

```
import pandas as pd
df_animal = pd.read_csv("data/msleep.csv")
df_animal
```

	conservation	name	genus	vore	order
0	lc	Cheetah	Acinonyx	carni	Carnivora
1	NaN	Owl monkey	Aotus	omni	Primates
2	nt	Mountain beaver	Aplodontia	herbi	Rodentia
3	lc	Greater short-tailed shrew	Blarina	omni	Soricomorpha
4	domesticated	Cow	Bos	herbi	Artiodactyla
..	
78	NaN	Tree shrew	Tupaia	omni	Scandentia
79	NaN	Bottle-nosed dolphin	Tursiops	carni	Cetacea
80	NaN	Genet	Genetta	carni	Carnivora
81	NaN	Arctic fox	Vulpes	carni	Carnivora
82	NaN	Red fox	Vulpes	carni	Carnivora

	sleep_total	sleep_rem	sleep_cycle	awake	brainwt	bodywt
0	12.1	NaN	NaN	11.9	NaN	50.000
1	17.0	1.8	NaN	7.0	0.01550	0.480
2	14.4	2.4	NaN	9.6	NaN	1.350
3	14.9	2.3	0.133333	9.1	0.00029	0.019
4	4.0	0.7	0.666667	20.0	0.42300	600.000
..
78	8.9	2.6	0.233333	15.1	0.00250	0.104
79	5.2	NaN	NaN	18.8	NaN	173.330
80	6.3	1.3	NaN	17.7	0.01750	2.000
81	12.5	NaN	NaN	11.5	0.04450	3.380
82	9.8	2.4	0.350000	14.2	0.05040	4.230

[83 rows x 11 columns]

```
import numpy as np
np.mean(df_animal.sleep_total)
np.median(df_animal.sleep_total)
```

#sözel verilerde value.counts kullanarak mod bulabilirsiniz

```

df_animal.vore.value_counts()
#iki tane mod fonksiyonu var statistics kütüphanesi de kullanılabilir
#sadece 32 değerini görmek istersen ne yaparsın
#df_animal.vore.value_counts.index[0]
df_animal.vore.value_counts(normalize=True) #ondalık şeklinde verir
df_animal[df_animal.vore == "insecti"] #hayvanlar içerisinde böcek
yiyenleri listele
df_animal[df_animal.vore == "insecti"]
["sleep_rem"].agg(["mean","median"]) #aynı alanın birden fazla
fonksiyonu
#istiyorsak agg kullanılırız. insecti olanlardan sleepremlerinin mean
ve medianlarını bulduk
#df_animal.loc[len(df_animal.index)] = ["New Insect", "",
"insecti","",0.0,0.0,0.0]#dataframe yeni bir satır ekler

```

```

mean      3.525
median    3.000
Name: sleep_rem, dtype: float64

```

```

df_animal.loc[len(df_animal.index)] = ["New Insect", "",
"insecti","",0.0,0.0,0.0,0.0,0.0,0.0,0.0]#dataframe yeni bir satır
ekler

```

```
df_animal.tail()
```

	name	genus	vore	order	conservation \
79	Bottle-nosed dolphin	Tursiops	carni	Cetacea	NaN
80	Genet	Genetta	carni	Carnivora	NaN
81	Arctic fox	Vulpes	carni	Carnivora	NaN
82	Red fox	Vulpes	carni	Carnivora	NaN
83	New Insect		insecti		0.0

	sleep_total	sleep_rem	sleep_cycle	awake	brainwt	bodywt
79	5.2	NaN	NaN	18.8	NaN	173.33
80	6.3	1.3	NaN	17.7	0.0175	2.00
81	12.5	NaN	NaN	11.5	0.0445	3.38
82	9.8	2.4	0.35	14.2	0.0504	4.23
83	0.0	0.0	0.00	0.0	0.0000	0.00

```

import statistics as stat
stat.mode(df_animal.vore)

```

```
'herbi'
```

```

#eksik değerleri nasıl dolduracağız merkez ölçüsü olarak ne alacağız
range_sleep_total = df_animal["sleep_total"].max() -
df_animal["sleep_total"].min()
range_sleep_total

```

```
18.0
```

```

np.var(df_animal.sleep_total,ddof=1) #varyans. ortalamadan sapmanın
büyüklüğünü ifade eder
np.sqrt(np.var(df_animal.sleep_total,ddof=1)) #kök
#standart sapma
np.std(df_animal.sleep_total,ddof=1)

#mean absolute deviation, ortalamadan sapması
dists=df_animal.sleep_total-np.mean(df_animal.sleep_total)
np.mean(np.abs(dists)) #ortalama mutlak sapma

3.5667005370881117

np.var(df_animal.sleep_total,ddof=1)

19.805677343520422

#Quantile, bir veri setini belirli yüzdelik dilimlere bölme işlemidir
#Veri setini sıralama: Veriler küçükten büyüğe sıralanır.
    #2. Pozisyonu (indisi) hesaplama:
    #• Quantile formülü şu şekildedir:

#{Pozisyon} = p \times (n - 1)

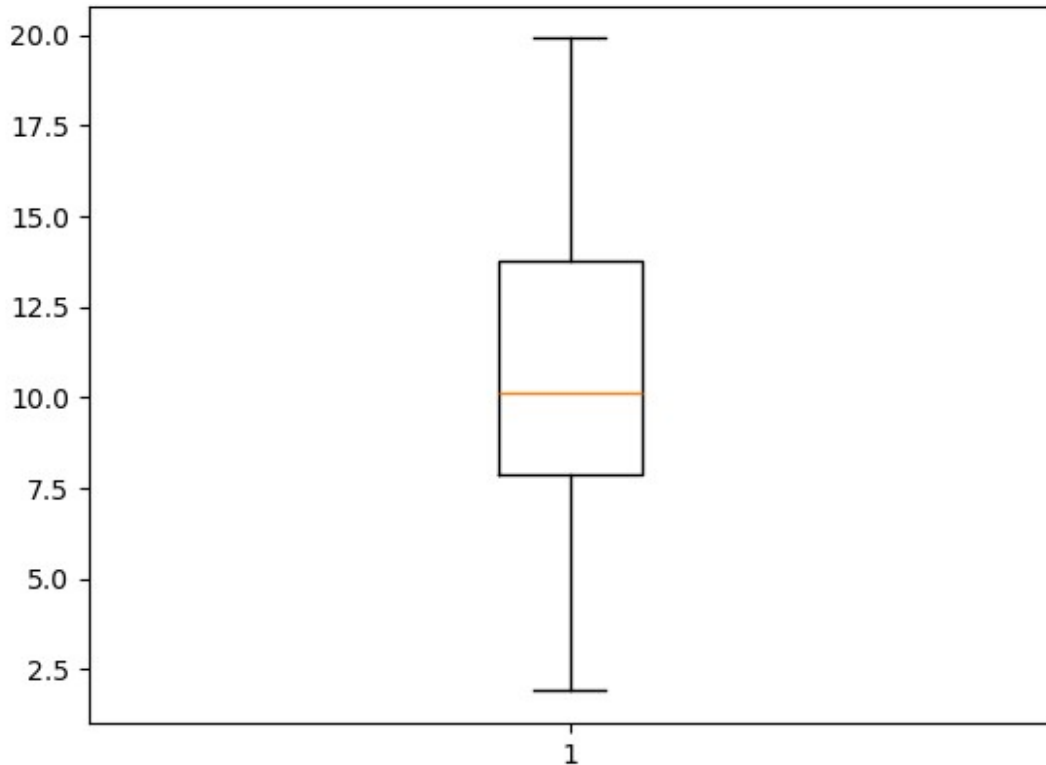
#burada:
# • p : Quantile yüzdesi (örneğin, medyan için p = 0.5 ,
çeyrekler için p = 0.25 ve p = 0.75 ).
# • n : Veri setindeki toplam eleman sayısı.
np.quantile(df_animal.sleep_total,0.5) #mediandır çünkü yüzde 50lik
quantile mediandır
np.quantile(df_animal.sleep_total, ([0,0.25,0.50,0.75,1]))
np.quantile(df_animal.sleep_total,np.linspace(0,1,5)) #verilerin bir
aralığı istediğiniz kadar eşit parçaya bölmenizi sağlar. 0 ile 1
arasında eşit aralıklarla 5 nokta oluşturur. Bu noktalar: [0, 0.25,
0.50, 0.75, 1] .

array([ 1.9 ,  7.85, 10.1 , 13.75, 19.9 ])

#boxplot çizme. Boxplot veri dağılımını ve aykırı değerleri
görselleştiren etkili bir grafik türüdür
import matplotlib.pyplot as plt
plt.boxplot(df_animal.sleep_total)

{'whiskers': [<matplotlib.lines.Line2D at 0x148e3f380>,
<matplotlib.lines.Line2D at 0x148e3f680>],
'caps': [<matplotlib.lines.Line2D at 0x148e3f950>,
<matplotlib.lines.Line2D at 0x148e3fc50>],
'boxes': [<matplotlib.lines.Line2D at 0x148e3f080>],
'medians': [<matplotlib.lines.Line2D at 0x148e3ff50>],
'fliers': [<matplotlib.lines.Line2D at 0x148e78290>],
'means': []}

```



```
import numpy as np
from scipy.stats import iqr
# aykırı değerlerin belirlenmesi için kullanılan IQR (interquartile
range) yöntemini kullanarak df_animal DataFrame'inde bodywt (vücut
ağırlığı) sütunundaki aykırı değerleri bulmayı amaçlıyo
# Q3 ve Q1 hesaplama
q3 = np.quantile(df_animal.sleep_total, 0.75)
q1 = np.quantile(df_animal.sleep_total, 0.25)

# IQR hesaplama
iqr_value = iqr(df_animal.bodywt) # • IQR, verilerin yayılımını
ölçmek için kullanılır. Özellikle aykırı değerleri belirlemede çok
yararlıdır.

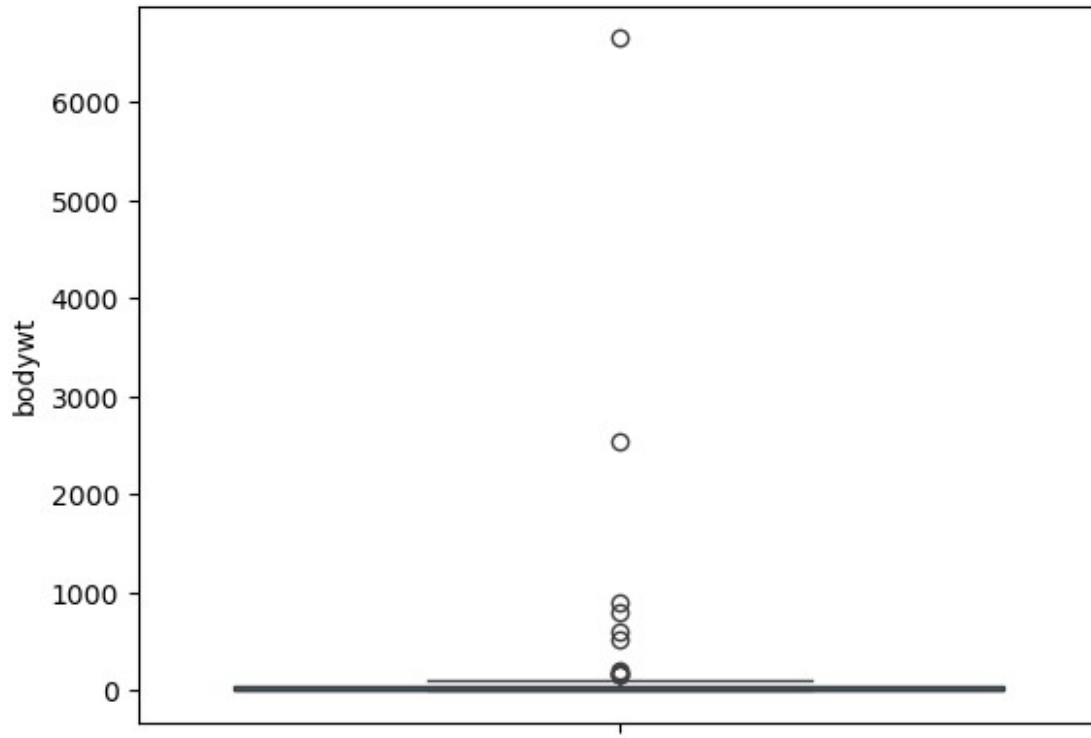
# Alt ve üst eşiklerin hesaplanması. Bu eşikler, verilerin makul
aralıklarını belirler. Veriler bu aralıkların dışında kaldığında
aykırı olarak değerlendirilir.
lower_threshold = np.quantile(df_animal.bodywt, 0.25) - 1.5 *
iqr_value
upper_threshold = np.quantile(df_animal.bodywt, 0.75) + 1.5 *
iqr_value

# Aykırı değerlerin filtrelenmesi
df_animal[(df_animal.bodywt < lower_threshold) | (df_animal.bodywt >
upper_threshold)]
```

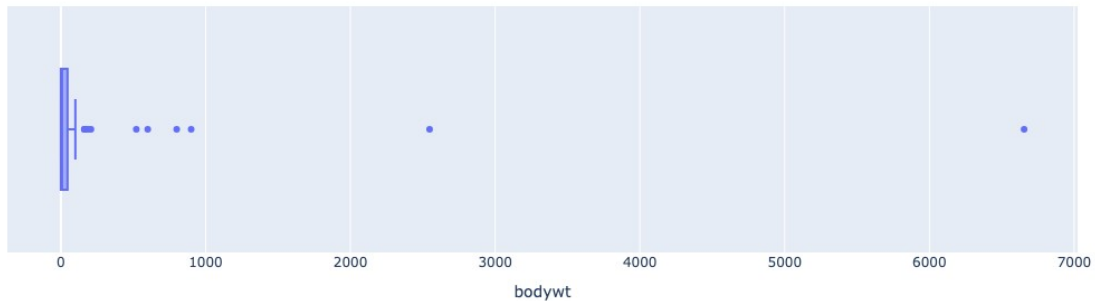
	name	genus	vore	order		
conservation \						
4	Cow	Bos	herbi	Artiodactyla		
domesticated						
20	Asian elephant	Elephas	herbi	Proboscidea		
en						
22	Horse	Equus	herbi	Perissodactyla		
domesticated						
23	Donkey	Equus	herbi	Perissodactyla		
domesticated						
29	Giraffe	Giraffa	herbi	Artiodactyla		
cd						
30	Pilot whale	Globicephalus	carni	Cetacea		
cd						
35	African elephant	Loxodonta	herbi	Proboscidea		
vu						
50	Tiger	Panthera	carni	Carnivora		
en						
52	Lion	Panthera	carni	Carnivora		
vu						
76	Brazilian tapir	Tapirus	herbi	Perissodactyla		
vu						
79	Bottle-nosed dolphin	Tursiops	carni	Cetacea		
NaN						
	sleep_total	sleep_rem	sleep_cycle	awake	brainwt	bodywt
4	4.0	0.7	0.666667	20.00	0.423	600.000
20	3.9	NaN	NaN	20.10	4.603	2547.000
22	2.9	0.6	1.000000	21.10	0.655	521.000
23	3.1	0.4	NaN	20.90	0.419	187.000
29	1.9	0.4	NaN	22.10	NaN	899.995
30	2.7	0.1	NaN	21.35	NaN	800.000
35	3.3	NaN	NaN	20.70	5.712	6654.000
50	15.8	NaN	NaN	8.20	NaN	162.564
52	13.5	NaN	NaN	10.50	NaN	161.499
76	4.4	1.0	0.900000	19.60	0.169	207.501
79	5.2	NaN	NaN	18.80	NaN	173.330

```
import seaborn as sns
#boxplotu seaborn kullanarak yaptık, veri noktalarını vermiyor
outliner değerleri yok bunun alttaki kütüphane kullanılır
sns.boxplot(data=df_animal, y="bodywt")
```

```
<Axes: ylabel='bodywt'>
```



```
import plotly.express as px
px.box(df_animal, x="bodywt")
```



```
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") # agg(sum="sum"): amount sütunundaki
değerleri, her grup için topluyorsun. Yani, her num_users için, amount
değerlerinin toplamı hesaplanıyor.
df_sales_users
```

```
sum
num_users
```

```
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 x 1 columns]
```

```
df_sales_users.sample() #içinden rastgele bir örnek seçer
```

```
import numpy as np
np.random.seed(42) # her çalıştığında aynı değerin gelmesini istersen
seed'i set etmelisin. Yani bu işlemi yaptıktan sonra sample()
fonksiyonunu her çalıştırdığında, aynı rastgele satırın seçildiğini
göreceksin.
```

```
df_sales_users.sample()
```

```
          sum
num_users
33      7077.48
```

```
df_sales_users.sample(5,replace=True) #seçtiğin örneği yerine
koyabilirsin replace diyerek
```

```
          sum
num_users
68      11225.36
43      12319.18
31      15163.88
15       4179.72
48      13355.00
```

1. CSV Dosyasını Okuma:

python

Kodu kopyala

```
import pandas as pd
df_animal = pd.read_csv("data/msleep.csv")
```

Bu satırda msleep.csv dosyasını Pandas kullanarak bir DataFrame olarak yüklüyorsun. Bu doğru.

2. NumPy Import:

python

Kodu kopyala

```
df_animal import numpy as np
```

Bu satırda bir hata var. Burada `import` yanlış bir şekilde yazılmış. Düzeltilmesi gereken kısım:

```
python
Kodu kopyala
import numpy as np
3. Ortalama ve Medyan Hesaplama:
```

```
python
Kodu kopyala
np.mean(df_animal.sleep_total)
np.median(df_animal.sleep_total)
Bu iki satırda sleep_total kolonunun ortalama ve medyanını
hesaplıyorsun. NumPy ile doğru bir şekilde yapılmış.
```

4. Sözel Verilerde Mod Bulma:

```
python
Kodu kopyala
df_animal.vore.value_counts()
Bu satır, vore kolonundaki kategorik değerlerin frekansını saymak için
kullanılır. Doğru.
```

```
python
Kodu kopyala
# iki tane mod fonksiyonu var statistics kütüphanesi de kullanılabilir
Açıklama doğru. Hem Pandas ile value_counts() fonksiyonunu kullanarak
mod bulunabilir hem de statistics modülüyle yapılabilir.
```

5. Belirli Bir Değerin Frekansını Görüntüleme:

```
python
Kodu kopyala
df_animal.vore.value_counts.index[0]
Bu satırda bir hata var, yanlış bir şekilde yazılmış. Doğru hali şu
şekilde olmalı:
```

```
python
Kodu kopyala
df_animal.vore.value_counts().index[0]
Bu, vore kolonunda en çok tekrar eden değeri döndürür.
```

6. Normalize Edilmiş Değer Sayısı:

```
python
Kodu kopyala
df_animal.vore.value_counts(normalize=True)
Bu satır, vore kolonundaki frekansları normalize edilmiş (yüzdelik)
olarak döndürür. Doğru.
```

7. Belirli Kategoriyi Filtreleme:

```
python
Kodu kopyala
df_animal[df_animal.vore == "insecti"]
```


Bu satır, vore kolonunda "insecti" olan satırları filtreler. Doğru. Ancak insecti doğru yazıldığından emin ol (büyük küçük harf hatası olabilir).

8. Birden Fazla İstatistiksel Değer Hesaplama:

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

Bu satırda, böcek yiyen hayvanların sleep_rem kolonundaki verileri için hem ortalama hem de medyan hesaplanıyor. Doğru.

9. DataFrame'e Yeni Satır Ekleme:

```
df_animal.loc[len(df_animal.index)] = ["New Insect", "", "insecti",  
"", 0.0, 0.0, 0.0]
```

Bu satırda DataFrame'e yeni bir satır ekleniyor. Doğru. Ancak, eklerken dikkat edilmesi gereken kolon isimlerinin sırasına ve eksik değerler için uygun boşlukları kullanmak.

10. Mod Bulma:

```
import statistics as stat  
stat.mode(df_animal.vore)
```

Bu, statistics modülü kullanarak vore kolonundaki mod değerini bulur. Doğru.

11. Range (Aralık) Hesaplama:

```
range_sleep_total = df_animal["sleep_total"].max() -  
df_animal["sleep_total"].min()
```

Bu satırda sleep_total kolonunun maksimum ve minimum değerleri arasındaki fark hesaplanıyor (range). Doğru.

12. Varyans ve Standart Sapma:

```
np.var(df_animal.sleep_total, ddof=1)  
np.sqrt(np.var(df_animal.sleep_total, ddof=1))  
np.std(df_animal.sleep_total, ddof=1)
```

Bu satırlarda varyans, varyansın karekökü (standart sapma) ve direkt standart sapma hesaplanıyor. Burada ddof=1 kullanımı doğru çünkü bu örnek varyans ve standart sapmayı hesaplamak için gereklidir.

13. Ortalama Mutlak Sapma (MAD):

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

Bu satırda ortalama mutlak sapma hesaplanıyor. Doğru ve önemli bir istatistiksel ölçüdür.

14. Çeyrek Değerler (Quantile):

```
np.quantile(df_animal.sleep_total, 0.5)
np.quantile(df_animal.sleep_total, [0, 0.25, 0.50, 0.75, 1])
```

Bu satırlarda %50'lik quantile (medyan) ve diğer çeyrek değerler hesaplanıyor. Doğru.

15. Eşit Parçalara Bölme:

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

Bu satırda sleep_total verilerini 5 eşit parçaya bölersin. Doğru.

16. Boxplot Çizme:

```
import matplotlib.pyplot as plt
plt.boxplot(df_animal.sleep_total)
```

Bu satırda sleep_total için bir boxplot (kutu grafiği) çiziliyor. Doğru.

17. IQR (Interquartile Range) Hesaplama ve Aykırı Değerleri Filtreleme:

```
from scipy.stats import iqr
q3 = np.quantile(df_animal.sleep_total, 0.75)
q1 = np.quantile(df_animal.sleep_total, 0.25)
iqr_value = iqr(df_animal.bodywt)
lower_threshold = np.quantile(df_animal.bodywt, 0.25) - 1.5 *
iqr_value
upper_threshold = np.quantile(df_animal.bodywt, 0.75) + 1.5 *
iqr_value
df_animal[(df_animal.bodywt < lower_threshold) | (df_animal.bodywt >
upper_threshold)]
```

Bu satırlarda IQR hesaplanıyor ve aykırı değerleri belirlemek için alt ve üst eşikler bulunuyor. Sonrasında bu eşiklerin dışındaki aykırı değerler filtreleniyor. Doğru.

18. Plotly ile Boxplot:

```
import plotly.express as px
px.box(df_animal, x="bodywt")
```

Bu satır Plotly kullanarak bodywt için bir kutu grafiği çizer. Doğru.

19. amir_deals.csv Dosyasını Okuma:

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

Bu satırlarda amir_deals.csv dosyasından veriler okunup num_users bazında amount sütunlarının toplamı hesaplanıyor. Doğru.

20. Rastgele Örnek Seçme:

```
df_sales_users.sample()
```

Bu satır, verilerden rastgele bir örnek seçer. Doğru.

21. Örnekleme ve Seed Ayarlama:

```
np.random.seed(42)
```

```
df_sales_users.sample(5, replace=True)
```

Bu satırlarda rastgele örnekleme yaparken aynı sonuçları almak için seed ayarlanıyor ve replace=True ile örneklerin tekrar seçilmesine izin veriliyor. Doğru.

Eksik Olanlar:

Kovaryans: İki değişken arasındaki ilişkinin yönünü gösteren bir ölçü olabilir.

```
np.cov(df_animal.sleep_total, df_animal.sleep_rem)
```

Korelasyon: İki değişken arasındaki ilişkinin gücünü ve yönünü gösterebilir. Özellikle sleep_total ve sleep_rem gibi değişkenler için ilginç olabilir.

```
df_animal[['sleep_total', 'sleep_rem']].corr()
```

Z-Skoru Hesaplama: Aykırı değerleri daha iyi tespit etmek için z-skoru kullanılabilir.

```
from scipy import stats
```

```
z_scores = stats.zscore(df_animal.sleep_total)
```

```
df_animal[(z_scores > 3) | (z_scores < -3)]
```