

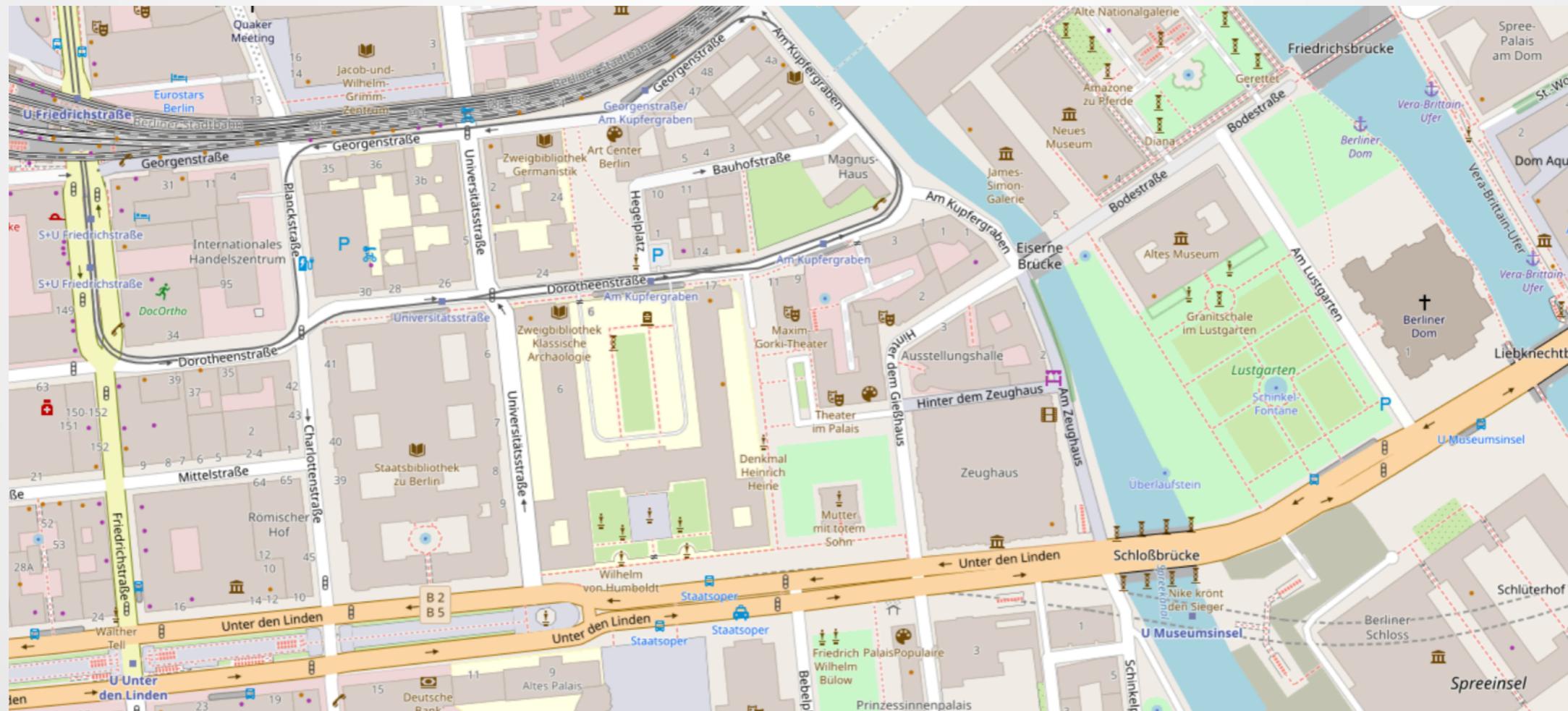
# Digital Economy and Decision Analytics - Blockchain and Cryptocurrency Seminar (WS 21/22) project

Kotik Ivan

Humboldt-Universität zu Berlin  
IRTG 1792  
Ladislaus von Bortkiewicz Professor of Statistics  
BRC Blockchain Research Center  
Charles University, WISE XMU, NCTU 玉山學者

# 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.

OpenStreetMap Data in Layered GIS Format // Free Shapefiles	
<b>OpenStreetMap Data in Layered GIS Format</b>	
Free shapefiles – 2021-07-08	
Frederik Ramm < <a href="mailto:frederik.ramm@geofabrik.de">frederik.ramm@geofabrik.de</a> >	
<b>Table of Contents</b>	
1 Preface.....	2
2 Introduction.....	2
2.1 Versions.....	2
2.2 Map datum.....	3
2.3 Character encoding.....	3
2.4 Attribute names.....	3
2.5 Common attributes.....	3
2.6 International Names.....	3
2.7 Layers.....	4
2.8 Points and Areas.....	4
2.9 Spillover Shape Files.....	4
3 Feature Catalogue - Overview.....	4
4 Point Features.....	5
4.1 Places ("places").....	5
4.2 Points of Interest.....	6
4.3 Places of Worship ("pofw").....	11
4.4 Natural Features ("natural").....	11
4.5 Traffic Related ("traffic").....	12
4.6 Transport Infrastructure ("transport").....	13
5 Line Features.....	14
5.1 Roads and Paths ("roads").....	14
5.2 Railways, Subways, Trams, Lifts, and Cable Cars ("railways").....	15
5.3 Waterways ("waterways").....	16
6 Polygon Features.....	16
6.1 Building outlines ("buildings").....	16
6.2 Land use and land cover ("landuse").....	17
6.3 Bodies of Water ("water").....	17
7 Differences to Commercial "0.7" Shape Files.....	18
7.1 Associated Area Layer.....	18
7.2 Additional Layers.....	18
7.3 Additional Feature Classes.....	18
7.4 Additional Fields.....	19

## 4.2 Points of Interest

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

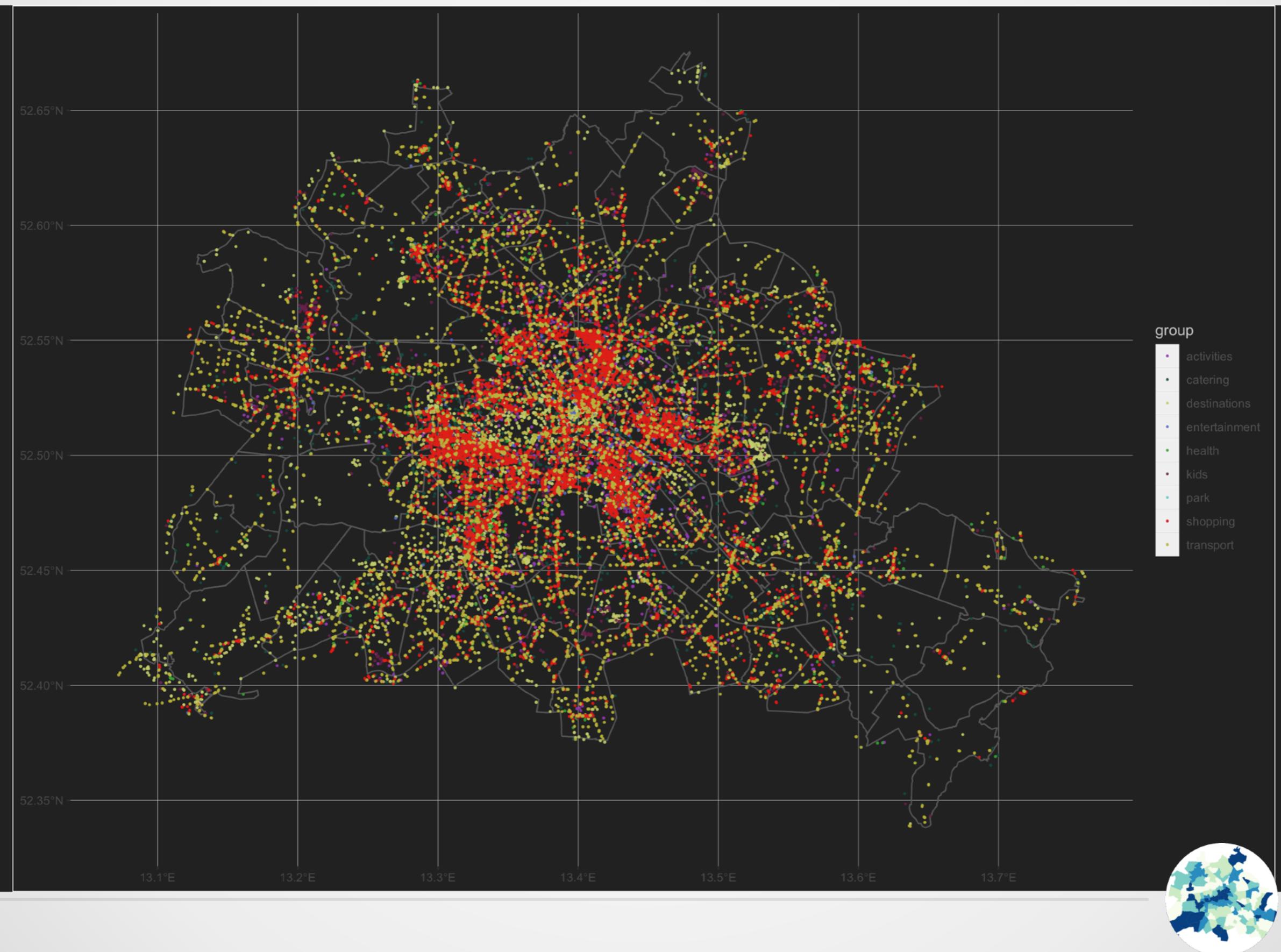
The following feature classes exist in this layer:

code	layer	fclass	Description	OSM Tags
20xx	<b>public</b>			
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
...				

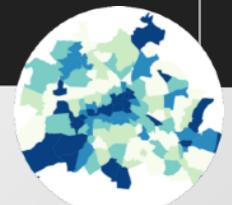
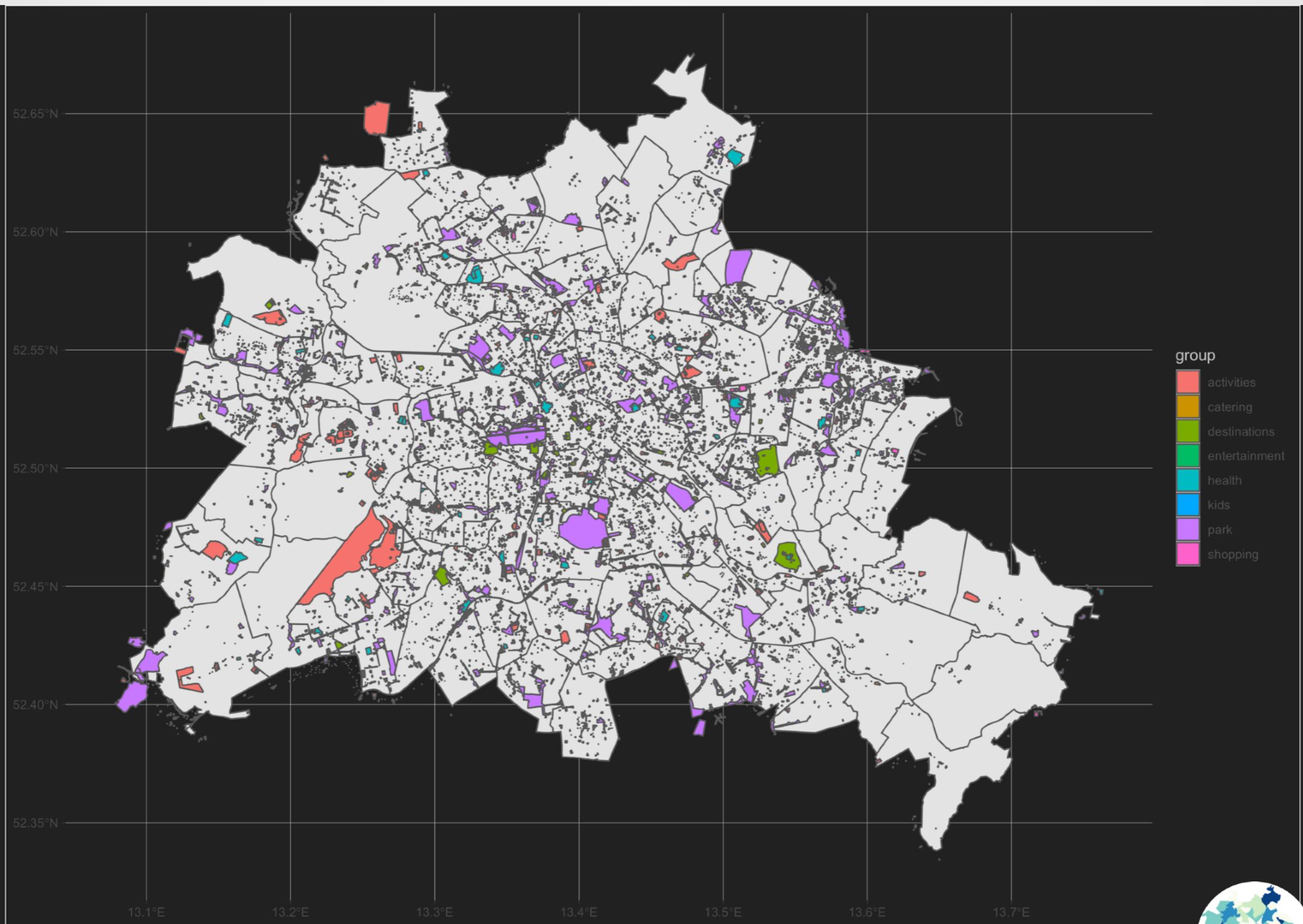
code	layer	fclass	Description	OSM Tags
22xx	<b>leisure</b>			
2201		theatre	A theatre.	amenity=theatre
2202		nightclub	A night club, or disco.	amenity=nightclub
2203		cinema	A cinema.	amenity=cinema
2204		park	A park.	leisure=park
2205		playground	A playground for children.	leisure=playground
2206		dog_park	An area where dogs are allowed to run free without a leash.	leisure=dog_park
Sports				
225x		sports_centre	A facility where a range of sports activities can be pursued.	leisure=sports_centre
2251				
2252		pitch	An area set aside for a specific sport.	leisure=pitch
2253		swimming_pool	A swimming pool or water park.	amenity=swimming_pool, leisure=swimming_pool, sport=swimming, leisure=water_park



# How the data looks like (polygons)



# How the data looks like (multi-polygons)



## The counter-table

After cleaning data is it time to calculate how many infrastructural objects are located inside the Bezirks of Berlin.

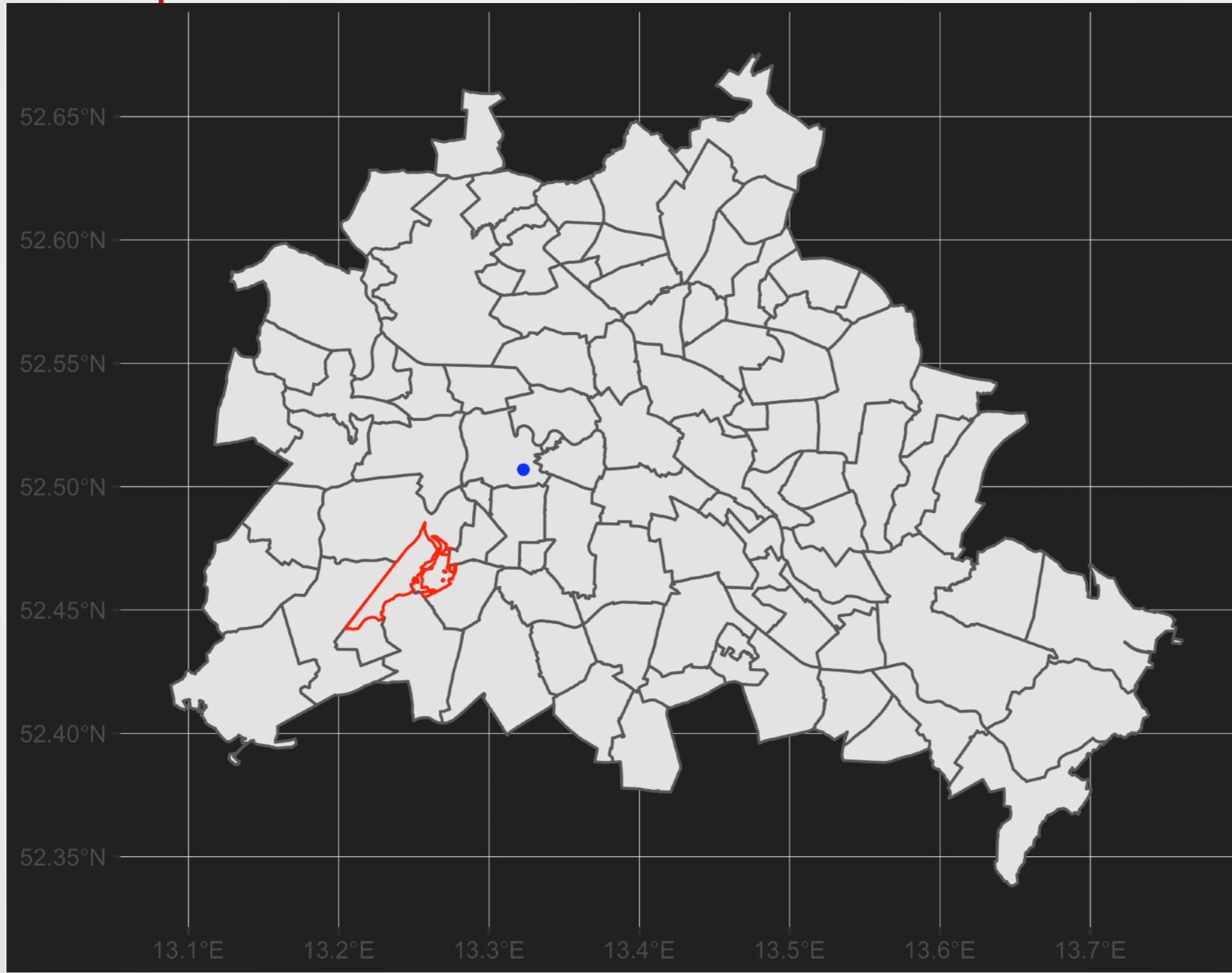
For polygon data the count goes on based on whether the point is inside the multi polygon of the Bezirk



For multipolygon data I calculate how much of the area intersects between the shapes



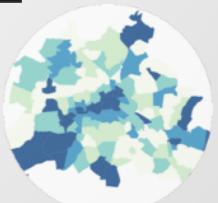
## Example



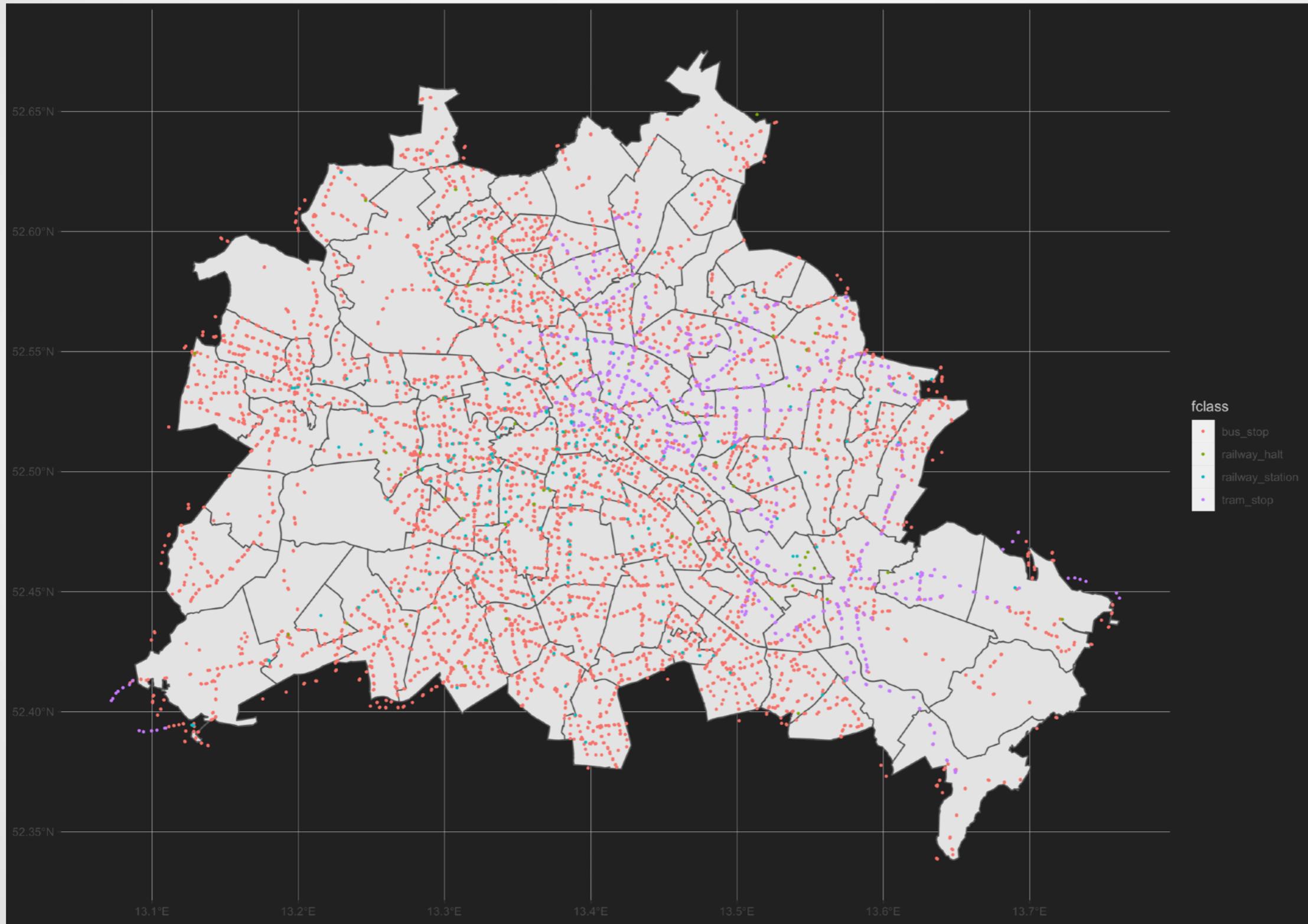
# The counter-table

The resulting data:

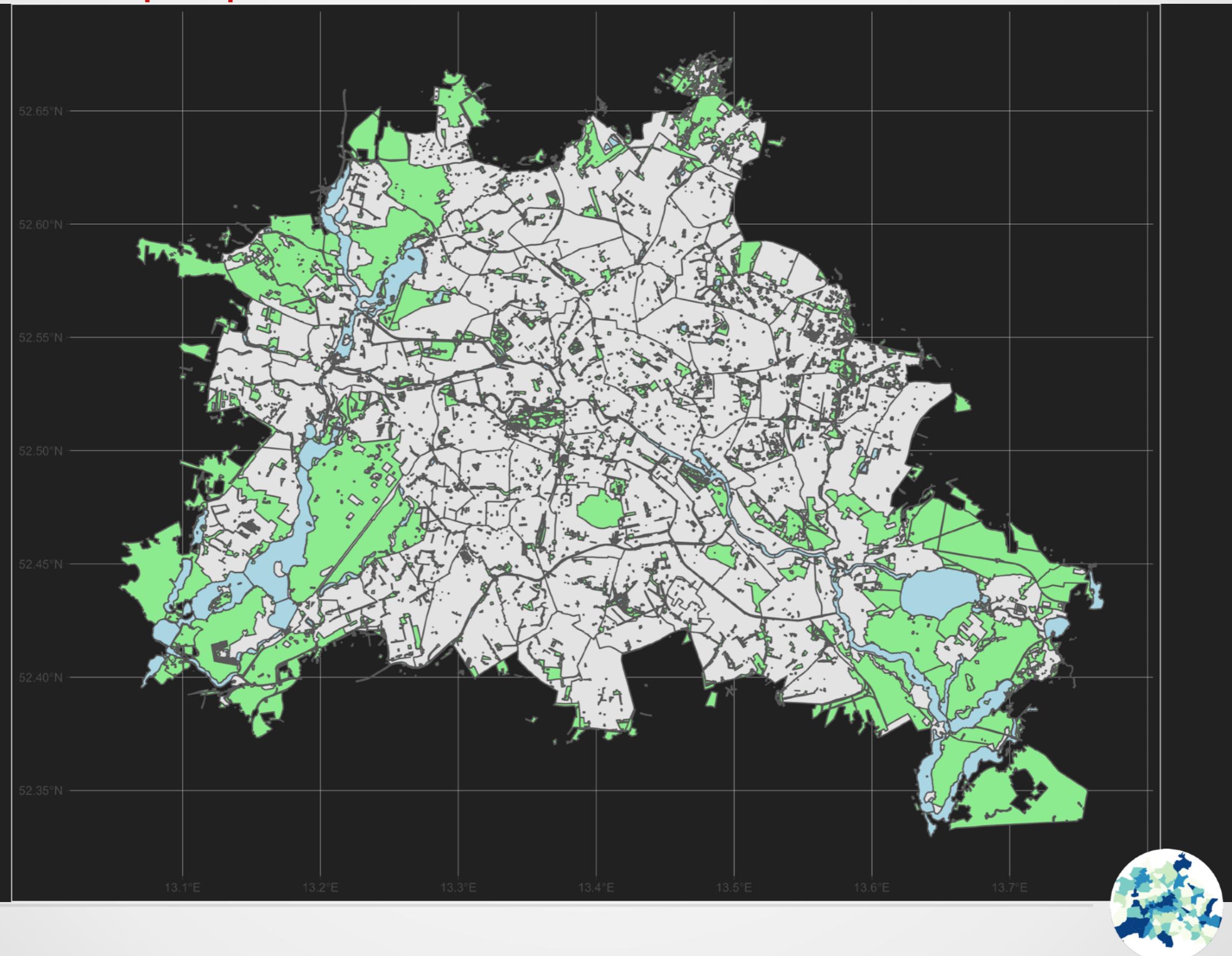
	bezirk	activities	catering	destinations	entertainment	health	kids		park	shopping	transport	water
1	Adlershof	35	65	15	2	26	26	765109.71 [m^2]	111	0	110962.9460 [m^2]	
2	Alt-Hohenschönhausen	62	65	18	0	12	57	627849.70 [m^2]	159	0	114598.2137 [m^2]	
3	Alt-Treptow	13	59	32	7	24	16	2703963.82 [m^2]	76	0	69677.4434 [m^2]	
4	Altglienicke	91	23	12	1	11	27	1702878.99 [m^2]	69	0	13869.5570 [m^2]	
5	Baumschulenweg	32	36	13	0	7	18	442526.53 [m^2]	75	0	80191.8301 [m^2]	
6	Biesdorf	47	29	13	2	13	46	2915368.93 [m^2]	94	0	160110.1798 [m^2]	
7	Blankenburg	9	4	0	0	4	7	232503.29 [m^2]	15	0	3741.7319 [m^2]	
8	Blankenfelde	10	4	14	0	0	3	4642426.26 [m^2]	4	0	278218.7221 [m^2]	
9	Bohnsdorf	19	12	6	0	10	11	474653.00 [m^2]	43	0	2862.1893 [m^2]	
10	Borsigwalde	20	11	0	0	5	9	87421.89 [m^2]	35	0	517.8005 [m^2]	
11	Britz	109	78	74	2	17	77	4128021.97 [m^2]	109	0	303221.3470 [m^2]	
12	Buch	24	17	36	0	5	25	7478168.17 [m^2]	36	0	222901.9041 [m^2]	
13	Buckow	34	48	6	2	21	60	262988.63 [m^2]	69	0	4991.3043 [m^2]	
14	Charlottenburg	115	974	264	40	130	82	1862658.96 [m^2]	1399	0	341122.8299 [m^2]	
15	Charlottenburg-Nord	43	18	12	0	4	24	1444173.00 [m^2]	36	0	389435.6137 [m^2]	
16	Dahlem	60	59	87	3	14	32	2588019.98 [m^2]	49	0	27856.8855 [m^2]	
17	Falkenberg	5	1	10	0	0	6	334092.95 [m^2]	4	0	14722.1584 [m^2]	
18	Falkenhagener Feld	73	18	8	0	4	33	1664381.46 [m^2]	27	0	187732.0863 [m^2]	
19	Fennpfuhl	52	35	17	0	10	27	626735.80 [m^2]	60	0	20844.7992 [m^2]	
20	Französisch Buchholz	19	14	6	0	4	30	1155676.52 [m^2]	47	0	171005.8046 [m^2]	
21	Friedenau	30	120	103	4	114	16	42890.41 [m^2]	215	0	NA [m^2]	
22	Friedrichsfelde	104	58	97	3	32	124	1265834.02 [m^2]	141	0	30681.1121 [m^2]	
23	Friedrichshagen	37	47	17	3	9	16	7589062.22 [m^2]	84	0	1966711.9260 [m^2]	
24	Friedrichshain	156	712	113	46	94	131	1714410.44 [m^2]	921	0	1066155.5237 [m^2]	
25	Frohnau	22	17	7	1	21	5	2476489.92 [m^2]	47	0	29020.3939 [m^2]	
26	Gatow	9	12	1	0	1	8	2808994.10 [m^2]	3	0	1112743.4493 [m^2]	
27	Gesundbrunnen	131	233	41	11	42	110	1453280.64 [m^2]	359	0	21582.7832 [m^2]	
28	Gropiusstadt	45	22	2	1	9	42	408288.48 [m^2]	90	0	1172.4550 [m^2]	
29	Grünau	15	15	8	0	1	6	5724203.70 [m^2]	19	0	985384.2882 [m^2]	

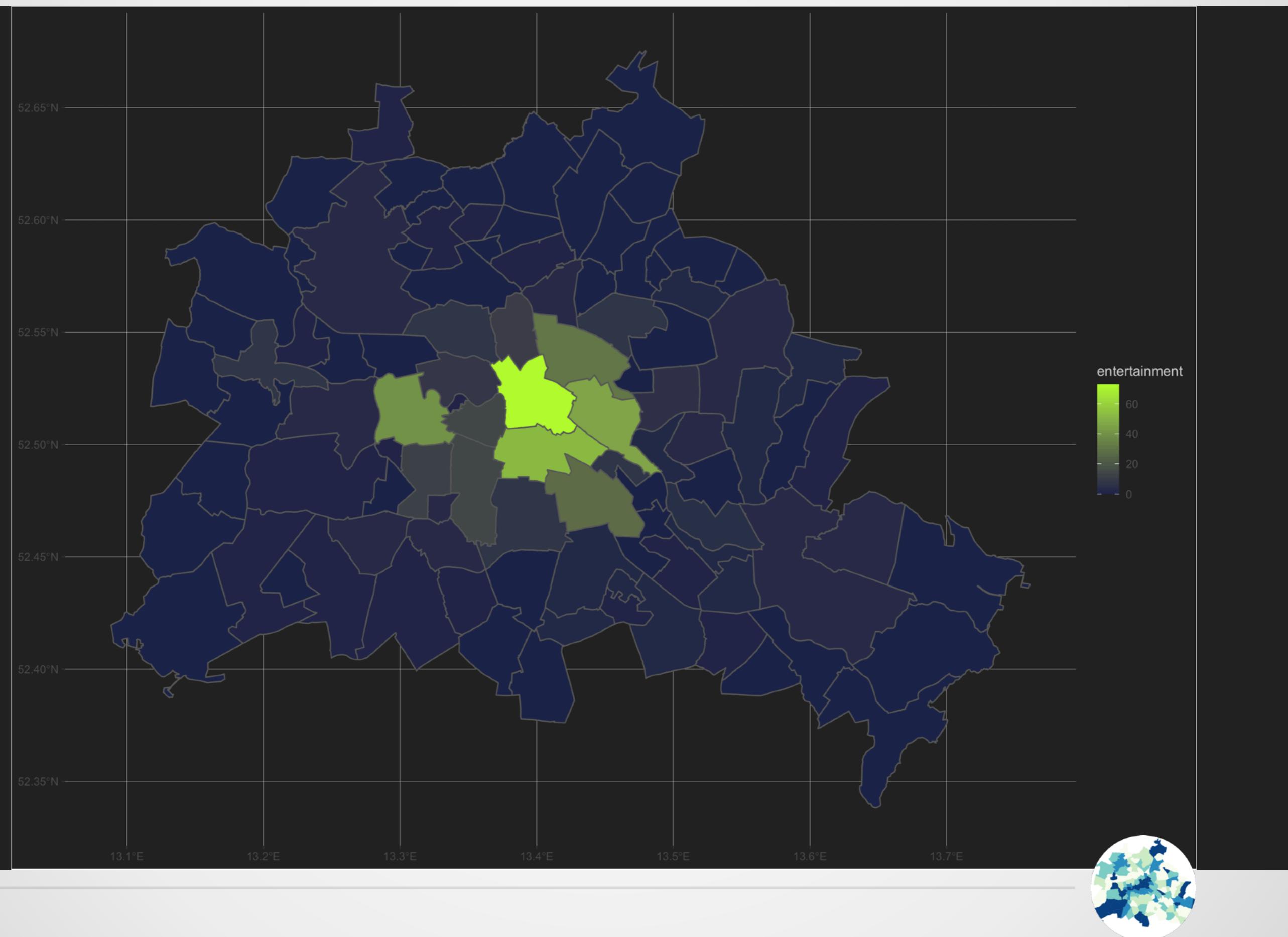


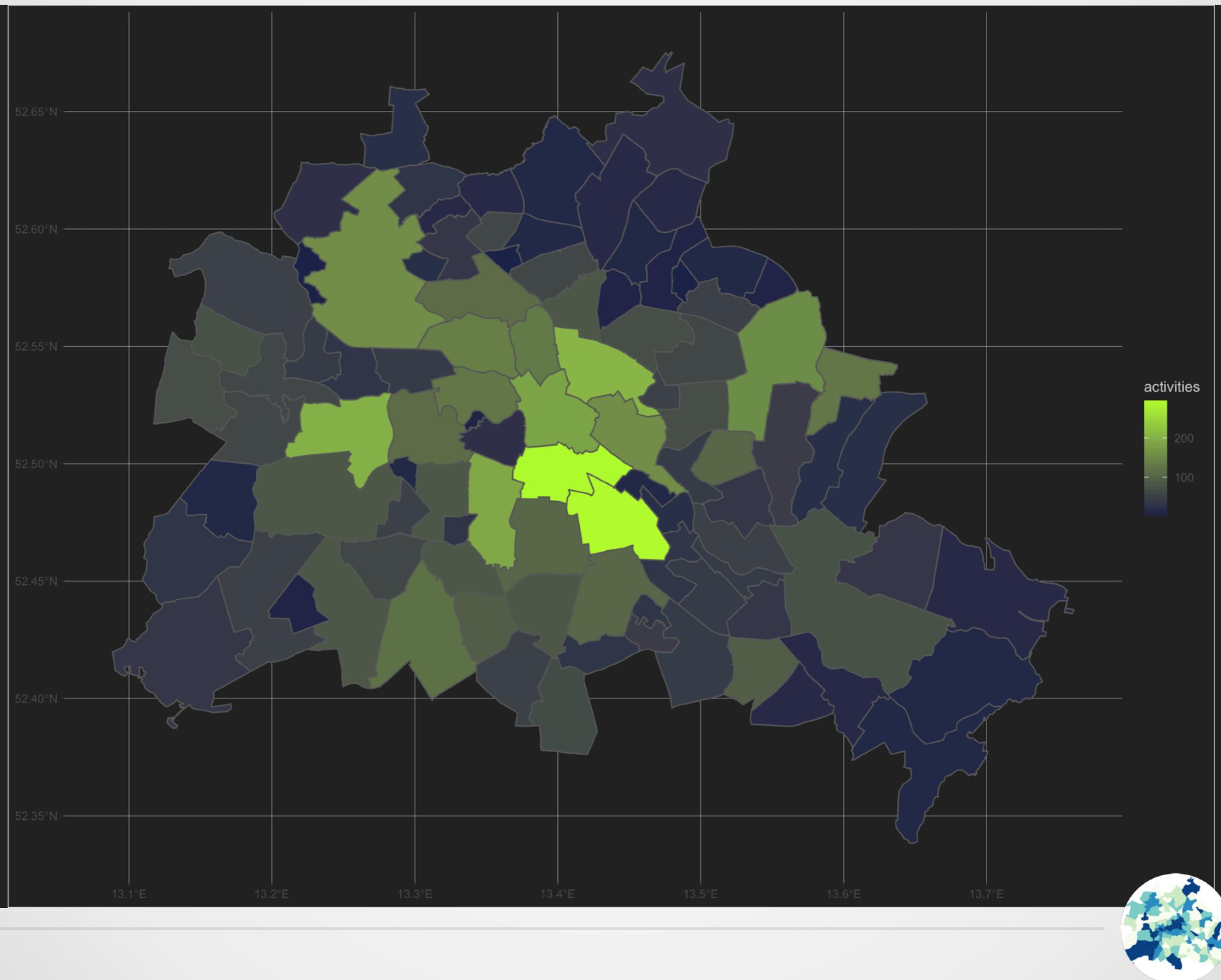
## More maps: transport

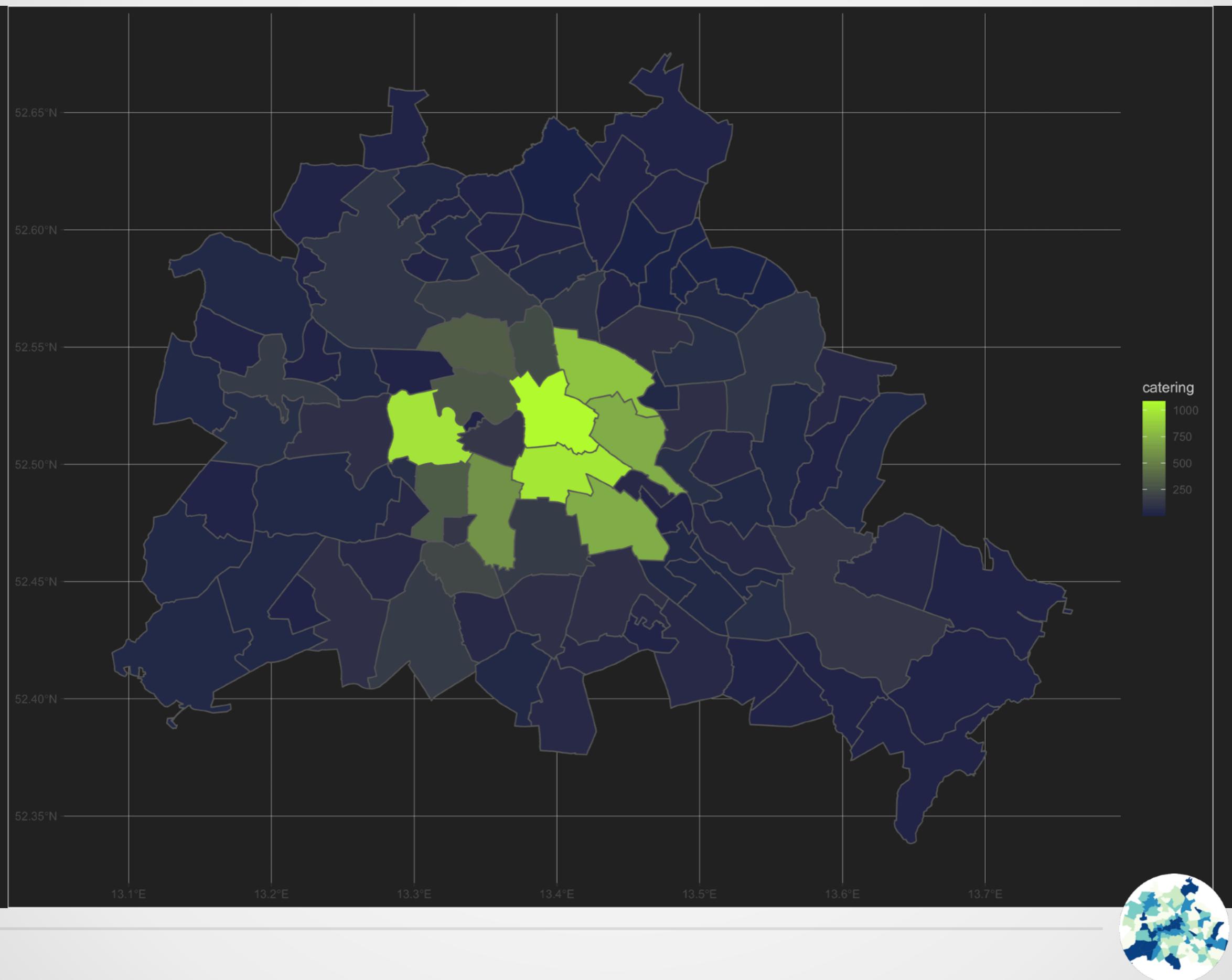


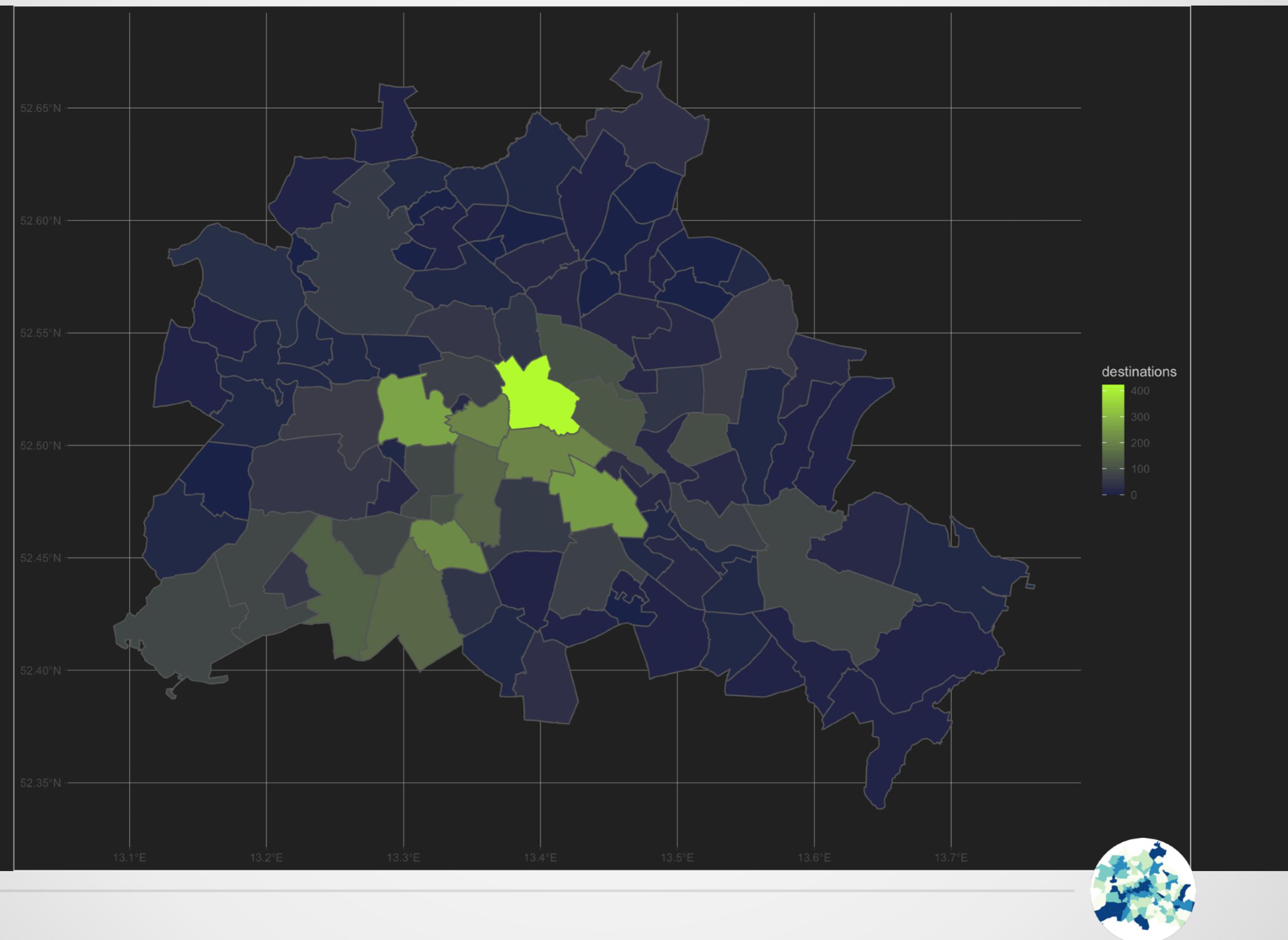
## More maps: parks and rivers

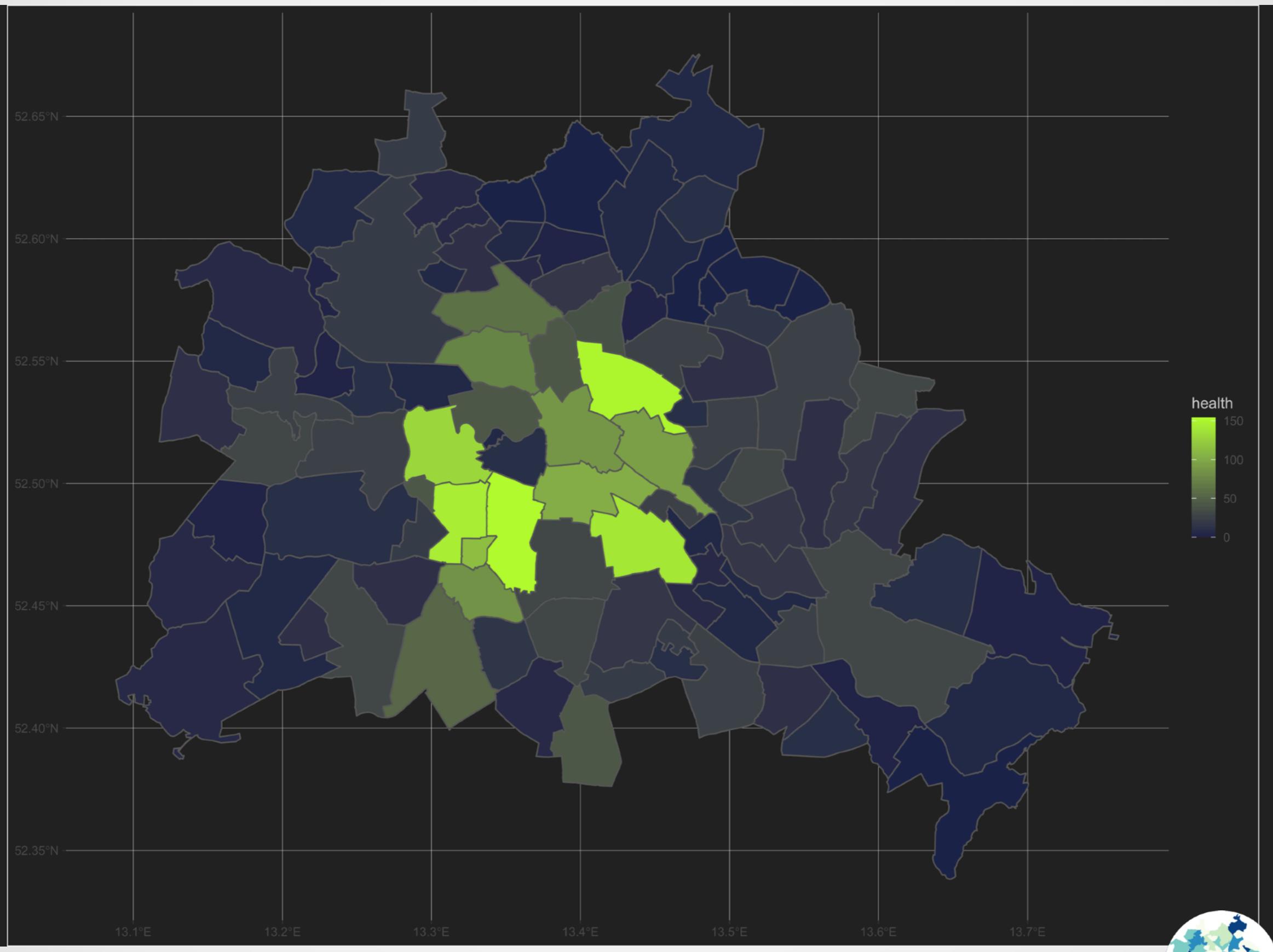


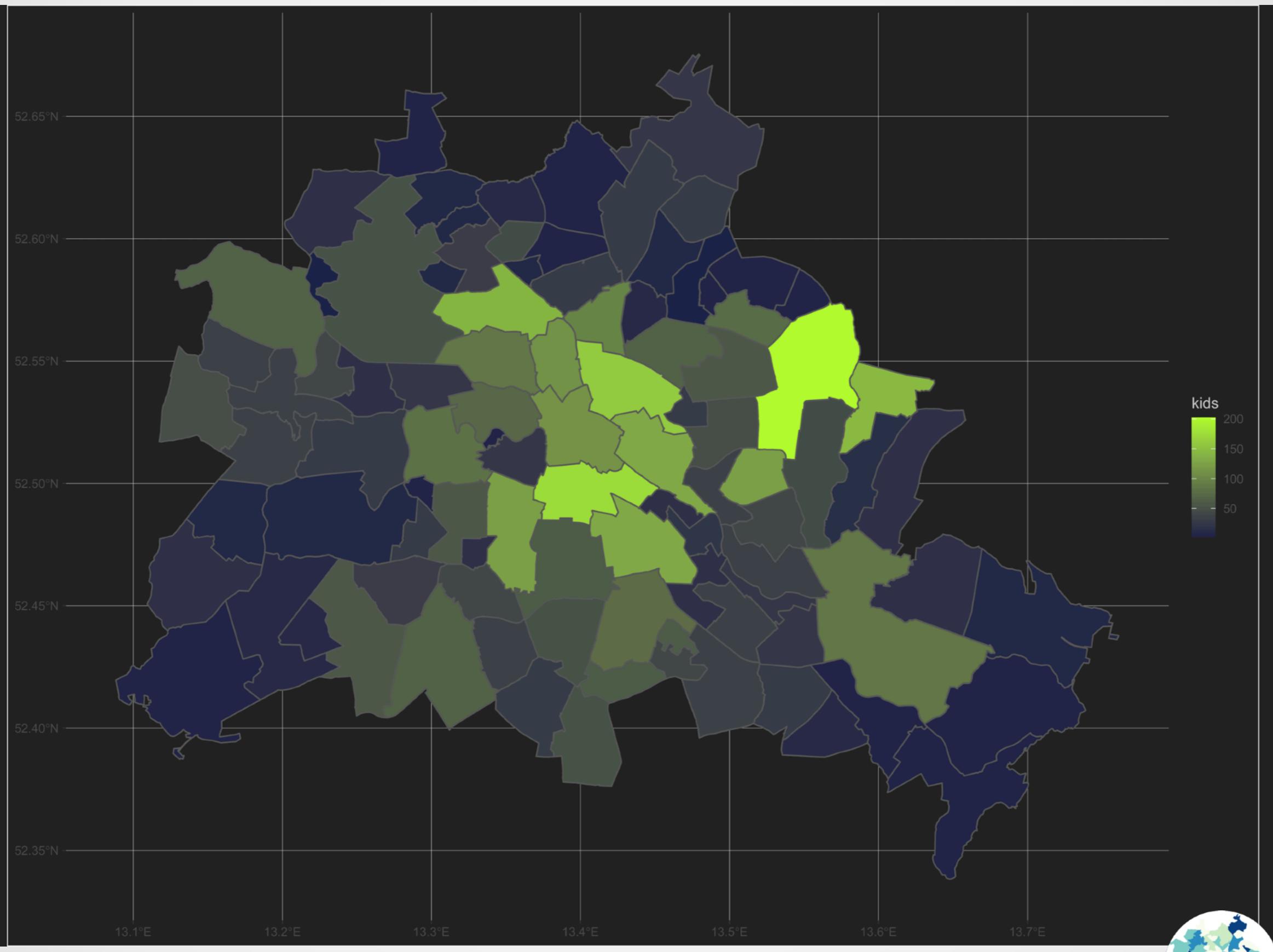


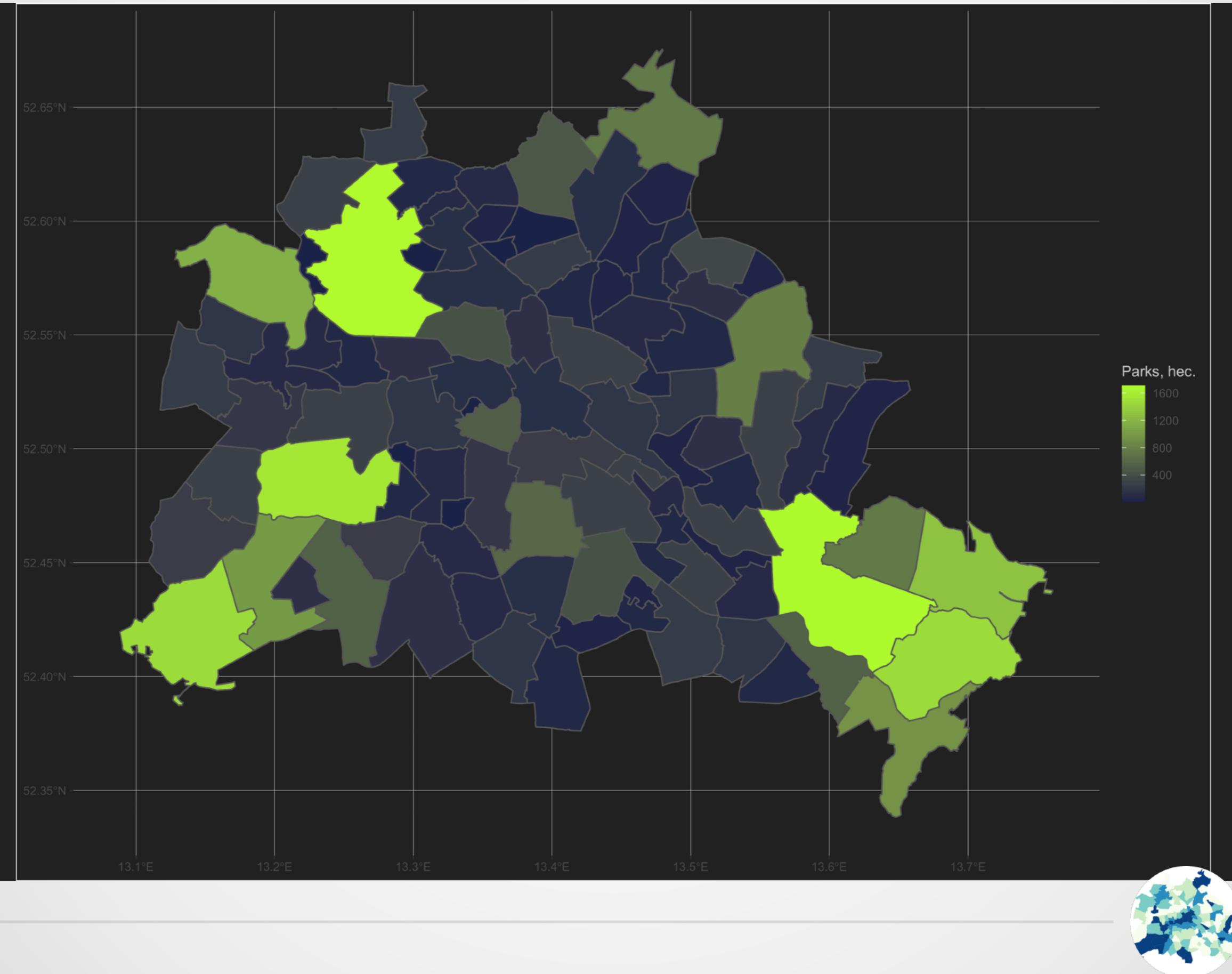


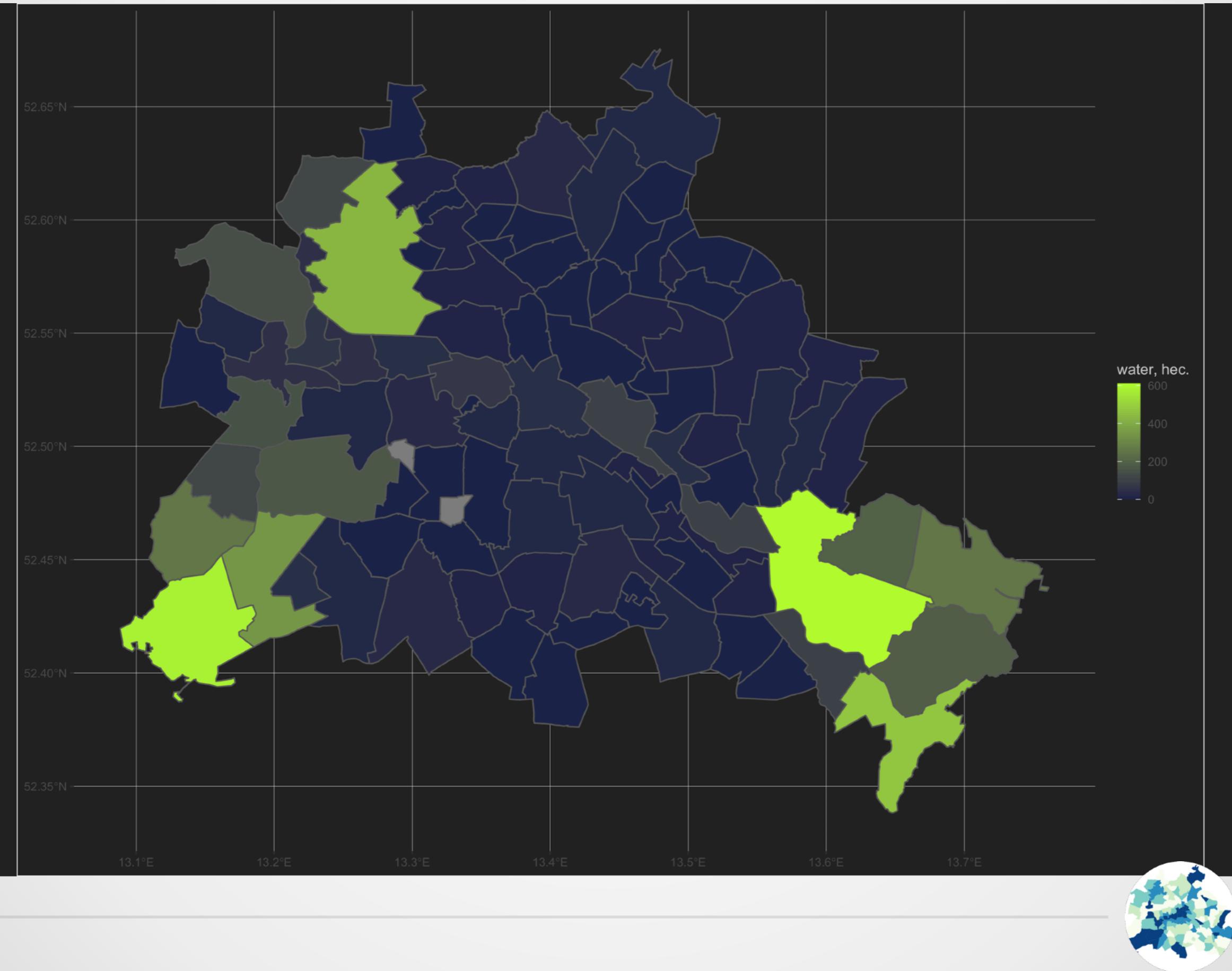


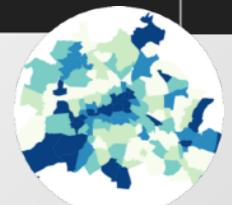
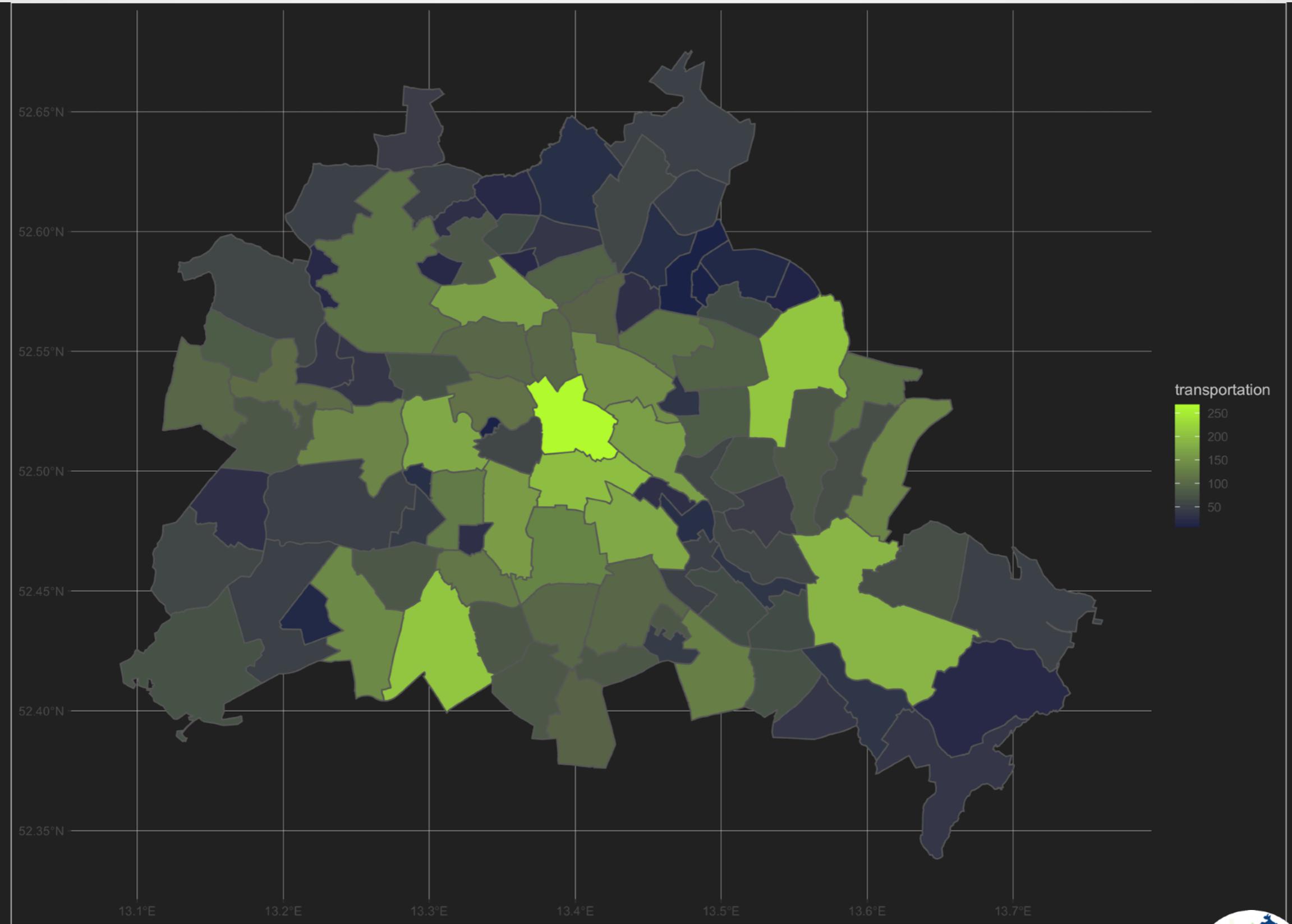












# Spatial framework done.

What now?  
Rent prices!

```
✓ └── data = Table: 268,850 x 49 ... View Table
    > ⚡ regio1 = "Nordrhein_Westfalen" "Rheinland_Pfalz" "Sachsen" ... View
    > ⚡ serviceCharge = 245.00 134.00 255.00 58.15 138.00 142.00 70.00 88.00 110.00 95.00 ... View
    > ⚡ heatingType = "central_heating" "self_contained_central_heating" ... View
    > ⚡ telekomTvOffer = "ONE_YEAR_FREE" "ONE_YEAR_FREE" "ONE_YEAR_FREE" "ONE_YEAR_FREE" ... View
    > ⚡ telekomHybridUploadSpeed = NA NA 10 NA NA NA 10 10 NA NA NA 10 NA NA 10 10 NA NA NA ... View
    > ⚡ newlyConst = FALSE FALSE TRUE FALSE ... View
    > ⚡ balcony = FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE FALSE TRUE FALSE ... View
    > ⚡ picturecount = 6 8 8 9 19 5 9 5 5 7 11 9 4 3 12 5 17 12 5 35 ... View
    > ⚡ pricetrend = 4.62 3.47 2.72 1.53 2.46 4.48 1.01 1.89 3.77 1.92 0.21 0.98 2.77 3.45 3.76 ... View
    > ⚡ telekomUploadSpeed = 10.0 10.0 2.4 40.0 NA 2.4 2.4 2.4 40.0 40.0 40.0 2.4 2.4 40.0 2.4 ... View
    > ⚡ totalRent = 840.00 NA 1300.00 NA 903.00 NA 380.00 584.25 690.00 ... View
    > ⚡ yearConstructed = 1965 1871 2019 1964 1950 1999 NA 1959 1970 1953 1930 1892 1912 1951 2018 ... View
    > ⚡ scoutId = 96107057 111378734 113147523 108890903 114751222 115531145 114391930 ... View
    > ⚡ noParkSpaces = 1 2 1 NA NA NA NA NA 1 NA NA NA NA NA 1 NA NA NA NA 1 ... View
    > ⚡ firingTypes = "oil" "gas" NA "district_heating" ... View
    > ⚡ hasKitchen = FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE ... View
    > ⚡ geo_bln = "Nordrhein_Westfalen" "Rheinland_Pfalz" "Sachsen" ... View
    > ⚡ cellar = TRUE FALSE TRUE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE ... View
    > ⚡ yearConstructedRange = 2 1 9 2 1 5 NA 2 2 2 1 1 1 2 9 4 NA 1 1 8 ... View
    > ⚡ baseRent = 595.00 800.00 965.00 343.00 765.00 315.20 310.00 452.25 580.00 ... View
    > ⚡ houseNumber = "244" NA "4" "35" "10" "1e" "14" "35" NA "30" "28" NA ... View
    > ⚡ livingSpace = 86.00 89.00 83.80 58.15 84.97 53.43 62.00 60.30 53.00 60.00 ... View
    > ⚡ geo_krs = "Dortmund" "Rhein_Pfalz_Kreis" ... View
    > ⚡ condition = "well_kept" "refurbished" ... View
    > ⚡ interiorQual = "normal" "normal" "sophisticated" NA ... View
    > ⚡ petsAllowed = NA "no" NA NA NA ... View
    > ⚡ street = "Sch&uuml;rferstra&szlig;e" "no_information" ... View
    > ⚡ streetPlain = "Sch&uuml;rferstraße" NA ... View
    > ⚡ lift = FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE ... View
    > ⚡ baseRentRange = 4 5 6 2 5 2 2 3 4 1 1 2 5 6 6 1 5 1 2 7 ... View
    > ⚡ typeOfFlat = "ground_floor" "ground_floor" "apartment" "other" "apartment" ... View
    > ⚡ geo_plz = 44269 67459 1097 9599 28213 24891 9599 28717 79211 45888 9130 9669 ... View
```



# Clear the data

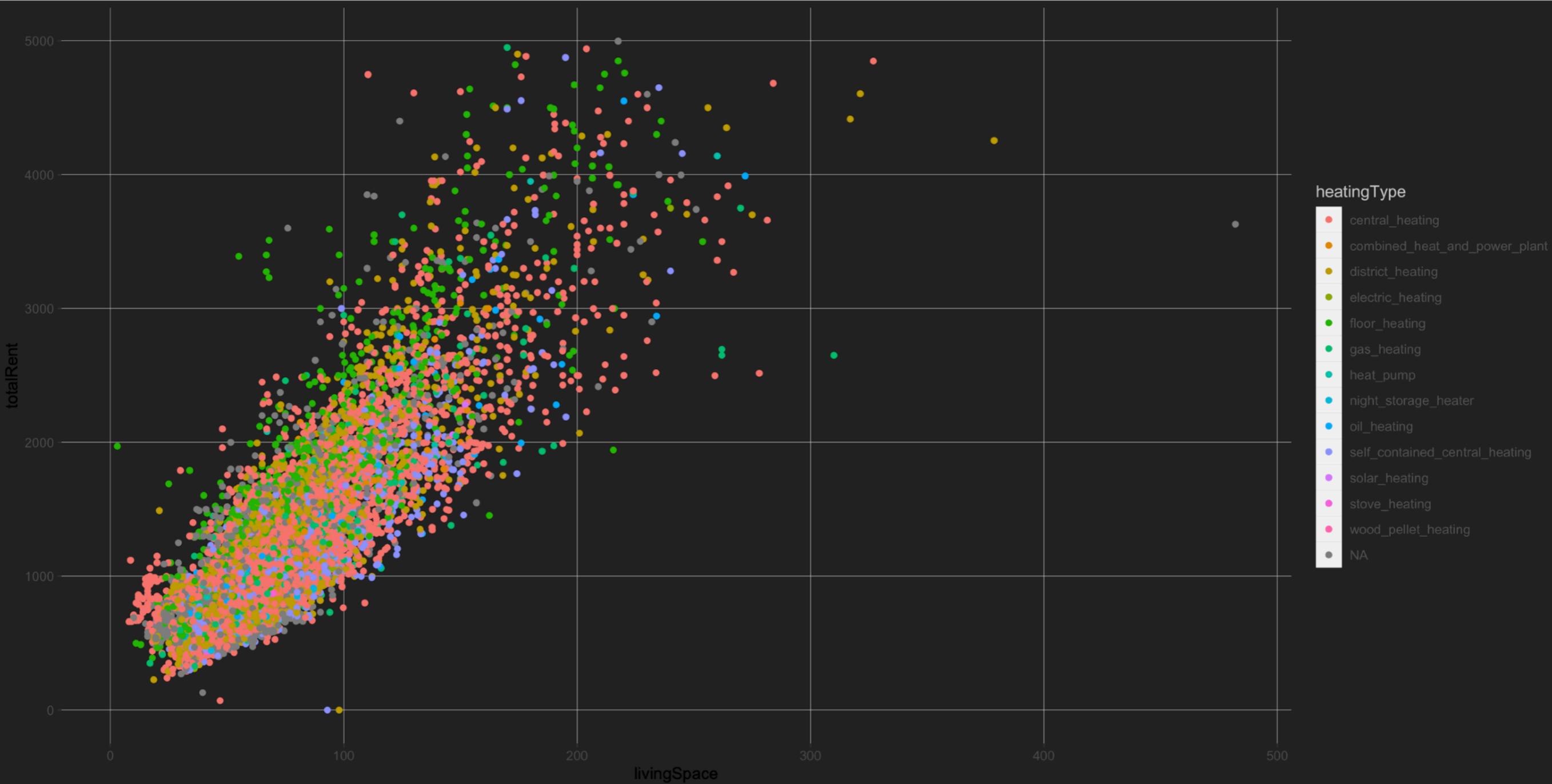
Take out Berlin from all the areas in Germany, filter the columns

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			

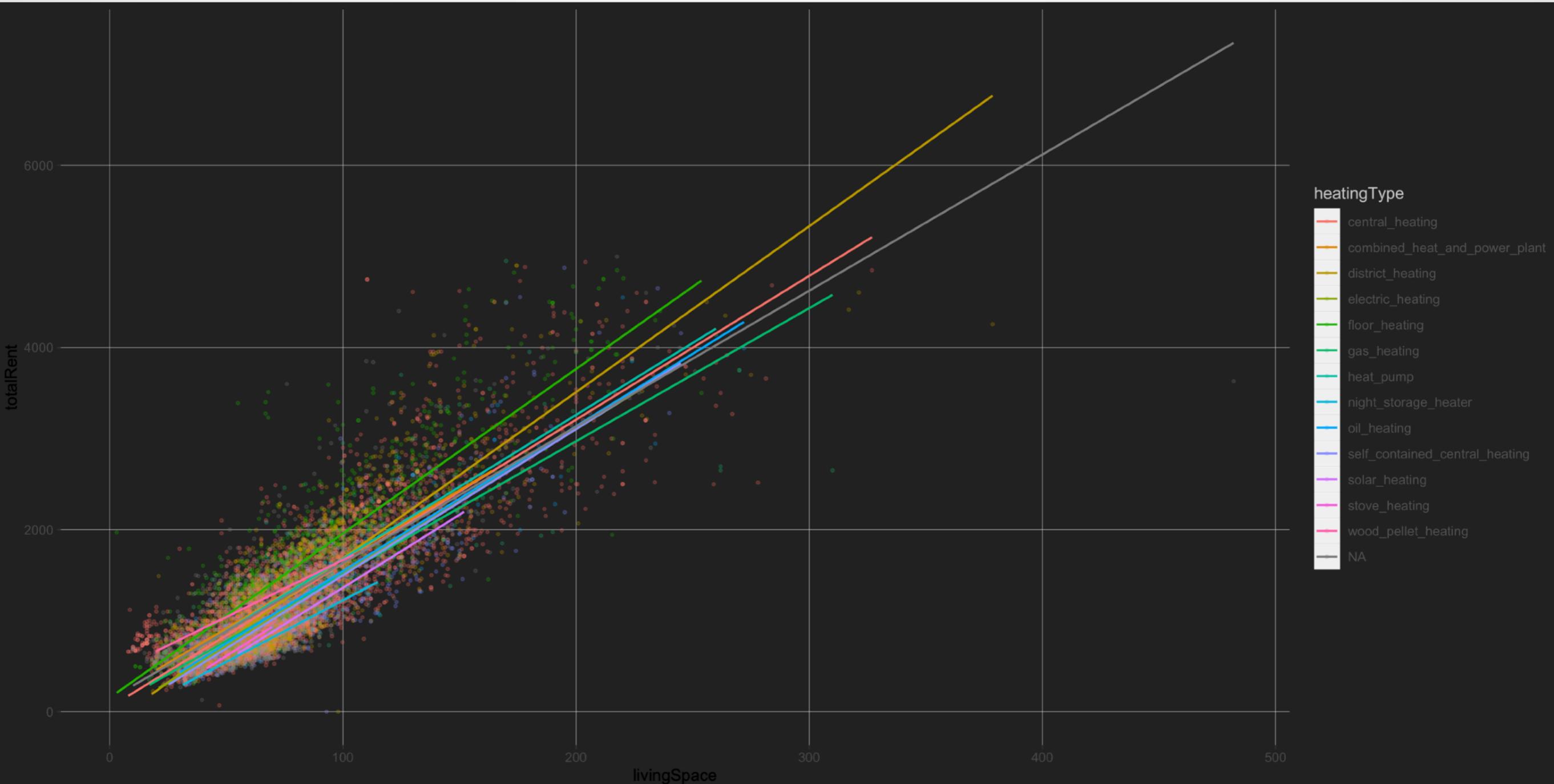
...and more cleaning



## Example: do we care about heating?



## Example: do we care about heating?



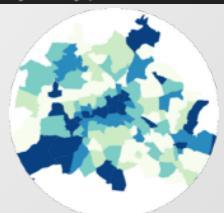
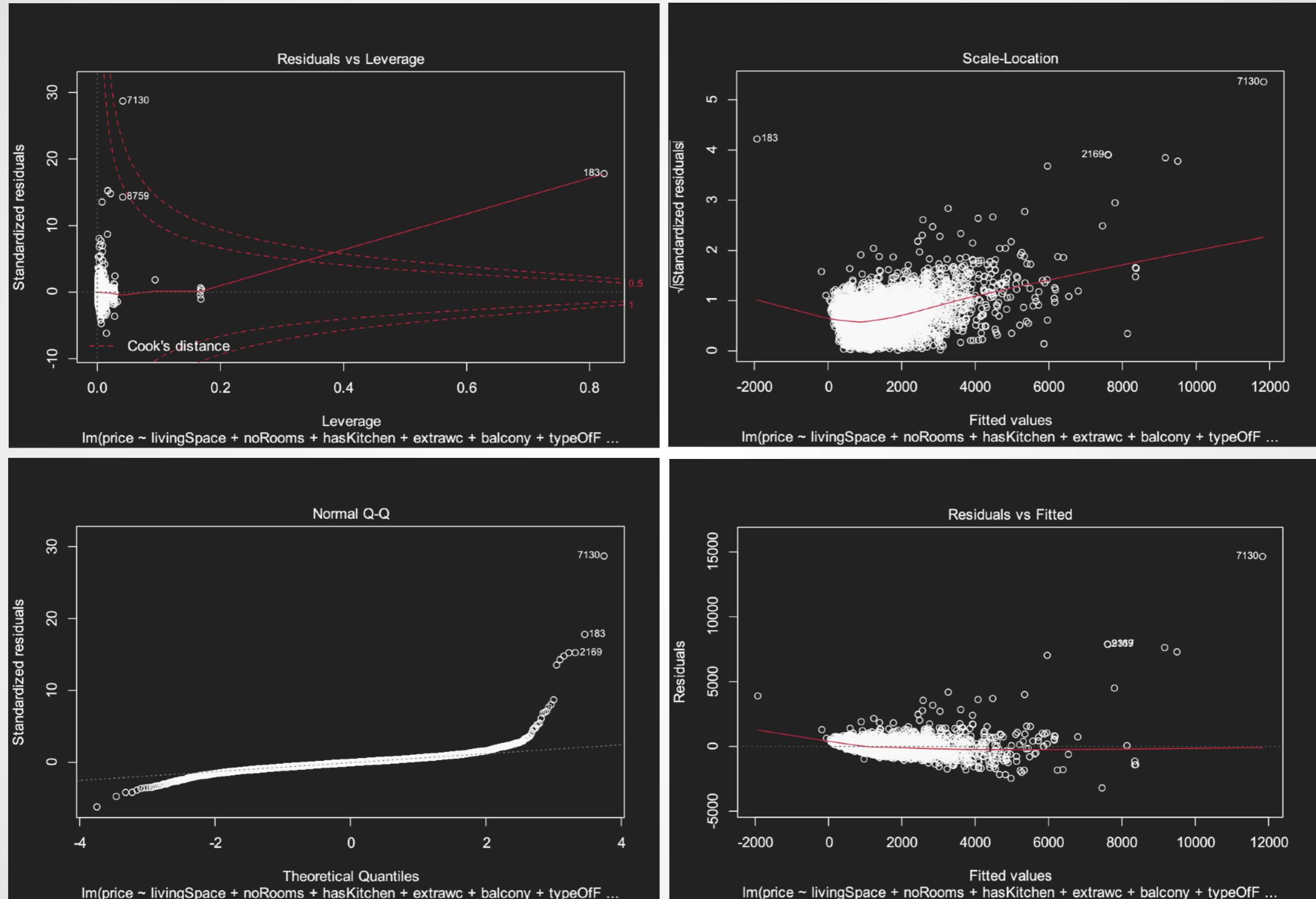
... seems like not so much.



# The general model:

Coefficients:					
	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	-676.6920	363.8389	-1.860		0.062957 .
livingSpace	20.6413	0.2020	102.171 < 0.00000000000002 ***		
noRooms	-20.9572	4.8091	-4.358 0.00001336998838628 ***		
hasKitchenTRUE	198.2198	15.5389	12.756 < 0.00000000000002 ***		
extrawcTRUE	-88.8065	23.3628	-3.801 0.000146 ***		
balconyTRUE	-116.4264	18.2529	-6.379 0.00000000019325457 ***		
typeOfFlatground_floor	7.4578	29.0540	0.257 0.797428		
typeOfFlathalf_basement	-141.0655	213.4316	-0.661 0.508678		
typeOfFlatloft	218.3554	82.4231	2.649 0.008091 **		
typeOfFlatmaisonette	-168.2702	41.7036	-4.035 0.00005535666547027 ***		
typeOfFlatother	-76.1280	55.6877	-1.367 0.171664		
typeOfFlatpenthouse	244.1591	49.1194	4.971 0.0000068703244169 ***		
typeOfFlatraised_ground_floor	-33.0369	52.5983	-0.628 0.529965		
typeOfFlatroof_storey	-61.1609	24.8792	-2.458 0.013989 *		
typeOfFlatterraced_flat	8.9068	53.9939	0.165 0.868982		
gardenTRUE	23.6984	20.1843	1.174 0.240406		
cellarTRUE	13.1139	15.0920	0.869 0.384924		
yearConstructed	0.1073	0.1885	0.569 0.569161		
newlyConstTRUE	168.8351	21.0551	8.019 0.000000000000129 ***		
floor	20.4250	4.8709	4.193 0.00002792798535111 ***		
numberOfFloors	4.9745	4.2637	1.167 0.243376		
liftTRUE	199.2381	18.8328	10.579 < 0.00000000000002 ***		
---					
Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '
Residual standard error:	521.5	on 5558 degrees of freedom			
(4826 observations deleted due to missingness)					
Multiple R-squared:	0.7649,	Adjusted R-squared:	0.7641		
F-statistic:	861.3	on 21 and 5558 DF,	p-value: < 0.000000000000022		





## The filtered model:

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-215.7446	12.8122	-16.839 < 0.00000000000002	***
livingSpace	19.9661	0.1856	107.587 < 0.00000000000002	***
noRooms	-92.2805	7.0667	-13.058 < 0.00000000000002	***
hasKitchenTRUE	232.2461	9.0328	25.711 < 0.00000000000002	***
balconyTRUE	-58.4699	10.3806	-5.633	0.000000183 ***
liftTRUE	237.4880	9.0490	26.245 < 0.00000000000002	***

---

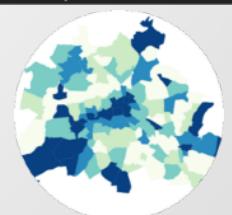
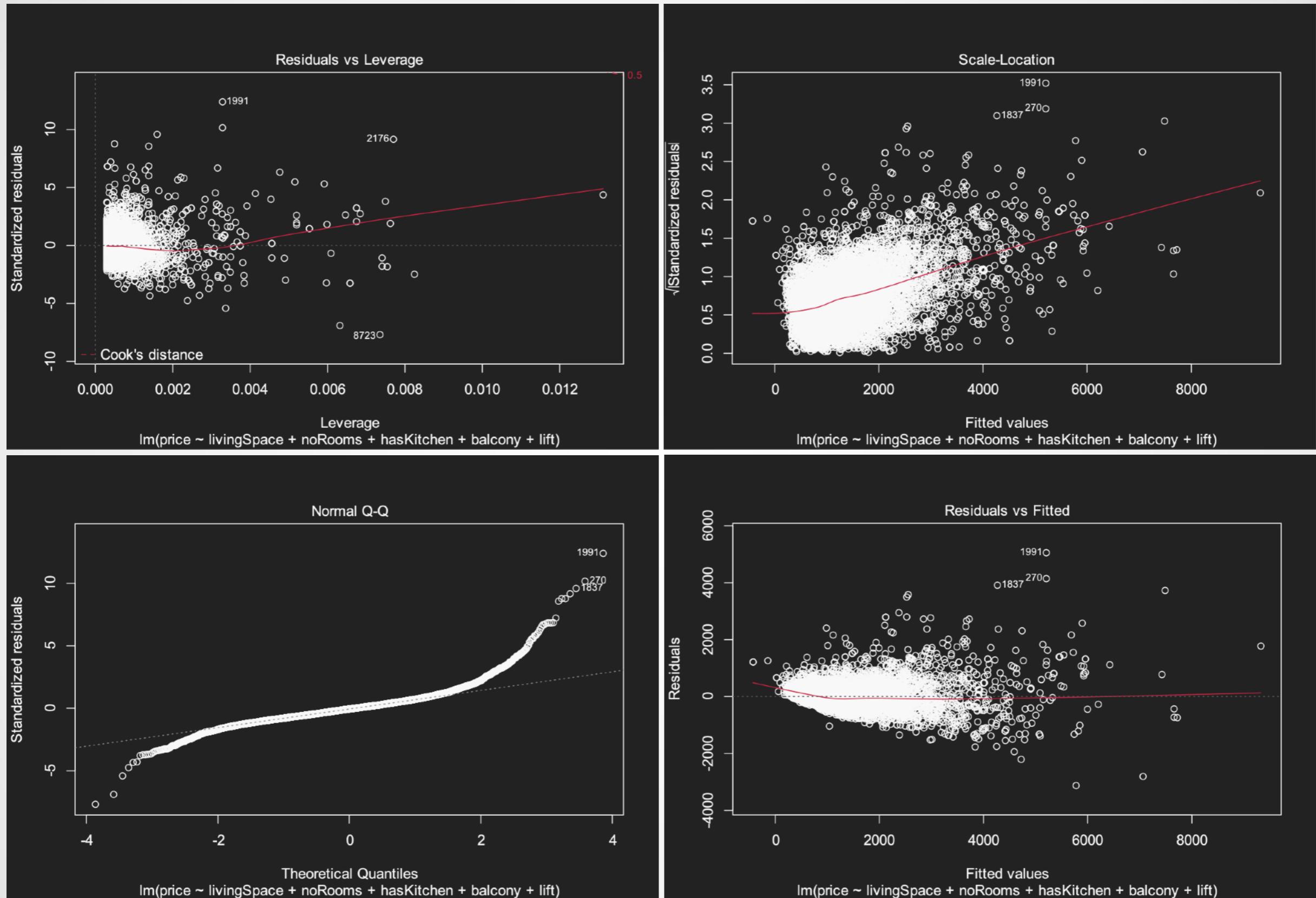
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 408.5 on 8959 degrees of freedom

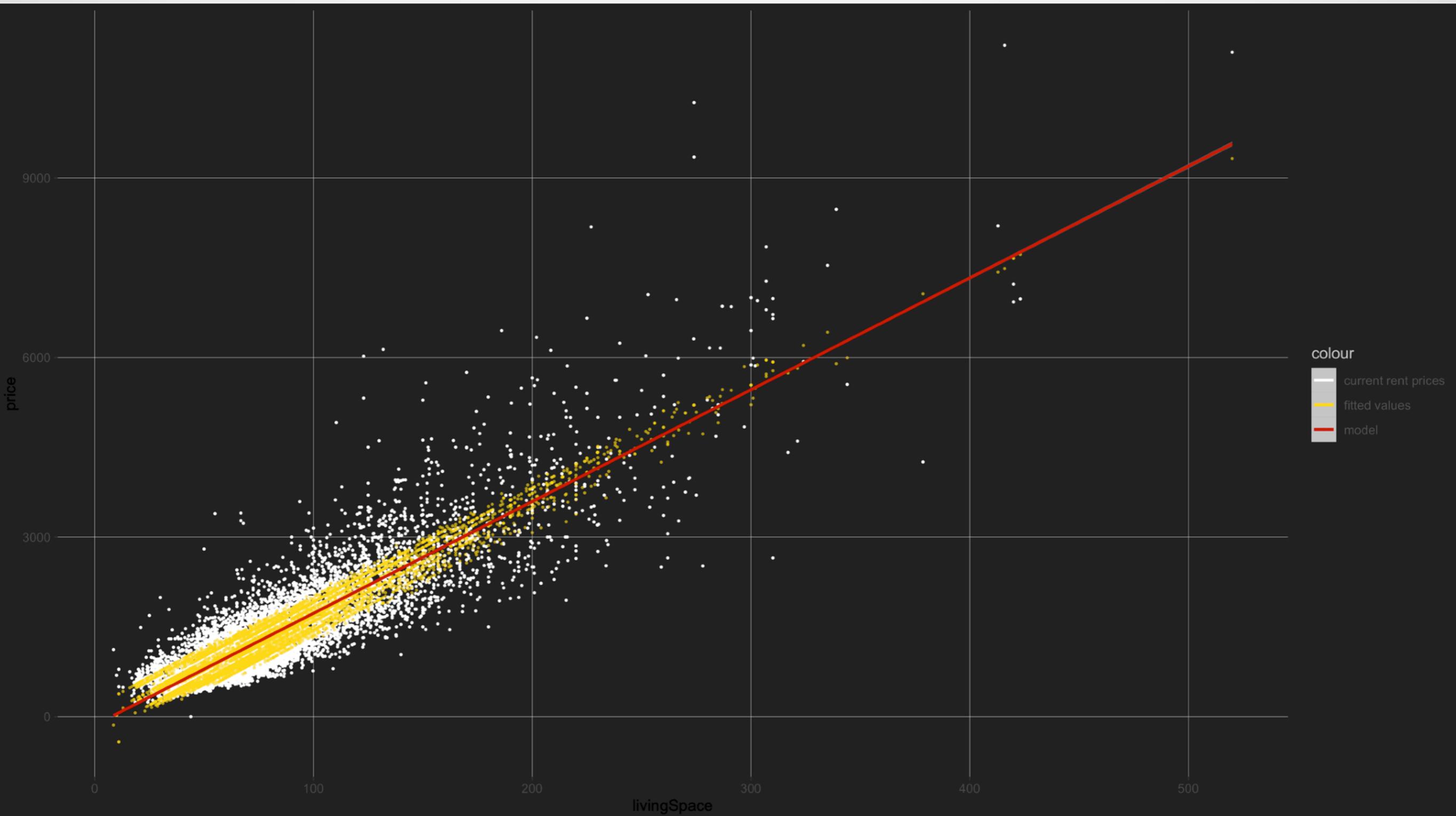
Multiple R-squared: 0.7981, Adjusted R-squared: 0.798

F-statistic: 7082 on 5 and 8959 DF, p-value: < 0.000000000000022





## The filtered model:



# Price model done.

What now?

Geocoding!

## 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)



# The project:

We have the map

We have the prices (and the addresses)

We **need** the coordinates of the addresses

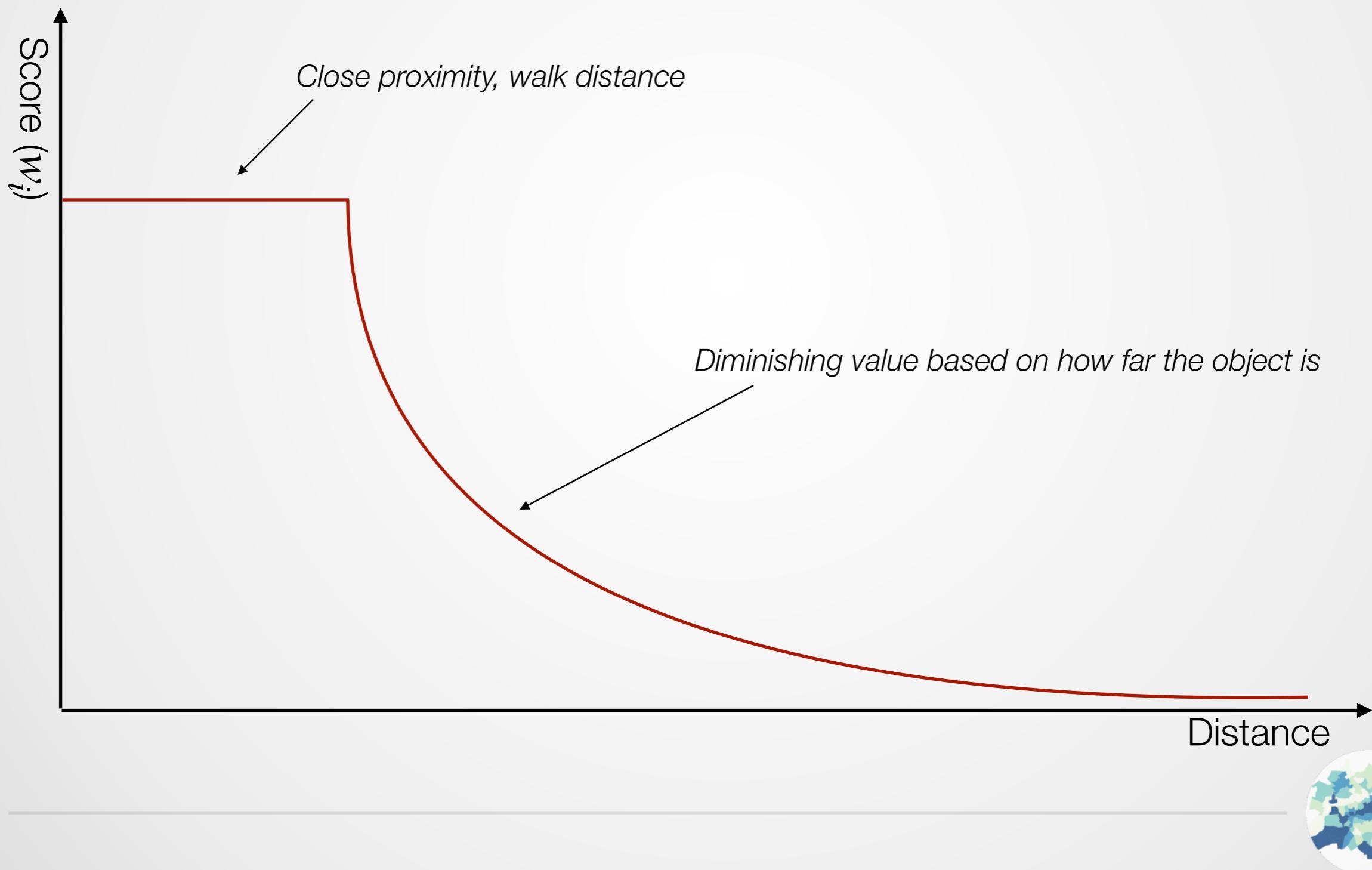
```
> test_address <- "Berlin 10557 Heidestrasse 19"
> geocode_OSM("Berlin 10557 Heidestrasse 19", as.sf = TRUE)
Simple feature collection with 1 feature and 7 fields
Active geometry column: point
Geometry type: POINT
Dimension: XY
Bounding box: xmin: 13.36456 ymin: 52.53469 xmax: 13.36456 ymax: 52.53469
Geodetic CRS: WGS 84
      query      lat      lon lat_min lat_max lon_min lon_max           bbox          point
1 Berlin 10557 Heidestrasse 19 52.53469 13.36456 52.53464 52.53474 13.36451 13.36461 POLYGON ((13.36451 52.53464... POINT (13.36456 52.53469)
```

```
for(i in 1:8856) {
  placeholderaddress <- geocode_OSM(addresses_for_geo[i], as.sf = TRUE)
  coord_list <- bind_rows(coord_list, placeholderaddress)
  print(Sys.time())
  Sys.sleep( time: 1.5)
}
```



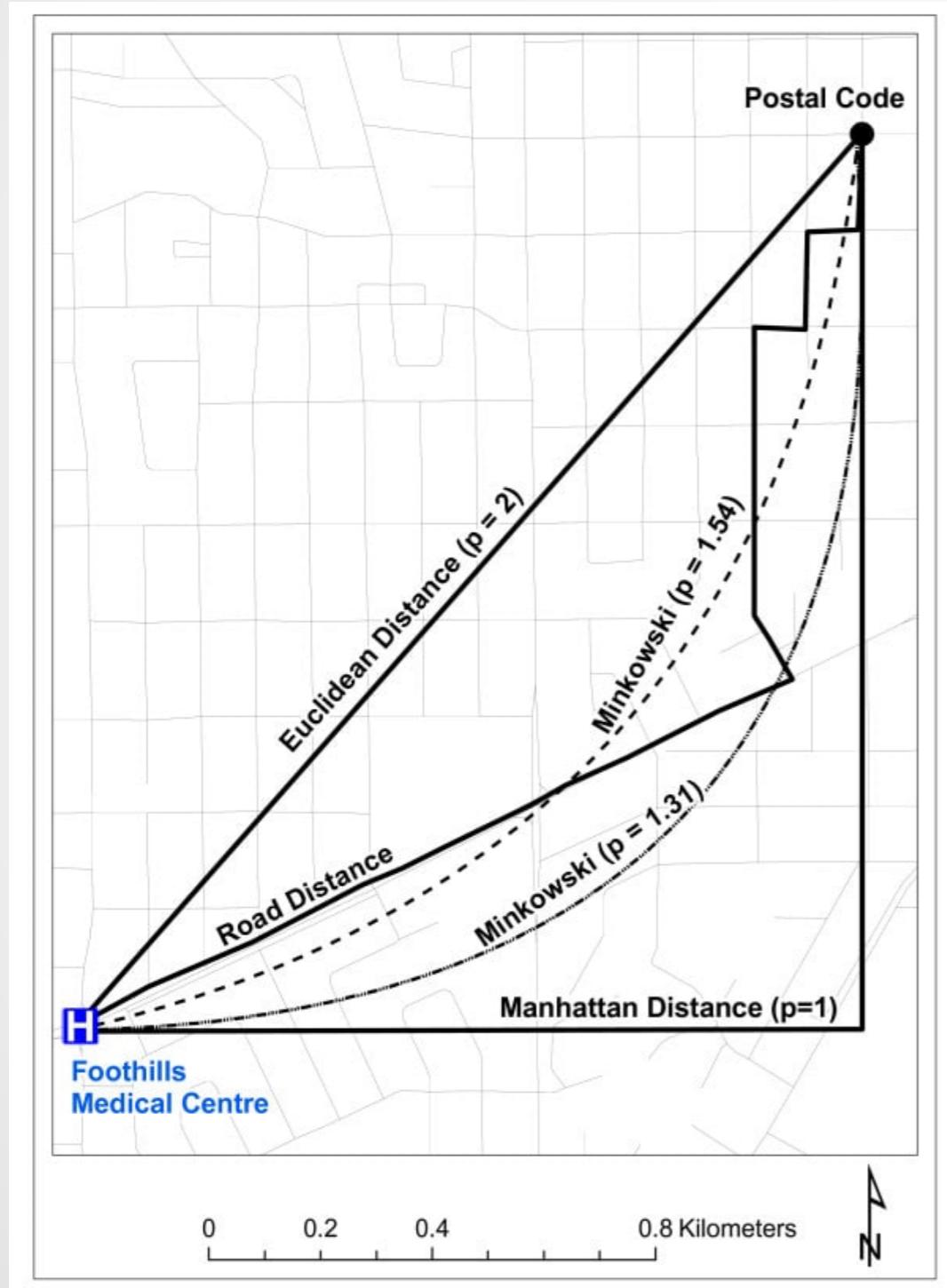
## What's next?

Create a model to account for distance between an apartment and the surrounding infrastructure.



## What's next?

How to measure the distance in the presence of city-terrain?



The Minkowski distance of order  $p$  (where  $p$  is an integer) between two points

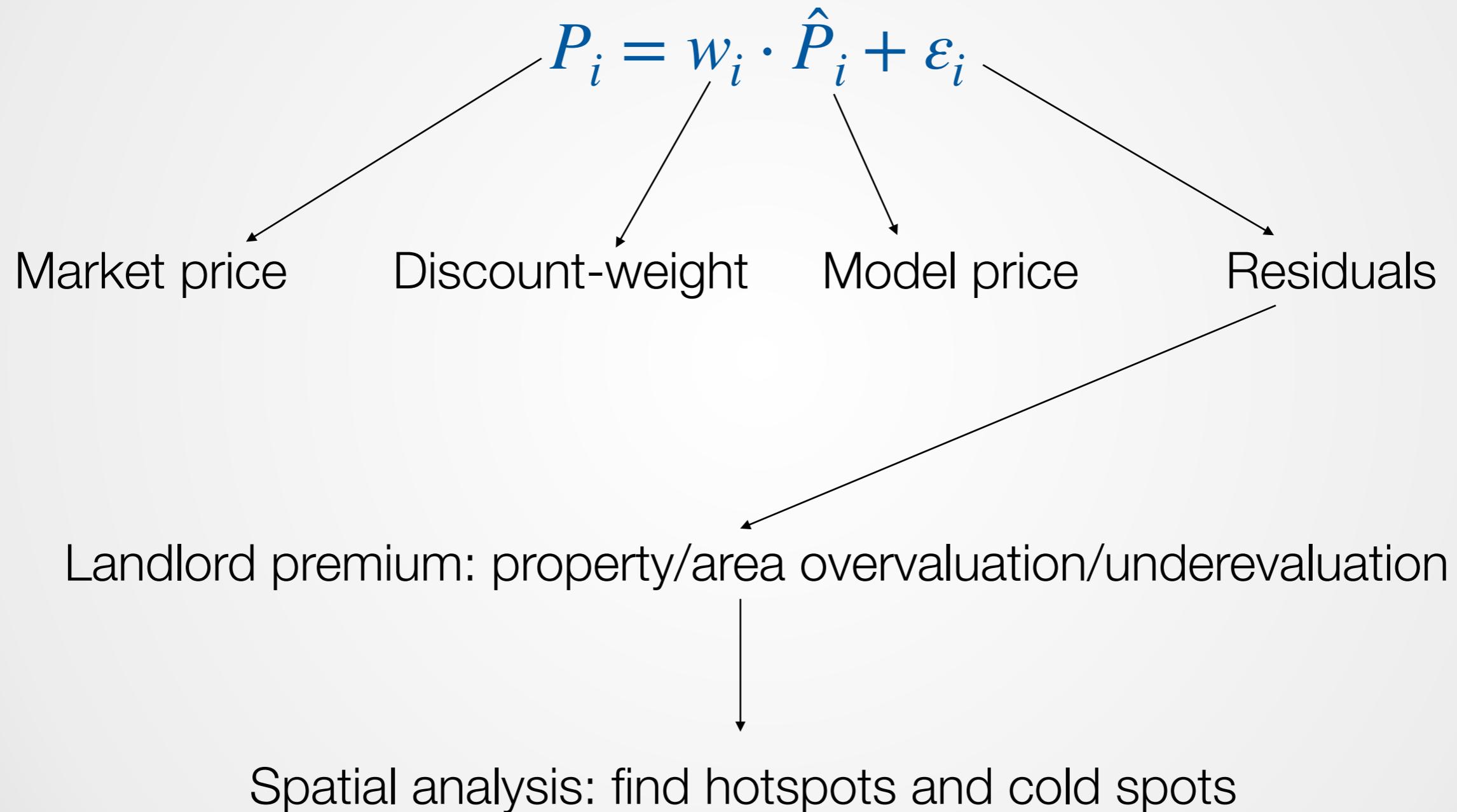
$$X = (x_1, x_2, \dots, x_n) \text{ and } Y = (y_1, y_2, \dots, y_n) \in \mathbb{R}^n$$

is defined as:

$$D(X, Y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}.$$



## What's next?



# Final graph idea?

