Университет ИТМО

# Практическая работа №3

по дисциплине «Визуализация и моделирование»

**Автор:** Бичук Екатерина Дмитриевна
**Поток:** ВИМ 1.2
**Группа:** К3221
**Факультет:** ИКТ
**Преподаватель:** Чернышева А.В.

Санкт-Петербург, 2021 г.

Таблица 1: Информация хранящаяся в столбцах

| Название столбца в датасете | Данные, хранящиеся в столбце | Тип данных | Шкала | Проблема | Решение |
|---|---|---|---|---|---|
| id | Индивидуальный номер объявления | int | Номинальная | - | - |
| name | Название объявления | str | Номинальная | - | - |
| host_id | Индивидуальный номер арендодателя | int | Номинальная | - | - |
| host_name | Имя арендодателя | str | Номинальная | Текст неудобно использовать при построение модели, а у каждого арендодателя есть id | Удалить столбец с данными |
| neighbourhood_group | Боро( единица административного деления Нью-Йорка) | str | Номинальная | - | - |
| neighbourhood | Название района | str | Номинальная | - | - |
| latitude | Координаты широты | float | Номинальная | - | - |
| longitude | Координаты долготы | float | Номинальная | - | - |
| room_type | Тип жилья | str | Порядковая | Текст неудобно использовать при построении модели | Перевод в число |
| price | Цена за сутки в долларах | int | Относительная | - | - |
| minimum_nights | Минимальное количество ночей | int | Относительная | - | - |
| number_of_reviews | Количество оценок на объявлении | int | Относительная | - | - |
| last_review | Дата последней оценки | str | Номинальная | Даты хранятся как строки и есть значения nan | Перевод в дату и замена nan |
| reviews_per_month | Количество оценок в месяц | float | Относительная | Есть значения nan | Замена на другие значения |
| calculated_host_list | Количество объявлений у одного арендодателя | int | Относительная | Возможны выбросы | Удалить выбросы |
| availability_365 | Количество дней в году для съема жилья | int | Относительная | - | - |

В датасете хранятся данные об объявлениях о сдаче жилья в Нью-Йорке с сайта airbnb за 2019 год. Также информация об самом жилье и арендодателях.

Имя арендодателя не удобно использовать при построении модели и у каждого арендодателя есть свой id. Поэтому удалим столбец.

```
df_norm = df.drop(columns=["host_name"])
df_norm.head()
```

| | id | name | host_id | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | minimum_nights | number_of_reviews | last_revi |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2539 | Clean & quiet apt home by the park | 2787 | Brooklyn | Kensington | 40.64749 | -73.97237 | Private room | 149 | 1 | 9 | 2018-10- |
| 1 | 2595 | Skylit Midtown Castle | 2845 | Manhattan | Midtown | 40.75362 | -73.98377 | Entire home/apt | 225 | 1 | 45 | 2019-05- |
| 2 | 3647 | THE VILLAGE OF HARLEM....NEW YORK ! | 4632 | Manhattan | Harlem | 40.80902 | -73.94190 | Private room | 150 | 3 | 0 | N |
| 3 | 3831 | Cozy Entire Floor of Brownstone | 4869 | Brooklyn | Clinton Hill | 40.68514 | -73.95976 | Entire home/apt | 89 | 1 | 270 | 2019-07- |
| 4 | 5022 | Entire Apt: Spacious Studio/Loft by central park | 7192 | Manhattan | East Harlem | 40.79851 | -73.94399 | Entire home/apt | 80 | 10 | 9 | 2018-11- |

Рис. 1: Удаление столбца host_name

Столбец room_type содержит текст: Private room, Entire home/apt, Shared room, который неудобно использовать для визуализации. Заменим текст на числа Private room == 1, Entire home/apt == 2, а Shared room == 3.

```
df_norm["room_type"] = df_norm["room_type"].apply(norm_room_type)
df_norm.head()
```

| | id | name | host_id | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | minimum_nights | number_of_reviews | last_rev |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2539 | Clean & quiet apt home by the park | 2787 | Brooklyn | Kensington | 40.64749 | -73.97237 | 1 | 149 | 1 | 9 | 2018-10 |
| 1 | 2595 | Skylit Midtown Castle | 2845 | Manhattan | Midtown | 40.75362 | -73.98377 | 2 | 225 | 1 | 45 | 2019-05 |
| 2 | 3647 | THE VILLAGE OF HARLEM....NEW YORK ! | 4632 | Manhattan | Harlem | 40.80902 | -73.94190 | 1 | 150 | 3 | 0 | N |
| 3 | 3831 | Cozy Entire Floor of Brownstone | 4869 | Brooklyn | Clinton Hill | 40.68514 | -73.95976 | 2 | 89 | 1 | 270 | 2019-07 |
| 4 | 5022 | Entire Apt: Spacious Studio/Loft by central park | 7192 | Manhattan | East Harlem | 40.79851 | -73.94399 | 2 | 80 | 10 | 9 | 2018-11 |

```
df_norm["room_type"].unique()
```

```
array([1, 2, 3], dtype=int64)
```

Рис. 2: Перевод в число

Проверим столбцы neighbourhood_group и neighbourhood. В столбце neighbourhood_group содержится всего пять разных Боро, значит странных данных нет. Столбец neighbourhood

хранит больше уникальных значений. Посмотрим названия районов, которые встречаются несколько раз, чтобы выявить опечатки. В ходе недолгого анализа получилось, что все эти районы существуют.

```
from operator import itemgetter

region_dt = [(name, df["neighbourhood"].to_list().count(name))
                    for name in df["neighbourhood"].unique() ]
region_dt = sorted(region_dt, key=itemgetter(1))
region_dt
```

```
[('Woodrow', 1),
 ('Richmondtown', 1),
 ('Fort Wadsworth', 1),
 ('New Dorp', 1),
 ('Rossville', 1),
 ('Willowbrook', 1),
 ('Co-op City', 2),
 ('Lighthouse Hill', 2),
 ('West Farms', 2),
 ('Silver Lake', 2),
 ('Howland Hook', 2),
 ('Westerleigh', 2),
 ('Bay Terrace, Staten Island', 2),
 ('Graniteville', 3),
 ('Eltingville', 3),
 ('Neponsit', 3),
 ('Huguenot', 3),
 ('Breezy Point', 3),
 ('Spuyten Duyvil', 4),
 ('Castleton Corners', 4)
```

```
df_norm["neighbourhood_group"].unique()
```

```
array(['Brooklyn', 'Manhattan', 'Queens', 'Staten Island', 'Bronx'],
      dtype=object)
```

Рис. 3: Корректность данных

Проверим на выбросы столбец calculated_host_list - количество объявлений у одного арендодателя. Получилось, что в 5 процентов с начала и конца входят 2402 значения, учитывая что в датасете около 50000 строк, удалим эти выбросы.

```
calculated_host_listings_count_stat = {"mean": df_norm["calculated_host_listings_count"].mean(),
            "median": df_norm["calculated_host_listings_count"].median(),
            "mode": df_norm["calculated_host_listings_count"].mode().to_list(),
            "interquartile_range": df_norm["calculated_host_listings_count"].quantile(0.75) - df_norm["calculated_host_listings_
            }
calculated_host_listings_count_stat
```

```
{'mean': 7.143982002249719,
 'median': 1.0,
 'mode': [1],
 'interquartile_range': 1.0}
```

```
print(df_norm["calculated_host_listings_count"].quantile(0.05))
print(df_norm["calculated_host_listings_count"].quantile(0.95))
```

```
1.0
15.0
```

```
print(df_norm["calculated_host_listings_count"][df_norm.calculated_host_listings_count < 1.0].count())
print(df_norm["calculated_host_listings_count"][df_norm.calculated_host_listings_count > 15.0].count())
```

```
0
2402
```

```
# Удалим выброс. Учитывая что в датасете около 50000 строк
df_norm = df_norm[df_norm.calculated_host_listings_count <= 15.0]
df_norm
```

Рис. 4: Удаление выбросов

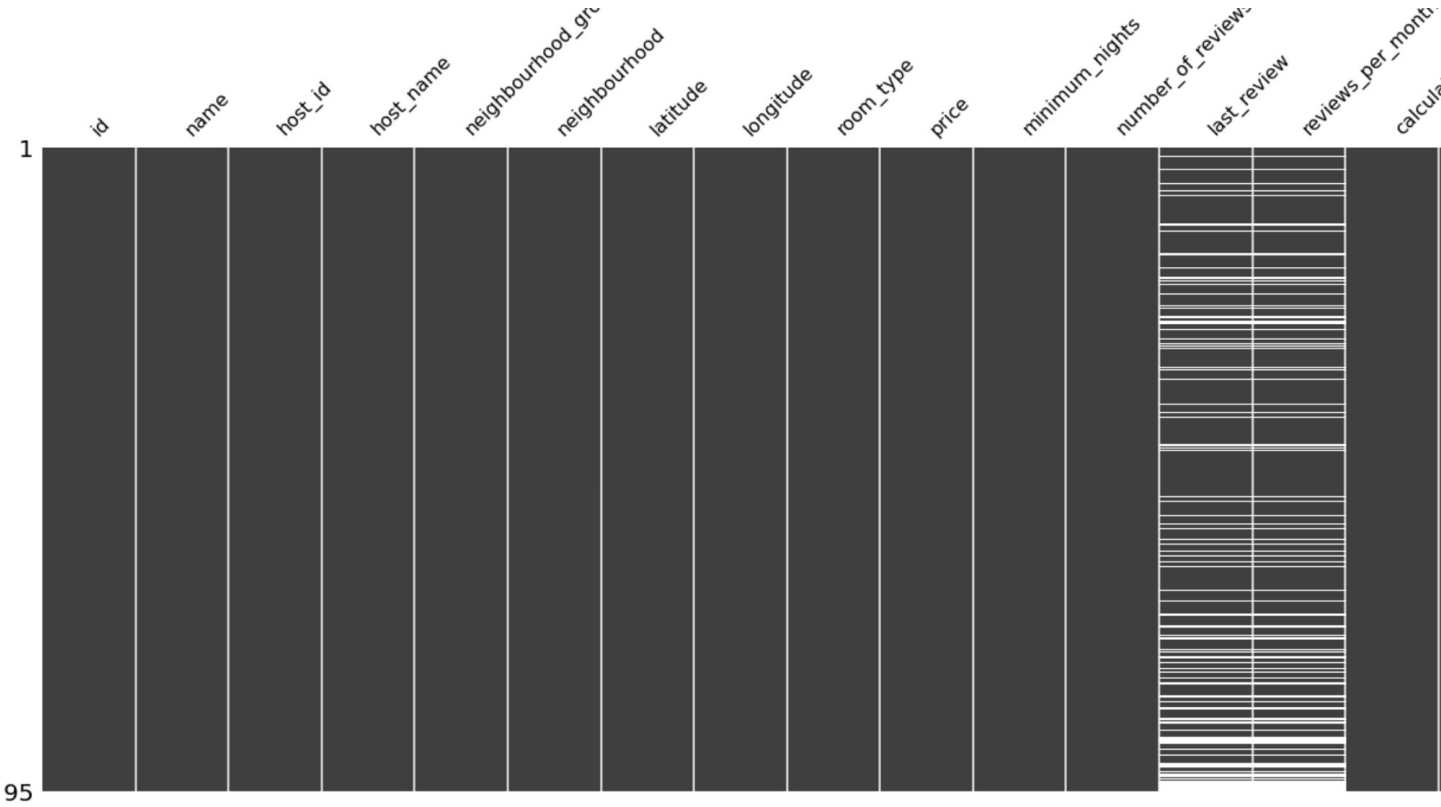


Рис. 5: Заполненность датасета

Обработаем пустые значения в столбце reviews_per_month. Этот столбец связан со столбцами количества оценок и дата последней. Если количество оценок равно нулю, то значит количество оценок в месяц тоже равно нулю. Поэтому заполним пустые значения нулями.

```
df2["reviews_per_month"] = df2["reviews_per_month"].apply(nan_to_median, col=df_norm["reviews_per_month"].to_list())
df2.head()
```

| ood | latitude | longitude | room_type | price | minimum_nights | number_of_reviews | last_review | reviews_per_month | calculated_host_listings_count | availability_365 |
|---|---|---|---|---|---|---|---|---|---|---|
| gton | 40.64749 | -73.97237 | 1 | 149 | 1 | 9 | 2018-10-19 | 0.21 | 6 | 365 |
| own | 40.75362 | -73.98377 | 2 | 225 | 1 | 45 | 2019-05-21 | 0.38 | 2 | 355 |
| lem | 40.80902 | -73.94190 | 1 | 150 | 3 | 0 | NaN | 0.00 | 1 | 365 |
| Hill | 40.68514 | -73.95976 | 2 | 89 | 1 | 270 | 2019-07-05 | 4.64 | 1 | 194 |
| lem | 40.79851 | -73.94399 | 2 | 80 | 10 | 9 | 2018-11-19 | 0.10 | 1 | 0 |

Рис. 6: Обработка пустых ячеек в столбце reviews_per_month

Переведем даты в подходящий тип данных. Заменим значения NaN. В данном случае пустое значение означает что объявление не имеет ни одной оценки, вследствие даты тоже нет. Поставим дату которая точно не входит в датасет.

```python
def nan_to_date(cell, col):
    if cell != cell:
        return '2001-01-01'
    else:
        return cell
```

```python
vy["last_review"] = vy["last_review"].apply(nan_to_date, col=vy["last_review"].to_list())
```

```python
vy["last_review"] = vy["last_review"].astype('datetime64[D]')
vy["last_review"]
```

```
0       2018-10-19
1       2019-05-21
2       2001-01-01
3       2019-07-05
4       2018-11-19
           ...
48890   2001-01-01
48891   2001-01-01
48892   2001-01-01
48893   2001-01-01
48894   2001-01-01
Name: last_review, Length: 46493, dtype: datetime64[ns]
```

Рис. 7: Обработка пустых ячеек в столбце last_review

В столбце availability_365 находятся значения от 0 до 365. Выбросов нет.

```python
# выбросы
print(df_norm1["availability_365"].max())
print(df_norm1["availability_365"].min()) # Выбросов нет
```

```
365
0
```

Рис. 8: Проверка выбросов

Гипотезы:

1. Больше 80 процентов арендодателей у которых больше 1 объявления сдают квартиры в одном боро. Потому что так легче следить за состоянием жилья.

2. Чем меньше последняя дата оценки, тем меньше отзывов на объявлении. То есть линейная зависимость.

3. Если убрать закрытые объявления, то распределение по популярности районов остается тем же. Распределение по боро тоже остается - т.е. Манхэттен, Бруклин, Квинс, Бронкс, Статен Айленд.

4. Больше 90 процентов арендодателей, у которых несколько объявлений, не имеют закрытых объявлений.

5. Закрытые объявления влияют на распределение по средней цене в районах и в боро.