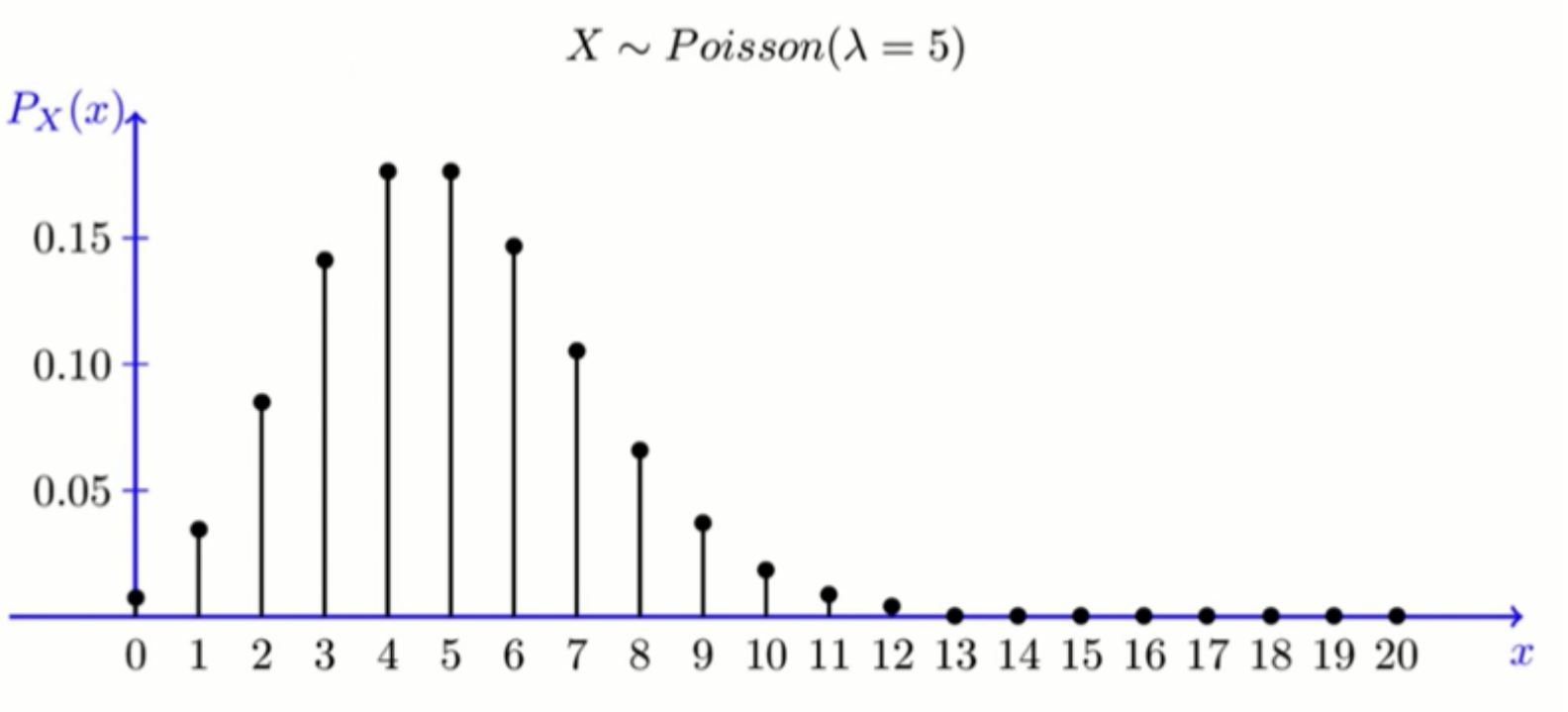
- Belirli bir zaman aralığında belirli bir alanda **nadiren** rastlanan olayların olasılıklarını hesaplamak için kullanılır.
- Yani gözlem sayısının çok yüksek ve beklenen sonucun gelme olasılığının çok küçük olduğu zamanlarda kullanılır. (n: büyük, p: küçük)
- Binom dağılımının özel bir halidir.
- Buradaki nadirlik ölçüsü, yani bir olayın nadir olay olarak kabul edilebilmesi için genel görüş **n > 50 ve n * p < 5** olmalıdır.
- Bu dağılım için:
  - Rassal denemeler iki sonuçlu olmalıdır.
  - Aynı koşullar altında gerçekleştirilmelidir.
  - Rassal denemeler birbirinden bağımsız olmalıdır.

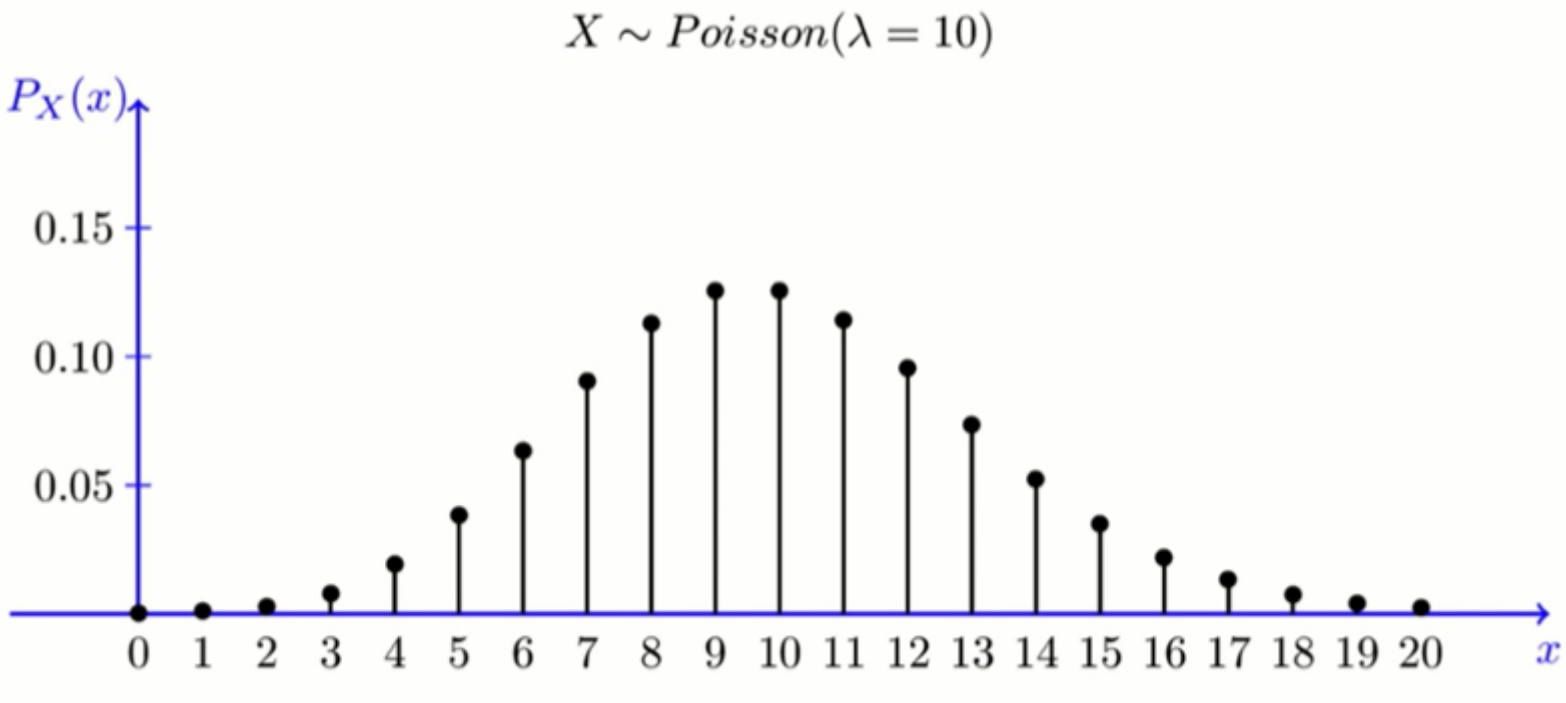
$$f(x,\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \qquad x = 0,1,2,\dots,n$$

$$E(X) = \lambda \quad Var(X) = \lambda$$

- Burada lambda, beklenen sonucun ortalama gerçekleşme sayısıdır.


$X \sim Poisson(\lambda = 1)$
$P_X(x)$
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
x


$X \sim Poisson(\lambda = 5)$
$P_X(x)$
0.15
0.10
0.05
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
x


$X \sim Poisson(\lambda = 10)$
$P_X(x)$
0.15
0.10
0.05
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
x

- 10 BİN kelimeden oluşan bir kitapta hatalı kelime sayısı
- 4000 öğrencili okulda not girişinde hata yapılması
- Bir iş gününde çağrı merkezine gelen taktir sayısı
- Kredi kartı işlemlerinde sahtekarlık olması
- Rötara düşen uçuş sefer sayısı

Örnek:

Bir üniversitede 5000 not girişinde 5 tane notun yanlış girilmesi olasılığı nedir?

Dağılımın Poisson olduğu biliniyor ve Lambda = 0.2

$$f(5;0.2) = \frac{0.2^5\, e^{-0.2}}{5!} = 0.00000218328201$$

İş Uygulaması: İlan Girişi Hata Olasılıkları

- Problem:

  Hatalı ilan girişi olasılıkları hesaplanmak isteniyor.

- Detaylar:
  - Bir yıl süresince ölçümler yapılıyor
  - Dağılım biliniyor (Poisson) ve Lambda 0.1 (ortalama hata sayısı)
  - Hiç hata olmaması, 3 hata olması ve 5 hata olması olasılıkları nedir?

$$f(0;0.1) = \frac{0.1^0\, e^{-0.1}}{0!} = 0.9048374180$$

$$f(3;0.1) = \frac{0.1^3\, e^{-0.1}}{3!} = 0.0001508062$$

$$f(5;0.1) = \frac{0.1^5\, e^{-0.1}}{5!} = 0.0000000754$$

Kod ile bunu yapalım:

In [3]:
```
from scipy.stats import poisson
```

In [4]:
```
lambda_ = 0.1
```

In [5]:
```
rv = poisson(mu = lambda_)
print(rv.pmf(k = 0))
```

```
0.9048374180359595
```

In [6]:
```
print(rv.pmf(k = 3))
```

```
0.0001508062363633932676
```

In [7]:
```
print(rv.pmf(k = 5))
```

```
7.54031181696634e-08
```

Normal Dağılım

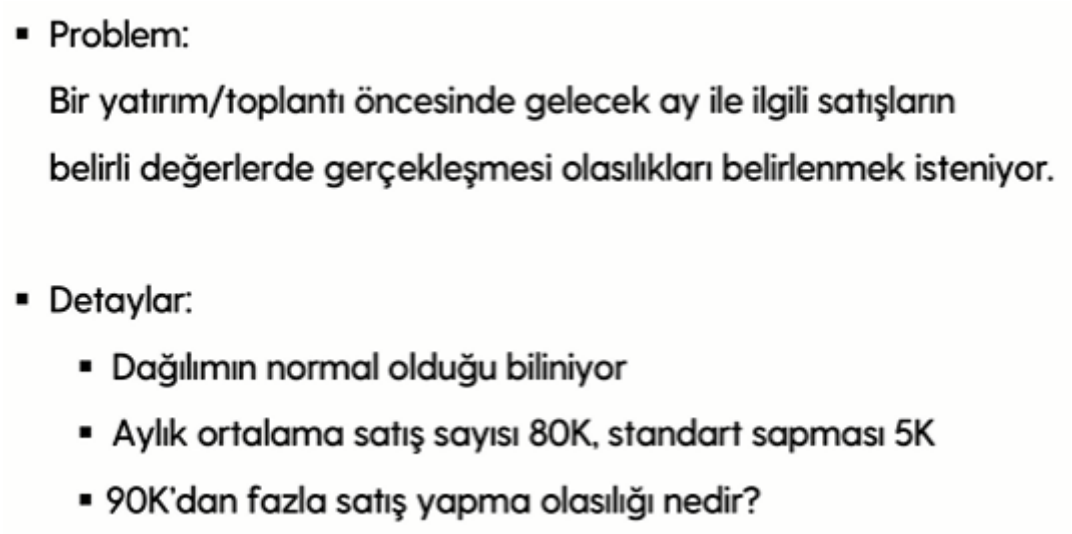
- Normal dağıldığı bilinen sürekli rassal değişkenler için olasılık hesaplaması için kullanılır.

Not: Dağılımı bilinen herhangi bir değişken için olasılık hesabı yapabiliriz.

Not: Kesikli dağılımlarda olasılık hesabı toplam sembolü ile, sürekli dağılımlarda olasılık hesabı integral ile yapılır.

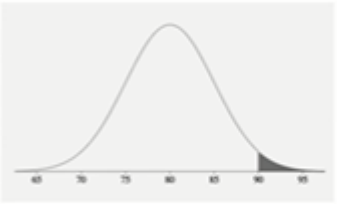
$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

İş Uygulaması: Satış Olasılıklarının Hesaplanması



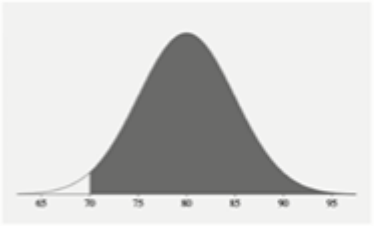
İçgüdüsel olarak düşük bir olasılık beklenir. Çünkü ortalama 80K, standart sapması ise 5K'dır. Yani satış sayısı 75K ila 85K arasında değişecektir. Bu yüzden olasılık düşük olacaktır.

90K'dan fazla olması olasılığı nedir?



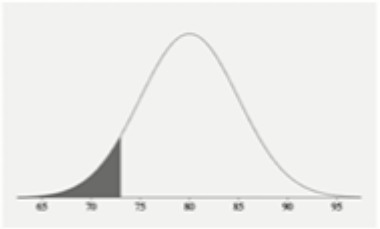
$P(X>90)=0.0228$

70K'dan fazla olması olasılığı nedir?

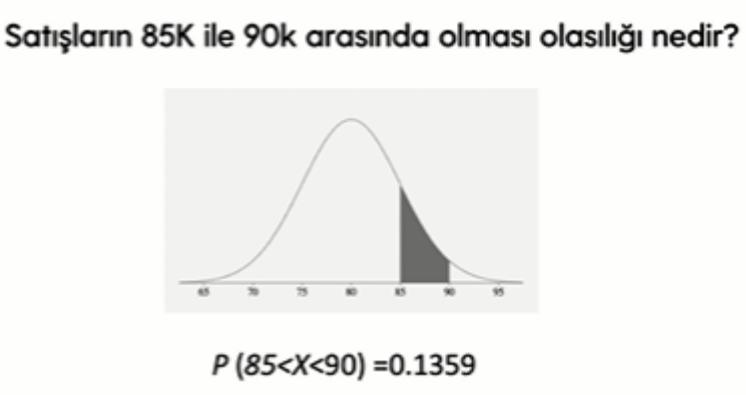


$P\ (X>70)=0.9772$

73K'dan az olması olasılığı nedir?



$P\ (X<73) = 0.0808$



Bunu normal dağılım grafiğinde incelediğimizde örneğin 90K'dan fazlası için düşük, 70K'dan fazlası için yüksek olasılık olduğu görülür. Çünkü ortalama 80K olduğundan olayın gerçekleşme olasılığı da yüksek olacaktır.

Not: Normalde bu olasılıkların hesaplanması için integralle alan hesabı bulunarak yapılır.

Şimdi kod ile bunu inceleyelim:

In [2]:
```
from scipy.stats import norm
```

90'dan Fazla Olması Olasılığı

In [3]:
```
1 - norm.cdf(90, 80, 5)
```

Out[3]:
```
0.02275013194817921
```

`cdf()` dememizin sebebi kesikli dağılımlardan farklı olarak alan hesabı yapmış olmamızdır. **Cumulative density function (kümülatif yoğunluk fonksiyonu)** demektir.

1'den çıkarmamızın sebebi, normal dağılım eğrisinin altında kalan alan 1'e eşittir. Bize ise belirli bir integral kısmı lazımdı. Bu noktada `cdf()` fonksiyonunu kullanarak 90'dan yukarı olan olasılığı hesaplamak için bu şekilde 1'den çıkarma işlemi gerçekleştirdik.

70'ten Fazla Olması Olasılığı

In [5]:
```
1 - norm.cdf(70, 80, 5)
```

Out[5]:
```
0.9772498680518208
```

73'ten Az Olması Olasılığı

In [7]:
```
norm.cdf(73, 80, 5)
```

Out[7]: 0.08075665923377107

85 ile 90 Arasında Olması Olasılığı

In [10]:
```
norm.cdf(90, 80, 5) - norm.cdf(85, 80, 5)

#Eksi değer olmaması için büyükten küçüğü çıkardık.
#Burada aslında bu aralıklar için alan hesabı yapmış oldu.
```

Out[10]: 0.13590512198327787

Veri Bilimi için İstatistik 201

Hipotez Testleri

Hipotez Testi Nedir?

- Bir inanışı (bir savı, bir tahmini vs.) test etmek için kullanılan istatistiksel bir tekniktir.
- Örneğin bir doktorun ilaç bulması halinde bunun etkili olduğunu düşünüp bilimsel olarak bunu test etmesi hipotez testi konusuna girmektedir.
- İlgili kişi hipotezin doğruluğunu öyle bir kanıtlamalı ki ortaya çıkan etkinin şans eseri oluşma ihtimali göz önünde bulundurulduğu halde kanıtlamalıdır. Yani hipotez testleri şans eseri ortaya çıkma durumunu da göz önünde bulundurarak **şansa yer vermeyecek şekilde** bize ilgilenmiş olduğumuz konuda ispat etme imkanı sağlar.

Hipotezler ve Türleri

$H_0: \mu = 50$ $H_0: \mu <= 50$ $H_0: \mu >= 50$

$H_1: \mu \neq 50$ $H_1: \mu > 50$ $H_1: \mu < 50$

- Tek yönlü ve çift yönlü olmak üzere ikiye ayrılır.
- İddia ettiğimiz hipotez H1 hipotezi olacaktır.
- Örneğin örneklem konusunda yaptığımız uygulamada 10000 kişilik bir popülasyondan 100 örneklem seçmiştik. Bunların yaşlarına ilişkin şöyle hipotezler kurulabilir:
  - Bu beldenin yaş ortalaması 50'den farklıdır.
  - Bu beldenin yaş ortalaması 50'den büyüktür.
  - Bu beldenin yaş ortalaması 50'den küçüktür.

Hata Tipleri

$H_0: \mu = 50$ $H_0: \mu <= 50$ $H_0: \mu >= 50$

$H_1: \mu \neq 50$ $H_1: \mu > 50$ $H_1: \mu < 50$

Üstteki hipotezlerden herhangi birini kurduğumuzu düşünelim. Önceki örneğimizde 10000 popülasyon vardı ve bunun yaşları ortalamasını kodsal olarak biliyoruz fakat gerçekte bunu bilemeyiz. Yani H0 doğru da olabilir yanlış da olabilir. Test yapıp belirsizliği azaltmaya çalışıyoruz.

		Hipotez Testi Sonucu Verilen Karar	
		H_0 reddedilmedi	H_0 reddedildi
Gerçek	H_0 doğru	Doğru Karar $(1-a) \rightarrow$ Güven Düzeyi	I. Tip Hata a
	H_0 yanlış	II. Tip Hata β	Doğru Karar $(1-\beta) \rightarrow$ Testin Gücü

Not: Gerçek H0'ı hiçbir zaman bilemeyeceğiz. Bu tablo teorik bir dayanaktır.

- Daha çok I. tip hata ile ilgileniyor olacağız.
- "H0 red" veya "H0 reddedilemez" ifadeleri doğrudur. "H0 kabul" ifadesi teorik olarak hiçbir zaman doğru DEĞİLDİR. Çünkü H0 gerçekte doğru iken onu reddettiğimizde yapacağımızı hatayı biliyorken(alpha) H0'ı kabul ettiğimizde yapacak olduğumuz hatayı(beta) bilmiyoruz.

P Value

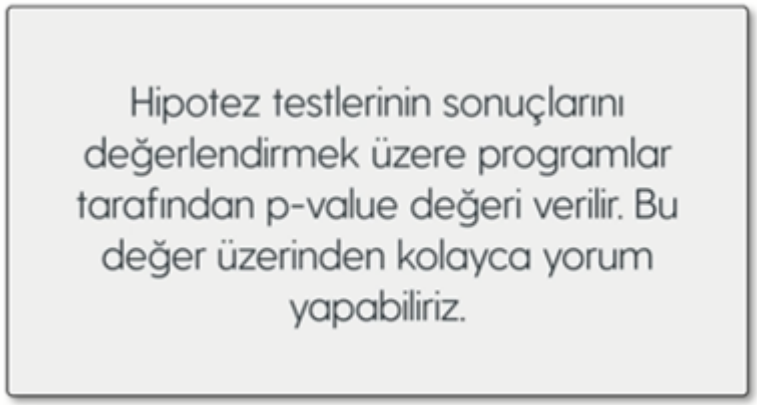
$a = 0.05$

$p<0.05$

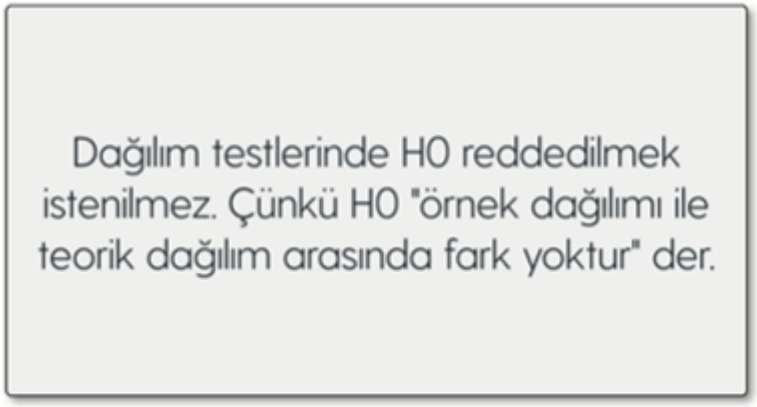
Tanımsal olarak:

- P değeri, gözlemlenen sonuçların aslında test edilmek istenen durumla hiçbir alakası olmamasının olasılığıdır. Farklı bir deyişle, modelin (sıfır hipotezi) doğru olduğu kabul edilirse, p değeri test edilen değere eşit ya da aşırı değerler elde etme olasılığıdır. -Wikipedia
- P değeri, bir karşılaştırmada "istatistiksel anlamlı fark vardır" kararı vereceğimiz zaman yapacağımız olası hata miktarını gösterir. ... Bir test sonucunda bulunan P değeri 0,05'in altında bir değer ise karşılaştırma sonucunda anlamlı farklılık bulunduğu anlamına gelir. -p005.net

Hipotez testi sonucunda bir p değeri elde edilir. Eğer **p < alpha** ise (genel olarak alpha 0.05'tir) H0 hipotezini reddedip seviniyoruz. Çünkü ortaya attığımız hipotez H1 hipotezi olduğundan bu durumda H1 hipotezi kabul olmuş oluyor.



Dağılıma uygunluk testlerinde:



Bu durumda p < alpha durumuna sevinemeyiz. Çünkü H0 red olmuş olur ve örnek dağılımımız teorik dağılıma (normal dağılım vs.) uygun olmamış olur. Fakat bu testte biz uygun olmasını yani H0'ı reddetmemek istiyoruz.

Hipotez Testi Adımları

Adım 1: Hipotezlerin kurulması ve yönlerinin belirlenmesi

Adım 2: Anlamlılık düzeyinin ve tablo değerinin belirlenmesi

Adım 3: Test istatistiğinin belirlenmesi ve test istatistiğinin hesaplanması

Adım 4: Hesaplanan test istatistiği ile alfa'ya karşılık gelen tablo

değerinin karşılaştırılması.

Test İstatistiği (Zh) > Tablo Değeri (Zt) ise H0 Red

Adım 5: Yorum

Tek Örneklem T Testi ve Tek Örneklem Oran Testi

Tek Örneklem T Testi

Tanımı: Popülasyon ortalaması ile varsayımsal bir değer arasında istatistiksel olarak anlamlı bir farklılık olup olmadığını test etmek için kullanılan parametrik bir testtir.

Hipotezler:

$H_0: \mu = 50$	$H_0: \mu <= 50$	$H_0: \mu >= 50$
$H_1: \mu \neq 50$	$H_1: \mu > 50$	$H_1: \mu < 50$

Test İstatistiği:

$$t = \frac{\overline{x} - \mu}{\frac{s}{\sqrt{n}}} \qquad z = \frac{\overline{x} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

1. Anakütle standart sapması biliniyorsa z istatistiği kullanılır.
2. Anakütle standart sapması bilinmiyorsa ve n > 30 ise z istatistiği kullanılır.
3. Anakütle standart sapması bilinmiyor ve n < 30 ise t istatistiği kullanılır.

n büyüdükçe t, z'ye yaklaşır

Not: Yapacağımız örneklerde çoğunluk t testi kullanılacaktır.

Varsayımlar:

- Örneklem normal dağılıma sahip olmalıdır.

İş Uygulaması: Ürün Satın Alma Adım Optimizasyonu

- Problem:

  Sepete ürün ekleme işlemi sonrasında ödeme ekranında 5 adım vardır ve bu adımların birisi sorgulanmaktadır.

- Detaylar:
  - Her adımın 20'şer sn. olması hedefi var. 4. adım sorgulanıyor.
  - Bu durumu test etmek için 100 örnek alınıyor.
  - Örnek standart sapması 5 saniyedir. Örnek ortalaması ise 19 saniyedir.

"Biz neyin testini yapıyoruz? Zaten örnek ortalaması 19 değil mi?" şeklinde soru aklımıza gelebilir. Burada testi yapmamızın amacı **istatistiksel olarak anlamlı bir fark olup olmadığını** bulmaktır. Yani şansa yer bırakmayacak şekilde ispat etmeye çalışmaktır.

Çözüm:

Adım 1: Hipotezlerin kurulması ve yönlerinin belirlenmesi

$$H_0: \mu = 20$$
$$H_1: \mu \neq 20$$

Adım 2: Anlamlılık düzeyinin ve tablo değerinin belirlenmesi

$$\alpha = 0{,}05 \qquad \frac{\alpha}{2} = 0{,}025$$

Ztablo tablo olasılık değeri: 0,5 – 0,025 = 0,475

Ztablo kritik değer = -/+ 1,96

Areas Under the Standard Normal Curve

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.0000	0.0040	0.0080	0.0120	0.0160	0.0199	0.0239	0.0279	0.0319	0.0359
0.1	.0398	.0438	.0478	.0517	.0557	.0596	.0636	.0675	.0714	.0753
0.2	.0793	.0832	.0871	.0910	.0948	.0987	.1026	.1064	.1103	.1141
0.3	.1179	.1217	.1255	.1293	.1331	.1368	.1406	.1443	.1480	.1517
0.4	.1554	.1591	.1628	.1664	.1700	.1736	.1772	.1808	.1844	.1879
0.5	.1915	.1950	.1985	.2019	.2054	.2088	.2123	.2157	.2190	.2224
0.6	.2257	.2291	.2324	.2357	.2389	.2422	.2454	.2486	.2517	.2549
0.7	.2580	.2611	.2642	.2673	.2704	.2734	.2764	.2794	.2823	.2852
0.8	.2881	.2910	.2939	.2967	.2995	.3023	.3051	.3078	.3106	.3133
0.9	.3159	.3186	.3212	.3238	.3264	.3289	.3315	.3340	.3365	.3389
1.0	.3413	.3438	.3461	.3485	.3508	.3531	.3554	.3577	.3599	.3621
1.1	.3643	.3665	.3686	.3708	.3729	.3749	.3770	.3790	.3810	.3830
1.2	.3849	.3869	.3888	.3907	.3925	.3944	.3962	.3980	.3997	.4015
1.3	.4032	.4049	.4066	.4082	.4099	.4115	.4131	.4147	.4162	.4177
1.4	.4192	.4207	.4222	.4236	.4251	.4265	.4279	.4292	.4306	.4319
1.5	.4332	.4345	.4357	.4370	.4382	.4394	.4406	.4418	.4429	.4441
1.6	.4452	.4463	.4474	.4484	.4495	.4505	.4515	.4525	.4535	.4545
1.7	.4554	.4564	.4573	.4582	.4591	.4599	.4608	.4616	.4625	.4633
1.8	.4641	.4649	.4656	.4664	.4671	.4678	.4686	.4693	.4699	.4706
1.9	.4713	.4719	.4726	.4732	.4738	.4744	.4750	.4756	.4761	.4767

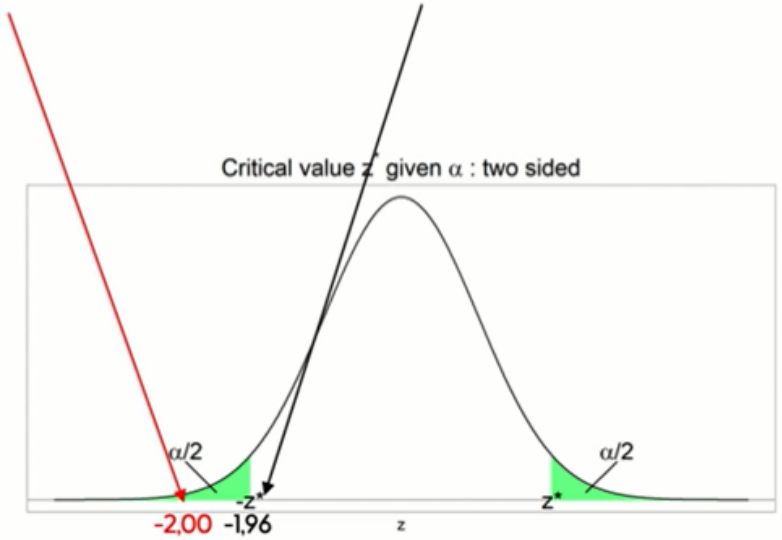
Adım 3: Test istatistiğinin belirlenmesi ve test istatistiğinin hesaplanması

$$z = \frac{\overline{x} - \mu}{\frac{\sigma(s)}{\sqrt{n}}} \qquad zhesap = \frac{19 - 20}{5/\sqrt{100}} = -2{,}00$$

n = 100, standart sapma = 5, örnek ortalaması 19 sn

Adım 4: Ztablo ve Zhesap karşılaştırması Zh > Zt ya da -Zh < -Zt ise H0 Red

Zhesap = -2,00 < Ztablo = -1,96 olduğu için H0 reddedilir.



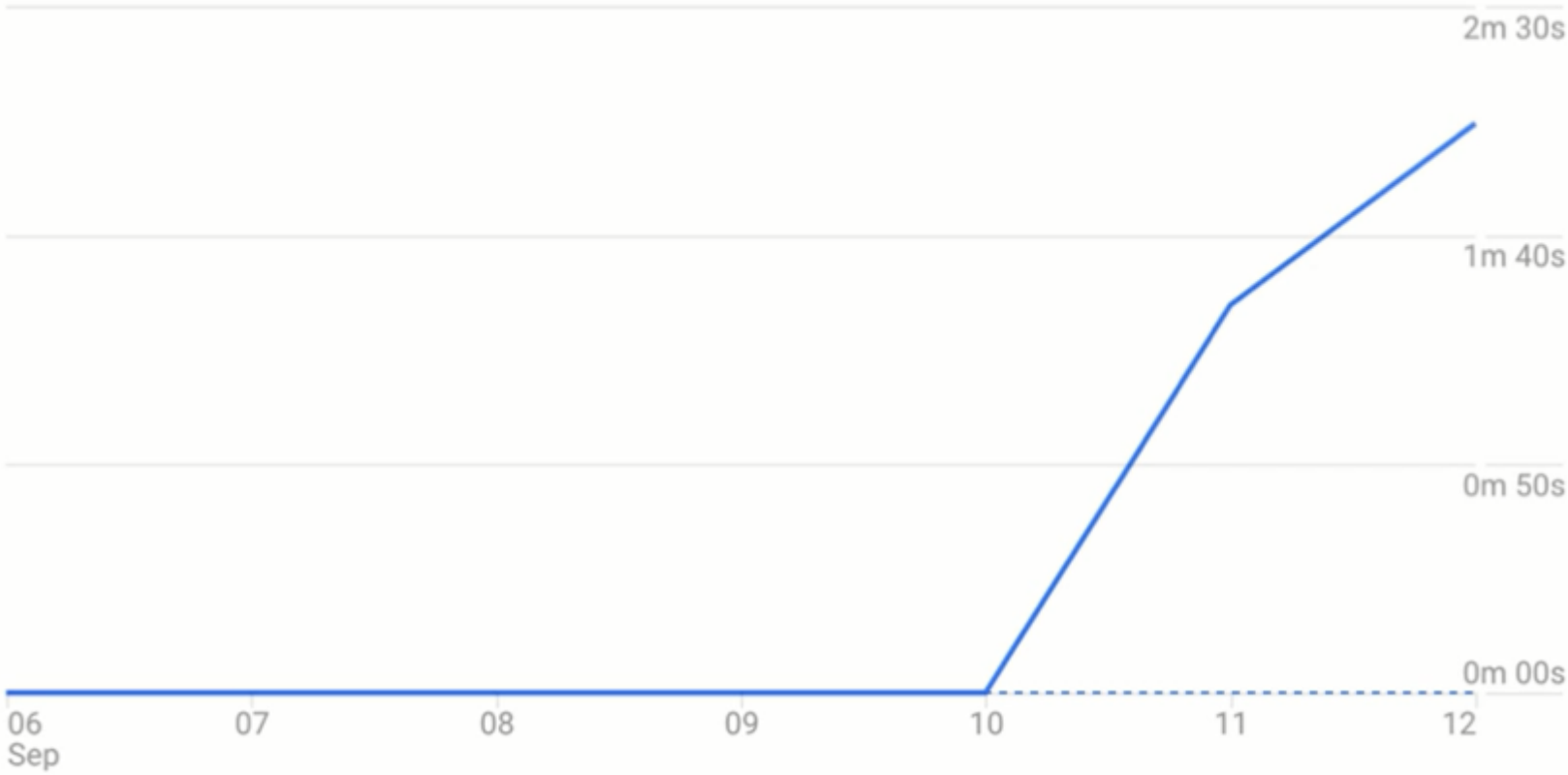
Adım 5: Yorum

$$H_0: \mu = 20$$
$$H_1: \mu \neq 20$$

4. adımda geçirilen sürenin 20 saniye olduğunu iddia eden H0 hipotezi reddedilmiştir. Buna göre kullanıcılar istatistiksel olarak yüzde 95 güvenilirlik ile 4. adımda 20 saniyeden farklı zaman geçirmektedir.

İş Uygulaması: Web Sitesinde Geçirilen Sürenin Testi

Users	Sessions	Bounce Rate	**Session Duration**
239	282	64.54%	1m 58s
-	-	-	-



- Problem:

  Web sitemizde geçirilen ortalama süre gerçekten 170 saniye mi?

- Detaylar:
  - Yazılımlardan elde edilen web sitesinde geçirilen ort. süreler var.
  - Bu veriler incelendiğinde bir yönetici ya da çalışanımız bu değerlerin böyle olmadığına yönelik düşünceler taşıyor ve bu durumu test etmek istiyorlar.

$$H_0: \mu = 170$$
$$H_1: \mu \neq 170$$

In [4]:
```
olcumler = np.array([17, 160, 234, 149, 145, 107, 197, 75, 201, 225, 211, 119,
                     157, 145, 127, 244, 163, 114, 145, 65, 112, 185, 202, 146,
                     203, 224, 203, 114, 188, 156, 187, 154, 177, 95, 165, 50, 110,
                     216, 138, 151, 166, 135, 155, 84, 251, 173, 131, 207, 121, 120])
```

In [4]:
```
olcumler[0:10]
```

Out[4]:
```
array([ 17, 160, 234, 149, 145, 107, 197,  75, 201, 225])
```

In [5]:
```
st.describe(olcumler)
```

Out[5]:
```
DescribeResult(nobs=50, minmax=(17, 251), mean=154.38, variance=2578.0363265306123, skewness=-0.32398897278694483, kurtosis=-0.05849823498415
985)
```

"Ortalama 154 olarak gözüküyor. 170'ten küçük işte daha ne yapıyoruz?" dediğinizi tekrar duyar gibiyim. :) Kaç defa açıklayacağım? Biz burada **istatistiksel olarak anlamlı mı** ona bakıyoruz. O zaman size şu soruyu yönelteyim. Bu çektiğimiz örneklem şansa bağlı olamaz mı? Yani biz başka 50 tane örneklem çektiğimizde belki 170'ten fazla gelecek. Bu olamaz mı? Yaa :). Anlaşıldı mı niye hipotez testi yapıyoruz? 😅😅

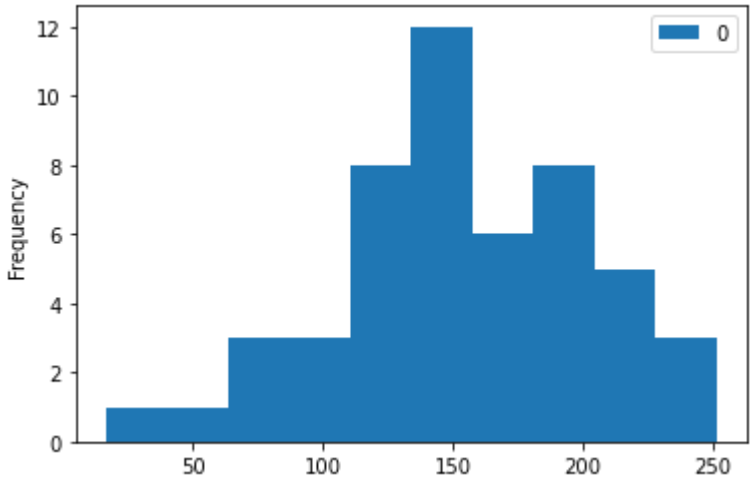
Yani şansa bağlı olarak oluşmuş olma ihtimalini bir hipotez kurarak test ettiğimizde ortadan kaldırmış oluyoruz.

Tek Örneklem T Testi Varsayım Kontrolü

Normallik Varsayımı

- **Histogram grafiği**

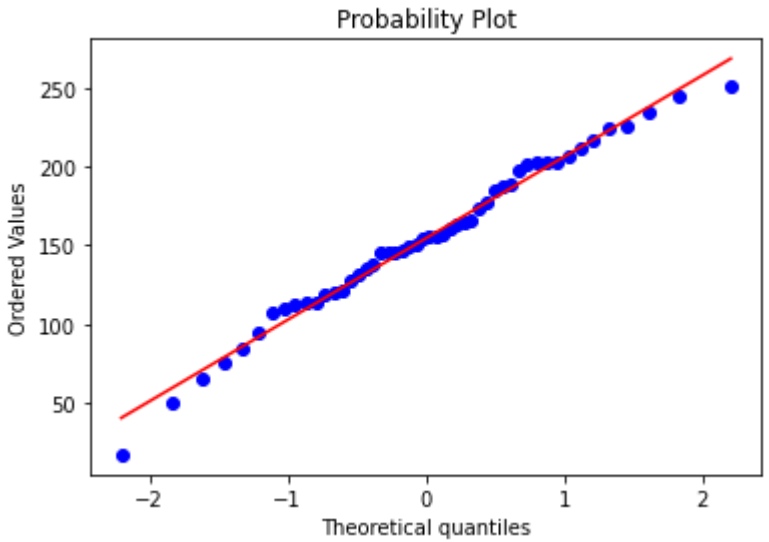
In [6]:
```
pd.DataFrame(olcumler).plot.hist();
```



Grafiğe göre aşırı bir bozukluk görülmemiş olup normal dağıldığı söylenebilir.

- **QQPlot**

In [7]:
```
import pylab
st.probplot(olcumler, dist="norm", plot=pylab)
pylab.show()
```



Bu grafiğe göre mavi noktalar (örnek değerleri) kırmızı çizgi etrafında (teorik dağılım) olduğundan normal dağıldığı söylenebilir.

- **Shapiro-Wilks Testi**

Bu testi yapmak için de hipotez testi yapıyoruz. :)

$H_0:$ Örneklem dağılım ile teorik normal dağılım arasında ist. ol. anl. bir fark yoktur.
$H_1:$ ... fark vardır.

In [8]:
```
st.shapiro(olcumler)
```

Out[8]:
```
ShapiroResult(statistic=0.9853105545043945, pvalue=0.7848747968673706)
```

Görüldüğü üzere pvalue değeleri alpha değerinden büyük olduğundan H0 reddedilemez. Yani örneklem dağılımın normal dağıldığı söylenir.

Tek Örneklem T Testi Uygulaması

Varsayımlar sağlandığına göre t testini uygulayabiliriz.

In [9]:
```
st.ttest_1samp(olcumler, popmean = 170)
```

Out[9]:
```
Ttest_1sampResult(statistic=-2.1753117985877966, pvalue=0.034460415195071446)
```

Görüldüğü üzere $pvalue = 0.03 < alpha = 0.05$ olduğundan H0 reddedilir. Yani **web sitesinde geçirilen ortalama süre 170 saniyeden farklıdır.**
Dolayısıyla bulmuş olduğumuz örneklem ortalaması 154 saniye olduğundan bu süre **170 saniyeden küçüktür** ve **istatistiksel olarak anlamlıdır** diyebiliriz.

Nonparametrik Tek Örneklem Testi

Parametrik testlerde gerekli varsayımlar sağlanmadığında non-parametrik testler uygulanır.

Tek örneklem testi için ilgili varsayımlar sağlanmadığında non-parametrik test olan **Sign test** veya **1 örneklem Wilcoxon testi** uygulanır.

In [5]:
```
from statsmodels.stats.descriptivestats import sign_test
```

In [6]:
```
sign_test(olcumler, 170)
```

Out[6]:
```
(-7.0, 0.06490864707227219)
```

Bu testte görüldüğü üzere $p = 0.06 > alpha = 0.05$ olduğundan H0 reddedilemez. Fakat örneklem dağılımı normal olduğundan bu sonuç ile ilgilenmiyoruz. Eğer varsayımlar sağlanmasaydı bu sonuca dayanarak yorum yapacaktık.

Tek Örneklem Oran Testi

Hatırlatma:

Bilinmeyen ve ulaşılmak istenen anakütle parametresinin değerini, örneklem çekerek belirli testlerden sonra ulaşmayı amaçlıyoruz. O değer tam olarak bulunamamakta fakat çıkarım yapılabilmektedir.

Tanımı:

Oransal bir ifade test edilmek istendiğinde kullanılır.

Hipotezler:

$H_0: p = p_o$ $H_1: p \neq p_o$

$H_0: p <= p_o$ $H_1: p > p_o$

$H_0: p >= p_o$ $H_1: p < p_o$

Test İstatistiği:

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$$

$p:$ Örneklem oranı

$p_0:$ Popülasyon oranı

Varsayımlar:

- $n > 30$ olmalıdır.

İş Uygulaması: Dönüşüm Oranı Testi

Dönüşüm Oranı: O olayı gerçekleştirmek üzere o duruma maruz kalan kişi sayısının ilgilenilen olayı bölümüdür.
Örneğin bir ürün bir sitede satışa çıkarıldığında o ürünü gören sayısı 100, satın alan sayısı 1 ise dönüşüm oranı $1/100$ dür.

- Problem:

  Bir yazılım ile bir mecrada reklam verilmiş ve bu reklama ilişkin yazılım tarafından 0.125 dönüşüm oranı elde edildiği ifade edilmiş. Fakat bu durum kontrol edilmek isteniyor. Çünkü bu yüksek bir oran ve gelirler incelendiğinde örtüşmüyor.

- Detaylar:
  - 500 kişi dış mecrada reklamlara tıklamış, 40 tanesi sitemize gelip alışveriş yapmış.
  - Örnek üzerinden elde edilen dönüşüm oranı: 40/500 = 0,08

Detaylara baktığımızda sanki iki farklı dönüşüm oranı olabilir. Reklamı görenlerin tıklayanlara oranı, veya reklamı görenlerin alışveriş yapanları oranı şeklinde iki dönüşüm oranı olabilir. Fakat burada 2. bahsedilen dönüşüm oranını inceleyeceğiz.

Not: Türkçede ondalıklı sayılar ***virgül*** ile, İngilizcede ise ***nokta*** ile ifade edilir.

Hipotezler:

$$H_0: p = 0.125$$

$$H_1: p \neq 0.125$$

In [1]:
```
from statsmodels.stats.proportion import proportions_ztest
```

In [2]:
```
count = 40 #Başarı sayısı
nobs = 500 #Gözlem sayısı
value = 0.125 #Test etmek istenilen oran
```

Burada fonksiyonun istediği argümanları yazdık.

In [3]:
```
proportions_ztest(count, nobs, value)
```

Out[3]:
```
(-3.7090151628513017, 0.0002080669689845979)
```

Görüldüğü üzere $p < alpha$ olduğundan H_0 hipotezi reddedilir. Yani dönüşüm oranı 0.125'ten farklıdır.

Bağımsız İki Örneklem T Testi

Bağımsız İki Örneklem T Testi Teorisi

İki grup ortalaması arasında karşılaştırma yapılmak istenildiğinde kullanılır. Veri bilimi dünyasında **AB Testi** olarak geçmektedir.

Hipotezler:

$H_0: \mu_1 = \mu_2$	$H_0: \mu_1 <= \mu_2$	$H_0: \mu_1 >= \mu_2$
$H_1: \mu_1 \neq \mu_2$	$H_1: \mu_1 > \mu_2$	$H_1: \mu_1 < \mu_2$

Test İstatistiği:

Örnek sayıları aynı, varyanslar homojen ise:

$$t = \frac{\overline{X}_1 - \overline{X}_2}{S_p\sqrt{\frac{2}{n}}}, \quad S_p = \sqrt{\frac{s^2_{X1} + s^2_{X2}}{2}}$$

Örnek sayısı farklı, varyanslar homojen ise:

$$t = \frac{\overline{X}_1 - \overline{X}_2}{S_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, \quad S_p = \sqrt{\frac{(n_1 - 1)s^2_{X_1} + (n_2 - 1)s^2_{X2}}{n_1 + n_2 - 2}}$$

Örnek sayıları farklı varyanslar homojen değil ise:

$$t = \frac{\overline{X}_1 - \overline{X}_2}{S_{\overline{\Delta}}}, \qquad S_{\overline{\Delta}} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

Önemli Not: Son iki test için örnek sayıları aynı olsa bile kullanılabilir. Son teste **Welch testi** de denir.

Varsayımlar:

- Normallik
- Varyans Homojenliği

İş Uygulaması: ML Modelinin Başarı Testi

- Problem:

  Bir ML projesine yatırım yapılmış. Ürettiği tahminler neticesinde oluşan gelir ile eski sistemin ürettiği gelirler karşılaştırılıp anlamlı farklılık olup olmadığı test edilmek isteniyor.

- Detaylar:
  - Model geliştirilmiş ve web sitesine entegre edilmiş.
  - Site kullanıcıları belirli bir kurala göre ikiye bölünmüş olsun.
  - A grubu eski B grubu yeni sistem.
  - Gelir anlamında anlamlı bir iş yapılıp yapılmadığı test edilmek isteniyor.

Hipotezler:

$$H_0: \mu_1 = \mu_2$$

$$H_1: \mu_1 \neq \mu_2$$

Sözel olarak:

$H_0:$ Eski sistemin gösterilerine göre insanlar ürün aldığında belirli bir süre zarfında ortaya çıkan gelirlerin ortalaması ile yeni sistemin önerilerine göre insanlar ürünleri satın aldığında ortaya çıkan gelirlerin ortalaması birbirine eşittir.
$H_1:$ ... eşit değildir.

Kodu:

Veri Tipi I (1. Senaryo)

Veriler farklı formatlarda (.xlsx, .csv, .txt) gelebilir veya veri tabanından çekmek durumunda kalabiliriz. Ayrı ayrı gelen verileri birleştirmek için aşağıda yazılan kod kullanılabilir.

In [2]:
```
A = pd.DataFrame([30,27,21,27,29,30,20,20,27,32,35,22,24,23,25,27,23,27,23,
                  25,21,18,24,26,33,26,27,28,19,25])

B = pd.DataFrame([37,39,31,31,34,38,30,36,29,28,38,28,37,37,30,32,31,31,27,
                  32,33,33,33,31,32,33,26,32,33,29])

A_B = pd.concat([A, B], axis = 1)
```

```
A_B.columns = ["A","B"]

A_B.head()
```

Out[2]:

	A	B
0	30	37
1	27	39
2	21	31
3	27	31
4	29	34

Veri Tipi II (2. Senaryo)

Kullanacağımız bazı fonksiyonlar argümanları neticesinde bizden veriyi isteği doğrultusunda biçimlendirmemizi isteyebilir. Örneğin yeni bir sütun ekleyip grupların olduğu bir değişken oluşturmak gerekebilir. Böyle durumlarda aşağıda yazılan kod kullanılabilir.
-Okuyan kişi: Türkçen bayağı iyiymiş karşim. :))

In [3]:

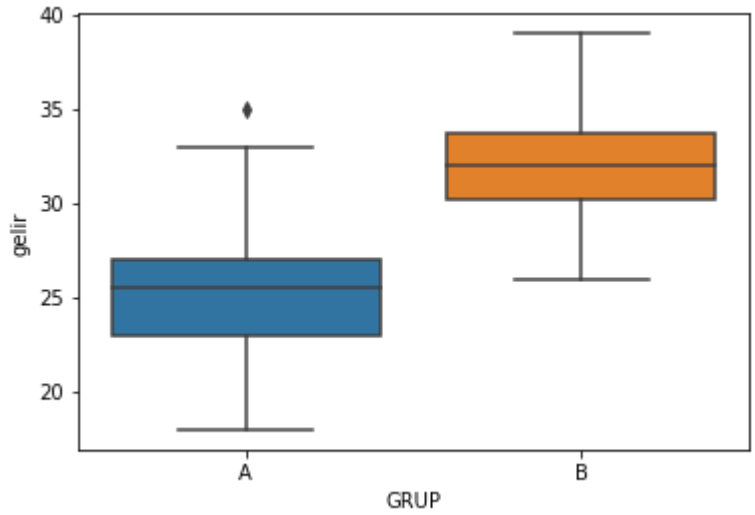
```
A = pd.DataFrame([30,27,21,27,29,30,20,20,27,32,35,22,24,23,25,27,23,27,23,
                  25,21,18,24,26,33,26,27,28,19,25])

B = pd.DataFrame([37,39,31,31,34,38,30,36,29,28,38,28,37,37,30,32,31,31,27,
                  32,33,33,33,31,32,33,26,32,33,29])

#A ve A'nın Grubu
GRUP_A = np.arange(len(A))
GRUP_A = pd.DataFrame(GRUP_A)
GRUP_A[:] = "A"
A = pd.concat([A, GRUP_A], axis = 1)

#B ve B'nin Grubu
GRUP_B = np.arange(len(B))
GRUP_B = pd.DataFrame(GRUP_B)
GRUP_B[:] = "B"
B = pd.concat([B, GRUP_B], axis = 1)

#Tüm veri
AB = pd.concat([A, B])
AB.columns = ["gelir", "GRUP"]
print(AB.head())
print(AB.tail())
```

```
   gelir GRUP
```

```
0     30    A
1     27    A
2     21    A
3     27    A
4     29    A
    gelir GRUP
25     33    B
26     26    B
27     32    B
28     33    B
29     29    B
```

Grafikleri:

In [5]:
```
sns.boxplot(x = "GRUP", y = "gelir", data = AB);
```



Grafikten görüldüğü üzere B sisteminin gelir olarak daha yukarıda olduğu gözleniyor. Fakat bu farkın **istatistiksel olarak anlamlı olup olmadığını** test etmemiz gerekiyor.

Bağımsız İki Örneklem T Testi Varsayım Kontrolü

In [6]:
```
A_B.head()
```

Out[6]:

	A	B
0	30	37
1	27	39
2	21	31

	A	B
3	27	31
4	29	34

In [7]:
```
AB.head()
```

Out[7]:

	gelir	GRUP
0	30	A
1	27	A
2	21	A
3	27	A
4	29	A

Normallik Varsayımı:

H_0: Normal dağılıma uygundur.
H_1: Normal dağılıma uygun değildir.

`shapiro()` fonksiyonu bizden tek bir değişken yazmamızı bekliyor. Bu yüzden ilk veri setini kullanıp hem A'nın hem de B'nin normalliğine bakmalıyız.

In [8]:
```
st.shapiro(A_B.A)
```

Out[8]:
```
ShapiroResult(statistic=0.9789242148399353, pvalue=0.7962799668312073)
```

In [9]:
```
st.shapiro(A_B.B)
```

Out[9]:
```
ShapiroResult(statistic=0.9561260342597961, pvalue=0.24584221839904785)
```

Sonuçlardan görüldüğü üzere $p > alpha$ olduğundan H_0 hipotezi reddedilemez. Yani normallik varsayımı sağlanmaktadır.

Varyans Homojenliği Varsayımı:

H_0: Varyanslar homojendir.
H_1: Varyanslar homojen değildir.

In [10]:
```
st.levene(A_B.A, A_B.B)
```

Out[10]:
```
LeveneResult(statistic=1.1101802757158004, pvalue=0.2964124900636569)
```

Sonuçtan görüldüğü üzere H_0 reddedilemez yani varyans homojenliği varsayımı da sağlanmaktadır.

Bağımsız İki Örneklem T Testi Uygulaması

Gerekli aşamaları inceleyip tüm varsayımlar sağlandığına göre testi uygulayabiliriz.

In [11]:
```
st.ttest_ind(A_B["A"], A_B["B"], equal_var = True)

#equal_var: Varyans homojenliği (True veya False)
```

Out[11]:
```
Ttest_indResult(statistic=-7.028690967745927, pvalue=2.6233215605475075e-09)
```

Virgülden sonraki basamak sayısı çok fazla olduğundan formatlarını değiştirelim:

In [12]:
```
test_istatistigi, pvalue = st.ttest_ind(A_B["A"], A_B["B"], equal_var = True)
print('Test istatistiği = %.4f, p-değeri = %.4f' % (test_istatistigi, pvalue))
```

```
Test istatistiği = -7.0287, p-değeri = 0.0000
```

Sonuçtan görüldüğü üzere $p < alpha$ olduğunda H_0 reddedilir. Yorum olarak ise, eski sistem olan A sisteminden elde edilen gelir ile makine öğrenmesi modeli uygulanan B sistemi birbirinden farklıdır. Yukarıda daha önce çizilen boxplot grafiğinde B sistemin geliri daha yüksek olduğundan bu fark **istatistiksel olarak anlamlıdır** ve dolayısıyla B sisteminden elde edilen gelir daha çoktur.

Nonparametrik Bağımsız İki Örneklem Testi

Gerekli varsayımlar sağlanmadığında bağımsız iki örneklem t testinin non parametrik alternatifi olan **Mann-Whitney U Testi** uygulanır.

In [14]:
```
st.mannwhitneyu(A_B["A"], A_B["B"])
```

Out[14]:
```
MannwhitneyuResult(statistic=89.5, pvalue=4.778975189306267e-08)
```

In [15]:
```
test_istatistigi, pvalue = st.mannwhitneyu(A_B["A"], A_B["B"])
print('Test istatistiği = %.4f, p-değeri = %.4f' % (test_istatistigi, pvalue))
```

```
Test istatistiği = 89.5000, p-değeri = 0.0000
```

Bu sonuçta da görüldüğü üzere H_0 reddedilir.

Bağımlı İki Örneklem T Testi

Bağımlı İki Örneklem T Testi Teorisi

Bağımlı iki grup ortalaması arasında karşılaştırma yapılmak istendiğinde kullanılır. Bağımlıdan kasıt bir olaydan öncesi ve sonrasını ifade eder.

Hipotezler:

$H_0: \mu_{ö} = \mu_s$ $H_0: \mu_{ö} <= \mu_s$ $H_0: \mu_{ö} >= \mu_s$

$H_1: \mu_{ö} \neq \mu_s$ $H_1: \mu_{ö} > \mu_s$ $H_1: \mu_{ö} < \mu_s$

Test İstatistiği:

$$t = \frac{\overline{x}_D - \mu_0}{\frac{s_D}{\sqrt{n}}}$$

Varsayımlar:

- Normallik
- Varyans Homojenliği

İş Uygulaması: Şirket İçi Eğitimin Performans Etkisi Ölçümü

- Problem:

  Belirli uğraşlar sonucunda alınan bir eğitimin katma değer sağlayıp sağlamadığı ölçülmek isteniyor.

- Detaylar:
  - Bir departman bir konuda eğitim talep ediyor
  - Gerekli/gereksiz değerlendirmeleri neticesinde eğitim alınıyor
  - Eğitimden önce ve sonra olacak şekilde gerekli ölçümler yapılıyor
  - Eğitim sonrasında eğitimin sağladığı katma değer test edilmek isteniyor

Hipotez:

$$H_0: \mu_{ö} = \mu_{s}$$

$$H_1: \mu_{ö} \neq \mu_{s}$$

Sözel olarak:

H_0: Eğitim öncesi ve sonrasındaki performans (gelir vs.) birbirine eşit olup eğitim etkili olmamıştır.

H_1: ... eşit olmayıp eğitim etkili olmuştur.

Kod:

In [5]:
```
oncesi = pd.DataFrame([123,119,119,116,123,123,121,120,117,118,121,121,123,119,
                       121,118,124,121,125,115,115,119,118,121,117,117,120,120,
                       121,117,118,117,123,118,124,121,115,118,125,115])

sonrasi = pd.DataFrame([118,127,122,132,129,123,129,132,128,130,128,138,140,130,
                        134,134,124,140,134,129,129,138,134,124,122,126,133,127,
                        130,130,130,132,117,130,125,129,133,120,127,123])
```

In [6]:
```
oncesi[0:5]
```

Out[6]:

	0
0	123

	0
1	119
2	119
3	116
4	123

In [7]:
```
sonrasi[0:5]
```

Out[7]:

	0
0	118
1	127
2	122
3	132
4	129

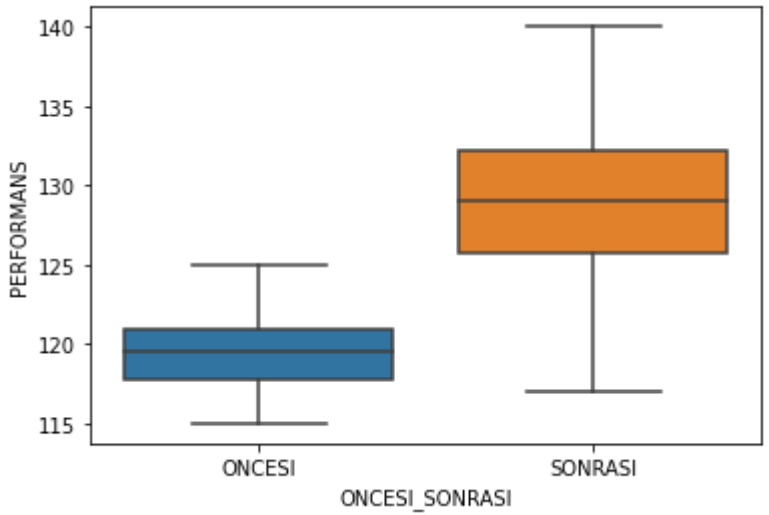
In [8]:
```
#BIRINCI VERI SETI
AYRIK = pd.concat([oncesi, sonrasi], axis = 1)
AYRIK.columns = ["ONCESI", "SONRASI"]
print("'AYRIK' Veri Seti: \n\n", AYRIK.head(), "\n\n")

#IKINCI VERI SETI
#ONCESI FLAG/TAG'INI OLUSTURMA
GRUP_ONCESI = np.arange(len(oncesi))
GRUP_ONCESI = pd.DataFrame(GRUP_ONCESI)
GRUP_ONCESI[:] = "ONCESI"
#FLAG VE ONCESI DEGERLERINI BIR ARAYA GETIRME
A = pd.concat([oncesi, GRUP_ONCESI], axis = 1)

#SONRASI FLAG/TAG'INI OLUSTURMA
GRUP_SONRASI = np.arange(len(sonrasi))
GRUP_SONRASI = pd.DataFrame(GRUP_SONRASI)
GRUP_SONRASI[:] = "SONRASI"
#FLAG VE ONCESI DEGERLERINI BIR ARAYA GETIRME
B = pd.concat([sonrasi, GRUP_SONRASI], axis = 1)

#TUM VERIYI BIR ARAYA GETIRME
BIRLIKTE = pd.concat([A, B])
BIRLIKTE
```

```
#ISIMLENDIRME
BIRLIKTE.columns = ["PERFORMANS", "ONCESI_SONRASI"]
print("'BIRLIKTE' Veri Seti: \n\n", BIRLIKTE.head(), "\n")
```

```
'AYRIK' Veri Seti:

    ONCESI  SONRASI
0      123      118
1      119      127
2      119      122
3      116      132
4      123      129


'BIRLIKTE' Veri Seti:

    PERFORMANS ONCESI_SONRASI
0          123         ONCESI
1          119         ONCESI
2          119         ONCESI
3          116         ONCESI
4          123         ONCESI
```

Grafik:

In [10]:
```
sns.boxplot(x = "ONCESI_SONRASI", y = "PERFORMANS", data = BIRLIKTE);
```



Grafikten görüldüğü üzere eğitimden sonrası ifade eden "SONRASI" değişkeninin yukarıda olduğu gözleniyor. Bu farkın **istatistiksel olarak anlamlı olup olmadığını** inceleyeceğiz.

Bağımlı İki Örneklem T Testi Varsayım Kontrolü

Hatırlatma: Bu varsayımları yapmamızın sebebi gerekli varsayımlar sağlandığında ilgili testin bilimsel bir referans/karşılaştırma noktasının güvenli bir şekilde vermesi, arka planda yapılan standartlaştırma işlemleri sonucunda verilebilecek kararları bilimsel olarak verebilme imkanını sağlar.

Normallik:

In [11]:
```
st.shapiro(AYRIK.ONCESI)
```

Out[11]:
```
ShapiroResult(statistic=0.9543656706809998, pvalue=0.10722451657056808)
```

In [12]:
```
st.shapiro(AYRIK.SONRASI)
```

Out[12]:
```
ShapiroResult(statistic=0.9780089259147644, pvalue=0.6159515380859375)
```

Sonuçlardan görüldüğü üzere $p > alpha$ olduğundan H_0 reddedilemez ve normal dağıldığı söylenebilir.

Bağımlı 2 örneklem t testinde değişkenlerin ayrı ayrı normalliğine bakabildiğimiz gibi, değişken değerlerinin farklarını alıp o farkın oluştuğu tek değişkenin normalliğine de bakabiliriz. **Fakat önerilen ayrı ayrı bakılmasıdır.**

Varyans Homojenliği:

In [13]:
```
st.levene(AYRIK.ONCESI, AYRIK.SONRASI)
```

Out[13]:
```
LeveneResult(statistic=8.31303288672351, pvalue=0.0050844511807370246)
```

Sonuçtan görüldüğü üzere $p < alpha$ olduğundan H_0 reddedilir. Yani varyanslar homojen olmayıp bu varsayım sağlanmamaktadır. Böyle bir durumda iki yol izlenebilir.

1. Sadece bu varsayım için değil, bir veya birden fazla varsayım sağlanmazsa aykırı değer analizi ve standartlaştırma işlemleri gerçekleştirilip ilgili varsayımlar sağlanmaya zorlanır.
2. Varyans homojenliği sağlanmadığında bu varsayım bir miktar göz ardı edilebilir ve ilgili test gerçekleştirilebilir. Non-parametrik teste geçme zorunluluğu olmaz.

Bağımlı İki Örneklem T Testi Uygulaması

In [14]:
```
st.ttest_rel(AYRIK.ONCESI, AYRIK.SONRASI)
```

Out[14]:
```
Ttest_relResult(statistic=-9.281533480429937, pvalue=2.0235251764440722e-11)
```

In [15]:
```
test_istatistigi, pvalue = st.ttest_rel(AYRIK.ONCESI, AYRIK.SONRASI)
print('Test istatistiği = %.4f, p-değeri = %.4f' % (test_istatistigi, pvalue))
```

```
Test istatistiği = -9.2815, p-değeri = 0.0000
```

Sonuçtan görüldüğü üzere $p < alpha$ olduğundan H_0 reddedilir. Yorum olarak ise eğitim öncesi performans ile eğitim sonrası performans farklı olup bu fark **istatistiksel olarak anlamlıdır** ve boxplot grafikten görüldüğü üzere eğitim sonrası performans daha fazladır. Yani eğitim işe yaramıştır. :)

Nonparametrik Bağımlı İki Örneklem Testi

Bağımlı iki örneklem t testi için gerekli varsayımlar sağlanmadığında **Wilcoxon testi** uygulanır.

In [16]:
```
st.wilcoxon(AYRIK.ONCESI, AYRIK.SONRASI)
```

Out[16]:
```
WilcoxonResult(statistic=15.0, pvalue=2.491492033374464e-07)
```

In [17]:
```
test_istatistigi, pvalue = st.wilcoxon(AYRIK.ONCESI, AYRIK.SONRASI)
print('Test istatistiği = %.4f, p-değeri = %.4f' % (test_istatistigi, pvalue))
```

```
Test istatistiği = 15.0000, p-değeri = 0.0000
```

Sonuçtan görüldüğü üzere $p < alpha$ olduğundan H_0 reddedilir.

Önemli Not: Python'da birçok hipotez testi fonksiyonları tek yönlü(<, >) hipotezi desteklememektedir. Sadece "farklıdır" hipotezi kurulabilmektedir. Bunun sebebi "farklıdır" hipotezi kabul edildiğinde ortalamalarına veya grafiğine bakılarak hangisini daha fazla veya daha az olduğu açıkça görülebilir.

İki Örneklem Oran Testi

İki Örneklem Oran Testi Teorisi

İki oran arasında karşılaştırma yapmak için kullanılır.

Hipotezler:

$H_0: p_1 = p_2$ $\quad$ $H_0: p_1 <= p_2$ $\quad$ $H_0: p_1 >= p_2$

$H_1: p_1 \neq p_2$ $\quad$ $H_1: p_1 > p_2$ $\quad$ $H_1: p_1 < p_2$

Test İstatistiği:

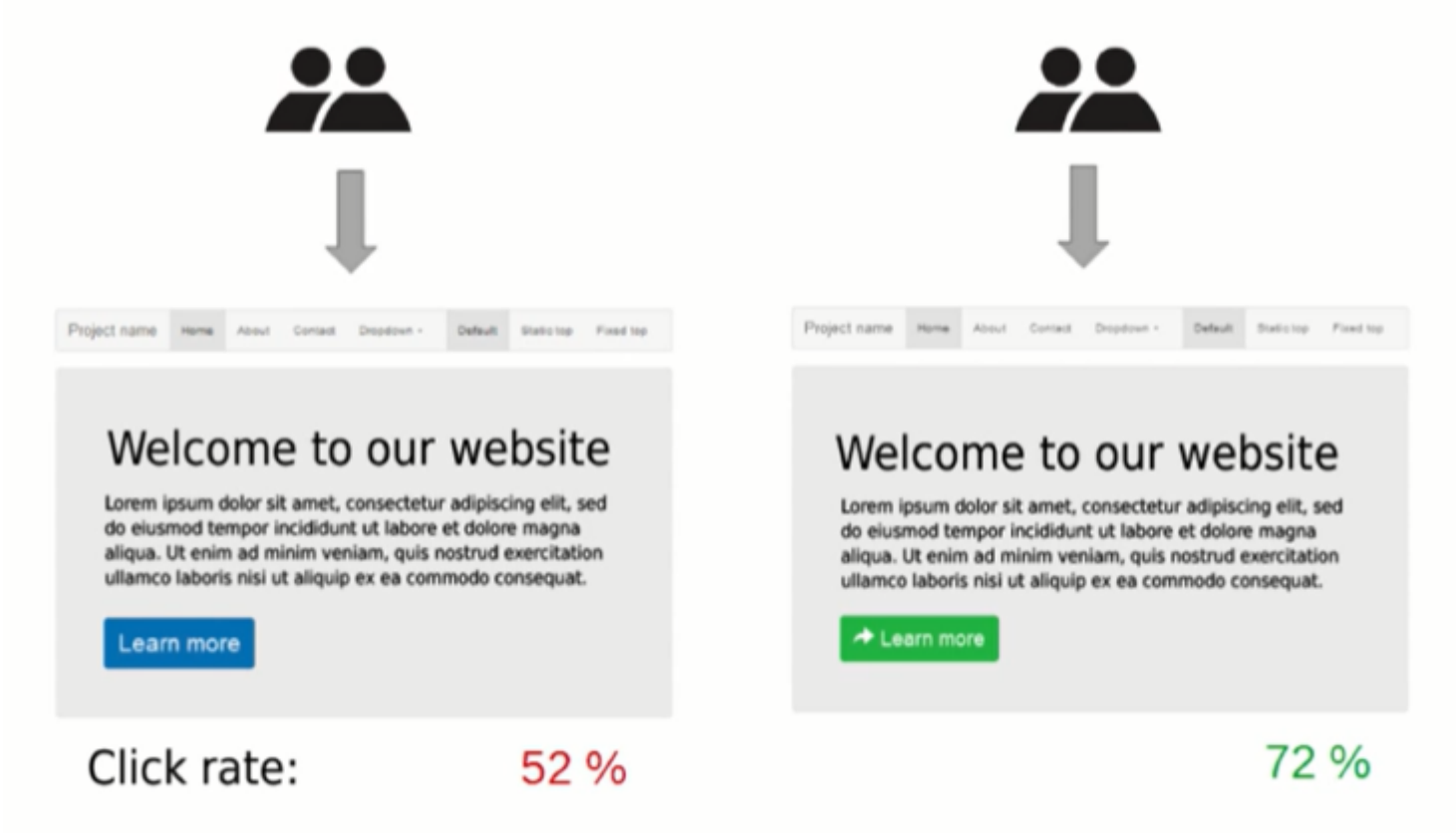
$$Z_h = \frac{(p_1 - p_2)}{\sqrt{p(1-p)(\frac{1}{n_1} + \frac{1}{n_2})}}$$

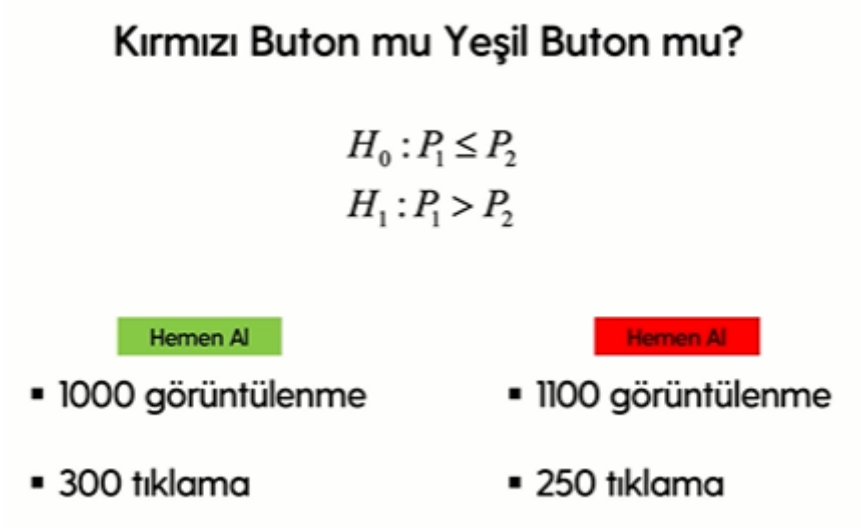
Varsayımlar:

- n1 > 30
- n2 > 30

İş Uygulaması: Kullanıcı Arayüzü Deneyi

Bu da t testi gibi literatürde **AB Testi** olarak geçmektedir.





Bu sonuçlara göre yeşil butona tıklama oranı %30, kırmızı butona tıklama oranı %22,7'dir.

In [2]:
```
from statsmodels.stats.proportion import proportions_ztest
```

In [5]:
```
basari_sayisi = np.array([300, 250])
gozlem_sayilari = np.array([1000, 1100])
```

In [6]:
```
proportions_ztest(count = basari_sayisi, nobs = gozlem_sayilari)
```

Out[6]:
```
(3.7857863233209255, 0.0001532232957772221)
```

Sonuçtan görüldüğü üzere $p < alpha$ olduğundan H_0 reddedilir. Yani oranlar birbirinden farklı olup bu fark **istatistiksel olarak anlamlıdır** ve yeşil buton tıklamaya daha çok etki etmiştir.

Varyans Analizi

Varyans Analizi Teorisi

İki ya da daha fazla grup ortalaması arasında istatistiksel olarak anlamlı farklılık olup olmadığı öğrenilmek istenildiğinde kullanılır.

Hipotez:

H_0: $\mu_1 = \mu_2 = \mu_3$

H_1: Eşit değillerdir (en az birisi farklıdır)

Test İstatistiği:

$$F_s = \frac{MS_{(between)}}{MS_{(within)}}$$

Varsayımlar:

- Gözlemlerin birbirinden bağımsız olması (grupların)
- Normal dağılım
- Varyans homojenliği

Önemli Not: Varyans analizinde **varyans homojenliği** varsayımı **muhakkak** sağlanmalıdır.

İş Uygulaması: Anasayfa İçerik Stratejisi Belirleme

- Problem:

  Anasayfa'da geçirilen süre artırılmak isteniyor

- Detaylar:
  - Bir web sitesi için başarı kriterleri: ortalama ziyaret süresi, hemen çıkış oranı vb
  - Uzun zaman geçiren kullanıcıların reklamlara daha fazla tıkladığı ve markaya olan bağlılıklarının arttığı biliniyor.
  - Buna yönelik olarak benzer haberler farklı resimler ya da farklı formatlarda hazırlanarak oluşturulan test gruplarına gösteriliyor.
  - A: Doğal Şekilde, B: Yönlendirici, C: İlgi Çekici

In [12]:
```
A = pd.DataFrame([28,33,30,29,28,29,27,31,30,32,28,33,25,29,27,31,31,30,31,34,30,32,31,34])

B = pd.DataFrame([31,32,30,30,33,32,34,27,36,30,31,30,38,29,30,34,34,31,35,35,33,30,28,29])

C = pd.DataFrame([40,33,38,41,42,43,38,35,39,39,36,34,35,40,38,36,39,36,33,35,38,35,40,40])
#Veriler normalde daha fazla fakat hoca göstermedi. :(
#Bu yüzden sonuçlar bir tık farklı olacak fakat genel sonuç değişmeyecektir.

dfs = [A, B, C]

ABC = pd.concat(dfs, axis = 1)
ABC.columns = ["GRUP_A", "GRUP_B", "GRUP_C"]
ABC.head()
```

Out[12]:

	GRUP_A	GRUP_B	GRUP_C
0	28	31	40
1	33	32	33
2	30	30	38
3	29	30	41
4	28	33	42

Varyans Analizi Varsayım Kontrolü

İlk varsayım olan grupların bağımsız olması varsayımı, gruplar birbirine etki etmediği için varsayımın sağlandığı kabul edilir.

Normallik:

In [8]:
```
st.shapiro(ABC.GRUP_A)
```

Out[8]:
```
ShapiroResult(statistic=0.9731682538986206, pvalue=0.7451096177101135)
```

In [9]:
```
st.shapiro(ABC.GRUP_B)
```

Out[9]:
```
ShapiroResult(statistic=0.9626312851905823, pvalue=0.49352705478668213)
```

In [10]:
```
st.shapiro(ABC.GRUP_C)
```

Out[10]:
```
ShapiroResult(statistic=0.9583170413970947, pvalue=0.4055526554584503)
```

Sonuçlardan görüldüğü üzere tüm p değerleri alpha'dan büyük olduğundan H_0 hipotezi reddedilemez ve normal dağıldığı söylenir.

Varyans Homojenliği:

In [11]:
```
st.levene(ABC.GRUP_A, ABC.GRUP_B, ABC.GRUP_C)
```

Out[11]:
```
LeveneResult(statistic=0.6585903083700435, pvalue=0.5208031835357625)
```

Görüldüğü üzere p değeri alpha'dan büyük olduğundan H_0 reddedilemez ve varyans homojenliği de sağlanmaktadır.

Varyans Analizi Hipotez Testinin Uygulanması

In [13]:
```
st.f_oneway(ABC.GRUP_A, ABC.GRUP_B, ABC.GRUP_C)
```

Out[13]:
```
F_onewayResult(statistic=54.9976, pvalue=5.215138541582314e-15)
```

Virgülden sonra çok basamak olduğundan önceki derslerden farklı olarak formatını değiştirelim.

In [14]:
```
print('{:.5f}'.format(st.f_oneway(ABC.GRUP_A, ABC.GRUP_B, ABC.GRUP_C)[1]))
```

```
0.00000
```

`st.f_oneway(ABC.GRUP_A, ABC.GRUP_B, ABC.GRUP_C)[1]` yazarak pvalue'yu almış olduk.

Sonucu yorumlarsak $p < alpha$ olduğundan H_0 reddedilir. Yani ilgili gruplardan en az biri site tasarımına göre sitede zaman geçirme ortalaması farklıdır. "Bu

grup veya gruplar hangisi?" şeklinde sorulursa ortalamaları incelenebilir.

In [15]:
```
ABC.describe().T
```

Out[15]:

	count	mean	std	min	25%	50%	75%	max
GRUP_A	24.0	30.125	2.290102	25.0	28.75	30.0	31.25	34.0
GRUP_B	24.0	31.750	2.706675	27.0	30.00	31.0	34.00	38.0
GRUP_C	24.0	37.625	2.794599	33.0	35.00	38.0	40.00	43.0

Görüldüğü üzere C grubunun ortalaması farklı olup bu gruptaki kişiler sitede daha fazla zaman geçirmiştir. Site tasarımı C grubuna göre yapılmalıdır.

Nonparametrik Hipotez Testi

Varyans analizinin gerekli varsayımları sağlanmadığında **Kruskal-Wallis H testi** uygulanır.

In [16]:
```
st.kruskal(ABC.GRUP_A, ABC.GRUP_B, ABC.GRUP_C)
```

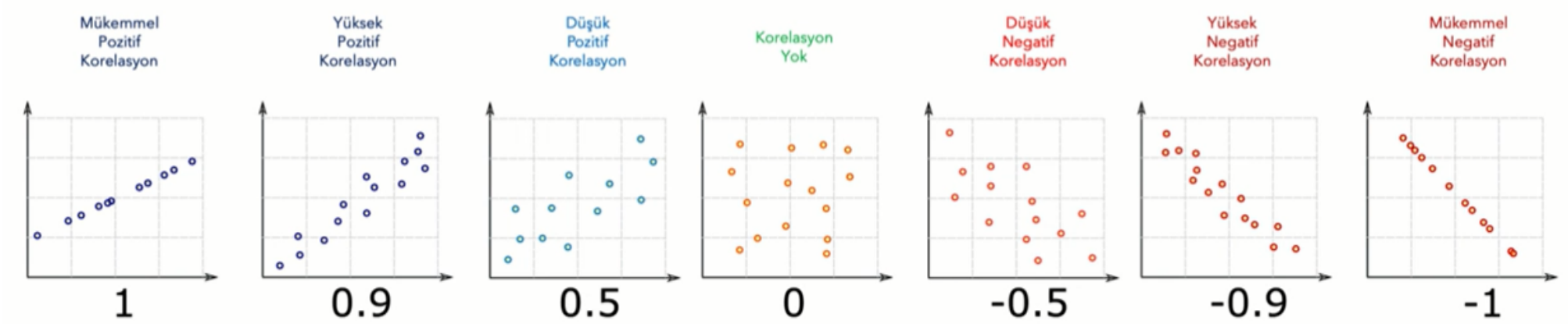
Out[16]:
```
KruskalResult(statistic=42.43187215942825, pvalue=6.10992715116595e-10)
```

Sonuçtan görüldüğü üzere parametrikte olduğu gibi $p < alpha$ olduğundan H_0 reddedilir.

Korelasyon Analizi

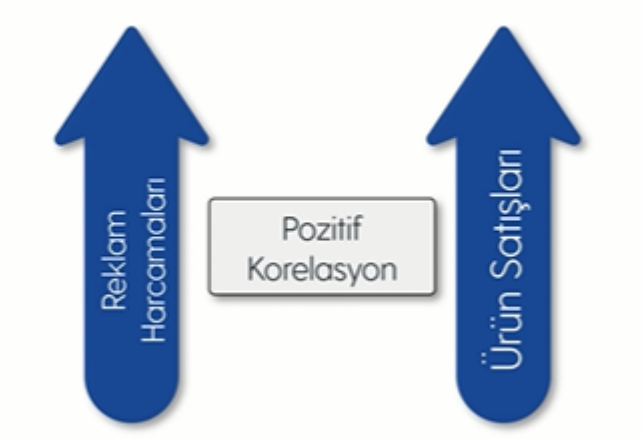
Korelasyon Analizi Teorisi

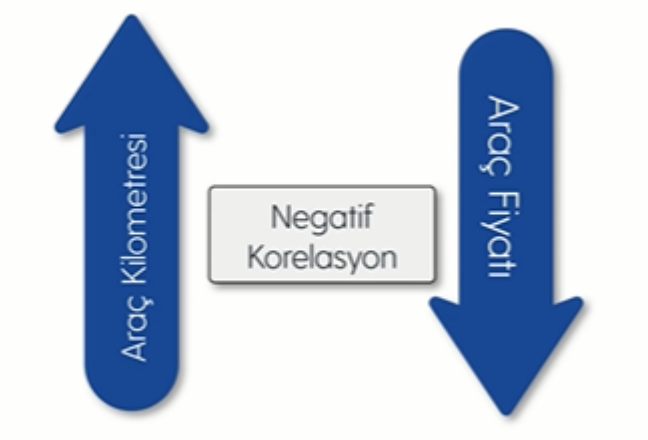
Değişkenler arasındaki ilişki, bu ilişkinin yönü ve şiddeti ile ilgili bilgiler sağlayan istatistiksel bir yöntemdir.



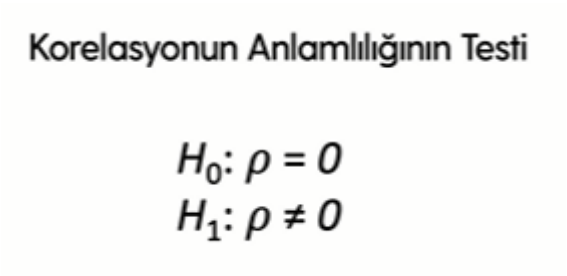
0.5 - 0.9 değerleri arasındaki ilişki **anlamlı** olarak değerlendirilir.

Örnekler:





Hipotezler:



Hesaplanan korelasyon katsayısının ($ρ$: ro, yani popülasyona ilişkin korelasyon katsayısı) anlamlı olup olmadığının testini yapacağız.

Sözel olarak:

H_0: Değişkenler arasında ilişki yoktur.

H_1: ... vardır.

Test İstatistiği:

$$r_{xy} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sqrt{(\sum x_i^2 - n\bar{x}^2)}\sqrt{(\sum y_i^2 - n\bar{y}^2)}}$$

$$t = r\sqrt{\frac{n-2}{1-r^2}}$$

r_{xy}: Pearson Korelasyon Katsayısı(örneklem korelasyon katsayısına karşılık gelir.)
t: Test istatistiği

Varsayımlar:

- İki değişken içinde normallik varsayımı.
- Varsayım sağlanıyorsa Pearson Korelasyon Katsayısı
- Varsayım sağlanmıyorsa Spearman Korelasyon Katsayısı

İş Uygulaması: Bahşiş İle Ödenen Hesap Arasındaki İlişkinin İncelenmesi

Hipotez:

$$H_0: \rho = 0$$
$$H_1: \rho \neq 0$$

Önceki derslerde kullanılan "tips" veri seti kullanılacaktır.

total_bill: Yemeğin toplam fiyatı (bahşiş ve vergi dahil)
tip: Bahşiş
sex: Ücreti ödeyen kişinin cinsiyeti (0: male, 1: female)
smoker: Grupta sigara içen var mı? (0: No, 1: Yes)
day: Gün (3: Thur, 4: Fri, 5: Sat, 6: Sun)

time: Ne zaman (0: Lunch(ya da Gündüz), 1: Dinner(ya da Akşam))

size: Grupta kaç kişi var?

In [2]:
```
tips = sns.load_dataset("tips")
df = tips.copy()
df.head()
```

Out[2]:

	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

Buradaki sorun **total_bill**(toplam hesap) değişkenine **tip**(bahşiş) değişkeni de eklenerek hesaplanmıştır. **Korelasyona bakılırken bu çok dikkat edilmelidir. Yani bir değişken diğerini etkilemişse öncelikle bu tespit edilip arındırılmalıdır.**

Bunun için toplam hesaptan bahşişi çıkarmamız yeterlidir.

In [4]:
```
df["total_bill"] = df["total_bill"] - df["tip"]
```

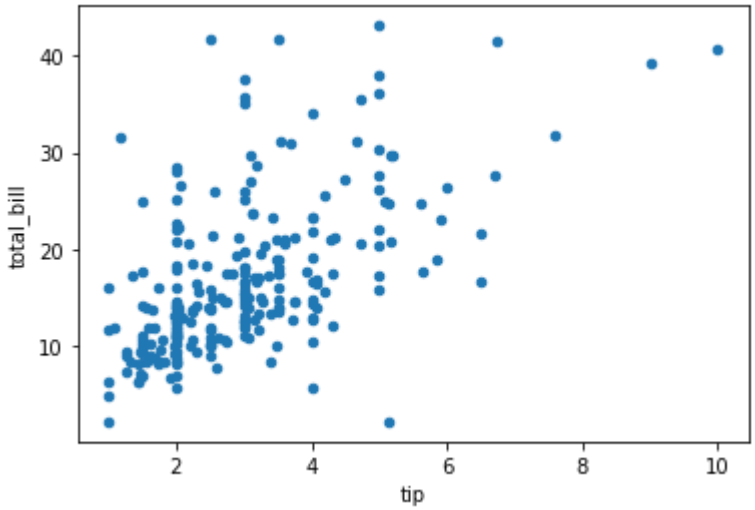
In [5]:
```
df.head()
```

Out[5]:

	total_bill	tip	sex	smoker	day	time	size
0	15.98	1.01	Female	No	Sun	Dinner	2
1	8.68	1.66	Male	No	Sun	Dinner	3
2	17.51	3.50	Male	No	Sun	Dinner	3
3	20.37	3.31	Male	No	Sun	Dinner	2
4	20.98	3.61	Female	No	Sun	Dinner	4

Şimdi bu değişkenlerin grafiğini çizdirelim. Bunun için "Pandas"ın kendi kütüphanesinden yararlanalım:

In [6]:
```
df.plot.scatter("tip", "total_bill");
```



Grafikten görüldüğü üzere hesap arttıkça bahşişlerin de arttığı gözlemleniyor. Bu artışın **anlamlı olup olmadığını** bulalım.

Korelasyon Varsayım Kontrolü

Normallik Varsayımı:

In [7]:
```
test_istatistigi, pvalue = st.shapiro(df.tip)
print('Test istatistiği = %.4f, p-değeri = %.4f' % (test_istatistigi, pvalue))

test_istatistigi, pvalue = st.shapiro(df.total_bill)
print('Test istatistiği = %.4f, p-değeri = %.4f' % (test_istatistigi, pvalue))
```

```
Test istatistiği = 0.8978, p-değeri = 0.0000
Test istatistiği = 0.9136, p-değeri = 0.0000
```

Sonuçlardan görüldüğü üzere $p < alpha$ olduğundan normallik varsayımı için H_0 reddedilir. Yani **örnek dağılım ile teorik normal dağılım arasında anlamlı bir farklılık vardır** demiş olup normallik varsayımı sağlanmamaktadır.

Fakat ders gereği öncelikle sanki sağlanmış gibi düşünüp **parametrik testi(Pearson Korelasyon Katsayısı)** uygulayacağız, sonrasında **nonparametrik testine(Spearman Korelasyon Katsayısı)** geçeceğiz.

Korelasyon Katsayısı Hipotez Testi

Korelasyon Katsayısı:

In [10]:
```
df["tip"].corr(df["total_bill"])
```

Out[10]:
```
0.5766634471096374
```

Bu fonksiyon ön tanımlı olarak **"Pearson Korelasyon Katsayısı"** değerini hesaplar. **"Spearman Korelasyon Katsayısı"** için argüman olarak `method = "spearman"` şeklinde yazmalıyız.

In [9]:
```
df["tip"].corr(df["total_bill"], method = "spearman")
```

Out[9]:
```
0.593691939408997
```

Korelasyon katsayıları hesaplandı. Bu sonucu yorumlarsak bu değişkenlerin **pozitif yönlü orta şiddette bir ilişki** vardır. Fakat bunun **anlamlı olup olmadığını** bulmak için korelasyon için hipotez testi yapmalıyız.

Not: Burada tersten bir işlem gerçekleştirdik. Önce hipotezi kurup sonra katsayıyı hesaplamalıydık.

In [11]:
```
test_istatistigi, pvalue = st.pearsonr(df.tip, df.total_bill)
print('Test istatistiği = %.4f, p-değeri = %.4f' % (test_istatistigi, pvalue))
```

```
Test istatistiği = 0.5767, p-değeri = 0.0000
```

Sonuçtan görüldüğü üzere $p < alpha$ olduğundan H_0 reddedilir. Yani değişkenler arasında anlamlı bir ilişki vardır. Fakat bu yorum parametrik teste göre olduğu için ve varsayımlar sağlanmadığı için bu yorum geçersizdir.

Not: Buradaki test istatistiği aslında hesaplamış olduğumuz korelasyon katsayısıdır.

Nonparametrik Korelasyon Hipotez Testi

Parametrik test olan **Pearson Testi** için gerekli varsayımlar sağlanmadığında **Spearman Testi** veya **Kendall Tau Testi** yapılır.

In [16]:
```
st.spearmanr(df.tip, df.total_bill)
```

Out[16]:
```
SpearmanrResult(correlation=0.593691939408997, pvalue=1.2452285137560276e-24)
```

Formatını düzenleyelim:

In [14]:
```
test_istatistigi, pvalue = st.spearmanr(df.tip, df.total_bill)
print('Korelasyon Katsayısı = %.4f, p-değeri = %.4f' % (test_istatistigi, pvalue))
```

```
Korelasyon Katsayısı = 0.5937, p-değeri = 0.0000
```

Sonuçtan görüldüğü üzere $p < alpha$ olduğundan H_0 reddedilir yani **istatistiksel olarak anlamlı** bir ilişki vardır. Bu ilişki de **pozitif yönlü orta şiddette** şeklindedir.

Diğer bir test olan **Kendall Tau** testini de gerçekleştirelim. **Spearman Testi** de genel olarak iş gören bir testtir fakat garanti olması açısından bu test de incelenebilir.

In [17]:
```
test_istatistigi, pvalue = st.kendalltau(df.tip, df.total_bill)
print('Korelasyon Katsayısı = %.4f, p-değeri = %.4f' % (test_istatistigi, pvalue))
```

```
Korelasyon Katsayısı = 0.4401, p-değeri = 0.0000
```

Bu testten de görüldüğü üzere değişkenler arasında istatistiksel olarak anlamlı bir ilişki olduğu söylenebilir.

Bu örnek için varsayımlardan yola çıkarak sonucunu kabul edeceğimiz test **Spearman Testidir.**

Bir serinin daha sonuna geldik. Okuduğunuz için teşekkür ederim. :)

In [ ]:
```
```