Тема: Data Science.

Исследовательский анализ данных о том, как ученики онлайн-школы «Сириус.Курсы» учились на нескольких курсах.

Что такое Data Science?

Наука о данных (англ. data science) — раздел информатики, изучающий проблемы анализа, обработки и представления данных в цифровой форме.

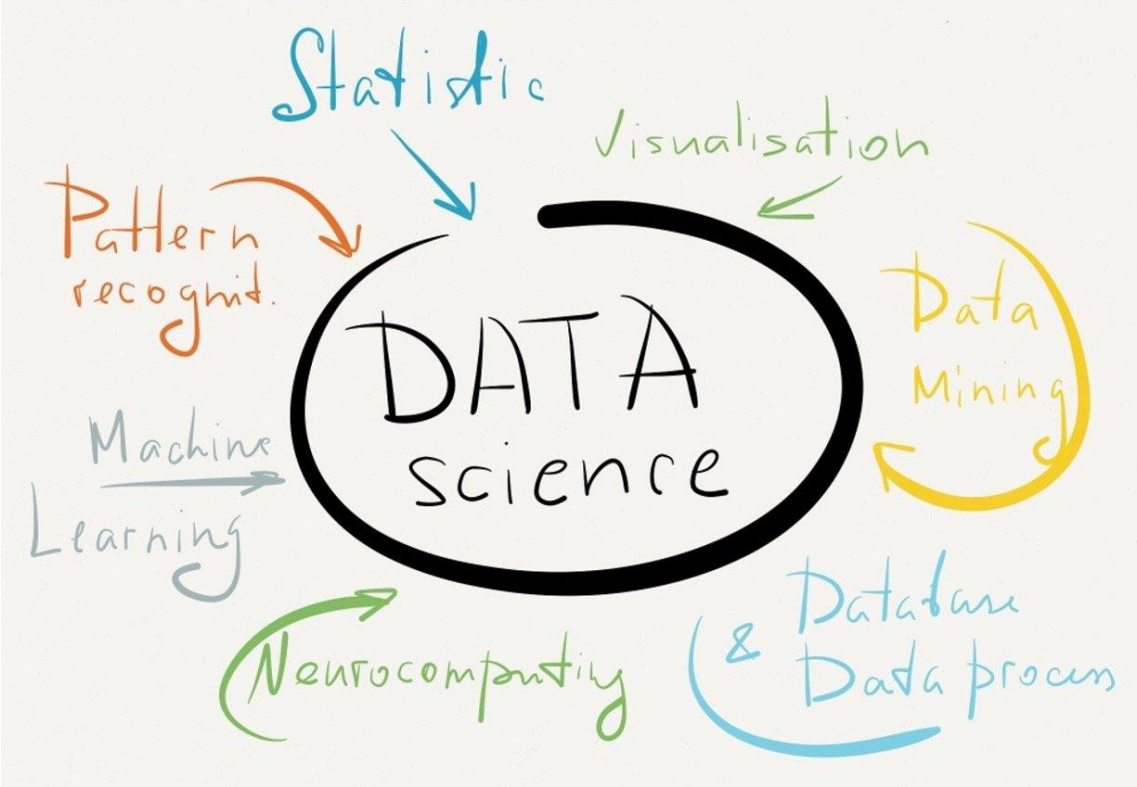
В рамках науки о данных рассматриваются:

- методы обработки данных в условиях больших объёмов и высокого уровня параллелизма;

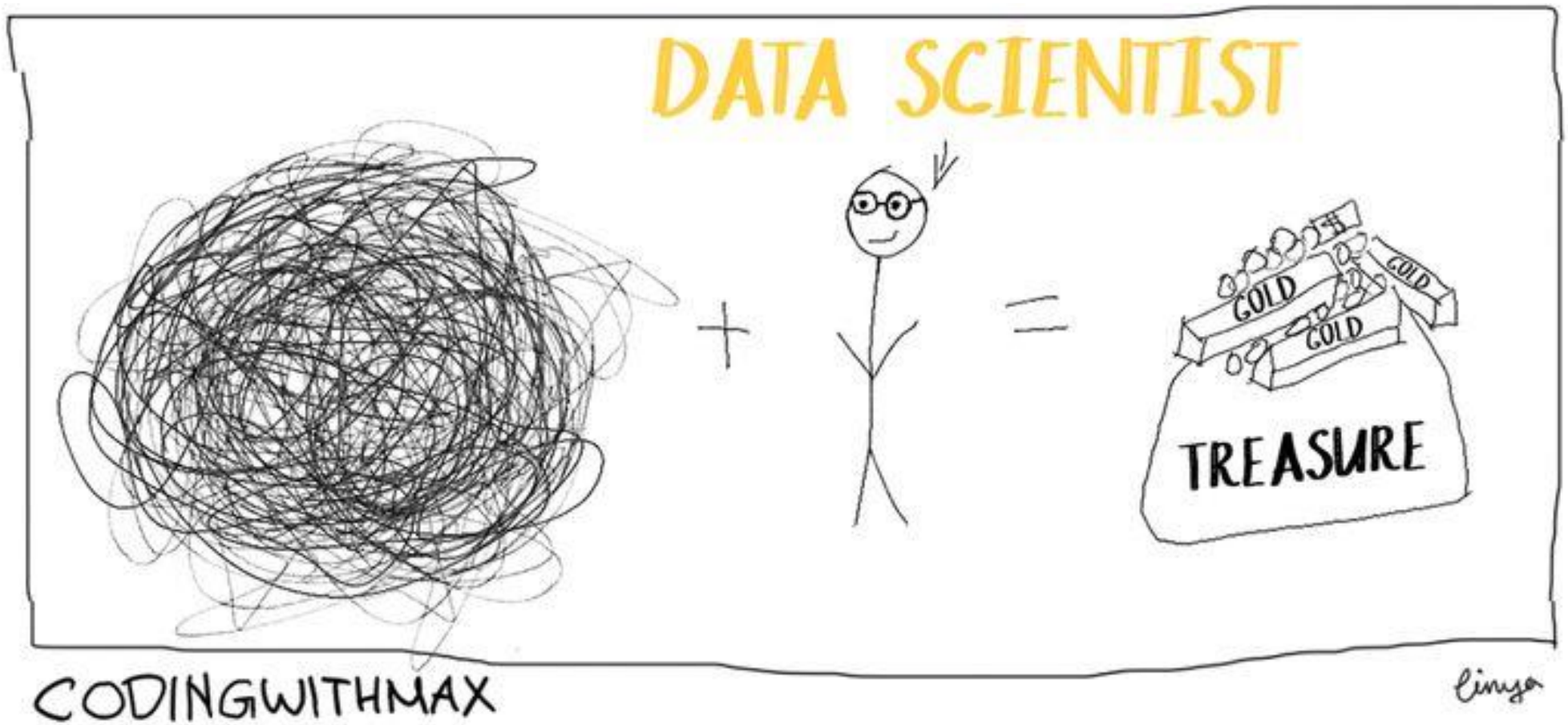
- статистические методы;

- методы интеллектуального анализа данных и приложения искусственного интеллекта для работы с данными;

- методы проектирования и разработки баз данных.



Чем занимается Data Scientist?



Что такое Pandas?

- — это **библиотека Python для обработки и анализа структурированных данных**, её название происходит от «panel data» («панельные данные»). Панельными данными называют информацию, полученную в результате исследований и структурированную в виде таблиц. Для работы с такими массивами данных и создан Pandas.

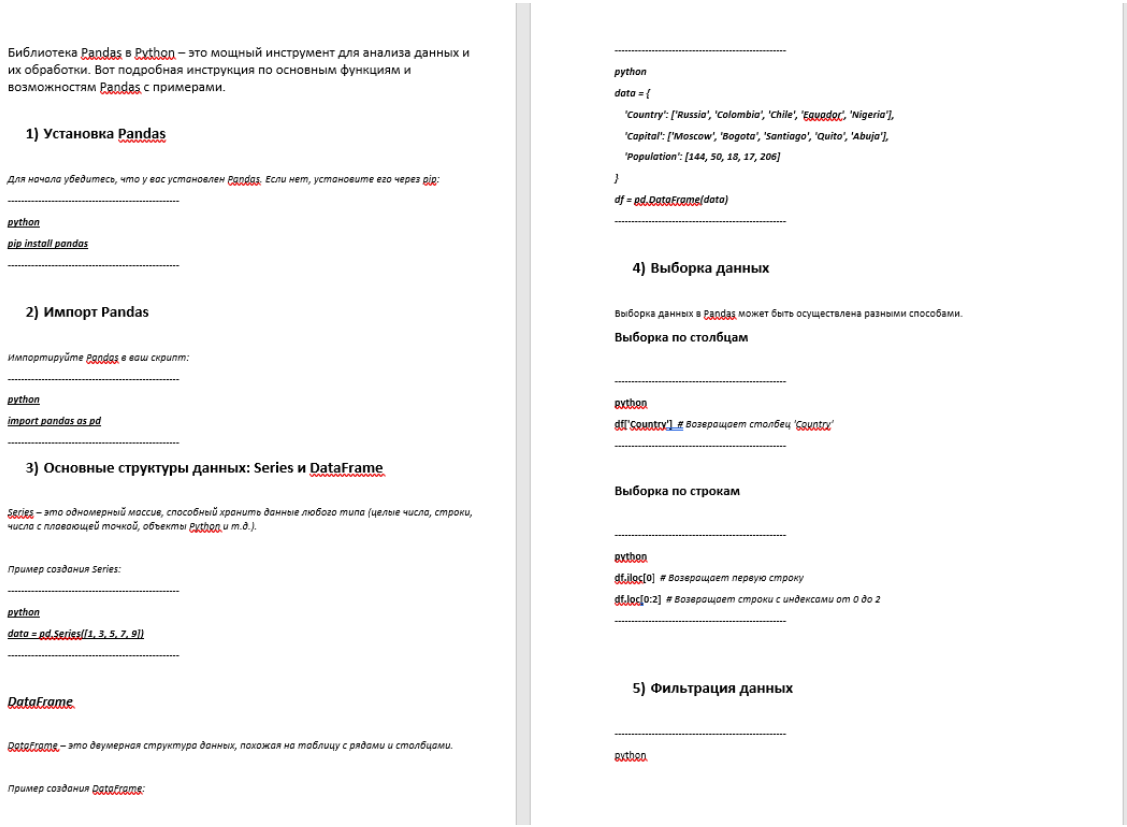
Для чего используется Pandas?

Pandas может использоваться во всех процессах анализа данных. С помощью этой библиотеки можно:

- Импортировать наборы данных из баз данных, электронных таблиц, CSV-файлов и т.д.
- Очищать наборы данных, например, устраняя пропущенные значения.
- Упорядочивать наборы данных путем преобразования их структуры в формат, пригодный для анализа.
- Агрегировать данные, вычисляя сводную статистику, например, среднее значение столбцов, корреляцию между ними и т.д.
- Визуализировать наборы данных и открывать новые возможности.
- Pandas также имеет функционал для анализа временных рядов и текстовых данных.

Создание инструкции по Pandas

- Были изучены и подробно описаны в отдельном текстовом документе многие функции библиотеки Pandas



Библиотека Pandas в Python – это мощный инструмент для анализа данных и их обработки. Вот подробная инструкция по основным функциям и возможностям Pandas с примерами.

1) Установка Pandas

Для начала убедитесь, что у вас установлен Pandas. Если нет, установите его через pip:

---

python

pip install pandas

---

2) Импорт Pandas

Импортируйте Pandas в ваш скрипт:

---

python

import pandas as pd

---

3) Основные структуры данных: Series и DataFrame

Series – это одномерный массив, способный хранить данные любого типа (целые числа, строки, числа с плавающей точкой, объекты Python и т.д.).

Пример создания Series:

---

python

data = pd.Series([1, 3, 5, 7, 9])

DataFrame

DataFrame – это двумерная структура данных, похожая на таблицу с рядами и столбцами.

Пример создания DataFrame:

---

python

data = {

'Country': ['Russia', 'Colombia', 'Chile', 'Ecuador', 'Nigeria'],

'Capital': ['Moscow', 'Bogota', 'Santiago', 'Quito', 'Abuja'],

'Population': [144, 50, 18, 17, 206]

}

df = pd.DataFrame(data)

---

4) Выборка данных

Выборка данных в Pandas может быть осуществлена разными способами.

Выборка по столбцам

---

python

df['Country'] *# Возвращает столбец 'Country'*

Выборка по строкам

---

python

df.iloc[0] *# Возвращает первую строку*

df.loc[0:2] *# Возвращает строки с индексами от 0 до 2*

---

5) Фильтрация данных

---

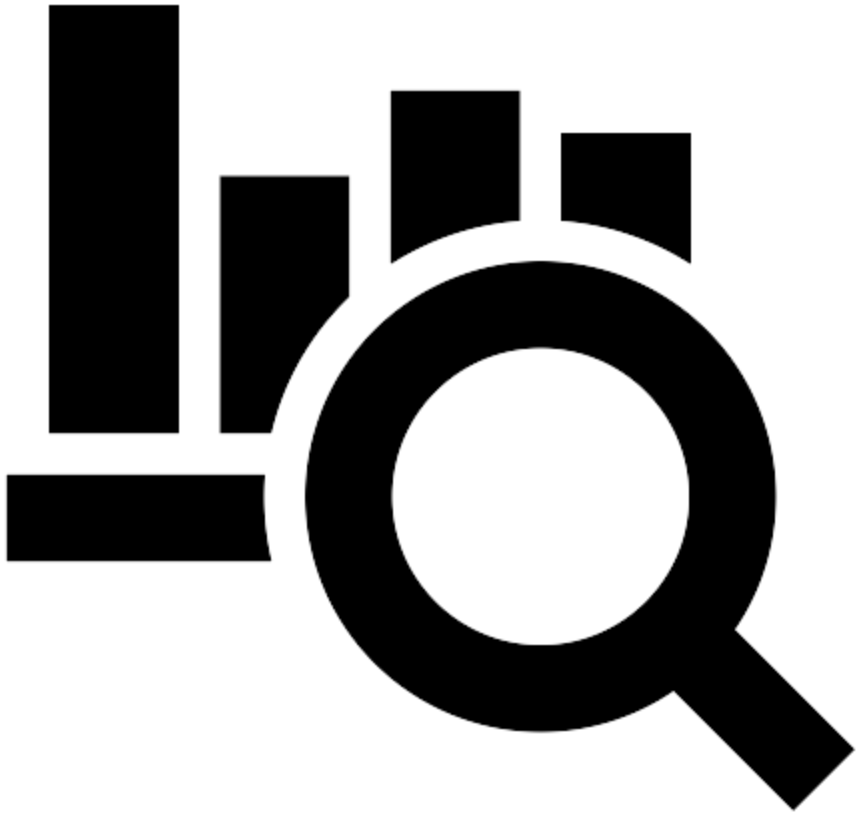
python

Основная цель - найти признаки по которым можно определить будет ли продолжать обучение ученик

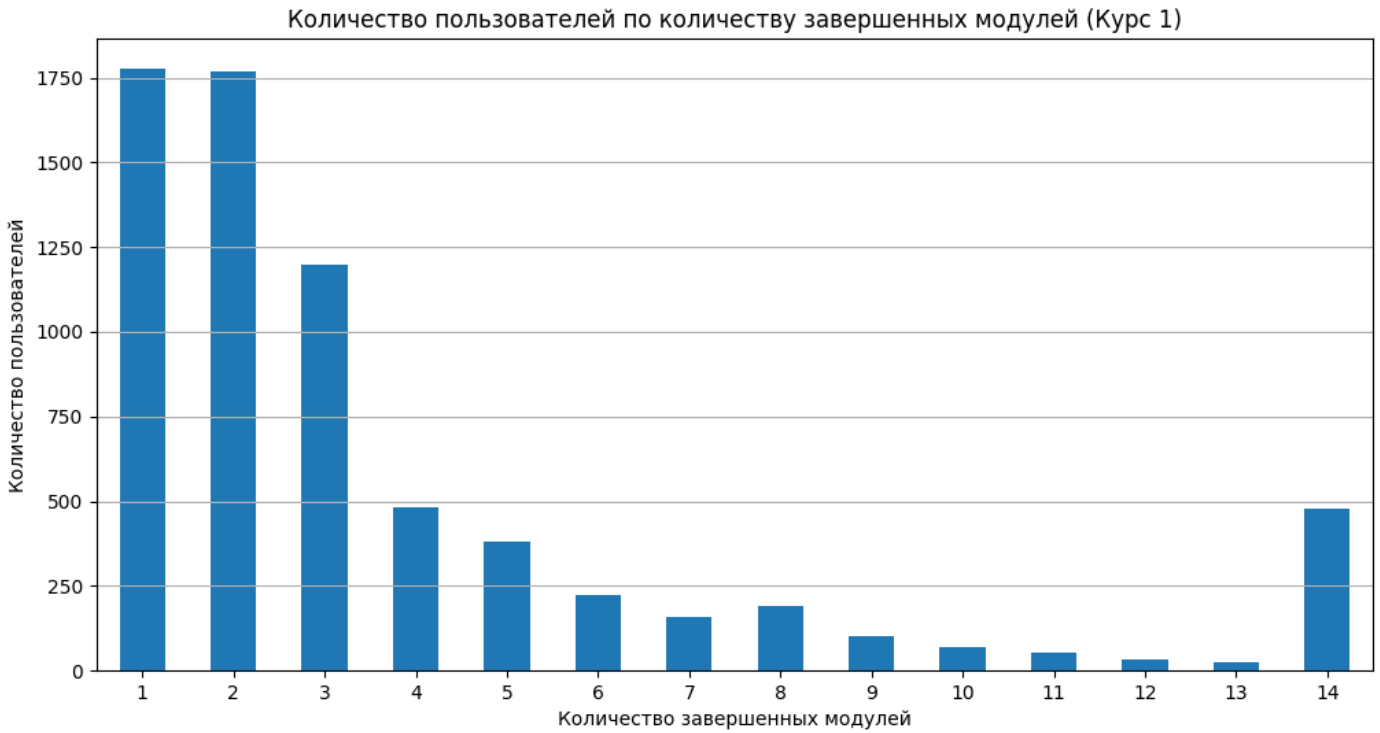
Выбор метрик.

1) Число завершенных модулей
2) Среднее количество попыток
3) Соотношение успешных и неудачных попыток
4) Регулярность занятий
5) Временные интервалы между входами в систему
6) Среднее время на выполнение модуля

Анализ полученных данных.

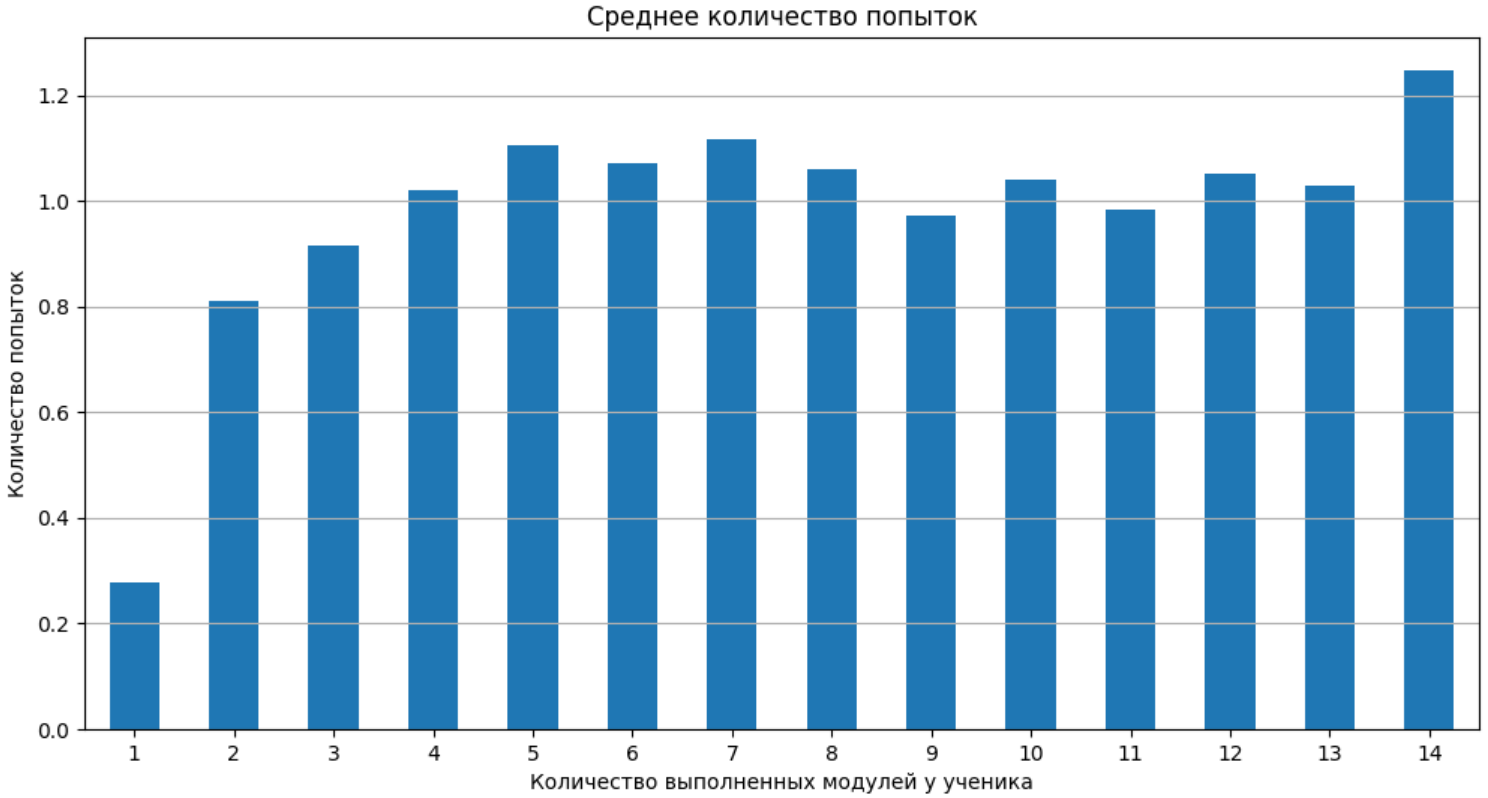


Число завершенных модулей



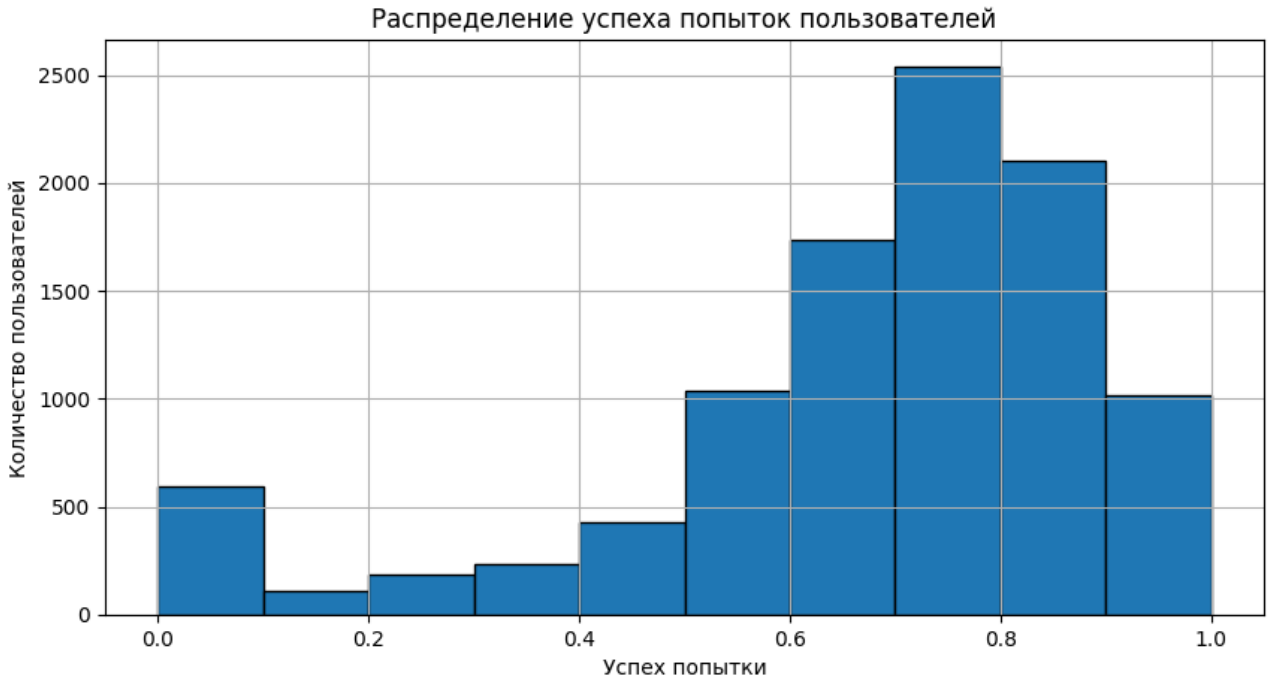

```python
#первый про модули
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('user_module_progress.csv')

course_1_data = data[data['course_id'] == 1]

completed_modules_course_1 = course_1_data[course_1_data['is_achieved'] == True]

user_completed_modules = completed_modules_course_1.groupby('user_id').size()

user_count_by_completed_modules = user_completed_modules.value_counts().sort_index()

# Построение графика
plt.figure(figsize=(12, 6))
user_count_by_completed_modules.plot(kind='bar')
plt.title('Количество пользователей по количеству завершенных модулей (Курс 1)')
plt.xlabel('Количество завершенных модулей')
plt.ylabel('Количество пользователей')
plt.xticks(rotation=0)
plt.grid(axis='y')
plt.show()

```

Среднее количество попыток



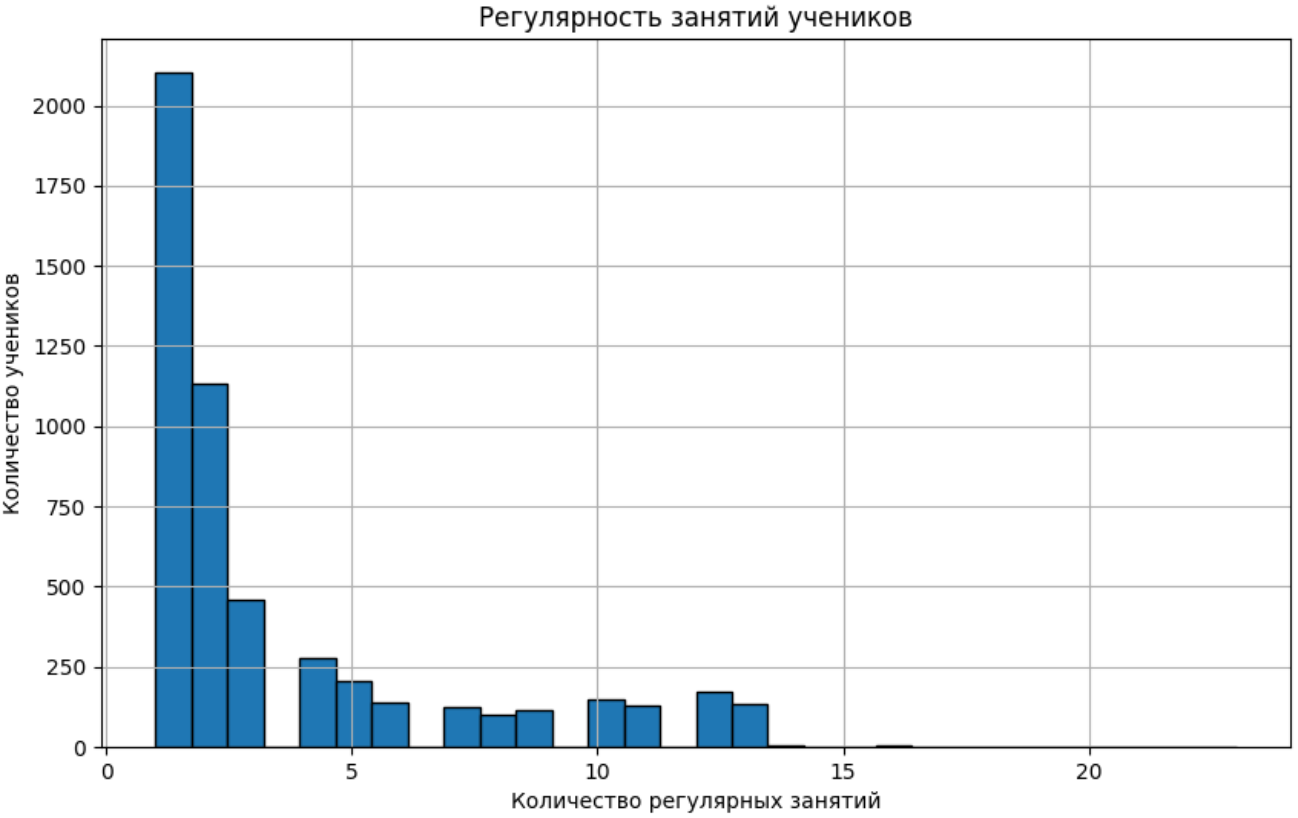

```python
#про кол-во попыток
import pandas as pd
import matplotlib.pyplot as plt

user_element_progress = pd.read_csv('user_element_progress.csv')

course_1_progress = user_element_progress[user_element_progress['course_id'] == 1]

modules_completed_course_1 = course_1_progress[course_1_progress['is_achieved'] == True] \
                             .groupby('user_id')['course_module_id'] \
                             .nunique()

avg_tries_per_user_course_1 = course_1_progress.groupby('user_id')['tries_count'].mean()

user_stats_course_1 = pd.DataFrame({'modules_completed': modules_completed_course_1, 'avg_tries': avg_tries_per_user_course_1})

avg_tries_by_module_count_course_1 = user_stats_course_1.groupby('modules_completed')['avg_tries'].mean()

#график
plt.figure(figsize=(12, 6))
avg_tries_by_module_count_course_1.plot(kind='bar')
plt.title('Среднее количество попыток')
plt.xlabel('Количество выполненных модулей у ученика')
plt.ylabel('Количество попыток')
plt.xticks(rotation=0)
plt.grid(axis='y')
plt.show()
```

Соотношение успешных и неудачных попыток




```python
import pandas as pd
import matplotlib.pyplot as plt

element_solution = pd.read_csv('user_element_solution.csv')
element_progress = pd.read_csv('user_element_progress.csv')

# Оценка 'verdict' как 1 для 'ok' и 0 для других значений
element_solution['verdict'] = element_solution['verdict'].apply(lambda x: 1 if x == 'ok' else 0)

solution_stats = element_solution.groupby('element_progress_id')['verdict'].mean().reset_index()

progress_with_solutions = pd.merge(element_progress, solution_stats, how='left', left_on='id', right_on='element_progress_id')

user_success_rate = progress_with_solutions.groupby('user_id')['verdict'].mean()

plt.figure(figsize=(10, 5))
user_success_rate.hist(bins=10, edgecolor='black')
plt.title('Распределение успеха попыток пользователей')
plt.xlabel('Успех попытки')
plt.ylabel('Количество пользователей')
plt.show()
```

Регулярность занятий




```python
import pandas as pd
import matplotlib.pyplot as plt

user_module_progress = pd.read_csv('user_module_progress.csv')

user_module_progress['time_achieved'] = pd.to_datetime(user_module_progress['time_achieved'], errors='coerce')

completed_modules = user_module_progress[user_module_progress['is_achieved'] == True].copy()

completed_modules.sort_values(by=['user_id', 'time_achieved'], inplace=True)
completed_modules['time_diff'] = completed_modules.groupby('user_id')['time_achieved'].diff()

completed_modules['days_between_achievements'] = completed_modules['time_diff'].dt.days

regular_achievements = completed_modules[completed_modules['days_between_achievements'] <= 7]

regularity_counts = regular_achievements.groupby('user_id').size()

plt.figure(figsize=(10, 6))
regularity_counts.hist(bins=30, edgecolor='black')
plt.title('Регулярность занятий учеников')
plt.xlabel('Количество регулярных занятий')
plt.ylabel('Количество учеников')
plt.show()
```

Временные интервалы между входами в систему


```python
import pandas as pd
import matplotlib.pyplot as plt

user_module_progress = pd.read_csv('user_module_progress.csv')

user_module_progress['time_unlocked'] = pd.to_datetime(user_module_progress['time_unlocked'], errors='coerce')

sorted_progress = user_module_progress.sort_values(by=['user_id', 'time_unlocked'])
sorted_progress['time_diff'] = sorted_progress.groupby('user_id')['time_unlocked'].diff()
sorted_progress['hours_between_sessions'] = sorted_progress['time_diff'].dt.total_seconds() / 3600

interval_data = sorted_progress[sorted_progress['hours_between_sessions'].notna()]
plt.figure(figsize=(10, 6))
interval_data['hours_between_sessions'].hist(bins=50, range=(0, 50), edgecolor='black')
plt.title('Интервалы времени между входами в систему')
plt.xlabel('Часы между сессиями')
plt.ylabel('Количество сессий')
plt.xticks(range(0, 51, 5))  # Установка меток с шагом в 5 часов
plt.grid(True)
plt.show()
```

Среднее время на выполнение модуля


```python
import pandas as pd
import matplotlib.pyplot as plt

# Загрузка данных
user_module_progress = pd.read_csv('user_module_progress.csv')

# Преобразование дат
user_module_progress['time_unlocked'] = pd.to_datetime(user_module_progress['time_unlocked'], errors='coerce')
user_module_progress['time_achieved'] = pd.to_datetime(user_module_progress['time_achieved'], errors='coerce')

# Расчет времени на достижение модуля
user_module_progress['time_to_achieve'] = (user_module_progress['time_achieved'] - user_module_progress['time_unlocked']).dt.total_seconds() / 3600  # в часах
completed_modules_time = user_module_progress[user_module_progress['is_achieved'] == True]
average_time_per_module = completed_modules_time.groupby('user_id')['time_to_achieve'].mean()

# Визуализация с уточненными данными
plt.figure(figsize=(10, 6))
upper_limit = average_time_per_module.quantile(0.75)
num_bins = 20
average_time_per_module.hist(bins=num_bins, range=(0, upper_limit), edgecolor='black')
plt.title('Среднее время на достижение модуля')
plt.xlabel('Время (часы)')
plt.ylabel('Количество пользователей')
plt.xticks(range(0, int(upper_limit) + 1, int(upper_limit / num_bins)))
plt.grid(True)
plt.show()
```