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

Инструменты хранения и анализа больших данных

Смоляков Руслан Игоревич БД-241м

**Практическая работа 3.1. Анализ и визуализация больших данных. Машинное обучение на больших данных с использованием Apache Spark MLlib**

**Вариант 23**

Направление подготовки/специальность

38.04.05 - Бизнес-информатика

Бизнес-аналитика и большие данные

(очная форма обучения)

Руководитель дисциплины:

Босенко Т.М., доцент департамента

информатики, управления и технологий,

кандидат технических наук

Москва

2025

**Введение**

**Цель и задачи работы:**

* **Познакомиться с понятием «большие данные» и способами их обработки;**
* **Познакомиться с инструментом Apache Spark и возможностями, которые он предоставляет для обработки больших данных.**
* **Получить навыки выполнения разведочного анализа данных использованием pyspark.**

**Индивидуальные задания:**

**Вариант – 23**

|  |
| --- |

**Задание 1 (Интерпретация) - Оцените названия столбцов в исходном датасете (раздел 2). Насколько они понятны для бизнес-пользователя? Есть ли столбцы, требующие переименования или дополнительного описания?**

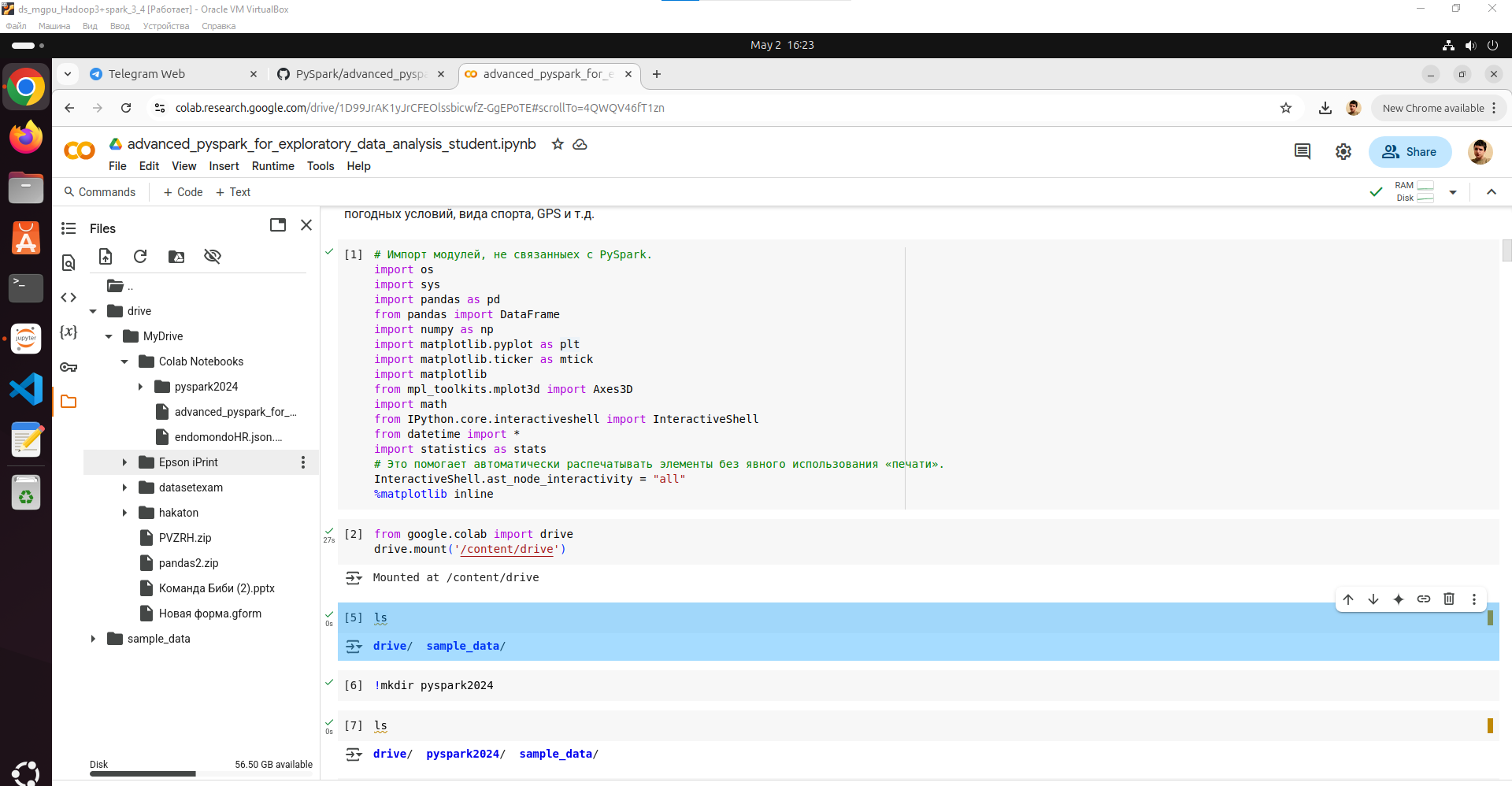
**Задание 2 (Интерпретация) - В разделе 6 рассчитывается процент пользователей, занимающихся более чем 1 видом спорта. Как бы вы интерпретировали этот показатель для бизнеса? Это высокий или низкий процент?**

**Задание 3 (Интерпретация) - Если бы у вас были данные о погоде во время тренировки, как бы вы могли их использовать совместно с данными о duration или heart\_rate?**

**Задание 4 (Практика PySpark/Python) - Напишите код PySpark, чтобы посчитать количество строк в датафрейме df до и после удаления дубликатов по столбцу id (уникальный ID тренировки).**

**Задание 5 (MLlib Концепция) - Почему удаление дубликатов важно перед обучением моделей ML в Spark MLlib? Как дубликаты могут повлиять на качество модели и оценку ее производительности?**

**Основная часть:**Работаю в google collab на ВМ

****

Скачал блокнот, импортировал все необходимые модули, дал доступ к своему диску, создал директорию pyspark2024.

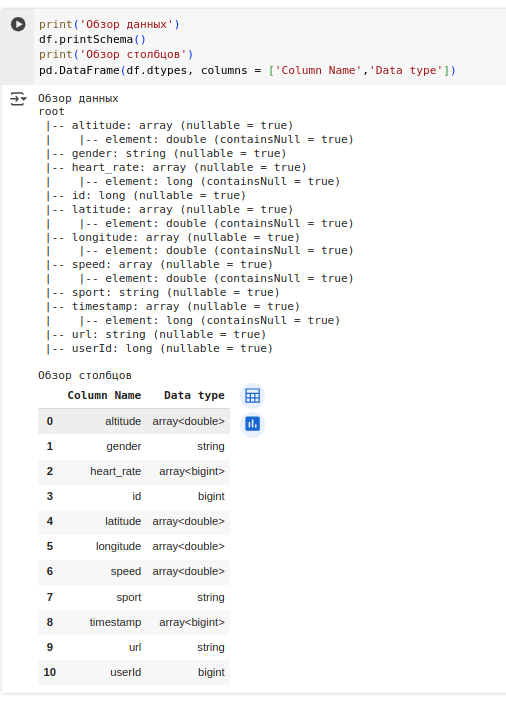


Перешел в новую директорию, разархивировал файл с данными, предварительно закинул его на диск в папку «Colab Notebooks»

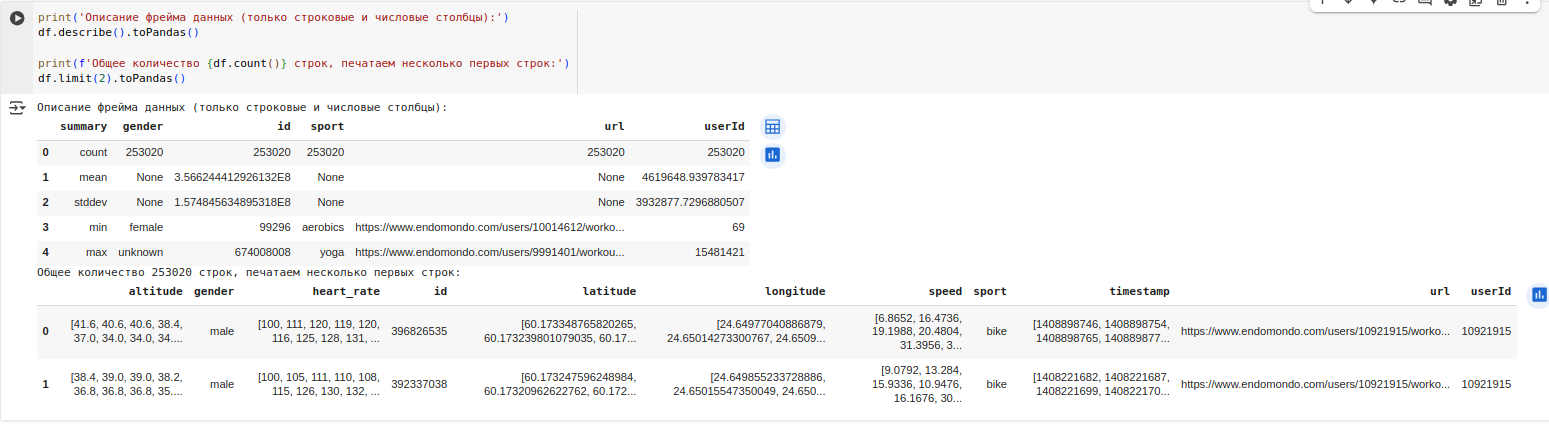


Запустили pyspark.

Приступаем к обзору набора данных:

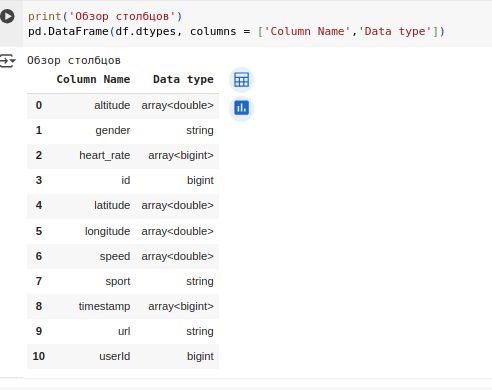


Произвели обзор данных и столбцов.



Описание фрейма данных и вывод кол-ва строк + первые 2 строки.

Можно приступать к «очистке» данных. Сначала будем искать пропуски и нули:



Еще раз обзор столбоцов.

Для строковых столбцов проверяем наличие None и null.

Для числовых столбцов проверяем наличие нулей и NaN.

Для столбцов типа массив проверяем, содержит ли массив нули или NaN

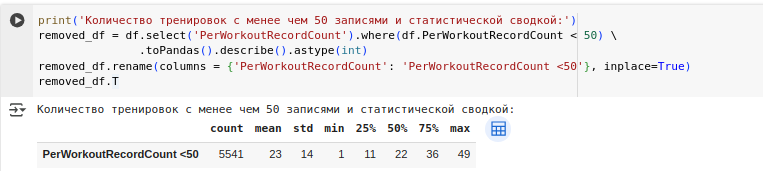


(15 минут выполнялся, думал что зависло все)

Увидели количество пропущенных значених.



Общее кол-во записей более 110 миллионов.



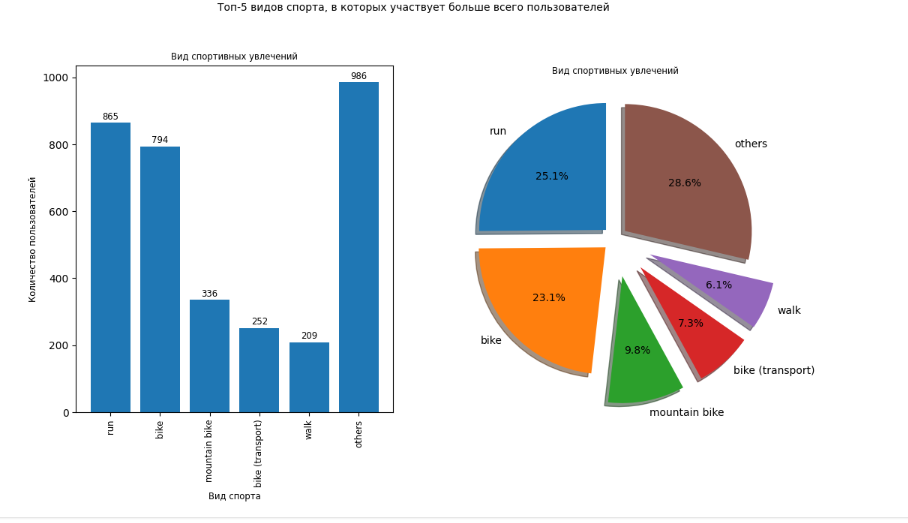
Переходим к ленивой оценке:  
**Ленивая оценка** расширяет возможности Apache Spark за счет сокращения времени выполнения операций RDD.

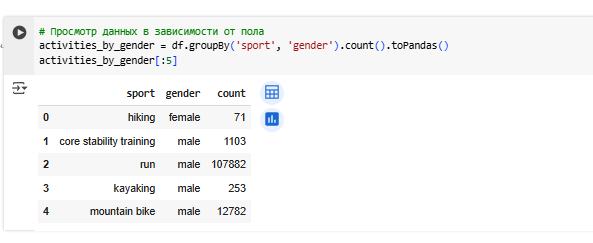
Оцениваем лучшие типы тренировок и выводим их  


Приступаем к исследовательскому анализу данных:



Составили 2 диаграммы с лучшими видами тренировок:



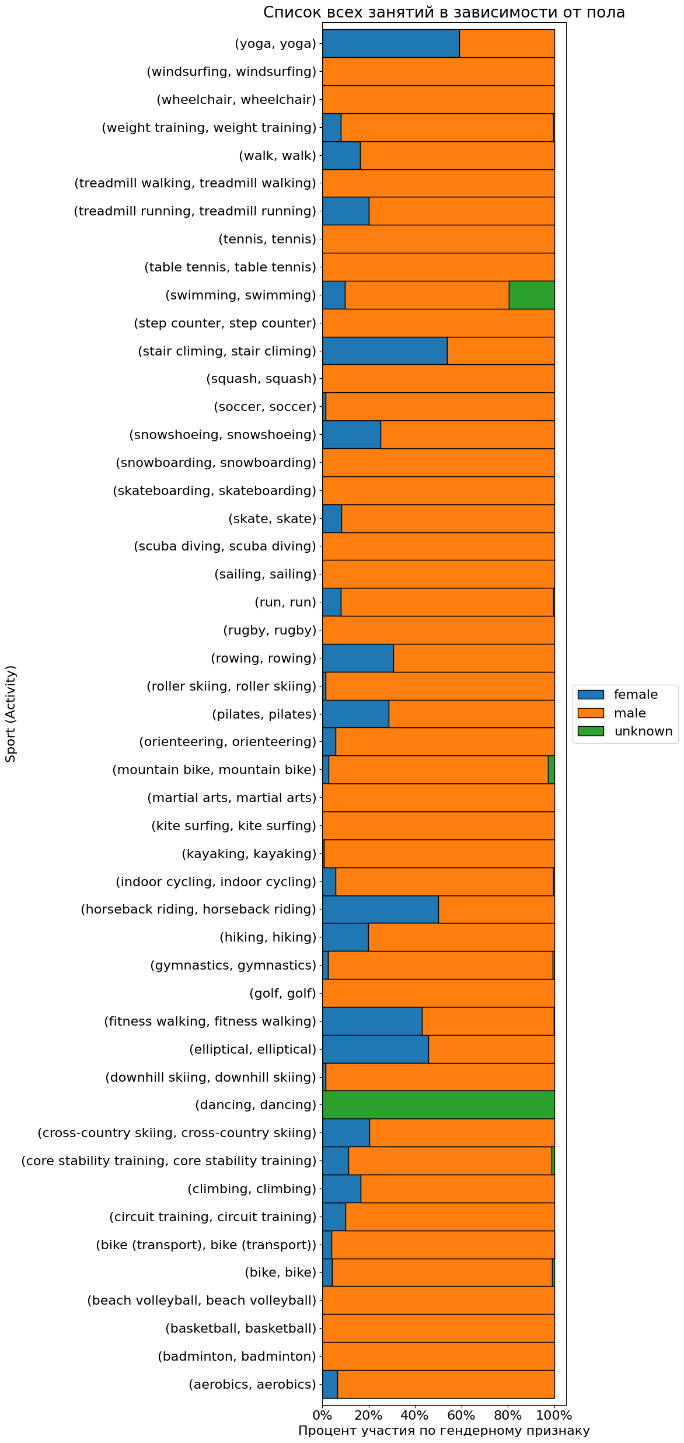


Просмотрели данные в зависимости от пола

Требуется изменить форму приведенной выше таблицы, чтобы разбить столбец пол, для последующей визуализации.

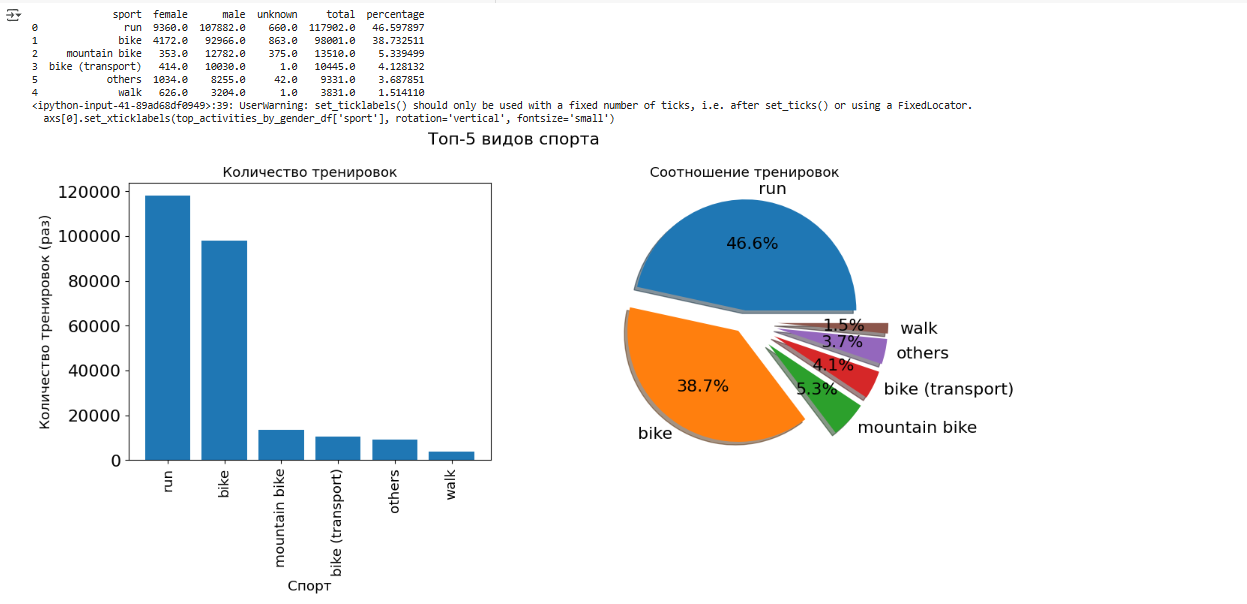
Определим принцип работы с категориями:

* метод DataFrame.stack(): "сводит" уровень (возможно, иерархический) меток столбцов, возвращая a DataFrame с новым, самым внутренним уровнем меток строк;
* метод DataFrame.unstack(): (обратная операция DataFrame.stack()) "сводит" уровень индекса строки (возможно, иерархический) к оси столбца, создавая измененный DataFrame с новым, самым внутренним уровнем меток столбцов.

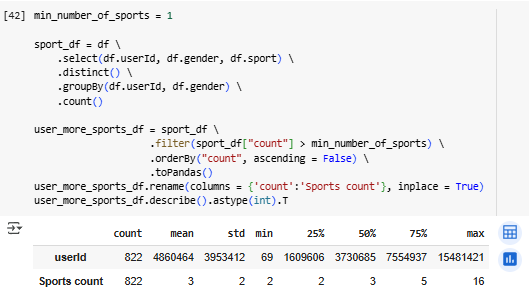


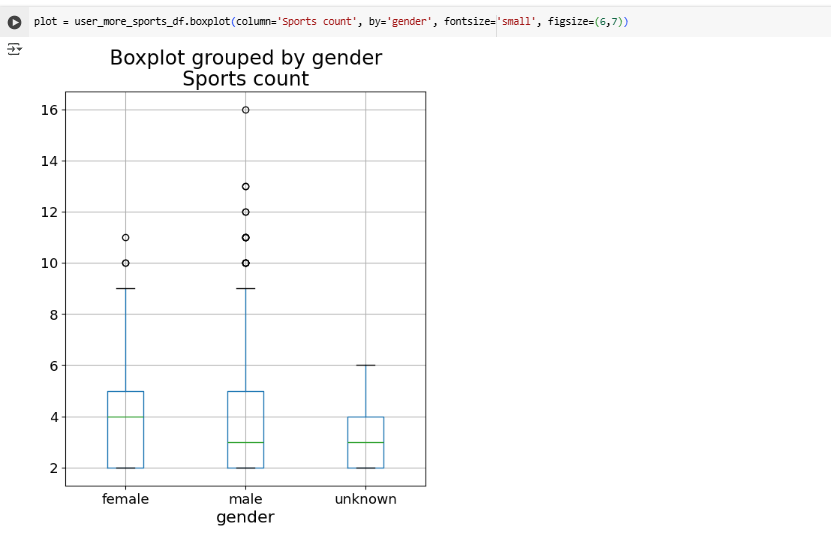
Смотрим верхнее парето из 5 видов спорта, в которых больше всего участников.  


Вывод:



Узнаем сколько людей участвовало более чем в одном виде спорта.



Разобьем по полу  


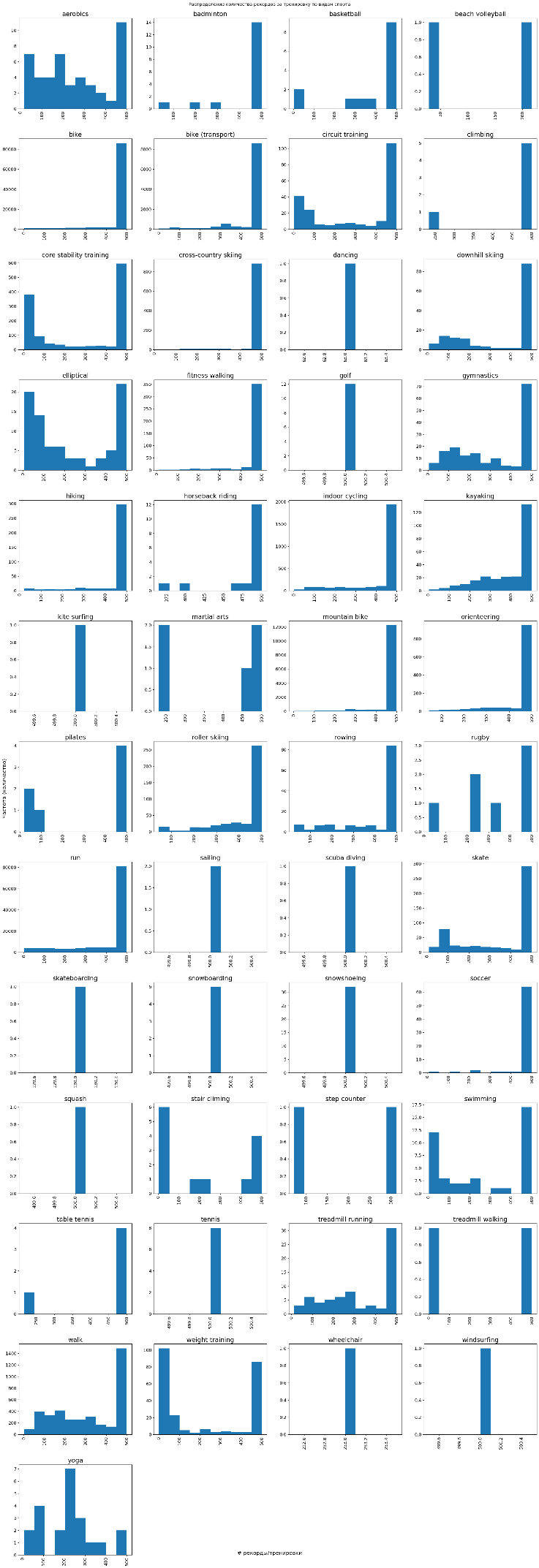
Выбросы есть, но в целом разница между мужчинами и женщинами не очень большая.

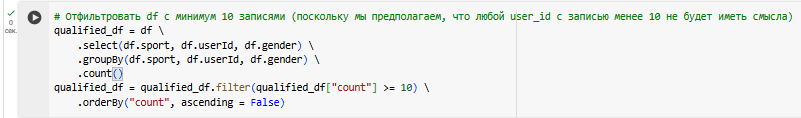
Для более детального наблюдения мы разбили количество рекордов по каждому виду деятельности по каждому отдельному виду спорта.

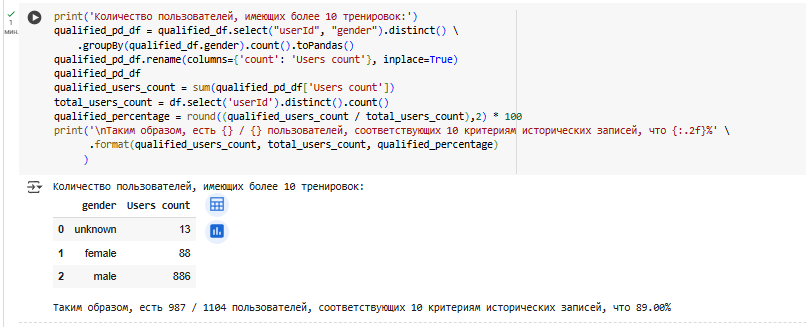
Исходя из распределения, максимальное количество записей за тренировку составляет 500, но не все тренировки и виды спорта достигают этого числа.



Вывод:

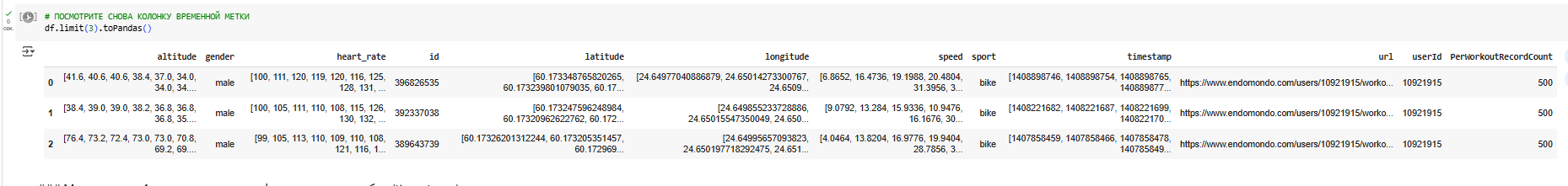


Отфильтруем по тем, у кого больше 10 тренировок:  




Переходим к пользовательским функциям:

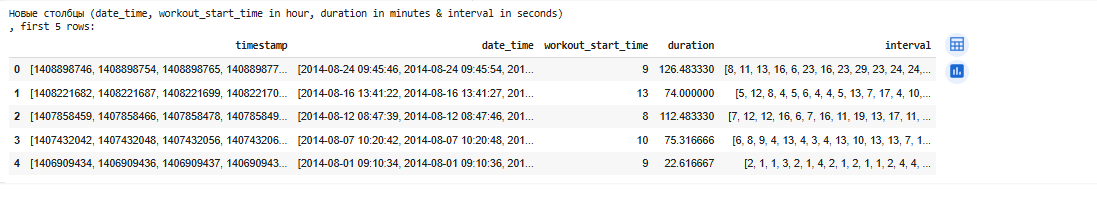
Сначала смотрим колонку временной метки:



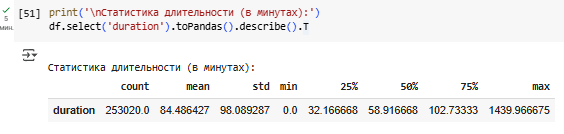
Далее создаем 4 вспомогательные функции для столбца 'timestamp', как описано выше, а затем преобразуем их в UDF.



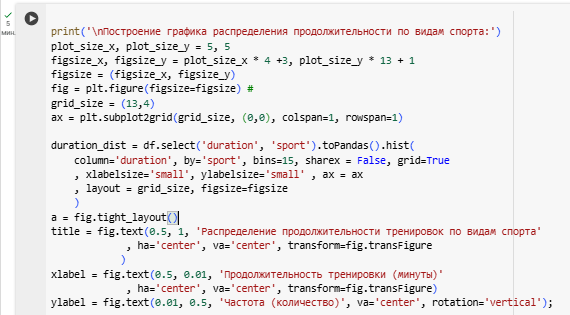
Вывод:



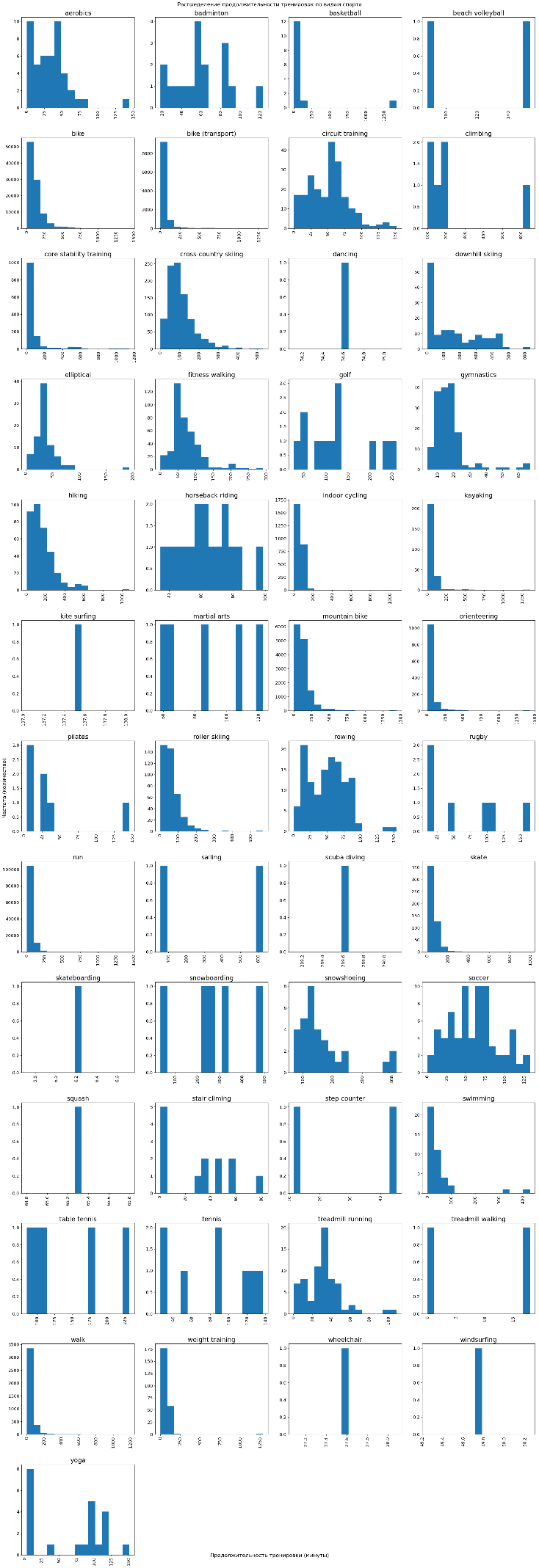
Смотрим на продолжительность каждой тренировки в минутах.



Строим график продолжительности:

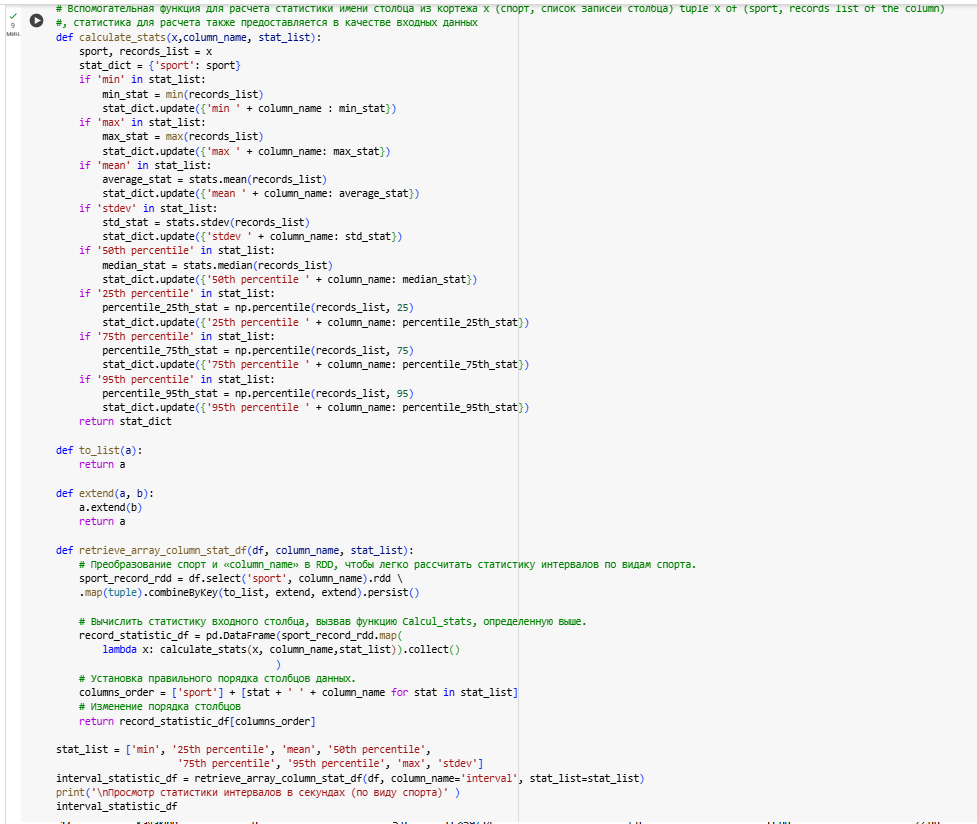


Графики:

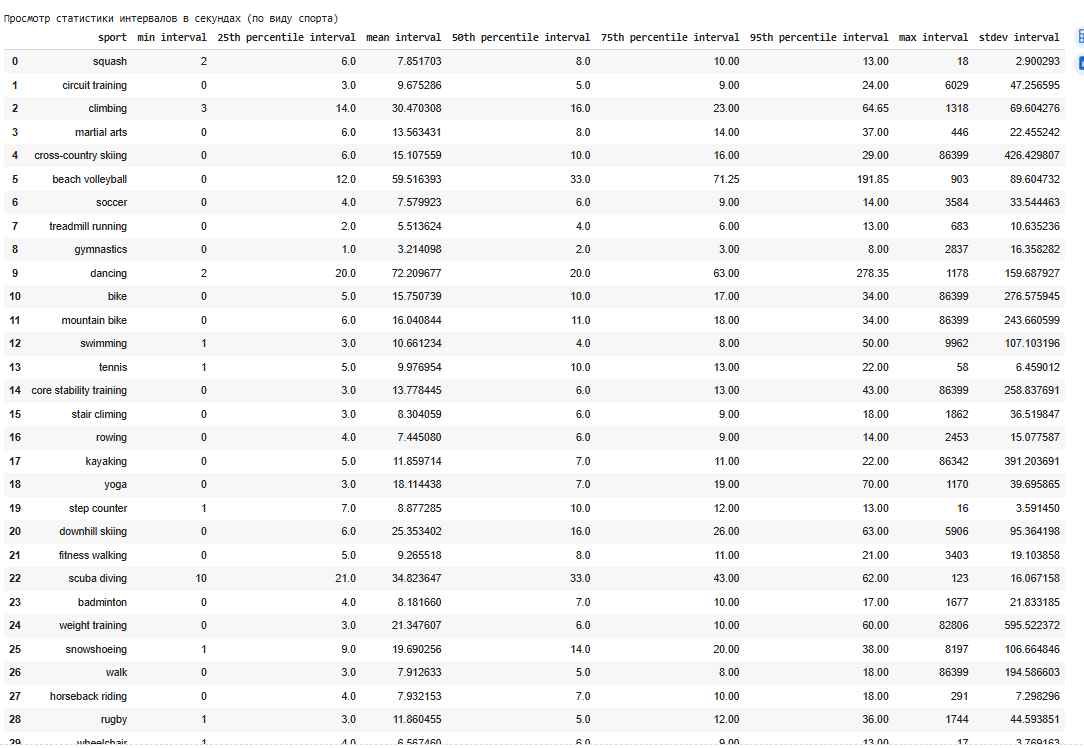


Переходим к следующему этапу: **Преобразование объектов строк в устойчивый распределенный набор данных Spark**

Рассчитаем некоторые основные статистические данные (мин/макс/среднее/среднее/стандартное отклонение и 4 квантиля 25/50/75/95) в pySpark, преобразуем в Rdd и построим их на графике.

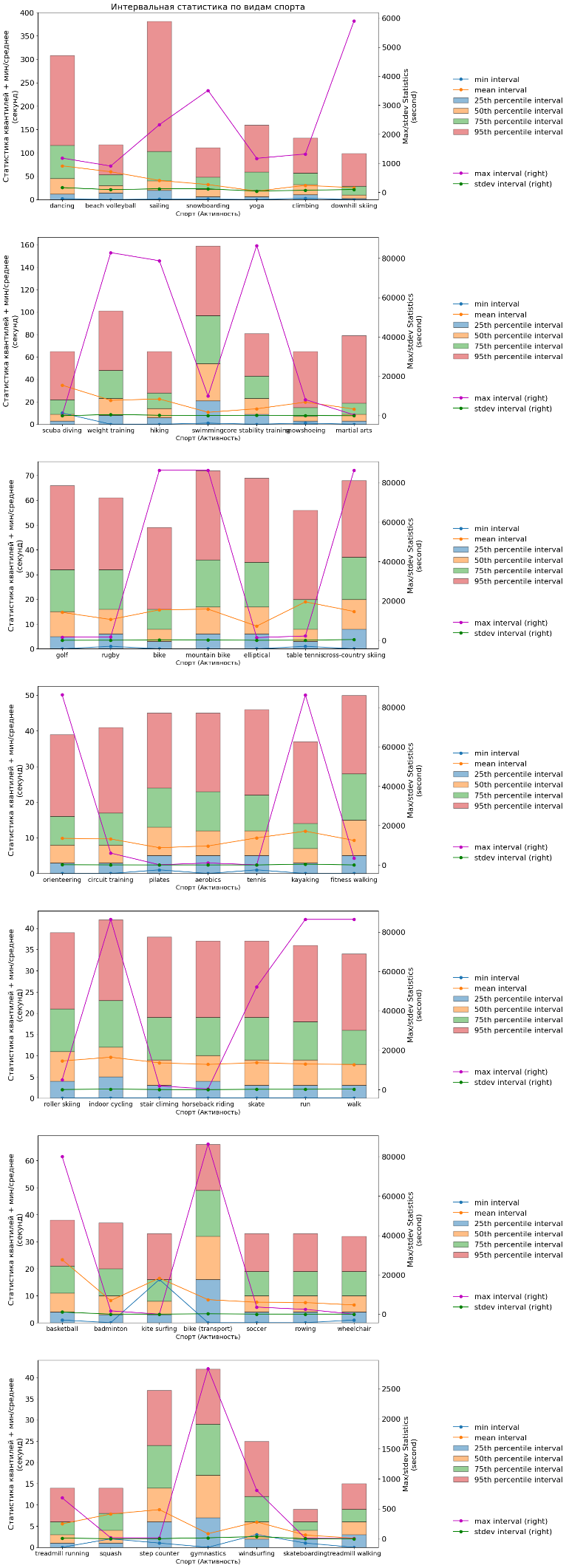


Вывод:

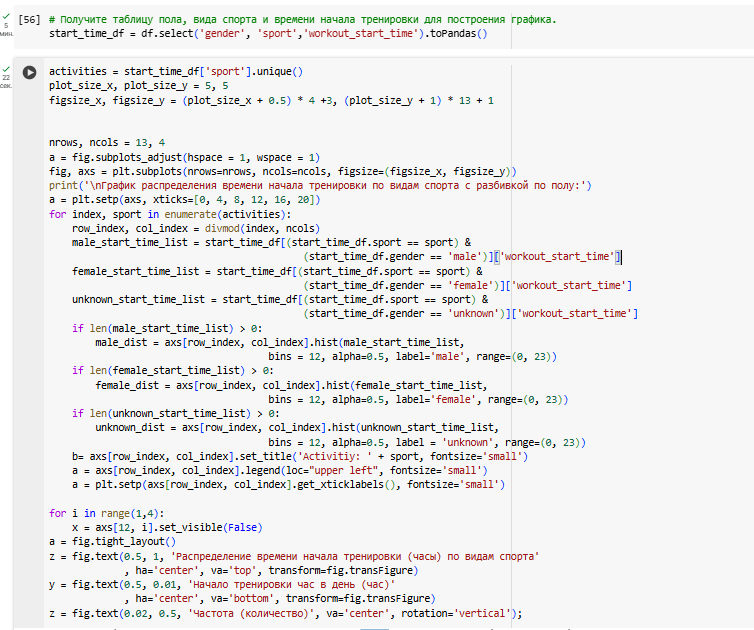


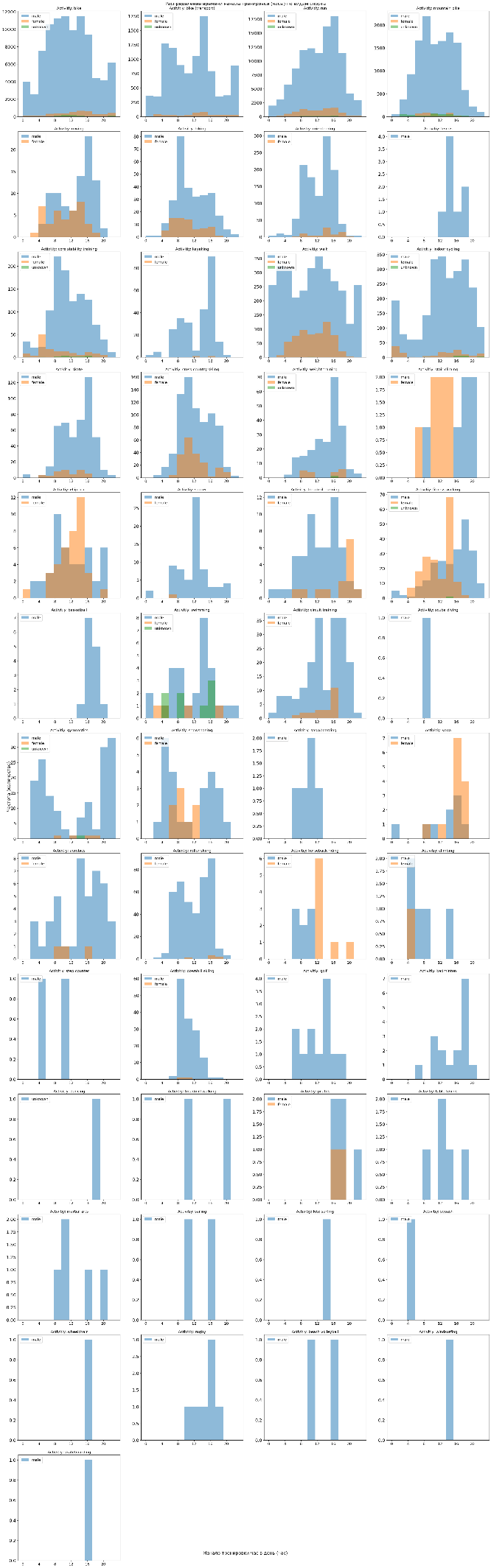
Отображение в виде столбцов и линейных диаграмм:  


Диаграммы:



Используем гистограмму, чтобы посмотреть на распределение часов начала тренировок, сгруппированных по видам спорта и с разбивкой по полу. Мы делим день на интервалы по 2 часа, всего получается 12 частей.

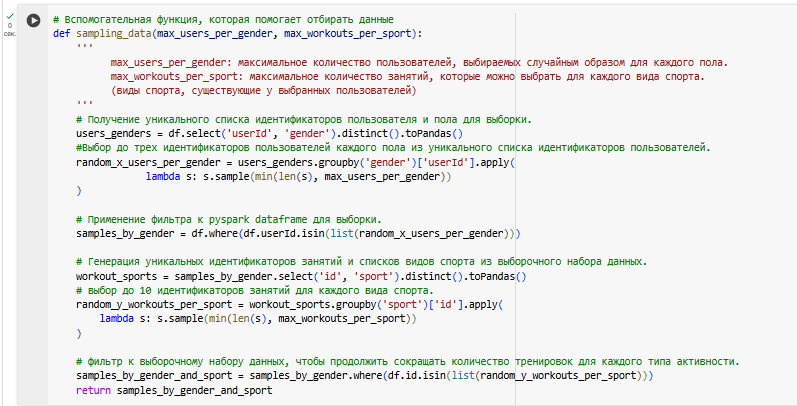


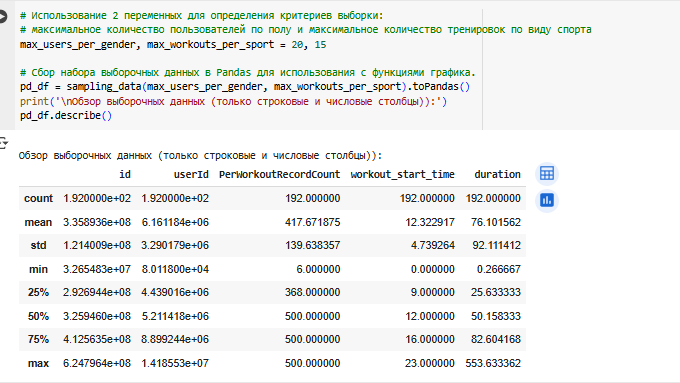
Графики:  


Смотрим глубже на информацию на уровне строки:

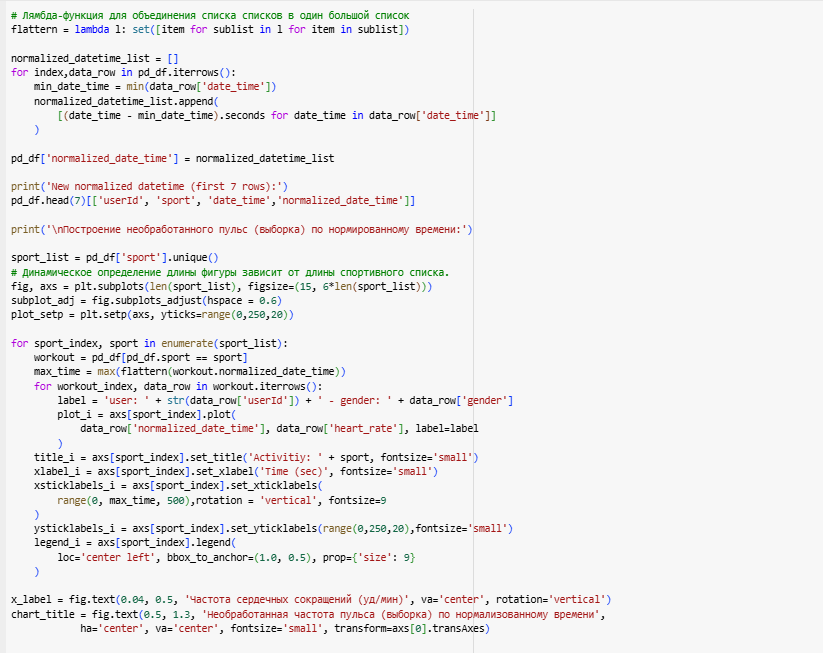


Из-за огромного количества пользователей и количества тренировок мы случайным образом выбрали до x количества пользователей каждого пола (например, 5) и до y тренировок по каждому типу активности (например, 10).

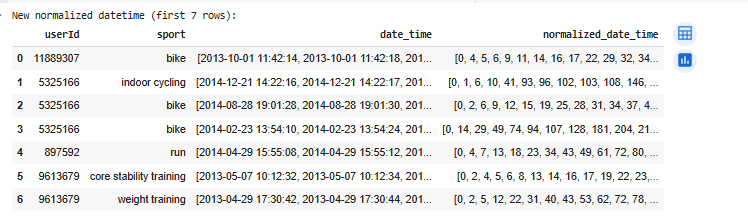


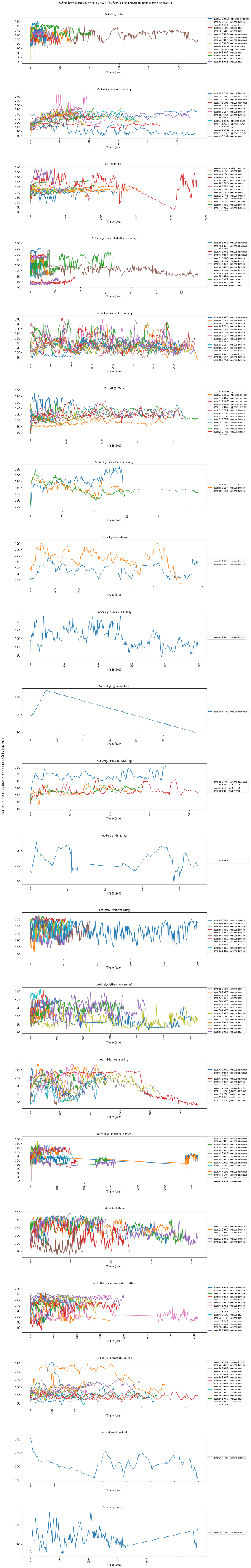


Нормализуем время для всех тренировок, рассчитав продолжительность (в секундах) каждой записи временной метки из первой записи тренировки (первый элемент datetime списка в этой тренировке).  
Затем отображаем частоту сердечных сокращений в зависимости от этого нормализованного времени, группируя по видам спорта.



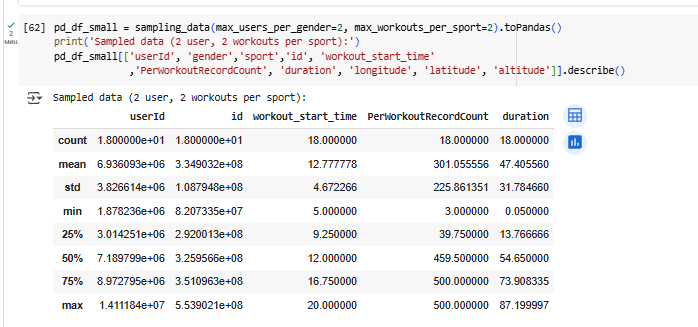
Вывод и графики:

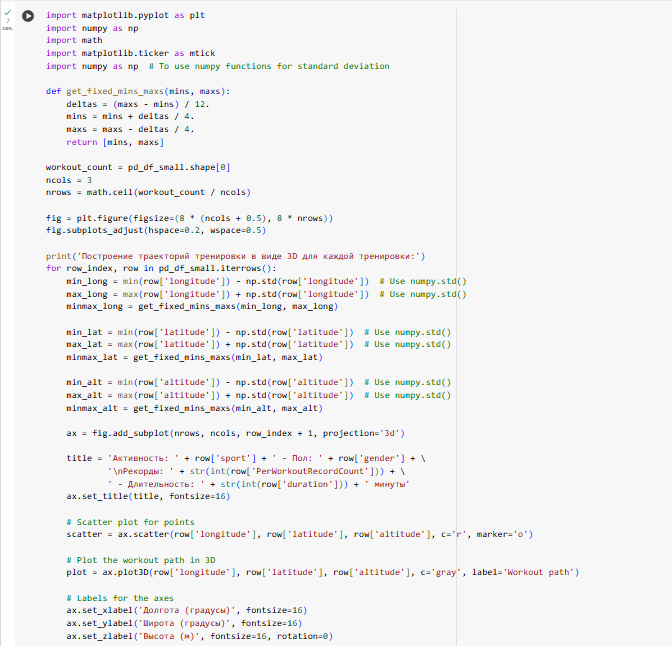
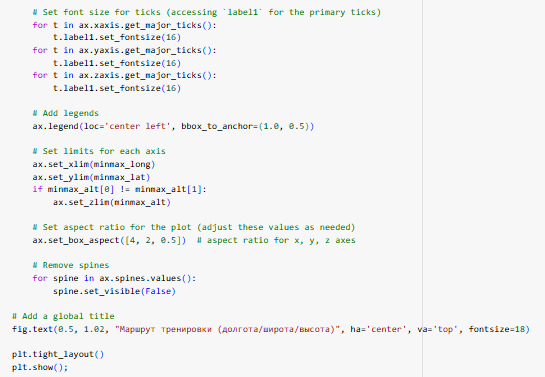


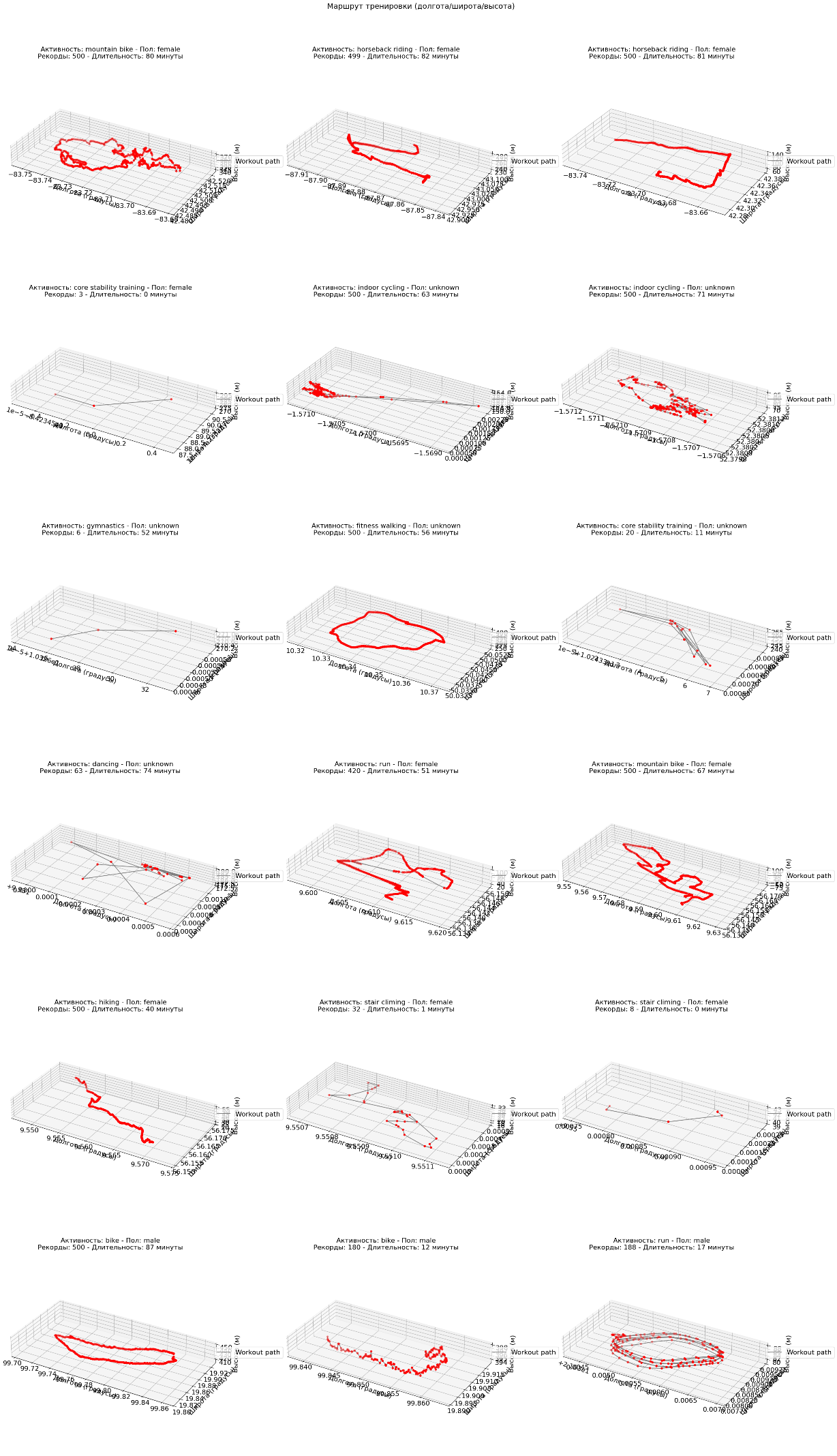


Перемещения во время тренировки

У нас будет некоторая визуализация в трех столбцах с информацией о смещении/геометрии (longitude,latitude и altitude). Поскольку расположение каждого пользователя и тренировки отличается друг от друга, мы отображаем только несколько отдельных тренировок на 3D-графиках, чтобы просмотреть маршрут тренировки.



Графики:  


**Индивидуальные задания:**

**Вариант – 23**

|  |
| --- |

**Задание 1 (Интерпретация) - Оцените названия столбцов в исходном датасете (раздел 2). Насколько они понятны для бизнес-пользователя? Есть ли столбцы, требующие переименования или дополнительного описания?**

**Задание 2 (Интерпретация) - В разделе 6 рассчитывается процент пользователей, занимающихся более чем 1 видом спорта. Как бы вы интерпретировали этот показатель для бизнеса? Это высокий или низкий процент?**

**Задание 3 (Интерпретация) - Если бы у вас были данные о погоде во время тренировки, как бы вы могли их использовать совместно с данными о duration или heart\_rate?**

**Задание 4 (Практика PySpark/Python) - Напишите код PySpark, чтобы посчитать количество строк в датафрейме df до и после удаления дубликатов по столбцу id (уникальный ID тренировки).**

**Задание 5 (MLlib Концепция) - Почему удаление дубликатов важно перед обучением моделей ML в Spark MLlib? Как дубликаты могут повлиять на качество модели и оценку ее производительности?**

**Задание 1.**

Названия столбцов:



Понятные:

* gender – пол пользователя. Понятное название.
* sport – вид спорта. Интуитивно ясно.
* url – ссылка на запись или ресурс. Общепринятое обозначение.
* userId – уникальный идентификатор пользователя. Часто используется в бизнес-аналитике.

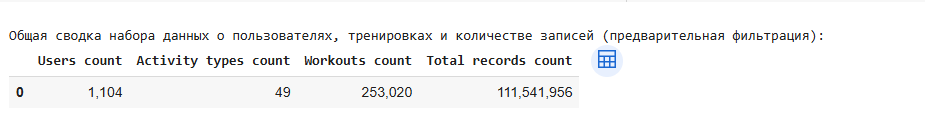
Требуют доп информации:

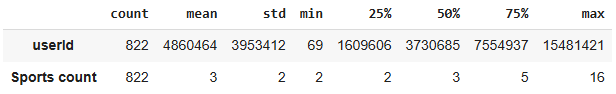
* id – слишком общее название.
* altitude
* latitude
* longitude
* speed
* timestamp
* heart\_rate.

Все эти поля являются массивами, обозначения полей понятны, но следует добавить единицы измерения, например «altitude\_m» или «heart\_rate\_bpm».

**Задание 2.**

Общее количество пользователей (из раздела 2):



Количество пользователей, участвующих более чем в одном виде спорта:  


Рассчитаем процентное соотношение:  
Процент пользователей= 822/1104 ​× 100 ≈ 74.45%

Это высокий процент , который указывает на то, что большинство пользователей платформы имеют разнообразные интересы и используют сервис для разных видов физической активности.

**Задание 3.**

Приведу по 3 примера:

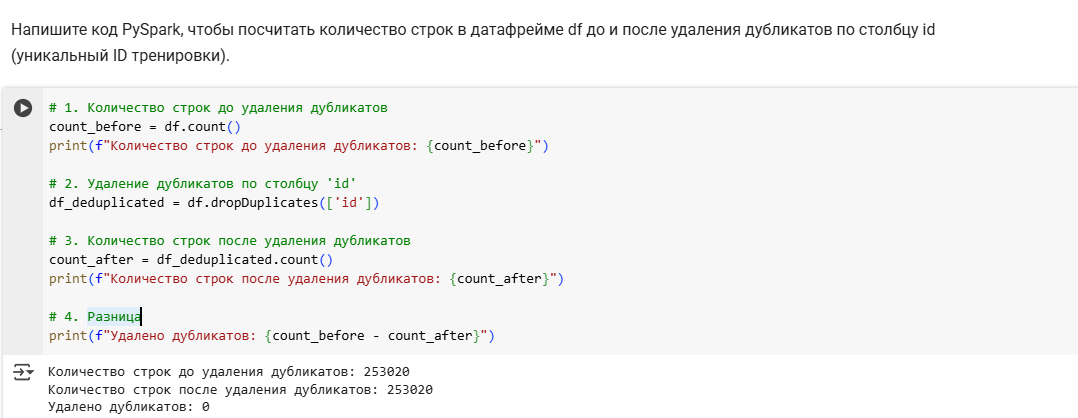
**duration:**

1. Анализ влияния дождя на продолжительность тренировок:
   * Понять, как осадки влияют на желание пользователей заниматься на улице.
   * Например: "Во время дождя средняя продолжительность бега снижается на 20%".
2. Сравнение сезонной активности (весна vs лето vs зима):
   * Определить, в какие сезоны пользователи тренируются дольше.
   * Это помогает планировать маркетинговые кампании и рекомендации.
3. Прогнозирование продолжительности тренировки на основе прогноза погоды:
   * Строить модель, которая предсказывает, сколько времени пользователь может потратить на тренировку завтра, исходя из погоды.

**heart\_rate:**

1. **Оценка влияния температуры на пульс во время тренировки:**
   * **Выявить, увеличивается ли пульс при высокой температуре даже при одинаковой нагрузке.**
   * **Например: "При температуре выше 30°C пульс растёт на 10% по сравнению с комфортными условиями".**
2. **Изучение влияния влажности на восстановление после тренировки:**
   * **Проверить, как влажный воздух влияет на восстановление пульса после окончания активности.**
3. **Персонализированные уведомления о возможном перегреве или переохлаждении:**
   * **На основе погоды и текущего пульса отправлять пользователям рекомендации:  
     *"Сегодня очень жарко. Ваш пульс повышен — возможно, стоит снизить интенсивность."***

**Задание 4**.

****

На всякий случай проверил строки с дубликатами отдельно

****

**Задание 5.**

Удаление дубликатов важно, потому что они искажают распределение данных, ведут к переобучению модели и завышенной оценке качества на тесте. Дубликаты нарушают независимость наблюдений, что делает результаты модели некорректными и менее обобщающими.

Вывод:

Работа демонстрирует практическое применение инструментов Apache Spark и PySpark для обработки, анализа и подготовки больших данных с последующей подготовкой к задачам машинного обучения. Полученные результаты могут быть использованы для улучшения пользовательского опыта, персонализации рекомендаций и анализа физической активности.