Статистика, прикладной поток

Практическое задание 4

В данном задании вы потренируетесь работать с библиотекой pandas, посмотрите на свойства робастных оценок, а также реализуете приближенный поиск оценок максимального правдоподобия.

Правила:

- Дедлайн **5 ноября 23:59**. После дедлайна работы не принимаются кроме случаев наличия уважительной причины.
- Выполненную работу нужно отправить на почту `mipt.stats@yandex.ru`, указав тему письма "`[applied] Фамилия Имя - задание 4`". Квадратные скобки обязательны. Если письмо дошло, придет ответ от автоответчика.
- Прислать нужно ноутбук и его pdf-версию (без архивов). Названия файлов должны быть такими: `4.N.ipynb` и `4.N.pdf`, где `N` - ваш номер из таблицы с оценками.
- Решения, размещенные на каких-либо интернет-ресурсах не принимаются. Кроме того, публикация решения в открытом доступе может быть приравнена к предоставлении возможности списать.
- Для выполнения задания используйте этот ноутбук в качестве основы, ничего не удаляя из него.
- Никакой код из данного задания при проверке запускаться не будет.

Баллы за задание:

- Задача 1 - 20 баллов **О3**
- Задача 2 - 7 баллов **О3**
- Задача 3 - 5 баллов **О3**
- Задача 4 - 15 баллов **О2**
- Задача 5 - 15 баллов **О3**
- Задача 6 - 5 баллов **О3**

In [1]:

```
import numpy as np
import pandas as pd
import scipy.stats as sps
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

%matplotlib inline
```

Задача 1. Yelp

Yelp (yelp.com) — веб-сайт для поиска на местном рынке услуг, например ресторанов или парикмахерских, с возможностью добавлять и просматривать рейтинги и обзоры этих услуг. Для популярных бизнесов имеются сотни обзоров. Для обозревателей на сайте предусмотрены элементы социальной сети.

Вам предоставляется следующая информация о компаниях на Yelp:

Файл `yelp_business.csv`:

- `business_id` — уникальный идентификатор компании;
- `name` — имя компании;
- `address`, `city`, `state` — месторасположении компании;
- `latitude`, `longitude` — географические координаты;
- `categories` — категории услуг компании.

Файл `yelp_review.csv`, содержащий оценки пользователей:

- `business_id` — идентификатор компании, соответствующий файлу `yelp_business.csv`;
- `stars` — поставленная пользователем оценка от 1 до 5.

В целях сокращения объема файла, текстовые отзывы пользователей не были включены.

Оригинальную версию датасета в формате json можно посмотреть по ссылке https://www.kaggle.com/yelp-dataset/yelp-dataset/data (https://www.kaggle.com/yelp-dataset/yelp-dataset/data)

Задача:

- Найти город с наибольшим количеством компаний;
- Для этого города определить районы с наиболее качественными услугами. Пример с несколько другой задачей: https://yandex.ru/company/researches/2017/msk_mobile_map (https://yandex.ru/company/researches/2017/msk_mobile_map)
- А также найти рестораны с наилучшими отзывами.

In [2]:

```
business = pd.read_csv('yelp_business.csv', index_col=0)
```

Найдите пять городов, по которым присутствует информация о наибольшем количестве компаний. Для этого стоит воспользоваться методами `groupby`, `count`, `sort_values`, `head`. В таблице должен быть указан город (название) и количество компаний в этом городе.

In [4]:

```
top_cities = business.groupby('city').count().sort_values('name', ascending=False).head
()[[]]
top_cities
```

Out[4]:

city
Las Vegas
Phoenix
Toronto
Charlotte
Scottsdale

Пусть N -- город с наибольшим количеством компаний.

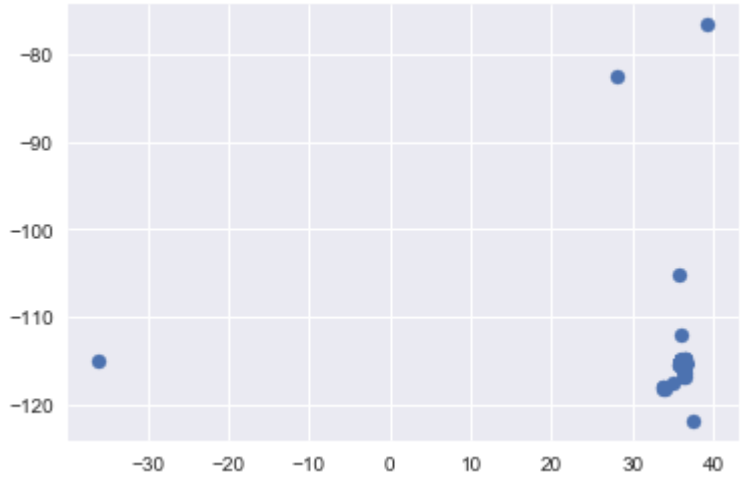
Оставьте в таблице только записи, соответствующие городу N. Нанесите все эти компании на график, в котором по оси x отметьте долготу, а по оси y -- долготу.

In [3]:

```
vegas = business[business.city == 'Las Vegas']
plt.scatter(vegas.latitude, vegas.longitude)
```

Out[3]:

```
<matplotlib.collections.PathCollection at 0x18b84e68c88>
```



Сам город находится в сгустке точек. Есть какие-то компании, которые приписаны к этому городу, но находятся далеко от него. Избавьтесь от них, подобрав некоторые границы значений широты и долготы. Изобразите все компании на новом графике.

На этом графике должны выделяться некоторые улицы. Откройте карту города N и сравните ее с построенным графиком.

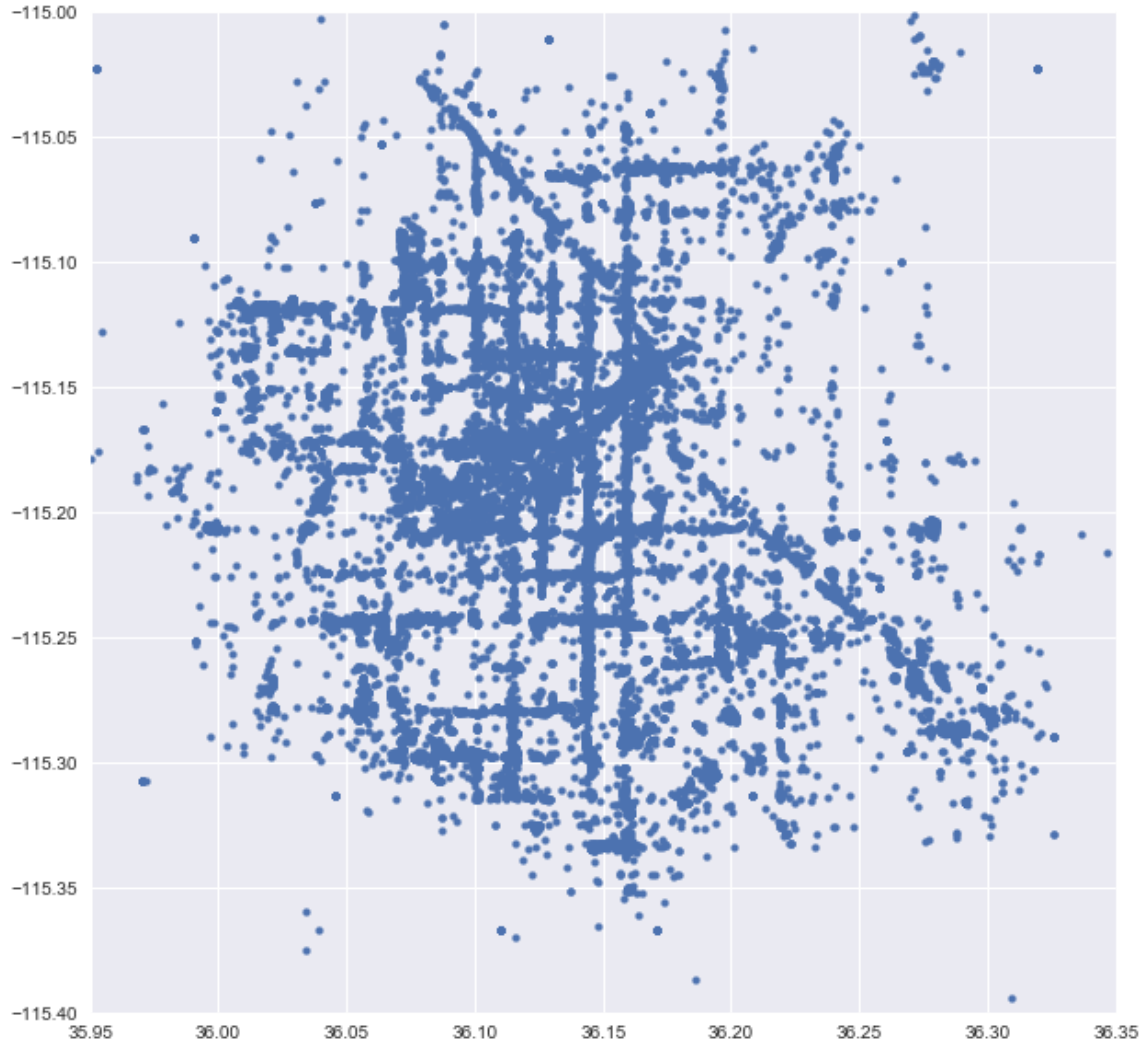
При желании вы можете разобраться с тем, как автоматически подгружать карту города в качестве фона графика.

In [34]:

```
plt.figure(figsize=(10, 10))
plt.scatter(vegas.latitude, vegas.longitude, s=15)
plt.xlim(35.95, 36.35)
plt.ylim(-115.4, -115)
```

Out[34]:

```
(-115.4, -115)
```



Оценки компаний

Для выполнения задания нужно посчитать среднюю оценку каждой компании, а также количество выставленных оценок.

Загрузите таблицу оценок `yelp_review.csv`.

In [4]:

```
review = pd.read_csv('yelp_review.csv', index_col=0)
```

```
C:\Users\1\Anaconda3\lib\site-packages\numpy\lib\arraysetops.py:472: Futur
eWarning: elementwise comparison failed; returning scalar instead, but in
the future will perform elementwise comparison
  mask |= (ar1 == a)
```

В подгруженной таблице оценок оставьте только компании города N. Для этого воспользуйтесь функцией `np.in1d(x, y)`, которая вернет массив того же размера, что и `x`, а на i-м месте будет `True`, если элемент `x[i]` встречается в `y`.

Внимание! Такая операция может выполняться довольно долго. После выполнения операции можно сохранить ее результат в файл, чтобы в дальнейшем не выполнять ее заново.

In [5]:

```
review = review[np.in1d(review['business_id'], vegas['business_id'])]
```

In [7]:

```
review.to_csv('vegas_review_new.csv')
```

Теперь посчитайте среднюю оценку каждой компании, а также количество выставленных компании оценок. Помочь в этом могут функции `groupby` и `aggregate([np.mean, np.size])`.

In [18]:

```
vegas_review = review.groupby('business_id').aggregate([np.mean, np.size])
```

Назовите колонки таблицы красивыми именами, изменив `<имя таблицы>.columns`, после чего напечатайте несколько строк полученной таблицы.

In [27]:

```
vegas_review.columns = ('stars', 'count')
vegas_review.head()
```

Out[27]:

	stars	count
business_id		
--9e1ONYQuAa-CB_Rrw7Tw	4.088904	1451
--DdmeR16TRb3LsjG0ejrQ	3.200000	5
--WsruI0lGEoeRmkErU5Gg	4.928571	14
--Y7NhBKzLTbNliMUX_wfg	4.875000	8
--e8PjCNhEz32pprnPhCwQ	3.473684	19

Соедините две полученные ранее таблицы по компаниям города N в одну. Для этого сначала установите поле `business_id` в качестве индекса в обеих таблицах с помощью `set_index` (в одной из них это уже должно было быть сделано). Соединение таблиц можно выполнить с помощью `join`. Индексы у этих таблиц одинаковые, так что тип джойна не имеет значения. В полученной таблице должны получится поля `latitude`, `longitude`, `categories`, `name`, `stars`, `count`.

In [31]:

```
#vegas = vegas.set_index('business_id')[['latitude', 'longitude', 'categories', 'nam
e']]
joined = vegas.join(vegas_review)
joined.head()
```

Out[31]:

	latitude	longitude	categories	na
business_id				
kCoE3jvEtg6UVz5SOD3GVw	36.207430	-115.268460	Real Estate Services;Real Estate;Home Services...	"BDJ Realty'
OD2hnuuTJl9uotcKycxg1A	36.197484	-115.249660	Shopping;Sporting Goods	"Socce Zone"
VBHEsoXQb2AQ76J9l8h1uQ	36.085051	-115.119421	Shopping;Jewelry;Watch Repair;Local Services	"Alfred Jewelr
1Jp_hmPNUZArNqzpbm7B0g	36.056382	-115.269332	Home Services;Lighting Fixtures & Equipment;Lo...	"Task Electri
DPQnTnNw2PJj7DdENM98Cw	36.105196	-115.056880	Nurseries & Gardening;Home & Garden;Shopping	"Star Nurser

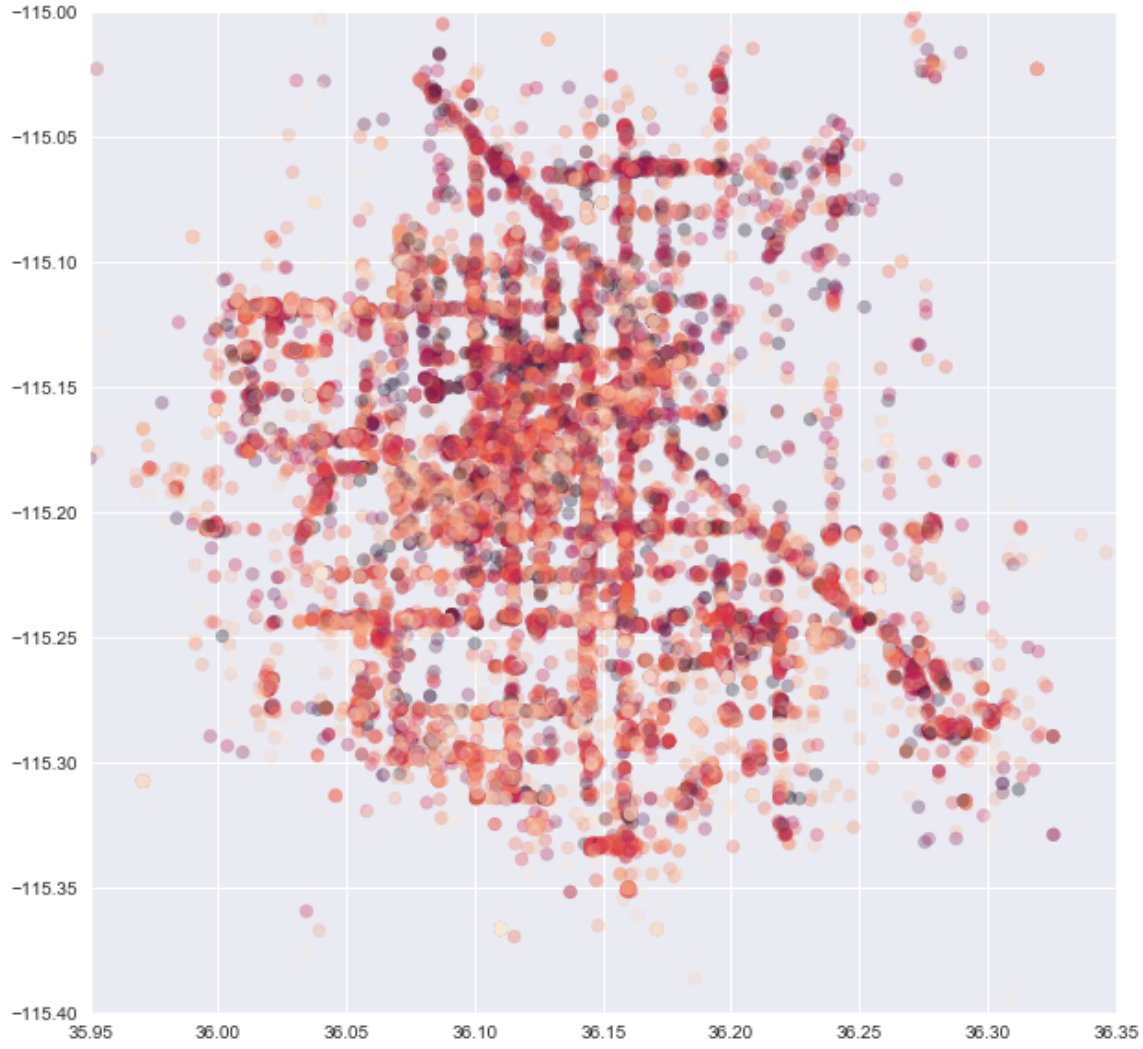
Изобразите все компании на графике, раскрасив точку в цвет, оттенок которого соответствует средней оценке компании. Прозрачность точки выставляйте не более 0.3.

In [35]:

```
plt.figure(figsize=(10, 10))
plt.scatter(joined.latitude, joined.longitude, alpha=0.3, c=joined.stars)
plt.xlim(35.95, 36.35)
plt.ylim(-115.4, -115)
```

Out[35]:

```
(-115.4, -115)
```



Чтобы получить районы города, округлите значения широты и долготы, подобрав оптимальный размер района. Например, можно сделать так `np.round(долгота*4, decimals=1)*0.25`.

In [37]:

```
joined['x'] = np.round(joined.latitude*4, decimals=1)*0.25
joined['y'] = np.round(joined.longitude*4, decimals=1)*0.25
```

Для получения средней оценки компании по району постройте сводную таблицу при помощи `pd.pivot_table`, взяв в качестве индексов и колонок округленные широту и долготу, а в качестве значений -- оценки. Аггрегирующей функцией является среднее.

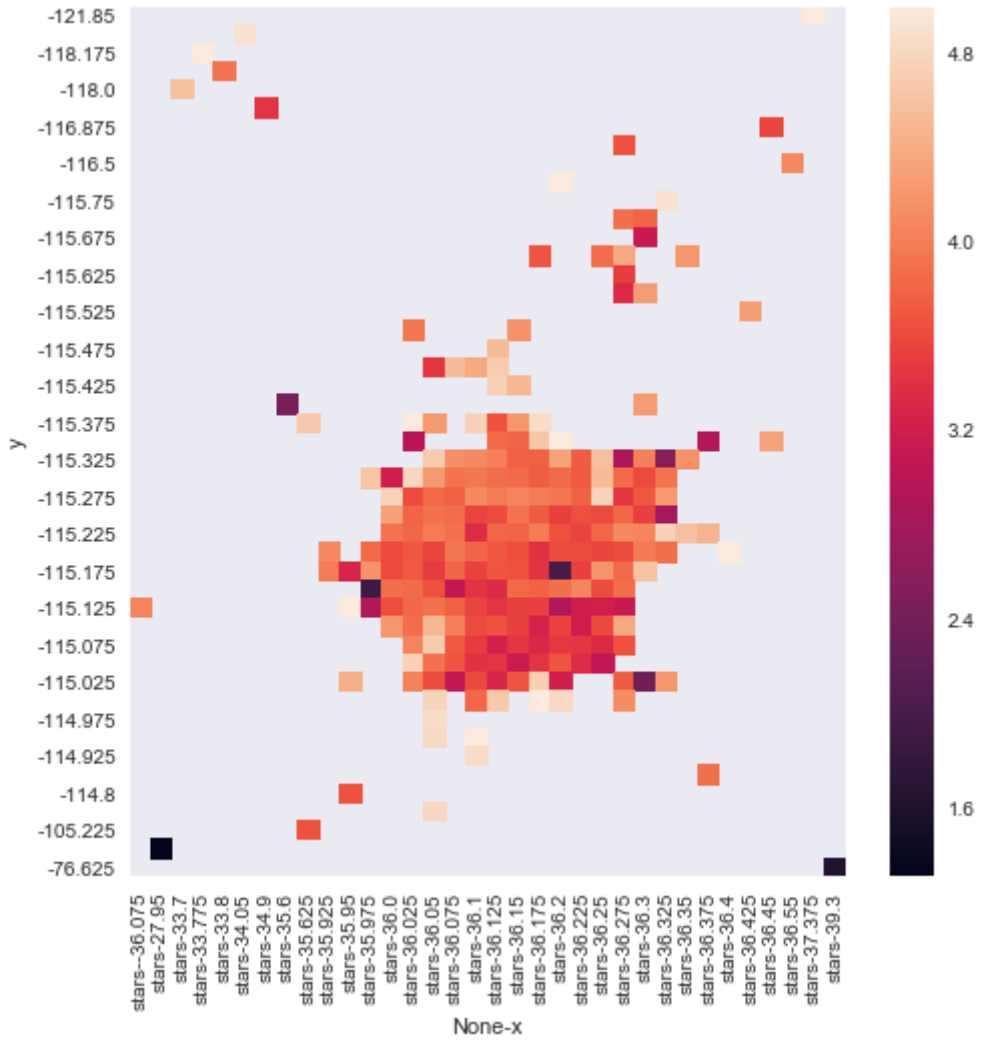
Изобразите полученную таблицу при помощи `sns.heatmap`.

In [88]:

```
pivot_stars = pd.pivot_table(joined, values=['stars'], index=['y'], columns=['x'], aggf
unc=np.mean)

plt.figure(figsize=(8,8))
sns.heatmap(pivot_stars)
```

Out[88]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x18b8da0fba8>
```



Полученный график имеет ряд недостатков. Во-первых, не очень правильно судить о районе, если в нем мало компаний. Во-вторых, на графике цветовая гамма автоматически подстроилась под минимальное и максимальное значения оценки.

Почему эти недостатки могут быть существенными?

Ответ: Если в районе мало компаний, то его оценки могут сильно разниться, то есть почти наверняка лучший и худший район будет иметь мало оценок. Также информация о непопулярном районе просто не интересна.

Если цветовая схема автоматически подстраивается под максимальное и минимальное значение, то для сложно сравнивать оценки для разных городов.

Оставьте районы, в которых имеется информация о не менее 30 компаний. Постройте новый график районов, использовав параметры `vmin` и `vmax` у функции `sns.heatmap`.

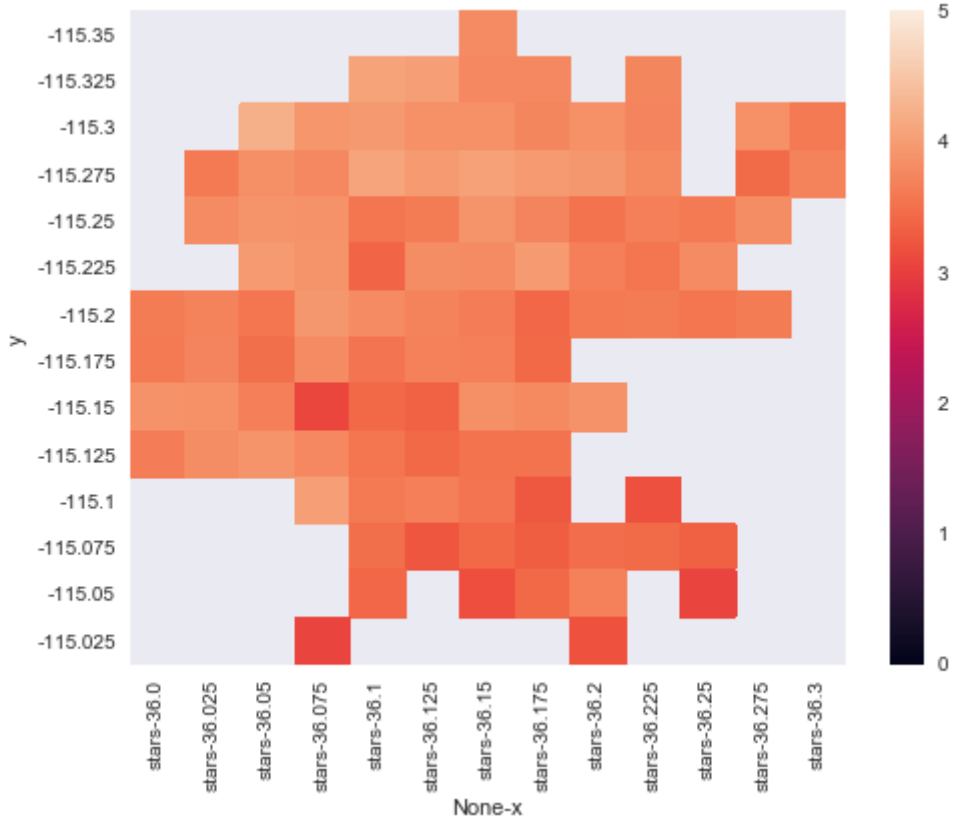
In [94]:

```
def aggfunc(x):
    if np.count_nonzero(x)>=30:
        return np.mean(x)
    return float('NaN')

new_pivot_stars = pd.pivot_table(joined, values=['stars'], index=['y'], columns=['x'],
aggfunc=aggfunc)
new_pivot_stars = new_pivot_stars.dropna(axis=(0,1), how='all')

plt.figure(figsize=(8, 6))
sns.heatmap(new_pivot_stars, vmin=0, vmax=5)
```

```
C:\Users\1\Anaconda3\lib\site-packages\ipykernel_launcher.py:7: FutureWarn
ing: supplying multiple axes to axis is deprecated and will be removed in
a future version.
  import sys
```

Out[94]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x18b8ff75860>
```



Сравните полученный график с предыдущим и сделайте вывод.

Вывод: По второму графику легче увидеть, что все оценки в Лас Вегасе примерно одинаковые и все высокие.

Рестораны

Будем считать компанию рестораном, если в поле `categories` *содержится* слово `Restaurant`. Составьте таблицу, в которой будет информация о всех ресторанах города N, для которых имеется не менее 5 отзывов. Далее постройте график районов, в котором каждому району сопоставьте среднюю оценку по ресторанам этого района. Рассматривайте только те районы, в которых есть не менее 10 ресторанов, для каждого из которых есть не менее 5 отзывов.

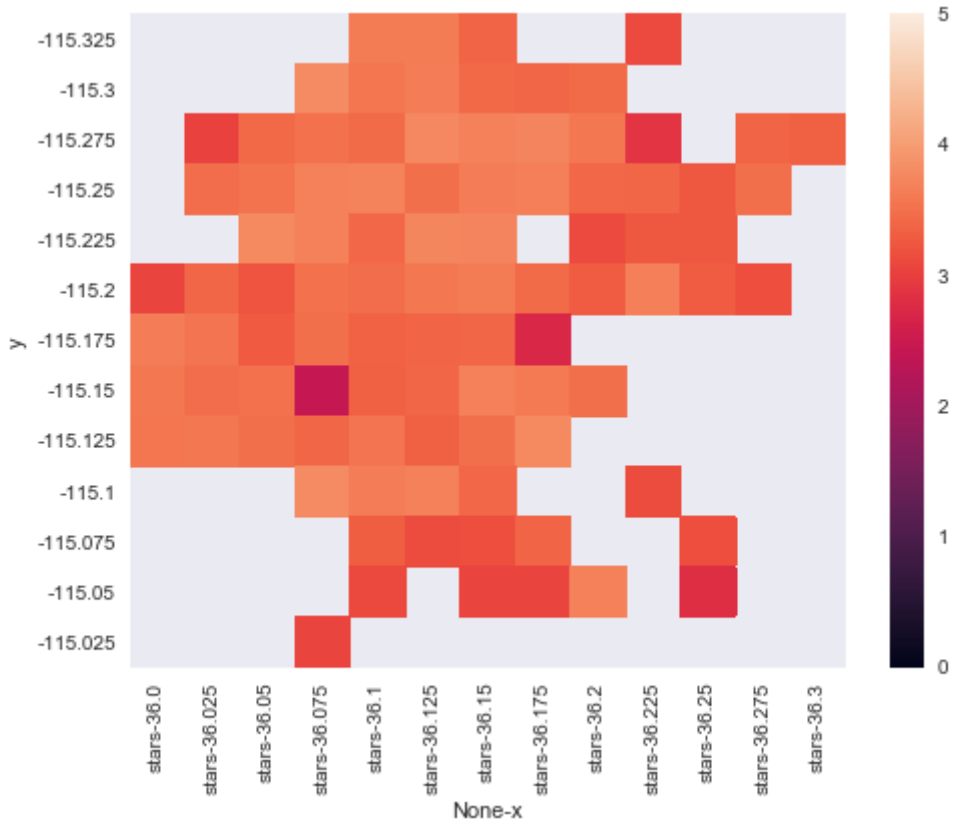
In [114]:

```
func = np.vectorize(lambda x: 'Restaurant' in x)
restaurants = joined[(func(joined.categories)) & (joined['count'] >= 5)]

def restaurant_func(x):
    if np.count_nonzero(x) >= 10:
        return np.mean(x)
    return float('NaN')

districts = pd.pivot_table(restaurants, values=['stars'], index=['y'], columns=['x'], a
ggfunc=restaurant_func)
districts = districts.dropna(axis=(0,1), how='all')

plt.figure(figsize=(8, 6))
sns.heatmap(districts, vmin=0, vmax=5)
```

```
C:\Users\1\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: FutureWar
ning: supplying multiple axes to axis is deprecated and will be removed in
a future version.
  # Remove the CWD from sys.path while we load stuff.
```

Out[114]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x18b8d9112b0>
```



Чем полезны ограничения снизу на количество отзывов для ресторана и количество ресторанов в районе?

Ответ: Ограничения снизу на количество отзывов полены тем, что они убирают маленькие выборки, которые не репрезентативны и часто оказываются выбросами.

Кот Василий из таблицы с баллами очень придирчив к выбору ресторана. Он доверяет только ресторанам с высоким рейтингом, который основывается на большом количестве отзывов. Напечатайте в виде таблицы информацию 10 ресторанах с самым большим рейтингом в порядке убывания рейтинга. Для каждого из этих ресторанов должно быть не менее 50 отзывов. По каждому ресторану необходимо вывести следующую информации: название ресторана, средняя оценка, количество отзывов, географические координаты, категории.

In [124]:

```
for_Vasya = restaurants[restaurants['count'] >= 50]
for_Vasya = for_Vasya.sort_values('stars', ascending=False).head(10)
for_Vasya = for_Vasya.set_index('name')
for_Vasya[['stars', 'count', 'latitude', 'longitude', 'categories']]
```

Out[124]:

name	stars	count	latitude	longitude	cat
"Lip Smacking Foodie Tours"	4.966480	179	36.114537	-115.172678	Food Tours;Restaurants;Event Plar Servic...
"Pepito Shack"	4.907692	65	36.152477	-115.151945	Restaurants;Burgers;Food Stands;Sandwiches;Hot...
"Bosa Boba Cafe"	4.890909	55	36.125960	-115.184846	Vietnamese;Bubble Tea;Sandwiches;Food;Coffee &...
"Garden Grill"	4.868132	91	36.166783	-115.286197	Tacos;Street Vendors;Farmers Market;Vegetarian...
"Brew Tea Bar"	4.848069	1165	36.054195	-115.242443	Cafes;Tea Rooms;Food;Bubble Tea;Restaurants;De...
"Poppa Naps BBQ"	4.836538	104	36.116549	-115.088115	Food Stands;Hot Dogs;Caterers;Restaurants;Amer...
"Zenaida's Cafe"	4.833333	180	36.101741	-115.100359	Restaurants;Breakfast & Brunch;Ca
"El Frescos Cocina Mexicana"	4.816754	191	36.098527	-115.148446	Caterers;Mexican;Restaurants;Foo Planni...
"Blaqcat Ultra Hookah Lounge"	4.809524	63	36.159742	-115.232738	Adult Entertainment;Lounges;Hook Bars;Restau...
"California Sushi Burrito"	4.807018	57	36.125636	-115.202487	Asian Fusion;Fast Food;Restaurants;Japanese;Po...

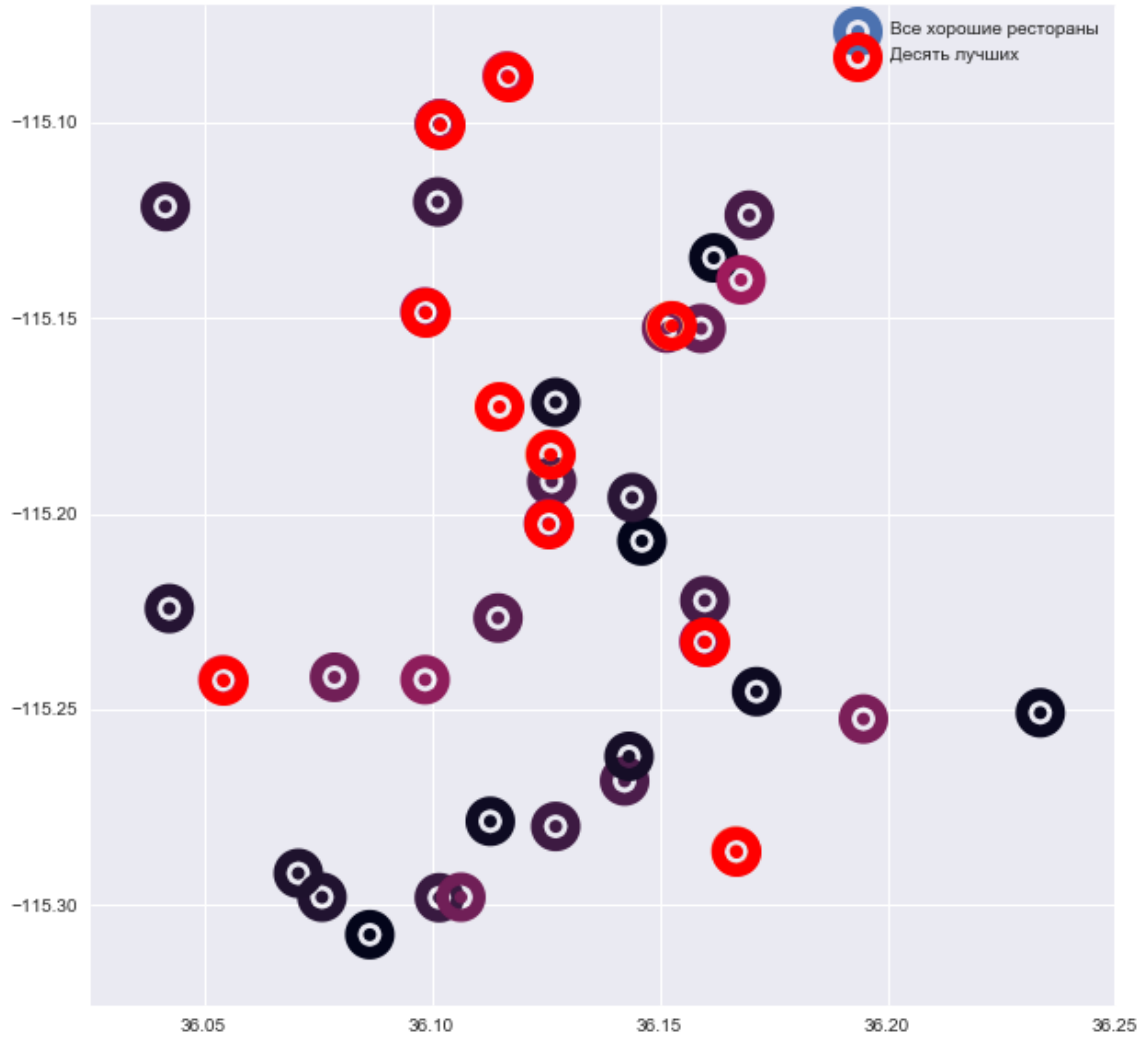
Нанесите на карту все рестораны со средней оценкой не менее 4.7, которая посчитана по не менее 50 отзывам. Отдельным цветом отметьте 10 ресторанов, которые вы получили ранее.

In [138]:

```
top_resturants = restaurants[(restaurants['stars'] >= 4.7) & (restaurants['count'] >= 5
0)]

plt.figure(figsize=(10, 10))
plt.scatter(top_resturants.latitude, top_resturants.longitude, c=top_resturants.stars,
linewidths=20,
            label='Все хорошие рестораны')
plt.scatter(for_Vasya.latitude, for_Vasya.longitude, c='r', linewidth=20, label='Десять
 лучших')
plt.legend()
```

Out[138]:

```
<matplotlib.legend.Legend at 0x18b90c38b00>
```



Охарактеризуйте кота Василия, а также сделайте общий вывод по задаче.

Вывод: Кот Василий весьма придирчив. Я научилась строить графики и таблицы для таких котов.

Задача 2. Airquality

Загрузите с помощью `pandas` из файла `airquality.csv` данные о качестве воздуха в Нью-Йорке с мая по сентябрь 1973 года по дням. Данные содержат измерения нескольких величин, описания которых можно прочитать по ссылке
https://www.rdocumentation.org/packages/datasets/versions/3.5.1/topics/airquality
(https://www.rdocumentation.org/packages/datasets/versions/3.5.1/topics/airquality)

In [139]:

```
airquality = pd.read_csv('airquality.csv')
```

Выведите описательные статистики (метод `describe`)

In [140]:

```
airquality.describe()
```

Out[140]:

	Ozone	Solar.R	Wind	Temp	Month	Day
count	116.000000	146.000000	153.000000	153.000000	153.000000	153.000000
mean	42.129310	185.931507	9.957516	77.882353	6.993464	15.803922
std	32.987885	90.058422	3.523001	9.465270	1.416522	8.864520
min	1.000000	7.000000	1.700000	56.000000	5.000000	1.000000
25%	18.000000	115.750000	7.400000	72.000000	6.000000	8.000000
50%	31.500000	205.000000	9.700000	79.000000	7.000000	16.000000
75%	63.250000	258.750000	11.500000	85.000000	8.000000	23.000000
max	168.000000	334.000000	20.700000	97.000000	9.000000	31.000000

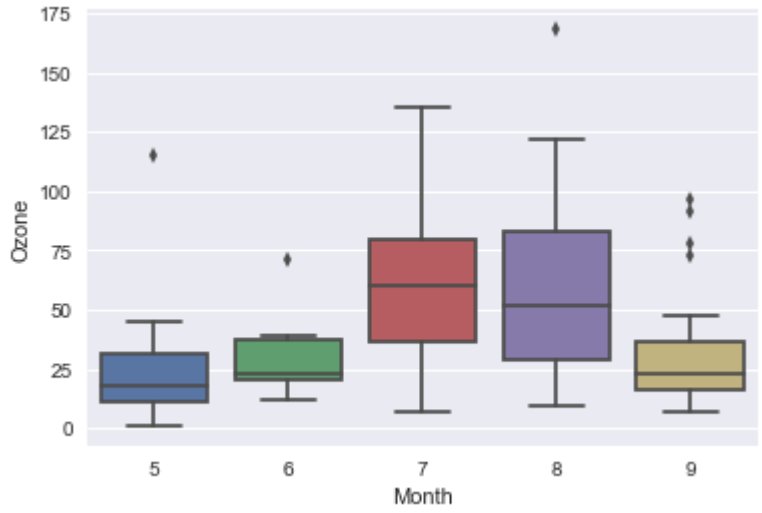
Что можно сказать о наличии в данных выбросов, сравнивая выборочную медиану и выборочное среднее?

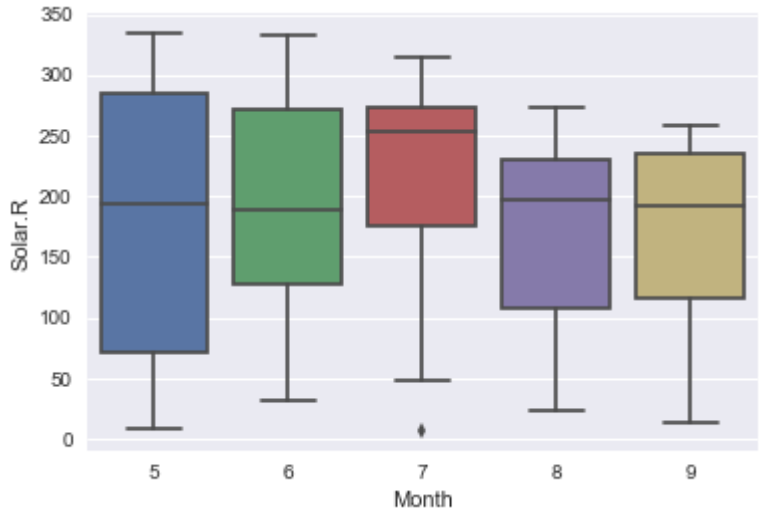
Ответ: На медиану слабо влияют выбросы, а на среднее влияют сильно. Теоретически медиана и среднее должны быть равны. Их сильное различие это свидетельство наличия большого количества выбросов.

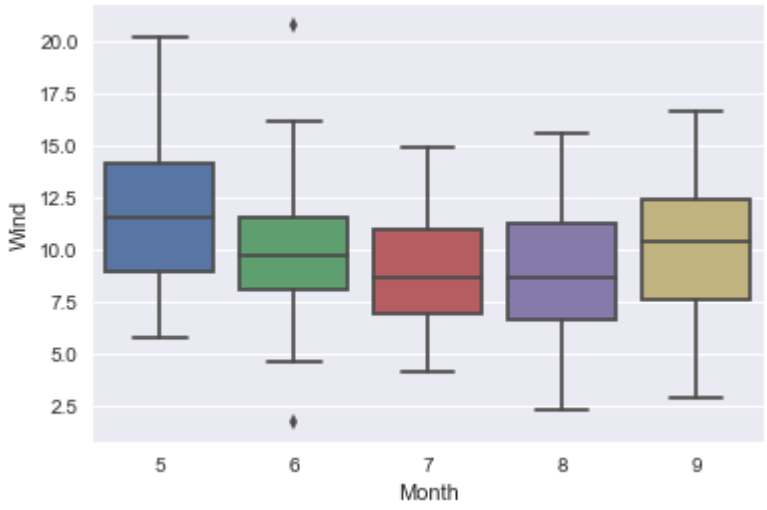
Для каждого параметра постройте график ящиков с усами (boxplot), в котором ось x соответствует номеру месяца, а ось y --- значениям параметра (т.е. свой ящик для каждого месяца). Используйте функцию `seaborn.boxplot`.

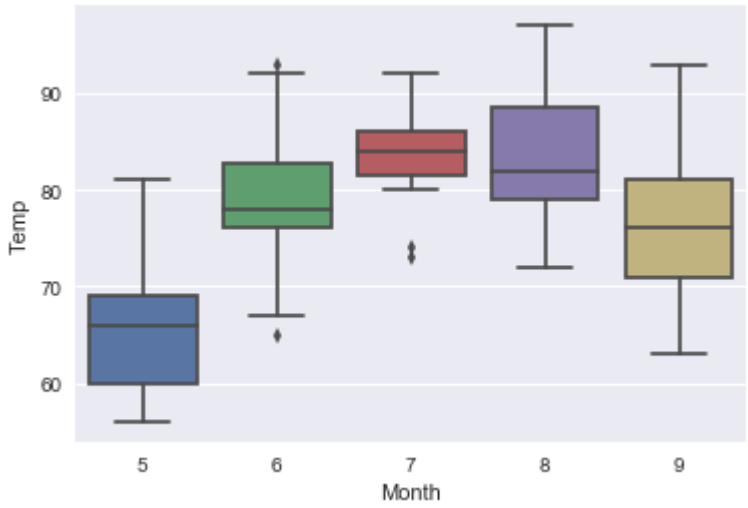
In [141]:

```
for param in ['Ozone', 'Solar.R', 'Wind', 'Temp']:
    sns.boxplot(x='Month', y=param, data=airquality)
    plt.show()
```


Ozone
175
150
125
100
75
50
25
0
5
6
7
8
9
Month


Solar.R
350
300
250
200
150
100
50
0
5
6
7
8
9
Month


Wind
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
5
6
7
8
9
Month



Какую информацию показывают ящики с усами? Какие выводы можно сделать в данном случае?

Ответ: ящик с усами показывает интервал, в котором скорее всего находятся значения. Так же он показывает выбросы. По графику озона можно увидеть, что данные по нему содержат много выбросов. Среди других характерсик выбросов намного меньше.

Для параметра с наибольшим числом наблюдений, признанных выбросами, сравните значения выборочного среднего, выборочной медианы и медианы средних Уолша.

In [143]:

```
ozone = airquality['Ozone']
n = len(ozone)
u, v = np.meshgrid(ozone, ozone)

Uolsh_median = np.nanmedian(u+v)/2

print('Среднее: %f, медиана: %f, медиана средних Уолша: %f' % (ozone.mean(), ozone.medi
an(), Uolsh_median))
```

```
Среднее: 42.129310, медиана: 31.500000, медиана средних Уолша: 38.500000
```

При подсчете этих статистик обычно предполагается, что наблюдения независимы. Выполено ли это свойство в данном случае?

Ответ: Наблюдения погоды не независимы, потому что наблюдения сегодня зависят от вчерашних наблюдений.

Вывод: Непонятно.

Задача 3. Laplace

Предлагается изучить некоторые свойства распределения Лапласа с параметром сдвига θ, обладающего плотностью распределения $p_\theta(x) = \frac{1}{2}e^{-|x-\theta|}$.

1. На отрезке $[-4, 4]$ постройте плотность стандартного нормального распределения и стандартного распределения Лапласа ($\theta = 0$). Не забудьте добавить легенду.

Решение:

In [ ]:

```
<...>
```

Вывод: <...>

2. Постройте график зависимости асимптотической дисперсии σ^2_α усеченного среднего $\overline{X}_\alpha$, $0 < \alpha < 1/2$, для распределения Лапласа. Помочь в ее вычислении может теорема, упомянутая на лекциях.

Является ли эта функция монотонной? Найдите пределы функции при $\alpha \to +0$ и $\alpha \to 1/2 - 0$. Сравните со значениями асимптотической дисперсии для выборочного среднего и выборочной медианы (не забудьте отметить их на графике). Сделайте вывод.

Решение:

In [ ]:

```
<...>
```

Вывод: <...>

3. Сгенерируйте выборку $X = (X_1, \ldots, X_{1000})$ из стандартного распределения Лапласа. Для всех $n \leqslant 1000$ по первым n элементам выборки $X_1, \ldots, X_n$ вычислите значения следующих оценок:

- $\overline{X}$ — выборочное среднее;
- $\widehat{\mu}$ — выборочная медиана;
- W — медиана по всем значениям $Y_{ij} = \frac{X_i + X_j}{2}, 1 \leq i \leq j \leq n$ — медиана средних Уолша.

На одном графике изобразите зависимость значений этих оценок от n. Настройте видимую область графика по оси y так, чтобы четко была отображена информативная часть графика. Сделайте вывод.

Решение:

In [ ]:

```
<...>
```

Вывод: <...>

Задача 4. Gamma-cats (Cauchy)

Предлагается изучить некоторые свойства распределения Коши с параметром сдвига θ, обладающего плотностью распределения $p_\theta(x) = \frac{1}{\pi(1+(x-\theta)^2)}$.

Замечание: Такое распределение встречается, к примеру, в следующей задаче. На высоте 1 метр от точки θ находится источник γ-излучения, причем направления траекторий γ-квантов случайны, т.е. равномерно распределены по полуокружности. Тогда $X_i, i = 1, \ldots, n$ — зарегистрированные координаты точек пересечения γ-квантов с поверхностью детекторной плоскости — образуют выборку из распределения Коши со сдвигом θ.

1. На отрезке $[-7, 7]$ постройте плотность стандартного нормального распределения и стандартного распределения Коши. Не забудьте добавить легенду.

Решение:

In [ ]:

```
<...>
```

Вывод: <...>

2. Постройте график зависимости асимптотической дисперсии σ^2_α усеченного среднего $\overline{X}_\alpha$, $0 < \alpha < 1/2$, для распределения Коши. Помочь в ее вычислении может теорема, упомянутая на лекциях.

Настройте видимую область графика по оси *y* так, чтобы четко была отображена информативная часть графика. Отметьте минимум функции.

Решение:

In [ ]:

```
<...>
```

При каком значении α асимптотическая дисперсия σ^2_α минимальна и чему она равна?

Ответ: <...>

3. Сгенерируйте выборку $X = (X_1, \ldots, X_{1000})$ из стандартного распределения Коши. Для всех $n \leqslant 1000$ по первым n элементам выборки $X_1, \ldots, X_n$ вычислите значения следующих оценок:

- $\overline{X}$ — выборочное среднее;
- $\overline{X}_\alpha$ — усеченное среднее, где α — значение, на котором достигается минимум σ^2_α;
- $\widehat{\mu}$ — выборочная медиана;
- W — медиана по всем значениям $Y_{ij} = \frac{X_i + X_j}{2}, 1 \leq i \leq j \leq n$ — медиана средних Уолша;

а также, по каждой из этих оценок, одношаговую оценку.

Напоминание: если $\hat{\theta}_0$ — асимптотически нормальная оценка, то одношаговая оценка $\hat{\theta}_1$ вычисляется как $\hat{\theta}_1 = \hat{\theta}_0 - \left(l''_X(\hat{\theta}_0)\right)^{-1} l'_X(\hat{\theta}_0)$, где $l_X(\theta)$ — логарифмическая функция правдоподобия. Заметим, что обычное выборочное среднее не является асимптотически нормальной оценкой, и оценка, вычисленная по формуле выше, формально не является одношаговой, однако ее все равно требуется посчитать.

На одном графике изобразите зависимость значений этих оценок от n. Для каждой оценки $\hat{\theta}_0$ соответствующая оценка $\hat{\theta}_1$ должна быть изображена на графике пунктиром тем же цветом, что и $\hat{\theta}_0$. Сделайте вывод.

Замечание: если некоторые оценки имеют большой разброс, и разница между графиками зависимостей оценок с малыми значениями недостаточно заметна, стоит сделать два графика, на одном из которых будут изображены все оценки, а на втором — только достаточно хорошие.

Решение:

In [ ]:

```
<...>
```

Вывод: <...>

Задача 5. Baltic macoma (Zero-inflated Poisson)

Пуассоновское распределение обычно используется для моделирования количества событий в некоторый отрезок времени или для моделирования количества объектов в некоторой области в предположении, что события или объекты появляются случайно и независимо. В курсе случайных процессов мы изучим пуассоновские процессы и поймем их связь с экспоненциальным распределением.

Пуассоновское распределение, завышенное в нуле (zero-inflated Poisson distribution), используется для моделирования случаев, в которых наблюдается завышенное содержание нулевых исходов. Например, число страховых исков в рамках населения будет иметь завышенное в нуле распределение из-за наличия тех людей, которые не оформили страховку.

Рассмотрим данные о колличествах балтийской макомы -- вид морских двустворчатых моллюсков из семейства теллинид, распространенного в северной части Атлантического и Тихого океана. В результате проведенных исследований оказалось, что во многих локациях численность видов равна нулю, поэтому стоит ожидать, что данные имеют пуассоновское распределение, завышенное в нуле.

Загрузите данные из файла `macoma.csv` и выберите столбец `macoma`. Постройте по данным гистограмму.

Внимание! Поскольку распределение дискретно, бины гистограммы должны соответствовать значениям величины. Для этого воспользуйтесь функцией `plt.hist(sample, range=(0, N), bins=N)`.

In [ ]:

```
<...>
```

Из теоретического домашнего задания вам известен метод поиска оценки максимального правдоподобия параметров распределения. Выпишите готовые формулы:

Ответ: <...>

Реализуйте метод для выданных данных. Постройте графики траекторий значений параметров в зависимости от номера итерации метода.

In [ ]:

```
<...>
```

Чтобы убедиться, что вы нашли правильное решение, посчитайте значения логарифмической функции правдоподобия по двумерной сетке значений параметров (ε, λ) и найдите максимум, используя функцию `cool_argmax` из предыдущего задания. Сравните его со значением, найденным ранее.

<...>

Распределение с подобранными параметрами сравните с гистограммой. На какой итерации визуально приближение получается наилучшим?

In [ ]:

```
<...>
```

Ответ: <...>

Почему при увеличении количества итерации получается плохое приближение? Предложите способ исправить этот недочет и реализуйте его.

Описание решения: <...>

Реализация решения:

In [ ]:

```
<...>
```

Ответ (значения оценок параметров): <...>

Вывод: <...>

Задача 6. Image Denoising

В качестве параллельного практического задания вам предлагаются конкурсы на Kaggle по восстановлению зашумленных фотографий семинаристов. Перечислите недостатки предлагаемых подходов и способа оценки качества восстановленного изображения.

Ответ: <...>