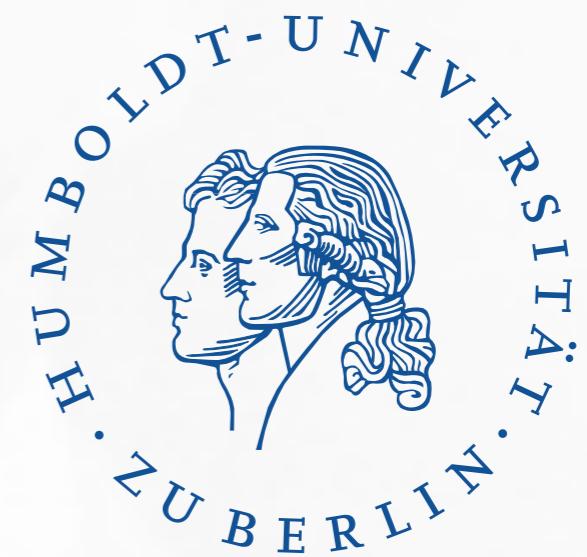


Digital Economy and Decision Analytics - Blockchain and Cryptocurrency Seminar (WS 21/22)

Spatial Analysis of Berlin Rent
Prices and Landlord Premiums

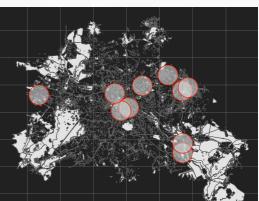
Ivan Kotik

Ladislaus von Bortkiewicz Chair of Statistics
C.A.S.E. - Center for Applied Statistics
and Economics
Humboldt-Universität zu Berlin
lvb.wiwi.hu-berlin.de



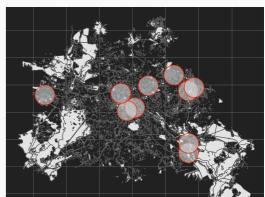
Outline

1. Motivation
2. Developing the spatial framework
3. Modeling the rent prices
4. Geocoding
5. Distance measurement and normalization
6. Evaluating landlord premiums



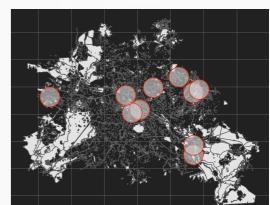
Motivation

- Rent prices differ a lot between the same type of apartments
- Rent prices differ a lot between neighbourhoods
- Usually people “feel” these prices out of their own experiences
- Always a long term investment
- No easy way to create a benchmark



Part 1: developing the spatial framework

Cartographical data is taken from  **OpenStreetMap**, which is a collaborative project to create a free editable geographic database of the world.



What does the data look like?

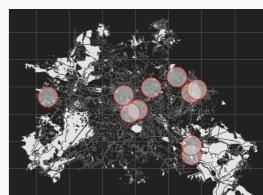
```

a_ber_poi_multipolygon = Table: 18,246 x 5 ... View Table
a_ber_poi_polygon = Table: 75,859 x 5 ... View Table
b_ber_landuse_multipolygon = Table: 33,291 x 5 ... View Table
c_ber_transport_polygon = Table: 36 x 5 ... View Table
d_ber_water_multipolygons = Table: 1,910 x 5 ... View Table
e_ber_map = Table: 109 x 6 ... View Table

```

```
> a_ber_poi_polygon
Simple feature collection with 75859 features and 4 fields
Geometry type: POINT
Dimension: XY
Bounding box: xmin: 13.08407 ymin: 52.33794 xmax: 13.76027 ymax: 52.6727
Geodetic CRS: WGS 84
# A tibble: 75,859 × 5
  osm_id   code fclass      name      geometry
  <chr>   <int> <chr>      <chr>      <POINT [°]>
1 16541597 2907 camera_surveillance Aral      (13.34544 52.54644)
2 26735749 2301 restaurant          Aida      (13.32282 52.50691)
3 26735753 2006 telephone          NA        (13.32214 52.50645)
4 26735759 2301 restaurant          Madame Ngo (13.31808 52.50621)
5 26735763 2301 restaurant          Thanh Long (13.32078 52.50732)
6 26754448 2701 tourist_info       NA        (13.37075 52.52329)
7 26865440 2307 biergarten         Spinnerbrücke (13.19073 52.43336)
8 26867409 2031 recycling_glass    NA        (13.29683 52.50133)
9 26972366 2724 memorial           Konrad Zuse (13.34722 52.52277)
10 27318009 2307 biergarten        Loretta     (13.17635 52.42009)
# ... with 75,849 more rows
```

```
> b_ber_landuse_multipolygon
Simple feature collection with 33291 features and 4 fields
Geometry type: MULTIPOLYGON
Dimension: XY
Bounding box: xmin: 13.05376 ymin: 52.32824 xmax: 13.76513 ymax: 52.68608
Geodetic CRS: WGS 84
# A tibble: 33,291 × 5
  osm_id   code fclass      name      geometry
  <chr>   <int> <chr>      <chr>      <MULTIPOLYGON [°]>
1 4401982 7206 cemetery Friedhof Wilmers... (((13.30865 52.48411, 13.30885 52.48...
2 4413796 7202 park    Preußenspark    (((13.31033 52.49362, 13.31039 52.49...
3 4440110 7202 park    NA            (((13.39298 52.39897, 13.39326 52.39...
4 4535352 7206 cemetery Städtischer Friem... (((13.32128 52.47605, 13.32128 52.47...
5 4537560 7201 forest   NA            (((13.2787 52.5447, 13.28131 52.5447...
6 4582178 7207 allotments Kleingartenkolon... (((13.30467 52.48664, 13.30539 52.48...
7 4582243 7202 park    Volkspark Wilmer... (((13.30852 52.48381, 13.30862 52.48...
8 4582244 7202 park    Volkspark Wilmer... (((13.31376 52.48316, 13.31378 52.48...
9 4585104 7218 grass   NA            (((13.36668 52.47268, 13.3667 52.472...
10 1657757 7202 park   Fritz-Schloß-Park (((13.35311 52.53125, 13.35319 52.53...
# ... with 33,281 more rows
```



Cleaning the data:

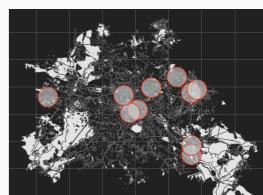
Next step is to clean the data and leave out the things that are not related to factors that could have an impact on rent prices. That is excluding such factors as police stations and hydrants.

4.2 Points of Interest

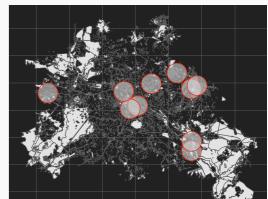
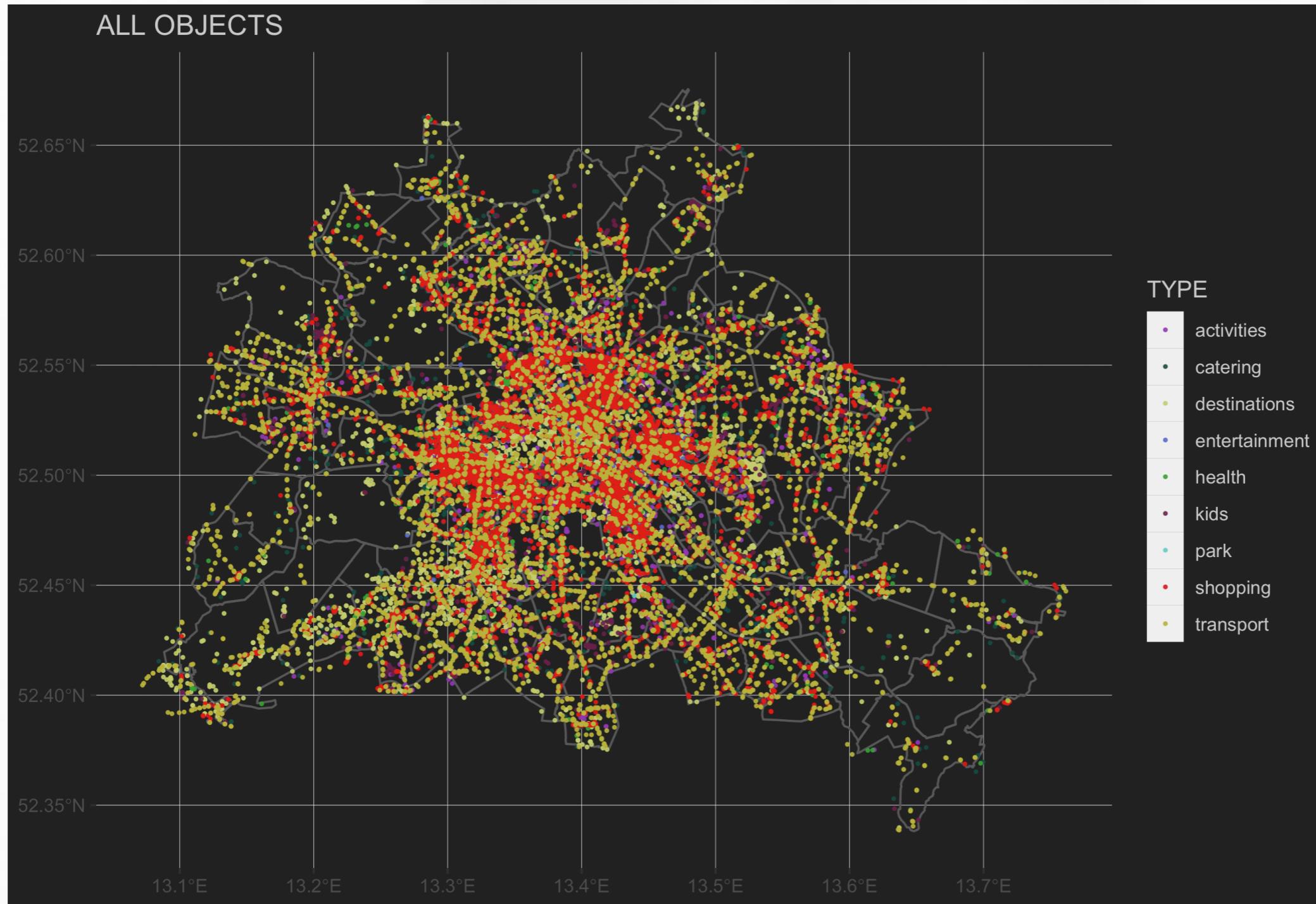
This layer has an associated area layer (see section 2.8).

The following feature classes exist in this layer:

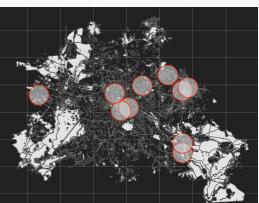
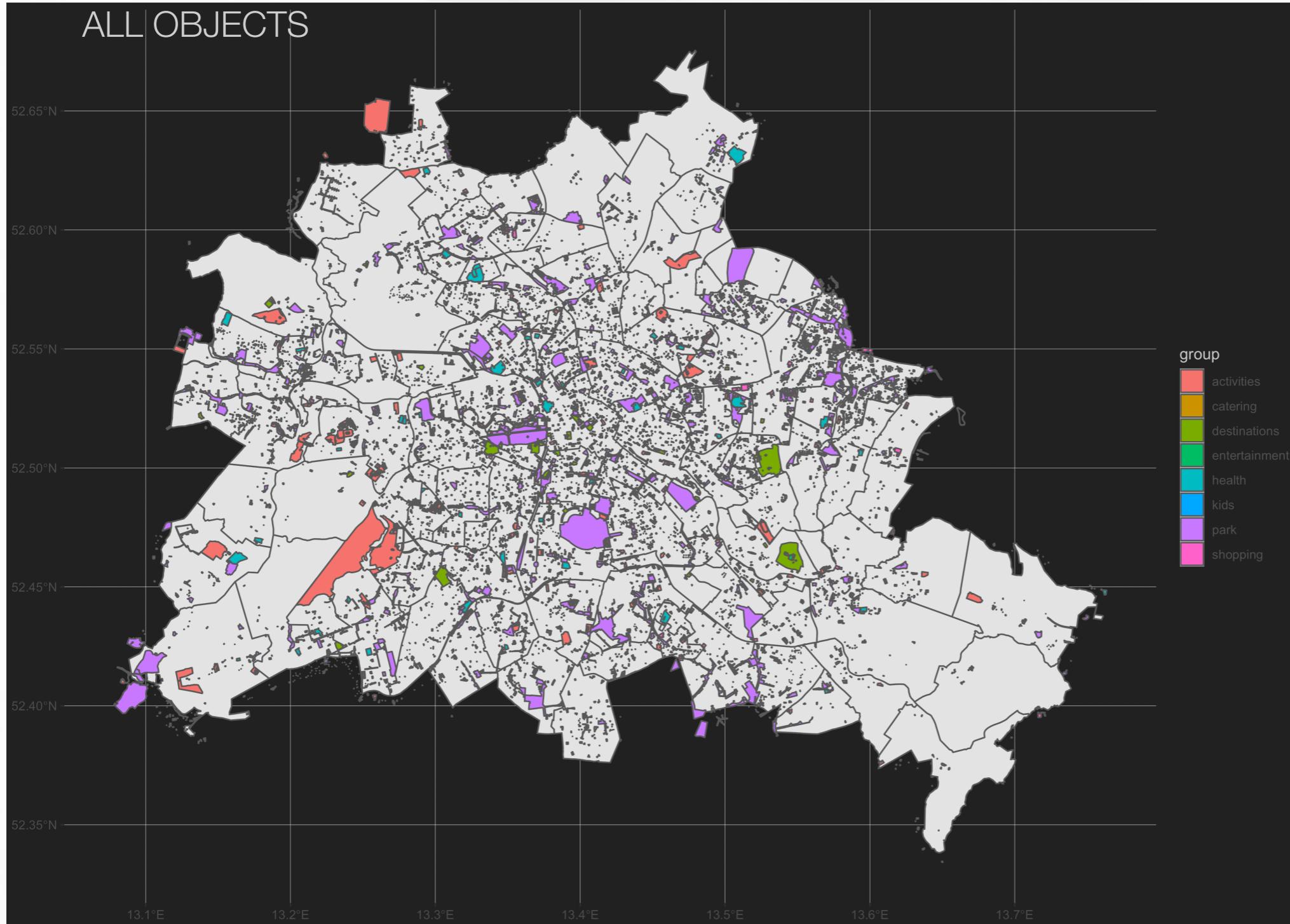
code	layer	fclass	Description	OSM Tags
20xx	public			
2001		police	A police post or station.	amenity=police
2002		fire_station	A fire station.	amenity=fire_station
2004		post_box	A post box (for letters).	amenity=post_box
2005		post_office	A post office.	amenity=post_office
2006		telephone	A public telephone booth.	amenity=telephone
2007		library	A library.	amenity=library
2008		town_hall	A town hall.	amenity=townhall
2009		courthouse	A court house.	amenity=courthouse
2010		prison	A prison.	amenity=prison
2011		embassy	An embassy or consulate.	amenity=embassy or office=diplomatic



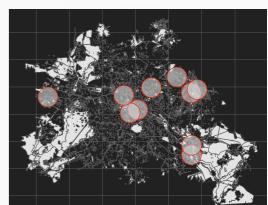
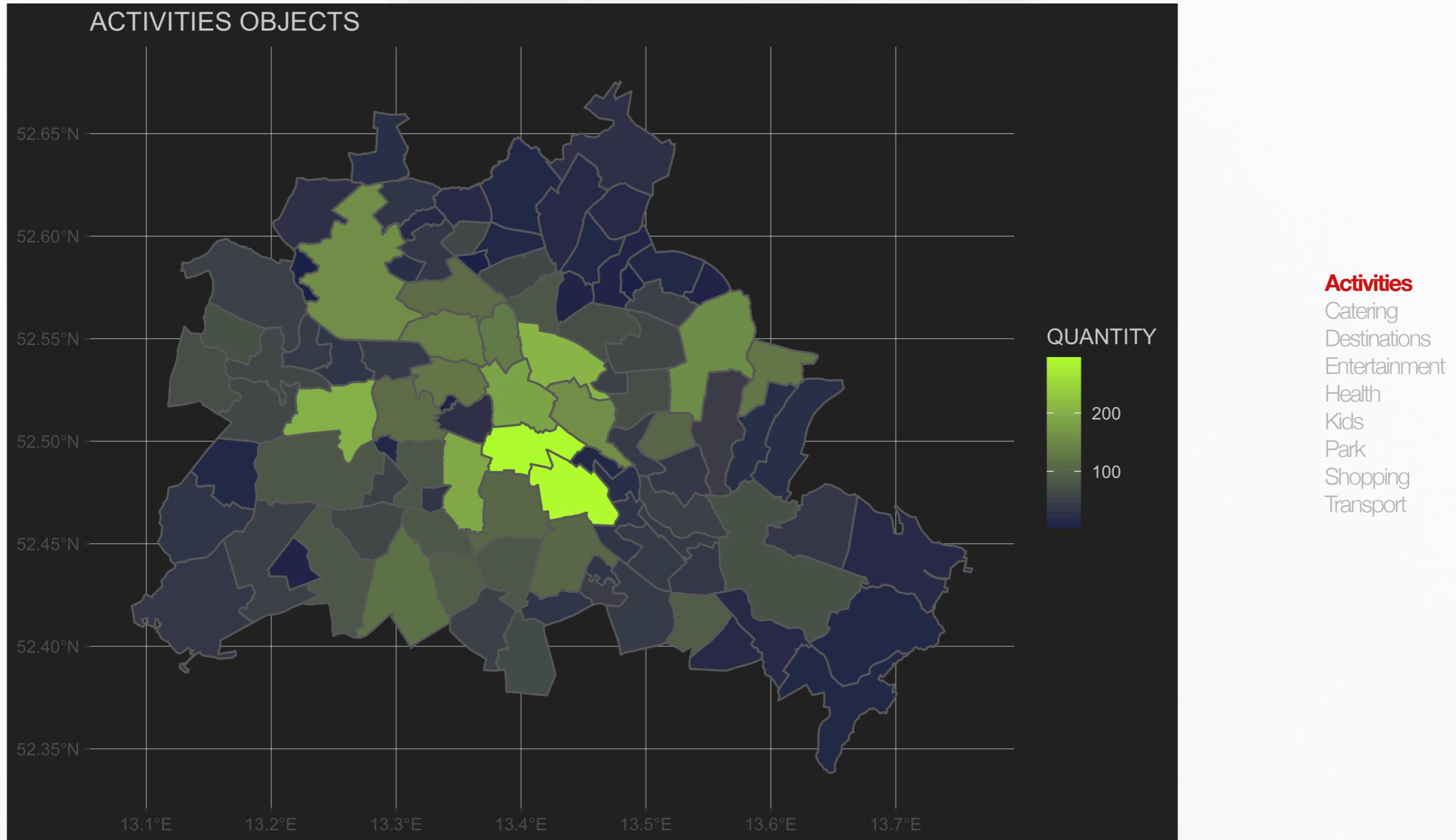
How the data looks like (polygons)



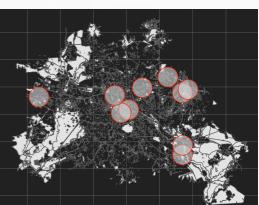
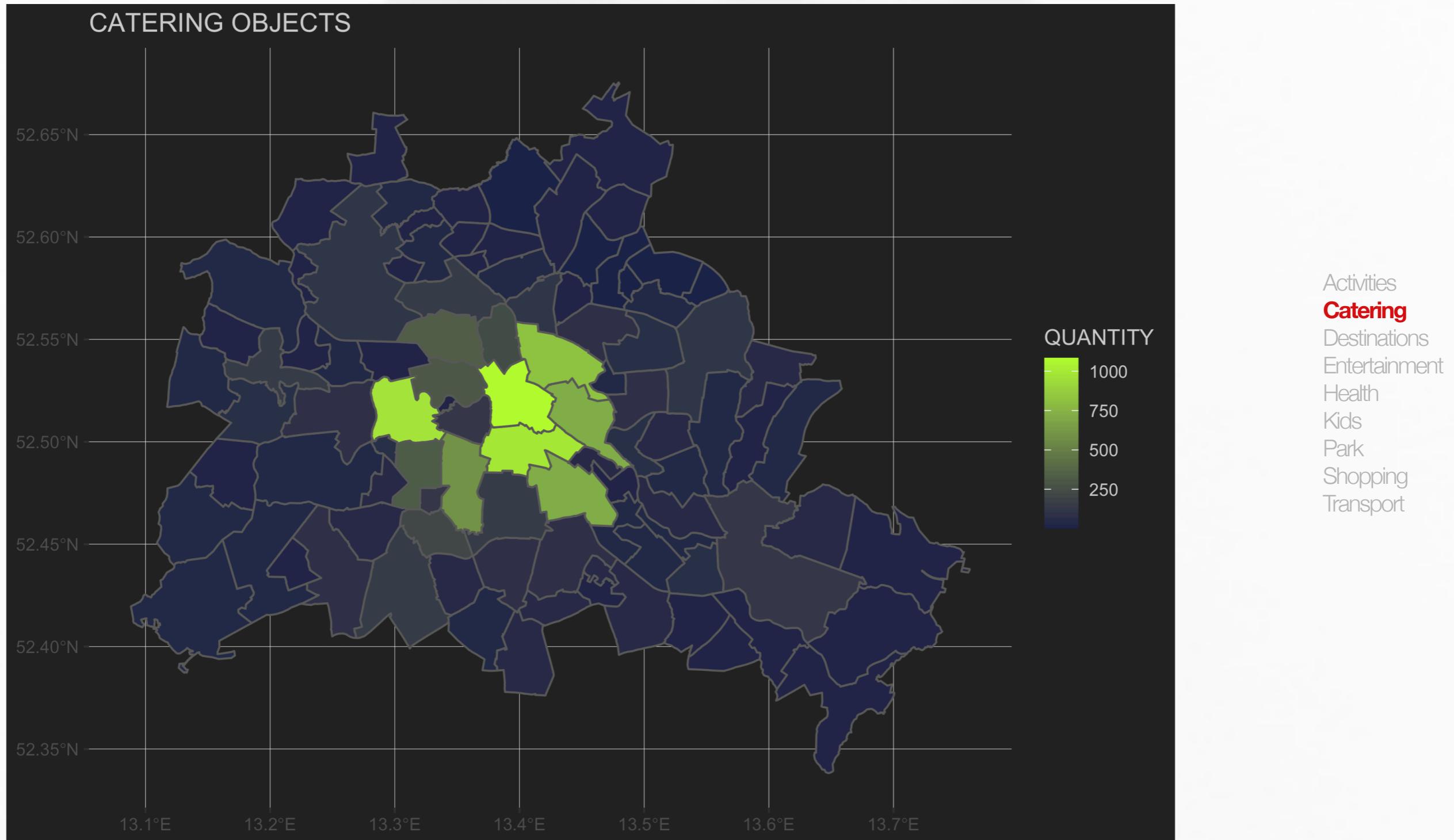
How the data looks like (multipolygons)



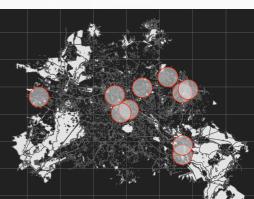
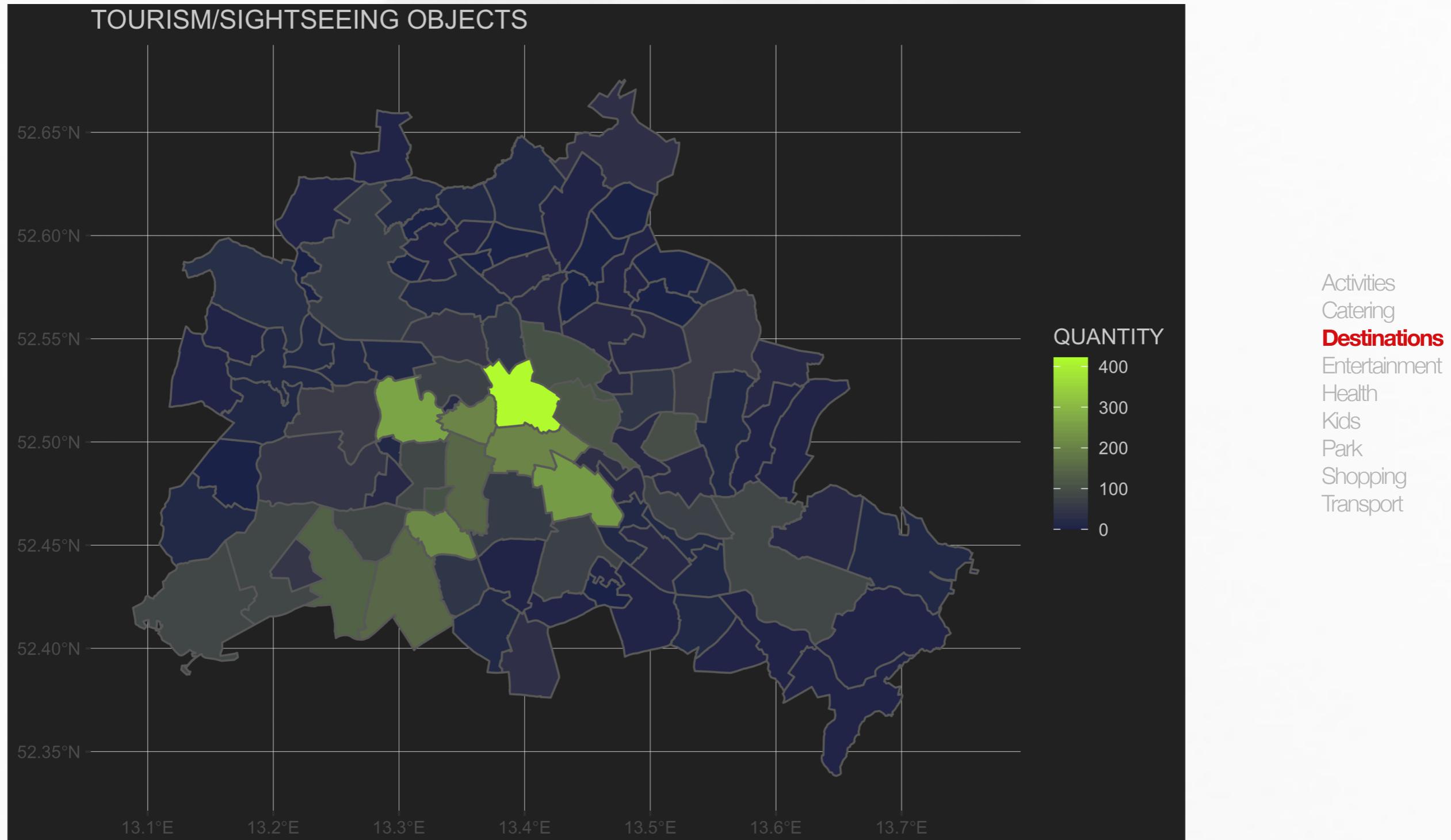
More maps: gyms, sports objects, etc.



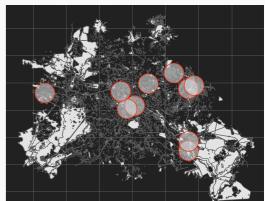
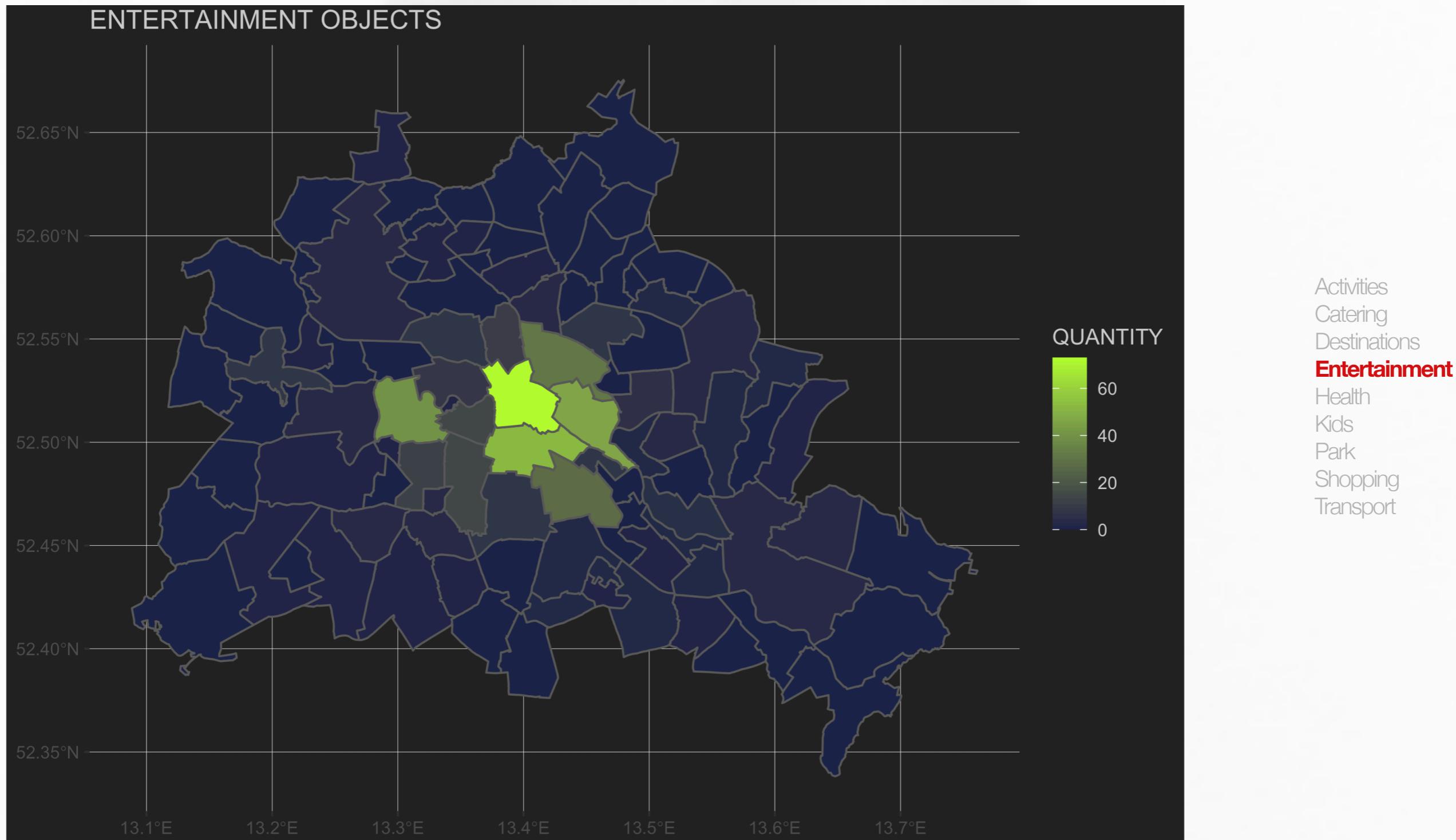
More maps: restaurants, bars, etc.



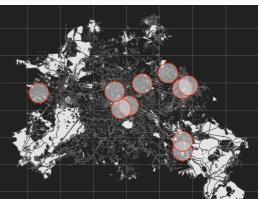
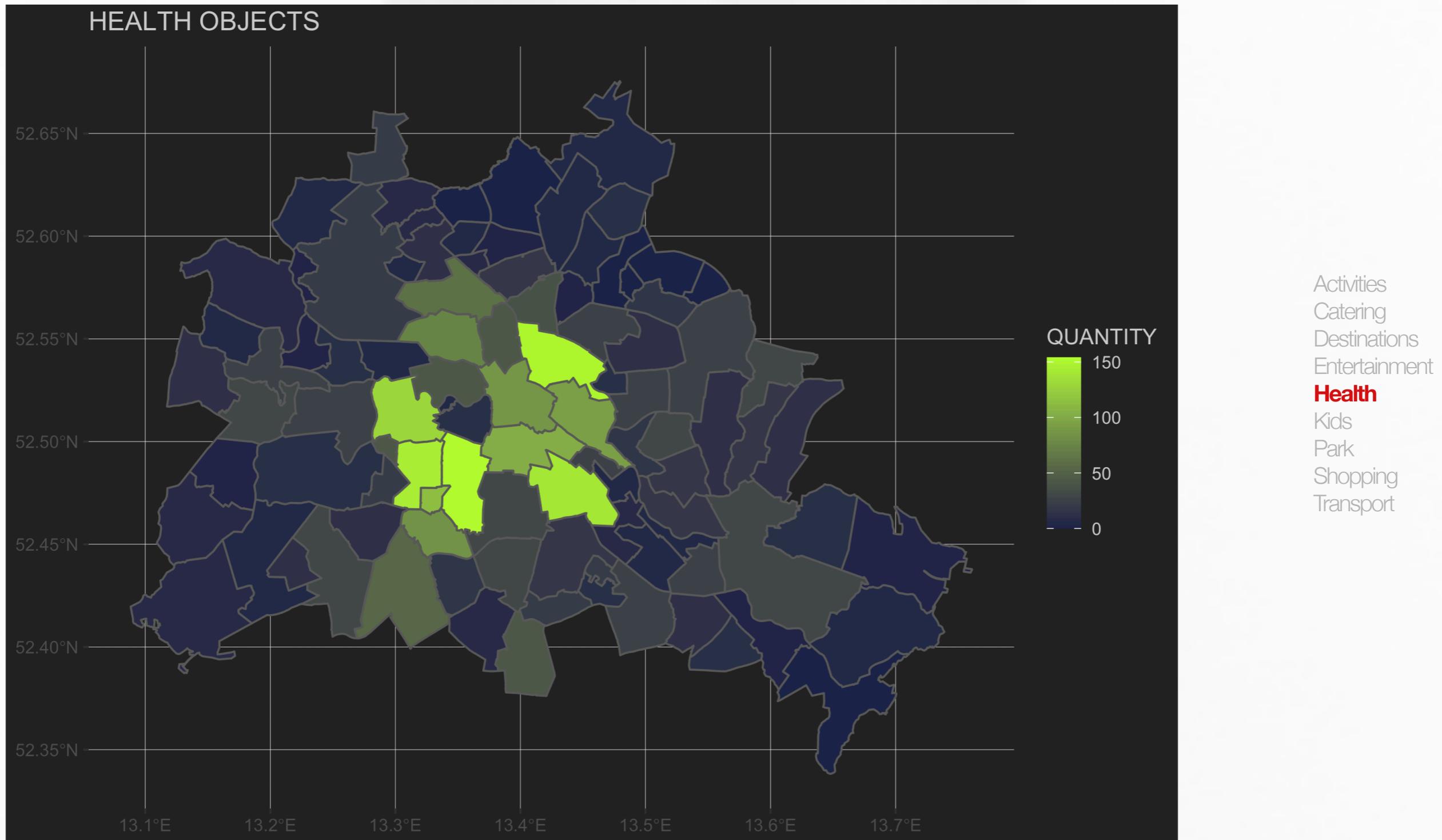
More maps: sightseeing, museums, etc.



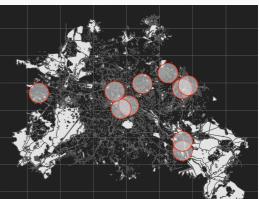
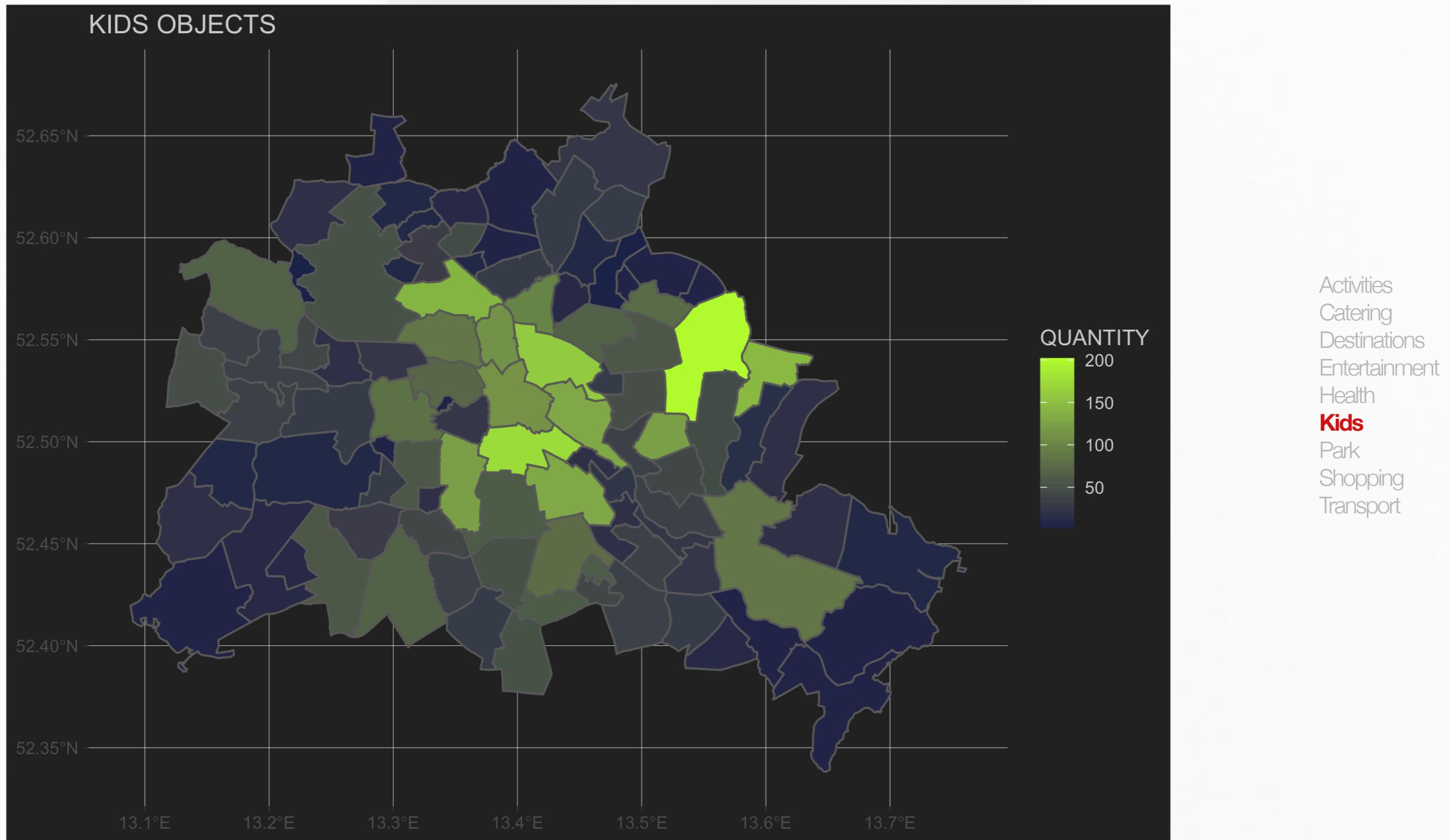
More maps: cinemas, night clubs, etc.



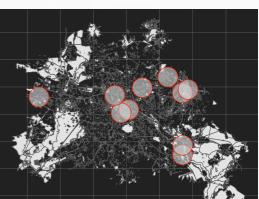
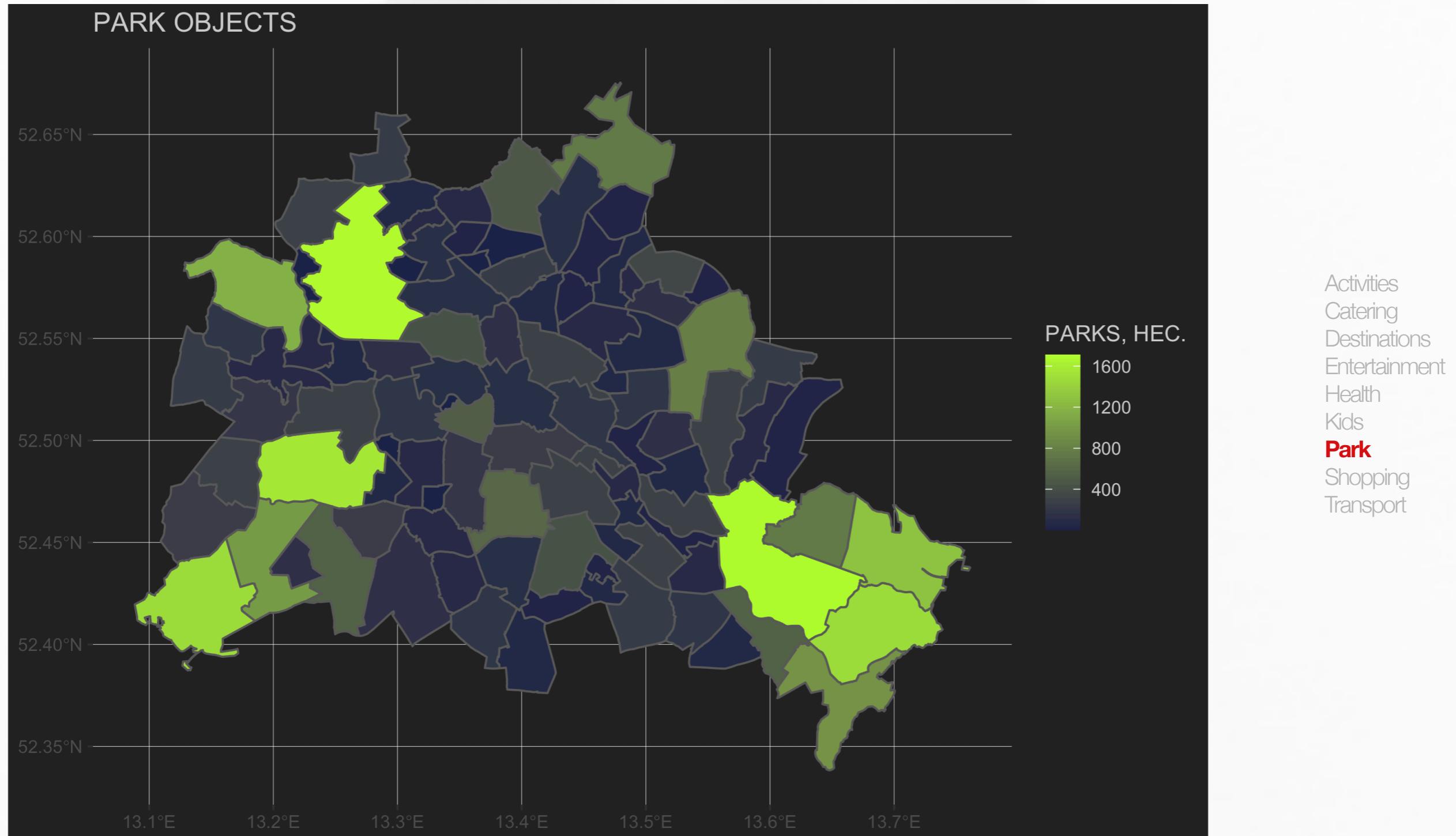
More maps: doctors, pharmacies, etc.



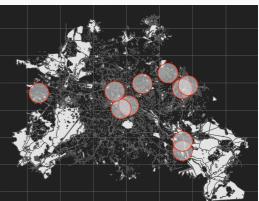
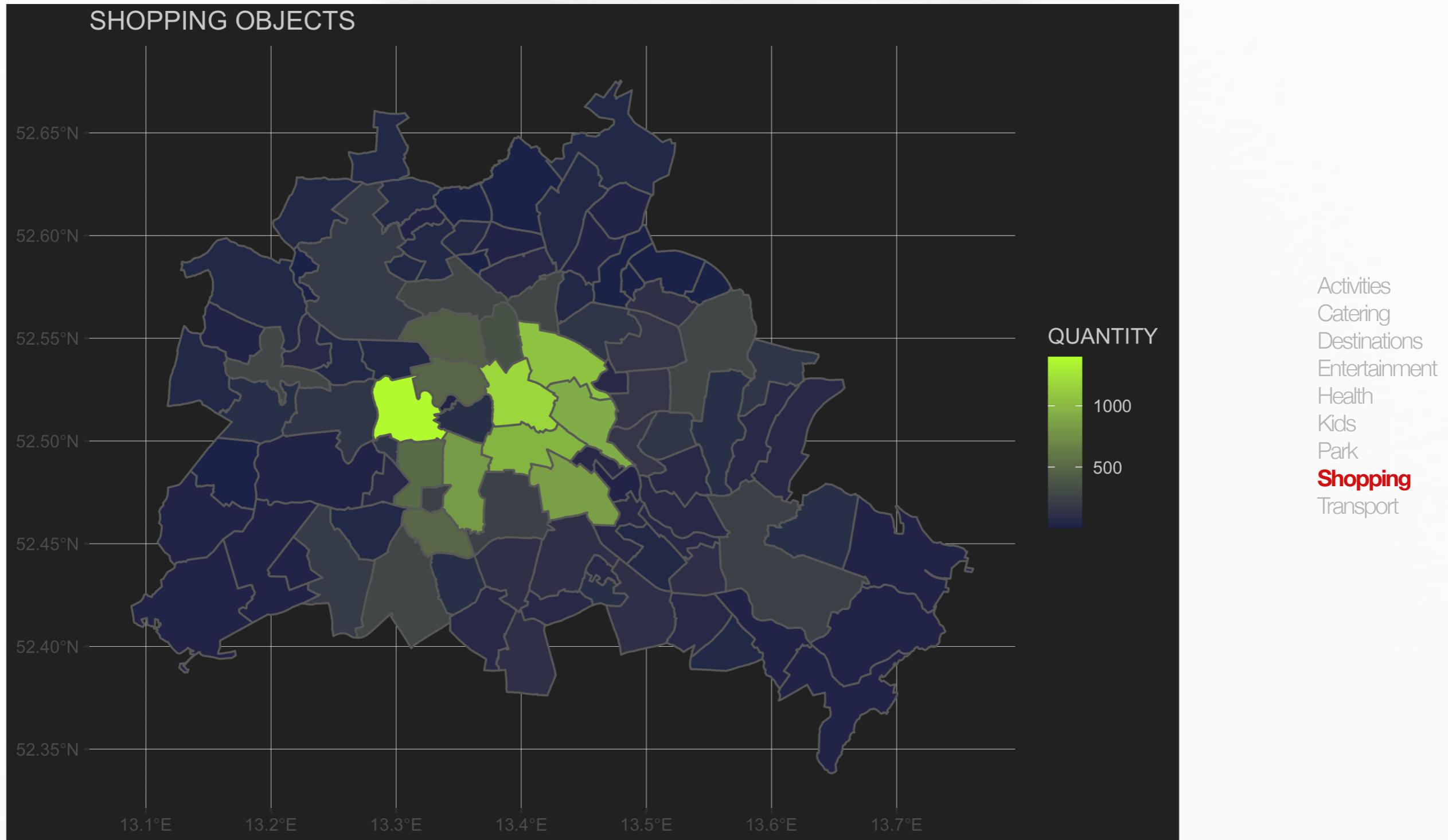
More maps: kitas, playgrounds, etc.



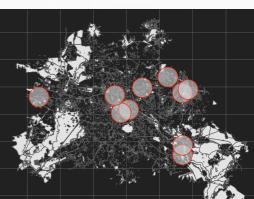
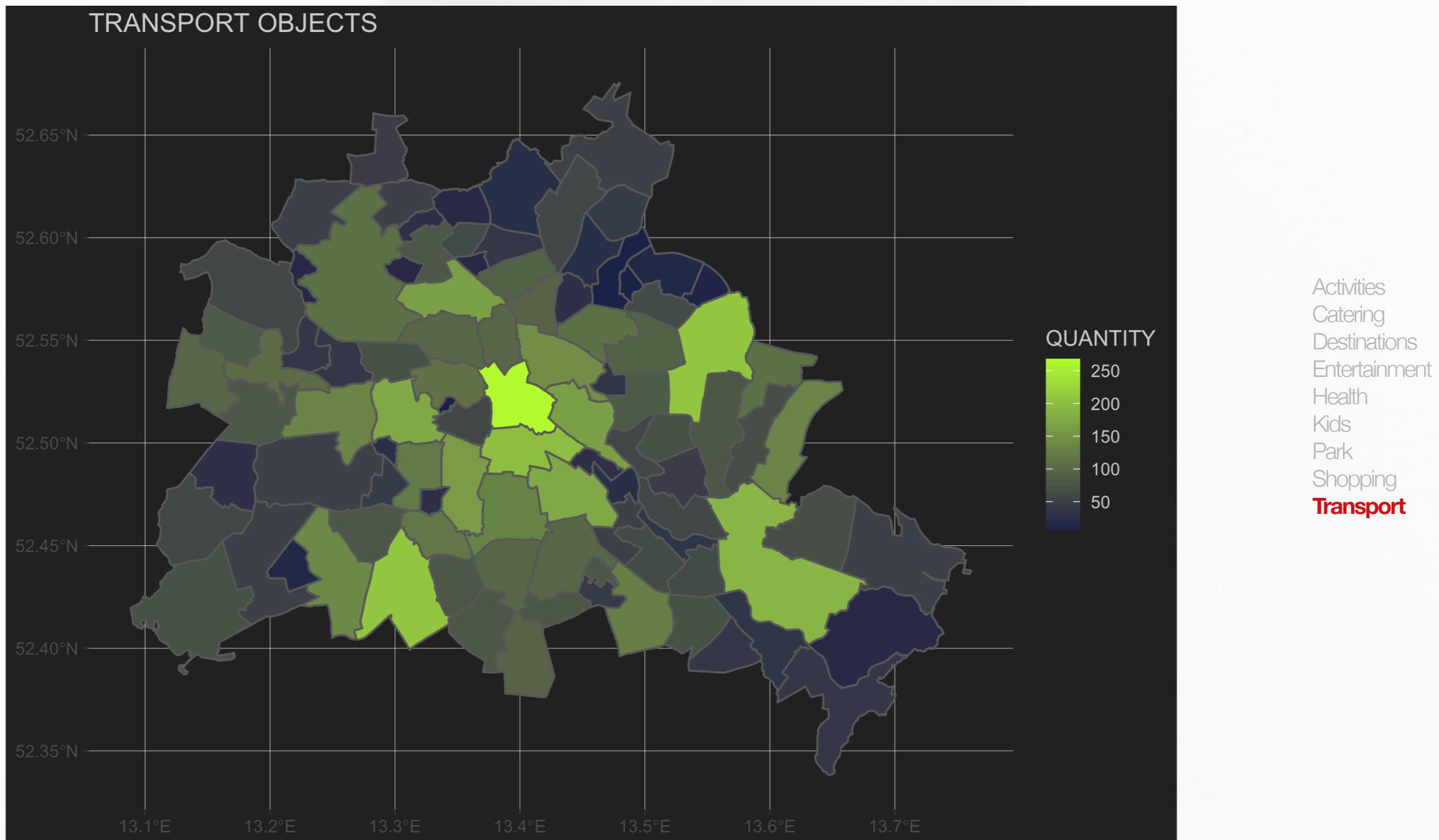
More maps: parks, dog parks, etc.



More maps: shops, grocery stores, etc.



More maps: S-Bahn, U-Bahn, etc.



Modeling the rent prices

Data taken from Kaggle, scraped from Immobilienscout24:

The screenshot shows a Kaggle dataset page for 'Apartment rental offers in Germany'. The title is 'Apartment rental offers in Germany' and the subtitle is 'Rental offers scraped from Germany biggest real estate online plattform'. The dataset was created by CorrieBar and updated 2 years ago (Version 6). The page includes tabs for Data (selected), Code (14), Discussion, Activity, and Metadata. Buttons for Download (285 MB) and New Notebook are present. Below the main header, there are sections for Usability (10.0), License (Data files © Original Authors), and Tags (real estate). A large image of a multi-story apartment building with balconies is displayed at the top. The 'Description' section contains the following text:

Where is the data from?

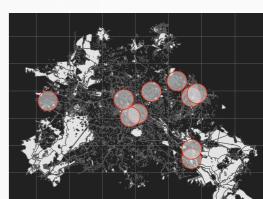
The data was scraped from Immoscout24, the biggest real estate platform in Germany. Immoscout24 has listings for both rental properties and homes for sale, however, the data only contains offers for rental properties.

The scraping process is described in this [blog post](#) and the corresponding code for scraping and minimal processing afterwards can be found in this [Github repo](#).

At a given time, all available offers were scraped from the site and saved. This process was repeated three times, so the data set contains offers from the dates 2018-09-22, 2019-05-10 and 2019-10-08.

Content

<https://www.kaggle.com/corrieaar/apartment-rental-offers-in-germany/version/6>

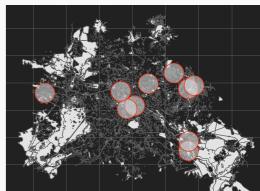


The data and data clearing

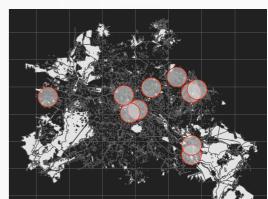
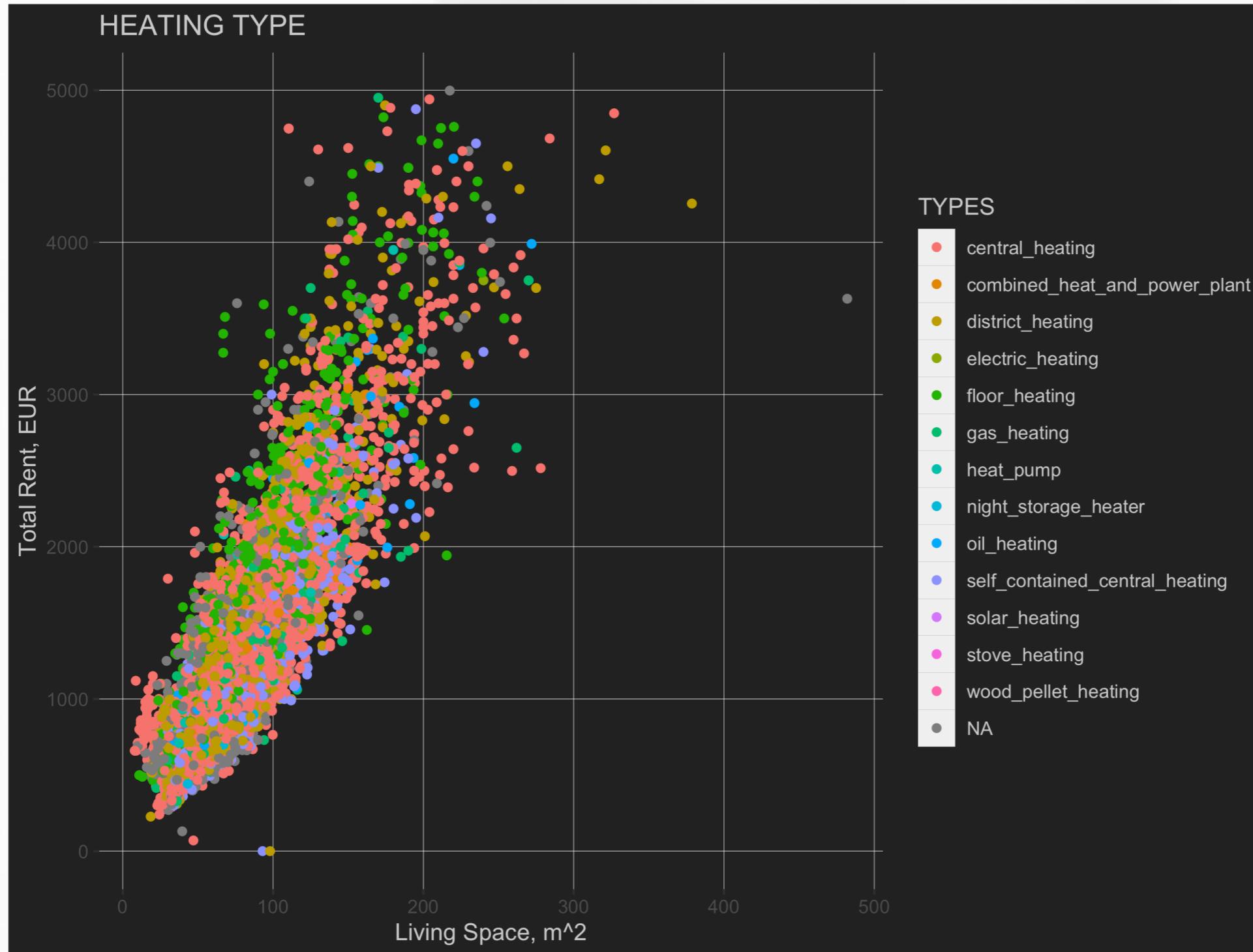
Extract Berlin from all the areas in Germany, filter the other columns

Baden_Württemberg		Bayern	Berlin
16091		21609	10406
Brandenburg		Bremen	Hamburg
6954		2965	3759
Hessen	Mecklenburg_Vorpommern		Niedersachsen
17845		6634	16593
Nordrhein_Westfalen	Rheinland_Pfalz		Saarland
62863		8368	1429
Sachsen	Sachsen_Anhalt		Schleswig_Holstein
58154		20124	6668
Thüringen			
8388			

...and more cleaning



Cleaning example: do we care about heating?



The general model:

$$\text{Price}_i = \beta_0 + \beta_1 \text{heating}_i + \beta_2 \text{living space}_i + \beta_3 \# \text{rooms}_i + \dots + \beta_{max} \text{lift}_i$$

(dummy) (dummy)

```

Call:
lm(formula = price ~ ., data = data_berlin_model_alltime)

Residuals:
    Min      1Q  Median      3Q     Max 
-3046.9 -190.8 -14.3  160.9 7377.8 

Coefficients:
                                         Estimate Std. Error t value    Pr(>|t|)    
(Intercept)                         747.4011   365.1490  2.047    0.049745 *  
heatingTypecombined_heat_and_power_plant -4.3181    50.7278 -0.085    0.932168    
heatingTypedistrict_heating            61.0344   19.2824  3.165    0.001562 **  
heatingTypeelectric_heating          -154.7923   152.0269 -1.018    0.308654    
heatingTypefloor_heating              95.7866   21.9377  4.366    0.000012984780 *** 
heatingTypegas_heating                -22.6382   34.1480 -0.663    0.507407    
heatingTypeheat_pump                 -34.9088   91.9724 -0.380    0.704296    
heatingTypenight_storage_heater       -158.6878   92.0323 -1.724    0.084745 .  
heatingTypeoil_heating                -18.1057   47.6448 -0.380    0.703957    
heatingTypeself_contained_central_heating -77.9583   24.6960 -3.157    0.001608 **  
heatingTypesolar_heating              -192.3944   201.8725 -0.953    0.340628    
heatingTypestove_heating              50.1163   362.2937  0.166    0.868334    
heatingTypewood_pellet_heating        -59.3090   284.2205 -0.209    0.834716    
livingSpace                           20.2205    0.3171  63.763 < 0.0000000000000002 *** 
noRooms                                -118.1389   11.3543 -10.405 < 0.0000000000000002 *** 
hasKitchenTRUE                          135.7235   15.6967  8.647 < 0.0000000000000002 *** 
extrawcTRUE                             -33.3685   21.1164 -1.580    0.114143    
balconyTRUE                            -37.4643   18.3682 -2.040    0.041459 *  
geschirrspTRUE                          -4.5796    49.8428 -0.092    0.926798    
washingmTRUE                            57.7797   16.6531  3.470    0.000527 ***  
typeOffFlatground_floor                10.7332   25.7367  0.417    0.676675    
typeOffFlathalf_basement               -166.8751   201.2550 -0.829    0.407061    
typeOffFlatloft                          166.2761   69.5983  2.389    0.016940 *  
typeOffFlatmaisonette                  -126.3728   37.3286 -3.385    0.000718 *** 
typeOffFlatother                         -111.4576   57.3925 -1.942    0.052210 .  
typeOffFlatpenthouse                   84.4989   42.7761  1.975    0.048300 *  
typeOffFlatraised_ground_floor          -24.3540   45.8593 -0.531    0.595409    
typeOffFlatrooft_storyy                 -45.6478   23.8133 -1.917    0.055328 .  
typeOffFlatterraced_flat               14.8045    48.7432  0.304    0.761354    
gardenTRUE                             -24.5034   17.8980 -1.369    0.171063    
interiorQualnormal                     -504.8291   27.2210 -18.546 < 0.0000000000000002 *** 
interiorQualsimple                      -485.9461   103.4259 -4.698    0.000002716618 *** 
interiorQualsophisticated              -420.6247   20.6580 -20.361 < 0.0000000000000002 *** 
conditionfirst_time_use_after_refurbishment -48.7411   32.4277 -1.503    0.132966    
conditionfully_renovated               -216.7098   35.8141 -6.051    0.000000001584 *** 
conditionmint_condition                -62.0043   29.8743 -2.077    0.037885 *  
conditionmodernized                    -195.8406   38.2705 -5.117    0.000000325282 *** 
conditionneed_of_renovation             -170.3806   95.6920 -1.781    0.075075 .  
conditionnegotiable                   -211.5881   98.8647 -2.329    0.019934 *  
conditionrefurbished                  -104.2009   35.4655 -2.938    0.003323 **  
conditionwell_kept                     -206.3251   33.0845 -6.236    0.000000000499 *** 
cellarTRUE                               -4.8510    14.7192 -0.330    0.741744    
yearConstructed                        -8.2045    0.1868 -1.095    0.273721    
newlyConstTRUE                          13.8662   29.6351  0.468    0.639886    
floor                                    14.2858    4.8651  2.936    0.003341 ** 
numberOfFloors                          9.6770    4.4361  2.181    0.029217 *  
liftTRUE                                 111.4863   18.8175  5.925    0.000000003418 *** 

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 399.9 on 3682 degrees of freedom
(5433 observations deleted due to missingness)
Multiple R-squared:  0.8253,    Adjusted R-squared:  0.8231 
F-statistic: 378.1 on 46 and 3682 DF,  p-value: < 0.0000000000000022

```

Heating types

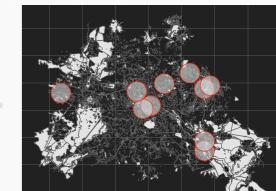
Apartment items

Type of apartment

Interior quality

Apartment conditions

House parameters



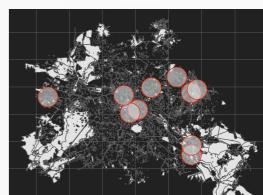
The general model:

After multiple backward selection iterations and making a 90% percentile cutoff for the price at 2400 EUR:

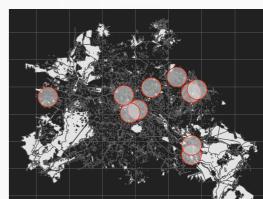
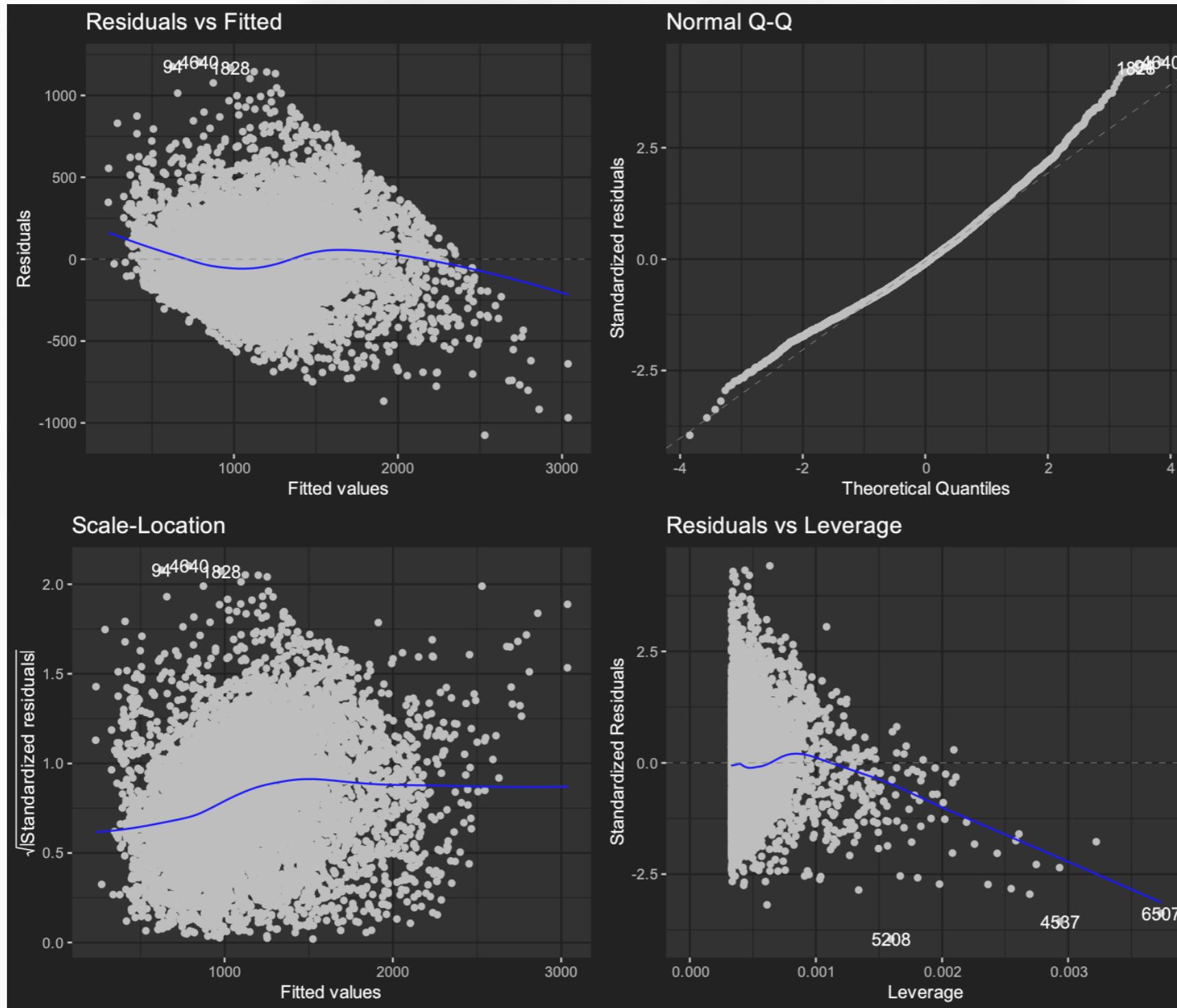
$$\text{Price}_i = \beta_0 + \beta_1 \text{Living space}_i + \beta_2 \text{Built-in-Kitchen}_i + \beta_3 \text{ Lift}_i$$

(dummy) *(dummy)*

```
Call:  
lm(formula = price ~ livingSpace + hasKitchen + lift, data = data_berlin_model_filter)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-1075.76 -195.87 -23.61  168.52 1202.85  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  24.446    9.138   2.675  0.00749 **  
livingSpace   13.160    0.109 120.754 < 0.0000000000000002 ***  
hasKitchenTRUE 217.212    6.119  35.498 < 0.0000000000000002 ***  
liftTRUE       150.700    6.094  24.730 < 0.0000000000000002 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 272.2 on 8238 degrees of freedom  
Multiple R-squared:  0.6792,    Adjusted R-squared:  0.6791  
F-statistic: 5813 on 3 and 8238 DF,  p-value: < 0.000000000000022
```



The general model:



Geocoding

Apartments are geographically represented with a physical address. In order to work with that data spatially what we would need is to geocode those addresses to latitudes and longitudes. How?

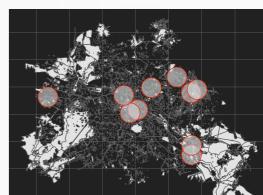
Via OpenStreetMap API Nominatim:

Nominatim API

Nominatim indexes named (or numbered) features within the OpenStreetMap (OSM) dataset and a subset of other unnamed features (pubs, hotels, churches, etc).

Its API has the following endpoints for querying the data:

- [`/search`](#) - search OSM objects by name or type
- [`/reverse`](#) - search OSM object by their location
- [`/lookup`](#) - look up address details for OSM objects by their ID
- [`/status`](#) - query the status of the server
- [`/deletable`](#) - list objects that have been deleted in OSM but are held back in Nominatim in case the deletion was accidental
- [`/polygons`](#) - list of broken polygons detected by Nominatim
- [`/details`](#) - show internal details for an object (for debugging only)



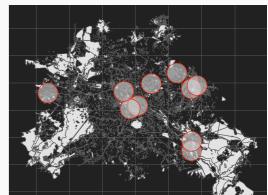
Distance measurement, scoring and normalization

The distance measurement chosen for this project is the Euclidian distance. The following formula was used to calculate the scores:

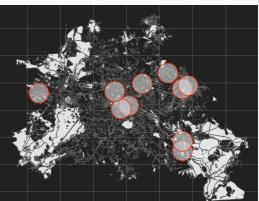
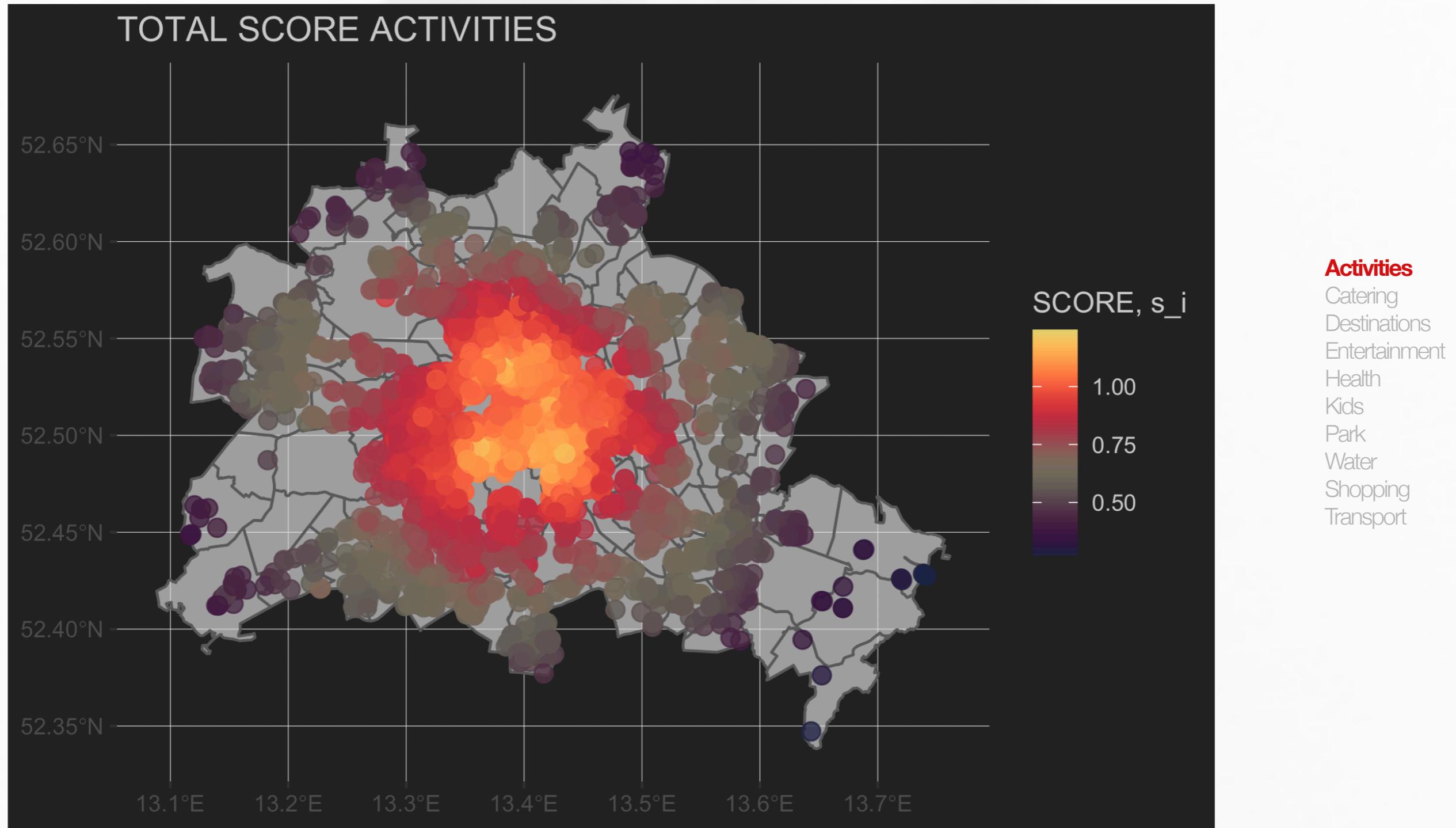
$$s_{i,k}^{dist} = \sum_{j=1}^n \left(\sqrt{(x_{A_i} - x_{O_{j,k}})^2 + (y_{A_i} - y_{O_{j,k}})^2} \right)^{-1}$$

Where x_{A_i}, y_{A_i} are the geographical coordinates of the apartment and x_{O_j}, y_{O_j} represent the infrastructure object coordinates.
 i – apartment subscript, j – infrastructure object subscript,
 k – category of infrastructural object subscript

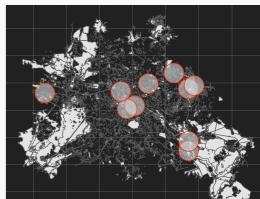
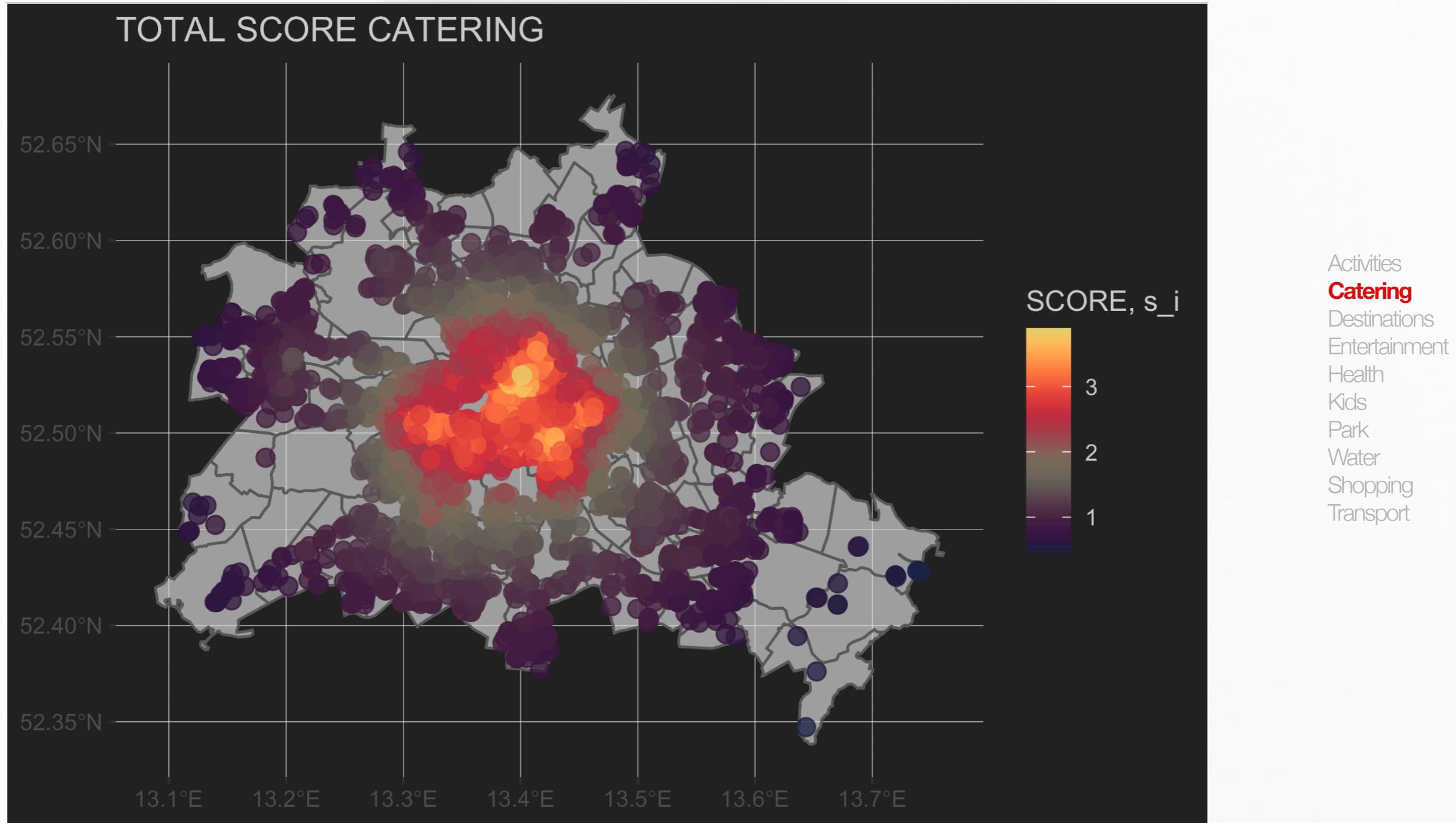
10 units (meters) are added to the distance in order to set the maximum score achievable for the measurement.



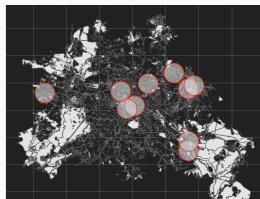
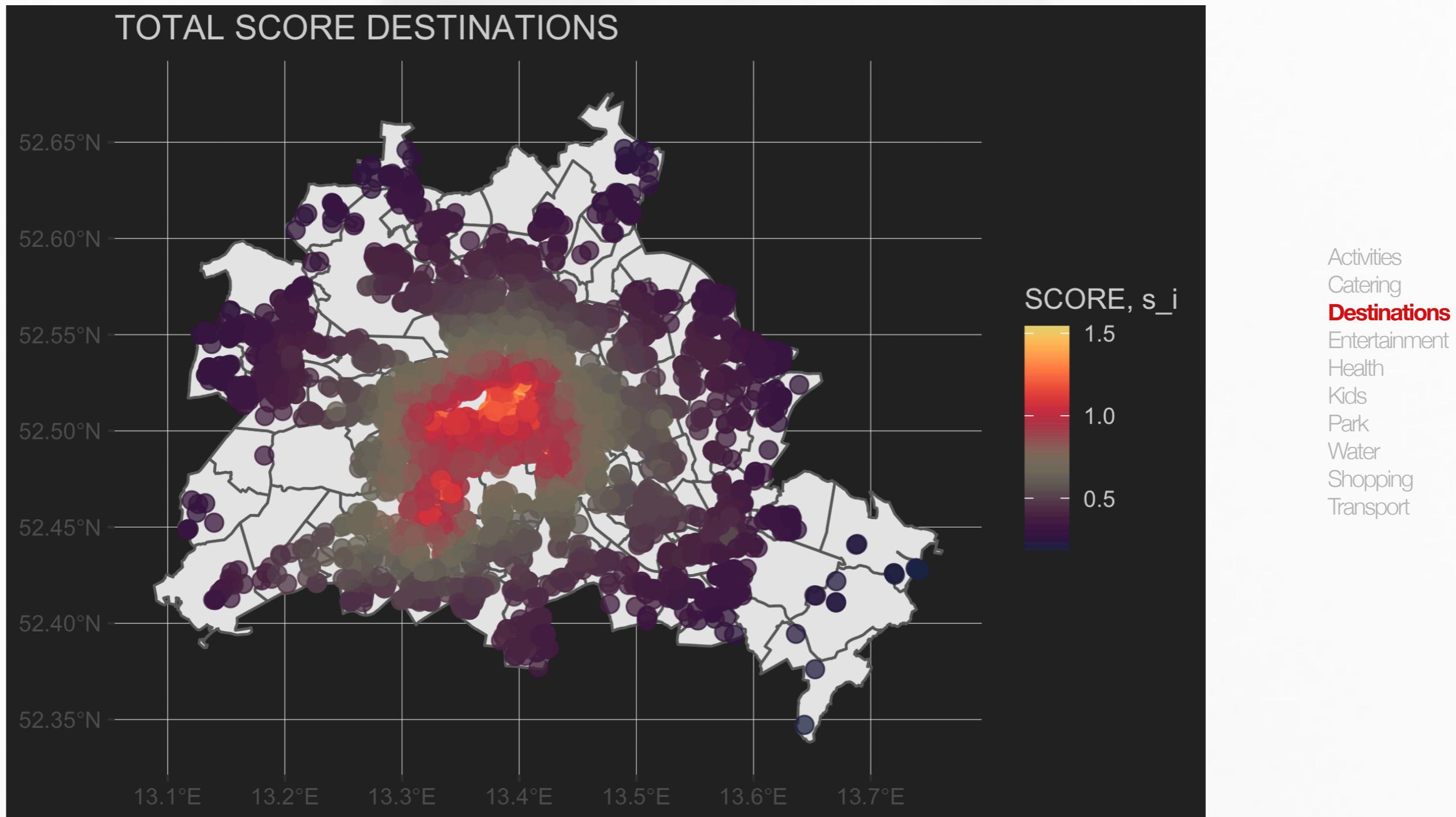
More maps: gyms, sports objects, etc.



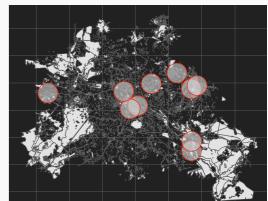
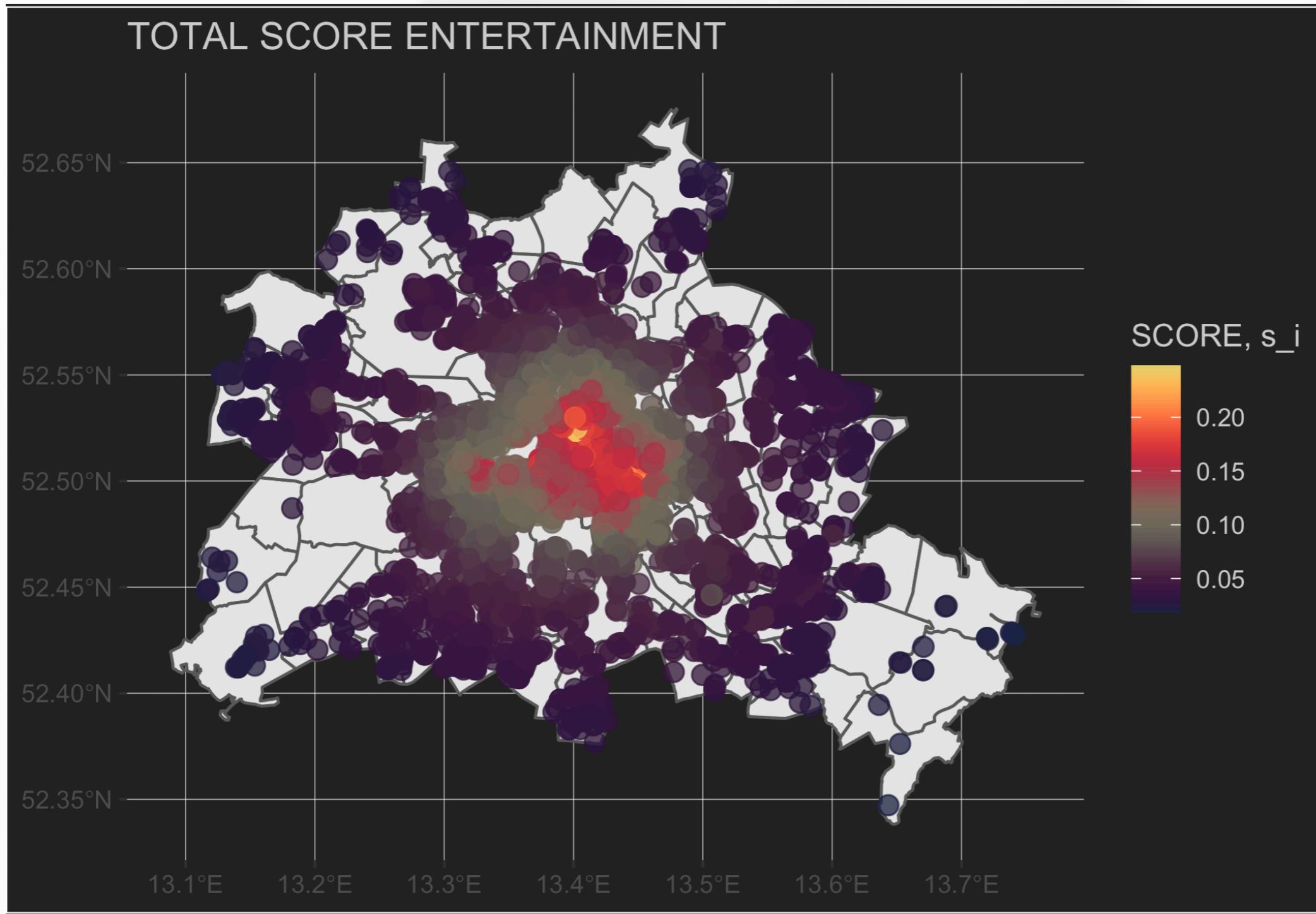
More maps: restaurants, bars, etc.



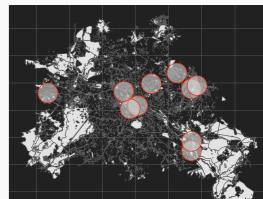
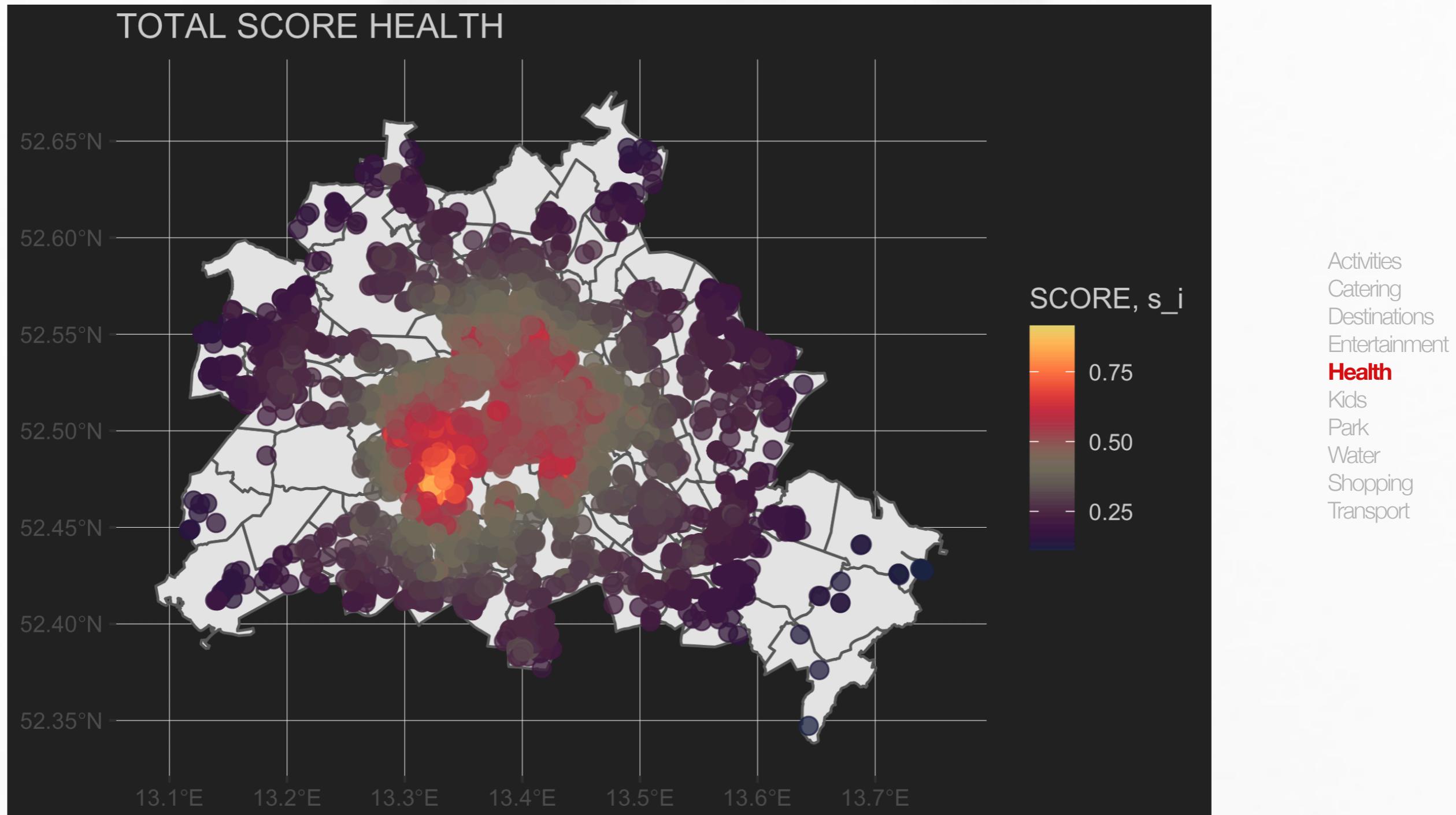
More maps: sightseeing, museums, etc.



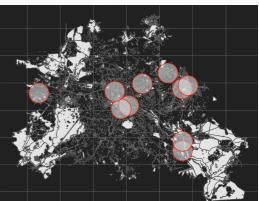
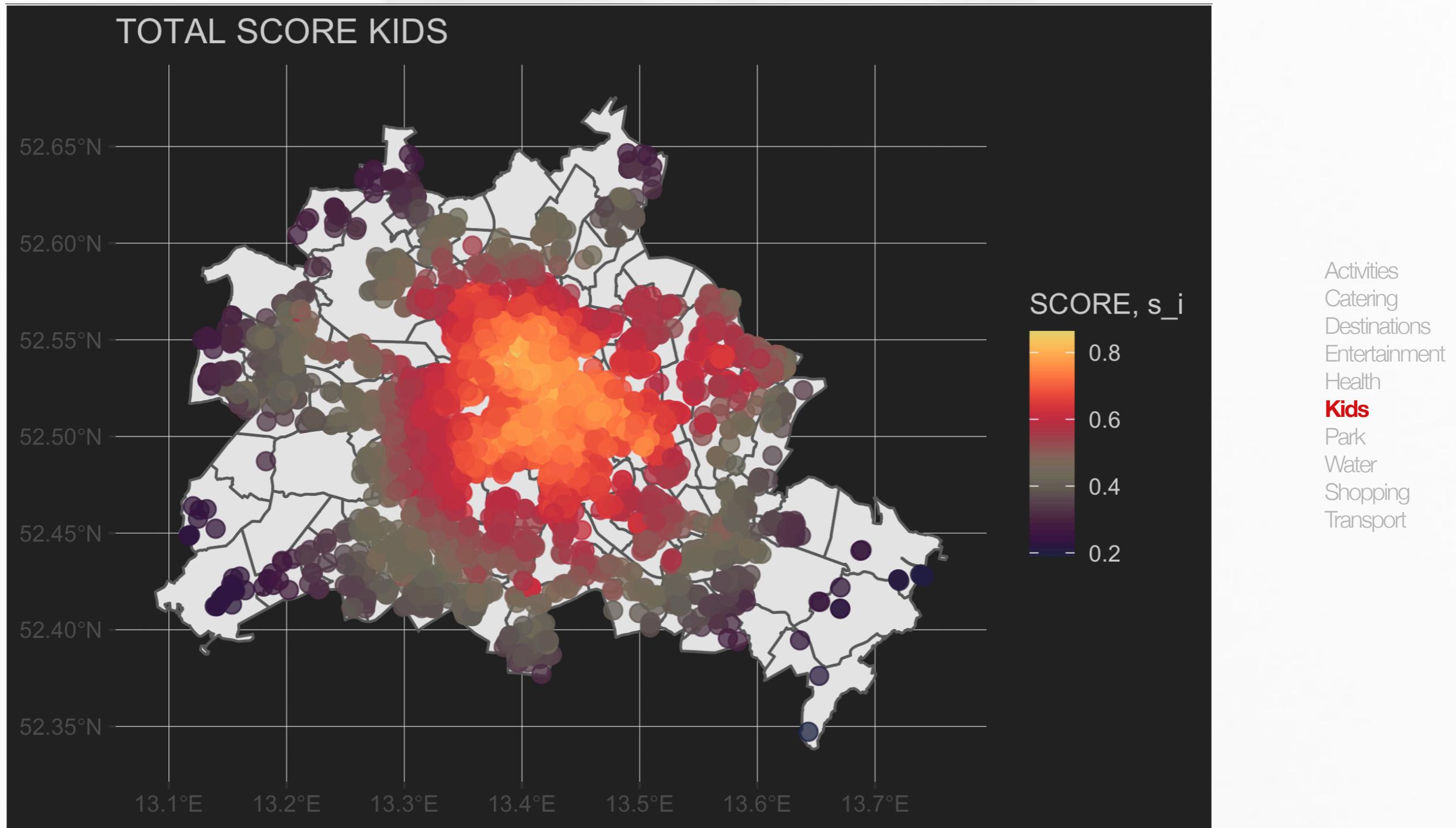
More maps: cinemas, night clubs, etc.



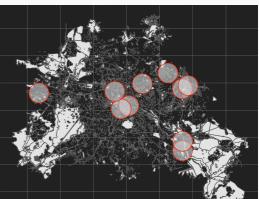
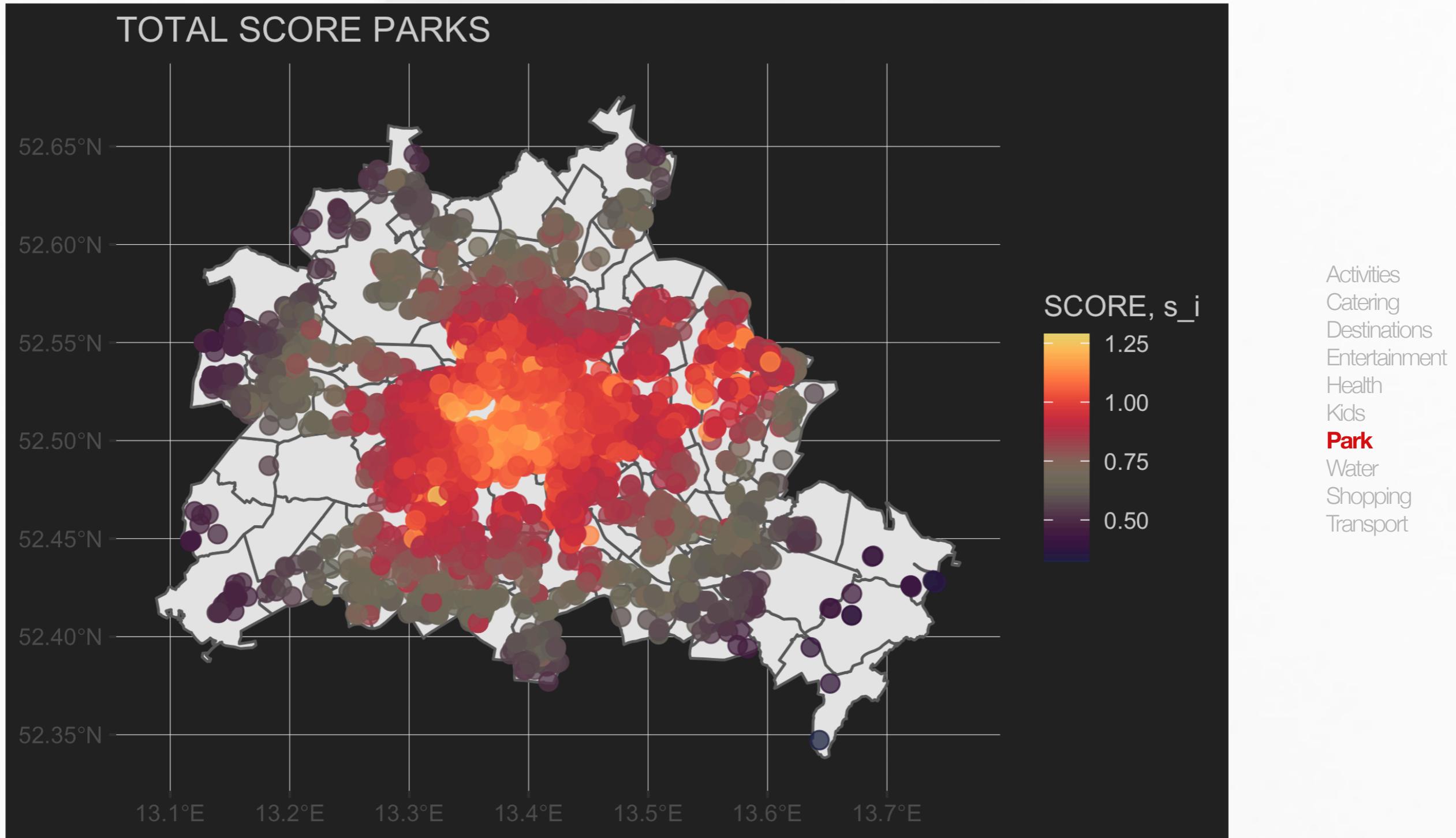
More maps: doctors, pharmacies, etc.



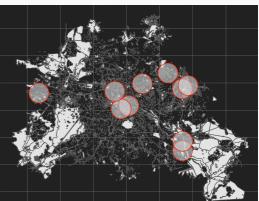
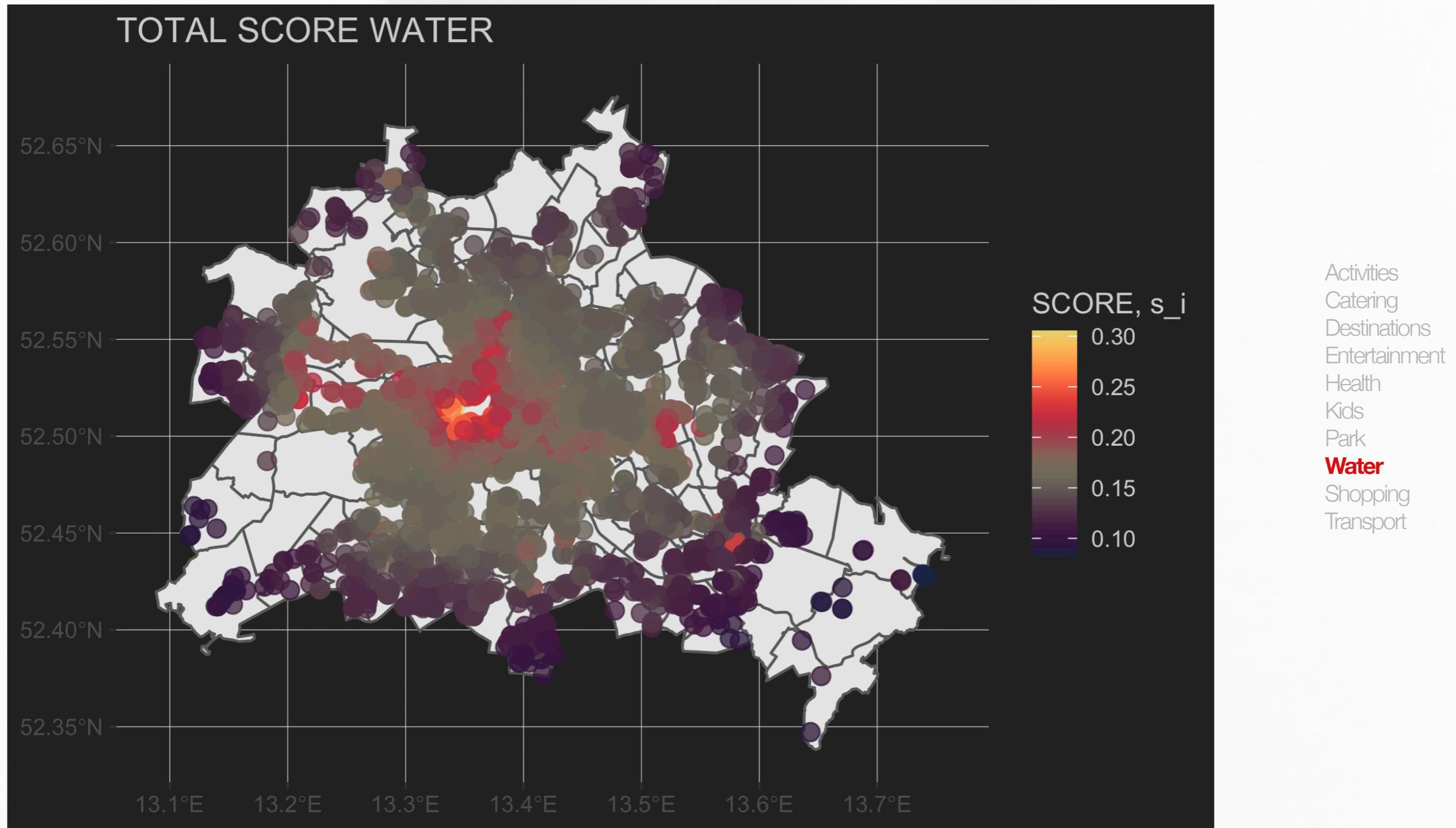
More maps: kitas, playgrounds, etc.



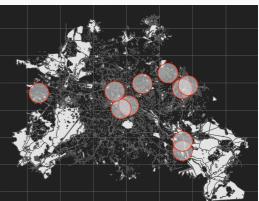
More maps: parks, dog parks, etc.



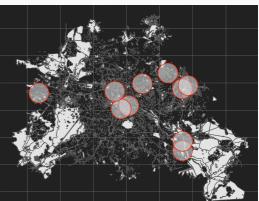
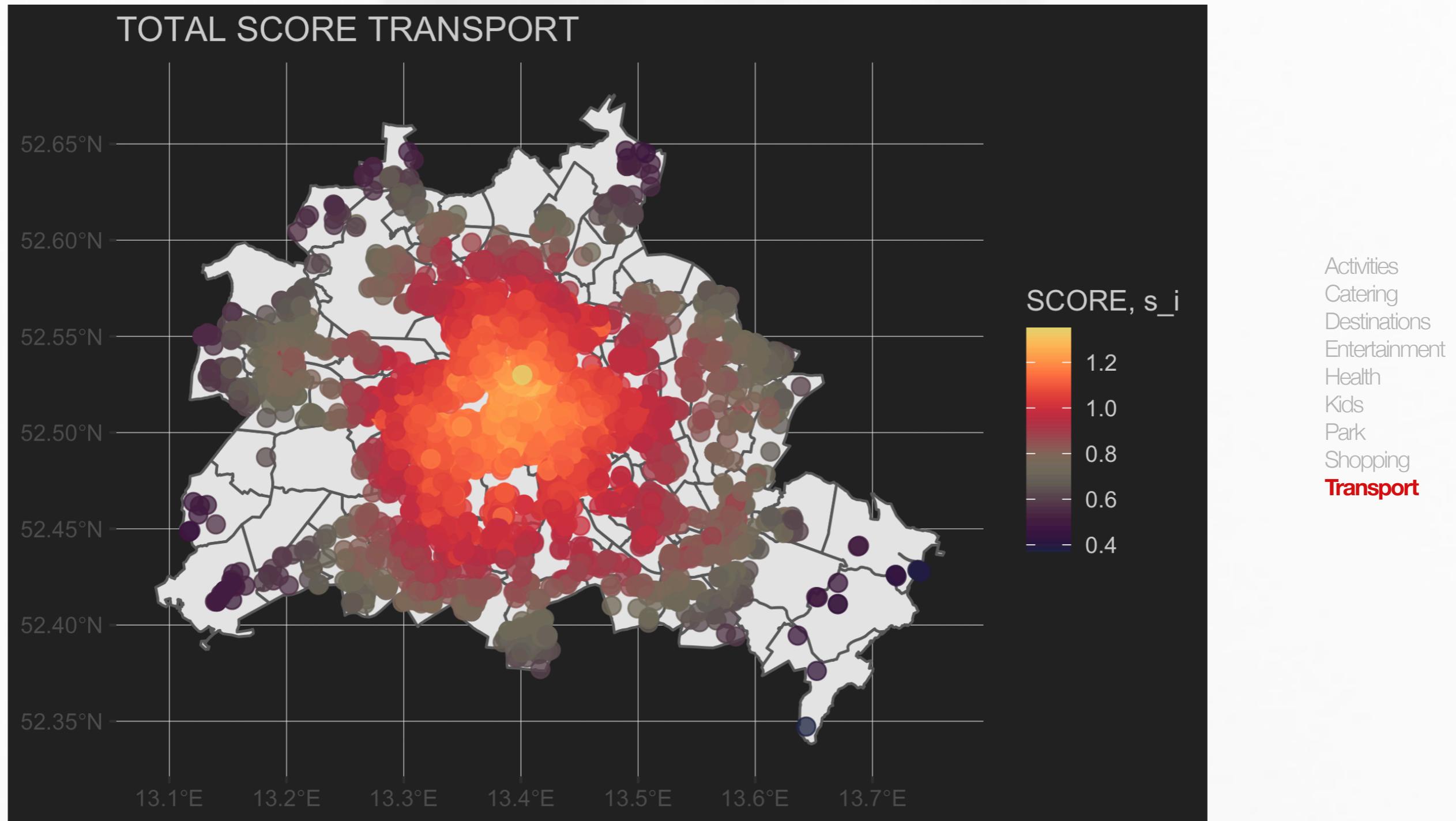
More maps: rivers, lakes etc.



More maps: shops, grocery stores, etc.



More maps: S-Bahn, U-Bahn, etc.



Distance measurement, scoring and normalization

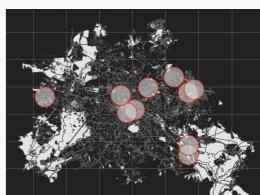
After that the scores are rescaled via min-max normalisation in order to get the weights:

$$s_{i,k}^{norm} = \frac{s_{i,k}^{dist} - \min s_k^{dist}}{\max s_k^{dist} - \min s_k^{dist}}$$

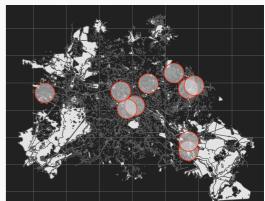
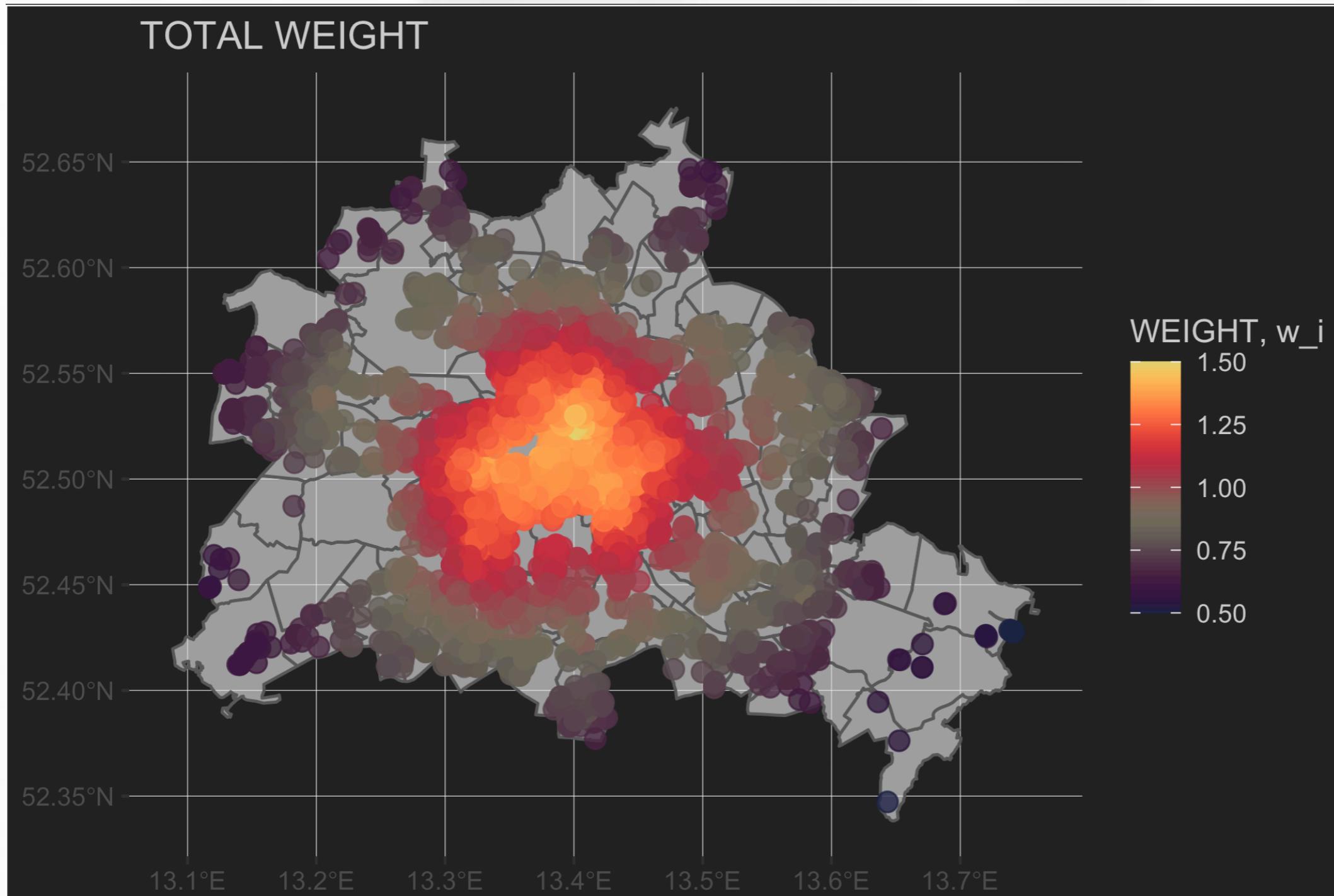
$$s_i^{norm} = \sum_k s_{i,k}^{norm}$$

$$w_i = \frac{s_i^{norm} - \min s^{norm}}{\max s^{norm} - \min s^{norm}} + \frac{1}{2}$$

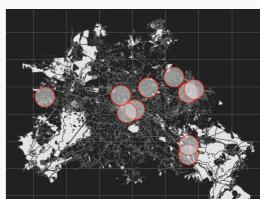
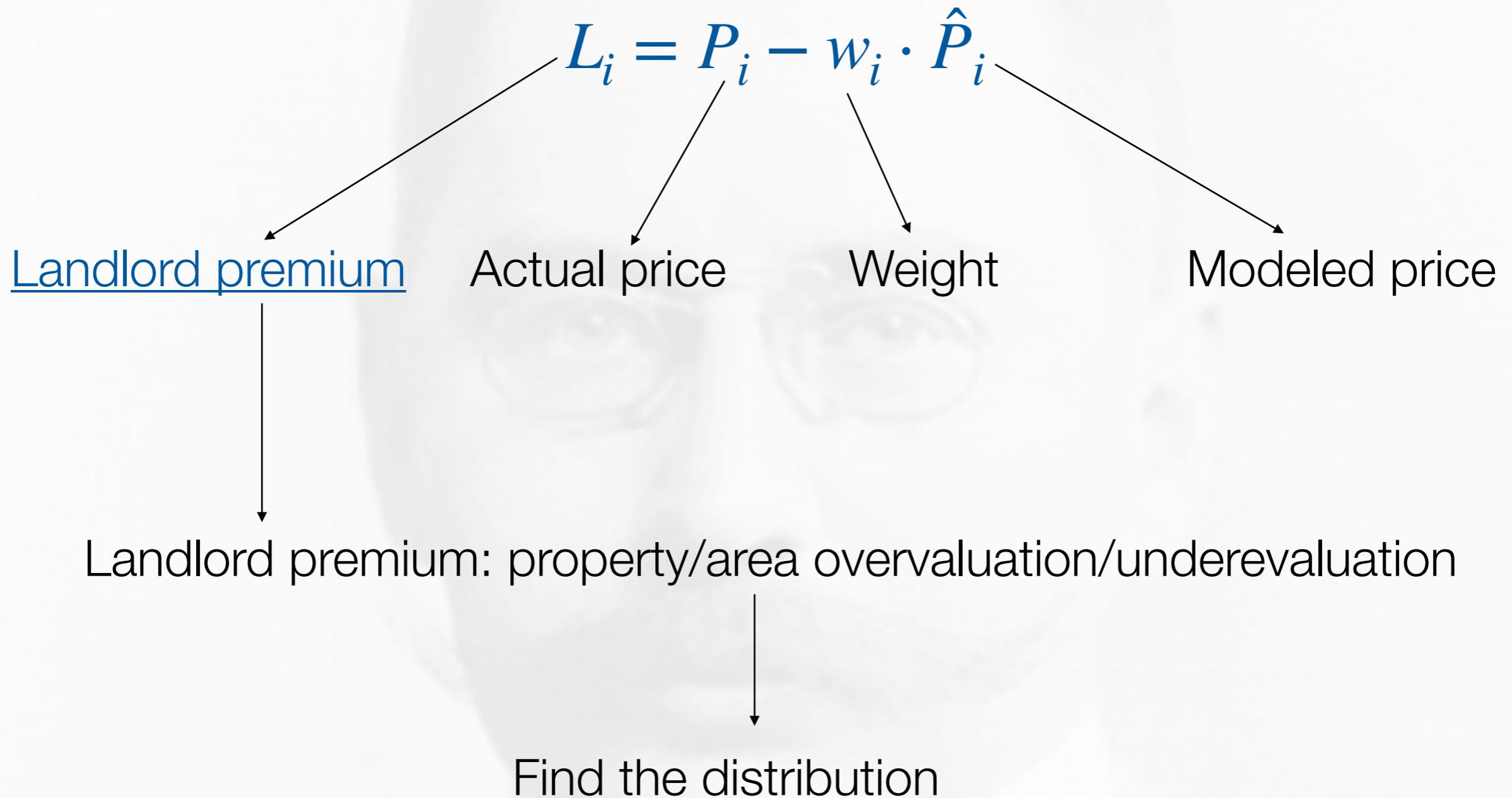
Here $\frac{1}{2}$ is added in order to make the minimum weight 0.5 and the maximum 1.5 (half-price discount or extra payment)



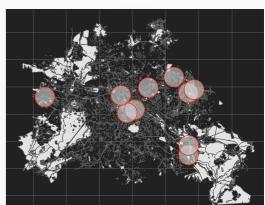
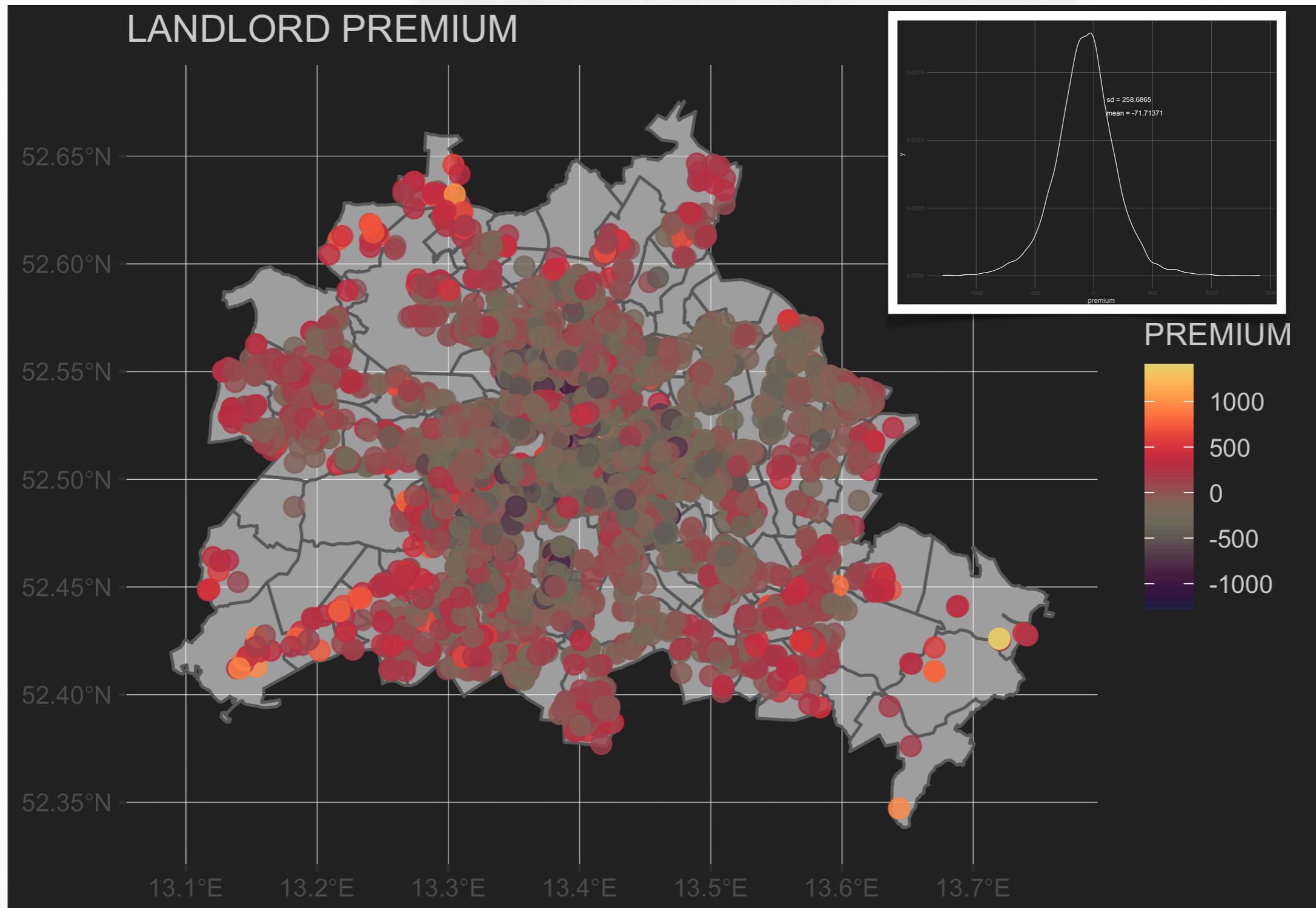
Weights



Evaluating landlord premiums



Premiums



Key takeaways:

- The size of the apartment, the presence of a kitchen, and a lift can be used as a solid foundation for a price check for the rent prices of apartments
- Infrastructural-wise Berlin tends to be a very single-centroid oriented city with few other areas of concentration emerging from different infrastructural objects
- Overall in Berlin, rent prices are not under- or overvalued based on the model used in this research, prices seem to be landlord premiums seem to be fairly normally distributed
- Overvalued properties seem to be more concentrated further outside the city centre and the more undervalued properties are located closer to the city center which can point to the lack of prestige living in the centre of the city

