

Kütüphaneler

```
In [2]: !pip3 install numpy
```

Requirement already satisfied: numpy in c:\users\asus\anaconda3\lib\site-packages (1.26.4)

```
In [3]: import math
        print(math.sqrt(16))
```

4.0

```
In [4]: from math import sqrt
```

```
In [5]: sqrt (16)
```

Out[5]: 4.0

```
In [6]: import math as m
```

```
In [7]: print(m.sqrt(64))
```

8.0

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

```
In [8]: import numpy as np
```

```
In [13]: a= np.array([1,2,3,4,5])
```

```
In [14]: a
```

Out[14]: array([1, 2, 3, 4, 5])

```
In [15]: print(a)
```

[1 2 3 4 5]

```
In [17]: m=np.array([[1,2,3],[4,5,6]])
```

```
In [18]: m
```

Out[18]: array([[1, 2, 3],
 [4, 5, 6]])

```
In [21]: a= np.array([1,2.6,3.8,4,5],dtype=float)
        m=np.array([[1,2,3],[4,5,6]],dtype=int)
```

```
In [22]: print(a)
        print(m)
```

[1. 2.6 3.8 4. 5.]
[[1 2 3]
 [4 5 6]]

```
In [27]: print(m.shape)
```

```
(2, 3)
```

```
In [25]: print(m.ndim) # kaç boyutlu
```

```
2
```

```
In [26]: print(m.size)
```

```
6
```

```
In [31]: print(m.itemsize)
print(a.itemsize)
```

```
4
```

```
8
```

```
In [32]: a = np.ones((5,))
```

```
In [33]: print(a)
```

```
[1. 1. 1. 1. 1.]
```

```
In [39]: a = np.ones((2,3), dtype= int)
```

```
In [40]: print(a)
```

```
[[1 1 1]
 [1 1 1]]
```

```
In [41]: i = np.identity(5)
print(i)
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

```
In [42]: a=np.arange(10)
print(a)
```

```
[0 1 2 3 4 5 6 7 8 9]
```

```
In [44]: a=np.arange(3,10,3)
print(a)
```

```
[3 6 9]
```

```
In [45]: a = np.random.random(3)
print(a)
```

```
[0.84655959 0.02548728 0.59644293]
```

```
In [52]: b=np.random.random ((3,4))
print(b)
```

```
[[0.3536592 0.71745197 0.15938617 0.40351946]
 [0.1223797 0.59757706 0.94050809 0.47485174]
 [0.52101281 0.58793318 0.78572802 0.99323427]]
```

```
In [62]: a=np.arange(1,10)
print(a)
```

```
[1 2 3 4 5 6 7 8 9]
```

```
In [56]: a[-1]
```

```
Out[56]: 9
```

```
In [63]: print(a[1:3])
```

```
[2 3]
```

```
In [64]: print(a[1:])
```

```
[2 3 4 5 6 7 8 9]
```

```
In [72]: m=np.array([[1,2,3],[4,5,6],[7,8,9]])
```

```
In [74]: print(m)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

```
In [75]: print(m[0])
```

```
[1 2 3]
```

```
In [76]: print(m[0,1])
```

```
2
```

```
In [77]: print(m[2,1]) # önce satır sonra sutun
```

```
8
```

```
In [78]: print(m[1:,1])
```

```
[5 8]
```

```
In [81]: import numpy as nm
a=np.array([1,2,3,4,5])
print(a.min())
print(a.max())
print(a.sum())
print(a.min())
print(a.mean())
print(a.prod())
```

```
1
5
15
1
3.0
120
```

Pandas

```
In [82]: import pandas as pd
```

```
In [83]: mountains_height=pd.Series([2061,2035.8,2028.5,2022.5,2016.4])
```

```
In [84]: print(mountains_height)
```

```
0    2061.0
1    2035.8
2    2028.5
3    2022.5
4    2016.4
dtype: float64
```

```
In [85]: mountains_height=pd.Series(
    data=[2061,2035.8,2028.5,2022.5,2016.4],
    index=["Goverla","Brebensky","Pip_Ivan","Petros","Gutin_Tomnatik"],
    name="Height,m",
    dtype=float
)
print(mountains_height)
```

```
Goverla      2061.0
Brebensky    2035.8
Pip_Ivan     2028.5
Petros       2022.5
Gutin_Tomnatik 2016.4
Name: Height,m, dtype: float64
```

```
In [86]: print(mountains_height[0])
```

```
2061.0
```

C:\Users\asus\AppData\Local\Temp\ipykernel_4904\2488827803.py:1: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`

```
print(mountains_height[0])
```

```
In [87]: print(mountains_height["Goverla"])
```

```
2061.0
```

```
In [89]: print(mountains_height[1:3])
```

```
Brebensky    2035.8
Pip_Ivan     2028.5
Name: Height,m, dtype: float64
```

```
In [90]: print(mountains_height > 2030)
```

```
Goverla      True
Brebensky    True
Pip_Ivan     False
Petros       False
Gutin_Tomnatik False
Name: Height,m, dtype: bool
```

```
In [91]: print(mountains_height [mountains_height > 2030])
```

```
Goverla      2061.0
Brebensky    2035.8
Name: Height,m, dtype: float64
```

```
In [92]: mountains_height = mountains_height.sort_index()
```

```
print(mountains_height)
```

```
Brebenskly      2035.8
Goverla         2061.0
Gutin_Tomnatik  2016.4
Petros          2022.5
Pip_Ivan        2028.5
Name: Height,m, dtype: float64
```

```
In [94]: mountains_height.sort_index(inplace=True)
print(mountains_height)
```

```
Brebenskly      2035.8
Goverla         2061.0
Gutin_Tomnatik  2016.4
Petros          2022.5
Pip_Ivan        2028.5
Name: Height,m, dtype: float64
```

```
In [95]: mountains_height.sort_values(inplace=True,ascending=False)
print(mountains_height)
```

```
Goverla         2061.0
Brebenskly      2035.8
Pip_Ivan        2028.5
Petros          2022.5
Gutin_Tomnatik  2016.4
Name: Height,m, dtype: float64
```

```
In [96]: mountains_height.sort_values(inplace=True,ascending=True)
print(mountains_height)
```

```
Gutin_Tomnatik  2016.4
Petros          2022.5
Pip_Ivan        2028.5
Brebenskly      2035.8
Goverla         2061.0
Name: Height,m, dtype: float64
```

```
In [107... mountains_height=pd.Series(
    data={"Goverla":2061,"Brebenskly":2035.8,"Pip_Ivan":2028.5},
    index=["Goverla","Brebenskly","Pip_Ivan","Petros","Gutin_Tomnatik"],
    name="Height,m",
    dtype=float
)
print(mountains_height)
```

```
Goverla         2061.0
Brebenskly      2035.8
Pip_Ivan        2028.5
Petros          NaN
Gutin_Tomnatik  NaN
Name: Height,m, dtype: float64
```

```
In [108... mountains_height.fillna(1,inplace=True)
print(mountains_height)
```

```
Goverla      2061.0
Brebenskly   2035.8
Pip_Ivan     2028.5
Petros       1.0
Gutin_Tomnatik 1.0
Name: Height,m, dtype: float64
```

```
In [105... mountains_height.fillna(1,inplace=True)
print(mountains_height)
```

```
Goverla      2061.0
Brebenskly   2035.8
Pip_Ivan     2028.5
Petros       0.0
Gutin_Tomnatik 0.0
Name: Height,m, dtype: float64
```

Dataframe

```
In [115... #dictionary
data={'ID': [1,2], 'Name': ['Alice', 'Bob']}
df=pd.DataFrame(data)
print(df)
```

```
   ID  Name
0    1  Alice
1    2   Bob
```

```
In [116... # list
data=[1,2], ['Alice', 'Bob']
df=pd.DataFrame(data, columns= ['ID', 'Name'])
print(df)
```

```
   ID Name
0    1    2
1  Alice Bob
```

```
In [117... #numpy array
data=np.array([[1,2], ['Alice', 'Bob']])
df=pd.DataFrame(data, columns= ['ID', 'Name'])
print(df)
```

```
   ID Name
0    1    2
1  Alice Bob
```

```
In [118... df.shape
```

```
Out[118... (2, 2)
```

```
In [119... df.columns
```

```
Out[119... Index(['ID', 'Name'], dtype='object')
```

```
In [120... print(df.index)
```

```
RangeIndex(start=0, stop=2, step=1)
```

```
In [122... print(df.dtypes)
```

```
ID      object
Name    object
dtype: object
```

```
In [ ]: contacts=pd.DataFrame(
        {
            "Name": [
                "Allan Raymond",
                "Chain Lewis",
                "Kenndy Lane",
                "Wylie Pop"
```

```
In [126... df=pd.DataFrame({
    'A': [1,np.nan,3],
    'B': [np.nan,5,np.nan]
})
```

```
In [127... df
```

```
Out[127...
      A    B
0  1.0  NaN
1  NaN  5.0
2  3.0  NaN
```

```
In [129... df.fillna(0,inplace=True)
```

```
In [130... df
```

```
Out[130...
      A    B
0  1.0  0.0
1  0.0  5.0
2  3.0  0.0
```

```
In [133... df=pd.DataFrame({
    'A': [1,np.nan,3],
    'B': [np.nan,5,np.nan]
})
```

```
In [134... df
```

```
Out[134...
      A    B
0  1.0  NaN
1  NaN  5.0
2  3.0  NaN
```

```
In [135... df.dropna(inplace=True)
```

In [136... df

Out[136... **A B**In []: *## Farklı kaynaklarda Df oluşturma*

In []: df= pd.read_csv('')

In [138... df.to_csv('null_df.csv')

In []: excel_df=pd.read_excel('null_df.xlsx')
df.to_excel('null_df.xlsx', sheet_name='null_sheet')In []: *## json*
*## split, records, index, columns, values*In [147... records = [
 { "name": "Michael",
 "Country": "Canada"
 },
 { "name": "John",
 "Country": "USA"
 },
 { "name": "Liza",
 "Country": "Australia"
 }
]In []: employee_df = pd.read_json('employees.json',orient ='records')
employee_df

In []:

In [211... students_data = {
 'İsim' : ['Anna','Bohdan','Olena','Ivan','Maria','Petro','Sophia','Max', 'Na
 'Yaş' : ['21','22','20','19','23','22','21','20', '19','21'],
 'Uzmanlık' : ['Math','Physics','Biology','Chemistry','Math','Physics','Biolo
}
students_df = pd.DataFrame(students_data)
students_df['Yaş']= students_df['Yaş'].astype(int)
print(students_df)

	İsim	Yaş	Uzmanlık
0	Anna	21	Math
1	Bohdan	22	Physics
2	Olena	20	Biology
3	Ivan	19	Chemistry
4	Maria	23	Math
5	Petro	22	Physics
6	Sophia	21	Biology
7	Max	20	Chemistry
8	Natalia	19	Math
9	Vdim	21	Physics

In [200... students_df['Yaş']= students_df['Yaş'].astype(int)

In [208... `older_students = students_df[students_df['Yaş'] > 20]`

In [210... `older_students`

Out[210...

	İsim	Yaş	Uzmanlık
--	------	-----	----------

0	Anna	21	Math
---	------	----	------

1	Bohdan	22	Phsics
---	--------	----	--------

4	Maria	23	Math
---	-------	----	------

5	Petro	22	Phsics
---	-------	----	--------

6	Sophia	21	Biology
---	--------	----	---------

9	Vdim	21	Phsics
---	------	----	--------

In [214... `older_Phy_students = students_df[(students_df['Yaş'] > 20)& (students_df['Uzmanl`

In [215... `older_Phy_students`

Out[215...

	İsim	Yaş	Uzmanlık
--	------	-----	----------

1	Bohdan	22	Physics
---	--------	----	---------

5	Petro	22	Physics
---	-------	----	---------

9	Vdim	21	Physics
---	------	----	---------

In [190... `students_df.head(3)`

Out[190...

	İsim	Yaş	Uzmanlık
--	------	-----	----------

0	Anna	21	Math
---	------	----	------

1	Bohdan	22	Phsics
---	--------	----	--------

2	Olena	20	Biology
---	-------	----	---------

In []:

In [192... `students_df.shape`

Out[192... `(10, 3)`

In [196... `students_df.describe()`

Out[196...

	İsim	Yaş	Uzmanlık
--	------	-----	----------

count	10	10	10
-------	----	----	----

unique	10	5	4
--------	----	---	---

top	Anna	21	Math
-----	------	----	------

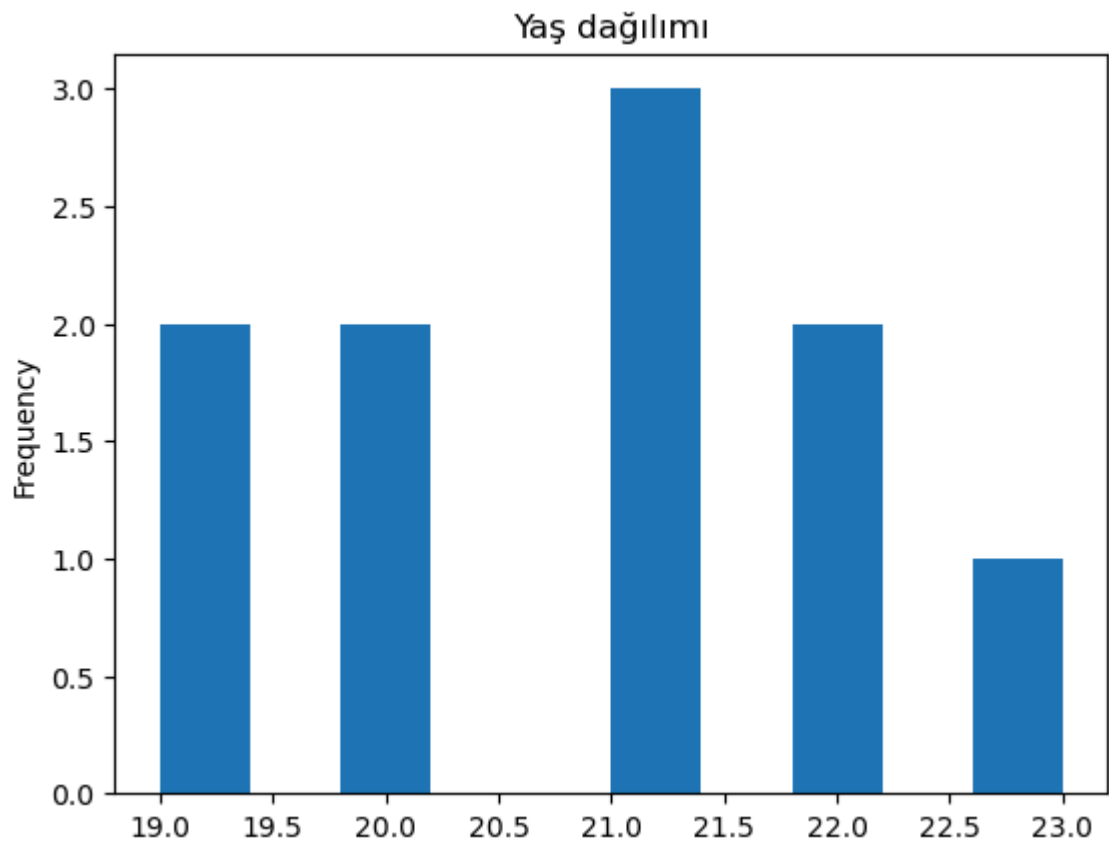
freq	1	3	3
------	---	---	---

In [204... `students_df.info()`

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 10 entries, 0 to 9  
Data columns (total 3 columns):  
#   Column      Non-Null Count  Dtype  
---  ---  
0   İsim        10 non-null     object  
1   Yaş         10 non-null     int32  
2   Uzmanlık   10 non-null     object  
dtypes: int32(1), object(2)  
memory usage: 332.0+ bytes
```

In [226... `students_df['Yaş'].plot(kind = 'hist', title = 'Yaş dağılımı')`

Out[226... `<Axes: title={'center': 'Yaş dağılımı'}, ylabel='Frequency'>`



In [227... `stack_df= pd.read_csv('survey_results_survey')`

```

-----
FileNotFoundError                                Traceback (most recent call last)
Cell In[227], line 1
----> 1 stack_df= pd.read_csv('survey_results_survey')

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1026, in read_csv
(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtype, en
gine, converters, true_values, false_values, skipinitialspace, skiprows, skipfoot
er, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, pars
e_dates, infer_datetime_format, keep_date_col, date_parser, date_format, dayfirs
t, cache_dates, iterator, chunksize, compression, thousands, decimal, linetermina
tor, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_err
ors, dialect, on_bad_lines, delim_whitespace, low_memory, memory_map, float_preci
sion, storage_options, dtype_backend)
    1013 kwds_defaults = _refine_defaults_read(
    1014     dialect,
    1015     delimiter,
    (... )
    1022     dtype_backend=dtype_backend,
    1023 )
    1024 kwds.update(kwds_defaults)
-> 1026 return _read(filepath_or_buffer, kwds)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:620, in _read(fil
epath_or_buffer, kwds)
    617 _validate_names(kwds.get("names", None))
    619 # Create the parser.
--> 620 parser = TextFileReader(filepath_or_buffer, **kwds)
    622 if chunksize or iterator:
    623     return parser

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1620, in TextFile
Reader.__init__(self, f, engine, **kwds)
    1617     self.options["has_index_names"] = kwds["has_index_names"]
    1619 self.handles: IOHandles | None = None
-> 1620 self._engine = self._make_engine(f, self.engine)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1880, in TextFile
Reader._make_engine(self, f, engine)
    1878     if "b" not in mode:
    1879         mode += "b"
-> 1880 self.handles = get_handle(
    1881     f,
    1882     mode,
    1883     encoding=self.options.get("encoding", None),
    1884     compression=self.options.get("compression", None),
    1885     memory_map=self.options.get("memory_map", False),
    1886     is_text=is_text,
    1887     errors=self.options.get("encoding_errors", "strict"),
    1888     storage_options=self.options.get("storage_options", None),
    1889 )
    1890 assert self.handles is not None
    1891 f = self.handles.handle

File ~\anaconda3\Lib\site-packages\pandas\io\common.py:873, in get_handle(path_or
_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    868 elif isinstance(handle, str):
    869     # Check whether the filename is to be opened in binary mode.
    870     # Binary mode does not support 'encoding' and 'newline'.
    871     if ioargs.encoding and "b" not in ioargs.mode:

```

```
872         # Encoding
--> 873         handle = open(
874             handle,
875             ioargs.mode,
876             encoding=ioargs.encoding,
877             errors=errors,
878             newline="",
879         )
880     else:
881         # Binary mode
882         handle = open(handle, ioargs.mode)
```

FileNotFoundError: [Errno 2] No such file or directory: 'survey_results_survey'

In []: CodingActivities

In []:

```
In [ ]: import pandas as pd
stack_pd= pd.csv('survey.csv')
stack_pd.dropna(subset='comp',inplace=True).quantile(0.25)
```