

Модели стохастических объектов (методы анализа данных)

Практическая работа №1

КИ18-16 Прекель В.А.

Используются matplotlib , numpy , pandas , seaborn , sklearn

```
In [49]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn import preprocessing
```

Подготовка

Считывает данные, сгенерированные в Lab01Scrapper.ipynb

```
In [50]: data = pd.read_json("memeblog.json")
data
```

```
Out[50]:
```

	OwnerId	Id	Date	AttachmentTypes	Comments	Likes	Reposts	Views
0	-120075923	1	2016-04-23 18:49:13+00:00	[photo]	28	61	1	NaN
1	-120075923	2	2016-04-23 19:18:26+00:00	[photo]	1	122	2	NaN
2	-120075923	3	2016-04-23 19:35:32+00:00	[photo]	0	225	9	NaN
3	-120075923	4	2016-04-23 19:35:39+00:00	[photo]	9	38	0	NaN
4	-120075923	7	2016-04-23 20:24:46+00:00	[]	9	45	0	NaN
...
38886	-120075923	785521	2021-02-22 11:10:42+00:00	[video]	39	427	451	12996.0
38887	-120075923	785563	2021-02-22 11:30:02+00:00	[video]	22	293	56	8832.0
38888	-120075923	785614	2021-02-22 12:00:10+00:00	[doc]	13	613	529	8418.0
38889	-120075923	785646	2021-02-22 12:30:02+00:00	[video]	23	403	115	8896.0
38890	-120075923	785695	2021-02-22 13:30:02+00:00	[video]	8	60	12	2361.0

38891 rows × 12 columns

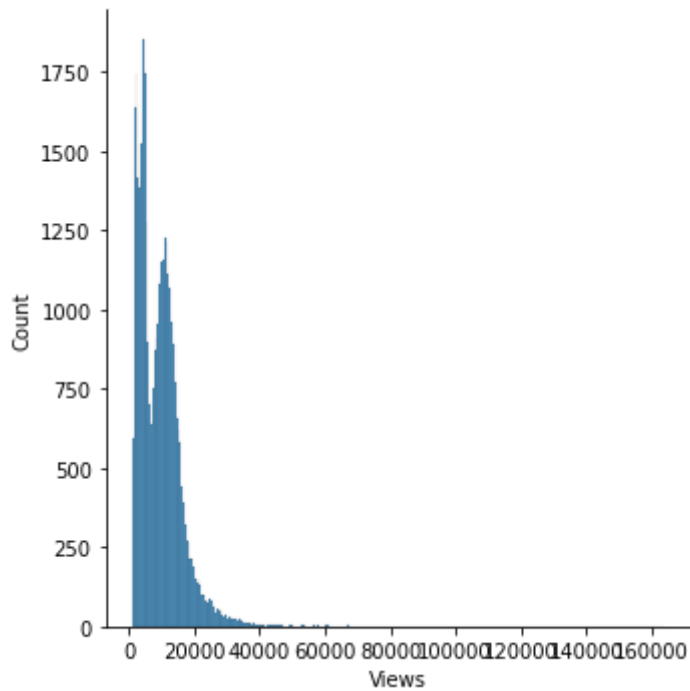


Графики

График распределения по просмотрам

```
In [51]: sns.displot(data.Views)
```

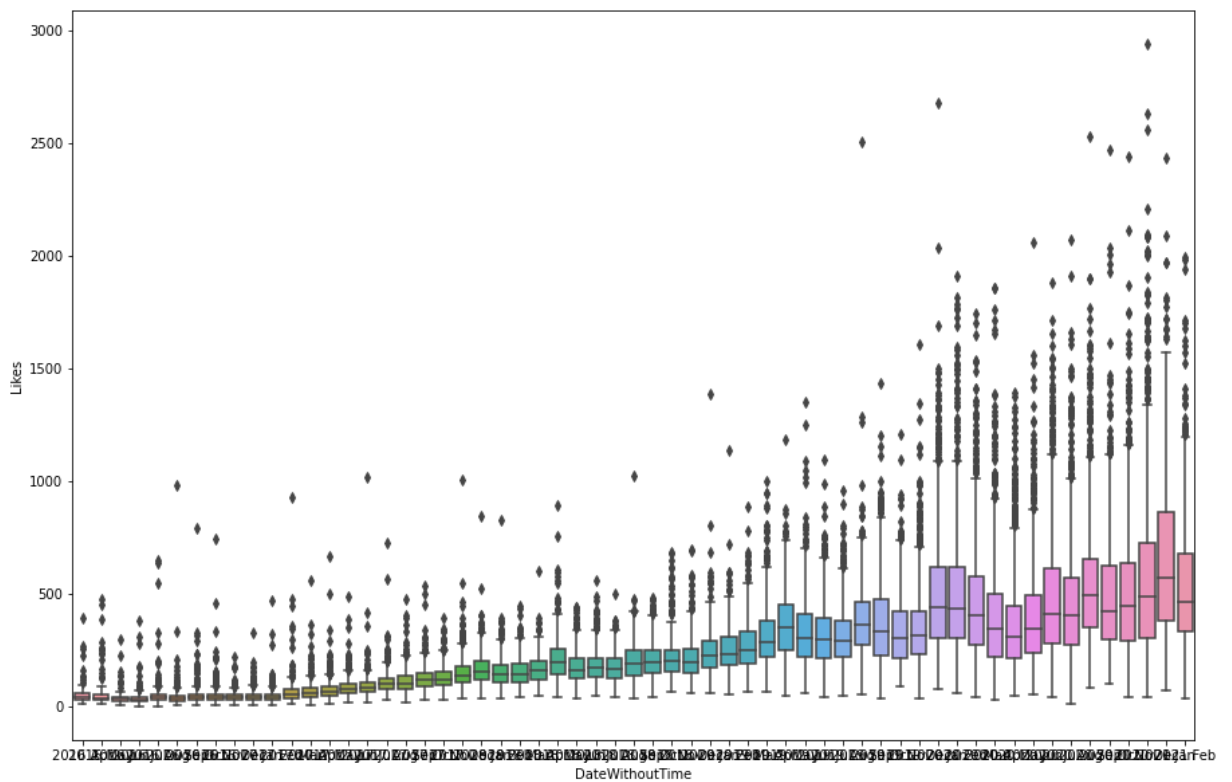
```
Out[51]: <seaborn.axisgrid.FacetGrid at 0x7f6519e1cbb0>
```



Круговая диаграмма по типу вложения

```
In [52]: plt.figure(figsize=(20, 20))  
data.AttachmentTypes.value_counts().plot.pie(autopct='%1.1f%%')
```

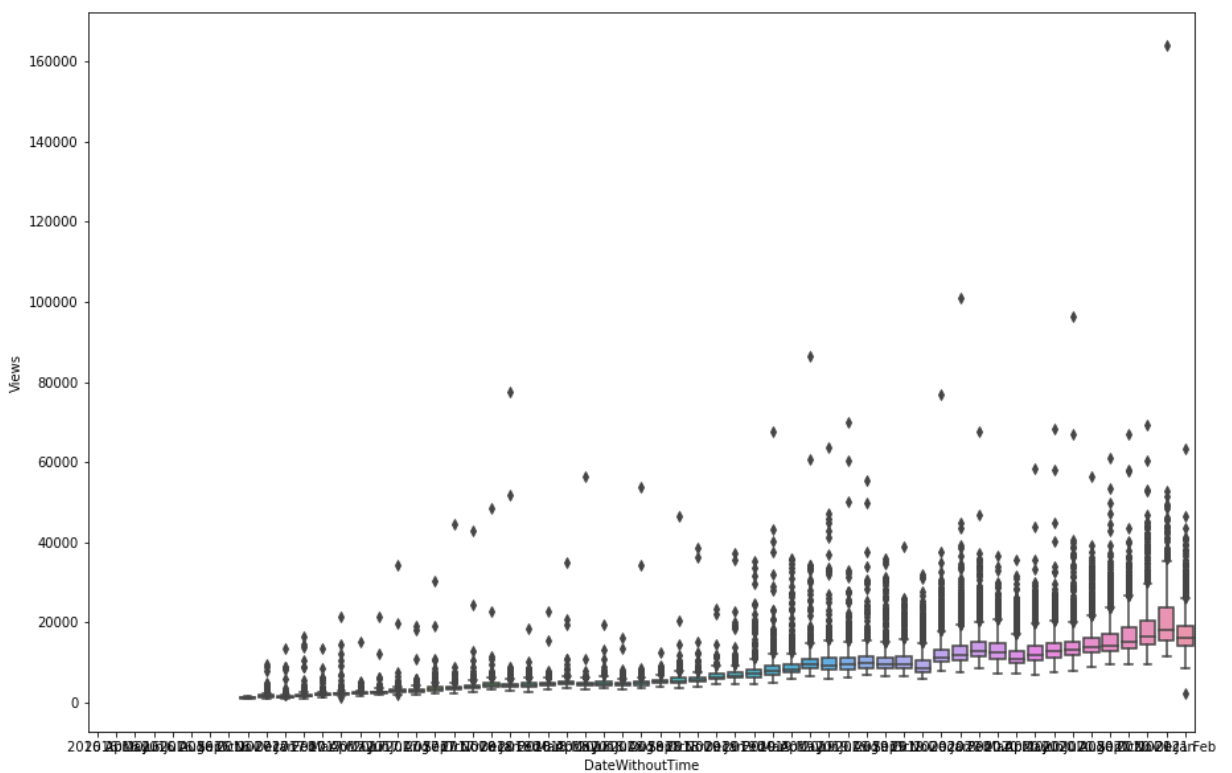
```
Out[52]: <AxesSubplot:ylabel='AttachmentTypes'>
```

Boxplot для просмотров по месяцам (видно, что просмотры в ВК появились с самого начала 2017 года)

```
In [57]: plt.figure(figsize=(15, 10))
sns.boxplot(x="DateWithoutTime", y="Views", data = data)
```

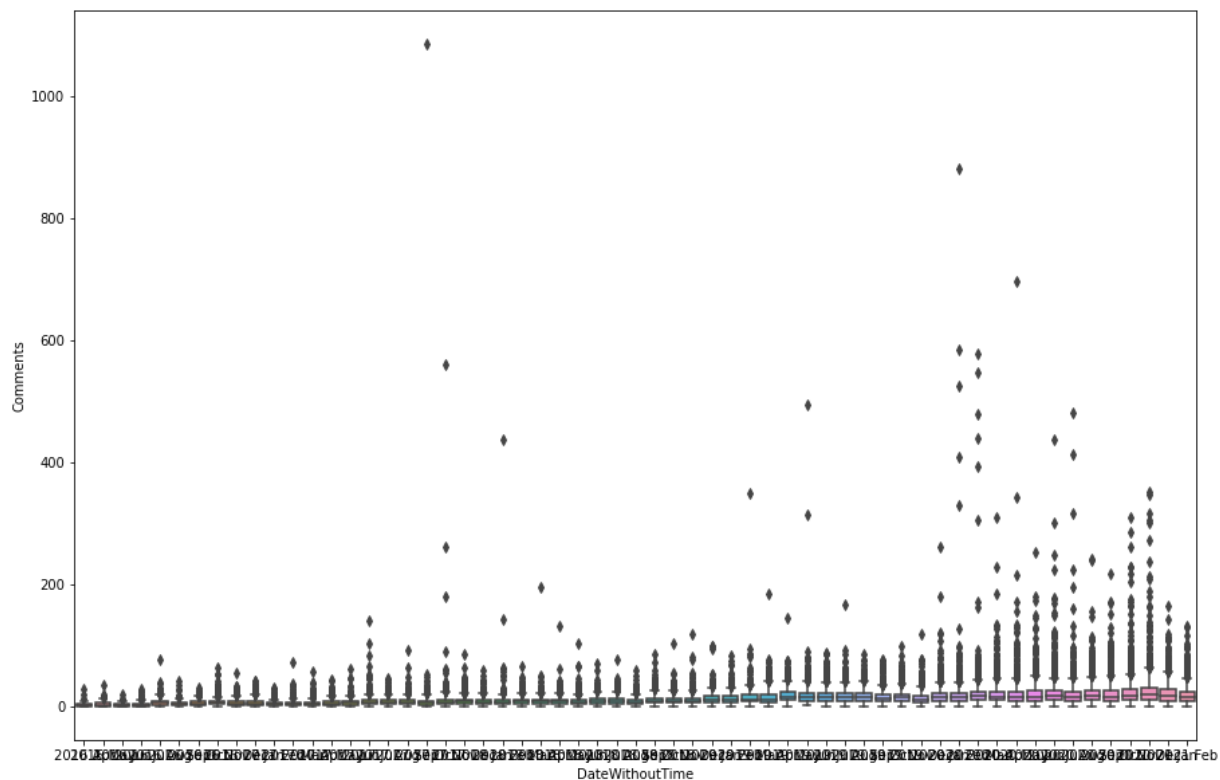
```
Out[57]: <AxesSubplot:xlabel='DateWithoutTime', ylabel='Views'>
```



Boxplot для комментариев по месяцам

```
In [58]: plt.figure(figsize=(15, 10))
sns.boxplot(x="DateWithoutTime", y="Comments", data = data)
```

```
Out[58]: <AxesSubplot:xlabel='DateWithoutTime', ylabel='Comments'>
```



Не графики

Среднее (матожидание) кол-во комментариев

```
In [59]: np.mean(data.Comments)
```

```
Out[59]: 14.763698542079144
```

Медиана кол-ва комментариев

```
In [60]: np.median(data.Comments)
```

```
Out[60]: 11.0
```

Дисперсия кол-ва комментариев

```
In [61]: np.var(data.Comments)
```

```
Out[61]: 410.8780537811244
```

Среднеквадратичное отклонение кол-ва комментариев

```
In [62]: np.std(data.Comments)
```

```
Out[62]: 20.270127127897457
```

Квантили 0.25, 0.5, 0.75, 0.9, 0.95, 0.99, 0.999 по кол-ву комментариев

```
In [63]: q = [0.25, 0.5, 0.75, 0.9, 0.95, 0.99, 0.999]
         [np.quantile(data.Comments, i) for i in q]
```

Out[63]: [6.0, 11.0, 18.0, 29.0, 39.0, 76.0, 241.220000000000116]

Нормализация комментариев

In [64]: `preprocessing.normalize([data.Comments])`

Out[64]: array([[0.00566189, 0.00020221, 0. , ..., 0.00262874, 0.00465084,
0.00161768]])

Стандартизация комментариев

In [65]: `preprocessing.scale(data.Comments)`

Out[65]: array([0.65299548, -0.67901392, -0.7283476 , ..., -0.08700974,
0.40632707, -0.33367815])

In []: