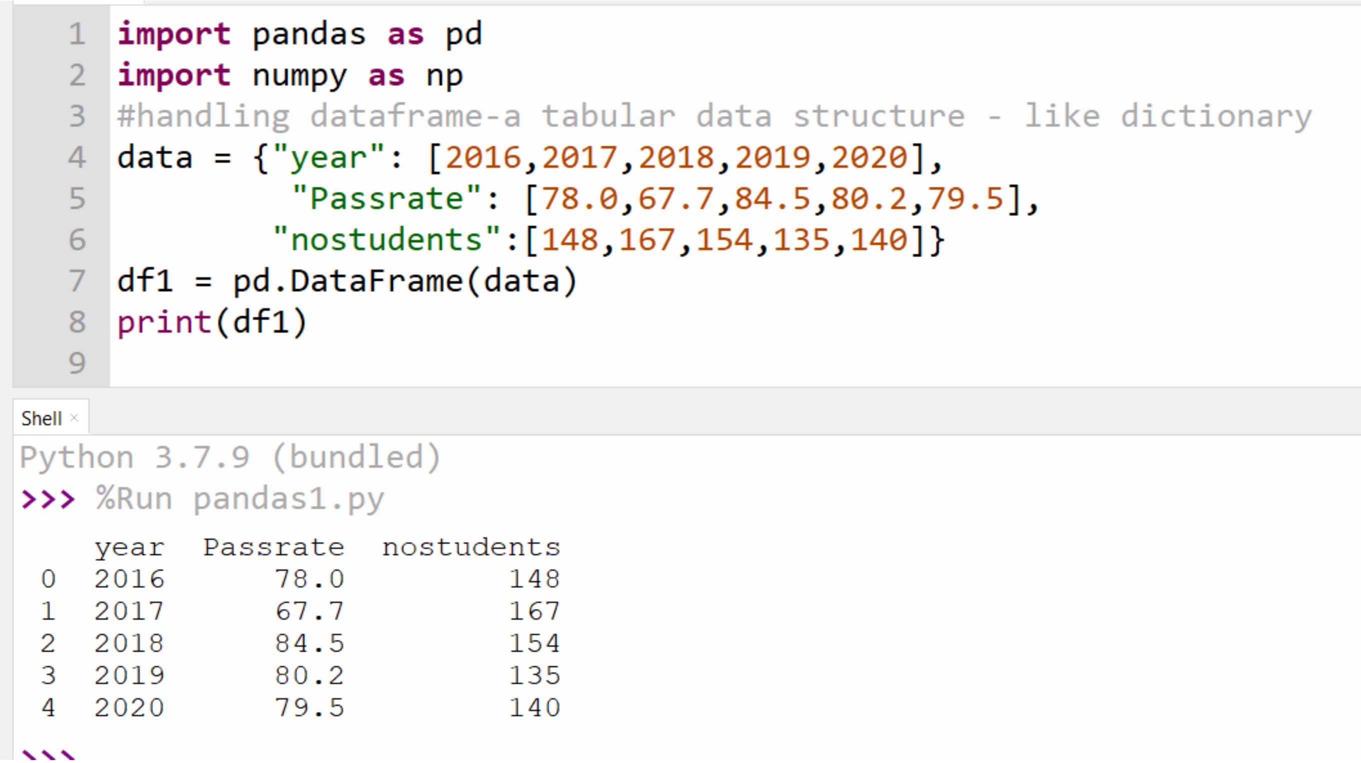
Data analytics in Python Programming

Week 10B:Python data analysis

Pandas: https://pandas.pydata.org/

It is a data analysis package which has computing functions that handle expressive data structures for data analysis.
Contains powerful arithmetic and statistical function to handle large sets of data.
In addition, it has robust input and output support for handling different types of data files including database, csv files and more.

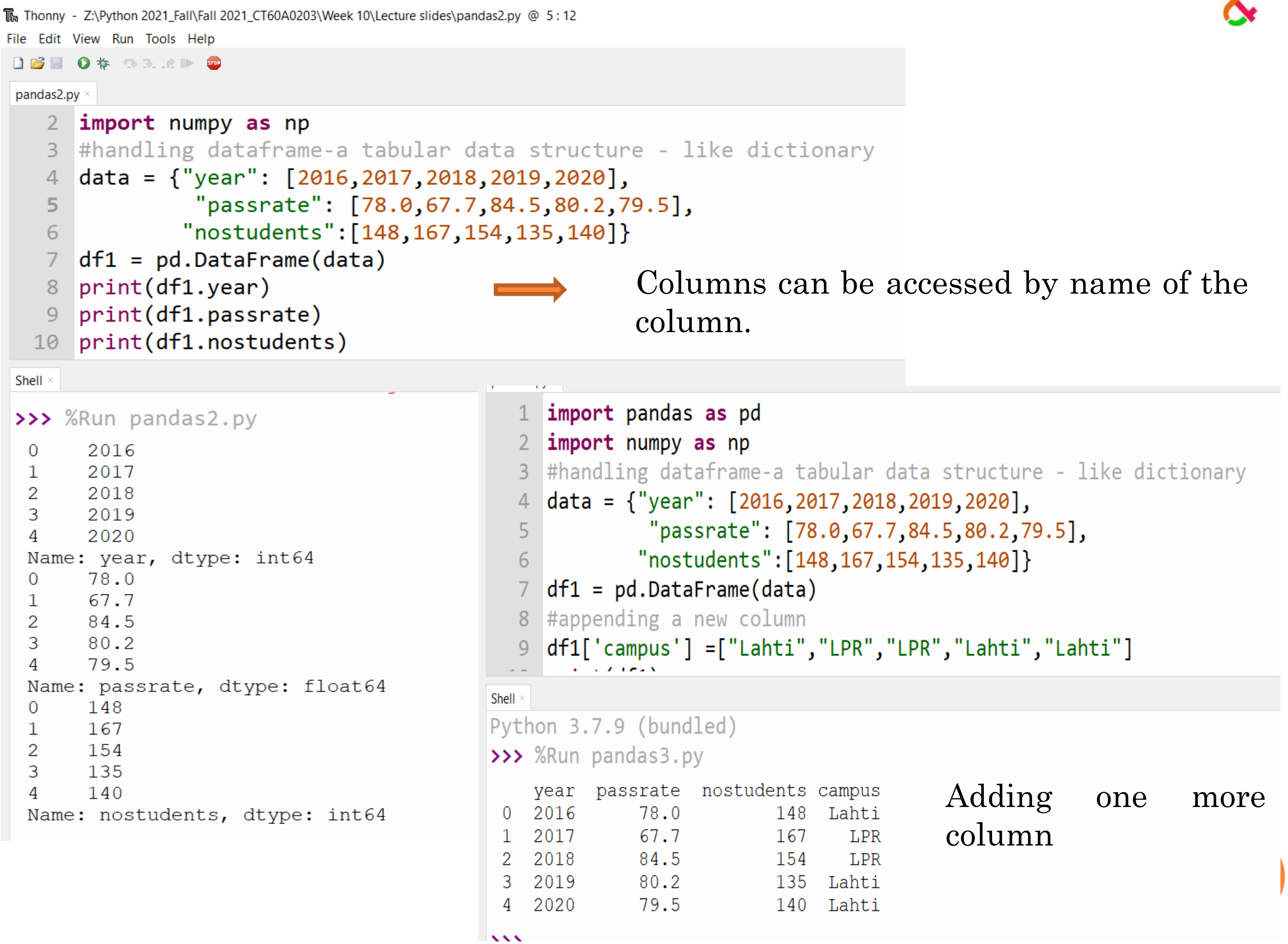


```python
import pandas as pd
import numpy as np
#handling dataframe-a tabular data structure - like dictionary
data = {"year": [2016,2017,2018,2019,2020],
         "Passrate": [78.0,67.7,84.5,80.2,79.5],
        "nostudents":[148,167,154,135,140]}
df1 = pd.DataFrame(data)
print(df1)
```

```
Python 3.7.9 (bundled)
>>> %Run pandas1.py
   year  Passrate  nostudents
0  2016      78.0         148
1  2017      67.7         167
2  2018      84.5         154
3  2019      80.2         135
4  2020      79.5         140

>>>
```

DataFrame is a tabular data structure which contain set of ordered columns and rows.



Thonny - Z:\Python 2021_Fall\Fall 2021_CT60A0203\Week 10\Lecture slides\pandas2.py @ 5 : 12
File Edit View Run Tools Help
pandas2.py
Shell

```python
import numpy as np
#handling dataframe-a tabular data structure - like dictionary
data = {"year": [2016,2017,2018,2019,2020],
         "passrate": [78.0,67.7,84.5,80.2,79.5],
        "nostudents":[148,167,154,135,140]}
df1 = pd.DataFrame(data)
print(df1.year)
print(df1.passrate)
print(df1.nostudents)
```

Columns can be accessed by name of the column.

```
>>> %Run pandas2.py

 0    2016
 1    2017
 2    2018
 3    2019
 4    2020
 Name: year, dtype: int64
 0    78.0
 1    67.7
 2    84.5
 3    80.2
 4    79.5
 Name: passrate, dtype: float64
 0    148
 1    167
 2    154
 3    135
 4    140
 Name: nostudents, dtype: int64
```

```python
import pandas as pd
import numpy as np
#handling dataframe-a tabular data structure - like dictionary
data = {"year": [2016,2017,2018,2019,2020],
         "passrate": [78.0,67.7,84.5,80.2,79.5],
        "nostudents":[148,167,154,135,140]}
df1 = pd.DataFrame(data)
#appending a new column
df1['campus'] =["Lahti","LPR","LPR","Lahti","Lahti"]
```

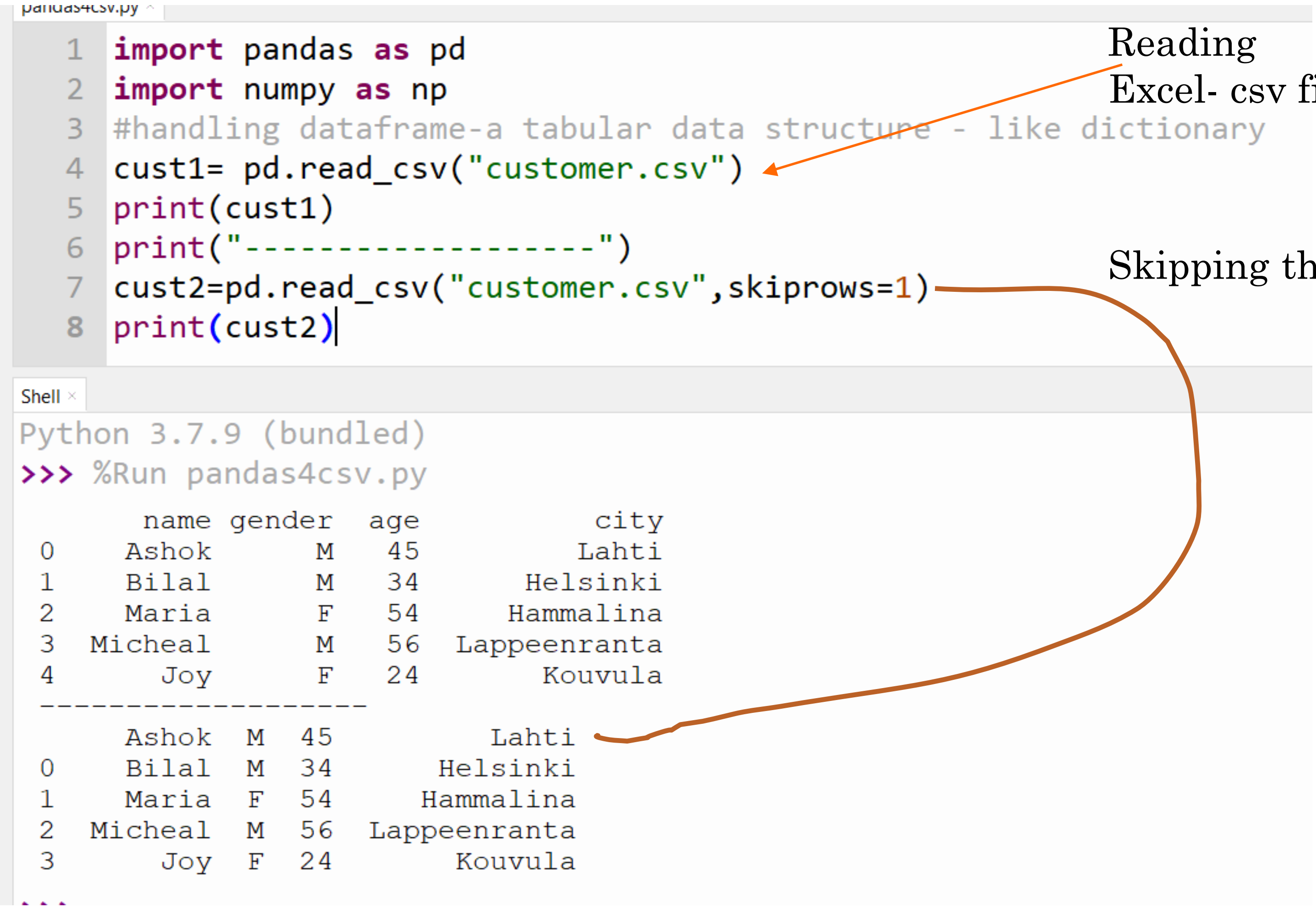
```
Python 3.7.9 (bundled)
>>> %Run pandas3.py

   year  passrate  nostudents campus
0  2016      78.0         148  Lahti
1  2017      67.7         167    LPR
2  2018      84.5         154    LPR
3  2019      80.2         135  Lahti
4  2020      79.5         140  Lahti

>>>
```

Adding one more column

Accessing .csv (comma-separated values) file and transforming it as DataFrame for data analysis



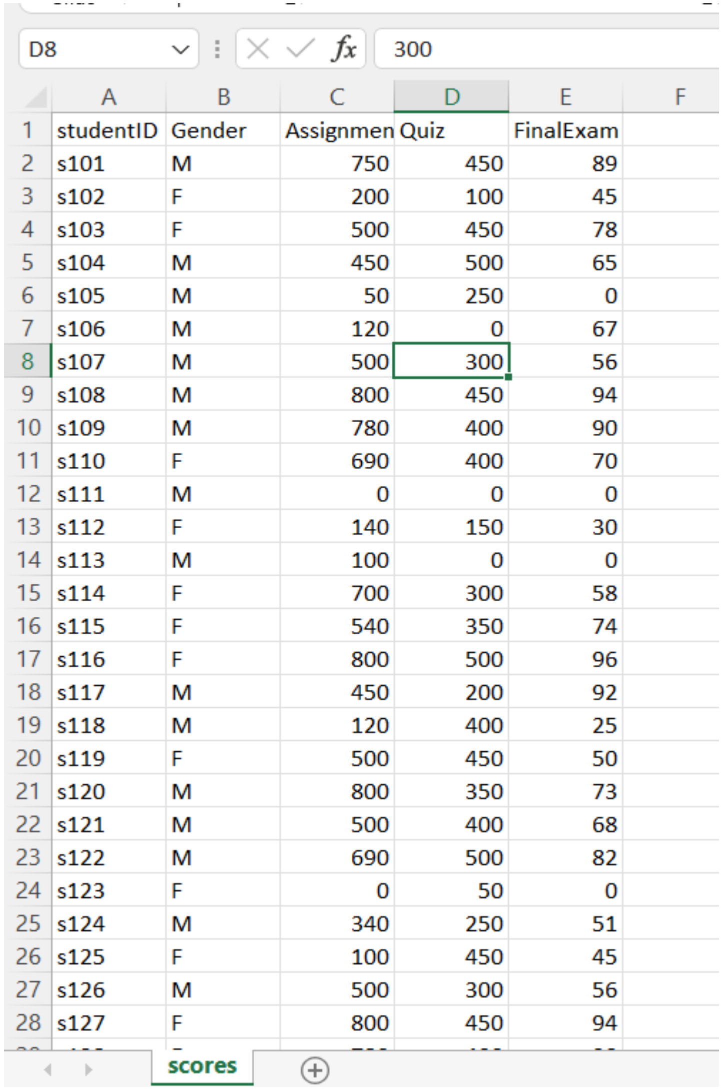
```python
import pandas as pd
import numpy as np
#handling dataframe-a tabular data structure - like dictionary
cust1= pd.read_csv("customer.csv")
print(cust1)
print("-------------------")
cust2=pd.read_csv("customer.csv",skiprows=1)
print(cust2)
```

```
Python 3.7.9 (bundled)
>>> %Run pandas4csv.py
      name gender  age         city
0    Ashok      M   45        Lahti
1    Bilal      M   34     Helsinki
2    Maria      F   54     Hammalina
3  Micheal      M   56  Lappeenranta
4      Joy      F   24      Kouvula
-------------------
     Ashok  M  45         Lahti
0    Bilal  M  34      Helsinki
1    Maria  F  54      Hammalina
2  Micheal  M  56  Lappeenranta
3      Joy  F  24       Kouvula
```

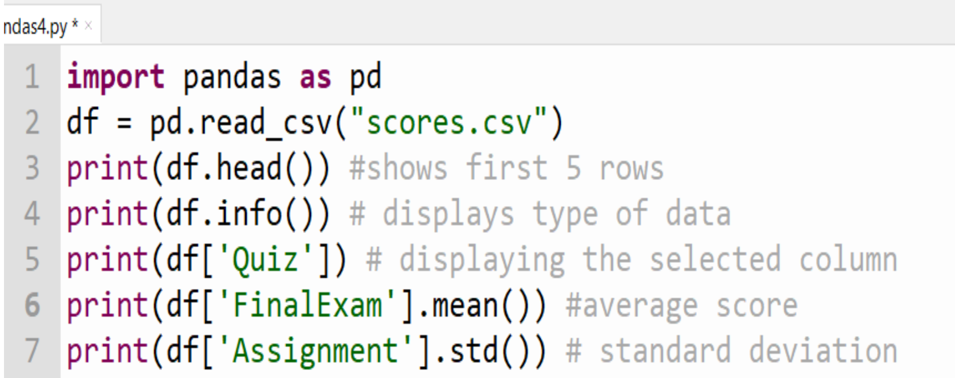
Reading data from Excel- csv file

Skipping the first row

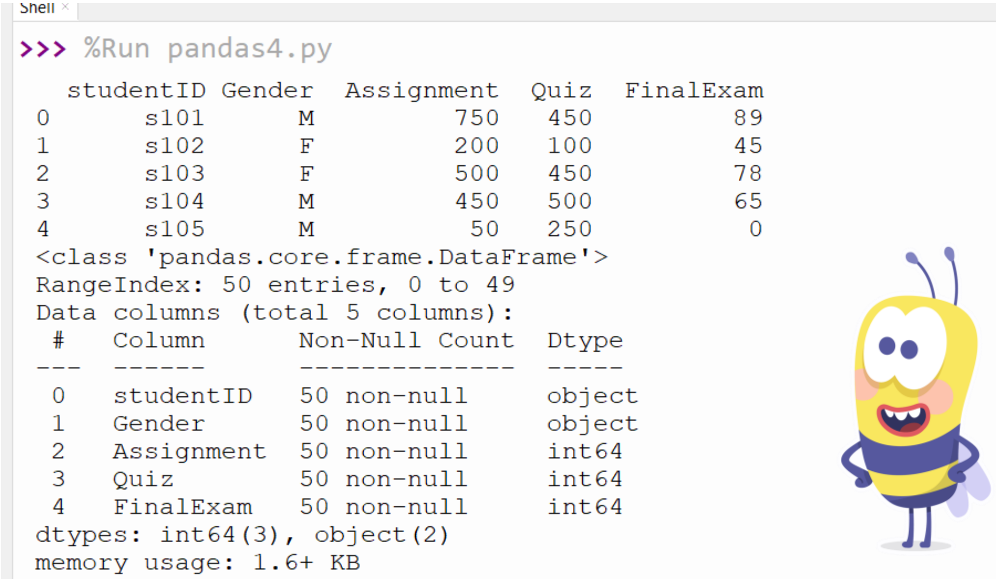
- **Statistics + Pandas + Python Programming**
- As noted, **Pandas** features artihemetic and statiscal functions for handling large volume of data.
- Lets begin with descriptive statistics!! **scores.csv**



	A	B	C	D	E	F
1	studentID	Gender	Assignmen	Quiz	FinalExam	
2	s101	M	750	450	89	
3	s102	F	200	100	45	
4	s103	F	500	450	78	
5	s104	M	450	500	65	
6	s105	M	50	250	0	
7	s106	M	120	0	67	
8	s107	M	500	300	56	
9	s108	M	800	450	94	
10	s109	M	780	400	90	
11	s110	F	690	400	70	
12	s111	M	0	0	0	
13	s112	F	140	150	30	
14	s113	M	100	0	0	
15	s114	F	700	300	58	
16	s115	F	540	350	74	
17	s116	F	800	500	96	
18	s117	M	450	200	92	
19	s118	M	120	400	25	
20	s119	F	500	450	50	
21	s120	M	800	350	73	
22	s121	M	500	400	68	
23	s122	M	690	500	82	
24	s123	F	0	50	0	
25	s124	M	340	250	51	
26	s125	F	100	450	45	
27	s126	M	500	300	56	
28	s127	F	800	450	94	

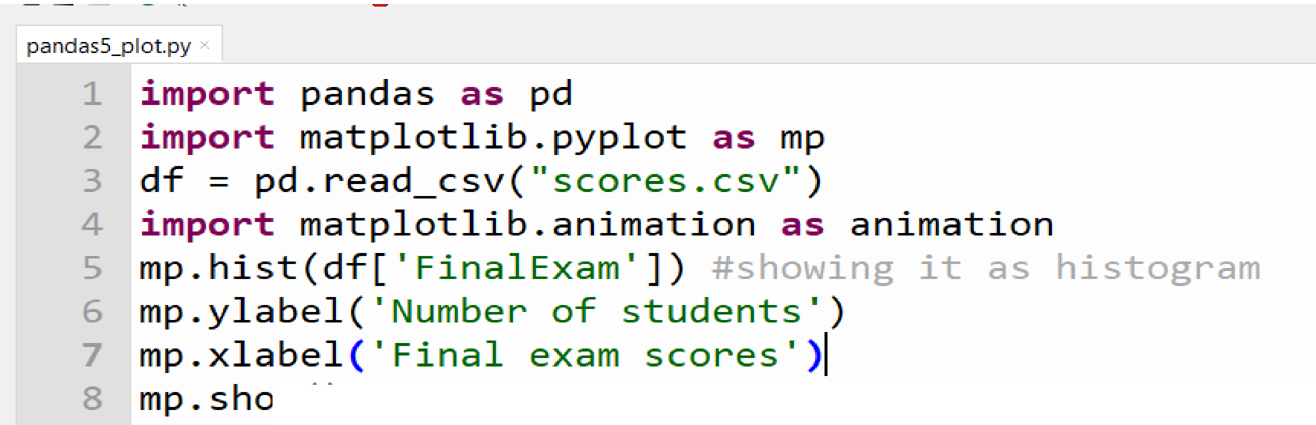


```python
import pandas as pd
df = pd.read_csv("scores.csv")
print(df.head()) #shows first 5 rows
print(df.info()) # displays type of data
print(df['Quiz']) # displaying the selected column
print(df['FinalExam'].mean()) #average score
print(df['Assignment'].std()) # standard deviation
```

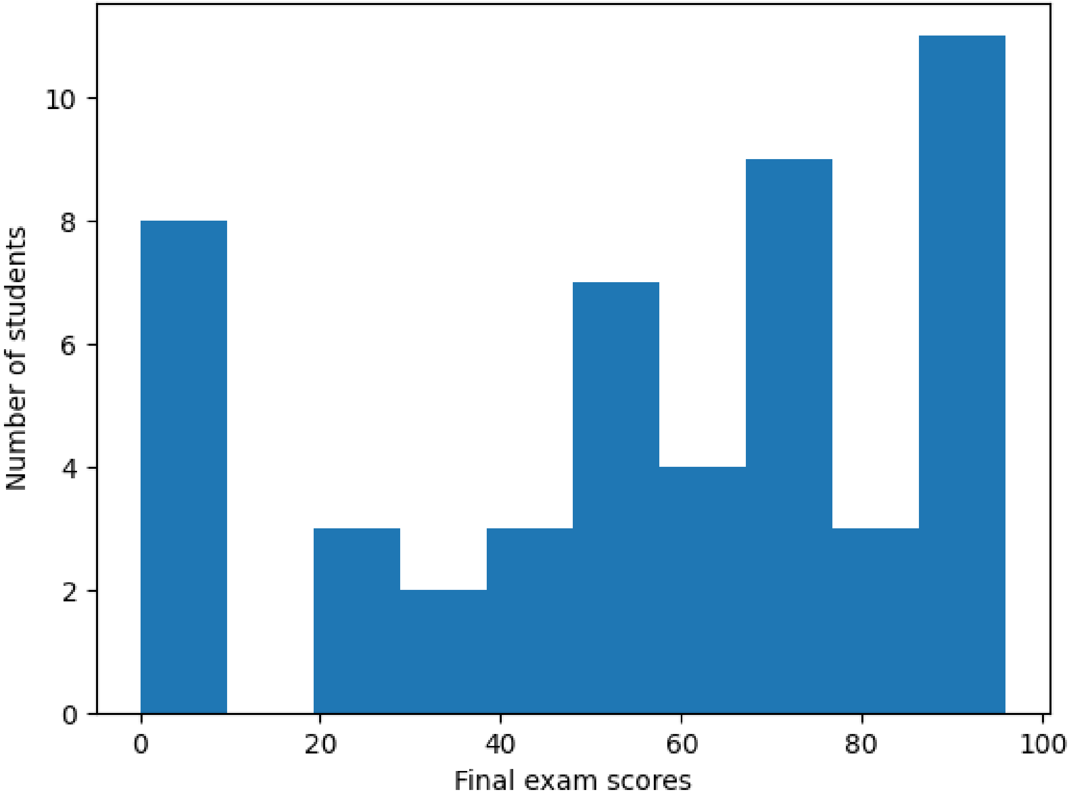


```
>>> %Run pandas4.py
  studentID Gender  Assignment  Quiz  FinalExam
0      s101      M         750   450         89
1      s102      F         200   100         45
2      s103      F         500   450         78
3      s104      M         450   500         65
4      s105      M          50   250          0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   studentID   50 non-null     object
 1   Gender      50 non-null     object
 2   Assignment  50 non-null     int64
 3   Quiz        50 non-null     int64
 4   FinalExam   50 non-null     int64
dtypes: int64(3), object(2)
memory usage: 1.6+ KB
```

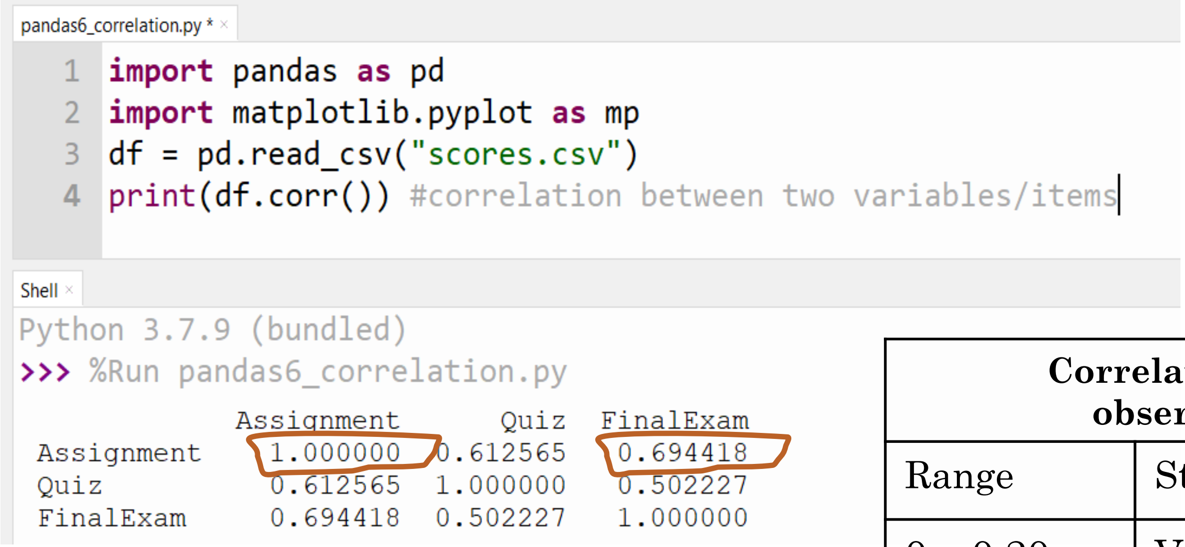
Pandas + matplotlib and Python Programming



```
import pandas as pd
import matplotlib.pyplot as mp
df = pd.read_csv("scores.csv")
import matplotlib.animation as animation
mp.hist(df['FinalExam']) #showing it as histogram
mp.ylabel('Number of students')
mp.xlabel('Final exam scores')
mp.sho
```



- **Correlation pandas**
- Correlation is a statistical measure that measures the relationship between two sets of data.
- Example: Is there any relationship between student assigment scores and final exam scores?



```python
import pandas as pd
import matplotlib.pyplot as mp
df = pd.read_csv("scores.csv")
print(df.corr()) #correlation between two variables/items
```

```
Python 3.7.9 (bundled)
>>> %Run pandas6_correlation.py
            Assignment      Quiz  FinalExam
Assignment    1.000000  0.612565   0.694418
Quiz          0.612565  1.000000   0.502227
FinalExam     0.694418  0.502227   1.000000
```

Correlation coefficient observation chart	
Range	Strength of relationship
0 – 0.20	Very low
0.20 – 0.40	Low
0.40 – 0.60	Moderate
0.60 – 0.80	High
0.80 – 1.00	Very high

Assignment Vs Quiz → 0.69 (High)
There is a positive relationship between student assignment scores and subsequent final exam scores. This implies that students that secured good scores in assignment may do well or will get good scores in the final exam.