In [ ]:
```
import jupyq as jp
import pandas as pd
import matplotlib as plt
import matplotlib.pyplot as plt
```

In [ ]:
```
dane = pd.read_csv('reviews_courses.csv',parse_dates=['Timestamp'])
dane
```

Out[ ]:

	Course Name	Timestamp	Rating	Comment
0	The Python Course: AI/ML in Python	2021-04-02 06:25:52+00:00	4.0	NaN
1	The Python Course: AI/ML in Python	2021-04-02 05:12:34+00:00	4.0	NaN
2	The Python Course: AI/ML in Python	2021-04-02 05:11:03+00:00	4.0	NaN
3	The Python Course: AI/ML in Python	2021-04-02 03:33:24+00:00	5.0	NaN
4	The Python Course: AI/ML in Python	2021-04-02 03:31:49+00:00	4.5	NaN
...	...	...	...	...
44995	The Python Course: From Beginner to Expert	2018-01-01 01:11:26+00:00	4.0	NaN
44996	The Python Course: AI/ML in Python	2018-01-01 01:09:56+00:00	5.0	NaN
44997	The Python Course: AI/ML in Python	2018-01-01 01:08:11+00:00	5.0	NaN
44998	The Python Course: From Beginner to Expert	2018-01-01 01:05:26+00:00	5.0	NaN
44999	The Python Course: AI/ML in Python	2018-01-01 01:01:16+00:00	5.0	NaN

45000 rows × 4 columns

In [ ]:
```
dane['Course Name'].unique()
```

Out[ ]:
```
array(['The Python Course: AI/ML in Python',
       'The Python Course: OpenCV in Python',
       'The Python Course: 100 Excercises for Beginners',
       'The Python Course: Interactive Visualizations',
       'The Python Course: From Beginner to Expert',
       'The Python Excercises:Data Processing by use of Python',
       'The Python Excercises: From Beginner to Expert',
       'Learn GIS in One Hour'], dtype=object)
```

In [ ]:
```
dane[dane['Course Name'] == 'Learn GIS in One Hour'].mean()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: DataFrame.mean and DataFrame.median with numeric_only=None will include datetime64 and datetime64tz columns in a future version.
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid co
lumns before calling the reduction.
  """Entry point for launching an IPython kernel.
```

Out[ ]:
```
Rating    4.971142
dtype: float64
```

In [ ]:
```
from datetime import datetime
from pytz import utc
```

In [ ]:
```
dane = pd.read_csv('reviews_courses.csv',parse_dates=['Timestamp'])
dane
```

Out[ ]:

	Course Name	Timestamp	Rating	Comment
0	The Python Course: AI/ML in Python	2021-04-02 06:25:52+00:00	4.0	NaN
1	The Python Course: AI/ML in Python	2021-04-02 05:12:34+00:00	4.0	NaN
2	The Python Course: AI/ML in Python	2021-04-02 05:11:03+00:00	4.0	NaN
3	The Python Course: AI/ML in Python	2021-04-02 03:33:24+00:00	5.0	NaN
4	The Python Course: AI/ML in Python	2021-04-02 03:31:49+00:00	4.5	NaN
...	...	...	...	...
44995	The Python Course: From Beginner to Expert	2018-01-01 01:11:26+00:00	4.0	NaN
44996	The Python Course: AI/ML in Python	2018-01-01 01:09:56+00:00	5.0	NaN
44997	The Python Course: AI/ML in Python	2018-01-01 01:08:11+00:00	5.0	NaN
44998	The Python Course: From Beginner to Expert	2018-01-01 01:05:26+00:00	5.0	NaN
44999	The Python Course: AI/ML in Python	2018-01-01 01:01:16+00:00	5.0	NaN

45000 rows × 4 columns

In [ ]:
```
kursy = dane[
        (dane['Timestamp']) < datetime(2020, 10, 1, tzinfo=utc)
].count()
kursy
```

Out[ ]:
```
Course Name    39139
Timestamp      39139
Rating         39139
Comment         6091
dtype: int64
```

In [ ]:
```
kom = dane[
        dane['Comment'].isnull()
].mean
kom
```

Out[ ]:
```
<bound method NDFrame._add_numeric_operations.<locals>.mean of                                        Course Name                Timestamp  \
0                The Python Course: AI/ML in Python 2021-04-02 06:25:52+00:00
1                The Python Course: AI/ML in Python 2021-04-02 05:12:34+00:00
2                The Python Course: AI/ML in Python 2021-04-02 05:11:03+00:00
3                The Python Course: AI/ML in Python 2021-04-02 03:33:24+00:00
4                The Python Course: AI/ML in Python 2021-04-02 03:31:49+00:00
...                                             ...                       ...
44995  The Python Course: From Beginner to Expert 2018-01-01 01:11:26+00:00
44996            The Python Course: AI/ML in Python 2018-01-01 01:09:56+00:00
44997            The Python Course: AI/ML in Python 2018-01-01 01:08:11+00:00
44998  The Python Course: From Beginner to Expert 2018-01-01 01:05:26+00:00
44999            The Python Course: AI/ML in Python 2018-01-01 01:01:16+00:00

       Rating Comment
0         4.0     NaN
1         4.0     NaN
2         4.0     NaN
3         5.0     NaN
4         4.5     NaN
...       ...     ...
44995     4.0     NaN
44996     5.0     NaN
44997     5.0     NaN
44998     5.0     NaN
44999     5.0     NaN

[38201 rows x 4 columns]>
```

In [ ]:
```
dane[
        dane['Comment'].str.contains('the', na= False)
].count()
```

Out[ ]:
```
Course Name    2791
Timestamp      2791
Rating         2791
Comment        2791
dtype: int64
```

In [ ]:
```
dane['Day'] = dane['Timestamp'].dt.date
df_2 = dane.groupby(['Day']).mean()
df_2
```

Out[ ]:

	Rating
Day	
2018-01-01	4.532609
2018-01-02	4.522807
2018-01-03	4.360465
2018-01-04	4.531250
2018-01-05	4.423077
...	...
2021-03-29	4.240000
2021-03-30	4.428371
2021-03-31	4.431325
2021-04-01	4.592593
2021-04-02	4.357143

1188 rows × 1 columns

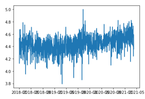
In [ ]:
```
dane.head()
```

Out[ ]:

	Course Name	Timestamp	Rating	Comment	Day
0	The Python Course: AI/ML in Python	2021-04-02 06:25:52+00:00	4.0	NaN	2021-04-02
1	The Python Course: AI/ML in Python	2021-04-02 05:12:34+00:00	4.0	NaN	2021-04-02
2	The Python Course: AI/ML in Python	2021-04-02 05:11:03+00:00	4.0	NaN	2021-04-02
3	The Python Course: AI/ML in Python	2021-04-02 03:33:24+00:00	5.0	NaN	2021-04-02
4	The Python Course: AI/ML in Python	2021-04-02 03:31:49+00:00	4.5	NaN	2021-04-02

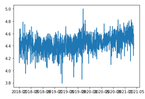
In [ ]:
```
%matplotlib inline
plt.plot(df_2)
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x7ff515e659b0>]
```



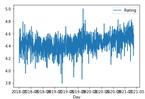
In [ ]:
```
plt.plot(df_2.index, df_2['Rating'])
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x7ff51a07e9d0>]
```



In [ ]:
```
df_2.plot()
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff51aa75290>
```

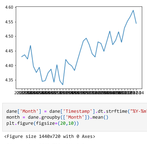


In [ ]:
```
dane['Month'] = dane['Timestamp'].dt.strftime("%Y-%m")
```

In [ ]:
```
df_month = dane.groupby(['Month']).mean()
```

In [ ]:
```
plt.plot(df_month)
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x7ff51a072b10>]
```



In [ ]:
```
dane['Month'] = dane['Timestamp'].dt.strftime("%Y-%m")
month = dane.groupby(['Month']).mean()
plt.figure(figsize=(20,10))
```

Out[ ]:
```
<figure size 1440x720 with 0 Axes>
```

```
<figure size 1440x720 with 0 Axes>
```

In [ ]:
```
plt.plot(month.index, month['Rating'])
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x7ff51a07b15d>]
```

In [ ]:
```
name = dane.groupby(['Month', 'Course Name']).mean()
```

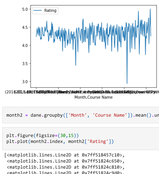
In [ ]:
```
plt.figure(figsize=(20,10))
```

Out[ ]:
```
<figure size 1440x720 with 0 Axes>
```

```
<figure size 1440x720 with 0 Axes>
```

In [ ]:
```
name.plot()
```

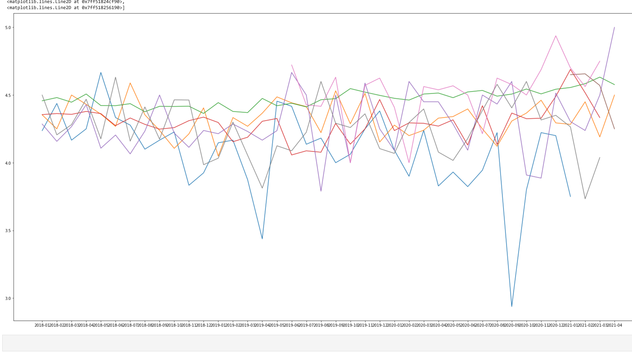
Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff51f3f74c0>
```



In [ ]:
```
month2 = dane.groupby(['Month', 'Course Name']).mean().unstack()
```

In [ ]:
```
plt.figure(figsize=(30,15))
plt.plot(month2.index, month2['Rating'])
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x7ff51845f c10>,
 <matplotlib.lines.Line2D at 0x7ff51824c658>,
 <matplotlib.lines.Line2D at 0x7ff51824c438>,
 <matplotlib.lines.Line2D at 0x7ff51824c908>,
 <matplotlib.lines.Line2D at 0x7ff51824cbe0>,
 <matplotlib.lines.Line2D at 0x7ff51824cd68>,
 <matplotlib.lines.Line2D at 0x7ff51824cf98>,
 <matplotlib.lines.Line2D at 0x7ff51825619e>]
```



In [ ]:
```
```