**Яндекс.Музыка**

Сравнение Москвы и Петербурга окружено мифами. Например:

* Москва — мегаполис, подчинённый жёсткому ритму рабочей недели;
* Петербург — культурная столица, со своими вкусами.

На данных Яндекс Музыки вы сравните поведение пользователей двух столиц.

**Цель исследования** — проверьте три гипотезы:

1. Активность пользователей зависит от дня недели. Причём в Москве и Петербурге это проявляется по-разному.
2. В понедельник утром в Москве преобладают одни жанры, а в Петербурге — другие. Так же и вечером пятницы преобладают разные жанры — в зависимости от города.
3. Москва и Петербург предпочитают разные жанры музыки. В Москве чаще слушают поп-музыку, в Петербурге — русский рэп.

**Ход исследования**

Данные о поведении пользователей вы получите из файла yandex\_music\_project.csv. О качестве данных ничего не известно. Поэтому перед проверкой гипотез понадобится обзор данных.

Вы проверите данные на ошибки и оцените их влияние на исследование. Затем, на этапе предобработки вы поищете возможность исправить самые критичные ошибки данных.

Таким образом, исследование пройдёт в три этапа:

1. Обзор данных.
2. Предобработка данных.
3. Проверка гипотез.

## Обзор данных

Составьте первое представление о данных Яндекс Музыки.

**Задание 1**

Основной инструмент аналитика — pandas. Импортируйте эту библиотеку.

**import** pandas **as** pd

**Задание 2**

Прочитайте файл yandex\_music\_project.csv из папки /datasets и сохраните его в переменной df:

df**=**pd**.**read\_csv('/datasets/yandex\_music\_project.csv')

**Задание 3**

Выведите на экран первые десять строк таблицы:

df**.**head(10)

**Задание 4**

Одной командой получить общую информацию о таблице c помощью метода info():

df**.**info()

Итак, в таблице семь столбцов. Тип данных во всех столбцах — object.

Согласно документации к данным:

* userID — идентификатор пользователя;
* Track — название трека;
* artist — имя исполнителя;
* genre — название жанра;
* City — город пользователя;
* time — время начала прослушивания;
* Day — день недели.

Количество значений в столбцах различается. Значит, в данных есть пропущенные значения.

**Задание 5**

**Вопрос со свободной формой ответа**

В названиях колонок видны три нарушения стиля:

1. Строчные буквы сочетаются с прописными.
2. Встречаются пробелы.

название не в змеином\_регистре

**Выводы**

В каждой строке таблицы — данные о прослушанном треке. Часть колонок описывает саму композицию: название, исполнителя и жанр. Остальные данные рассказывают о пользователе: из какого он города, когда он слушал музыку.

Предварительно можно утверждать, что, данных достаточно для проверки гипотез. Но встречаются пропуски в данных, а в названиях колонок — расхождения с хорошим стилем.

Чтобы двигаться дальше, нужно устранить проблемы в данных.

## Предобработка данных

Исправьте стиль в заголовках столбцов, исключите пропуски. Затем проверьте данные на дубликаты.

### Стиль заголовков

**Задание 6**

Выведите на экран названия столбцов:

df**.**columns

Out[5]:

**Задание 7**

Приведите названия в соответствие с хорошим стилем:

* несколько слов в названии запишите в «змеином\_регистре»,
* все символы сделайте строчными,
* устраните пробелы.

Для этого переименуйте колонки так:

* ' userID' → 'user\_id';
* 'Track' → 'track';
* ' City ' → 'city';
* 'Day' → 'day'.

df**=**df**.**rename(columns**=**{' userID':'user\_id', 'Track':'track', ' City ':'city', 'Day':'day'})

**Задание 8**

Проверьте результат. Для этого ещё раз выведите на экран названия столбцов:

df**.**columns

**Задание 9**

Сначала посчитайте, сколько в таблице пропущенных значений. Для этого достаточно двух методов pandas:

df**.**isna()**.**sum()

Не все пропущенные значения влияют на исследование. Так в track и artist пропуски не важны для вашей работы. Достаточно заменить их явными обозначениями.

Но пропуски в genre могут помешать сравнению музыкальных вкусов в Москве и Санкт-Петербурге. На практике было бы правильно установить причину пропусков и восстановить данные. Такой возможности нет в учебном проекте. Придётся:

* заполнить и эти пропуски явными обозначениями,
* оценить, насколько они повредят расчётам.

**Задание 10**

Замените пропущенные значения в столбцах track, artist и genre на строку 'unknown'. Для этого создайте список columns\_to\_replace, переберите его элементы циклом for и для каждого столбца выполните замену пропущенных значений:

columns\_to\_replace**=**['track', 'artist', 'genre']

**for** column **in** columns\_to\_replace:

df[column]**=**df[column]**.**fillna('unknown')

**Задание 11**

Убедитесь, что в таблице не осталось пропусков. Для этого ещё раз посчитайте пропущенные значения.

df**.**isna()**.**sum()

### Дубликаты

**Задание 12**

Посчитайте явные дубликаты в таблице одной командой:

df**.**duplicated()**.**sum()

**Задание 13**

Вызовите специальный метод pandas, чтобы удалить явные дубликаты:

df**=**df**.**drop\_duplicates()

**Задание 14**

Ещё раз посчитайте явные дубликаты в таблице — убедитесь, что полностью от них избавились:

df**.**duplicated()**.**sum()

Теперь избавьтесь от неявных дубликатов в колонке genre. Например, название одного и того же жанра может быть записано немного по-разному. Такие ошибки тоже повлияют на результат исследования.

**Задание 15**

Выведите на экран список уникальных названий жанров, отсортированный в алфавитном порядке. Для этого:

* извлеките нужный столбец датафрейма,
* примените к нему метод сортировки,
* для отсортированного столбца вызовите метод, который вернёт уникальные значения из столбца.

sorted\_df **=** df['genre']**.**sort\_values()

print(sorted\_df**.**unique())

**Задание 16**

Просмотрите список и найдите неявные дубликаты названия hiphop. Это могут быть названия с ошибками или альтернативные названия того же жанра.

Вы увидите следующие неявные дубликаты:

* *hip*,
* *hop*,
* *hip-hop*.

Чтобы очистить от них таблицу используйте метод replace() с двумя аргументами: списком строк-дубликатов (включащий *hip*, *hop* и *hip-hop*) и строкой с правильным значением. Вам нужно исправить колонку genre в таблице df: заменить каждое значение из списка дубликатов на верное. Вместо hip, hop и hip-hop в таблице должно быть значение hiphop:

df['genre']**.**unique()

df['genre']**=**df['genre']**.**replace(['hip', 'hop', 'hip-hop'], 'hiphop')

**Задание 17**

Проверьте, что заменили неправильные названия:

* hip
* hop
* hip-hop

Выведите отсортированный список уникальных значений столбца genre:

sorted\_df **=** df['genre']**.**sort\_values()

sorted\_df**.**unique()

**Выводы**

Предобработка обнаружила три проблемы в данных:

* нарушения в стиле заголовков,
* пропущенные значения,
* дубликаты — явные и неявные.

Вы исправили заголовки, чтобы упростить работу с таблицей. Без дубликатов исследование станет более точным.

Пропущенные значения вы заменили на 'unknown'. Ещё предстоит увидеть, не повредят ли исследованию пропуски в колонке genre.

Теперь можно перейти к проверке гипотез.

## Проверка гипотез

### Сравнение поведения пользователей двух столиц

Первая гипотеза утверждает, что пользователи по-разному слушают музыку в Москве и Санкт-Петербурге. Проверьте это предположение по данным о трёх днях недели — понедельнике, среде и пятнице. Для этого:

* Разделите пользователей Москвы и Санкт-Петербурга
* Сравните, сколько треков послушала каждая группа пользователей в понедельник, среду и пятницу.

**Задание 18**

Для тренировки сначала выполните каждый из расчётов по отдельности.

Оцените активность пользователей в каждом городе. Сгруппируйте данные по городу и посчитайте прослушивания в каждой группе.

df**.**groupby('city')['genre']**.**count()

В Москве прослушиваний больше, чем в Петербурге. Из этого не следует, что московские пользователи чаще слушают музыку. Просто самих пользователей в Москве больше.

**Задание 19**

Теперь сгруппируйте данные по дню недели и подсчитайте прослушивания в понедельник, среду и пятницу. Учтите, что в данных есть информация только о прослушиваниях только за эти дни.

df**.**groupby('day')['genre']**.**count()

В среднем пользователи из двух городов менее активны по средам. Но картина может измениться, если рассмотреть каждый город в отдельности.

**Задание 20**

Вы видели, как работает группировка по городу и по дням недели. Теперь напишите функцию, которая объединит два эти расчёта.

Создайте функцию number\_tracks(), которая посчитает прослушивания для заданного дня и города. Ей понадобятся два параметра:

* день недели,
* название города.

В функции сохраните в переменную строки исходной таблицы, у которых значение:

* в колонке day равно параметру day,
* в колонке city равно параметру city.

Для этого примените последовательную фильтрацию с логической индексацией (или сложные логические выражения в одну строку, если вы уже знакомы с ними).

Затем посчитайте значения в столбце user\_id получившейся таблицы. Результат сохраните в новую переменную. Верните эту переменную из функции.

**def** number\_tracks(day, city):

track\_list**=**df[(df['day']**==**day)**&**(df['city']**==**city)]

track\_list\_count**=**track\_list['user\_id']**.**count()

**return** track\_list\_count

**Задание 21**

Вызовите number\_tracks() шесть раз, меняя значение параметров — так, чтобы получить данные для каждого города в каждый из трёх дней.

number\_tracks('Monday', 'Moscow')

number\_tracks('Monday', 'Saint-Petersburg')

number\_tracks('Wednesday', 'Moscow')

number\_tracks('Wednesday', 'Saint-Petersburg')

number\_tracks('Friday', 'Moscow')

number\_tracks('Friday', 'Saint-Petersburg')

**Задание 22**

Создайте c помощью конструктора pd.DataFrame таблицу, где

* названия колонок — ['city', 'monday', 'wednesday', 'friday'];
* данные — результаты, которые вы получили с помощью number\_tracks.

info**=**pd**.**DataFrame(data**=**[['Moscow', 15347, 10865, 15680], ['Saint-Petersburg', 5519, 6913, 5802]],columns**=**['city','monday','wednesday','friday'])

**Выводы**

Данные показывают разницу поведения пользователей:

* В Москве пик прослушиваний приходится на понедельник и пятницу, а в среду заметен спад.
* В Петербурге, наоборот, больше слушают музыку по средам. Активность в понедельник и пятницу здесь почти в равной мере уступает среде.

Значит, данные говорят в пользу первой гипотезы.

### Музыка в начале и в конце недели

Согласно второй гипотезе, утром в понедельник в Москве преобладают одни жанры, а в Петербурге — другие. Так же и вечером пятницы преобладают разные жанры — в зависимости от города.

**Задание 23**

Сохраните таблицы с данными в две переменные:

* по Москве — в moscow\_general;
* по Санкт-Петербургу — в spb\_general.

moscow\_general **=** df[df['city'] **==** 'Moscow']

spb\_general **=** df[df['city'] **==** 'Saint-Petersburg']

**Задание 24**

Создайте функцию genre\_weekday() с четырьмя параметрами:

* таблица (датафрейм) с данными,
* день недели,
* начальная временная метка в формате 'hh:mm',
* последняя временная метка в формате 'hh:mm'.

Функция должна вернуть информацию о топ-10 жанров тех треков, которые прослушивали в указанный день, в промежутке между двумя отметками времени.

**def** genre\_weekday(df, day, time1, time2):

*# последовательная фильтрация*

*# оставляем в genre\_df только те строки df, у которых день равен day*

genre\_df **=** df[df['day'] **==** day]

*# оставляем в genre\_df только те строки genre\_df, у которых время меньше time2*

genre\_df **=** genre\_df[genre\_df['time'] **<** time2]

*# оставляем в genre\_df только те строки genre\_df, у которых время больше time1*

genre\_df **=** genre\_df[genre\_df['time'] **>** time1]

*# сгруппируем отфильтрованный датафрейм по столбцу с названиями жанров, возьмём столбец genre и посчитаем кол-во строк для каждого жанра методом count()*

genre\_df\_grouped **=** genre\_df**.**groupby('genre')['genre']**.**count()

*# отсортируем результат по убыванию (чтобы в начале Series оказались самые популярные жанры)*

genre\_df\_sorted **=** genre\_df\_grouped**.**sort\_values(ascending**=False**)

*# вернём Series с 10 самыми популярными жанрами в указанный отрезок времени заданного дня*

**return** genre\_df\_sorted[:10]

**Задание 25**

Cравните результаты функции genre\_weekday() для Москвы и Санкт-Петербурга в понедельник утром (с 7:00 до 11:00) и в пятницу вечером (с 17:00 до 23:00):

genre\_weekday(moscow\_general, 'Monday', '07:00', '11:00')

genre\_weekday(spb\_general, 'Monday', '07:00', '11:00')

genre\_weekday(moscow\_general,'Friday','17:00', '23:00')

genre\_weekday(spb\_general, 'Friday','17:00', '23:00')

**Выводы**

Если сравнить топ-10 жанров в понедельник утром, можно сделать такие выводы:

1. В Москве и Петербурге слушают похожую музыку. Единственное отличие — в московский рейтинг вошёл жанр “world”, а в петербургский — джаз и классика.
2. В Москве пропущенных значений оказалось так много, что значение 'unknown' заняло десятое место среди самых популярных жанров. Значит, пропущенные значения занимают существенную долю в данных и угрожают достоверности исследования.

Вечер пятницы не меняет эту картину. Некоторые жанры поднимаются немного выше, другие спускаются, но в целом топ-10 остаётся тем же самым.

Таким образом, вторая гипотеза подтвердилась лишь частично:

* Пользователи слушают похожую музыку в начале недели и в конце.
* Разница между Москвой и Петербургом не слишком выражена. В Москве чаще слушают русскую популярную музыку, в Петербурге — джаз.

Однако пропуски в данных ставят под сомнение этот результат. В Москве их так много, что рейтинг топ-10 мог бы выглядеть иначе, если бы не утерянные данные о жанрах.

### Жанровые предпочтения в Москве и Петербурге

Гипотеза: Петербург — столица рэпа, музыку этого жанра там слушают чаще, чем в Москве. А Москва — город контрастов, в котором, тем не менее, преобладает поп-музыка.

**Задание 26**

Сгруппируйте таблицу moscow\_general по жанру и посчитайте прослушивания треков каждого жанра методом count(). Затем отсортируйте результат в порядке убывания и сохраните его в таблице moscow\_genres.

moscow\_genres**=**moscow\_general**.**groupby('genre')['genre']**.**count()**.**sort\_values(ascending **=** **False**)

**Задание 27**

Выведите на экран первые десять строк moscow\_genres:

moscow\_genres**.**head(10)

**Задание 28**

Теперь повторите то же и для Петербурга.

Сгруппируйте таблицу spb\_general по жанру. Посчитайте прослушивания треков каждого жанра. Результат отсортируйте в порядке убывания и сохраните в таблице spb\_genres:

spb\_genres**=**spb\_general**.**groupby('genre')['genre']**.**count()**.**sort\_values(ascending **=** **False**)

**Задание 29**

Выведите на экран первые десять строк spb\_genres:

spb\_genres**.**head(10)

**Выводы**

Гипотеза частично подтвердилась:

* Поп-музыка — самый популярный жанр в Москве, как и предполагала гипотеза. Более того, в топ-10 жанров встречается близкий жанр — русская популярная музыка.
* Вопреки ожиданиям, рэп одинаково популярен в Москве и Петербурге.

## Итоги исследования

Вы проверили три гипотезы и установили:

1. День недели по-разному влияет на активность пользователей в Москве и Петербурге.

Первая гипотеза полностью подтвердилась.

1. Музыкальные предпочтения не сильно меняются в течение недели — будь то Москва или Петербург. Небольшие различия заметны в начале недели, по понедельникам:

* в Москве слушают музыку жанра “world”,
* в Петербурге — джаз и классику.

Таким образом, вторая гипотеза подтвердилась лишь отчасти. Этот результат мог оказаться иным, если бы не пропуски в данных.

1. Во вкусах пользователей Москвы и Петербурга больше общего чем различий. Вопреки ожиданиям, предпочтения жанров в Петербурге напоминают московские.

Третья гипотеза не подтвердилась. Если различия в предпочтениях и существуют, на основной массе пользователей они незаметны.

**На практике исследования содержат проверки статистических гипотез.** Из данных одного сервиса не всегда можно сделать вывод о всех жителях города. Проверки статистических гипотез покажут, насколько они достоверны, исходя из имеющихся данных. С методами проверок гипотез вы ещё познакомитесь в следующих темах.