TALLINN AND FOURSQUARE

Raivo Kartau

This projects tries to show that the Foursquare Api is not something that one should use in programs/apps etc.

Tallinn:

Area	
• **Total**	159.2 km^2 (61.5 sq mi)
Elevation	9 m (30 ft)
Population (2020)[1]	
• **Total**	437,619
• **Rank**	1st in Estonia
• **Density**	2,700/km^2 (7,100/sq mi)

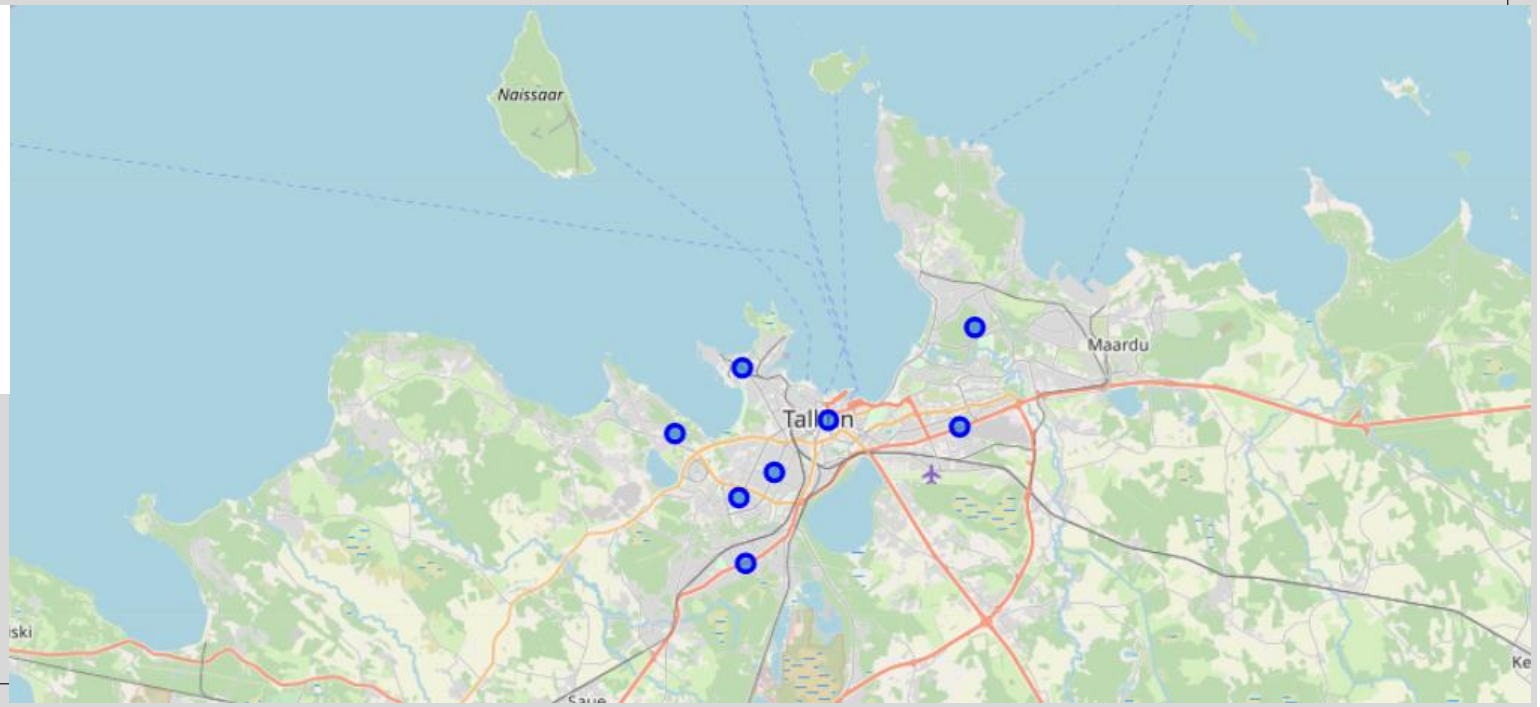
Location within Europe
Show map of Europe

Tallinn - Wikipedia

Borough in Tallinn

I remind you this is a city of almost 500 000 people living in it…

	Borough	Neighborhood	Latitude	Longitude
0	Tallinn	Haabersti	59.431328	24.629042
1	Tallinn	Kesklinn	59.436960	24.753530
2	Tallinn	Kristiine	59.415000	24.709150
3	Tallinn	Lasnamäe	59.433600	24.860170
4	Tallinn	Mustamäe	59.404820	24.681190



After setting the limit of venues to 999 in Tallinn

```
#The correct answer is:
LIMIT = 999 # limit of number of venues returned by Foursquare API
radius = 55000 # define radius
# create URL
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    neighborhood_latitude,
    neighborhood_longitude,
    radius,
    LIMIT)
url # display URL
```

```
'https://api.foursquare.com/v2/venues/explore?&client_id=A42H3MCRDBJ0UJ0LQREYLJNKB2R1AQOM0FT1NUHWY5AGLVF5&client_secret=PKVYIP2SRSATJBW0RGM1Z0CEZ1VZFSJ5I5EUOMQ23DJCZHXX&v=20180605&ll=59.4313282,24.6290422&radius=55000&limit=999'
```

```
[77]: results = requests.get(url).json()
      results
```


```
[77]: {'meta': {'code': 200, 'requestId': '60409e00add52e5ee2b99d68'},
       'response': {'suggestedFilters': {'header': 'Tap to show:',
         'filters': [{'name': 'Open now', 'key': 'openNow'}]},
        'headerLocation': 'Tallinn',
        'headerFullLocation': 'Tallinn',
        'headerLocationGranularity': 'city',
        'totalResults': 174,
        'suggestedBounds': {'ne': {'lat': 59.9263286950005,
          'lng': 25.60054187461796},
         'sw': {'lat': 58.936327704999506, 'lng': 23.65754252538204}},
        'groups': [{'type': 'Recommended Places',
          'name': 'recommended',
```


```
results = requests.get(url).json()
results
results['response'] ['totalResults']
```

```
174
```

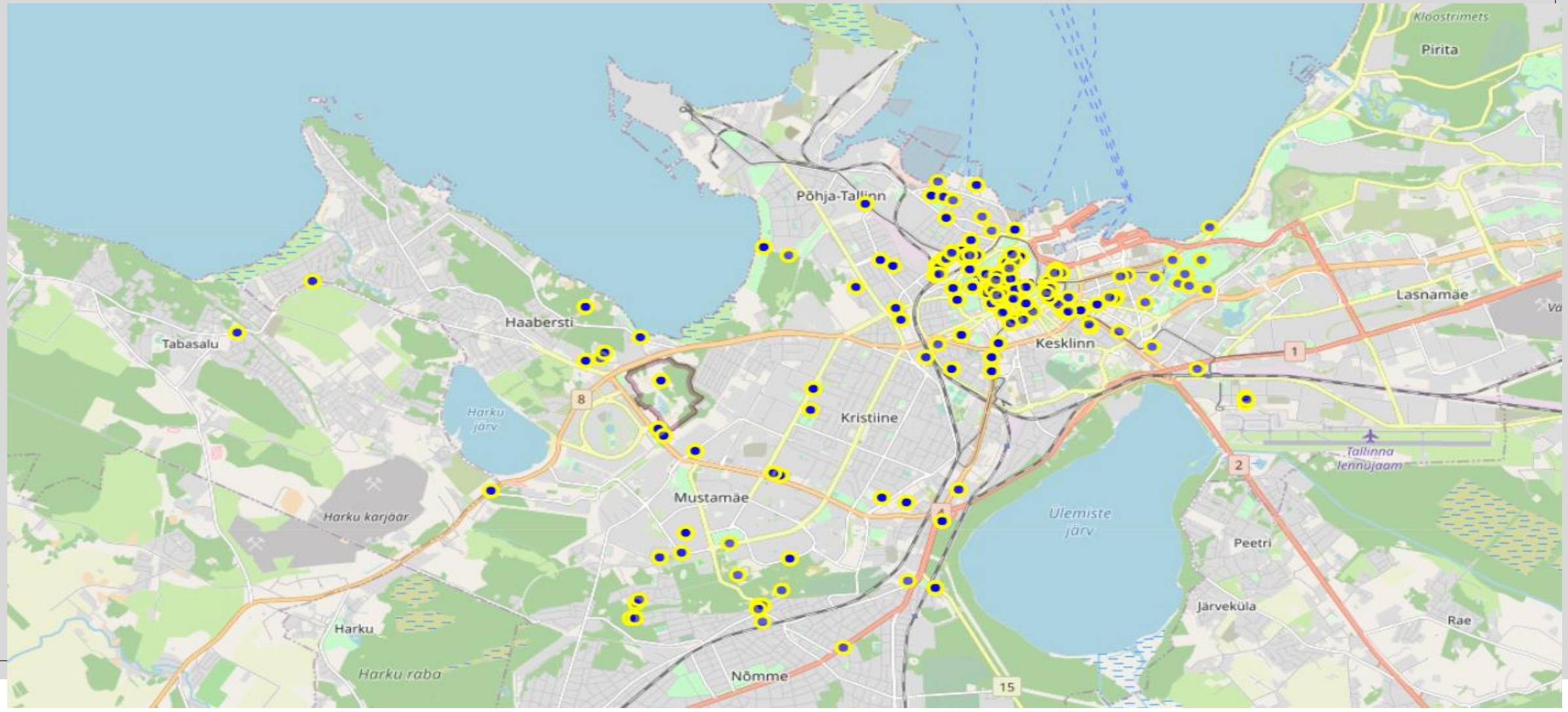
After Consolidating the Venues to Borough it was clear that the data is crooked towards central city= Kesklinn

[48]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Haabersti	7	7	7	7	7	7
Kesklinn	92	92	92	92	92	92
Kristiine	10	10	10	10	10	10
Lasnamäe	5	5	5	5	5	5
Mustamäe	8	8	8	8	8	8
Nõmme	4	4	4	4	4	4
Pirita	3	3	3	3	3	3
Põhja-Tallinn	6	6	6	6	6	6

```
There are 76 uniques categories.
```

On map it looks like this



I still analysed the crooked info based on area and the venue categories:

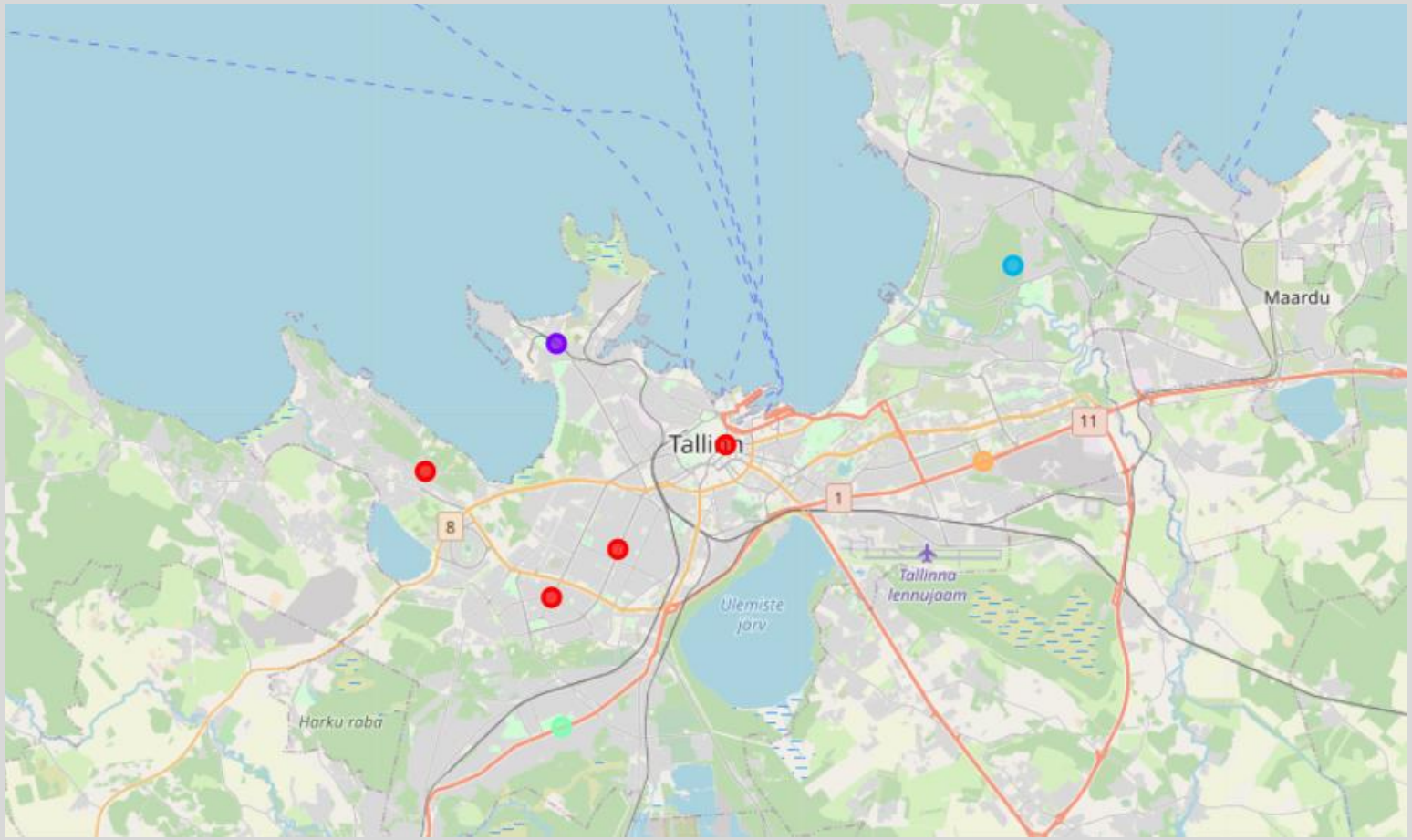
- The venue type based on a location

	Neighborhood	Asian Restaurant	Athletics & Sports	Auto Garage	Bar	Basketball Court	Beer Bar	Bistro	Bookstore	Boutique	Bus Line	Bus Station	Bus Stop	Cafeteria	Café	Candy Store	Cheese Shop	Cocktail Bar	Coffee Shop	Concert Hall	Convenience Store	Cosmetics Shop	Cupcake Shop	Dance Studio	Department Store	Dessert Shop	Dur Rest
0	Haabersti	0.000000	0.000000	0.0	0.000000	0.000000	0.00000	0.00000	0.000000	0.00000	0.142857	0.142857	0.142857	0.00000	0.000000	0.00000	0.00000	0.000000	0.00000	0.000000	0.142857	0.000000	0.00000	0.00000	0.00000	0.00000	0
1	Kesklinn	0.032609	0.000000	0.0	0.043478	0.000000	0.01087	0.01087	0.021739	0.01087	0.000000	0.000000	0.000000	0.01087	0.043478	0.01087	0.01087	0.054348	0.01087	0.021739	0.000000	0.021739	0.01087	0.01087	0.01087	0.01087	0
2	Kristiine	0.000000	0.000000	0.1	0.000000	0.000000	0.00000	0.10000	0.000000	0.00000	0.100000	0.100000	0.000000	0.00000	0.100000	0.00000	0.00000	0.000000	0.00000	0.000000	0.000000	0.000000	0.00000	0.00000	0.00000	0.00000	0
3	Lasnamäe	0.000000	0.000000	0.0	0.000000	0.000000	0.00000	0.00000	0.000000	0.00000	0.000000	0.000000	0.000000	0.00000	0.200000	0.00000	0.00000	0.000000	0.00000	0.000000	0.000000	0.000000	0.00000	0.00000	0.00000	0.00000	0
4	Mustamäe	0.000000	0.000000	0.0	0.000000	0.000000	0.00000	0.00000	0.000000	0.00000	0.125000	0.000000	0.125000	0.00000	0.000000	0.00000	0.00000	0.000000	0.00000	0.000000	0.000000	0.000000	0.00000	0.00000	0.00000	0.00000	0
5	Nõmme	0.000000	0.000000	0.0	0.000000	0.000000	0.00000	0.00000	0.000000	0.00000	0.000000	0.000000	0.250000	0.00000	0.000000	0.00000	0.00000	0.000000	0.00000	0.000000	0.000000	0.000000	0.00000	0.00000	0.00000	0.00000	0
6	Pirita	0.000000	0.333333	0.0	0.000000	0.000000	0.00000	0.00000	0.000000	0.00000	0.000000	0.000000	0.000000	0.00000	0.000000	0.00000	0.00000	0.000000	0.00000	0.000000	0.000000	0.000000	0.00000	0.00000	0.00000	0.00000	0
7	Põhja-Tallinn	0.000000	0.000000	0.0	0.000000	0.166667	0.00000	0.00000	0.000000	0.00000	0.000000	0.333333	0.166667	0.00000	0.000000	0.00000	0.00000	0.000000	0.00000	0.000000	0.000000	0.000000	0.00000	0.00000	0.00000	0.00000	0

- 10 most common in similar areas:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venu
0	Haabersti	Bus Line	Bus Stop	Supermarket	Pet Store	Pharmacy	Convenience Store	Bus Station	Cupcake Shop	Dance Studio	Department Sto
1	Kesklinn	Hotel	Cocktail Bar	Italian Restaurant	Gym / Fitness Center	Bar	Café	Restaurant	Asian Restaurant	Cosmetics Shop	Pu
2	Kristiine	Playground	Thai Restaurant	Auto Garage	Flower Shop	Bistro	Fast Food Restaurant	Bus Line	Pizza Place	Bus Station	Ca
3	Lasnamäe	Gym / Fitness Center	Electronics Store	Tennis Court	Café	Eastern European Restaurant	Cupcake Shop	Dance Studio	Department Store	Dessert Shop	Dumpling Restaura
4	Mustamäe	Pizza Place	Bus Line	Grocery Store	Farmers Market	Bus Stop	Pub	Mobile Phone Shop	Cupcake Shop	Dance Studio	Department Sto

Did the map on most similar areas based on 5 clusters



Clusters and the venues in there:

- Cluster 0

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Haabersti	Bus Line	Bus Stop	Supermarket	Pet Store	Pharmacy	Convenience Store	Bus Station	Cupcake Shop	Dance Studio	Department Store
1	Kesklinn	Hotel	Cocktail Bar	Italian Restaurant	Gym / Fitness Center	Bar	Café	Restaurant	Asian Restaurant	Cosmetics Shop	Pub
2	Kristiine	Playground	Thai Restaurant	Auto Garage	Flower Shop	Bistro	Fast Food Restaurant	Bus Line	Pizza Place	Bus Station	Café
4	Mustamäe	Pizza Place	Bus Line	Grocery Store	Farmers Market	Bus Stop	Pub	Mobile Phone Shop	Cupcake Shop	Dance Studio	Department Store

- Cluster 1

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
7	Põhja-Tallinn	Bus Station	Basketball Court	Liquor Store	Light Rail Station	Bus Stop	Wine Bar	Dance Studio	Department Store	Dessert Shop	Dumpling Restaurant

- Cluster 2

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
6	Pirita	Athletics & Sports	Garden	Park	Wine Bar	Electronics Store	Cupcake Shop	Dance Studio	Department Store	Dessert Shop	Dumpling Restaurant

- Cluster 3

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
5	Nõmme	Park	Bus Stop	Restaurant	Wine Bar	Eastern European Restaurant	Cosmetics Shop	Cupcake Shop	Dance Studio	Department Store	Dessert Shop

- Cluster 4

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
3	Lasnamäe	Gym / Fitness Center	Electronics Store	Tennis Court	Café	Eastern European Restaurant	Cupcake Shop	Dance Studio	Department Store	Dessert Shop	Dumpling Restaurant

Then I had a AHAA moment

- What if I change the code and make 7 unique request not combined request as the Lecture lab did it??

This means make unique Borough request on every area of city

Haabersti linnaosa

```
address = 'Haabersti, estonia'
geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

```
59.4313282 24.6290422
```

```
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    latitude,
    longitude,
    radius,
    limit)

# get the result to a json file
results = requests.get(url).json()
print("done")
print(url)
```

```
done
https://api.foursquare.com/v2/venues/explore?&client_id=A42H3MCRD8J0UJ0LQREYLJNKB2R1AQOM0FT1NUHWY5AGLVF5&client_secret=PKVYIP2SRSATJBW0RGM1Z0CEZ1VZFSJ5I5EUOMQ23DJCZHXX&v=20180605&ll=59.431
```

```
results = requests.get(url).json()
results
# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
nearby_venues.to_csv('Haabersti.csv')
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages/ipykernel_launcher.py:16: FutureWarning: pandas.io.json.json_normalize is deprecated, use pandas.json_normalize instead
  app.launch_new_instance()
```

I did this in every Borough and made a combined/ full database

```
full_df=pd.concat([df1, df2 ,df3 ,df4 ,df5 ,df6 ,df7])
full_df.to_csv('./kokku.csv')
```

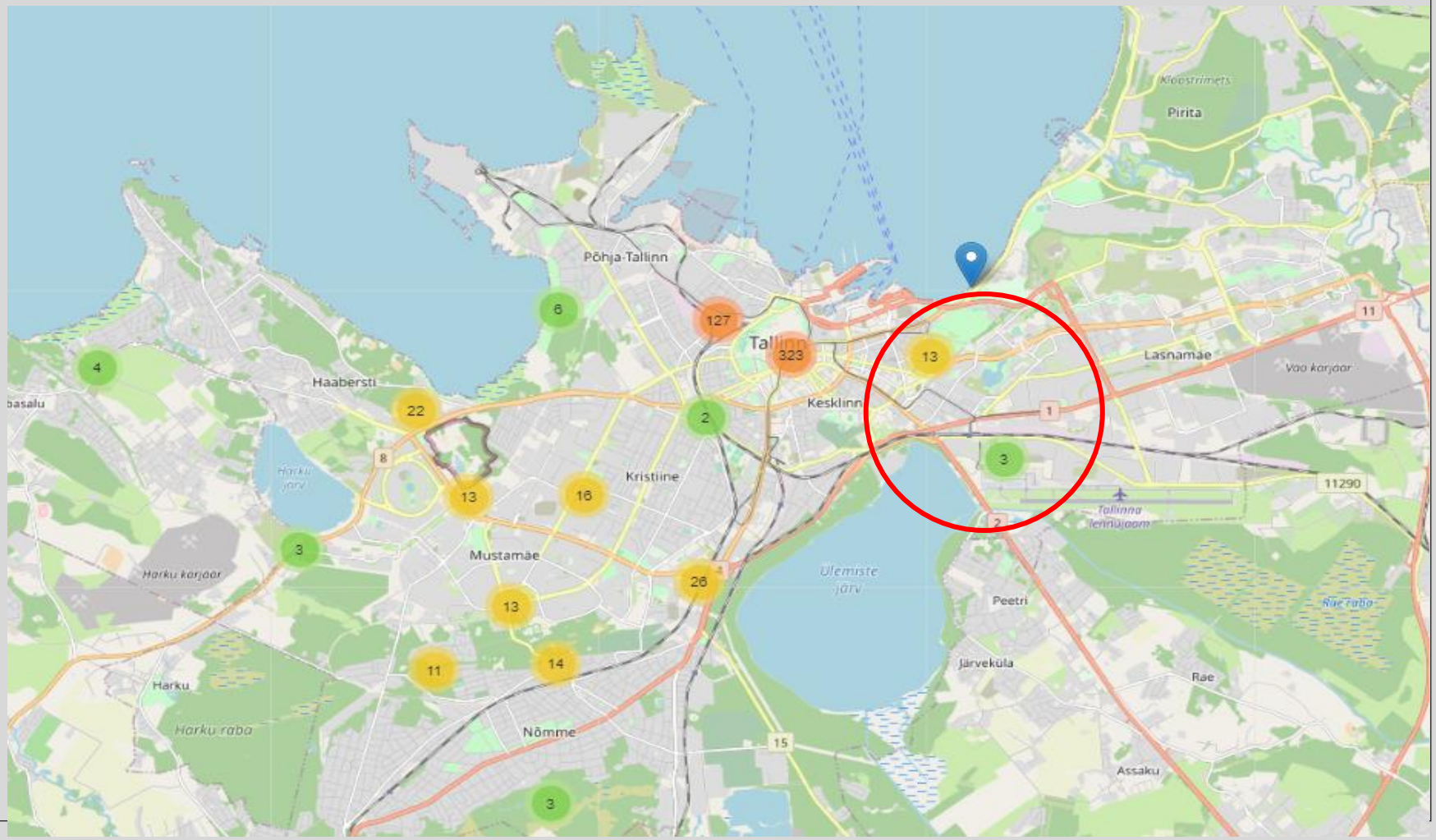
The result of this is much better:

```
df_incidents= pd.read_csv("./kokku.csv")
df_incidents.shape
```

```
(617, 4)
```

To put this in map- it looks like this:

- Data is still crooked on the side of central city with 127+323 (more then half) of datapoints in there
- Data in the area of City where most people live in is 13+3 = not valid for sure
- As the difference in data amount in areas is so huge then the comparison is like comparing apples with planes



So its still not good……

- Because the real business count (let alone the buss stations and parks etc what foursquare reads as venues is 10 time bigger)……

Exporting economic units by administrative unit, years

9 October 2020

	2011	2012	2013	2014	2015	2016	2017	2018	2019
Whole country	11 987	13 190	14 361	14 486	14 851	15 376	16 148	16 090	16 290
Harju county	6 806	7 305	7 617	7 731	7 934	8 162	8 513	8 518	8 562
..Tallinn	5 419	5 765	5 988	6 055	6 192	6 356	6 557	6 566	6 516
Hiiu county	45	41	51	59	50	56	68	68	61
Ida-Viru county	347	384	396	404	430	450	462	427	418
..Kohtla-Järve city	64	70	73	78	72	72	74	70	62
..Narva city	120	133	150	146	183	177	186	187	173
..Sillamäe city	35	41	44	48	45	54	51	42	45
Jõgeva county	120	112	130	138	156	136	165	161	159
Järva county	150	160	170	186	194	184	180	197	188

https://vana.stat.ee/413461

As a result:

Nice ap and API and really learned a lot, but its not valid in every location and a alternative is needed……..