Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК5 «Информатика и вычислительная техника»

Лабораторная работа №1

«Инструментальные средства, подготовка эксперимента и анализ результатов»

ДИСЦИПЛИНА: «Проектирование программного обеспечения»

Выполнил: студент гр. ИУК4-11М _____ (Сафронов Н.С.)
(подпись) (Ф.И.О.)

Проверил: _____ (Потапов А.Е.)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2024

Цель работы: формирование практических навыков подготовки и настройки инструментальной среды анализа данных, умений подготовить эксперимент, представить и проанализировать полученные данные.

Задачи: подготовить и настроить среду исполнения Python, выполнить предобработку данных эксперимента, реализовать простейшие алгоритмы анализа данных (получение статистических характеристик), реализовать адекватное графическое представление результатов**.**

Результаты выполнения работы

Задание 1

```
In [19]: import pandas as pd
```

```
In [6]: data: pd.DataFrame = pd.read_csv("./data/adult.data.csv")
        data.head()
```

Out[6]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hour pe we
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	

Рисунок 1 – Результат загрузки данных в pandas

Задача 1

Сколько мужчин и женщин (признак sex) представлено в этом наборе данных?

```
In [7]: data['sex'].value_counts()
```

```
Out[7]: sex
        Male      21790
        Female    10771
        Name: count, dtype: int64
```

Рисунок 2 – Результат получения количества мужчин и женщин в выборке

Задача 2

Каков средний возраст (признак age) женщин?

```
In [8]: data[data['sex'] == 'Female']['age'].mean()
Out[8]: np.float64(36.85823043357163)
```

Рисунок 3 – Вычисленный средний возраст женщин в выборке

Задача 3

Какова доля граждан Германии (признак native-country)?

```
In [9]: data[data['native-country'] == 'Germany']['native-country'].count() / data['native-country'].count()
Out[9]: np.float64(0.004207487485028101)
```

Рисунок 4 – Доля граждан Германии в выборке

Таким образом, доля граждан Германии – 0.0042 или 0.42%.

Задачи 4-5

Каковы средние значения и среднеквадратичные отклонения возраста тех, кто получает более 50K в год (признак salary) и тех, кто получает менее 50K в год?

```
In [10]: data[data['salary'] == '>50K']['age'].mean(), data[data['salary'] == '>50K']['age'].std()
Out[10]: (np.float64(44.24984058155847), np.float64(10.519027719851826))
```

Рисунок 5 – Среднее значение и среднеквадратичное отклонение возраста тех, кто получает более 50K в год

Таким образом, среднее значение возраста тех, кто получает более 50K в год, - 44.25; среднеквадратичные отклонения – 10.52.

```
In [11]: data[data['salary'] == '<=50K']['age'].mean(), data[data['salary'] == '<=50K']['age'].std()
Out[11]: (np.float64(36.78373786407767), np.float64(14.02008849082488))
```

Рисунок 6 – Среднее значение и среднеквадратичное отклонение возраста тех, кто получает менее либо 50K в год

Таким образом, среднее значение возраста тех, кто получает более 50K в год, - 36.78; среднеквадратичные отклонения – 14.02.

Задача 6

Правда ли, что люди, которые получают больше 50k, имеют как минимум высшее образование? (признак education – Bachelors, Prof-school, Assoc-acdm, Assoc-voc, Masters или Doctorate)

```
In [12]: df1 = data[data['salary'] == '>50K']
         df2 = df1[df1['education'].isin(['Bachelors', 'Prof-school', 'Assoc-acdm', 'Assoc-voc', 'Masters', 'Doctorate'])]
         df1.count == df2.count

Out[12]: False
```

Рисунок 7 – Результат сравнения числа тех, кто имеет и не имеет высшее образование, среди зарабатывающих более 50К

Таким образом, утверждение не является правдой, так как число тех, кто имеет и не имеет высшее образование, среди зарабатывающих более 50К не является равным.

Задача 7

Выведите статистику возраста для каждой расы (признак race) и каждого пола. Используйте groupby и describe. Найдите таким образом максимальный возраст мужчин расы Amer-Indian-Eskimo.

```
In [15]: data.groupby(data['race'])['age'].describe()
```

Out[15]:

race	count	mean	std	min	25%	50%	75%	max
Amer-Indian-Eskimo	311.0	37.173633	12.447130	17.0	28.0	35.0	45.5	82.0
Asian-Pac-Islander	1039.0	37.746872	12.825133	17.0	28.0	36.0	45.0	90.0
Black	3124.0	37.767926	12.759290	17.0	28.0	36.0	46.0	90.0
Other	271.0	33.457565	11.538865	17.0	25.0	31.0	41.0	77.0
White	27816.0	38.769881	13.782306	17.0	28.0	37.0	48.0	90.0

Рисунок 8 – Статистика возраста для каждой расы

Исходя из данных таблицы, максимальный возраст мужчин расы Amer-Indian-Eskimo – 82 года.

Задача 8

Среди кого больше доля зарабатывающих много (>50K): среди женатых или холостых мужчин (признак marital-status)? Женатыми считаем тех, у кого marital-status начинается с Married (Married-civ-spouse, Married-spouse-absent или Married-AF-spouse), остальных считаем холостыми.

```
In [16]: df = data[data['sex'] == 'Male']
         df[df['marital-status'].str.startswith('Married')]['salary'].value_counts() / df[df['marital-status'].str.startswi

Out[16]: salary
         <=50K    0.559486
         >50K     0.440514
         Name: count, dtype: float64
```

Рисунок 9 – Доля зарабатывающих много и мало среди женатых

```
In [21]: df = data[data['sex'] == 'Male']
         df[~df['marital-status'].str.startswith('Married')]['salary'].value_counts() / df[~df['marital-status'].str.starts

Out[21]: salary
         <=50K    0.915505
         >50K     0.084495
         Name: count, dtype: float64
```

Рисунок 10 – Доля зарабатывающих много и мало среди неженатых

Таким образом, доля зарабатывающих много больше у женатых мужчин.

Задача 9

Какое максимальное число часов человек работает в неделю (признак hours-per-week)? Сколько людей работают такое количество часов и каков среди них процент зарабатывающих много?

```
In [11]: m = data['hours-per-week'].max()
         max_salary_df = data[data['hours-per-week'] == m]
         c = max_salary_df['hours-per-week'].count()
         top_c = max_salary_df[max_salary_df['salary'] == '>50K']['hours-per-week'].count()
         print('Max:', m)
         print('Count of Max:', c)
         print(f'Percentage of Max with high salary: {top_c / c * 100:.2f}%')

         Max: 99
         Count of Max: 85
         Percentage of Max with high salary: 29.41%
```

Рисунок 11 – Расчёты для времени работы в неделю

Таким образом, максимальное число часов человек работает в неделю равно 99 часам; 85 человек работают такое количество часов и 29.41% из них зарабатывают много.

Задача 10

Посчитайте среднее время работы (hours-per-week) зарабатывающих мало и много (salary) для каждой страны (native-country).

```
print('>50K', df1.groupby('native-country')['hours-per-week'].mean())
```

```
>50K native-country
?                         45.547945
Cambodia                  40.000000
Canada                    45.641026
China                     38.900000
Columbia                  50.000000
Cuba                      42.440000
Dominican-Republic        47.000000
Ecuador                   48.750000
El-Salvador               45.000000
England                   44.533333
France                    50.750000
Germany                   44.977273
Greece                    50.625000
Guatemala                 36.666667
Haiti                     42.750000
Honduras                  60.000000
Hong                      45.000000
Hungary                   50.000000
India                     46.475000
Iran                      47.500000
Ireland                   48.000000
Italy                     45.400000
Jamaica                   41.100000
Japan                     47.958333
Laos                      40.000000
Mexico                    46.575758
Nicaragua                 37.500000
Peru                      40.000000
Philippines               43.032787
Poland                    39.000000
Portugal                  41.500000
Puerto-Rico               39.416667
Scotland                  46.666667
South                     51.437500
Taiwan                    46.800000
Thailand                  58.333333
Trinadad&Tobago           40.000000
United-States             45.505369
Vietnam                   39.200000
Yugoslavia                49.500000
Name: hours-per-week, dtype: float64
```

Рисунок 12 – Среднее время работы зарабатывающих много по странам

```
print('<=50K', df2.groupby('native-country')['hours-per-week'].mean())
```

```
<=50K native-country
?                         40.164760
Cambodia                  41.416667
Canada                    37.914634
China                     37.381818
Columbia                  38.684211
Cuba                      37.985714
Dominican-Republic        42.338235
Ecuador                   38.041667
El-Salvador               36.030928
England                   40.483333
France                    41.058824
Germany                   39.139785
Greece                    41.809524
Guatemala                 39.360656
Haiti                     36.325000
Holand-Netherlands        40.000000
Honduras                  34.333333
Hong                      39.142857
Hungary                   31.300000
India                     38.233333
Iran                      41.440000
Ireland                   40.947368
Italy                     39.625000
Jamaica                   38.239437
Japan                     41.000000
Laos                      40.375000
Mexico                    40.003279
Nicaragua                 36.093750
Outlying-US(Guam-USVI-etc) 41.857143
Peru                      35.068966
Philippines               38.065693
Poland                    38.166667
Portugal                  41.939394
Puerto-Rico               38.470588
Scotland                  39.444444
South                     40.156250
Taiwan                    33.774194
Thailand                  42.866667
Trinadad&Tobago           37.058824
United-States             38.799127
Vietnam                   37.193548
Yugoslavia                41.600000
Name: hours-per-week, dtype: float64
```

Рисунок 13 – Среднее время работы зарабатывающих мало по странам

Задание 2

```
In [21]: import pandas as pd

         %matplotlib inline
         import matplotlib.pyplot as plt
         import seaborn as sns

In [22]: df = pd.read_csv("./data/howpop_train.csv")

In [23]: df.drop(
             filter(lambda c: c.endswith("_lognorm"), df.columns),
             axis=1,
             inplace=True,
         )

In [24]: sns.set_style("dark")
         sns.set_palette("RdBu")
         sns.set_context(
             "notebook", font_scale=0.75, rc={"figure.figsize": (15, 5), "axes.titlesize": 18}
         )

In [25]: df["published"] = pd.to_datetime(df.published, yearfirst=True)

         df["year"] = [d.year for d in df.published]
         df["month"] = [d.month for d in df.published]
         df['day'] = [d.day for d in df.published]

         df["dayofweek"] = [d.isoweekday() for d in df.published]
         df["hour"] = [d.hour for d in df.published]
```

Рисунок 14 – Подготовительные действия над датасетом

Задача 1

В каком месяце (и какого года) было больше всего публикаций?

```
In [26]: monthly_publications = df.groupby([df['year'], df['month']]).size()
monthly_df = monthly_publications.reset_index()
monthly_df.columns = ['year', 'month', 'count']
monthly_df['year_by_months'] = [f'{month:02} {year:4}' for year, month in zip(monthly_df.year, monthly_df.month)]

sns.lineplot(data=monthly_df, x='year_by_months', y='count', hue='year')
max_monthly = monthly_df[monthly_df['count'] == monthly_df['count'].max()]

plt.vlines(max_monthly['year_by_months'], 0, max_monthly['count'], ls='--', label=max_monthly['year_by_months'].it
plt.legend(loc="lower left")

Out[26]: <matplotlib.legend.Legend at 0x70534b4c2a10>
```

Рисунок 15 – Графическое отображение максимума на графике числа публикаций в месяц

```
In [27]: monthly_of_2015 = monthly_df[monthly_df['year'] == 2015]
sns.lineplot(data=monthly_of_2015, x='month', y='count')
max_monthly_of_2015 = monthly_of_2015[monthly_of_2015['count'] == monthly_of_2015['count'].max()]

plt.vlines(max_monthly_of_2015['month'], 0, max_monthly_of_2015['count'], ls='--', label=max_monthly_of_2015['year
plt.legend(loc="lower right")

Out[27]: <matplotlib.legend.Legend at 0x70534ab868d0>
```

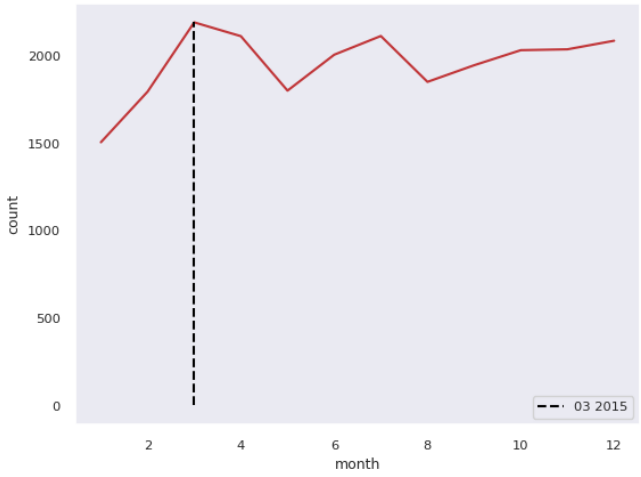

Рисунок 16 – Число публикаций в 2015 году по месяцам

Получаем, что наибольшее число публикаций было в марте 2015 года.

Задача 2

Проанализируйте публикации в месяце из предыдущего вопроса.

Выберите один или несколько вариантов:

- Один или несколько дней сильно выделяются из общей картины.
- На хабре всегда больше статей, чем на гиктаймсе.
- По субботам на гиктаймс и на хабрахабр публикуют примерно одинаковое число статей.

```
In [28]: max_year_df = df[df['year'] == max_monthly['year'].item()]
         max_month_df = max_year_df[max_year_df['month'] == max_monthly['month'].item()]

         daily_groups = max_month_df.groupby([max_month_df['day'], max_month_df['domain']]).size()

         daily_df = daily_groups.reset_index()
         daily_df.columns = ['day', 'domain', 'count']

         sns.barplot(data=daily_df, x='day', y='count', hue='domain')

Out[28]: <Axes: xlabel='day', ylabel='count'>
```

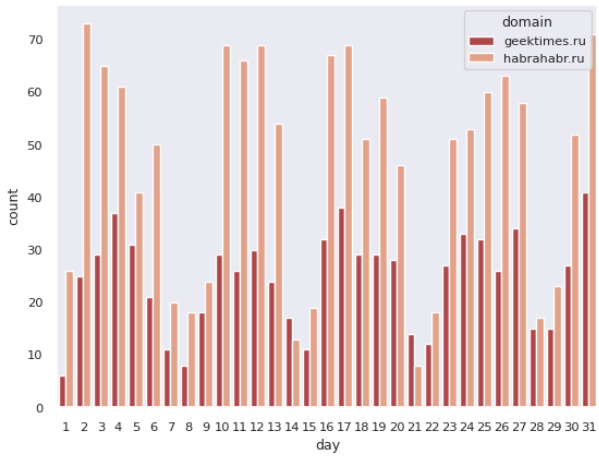

Рисунок 17 – Число публикаций в марте 2015 году по дням на разных доменах

Таким образом, график цикличен, а следовательно, ни один из дней не выделяется из общей картины.

```
In [29]: habrahabr_df = daily_df[daily_df['domain'] == 'habrahabr.ru']
         geektimes_df = daily_df[daily_df['domain'] == 'geektimes.ru']

         composed_df = habrahabr_df.merge(geektimes_df, on='day', suffixes=('_habr', '_geek'), how='inner')

         composed_df[composed_df['count_habr'] < composed_df['count_geek']]
```

Out[29]:

	day	domain_habr	count_habr	domain_geek	count_geek
13	14	habrahabr.ru	13	geektimes.ru	17
20	21	habrahabr.ru	8	geektimes.ru	14

Рисунок 18 – Дни, когда на гиктаймс было больше статей, чем на хабре

Таким образом, на хабре не всегда больше статей, чем на гиктаймсе.

```
In [30]: sat_groups = max_month_df[max_month_df['dayofweek'] == 6].groupby([max_month_df['day'], max_month_df['domain']]).s

         sat_df = sat_groups.reset_index()
         sat_df.columns = ['day', 'domain', 'count']

         sns.barplot(data=sat_df, x='day', y='count', hue='domain')

Out[30]: <Axes: xlabel='day', ylabel='count'>
```

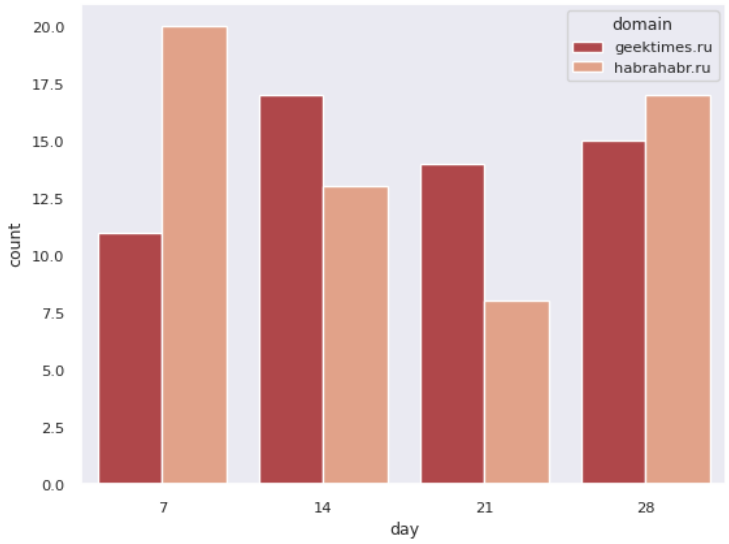

Рисунок 19 – Публикации в субботу на доменах

Таким образом, в субботу действительно на обоих доменах приблизительно одинаковое число статей.

Выбираем только пункт *«По субботам на гиктаймс и на хабрахабр публикуют примерно одинаковое число статей».*

Задача 3

Когда лучше всего публиковать статью?

- Больше всего просмотров набирают статьи, опубликованные в 12 часов дня.
- У опубликованных в 10 утра постов больше всего комментариев.
- Больше всего просмотров набирают статьи, опубликованные в 6 часов утра.
- Максимальное число комментариев на гиктаймсе набрала статья, опубликованная в 9 часов вечера.
- На хабре дневные статьи комментируют чаще, чем вечерние.

```
In [31]: hourly_publications = df.groupby([df['hour']]).mean(True)
         hourly_domain_publications = df.groupby([df['hour'], df['domain']]).mean(True)

         sns.barplot(data=hourly_publications, x='hour', y='views')

Out[31]: <Axes: xlabel='hour', ylabel='views'>
```

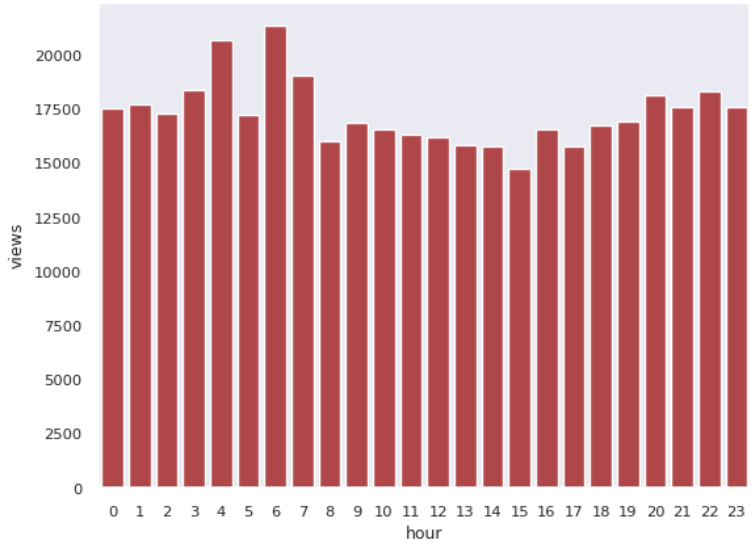

Рисунок 20 – Просмотры статей относительно времени

Таким образом, больше всего просмотров набирают статьи, опубликованные в 6 часов утра, а не в 12 часов дня.

```
In [33]: sns.barplot(data=hourly_publications, x='hour', y='comments')

Out[33]: <Axes: xlabel='hour', ylabel='comments'>
```

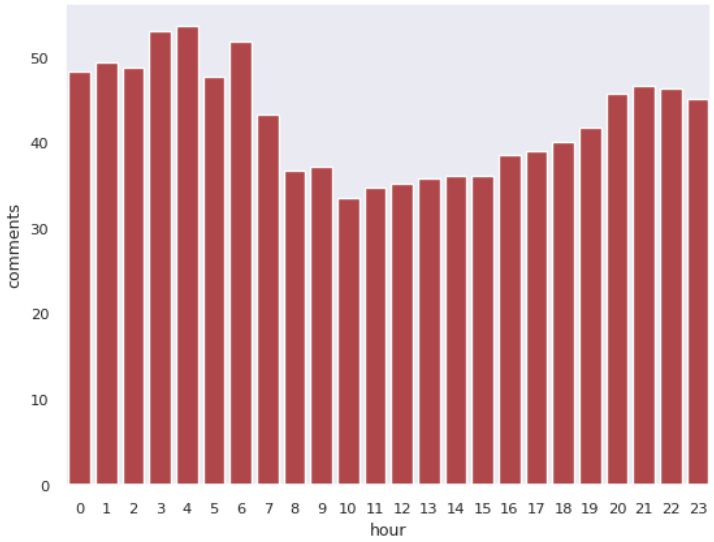

Рисунок 21 – Комментарии статей относительно времени

Таким образом, тезис «У опубликованных в 10 утра постов больше всего комментариев» опровергается.

```
In [34]: sns.barplot(data=hourly_domain_publications, x='hour', y='comments', hue='domain')
Out[34]: <Axes: xlabel='hour', ylabel='comments'>
```

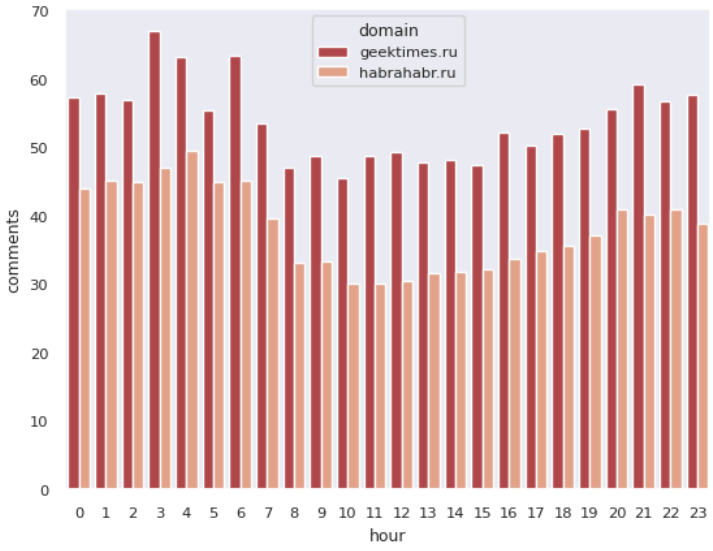

Рисунок 22 – Комментарии статей относительно времени по доменам

Таким образом, тезис «На хабре дневные статьи комментируют чаще, чем вечерние» опровергается.

```
In [35]: geektimes_hourly = df[df['domain'] == 'geektimes.ru'].groupby('hour').max(True)
         sns.barplot(data=geektimes_hourly, x='hour', y='views')
Out[35]: <Axes: xlabel='hour', ylabel='views'>
```

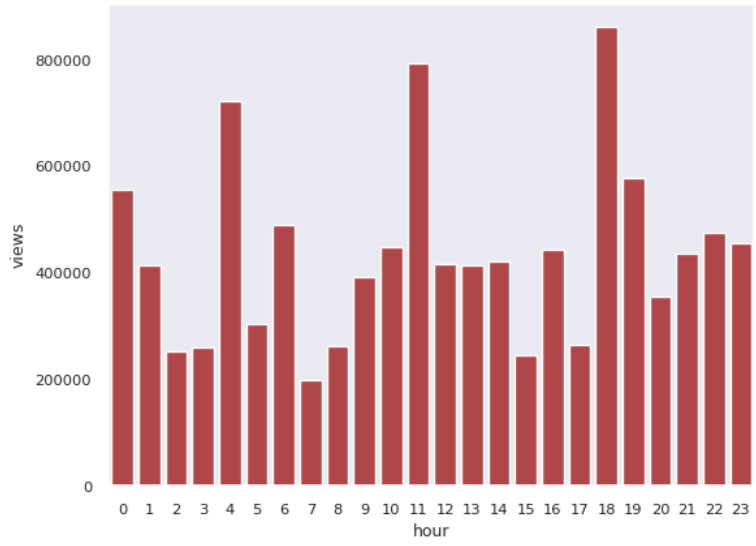

Рисунок 23 – Максимальные просмотры статей, опубликованных в заданный час, на гиктаймс

Таким образом, тезис «Максимальное число комментариев на гиктаймсе набрала статья, опубликованная в 9 часов вечера» опровергается.

Подтвердился лишь один тезис: «*Больше всего просмотров набирают статьи, опубликованные в 6 часов утра*».

Задача 4

Кого из топ-20 авторов (по количеству статей) чаще всего минусуют?

```
In [36]: top_20 = df.groupby([df['author']])['author'].count().sort_values(ascending=False).iloc[:20].keys()
         top_20

Out[36]: Index(['@alizar', '@marks', '@SLY_G', '@ivansychev', '@semen_grinshtein',
                '@jeston', '@aleksandrit', '@XaocCPS', '@Mithgol', '@Mordatyj',
                '@Shapelez', '@ilya42', '@atomlib', '@ragequit', '@Tylerskald',
                '@andorro', '@jasiejames', '@lozga', '@Sterhel', '@azazelis'],
               dtype='object', name='author')

In [37]: top_20_df = df[df['author'].isin(top_20)]
         most_minused = top_20_df.groupby(df['author'])['votes_minus'].mean().sort_values(ascending=False)[:5]
         sns.barplot(data=most_minused)

Out[37]: <Axes: xlabel='author', ylabel='votes_minus'>
```

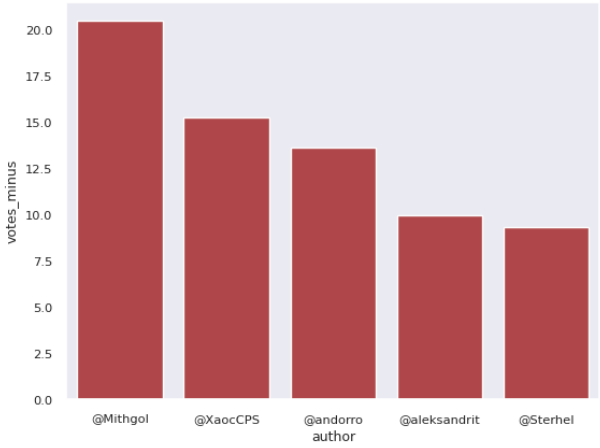

Рисунок 24 – Среднее количество минусов на статью для авторов

Таким образом, самым минусуемым автором из топ-20 является «Mithgol».

Задача 5

Сравните субботы и понедельники. Правда ли, что по субботам авторы пишут в основном днём, а по понедельникам — в основном вечером?

```
In [38]: daily_groups = df[df['dayofweek'].isin([1, 6])].groupby([df['dayofweek'], df['hour']]).size()

         daily_df = daily_groups.reset_index()
         daily_df.columns = ['dayofweek', 'hour', 'count']

         sns.barplot(data=daily_df, x='hour', y='count', hue='dayofweek')

Out[38]: <Axes: xlabel='hour', ylabel='count'>
```

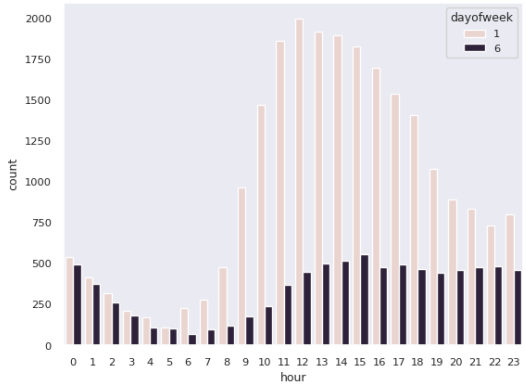

Рисунок 25 – Количество публикаций по дням недели для суббот и понедельников

Таким образом, по понедельникам и субботам пишут приблизительно в одно и то же время, но разные количества статей, так что тезис опровержен.

Вывод: в ходе выполнения лабораторной работы были получены практические навыки работы с большими массивами данных в pandas, а также навыки их визуализации с использованием библиотеки seaborn.