Департамент образования и науки города Москвы Государственное автономное образовательное учреждение высшего образования города Москвы «Московский городской педагогический университет» Институт цифрового образования Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

Инструменты для хранения и обработки больших данных

Практическая работа 01-1

Тема:

Визуализация данных из CSV-файла в DataLeans

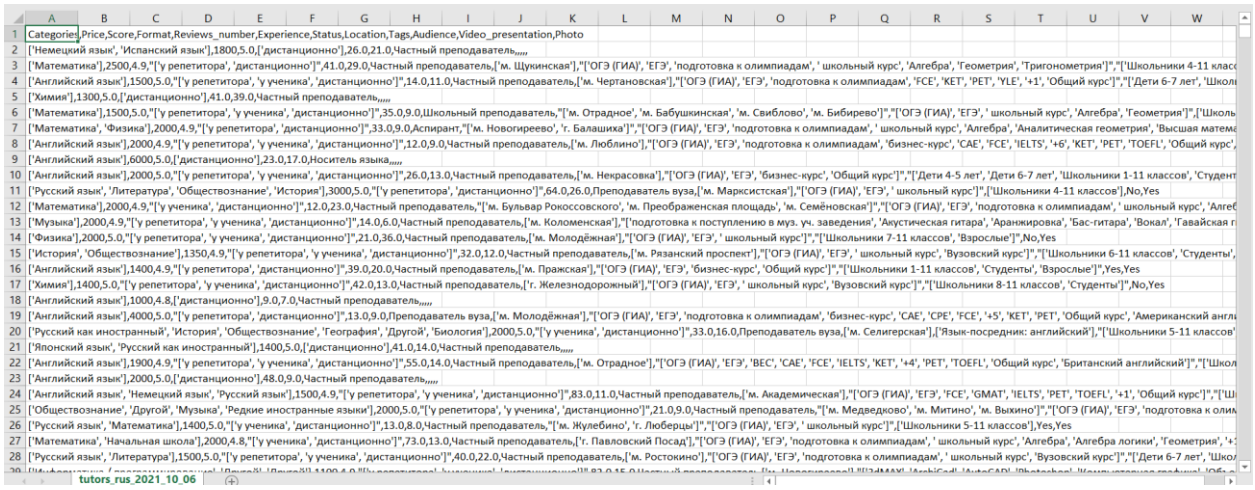
Выполнила: Соколова М. С., группа: АДЭУ-201

Преподаватель: Босенко Т. М.

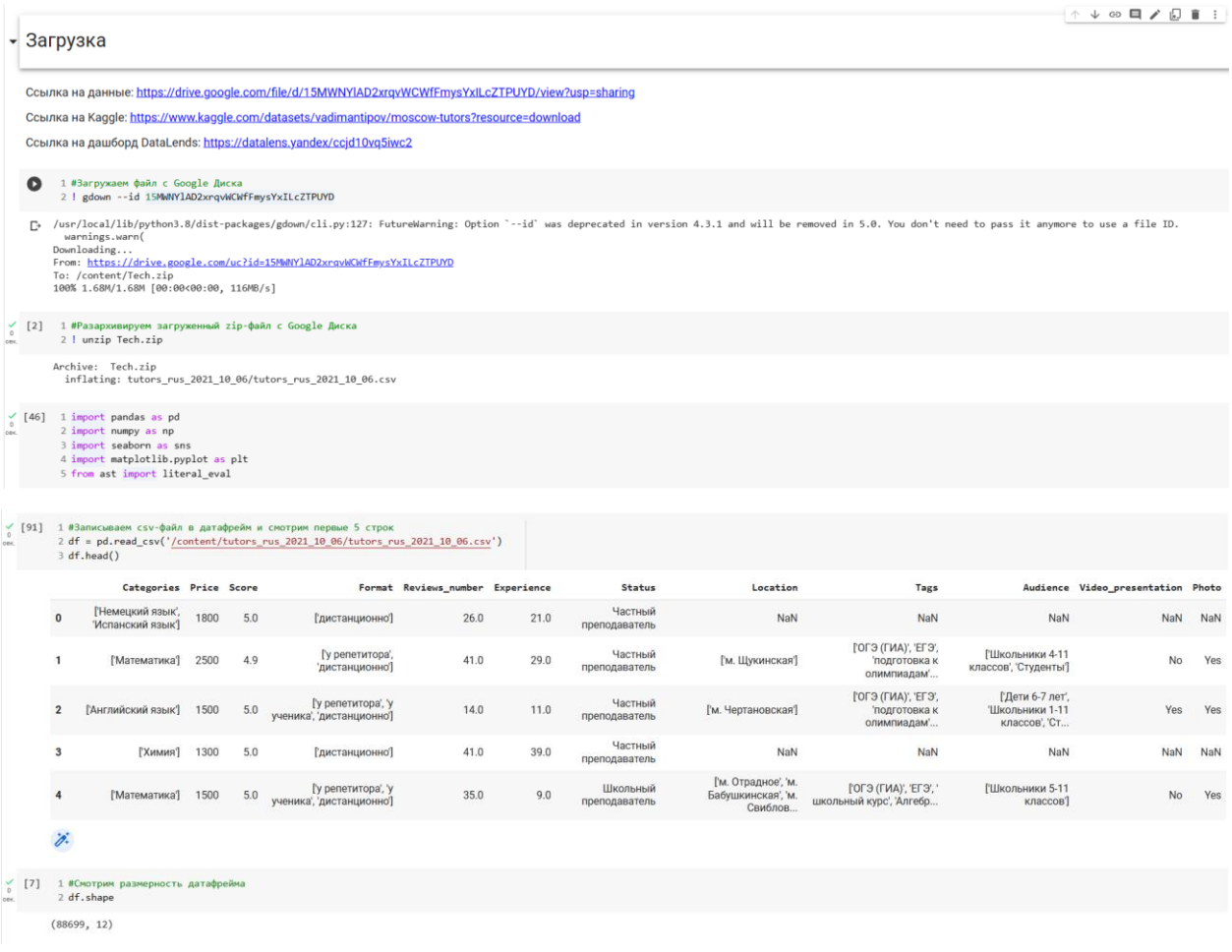
Москва

2023

Данные, которые были предоставлены со станицы [Moscow tutors | Kaggle]:



Переходим в Colab Research [Big-Data.Pr 01-1 - Colaboratory (google.com)]:



Загрузка

Ссылка на данные: https://drive.google.com/file/d/15MWNYlAD2xrqvWCWfFmysYxILcZTPUYD/view?usp=sharing

Ссылка на Kaggle: https://www.kaggle.com/datasets/vadimantipov/moscow-tutors?resource=download

Ссылка на дашборд DataLends: https://datalens.yandex/ccjd10vq5iwc2

```
1 #Загружаем файл с Google Диска
2 ! gdown --id 15MWNYlAD2xrqvWCWfFmysYxILcZTPUYD
```

```
/usr/local/lib/python3.8/dist-packages/gdown/cli.py:127: FutureWarning: Option `--id` was deprecated in version 4.3.1 and will be removed in 5.0. You don't need to pass it anymore to use a file ID.
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=15MWNYlAD2xrqvWCWfFmysYxILcZTPUYD
To: /content/Tech.zip
100% 1.68M/1.68M [00:00<00:00, 116MB/s]
```

```
[2] 1 #Разархивируем загруженный zip-файл с Google Диска
    2 ! unzip Tech.zip
```

```
Archive:  Tech.zip
  inflating: tutors_rus_2021_10_06/tutors_rus_2021_10_06.csv
```

```
[46] 1 import pandas as pd
     2 import numpy as np
     3 import seaborn as sns
     4 import matplotlib.pyplot as plt
     5 from ast import literal_eval
```

```
[91] 1 #Записываем csv-файл в датафрейм и смотрим первые 5 строк
     2 df = pd.read_csv('/content/tutors_rus_2021_10_06/tutors_rus_2021_10_06.csv')
     3 df.head()
```

	Categories	Price	Score	Format	Reviews_number	Experience	Status	Location	Tags	Audience	Video_presentation	Photo
0	['Немецкий язык', 'Испанский язык']	1800	5.0	['дистанционно']	26.0	21.0	Частный преподаватель	NaN	NaN	NaN	NaN	NaN
1	['Математика']	2500	4.9	['у репетитора', 'дистанционно']	41.0	29.0	Частный преподаватель	['м. Щукинская']	['ОГЭ (ГИА)', 'ЕГЭ', 'подготовка к олимпиадам'...	['Школьники 4-11 классов', 'Студенты']	No	Yes
2	['Английский язык']	1500	5.0	['у репетитора', 'у ученика', 'дистанционно']	14.0	11.0	Частный преподаватель	['м. Чертановская']	['ОГЭ (ГИА)', 'ЕГЭ', 'подготовка к олимпиадам'...	['Дети 6-7 лет', 'Школьники 1-11 классов', 'Ст...	Yes	Yes
3	['Химия']	1300	5.0	['дистанционно']	41.0	39.0	Частный преподаватель	NaN	NaN	NaN	NaN	NaN
4	['Математика']	1500	5.0	['у репетитора', 'у ученика', 'дистанционно']	35.0	9.0	Школьный преподаватель	['м. Отрадное', 'м. Бабушкинская', 'м. Свиблов...	['ОГЭ (ГИА)', 'ЕГЭ', ' школьный курс', 'Алгебр...	['Школьники 5-11 классов']	No	Yes

```
[7] 1 #Смотрим размерность датафрейма
    2 df.shape
```

```
(88699, 12)
```

Обработка данных

```
[8]  1 #Смотрим сколько процентов данных с пустыми значениями
     2 df.isna().mean()
```

```
Categories          0.000000
Price               0.000000
Score               0.000000
Format              0.000000
Reviews_number      0.712229
Experience          0.712229
Status              0.712229
Location            0.737393
Tags                0.737539
Audience            0.737551
Video_presentation  0.737551
Photo               0.737551
dtype: float64
```

Почти 74 % данных отсутствуют или пусты для 8 последних столбцов.

Учитывая тот факт, что набор данных большой и недостающие значения распределены по всему набору данных равномерно, мы можем удалить строки, содержащие любое количество значений Nan, и по-прежнему иметь репрезентативный большой набор данных

```
1 #Удаляем строки, в которых есть ячейки со значением NaN и восстанавливаем индексацию
2 df = df.dropna(axis=0).reset_index(drop=True)
3 df.tail()
```

	Categories	Price	Score	Format	Reviews_number	Experience	Status	Location	Tags	Audience	Video_presentation	Photo
23274	['Русский язык', 'Занятия с дошкольниками']	1300	3.6	['у ученика']	1.0	18.0	Частный преподаватель	['м. Алтуфьево']	['школьный курс']	['Школьники 1-11 классов']	No	Yes
23275	['Занятия с дошкольниками', 'Начальная школа']	1000	4.6	['у ученика']	5.0	14.0	Частный преподаватель	['м. Щёлковская']	['английский для малышей', 'Общий курс', 'Мате...	['Дети 4-5 лет', 'Дети 6-7 лет']	No	No
23276	['Английский язык', 'Русский как иностранный',...	700	4.7	['у репетитора', 'у ученика', 'дистанционно']	6.0	13.0	Частный преподаватель	['м. Университет']	['бизнес-курс', 'Общий курс']	['Дети 1-3 года', 'Дети 4-5 лет', 'Дети 6-7 ле...	No	No
23277	['Французский язык']	1000	5.0	['у ученика']	1.0	38.0	Школьный преподаватель	['г. Люберцы']	['ОГЭ (ГИА)', 'ЕГЭ', 'бизнес-курс', 'Общий курс']	['Школьники 1-11 классов', 'Студенты', 'Взросл...	No	No
23278	['Английский язык']	500	5.0	['у репетитора', 'у ученика', 'дистанционно']	1.0	4.0	Студент	['м. Курская (радиальная)', 'м. Курская (кольц...	['ОГЭ (ГИА)', 'A-Level', 'Общий курс']	['Дети 6-7 лет', 'Школьники 1-9 классов']	No	No

```
1 #Просматриваем типы данных
2 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23279 entries, 0 to 23278
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Categories          23279 non-null  object 
 1   Price               23279 non-null  int64  
 2   Score               23279 non-null  float64
 3   Format              23279 non-null  object 
 4   Reviews_number      23279 non-null  float64
 5   Experience          23279 non-null  float64
 6   Status              23279 non-null  object 
 7   Location            23279 non-null  object 
 8   Tags                23279 non-null  object 
 9   Audience            23279 non-null  object 
 10  Video_presentation  23279 non-null  object 
 11  Photo               23279 non-null  object 
dtypes: float64(3), int64(1), object(8)
memory usage: 2.1+ MB
```

Описание полей:

1. **Categories** - список преподаваемых предметов;
2. **Price** - Цена в рублях за час;
3. **Score** - Средний балл, основанный на отзывах;
4. **Format** - Варианты форматов обучения: дистанционно, у репетитора, у студента;
5. **Reviews_number** - Количество отзывов в профиле преподавателя;
6. **Experience** - Опыт в годах;
7. **Status** - Текущий статус преподавателя: Частный репетитор, Школьный учитель, Аспирант, носитель языка, профессор университета, Студент, не указано;
8. **Местоположение** - Станции метро или города Московской области;
9. **Tags** - Услуги репетитора. Они изложены преподавателями и могут отличаться;
10. **Audience** - целевая аудитория преподавателя. Например: студенты, учащиеся 10 классов и т.д;
11. **Video_presentation** - Доступность видеопрезентации;
12. **Photo** - Наличие фотографий в профиле.

Столбцы Reviews_number и Experience содержат числа с плавающей запятой, хотя должны быть целыми числами.

```
[94] 1 #Изменение типов данных
     2 df['Reviews_number'] = df['Reviews_number'].astype(int)
     3 df['Experience'] = df['Experience'].astype(int)
```

Изменяем стобец Format

```
[95] 1 df['Format'].unique()
```

```
array(["['у репетитора', 'дистанционно']",
       "['у репетитора', 'у ученика', 'дистанционно']",
       "['у ученика', 'дистанционно']", "['у репетитора']",
       "['у репетитора', 'у ученика']", "['у ученика']"], dtype=object)
```

```
[96] 1 def format(x):
     2     if x == "['у репетитора', 'дистанционно']":
     3         return 'У репетитора или дистанционно'
     4     if x == "['у репетитора', 'у ученика', 'дистанционно']":
     5         return 'Любой формат'
     6     if x == "['у ученика', 'дистанционно']":
     7         return 'У ученика или дистанционно'
     8     if x == "['у ученика']":
     9         return 'У ученика'
    10     if x == "['у репетитора']":
    11         return 'У репетитора'
    12     if x == "['у репетитора', 'у ученика']":
    13         return 'У репетитора или ученика'
```

```
[97] 1 df['Format_new'] = df['Format'].apply(format)
```

Разбиваем столбец Price на три сегмента:

1. Низкий ценовой сегмент
2. Средний ценовой сегмент
3. Высокий ценовой сегмент

```
[98] 1 df['Price_group_count'] = pd.qcut(df['Price'], 3)
```

Удаляем столбец Tags, т.к он не пригодиться нам для анализа

```
[99] 1 df = df.drop('Tags', axis=1)
```

```
[100] 1 df.head()
```

	Categories	Price	Score	Format	Reviews_number	Experience	Status	Location	Audience	Video_presentation	Photo	Format_new	Price_group_count
0	['Математика']	2500	4.9	['у репетитора', 'дистанционно']	41	29	Частный преподаватель	['м. Щукинская']	['Школьники 4-11 классов', 'Студенты']	No	Yes	У репетитора или дистанционно	(1100.0, 6700.0]
1	['Английский язык']	1500	5.0	['у репетитора', 'у ученика', 'дистанционно']	14	11	Частный преподаватель	['м. Чертановская']	['Дети 6-7 лет', 'Школьники 1-11 классов', 'Ст...	Yes	Yes	Любой формат	(1100.0, 6700.0]
2	['Математика']	1500	5.0	['у репетитора', 'у ученика', 'дистанционно']	35	9	Школьный преподаватель	['м. Отрадное', 'м. Бабушкинская', 'м. Свиблов...	['Школьники 5-11 классов']	No	Yes	Любой формат	(1100.0, 6700.0]
3	['Математика', 'Физика']	2000	4.9	['у репетитора', 'дистанционно']	33	9	Аспирант	['м. Новогиреево', 'г. Балашиха']	['Школьники 7-11 классов', 'Студенты', 'Взросл...	No	Yes	У репетитора или дистанционно	(1100.0, 6700.0]
4	['Английский язык']	2000	4.9	['у репетитора', 'у ученика', 'дистанционно']	12	9	Частный преподаватель	['м. Люблино']	['Дети 1-3 года', 'Дети 4-5 лет', 'Дети 6-7 ле...	Yes	Yes	Любой формат	(1100.0, 6700.0]

Анализ

```
[20] 1 #Просматриваем ститистику
     2 df.describe()
```

	Price	Score	Reviews_number	Experience
count	23279.000000	23279.000000	23279.000000	23279.000000
mean	1075.114910	4.892109	6.214614	15.244340
std	511.798633	0.229869	9.177704	9.929735
min	150.000000	2.200000	0.000000	0.000000
25%	700.000000	4.900000	1.000000	8.000000
50%	1000.000000	5.000000	3.000000	13.000000
75%	1300.000000	5.000000	7.000000	20.000000
max	6700.000000	5.000000	172.000000	64.000000

Корреляция Пирсона

```
[30] 1 df.corr()
```

	Price	Score	Reviews_number	Experience
Price	1.000000	0.040723	0.159178	0.144693
Score	0.040723	1.000000	-0.002893	0.005044
Reviews_number	0.159178	-0.002893	1.000000	0.178781
Experience	0.144693	0.005044	0.178781	1.000000

```
[51]  1 sns.heatmap(df.corr(), annot = True)
```

```
<AxesSubplot:>
```

Корреляция Фика

```
[32]  1 #Устанавливаем библиотеку phik
      2 ! pip install phik
```

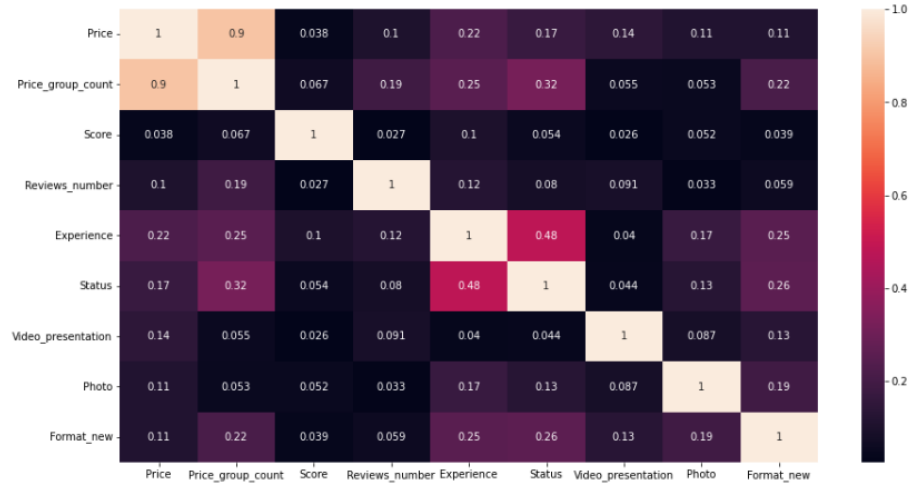
```
[41]  1 #Подключаем библиотеку
      2 import phik
      3 from phik.report import plot_correlation_matrix
      4 from phik import report
```

```
[55]  1 #Создаем матрицу корреляции
      2 phik_overview = df[['Price', 'Price_group_count', 'Score',  'Reviews_number', 'Experience', 'Status', 'Video_presentation', 'Photo', 'Format_new']].phik_matrix()
```

```
interval columns not set, guessing: ['Price', 'Score', 'Reviews_number', 'Experience']
```

```
[56]  1 plt.figure(figsize = (15,8))
      2 sns.heatmap(phik_overview, annot = True)
```

```
<AxesSubplot:>
```



Группировка

Цена

```
[58] 1 #Делаем группировку по столбцы Price_group_count, добавляя среднее значение показателей
     2 price = df.groupby('Price_group_count',dropna=False)[['Price', 'Score', 'Experience', 'Reviews_number']].agg(['count', 'mean'])
     3 #Убираем двухэтажные названия
     4 price.columns = ['_'.join(col).strip() for col in price.columns.values]
     5 #Удаляем повторяющиеся столбецы количества
     6 price = price.drop(columns=['Score_count', 'Experience_count', 'Reviews_number_count'])
     7 #Переименовываем столбцы
     8 price = price.rename(columns={'Price_count':'Количество', 'Price_mean':'Средняя цена', 'Score_mean':'Средняя оценка',
     9                              'Experience_mean':'Средний страж', 'Reviews_number_mean':'Среднее кол-во отзывов'})
    10 price
```

Price_group_count	Количество	Средняя цена	Средняя оценка	Средний страж	Среднее кол-во отзывов
(149.999, 800.0]	8420	639.536817	4.881033	13.291093	4.502850
(800.0, 1100.0]	7101	990.550627	4.892860	15.462048	6.342487
(1100.0, 6700.0]	7758	1625.264243	4.903442	17.164991	7.955401

Можем увидеть четкую зависимость между всеми метриками и ценовым сегментом. Чем выше стоимость, тем выше все показатели.

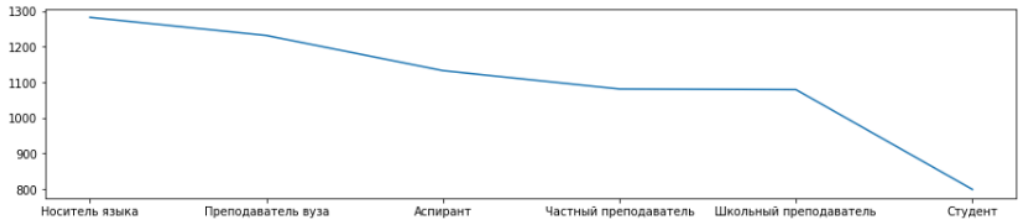
Статус репетитора

```
[59] 1 #Делаем группировку по столбцы Status, добавляя среднее значение показателей
     2 status = df.groupby('Status',dropna=False)[[ 'Price', 'Score', 'Experience', 'Reviews_number']].agg(['count', 'mean'])
     3 #Убираем двухэтажные названия
     4 status.columns = ['_'.join(col).strip() for col in status.columns.values]
     5 #Сортируем по цене по убыванию
     6 status = status.sort_values(by = 'Price_mean', ascending = False)
     7 #Удаляем повторяющиеся столбецы количества
     8 status = status.drop(columns=['Score_count', 'Experience_count', 'Reviews_number_count'])
     9 #Переименовываем столбцы
    10 status = status.rename(columns={'Price_count':'Количество', 'Price_mean':'Средняя цена', 'Score_mean':'Средняя оценка',
    11                               'Experience_mean':'Средний страж', 'Reviews_number_mean':'Среднее кол-во отзывов'})
    12 status
```

Status	Количество	Средняя цена	Средняя оценка	Средний страж	Среднее кол-во отзывов
Носитель языка	438	1282.305936	4.826484	10.678082	5.392694
Преподаватель вуза	1862	1231.364125	4.900859	21.162728	8.264232
Аспирант	167	1132.634731	4.905988	7.365269	5.263473
Частный преподаватель	16013	1081.012303	4.891482	15.420783	6.364766
Школьный преподаватель	2994	1079.325317	4.909920	18.608550	6.395792
Студент	1805	799.030471	4.873740	3.830471	2.755125

```
1 plt.figure(figsize = (15, 3))
2 status['Средняя цена'].plot()
```

```
<AxesSubplot:xlabel='Status'>
```



Можем увидеть, что максимальная средняя стоимость у носителя языка, а минимальная у студента, это логично.

Формат

```
[72]  1 #Делаем группировку по столбцу Format_new, добавляя среднее значение показателей
      2 format = df.groupby('Format_new',dropna=False)[[ 'Price', 'Score', 'Experience', 'Reviews_number']].agg(['count', 'mean'])
      3 #Убираем двухэтажные названия
      4 format.columns = ['_'.join(col).strip() for col in format.columns.values]
      5 #Сортируем по цене по убыванию
      6 format = format.sort_values(by = 'Price_mean', ascending = False)
      7 #Удаляем повторяющиеся столбецы количества
      8 format = format.drop(columns=['Score_count', 'Experience_count', 'Reviews_number_count'])
      9 #Удаляем повторяющиеся столбецы количества
     10 format = format.rename(columns={'Price_count':'Количество', 'Price_mean':'Средняя цена', 'Score_mean':'Средняя оценка',
     11                               'Experience_mean':'Средний страж', 'Reviews_number_mean':'Среднее кол-во отзывов'})
     12 format
```

Format_new	Количество	Средняя цена	Средняя оценка	Средний страж	Среднее кол-во отзывов
У ученика	2292	1143.542757	4.880846	15.194154	4.370419
У репетитора или ученика	2720	1138.216912	4.878015	18.522426	5.132353
У репетитора	921	1123.778502	4.899783	23.676439	5.685125
У репетитора или дистанционно	3619	1118.872617	4.914396	17.508151	7.168555
Любой формат	9138	1053.250164	4.892362	14.351499	6.809696
У ученика или дистанционно	4589	1002.800174	4.886468	11.626716	5.946176

Можем увидеть, что максимальная стоимость при формате обучения у ученика, а максимальная стоимость и максимальная оценка у репетитора или дистанционно.

Отдельные сводные таблицы

Формат

```
[101]  1 df['Format']
```

```
0                     ['у репетитора', 'дистанционно']
1        ['у репетитора', 'у ученика', 'дистанционно']
2        ['у репетитора', 'у ученика', 'дистанционно']
3                     ['у репетитора', 'дистанционно']
4        ['у репетитора', 'у ученика', 'дистанционно']
                             ...
23274                                  ['у ученика']
23275                                  ['у ученика']
23276    ['у репетитора', 'у ученика', 'дистанционно']
23277                                  ['у ученика']
23278    ['у репетитора', 'у ученика', 'дистанционно']
Name: Format, Length: 23279, dtype: object
```

```
[113]  1 #Удаляем лишние знаки и преобразовываем в список столбец Format
       2 for i in ['Format']:
       3     df[i] = df[i].apply(lambda s: list(literal_eval(str(s))) if s != np.nan else s)
       4 #Разделяем каждый список на отдельные строки
       5 format_study = df['Format'].explode()
       6 #Добавляем сводную таблицу
       7 format_study_preprocessed = df.join(pd.crosstab(format_study.index, format_study))
       8 #Выбираем нужные столбцы и считаем сумму
       9 format_list = format_study_preprocessed.columns[13:16]
      10 df1 = format_study_preprocessed[format_list].sum()
      11 df1
```

```
дистанционно    17346
у репетитора    16398
у ученика       18739
dtype: int64
```

Аудитория

```
for i in ['Audience']:
    df[i] = df[i].apply(lambda s: list(literal_eval(str(s))) if s != np.nan else s)

audience_series = df['Audience'].explode()
audience_series_preprocessed = df.join(pd.crosstab(audience_series.index, audience_series))

audience_series_list = audience_series_preprocessed.columns[13:100]
df2 = audience_series_preprocessed[audience_series_list].sum()
df2
```

```
10 классов                15
10-11 классов             11
11 классов                21
3-11 классов               3
4-11 классов               4
                         ...
Школьники 5-8 классов     74
Школьники 5-9              4
Школьники 5-9 классов    357
Школьники 6                1
Школьники 6-10 классов    40
Length: 87, dtype: int64
```

Курсы

Посмотрим топ-10 самых популярных предметов

```
for i in ['Categories']:
    df[i] = df[i].apply(lambda s: list(literal_eval(str(s))))

categories_series = df['Categories'].explode()

tutors_data_preprocessed = df.join(pd.crosstab(categories_series.index, categories_series))

categories_list = tutors_data_preprocessed.columns[13:39]
tutors_data_preprocessed[categories_list].sum()

df3 = tutors_data_preprocessed[categories_list].sum()
df3.sort_values(ascending=False).head(10)
```

```
Английский язык           7484
Математика                5800
Русский язык              3331
Физика                    2112
Музыка                    2026
Начальная школа           1756
Другой                    1580
Химия                     1473
Занятия с дошкольниками   1432
Обществознание            1393
dtype: int64
```

Выгрузка

```
from google.colab import files
df.to_csv('df.csv')
files.download('df.csv')
```

```
from google.colab import files
df1.to_csv('df1.csv')
files.download('df1.csv')
```

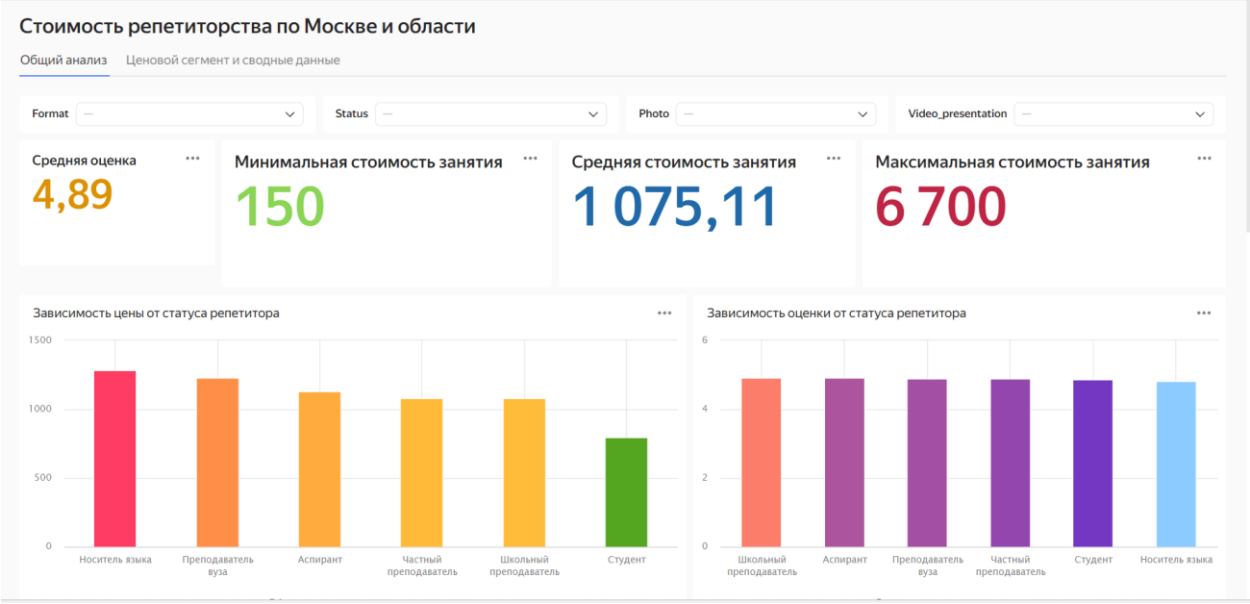
```
from google.colab import files
df2.to_csv('df2.csv')
files.download('df2.csv')
```

```
from google.colab import files
df3.to_csv('df3.csv')
files.download('df3.csv')
```

Данные после обработки:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Column1	Categories	Price	Score	Format	Reviews_n	Experience	Status	Location	Audience	Video_pre	Photo	Format_ne	Price_group_count		
2	0	['Математ	2500	4.9	['у репети	41	29	Частный г	['м. Щукин	['Школьни	No	Yes	У репетит	(1100.0, 6700.0]		
3	1	['Английс	1500	5.0	['у репети	14	11	Частный г	['м. Черта	['Дети 6-7	Yes	Yes	Любой фс	(1100.0, 6700.0]		
4	2	['Математ	1500	5.0	['у репети	35	9	Школьный	['м. Отрад	['Школьни	No	Yes	Любой фс	(1100.0, 6700.0]		
5	3	['Математ	2000	4.9	['у репети	33	9	Аспирант	['м. Новог	['Школьни	No	Yes	У репетит	(1100.0, 6700.0]		
6	4	['Английс	2000	4.9	['у репети	12	9	Частный г	['м. Любл	['Дети 1-3	Yes	Yes	Любой фс	(1100.0, 6700.0]		
7	5	['Английс	2000	5.0	['у репети	26	13	Частный г	['м. Некра	['Дети 4-5	Yes	Yes	Любой фс	(1100.0, 6700.0]		
8	6	['Русский	3000	5.0	['у репети	64	26	Преподав	['м. Маркс	['Школьни	No	Yes	У репетит	(1100.0, 6700.0]		
9	7	['Математ	2000	4.9	['у ученик	12	23	Частный г	['м. Бульв	['Школьни	Yes	Yes	У ученика	(1100.0, 6700.0]		
10	8	['Музыка']	2000	4.9	['у репети	14	6	Частный г	['м. Колом	['Дети 1-3	Yes	Yes	Любой фс	(1100.0, 6700.0]		
11	9	['Физика']	2000	5.0	['у репети	21	36	Частный г	['м. Моло	['Школьни	No	Yes	Любой фс	(1100.0, 6700.0]		
12	10	['История	1350	4.9	['у репети	32	12	Частный г	['м. Рязан	['Школьни	No	Yes	Любой фс	(1100.0, 6700.0]		
13	11	['Английс	1400	4.9	['у репети	39	20	Частный г	['м. Пражс	['Школьни	Yes	Yes	У репетит	(1100.0, 6700.0]		
14	12	['Химия']	1400	5.0	['у репети	42	13	Частный г	['г. Железн	['Школьни	No	Yes	Любой фс	(1100.0, 6700.0]		
15	13	['Английс	4000	5.0	['у репети	13	9	Преподав	['м. Моло	['Дети 6-7	Yes	Yes	У репетит	(1100.0, 6700.0]		
16	14	['Русский	2000	5.0	['у ученик	33	16	Преподав	['м. Селиг	['Школьни	No	Yes	У ученика	(1100.0, 6700.0]		
17	15	['Английс	1900	4.9	['у репети	55	14	Частный г	['м. Отрад	['Школьни	No	Yes	Любой фс	(1100.0, 6700.0]		
18	16	['Английс	1500	4.9	['у репети	83	11	Частный г	['м. Акаде	['Школьни	No	Yes	Любой фс	(1100.0, 6700.0]		
19	17	['Обществ	2000	5.0	['у репети	21	9	Частный г	['м. Медв	['Школьни	Yes	Yes	Любой фс	(1100.0, 6700.0]		
20	18	['Русский	1400	5.0	['у ученик	13	8	Частный г	['м. Жулеб	['Школьни	Yes	Yes	У ученика	(1100.0, 6700.0]		
21	19	['Математ	2000	4.8	['у ученик	73	13	Частный г	['г. Павло	['Школьни	No	Yes	У ученика	(1100.0, 6700.0]		
22	20	['Русский	1500	5.0	['у репети	40	22	Частный г	['м. Росто	['Дети 6-7	No	Yes	Любой фс	(1100.0, 6700.0]		
23	21	['Информа	1100	4.9	['у репети	82	15	Частный г	['м. Новог	['Студенть	No	Yes	Любой фс	(800.0, 1100.0]		
24	22	['Математ	2000	5.0	['у репети	13	6	Частный г	['м. Улице	['Школьни	Yes	Yes	Любой фс	(1100.0, 6700.0]		
25	23	['Изобраз	1000	5.0	['у репети	63	12	Частный г	['м. Арбат	['Школьни	No	Yes	У репетит	(800.0, 1100.0]		
26	24	['Изобраз	600	5.0	['у репети	17	3	Студент	['м. Селиг	['Дети 1-3	Yes	Yes	Любой фс	(149.999, 800.0]		
27	25	['Английс	1500	5.0	['у репети	20	5	Частный г	['м. Дмитр	['Дети 1-3	Yes	Yes	Любой фс	(1100.0, 6700.0]		
28	26	['Математ	2000	4.9	['у репети	33	49	Частный г	['м. Алекс	['Школьни	Yes	Yes	У репетит	(1100.0, 6700.0]		

Переходим в DataLens: https://datalens.yandex/ccjd10vq5iwc2



Средняя стоимость в зависимости от наличия фото и видео

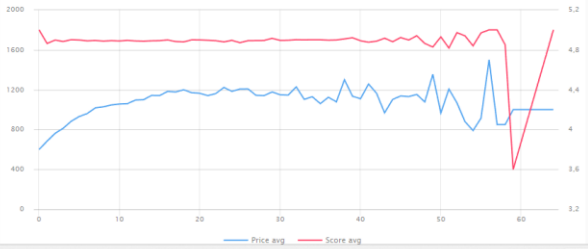
Photo	Video_presentation	Price avg
Yes	Yes	1 454,64
	No	1 082,53
No	Yes	600,00
	No	945,92

Варианты режимов обучения

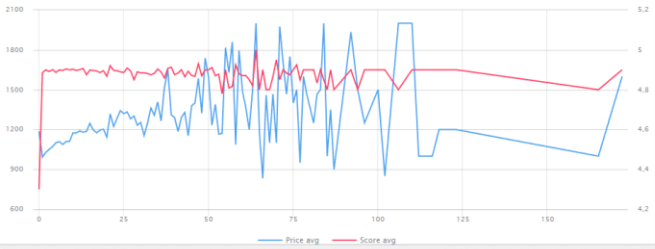
Наличие фото в профиле

Наличие видеопрезентации

Зависимость цены и оценок от опыта

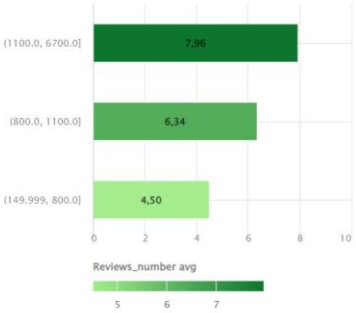


Зависимость цены и оценок от наличия отзывов



Общий анализ | Ценовой сегмент и сводные данные

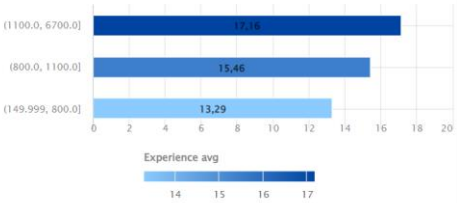
Средняя количество отзывов по ценовым сегме...



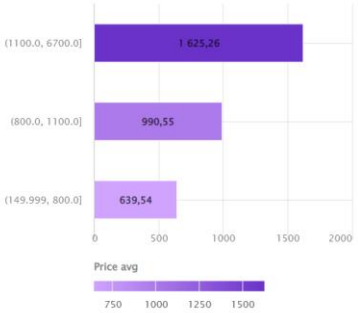
Ценовой сегмент влияет на все метрики

Можем увидеть зависимость всех метрик от ценового сегмента. Чем выше сегмент, тем лучше показатели.

Средний стаж по ценовым сегментам



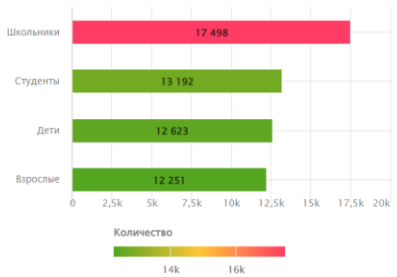
Средняя цена по ценовым сегментам



Сводные данные

Доля форматов обучения

Аудитория



Топ-5 предметов