Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Пермский национальный исследовательский политехнический университет»

Кафедра информационных технологий и автоматизированных систем

**ОТЧЁТ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ № 4**

|  |
| --- |
| по дисциплине «Технологии анализа данных DataMining»  Тема: «Использование библиотеки pandas» |
|  |

|  |  |
| --- | --- |
|  | Выполнил студент гр. АСУ8-24-1м  Пельц Данил Андреевич  (Фамилия И.О.)  (номер зачетной книжки)  Проверил доцент каф. ИТАС  (должность)  Городилов Алексей Юрьевич  (Фамилия И.О)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата, подпись) |

Пермь 2025 г.

Цель работы: научиться применять библиотеки pandas, numpy, scipy и matplotlib для подготовки, анализа и визуализации данных.

Задачи:

1. Ознакомиться с основными функциями и средствами обработки данных в библиотеке pandas.
2. Изучить методы численных вычислений и работы с массивами в numpy.
3. Применить статистические и математические функции библиотеки scipy для анализа данных.
4. Построить графики и визуализировать результаты анализа с помощью matplotlib.

**Решение задания 1:**

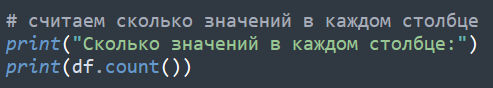
Для начала импортируем библиотеки для работы с данными и визуализацией.



После чего загружаем данные из CSV-файла athlete\_events.csv в DataFrame df.



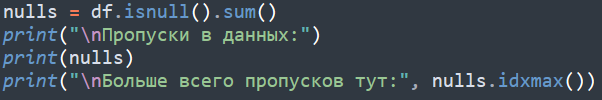
Считаем, сколько значений в каждом столбце, чтобы понять полноту данных.



Выводим общую информацию о DataFrame, включая типы данных и количество значений в столбцах.



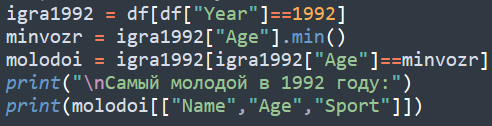
Далее ищем пропущенные значения в столбцах и определяем столбец с наибольшим числом пропусков.



Рассчитываю базовую статистику по числовым столбцам Age, Height и Weight.



Находим самого молодого участника Олимпиады 1992 года и выводим его имя, возраст и вид спорта.



После чего выводим список всех видов спорта, представленных в датасете.



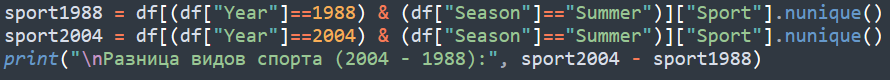
Вычисляем средний рост женщин-теннисисток на Олимпиаде 2000 года.



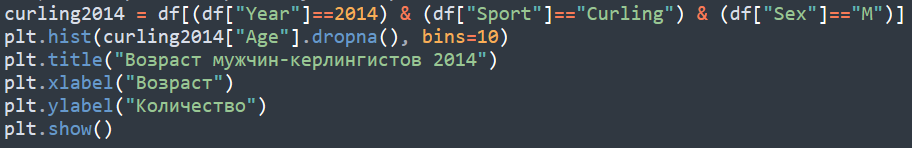
Считаем количество золотых медалей, завоеванных сборной Китая в настольном теннисе на Олимпиаде 2008 года.



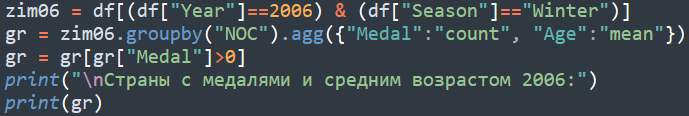
Сравниваем число уникальных видов спорта на летних Олимпиадах 1988 и 2004 годов и выводим разницу.



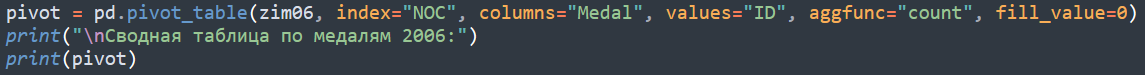
Для визуального представления строим гистограмму распределения возраста мужчин-керлингистов на Олимпиаде 2014 года.



Анализируем зимние Олимпийские игры 2006 года: считаем количество медалей и средний возраст спортсменов по странам.

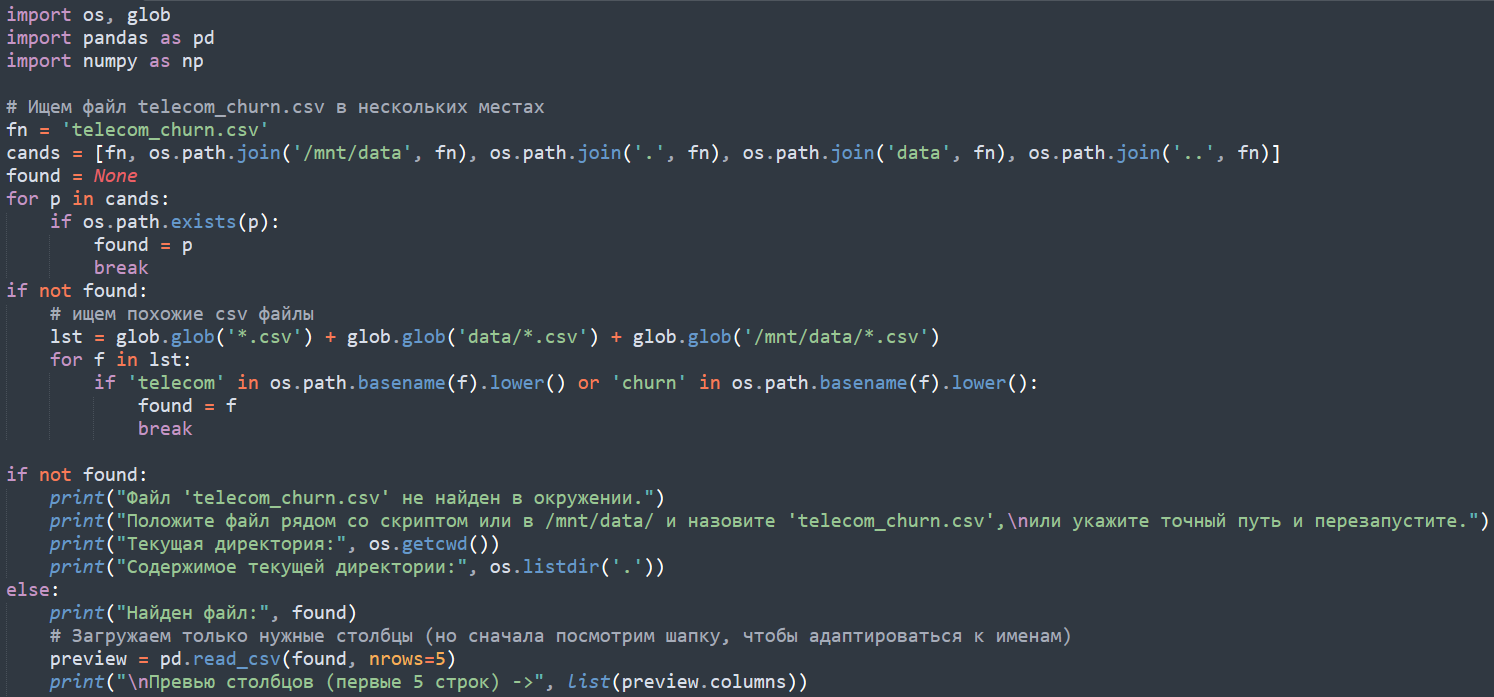


В конце строим сводную таблицу с количеством медалей разных типов для каждой страны на зимних Олимпийских играх 2006 года.

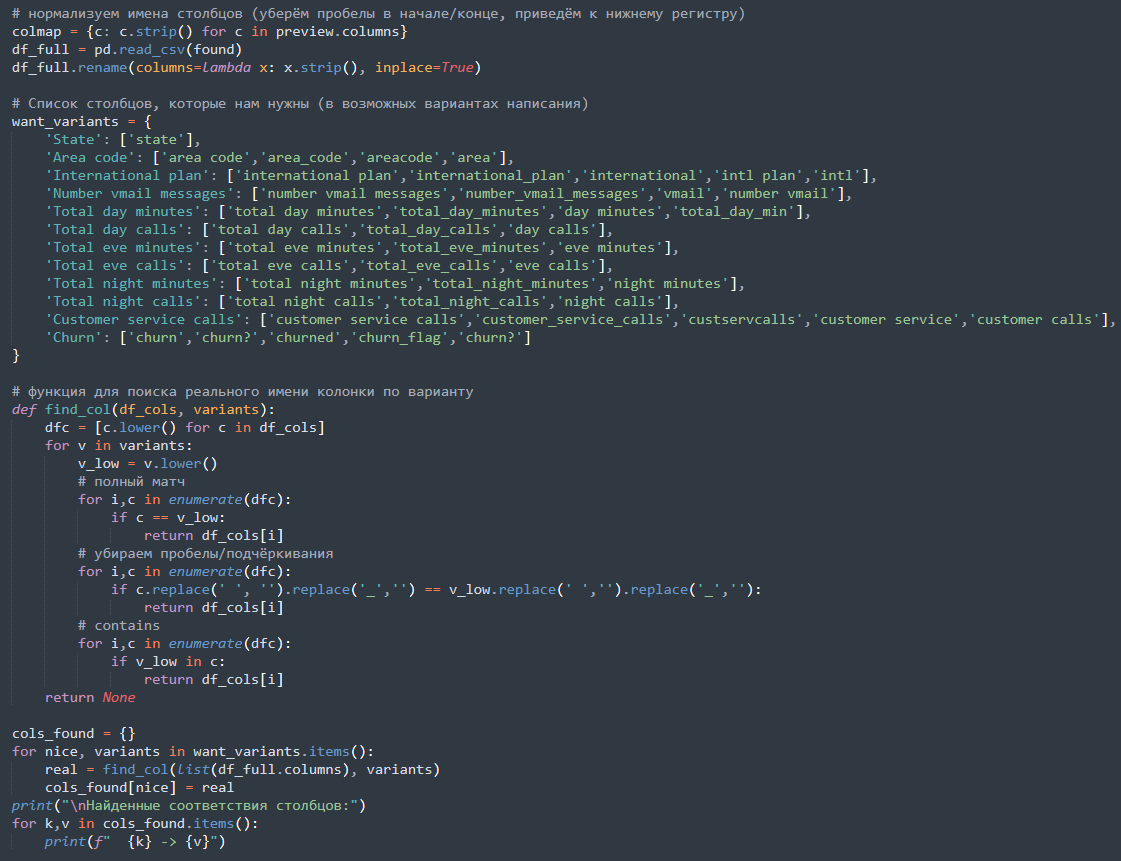


**Решение задания 2:**

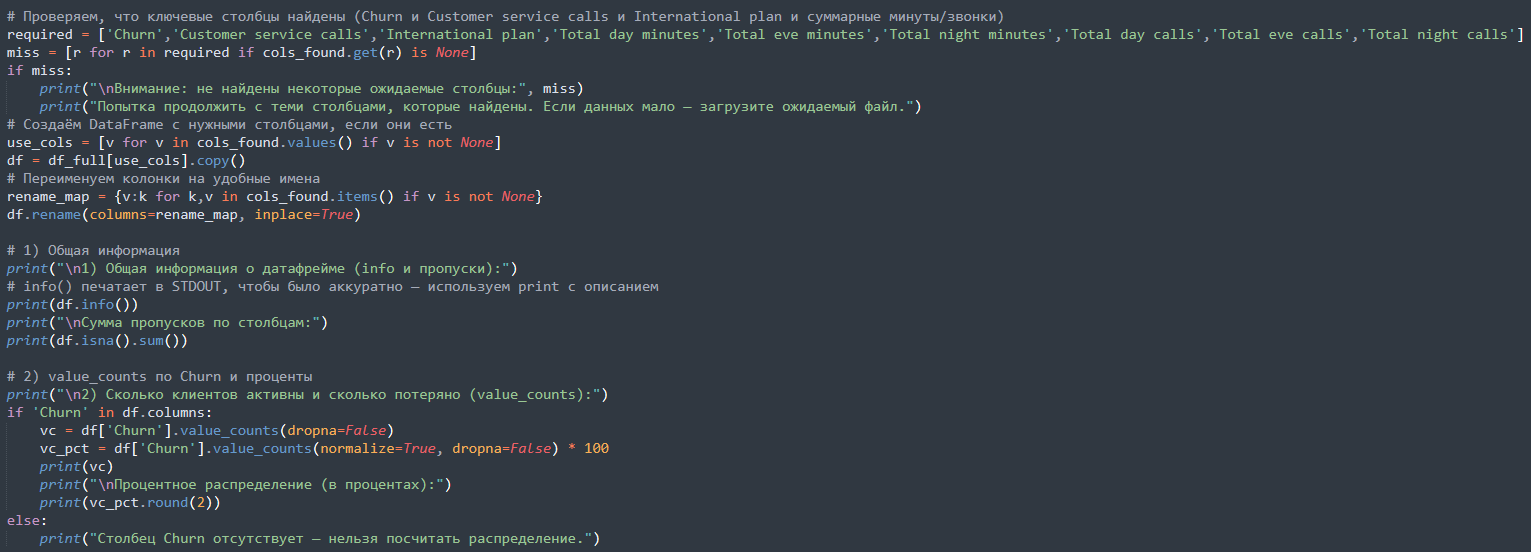
Для начала был произведен поиск файла данных telecom\_churn.csv в нескольких возможных директориях: текущей, родительской, в папках data и /mnt/data. Если файл не обнаружен, выполняется поиск похожих CSV-файлов с ключевыми словами «telecom» или «churn» в названии. В случае неудачи выводится информативное сообщение с рекомендациями о расположении файла. Если файл найден, выводятся имена столбцов первых пяти строк для понимания структуры данных перед загрузкой полного набора. Такой подход обеспечивает гибкую и надежную загрузку данных без необходимости жестко фиксировать путь к файлу.



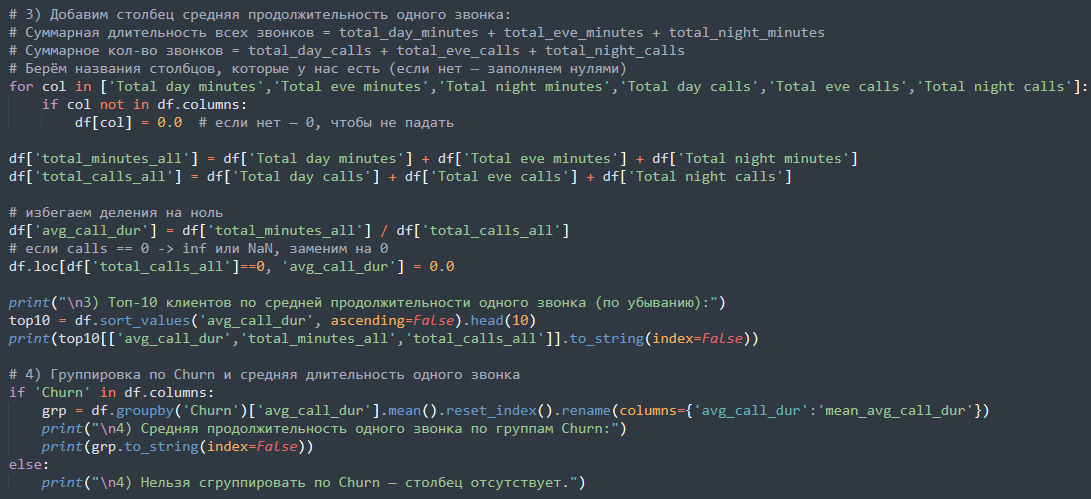
После чего выполняется нормализация и стандартизация названий столбцов для удобства работы с данными: пробелы удаляются, а имена приводятся к нижнему регистру. Затем задается словарь с нужными столбцами и их возможными вариантами написания, чтобы обеспечить гибкий поиск. Функция find\_col реализует поиск реального имени столбца в датасете, используя несколько методов сравнения (точное совпадение, без пробелов и подчеркиваний, вхождение подстроки), что позволяет корректно сопоставлять столбцы даже при различиях в форматировании и именах. После этого собирается словарь найденных соответствий для последующего использования в анализе.



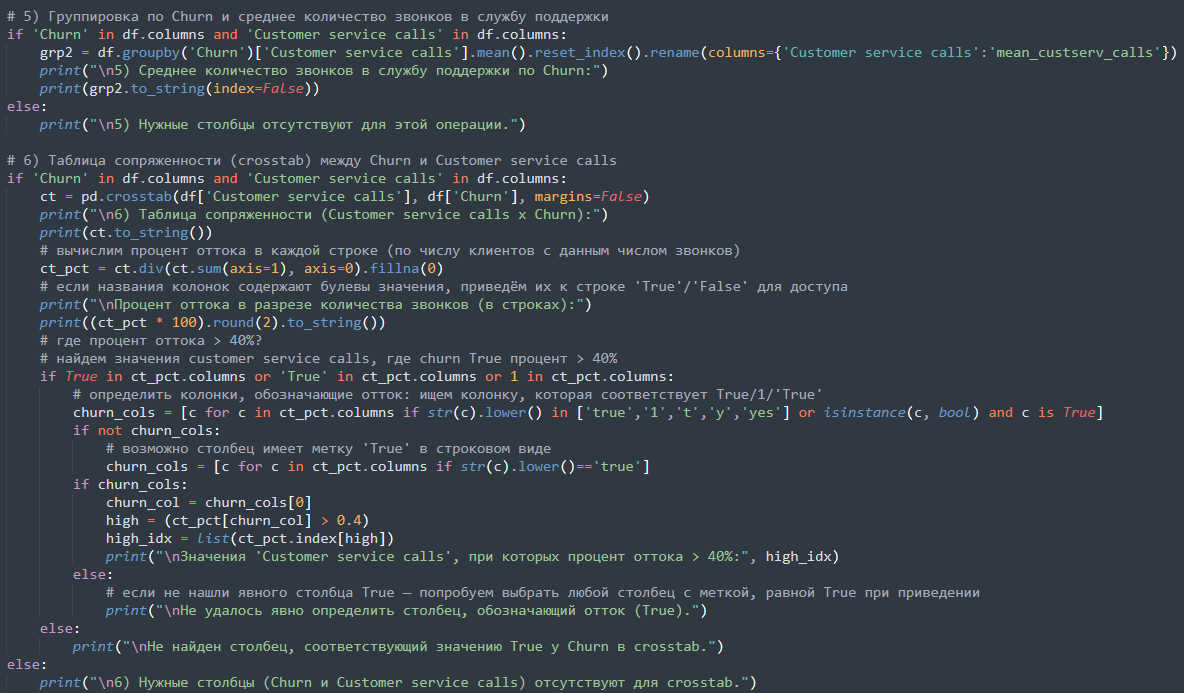
На следующем этапе выполняется проверка наличия всех ключевых столбцов, необходимых для дальнейшего анализа, таких как «Churn», «Customer service calls», «International plan» и суммарные минуты и звонки по разным периодам дня. В случае отсутствия каких-либо из них выводится предупреждение, однако анализ продолжается с доступными данными. Затем создается новый DataFrame, содержащий только найденные нужные столбцы, которые переименовываются в удобочитаемые названия для упрощения работы с ними. Далее производится вывод общей информации о DataFrame, включая типы данных, размер и количество пропущенных значений с помощью метода info(), что позволяет оценить качество и полноту данных. После этого выводится распределение значений в столбце «Churn» и их процентное соотношение, что помогает понять соотношение активных и ушедших клиентов в выборке.



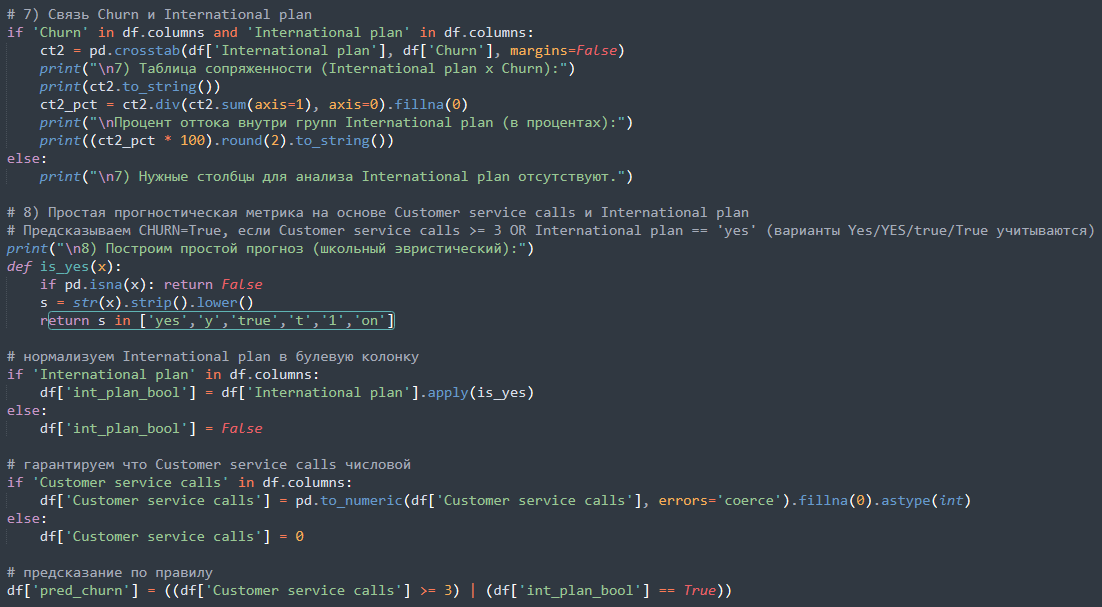
Далее добавляется новый столбец со средней продолжительностью одного звонка, рассчитываемой как отношение суммарного времени звонков за день, вечер и ночь к их общему количеству. Для устойчивости к отсутствующим данным столбцы с минутами и звонками, которые отсутствуют, заполняются нулями, чтобы избежать ошибок. Также предусмотрена обработка деления на ноль: когда количество звонков равно нулю, средняя длительность устанавливается в 0, что предотвращает появление бесконечностей или NaN и позволяет корректно выполнять дальнейший анализ. Затем выводится топ-10 клиентов с наибольшей средней длительностью звонка для выявления наиболее активных. Далее, при наличии столбца «Churn», происходит группировка данных для оценки средней длительности звонка в зависимости от статуса клиента (отток или нет), что помогает выявить закономерности в поведении пользователей.



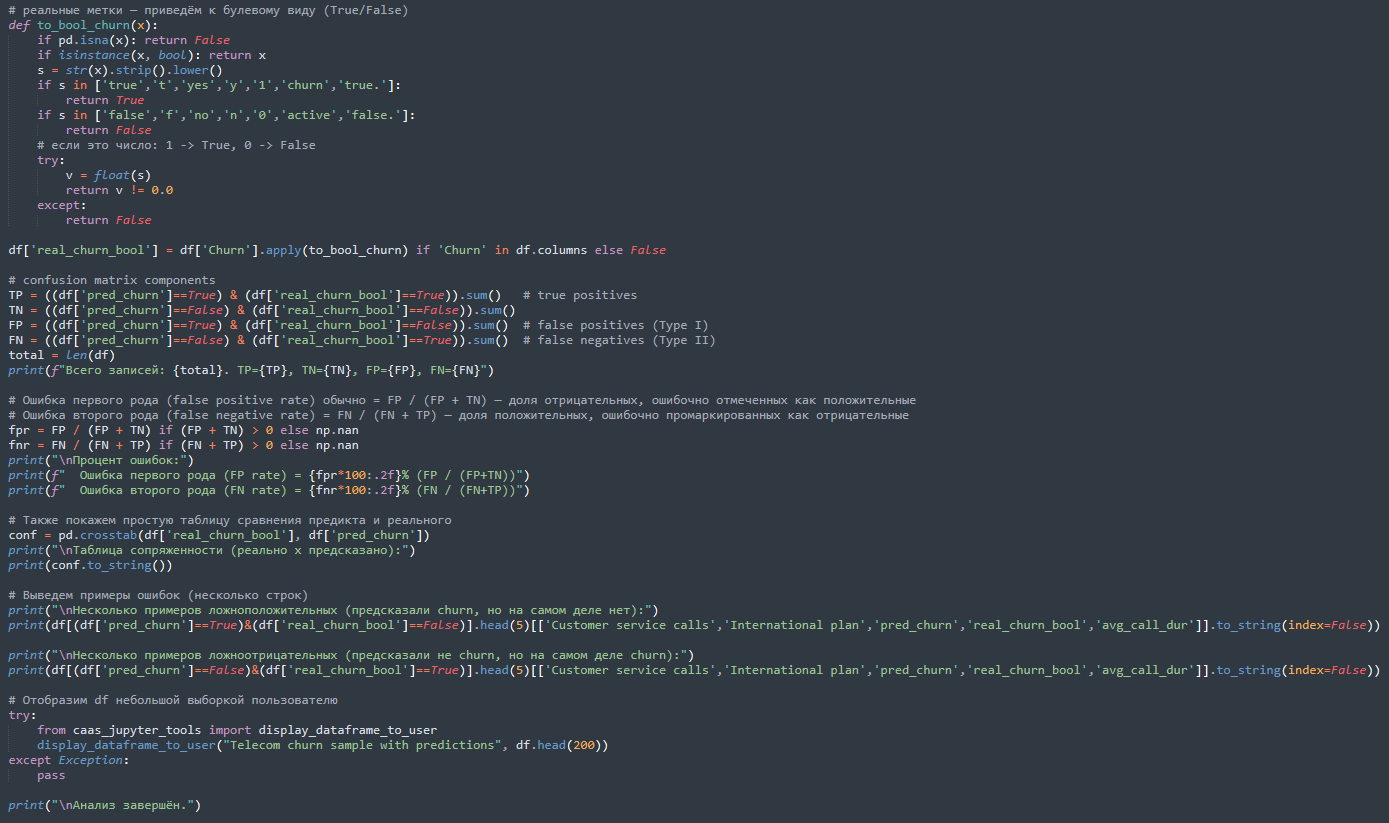
После чего выполняем группировку данных по признаку "Churn" для вычисления среднего количества звонков в службу поддержки у клиентов, которые ушли или остались, а также строим таблицу сопряженности, показывающую связь между числом звонков и уходом клиентов. В случае, если данные о «Churn» и «Customer service calls» присутствуют, он выводит среднее значение и таблицу с пропорциями, а также идентифицирует случаи, когда процент оттока превышает 40% по определённому количеству звонков, что помогает выявить повышенный риск ухода. Если необходимые столбцы отсутствуют, выводится сообщение об отсутствии информации для анализа.



Следующий этап – анализ связи между оттоком клиентов (Churn) и наличием у них международного плана (International plan) с помощью таблицы сопряженности, позволяющей визуализировать распределение клиентов по этим категориям и вычислить процент оттока внутри групп. После чего формируется простая эвристическая прогностическая модель, которая предсказывает отток, если клиент либо имеет более 3 звонков в службу поддержки, либо подключен к международному плану. Для этого международный план нормализуется в булев тип по стандартным значениям "yes", "true" и т.п., а количество звонков приводится к числовому виду.



Ну и в конце реализуется преобразование реальных меток оттока (Churn) к булевому типу для унифицирования значения для последующего сравнения с прогнозом. Затем вычисляются компоненты матрицы ошибок (confusion matrix): истинно-положительные (TP), истинно-отрицательные (TN), ложноположительные (FP) и ложноотрицательные (FN) случаи, по которым рассчитываются ошибки первого и второго рода (false positive rate и false negative rate). Код выводит сводную таблицу с сопоставлением реальных и предсказанных значений, а также примеры ложных срабатываний.

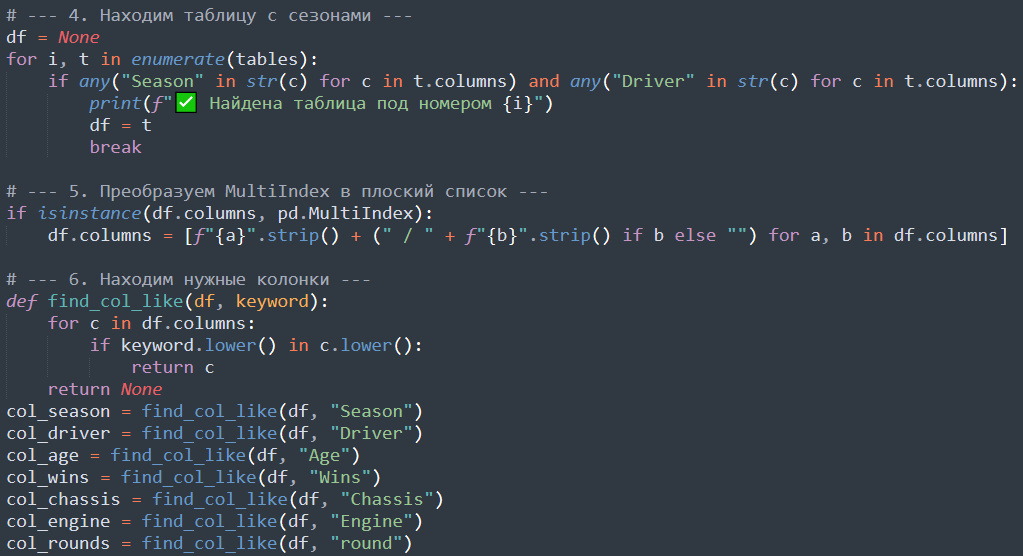


**Решение задания 3:**

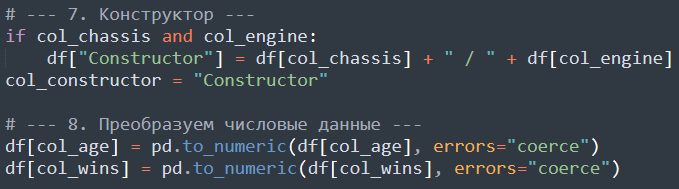
Дня начала загружаем страницу Википедии с перечнем чемпионов Формулы 1 по годам, используя requests и BeautifulSoup для получения и парсинга HTML кода. После получения HTML передаем его в pandas для извлечения всех таблиц, присутствующих на странице, и выводим количество найденных таблиц. Это подготовительный этап для дальнейшей работы с таблицей чемпионов — последующего выбора нужной таблицы и очистки данных.



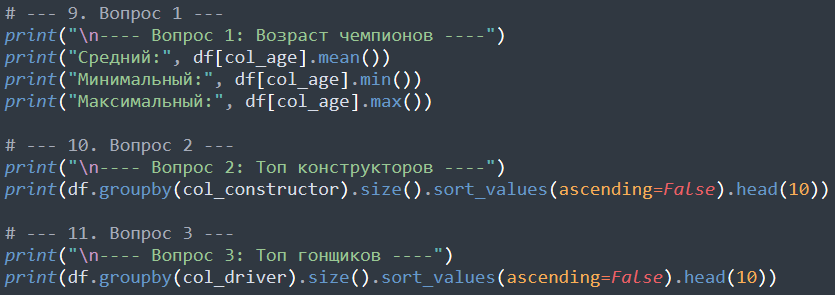
После чего происходит проверка, является ли индекс столбцов MultiIndex, и если да, то он преобразуется в плоский список строк, объединяя уровни через разделитель " / ". Это упрощает дальнейшую работу с таблицей, так как устраняет вложенность в названиях колонок. Затем определяется функция для поиска столбцов, которые содержат заданное ключевое слово (например, "Season", "Driver", "Age" и т.д.), и с её помощью находятся нужные названия столбцов для последующего анализа.



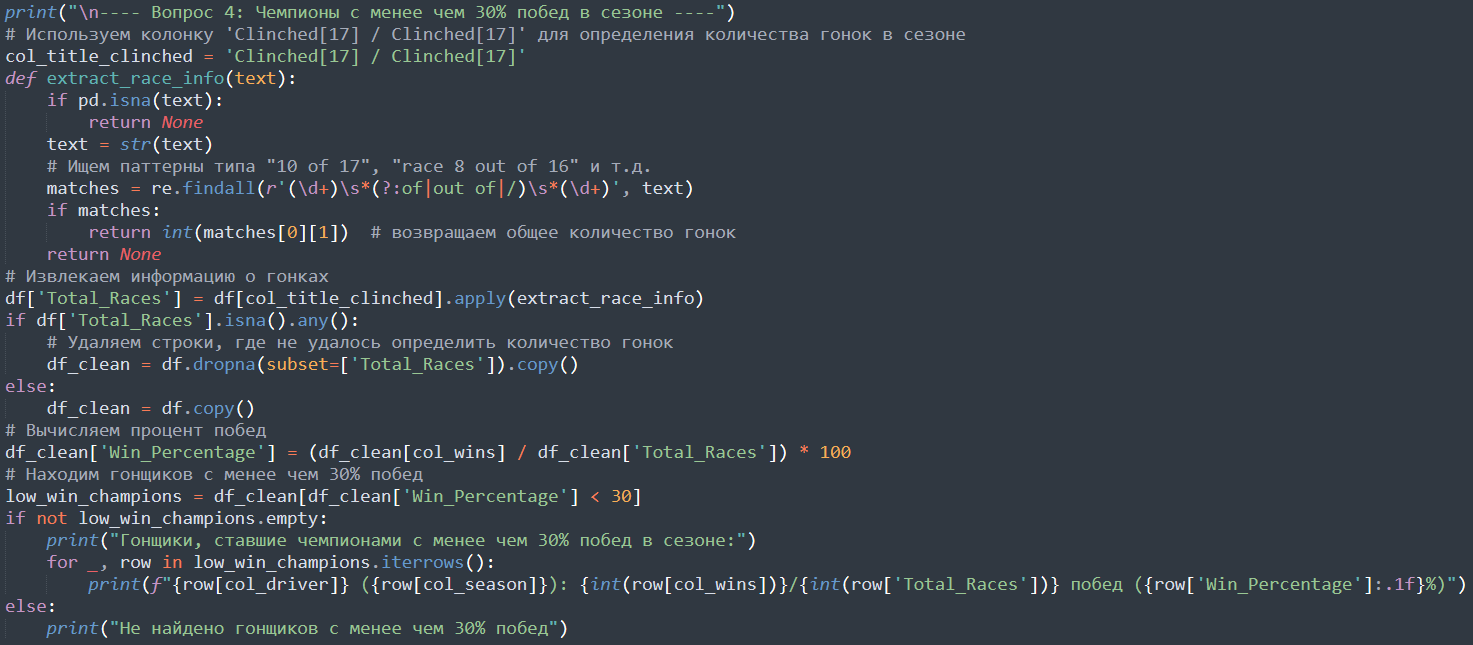
Следующий этап задания – новый столбец "Constructor", который объединяет название шасси и двигателя через " / " для формирования полного название команды-конструктора. После этого числовые столбцы, такие как возраст гонщика и количество побед, преобразуются из строковых значений в числовой тип с помощью pandas. Это необходимо для корректного математического анализа и обработки данных.



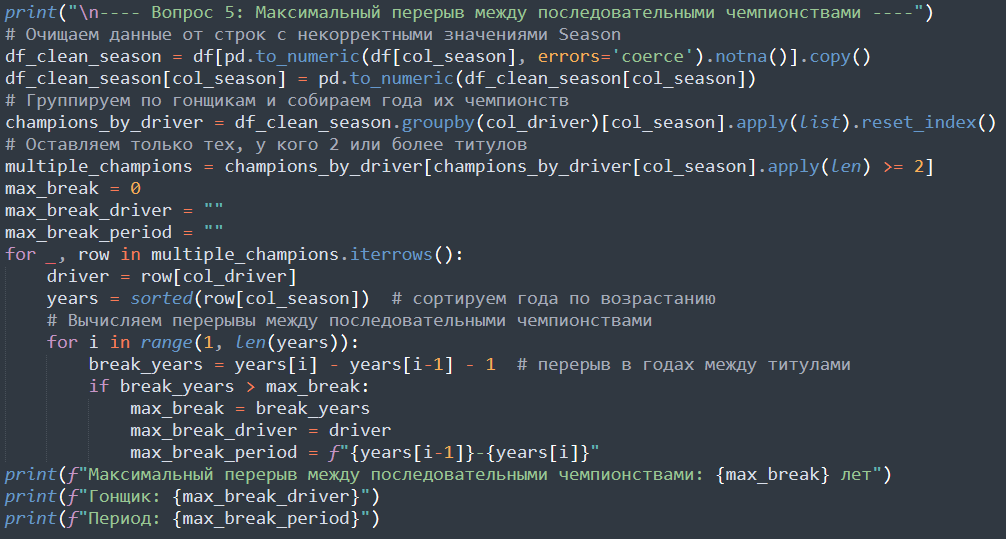
Далее на фото решаются три задачи. Во-первых, вычисляются и выводятся средний, минимальный и максимальный возраст чемпионов Формулы 1, используя числовой столбец с возрастом. Во-вторых, определяется топ конструкторов (команд), которые набрали наибольшее количество чемпионских титулов, группируя по объединенному названию шасси и двигателя. В-третьих, выводится топ гонщиков по числу завоеванных титулов. Получился базовый аналитический блок для получения ключевых статистик по возрасту чемпионов и их командам.



Следующий этап – решение задачи выявления чемпионов с менее чем 30% побед в сезоне. Из столбца со сведениями о количестве гонок в сезоне методом регулярных выражений извлекается общее число гонок, затем вычисляется процент побед каждого чемпиона относительно числа гонок, а записи с неизвестным числом гонок удаляются для точности анализа. Далее фильтруются и выводятся гонщики, чей процент побед меньше 30%. Это позволяет выявить чемпионов, выигравших меньше трети гонок в своем выигрышном сезоне, что является показателем нестандартного пути к титулу.



Последняя часть кода – определение максимального перерыва между последовательными чемпионствами гонщиков, которые становились чемпионами как минимум дважды. Для этого данные очищаются от некорректных значений сезонов, затем для каждого гонщика формируется список годов его титулов, после чего вычисляются интервалы между соседними чемпионствами. Максимальный промежуток запоминается с соответствующим именем гонщика и периодом. Такой подход позволяет выявить самого "устойчивого" или "возвращающегося" чемпиона в истории Формулы 1.



**Вывод:**

В ходе выполнения лабораторной работы были изучены и практически применены основные методы и инструменты библиотек pandas, numpy, scipy и matplotlib для анализа и визуализации данных. Полученные результаты анализа данных позволили наглядно продемонстрировать использование статистических функций, обработку массивов и построение графиков.

Выполненные задачи позволили понять важность комплексного подхода к работе с данными — от предобработки и описательной статистики до визуальной интерпретации результатов. Результаты работы могут быть полезны при дальнейшем анализе больших объемов информации и построении более сложных моделей.

Также выявлены возможности для улучшения работы, включая расширение спектра применяемых методов и углубление анализа. Это может стать основой для последующих исследований и практических применений в области анализа данных.