Spatially weighted averages in R with sf

Markus Konrad

6/8/2021

Introduction

Spatial joins allow to augment one spatial dataset with information from another spatial dataset by linking overlapping features. In this post I will provide an example to show how to augment a dataset containing school locations with socioeconomic data of their surrounding statistical region using R and the package *sf*. This approach has the drawback that the surrounding statistical region doesn't reflect the actual catchment area of the school. I will present an alternative approach where the overlaps of the schools' catchment areas with the statistical regions allow to calculate the weighted average of the socioeconomic statistics. If we have no data about the actual catchment areas of the schools, we may resort to approximating these as circular regions or as Voronoi regions around schools.

Data

For this example, I'd like to compare the percentage of children whose parents obtain social welfare in the neighborhood regions around public and private primary schools in Berlin. This blog post concentrates on how to join the point samples (the schools) with the surrounding statistical regions, so I will present only a few summary statistics in the end since proper spatial modeling is beyond the scope of this blog post.

We will work with several datasets: The first spatial dataset contains the shape of the statistical regions in Berlin, the second dataset contains the socioeconomic data for these regions, the third and fourth datasets contain the locations and other attributes of public and private primary schools in Berlin, respectively.

All data and the code are available in the GitHub repository. We will use the *sf* package for working with spatial data in R, *dplyr* for data management and *ggplot2* for a few more advanced visualizations, i.e. when base `plot()` is not sufficient.

```r
library(sf)
library(dplyr)
library(ggplot2)
```

Socioeconomic data for statistical regions

We will at first load a dataset with the most granular official statistical regions for Berlin, called *Planungsräume* (planning areas). We select the area ID and name as spatial attributes. The result is a spatial dataframe (a *simple feature (sf)* collection).

```r
bln_plan <- read_sf('data/berlin_plr.shp') %>%
  mutate(areaid = as.integer(SCHLUESSEL)) %>%   # transform character SCHLUESSEL to numeric area ID
  select(areaid, name = PLR_NAME)
head(bln_plan)
```

```
## Simple feature collection with 6 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 386668.2 ymin: 5817761 xmax: 390764.3 ymax: 5820432
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 6 x 3
##    areaid name                                                          geometry
##     <int> <chr>                                               <MULTIPOLYGON [m]>
## 1 1011101 Stülerstr.      (((387256.6 5818552, 387323.1 5818572, 387418.9 58186~
## 2 1011102 Großer Tiergar~ (((386767.5 5819393, 386768.3 5819389, 386769.6 58193~
## 3 1011103 Lützowstr.      (((387952.6 5818275, 387986.7 5818313, 387994.6 58183~
## 4 1011104 Körnerstr.      (((388847.1 5817875, 388855.5 5817899, 388865.1 58179~
## 5 1011105 Nördlicher Lan~ (((388129.5 5819015, 388157.1 5819017, 388170.8 58190~
## 6 1011201 Wilhelmstr.     (((389845.7 5819286, 389840.9 5819311, 389846.1 58193~
```

When printing this dataframe, the header reveals another important information: The coordinate reference system (CRS) of this dataset is ETRS89 / UTM zone 33N. We will later need to make sure that the coordinates of the school locations and the coordinates of the planning areas use the same coordinate system.

This data can be joined with socioeconomic information provided from official sources. Luckily, Helbig/Salomo 2021 compiled these information for some cities in Germany (available for download) among which is data for Berlin from 2020. I've created an excerpt with percentages of residents receiving social welfare (`welfare`) and percentage of children under 15 years whose parents receive social welfare (`welfare_chld`):

```
bln_welfare <- read.csv('data/berlin_welfare.csv', stringsAsFactors = FALSE)
head(bln_welfare)
```

```
##    areaid                 areaname welfare welfare_chld
## 1 1011101             Stülerstraße   10.09        15.44
## 2 1011102        Großer Tiergarten    4.76         0.00
## 3 1011103             Lützowstraße   22.21        36.80
## 4 1011104             Körnerstraße   24.81        42.14
## 5 1011105 Nördlicher Landwehrkanal    2.82         3.53
## 6 1011201            Wilhelmstraße   12.13        19.03
```

We can use the area ID for augmenting the planning areas with the welfare statistics. We're joining a spatial with an ordinary dataframe, so we can use dplyr's `inner_join`. Before that we can check that for each planning area we have welfare statistics information and vice versa: [1]

```
setequal(bln_plan$areaid, bln_welfare$areaid)
```

```
## [1] TRUE
```

```
bln <- inner_join(bln_plan, bln_welfare, by = 'areaid') %>%
  select(-name)
head(bln)
```

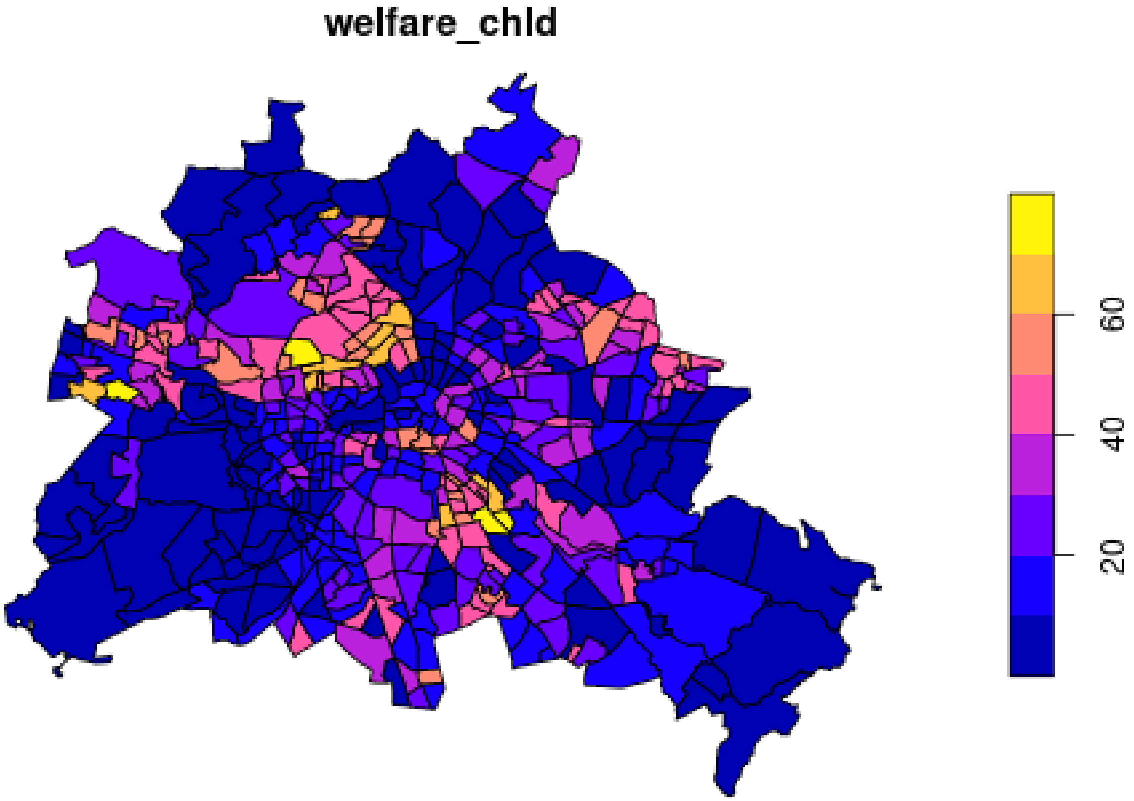
```
## Simple feature collection with 6 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 386668.2 ymin: 5817761 xmax: 390764.3 ymax: 5820432
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 6 x 5
##    areaid                              geometry areaname    welfare welfare_chld
##     <int>                    <MULTIPOLYGON [m]> <chr>         <dbl>        <dbl>
## 1 1011101 (((387256.6 5818552, 387323.1 581857~ Stülerstra~    10.1         15.4
```

[1] Note that when joining spatial and ordinary dataframes, the order of arguments in the join function matters. If you have a spatial dataframe on the "left side" (`x` argument), the result will be a spatial dataframe. If you have an ordinary dataframe on the left side, the result will be an ordinary dataframe, i.e. the merged dataset loses its "spatial nature" and spatial operations won't work with it any more (unless you convert it back to a spatial dataframe again with `st_as_sf`).

```
## 2 1011102 (((386767.5 5819393, 386768.3 581938~ Großer Tie~    4.76         0
## 3 1011103 (((387952.6 5818275, 387986.7 581831~ Lützowstra~   22.2         36.8
## 4 1011104 (((388847.1 5817875, 388855.5 581789~ Körnerstra~   24.8         42.1
## 5 1011105 (((388129.5 5819015, 388157.1 581901~ Nördlicher~    2.82         3.53
## 6 1011201 (((389845.7 5819286, 389840.9 581931~ Wilhelmstr~   12.1         19.0
```

A quick plot confirms that it is similar to the one from the dashboard of the Helbig/Salomo study.[2]

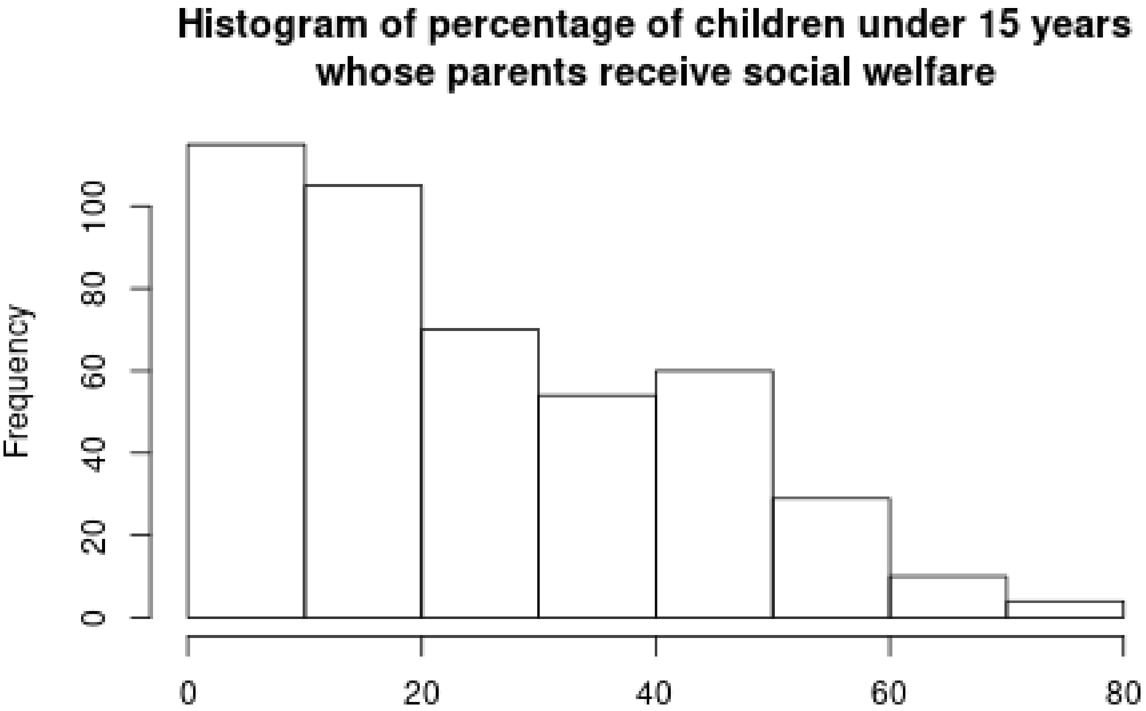
```
plot(bln['welfare_chld'])
```



The median percentage of children whose parents receive social welfare is ~20% with an interquartile range of about 29%. The following shows the distribution of this welfare rate:

```
hist(bln$welfare_chld,
     main = 'Histogram of percentage of children under 15 years\nwhose parents receive social welfare',
     xlab = '')
```

[2] I prefer using the base `plot` function for quick exploration of spatial data and usually only turn to ggplot2 for more advanced or "publication ready" plots. The help page for `plot.sf` provides some information about the arguments of this plotting function used for sf objects.



Public and private primary schools

The Berlin geodata catalog “FIS Broker” provides the locations of public schools in Berlin.[3] I obtained the data and converted it to GeoJSON, which we can now load. We’ll only retain primary schools and add an variable denoting that these are public schools. We also see that the CRS of the school locations matches the CRS of the Berlin statistical regions data.

```
pubschools <- read_sf('data/berlin_pubschools.geojson') %>%
  filter(SCHULART == 'Grundschule') %>%
  select(name = NAME) %>%
  mutate(ownership = 'pub', .before = 1)
head(pubschools)
```

```
## Simple feature collection with 6 features and 2 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 390139.2 ymin: 5819341 xmax: 393086.4 ymax: 5821994
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 6 x 3
##   ownership name                                           geometry
##   <chr>     <chr>                                       <POINT [m]>
## 1 pub       Grundschule am Arkonaplatz       (391497.3 5821994)
## 2 pub       Papageno-Grundschule             (390876.3 5821514)
## 3 pub       Kastanienbaum-Grundschule        (391579.6 5820819)
## 4 pub       Grundschule Neues Tor            (390139.2 5820930)
## 5 pub       GutsMuths-Grundschule            (393086.4 5819617)
## 6 pub       Grundschule am Brandenburger Tor   (390255 5819341)
```

Now to the private schools’ locations. Marcel Helbig, Rita Nikolai and me collected data on school locations in East Germany from 1992 to 2015 in order to analyze the development of the network of schools in East

[3] The catalog is a bit clumsy to use, but actually works quite well: You search for the data, get an URL to the WFS endpoint from the data’s metainformation panel and use that URL to obtain the data e.g. via a WFS layer in QGIS.

Germany and which role private schools play in it. Besides creating an interactive map, we also published the data and are planning an update with newer data (until 2020) from which will we now use an excerpt. This dataset provides school locations from 2019 as longitude/latitude WGS84 coordinates which we can load and convert into a spatial dataset using `st_as_sf`. We also transform these locations to the ETRS89 CRS used in all prior spatial datasets.

```
privschools <- read.csv('data/grundschulen_berlin_2019.csv', stringsAsFactors = FALSE) %>%
  filter(traeger == 'priv') %>%
  select(ownership = traeger, name, lng, lat) %>%
  st_as_sf(coords = c('lng', 'lat'), crs = 4326) %>%  # EPSG 4326 is WGS84 lat/long coord.
  st_transform(crs = st_crs(pubschools))  # transform to same CRS as publ. schools
head(privschools)
```

```
## Simple feature collection with 6 features and 2 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 374724.8 ymin: 5813179 xmax: 395177.8 ymax: 5822721
## Projected CRS: ETRS89 / UTM zone 33N
##   ownership                                          name
## 1      priv              Freie Waldorfschule Berlin Mitte
## 2      priv                 Freie Waldorfschule Kreuzberg
## 3      priv        Freie Waldorfschule am Prenzlauer Berg
## 4      priv                           Annie-Heuser-Schule
## 5      priv Freie Waldorfschule Havelhöhe - Eugen Kolisko
## 6      priv                  Rudolf-Steiner-Schule Berlin
##                  geometry
## 1 POINT (391783.5 5820737)
## 2   POINT (391544 5818222)
## 3 POINT (395177.8 5822721)
## 4 POINT (384902.8 5816817)
## 5 POINT (374724.8 5813820)
## 6 POINT (382728.5 5813179)
```

The variable `ownership` encodes whether a given facility is a public ("pub") or private ("priv") primary school. We can now append the public and private primary schools datasets to form a single `schools` dataset. The public school data comes from 2020 and the private school data from 2019, but this shouldn't be an issue because the number of public and private schools has been quite stable in recent years.

```
schools <- bind_rows(pubschools, privschools) %>%
  mutate(schoolid = 1:nrow(.), .before = 1)
head(schools)
```

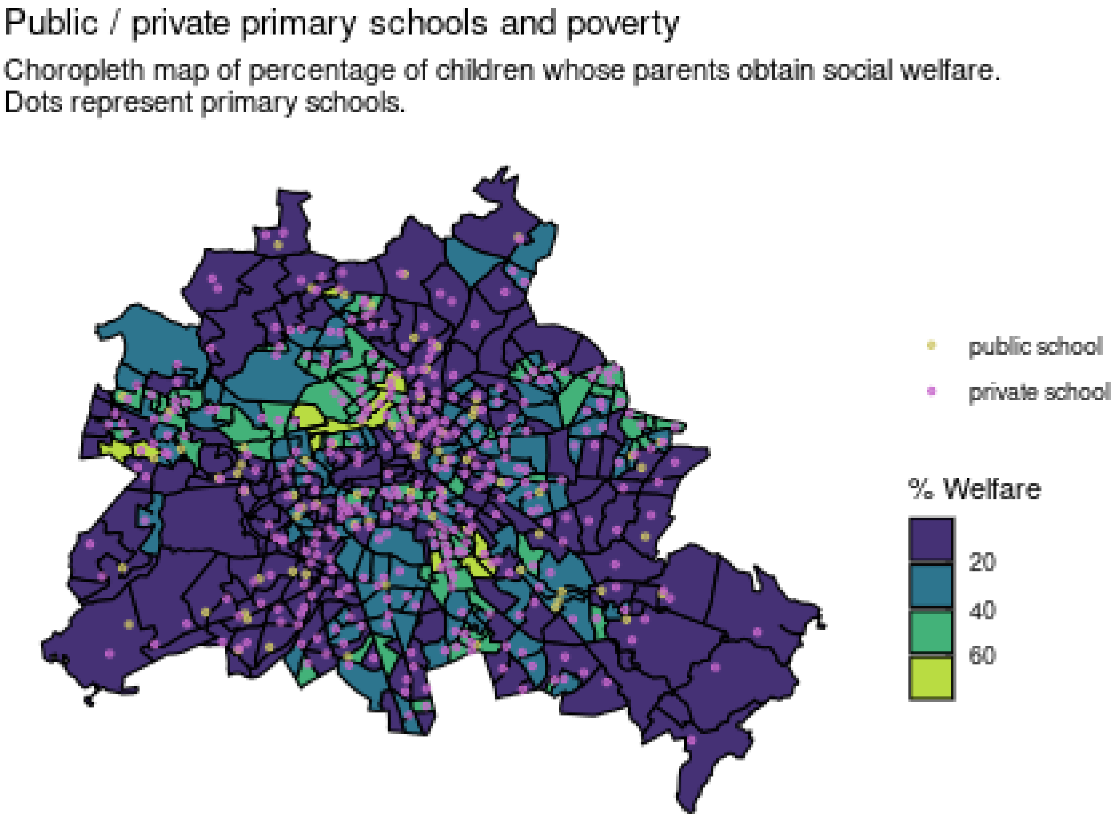
```
## Simple feature collection with 6 features and 3 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 390139.2 ymin: 5819341 xmax: 393086.4 ymax: 5821994
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 6 x 4
##   schoolid ownership name                                          geometry
##      <int> <chr>     <chr>                                      <POINT [m]>
## 1        1 pub       Grundschule am Arkonaplatz       (391497.3 5821994)
## 2        2 pub       Papageno-Grundschule             (390876.3 5821514)
## 3        3 pub       Kastanienbaum-Grundschule        (391579.6 5820819)
## 4        4 pub       Grundschule Neues Tor            (390139.2 5820930)
## 5        5 pub       GutsMuths-Grundschule            (393086.4 5819617)
## 6        6 pub       Grundschule am Brandenburger Tor   (390255 5819341)
```

In our dataset we now have 361 public and 71 private primary schools in Berlin.

Public / private primary schools and poverty by statistical region

Both datasets use the same coordinate system now, so we can plot the school locations on top of the planning areas. I will use ggplot2 this time to make a choropleth map of the `welfare_chld` variable and overlay that with the public and private primary school locations.

```
ggplot() +
  geom_sf(aes(fill = welfare_chld), color = 'black', data = bln) +
  geom_sf(aes(color = ownership), size = 1, alpha = 0.75, data = schools) +
  scale_fill_binned(type = 'viridis', guide = guide_bins(title = '% Welfare')) +
  scale_color_manual(values = c('pub' = '#c767cb',  'priv' = '#cdc566'),
                     labels = c('public school', 'private school'),
                     guide = guide_legend(title = '')) +
  coord_sf(datum = NA) +  # disable graticule
  labs(title = "Public / private primary schools and poverty",
       subtitle = "Choropleth map of percentage of children whose parents obtain social welfare.\nDots
  theme_minimal()
```


Public / private primary schools and poverty
Choropleth map of percentage of children whose parents obtain social welfare.
Dots represent primary schools.
public school
private school
% Welfare
20
40
60

From the figure alone, it's probably hard to assess whether there's a pattern in the distribution of private and public schools regarding areas with higher welfare rate in the city. In order to compare the social welfare statistics of regions around private schools with those around public schools, we can join the schools' data with the socioeconomic information of the planning areas they're located in. This can be done with a spatial join using `st_join`. By default, this function joins the spatial features of the first argument with features of the second argument **when they intersect** – in our case this means a school is linked with the planning area it's located in. Note that the order of arguments matters here and that the spatial geometry of the first argument is retained in the resulting dataset.

```
schools_plan <- st_join(schools, bln)
head(schools_plan)
```

```
## Simple feature collection with 6 features and 7 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 390139.2 ymin: 5819341 xmax: 393086.4 ymax: 5821994
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 6 x 8
##   schoolid ownership name                      geometry areaid areaname  welfare
##      <int> <chr>     <chr>                  <POINT [m]>  <int> <chr>       <dbl>
## 1        1 pub       Grundsch~       (391497.3 5821994) 1.01e6 Arkonapl~    4.46
## 2        2 pub       Papageno~       (390876.3 5821514) 1.01e6 Invalide~    6.29
## 3        3 pub       Kastanie~       (391579.6 5820819) 1.01e6 Oranienb~    8.16
## 4        4 pub       Grundsch~       (390139.2 5820930) 1.01e6 Charitév~    3.68
## 5        5 pub       GutsMuth~       (393086.4 5819617) 1.01e6 Karl-Mar~   23.5
## 6        6 pub       Grundsch~         (390255 5819341) 1.01e6 Wilhelms~   12.1
## # ... with 1 more variable: welfare_chld <dbl>
```

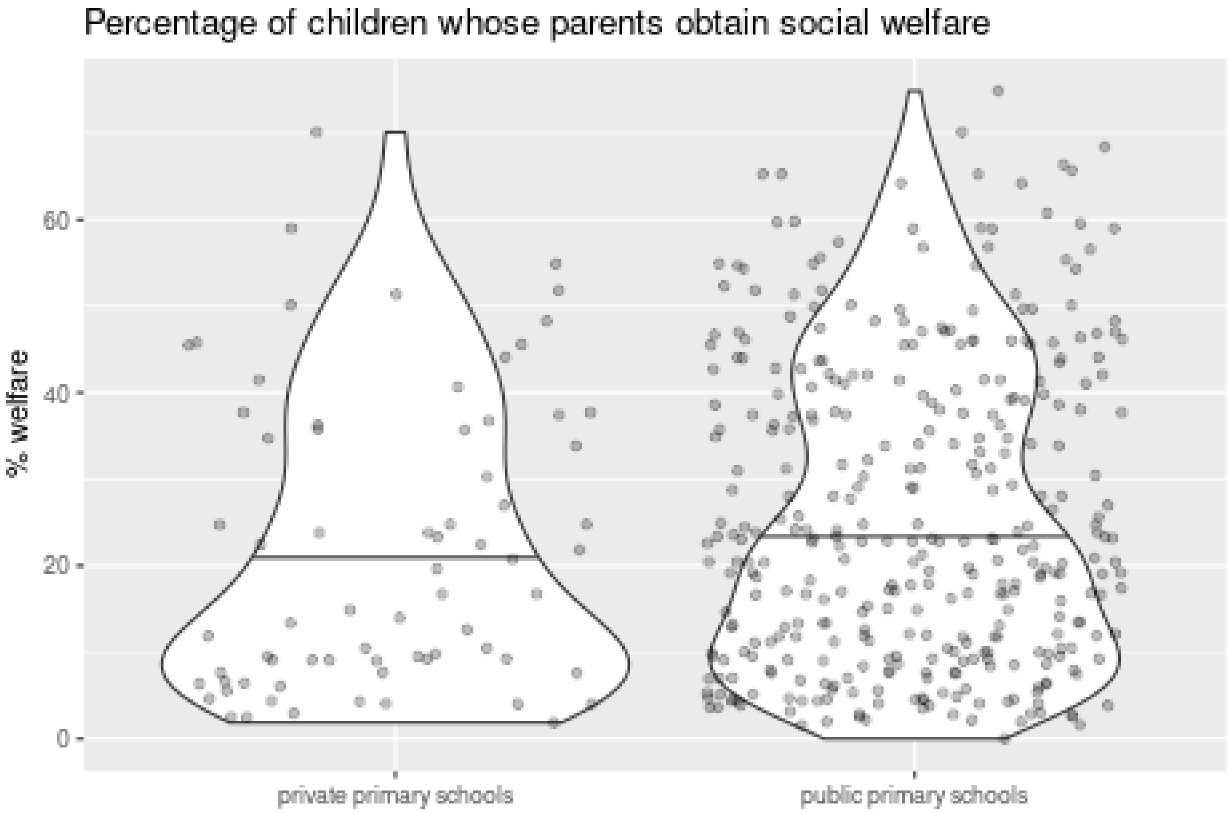
We can see that the schools' data was linked with the data from the planning areas. We should also check whether there's a school that was not located in any planning area (this may for example happen when a school is very close to the Berlin-Brandenburg border):

```
sum(is.na(schools_plan$areaid))
```

```
## [1] 0
```

All schools were linked with their planning area, so we can now compare the percentage of children whose parents obtain social welfare between public and private primary schools:

```
ggplot(schools_plan) +
  geom_violin(aes(x = ownership, y = welfare_chld), draw_quantiles = c(0.5)) +
  geom_jitter(aes(x = ownership, y = welfare_chld), alpha = 0.25) +
  scale_x_discrete(labels = c('pub' = 'public primary schools', 'priv' = 'private primary schools')) +
  labs(title = 'Percentage of children whose parents obtain social welfare', x = '', y = '% welfare')
```



Our descriptive results indicate that the median percentage of children whose parents obtain social welfare is around six percent higher in the statistical regions around public schools than around private schools: [4]

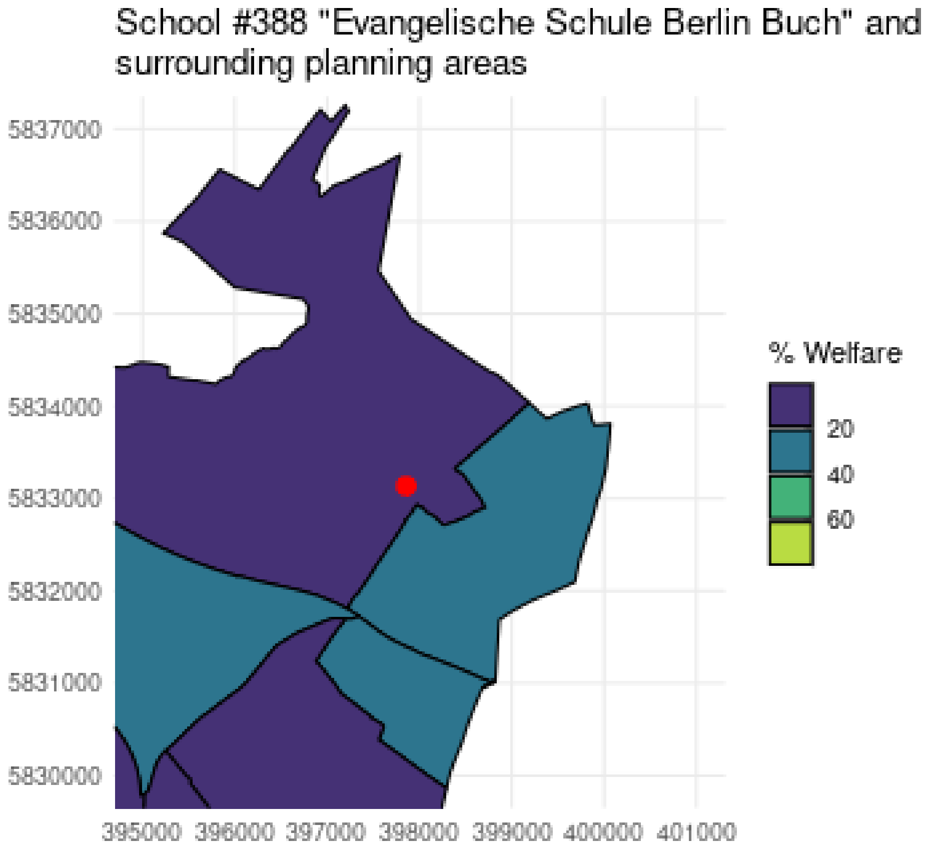
```
st_drop_geometry(schools_plan) %>%
  group_by(ownership) %>%
  summarise(median_welfare_chld = median(welfare_chld))
```

```
## # A tibble: 2 x 2
##   ownership median_welfare_chld
##   <chr>                   <dbl>
## 1 priv                     16.8
## 2 pub                      22.8
```

This is an interesting descriptive result and we may continue with our spatial analysis from here. However, our current approach doesn't consider the catchment area of a school correctly: Children from nearby planning areas will most likely visit a school, but at the moment we only consider the one planning area in which a school is located. As an example, let's zoom to school #388 "Evangelische Schule Berlin Buch" in the north of Berlin. As you can see, only considering the planning area in which this school is located omits the higher welfare rates in nearby areas:

```
ggplot(bln) +
  geom_sf(aes(fill = welfare_chld), color = 'black') +
  geom_sf(data = filter(schools, schoolid == 388), size  = 3, color = 'red') +
  scale_fill_binned(type = 'viridis', guide = guide_bins(title = '% Welfare')) +
  coord_sf(datum = st_crs(bln), xlim = c(395e3, 401e3), ylim = c(583e4, 5837e3)) +
  labs(title = 'School #388 "Evangelische Schule Berlin Buch" and\nsurrounding planning areas') +
  theme_minimal()
```

[4] I'm using `st_drop_geometry` here, because otherwise a spatial aggregation would be performed which takes much longer to compute and is not necessary here.



Spatial weighting with official school catchment areas

In Berlin, parents can send their children to a primary school that is within the official school catchment area of their home address *("Grundschuleinzugsbereiche")*.[5] Luckily, there's spatial data for these catchment areas again available in the Berlin geodata catalog. I again converted the obtained data from the Berlin geodata catalog to GeoJSON, which we can load now. I also generate a catchment area ID `catchid` which however has nothing to do with the school ID from `schools` dataset.

```
schoolareas <- read_sf('data/berlin_ezb.geojson') %>%
  select(-BSN, -BEREICH) %>%
  mutate(catchid = 1:nrow(.), .before = 1)
head(schoolareas)
```

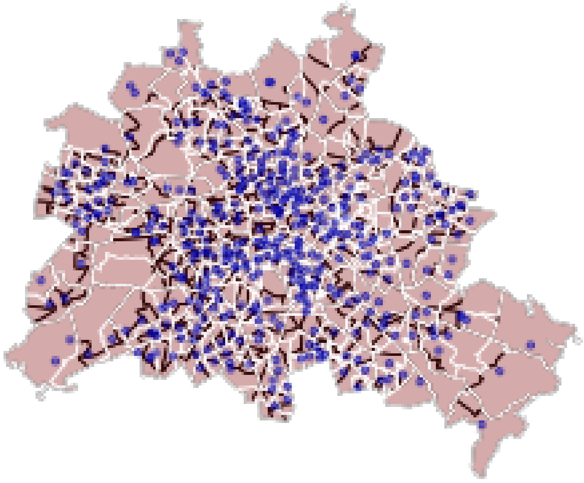
```
## Simple feature collection with 6 features and 1 field
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 384834.8 ymin: 5821206 xmax: 391331.4 ymax: 5825412
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 6 x 2
##   catchid                                                              geometry
##     <int>                                                    <MULTIPOLYGON [m]>
## 1       1 (((384834.8 5823436, 384868.2 5823434, 385030 5823424, 385075.1 58234~
## 2       2 (((388073.3 5824827, 388062.1 5824807, 387940.3 5824687, 387718.8 582~
## 3       3 (((389274.1 5824382, 389277.2 5824350, 389278.2 5824340, 389282.7 582~
## 4       4 (((387025.5 5822534, 387032.1 5822532, 387032.7 5822532, 387049.5 582~
```

[5] There's much debate about this and parents can try to register their children in a primary school outside their home catchment area but this comes with juristic obstacles, so we can assume for now that most will stay within their area.

```
## 5       5 (((388654.7 5823154, 388734.6 5823076, 388917.1 5822903, 388926 58229~
## 6       6 (((388843.8 5822105, 388843.3 5822105, 388770.6 5822038, 388728.5 582~
```

Let's generate a plot that overlays the planning areas, catchment areas and school locations. We can see that the catchment areas differ from the planning areas:

```
plot(bln$geometry)
plot(schoolareas$geometry, col = '#80000055', border = 'white', add = TRUE)
plot(schools$geometry, col = '#0000AA77', cex = 0.5, pch = 19, add = TRUE)
```

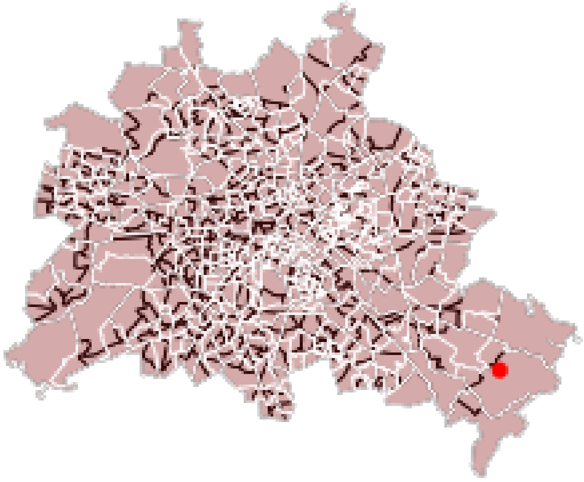


The goal is now to calculate the weighted average of the welfare rate for a given school by taking into account all planning areas that the school's catchment area intersects with. The weights will be determined by the intersection area between the catchment area and the planning areas. I will first do this with a single school only to illustrate how it works. This school will be #269 “Müggelheimer Schule” located in the south east of Berlin:

```
(exampleschool <- schools[schools$schoolid == 269,])
```

```
## Simple feature collection with 1 feature and 3 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 409366.8 ymin: 5807989 xmax: 409366.8 ymax: 5807989
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 1 x 4
##   schoolid ownership name                                          geometry
##      <int> <chr>     <chr>                                      <POINT [m]>
## 1      269 pub       Müggelheimer Schule (Grundschule) (409366.8 5807989)
```

```
plot(bln$geometry)
plot(schoolareas$geometry, col = '#80000055', border = 'white', add = TRUE)
plot(exampleschool$geometry, col = 'red', pch = 19, add = TRUE)
```



First, we need the catchment area of that school. We can again apply `st_join` for this in order to get the catchment area that intersects with the school. Note that the catchment areas should be the first argument in the `st_join` function since we want to retain the catchment areas' geometries in the resulting dataset. We also use an inner join instead of a left join by setting `left = FALSE` so that the result set only contains the single catchment area that intersects with the school.

```
(example_catchment_area <- st_join(schoolareas, exampleschool, left = FALSE))
```

```
## Simple feature collection with 1 feature and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 405754.5 ymin: 5804206 xmax: 414197.3 ymax: 5810542
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 1 x 5
##   catchid                                geometry schoolid ownership name
## *   <int>                      <MULTIPOLYGON [m]>    <int> <chr>     <chr>
## 1     254 (((405754.5 5807399, 405773.6 5807400,~      269 pub       Müggelheim~
```

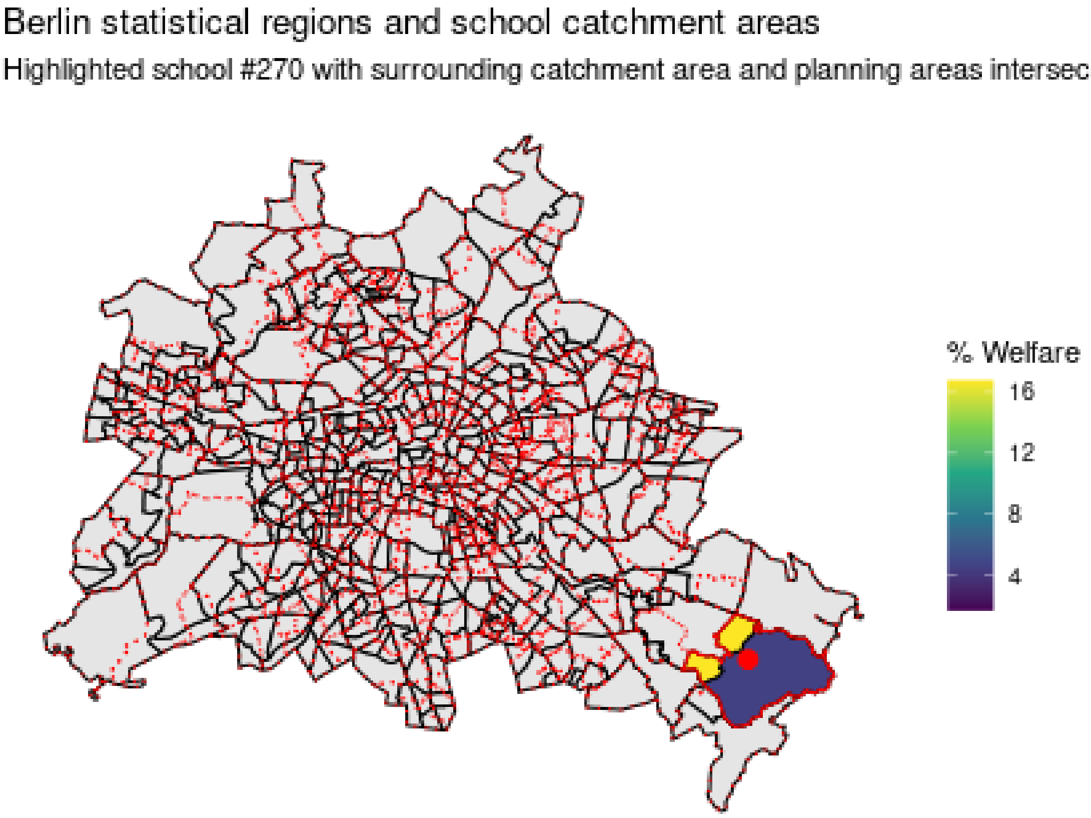
The next step is to get the intersections between the planning areas and catchment areas, i.e. to clip the planning areas according to the school's catchment area. We do this with the help of `st_intersection`, which calculates the intersection between spatial objects. The result is a spatial dataframe of seven planning regions that overlap with the school's catchment area:

```
(example_plr <- st_intersection(bln, example_catchment_area))
```

```
## Simple feature collection with 7 features and 8 fields
## Geometry type: GEOMETRY
## Dimension:     XY
## Bounding box:  xmin: 405754.5 ymin: 5804206 xmax: 414197.3 ymax: 5810542
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 7 x 9
##    areaid areaname    welfare welfare_chld catchid schoolid ownership name
## *   <int> <chr>         <dbl>        <dbl>   <int>    <int> <chr>     <chr>
## 1 9031101 Grünau         8.8         10.4      254      269 pub       Müggelhei~
## 2 9031201 Karolinenh~    2.01         1.73     254      269 pub       Müggelhei~
## 3 9031202 Schmöckwit~    6.25         9.47     254      269 pub       Müggelhei~
## 4 9041301 Kietzer Fe~   12.8         16.8      254      269 pub       Müggelhei~
## 5 9041601 Müggelheim     3.68         4.54     254      269 pub       Müggelhei~
## 6 9051702 Bölschestr~    7.63         7.58     254      269 pub       Müggelhei~
## 7 9051801 Rahnsdorf/~    5.89         5.67     254      269 pub       Müggelhei~
## # ... with 1 more variable: geometry <GEOMETRY [m]>
```

We can put that a little bit into perspective again and display it on the Berlin planning areas map overlayed with the schools' catchment areas. Here we can see that our example school's catchment area mainly intersects with only two planning areas. The other five intersections listed above are only tiny overlaps from surrounding planning areas, as we can also confirm next by computing their surface areas.

```
ggplot() +
  geom_sf(color = 'black', data = bln) +
  geom_sf(fill = NA, color = 'red', linetype = 'dotted', data = schoolareas) +
  geom_sf(aes(fill = welfare_chld), color = 'black', data = example_plr) +
  geom_sf(fill = NA, color = 'red', data = example_catchment_area) +
  geom_sf(color = 'red', size = 3, data = exampleschool) +
  scale_fill_continuous(type = 'viridis', guide = guide_colorbar(title = '% Welfare')) +
  coord_sf(datum = NA) +
  labs(title = "Berlin statistical regions and school catchment areas",
       subtitle = "Highlighted school #270 with surrounding catchment area and planning areas intersect
  theme_minimal()
```


Berlin statistical regions and school catchment areas

Highlighted school #270 with surrounding catchment area and planning areas intersection.

All that is left now for our example school is to take the weighted average of the welfare rate. The weights are the area of the planning area intersections so that planning areas with larger overlap in the catchment area have a higher influence on the overall average. The following shows the planning area intersections along with their area as calculated via `st_area`. We can see that the welfare rate of ~5% in Müggelheim will have the largest weight, followed by the ~17% rate in Kietzer Feld/Nachtheide:

```
cbind(example_plr[c('areaname', 'welfare_chld')], area = st_area(example_plr)) %>%
  mutate(weight = as.numeric(area / sum(area))) %>%
  arrange(desc(weight))
```

```
## Simple feature collection with 7 features and 4 fields
## Geometry type: GEOMETRY
## Dimension:     XY
## Bounding box:  xmin: 405754.5 ymin: 5804206 xmax: 414197.3 ymax: 5810542
## Projected CRS: ETRS89 / UTM zone 33N
##                        areaname welfare_chld               area       weight
## 1                   Müggelheim         4.54 2.217476e+07 [m^2] 8.013399e-01
## 2      Kietzer Feld/Nachtheide        16.75 5.479226e+06 [m^2] 1.980054e-01
## 3       Rahnsdorf/Hessenwinkel         5.67 1.683612e+04 [m^2] 6.084149e-04
## 4 Schmöckwitz/Rauchfangswerder         9.47 1.179133e+03 [m^2] 4.261089e-05
## 5                 Karolinenhof         1.73 7.181086e+01 [m^2] 2.595063e-06
## 6                       Grünau        10.39 2.955579e+01 [m^2] 1.068072e-06
## 7                Bölschestraße         7.58 1.774700e-02 [m^2] 6.413317e-10
##                         geometry
## 1 POLYGON ((406890.4 5806615,...
## 2 MULTIPOLYGON (((408688.7 58...
## 3 MULTIPOLYGON (((412927.1 58...
## 4 MULTIPOLYGON (((408424.5 58...
## 5 MULTIPOLYGON (((407010.6 58...
## 6 MULTIPOLYGON (((406448 5807...
## 7 POLYGON ((408688.9 5810542,...
```

We pass these area measurements to `weighted.mean` (stripping the m^2 unit via `as.numeric` since `weighted.mean` can't handle it) and obtain a weighted average welfare rate of ~7% which is quite a bit higher than the ~4.5% we get when using the former approach (linking the school with its planning area "Müggelheim"):

```
weighted.mean(example_plr$welfare_chld, as.numeric(st_area(example_plr)))
```

```
## [1] 6.958543
```

```
# former approach: linking the school with its planning area
schools_plan[schools_plan$schoolid == 269, ]$welfare_chld
```

```
## [1] 4.54
```

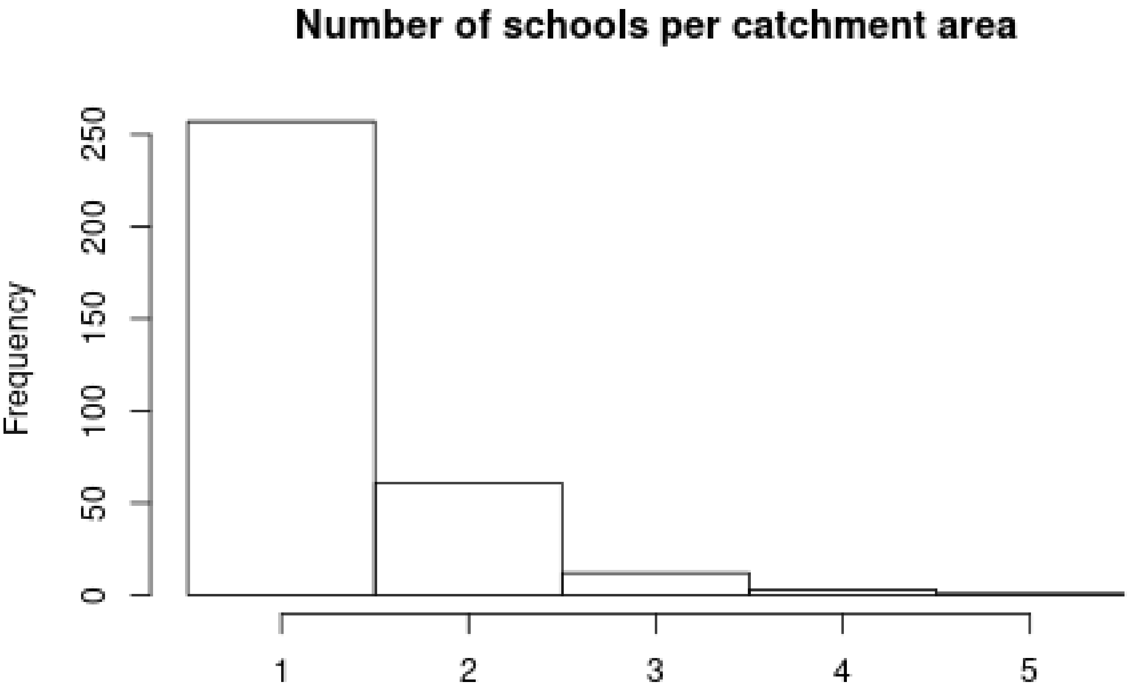
We'll next perform these calculations for all schools. First, we link each school with its catchment area using `st_join` as before:

```
schools_catch <- st_join(schoolareas, schools, left = FALSE)
head(schools_catch)
```

```
## Simple feature collection with 6 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 384834.8 ymin: 5822534 xmax: 391331.4 ymax: 5825412
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 6 x 5
##   catchid                                geometry schoolid ownership name
##     <int>                      <MULTIPOLYGON [m]>    <int> <chr>     <chr>
## 1       1 (((384834.8 5823436, 384868.2 5823434,~       25 pub       Möwensee-G~
## 2       1 (((384834.8 5823436, 384868.2 5823434,~       27 pub       Anna-Lindh~
## 3       2 (((388073.3 5824827, 388062.1 5824807,~       13 pub       Gottfried-~
## 4       2 (((388073.3 5824827, 388062.1 5824807,~       26 pub       Erika-Mann~
## 5       3 (((389274.1 5824382, 389277.2 5824350,~       17 pub       Wilhelm-Ha~
## 6       3 (((389274.1 5824382, 389277.2 5824350,~       19 pub       Carl-Kraem~
```

We confirm that there can be several schools in the same catchment area:

```
st_drop_geometry(schools_catch) %>%
  count(catchid) %>%
  pull(n) %>%
  hist(main = 'Number of schools per catchment area', breaks = 1:6 - 0.5, xlab = '')
```

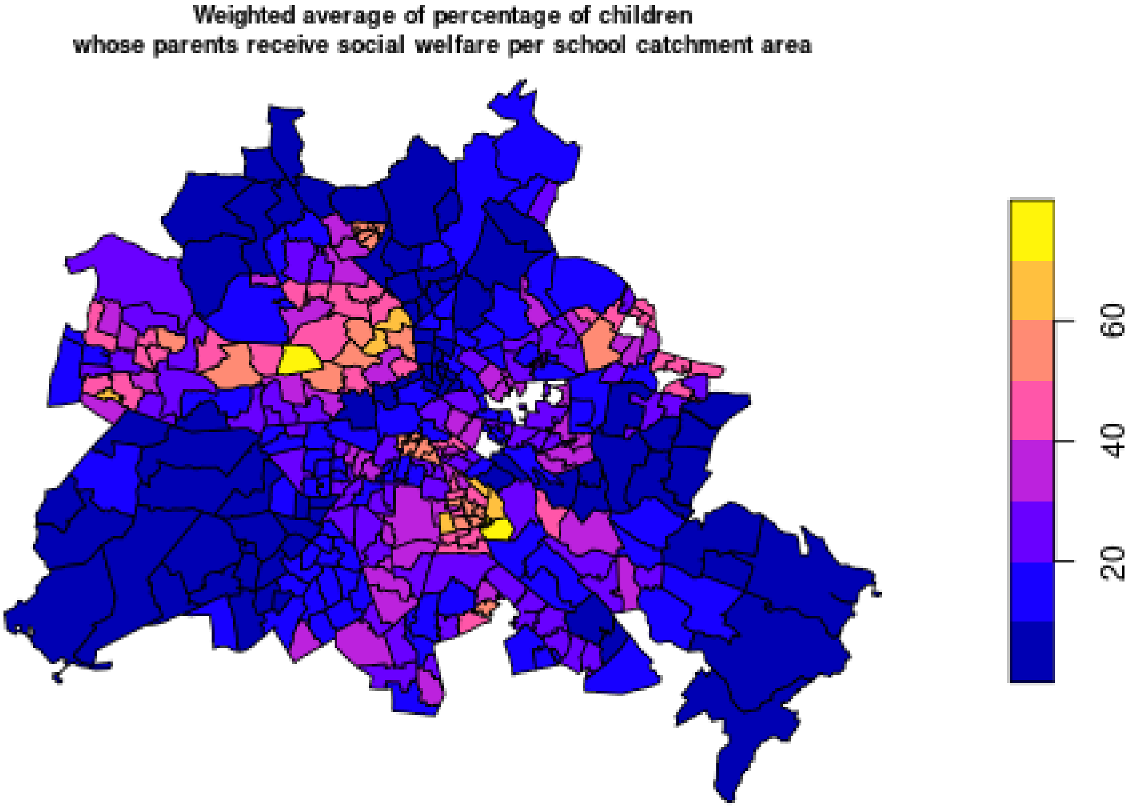


Next we calculate the planning area intersections, their areas and weighted average of the welfare rate for each school's catchment area using `sapply`. This computation takes some seconds to complete and in the end adds the weighted average of the welfare rate as `welfare_chld` variable to the schools' catchment area dataset:

```
spat_weighted_mean <- function(catch) {
  # the catchment area polygon "catch" loses the CRS during sapply -> set it here again
  catch <- st_sfc(catch, crs = st_crs(bln))
  areas <- st_intersection(bln, catch)
  weighted.mean(areas$welfare_chld, as.numeric(st_area(areas)))
}

schools_catch$welfare_chld <- sapply(schools_catch$geometry, spat_weighted_mean)
select(schools_catch, catchid, schoolid, ownership, name, welfare_chld) %>% head()
```

```
## Simple feature collection with 6 features and 5 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 384834.8 ymin: 5822534 xmax: 391331.4 ymax: 5825412
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 6 x 6
##   catchid schoolid ownership name     welfare_chld                       geometry
##     <int>    <int> <chr>     <chr>           <dbl>             <MULTIPOLYGON [m]>
## 1       1       25 pub       Möwense~         47.9 (((384834.8 5823436, 384868.~
## 2       1       27 pub       Anna-Li~         47.9 (((384834.8 5823436, 384868.~
## 3       2       13 pub       Gottfri~         50.1 (((388073.3 5824827, 388062.~
## 4       2       26 pub       Erika-M~         50.1 (((388073.3 5824827, 388062.~
## 5       3       17 pub       Wilhelm~         65.2 (((389274.1 5824382, 389277.~
## 6       3       19 pub       Carl-Kr~        65.2 (((389274.1 5824382, 389277.~
```

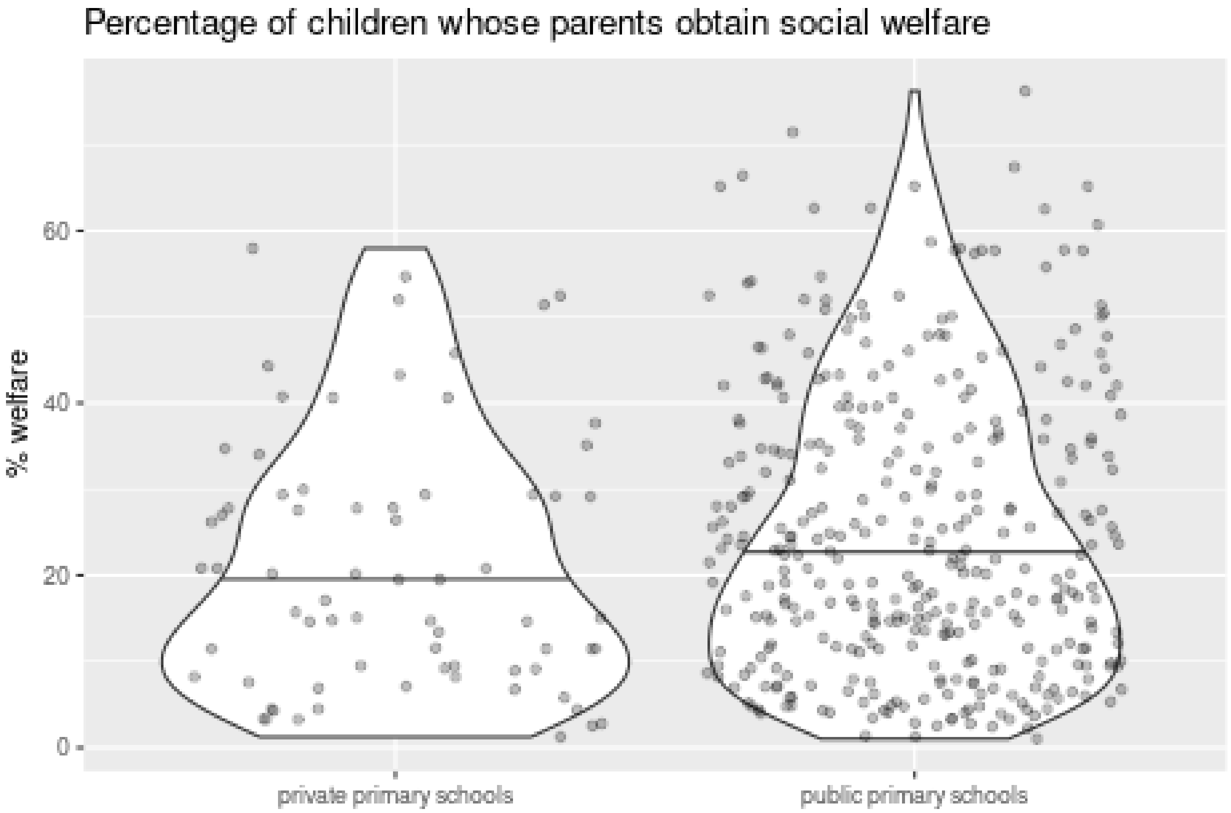
```
plot(distinct(schools_catch['welfare_chld']),
     main = 'Weighted average of percentage of children\nwhose parents receive social welfare per school
     cex.main = 0.75)
```


Weighted average of percentage of children
whose parents receive social welfare per school catchment area
60
40
20

Note that blank areas in the above figure represent catchment areas in which no primary school was located – this may be a flaw in the official data (12 such areas in total).

We again compare public and private schools, this time with our revised calculations:

```
ggplot(schools_catch) +
  geom_violin(aes(x = ownership, y = welfare_chld), draw_quantiles = c(0.5)) +
  geom_jitter(aes(x = ownership, y = welfare_chld), alpha = 0.25) +
  scale_x_discrete(labels = c('pub' = 'public primary schools', 'priv' = 'private primary schools')) +
  labs(title = 'Percentage of children whose parents obtain social welfare', x = '', y = '% welfare')
```



The median percentage of children whose parents obtain social welfare is still higher for public schools, but the difference is now five instead of six percent.

```
st_drop_geometry(schools_catch) %>%
  group_by(ownership) %>%
  summarise(median_welfare_chld = median(welfare_chld))
```

```
## # A tibble: 2 x 2
##   ownership median_welfare_chld
##   <chr>                   <dbl>
## 1 priv                     17.1
## 2 pub                      22.0
```

Our updated approach led to a difference that is only a bit smaller. The difference is not so large because of the very small catchment areas for the many schools in the inner city that result in a weighted average of the welfare rate that is very close to the rate of the schools' planning area. For other data, where catchment areas are bigger than the statistical regions (like in the example school in the south east of Berlin), you can expect a larger difference between the two approaches.

Approximating catchment areas as circular regions or Voronoi regions around schools

So far, we've assumed that private primary schools have the same catchment area as their nearby public schools, since there are no official catchment areas for private primary schools and parents can choose more freely which school they send their children to when they prefer a private school. So if we have no spatial data for about the catchment areas of private primary schools, what can we do?

One possibility would be to construct a circle around each private school which represents the catchment area for a certain radius. This can be done via `st_buffer`. However, it's hard to justify a certain value for that radius and the radius for such a catchment area should probably vary depending on where the school is located (smaller catchment areas in inner city schools than for schools in the outskirts).

Another approach relies on Voronoi regions. They partition the space between given points so that the Voronoi region around each point covers an area of minimal distance to that origin point. In other words: the Voronoi region around a school is the area in which all households are located that are closest to that school. It is reasonable to assume that most parents choose among the closest private schools to their home. This means approximating the catchment area of private primary schools as Voronoi regions may be a good option, while still using the official public primary school catchment areas only for the public schools.

Voronoi regions can be generated with `st_voronoi`, which accepts the points as `MULTIPOINT` geometry object. The second argument is an envelope polygon for which we'll use the the Berlin borders. The resulting object is a `GEOMETRYCOLLECTION` geometry object which we pass on to `st_collection_extract` and `st_sfc` in order to transform this to a *geometry set* object that has the same CRS as our other spatial data (ETRS89).

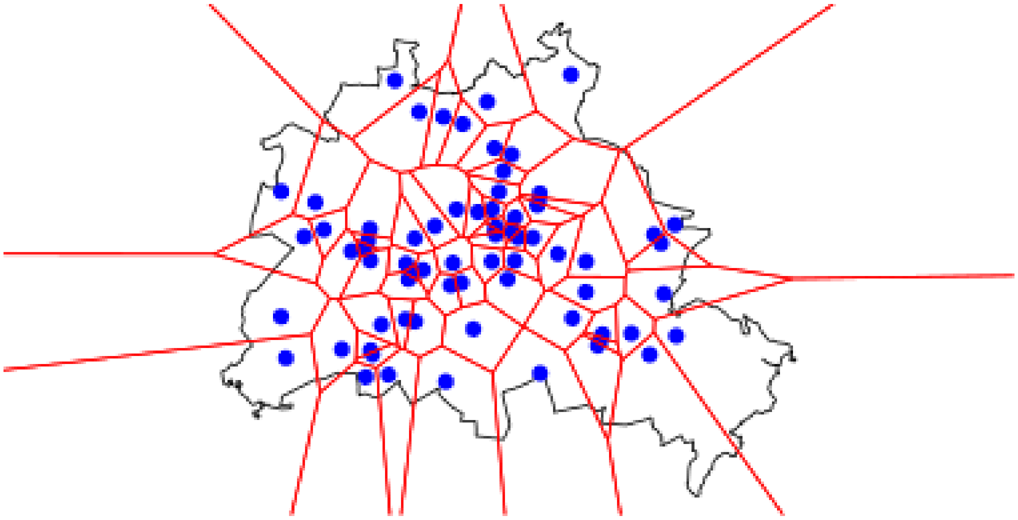
Let's generate the Voronoi regions around all private primary schools:

```r
bln_outline <- st_union(bln$geometry)  # Berlin borders

privschools <- filter(schools, ownership == 'priv')
pubschools <- filter(schools, ownership == 'pub')

(priv_voro <-  st_coordinates(privschools$geometry) %>%
  st_multipoint() %>%
  st_voronoi(bln_outline) %>%
  st_collection_extract() %>%
  st_sfc(crs = st_crs(bln)))
```

```
## Geometry set for 71 features
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 343176.8 ymin: 5777090 xmax: 437820.8 ymax: 5864683
## Projected CRS: ETRS89 / UTM zone 33N
## First 5 geometries:

## POLYGON ((375671.8 5822022, 378042.6 5829508, 3...

## POLYGON ((343176.8 5818841, 343176.8 5864683, 3...

## POLYGON ((343176.8 5808476, 343176.8 5818841, 3...

## POLYGON ((377498.9 5816477, 369373 5818823, 375...

## POLYGON ((382822.9 5777090, 370640.2 5777090, 3...
```

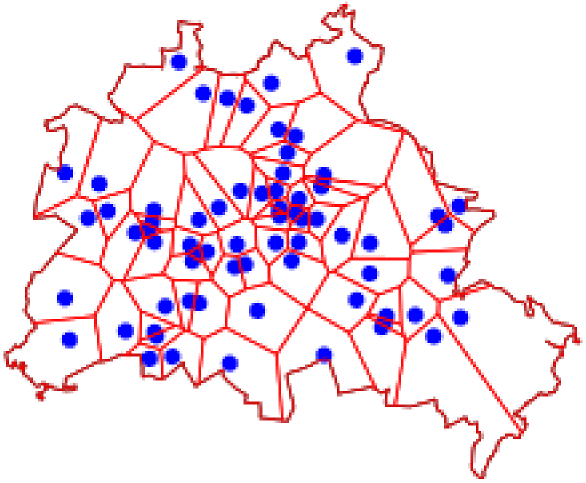
We can now plot the generated regions along with the school locations:

```r
plot(bln_outline)
plot(privschools$geometry, col = 'blue', pch = 19, add = TRUE)
plot(priv_voro, border = 'red', col = NA, add = TRUE)
```



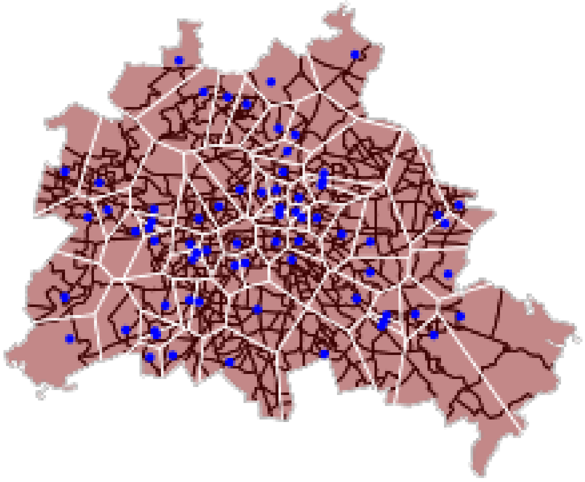
We can see that the Voronoi regions extend beyond the borders of Berlin so we should take the intersection between the Voronoi regions and the Berlin border in order to clip these regions:

```
priv_voro <- st_intersection(priv_voro, bln_outline)

plot(bln_outline)
plot(privschools$geometry, col = 'blue', pch = 19, add = TRUE)
plot(priv_voro, border = 'red', col = NA, add = TRUE)
```



Let's overlay the planning areas with the private schools' Voronoi regions to see how they differ.

```
plot(bln$geometry)
plot(priv_voro,  col = '#80000077', border = 'white', add = TRUE)
plot(privschools$geometry, col = 'blue', pch = 19, cex = 0.5, add = TRUE)
```



Once we have circular regions or Voronoi regions around the private schools, the rest of the calculations are similar to those with the official catchment areas. We link the private schools with their approximated catchment areas and apply the `spat_weighted_mean` function that we've defined before. I've done this in the R notebook for this blog article.

Linking the private schools with their Voronoi regions:

```
privschools_voro <- st_as_sf(priv_voro) %>%
  mutate(voroid = 1:nrow(.)) %>%
  st_join(privschools, left = FALSE) %>%
  rename(geometry = x)
head(privschools_voro)
```

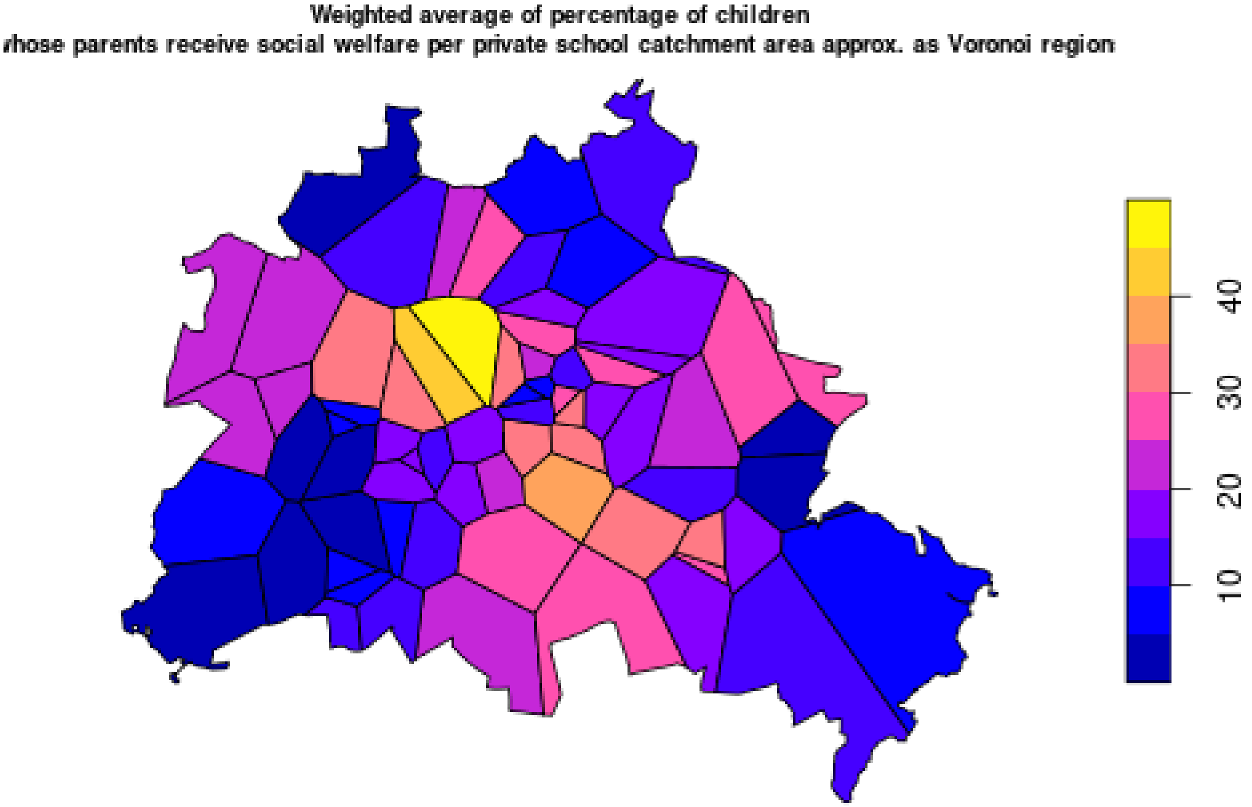
```
## Simple feature collection with 6 features and 4 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 370000.7 ymin: 5805751 xmax: 382549.8 ymax: 5829223
## Projected CRS: ETRS89 / UTM zone 33N
##   voroid schoolid ownership
## 1      1      399      priv
## 2      2      400      priv
## 3      3      366      priv
## 4      4      426      priv
## 5      5      368      priv
## 6      6      405      priv
##                                                     name
## 1 Katholische Schule Bernhard Lichtenberg (Grundschule)
## 2                                  Immanuel-Grundschule
## 3        Freie Waldorfschule Havelhöhe - Eugen Kolisko
## 4                                  Wilhelmstadt Schulen
## 5                                      Emil-Molt-Schule
## 6  Internationale Montessorischule Berlin (Grundschule)
##                           geometry
## 1 POLYGON ((375671.8 5822022,...
## 2 POLYGON ((377622.8 5828183,...
## 3 POLYGON ((374067.4 5817468,...
## 4 POLYGON ((377498.9 5816477,...
## 5 POLYGON ((379582.4 5809287,...
## 6 POLYGON ((372195.4 5811848,...
```

Checking that there's really only one private school per Voronoi region:

```
st_drop_geometry(privschools_voro) %>%
  count(voroid) %>%
  pull(n) %>%
  all(. == 1) %>%
  stopifnot()
```

Generating the weighted averages and plotting the corresponding choropleth map for the private schools:

```
privschools_voro$welfare_chld <- sapply(privschools_voro$geometry, spat_weighted_mean)
plot(privschools_voro['welfare_chld'],
     main = 'Weighted average of percentage of children\nwhose parents receive social welfare per priva
     cex.main = 0.75)
```


Weighted average of percentage of children
whose parents receive social welfare per private school catchment area approx. as Voronoi region
10
20
30
40

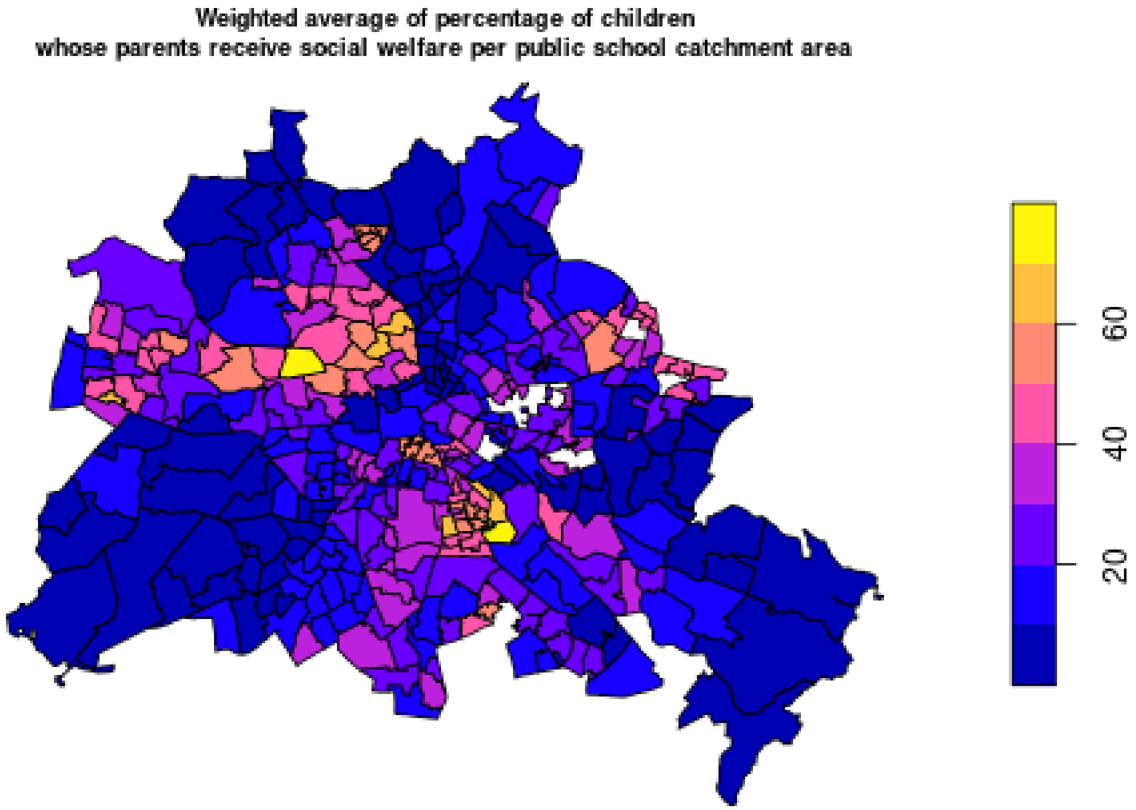
Linking the public schools with their catchment areas:

```
pubschools_catch <- st_join(schoolareas, pubschools, left = FALSE)
head(pubschools_catch)
```

```
## Simple feature collection with 6 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 384834.8 ymin: 5822534 xmax: 391331.4 ymax: 5825412
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 6 x 5
##   catchid                                geometry schoolid ownership name
##     <int>                      <MULTIPOLYGON [m]>    <int> <chr>     <chr>
## 1       1 (((384834.8 5823436, 384868.2 5823434,~       25 pub       Möwensee-G~
## 2       1 (((384834.8 5823436, 384868.2 5823434,~       27 pub       Anna-Lindh~
## 3       2 (((388073.3 5824827, 388062.1 5824807,~       13 pub       Gottfried-~
## 4       2 (((388073.3 5824827, 388062.1 5824807,~       26 pub       Erika-Mann~
## 5       3 (((389274.1 5824382, 389277.2 5824350,~       17 pub       Wilhelm-Ha~
## 6       3 (((389274.1 5824382, 389277.2 5824350,~       19 pub       Carl-Kraem~
```

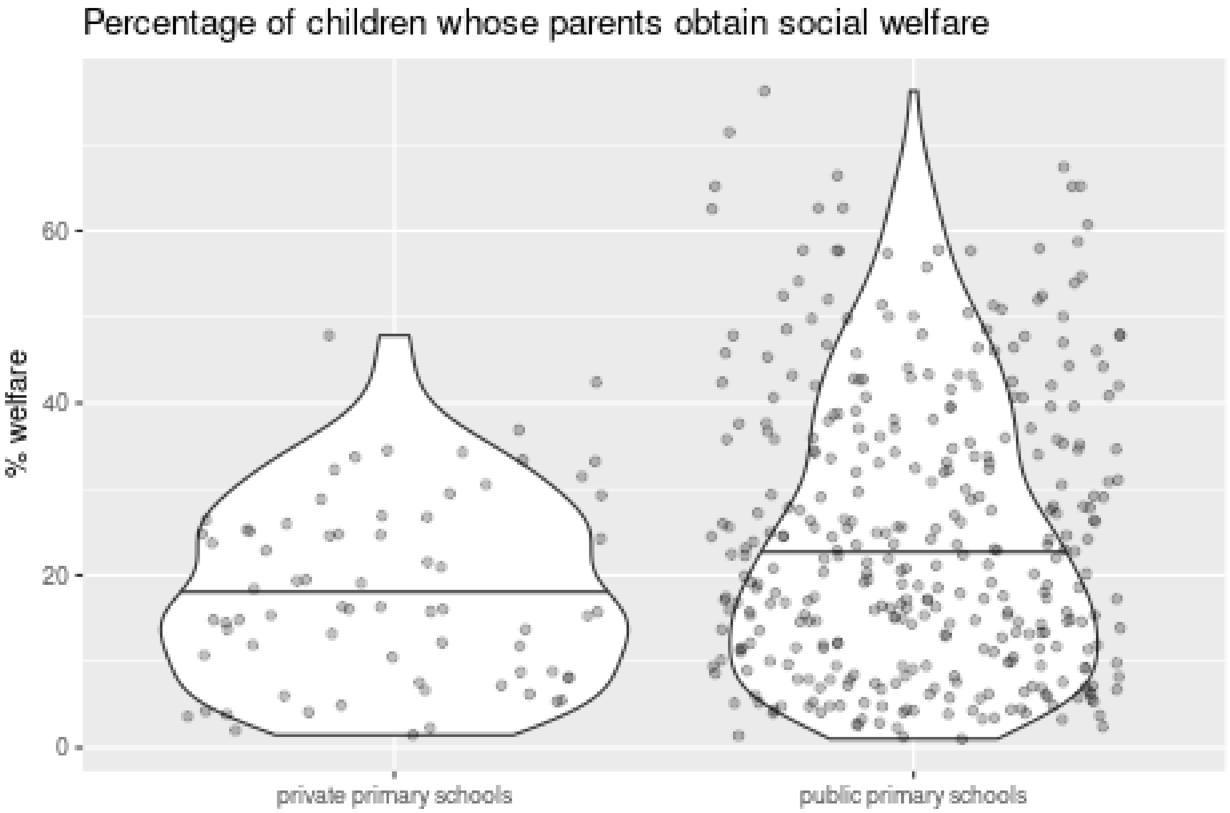
Generating the weighted averages and plotting the corresponding choropleth map for the public schools:

```
pubschools_catch$welfare_chld <- sapply(pubschools_catch$geometry, spat_weighted_mean)
plot(distinct(pubschools_catch['welfare_chld']),
     main = 'Weighted average of percentage of children\nwhose parents receive social welfare per publi
     cex.main = 0.75)
```


Weighted average of percentage of children
whose parents receive social welfare per public school catchment area
60
40
20

Combining the results from the public and private schools:

```
pubpriv <- bind_rows(select(pubschools_catch, ownership, welfare_chld),
                     select(privschools_voro, ownership, welfare_chld)) %>%
  st_drop_geometry()

ggplot(pubpriv) +
  geom_violin(aes(x = ownership, y = welfare_chld), draw_quantiles = c(0.5)) +
  geom_jitter(aes(x = ownership, y = welfare_chld), alpha = 0.25) +
  scale_x_discrete(labels = c('pub' = 'public primary schools', 'priv' = 'private primary schools')) +
  labs(title = 'Percentage of children whose parents obtain social welfare', x = '', y = '% welfare')
```



```
group_by(pubpriv, ownership) %>%
  summarise(median_welfare_chld = median(welfare_chld))
```

```
## # A tibble: 2 x 2
##   ownership median_welfare_chld
##   <chr>                   <dbl>
## 1 priv                     16.0
## 2 pub                      22.0
```

Conclusion

The descriptive results suggest that private schools in Berlin may tend to be located in areas with lower rates of children whose parents obtain social welfare as compared to public schools. Further spatial analysis could be done to test this hypothesis.

We have seen how we can calculate a weighted average for some variable of interest for a catchment area around sample points, when this variable of interest was measured for regions that overlap with that catchment area. In the best case scenario, you know the geometry of the catchment areas. Otherwise you may need to approximate them, for example as circular regions around the points or as Voronoi regions. Additionally, you may consider nearest-feature-joins or travel time isochrones. Which option is more appropriate depends on your use-case.