

Анализ удовлетворённости студентов учебным процессом с использованием Python и интерактивной визуализации данных

Екимова Яна Сергеевна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной научной статье рассматривается пошаговый анализ данных опроса студентов с использованием библиотек Python. Рассмотрено, как загружать данные, вычислять описательную статистику, строить интерактивные графики (гистограммы, boxplot, тепловые карты и круговые диаграммы) и интерпретировать результаты с использованием Google Colaboratory. Для достижения поставленной цели использовался набор данных, содержащий результаты опросов студентов.

Ключевые слова: опрос, Python, pandas, seaborn, plotly, визуализация данных.

Analysis of student satisfaction with the learning process using Python and interactive data visualization

Ekimova Yana Sergeevna

Sholom Aleichem Priamurskiy State University

Student

Abstract

This scientific article discusses a step-by-step analysis of student survey data using Python libraries. It discusses how to download data, calculate descriptive statistics, build interactive graphs (histograms, boxplot, heat maps, and pie charts), and interpret the results using Google Colaboratory. To achieve this goal, a data set containing the results of student surveys was used.

Keywords: survey, Python, pandas, seaborn, plotly, data visualization.

1 Введение

1.1 Актуальность

В современном образовательном процессе важно понимать, как студенты воспринимают различные аспекты учебного курса, чтобы своевременно выявлять сильные и слабые стороны и принимать меры для улучшения качества обучения. Опросы студентов являются одним из ключевых инструментов для сбора обратной связи, однако анализ таких данных требует применения современных методов интеллектуальной обработки и визуализации. Использование библиотек Python, таких как pandas, seaborn и plotly, позволяет не только автоматизировать процесс

анализа, но и создавать интерактивные визуализации, которые делают результаты более наглядными и удобными для интерпретации. Данная статья демонстрирует, как с помощью инструментов анализа данных можно эффективно обрабатывать и визуализировать результаты опросов, чтобы получить ценные идеи для улучшения учебного процесса. Обратная связь от студентов является важным источником информации для оценки качества образовательных программ. Однако ручной анализ данных опросов может быть трудоёмким и недостаточно эффективным. Современные методы анализа данных, такие как описательная статистика, корреляционный анализ и визуализация, позволяют автоматизировать этот процесс и выявить ключевые закономерности.

1.2 Обзор исследований

Э. А.Насыпова, И. Ф. Развеева применили методы машинного обучения для формирования портрета клиента психологического центра [1]. Разведочный анализ данных о прогнозировании энергопотребления провела А.С. Матвеева [2]. А. А. Антонов провел прогнозирование размера заработной платы с использованием методов машинного обучения по обработке естественного языка [3]. Качественный анализ данных сделали А. Е. Войскунский, С. В. Скрипкин [4]. Е. А. Григорьев, Н. С. Климов привели пример разведочного анализа данных с помощью python [5].

1.3 Цель исследования

Цель исследования – выполнить анализ данных с помощью Google Colaboratory.

2 Материалы и методы

В данном исследовании используется Google Colaboratory для анализа данных. Набор данных о результатах опроса студентов. Он охватывает обратную связь, организацию курса, техническую поддержку, качество преподавания, материал лекций, практические задания, сложность курса. Данные для работы можно скачать по ссылке: https://drive.google.com/file/d/1cmX_nrZZEXbKOx-3o7ZqAdPMF5UhO1eM/view?usp=sharing.

Ссылка на ноутбук Google Colaboratory: https://colab.research.google.com/drive/1nuYUIO0t4sFyTm6FaPk_8PBZMchX4tI?usp=sharing.

3 Результаты

Для проведения интеллектуального анализа данных опроса студентов, импортируем нужные библиотеки и модели (рис.1).

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

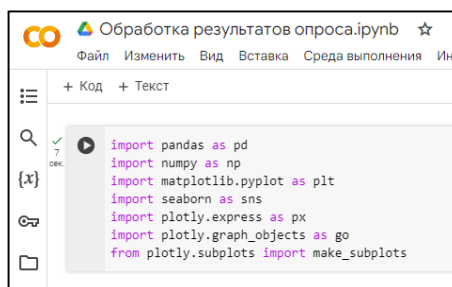


Рисунок 1. Импорт библиотек

Загрузим данные опроса из CSV-файла с использованием библиотеки `pandas`. Первый столбец данных содержит имена студентов, а последующие столбцы — ответы на вопросы опроса, оцененные по шкале от 1 до 5. CSV-файл содержит следующие столбцы:

- **name** — имя студента (строковый тип данных).
- **столбец 2** — ответ на вопрос 1: "Как вы оцениваете качество преподавания?" (целочисленный тип данных, значения от 1 до 5).
- **столбец 3** — ответ на вопрос 2: "Как вы оцениваете материалы курса?" (целочисленный тип данных, значения от 1 до 5).
- **столбец 4** — ответ на вопрос 3: "Насколько интересны лекции?" (целочисленный тип данных, значения от 1 до 5).
- **столбец 5** — ответ на вопрос 4: "Насколько полезны практические задания?" (целочисленный тип данных, значения от 1 до 5).
- **столбец 6** — ответ на вопрос 5: "Как вы оцениваете уровень сложности курса?" (целочисленный тип данных, значения от 1 до 5).
- **столбец 7** — ответ на вопрос 6: "Как вы оцениваете обратную связь от преподавателя?" (целочисленный тип данных, значения от 1 до 5).
- **столбец 8** — ответ на вопрос 7: "Насколько комфортно вам было учиться?" (целочисленный тип данных, значения от 1 до 5).
- **столбец 9** — ответ на вопрос 8: "Как вы оцениваете организацию курса?" (целочисленный тип данных, значения от 1 до 5).
- **столбец 10** — ответ на вопрос 9: "Как вы оцениваете техническую поддержку курса?" (целочисленный тип данных, значения от 1 до 5).
- **столбец 11** — ответ на вопрос 10: "Порекомендуете ли вы этот курс другим?" (целочисленный тип данных, значения от 1 до 5).

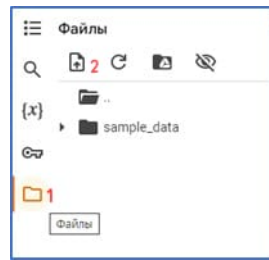


Рисунок 2. Загрузка файла

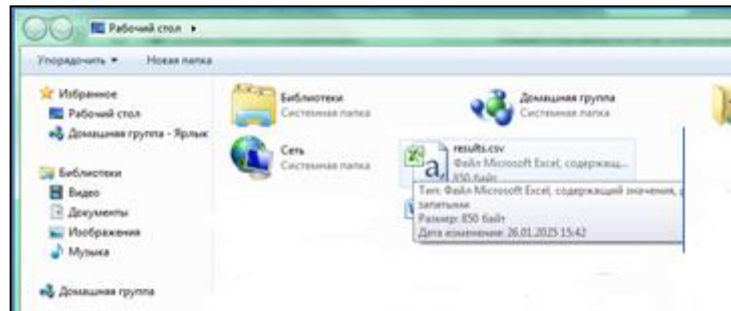


Рисунок 3. Выбор нужного файла

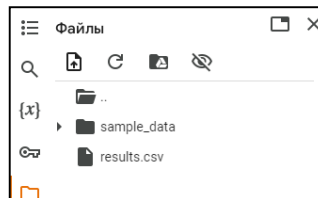


Рисунок 4. Загрузка файла

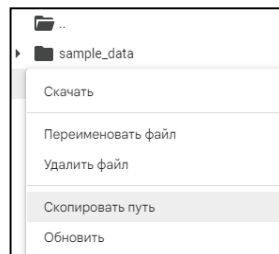


Рисунок 5. Копируем путь файла

```
# Загрузка данных из CSV-файла
url = '/content/results.csv' # Замените на путь к вашему файлу
data = pd.read_csv(url)
```

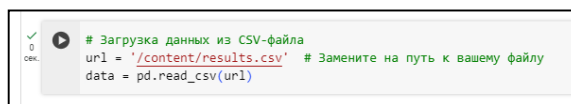


Рисунок 6. Загрузка данных

Для ознакомления с данными выведем первые строки таблицы и имена столбцов.

```
# Просмотр первых строк данных
print("Первые строки данных:")
```

```
print(data.head())
```

```
# Просмотр первых строк данных
print("Первые строки данных:")
print(data.head())
```

Первые строки данных:

	Имя	Качество преподавания	Материалы курса	Интерес лекций	\
0	Павел	5	5	5	
1	Павел	3	4	3	
2	Мария	2	3	4	
3	Яна	3	2	4	
4	Владимир	3	2	4	

	Практические задания	Сложность курса	Обратная связь	Комфорт обучения	\
0	5	5	5	5	
1	5	5	5	1	
2	4	5	5	2	
3	4	3	5	2	
4	5	3	5	4	

	Организация курса	Техническая поддержка	Рекомендация курса	\
0	5	5	5	
1	5	4	5	
2	2	2	3	
3	2	1	2	
4	2	4	5	

Рисунок 7. Просмотр первых строк

Теперь напишем код, который используется для вычисления и вывода **описательной статистики** по числовым столбцам данных. Описательная статистика предоставляет основные метрики, которые помогают понять распределение и характеристики данных. Это важный этап предварительного анализа данных, который позволяет исследователю получить общее представление о наборе данных.

```
# Описательная статистика
print("\nОписательная статистика:")
print(data.describe())
```

```
# Описательная статистика
print("\nОписательная статистика:")
print(data.describe())
```

Описательная статистика:

	Качество преподавания	Материалы курса	Интерес лекций	\
count	15.000000	15.000000	15.000000	
mean	3.266667	3.600000	3.666667	
std	1.162919	1.055597	0.617213	
min	1.000000	2.000000	3.000000	
25%	3.000000	3.000000	3.000000	
50%	3.000000	4.000000	4.000000	
75%	4.000000	4.000000	4.000000	
max	5.000000	5.000000	5.000000	

	Практические задания	Сложность курса	Обратная связь	\
count	15.000000	15.000000	15.000000	
mean	3.733333	4.133333	4.200000	
std	1.099784	0.915475	0.774597	
min	2.000000	2.000000	3.000000	
25%	3.000000	4.000000	4.000000	
50%	4.000000	4.000000	4.000000	
75%	5.000000	5.000000	5.000000	
max	5.000000	5.000000	5.000000	

	Комфорт обучения	Организация курса	Техническая поддержка	\
count	15.000000	15.000000	15.000000	
mean	3.733333	4.133333	4.200000	
std	1.099784	0.915475	0.774597	
min	2.000000	2.000000	3.000000	
25%	3.000000	4.000000	4.000000	
50%	4.000000	4.000000	4.000000	
75%	5.000000	5.000000	5.000000	
max	5.000000	5.000000	5.000000	

Рисунок 8. Статистика по числовым столбцам данных

+ Код	+ Текст	5.000000	4.000000	3.000000
50%		4.000000	4.000000	4.000000
75%		5.000000	5.000000	5.000000
max		5.000000	5.000000	5.000000
Комфорт обучения				
count		15.000000	15.000000	15.000000
mean		3.333333	3.733333	3.533333
std		1.290994	1.279881	1.302013
min		1.000000	2.000000	1.000000
25%		2.000000	3.000000	3.000000
50%		4.000000	3.000000	4.000000
75%		4.000000	5.000000	4.000000
max		5.000000	5.000000	5.000000
Организация курса				
count		15.000000	15.000000	15.000000
mean		3.333333	3.733333	3.533333
std		1.290994	1.279881	1.302013
min		1.000000	2.000000	1.000000
25%		2.000000	3.000000	3.000000
50%		4.000000	3.000000	4.000000
75%		4.000000	5.000000	4.000000
max		5.000000	5.000000	5.000000
Техническая поддержка				
count		15.000000	15.000000	15.000000
mean		3.333333	3.733333	3.533333
std		1.290994	1.279881	1.302013
min		1.000000	2.000000	1.000000
25%		2.000000	3.000000	3.000000
50%		4.000000	3.000000	4.000000
75%		4.000000	5.000000	4.000000
max		5.000000	5.000000	5.000000
Рекомендация курса				
count		15.000000	15.000000	15.000000
mean		3.800000	3.800000	3.800000
std		1.373213	1.373213	1.373213
min		1.000000	1.000000	1.000000
25%		3.000000	3.000000	3.000000
50%		4.000000	4.000000	4.000000
75%		5.000000	5.000000	5.000000
max		5.000000	5.000000	5.000000

Рисунок 9. Статистика по числовым столбцам данных

Результаты описательной статистики позволяют сделать следующие выводы о восприятии студентами различных аспектов учебного курса:

- Качество преподавания получило среднюю оценку 3.27, что указывает на умеренную удовлетворённость студентов. Стандартное отклонение (1.16) показывает значительный разброс мнений: некоторые студенты оценили преподавание как "отличное" (5.0), а другие — как "плохое" (1.0). Медиана (3.0) подтверждает, что большинство студентов оценили преподавание как "удовлетворительное".

- Материалы курса получили более высокую оценку (3.60), чем качество преподавания. Медиана (4.0) указывает на то, что большинство студентов оценили материалы как "хорошие". Разброс оценок ($std = 1.06$) меньше, чем для качества преподавания, что говорит о более согласованном мнении.

- Лекции получили среднюю оценку 3.67, что указывает на умеренный интерес студентов. Низкое стандартное отклонение (0.62) свидетельствует о том, что мнения студентов относительно интереса лекций достаточно согласованы. Медиана (4.0) подтверждает, что большинство студентов считают лекции "интересными".

- Практические задания получили оценку 3.73, что указывает на их полезность для студентов. Разброс оценок ($std = 1.10$) показывает, что мнения студентов варьируются от "удовлетворительно" до "отлично". Медиана (4.0) подтверждает, что большинство студентов оценили задания как "хорошие".

- Сложность курса получила высокую оценку (4.13), что указывает на то, что студенты считают курс достаточно сложным. Медиана (4.0) подтверждает, что большинство студентов оценили сложность как "высокую". Разброс оценок ($std = 0.92$) показывает, что мнения студентов относительно сложности курса варьируются.

- Обратная связь получила высокую оценку (4.20), что указывает на её эффективность. Низкое стандартное отклонение (0.77) свидетельствует о согласованности мнений студентов. Медиана (4.0) подтверждает, что большинство студентов оценили обратную связь как "хорошую".

- Комфорт обучения получил среднюю оценку 3.33, что указывает на умеренный уровень комфорта. Высокое стандартное отклонение (1.29) показывает значительный разброс мнений: от "очень дискомфортно" (1.0) до "очень комфортно" (5.0). Медиана (4.0) подтверждает, что большинство студентов чувствовали себя "комфортно".

- Организация курса получила оценку 3.73, что указывает на её удовлетворительный уровень. Разброс оценок ($std = 1.28$) показывает, что мнения студентов варьируются. Медиана (3.0) подтверждает, что большинство студентов оценили организацию как "удовлетворительную".

- Техническая поддержка получила среднюю оценку 3.53, что указывает на её умеренную эффективность. Высокое стандартное отклонение (1.30) показывает значительный разброс мнений. Медиана (4.0) подтверждает, что большинство студентов оценили поддержку как "хорошую".

- Рекомендация курса получила оценку 3.80, что указывает на умеренную готовность студентов рекомендовать курс другим. Высокое стандартное отклонение (1.37) показывает значительный разброс мнений. Медиана (4.0) подтверждает, что большинство студентов готовы рекомендовать курс.

Положительные аспекты: Студенты высоко оценили **обратную связь** (4.20) и **сложность курса** (4.13), что указывает на их удовлетворённость этими аспектами. **Материалы курса** (3.60) и **практические задания** (3.73) также получили положительные оценки.

Аспекты для улучшения: **Качество преподавания** (3.27) и **комфорт обучения** (3.33) получили более низкие оценки, что указывает на необходимость улучшения этих аспектов. **Техническая поддержка** (3.53) и **организация курса** (3.73) также требуют внимания.

Разброс мнений: Наибольший разброс мнений наблюдается для **качества преподавания**, **комфорта обучения** и **технической поддержки**, что указывает на неоднозначность восприятия этих аспектов.

Рекомендации: Улучшение качества преподавания и комфорта обучения может повысить общую удовлетворённость студентов. Усиление технической поддержки и улучшение организации курса также могут положительно повлиять на восприятие курса.

Результаты описательной статистики показывают, что студенты в целом положительно оценивают курс, но есть аспекты, требующие улучшения. Эти выводы могут быть использованы для разработки стратегий повышения качества учебного процесса.

Подготовим данные для визуализации распределения оценок по каждому вопросу опроса. Код извлекает названия столбцов, соответствующих вопросам, исключая первый столбец (содержащий имена студентов).

```
# Визуализация распределения оценок для каждого вопроса (интерактивно с Plotly)
questions = data.columns[1:] # Пропускаем первый столбец с именами
```

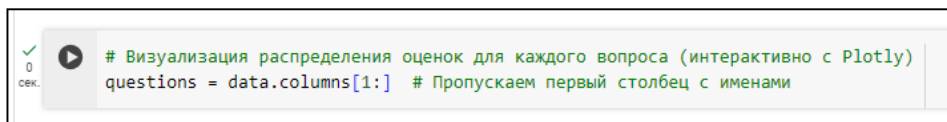


Рисунок 10. Подготовка данных к визуализации

Напишем код, который создаст интерактивные гистограммы для каждого вопроса опроса с использованием библиотеки `plotly`. Сначала создаётся сетка из 5 строк и 2 столбцов (`make_subplots`), в которой каждая ячейка соответствует одному вопросу. Для каждого вопроса вычисляется количество каждой оценки (`value_counts`), и результаты отображаются в виде столбчатой диаграммы (`go.Bar`). Гистограммы добавляются в соответствующую ячейку сетки с помощью `add_trace`, где позиция ячейки определяется по формуле $(i-1) // 2 + 1$ для строки и $(i-1) \% 2 + 1$ для столбца. Затем задаётся общий заголовок графика (`title_text`), высота и ширина фигуры, а также отключается легенда (`showlegend=False`). В конце график отображается с помощью `fig.show()`. Этот код позволяет визуализировать распределение оценок по всем вопросам в одном интерактивном графике, что упрощает анализ данных.

```
# Создаем интерактивные гистограммы для каждого вопроса
fig = make_subplots(rows=5, cols=2, subplot_titles=questions)

for i, question in enumerate(questions, 1):
    counts = data[question].value_counts().sort_index()
    fig.add_trace(
        go.Bar(x=counts.index, y=counts.values, name=question),
        row=(i-1)//2 + 1, col=(i-1)%2 + 1
    )

fig.update_layout(
    title_text="Распределение оценок по вопросам",
    height=1200,
    width=1000,
    showlegend=False
)
fig.show()
```

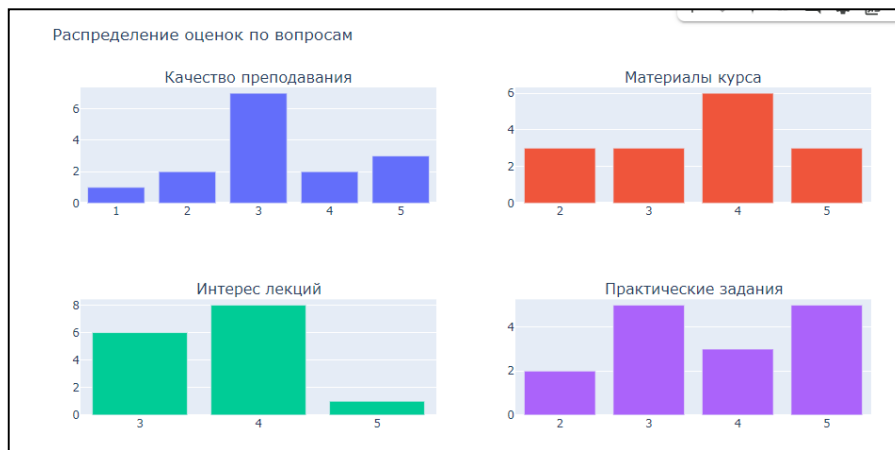


Рисунок 11. Визуализация вопросов 1-4

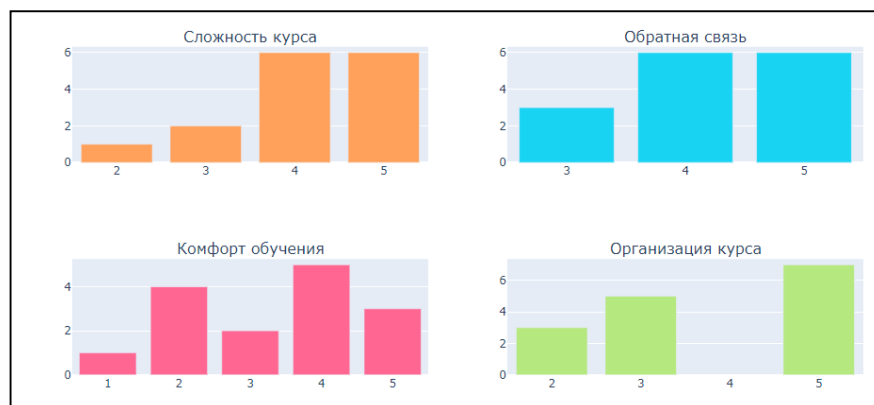


Рисунок 12. Визуализация вопросов 5-8

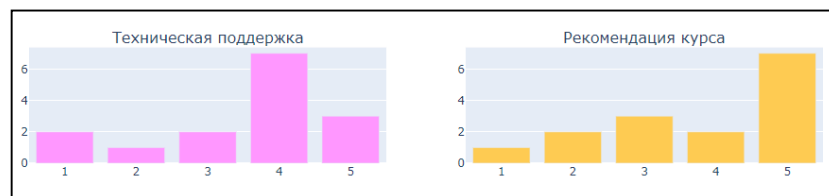


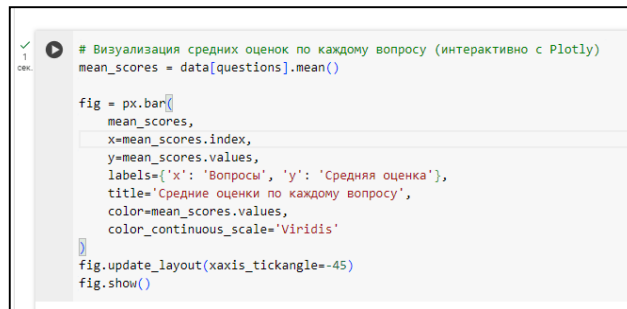
Рисунок 13. Визуализация вопросов 9-10

Далее напишем код, который визуализирует средние оценки по каждому вопросу опроса в виде интерактивной столбчатой диаграммы с использованием библиотеки `plotly.express`. Сначала вычисляются средние значения оценок для каждого вопроса с помощью `data[questions].mean()`, где `questions` — это список столбцов с вопросами. Затем создаётся столбчатая диаграмма (`px.bar`), где по оси `X` откладываются названия вопросов (`mean_scores.index`), а по оси `Y` — соответствующие средние оценки (`mean_scores.values`). Диаграмма настраивается с помощью параметров: заголовков (`title`), подписи осей (`labels`), цветовая шкала (`color` и `color_continuous_scale`), а также угол наклона подписей на оси `X` (`xaxis_tickangle=-45`). В конце график отображается с помощью `fig.show()`. Этот код позволяет наглядно

сравнить средние оценки по всем вопросам и выявить наиболее и наименее успешные аспекты курса.

```
# Визуализация средних оценок по каждому вопросу (интерактивно с Plotly)
mean_scores = data[questions].mean()
```

```
fig = px.bar(
    mean_scores,
    x=mean_scores.index,
    y=mean_scores.values,
    labels={'x': 'Вопросы', 'y': 'Средняя оценка'},
    title='Средние оценки по каждому вопросу',
    color=mean_scores.values,
    color_continuous_scale='Viridis'
)
fig.update_layout(xaxis_tickangle=-45)
fig.show()
```



```
# Визуализация средних оценок по каждому вопросу (интерактивно с Plotly)
mean_scores = data[questions].mean()

fig = px.bar(
    mean_scores,
    x=mean_scores.index,
    y=mean_scores.values,
    labels={'x': 'Вопросы', 'y': 'Средняя оценка'},
    title='Средние оценки по каждому вопросу',
    color=mean_scores.values,
    color_continuous_scale='Viridis'
)

fig.update_layout(xaxis_tickangle=-45)
fig.show()
```

Рисунок 14. Визуализация интерактивной столбчатой диаграммы

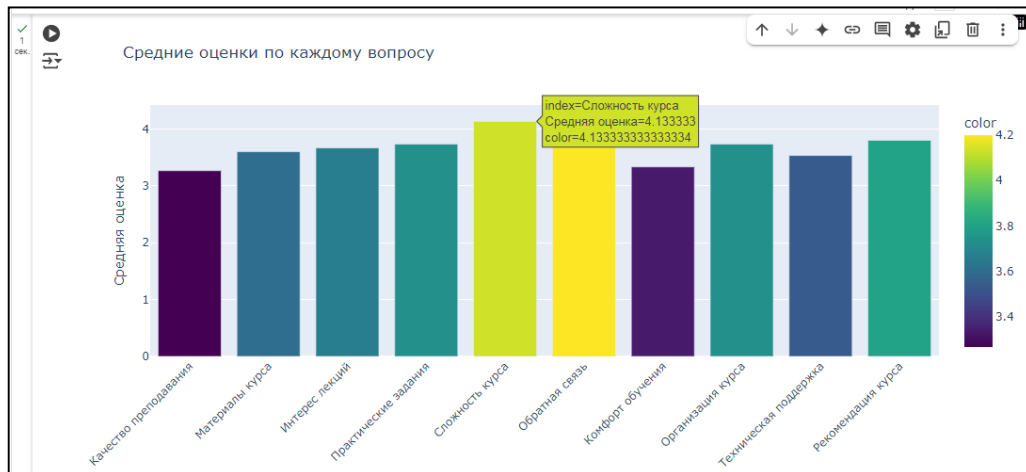


Рисунок 15. Визуализация интерактивной столбчатой диаграммы

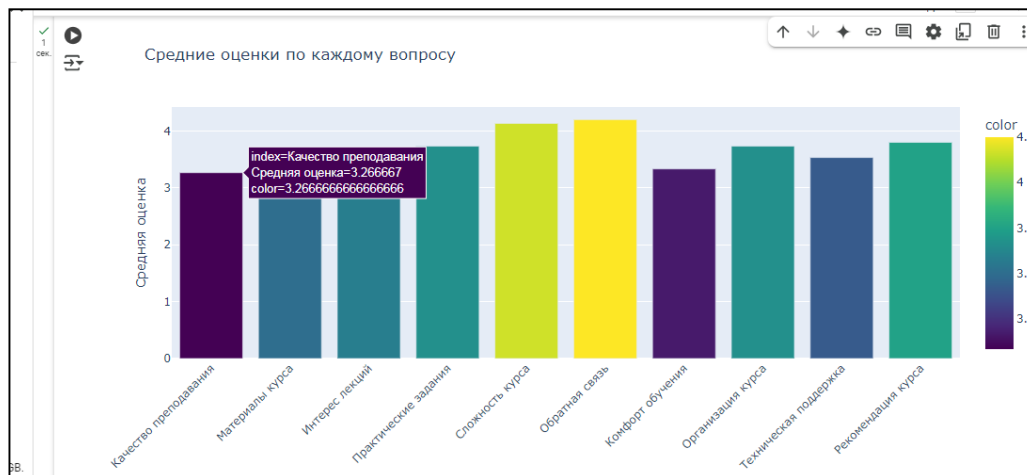


Рисунок 16. Визуализация интерактивной столбчатой диаграммы

Теперь напишем код, который вычисляет корреляционную матрицу для вопросов опроса и визуализирует её в виде интерактивной тепловой карты с использованием библиотеки `plotly`.

Сначала с помощью `data[questions].corr()` вычисляется корреляционная матрица, где каждая ячейка содержит коэффициент корреляции между двумя вопросами. Затем создаётся тепловая карта (`go.Heatmap`), где значения корреляций (z) отображаются в виде цветовых градиентов, а оси X и Y соответствуют названиям вопросов (`corr_matrix.columns`). Цветовая шкала (`colorscale='RdBu'`) задаёт синий цвет для положительных корреляций и красный для отрицательных, а параметры `zmin=-1` и `zmax=1` ограничивают диапазон значений корреляции. На тепловую карту добавляются числовые значения корреляций, округлённые до двух знаков после запятой (`text=corr_matrix.round(2).values`), которые отображаются как при наведении курсора (`hoverinfo="text"`), так и непосредственно на карте (`texttemplate="%{text}"`). График настраивается с помощью `update_layout`, где задаются заголовок (`title`), количество меток на осях (`xaxis_nticks=36`, `yaxis_nticks=36`) и подписи осей (`xaxis_title`, `yaxis_title`). В конце тепловая карта отображается с помощью `fig.show()`. Этот код позволяет наглядно анализировать взаимосвязи между вопросами опроса, выявляя сильные и слабые корреляции.

```
# Вычисление корреляционной матрицы
corr_matrix = data[questions].corr()

# Создание интерактивной тепловой карты
fig = go.Figure(data=go.Heatmap(
    z=corr_matrix,
    x=corr_matrix.columns,
    y=corr_matrix.columns,
    colorscale='RdBu',
    zmin=-1,
```

```

zmax=1,
text=corr_matrix.round(2).values, # Числовые значения
hoverinfo="text", # Показывать значения при наведении
texttemplate="%{text}" # Отображать значения на карте
))

fig.update_layout(
    title='Корреляция между вопросами',
    xaxis_nticks=36,
    yaxis_nticks=36,
    xaxis_title="Вопросы",
    yaxis_title="Вопросы"
)

fig.show()

```

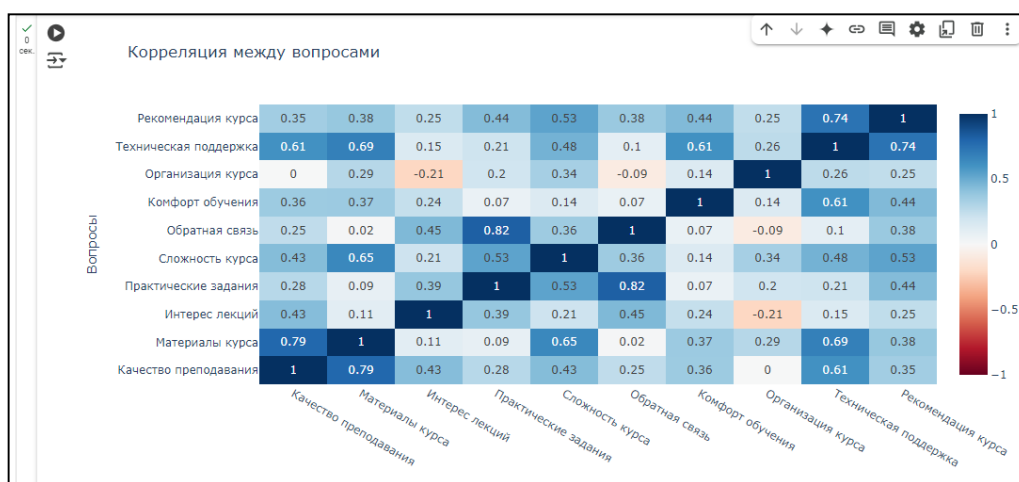


Рисунок 17. Корреляционная матрица

Результаты корреляционного анализа показали, что наиболее сильные положительные корреляции наблюдаются между **"Обратной связью"** и **"Практическими знаниями"** (0.82), а также между **"Материалами курса"** и **"Качеством преподавания"** (0.79). Это указывает на то, что студенты, которые высоко оценивают обратную связь от преподавателей, также склонны положительно оценивать практическую полезность курса. Аналогично, высокое качество преподавания тесно связано с удовлетворённостью материалами курса, что подчеркивает важность профессионального подхода преподавателей к подготовке и подаче учебного контента.

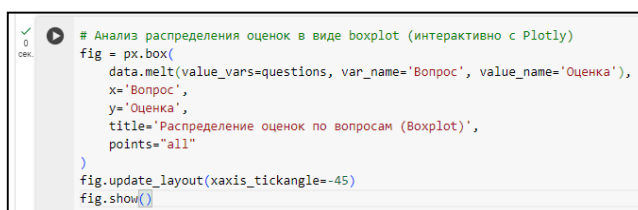
Напротив, выявлена **отрицательная корреляция** между **"Организацией курса"** и **"Интересом лекций"** (-0.21). Это может свидетельствовать о том, что недостатки в организации курса (например, неудобное расписание или плохая структура) снижают интерес студентов к лекциям. Хотя корреляция слабая, это сигнал о необходимости улучшения организационных аспектов для повышения вовлеченности студентов.

Слабая корреляция между **"Технической поддержкой"** и другими аспектами курса (0.1–0.25) говорит о том, что техническая поддержка воспринимается студентами как отдельный элемент, который практически не

влияет на их общую удовлетворённость курсом. Это может указывать на то, что техническая поддержка либо уже находится на приемлемом уровне, либо не является критически важной для студентов в контексте других аспектов обучения.

Напишем код, который создаёт интерактивный `boxplot` (ящик с усами) для визуализации распределения оценок по каждому вопросу опроса с использованием библиотеки `plotly.express`. Сначала данные преобразуются с помощью метода `melt`, который объединяет все столбцы с вопросами в два столбца: `Вопрос` (название вопроса) и `Оценка` (значение оценки). Это позволяет удобно визуализировать данные в формате "длинной" таблицы. Затем создаётся `boxplot` (`px.box`), где по оси `X` откладываются названия вопросов (`x='Вопрос'`), а по оси `Y` — соответствующие оценки (`y='Оценка'`). Параметр `points="all"` добавляет все точки данных на график, что позволяет увидеть распределение оценок более детально. График настраивается с помощью `update_layout`, где задаётся угол наклона подписей на оси `X` (`xaxis_tickangle=-45`) для улучшения читаемости. В конце `boxplot` отображается с помощью `fig.show()`. Этот код позволяет наглядно анализировать распределение оценок, выявлять медиану, квартили, выбросы и общие тенденции по каждому вопросу, что помогает оценить согласованность мнений студентов и выявить потенциальные проблемы.

```
# Анализ распределения оценок в виде boxplot (интерактивно с Plotly)
fig = px.box(
    data.melt(value_vars=questions, var_name='Вопрос', value_name='Оценка'),
    x='Вопрос',
    y='Оценка',
    title='Распределение оценок по вопросам (Boxplot)',
    points="all"
)
fig.update_layout(xaxis_tickangle=-45)
fig.show()
```

A screenshot of a code editor window. The code is in Python and uses Plotly Express to create an interactive boxplot. The code includes comments in Russian and uses Cyrillic characters for variable names. The code is as follows:

```
✓ 0 сек. # Анализ распределения оценок в виде boxplot (интерактивно с Plotly)
fig = px.box(
    data.melt(value_vars=questions, var_name='Вопрос', value_name='Оценка'),
    x='Вопрос',
    y='Оценка',
    title='Распределение оценок по вопросам (Boxplot)',
    points="all"
)
fig.update_layout(xaxis_tickangle=-45)
fig.show()
```

Рисунок 18. Создание интерактивного `boxplot`

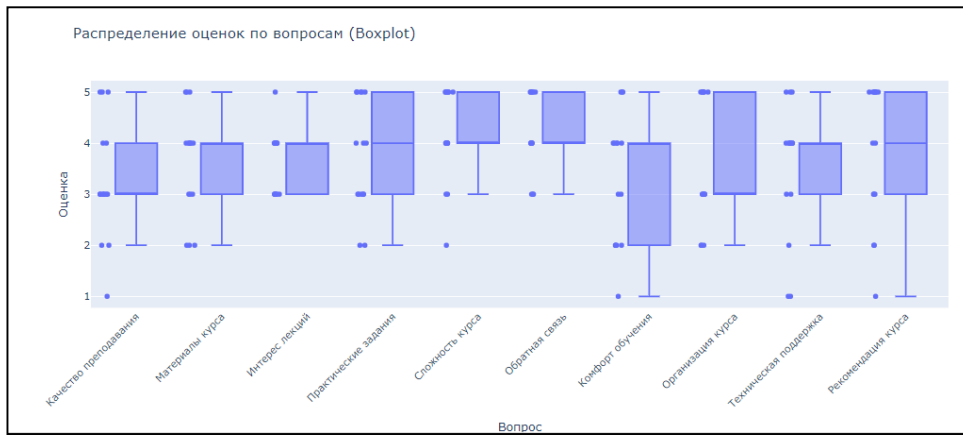


Рисунок 19. Интерактивный boxplot (ящик с усами)

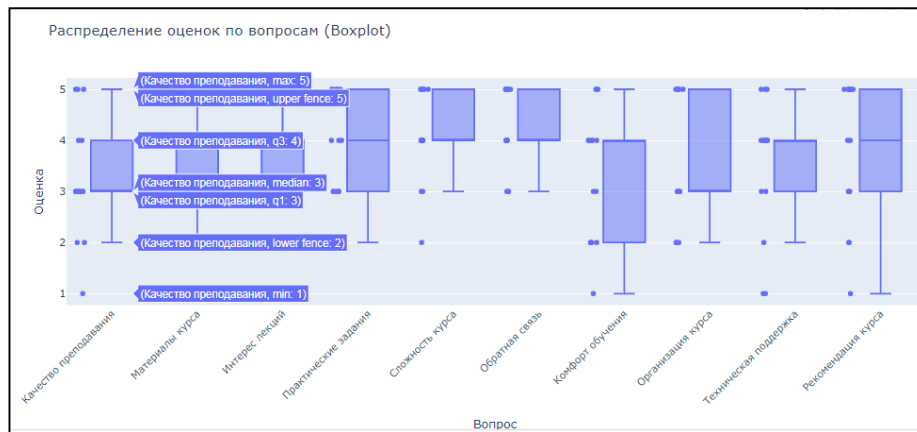


Рисунок 20. Интерактивный boxplot (ящик с усами)

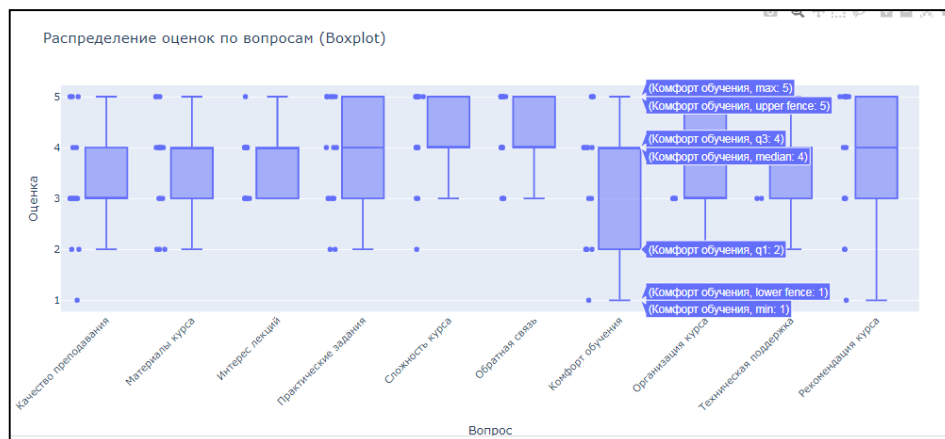


Рисунок 21. Интерактивный boxplot (ящик с усами)

Напишем код, который анализирует общий уровень удовлетворённости студентов, вычисляя сумму баллов по всем вопросам опроса для каждого студента и визуализируя распределение этих сумм с помощью интерактивной гистограммы, созданной с использованием библиотеки `plotly.express`. Сначала в `DataFrame data` добавляется новый столбец `Общий балл`, который содержит сумму оценок по всем вопросам для каждого студента (`data[questions].sum(axis=1)`). Затем строится гистограмма (`px.histogram`), где по оси `X` откладываются значения общего балла

(`x='Общий балл'`), а по оси `Y` — частота встречаемости этих значений. Параметр `nbins=20` задаёт количество интервалов (бинов) для группировки данных, а `title` добавляет заголовок графика. Подписи осей настраиваются с помощью `labels`, где ось `X` обозначается как "Сумма баллов". Цвет гистограммы задаётся с помощью `color_discrete_sequence=['#1f77b4']`, что делает график визуально привлекательным. В конце гистограмма отображается с помощью `fig.show()`. Этот код позволяет наглядно оценить распределение общего уровня удовлетворённости студентов, выявить наиболее частые значения и определить, насколько равномерно распределены оценки.

```
# Анализ общего уровня удовлетворенности (сумма баллов по всем вопросам)
data['Общий балл'] = data[questions].sum(axis=1)

fig = px.histogram(
    data,
    x='Общий балл',
    nbins=20,
    title='Распределение общего уровня удовлетворенности',
    labels={'Общий балл': 'Сумма баллов'},
    color_discrete_sequence=['#1f77b4']
)
fig.show()
```

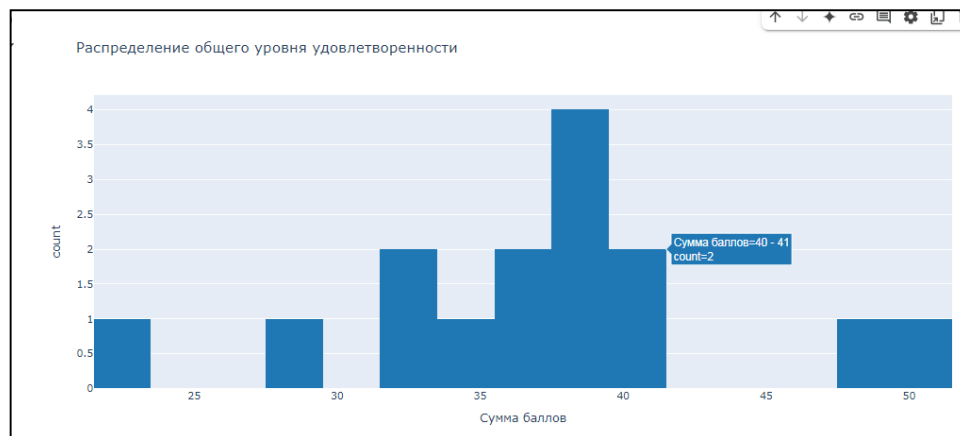


Рисунок 22. Интерактивная гистограмма

Теперь напишем код, который анализирует ответы студентов на вопрос о рекомендации курса (вопрос 10) и визуализирует их в виде интерактивной круговой диаграммы (`pie chart`) с использованием библиотеки `plotly.express`. Сначала с помощью `data['Рекомендация курса'].value_counts().sort_index()` подсчитывается количество студентов, выбравших каждую из возможных оценок, и результаты сортируются по возрастанию. Затем создаётся круговая диаграмма (`px.pie`), где `names` задаёт категории (оценки), а `values` —

соответствующие им количества студентов. Параметр `title` добавляет заголовок графика, а `labels` настраивает подписи для категорий и значений. В конце диаграмма отображается с помощью `fig.show()`. Этот код позволяет наглядно оценить, какой процент студентов готов рекомендовать курс, и выявить распределение мнений, что помогает понять общий уровень удовлетворённости и лояльности студентов.

```
# Анализ рекомендаций (вопрос 10)
recommendation_counts = data['Рекомендация курса'].value_counts().sort_index()

fig = px.pie(
    names=recommendation_counts.index,
    values=recommendation_counts.values,
    title='Рекомендации курса (Вопрос 10)',
    labels={'names': 'Оценка', 'values': 'Количество студентов'}
)
fig.show()
```

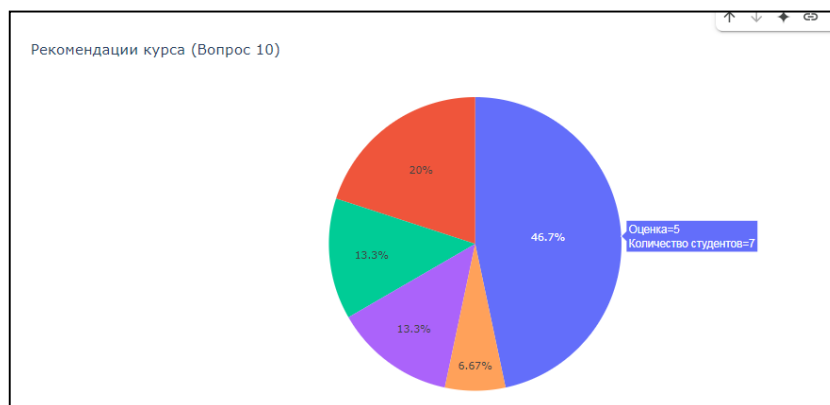


Рисунок 23. Интерактивная круговая диаграмма

4 Выводы

В данной научной статье был рассмотрен пошаговый анализ данных опроса студентов с использованием библиотек Python. Рассмотрено, как загружать данные, вычислять описательную статистику, строить интерактивные графики (гистограммы, `boxplot`, тепловые карты и круговые диаграммы) и интерпретировать результаты с использованием Google Colaboratory. Для достижения поставленной цели использовался набор данных, содержащий результаты опросов студентов.

Данная работа может использоваться в образовательных учреждениях для проведения опросов среди студентов, а также в научных исследованиях для анализа общественного мнения. Кроме того, работа демонстрирует возможности современных веб-технологий для создания интерактивных и наглядных инструментов анализа данных.

Библиографический список

1. Насыпова Э. А., Развеева И. Ф. Применение методов машинного обучения для формирования портрета клиента психологического центра //XXIII Всероссийская студенческая научно-практическая конференция Нижневартовского государственного университета. 2021. С. 41-45.
2. Матвеева А. С. Разведочный анализ данных о прогнозировании энергопотребления //Постулат. 2024. №. 1 январь.
3. Антонов А. А. Прогнозирование размера заработной платы с использованием методов машинного обучения по обработке естественного языка //лучшие исследовательские работы студентов 3. 2024. С. 30.
4. Войскунский А. Е., Скрипкин С. В. Качественный анализ данных //Вестник Московского университета. Серия. 2001. Т. 14. С. 93-109.
5. Григорьев Е. А., Климов Н. С. Разведочный анализ данных с помощью python //E-Scio. 2020. №. 2 (41). С. 165-176.