Приложение №5

```
import pandas as pd
import numpy as np
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Импорт датасетов

```
groups = pd.read_csv('/content/drive/MyDrive/датасеты/groups.csv')
```

Группы-тест

```
groups_test = groups[['уникальный номер', 'направление 2',  'направление 3']]
```

Функция приведения списка адресов в единый вид

```
def clean_group_address(address: str):
  """Function for delete replicas in address"""
  address = address.replace('г.о.', 'город')
  addresses = address.split(',')
  for i in range(len(addresses)):
      if 'Москва' in addresses[i] or 'москва' in addresses[i]:
          addresses[i] = (addresses[i].replace('город', '').replace('г.', '').replace('Город', '').replace('Г',
  addresses = ', '.join(addresses)
```

```
  addresses = addresses.split('Москва,')
  # убрать пустой список вначале, запятые в конце списка
  addresses = ['город Москва, ' + i.strip().strip(',') for i in addresses if i != ' ' and i != '']
  return addresses
```

```
groups['округ площадки'] = groups['округ площадки'].fillna('')
groups['адрес площадки'] = groups['адрес площадки'].fillna('')
```

Выделение онлайн и оффлайн групп

```
groups_on = groups.loc[groups['направление 2'].str.contains('ОНЛАЙН')]
groups_off = groups.loc[~groups['направление 2'].str.contains('ОНЛАЙН')]
```

```
groups_on['адрес площадки'] = 'NaN'
groups_on['округ площадки'] = 'NaN'
```

```
    <ipython-input-117-8df8084292fd>:1: SettingWithCopyWarning:
    A value is trying to be set on a copy of a slice from a DataFrame.
    Try using .loc[row_indexer,col_indexer] = value instead

    See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returr
      groups_on['адрес площадки'] = 'NaN'
    <ipython-input-117-8df8084292fd>:2: SettingWithCopyWarning:
    A value is trying to be set on a copy of a slice from a DataFrame.
    Try using .loc[row_indexer,col_indexer] = value instead

    See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returr
      groups_on['округ площадки'] = 'NaN'
```

Приведение адресов и округов в единый вид

```
groups_off['адрес площадки'] = groups_off['адрес площадки'].apply(lambda x: clean_group_address(x))
```

```
    <ipython-input-118-dcd2d9dcd77d>:1: SettingWithCopyWarning:
    A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returr
  groups_off['адрес площадки'] = groups_off['адрес площадки'].apply(lambda x: clean_group_address(x))
```

```
groups_off['округ площадки']=groups_off['округ площадки'].str.split(',')
```

```
<ipython-input-119-57f63e7b73ea>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returr
  groups_off['округ площадки']=groups_off['округ площадки'].str.split(',')
```

```
groups_off.sample(3)
```

	уникальный номер	направление 1	направление 2	направление 3	адрес площадки	округ площадки	район площадки	расписани в активны периода
15252	801360464	Физическая активность	Фитнес, тренажеры	Тренажеры	[город Москва, 2-я Вольская улица, дом 16, к...	[Юго-Восточный административный округ]	муниципальный округ Некрасовка	Na
17491	801372208	Физическая активность	Спортивные игры	Настольный теннис	[город Москва, Олонецкий проезд,	[Северо-Восточный административный округ]	муниципальный округ Бабушкинский	21.02.202 г 31.12.202 П

Выявление несоответствия длин списков по адресам и округам. Удаление лишних, так как их 12 на список из 27000.

```
groups_off['okrug'] = groups_off['округ площадки'].apply(lambda x: len(x))
groups_off['addresses'] = groups_off['адрес площадки'].apply(lambda x: len(x))
groups_off['compare'] = groups_off['addresses'] == groups_off['okrug']
```

```
<ipython-input-121-310f6eeb05c1>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returr
  groups_off['okrug'] = groups_off['округ площадки'].apply(lambda x: len(x))
<ipython-input-121-310f6eeb05c1>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returr
  groups_off['addresses'] = groups_off['адрес площадки'].apply(lambda x: len(x))
<ipython-input-121-310f6eeb05c1>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returr
  groups_off['compare'] = groups_off['addresses'] == groups_off['okrug']
```

```
res = groups_off.loc[groups_off['compare']==False]
res.shape
```

```
(12, 13)
```

```
groups_off = groups_off.loc[groups_off['compare']==True]
```

```
groups_off = groups_off.drop(columns=['okrug', 'addresses', 'compare'], axis=1)
groups_off.head(1)
```

	уникальный номер	направление 1	направление 2	направление 3	адрес площадки	округ площадки	район площадки	расписание в активных периодах	ра в
0	801357270	Физическая активность	ОФП	ОФП	[город Москва, Саратовская улица, дом	[Юго-Восточный административный округ]	муниципальный округ Текстильщики	NaN	0 3

Разбиение этих списков на строки

```
groups_off = groups_off.explode(['округ площадки', 'адрес площадки']).reset_index(drop= True )
```

```
groups = pd.concat([groups_off, groups_on], ignore_index=True)
```

Преобразование округов из списка онлайн групп. Адреса не нужно преобразовывать, в них много других городов.

```
def clean_group_district(district: str):
  """Function for delete replicas in district"""
  district = district.replace(' и ', ' округ,').replace('округа', 'округ').split(',')
  districts_list = [i.strip() for i in district if i != ' ']
  return districts_list
```

```
groups['округ площадки'] = groups['округ площадки'].apply(lambda x: clean_group_district(x) if x != '' else x)
```

```
groups = groups.explode(['округ площадки']).reset_index(drop= True )
```

```
groups['округ площадки'] = groups['округ площадки'].replace('Новомосковский административные округ','Новомоскс
                                                    replace ('Троицкий округ', 'Троицкий административный округ'
```

```
groups['округ площадки'].value_counts()
```

```
NaN                                        7497
Восточный административный округ           3403
Южный административный округ               3014
Северо-Восточный административный округ    2879
Западный административный округ            2171
Юго-Западный административный округ        2170
Центральный административный округ          2034
Юго-Восточный административный округ        1886
Северный административный округ             1855
Северо-Западный административный округ      1575
Троицкий административный округ              526
Новомосковский административный округ        526
Зеленоградский административный округ        496
                                             119
Name: округ площадки, dtype: int64
```

```
len(groups)
```

```
30157
```

```
groups.sample(3)
```

	уникальный номер	направление 1	направление 2	направление 3	адрес площадки	округ площадки	район площадки	ра в
24590	801347025	Образование	ОНЛАЙН Психология и коммуникации	ОНЛАЙН Психологические тренинги	NaN	NaN	NaN	

20587	801364574	Физическая активность	Гимнастика	Адаптивная и тонизирующая гимнастика	город Москва, Московский проспект,	Восточный административный округ	муниципальный округ Измайлово	02 31

Функции по преобразованию расписания в активных периодах (так как оно нужно для работы) в единый вид

```
def extract_time_weekday(elem: list) -> dict:
    weeksdays_list = ['Пн.', 'Вт.', 'Ср.', 'Чт.', 'Пт.', 'Сб.', 'Вс.']
    times = []
    weeksdays = []
    schedule_dict = {}
    for j in elem:
        if j in weeksdays_list:
            weeksdays.append(j[:-1])
        if '-' in j:
            times.append(j)
    if len(weeksdays) == len(times):
        schedule_dict = dict(zip(weeksdays, times))
    elif len(weeksdays) > len(times):
        for i in weeksdays:
            schedule_dict[i] = times[0]
    else:
        for i in times:
            times[i] = weeksdays
```

```
            times[1] = weekdays
    if schedule_dict:
        return schedule_dict


def clean_str(s: str) -> str:
    str_clean = (s.replace('с', '')
                 .replace('по', '')
                 .replace('без перерыва', '')
                 .replace(',', '')).strip()
    return str_clean


def case_one_time(s: str) -> list:
    """
    обработка случая 'с 31.03.2023 по 31.12.2023, Пн., Ср. 17:00-19:00, без перерыва'
    """
    str_clean = clean_str(s)
    spl = [i for i in str_clean.split(' ')]
    lst = []
    schedule_dict = extract_time_weekday(spl)
    if schedule_dict:
        for day, times in schedule_dict.items():
            lst.append([spl[0], spl[2], day, times])
    return lst


def case_two_dates_two_time(s: str) -> list:
    """
    обработка случая 'с 31.03.2023 по 31.12.2023, Вт. 17:00-19:00, без перерыва, Пт. 13:00-15:00, без перерыва
    """
    str_split = s.split('без перерыва')
    str_clean = clean_str(str_split[0])
    spl = [i for i in str_clean.split(' ')]
    lst = []
    schedule_dict = extract_time_weekday(spl)
```

```
    schedule_dict = extract_time_weekday(spl)
    if schedule_dict:
        for day, times in schedule_dict.items():
            lst.append([spl[0], spl[2], day, times])
    for i in range(1, len(str_split)):
        new_str_split = [i for i in clean_str(str_split[i]).split(' ')]
        schedule_dict_ = extract_time_weekday(new_str_split)
        if schedule_dict_:
            for day_, times_ in schedule_dict_.items():
                lst.append([spl[0], spl[2], day_, times_])
    return lst


def case_two_dates_one_time(s: str) -> list:
    """
    обработка случая 'с 01.06.2022 по 11.08.2022, Пн., Ср. 12:05-13:05, без перерыва;
    с 01.01.2022 по 31.05.2022, Пн., Ср. 12:15-13:15, без перерыва;
    """
    result = []
    spl = [i.strip() for i in s.split(';')]
    for lst in spl:
        if lst.count('перерыв') > 1:
            lst = case_two_dates_two_time(lst)
            result += lst
        else:
            lst = case_one_time(lst)
            result += lst
    return result
```

```
def extract(s) -> list:
    if s.count(';') == 0:
        if s.count('перерыв') > 1:
            result = case_two_dates_two_time(s)
        else:
```

```
            result = case_one_time(s)
    else:
        result = case_two_dates_one_time(s)
    return result
```

```
groups['расписание в активных периодах'] = groups['расписание в активных периодах'].fillna('')
```

```
groups['расписание в активных периодах'] = groups['расписание в активных периодах'].apply(lambda x: extract(x)
```

```
groups.sample(3)
```

	уникальный номер	направление 1	направление 2	направление 3	адрес площадки	округ площадки	район площадки	расписани в активны периода
2847	801355632	Физическая активность	Спортивные игры	Волейбол	город Москва, Перовская улица, дом 37	Восточный административный округ	муниципальный округ Перово	
7119	801356514	Образование	Английский	Английский	город Москва, Главная	Восточный административный	муниципальный округ Восточный,	

Чтобы изменить содержимое ячейки, дважды нажмите на нее (или выберите "Ввод")

```
groups = groups.explode(['расписание в активных периодах']).reset_index(drop= True )
```

```
groups['день недели'] = groups['расписание в активных периодах'].str[2]
groups['дата начала'] = groups['расписание в активных периодах'].str[0]
groups['дата окончания'] = groups['расписание в активных периодах'].str[1]
```

```
groups['время начала'] = groups['расписание в активных периодах'].str[3]
groups['время начала'] = groups['время начала'].str.split('-')
groups['время окончания'] = groups['время начала'].str[1]
groups['время начала'] = groups['время начала'].str[0]
```

```
WEEK = {'Пн': 'понедельник', 'Вт': 'вторник', 'Ср': 'среда','Чт': 'четверг', 'Пт': 'пятница', 'Сб': 'суббота',
groups = groups.replace({'день недели': WEEK})
```

```
groups.reset_index(inplace= True)
groups = groups.rename(columns={'index': 'uniq_index'})
```

```
#groups_correct = groups[['uniq_index', 'расписание в активных периодах', 'уникальный номер', 'направление 2',
 #                        'weekday', 'start_date', 'end_date', 'start_time', 'end_time']]
groups_correct = groups.drop(['направление 1', 'расписание в закрытых периодах', 'расписание в плановом период
```

```
groups_correct.sample(5)
```

	uniq_index	уникальный номер	направление 2	направление 3	адрес площадки	округ площадки	расписание в активных периодах	дек недел
29118	29118	801355584	ОНЛАЙН Рисование	ОНЛАЙН Масляная живопись	NaN	NaN		Na

24483	24483	801363832	Гимнастика	Гимнастика	город Москва, проспект Андропова, дом 22	Южный административный округ	[02.03.2023, 31.12.2023, Чт, 11:00-12:00]	четвер
15832	15832	801367587	Рисование	Различные техники рисования	город Москва, Партизанская улица, дом 11	Западный административный округ		Na
			Фитнес,		город Москва, улица	Северо-Восточный	[03.03.2023, 31.12.2023,	

```
attend = pd.read_csv('/content/drive/MyDrive/датасеты/attend.csv')
users = pd.read_csv('/content/drive/MyDrive/датасеты/users.csv')
dictx = pd.read_excel('/content/drive/MyDrive/датасеты/dict.xlsx')
```

```
groups_corr_test = groups_correct.loc[groups_correct['расписание в активных периодах'] != '']
```

```
groups_corr_test = groups_corr_test.drop(columns=['расписание в активных периодах'], axis=1)
groups_correct = groups_correct.drop(columns=['расписание в активных периодах'], axis=1)
```

```
users_test = users.sample(frac = 0.1)
```

```
attend_test =  attend.loc[attend['уникальный номер группы'].isin(groups_corr_test['уникальный номер'])]
```

```
attend_test =  attend_test.loc[attend['уникальный номер участника'].isin(users_test['уникальный номер'])]
```

```
attend_test = attend_test.sample(frac = 0.5)
```

```
attend_test.shape
```

```
(40667, 9)
```

```
attend_test.to_csv('/content/drive/MyDrive/датасеты/attend_test.csv', index=False)
users_test.to_csv('/content/drive/MyDrive/датасеты/users_test.csv', index=False)
groups_corr_test.to_csv('/content/drive/MyDrive/датасеты/groups_test_corr.csv', index=False)
groups_test.to_csv('/content/drive/MyDrive/датасеты/groups_test.csv', index=False)
```