**Python**

**Описание данных**

**Таблица «*transactions.xlsx*»** - содержит историю транзакций клиентов банка за один год и три месяца, ссылка на таблицу для выполнения задания по python: <https://cloud.mail.ru/public/SmEX/wwYjGY3Qs>

Формат данных:

customer\_id,tr\_datetime,mcc\_code,tr\_type,amount,term\_id

111111,15 01:40:52,1111,1000,-5224,111111

111112,15 15:18:32,3333,2000,-100,11122233

...

Описание полей:

* customer\_id — идентификатор клиента;
* tr\_datetime — день и время совершения транзакции (дни нумеруются с начала данных);
* mcc\_code — mcc-код транзакции;
* tr\_type — тип транзакции;
* amount — сумма транзакции в условных единицах со знаком; + — начисление средств клиенту (приходная транзакция), - — списание средств (расходная транзакция);
* term\_id — идентификатор терминала;

**Таблица «*gender\_train.csv*»** - содержит информацию по полу для части клиентов, для которых он известен. Для остальных клиентов пол неизвестен.

Формат данных:

customer\_id,gender

111111,0

111112,1

...

Описание полей:

* customer\_id — идентификатор клиента;
* gender — пол клиента;

**Таблица «*tr\_mcc\_codes.csv*»** - содержит описание mcc-кодов транзакций.

Формат данных:

mcc\_code;mcc\_description

1000;словесное описание mcc-кода 1000

2000;словесное описание mcc-кода 2000

...

Описание полей:

* mcc\_code – mcc-код транзакции;
* mcc\_description — описание mcc-кода транзакции.

**Таблица «*tr\_types.csv*»** - содержит описание типов транзакций.

Формат данных:

tr\_type;tr\_description

1000;словесное описание типа транзакции 1000

2000;словесное описание типа транзакции 2000

...

Описание полей:

* tr\_type – тип транзакции;
* tr\_description — описание типа транзакции;

***Задача №1***

* Считать файл transactions только первые 1000000 строк.
* заполнить Nan значение знаком '-'
* изменить название столбца term\_id на term\_id\_1
* просуммировать значения в столбце amount, вывести минимальное и максимальное значение, просуммировать значения больше 0
* в столбце tr\_datetime отделить время [0 10:23:26, 1 10:19:29] -> [10:23:26, 10:19:29]
* в столбце tr\_datetime имеются ошибочные данные в секундах, заменить 60 на 59 [06:27:60] -> [06:27:59]. Сделать двумя способами, в одном из них использовать try - except
* отфильтровать датафрейм по столбцу mcc\_code > 6000 и столбцу tr\_datetime позже 10:00:00, посчитать среднее значение столбца amount
* вывести дф в обратном порядке, вывести каждую 10 строку дф.
* сгруппировать по customer\_id, создать новый дф из строк в которых среднее значение больше 0.
* добавить в новый дф столбец из рандомных чисел от 0 до 100
* используя регулярные выражения отфильтровать по столбцу customer\_id строки в которых содержатся цифры 3, 6, 8.

***Задача №2***

* В датафрейме transactions задайте столбец customer\_id в качестве индекса.
* Выделите клиента с максимальной суммой транзакции (то есть с максимальным приходом на карту). *(\*)*
* Найдите у него наиболее часто встречающийся модуль суммы приходов/расходов. *(\*\*)*
* Выберите все верные пункты:
  1. 1122957.89
  2. 15721.41
  3. 22459.16
  4. 13475494.63
  5. 107407.78
  6. 65019.26

*Пояснения:*

*(\*)* Если у клиента были транзакции [-10000, 10, 0, -10], то максимумом будет являться значение 10.  
*(\*\*)* Если у клиента были транзакции [-10000, 10, 0, -10], то наиболее встречающийся модуль суммы транзакций равен 10, и встретился он 2 раза.

***Задача №3***

* Соедините transactions с всеми остальными таблицами (tr\_mcc\_codes, tr\_types, gender\_train). Причём с gender\_train необходимо смёрджиться с помощью left join, а с оставшимися датафреймами - через inner. После получения результата таблицы gender\_train, tr\_types, tr\_mcc\_codes можно удалить. В результате соединения датафреймов должно получиться 999584 строки.
* Выделите из поля tr\_datetime относительный день tr\_day (первое число до точного времени). *(\*)*
* Отфильтруйте строки таким образом, чтобы оставить только те транзакции, у которых в соответствующий относительный день tr\_day количество уникальных MCC кодов при транзакциях было больше 75 (можно воспользоваться функцией nunique())
* Сгруппируйте полученный отфильтрованный датафрейм по MCC коду и полу, после чего, пронализировав результат, выберите верные варианты ответов ниже *(\*\*)*:
  1. gender == 0 - женщины, gender == 1 – мужчины
  2. gender == 1 - женщины, gender == 0 – мужчины
  3. Абсолютное значение медианы с типом "Флористика" (расходов/приходов) у мужчин выше той же медианы у женщин
  4. Абсолютное значение медианы женских трат (расходов/приходов) на ценные бумаги выше мужских
  5. Абсолютное значение медианы женских трат (расходов/приходов) в категории "Бары, коктейль-бары, дискотеки, ночные клубы и таверны — места продажи алкогольных напитков" ниже мужских

*Пояснения:*

*(\*)* Для того, чтобы выделить всё, что стоит до первого пробела, можно использовать строковые методы для датафрейма - .str.split(), например. Либо же реализовывать логику выделения подстроки с помощью метода apply.  
*(\*\*)* Понять, какой класс к какому типу транзакций (мужские/женские) относится можно, если поизучать типичные для мужчин/женщин категории и сравнить средние/медианы расходов и/или приходов в них.