In [1]:

```
import scipy.stats as sps
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from tqdm.notebook import tqdm

import datetime

import plotly.graph_objects as go
import plotly.express as px
import plotly.offline

sns.set(font_scale=1.5)
```

Задача исследования групповых поездок.

Загрузим оба датасета с данными о поездках и станциях велопроката.

In [2]:

```
stations = pd.read_csv('cycle-share-dataset/station.csv')

stations.head()
```

Out[2]:

	station_id	name	lat	long	install_date	install_dockcount	modification_date	current_dockcount	decommission_date
0	BT-01	3rd Ave & Broad St	47.618418	-122.350964	10/13/2014	18	NaN	18	NaN
1	BT-03	2nd Ave & Vine St	47.615829	-122.348564	10/13/2014	16	NaN	16	NaN
2	BT-04	6th Ave & Blanchard St	47.616094	-122.341102	10/13/2014	16	NaN	16	NaN
3	BT-05	2nd Ave & Blanchard St	47.613110	-122.344208	10/13/2014	14	NaN	14	NaN
4	CBD-03	7th Ave & Union St	47.610731	-122.332447	10/13/2014	20	NaN	20	NaN

In [3]:
```
trips = pd.read_csv('cycle-share-dataset/trip.csv',
                    error_bad_lines=False,
                    parse_dates=[1, 2])
```

```
b'Skipping line 50794: expected 12 fields, saw 20\n'
```

In [4]:
```
trips.head(5)
```

Out[4]:

	trip_id	starttime	stoptime	bikeid	tripduration	from_station_name	to_station_name	from_station_id	to_station_id	usertype	gender	birthyear
0	431	2014-10-13 10:31:00	2014-10-13 10:48:00	SEA00298	985.935	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	1960.0
1	432	2014-10-13 10:32:00	2014-10-13 10:48:00	SEA00195	926.375	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	1970.0
2	433	2014-10-13 10:33:00	2014-10-13 10:48:00	SEA00486	883.831	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Female	1988.0
3	434	2014-10-13 10:34:00	2014-10-13 10:48:00	SEA00333	865.937	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Female	1977.0
4	435	2014-10-13 10:34:00	2014-10-13 10:49:00	SEA00202	923.923	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	1971.0

Предобработка данных

Упорядочим данные по времени начала поездки

In [5]:
```
sorted_by_time = trips.sort_values(by='starttime').reset_index(drop=True)
```

In [6]:
```
1 sorted_by_time.head()
```

Out[6]:

	trip_id	starttime	stoptime	bikeid	tripduration	from_station_name	to_station_name	from_station_id	to_station_id	usertype	gender	birthyear
0	431	2014-10-13 10:31:00	2014-10-13 10:48:00	SEA00298	985.935	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	1960.0
1	431	2014-10-13 10:31:00	2014-10-13 10:48:00	SEA00298	985.935	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	1960.0
2	432	2014-10-13 10:32:00	2014-10-13 10:48:00	SEA00195	926.375	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	1970.0
3	432	2014-10-13 10:32:00	2014-10-13 10:48:00	SEA00195	926.375	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	1970.0
4	433	2014-10-13 10:33:00	2014-10-13 10:48:00	SEA00486	883.831	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Female	1988.0

Заметим, что у первых четырёх поездок совпадают маршруты и примерно совпадают времена начала поездки и окончания. Скорее всего это была групповая поездка.

Попробуем выделить схожие групповые поездки.

Заведём столбец `group_id`. Сначала для для всех поездок `group_id = -1`. Затем, для каждой поездки будем находить поездки, со схожими в пределах 5 минут временами начала и окончания поездки и совпадающим маршрутом.

Такие поездки будем считать за групповые поездки и присваивать группе уникальный `group_id`.

In [7]:

```
sorted_by_time['group_id'] = -1

curr_id = 0
for i in tqdm(range(sorted_by_time.shape[0])):
    if sorted_by_time.loc[i, :].group_id == -1:
        curr_row = sorted_by_time.loc[i, :]

        sorted_by_time.loc[i, 'group_id'] = curr_id

        starttime = sorted_by_time.loc[i, 'starttime'].to_pydatetime()
        delay = starttime + datetime.timedelta(minutes=5)
        delta = 5 * 60

        j = i + 1
        while (j < sorted_by_time.shape[0] and
               sorted_by_time.loc[j, 'starttime'].to_pydatetime() <= delay):

            obs_row = sorted_by_time.loc[j, :]

            if (obs_row.from_station_id == curr_row.from_station_id and
                obs_row.to_station_id == curr_row.to_station_id and
                obs_row.tripduration >= curr_row.tripduration - delta and
                obs_row.tripduration <= curr_row.tripduration + delta):

                sorted_by_time.loc[j, 'group_id'] = curr_id

            j += 1


        curr_id += 1
```

100% 286857/286857 [09:52<00:00, 484.18it/s]

Полученный датафрейм.

In [8]:
```
1 sorted_by_time.head()
```

Out[8]:

	trip_id	starttime	stoptime	bikeid	tripduration	from_station_name	to_station_name	from_station_id	to_station_id	usertype	gender	birthyear
0	431	2014-10-13 10:31:00	2014-10-13 10:48:00	SEA00298	985.935	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	1960.0
1	431	2014-10-13 10:31:00	2014-10-13 10:48:00	SEA00298	985.935	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	1960.0
2	432	2014-10-13 10:32:00	2014-10-13 10:48:00	SEA00195	926.375	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	1970.0
3	432	2014-10-13 10:32:00	2014-10-13 10:48:00	SEA00195	926.375	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	1970.0
4	433	2014-10-13 10:33:00	2014-10-13 10:48:00	SEA00486	883.831	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Female	1988.0

Некоторые исследования групповых поездок

Среднее число участников групповых поездок.

In [9]:
```
1 sorted_by_time.shape[0] / np.unique(sorted_by_time.group_id).shape[0]
```

Out[9]:
```
1.4360010212203582
```

Подсчитаем количество участников каждой группы.

In [10]:
```
1 group_count = pd.DataFrame(sorted_by_time.groupby(by = 'group_id').
2                            count()['trip_id'])
3
4 group_count.columns = ['members_num']
5 group_count.reset_index(inplace=True)
```

In [11]:
```
group_av = (group_count[group_count.members_num > 1].shape[0] /
            group_count.shape[0])

print(f'Отношение количества групповых поездок ко всем остальным:\
 {group_av}')
```

```
Отношение количества групповых поездок ко всем остальным: 0.3477805978143882
```

Промежуточный вывод:

Можно видеть, что примерно 35% всех поездок - групповые. Это значит, что предполагаемой компании владельцу станций велопроката стоит разделять тарифные планы для одиночных поездок и групповых.

Исследуем, в какое время дня, в какой день недели, в какой месяц наиболее часто совершаются групповые поездки.

Визуализируем гистограммы.

In [12]:
```
group_trips = sorted_by_time.copy()
group_trips = group_trips.merge(group_count,
                                how='left', on='group_id')

group_trips = group_trips[group_trips.members_num > 1]
```

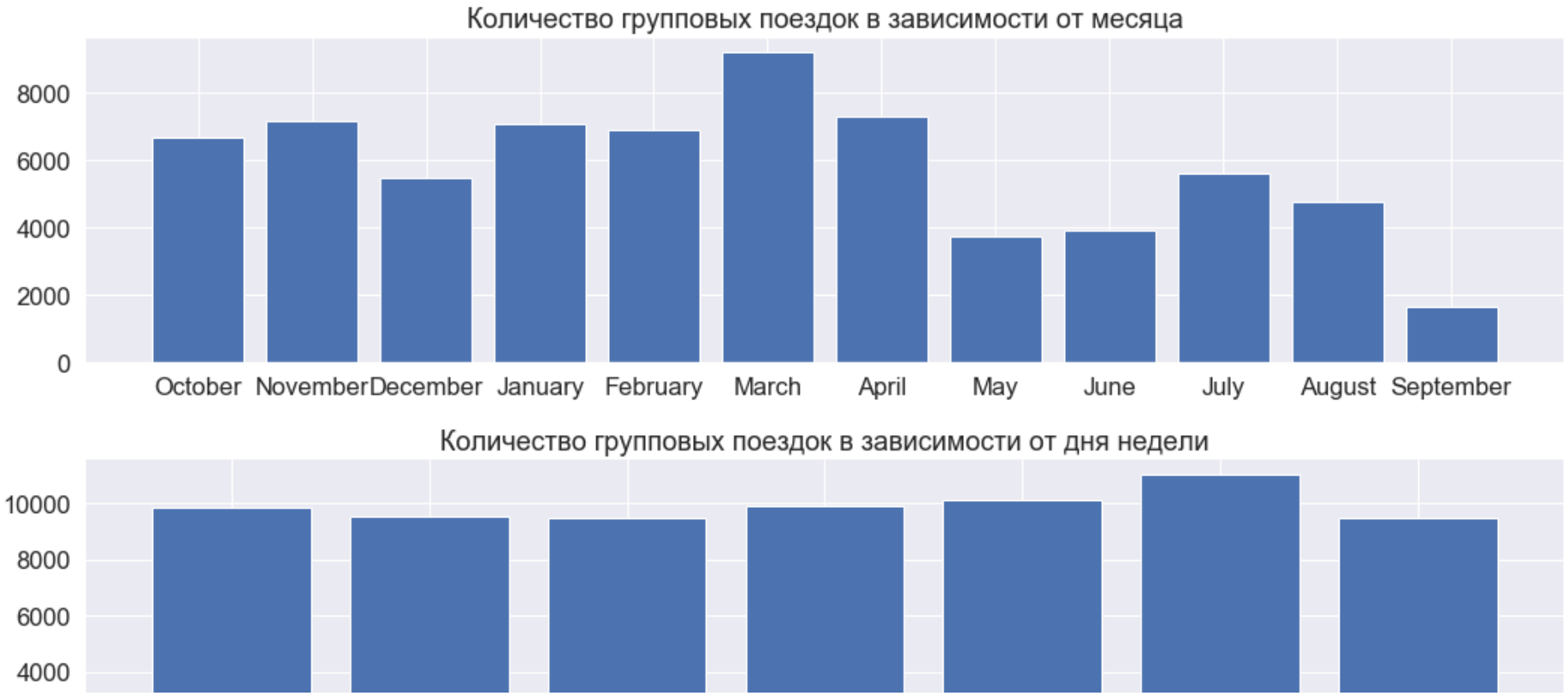
In [13]:
```
group_unique_trips = group_trips.drop_duplicates(subset=['group_id']).copy()
group_unique_times = group_unique_trips.loc[:, 'starttime'].copy()

group_unique_trips['month'] = group_unique_times.dt.month_name()
group_unique_trips['day'] = group_unique_times.dt.day_name()
group_unique_trips['time_of_day'] = ((group_unique_times.dt.hour.values >= 6).astype('int') +
                                     (group_unique_times.dt.hour.values >= 12).astype('int') +
                                     (group_unique_times.dt.hour.values >= 18).astype('int'))
```

In [14]:

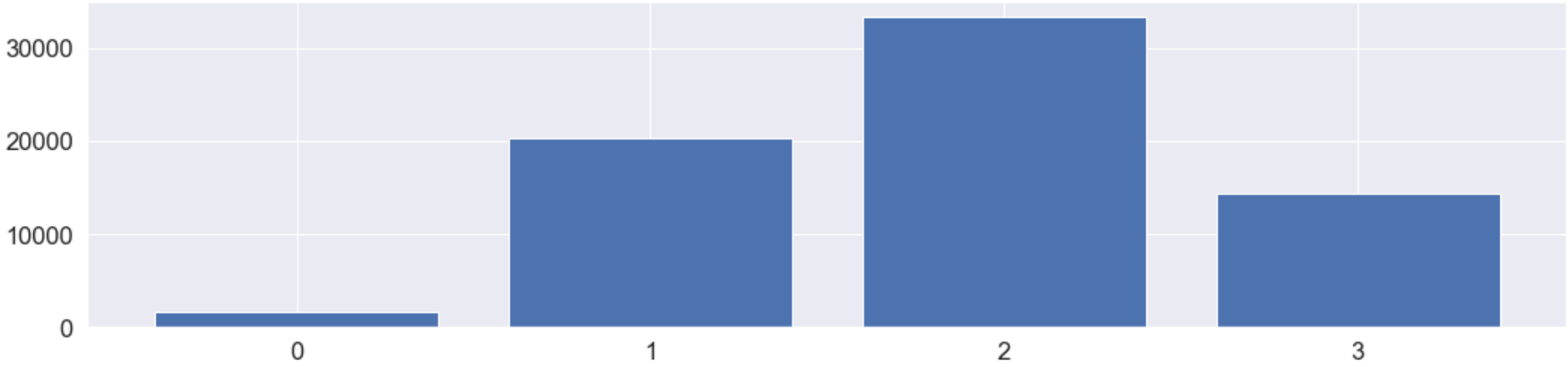
```
from collections import defaultdict

counts_month = defaultdict(int)
for e in group_unique_trips['month']:
    counts_month[e] += 1

counts_day = defaultdict(int)
for e in group_unique_trips['day']:
    counts_day[e] += 1

counts_day_time = defaultdict(int)
for e in group_unique_trips['time_of_day']:
    counts_day_time[e] += 1
```

In [15]:

```
plt.figure(figsize=(15, 12))

plt.subplot(3, 1, 1)
plt.title('Количество групповых поездок в зависимости от месяца')
plt.bar(list(counts_month.keys()), list(counts_month.values()))

plt.subplot(3, 1, 2)
plt.title('Количество групповых поездок в зависимости от дня недели')
plt.bar(list(counts_day.keys()), list(counts_day.values()))

plt.subplot(3, 1, 3)
plt.title('Количество групповых поездок в зависимости от времени суток\
 (раннее утро, утро, день, вечер)')
plt.bar(list(counts_day_time.keys()), list(counts_day_time.values()))
plt.xticks([0, 1, 2, 3])

plt.tight_layout()
```



Количество групповых поездок в зависимости от времени суток (раннее утро, утро, день, вечер)



По гистограммам можно видеть, что наибольшее число поездок совершается весной, по субботам и в вечернее время суток.

Это также можно учесть в тарификации поездок.

Построим график зависимости длительности поездки от ведичины группы.

In [16]:
```
unique_trips = sorted_by_time.copy()
unique_trips = unique_trips.merge(group_count,
                                  how='left', on='group_id')

unique_trips = unique_trips.drop_duplicates(subset=['group_id']).copy()
```

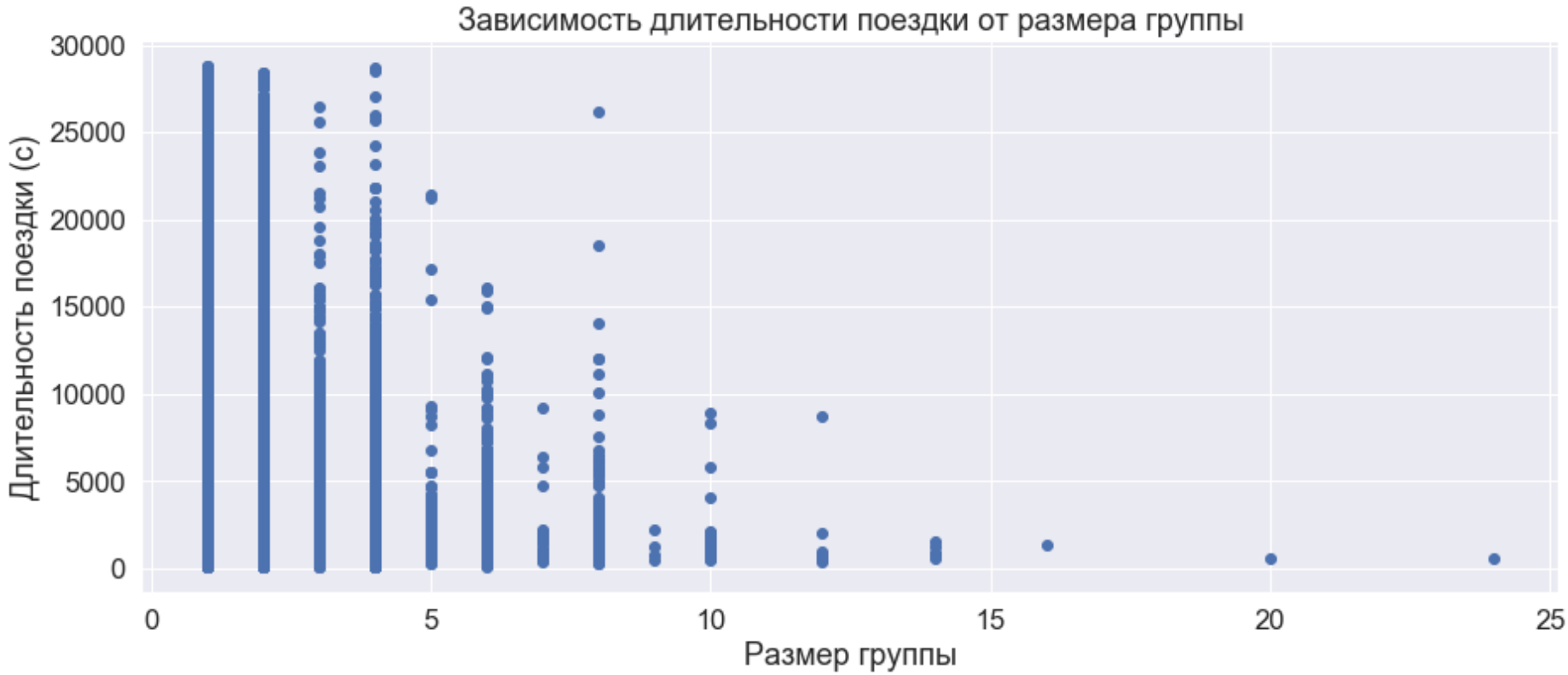
In [17]:
```
1 unique_trips
```

Out[17]:

	trip_id	starttime	stoptime	bikeid	tripduration	from_station_name	to_station_name	from_station_id	to_station_id	usertype	gender	bi
0	431	2014-10-13 10:31:00	2014-10-13 10:48:00	SEA00298	985.935	2nd Ave & Spring St	Occidental Park / Occidental Ave S & S Washing...	CBD-06	PS-04	Member	Male	
12	440	2014-10-13 11:35:00	2014-10-13 11:45:00	SEA00434	587.634	Occidental Park / Occidental Ave S & S Washing...	King Street Station Plaza / 2nd Ave Extension ...	PS-04	PS-05	Member	Male	
36	450	2014-10-13 11:40:00	2014-10-13 11:49:00	SEA00107	499.734	Occidental Park / Occidental Ave S & S Washing...	City Hall / 4th Ave & James St	PS-04	CBD-07	Member	Female	
38	453	2014-10-13 11:41:00	2014-10-13 11:51:00	SEA00178	571.807	Occidental Park / Occidental Ave S & S Washing...	1st Ave & Marion St	PS-04	CBD-05	Member	Male	
58	463	2014-10-13 11:44:00	2014-10-13 12:00:00	SEA00296	920.055	Occidental Park / Occidental Ave S & S Washing...	2nd Ave & Spring St	PS-04	CBD-06	Member	Female	
...	...	...	...	...	...	...	...	...	...	...	...	...
286847	255236	2016-08-31 22:13:00	2016-08-31 22:25:00	SEA00254	674.993	3rd Ave & Broad St	Occidental Park / Occidental Ave S & S Washing...	BT-01	PS-04	Member	Male	
286848	255237	2016-08-31 22:37:00	2016-08-31 22:39:00	SEA00330	144.477	Summit Ave & E Denny Way	Summit Ave E & E Republican St	CH-01	CH-03	Member	Male	
286849	255238	2016-08-31 22:44:00	2016-08-31 23:03:00	SEA00336	1106.063	Pier 69 / Alaskan Way & Clay St	2nd Ave & Blanchard St	WF-01	BT-05	Short-Term Pass Holder	NaN	
286852	255241	2016-08-31 23:34:00	2016-08-31 23:45:00	SEA00201	679.532	Harvard Ave & E Pine St	2nd Ave & Spring St	CH-09	CBD-06	Short-Term Pass Holder	NaN	
286853	255243	2016-08-31 23:47:00	2016-09-01 00:20:00	SEA00300	1951.173	Cal Anderson Park / 11th Ave & Pine St	6th Ave S & S King St	CH-08	ID-04	Short-Term Pass Holder	NaN	

199761 rows × 14 columns

In [18]:
```
plt.figure(figsize=(15, 6))

plt.title('Зависимость длительности поездки от размера группы')
plt.scatter(unique_trips.members_num, unique_trips.tripduration)

plt.xlabel('Размер группы')
plt.ylabel('Длительность поездки (с)')
plt.show()
```



Наблюдается тенденция к уменьшению длительности при увеличении размера группы.

Проверим, есть ли зависимость, статистическими методами с помощью корреляций.

Коэффициент корреляции Спирмена. Он подходит для выборок большого размера способен выявлять нелинейные зависимости.

In [19]:
```
sps.spearmanr(unique_trips.members_num, unique_trips.tripduration)
```

Out[19]:
```
SpearmanrResult(correlation=0.20887796587249194, pvalue=0.0)
```

p_value < 0.05 гипотеза о независимости выборок отверглась. Следовательно, можно сказать, что есть тенденция к уменьшению длительности поездки при увеличении размера группы.

Данный факт также может повлиять на тарификацию компании.

Визуализация

Визуализируем, теперь, наиболее популярные групповые маршруты.

Воспользуемся функциями из ноутбука по визуализации данных.

In [20]:
```
def most_popular(trip):
    d = {}

    for route in trip.values:
        from_to = (route[7], route[8])
        if from_to in d:
            d[from_to] += 1
        else:
            d[from_to] = 1

    d = {k: v for k, v in sorted(d.items(), key=lambda item: item[1])[::-1]}

    return d
```

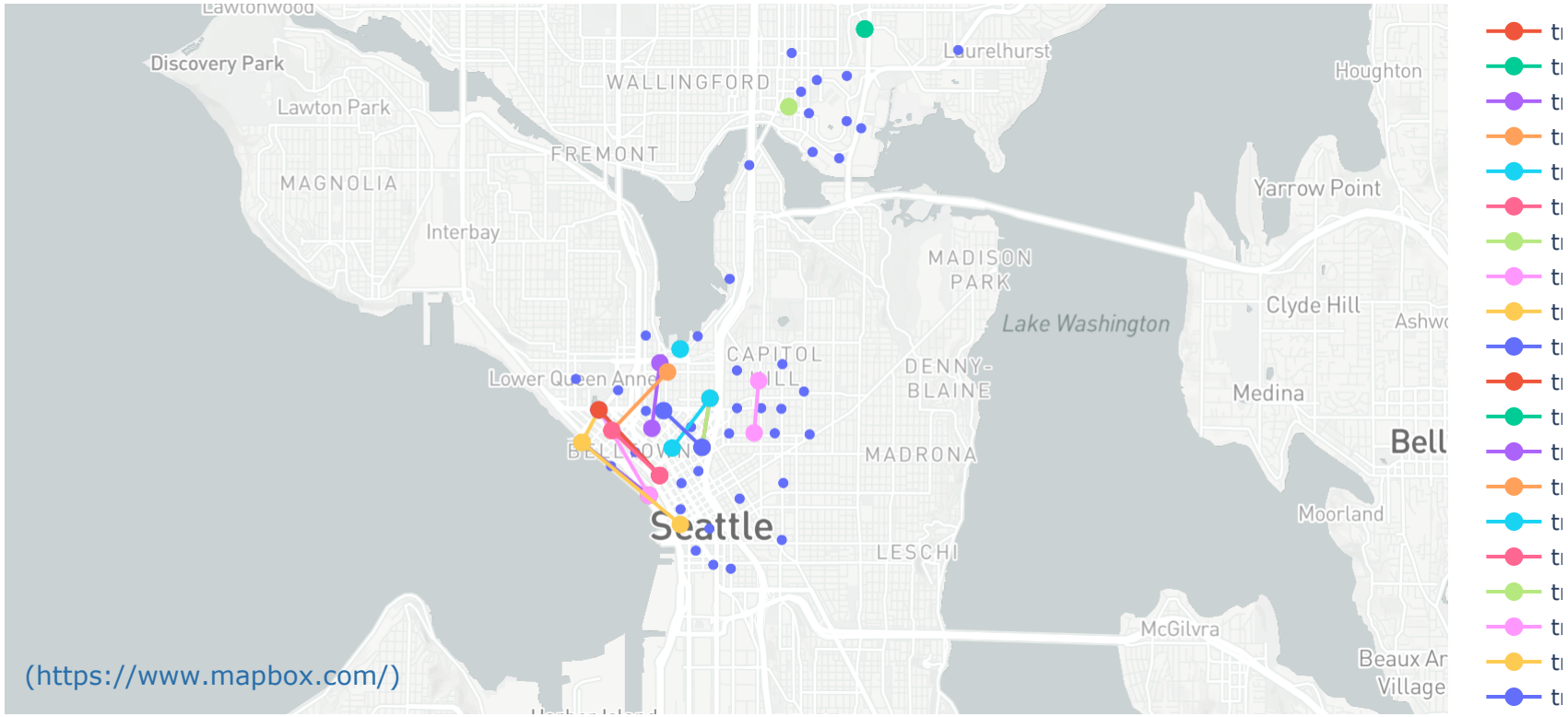
In [21]:

```
def plot_popular(popular_dict, n=10):
    px.set_mapbox_access_token(open("public_key").read())
    fig = px.scatter_mapbox(stations, lat="lat", lon="long",
                            zoom=11)
    top = list(popular_dict.keys())[:n]

    for route in top:
        st_from = route[0]
        st_to = route[1]
        from_lon = stations[stations.station_id == st_from].long.values[0]
        from_lat = stations[stations.station_id == st_from].lat.values[0]
        to_lon = stations[stations.station_id == st_to].long.values[0]
        to_lat = stations[stations.station_id == st_to].lat.values[0]

        fig.add_trace(go.Scattermapbox(mode = "markers+lines",
                                       lon = [from_lon, to_lon],
                                       lat = [from_lat, to_lat],
                                       marker = {'size': 10}))

    fig.show()
```

20 Наиболее популярных маршрутов у групп

In [22]:

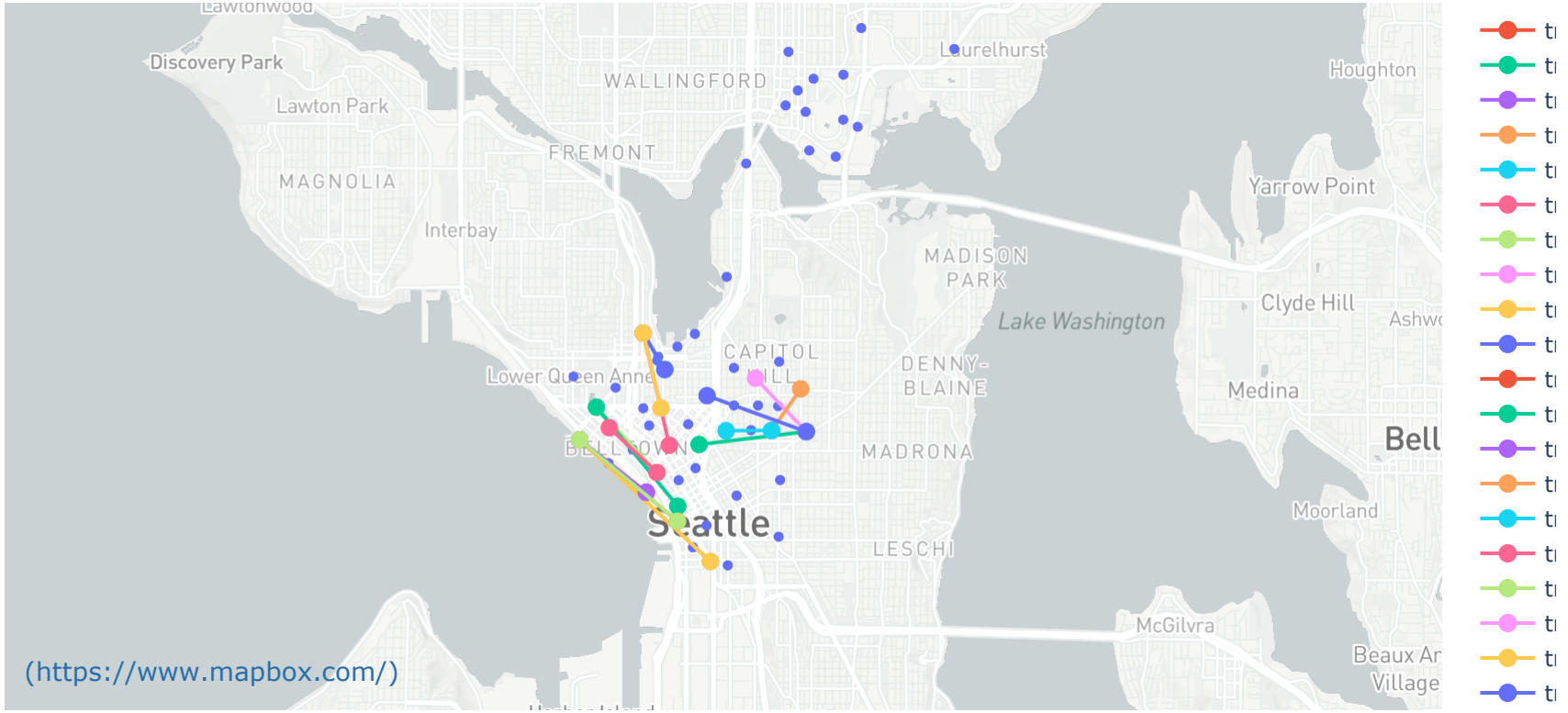
```
groups_popular = most_popular(unique_trips[unique_trips.members_num > 1])
plot_popular(groups_popular, 20)
```



20 Наиболее популярных маршрутов одиночных поездок

In [23]:

```
solo_popular = most_popular(unique_trips[unique_trips.members_num == 1])
plot_popular(solo_popular, 20)
```



Наблюдение

Видно отличие среди популярных маршрутов групповых и одиночных поездок. Достаточно много одинчных поездок сосредоточено в районе Capitol Hill с большим количеством баров и клубов.

Большинство же групповых поездок пролегают в районе Belltown. Это может быть связано как с густотой населения этого района, как и с обилием (судя по описанию) культурных достопримечательностей. Такие групповые поездки могут быть экскурсионными.

Также, много групповых поездок сосредоточено поблизости района Laurelhurst. Это может быть связано с тем, что в данном районе много парков и частных домов (семейные поездки/отдых на природе).

В данном вопросе достаточно много вопросов для исследования. Таких, как сравнение распределений одиночных и групповых поездок в зависимости от погодных условий, выявление зависимости параметров от полового состава группы.

Планируется продожение работы над групповыми поездками...

In [ ]:
```
1
```