

# Python - Analiza danych z modułem PANDAS

www.udemy.com (R)

## LAB - S08-L001 - plot

1. Zaimportuj **moduł pandas, numpy i matplotlib** i nadaj im standardowe aliasy
2. Uruchom poniższy fragment kodu, aby przygotować dane do rysowania wykresu (wszystkie zastosowane polecenia powinny Ci już być na tym etapie znane):

```
import datetime

#Import data
nasa = pd.read_csv("nasa_facebook_statuses.csv",
                  usecols=["status_published", "num_reactions",
                           "num_comments", "num_shares", "num_likes"])

#Convert column type to date time
nasa["status_published"] = pd.to_datetime(nasa["status_published"])

#Select observations between two datetimes - May 2015
filter =(nasa['status_published'] >= '2016-5-1') & (nasa['status_published']
< '2016-6-1')
nasa = nasa[filter]

#Add column with day only
nasa['day'] = nasa.apply(lambda row: row["status_published"].day,axis=1)

#Remove non-numeric column
nasa.drop('status_published', axis=1, inplace=True)

#Group all columns by sum
nasa_by_day = nasa.groupby(by='day').sum()

nasa_by_day.head()
```

3. Uruchom polecenie, które wyświetli na wykresie wszystkie dane numeryczne znajdujące się w zmiennej **nasa\_by\_day**. Prawdopodobnie na tym etapie nie zobaczysz ładnego wykresu, bo nie wykonałeś jeszcze jednego ważnego polecenia...
4. Uruchom instrukcję (tzw. funkcja magiczna), które spowoduje, że wykres będzie pokazywany wewnątrz w jupyter notebook
5. Ponownie uruchom polecenie z punktu (3) - tym razem wykres powinien się ładnie wyświetlić
6. Wyświetl wykres tylko dla serii **num\_shares**
7. Wyświetl wykres dla dwóch serii jednocześnie: **num\_shares,num\_comments**
8. Wykonaj jeszcze raz punkt (6), ale tym razem skorzystaj z innej metody - zastosuj parametr **y** polecenia **plot**

9. Wykonaj jeszcze raz punkt (7), ale tym razem skorzystaj z innej metody - zastosuj parametr **y** polecenia **plot**

## Rozwiązania:

Poniżej znajdują się propozycje rozwiązań zadań. Prawdopodobnie istnieje wiele dobrych rozwiązań, dlatego jeżeli rozwiązujesz zadania samodzielnie, to najprawdopodobniej zrobisz to inaczej, może nawet lepiej :) Możesz pochwalić się swoimi rozwiązaniami w sekcji Q&A

```
In [1]: import pandas as pd
import numpy as np
import matplotlib as plt
```

```
In [2]: import datetime

#Import data
nasa = pd.read_csv("nasa_facebook_statuses.csv",
                  usecols=["status_published", "num_reactions",
                           "num_comments", "num_shares", "num_likes"])

#Convert column type to date time
nasa["status_published"] = pd.to_datetime(nasa["status_published"])

#Select observations between two datetimes - May 2015
filter =(nasa['status_published'] >= '2016-5-1') & (nasa['status_published'] < '2016-6-1')
nasa = nasa[filter]

#Add column with day only
nasa['day'] = nasa.apply(lambda row: row["status_published"].day,axis=1)

#Remove non-numeric column
nasa.drop('status_published', axis=1, inplace=True)

#Group all columns by sum
nasa_by_day = nasa.groupby(by='day').sum()

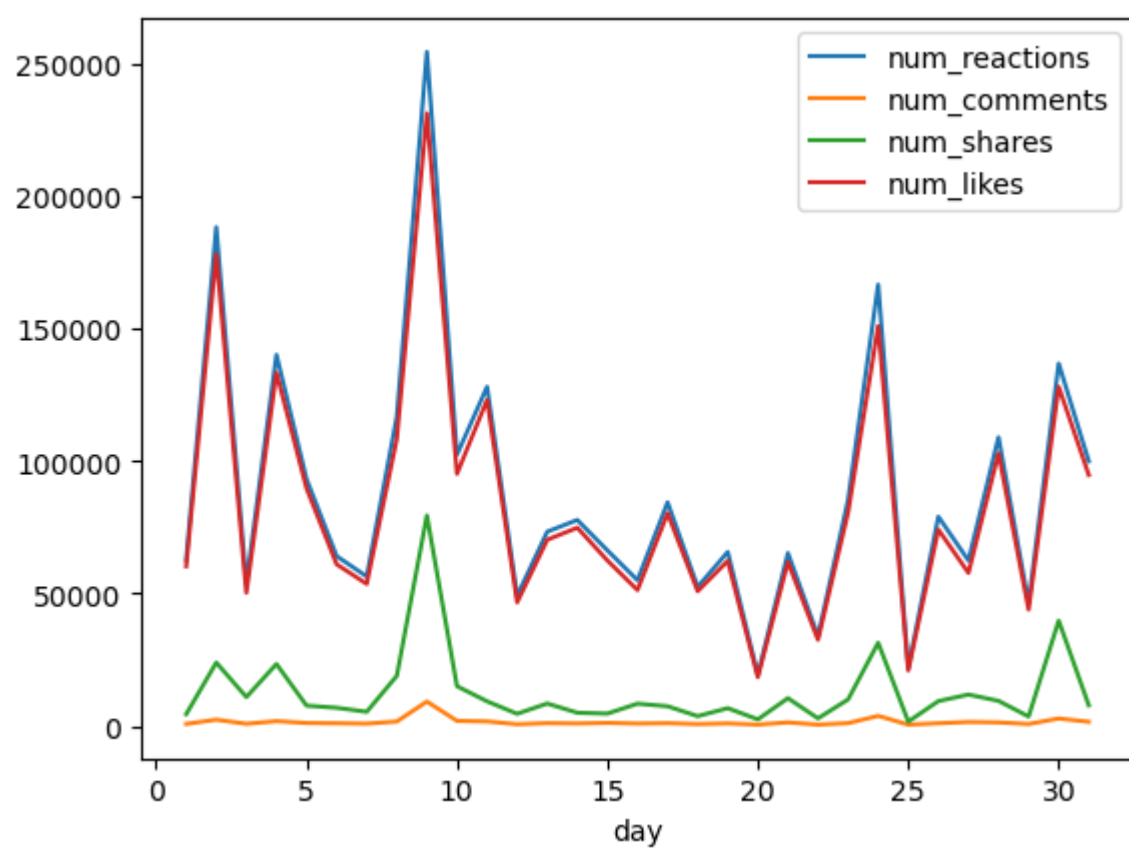
nasa_by_day.head()
```

```
Out[2]:
```

	num_reactions	num_comments	num_shares	num_likes
day				
1	62629	727	4403	60243
2	188255	2328	23897	177989
3	53700	770	10858	50264
4	140098	1776	23360	133302
5	93146	1070	7627	89657

```
In [3]: nasa_by_day.plot()
```

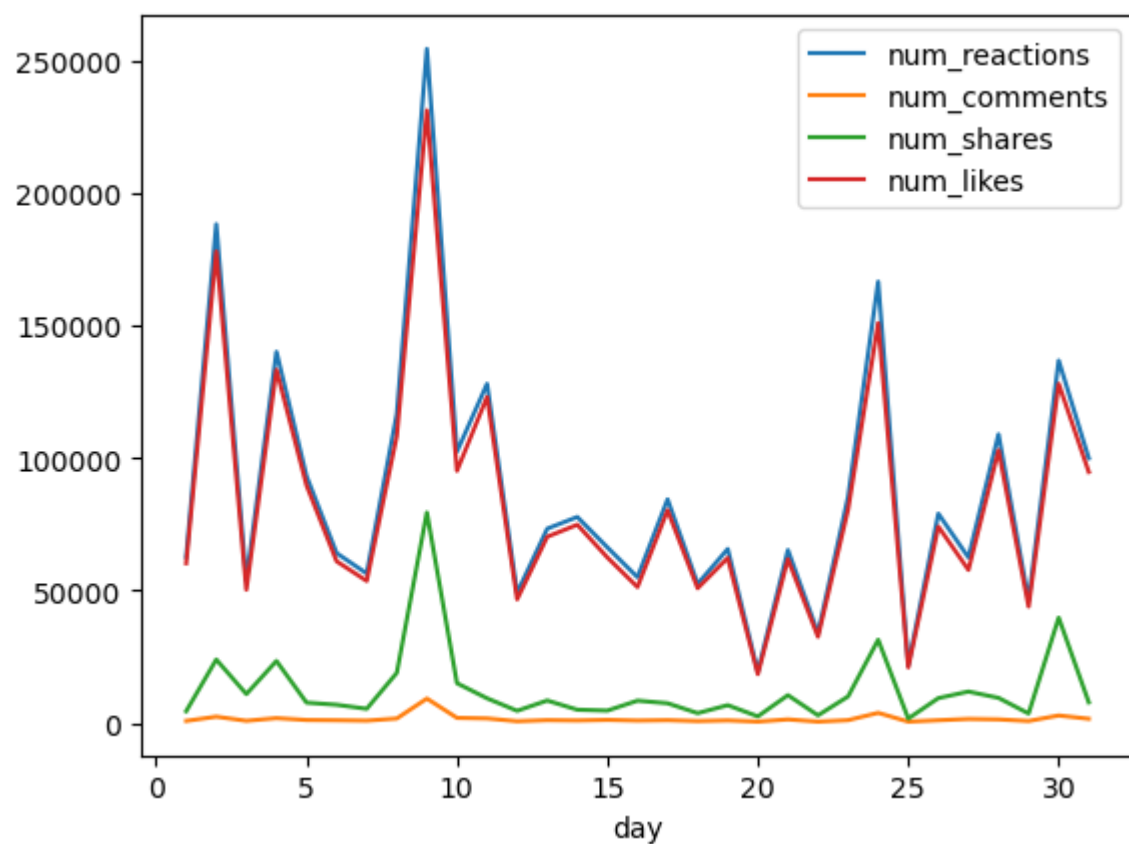
```
Out[3]: <Axes: xlabel='day'>
```



In [4]: `%matplotlib inline`

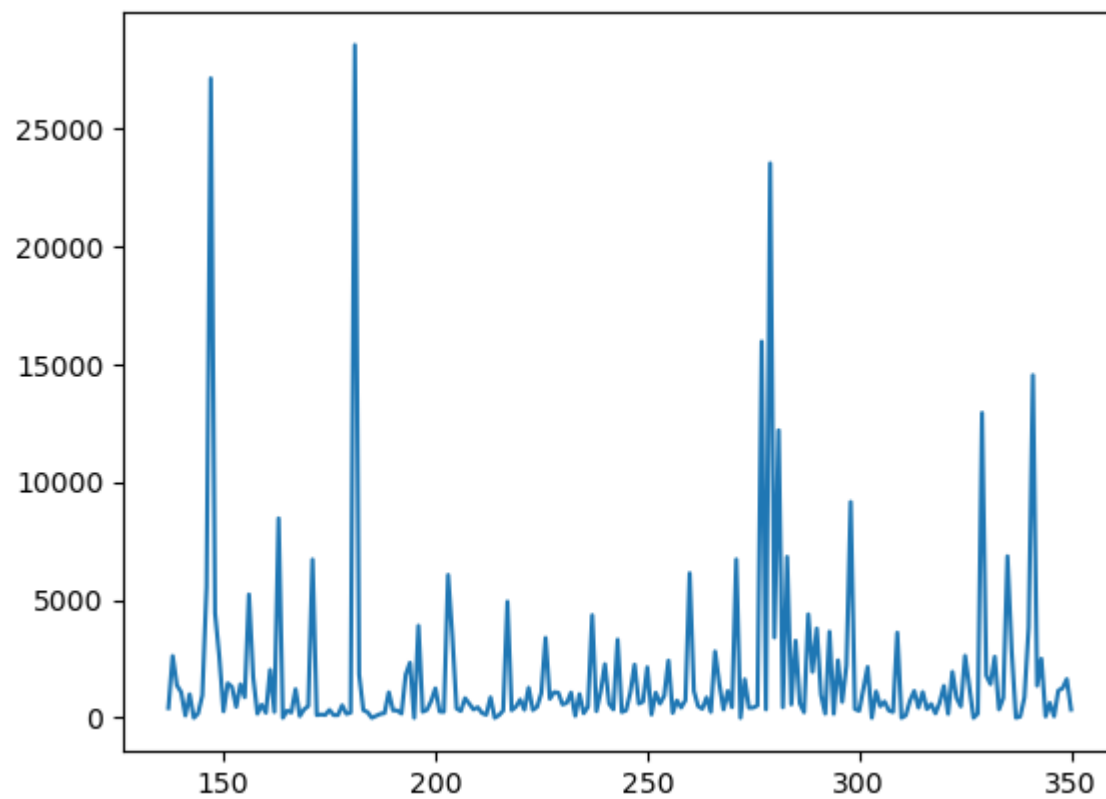
In [5]: `nasa_by_day.plot()`

Out[5]: `<Axes: xlabel='day'>`



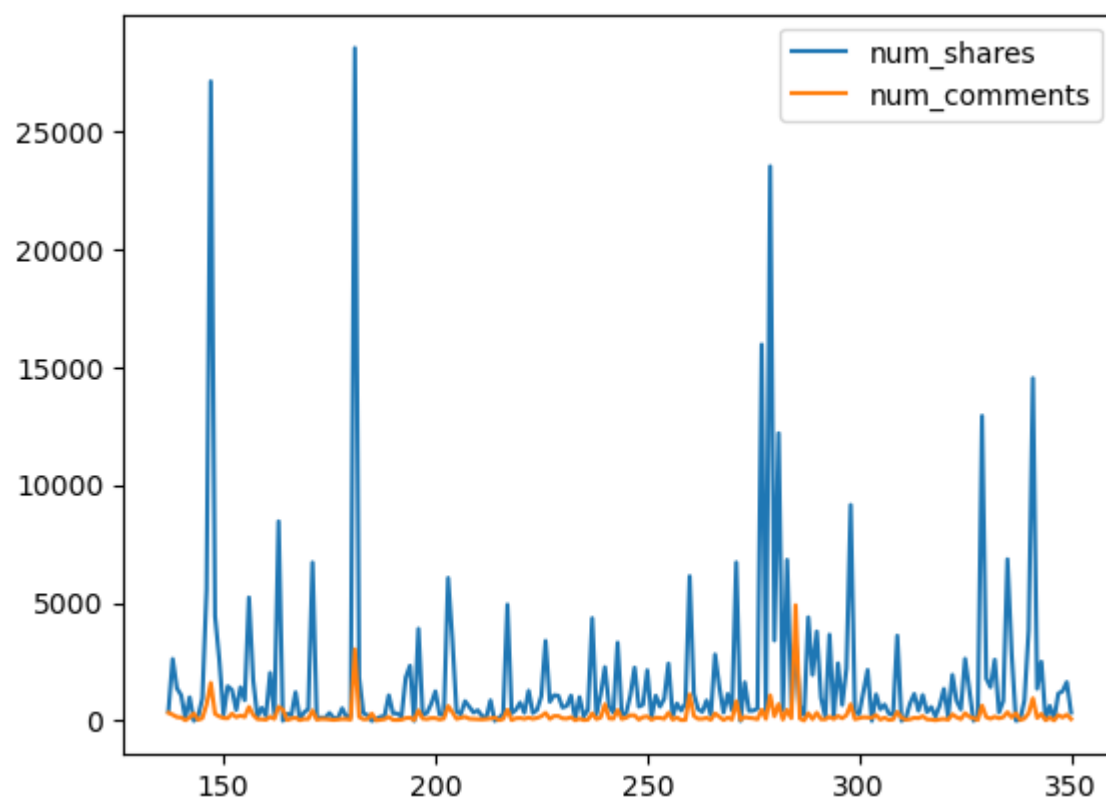
In [6]: `nasa["num_shares"].plot()`

Out[6]: `<Axes: >`



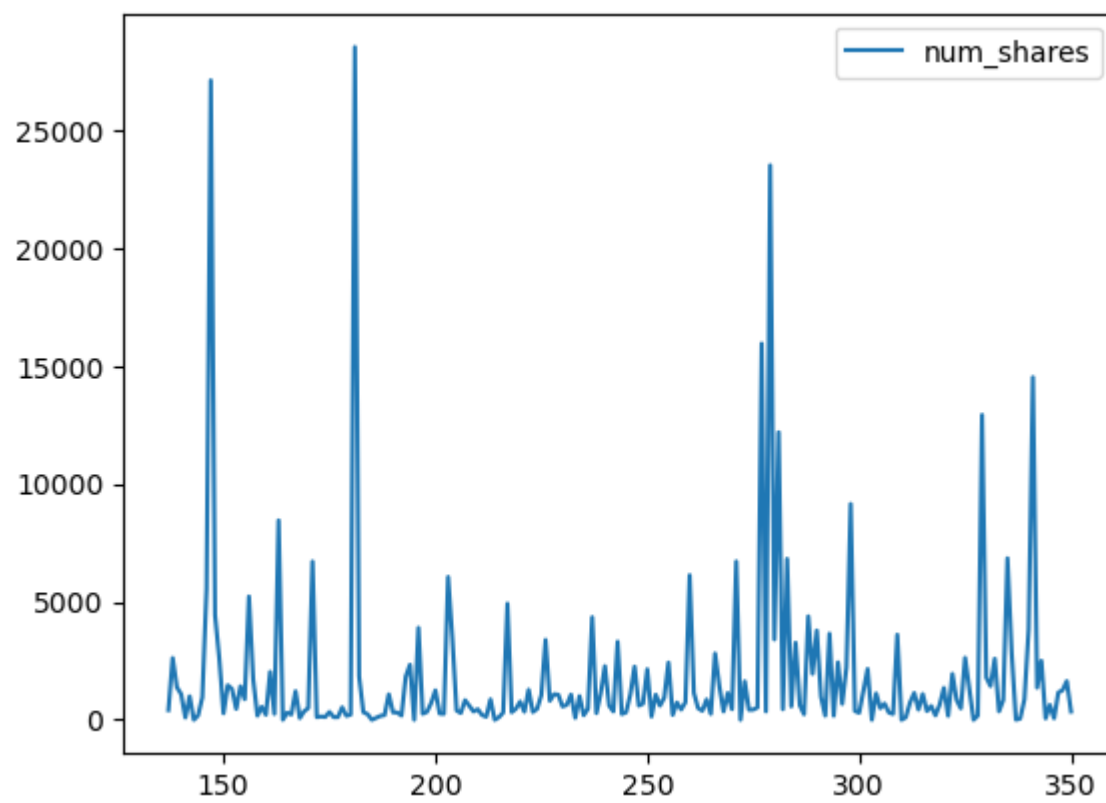
```
In [7]: nasa[["num_shares", "num_comments"]].plot()
```

```
Out[7]: <Axes: >
```



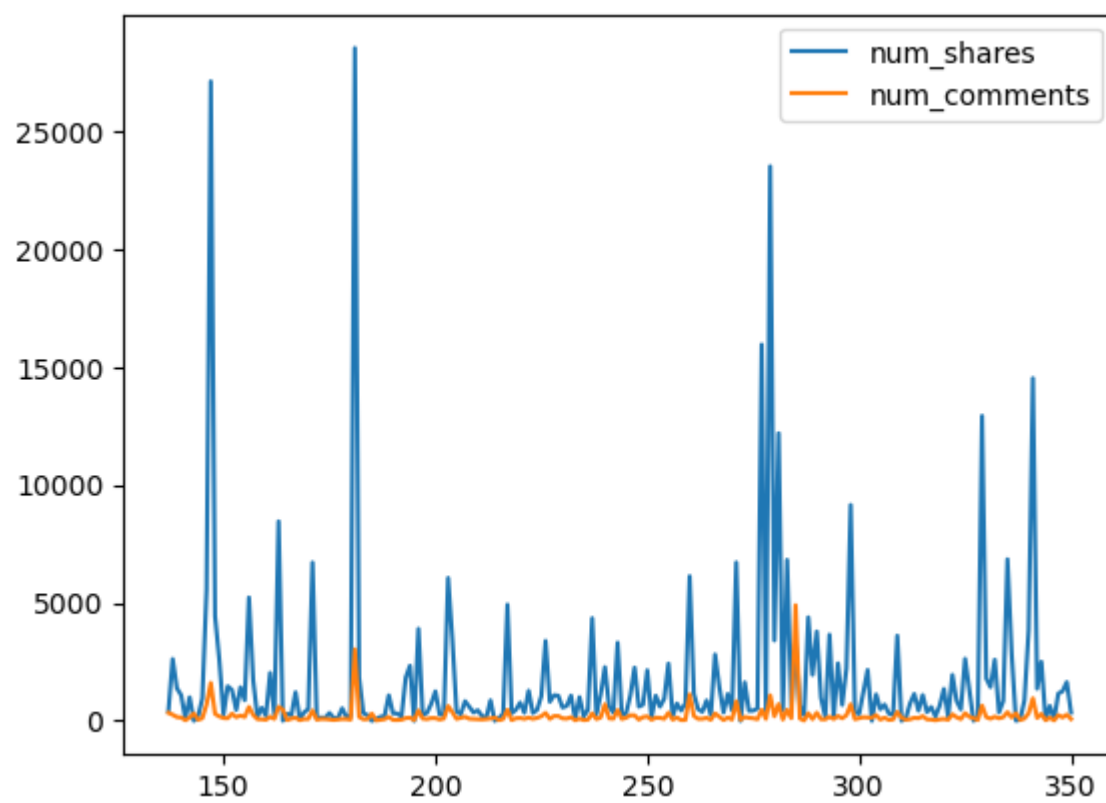
```
In [8]: nasa.plot(y="num_shares")
```

```
Out[8]: <Axes: >
```



```
In [9]: nasa.plot(y=["num_shares", "num_comments"])
```

Out[9]: <Axes: >



```
In [ ]:
```