



دوره جامع پایتون:  
بخش علوم داده  
جلسه شانزدهم

دکتر ذبیح اله ذبیحی

# نصب ماژول pandas

- `pip install pandas`

# فراخوانی

- import pandas
- import pandas as pd

---

```
import pandas as pd
```

```
print(pd.__version__)
```

- Series به عنوان یک آرایه یک بعدی و از نوع `immutable` در نظر گرفته می شود.
- DataFrame یک آرایه دو بعدی است که دقیقا به مانند جدول های بانک اطلاعاتی رفتار می کند.
- Panel یک آرایه سه بعدی است که دقیقا به مانند مکعب داده رفتار می کند. (در نسخه های جدید منسوخ شده)

## ساخت series

- `pandas.Series( data, index, dtype, copy)`

```
import pandas as pd  
a = [5, 6, 7]  
y = pd.Series(a)  
print(y)
```

---

```
import pandas as pd  
a = [5, 6, 7]  
y = pd.Series(a)  
print(y[0])
```

---

```
import pandas as pd
a = [5, 6, 7]
y = pd.Series(a,index = ["A", "B", "C"])
print(y)
```

---

```
import pandas as pd
a = [5, 6, 7]
y = pd.Series(a,index = ["A", "B", "C"])
print(y["B"])
```

```
import pandas as pd
temp = {"day1": 30, "day2": 32, "day3": 27}
y= pd.Series(temp)
print(y)
```

```
-----
import pandas as pd
temp = {"day1": 30, "day2": 32, "day3": 27}
y= pd.Series(temp,index = ["day1", "day2"])
print(y)
```



# DataFrames ساخت

- `pandas.DataFrame( data, index, columns, dtype, copy)`

```
import pandas as pd
data = {"date": [5, 15, 20], "temp": [30, 32, 28]}
y = pd.DataFrame(data)
print(y)
```

```
-----
import pandas as pd
data = {"date": [5, 15, 20], "temp": [30, 32, 28]}
y = pd.DataFrame(data)
print(y.loc[0])
```

```
-----
import pandas as pd
data = {"date": [5, 15, 20], "temp": [30, 32, 28]}
y = pd.DataFrame(data)
print(y.loc[[0, 1]])
```

```
import pandas as pd
data = {"date": [5, 15, 20], "temp": [30, 32, 28]}
y = pd.DataFrame(data, index = ["day1", "day2", "day3"])
print(y)
```

```
-----
import pandas as pd
data = {"date": [5, 15, 20], "temp": [30, 32, 28]}
y = pd.DataFrame(data, index = ["day1", "day2", "day3"])
print(y.loc["day2"])
```

```
import pandas as pd
x={"date":[15,20,25,30],"temp":[28,31,27,29]}
y=pd.DataFrame(x,index=["day1","day2","day3","day4"])
print(y)
print("-----")
print(y.loc["day1"]["temp"])
```

```
import pandas as pd
data={"day1":[24,23,29],"day2":[27,25,31],"day3":[29,25,30]}
y=pd.DataFrame(data,index=["sari","hamedan","tehran"])
print(y)
```

```
import pandas as pd
data = [[5, 30], [15, 32], [20, 20]]
y= pd.DataFrame(data, columns=['date', 'temp'])
print(y)
```

# استخراج اطلاعات آماری با تابع describe()

```
import pandas as pd  
data = [[5, 30], [15, 32], [20, 20]]  
y= pd.DataFrame(data, columns=['date', 'temp'])  
print(y.describe())
```

# تابع همبستگی

- تغییرات این تابع بین -1 تا 1 است.

```
import pandas as pd
data = {"date": [5, 15, 20], "temp": [30, 32, 28]}
y = pd.DataFrame(data, index = ["day1", "day2", "day3"])
print(y.corr())
```



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

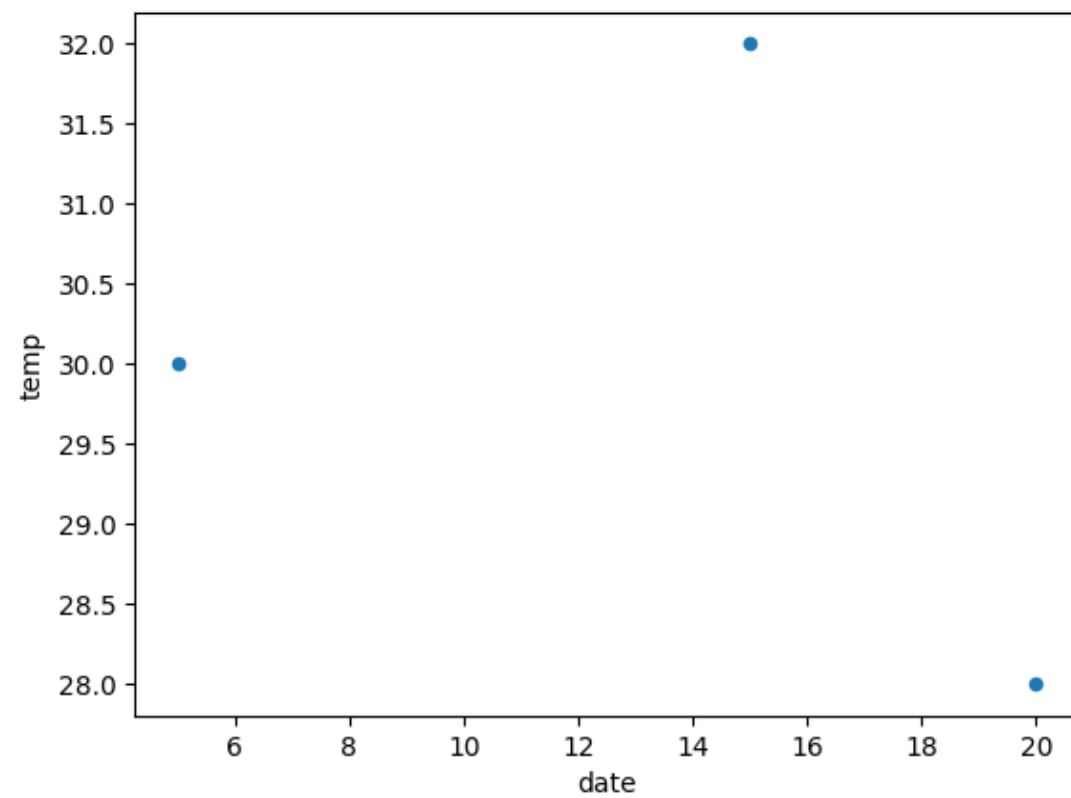
```
import pandas as pd
```

```
data = {"date": [5, 15, 20], "temp": [30, 32, 28]}
```

```
y = pd.DataFrame(data, index = ["day1", "day2", "day3"])
```

```
y.plot(kind = 'scatter', x = 'date', y = 'temp')
```

```
plt.show()
```



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

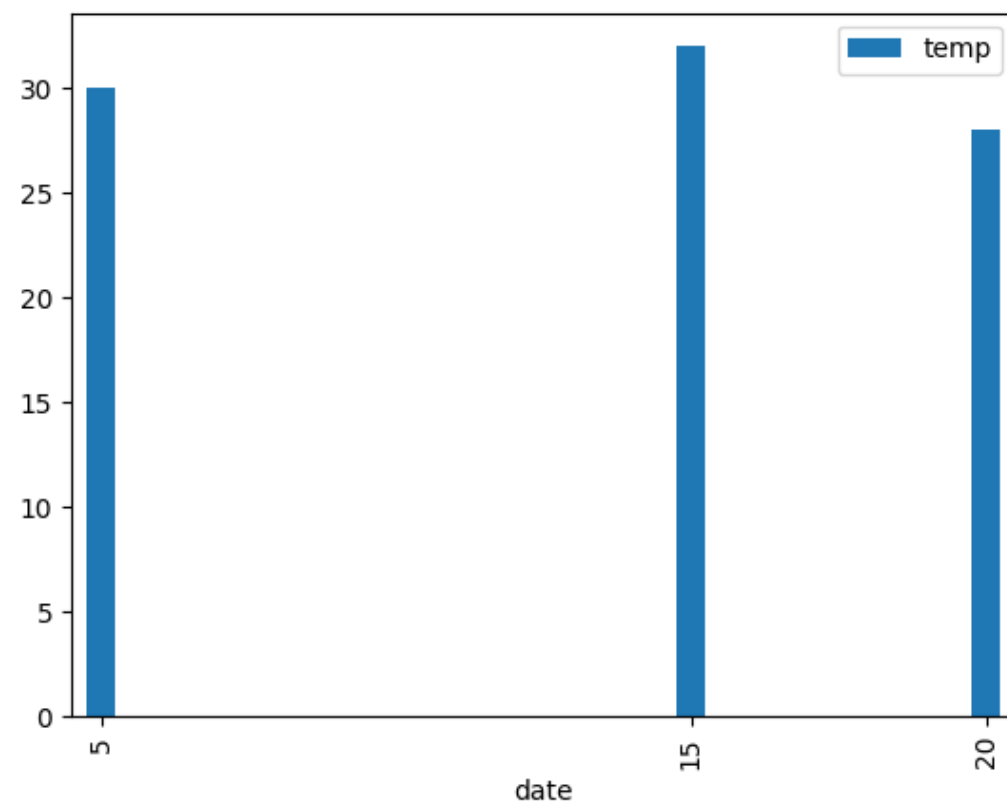
```
import pandas as pd
```

```
data = {"date": [5, 15, 20], "temp": [30, 32, 28]}
```

```
y = pd.DataFrame(data)
```

```
y.plot(kind = 'bar', x = 'date', y = 'temp')
```

```
plt.show()
```



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

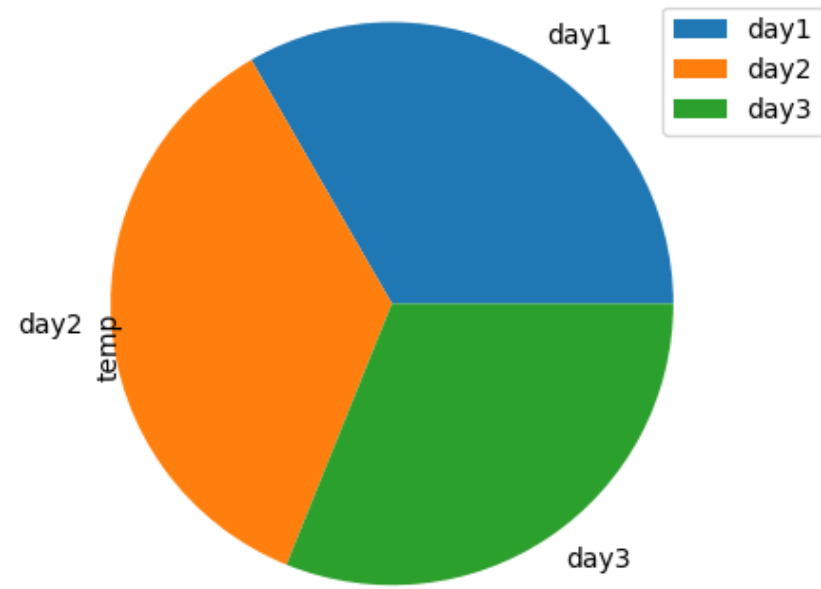
```
import pandas as pd
```

```
data = {"date": [5, 15, 20], "temp": [30, 32, 28]}
```

```
y = pd.DataFrame(data, index = ["day1", "day2", "day3"])
```

```
y.plot(kind = 'pie', x = 'date', y = 'temp')
```

```
plt.show()
```



```
import matplotlib.pyplot as plt
import pandas as pd
data={"day1":[24,23,29],"day2":[27,25,31],"day3":[29,25,30]}
y=pd.DataFrame(data,index=["sari","hamedan","tehran"])
y.plot.bar()
plt.show()
```

```
import matplotlib.pyplot as plt
import pandas as pd
data={"day1":[24,23,29],"day2":[27,25,31],"day3":[29,25,30]}
y=pd.DataFrame(data,index=["sari","hamedan","tehran"])
y.plot.barh()
plt.show()
```



```
import matplotlib.pyplot as plt
import pandas as pd
data={"sari":[24,23,29],"hamedan":[27,25,31],"tehran":[29,25,30]}
y=pd.DataFrame(data,index=["day1","day2","day3"])
y.plot.bar(stacked=True)
plt.show()
```

## ادغام با تابع merge()

```
import pandas as pd
```

```
data = {"date": [5, 15, 20], "temp": [30, 32, 28]}
```

```
y = pd.DataFrame(data, columns=['date', 'temp'])
```

```
data1 = {"date": [4, 15, 22], "temp": [25, 23, 29]}
```

```
y1 = pd.DataFrame(data1, columns=['date', 'temp'])
```

```
print(pd.merge(y, y1, on='date'))
```

## دستور groupby()

```
import pandas as pd  
data = {"date": [5,5,5,15, 15, 20],"temp": [30,27,29,30, 32, 28]}  
y = pd.DataFrame(data,columns=['date', 'temp'] )  
print(y.groupby(['date']).mean())
```

```
import pandas as pd
data = {"date": [5,5,5,15, 15, 20],"temp": [30,27,29,30, 32, 28]}
y = pd.DataFrame(data,columns=['date', 'temp'] )
print(y.groupby(['date']).sum())
```

# مرتب سازی DataFrame با تابع sort\_value

`DataFrame.sort_values(by, axis=0, ascending=True, inplace=False, kind='quicksort', na_position='last')`

- تمامی آرگومان ها به جز `by` دارای مقادیر پیش فرض اند.
- `by`: نام یک یا مجموعه ای از ستون ها برای مرتب سازی داده ها بر مبنای آن ستون ها است.
- `axis`: عدد 0 یا عبارت "index" برای سطر و 1 یا "columns" برای ستون.
- `ascending`: یک مقدار بولین که اگر `True` بود صعودی، و اگر `False` بود نزولی مرتب می کند.
- `inplace`: یک مقدار بولین است. اگر `True` بود در دیتافریم انتقالی به تابع، تغییراتی ایجاد می کند.
- `kind`: یک مقدار رشته ای است که می تواند سه مقدار `'quicksort'`، `'mergesort'` و یا `'heapsort'` را بگیرد و الگوریتم مرتب سازی بر اساس آن عمل کند.
- `na_position`: می تواند یکی از دو مقدار رشته ای `'last'` یا `'first'` را بگیرد و بر اساس آن تعیین کند که مکان مقادیر `NAN` یا همان تهی، پس از مرتب سازی در ابتدای جدول قرار گیرد یا در انتهای آن.

```
import pandas as pd
data = {"date": [5, 15, 20], "temp": [30, 32, 28]}
y = pd.DataFrame(data)
print(y.sort_values("temp"))
```

```
import pandas as pd
import numpy as np
y = pd.DataFrame({
    'day': ['day1', 'day1', 'day2', np.nan, 'day3', 'day3'],
    'city': ["tehran", "yazd", "sari", "hamedan", "mashhad", "shiraz"],
    'temp_location1': [32, 34, 25, 26, 29, 30],
    'temp_location2': [30, 31, 22, 23, 30, 29]})
print(y.sort_values(by=['temp_location1']))
```

## مقدار دهی مقادیر nan

```
import pandas as pd
import numpy as np
y = pd.DataFrame({
    'day': ['day1', 'day1', 'day2', "day2", 'day3', 'day3'],
    'city': ["tehran", "yazd", "sari", "hamedan", "mashhad", "shiraz"],
    'temp_location1': [32, np.nan, 25, 26, 29, 30],
    'temp_location2': [30, 31, 22, np.nan, 30, 29]})

print(y.fillna(0))
```



```
import pandas as pd
import numpy as np
y = pd.DataFrame({
    'day': ['day1', 'day1', 'day2', "day2", 'day3', 'day3'],
    'city': ["tehran", "yazd", "sari", "hamedan", "mashhad", "shiraz"],
    'temp_location1': [32, np.nan, 25, 26, 29, 30],
    'temp_location2': [30, 31, 22, np.nan, 30, 29]})

print(y.dropna())
```

# Pandas و کار با excle

• نصب

```
pip install xlwt
```

```
pip install openpyxl
```

```
pip install XlsxWriter
```

```
import pandas as pd
```

```
df = pd.DataFrame({'Data': [10, 20, 30, 20, 15, 30, 45]})
```

```
writer = pd.ExcelWriter('pandas_simple.xlsx', engine='xlsxwriter')
```

```
df.to_excel(writer, sheet_name='Sheet1')
```

```
writer.save()
```

## چاپ در اکسل

```
import pandas as pd
```

```
df = pd.DataFrame({'Date': [5, 10, 15, 20, 25, 30], "temp": [27, 25, 28, 29, 32, 29]})
```

```
writer = pd.ExcelWriter('pandas_simple.xlsx', engine='xlsxwriter')
```

```
df.to_excel(writer, sheet_name='Sheet1')
```

```
writer.save()
```

# خواندن داده از اکسل

```
import pandas as pd  
x=pd.read_excel('pandas_simple.xlsx', index_col=0)  
print(x)
```

# تمرین

- کدی بنویسید که بتوان جدول زیر را ایجاد کرد و همچنین اطلاعات آماری و ماتریس همبستگی را استخراج کنید.

ادبیات	ریاضی	شیمی	فیزیک	
۱۷	۱۴	۱۶/۲۵	۱۸	علی
۱۵	۱۳	۱۵	۱۲	محمد
۱۷	۱۸	۲۰	۱۹	رضا