İstatistik Nedir

İstatistik, veri toplama ve analiz etme uygulama ve çalışmasıdır.

İki ana dala ayrılır

- **Descriptive/Summary İstatistik** : Eldeki veriyi açıklar vey özetler
- **Inferential İstatistik :** Çıkarımsal istatistik, temsil edilen popülasyon hakkında sonuçlar çıkarmak için örneklemi kullanmayı içerir

İstatistik Neden Bu kadar Önemli?

Spor İstatistikleri 

Kişisel Finans


İstatistik ne yapabilir?

Pratikte soruları cevaplamak için

- Türkiye'de ortalama maaş nedir?
- Bir şirketin haftada kaç müşteri sorgusu alması muhtemeldir?

Toplum genelinde uygulamalar

- Otomobil veya uçak gibi daha güvenli ürünler geliştirmek
- Hükümetin nüfusunun ihtiyaçlarını anlamasına yardımcı olmak

COVID-19 aşıları gibi bilimsel atılımları doğrular

İstatistik hangi sorulara cevap verebilir?

- Birinin bir ürünü alma olasılığı ne kadardır? İnsanlar farklı bir ödeme sistemi kullanabilselerdi, satın alma olasılıkları daha mı yüksek olurdu?
- Otelinizde kaç kişi konaklayacak? Doluluk oranını nasıl optimize edersiniz?
- Nüfusun %95'ine uyabilmesi için kaç beden kot üretilmesi gerekiyor? Her bedenden aynı sayıda mı üretilmeli?
- Hangi reklam insanların bir ürünü satın almasını sağlamada daha etkilidir?(A/B Testi)

İstatistiğin Sınırlamaları

İstatistikler; geniş, açık sorular yerine, spesifik, ölçülebilir sorular gerektirir.

- Örneğin; istatistikler bize pop müziğin Türkiyede daha popüler olup olmadığını söyleyebilir.

Ancak, ilişkilerin neden var olduğunu bulmak için istatistikleri kullanamayız

- Örneğin; insanların neden farklı müzik türlerini sevdiğini veya kadınların neden erkeklerden fazla yaşadığını

İstatistik hangi sorulara cevap veremez?

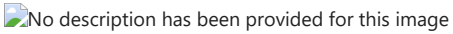
- Game of Thrones dizisi neden bu kadar popüler?: Herkese neden beğendikleri sorulabilir. Fakat yalan söyleyebilirler veya sebebini açıklamayabilirler
- Daha fazla şiddet sahnesi içeren filmlerin daha çok izleyici çekip çekmediği görülebilir. Fakat, Game of Thrones'daki şiddetin bunun sebebi olup olmadığına karar verilemez.

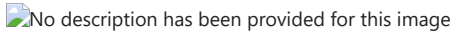
Veri Türleri

- Nümerik/Nicel
    1. Sürekli Veriler : Hisse Senedi Fiyatı, Günlük Rüzgar Hızı, Ürün kutusu ölçüleri ve ağırlığı
    2. Ayrık Veriler : Ürün incelemelerinin sayısı, Bir günde satılan bilet sayısı, Sınıftaki öğrenci sayısı

Nümerik Verileri Görselleştirme

Sayısal veriler arasındaki ilişkiyi görselleştirmenin en yaygın yolu dağılım grafiği kullanmaktır.





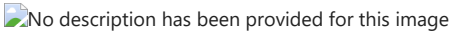
- Kategorik
    1. Nominal Veriler : Göz Rengi gibi sıralanmamış kategorileri tanımlayan veriler.
    2. Ordinal Veriler : Kategorilerin sıralandığı veriler. Likert ölçeği(Seviyorum, sevmiyorum..),

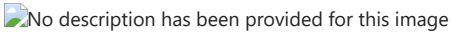
Kategorik Verileri Görselleştirme

Kategorik verileri ve bunların sayıları arasındaki ilişki görselleştirilebilir.

Descriptive/Summary İstatistik

 Dört arkadaşa işe nasıl gittikleri sorulduğunda; %50'si işe arabayla, %25'i otobüs %25'i ise bisikletle gittiğini belirtsin. Bunlar betimsel istatistiklerdir.





Inferential İstatistik

Bir popülasyon hakkında sonuç çıkarmak için bir örneklem kullanılır. Örneğin, 100 kişiye sosyal medya reklamlarını gördükten sonra kıyafet alıp almadıkları sorulabilir ve burdan elde edilen sonuç tüm insanların yüzde kaçının sosyal medya reklamı sonrası kıyafet aldığını anlamak için kullanılabilir.

Merkez Ölçümleri

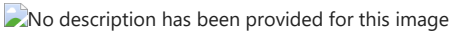
Merkez ölçümleri neden faydalıdır?

- Bir işyerinde aylık ortalama sipariş sayısının ne olduğu sorulabilir.
- Bir evin tipik maliyeti öğrenilebilir
- En yaygın saç rengi öğrenilebilir

Ortalama, tipik, en yaygın gibi termonolijilerin tümü merkez ölçümlerinin günlük yaşamda nasıl ifade edildiğine dair örneklerdir.

Suç Verisi

Bu veri setindeki verileri histogram yolu ile görselleştirelim. Histogram, veri noktalarını alır ve bunları bölmelere veya değer aralıklarına ayırır.

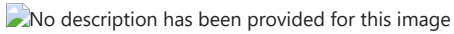


Yukarıda her biri ayrı yüksekliğe sahip sekiz kutulu araç suçlarının histogramı verilmiştir.Ortadaki zirve dokuz Londra ilçesinde son iki yılda 6000 ila 7300 arasında araç suçu işlendiğini göstermektedir.

Histogram sayısal verileri özetlemenin çok iyi bir yoludur ancak tanımlayıcı istatistikler de kullanılabilir.

Londra'da tipik araç suçu miktarı nedir?

Bu sorunun cevaplanabilmesi için verinin tipik veya merkez değerinin ne olduğunun belirlenmesi gerekir. Bunun histogram gibi veri görselleştirme yoluyla belirlenmesi zordur.



Merkezi hesaplamanın üç yolu bulunmaktadır.

- Ortalama
- Medyan
- Mod

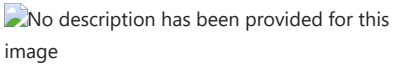
In [1]:
```
import pandas as pd
df_animal = pd.read_csv("data/msleep.csv")
df_animal
```

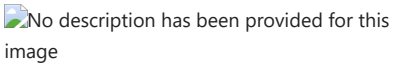
Out[1]:

	name	genus	vore	order	conservation	sleep_total	sleep_rem	sle
0	Cheetah	Acinonyx	carni	Carnivora	lc	12.1	NaN	
1	Owl monkey	Aotus	omni	Primates	NaN	17.0	1.8	
2	Mountain beaver	Aplodontia	herbi	Rodentia	nt	14.4	2.4	
3	Greater short-tailed shrew	Blarina	omni	Soricomorpha	lc	14.9	2.3	
4	Cow	Bos	herbi	Artiodactyla	domesticated	4.0	0.7	
...	...	...	...	...	...	...	...	
78	Tree shrew	Tupaia	omni	Scandentia	NaN	8.9	2.6	
79	Bottle-nosed dolphin	Tursiops	carni	Cetacea	NaN	5.2	NaN	
80	Genet	Genetta	carni	Carnivora	NaN	6.3	1.3	
81	Arctic fox	Vulpes	carni	Carnivora	NaN	12.5	NaN	
82	Red fox	Vulpes	carni	Carnivora	NaN	9.8	2.4	

83 rows × 11 columns

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.histplot(df_animal.sleep_total)
plt.title("Distribution of Sleep Times of Various Mammals")
plt.xlabel("Hours of Sleep")
```


No description has been provided for this image


No description has been provided for this image

Ortalama

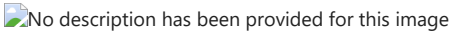
Verinin merkezini tanımlamanın en yaygın yollarından biridir.

Burada hırsızlığın en büyük ortalamaya sahip olduğu görülmektedir.

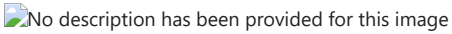
In [ ]:
```
import numpy as np

# np.mean(df_animal.sleep_total)
df_animal['sleep_total'].mean()
```

Medyan

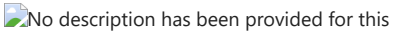
Verinin merkezini tanımlamanın bir diğer ölçüsü medyandır. Verilerin orta değeridir



Yukarıda hırsızlık için gösterildiği gibi veriler küçükten büyüğe sıralanır. Verilerin %50'si medyandan büyük %50'si medyandan küçük olmalıdır.



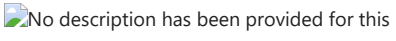
Çift sayıda veri olduğu için (32) ortaya en yakın iki değer alınır.



In [ ]:
```
import numpy as np
np.median(df_animal.sleep_total), df_animal.sleep_total.median()
```

Mod

Verideki en çok tekrar eden değerdir.



En sık görülen değer hırsızlıktır. Kategorik verilerin beklenen değeri aranıyorsa mod en uygun ölçüdür.

In [ ]:
```
df_animal.sleep_total.value_counts()
df_animal.vore.value_counts(dropna=False)
```

In [ ]:
```
import statistics as stat

stat.mode(df_animal.vore), df_animal.vore.mode()
```

In [2]:
```
df_animal[df_animal.vore == "insecti"]
```

Out[2]:

	name	genus	vore	order	conservation	sleep_total	sleep_rem
21	Big brown bat	Eptesicus	insecti	Chiroptera	lc	19.7	3.9
42	Little brown bat	Myotis	insecti	Chiroptera	NaN	19.9	2.0
61	Giant armadillo	Priodontes	insecti	Cingulata	en	18.1	6.1
66	Eastern american mole	Scalopus	insecti	Soricomorpha	lc	8.4	2.1
74	Short-nosed echidna	Tachyglossus	insecti	Monotremata	NaN	8.6	NaN

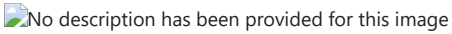
In [3]:
```
df_animal[df_animal.vore == "insecti"]["sleep_rem"].agg(["mean", "median"])
```

Out[3]:
```
mean      3.525
median    3.000
Name: sleep_rem, dtype: float64
```

In [ ]:
```
# add outlier
df_animal.loc[len(df_animal.index)] = ["New Insect", "", "insecti", "", "", 0.0,
df_animal
```

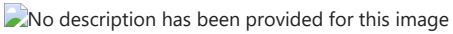
In [ ]:
```
df_animal[df_animal.vore == "insecti"]["sleep_rem"].agg(["mean", "median"])
# mean 3.525 -> 2.82
# median 3 -> 2.1

# Ortalama uç değerlere karşı daha hassastır. Bu nedenle bu tip durumlarda medya
# Yine çarpık verilerde de medyan kullanmak daha iyidir.
```

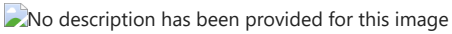
Hangi merkez ölçüsü kullanılmalıdır?



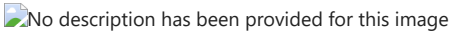
Yukarıdaki histogram incelendiğinde simetrik olduğu görülmektedir. Ortada zirve yapmakta ve her iki tarafa doğru azalmaktadır.

Veriler simetrik olduğunda ortalama ve medaynın her ikisin de kullanımı uygundur





Görüldüğü veriler simetrik değildir. Bir değer diğerinden önemli ölçüde farklı olduğunda bu değere aykırı değer (outlier) denir.



Bu aykırı değer ortalamayı kendisine doğru çekerken, medyan daha az etkilenir. Bunun nedeni, ortalama hesaplamanın tüm değerleri toplamayı gerektirmesidir. Daha büyük değerler sonucu etkiler, medyan ise sadece ortadaki değer bakar. Bu nedenler veriler simetrik olmadığında medyan kullanmak en iyi seçimdir.

Yayılım Ölçüleri

Özet istatistiğinin başka bir konusudur. Yayılım, veri noktalarının birbirinden ne kadar uzak olduğunu açıklar.

İlk histogram Londra ilçeleri genelinde araç suçlarının histogramını göstermektedir. İkinic histogram ise, Londra ilçeleri genelinde hırsızlık suçlarının histogramını göstermektedir. İkinci histogramda yayılımın dar olduğunu görmekteyiz.

Yayılma önemlidir çünkü, bize verilerimizde ne kadar çeşitlilik olabileceğini gösterir. Örneğin, kazakların maliyeti 30 TL ise ancak 10-200 TL aralığında başka yerlerden alınabiliyorsa, 30 TL'den bir tane bulma olasılığımız nedir? Kazak fiyatları 20-50 TL arasında olursa bu olasılık değişir mi?

In [ ]:
```
range_sleep_total = df_animal['sleep_total'].max() - \
df_animal['sleep_total'].min()
range_sleep_total
```

Hangi yayılım ölçüleri vardır

- **Range** range = maximum - minimum
   No description has been provided for this image range(Burglaries) = 5183 - 1432 range(Burglaries) = 3751 Thames'de son iki yılda Tower Hamlets'ten 3751 daha az hırsızlık olayı gerçekleşti.
- **Varyans** Varyans, her bir veri noktasının ortalamaya olan ortalama uzaklığını hesaplar. Varyans ne kadar büyükse veriler o kadar dağılmış demektir.
   No description has been provided for this image Yukarıdaki grafik, ortadaki dikey çizgi ortalamayı göstermek üzere, Londra'sa ilçe başına suçların dağılımını göstermektedir. Bir ilçenin ortalamadan çok uzakta olduğu görülmektedir. Varyansı hesaplamak için, her bir veri noktasının ortalama değere olan mesafesi hesaplanır.
   No description has been provided for this image Bu her bir veri noktası için tekrarlanır. Eğer hepsi toplanırsa toplamın sıfır olduğu görülür. Bu nedenle mesafelerin kareleri toplanır ve toplamı alınır.
   No description has been provided for this image variance(total crime) = 7.509.750.824 / 32(ilçe sayısı) = 234.769.713
- **Standart Sapma** Varyansı anlamak zordur. Bu nedenle standart sapma kullanılır Standart sapma varyansın karekökü alınarak hesaplanır. standart sapma = √234.769.713 = 15.319, 26 Standart sapma sıfıra ne kadar yakınsa verilerin ortalama etrafında o kadar yakın kümlendiği anlaşılır.
   No description has been provided for this image Yukarıda da görüldüğü gibi, ortalamadan bir veya iki standart sapma uzaklık işaretlenerek verilerin ne kadar dağınık olduğu görülebilir.
- **Quartiles** Yayılım, verileri dört eşit parçaya bölmenin bir yolu olan çeyrekler kullanılarak da ölçülebilir.  No description has been provided for this image Yukarıdaki tabloda Londra'daki çeşitli suçlar ve bunların minimum değerleri bulunmaktadır (%25, %50, %75, %100) Her çeyrek için değer, o sayıdan küçük veya ona eşit olan değerlerin yüzdesini temsil eder. Londra ilçelerinin %75'inde son iki yılda 4392'den az hırsızlık olduğu görülmektedir. İkinci çeyrek ortadaki değerdir ve medyana eşittir. **Boxplots** Bir boxplot kullanılarak çeyrekler görselleştirilebilir.
   No description has been provided for this image Aykırı değerler 4000'in üzerindeki nokta gibi yatay çizgilerin ötesinde gösterilir.
- **Interquartile Range (IQR)** Çeyrekler arası aralık olarak da tanımlanır. IQR = Q3 - Q1
   No description has been provided for this image IQR = 1976.5 - 895.75 IQR = 1080.75 IQR, aşırı koşullardan daha az etkilenir. Standart sapmadan daha değerlidir.

In [4]:
```
import numpy as np
np.var(df_animal.sleep_total, ddof=1)
```

Out[4]:
```
19.805677343520422
```

In [5]:
```
np.sqrt(np.var(df_animal.sleep_total, ddof=1))
```

Out[5]:
```
4.4503569905705795
```

In [6]:
```
np.std(df_animal.sleep_total, ddof=1)
```

Out[6]:
```
4.4503569905705795
```

In [ ]:
```
# Mean Absolute Deviation

dists = df_animal.sleep_total - \
np.mean(df_animal.sleep_total)
np.mean(np.abs(dists))
```

- Standart sapmada uzaklıkların karesi alınır, bu nedenle daha uzun mesafeler daha uzun cezalandırılır.
- Mean Absolute Deviation'da, her uzaklık eşit ceza alır.
- Biri diğerinden daha iyi değildir ama SD MAD'den daha yaygındır.

In [2]:
```
import numpy as np
np.quantile(df_animal.sleep_total, 0.5)
```

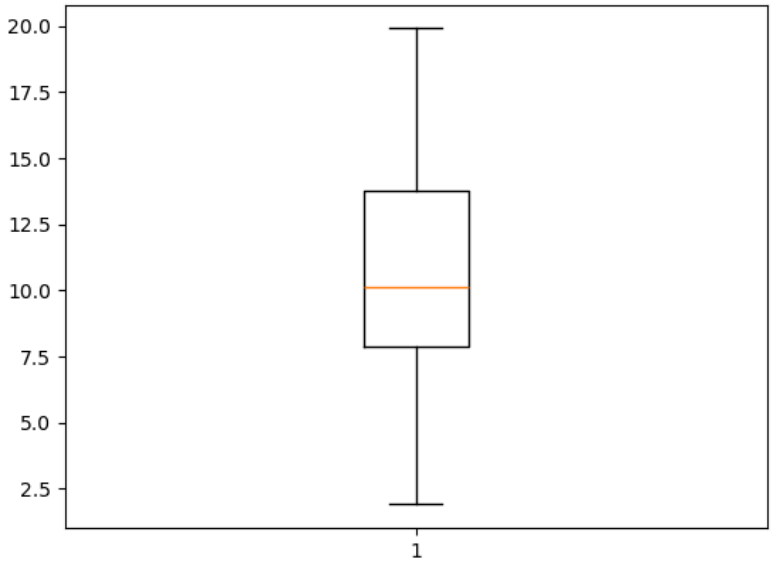
Out[2]:
```
10.1
```

In [3]:
```
np.quantile(df_animal.sleep_total, [0, 0.25, 0.50, 0.75, 1])
```

Out[3]:
```
array([ 1.9 ,  7.85, 10.1 , 13.75, 19.9 ])
```

In [4]:
```
import  matplotlib.pyplot as plt
plt.boxplot(df_animal.sleep_total)
```

Out[4]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7a1e9f46c110>,
  <matplotlib.lines.Line2D at 0x7a1e9f46c890>],
 'caps': [<matplotlib.lines.Line2D at 0x7a1e9f46cbc0>,
  <matplotlib.lines.Line2D at 0x7a1e9f46cec0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7a1e9f46c2c0>],
 'medians': [<matplotlib.lines.Line2D at 0x7a1e9f46d190>],
 'fliers': [<matplotlib.lines.Line2D at 0x7a1e9f46d490>],
 'means': []}
```



In [5]:
```
np.quantile(df_animal.sleep_total, \
            [0, 0.2, 0.4, 0.6, 0.8, 1])
```

Out[5]:
```
array([ 1.9 ,  6.24,  9.48, 11.14, 14.4 , 19.9 ])
```

In [7]:
```
np.linspace(0, 1, 5)
```

Out[7]:
```
array([0.  , 0.25, 0.5 , 0.75, 1.  ])
```

In [6]:
```
# np.linspace(start, stop, num)

np.quantile(df_animal.sleep_total, np.linspace(0, 1, 5))
```

Out[6]:
```
array([ 1.9 ,  7.85, 10.1 , 13.75, 19.9 ])
```

In [7]:
```
np.quantile(df_animal.sleep_total, 0.75) - \
np.quantile(df_animal.sleep_total, 0.25)
```

Out[7]:
```
5.9
```

In [8]:
```
from scipy.stats import iqr
iqr(df_animal.sleep_total)
```

Out[8]:
```
5.9
```

Outliers (Aykırı Değerler)

Diğerlerinden önemli ölçüde farklı olan veri noktalarıdır. Peki öenmli bir fark olduğuna nasıl karar verilir?

Q1 - 1.5IQR < data < Q3 + 1.5IQR

In [9]:
```
# Finding Outliers

from scipy.stats import iqr

iqr = iqr(df_animal.bodywt)

lower_thereshold = np.quantile(df_animal.bodywt, 0.25) \
- 1.5 * iqr
upper_thereshold = np.quantile(df_animal.bodywt, 0.75) \
+ 1.5 * iqr

df_animal[(df_animal.bodywt < lower_thereshold) | (df_animal.bodywt > upper_ther
```

Out[9]:

	name	genus	vore	order	conservation	sleep_total	sleep_rem	s
4	Cow	Bos	herbi	Artiodactyla	domesticated	4.0	0.7	
20	Asian elephant	Elephas	herbi	Proboscidea	en	3.9	NaN	
22	Horse	Equus	herbi	Perissodactyla	domesticated	2.9	0.6	
23	Donkey	Equus	herbi	Perissodactyla	domesticated	3.1	0.4	
29	Giraffe	Giraffa	herbi	Artiodactyla	cd	1.9	0.4	
30	Pilot whale	Globicephalus	carni	Cetacea	cd	2.7	0.1	
35	African elephant	Loxodonta	herbi	Proboscidea	vu	3.3	NaN	
50	Tiger	Panthera	carni	Carnivora	en	15.8	NaN	
52	Lion	Panthera	carni	Carnivora	vu	13.5	NaN	
76	Brazilian tapir	Tapirus	herbi	Perissodactyla	vu	4.4	1.0	
79	Bottle-nosed dolphin	Tursiops	carni	Cetacea	NaN	5.2	NaN	

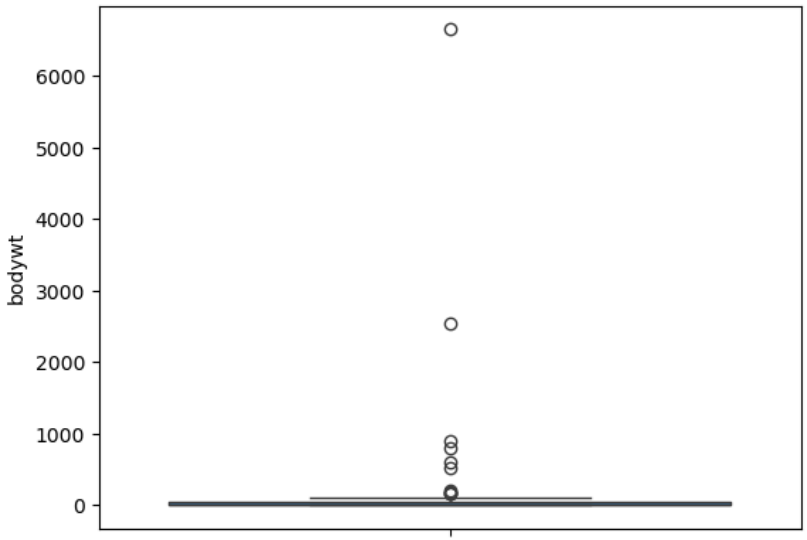
In [11]:
```
sp_lower = np.mean(df_animal.bodywt) - 3 * np.std(df_animal.bodywt, ddof=1)
sp_upper = np.mean(df_animal.bodywt) + 3 * np.std(df_animal.bodywt, ddof=1)

df_animal[(df_animal.bodywt < sp_lower) | (df_animal.bodywt > sp_upper)]
```

Out[11]:

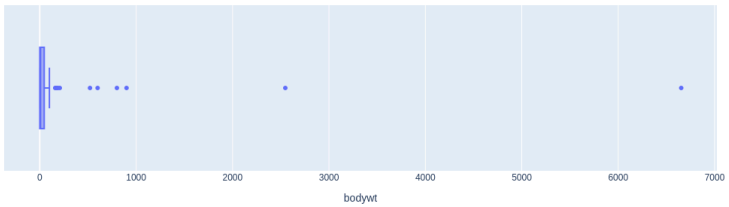
	name	genus	vore	order	conservation	sleep_total	sleep_rem	sleep_
20	Asian elephant	Elephas	herbi	Proboscidea	en	3.9	NaN	
35	African elephant	Loxodonta	herbi	Proboscidea	vu	3.3	NaN	

In [12]:
```
import seaborn as sns

sns.boxplot(data=df_animal, y='bodywt')
```

Out[12]:
```
<Axes: ylabel='bodywt'>
```



In [13]:
```
import plotly.express as px

px.box(df_animal, x='bodywt')
```



In [14]:
```
df_animal.describe().T
```

Out[14]:

	count	mean	std	min	25%	50%	75%
sleep_total	83.0	10.433735	4.450357	1.900000	7.850000	10.100000	13.750000
sleep_rem	61.0	1.875410	1.298288	0.100000	0.900000	1.500000	2.400000
sleep_cycle	32.0	0.439583	0.358680	0.116667	0.183333	0.333333	0.579167
awake	83.0	13.567470	4.452085	4.100000	10.250000	13.900000	16.150000
brainwt	56.0	0.281581	0.976414	0.000140	0.002900	0.012400	0.125500
bodywt	83.0	166.136349	786.839732	0.005000	0.174000	1.670000	41.750000

In [15]:
```
df_animal.bodywt.describe()
```

Out[15]:
```
count      83.000000
mean      166.136349
std       786.839732
min         0.005000
25%         0.174000
50%         1.670000
75%        41.750000
max      6654.000000
Name: bodywt, dtype: float64
```

Şans Nedir

İnsanlar genellikle şanstan bahsederler. Örneğin; bir satışın bitme, yarın yağmur yağma ve oyunu kazanma gibi.

Bir olayın sonucunun olma olasılığını tahmin edebilmek, birçok yönden faydalı olabilmektedir.

Peki, şansı nasıl ölçebiliriz

Bir olayın gerçekleşme olasılığı nedir?

P(Olay) = olayın gerçekleşebileceği yollar / Olası çıktıların toplam sayısı

Örneğin; bir yazı tura oyununda tura gelme olasılığı;

P(Tura) = 1 / 2 = 0.5

Olasılık her zaman 0 ile 100 arasındadır.


No description has been provided for this image

Bir başka örnek verirsek; Potansiyel bir müşteri ile yaklaşan bir toplantınız var ve siz de katılması için satış ekibinden birini göndermek istiyorsunuz. Her kişinin adını bir kutuya koyup, toplantıya kimin katılacağını belirlemek için rastgele birini seçeceksiniz.


No description has been provided for this image

Brian'ı seçme olasılığı 1/4 = 0.25'tir.

Peki, farklı zamanlarda iki toplantı yaparsak ne olur?

Her toplantı için dört ekip üyesinden birini rastgele seçebiliriz. İlk toplantı için seçilen kişinin, ikinci toplantı için seçilme şansını etkilenmez.

Örenğin ilk toplantı için Brain seçilmişse, Brain'ın öğleden sonraki toplantı için seçilme şansı yine %25'tir.

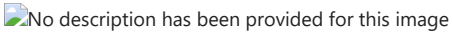
Örnek yerine geri yerleştirildiğinden ve tekrar seçilebildiğinden buna **değiştirmeli örnekleme (sampling with replacement)** denir.

Bu bağımsız olasılığın bir örneğidir.

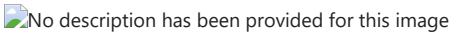
Bağımsız Olasılık

İkinci olayın olasılığı ilk olayın sonucuna bağlı olarak değişmiyorsa iki olay bağımsızdır.

Online Perakende Satış Veri Seti



Bu veri setini kullanarak bir sonraki siparişin mücevher olam olasılığı bulunmak istenirse. Bu durumda, tüm ürünleri sipariş tipine göre gruplayarak her bir ürün için verilen toplam sipariş sayısını bulabiliriz.



P(Mücevher) = Mücevher sipariş sayısı / Toplam Sipariş Sayısı

P(Mücevher) = 210 / 1767 = % 11.88

In [2]:
```
import pandas as pd

df_sales = pd.read_csv("data/amir_deals.csv")
df_sales_users = df_sales.groupby("num_users")["amount"].\
agg(sum="sum")

# select num_users, sum(amount) from amir_deals group by num_users

df_sales_users
```

Out[2]:

num_users	sum
1	13624.50
2	40732.68
3	24858.82
4	3880.07
5	12428.48
...	...
92	4509.96
94	4171.76
96	8180.81
98	5992.86
99	16750.45

79 rows × 1 columns

In [3]:
```
df_sales_users.sample()
```

Out[3]:

num_users	sum
92	4509.96

In [4]:
```
df_sales_users.sample()
```

Out[4]:

num_users	sum
82	13927.5

In [5]:
```
import numpy as np

np.random.seed(42)
```

In [6]:
```
df_sales_users.sample()
```

Out[6]:

num_users	sum
33	7077.48

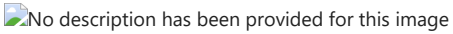
In [7]:
```
np.random.seed(42)
df_sales_users.sample()
```

Out[7]:

num_users	sum
33	7077.48

Koşullu Olasılık (Conditional Probability)

Bir olayın sonucunun başka bir olayı etkilemesi durumudur.



Önceki örnekten devaö edelim. Brain'ın ismi seçildi ve adı artık kutusa yer almıyor. Aynı anda başka bir toplantımız var ve başka bir temsilci seçmek zorundayız. Brain olmadığı içib geri klan üç kişiden birini seçmek zorundasınız.

Daha önce çıkardığımız kişiyi tekrar yerine koymadığımız için buna **değiştirmeden örnekleme (sampling without replacement)** denir.

Bu sefer de Clair seçilsin. Bu drumda olasılık %33.3'tür. Bu ilk olayın sonucunun ikinci olayın olasılığını değiştirdiği bağıölı olaylara bir örnektir.

Bağımlı Olasılık

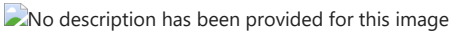
İlk olayın sonucunun ikinci olayın sonucunu etkilediği durumdur.

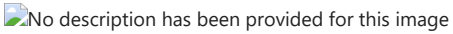
Örneğim ilk seçimde Claire çekilseydi, ikinci çekilişte Claire'in seçilme olasılığı %0'dır. Eğer başka biri ilk seçilirse Claire'in ikinci seçilme olasılığı %33.3'tür.

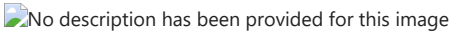
Koşullu olasılık, bağımlı olayların olasılığını hesaplamak için kullanılır.

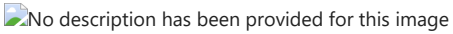
- Bir olayın olasılığı diğerinin sonucuna bağlıdır. Örneğin, bir önceki tren göz önüne alındığında, bir trenin zamanında varma olasılığı.

Venn Diagram









In [10]:
```
df_sales_users.sample(5, replace=True)
```

Out[10]:

num_users	sum
73	9299.91
11	3240.63
8	20224.18
37	4409.89
37	4409.89

Ayrık Dağılımlar

Standart altı yüzlü bir zarın atıldığını düşünelim. Alto olası sonuç vardır ve her birinin gerçekleşme şansı altıda birdir.

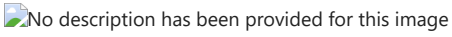
Bu daha önceki senaryoya benzemektedir ve burada sayıların yerini isimler almaktadır. Tıpkı zarın atılması gibi, her sonucun veya ismin seçilme şansı eşittir.ir

Olasılık Dağılımı

Bir senaryodaki her olası sonucun olasılığını açıklar.

Bir dağılımın ortalaması olan beklenen değeri de bulunabilir.

Bu, her değeri olasılığıyla (bu durumda altıda biri) çarpıp toplayarak hesaplanır.

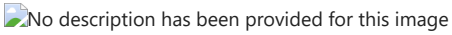


Olasılık Dağılımları Neden Önemlidir?

- Riski ölçmeye ve karar alma sürecini bilgilendirmeyi sağlar.
- Hipotez testlerinde sonuçların şans eseri çıkıp çıkmadığını anlamak için
-

Olasılık Dağılımlarının Görselleştirilmesi

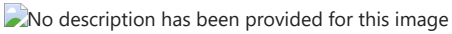
Histogram kullanarak olasılık dağılımları görselleştirilebilir. Burada her çubuk bir sonucu temsil eder ve her çubuğun yüksekliği bu sonucun olasılığını temsil eder.



Olasılık = Alan

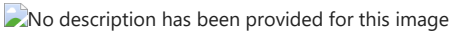
Olasılık dağılımının alanlarını bularak farklı sonuçların olasılıkları hesaplanabilir.

Örneğin, atılan zarın ikiye eşit veya ikiden düşük olma olasılığı nedir? P(zar atışı) ≤ 2 = ?



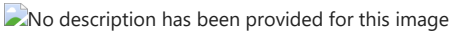
P(zar atışı) ≤ 2 = 1/3

Aşağıdaki şekilde de görüldüğü gibi zardaki iki kısmının üçe dönüştüğünü varsayalım.

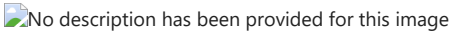


Bu durumda, artık zarın iki gelme şansı 0, 3 gelme şansı %33'dür.

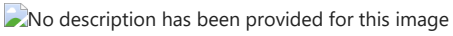
Böyle bir zarın beklenen değeri ise:



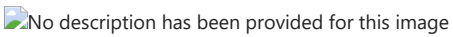
Bu yeni olasılıklara ait dağılım grafiği çizildiğinde çubukların yükseklikleri eşit olmayacaktır.



Böyle bir durumda P(zar atışı) ≤ 2 = ?



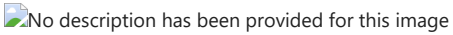
Şu ana kadar gördüğümüz olasılık dağılımları ayrık sonuçları olan durumları temsil ettikleri için ayrıktır. Bu nedenle sayım ve aralık verilerini temsil ederler. Örneğin, zar durumunda noktaları sayıyoruz. 1.5 veya 4.3 atılamaz.



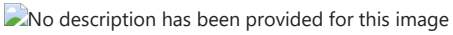
Adil bir zar kullanılması durumu gibi, tüm sonuçların aynı olasılığa sahip olaması durumuna discrete uniform distrubition adı verilir.

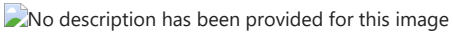
Ayrık bir dağılımdan örneklem alma

Adil bir zar atışında potansityel sonuçlar aşağıdaki gibi olsun.

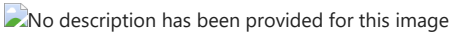


Aynı zar 10 kez atılırsa, aynı sonuç birden fazla kez alınabileceği için yerine koyarak örnekleme yapılmış olur.



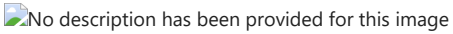


Örnekleme rastgele olduğundan, her sayının gelme olasılığı aynı olmasına rağmen, farklı sayıda elde edilmiştir.



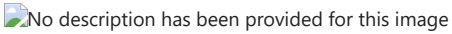
Burada örneğin ortalaması 3'tür ve beklenen değer olan 3.5'e pek yakın değildir.

Zarı 100 kere atarsak dağılım:



Görüldüğü gibi gelen sayılar biraz daha eşit görünüyor ve ortalama 3.5'e biraz daha yakındır.

1000 defa atılırsa sonuç şuan benzer.



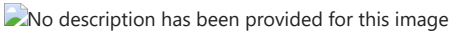
Burada teorik olasılık dağılımı ve 3.5 ile daha yakın eşleşiyor. Buna büyük sayılar kanunu (Law of Large Numbers) adı verilir.

Örnek boyutu artılırsa ortalanma teorik ortalamaya yaklaşacaktır.

Sürekli Dağılımlar

Şu durumlarda sürekli dağılımlar kullanılır

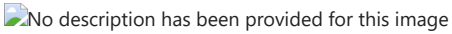
Otobüs bekliyorsunuz



Belediye otobüsü her 12 dakikada bir geliyor. yani rastgele bir saatte gelirseniz bir süre bekleyebilirsiniz. Otobüs tam durağa gelirken gelinirse sıfır dakikadan, otobüs kalkarken gelinirse 12 dakikaya kadar beklenebilir.

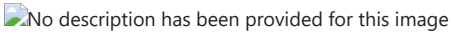
Bu olay bir olasılık dağılımı ile modellenebilir. Durakta, beklenebilecek sonsuz sayıda dakika vardır (5 dk., 1.5 dk, 1.53 dk. vb.). Bu nedenle sayım ve aralık verileriyle yapıldığı gibi bireysel bloklar oluşturulamaz.

Bunun yerine olasılığı temsil etmek için sürekli bir çizgi kullanılır. 0'dan 12'ye kadar herhangi bir süre bekleme olasılığı olduğu aynı olduğundan çizgi düzdür. Buna sürekli düzgün dağılım adı verilir (Continuous Uniform Disttibutions).

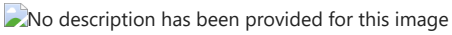


Olasılık = Alan

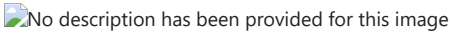
P(4 ≤ bekleme zamanı ≤ 7) = ?



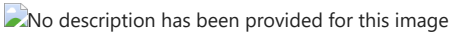
P(bekleme zamanı ≤ 7) = ?



P(0 ≤ bekleme zamanı ≤ 12) = ?

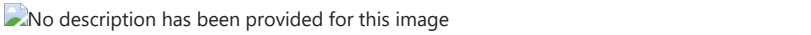


P(bekleme zamanı ≥ 7) = ?



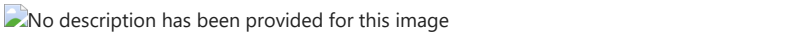
Bimodal Distributions

Sürekli dağılımlar, bazı değerlerin diğerlerinden daha yüksek olasılığa sahip olduğu tekdüze olmayan biçimler alabilir.

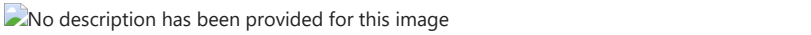


Örneğin yukarıda en sık ortaya çıkan iki değrin olduğu bir dağılım görülmektedir. İki modu olduğundan, iki modlu dağılım (bimodal distribution) olarak adlandırılır. İki modlu dağılama verilebilecek örneklerden bir tanesi, farklı özelliklere sahip kitap fiyatlarıdır. Kitabın ciltli veya ciltsiz olmasına göre ortaya çıkan değerler.

Normal Dağılım



Tansiyon ve emeklilik yaşı gibi verilerde bu dağılımı gözlemlemek çok yaygındır.



In [34]:
```
'''
scipy.stats.uniform.cdf(x, loc=0, scale=1)

x = Değeri x'e kadar olan olasılığı hesaplamak istediğiniz noktadır. Yani, P(X≤x

loc = Dağılımın başlangıç (lokasyon) parametresidir. Yani, uniform dağılımın baş
Default (varsayılan) değeri 0'dır, ancak farklı bir aralıkta bir uniform dağılım
Örnek: Eğer loc=2loc=2 ise, dağılım 2'den başlar.

scale =  Dağılımın uzunluğunu (aralığını) belirleyen bir parametredir.
scale, dağılımın genişliğini gösterir ve bu aralık [loc,loc+scale] aralığına yay
Varsayılan değer 1'dir, yani loc'dan 1 birim sağa doğru genişler.
Örnek: Eğer scale=3 ise, uniform dağılım loc'dan loc+3'e kadar uzanır.

'''

# P(wait_time ≤ 7)
from scipy.stats import uniform
uniform.cdf(7, 0, 12)
```

Out[34]: 0.5833333333333334

In [35]:
```
# P(wait_time ≥ 7) = 1 - P(wait_time ≤ 7)
1 - uniform.cdf(7, 0, 12)
```

Out[35]: 0.41666666666666663

In [36]:
```
# P(4 ≤ wait_time ≤ 7)
# P(4 ≤ wait_time ≤ 7) = P(wait_time ≤ 7) - P(wait_time ≤ 4)

uniform.cdf(7, 0, 12) - uniform.cdf(4, 0, 12)
```

Out[36]: 0.25000000000000006

In [37]:
```
# P(0 ≤ wait_time ≤ 12)

uniform.cdf(12, 0, 12)
```

Out[37]: 1.0

Uniform Dağılıma Göre Rastgele Sayılar Üretme

In [38]:
```
uniform.rvs(0, 5, size=10)
```

Out[38]:
```
array([3.85635173, 0.37022326, 1.79232864, 0.5793453 , 4.31551713,
       3.11649063, 1.65449012, 0.31779175, 1.55491161, 1.62591661])
```

scipy.stats.uniform.rvs(loc=0, scale=1, size=1, random_state=None)

loc (default = 0):

```
Açıklama: Bu parametre, dağılımın başlangıç noktasını
(lokasyonunu) belirler. Dağılım, bu değerden itibaren başlar.
Varsayılan değer 0'dır.
Örnek: Eğer loc = 5 verilirse, uniform dağılım 5'ten başlar ve
rastgele sayılar bu aralıktan üretilir.
Dağılım Aralığı: [loc,loc+scale][loc,loc+scale] aralığında
değerler üretir.
```

scale (default = 1):

```
Açıklama: Bu parametre, dağılımın genişliğini belirler. Uniform
dağılım bu aralık içinde rastgele değerler üretir. Varsayılan
değer 1'dir.
Örnek: Eğer scale = 10 verilirse, dağılım locloc'dan
loc+10loc+10'a kadar uzanır. Yani, üretilen rastgele değerler bu
aralıktan gelir.
Dağılım Aralığı: [loc,loc+scale][loc,loc+scale].
```

size (default = 1):

```
Açıklama: Üretilmesini istediğiniz rastgele sayıların sayısını
belirtir. Eğer bir sayı verirseniz, o kadar rastgele değer
üretilir. Bu parametre bir tamsayı ya da bir tuple olabilir.
Örnek:
    size=1 ise, tek bir rastgele sayı üretilir.
    size=5 ise, beş tane rastgele sayı üretilir.
    size=(3, 4) ise, 3x4 boyutunda bir matris şeklinde rastgele
sayılar üretilir.
```

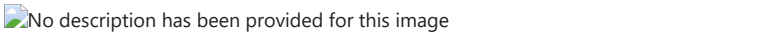
random_state (default = None):

```
Açıklama: Rastgele sayı üretiminde kullanılan seed (tohum)
değerini belirler. Eğer belirli bir seed verilirse, aynı
sonuçların tekrar üretilebilmesi sağlanır. Eğer
random_state=None verilirse, her çalıştırmada farklı rastgele
sayılar üretilir.
Örnek:
```

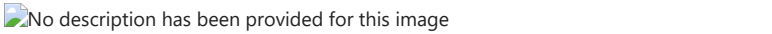
```
    random_state=42 verilirse, aynı koşullar altında aynı
rastgele sayılar üretilir.
    Eğer bu parametreye herhangi bir değer verilmezse (None),
sonuçlar her çalıştırmada farklı olur.
```

Binomial (Binom) Distribution

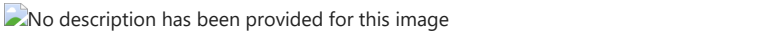
Binom dağılımı, bir dizi bağımsız denemedeki başarı sayısının olasılığını tanımlar.



Bir yazı tura oyunu oyanayalım. İki olasığığa sahiptir. Her birinin olasılığı %50'dir. Bu iki olası değerin oluşabileceği ikili duruma örnektir.

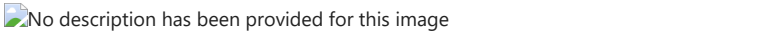


Bu sonuçlar yukarıdaki gibi farklı şekillerde ifade edilebilir.



Aynı madeni parayı birden fazla kez havaya atıp sonuçları kaydedebiliriz, örneğin burada yazı gelirse 1, tura gelirse 0 olarak gösterilir.

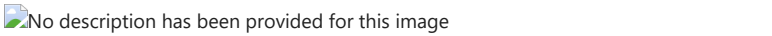
- Binom dağılımı, bir dizi bağımsız olaydaki başarı sayısının olasılığını tanımlar. Örneğin, bir dizi yazı-tura atışında belirli sayıda tura gelme olasılığını söyleyebilir.
- Sayılabilir bir sonuçla çalışıldığı için, ayrık bir dağılımdır.
- Binom dağılımındaki iki parametre n ve p olarak tanımlanabilir. n : gerçekleştirilen etkinlik sayısı p : başarı olasılığı



Yukarıda 10 atış için dağılımın nasıl görünmektedir. Burada en yüksek olasılık beş tura gelme, sıfır veya 10 tura gelme şansı ise çok daha düşüktür.

10 atışta 7 veya daha az tura gelme olasılığı:

10 atışta 8 veya daha fazla tura gelme olasılığı:

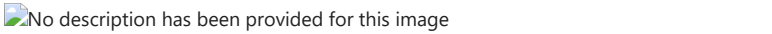


Expected Value = n * p

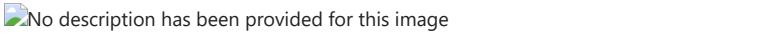
Yukarıdaki örnek için;

Expected Value = 10 * 0.5 = 5

Binom dağılımının uygulanabilmesi için her olayın bağımsız olması yani bir olayın sonucunun bir sonrakini etkilememesi gerekir.



Örneğin, yukarıdaki kartlardan rastgele seçim yapılıyorsa, sıfır veya bir seçme şansı %50 - %50'dir. Olasılıkların önceki bir olayın sonucuna göre değişmesi durumunda binom dağılımı uygulanmaz.



Ancak, yukarıda oldupu gibi yerine koymadan seçim yapılıyorsa, olasılıklar ilk olayın sonucuna bağlı olarak farklıdır. Yani burada, binom dağılımı uygulanamaz.

Binom dağılımı bağımsız olayların ikili sonuçlar ürettiği ve her sonuç için eşit olasılık gerektirmeyen senaryolarda kullanılabilir. Örnek olarak, bir ilacın etkili olup olmadığı durumu gösterilebilir.

from scipy.stats import binom

binom.rvs(n, p, loc=0, size=1, random_state=None)

n:

```
    Açıklama: Bu parametre, deneme sayısını (yani binom dağılımında
    yapılacak toplam deney sayısını) ifade eder.
    Örnek: Eğer n = 10 verilirse, 10 denemeden oluşan bir binom
    dağılımı üzerinde rastgele sayılar üretilir. Her bir deneme
    bağımsızdır ve p olasılığına göre başarı veya başarısızlıkla
    sonuçlanır.
```

p:

```
    Açıklama: Her bir denemenin başarı olasılığını ifade eder. Bu
    değer 0 ile 1 arasında olmalıdır.
    Örnek: Eğer p = 0.5 ise, her bir denemenin başarı olasılığı
    %50'dir. Yani, bir madeni paranın yazı-tura atışında yazı gelme
    olasılığı gibi düşünülebilir.
```

loc (default = 0):

```
    Açıklama: Sonuçta elde edilen rastgele sayılara bir kayma
    (offset) ekleyen parametredir. Varsayılan değeri 0'dır. Bu
    parametre, dağılımdan üretilen değerlere toplamsal bir kayma
    ekleyerek sonuçları değiştirir.
    Örnek: Eğer loc = 2 verilirse, üretilen her rastgele sayı
    üzerine 2 eklenir.
```

size (default = 1):

```
    Açıklama: Üretilmesini istediğiniz rastgele sayıların miktarını
    belirtir. Bu parametre, kaç tane binom dağılımına göre değer
    üretilmesi gerektiğini gösterir.
    Örnek:
```

```
    size=1 ise, tek bir rastgele sayı üretir.
    size=5 ise, beş tane rastgele sayı üretir.
    size=(3, 4) ise, 3x4 boyutunda bir matris şeklinde rastgele
sayılar üretir.
```

random_state (default = None):

```
    Açıklama: Rastgele sayı üretiminde kullanılan seed değerini
belirler. Eğer belirli bir seed verilirse, aynı sonuçların
tekrar üretilebilmesi sağlanır. Eğer random_state=None
verilirse, her çalıştırmada farklı rastgele sayılar üretilir.
Örnek:
    random_state=42 verilirse, aynı koşullar altında aynı
rastgele sayılar üretilir.
    Eğer bu parametreye herhangi bir değer verilmezse, sonuçlar
her çalıştırmada farklı olur.
```

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

binom.rvs(1, 0.5, size=1)
```

Out[1]:
```
array([1])
```

In [12]:
```
binom.rvs(1, 0.5, size=8)
```

Out[12]:
```
array([0, 0, 0, 1, 1, 0, 0, 0])
```

In [13]:
```
# 8 para aynı anda havaya atılıyor
binom.rvs(8, 0.5, size=1)
```

Out[13]:
```
array([3])
```

In [14]:
```
binom.rvs(3, 0.5, size=10)
```

Out[14]:
```
array([2, 2, 3, 1, 0, 2, 2, 2, 2, 1])
```

In [15]:
```
# örneğin elimizde bir tarafı ağır bir para olsun.
# Bu parada %25 yazı %75 tura gelme olasılığı olsun.
binom.rvs(3, 0.25, size=10)
```

Out[15]:
```
array([1, 1, 0, 0, 0, 1, 0, 1, 2, 0])
```

scipy.stats.pmf(k, n, p, loc=0)

olasılık kütle fonksiyonu (PMF) hesaplamak için kullanılır. PMF, bir dağılımın belirli bir olasılıkta kesikli bir değer alma ihtimalini hesaplar.

```
k:
    Açıklama: Başarı sayısını temsil eder. Yani, bu parametre,
yapılan denemelerde kaç kez başarı elde etmeyi beklediğinizi
gösterir.
    Örnek: Eğer 3 deneme yapıyorsanız (n=3) ve bu denemelerde 2
başarı elde etmenin olasılığını hesaplamak istiyorsanız, k = 2
olur.

n:
    Açıklama: Deneme sayısını ifade eder. Bu parametre, toplam
kaç bağımsız deneme yapılacağını belirtir. Binom dağılımında her
bir deneme başarı ya da başarısızlıkla sonuçlanır.
    Örnek: Eğer bir madeni para 5 kez atılıyorsa, her bir atış
bağımsız bir deneme sayılır. Bu durumda n = 5 olur.

p:
    Açıklama: Başarı olasılığıdır. Her bir denemenin başarıyla
sonuçlanma olasılığını gösterir. Bu, 0 ile 1 arasında bir değer
olmalıdır.
    Örnek: Eğer bir madeni paranın yazı gelme olasılığı %50 ise,
p = 0.5 olur.

loc (default = 0):
    Açıklama: Kayma (lokasyon) parametresidir. Binom dağılımı
genellikle 0'dan başlar, ancak bu parametreyi kullanarak
dağılımın başlangıç noktasını değiştirebilirsiniz.
    Örnek: Varsayılan olarak loc = 0 dır, yani başarı sayısı
0'dan başlayarak hesaplanır. Eğer loc = 1 ise, başarı sayıları
1'den itibaren sayılmaya başlanır.
```

Olasılık Kütle Fonksiyonu (PMF) Nedir?

Olasılık kütle fonksiyonu (PMF), kesikli bir rastgele değişkenin belirli bir değeri alma olasılığını hesaplar. Binom dağılımı gibi kesikli dağılımlarda, belirli bir başarı sayısının olasılığını verir. Örneğin, 10 madeni para atışında tam 5 kez yazı gelme olasılığını hesaplamak için kullanılır.

In [2]:
```
from scipy.stats import binom
# 10 jetondan 7 sinin yazı gelme olasılığı
binom.pmf(7, 10, 0.5)
```

Out[2]:
```
0.11718749999999999
```

In [ ]:
```
# 7 ve daha az yazı gelme olasılığı

binom.cdf(7, 10, 0.5)
```

In [ ]:
```
# 7'den fazla yazı gelme olasılığı

1 - binom.cdf(7, 10, 0.5)
```

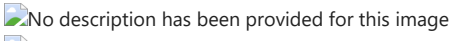
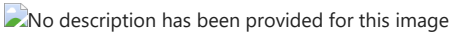
Normal Distribution

Sürekli bir olasılık dağılımıdır. Diğer olasılık dağılımlarından daha fazla gerçek dünya durumu için geçerlidir.

No description has been provided for this image

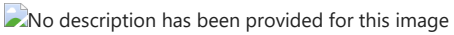
Çan eğrisi şeklindedir.

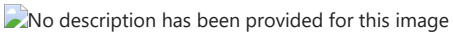
- Simetriktir. Dolayısıyla sol taraf sağ tarafın ayna görüntüsüdür.
- Herhangi bir olasılık dağılımı gibi, eğrinin altındaki alan 1'e eşittir.
- Uçların öyle görünse bile olasılık hiçbir zaman sıfıra ulaşmaz.

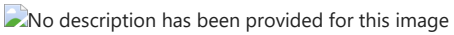



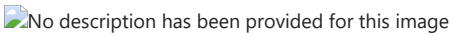
Örneğin bu dağılımda, 10'un üzerinde değer alma şansı 0.5'den az ama mümkündür.

Normal Dağılım, ortalaması ve standart sapması ile tamamlanır.









Bu duruma bazen 68-95-99.7 kuralı denmektedir.

Ortalaması 0 ve standart sapması 1 olan dağılıma standart normal dağılım denir.

Normal Dağılım Neden Önemlidir

- Birçok gerçek dünya verisi normal dağılıma çok benzemektedir.
  
- Hipotez testinde, bir örneğin ortalamasını temsil ettiği popülasyonla karşılaştırmak gibi birçok istatistiksel testi gerçekleştirmek için verilerin normal bir dağılıma uyması gerekir.
- Uçların öyle görünse bile olasılık hiçbir zaman sıfıra ulaşmaz.

scipy.stats.norm.cdf(x, loc=0, scale=1) fonksiyonu, normal dağılımın kümülatif dağılım fonksiyonunu (CDF) hesaplamak için kullanılır. Şimdi bu fonksiyonun parametrelerini detaylıca inceleyelim: Parametreler:

```
x:
    Açıklama: Bu, CDF'nin hesaplandığı noktadır. Yani, belirli
bir xx değerine kadar olan olasılığı bulmak için kullanılır.
    Fonksiyonun Anlamı: P(X≤x)P(X≤x), yani rastgele değişkenin
xx'e kadar olan kısmındaki kümülatif olasılığı hesaplar.
    Örnek: Eğer x=1.5x=1.5 ise, P(X≤1.5)P(X≤1.5) hesaplanır,
yani normal dağılımın 1.5'e kadar olan kısmındaki olasılık
bulunur.

loc (default = 0):
    Açıklama: Lokasyon parametresidir ve normal dağılımın
ortalamasını temsil eder. Varsayılan değeri 0'dır.
    Fonksiyonun Anlamı: Bu parametre, dağılımın merkezini
kaydırır. Normal dağılımın ortalaması, verilen locloc değeriyle
belirlenir.
    Örnek: Eğer loc = 2 ise, dağılımın ortalaması 2 olur. Bu
durumda, xx değerinin 2'den küçük veya eşit olma olasılığı
hesaplanır.

scale (default = 1):
    Açıklama: Scale parametresi, normal dağılımın standart
sapmasını temsil eder. Varsayılan olarak 1'dir.
    Fonksiyonun Anlamı: Standart sapma, dağılımın genişliğini
belirler. Daha büyük scale değerleri dağılımı yayarken, daha
küçük scale değerleri dağılımı daraltır.
    Örnek: Eğer scale = 3 ise, normal dağılımın standart sapması
3 olur. Bu, dağılımın yayılma genişliğini belirler.
```

Normal Dağılım ve CDF Nedir?

```
Normal dağılım (ya da Gauss dağılımı), birçok doğal olayın
dağılımını modellemek için kullanılan bir olasılık dağılımıdır.
Ortalama değerin etrafında simetriktir ve bir çan eğrisi
şeklindedir.
CDF (Cumulative Distribution Function), bir rastgele değişkenin
belirli bir değerin altında kalma olasılığını toplar. Normal
dağılımda, bu xx değerine kadar olan toplam olasılığı verir.
```

In [3]:
```
# örneğin ortalaması 161cm ve standart sapması 7 olan bir normal dağılım
# düşünelim. Bu kadınların boylarına ait dağılım olsun.
# kadınların % kaçının 154 cm'den kısa olduğunu bulmak isteyelim.

from scipy.stats import norm
norm.cdf(154, 161, 7)
```

Out[3]:
```
0.15865525393145707
```

In [4]:
```
# 154 cm'den uzun olanların yüzdesi

1 - norm.cdf(154, 161, 7)
```

Out[4]:
```
0.8413447460685429
```

In [5]:
```
# 154 ile 157 cm arasındaki kadınların yüzdesi

norm.cdf(157, 161, 7) - norm.cdf(154, 161, 7)
```

Out[5]:
```
0.1251993291672192
```

norm.ppf kullanılarak da yüzde hesaplanabilir

scipy.stats.norm.ppf(q, loc=0, scale=1) fonksiyonu, normal dağılımın ters kümülatif dağılım fonksiyonunu (percent-point function, PPF) hesaplar. PPF, kümülatif dağılım fonksiyonunun (CDF) tersidir. Yani, verilen bir olasılığa (q) karşılık gelen kritik değeri (x) hesaplar. Başka bir deyişle, belirli bir olasılığa karşılık gelen kesim noktasını verir.

Şimdi bu fonksiyonun parametrelerini açıklayalım: Parametreler:

```
q:
    Açıklama: Bu, kümülatif olasılığı (CDF'deki olasılık) temsil
eder. PPF fonksiyonu, bu olasılığa karşılık gelen kritik değeri
```

```
bulur. qq, 0 ile 1 arasında bir değer alır, çünkü kümülatif
olasılık 0 ile 1 arasında olur.
    Fonksiyonun Anlamı: qq, normal dağılımda bu kadar olasılığı
kapsayan değeri bulmanızı sağlar. Örneğin, q=0.95q=0.95 ise,
PPF, normal dağılımda %95'lik kısmın altında kalan değeri verir.
    Örnek: Eğer q=0.95q=0.95 ise, dağılımın altında kalan
%95'lik kısmı kapsayan xx değerini verir.

loc (default = 0):
    Açıklama: Lokasyon parametresi, normal dağılımın
ortalamasını belirler. Varsayılan olarak 0'dır.
    Fonksiyonun Anlamı: Bu parametre dağılımın merkezini
belirler. Yani, PPF fonksiyonu tarafından döndürülen kritik
değer, bu ortalama etrafında kaydırılır.
    Örnek: Eğer loc = 2 verilirse, dağılımın ortalaması 2 olarak
kabul edilir ve sonuç buna göre kaydırılır.

scale (default = 1):
    Açıklama: Scale parametresi, normal dağılımın standart
sapmasını belirler. Varsayılan olarak 1'dir.
    Fonksiyonun Anlamı: Standart sapma, dağılımın yayılımını
belirler. Daha büyük scale değerleri, dağılımı yayarken, daha
küçük scale değerleri dağılımı daraltır. Bulunan kritik değer,
bu yayılma ile ölçeklendirilir.
    Örnek: Eğer scale = 3 verilirse, standart sapma 3 olarak
kabul edilir ve bu, PPF fonksiyonu tarafından döndürülen kritik
değeri etkiler.
```

PPF'nin Anlamı:

PPF (percent-point function), belirli bir olasılığa karşılık gelen değeri hesaplar. PPF, CDF'nin tersidir:

```
CDF: Belirli bir değerin altında kalma olasılığını bulur.
PPF: Verilen bir olasılığa karşılık gelen değeri bulur.
```

In [3]:
```
from scipy.stats import norm
norm.ppf(0.9, 161, 7)
# kadınların %90'ı 169.97 cm'den kısadır
```

Out[3]:
```
169.9708609588122
```

In [4]:
```
# kadınların %90'ı şu boydan uzundur
norm.ppf((1 - 0.9), 161, 7)
```

Out[4]:
```
152.0291390411878
```

scipy.stats.norm.rvs(loc=0, scale=1, size=1, random_state=None) fonksiyonu, normal dağılımdan rastgele örnekler (random variates) üretmek için kullanılır. Normal dağılım, ortalaması loc ve standart sapması scale olan bir dağılımdır. Bu fonksiyon, belirtilen normal dağılımdan rastgele değerler üretir. Şimdi parametreleri detaylı olarak inceleyelim: Parametreler:

```
loc (default = 0):
    Açıklama: Bu parametre, dağılımın ortalamasını belirtir.
Varsayılan olarak 0'dır. Bu, normal dağılımın merkez noktasıdır.
    Örnek: Eğer loc = 5 ise, üretilen değerler, ortalaması 5
olan bir normal dağılımdan gelir. Yani, dağılım 5 etrafında
merkezlenir.

scale (default = 1):
    Açıklama: Bu parametre, dağılımın standart sapmasını
belirtir. Varsayılan olarak 1'dir. Standart sapma, dağılımın
yayılımını gösterir.
    Örnek: Eğer scale = 2 ise, dağılımın standart sapması 2
olur, yani dağılım 2 birim genişler. Standart sapma büyüdükçe,
üretilen değerler dağılımın ortalama etrafında daha geniş bir
aralığa dağılır.

size (default = 1):
    Açıklama: Üretilmesini istediğiniz rastgele sayıların
sayısını belirtir. Eğer bir sayı verirseniz, o kadar rastgele
sayı üretilir. Bu parametre bir tamsayı ya da bir tuple
olabilir.
    Örnek:
        size=1: Tek bir rastgele sayı üretir.
        size=5: Beş tane rastgele sayı üretir.
        size=(2, 3): 2x3 boyutunda bir matris şeklinde rastgele
sayılar üretir.

random_state (default = None):
    Açıklama: Rastgele sayı üretiminde kullanılan seed (tohum)
değerini belirler. Eğer belirli bir seed verilirse, aynı
koşullar altında tekrar çalıştırıldığında aynı sonuçlar
üretilir. Bu, yeniden üretilebilir sonuçlar elde etmek için
kullanılır.
    Örnek:
        random_state=42: Aynı koşullar altında aynı rastgele
sayıları üretir.
        Eğer random_state=None ise, her çalıştırmada farklı
rastgele sayılar üretilir.
```

In [5]:
```
norm.rvs(161, 7, size=10)
```

Out[5]:
```
array([160.38269823, 153.38682633, 159.06325788, 150.50315155,
       159.20234513, 161.83368275, 153.080613  , 176.74723586,
       169.06802644, 152.73262967])
```

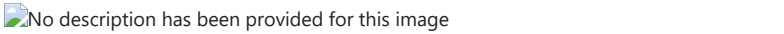
Skewness (Çarpıklık)

Veri dağılımını yorumlarken verilerin sona erdiği yönü tanımlayan çarpıklık terimi kullanılır.


No description has been provided for this image

Yukarıdaki çizim solda zirve yapar ve sağa doğru biter; dolayısıyla kuyruk daha büyük pozitif değerlerin olduğu yerde sağda olduğundan dağılım pozitif çarpık veya sağa çarpıktır.

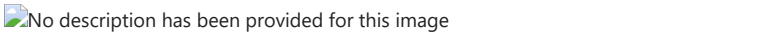
Tersine negatif çarpık veya sola çarpık dağılım sağda zirve yapar ve sola doğru sona erer.



Hane halkı geliri gibi gerçek dünya verilerinde çarpıklık gözlenmesi çok yaygın bir durumdur. Bazı hanelerin normal gelirden çok fazla kazanması nedeniyle genellikle pozitif çarpıktır.

Kurtosis (Basıklık)

Bir dağılım basıklığıyla da yorumlanabilir. Kurtosis, dağılımdaki aşırı değerlerin oluşumunu açıklamanın bir yoludur. Üç tür basıklık vardır.

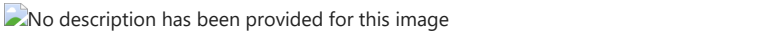


- Pozitif Basıklık (Leptokurtic) : Grafikte kırmızıyla gösterilmiştir. Ortalama etrafında, büyük bir tepe noktası ve daha küçük standart sapma ile karakterize edilir.
- Mesokurtic Basıklık, çizimde mavi olarak gösterilen normal dağılımdır.
- Negatif Basıklık (Platykurtic) : Çizimde yeşil renkle gösterilmiştir. Daha düşük zirveye ve daha büyük standart sapmaya sahip bir dağılımdır.

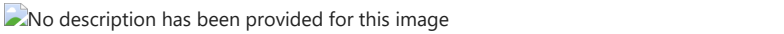
The Central Limit Theorem

Normal dağılım hakkında bilgi edindik. Şimdi bunu bu kadar önemli kılan teorem hakkında bilgi verelim.

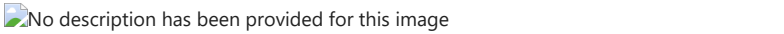
Adil bir zar beş defa atılsın ve elde edilen sonuçlar kaydedilsin.

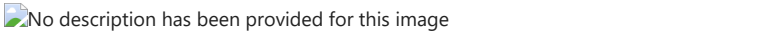


Aynı işlem farklı zamanlarda tekrar edilirse farklı ortalamalar elde edilir.



Bu 10 kez tekrar edilsin. Yani zar 5 kez atılsın, bu 5 atışın ortalaması alınsın ve bu işlem 10 kez tekrar edilsin.

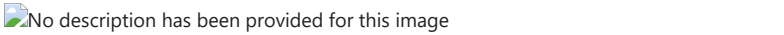




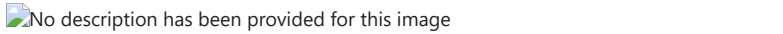
Ortalama gibi bir özet istatistiğinin dağılımına örneklem dağılımı denir.

Bu dağılım, özellikle, örnek ortalamasının bir örnekleme dağılımıdır.

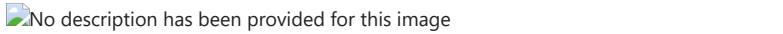
100 örnek ortalama almak için bu işlem 100 defa tekrarlanabilir.



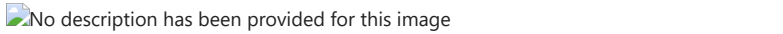
Yeni örneklem dağılımına baklırısa, her bir zar atımı için sonuçların dağılımı uniform olsa bile, şeklinin biraz normal dağılıma benzediği görülür.

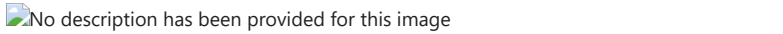


Bu örneklem dağılımı normal dağılıma daha çok benzemektedir.



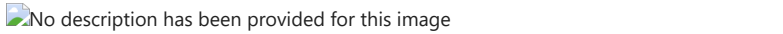
Şekil on bin örnekte tutarlı kalır.





Bu olay merkezi limit teoremi olarak bilinir ve örneklemin şu şekilde olduğunu belirtir.

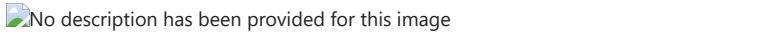
Bir istatistiğin örnekleme dağılım, örneklem büyüklüğü arttıkça normal dağılıma yaklaşır.



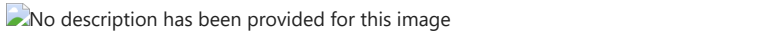
Merkezi limit teoreminin yalnızca örnekler rastgele alındığında ve bağımsız olduğunda geçerli olduğunu belirtmek önemlidir.

Genel olarak merkezi limit teoreminin uygulanabilmesi için örneklem büyüklüğünün en az 30 olması önemlidir.

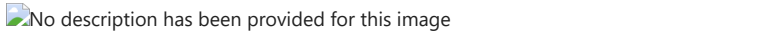
MLT diğer özet istatistikler için de geçerlidir.



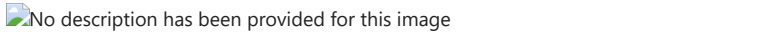
MLT'nin geçerli olduğu bir diğer özet istatistik proportion'dır (orantı).



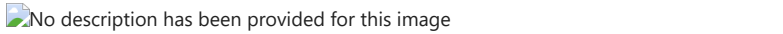
Örneğin bir zar 5 defa atılsın ve kaç kez 4 atıldığına bakılsın. Bu 1000 kez tekrarlanıp dağılıma baklırsa:



Bu örnekleme dağılımları normal olduğundan, bir dağılımın ortalaması, standart sapması veya oranı hakkında bir tahmin elde edilebilir.



Bu büyük sayılar yasasının uygulamasına bir örnektir.



In [7]:
```
import pandas as pd
import numpy as np

die = pd.Series([1, 2, 3, 4, 5, 6])
samp_5 = die.sample(5, replace=True)
samp_5
```

Out[7]:
```
2    3
0    1
0    1
0    1
3    4
dtype: int64
```

In [8]:
```
samp_5.mean()
```

Out[8]:
```
4.0
```

In [9]:
```
samp_5_2 = die.sample(5, replace=True)
samp_5_2
```

Out[9]:
```
2    3
0    1
3    4
3    4
0    1
dtype: int64
```

In [10]:
```
samp_5_2.mean()
```

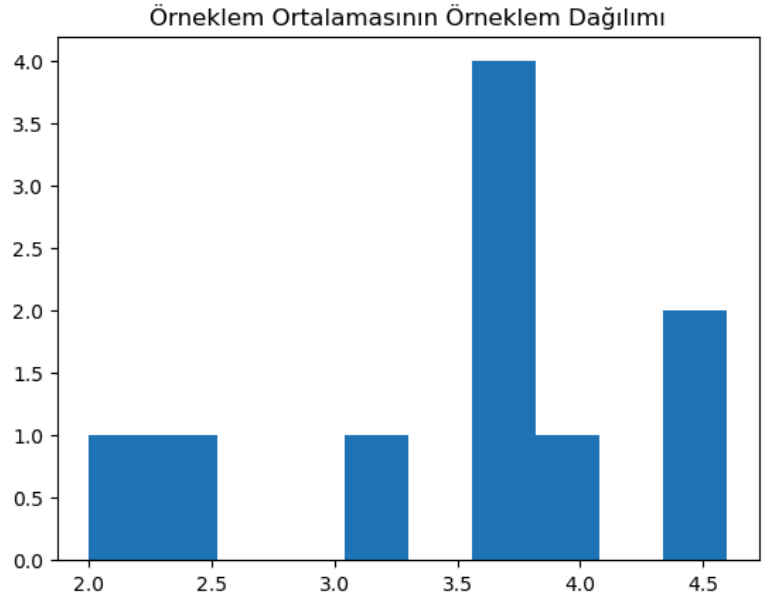
Out[10]:
```
2.6
```

In [11]:
```
samp_5_3 = die.sample(5, replace=True)
samp_5_3.mean()
```

Out[11]:
```
3.8
```

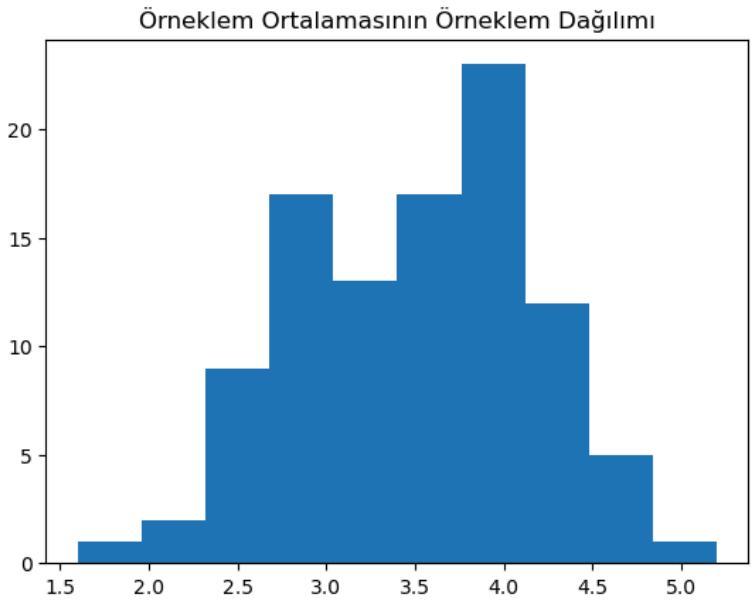
In [9]:
```
# Yukarıdaki işlem 10 defa yapılıp ortalama alınırsa
import matplotlib.pyplot as plt

sample_means = []
for i in range(10):
    samp_5 = die.sample(5, replace=True)
    sample_means.append(samp_5.mean())
plt.title("Örneklem Ortalamasının Örneklem Dağılımı")
plt.hist(sample_means)
# Böyle bir özet istatistiğinin dağılımına örneklem dağılımı denir
```

Out[9]:
```
(array([1., 1., 0., 0., 1., 0., 4., 1., 0., 2.]),
 array([2.  , 2.26, 2.52, 2.78, 3.04, 3.3 , 3.56, 3.82, 4.08, 4.34, 4.6 ]),
 <BarContainer object of 10 artists>)
```



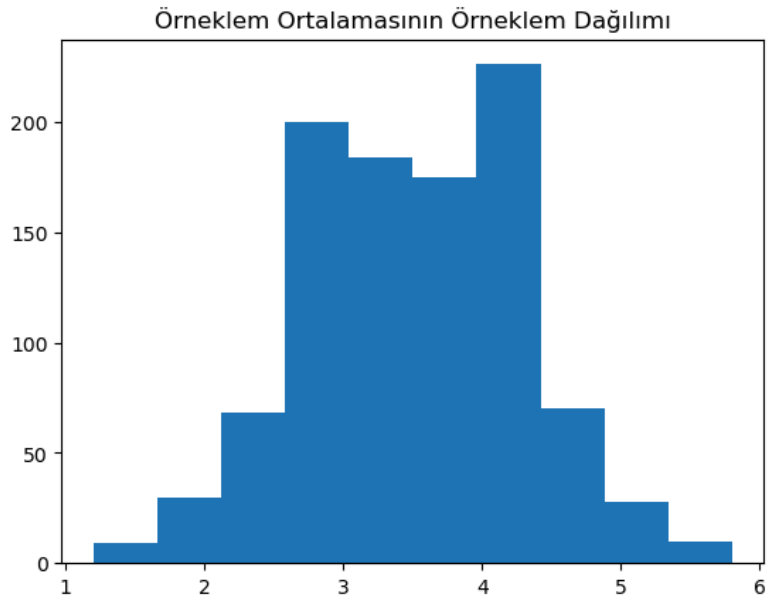
In [13]:
```
sample_means = []
for i in range(100):
    samp_5 = die.sample(5, replace=True)
    sample_means.append(samp_5.mean())
plt.title("Örneklem Ortalamasının Örneklem Dağılımı")
plt.hist(sample_means)
```

Out[13]:
```
(array([ 1.,  2.,  9., 17., 13., 17., 23., 12.,  5.,  1.]),
 array([1.6 , 1.96, 2.32, 2.68, 3.04, 3.4 , 3.76, 4.12, 4.48, 4.84, 5.2 ]),
 <BarContainer object of 10 artists>)
```



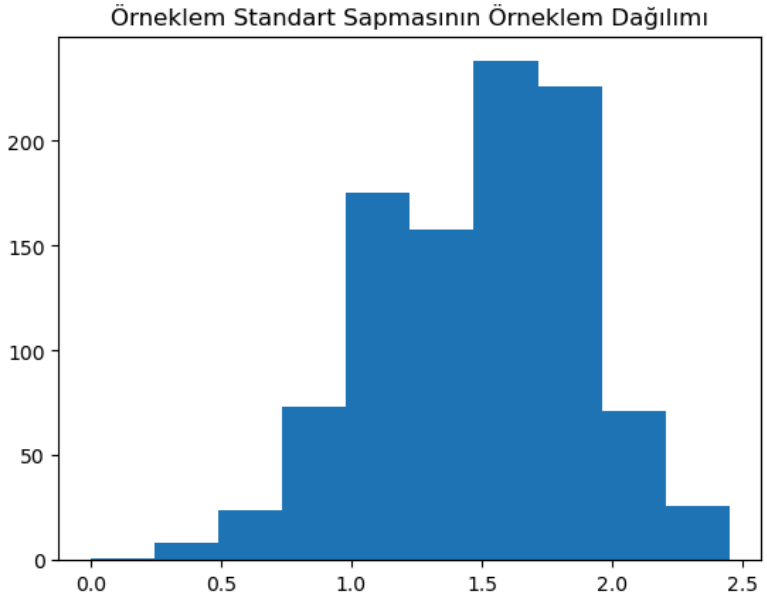
In [14]:
```
sample_means = []
for i in range(1000):
    samp_5 = die.sample(5, replace=True)
    sample_means.append(samp_5.mean())
plt.title("Örneklem Ortalamasının Örneklem Dağılımı")
plt.hist(sample_means)
```

Out[14]:
```
(array([  9.,  30.,  68., 200., 184., 175., 226.,  70.,  28.,  10.]),
 array([1.2 , 1.66, 2.12, 2.58, 3.04, 3.5 , 3.96, 4.42, 4.88, 5.34, 5.8 ]),
 <BarContainer object of 10 artists>)
```



In [10]:
```
sample_sts = []
for i in range(1000):
    samp_5 = die.sample(5, replace=True)
    sample_sts.append(np.std(samp_5))
plt.title("Örneklem Standart Sapmasının Örneklem Dağılımı")
plt.hist(sample_sts)
```

Out[10]:
```
(array([  1.,   8.,  24.,  73., 175., 158., 238., 226.,  71.,  26.]),
 array([0.        , 0.24494897, 0.48989795, 0.73484692, 0.9797959 ,
        1.22474487, 1.46969385, 1.71464282, 1.95959179, 2.20454077,
        2.44948974]),
 <BarContainer object of 10 artists>)
```



In [16]:
```
import pandas as pd
sales_team = pd.Series(['Amir', 'Brian', 'Claire', 'Damian'])
sales_team.sample(10, replace=True)
# %20 Cllire
```

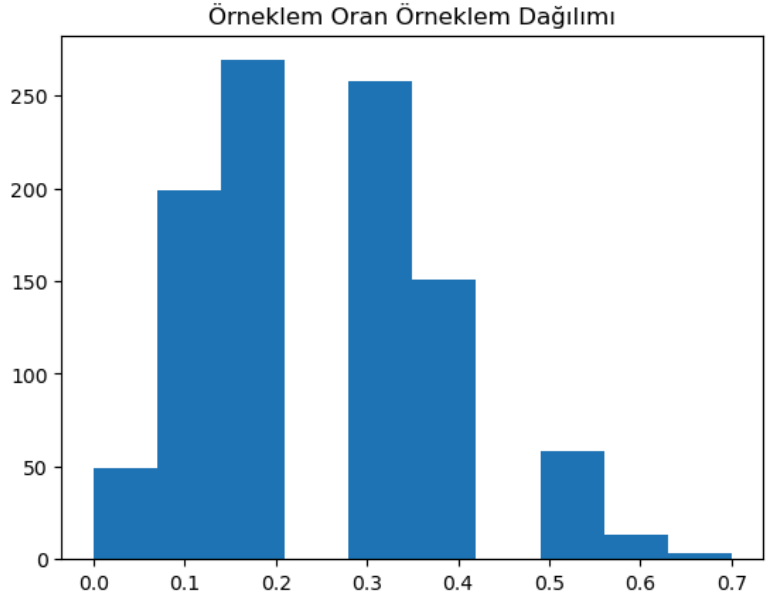
Out[16]:
```
3    Damian
3    Damian
1     Brian
0      Amir
3    Damian
3    Damian
3    Damian
3    Damian
1     Brian
0      Amir
dtype: object
```

In [17]:
```
sales_team.sample(10, replace=True)
# %30 Claire
```

Out[17]:
```
2    Claire
1     Brian
0      Amir
0      Amir
1     Brian
2    Claire
2    Claire
2    Claire
2    Claire
1     Brian
dtype: object
```

In [18]:
```
sample_prp = []
for i in range(1000):
    samp_5 = sales_team.sample(10, replace=True)
    try:
        sample_prp.append(samp_5.value_counts()['Claire'] / 10)
    except:
        sample_prp.append(0)
plt.title("Örneklem Oran Örneklem Dağılımı")
plt.hist(sample_prp)
```

Out[18]:
```
(array([ 49., 199., 269.,   0., 258., 151.,   0.,  58.,  13.,   3.]),
 array([0.  , 0.07, 0.14, 0.21, 0.28, 0.35, 0.42, 0.49, 0.56, 0.63, 0.7 ]),
 <BarContainer object of 10 artists>)
```



Poisson Distribution

Poisson Süreci

Poisson süreci, belirli bir zaman dilimindeki ortalama olay sayısının bilindiği, ancak olaylar arasındaki zaman veya boşluğun rastgele olduğu bir süreçtir.

Poisson süreçleri günlük hayatta çok yaygındır.

- Bir hayvan barınağından her hafta sahiplenilen hayvan sayısı poisson sürecidir.
- Saat başına bir restorana gelen kişi sayısı.
- Günlük web sitesi ziyaret sayısı.

Poisson dağılımı, belirli bir zaman diliminde bazı olayların meydaha gelme olasılığını açıklar.

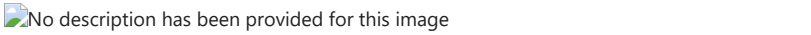
- Haftada en az beş hayvanın bir hayvan barınağından sahiplenilme olasılığı.
- Bir restorana saatte 12 kişinin gelme olasılığı.
- Bir web sitesinin bir günde 200'den az ziyaret edilme olasılığı.

Poisson dağılım Lambda(λ) ile tanımlanır.

λ = zaman periyodu başına ortalama olay sayısı

- Restoran örneğinde bu değer, saat başına ortalama müşteri sayısı olan 20'dir.
- Bu değer aynı zamanda dağılımın beklenen değeridir.

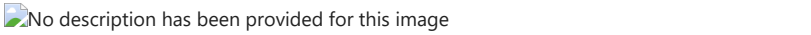
Örneklem büyüklüğü 200 ve beklenen değeri 20 olan bir poisson dağılımı şuna benzer.



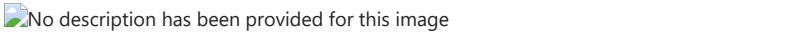
Olaylar sayıldığı için kesikli bir dağılımdır ve 20 bir saat içinde ziyaret etmesi en muhtemel müşteri sayısıdır.

Lambda dağılımın şeklini değiştirir.

Saat başına müşteri örneği kullanılarak bir poisson dağılımı görülebilir.

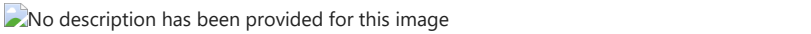


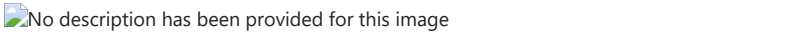
Lambda değeri ne olursa olsun dağılımın zirvesi her zaman lambda değerindedir.



Tıpkı diğer dağılımlarda olduğu gibi, çok sayıda örnek varsa ve her birinin ortalaması hesaplanırsa, Poisson Dağılımı olarak örnek ortalamalarının dağılımı normal dağılıma benzer.

Bir restoranı ziyaret eden belirli sayıda müşterinin olasılığı, bu değeri temsil eden çubuğun yüksekliği ölçülerek hesaplanabilir.





In [20]:
```
# Haftada ortalama 8 sahiplenmenin gerçekleştiği bir sığınakta
# bir haftada 5 sahiplenme gerçekleşme olasılığı nedir?

from scipy.stats import poisson

poisson.pmf(5, 8)


# 5 veya daha az sahiplenme olasılığı

poisson.cdf(5, 8)

#5'den fazla sahiplenme olasılığı

1 - poisson.cdf(5, 8)
```

Out[20]:
```
0.8087639379203747
```

In [21]:
```
poisson.rvs(8, size=10)
```

Out[21]:
```
array([ 9,  8,  5,  8,  8,  8, 10, 10,  6,  5])
```

Diğer Olasılık Dağılımları

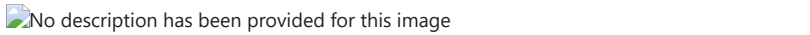
Exponential Distribution (Üstel Dağılım) Poisson olayları arasında belirli bir zaman geçme olasılığını temsil eden dağılımdır.

- Evlat edinmeler arasında 1 günden fazla zaman geçme olasılığı
- Restorana gelişler arasında 10 dakikadan az zaman geçme olasılığı
- Depremler arasında 6-8 ay geçme olasılığı

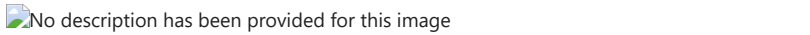
Üstel dağılım, Poisson dağılımında olduğu gibi oranı temsil eden aynı lambda değerini kullanır. Bu bağlamda lambda ve oranın aynı değer anlamına gelir. Ayrıca Poisson dağılımının aksine, zamanı temsil ettiği için süreklidir.

Müşteri hizmetleri talepleri

Örneğin, her 2 dakikada bir müşteri hizmetleri bileti oluşturulsun. Bu bir dakikalık bir zaman aralığı cinsinden yeniden ifade edilebilir, böylece her dakika bir biletin yarısı oluşturulur. Lambda değeri olarak 0.5 kullanılır. Yarım oranlı üstel dağılım şu şekilde görünür.



Rate(λ), dağılımın şeklini ve ne kadar dik bir şekilde azaldığını etkiler.

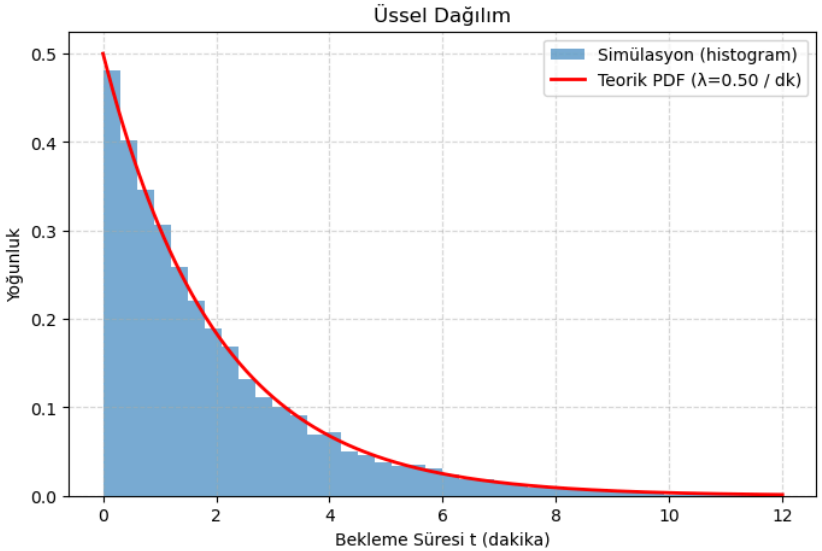


Üstel Dağılımın Beklenen Değeri

- Lambda, sıklığı oran veya olay sayısı cinsinden ölçen Poisson dağılımının beklenen değeridir. λ = 0.5 dakika başına istek sayısı
- Üstel dağılım, sıklığı olaylar arasındaki süre açısından ölçer. Üstel dağılımın beklenen değeri, 1 / λ ile hesaplanabilir. 1 / λ = 1 / 0.5 = 2 Her iki dakikada bir istek

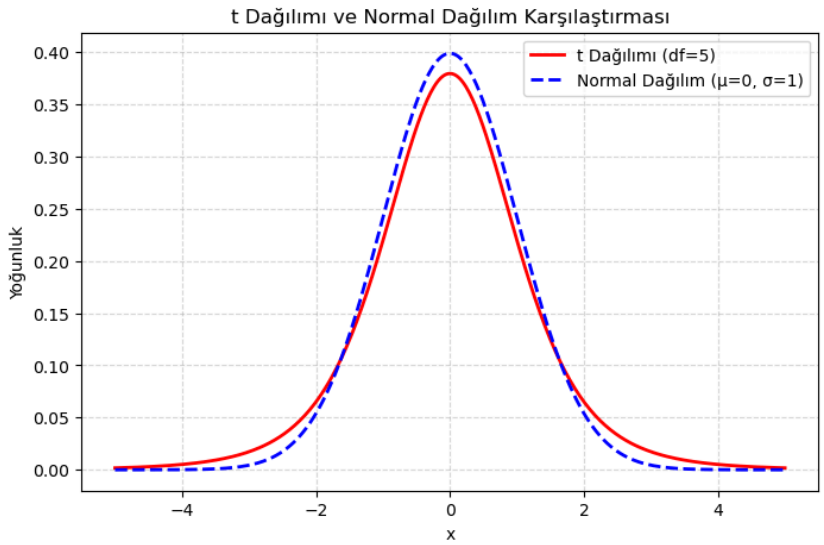
In [11]:
```
from scipy.stats import expon
import matplotlib.pyplot as plt
import numpy as np

ortalama = 2
lmb = 1 / 2

x = np.linspace(0, 12, 400)

orneklem = expon.rvs(scale=ortalama, size=10000, random_state=42)

pdf = expon.pdf(x, scale=ortalama)
cdf = expon.cdf(x, scale=ortalama)
```

```
plt.figure(figsize=(8, 5))
plt.hist(orneklem, bins=40, range=(0, 12), density=True, alpha=0.6, label="Simül
plt.plot(x, pdf, 'r-', lw=2, label=f"Teorik PDF (λ={lmb:.2f} / dk)")

plt.title("Üssel Dağılım")
plt.xlabel("Bekleme Süresi t (dakika)")
plt.ylabel("Yoğunluk")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.5)
```



In [ ]:
```
# Yeni bir talep oluşturulana kadar ne kadar süre geçecek?
from scipy.stats import expon

# yeni bir istek için 1 dakikadan az bekleme olasılığı

expon.cdf(1, scale=2)

# 4 dakikadan fazla bekleme olasılığı

1 - expon.cdf(4, scale=2)

# 1 ile 4 dakika arasında bekleme olasılığı

expon.cdf(4, scale=2) - expon.cdf(1, scale=2)
```

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import t, norm

# Parametreler
df = 5  # t dağılımı serbestlik derecesi
x = np.linspace(-5, 5, 400)

# PDF'ler
t_pdf = t.pdf(x, df)
norm_pdf = norm.pdf(x, 0, 1)  # Ortalama 0, std 1

# Grafik
plt.figure(figsize=(8, 5))
plt.plot(x, t_pdf, 'r-', lw=2, label=f"t Dağılımı (df={df})")
plt.plot(x, norm_pdf, 'b--', lw=2, label="Normal Dağılım (μ=0, σ=1)")

plt.title("t Dağılımı ve Normal Dağılım Karşılaştırması")
plt.xlabel("x")
plt.ylabel("Yoğunluk")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.5)
```



(Student's) t-distribution t-dağılım, küçük örneklem büyüklükleriyle uğraşırken veya popülasyon standart sapması bilinmediğinde istatistikte yaygın olarak kullanılır.

No description has been provided for this image

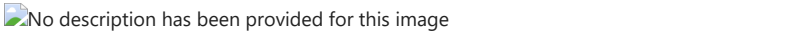
- Şekli normal dağılıma benzer, ancak tam olarak aynı değildir.
- Mavi ile gösterilen normal dağılım ile turuncu ile gösterilen bir serbestlik dereceli t-dağılımını karşılaştırırsak, t-dağılımının kuyruklarının daha kalın olduğunu görürüz.
- Bu, t-dağılımında gözlemlerin ortalamadan daha uzak düşme olasılığının daha yüksek olduğu anlamına gelir.

*__Degrees of freedom (Serbestlik Derecesi)__ Serbestlik derecesi (df), istatistikte temel bir kavramdır ve istatistiksel bir parametreyi tahmin etmek için kullanılabilen bağımsız değer veya bilgi parçalarının sayısını ifade eder. Daha basit bir ifadeyle, bir hesaplamada veriler tarafından empoze edilen herhangi bir kısıtlamayı ihlal etmeden serbestçe değişebilen değer sayısını temsil eder.

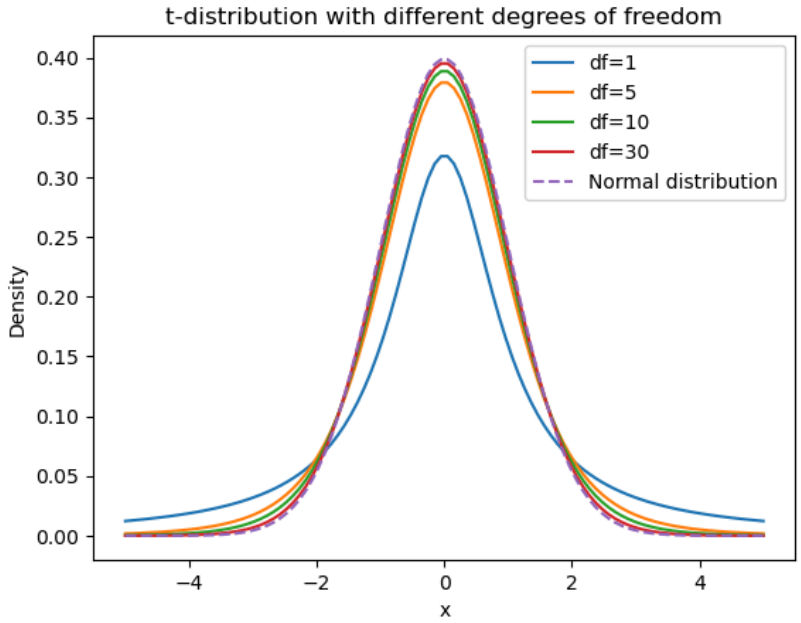
Serbestlik dereceleri t-dağılımının kuyruklarının ne kadar geniş ve düz olacağını etkiler. **Daha düşük serbestlik dereceleri (küçük df):** Dağılımın daha ağır (daha geniş) kuyrukları vardır ve daha yayılmıştır. Bu, normal dağılıma kıyasla daha yüksek uç değerler (aykırı değerler) olasılığı olduğu anlamına gelir. Örneklem boyutları küçük olduğunda daha fazla belirsizliğe neden olur. **Daha yüksek serbestlik dereceleri (büyük df):** Serbestlik dereceleri arttıkça t-dağılım normal dağılıma (çan eğrisi) yaklaşır. Serbestlik dereceleri yaklaşık 30 veya daha fazlasına ulaştığında, t-dağılım normal dağılımdan neredeyse ayırt edilemez hale gelir.

Çoğu istatistiksel testte, t-dağılımındaki serbestlik dereceleri genellikle örneklem büyüklüğüne bağlıdır. Örneğin, tek örneklemli bir t-testinde, serbestlik dereceleri n−1'dir, burada n örneklem büyüklüğüdür. **Küçük örneklem büyüklükleri (küçük df):** Küçük bir örneklem büyüklüğünüz olduğunda, t-dağılımının kuyrukları daha ağırdır çünkü popülasyon ortalamasını tahmin etmede daha fazla belirsizlik vardır. **Büyük örneklem büyüklükleri (büyük df):** Daha büyük örneklerde, t-dağılımının normal dağılıma yaklaşması, popülasyon ortalamasının tahmininin daha kesin hale gelmesidir.

Serbestlik dereceleri hipotez testindeki kritik değerleri de etkiler. **Daha düşük df (daha küçük örneklem boyutu):** Güven aralıkları veya önem testleri (t-testleri gibi) için kritik değerler daha büyük olacaktır. Bunun nedeni, daha az veri noktasıyla, ek belirsizliği hesaba katmak için daha geniş bir hata payına ihtiyaç duymanızdır. **Daha yüksek df (daha büyük örneklem boyutu):** Örneklem boyutu arttıkça kritik değerler küçülür, bu da güven aralıklarının daha dar hale geldiği ve hipotez testlerinin sıfır hipotezini reddetmek için daha az uç değer gerektirdiği anlamına gelir. Bu, daha büyük örneklerin daha büyük kesinliğini yansıtır.



In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

x = np.linspace(-5, 5, 100)

# Plot for different degrees of freedom
for df in [1, 5, 10, 30]:
    plt.plot(x, stats.t.pdf(x, df), label=f'df={df}')

plt.plot(x, stats.norm.pdf(x), label='Normal distribution',
         linestyle='--')
plt.legend()
plt.title('t-distribution with different degrees of freedom')
plt.xlabel('x')
plt.ylabel('Density')
plt.show()
```

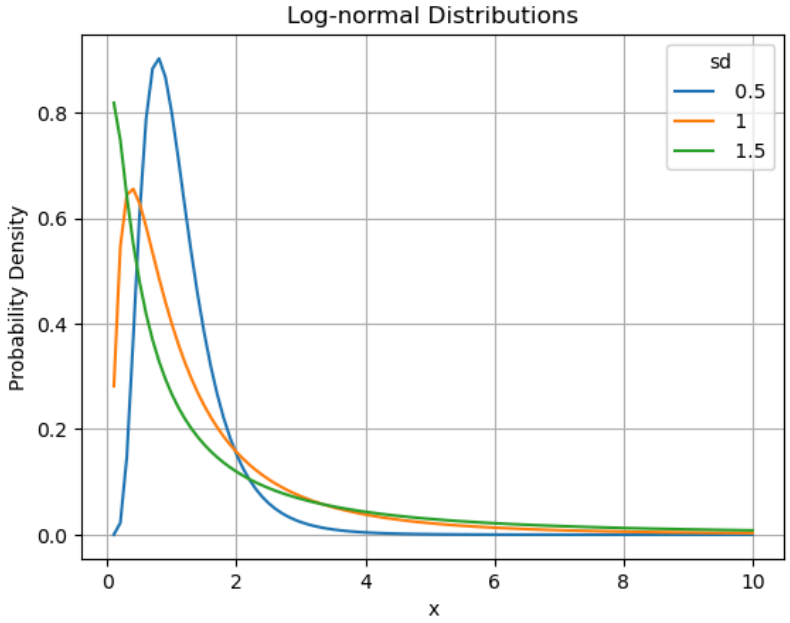


Log-normal distribution

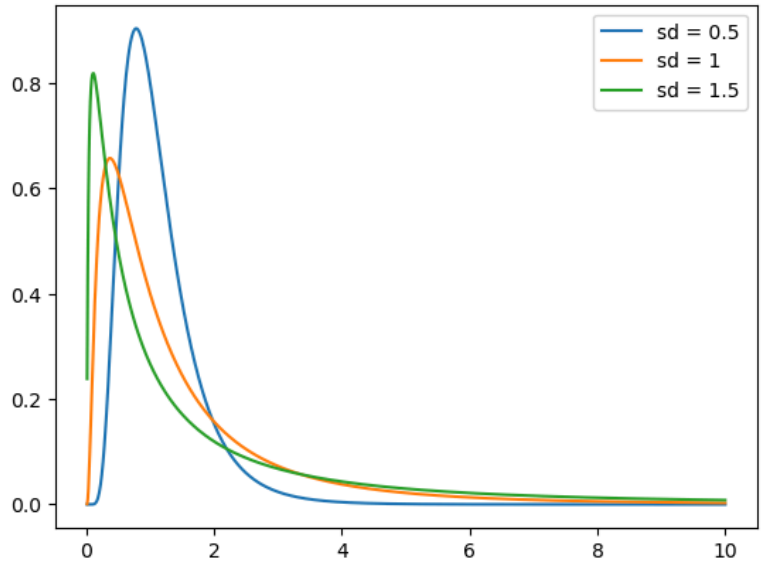
- Log-normal dağılımı izleyen değişkenlerin logaritması normal dağılım gösterir.
- Bu da normal dağılımdan farklı olarak çarpık dağılımlara neden olur.
- Satranç oyunlarının uzunluğu, yetişkinlerde kan basıncı ve 2003 SARS salgınında hastaneye yatış sayısı gibi bu dağılımı izleyen çok sayıda gerçek dünya örneği vardır.

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import lognorm

# X ekseni için değerler
x = np.linspace(0.1, 10, 100)
"""
s=Standard sapma (log) -> logaritması alınırsa normal dağılım standart sapması o
mean=Ortalama (log)
scale= Ölçek parametresi (e^mean) büyük olduğunda sağa doğru genişler (dağılımın

Log-normal dağılım, bir değişkenin logaritması normal dağılım gösteriyorsa ortay
Yani, eğer X log-normal dağılıma sahipse, log(X) normal bir dağılıma uyar.
"""

# Farklı log-normal dağılımlar için parametreler
params = [
    {'s': 0.5, 'scale': np.exp(0)},  # dar
    {'s': 1, 'scale': np.exp(0)},  # Orta
    {'s': 1.5, 'scale': np.exp(0)},  # geniş
]
```

```
for param in params:
    pdf = lognorm.pdf(x, param['s'], loc=0, scale=param['scale'])
    plt.plot(x, pdf, label=f" {param['s']}")

# Grafik ayarları
plt.title('Log-normal Distributions')
plt.xlabel('x')
plt.ylabel('Probability Density')
plt.legend(title="sd")
plt.grid(True)
plt.show()
```



In [4]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import lognorm

sd_values = [0.5, 1, 1.5]


x = np.linspace(0.01, 10, 1000)


for sd in sd_values:
    pdf = lognorm.pdf(x, sd)
    plt.plot(x, pdf, label=f'sd = {sd}')


plt.legend()
plt.show()
```



Hypothesis Testing

Hipotez testi, popülasyonları karşılaştırmak için kullanılan bir grup teori, yöntem ve tekniktir.

Neden hipotez testleri hakkında bilgi sahibi olmak gerekir?

- İlk olarak, birçok sektörde rutin olarak kullanılmaktadır. Örneğin, bir şirketin ürün fiyatını artırmanın geliri artıracağı veya bir web sitesinin adını değiştirmenin trafiği artırabileceği yönünde bir teorisi olabilir. Bir ilacın belirli sağlık koşullarının tedavisinde etkili olup olmadığını analiz etmek için hipotez testi kullanılabilir.

- Hipotez testlerinde, her zaman popülasyonlar arasında hiçbir fark olmadığı varsayımıyla başlanır. Bu, teste herhangi bir önyargı ekleme riskini azaltmak için yapılır.
- Buna sıfır(null) hipotezi denir.

Bir örnek vermek gerekirse;

Sıfır hipotezi, C vitamini takviyesi alan ve almayan kadınlar arasında cinsiyete göre doğum oranında bir fark olmadığı şeklindedir.

Daha sonra alternatif bir hipotez oluşturulur ve bu hipotez tipik olarak iki şekilde olabilir. C vitamini takviyesi alan ve almayan kadınlar arasında erkek ve kadın doğumları arasında bir fark olduğu söylenebilir. Ya da farkın yönü, örneğin C vitamini takviyesi alan nüfusun takviye almayanlara göre daha fazla kadın doğumuna sahip olduğunu belirtilebilir.

Hipotez Testi İş Akışı

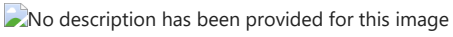
- Popülasyon tanımlanmalıdır. Aralarındaki farkı analiz etmek istediğimiz popülasyona karar verirlir. Bu durumda C vitamini takviyesi kullanan veya kullanmayan yetişkin kadınlar popülasyondur.
- Null ve Alternatif hipotez belirlenir. Ardından, her iki popülasyonda da doğumların erkek veya kız olma olasılığının eşit olduğu veya C vitamini takviyesi alan kadınlarda bebeklerin kız olma olasılığının daha yüksek olduğu şeklinde boş ve alternatif hipotezler geliştirilir.
- Örnek veriler toplanır veya bunlara erişim sağlanır.
- Veriler üzerinde istatistiksel testler gerçekleştirilir
- Sonuçlar örneklemin temsil ettiği nüfus hakkında sonuçlar çıkarmak için kullanılır.

Ne kadar veriye ihtiyaç var?

Peki kaç doğumun cinsiyeti kaydedilmelidir? Merkezi limit teoremi uygulanırsa, örneklem büyüklüğü arttıkça erkek ve kadın doğumlarının ortalama sayısı popülasyon ortalamalarına yaklaşır. Ancak, büyük örnekler toplamak çok fazla zaman ve kaynak gerektirebilir! Yaygın bir yaklaşım, örneklemlerin ne kadar büyük olduğunu bulmak için benzer hipotez testleri üzerine hakemli araştırmalara bakmaktır. Bu daha sonra bir ölçüt olarak kullanılabilir.

Independent and dependent variables

- Bağımsız değişken, diğer verilerden etkilenmesi beklenen veriyi tanımlar. C Vitamini Takviyesi
- Bağımlı Değişken, Bağımsız değişken veya değişkenler tarafından etkilenen değişken. Doğum cinsiyet oranı



Design of Experiments

- Veriler genellikle belirli bir soruyu yanıtlamayı amaçlayan bir çalışmanın sonucu olarak oluşturulur. Ancak, verilerin nasıl oluşturulduğuna ve çalışmanın nasıl tasarlandığına bağlı olarak verilerin farklı şekilde analiz edilmesi ve yorumlanması gerekir.
- Deneyler genellikle "Tedavinin tepki üzerindeki etkisi nedir?" biçimindeki bir soruyu yanıtlamayı amaçlar. Bu ortamda, tedavi açıklayıcı veya bağımsız değişkeni ifade eder ve tepki tepkiyi veya bağımlı değişkeni ifade eder. Örneğin, bir reklamın satın alınan ürün sayısı üzerindeki etkisi nedir? Bu durumda, tedavi bir reklamdır ve tepki satın alınan ürün sayısıdır.

Controlled experiments

- Kontrollü bir deneyde, katılımcılar rastgele tedavi grubuna veya kontrol grubuna atanır. Tedavi grubu tedaviyi alır ve kontrol grubu almaz.
- Bununen iyi örneklerinden biri A/B testidir. Örneğin, tedavi grubu bir reklam görecek, kontrol grubu ise görmeyecektir. Bu farkın dışında, grupların karşılaştırılabilir olması gerekir; böylece bir reklam görmenin insanların daha fazla satın almasına neden olup olmadığını belirlenebilir.
- Gruplar karşılaştırılabilir değilse, bu durum kafa karıştırıcılığa veya önyargıya yol açabilir. Tedavi grubundaki katılımcıların ortalama yaşı 25 ve kontrol grubundaki katılımcıların ortalama yaşı 50 ise, daha genç kişilerin daha fazla satın alma olasılığı daha yüksekse yaş potansiyel bir karıştırıcı olabilir ve bu da deneyi tedaviye doğru önyargılı hale getirecektir.

Deneylerin Altın Standardı

- Altın standart veya ideal deney, belirli araçları kullanarak mümkün olduğunca fazla önyargıyı ortadan kaldıracaktır.
- Kontrollü deneylerde önyargıyı ortadan kaldırmaya yardımcı olan ilk araç, randomize kontrollü bir deneme kullanmaktır. Randomize kontrollü bir denemede, katılımcılar tedavi veya kontrol grubuna rastgele atanır ve atamaları şanstan başka bir şeye dayanmaz. Bu tür rastgele atama, grupların karşılaştırılabilir olduğundan emin olmaya yardımcı olur.
- İkinci yol, tedaviye benzeyen ancak hiçbir etkisi olmayan bir şey olan plasebo kullanmaktır. Bu şekilde, katılımcılar tedavi veya kontrol grubunda olup olmadıklarını bilmezler. Bu, tedavinin etkisinin tedavinin kendisinden kaynaklandığını, tedaviyi alma fikrinden kaynaklanmadığını garanti eder. Bu, bir ilacın etkinliğini test eden klinik çalışmalarda yaygındır. Kontrol grubuna yine bir hap verilir, ancak bu, yanıt üzerinde minimum etkisi olan bir şeker hapıdır.
- Çift kör bir deneyde, tedaviyi uygulayan veya deneyi yürüten kişi, gerçek tedaviyi mi yoksa plaseboyu mu uyguladığını bilmez. Bu, yanıttaki önyargıya ve sonuçların analizine karşı koruma sağlar. Bu farklı araçların hepsi aynı prensibe dayanır: deneyinize önyargının sızması için daha az fırsat varsa, tedavinin yanıtı etkileyip etkilemediğine dair daha güvenilir bir sonuca varabilirsiniz.

Observational studies

- Gözlemsel bir çalışmada, katılımcılar gruplara rastgele atanmazlar. Bunun yerine, katılımcılar genellikle önceden var olan özelliklere göre kendilerini atarlar. Bu, kontrollü bir deney için elverişli olmayan soruları yanıtlamak için yararlıdır.
- Sigara içmenin kanser üzerindeki etkisini incelemek istiyorsanız, insanları sigara içmeye zorlayamazsınız. Benzer şekilde, geçmiş satın alma davranışının birinin bir ürünü satın alıp almayacağını nasıl etkilediğini incelemek istiyorsanız, insanları belirli geçmiş satın alma davranışlarına sahip olmaya zorlayamazsınız.
- Atama rastgele olmadığından, grupların her açıdan karşılaştırılabilir olacağını garantilemenin bir yolu yoktur, bu nedenle gözlemsel çalışmalar nedensellik kuramaz, yalnızca ilişki kurabilir. Tedavinin etkileri, belirli kişileri kontrol grubuna ve belirli kişileri tedavi grubuna sokan faktörler tarafından karıştırılabilir. Ancak, ilişki hakkındaki sonuçların güvenilirliğini güçlendirmeye yardımcı olabilecek karıştırıcıları kontrol etmenin yolları vardır.

Longitudinal vs. cross-sectional studies (Uzunlamasına ve kesitsel çalışmalar)

- Uzunlamasına bir çalışmada, aynı katılımcılar, tedavinin yanıt üzerindeki etkisini incelemek için bir süre boyunca takip edilir.
- Kesitsel bir çalışmada, veriler zaman içinde tek bir anlık görüntüden toplanır.
- Yaşın boy üzerindeki etkisini araştırmak isterseniz, kesitsel bir çalışma farklı yaşlardaki insanların boylarını ölçer ve bunları karşılaştırır. Ancak, sonuçlar doğum yılı ve yaşam tarzı tarafından karıştırılacaktır çünkü her neslin daha uzun olması mümkündür.
- Uzunlamasına bir çalışmada, aynı kişilerin boyları hayatlarının farklı noktalarında kaydedilir, böylece karıştırıcı faktör ortadan kalkar.
- Uzunlamasına çalışmaların daha pahalı olduğunu ve gerçekleştirilmesinin daha uzun sürdüğünü, kesitsel çalışmaların ise daha ucuz, daha hızlı ve daha rahat olduğunu belirtmek önemlidir.

Experiments

Deneyler, bir popülasyon hakkında sonuçlar çıkarmak için örnek veriler üzerinde istatistiksel testler yapmayı içeren hipotez testinin bir alt kümesidir.

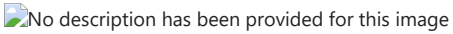
Bu sadece akademi ve araştırma için geçerli değildir; deneyler, özellikle ürün içgörüleri elde etmek ve ticari performansta iyileştirmeler sağlamak için endüstride de gerçekleştirilir.

Deneyler genellikle "Uygulamanın yanıt üzerindeki etkisi nedir?" şeklinde bir soruyu yanıtlamayı amaçlar; burada uygulama bağımsız değişkeni, yanıt ise bağımlı değişkeni ifade eder.

Advertising as a treatment

Bir deney örneği olarak, bir reklamın satın alınan ürün sayısı üzerinde ne gibi bir etkisi olduğu bilinmek istenebilir.

Bu durumda, uygulama bir reklam, yanıt ise satın alınan ürün sayısıdır.



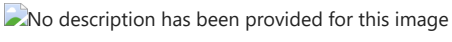
Sonuçların bir çubuk grafiği kullanılarak görselleştirilmesi, uygulamanın satın alınan ürün sayısını artırmada etkili olabileceğini göstermektedir.

Controlled experiments

Yaygın bir deney türü, katılımcıların rastgele bir şekilde uygulama grubuna ya da kontrol grubuna atandığı kontrollü bir deneydir.

Örnekte, uygulama grubu bir reklam görecek, kontrol grubu ise görmeyecektir. Bu fark dışında, gruplar karşılaştırılabilir olmalıdır, böylece bir reklam görmenin insanların daha fazla satın almasına neden olup olmadığı belirlenebilir.

Gruplar karşılaştırılabilir değilse, sonuçlara dayanarak yanlış sonuçlar çıkarılabilir. Uygulama grubundaki katılımcıların yaş ortalaması 25 ve kontrol grubundaki katılımcıların yaş ortalaması 50 ise, yaş potansiyel olarak sonuçları etkileyebilir; genç insanların daha fazla satın alma olasılığı daha yüksektir ve bu da deneyi uygulama lehine taraflı hale getirir.



The gold standard of experiments

Kontrollü deneylerde önyargıyı ortadan kaldırmaya yardımcı olan ilk yöntem rastgeleleştirmedir (Randomization).

- Katılımcılar Uygulama/Kontrol grubuna belirli özelliklerine göre değil rastgele atanır.
- Rastgeleleştirme grupların karşılaştırılabilir olmasını sağlamaya yardımcı olur.
- Buna randomize kontrollü çalışma adı verilir.

İkinci yöntem körleme(Blinding) kullanmaktır.

- Katılımcılar hangi grupta olduklarını bilmezler. Bu tedavinin etkisinin tedavi olam fikriden değil, tedavinin kendisinden kaynaklanmasını sağlar.
- Bu tedaviye benzeyen fakat hiçbir etkisi olmayan bir plasebo kullanımını içerebilir.
- Bu bir ilacın etkinliğini test eden klinik deneylerde yaygındır.

Üçüncü yöntem Çift kör randomize kontrollü bir çalışma (double-blind randomized controlled trial) kullanmaktır.

- Tedaviyi uygulayan veya deneyi yürüten kişi gerçek tedaviyi mi yoksa plaseboyu mu uyguladığını da bilmez.
- Bu, sonuçların analizinin yanı sıra yanıtta da önyargıya karşı koruma sağlar.
- Bu farklı araçların hepsi aynı ilkeye dayanır: Deneye önyargı girmesi için ne kadar az fırsat olursa, tedavinin yanıtı etkileyip etkilemediği sonucuna o kadar güvenilir bir şekilde varılabilir.

Randomized Controlled Trials vs. A/B testing

Amaç, bir ilacın farklı dozajları gibi birden fazla tedavi arasındaki farkı test etmekse, randomize kontrollü çalışmalar birden fazla tedavi grubuna sahip olabilir. Bunlar akademide, özellikle de bilimsel ve klinik araştırmalarda popülerdir.

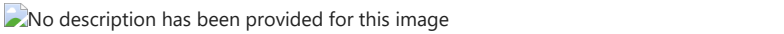
Randomize kontrollü denemeler, genellikle pazarlama ve mühendislik gibi sektörlerde kullanıldığında A/B testi olarak da adlandırılır. Aradaki fark, A/B testinin katılımcıları yalnızca uygulama ve kontrol olmak üzere iki gruba ayırmasıdır.

Correlation

Değişkenler arası ilişkiler daha önceki konularda anlatıldı. Şimdi bu ilişkiyi ölçmenin bir yolu olan Korelasyon konusundan bahsedilecektir.

Relationships between two variables

İki değişken arasındaki ilişkiyi görselleştirebilmek için dağılım grafiği kullanılır.



Burada, farklı şehirlerdeki spor salonu üyelik maliyetleri ile su maliyetlerini çiziyoruz. Bu iki değişken arasında net bir ilişki olup olmadığını belirlemek zordur.

Pearson correlation coefficient

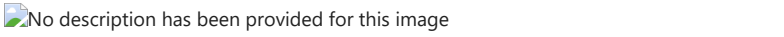
İşte bu noktada, genellikle korelasyon katsayısı olarak adlandırılan Pearson korelasyon katsayısı işe yarar.

Karl Pearson tarafından geliştirilmiş ve 1896 yılında yayınlanmıştır.

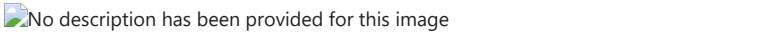
İki değişken arasındaki ilişkinin gücünü ölçer ve eksi bir ile bir arasında bir değer üretir. Bu sayı, değişkenler arasındaki ilişkinin gücüne karşılık gelir ve pozitif veya negatif işaret, ilişkinin yönüne karşılık gelir.

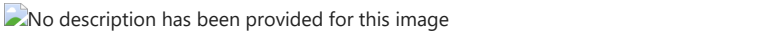
Linear relationships

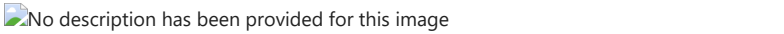
Pearson korelasyon katsayısı yalnızca doğrusal ilişkiler için kullanılabilir, yani değişkenler arasındaki değişiklikler orantılıdır.

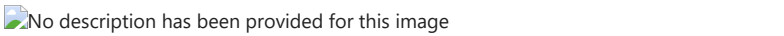


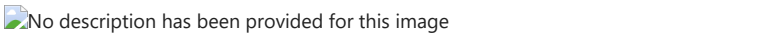
Örneğin, Londra'da bir şişe suyun bir dolar ve spor salonu üyeliğinin aylık fiyatının yirmi dolar olduğunu varsayalım. Eğer su Paris'te iki kat daha pahalıysa, o zaman spor salonu üyeliği 40 dolar olmalıdır.

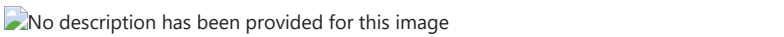


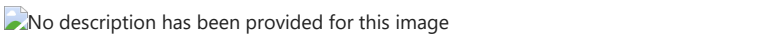




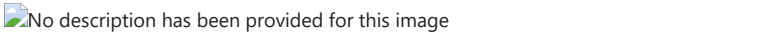




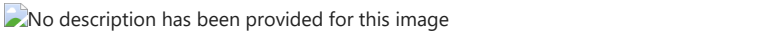




Net bir çizgi yok, bu da çok güçlü bir ilişki olmadığını gösteriyor, ancak her iki değer de birlikte artma eğiliminde. Yani, belki de zayıf-orta düzeyde pozitif bir korelasyon vardır.



Eğilim çizgisi ilişkiyi görselleştirmeyi kolaylaştırır. Pearson korelasyon katsayısı 0,35 olup, spor salonu üyeliğinin maliyeti ile bir şişe suyun maliyeti arasında zayıf ila orta düzeyde pozitif bir ilişki olduğunu teyit etmektedir.



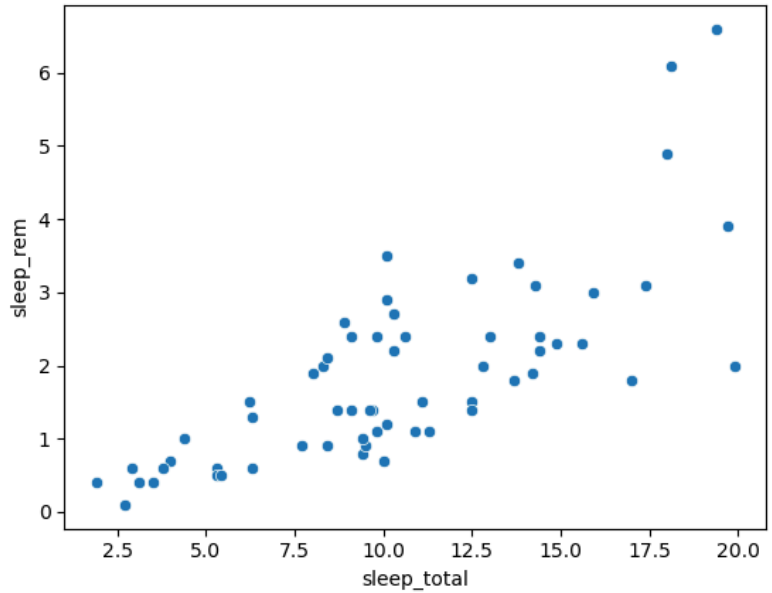
Korelasyon katsayısını kullanarak değişkenler arasındaki ilişkiyi yorumlarken dikkatli olunmalıdır. Burada yaşam beklentisi ve bir şişe suyun maliyetini gösteren bir grafik yer almaktadır. Korelasyon katsayısı 0,61'dir ve orta düzeyde pozitif bir ilişki olduğunu göstermektedir.

Korelasyon nedenselliğe eşit değildir. Bu, su maliyetinin artmasının ortalama yaşam süresini artıracağı anlamına mı geliyor? Bir ilişkinin var olmasının, su maliyetlerindeki değişikliklerin yaşam beklentisinde bir değişikliğe yol açacağı anlamına gelmediğini ayırt etmek önemlidir.

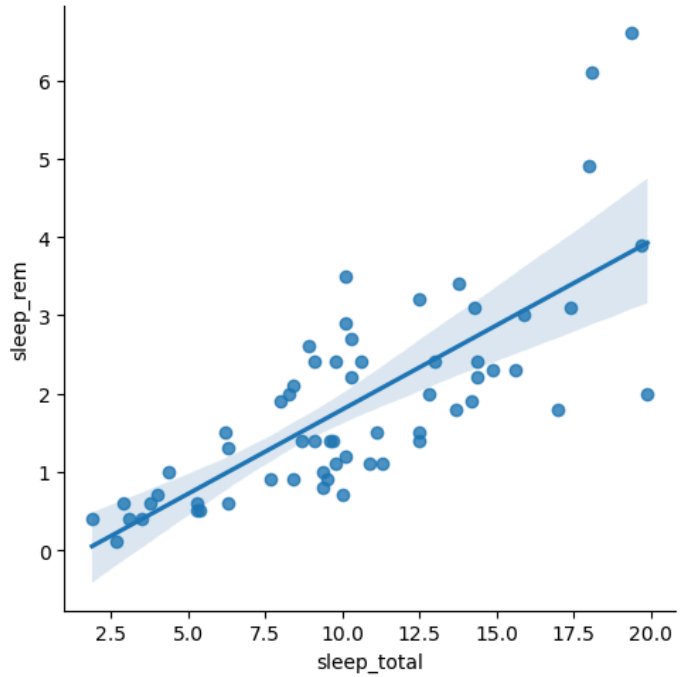
Confounding variables

Veriler arasındaki ilişkilere bakarken, değerleri başka nelerin etkiliyor olabileceğini sormak önemlidir. Bir şişe suyun maliyeti, daha güçlü ekonomilere sahip yerlerde genellikle daha yüksektir ve bu yerler yüksek kaliteli sağlık hizmetlerine daha iyi erişim sunabilir. Dolayısıyla, belki de yaşam beklentisi bir şişe suyun maliyetinden etkilenmiyor, aslında ekonominin gücünden etkileniyordur. Bu, analiz ettiğimiz verileri etkileyen ancak değişkenler arasındaki ilişkiyi değerlendirirken hesaba katılmayan bir şey olan karıştırıcı değişken olarak bilinir.

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

df_sleep = pd.read_csv("data/msleep.csv")

sns.scatterplot(x="sleep_total", y="sleep_rem", data=df_sleep)
plt.show()
```



In [3]:
```
import seaborn as sns
sns.lmplot(x="sleep_total", y="sleep_rem", data=df_sleep, ci=95)
plt.show()

'''
ci parametresi, çizilen regresyon doğrusunun güven aralığını kontrol etmek için
ci, regresyon çizgisi etrafındaki gölgeli alanı ifade eder ve bu gölgeli alan, r
sahip olduğunu gösterir.
ci Parametresi:

    Açıklama: ci, regresyon çizgisi etrafındaki güven aralığını belirler. Güven
    regresyon çizgisinin doğruluğunu görsel olarak temsil eder. Varsayılan olara
    Kullanım: Eğer ci=95 olarak ayarlanırsa, grafikte regresyon doğrusunun etraf
    Bu, regresyon çizgisinin %95 güvenle bulunduğu aralığı ifade eder.
    Değerler:
        Bir sayı olarak (örneğin, ci=95), bu güven aralığı yüzdesini belirtir.
        ci=None olarak ayarlanırsa, güven aralığı çizilmez ve sadece regresyon ç

Güven Aralığı Nedir?

Güven aralığı, tahmin edilen regresyon çizgisinin hangi aralıkta bulunabileceğin

Ne Zaman ci=None Kullanılabilir?

Eğer yalnızca regresyon çizgisi ile ilgileniyorsanız ve güven aralığına gerek yo
Özet:

    ci parametresi, regresyon çizgisi etrafında çizilen güven aralığını kontrol
    ci=95: %95 güven aralığı çizilir (varsayılan).
    ci=None: Güven aralığı çizilmez, sadece regresyon çizgisi gösterilir.
'''
```



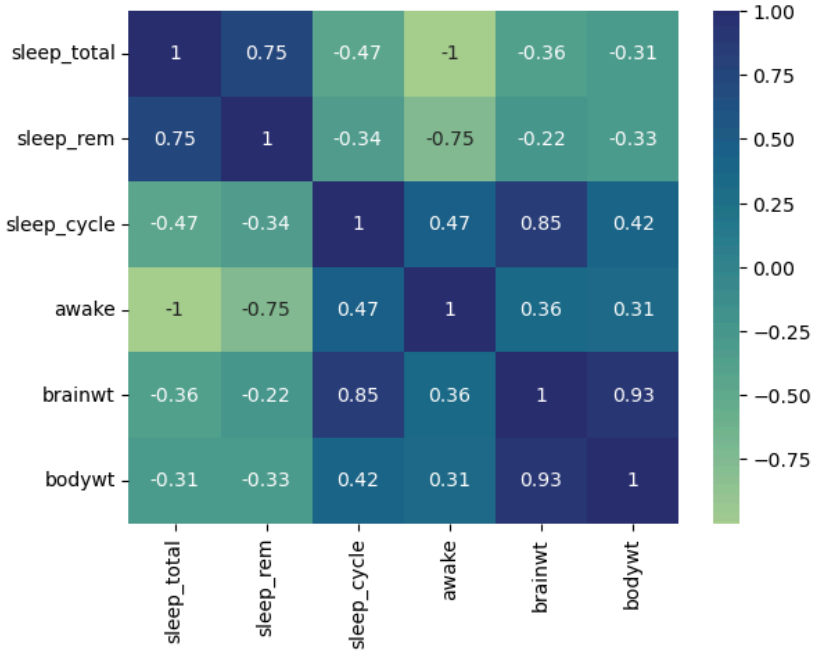
Out[3]:
```
'\nci parametresi, çizilen regresyon doğrusunun güven aralığını kontrol etmek i
çin kullanılır. \nci, regresyon çizgisi etrafındaki gölgeli alanı ifade eder ve
bu gölgeli alan, regresyon çizgisinin belirli bir güven aralığına \nsahip olduğ
unu gösterir.\nci Parametresi:\n\n    Açıklama: ci, regresyon çizgisi etrafında
ki güven aralığını belirler. Güven aralığı, verinin değişkenliğini ve \n    reg
resyon çizgisinin doğruluğunu görsel olarak temsil eder. Varsayılan olarak, bu
değer 95 (yani %95 güven aralığı) olarak ayarlanmıştır.\n    Kullanım: Eğer ci=
95 olarak ayarlanırsa, grafikte regresyon doğrusunun etrafında %95 güven aralığ
ı çizilir. \n    Bu, regresyon çizgisinin %95 güvenle bulunduğu aralığı ifade e
der.\n    Değerler:\n        Bir sayı olarak (örneğin, ci=95), bu güven aralığı
yüzdesini belirtir.\n        ci=None olarak ayarlanırsa, güven aralığı çizilmez
ve sadece regresyon çizgisi gösterilir.\n\nGüven Aralığı Nedir?\n\nGüven aralığ
ı, tahmin edilen regresyon çizgisinin hangi aralıkta bulunabileceğine dair bir
belirsizlik ölçüsüdür. \n\nNe Zaman ci=None Kullanılabilir?\n\nEğer yalnızca re
gresyon çizgisi ile ilgileniyorsanız ve güven aralığına gerek yoksa, grafiği sa
deleştirmek için ci=None ayarlaması yapılabilir. Bu, grafik üzerinde daha net b
ir görsellik sunar, özellikle çok fazla veri noktası olduğunda güven aralığı di
kkati dağıtabilir.\nÖzet:\n\n    ci parametresi, regresyon çizgisi etrafında çi
zilen güven aralığını kontrol eder.\n    ci=95: %95 güven aralığı çizilir (vars
ayılan).\n    ci=None: Güven aralığı çizilmez, sadece regresyon çizgisi gösteri
lir.\n'
```

In [4]:
```
df_sleep["sleep_total"].corr(df_sleep["sleep_rem"])
```

Out[4]:
```
0.7517549992287144
```

In [7]:
```
sns.heatmap(df_sleep.corr(numeric_only=True), annot=True,
            cmap='crest')
```

Out[7]:
```
<Axes: >
```



Korelasyon Uyarıları (Correlation caveats)

Korelasyon, ilişkileri ölçmek için yararlı bir yol olsa da, bazı uyarıları dikkate almak gereklidir.

Non-linear relationships

- No description has been provided for this image
- Yukarıdaki grafiği gözönünde bulunduralım. X ve Y arasında açıkça bir ilişki var, ancak korelasyonu hesapladığında 0.18 elde edilir. Bunun nedeni, iki değişken arasındaki ilişkinin doğrusal bir ilişki değil, ikinci dereceden bir ilişki olmasıdır. Korelasyon katsayısı sadece doğrusal ilişkilerin gücünü ölçer.
- Tüm özet istatistiklerde olduğu gibi korelasyon da körü körüne kullanılmamalı ve mümkün olduğunca veriler görselleştirilmelidir.

In [6]:
```
import pandas as pd

df_sleep = pd.read_csv("data/msleep.csv")
df_sleep
```

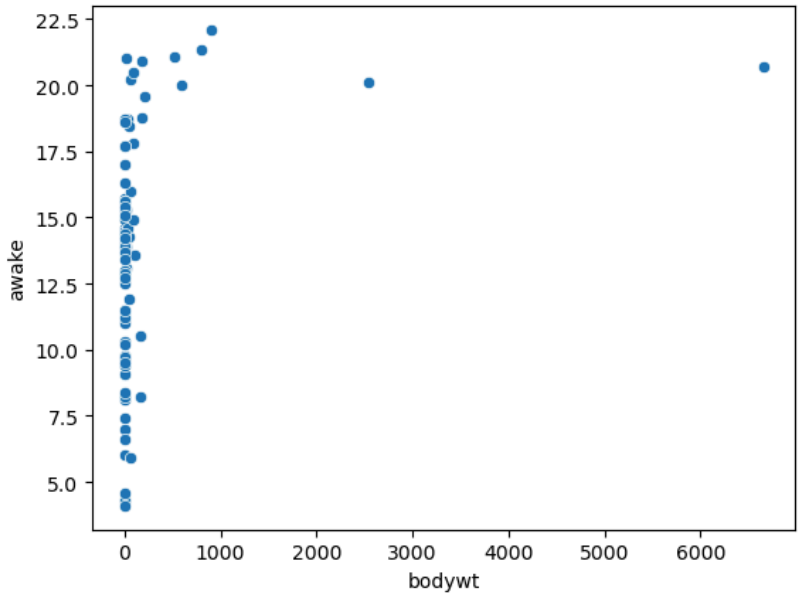
Out[6]:

	name	genus	vore	order	conservation	sleep_total	sleep_rem	sle
0	Cheetah	Acinonyx	carni	Carnivora	lc	12.1	NaN	
1	Owl monkey	Aotus	omni	Primates	NaN	17.0	1.8	
2	Mountain beaver	Aplodontia	herbi	Rodentia	nt	14.4	2.4	
3	Greater short-tailed shrew	Blarina	omni	Soricomorpha	lc	14.9	2.3	
4	Cow	Bos	herbi	Artiodactyla	domesticated	4.0	0.7	
...	...	...	...	...	...	...	...	
78	Tree shrew	Tupaia	omni	Scandentia	NaN	8.9	2.6	
79	Bottle-nosed dolphin	Tursiops	carni	Cetacea	NaN	5.2	NaN	
80	Genet	Genetta	carni	Carnivora	NaN	6.3	1.3	
81	Arctic fox	Vulpes	carni	Carnivora	NaN	12.5	NaN	
82	Red fox	Vulpes	carni	Carnivora	NaN	9.8	2.4	

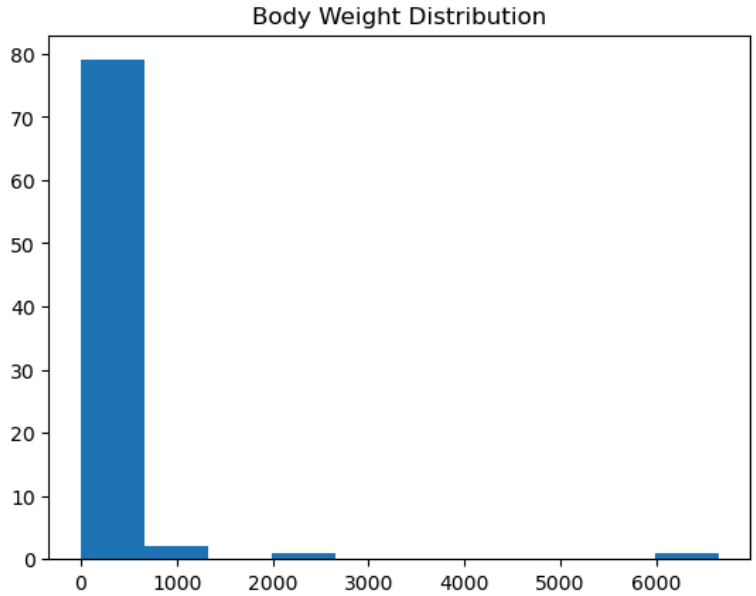
83 rows × 11 columns

In [7]:
```
import seaborn as sns
sns.scatterplot(x="bodywt", y="awake", data=df_sleep)
print(df_sleep["bodywt"].corr(df_sleep["awake"]))

# Burada her bir memelinin vücut ağırlığı ile her gün uyanık
# geçirdikleri sürenin dağılım grafiği yer almaktadır.
# Bu değişkenler arasındaki ilişki kesinlikle doğrusal bir ilişki
# değildir. Vücut ağırlığı ve uyanık kalma süresi arasındaki
# korelasyon
# sadece 0.3 civarındadır, bu da zayıf bir doğrusal ilişkidir.
```

```
0.31198014973502586
```

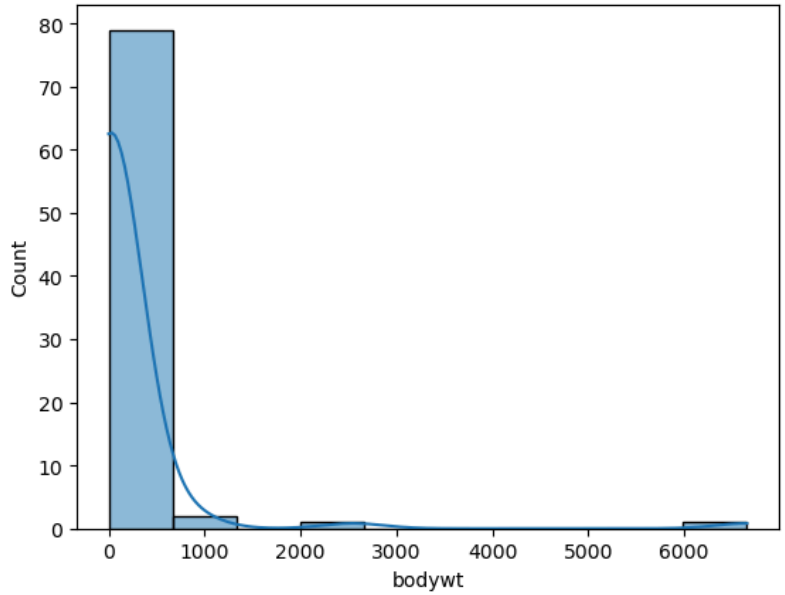


In [8]:
```
plt.hist(df_sleep["bodywt"])
plt.title("Body Weight Distribution")
plt.show()

# Vücut ağırlığı dağılımına daha yakından bakılırsa,
# oldukça çarpık olduğunu görülür. Çok sayıda düşük ağırlık ve
# diğerlerinden çok daha
# yüksek olan birkaç ağırlık vardır.
```



In [9]:
```
sns.histplot(data=df_sleep, x='bodywt', kde=True, bins=10)
```

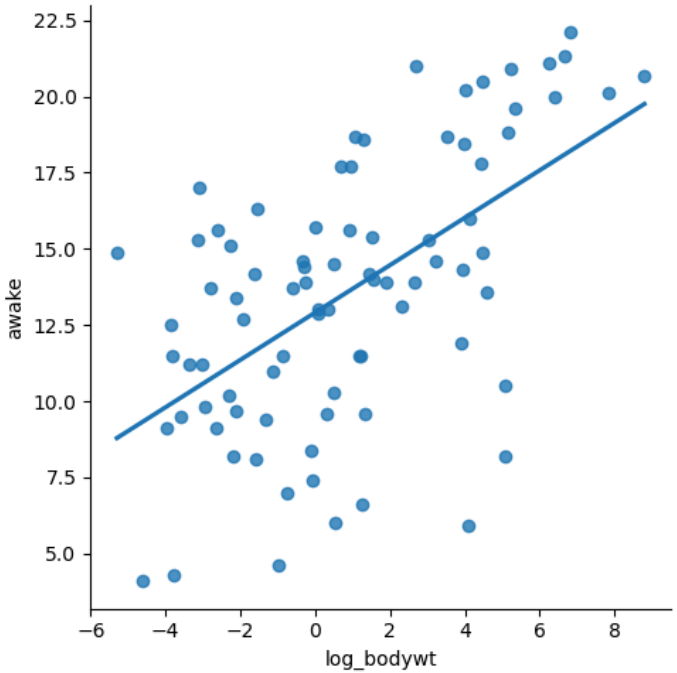
Out[9]: <Axes: xlabel='bodywt', ylabel='Count'>



In [8]:
```
# Log Transformation

# Veriler bu şekilde yüksek oranda çarpık olduğunda,
```

```
# bir log dönüşümü uygulanabilir.
# Her bir vücut ağırlığının logunu tutan log_bodywt adında
# yeni bir sütun oluşturulr. Bu np.log() kullanılarak yapılabilir. Vücut ağırlığ
# ilişki normal vücut ağırlığı ile uyanık kalma süresi arasındaki
# ilişkiden çok daha doğrusal görünür. Vücut ağırlığının logaritması ile
# uyanık kalma süresi arasındaki korelasyon yaklaşık 0.57'dir,
# bu da daha önce sahip olunan 0.31'den çok daha yüksektir.

import numpy as np

df_sleep["log_bodywt"] = np.log(df_sleep["bodywt"])

print(df_sleep["log_bodywt"].corr(df_sleep["awake"]))


sns.lmplot(x="log_bodywt", y="awake", data=df_sleep, ci=None)
plt.show()
print()
```

```
0.5687943427609856
```



Other Transformations

Log dönüşümüne ek olarak, bir ilişkiyi daha doğrusal hale getirmek için bir değişkenin karekökünü veya tersini almak gibi kullanılabilecek birçok başka dönüşüm vardır.

- Log Transformation (log(x))
- Square Root Transformation (sqrt(x))
- Reciprocal Transformation (1 / x)
- Bunların Kombinasyonları log(x) ve log(y) sqrt(x) ve 1 / y

Dönüşümün seçimi veriye ve ne kadar çarpık olduğuna bağlı olacaktır.

Transformation Neden Kullanılır?

Bazı istatistiksel yöntemler, korelasyon katsayısının hesaplanması gibi değişkenlerin doğrusal bir ilişkiye sahip olmasına dayanır.

Doğrusal regresyon, değişkenlerin doğrusal bir şekilde ilişkili olmasını gerektiren başka bir istatistiksel tekniktir.

Korelasyon nedensellik anlamına gelmez

Korelasyon nedensellik anlamına gelmez. Örneğin, burada ABD'de her yıl kişi başına düşen margarin tüketimi ile Maine eyaletindeki boşanma oranını gösteren bir dağılım grafiği yer almaktadır. Bu iki değişken arasındaki korelasyon 0.99'dur, yani neredeyse mükemmeldir. Ancak bu, daha fazla margarin tüketmenin daha fazla boşanmaya neden olacağı anlamına gelmez. **Bu tür bir korelasyon genellikle sahte korelasyon olarak adlandırılır.**


No description has been provided for this image

Confounding (Karıştırma)

Karıştırma adı verilen bir olgu sahte korelasyonlara yol açabilir. Diyelim ki kahve içmenin akciğer kanserine neden olup olmadığını bilmek istiyoruz. Verilere baktığımızda, kahve içme ve akciğer kanseri arasında korelasyon olduğunu görürüz; bu da bizi daha fazla kahve içmenin akciğer kanserine yol açacağını düşünmeye sevk edebilir.

Ancak, üçüncü ve gizli bir değişken daha vardır ki o da sigaradır.

Sigara içmenin kahve tüketimi ile ilişkili olduğu bilinmektedir.

Sigaranın akciğer kanserine neden olduğu da bilinmektedir.

Gerçekte, kahvenin akciğer kanserine neden olmadığı ve sadece onunla ilişkili olduğu, ancak üçüncü değişken olan sigara nedeniyle nedensel göründüğü ortaya çıkmıştır. Bu üçüncü değişkene karıştırıcı ya da gizlenen değişken denir. Bu da kahve ve akciğer kanseri arasındaki ilişkinin sahte bir korelasyon olduğu anlamına gelmektedir.

Bunun bir başka örneği de tatiller ve perakende satışlar arasındaki ilişkidir. Her ne kadar insanlar bayramlarda kutlama amacıyla daha fazla alışveriş yapıyor olsa da, satışlardaki artışın ne kadarının bayramlardan, ne kadarının ise bayramlarda yapılan özel fırsat ve promosyonlardan kaynaklandığını söylemek zordur. Burada, özel fırsatlar tatil ve satışlar arasındaki ilişkiyi karıştırmaktadır.


No description has been provided for this image

Interpreting Hypothesis Test Results

Chicago ve Bangkok'da Yaşam Süreleri

Chicago ve Bangkok'ta ortalama yaşam süreleri arasında bir fark olup olmadığının test edilmek istendiğini varsayalım.

- H0 = Chicago ve Bangkok'ta ortalama yaşam süreleri arasında bir fark yoktur.
- H1 = Chicago'da yaşayanlar Bangkok'ta yaşayanlardan daha uzun bir yaşam süresine sahiptir.

Örneklem Dağılımı

- Chicago ve Bangkok'ta yaşayan 100'er kişiden ölüm yaşına ilişkin veri toplansın.
- Bu histogram ile her bir şehir için yaşam beklentisi örnek dağılımlarını gösterilsin.
 No description has been provided for this image
- Chicago örnekleminin ortalama yaşam beklentisi 79.3, Bangkok için ise 73.9'dur.
- Ancak bunların her bir nüfus için gerçek ortalama değerler olduğunu nasıl bilinebilir?

Farklı Örneklemler

- Her iki şehirde yaşayan 100 kişiden daha ölüm yaşı verisi toplanabilir.
- Bu sefer farklı sonuçlar elde edilebilir.
 No description has been provided for this image
- Peki, yaşam beklentisinde gerçekten bir fark olduğundan emin miyiz, yoksa sonuçlar şansa mı bağlı?
- Başka bir deyişle, örnekler bu nüfusları gerçekten temsil ediyor mu?

Ortalama yaşam beklentisinin örnekleme dağılımı

- Tüm nüfus verilerini topanamaz, bu nedenle bir yaklaşım, her şehirden orijinal veriler üzerinde değiştirme ile örnekleme yapmak ve her örnek için ortalama yaşam beklentisini hesaplamaktır.
- Bu 10000 kez tekrarlanarak ve sonuçları görselleştirilerek, Bangkok ve Chicago'daki ortalama yaşam beklentisi için normal dağılımlar görülebilir ve Chicago daha büyük bir beklenen değere sahiptir!  No description has been provided for this image
- Öyleyse, artık yaşam beklentisinde gerçekten bir fark olduğu sonucuna varilabilir mi?

p-value

- Hipotez testlerinde sonuç çıkarırken p-değeri adı verilen bir ölçüt kullanılır.
- Bu, sıfır hipotezinin doğru olduğunu varsayarak en az gözlemlediğimiz kadar uç bir sonuç elde etme olasılığıdır.
- p-değeri = gözlediğiniz sonucun sadece şansla açıklanabilme ihtimali
- Diyelim ki, 79,3'lük bir popülasyon ortalaması göz önüne alındığında, Chicago yaşam beklentisi için örnek ortalamasının 82'ye eşit veya daha fazla olma olasılığını bilmek istiyoruz. Örneklem ortalamaları dağılımını görselleştirebilir ve 82'den itibaren toplam alana bakarak p-değerinin 0,037 olduğunu, yani ortalama yaşam beklentisinin 82 veya daha fazla olduğunu gözlemleme şansımızın yüzde 3,7 olduğunu belirleyebiliriz.

 No description has been provided for this image

 No description has been provided for this image

İki örnek ortalama dağılımı için p-değeri, aralarında örtüşen toplam alan olarak görselleştirilebili. Peki, sonuçtan emin olmak için ne kadar küçük bir örtüşme gerekir?

Significance level (α)

- Yanlış bir sonuca varma riskini azaltmak için, sıfır hipotezini yanlışlıkla reddetmek için bir olasılık eşiği belirlenir. Bu olasılık eşiği alfa veya anlamlılık düzeyi olarak bilinir.
- Önyargıyı en aza indirmek için veri toplanmadan önce karar verilir, çünkü bir araştırmacı verileri gördükten sonra kendi çıkarlarına hizmet eden bir sonuç çıkarmak için farklı bir eşik seçebilir.
- Bunun için tipik bir değer 0,05'tir, yani Chicago sakinlerinin Bangkok sakinlerinden daha uzun yaşadığı sonucuna yanlış bir şekilde varmak için yüzde beş şans vardır.
- Veri toplandıktan sonra, p-değerinin alfa değerinden küçük ya da eşit olup olmadığına bakılır. Eğer p-değeri bu kriteri karşılıyorsa, sıfır hipotezini reddetme konusunda güvenilebilir. Bu gerçekleşirse sonuçlar istatistiksel olarak anlamlı olarak tanımlanır.

Type I/II Error

- Yanlış olduğu halde sıfır hipotezi yanlışlıkla kabul edilebilir. Bu, ikinci tip hata olarak bilinir.
- Doğru olduğunda boş hipotezi yanlış bir şekilde reddedilebilir. Bu birinci tip hata olarak kabul edilir.

 No description has been provided for this image

Result

- Alfa değerini belirledikten sonra, artık örneklem ortalama dağılımlarına dayanarak bir sonuç çıkarılabilir.
- Dağılımların örtüşmesi, alfa eşiği olan 0.05'ten daha düşük bir değere karşılık gelmektedir; Bu da iki şehir arasındaki ortalama yaşam beklentisi farkının tesadüfen ortaya çıkma olasılığının %5'ten daha az olduğu anlamına gelmektedir.
- Dolayısıyla, sıfır hipotezini reddedebilir ve Chicago'daki ortalama yaşam süresinin Bangkok'tan daha yüksek olduğu sonucuna varılabilir!

In [ ]: