

```

first_number = 45
second_number = 32

first_number + second_number #böyle de çalışır
print(f'{first_number} + {second_number} = {first_number +
second_number}')

45 + 32 = 77

import pandas as pd

pd.read_csv('data/msleep.csv') #birden fazla satır varsa dataframe döner!

```

	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]

```
df_animal = pd.read_csv('data/msleep.csv')
df_animal  #okuma işlemini göstermek gerekir
df_animal.head() #ilk 5 satırı gösterir. parantezin içine sayı
              verilerek değiştirilebilir bu sayı
df_animal.tail() #sondan ilk 5 satırı gösterir. parantezin içine sayı
              verilerek değiştirilebilir bu sayı
df_animal.columns #dataframedeki columnların adını verir. propertydir
df_animal.dtypes #her bir sütunun hangi tip olduğunu seri olarak
                 verir. seri tek satırdır, dataframe birden fazla satırdan oluşur.
                 object -> string demektir
df_animal.shape #dfnin kaç satır ve sütundan oluştuğunu. tuple ile
                 döndürür, tuple değiştirilmez!
# df_animal.shape[0,1] #0. satır 1. sütundaki veriyi ver
df_animal.info() #boş kayıtlara sahip olup olmadığını verir, ramde ne
                 kadar yer kapladığına vs, hangi türde veriler vs
# df_animal.description #sayısal kısımlardaki özet istatistikleri
                 verir. ort vs. transpoz kısmını kaçırdım
df_animal.describe().transpose()
```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 83 entries, 0 to 82

Data columns (total 11 columns):

#	Column	Non-Null Count	Dtype
0	name	83 non-null	object
1	genus	83 non-null	object
2	vore	76 non-null	object
3	order	83 non-null	object
4	conservation	54 non-null	object
5	sleep_total	83 non-null	float64
6	sleep_rem	61 non-null	float64
7	sleep_cycle	32 non-null	float64
8	awake	83 non-null	float64
9	brainwt	56 non-null	float64
10	bodywt	83 non-null	float64

dtypes: float64(6), object(5)

memory usage: 7.3+ KB

	count	mean	std	min	25%
50% \					
sleep_total	83.0	10.433735	4.450357	1.900000	7.850000
10.100000					
sleep_rem	61.0	1.875410	1.298288	0.100000	0.900000
1.500000					
sleep_cycle	32.0	0.439583	0.358680	0.116667	0.183333
0.333333					

```

awake      83.0    13.567470    4.452085    4.100000    10.250000
13.900000
brainwt    56.0     0.281581    0.976414    0.000140    0.002900
0.012400
bodywt     83.0   166.136349   786.839732    0.005000    0.174000
1.670000

```

```

              75%      max
sleep_total  13.750000   19.900
sleep_rem    2.400000    6.600
sleep_cycle  0.579167    1.500
awake        16.150000   22.100
brainwt      0.125500    5.712
bodywt       41.750000  6654.000

```

```

df_animal.genus #isimlerin arasında boşluk yoksa bu kullanılır. seri döndürür!
df_animal['genus']
df_animal[['name', 'genus']] #birden fazla satır olduğu için bu df olarak döner
df_animal['name'].str.len() #name sütunundaki her bir elemanın isminin uzunluğunu döndürür
df_animal[df_animal['vore'] == 'herbi'] #vore değeri herbi olanları verir. where gibi yani
df_animal[(df_animal['vore'] == 'herbi') & (df_animal['sleep_total'] > 9)] # VE

```

	name	genus	vore	order
2	Mountain beaver	Aplodontia	herbi	Rodentia
5	Three-toed sloth	Bradypus	herbi	Pilosa
11	Guinea pig	Cavis	herbi	Rodentia
13	Chinchilla	Chinchilla	herbi	Rodentia
26	Western american chipmunk	Eutamias	herbi	Rodentia
34	Mongoose lemur	Lemur	herbi	Primates
38	Mongolian gerbil	Meriones	herbi	Rodentia
39	Golden hamster	Mesocricetus	herbi	Rodentia
40	Vole	Microtus	herbi	Rodentia
41	House mouse	Mus	herbi	Rodentia
43	Round-tailed muskrat	Neofiber	herbi	Rodentia

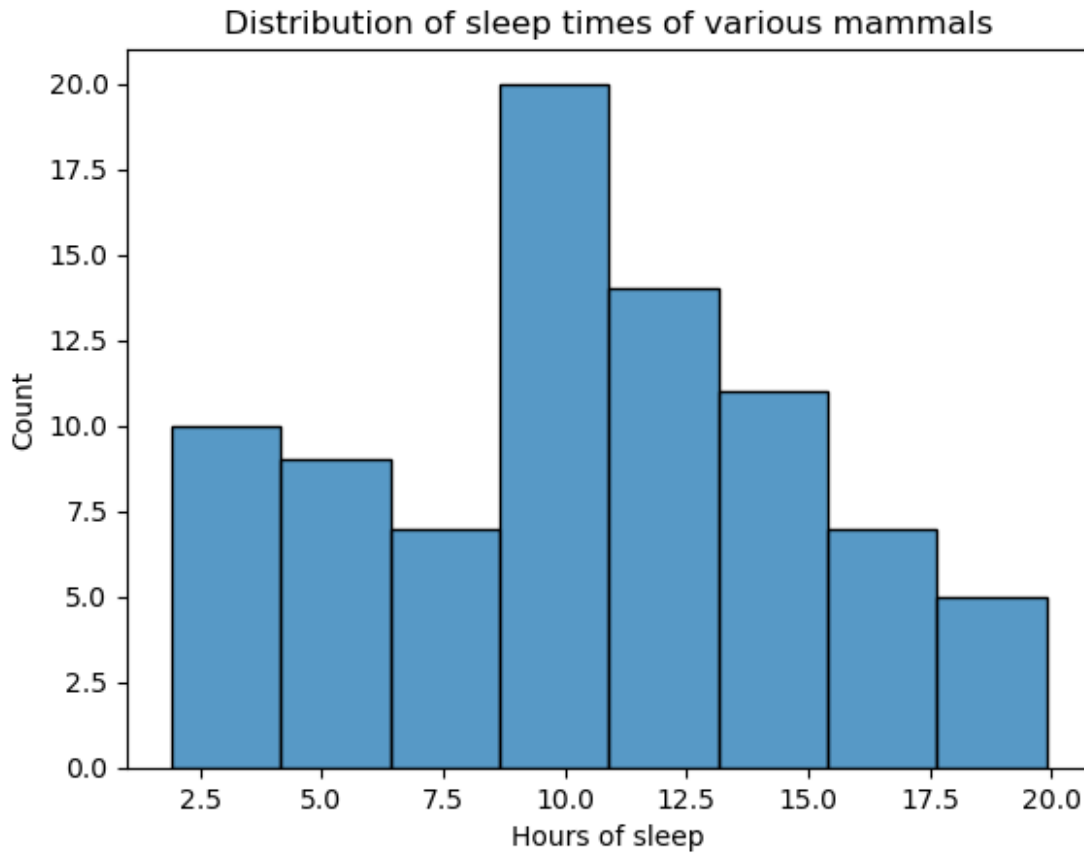
60		Potoroo	Potorous	herbi	Diprotodontia	
63		Laboratory rat	Rattus	herbi	Rodentia	
67		Cotton rat	Sigmodon	herbi	Rodentia	
69		Arctic ground squirrel	Spermophilus	herbi	Rodentia	
70	Thirteen-lined ground squirrel	Spermophilus	herbi	Rodentia		
71	Golden-mantled ground squirrel	Spermophilus	herbi	Rodentia		
75		Eastern american chipmunk	Tamias	herbi	Rodentia	
	conservation	sleep_total	sleep_rem	sleep_cycle	awake	brainwt
	bodywt					
2	nt	14.4	2.4	NaN	9.6	NaN
1.350						
5	NaN	14.4	2.2	0.766667	9.6	NaN
3.850						
11	domesticated	9.4	0.8	0.216667	14.6	0.00550
0.728						
13	domesticated	12.5	1.5	0.116667	11.5	0.00640
0.420						
26	NaN	14.9	NaN	NaN	9.1	NaN
0.071						
34	vu	9.5	0.9	NaN	14.5	NaN
1.670						
38	lc	14.2	1.9	NaN	9.8	NaN
0.053						
39	en	14.3	3.1	0.200000	9.7	0.00100
0.120						
40	NaN	12.8	NaN	NaN	11.2	NaN
0.035						
41	nt	12.5	1.4	0.183333	11.5	0.00040
0.022						
43	nt	14.6	NaN	NaN	9.4	NaN
0.266						
60	NaN	11.1	1.5	NaN	12.9	NaN
1.100						
63	lc	13.0	2.4	0.183333	11.0	0.00190
0.320						
67	NaN	11.3	1.1	0.150000	12.7	0.00118
0.148						
69	lc	16.6	NaN	NaN	7.4	0.00570
0.920						
70	lc	13.8	3.4	0.216667	10.2	0.00400
0.101						
71	lc	15.9	3.0	NaN	8.1	NaN

```
0.205
75      NaN      15.8      NaN      NaN      8.2      NaN
0.112
```

```
df_animal['vore'].value_counts() #hangi beslenme türünden kaç tane.
seri döndürür. groupby'ya karşılık gelir!
df_animal.groupby('vore')['vore'].count() #bana voreları grupta,
voreları sayarak ver. sum olabilir count olabilir avg olabilir. nan
değerleri vermez
df_animal['vore'].unique() #unique olarak hangi değerler varsa verir.
nanları verir bu
df_animal['vore'].nunique(dropna=False) #nanları vermez. dropna değeri
nanları siler, false verirsene gösterir
```

5

```
import matplotlib.pyplot as plt
import seaborn as sns
# Histogram, verilerin dağılımını görselleştirmek için kullanılan bir
grafiktir. Verileri sınıf aralıklarına (bins) böler ve her bir
aralıkta kaç veri olduğunu gösterir. Histogram, sürekli veri
dağılımını analiz etmek için kullanılır ve genellikle verilerin
yoğunlaştığı bölgeleri ve dağılımın genel yapısını görselleştirmede
etkilidir.
sns.histplot(df_animal.sleep_total)
plt.title("Distribution of sleep times of various mammals")
plt.xlabel("Hours of sleep")
Text(0.5, 0, 'Hours of sleep')
```



```
# ortalama(mean)
import numpy as np

np.mean(df_animal.sleep_total)
df_animal['sleep_total'].mean() #böyle de olur

10.433734939759034

#medyan. değerlerde uç değer varsa ort yeni medyan daha kullanışlıdır
np.median(df_animal.sleep_total)

10.1

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

'herbi'

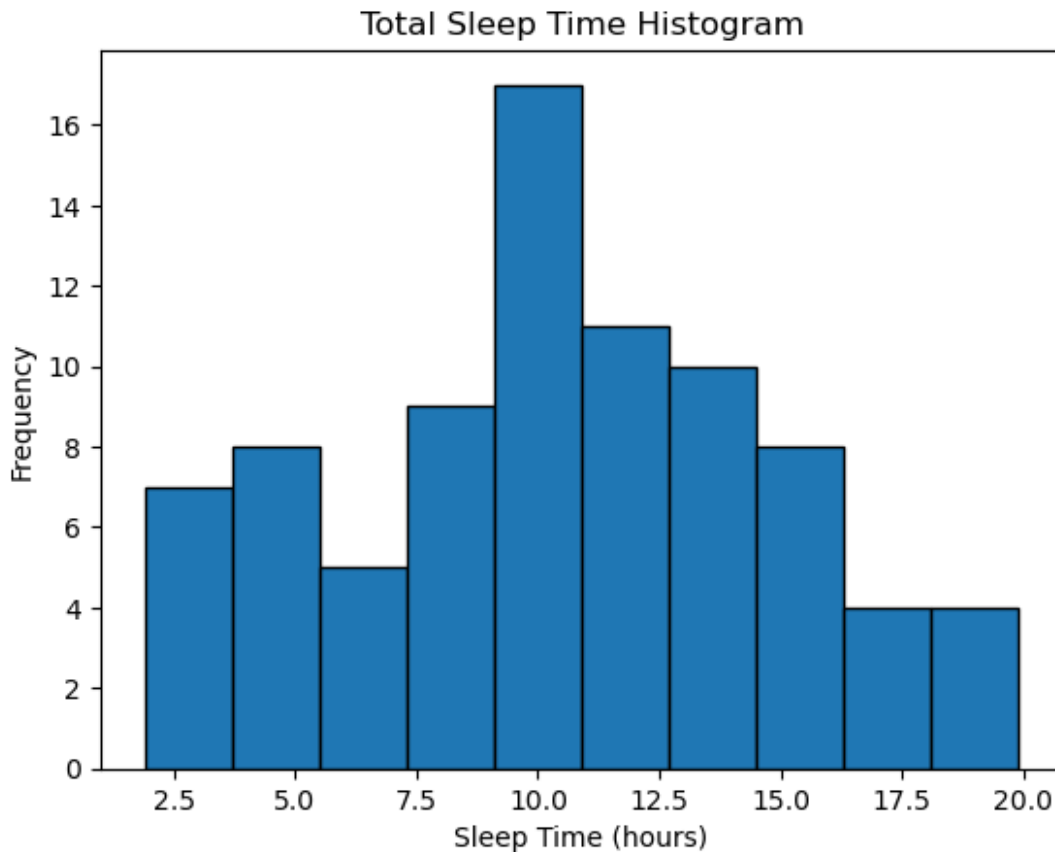
#ödev
import matplotlib.pyplot as plt

# Histogram oluşturmak için 'sleep_total' sütununu seç
sleep_data = df_animal['sleep_total']
```

```
# Histogramı oluştur, bins=10 ile verileri 10 aralığa (bin) böleceğiz
plt.hist(sleep_data.dropna(), bins=10, edgecolor='black')

# Grafiğe başlık ve etiketler ekle
plt.title('Total Sleep Time Histogram')
plt.xlabel('Sleep Time (hours)')
plt.ylabel('Frequency')

# Histogramı göster
plt.show()
```



1. CSV Dosyasını Okuma

```
df_animal = pd.read_csv('data/msleep.csv')
df_animal
```

- `pd.read_csv()` fonksiyonu ile `msleep.csv` dosyasını bir pandas DataFrame'e yüklüyorsun.
- `df_animal` ifadesi ile DataFrame'i ekrana basıyorsun. Bu işlem veri yüklemenin başarılı olup olmadığını görmeni sağlar.

2. İlk 5 Satırı Gösterme

```
df_animal.head()
```

- `head()` fonksiyonu, veri çerçevesinin ilk 5 satırını döndürür. İçine sayı koyarsan, o kadar satırı getirir. Örneğin, `df_animal.head(10)` ilk 10 satırı getirir.

3. Son 5 Satırı Gösterme

```
df_animal.tail()
```

- `tail()` fonksiyonu son 5 satırı gösterir. Yine, paranteze bir sayı koyarak son kaç satırı görmek istediğini belirtebilirsin.

4. Sütun İsimlerini Listeleme

```
df_animal.columns
```

- `columns` özelliği, veri çerçevesindeki sütun isimlerini bir liste olarak döndürür. Bu bilgi, hangi sütunların mevcut olduğunu görmeni sağlar.

5. Sütunların Veri Tiplerini Gösterme

```
df_animal.dtypes
```

- `dtypes`, her sütunun veri tipini gösterir. Bu sayede hangi sütunların sayısal (`int`, `float`), hangilerinin metinsel (`object` tipi) olduğunu anlayabilirsin.

6. DataFrame'in Boyutlarını Öğrenme

```
df_animal.shape
```

- `shape`, satır ve sütun sayısını bir `tuple` (örneğin (N, M)) olarak döndürür. (100, 8) gibi bir sonuç alırsan, bu 100 satır ve 8 sütun olduğunu gösterir.

7. Veri Bilgilerini Detaylı Gösterme

```
df_animal.info()
```

- Bu fonksiyon, veri çerçevesinin genel yapısını gösterir: sütun isimleri, veri tipleri, boş değerler (NaN), ve RAM'de ne kadar yer kapladığını.

8. Sayısal Verilerin Özet İstatistiklerini Gösterme

```
df_animal.describe().transpose()
```

- `describe()` sayısal sütunlar için ortalama (mean), standart

sapma (std), minimum (min), maksimum (max) ve yüzde birimleri (percentiles) gibi özet istatistikleri verir.

- transpose() ile satır ve sütunları yer değiştirerek daha okunaklı hale getirebilirsin.

9. Sütuna Erişim

```
df_animal.genus  
df_animal['genus']
```

- df_animal.genus ve df_animal['genus'] aynı işlemi yapar, genus sütununu alır. Ancak boşluk veya özel karakter içeren sütunlar için df_animal['genus'] yöntemi daha güvenlidir.

10. Birden Fazla Sütunu Seçme

```
df_animal[['name', 'genus']]
```

- Çift köşeli parantez ile birden fazla sütunu seçiyorsun. Bu işlem, bir DataFrame döndürür.

11. String Uzunluğu Bulma

```
df_animal['name'].str.len()
```

- name sütunundaki her bir string ifadenin uzunluğunu hesaplar.

12. Koşullu Filtreleme (Where)

```
df_animal[df_animal['vore'] == 'herbi']
```

- Bu filtreleme işlemi, vore sütununda 'herbi' (otçul) olanları getirir.

13. Birden Fazla Koşulla Filtreleme

```
df_animal[(df_animal['vore'] == 'herbi') & (df_animal['sleep_total'] > 9)]
```

- Burada iki koşul birden var: vore sütunu 'herbi' olmalı ve sleep_total değeri 9'dan büyük olmalı. & operatörü VE anlamına gelir.

14. Kategorik Değerlerin Sayısını Bulma

```
df_animal['vore'].value_counts()
```

- vore sütunundaki kategorik değerlerin kaç kez tekrarlandığını sayar. Örneğin, kaç tane otçul, kaç tane etçil olduğunu gösterir.

15. Gruplayarak Sayma

```
df_animal.groupby('vore')['vore'].count()
```

- `groupby()`, vore sütununa göre gruplama yapar ve her gruptaki öğeleri sayar. Nan değerler hariç tutulur.

16. Benzersiz Değerleri Bulma

```
df_animal['vore'].unique()
```

- `unique()`, vore sütunundaki benzersiz değerleri verir. Nan değerler de gösterilir.

17. Benzersiz Değer Sayısını Bulma

```
df_animal['vore'].nunique(dropna=False)
```

- Bu, Nan dahil olmak üzere kaç farklı benzersiz değer olduğunu döndürür. `dropna=False` ile Nan'ları hariç tutabilirsiniz.

18. Histogram Çizimi (Seaborn)

```
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")
plt.show()
```

- Bu kodla, `sleep_total` sütununa ait histogram oluşturuluyor. Histogram, sütun verisinin dağılımını görselleştirir. Başlık ve eksen adları da eklenmiş.

19. Ortalama ve Medyan Hesaplama

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

- `mean()` ortalama değer, `median()` ise medyan değeri hesaplar. Medyan, uç değerlerden etkilenmez, bu nedenle bazen ortalamadan daha kullanışlıdır.

20. Mod (En Çok Tekrar Eden Değer) Hesaplama

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

- `mode()`, en çok tekrar eden değeri bulur. Kategorik veriler

için özellikle faydalıdır.

21. Matplotlib ile Histogram Çizimi

```
import matplotlib.pyplot as plt
```

```
sleep_data = df_animal['sleep_total']  
plt.hist(sleep_data.dropna(), bins=10, edgecolor='black')  
plt.title('Total Sleep Time Histogram')  
plt.xlabel('Sleep Time (hours)')  
plt.ylabel('Frequency')  
plt.show()
```

- Bu kod, sleep_total sütunundaki veriler için matplotlib kullanarak bir histogram çizer. bins=10 veriyi 10 aralığa böler.

Ek Olasılıklar ve Alternatifler

1. Eksik Verileri Doldurma: Eğer veri çerçevesinde eksik (NaN) değerler varsa, bunları doldurabilirsiniz.

```
df_animal.fillna(value)
```

2. Veri Tipi Dönüştürme:

```
df_animal['sleep_total'] = df_animal['sleep_total'].astype('float')
```

3. Sütun Ekleme: Yeni bir sütun oluşturmak istersen:

```
df_animal['new_column'] = df_animal['sleep_total'] / 2
```

4. Gruplama ile Toplam, Ortalama Bulma:

```
df_animal.groupby('vore')['sleep_total'].mean()
```

5. Veri Setini Sıralama:

```
df_animal.sort_values(by='sleep_total', ascending=False)
```

```
df_animal.groupby('vore')['sleep_total'].mean()
```

```
vore  
carni    10.378947  
herbi     9.509375  
insecti  14.940000
```

```
omni      10.925000  
Name: sleep_total, dtype: float64
```