In [1]:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import plotly.express as px
import seaborn as sns


sns.set(font_scale=1.5)
%matplotlib inline
```

executed in 41.5s, finished 18:27:24 2020-05-09

Формулировка задачи на простом "пользовательском" языке:

> Мы займемся определением самых популярных станций, их визуализацией на карте города, а также определением самых популярных маршрутов.

> Рассмотрим влияние погодных условий на количество поездок.

Формулировка задачи на математическом языке:

> По данным о поездках, определить наиболее часто используемые станции (пары станций в случае для маршрута), а также нанести их на карту города

> С помощью коэффициента корреляции выясним, есть ли зависимость между погодными условиями и количеством совершаемых поездок.

Загрузим чуть-чуть обработанные данные, где удалены дубликаты первых 50000 поездок в таблице `trips.csv` .

In [2]:

```
DATA_DIR = "../data/processed/"
```

executed in 4ms, finished 18:27:24 2020-05-09

In [3]:

```
stations_data = pd.read_csv(DATA_DIR+"station.csv")
trips = pd.read_csv(DATA_DIR+"trips.csv", error_bad_lines=False, index_col=0)
weather = pd.read_csv(DATA_DIR+"weather.csv")
```

executed in 11.7s, finished 18:27:36 2020-05-09

Самые популярные станции, откуда уезжают.

In [4]:

```
1 print(trips['from_station_name'].value_counts()[:10])
```

executed in 117ms, finished 18:27:36 2020-05-09

```
Pier 69 / Alaskan Way & Clay St           11274
E Pine St & 16th Ave                       9466
3rd Ave & Broad St                         9392
2nd Ave & Pine St                          8198
Westlake Ave & 6th Ave                     8188
Cal Anderson Park / 11th Ave & Pine St     7690
E Harrison St & Broadway Ave E             7685
2nd Ave & Vine St                          6568
Key Arena / 1st Ave N & Harrison St        6402
REI / Yale Ave N & John St                 6401
Name: from_station_name, dtype: int64
```

In [5]:

```
1 from_top_stations = trips['from_station_name'].value_counts()[:10].to_frame().
```

executed in 455ms, finished 18:27:36 2020-05-09

In [6]:

```
1 top_stations_data = stations_data.loc[
2     stations_data['name'].isin(from_top_stations)
3 ]
4 top_stations_data = top_stations_data.assign(
5     from_counter=trips['from_station_name'].value_counts()[:10].values
6 )
7 top_stations_data[['station_id', 'name', 'from_counter']]
```

executed in 907ms, finished 18:27:37 2020-05-09

Out[6]:

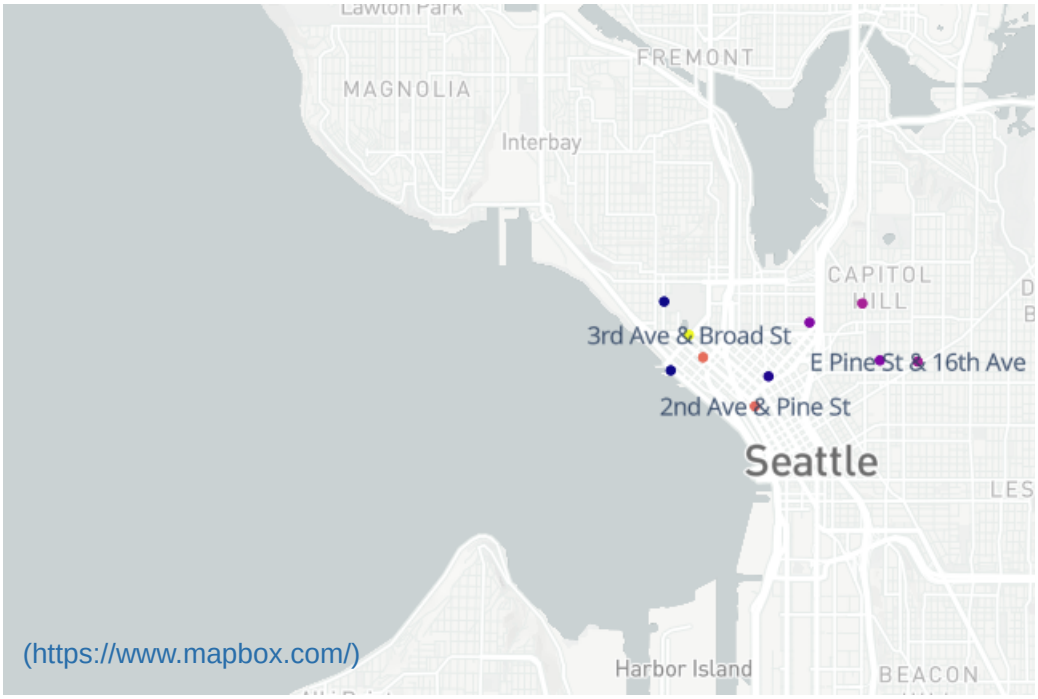
	station_id	name	from_counter
0	BT-01	3rd Ave & Broad St	11274
1	BT-03	2nd Ave & Vine St	9466
9	CBD-13	2nd Ave & Pine St	9392
12	CH-02	E Harrison St & Broadway Ave E	8198
16	CH-07	E Pine St & 16th Ave	8188
17	CH-08	Cal Anderson Park / 11th Ave & Pine St	7690
31	SLU-01	REI / Yale Ave N & John St	7685
35	SLU-15	Westlake Ave & 6th Ave	6568
39	SLU-19	Key Arena / 1st Ave N & Harrison St	6402
52	WF-01	Pier 69 / Alaskan Way & Clay St	6401

In [7]:

```
px.set_mapbox_access_token(open("../mapbox_token").read())
fig = px.scatter_mapbox(
    top_stations_data,
    lat="lat",
    lon="long",
    color="from_counter",
    text="name",
    size_max=15,
    zoom=11,
    title='Самые популярные станции отправления'
)
fig.show()
```

executed in 35.9s, finished 18:28:13 2020-05-09

Самые популярные станции отправления



In [8]:

```
to_top_stations = np.array(trips['to_station_name'].value_counts()[:10].to_fra
top_stations_data = stations_data.loc[
    stations_data['name'].isin(from_top_stations)
]
top_stations_data = top_stations_data.assign(
    to_counter=trips['to_station_name'].value_counts()[:10].values
)
```

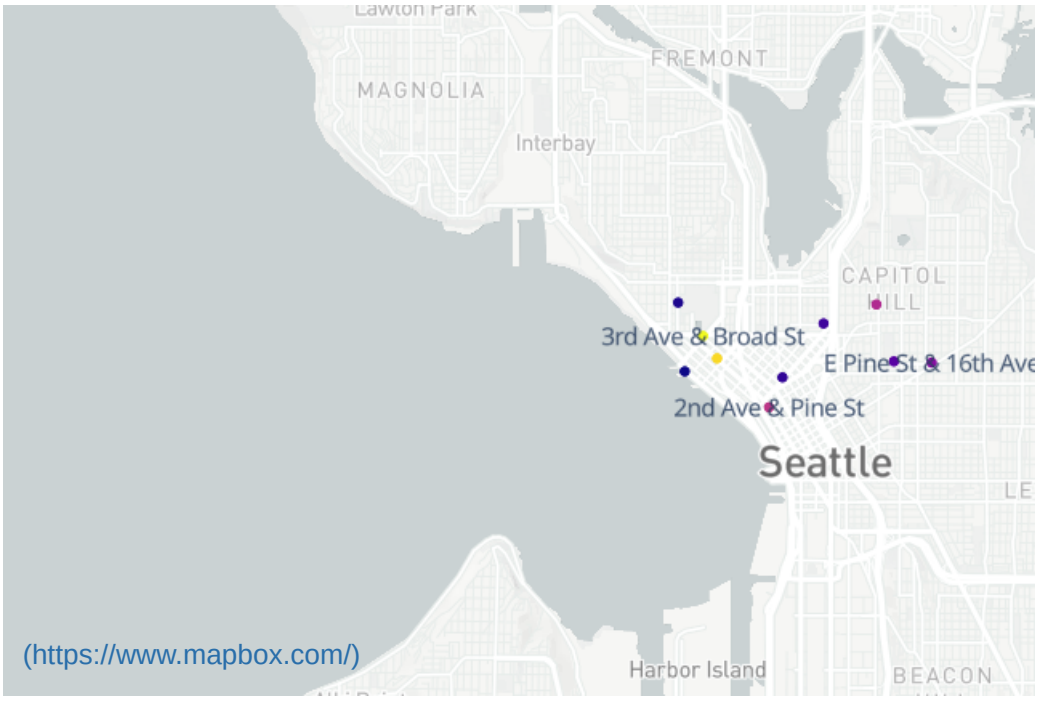
executed in 69ms, finished 18:28:13 2020-05-09

In [9]:

```
px.set_mapbox_access_token(open("../mapbox_token").read())
fig = px.scatter_mapbox(
    top_stations_data,
    lat="lat",
    lon="long",
    color="to_counter",
    text="name",
    size_max=15,
    zoom=11,
    title='Самые популярные станции прибытия'
)
fig.show()
```

executed in 765ms, finished 18:28:14 2020-05-09

Самые популярные станции прибытия



In [10]:

```
popular_trips = trips.groupby(['from_station_id', 'to_station_id']).size().res
popular_trips = popular_trips.rename(columns={0:'counter'})
```

executed in 52ms, finished 18:28:14 2020-05-09

In [11]:

```
top_trips = popular_trips.sort_values(by='counter', ascending=False).head(10)
top_trips
```

executed in 2.35s, finished 18:28:16 2020-05-09

Out[11]:

	from_station_id	to_station_id	counter
2866	WF-01	WF-01	4209
2868	WF-01	WF-04	2291
2940	WF-04	WF-01	1605
2942	WF-04	WF-04	1150
10	BT-01	CBD-13	1101
2844	WF-01	PS-05	962
478	CBD-13	BT-01	962
2062	SLU-17	SLU-17	960
896	CH-07	SLU-16	934
2368	UD-01	UD-01	880

Кажется, что достаточно распространенная практика - это взять велосипед на прогулку и вернуть его туда же.

In [12]:

```
top_trips = top_trips.merge(stations_data[['station_id', 'name']],
                            left_on='from_station_id',
                            right_on='station_id')

top_trips = top_trips.drop(['from_station_id'], axis=1)
top_trips = top_trips.rename(columns={'name':'from_station_name'})

top_trips = top_trips.merge(stations_data[['station_id', 'name']],
                left_on='to_station_id',
                right_on='station_id')

top_trips = top_trips.drop(['to_station_id'], axis=1)
top_trips = top_trips.rename(columns={'name':'to_station_name'})

top_trips
```

executed in 986ms, finished 18:28:17 2020-05-09

Out[12]:

	counter	station_id_x	from_station_name	station_id_y	to_station_name
0	4209	WF-01	Pier 69 / Alaskan Way & Clay St	WF-01	Pier 69 / Alaskan Way & Clay St
1	1605	WF-04	Seattle Aquarium / Alaskan Way S & Elliott Bay...	WF-01	Pier 69 / Alaskan Way & Clay St
2	2291	WF-01	Pier 69 / Alaskan Way & Clay St	WF-04	Seattle Aquarium / Alaskan Way S & Elliott Bay...
3	1150	WF-04	Seattle Aquarium / Alaskan Way S & Elliott Bay...	WF-04	Seattle Aquarium / Alaskan Way S & Elliott Bay...
4	962	WF-01	Pier 69 / Alaskan Way & Clay St	PS-05	King Street Station Plaza / 2nd Ave Extension ...
5	1101	BT-01	3rd Ave & Broad St	CBD-13	2nd Ave & Pine St
6	962	CBD-13	2nd Ave & Pine St	BT-01	3rd Ave & Broad St
7	960	SLU-17	Lake Union Park / Valley St & Boren Ave N	SLU-17	Lake Union Park / Valley St & Boren Ave N
8	934	CH-07	E Pine St & 16th Ave	SLU-16	Pine St & 9th Ave
9	880	UD-01	Burke-Gilman Trail / NE Blakeley St & 24th Ave NE	UD-01	Burke-Gilman Trail / NE Blakeley St & 24th Ave NE

Рассмотрим влияние погодных условий на количество поездок(как влияет на количество поездок дождь/ средняя температура и т.д., есть ли зависимость?)

In [13]:

```
trips['starttime'] = pd.to_datetime(trips['starttime'])
trips['stoptime'] = pd.to_datetime(trips['stoptime'])
trips['Date'] = pd.to_datetime(trips['starttime'].dt.date)
weather['Date']= pd.to_datetime(weather['Date'])
```

executed in 49.6s, finished 18:29:07 2020-05-09

In [14]:

```
num_trips_per_day = trips.groupby('Date').size().reset_index().rename(columns=
```

executed in 67ms, finished 18:29:07 2020-05-09

In [15]:

```
trips_and_weather = num_trips_per_day.merge(weather, on='Date')
```

executed in 87ms, finished 18:29:07 2020-05-09

Посмотрим какие события бывают в таблице с информацией о погоде.

In [16]:

```
trips_and_weather.Events.unique()
```

executed in 84ms, finished 18:29:07 2020-05-09

Out[16]:

```
array(['Rain', nan, 'Rain , Snow', 'Fog', 'Fog , Rain',
       'Rain , Thunderstorm', 'Fog-Rain', 'Snow', 'Rain-Thunderstorm',
       'Rain-Snow'], dtype=object)
```

В `Events` обычного дня, без событий типа `Rain` и т.д. проставим значение `Nothing` .

In [17]:

```
trips_and_weather['Events'] = trips_and_weather['Events'].fillna('Nothing')
```

executed in 95ms, finished 18:29:07 2020-05-09

In [18]:

```
correlations = trips_and_weather.drop(
    ['Date', 'Events'],
    axis=1
).corr(method='spearman')['trips_counter'].sort_values()[:-1]
correlations
```

executed in 192ms, finished 18:29:08 2020-05-09

Out[18]:

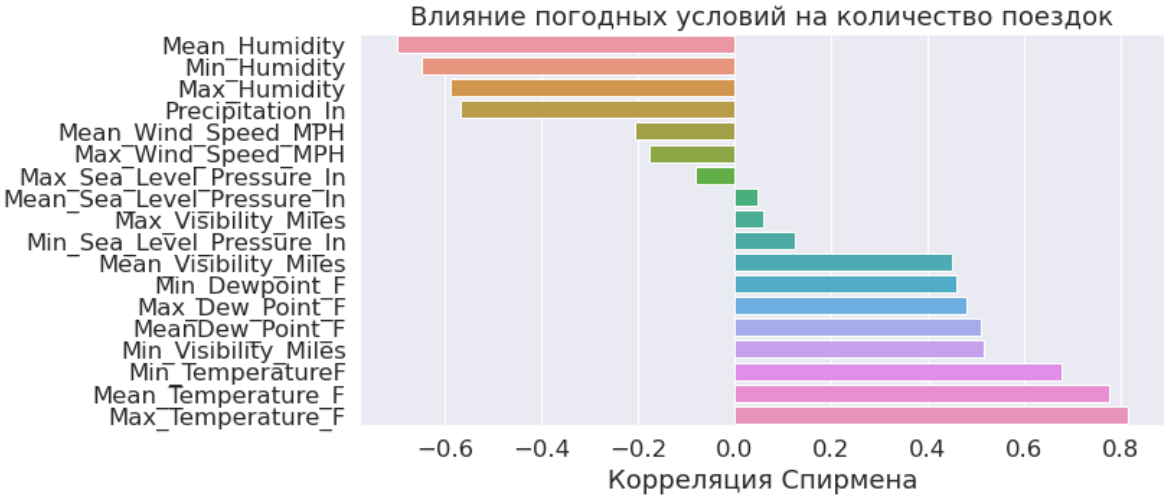
```
Mean_Humidity                -0.700554
Min_Humidity                 -0.647427
Max_Humidity                 -0.588857
Precipitation_In             -0.568086
Mean_Wind_Speed_MPH          -0.207843
Max_Wind_Speed_MPH           -0.175602
Max_Sea_Level_Pressure_In    -0.080565
Mean_Sea_Level_Pressure_In    0.045906
Max_Visibility_Miles          0.059613
Min_Sea_Level_Pressure_In     0.125072
Mean_Visibility_Miles         0.449956
Min_Dewpoint_F                0.460969
Max_Dew_Point_F               0.480049
MeanDew_Point_F               0.509847
Min_Visibility_Miles          0.515462
Min_TemperatureF              0.677961
Mean_Temperature_F            0.776625
Max_Temperature_F             0.815100
Name: trips_counter, dtype: float64
```

Визуализируем полученные корреляции.

In [19]:

```
plt.figure(figsize=(10, 5))
sns.barplot(y=correlations.index, x=correlations)
plt.title('Влияние погодных условий на количество поездок')
plt.xlabel('Корреляция Спирмена')
plt.show()
```

executed in 1.14s, finished 18:29:09 2020-05-09



Выводы:

- Видим, что есть зависимость между количеством поездок и, например, средней температуры, что, вообще говоря, можно было ожидать.
- Очень четко выделяются популярные станции отправления и прибытия, которые и образуют популярные маршурты.