Вы работаете аналитиком данных в онлайн-кинотеатре *СмотримВместе*.
Сегодня работаем с **ивентами** (event - *событие*). Ивенты - это записи о клиентских событиях, таких как включение плеера с видео или заход на сайт онлайн-кинотеатра. Одной из важнейших задач аналитики является изучение и правка ошибочных ивентов.

Ваша задача - изучить датасет на наличие ошибок. В рамках данного технического задания вам необходимо взять только те строки, у которых колонка screen_type принимает значение "player".

```
import pandas as pd
```

```
df = pd.read_excel('/Users/megatron/Documents/study/Python/Техническое задание Python 1/event_list.xlsx')
```

```
# Отбираем строки, в которых колонка screen_type принимает значение "player" и перезаписываем датафрейм.
df = df[df['screen_type'] == 'player'].sort_values('server_time', ascending=True)
df.head()
```

	account_id	server_time	screen_type	action_id	device_type	user_browser
3828	102682	2021-10-01 00:06:08	player	playback_stop	desktop_web	Chrome
5294	103052	2021-10-01 00:06:40	player	click	desktop_web	Safari
5295	103052	2021-10-01 00:06:53	player	playback_pause	desktop_web	Safari
3829	102682	2021-10-01 00:10:12	player	playback_stop	desktop_web	Chrome
3611	102660	2021-10-01 00:10:22	player	click	desktop_web	Chrome

Задача 1

Выделите 10 самых активных пользователей (тех, что сделали больше всего действий за данный период времени).
Каким браузером чаще всего пользовались эти 10 пользователей?
Какое действие занимает наибольший процент среди всех действий этих 10 пользователей?

```
#Выделите 10 самых активных пользователей (тех, что сделали больше всего действий за данный период времени).
df_top_users = df.groupby('account_id', as_index=False).agg(cnt_action_id = ('action_id', 'count'))\
.sort_values('cnt_action_id', ascending=False).head(10)
df_top_users
```

	account_id	cnt_action_id
36	102598	1004
145	103398	386
37	102605	288
76	102904	142
108	103149	141
120	103219	128
199	103804	128
10	102405	116
74	102883	107
138	103364	102

```
#Каким браузером чаще всего пользовались эти 10 пользователей?
df_browser = df[df['account_id'].isin(list(df_top_users['account_id']))].groupby('user_browser', as_index=False)\
.agg(cnt_browser = ('user_browser', 'count')).sort_values('cnt_browser', ascending=False)
df_browser
# Браузером Yandex
```

	user_browser	cnt_browser
4	Yandex	1004
0	Chrome	968
2	Safari	301
3	Samsung Browser	141
1	Edge	128

```
#Какое действие занимает наибольший процент среди всех действий этих 10 пользователей?
df_action = df[df['account_id'].isin(list(df_top_users['account_id']))].groupby('action_id', as_index=False)\
.agg(cnt_action = ('action_id', 'count')).sort_values('cnt_action', ascending=False)
df_action['rate_action'] = round(df_action['cnt_action'] / sum(df_action['cnt_action']) * 100, 2)
df_action[df_action['rate_action'] == max(df_action['rate_action'])]
# Действие 'click' - клик по плееру
```

	action_id	cnt_action	rate_action
0	click	1595	62.75

Задача 2

Проверьте данные на состоятельность:

1. Есть ли клиенты, которые снимали видео с паузы хотя бы раз, но при этом не ставили видео на паузу ни разу?
2. Есть ли клиенты, которые ставили видео на паузу хотя бы раз, но при этом не включали проигрыватель?

С каких устройств и браузеров заходят пользователи, у которых встречаются подобные аномалии?

Подсказка
Воспользуйтесь методом серии unique(), чтобы для каждого действия (ивента) создать списки с уникальными пользователями, которые совершали эти действия.
Посмотрите на соответствующие исключения списков друг из друга, чтобы определить пользователей с проблемной последовательностью событий.

```
#Задача 2.1
# клиенты, которые снимали видео с паузы
users_unpause = set(pd.Series(df[df['action_id'] == 'playback_unpause']['account_id'].unique()))
# все клиенты
users_all = set(pd.Series(df['account_id'].unique()))
# клиенты, которые включали проигрыватель
users_pause = set(pd.Series(df[df['action_id'] == 'playback_pause']['account_id'].unique()))
# вычислим клиенты, которые не ставили видео на паузу
users_non_pause = users_all - users_pause
# клиенты, которые не ставили видео на паузу - решение через пересечение множеств.
users_non_pause_and__unpause = users_non_pause & users_unpause
users_non_pause_and__unpause
# Наблюдается три клиента с id 102345, 102918, 103050, которые сняли видео с паузы, хотя при это на паузу не ставили
```

```
{102345, 102918, 103050}
```

```
#Задача 2.2
# клиенты, которые ставили видео на паузу
users_pause = set(pd.Series(df[df['action_id'] == 'playback_pause']['account_id'].unique()))
# все клиенты
users_all = set(pd.Series(df['account_id'].unique()))
# клиенты, которые включали проигрыватель
users_start = set(pd.Series(df[df['action_id'] == 'playback_start']['account_id'].unique()))
# клиенты, которые не включали проигрыватель
users_no_start = users_all - users_start
# клиенты, которые ставили видео на паузу, но при этом не включали проигрыватель
users_no_start_pause = users_pause & users_no_start
users_no_start_pause
# Наблюдается два клиента с id 102669, 103052, которые ставили видео на паузу, но при этом не включали проигрыватель
```

```
{102669, 103052}
```

```
# Задача 2.3
# С каких устройств и браузеров заходят пользователи, у которых встречаются подобные аномалии?
# делаю общее множество проблемных клиентов
users_anomaly = users_non_pause_and__unpause.union(users_no_start_pause)
# выводим информацию по проблемным клиентам
df[df['account_id'].isin(users_anomaly)].groupby(['device_type', 'user_browser'], as_index=False)\
.agg( cnt_action_anomaly = ('account_id', 'count'))
# видим, что все аномалии наблюдаются при пользовании ресурсом только с desktop pc и в браузерах Chrome и Safari
```

	device_type	user_browser	cnt_action_anomaly
0	desktop_web	Chrome	32
1	desktop_web	Safari	12

Задача 3

1. Изучите пользователей, у которых есть только одно событие. Какой вид события встречается чаще всего? Какие из встречающихся событий вы бы посчитали ошибочными?
2. Изучите пользователей, у которых есть только два события. Какие из их последовательностей событий вы бы посчитали ошибочными?

```
# Задача 3.1
# модифицируем таблицу для вывода списка пользователей с кол-вом событий и типом событий
df_grupped = df.groupby('account_id', as_index=False).agg(cnt_action_id = ('action_id', 'count'))\
.sort_values('cnt_action_id')
# создаем список с  пользователями с одним событием
users_cnt_action_1 = list(df_grupped[df_grupped['cnt_action_id'] == 1]['account_id'])
# создаем предварительную таблицу с клиентом, датой и типом  события, среди клиентов с всего одним событием для будущего анализа
df1 = df[df['account_id'].isin(users_cnt_action_1)]
df1
```

	account_id	server_time	screen_type	action_id	device_type	user_browser
4440	102865	2021-10-01 05:55:31	player	playback_stop	desktop_web	Safari
9920	103813	2021-10-01 07:17:19	player	playback_stop	desktop_web	Chrome
900	102461	2021-10-01 07:22:42	player	click	desktop_web	Safari
4439	102853	2021-10-01 08:55:10	player	click	desktop_web	Yandex
4395	102839	2021-10-01 15:32:53	player	playback_start	desktop_web	Opera
1444	102520	2021-10-01 19:51:25	player	playback_stop	desktop_web	Chrome
5843	103140	2021-10-01 21:02:42	player	playback_start	mobile_web	Mobile Safari

```
# проверяем тип и количесво событий среди пользователей с одним событием
grupped1_users_cnt_action_1 = df[df['account_id'].isin(users_cnt_action_1)].groupby('action_id').agg(cnt_action_1 = ('action_id', 'count'))
grupped1_users_cnt_action_1
```

	cnt_action_1
action_id	
click	2
playback_start	2
playback_stop	3

Вывод по Задаче 3.1.

Выборка была сделана по типу страницы "player", соответственно в анализ попадают только логи, связанные с действиями на странице с плеером.

1. Два действия 'click': Речь идет о простом клике по плееру, за которым не следуют другие действия. Это может означать, что пользователь либо закрыл страницу, либо начал взаимодействовать с другими элементами, не относящимися к плееру (и, следовательно, не попавшими в текущую выборку). Такие события не выглядят аномальными и, скорее всего, соответствуют ожидаемому пользовательскому поведению.

2. Два действия 'playback_start': Первое включение проигрывателя произошло через 15,5 часов после начала логирования, второе — через 21 час. Такие интервалы без каких-либо промежуточных действий выглядят подозрительно. Маловероятно, что пользователь находится на странице так долго без активности и затем запускает воспроизведение. Это может свидетельствовать о сбоях в логировании или ошибке в данных.

3. Три действия 'playback_stop': Остановка воспроизведения зафиксирована через 6, 7 и 20 часов после начала сессии. Учитывая отсутствие других взаимодействий (например, пауза, перемотка), а также то, что видеоконтент такой продолжительности практически отсутствует, эти действия также выглядят недостоверными.

Вывод: С высокой вероятностью, данные либо содержат ошибки, либо логирование действий настроено некорректно. Особенно это касается аномально длительных интервалов без активности, что нехарактерно для взаимодействия с видеоплеером.

```
# Задача 3.2
# создаем список с  пользователями с двумя событиями
users_cnt_action_2 = list(df_grupped[df_grupped['cnt_action_id'] == 2]['account_id'])
# смотрим последовательность событий у пользователей с двумя событиями
grupped_users_cnt_action_2 = df[df['account_id'].isin(users_cnt_action_2)][['account_id', 'server_time', 'action_id']]\
.sort_values(['account_id', 'server_time'])
grupped_users_cnt_action_2
```

	account_id	server_time	action_id
1507	102537	2021-10-01 18:48:43	playback_start
1508	102537	2021-10-01 18:50:18	playback_stop
3828	102682	2021-10-01 00:06:08	playback_stop
3829	102682	2021-10-01 00:10:12	playback_stop
4839	102943	2021-10-01 19:48:09	playback_start
4840	102943	2021-10-01 19:48:12	playback_pause
5294	103052	2021-10-01 00:06:40	click
5295	103052	2021-10-01 00:06:53	playback_pause
8076	103446	2021-10-01 18:43:42	playback_start
8077	103446	2021-10-01 18:44:14	playback_stop
8635	103566	2021-10-01 10:12:48	playback_start
8636	103566	2021-10-01 10:12:56	click
9382	103724	2021-10-01 13:24:46	playback_start
9383	103724	2021-10-01 13:24:52	playback_stop

Вывод по Задаче 3.2

Оценка последовательностей пользовательских действий

Нормальные последовательности:

playback_start → playback_stop — пользователь запускает воспроизведение и останавливает его. playback_start → playback_pause — пользователь ставит воспроизведение на паузу. playback_start → click — пользователь запускает воспроизведение, затем кликает по плееру, возможно, выбирая другое видео. Все эти последовательности соответствуют нормальному пользовательскому поведению в онлайн-кинотеатре.

Подозрительные случаи:

1. Повторное завершение воспроизведения: 102682 2021-10-01 00:06:08 playback_stop 102682 2021-10-01 00:10:12 playback_stop Два действия playback_stop с разницей в 4 минуты от одного клиента выглядят нелогично. Невозможно дважды завершить воспроизведение, не запустив его повторно между этими событиями. Вероятные объяснения: Пропущено событие playback_start между двумя stop. Первое событие ошибочно записано как stop, хотя на самом деле могло быть другим (например, pause или click).

2. Клик до паузы без запуска воспроизведения: 103052 2021-10-01 00:06:40 click 103052 2021-10-01 00:06:53 playback_pause Здесь пользователь сначала кликнул по плееру, а затем поставил видео на паузу. Отсутствие события playback_start между этими действиями вызывает сомнение. Возможные причины: Событие playback_start было пропущено при логировании. Событие click фактически обозначает другой тип взаимодействия, не соответствующий текущей интерпретации. Общий вывод:

Обнаруженные последовательности подтверждают наличие потенциальных проблем в логировании: либо пропущенные события, либо неверно определённые типы действий. Для более точной диагностики стоит проверить настройки логирования и провести выборочную проверку сессий с полным набором событий.