ПРОЕКТ №5
СЛУЖБА ТАКСИ

ИТОГОВЫЙ ПРОЕКТ

https://github.com/renasafetysea/final_project

задача

Необходимо, используя таблицу поездок для каждого дня, рассчитать процент поездок по количеству человек в машине (без пассажиров, 1, 2, 3, 4 и более пассажиров). По итогу должна получиться таблица (parquet)
Также добавить столбцы к предыдущим результатам с самой дорогой и самой дешевой поездкой для каждой группы.

Дополнительно: также провести аналитику и построить график на тему «Как пройденное расстояние и количество пассажиров влияет на чаевые» в любом удобном инструменте.

Входные данные – таблица (количество строк - 6 405 008) состоящая из поездок такси в Нью-Йорке. Сырые данные представляют собой файл весом около 566,1 мб.
ссылка на данные : https://disk.yandex.ru/d/DKeoopbGH1Ttuw

РЕАЛИЗАЦИЯ

использован следующий стек :

- В качестве СУБД использован Postgres, поскольку таблица не требует высокой скорости чтения для аналитики и при расширении функционала БД службы такси она скорее будет ориентирована на запись, причем небольшого количества строк за раз.
- Язык скриптов - python, библиотеки psycopg2 для подключения к БД и обмена данными и pandas для обработки данных датафреймами, а также выгрузки данных из БД в формате parquet.
- Проект завернут в docker-compose.

В проекте необходимо предварительно обработать данные для core-слоя и поместить их в нужную директорию.

- Требуеся предварительная подготовка файла с очищенными данными. Необходимо запустить скрипт dataframe_processing и поместить полученный файл в папку init_db/data. Здесь производится преобразование формата, проверка на существование всех необходимых значений строки, их адекватность.

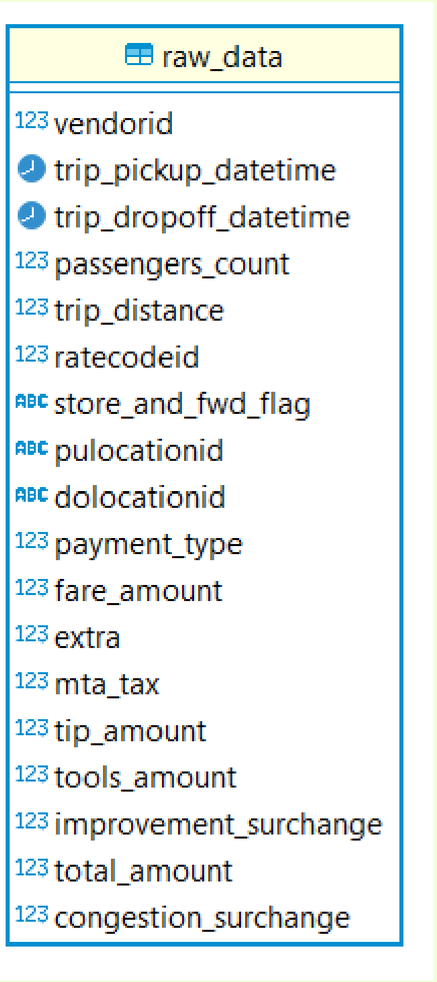
- При запуске docker-compose происходит загрузка данных в БД Postgres и создние витрины данных, а также ее выгрузка в формате parquet в отдельную директорию внутри контейнера.

Работа с данными

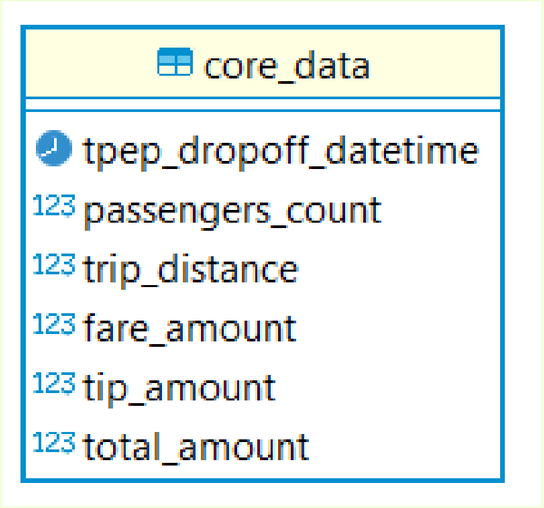
структура проекта

```
├── version1
│   │   docker-compose.yml
│   │
│   ├── app
│   │   │   Dockerfile
│   │   │   requirements.txt
│   │   │
│   │   └── scripts
│   │       │   config.py
│   │       │   dataframe_processing.py
│   │       │   data_analysis.py
│   │       │   data_load.py
│   │       │   main.py
│   │       │   queries.py
│   │       │   scatter_plot_1-1.png
│   │       │   scatter_plot_1-2.png
│   │       │
│   │       └── __pycache__
│   │               config.cpython-311.pyc
│   │
│   ├── figs
│   │       data_mart_fig.PNG
│   │       er.PNG
│   │       structure.PNG
│   │
│   ├── init_db
│   │   └── data
│   │           core_table_data.csv
│   │           yellow_tripdata_2020-01.csv
│   │
│   └── results
│           final_datamart.parquet
```

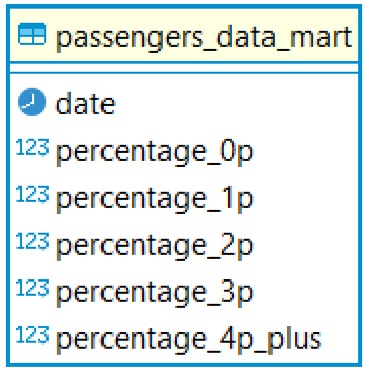
RAW

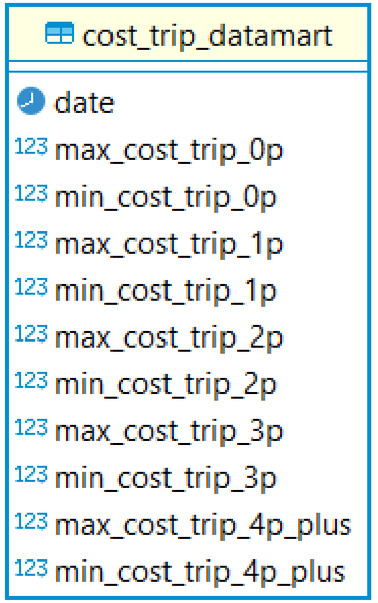


CORE



MART





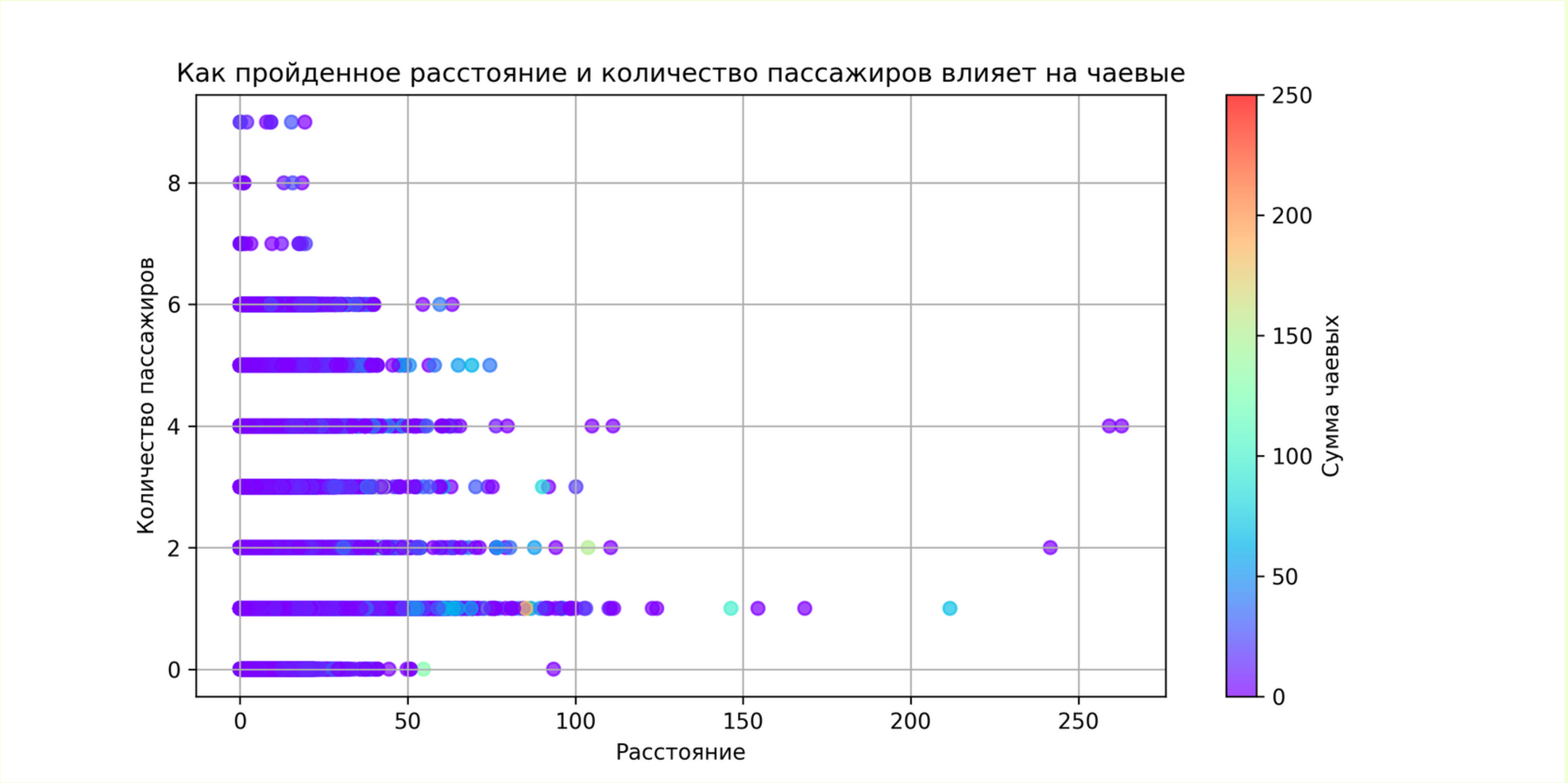
Витрина данных содержит информацию о процентном соотношении поездок такси с различным количеством пассажиров по датам:

	date	percentage_0p	percentage_1p	percentage_2p	percentage_3p	percentage_4p_plus
1	2008-12-31 00:00:00.000	[NULL]	1	[NULL]	[NULL]	[NULL]
2	2009-01-01 00:00:00.000	[NULL]	0,76	0,04	0,08	0,12
3	2019-12-31 00:00:00.000	[NULL]	0,587	0,143	0,048	0,222
4	2020-01-01 00:00:00.000	0,015	0,634	0,195	0,057	0,098
5	2020-01-02 00:00:00.000	0,017	0,687	0,16	0,048	0,088
6	2020-01-03 00:00:00.000	0,018	0,688	0,163	0,046	0,085
7	2020-01-04 00:00:00.000	0,017	0,656	0,184	0,051	0,091
8	2020-01-05 00:00:00.000	0,017	0,675	0,174	0,048	0,085
9	2020-01-06 00:00:00.000	0,018	0,737	0,136	0,037	0,072
10	2020-01-07 00:00:00.000	0,017	0,739	0,137	0,035	0,072
11	2020-01-08 00:00:00.000	0,018	0,739	0,136	0,035	0,071
12	2020-01-09 00:00:00.000	0,018	0,74	0,136	0,035	0,071
13	2020-01-10 00:00:00.000	0,018	0,719	0,149	0,039	0,074
14	2020-01-11 00:00:00.000	0,017	0,673	0,18	0,047	0,083
15	2020-01-12 00:00:00.000	0,016	0,687	0,172	0,046	0,08
16	2020-01-13 00:00:00.000	0,018	0,747	0,131	0,035	0,069
17	2020-01-14 00:00:00.000	0,018	0,744	0,132	0,035	0,071
18	2020-01-15 00:00:00.000	0,018	0,744	0,134	0,034	0,071
19	2020-01-16 00:00:00.000	0,018	0,74	0,136	0,035	0,071
20	2020-01-17 00:00:00.000	0,018	0,722	0,147	0,04	0,074
21	2020-01-18 00:00:00.000	0,017	0,669	0,179	0,047	0,087
22	2020-01-19 00:00:00.000	0,018	0,67	0,177	0,049	0,085
23	2020-01-20 00:00:00.000	0,018	0,708	0,155	0,043	0,076

Также составлена витрина для получения информации о минимальной и максимальной стоимости поездки за день:

1	2020-07-31 00:00:00.000	[NULL]	[NULL]	9,3	9,3	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
2	2020-02-02 00:00:00.000	[NULL]	[NULL]	49,92	49,92	9,8	9,8	9,3	9,3	[NULL]
3	2020-07-10 00:00:00.000	[NULL]	[NULL]	20,16	9,36	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
4	2020-01-03 00:00:00.000	187,42	3,3	358,92	0,31	370,3	0,31	260,3	3,8	271,3
5	2020-01-09 00:00:00.000	114,75	3,3	620,3	0,31	272,67	0,31	230,3	4,3	333,04
6	2020-02-01 00:00:00.000	53	10,35	284,79	5,3	200,04	8,3	59,42	7,8	293,16
7	2020-05-28 00:00:00.000	[NULL]	[NULL]	18,96	18,96	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
8	2020-01-30 00:00:00.000	137,35	3,3	349,1	0,31	243,49	3,3	212,9	3,3	245,15
9	2020-05-29 00:00:00.000	[NULL]	[NULL]	9,8	9,8	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
10	2020-01-11 00:00:00.000	143,45	4,3	416,15	0,31	228,02	0,31	174,8	0,31	424,29
11	2020-03-25 00:00:00.000	[NULL]	[NULL]	11,3	11,3	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
12	2021-01-02 00:00:00.000	[NULL]	[NULL]	17,16	14,8	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
13	2020-03-16 00:00:00.000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	8,76	8,76	[NULL]
14	2020-03-04 00:00:00.000	[NULL]	[NULL]	15,34	15,34	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
15	2020-01-04 00:00:00.000	152,54	0,31	965,8	0,31	481,3	0,31	158,8	3,3	577,8
16	2020-01-26 00:00:00.000	186,35	3,3	357,42	0,31	345,34	0,31	173,16	3,3	217,86
17	2020-01-12 00:00:00.000	161,8	3,3	380,53	0,31	431,6	3,3	331,42	0,31	658,35
18	2020-01-05 00:00:00.000	435,42	3,3	1 242,3	0,31	375,3	3,3	333,34	0,31	255,42
19	2020-01-19 00:00:00.000	160,92	4,3	464,8	1,3	407,8	3,3	499,56	0,31	222,36
20	2020-04-16 00:00:00.000	[NULL]	[NULL]	10,56	10,56	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
21	2020-01-13 00:00:00.000	276,29	3,3	534,12	0,31	400,3	3,3	178,92	3,8	300,3
22	2020-01-28 00:00:00.000	143,05	3,3	1 722,3	0,31	298,12	0,31	156,36	3,8	294,8

Скрипт data_analysis (не инициализируется docker-compose) содержит программу для построения графика рассеяния, отвечающего на вопрос "Как количество пассажиров и дальность поездки влияют на чаевые?". Получен график:



ВЫВОДЫ

1	Получен работоспособный проект, инициализирующий БД службы такси, состоящую из трех слоев данных. Написаны скрипты для обработки и очистки сырых данных и получения витрины.
2	При ответе на дополнительный вопрос сделан вывод, что чаще всего таксист получает большую сумму чаевых при поездке на расстояние 50-100, если в машине находятся 1, 2 или 5 человек.
3	Проект можно доработать улучшив качество очистки данных при инициализации core-слоя и обработки данных при аналитике, а также оптимизировав его с точки зрения затрат памяти.