Spatially weighted averages with Voronoi regions

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 catchment areas are approximated as Voronoi regions. Overlaps between these Voronoi regions and the statistical regions allow to calculate the weighted average of the socioeconomic statistics.

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 three datasets: The first spatial dataset contains the shape of the statistical regions in Berlin, the second dataset contains the socioeconomic data for these regions and the third dataset contains the locations and other attributes of primary schools in Berlin.

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.

```
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).

```
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 region 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)
```

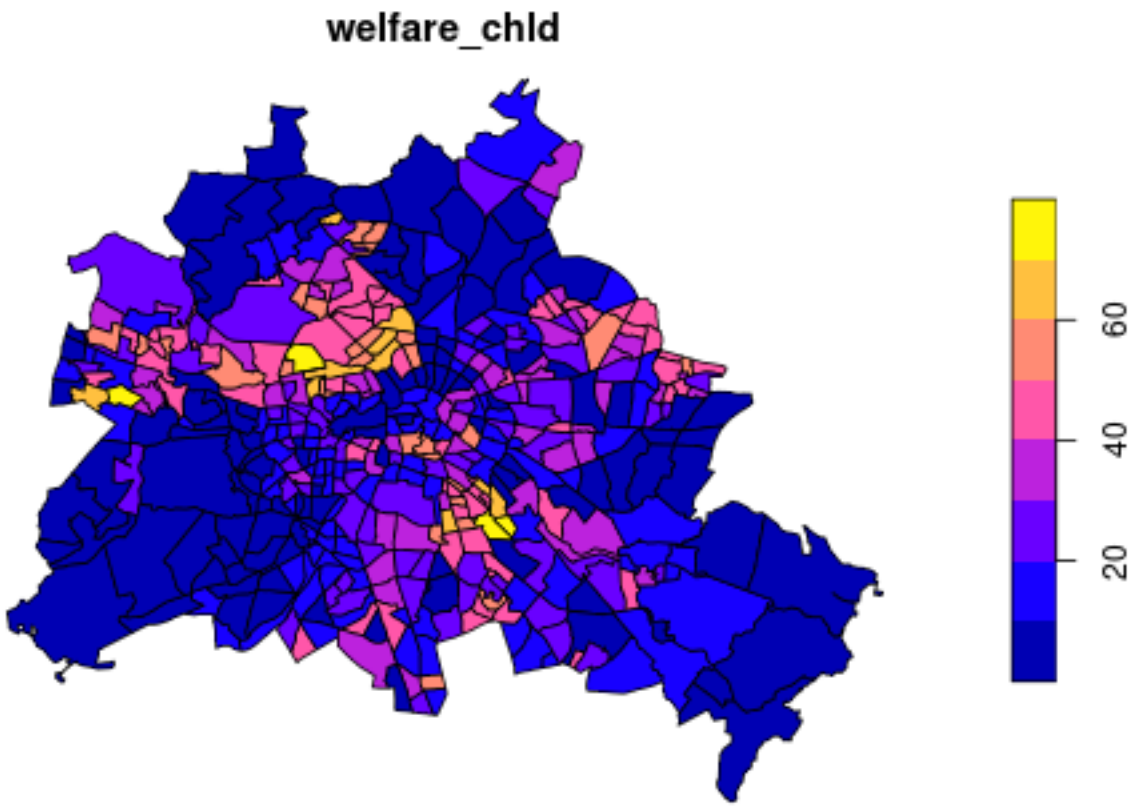
```
## Simple feature collection with 6 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 386668.2 ymin: 5817761 xmax: 390764.3 ymax: 5820432
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 6 x 5
##    areaid                              geometry areaname    welfare welfare_chld
##     <int>                      <MULTIPOLYGON [m]> <chr>         <dbl>        <dbl>
## 1 1011101 (((387256.6 5818552, 387323.1 581857~ Stülerstra~   10.1         15.4
## 2 1011102 (((386767.5 5819393, 386768.3 581938~ Großer Tie~    4.76         0
```

[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`).

```
## 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]

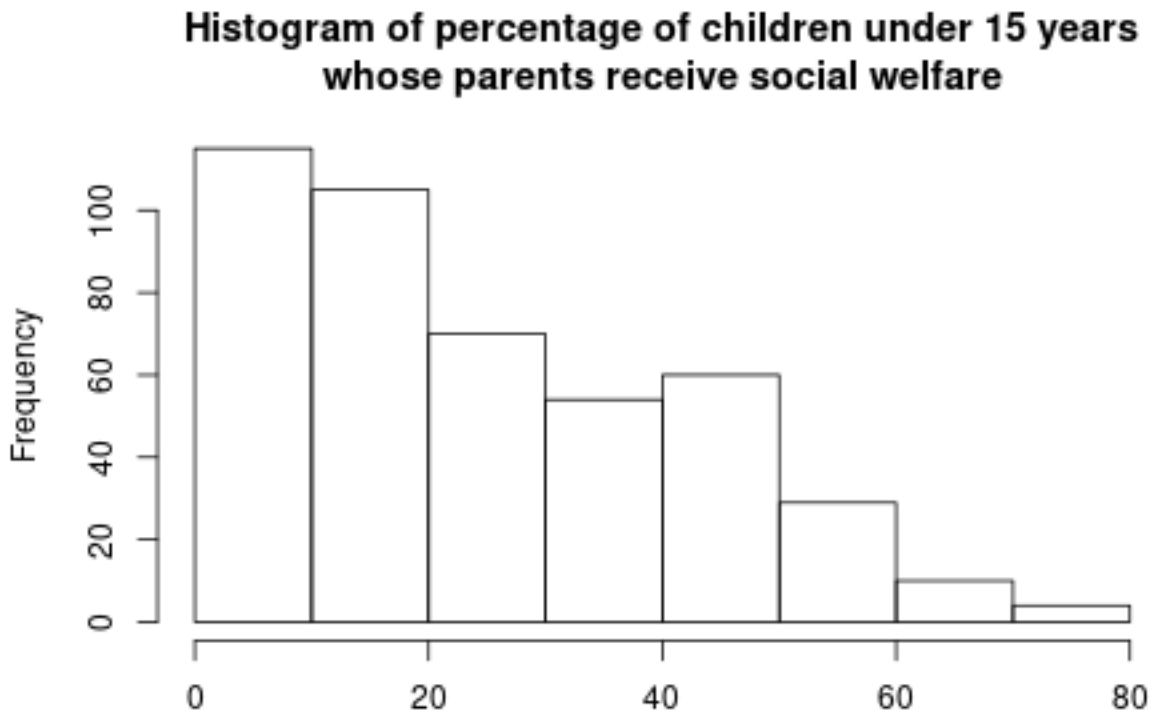
```
plot(bln['welfare_chld'])
```



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

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 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`.

```
schools <- read.csv('data/grundschulen_berlin_2019.csv', stringsAsFactors = FALSE) %>%
  st_as_sf(coords = c('lng', 'lat'), crs = 4326)  # EPSG 4326 is WGS84 lat/long coord.
head(schools)
```

```
## Simple feature collection with 6 features and 5 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 13.38033 ymin: 52.51312 xmax: 13.42395 ymax: 52.53684
## Geodetic CRS:  WGS 84
##   traeger priv_schule_typ                                name          ort   plz
## 1    oeff                        Grundschule am Arkonaplatz Berlin Mitte 10115
## 2    oeff                             Papageno-Grundschule Berlin Mitte 10115
## 3    oeff                         Kastanienbaum-Grundschule Berlin Mitte 10119
## 4    oeff                             Grundschule Neues Tor Berlin Mitte 10115
## 5    oeff                             GutsMuths-Grundschule Berlin Mitte 10179
## 6    oeff                    Grundschule am Brandenburger Tor Berlin Mitte 10117
##                     geometry
## 1 POINT (13.40039 52.53684)
```

```
## 2 POINT (13.39117 52.53274)
## 3  POINT (13.4018 52.52678)
## 4 POINT (13.38033 52.52759)
## 5 POINT (13.42395 52.51642)
## 6 POINT (13.38287 52.51312)
```

The variable `traeger` encodes whether a given facility is a public ("oeff") or private ("priv") primary school. The only other important feature will be the school's location, represented as point geometries that use the WGS84 CRS. The Berlin planning area data uses a different CRS, namely ETRS89 / UTM zone 33N. We need to make sure that both datasets use the same CRS, otherwise spatial operations such as spatial joins will fail. WGS84 uses spherical coordinates measured in degrees and calculations with these coordinates are quite complex because they happen on a curved surface. It's better to use a CRS that uses planar coordinates for all the following spatial operations. The ETRS89 CRS used in the Berlin planning area data uses planar coordinates measured in meters, so we should transform the school locations to this CRS using `st_transform`:

```
schools <- mutate(schools, schoolid = 1:nrow(schools), .before = 1) %>%
  st_transform(crs = st_crs(bln_plan))
head(schools)
```

```
## Simple feature collection with 6 features and 6 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 390124.5 ymin: 5819340 xmax: 393056.4 ymax: 5821952
## Projected CRS: ETRS89 / UTM zone 33N
##   schoolid traeger priv_schule_typ                              name
## 1        1    oeff                       Grundschule am Arkonaplatz
## 2        2    oeff                             Papageno-Grundschule
## 3        3    oeff                        Kastanienbaum-Grundschule
## 4        4    oeff                            Grundschule Neues Tor
## 5        5    oeff                            GutsMuths-Grundschule
## 6        6    oeff                 Grundschule am Brandenburger Tor
##            ort   plz                  geometry
## 1 Berlin Mitte 10115   POINT (391508 5821952)
## 2 Berlin Mitte 10115 POINT (390872.9 5821510)
## 3 Berlin Mitte 10119 POINT (391578.9 5820831)
## 4 Berlin Mitte 10115 POINT (390124.5 5820954)
## 5 Berlin Mitte 10179 POINT (393056.4 5819646)
## 6 Berlin Mitte 10117 POINT (390260.6 5819340)
```

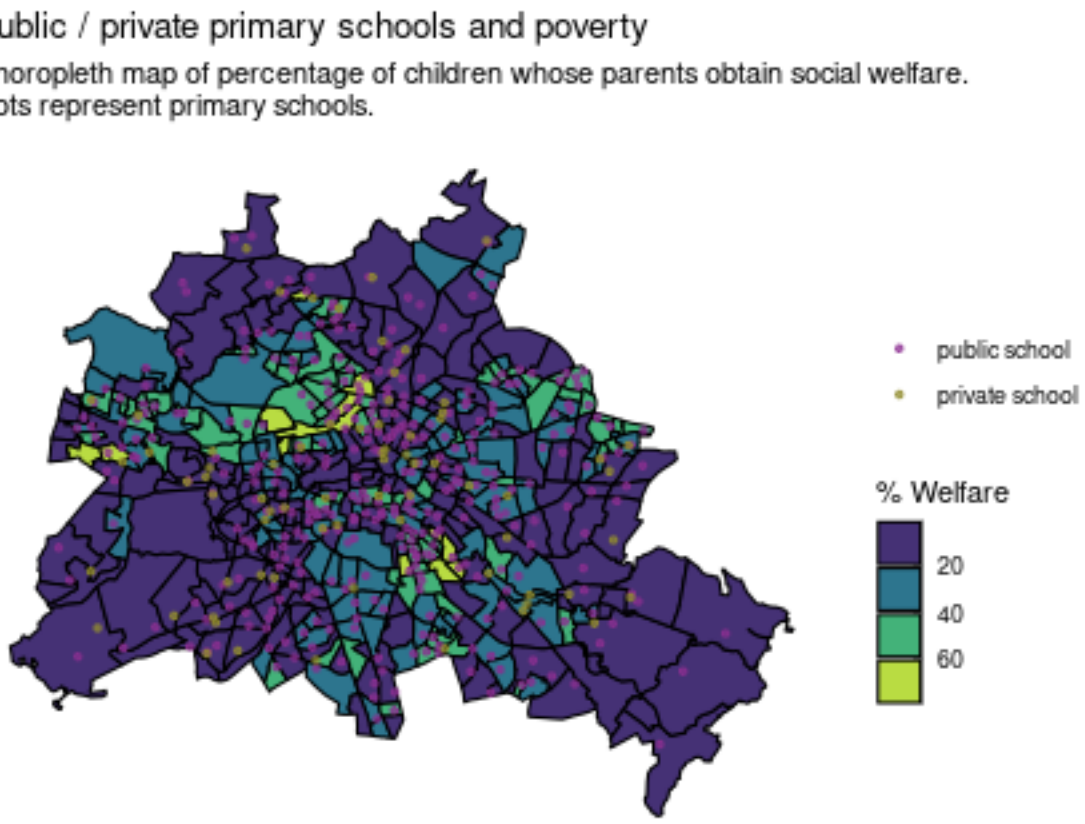
In this dataset from 2019, we have 364 public and 71 private primary schools in Berlin.

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 = traeger), size = 1, alpha = 0.75, data = schools) +
  scale_fill_binned(type = 'viridis', guide = guide_bins(title = '% Welfare')) +
  scale_color_manual(values = c('oeff' = '#8C2F92',  'priv' = '#928C2F'),
                     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 r
  theme_minimal()
```



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 10 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 390124.5 ymin: 5819340 xmax: 393056.4 ymax: 5821952
## Projected CRS: ETRS89 / UTM zone 33N
##   schoolid traeger priv_schule_typ                            name
## 1        1    oeff                      Grundschule am Arkonaplatz
## 2        2    oeff                            Papageno-Grundschule
## 3        3    oeff                     Kastanienbaum-Grundschule
## 4        4    oeff                         Grundschule Neues Tor
## 5        5    oeff                         GutsMuths-Grundschule
```

```
## 6        6    oeff                Grundschule am Brandenburger Tor
##            ort   plz  areaid             areaname welfare welfare_chld
## 1 Berlin Mitte 10115 1011402          Arkonaplatz    4.46         3.53
## 2 Berlin Mitte 10115 1011401      Invalidenstraße    6.29         7.70
## 3 Berlin Mitte 10119 1011302 Oranienburger Straße    8.16        10.44
## 4 Berlin Mitte 10115 1011301       Charitéviertel    3.68         3.92
## 5 Berlin Mitte 10179 1011304      Karl-Marx-Allee   23.54        36.33
## 6 Berlin Mitte 10117 1011201        Wilhelmstraße   12.13        19.03
##                   geometry
## 1   POINT (391508 5821952)
## 2 POINT (390872.9 5821510)
## 3 POINT (391578.9 5820831)
## 4 POINT (390124.5 5820954)
## 5 POINT (393056.4 5819646)
## 6 POINT (390260.6 5819340)
```

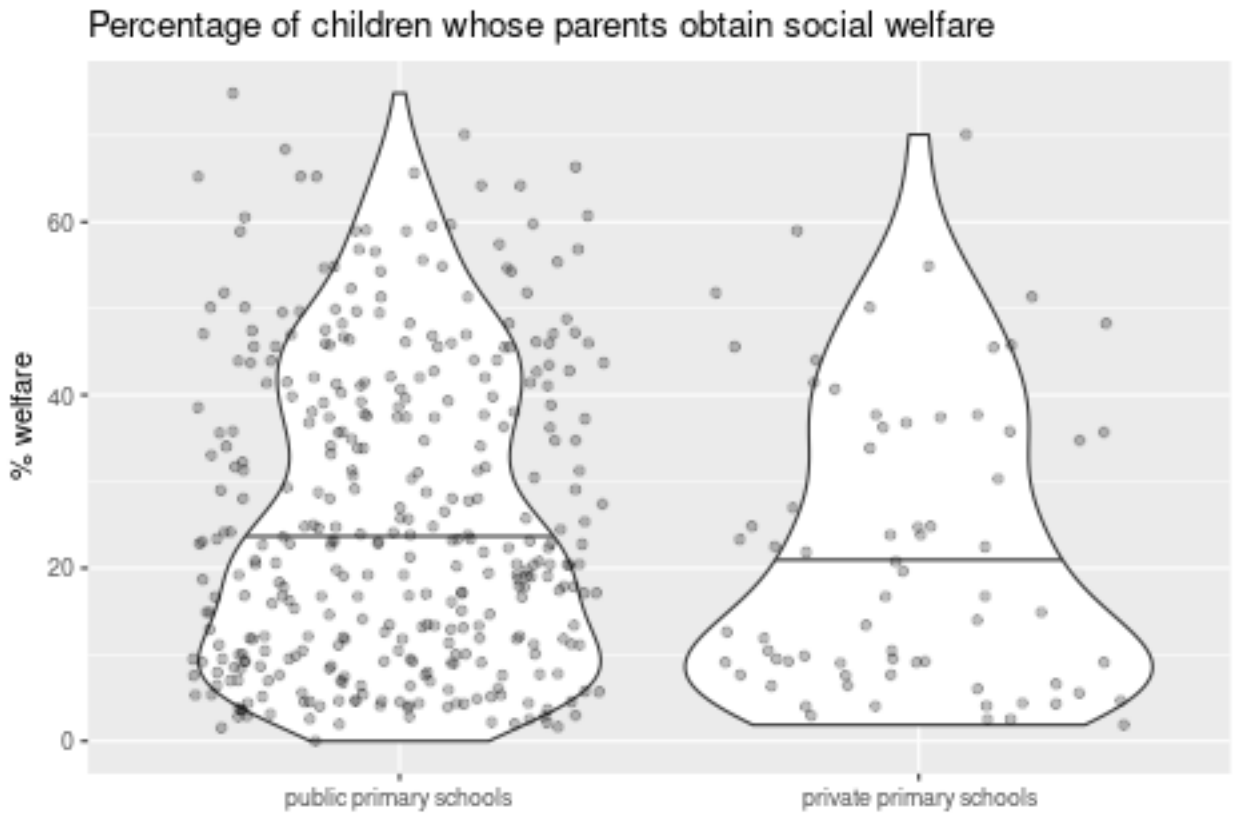
We can see that the schools' data was linked with the data from the planning areas. We should also check whether there's a school that was not located in any planning area (this may for example happen when a school is very close to the Berlin-Brandenburg border):

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

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

All schools were linked with their planning region, 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 = traeger, y = welfare_chld), draw_quantiles = c(0.5)) +
  geom_jitter(aes(x = traeger, y = welfare_chld), alpha = 0.25) +
  scale_x_discrete(labels = c('oeff' = '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: [3]

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

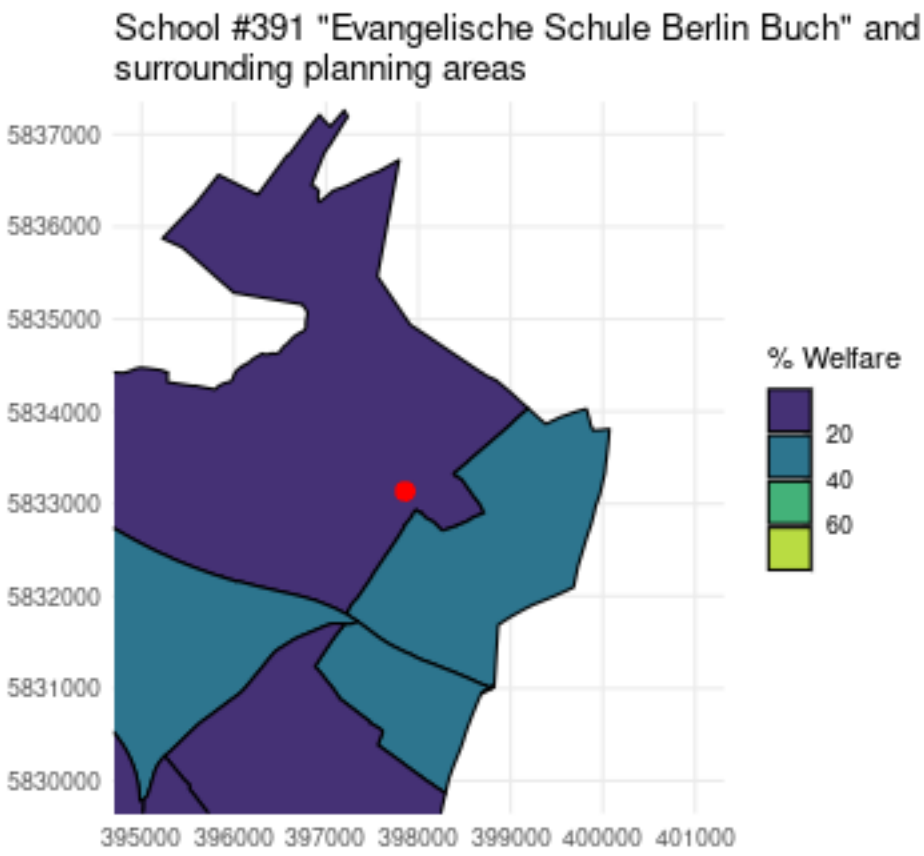
```
## # A tibble: 2 x 2
##   traeger median_welfare_chld
##   <chr>                 <dbl>
## 1 oeff                   22.8
## 2 priv                   16.8
```

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 #391 "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 == 391), 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 #391 "Evangelische Schule Berlin Buch" and\nsurrounding planning areas') +
```

[3] I'm using `st_drop_geometry` here, because otherwise a spatial aggregation would be performed which is not necessary here.

```
theme_minimal()
```



Using Voronoi regions as catchment areas

We can improve on this by using each schools' catchment area and then taking the weighted average of the welfare variable from the intersections with the planning areas. You may be able to get spatial data on school catchment areas from administrative sources, but this is not often the case. So how could we define such a catchment area? One possibility would be to construct a circle around each school which represents the catchment area for a certain radius. 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's a reasonable assumption that catchment areas for schools can be approximated by Voronoi regions in Berlin since children are by law assigned to schools near their place of residence. Parents can try to register their children at another school, but priority is given to children that live closer to that school. A problem is that this "automatical" assignment happens for public schools only and that the catchment areas of private schools are probably larger, but for the sake of simplicity in this example we'll stick to the Voronoi regions.

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).

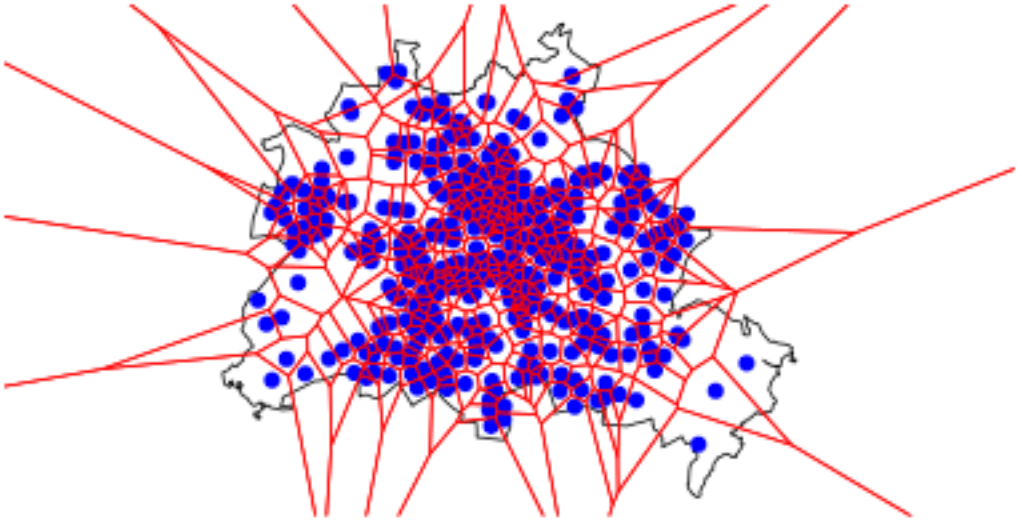
```
bln_outline <- st_union(bln$geometry)  # Berlin borders

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

```
## Geometry set for 435 features
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 333829.7 ymin: 5764709 xmax: 450831.3 ymax: 5872408
## Projected CRS: ETRS89 / UTM zone 33N
## First 5 geometries:
## POLYGON ((333829.7 5824739, 333829.7 5843705, 3...
## POLYGON ((333829.7 5805022, 333829.7 5824739, 3...
## POLYGON ((373574.4 5820108, 374364.5 5821773, 3...
## POLYGON ((360399.8 5809719, 373736.3 5814453, 3...
## POLYGON ((373887.8 5764709, 333829.7 5764709, 3...
```

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

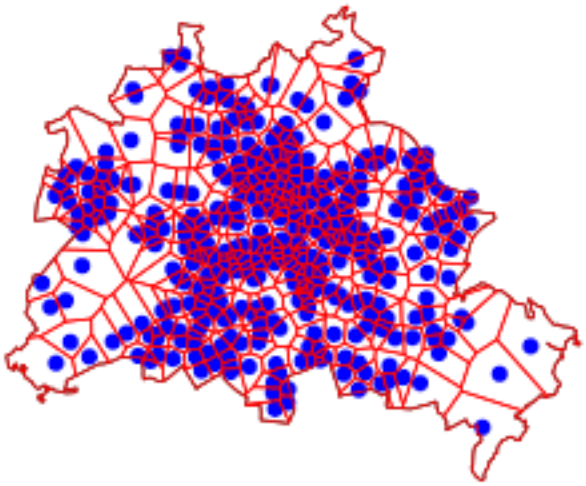
```
{
  plot(bln_outline)
  plot(schools$geometry, col = 'blue', pch = 19, add = TRUE)
  plot(voronois, border = 'red', col = NA, add = TRUE)
}
```



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:

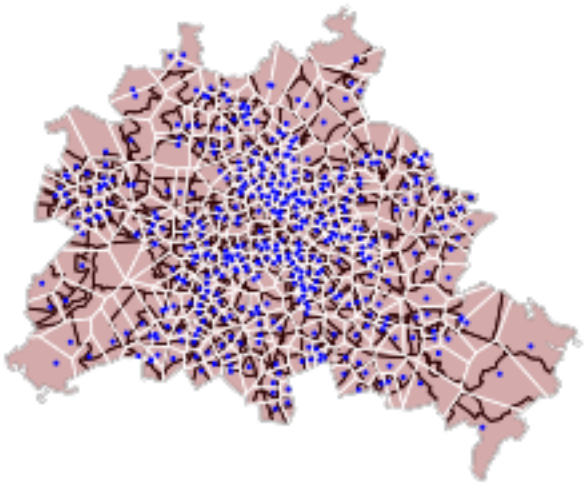
```
bln_vor <- st_intersection(voronois, bln_outline)

{
  plot(bln_outline)
  plot(schools$geometry, col = 'blue', pch = 19, add = TRUE)
  plot(bln_vor, border = 'red', col = NA, add = TRUE)
}
```



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

```
{
  plot(bln$geometry)
  plot(bln_vor, col = '#80000055', border = 'white', add = TRUE)
  plot(schools$geometry, col = 'blue', pch = 19, cex = 0.25, add = TRUE)
}
```

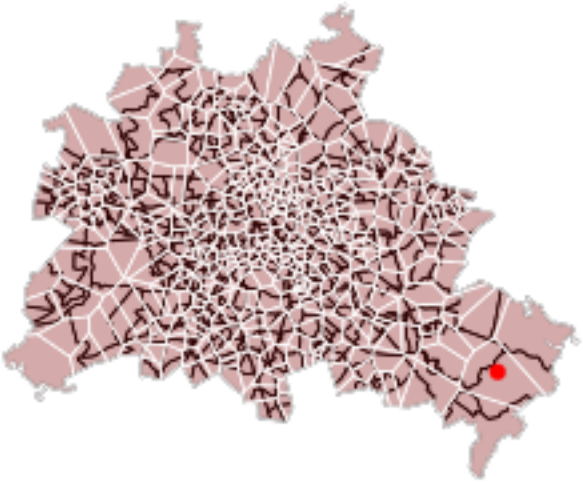


The goal is now to calculate the weighted average of the welfare rate for a given school by taking into account all planning areas that the school's Voronoi region intersects with. The weights will be determined by the intersection area between the Voronoi region and the planning areas. I will first do this with a single school only to illustrate how it works. This school will be #270 "Müggelheimer Schule" located in the south east of Berlin:

```
(exampleschool <- schools[schools$schoolid == 270,])
```

```
## Simple feature collection with 1 feature and 6 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 409354.2 ymin: 5807970 xmax: 409354.2 ymax: 5807970
## Projected CRS: ETRS89 / UTM zone 33N
##     schoolid traeger priv_schule_typ                                name
## 270      270    oeff                Müggelheimer Schule (Grundschule)
##                         ort   plz                 geometry
## 270 Berlin Treptow-Köpenick 12559 POINT (409354.2 5807970)
```

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



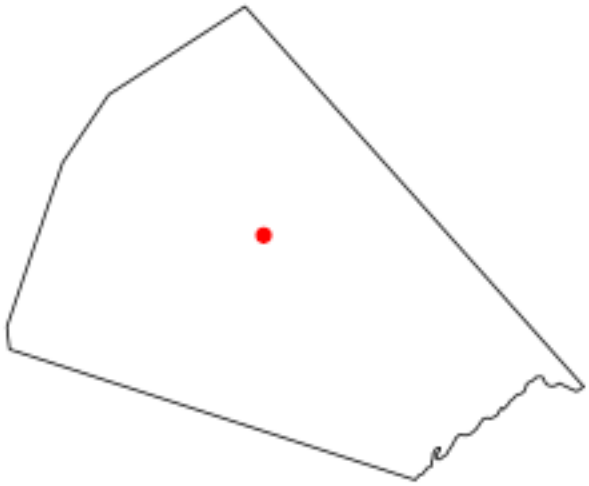
First, we need the Voronoi region of that school. We can again apply `st_join` for this in order to get the Voronoi region that intersects with the school. `st_join` only works with spatial dataframes so we need to convert the geometry set object via `st_as_sf`. Note that the Voronoi regions should be the first argument in the `st_join` function since we want to retain the region's geometry 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 Voronoi region that intersects with the school.

```
(examplevor <- st_join(st_as_sf(bln_vor), exampleschool, left = FALSE))
```

```
## Simple feature collection with 1 feature and 6 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 406269.6 ymin: 5805035 xmax: 413198.9 ymax: 5810714
## Projected CRS: ETRS89 / UTM zone 33N
##   schoolid traeger priv_schule_typ                                name
## 1      270    oeff                Müggelheimer Schule (Grundschule)
##                       ort   plz                       geometry
## 1 Berlin Treptow-Köpenick 12559 POLYGON ((406269.6 5806871,...
```

We can see that the extracted Voronoi region seems to be correct:

```
{
  plot(examplevor$geometry)
  plot(exampleschool$geometry, col = 'red', pch = 19, add = TRUE)
}
```



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

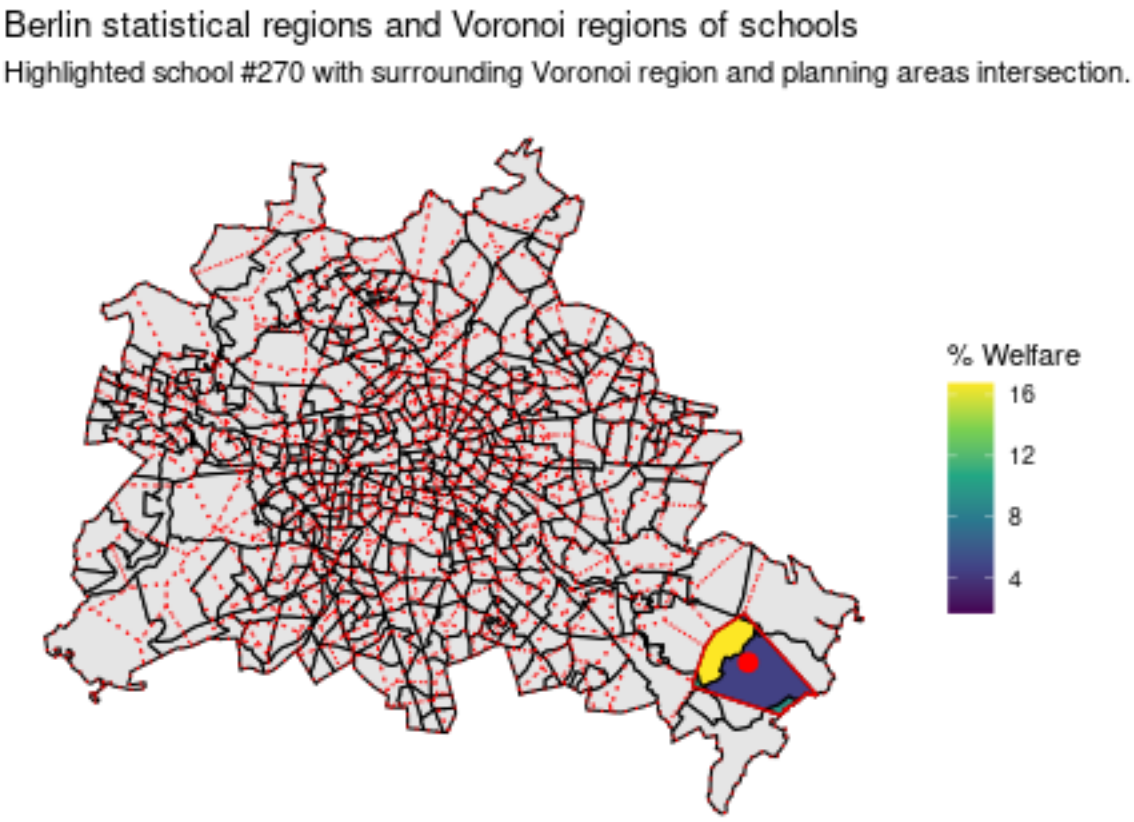
```
(exampleareas <- st_intersection(bln, examplevor))
```

```
## Simple feature collection with 6 features and 10 fields
## Geometry type: GEOMETRY
## Dimension:     XY
## Bounding box:  xmin: 406269.6 ymin: 5805035 xmax: 413198.9 ymax: 5810714
## Projected CRS: ETRS89 / UTM zone 33N
## # A tibble: 6 x 11
##    areaid areaname  welfare welfare_chld schoolid traeger priv_schule_typ name
## *   <int> <chr>       <dbl>        <dbl>    <int> <chr>   <chr>           <chr>
## 1 9031101 Grünau       8.8         10.4       270 oeff    ""              Mügge~
## 2 9031201 Karoline~    2.01         1.73      270 oeff    ""              Mügge~
## 3 9031202 Schmöckw~    6.25         9.47      270 oeff    ""              Mügge~
## 4 9041301 Kietzer ~   12.8         16.8       270 oeff    ""              Mügge~
## 5 9041601 Müggelhe~    3.68         4.54      270 oeff    ""              Mügge~
## 6 9051801 Rahnsdor~    5.89         5.67      270 oeff    ""              Mügge~
## # ... with 3 more variables: ort <chr>, plz <int>, geometry <GEOMETRY [m]>
```

```
{
  plot(exampleareas$geometry)
  plot(exampleschool$geometry, col = 'red', pch = 19, add = TRUE)
}
```

We can put that a little bit into perspective again and display this on the Berlin planning regions map overlayed with the schools' Voronoi regions:

```
ggplot() +
  geom_sf(color = 'black', data = bln) +
  geom_sf(fill = NA, color = 'red', linetype = 'dotted', data = bln_vor) +
  geom_sf(aes(fill = welfare_chld), color = 'black', data = exampleareas) +
  geom_sf(fill = NA, color = 'red', data = examplevor) +
  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 Voronoi regions of schools",
       subtitle = "Highlighted school #270 with surrounding Voronoi region and planning areas intersecti
  theme_minimal()
```



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

```
## Simple feature collection with 6 features and 4 fields
## Geometry type: GEOMETRY
## Dimension:     XY
## Bounding box:  xmin: 406269.6 ymin: 5805035 xmax: 413198.9 ymax: 5810714
## Projected CRS: ETRS89 / UTM zone 33N
##                       areaname welfare_chld              area       weight
## 1                   Müggelheim         4.54 13633749.67 [m^2] 0.6473603007
## 2      Kietzer Feld/Nachtheide        16.75  6117595.30 [m^2] 0.2904768263
## 3 Schmöckwitz/Rauchfangswerder         9.47   757552.06 [m^2] 0.0359702313
## 4                       Grünau        10.39   318930.01 [m^2] 0.0151434954
## 5       Rahnsdorf/Hessenwinkel         5.67   216763.29 [m^2] 0.0102923961
## 6                  Karolinenhof         1.73    15937.56 [m^2] 0.0007567502
##                         geometry
## 1 POLYGON ((406890.4 5806615,...
## 2 POLYGON ((408799.1 5810501,...
## 3 POLYGON ((410293.6 5805317,...
## 4 POLYGON ((406451.3 5807406,...
```

```
## 5 MULTIPOLYGON (((410155.4 58...
## 6 MULTIPOLYGON (((406681.9 58...
```

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 ~8.4% 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(examplepareas$welfare_chld, as.numeric(st_area(examplepareas)))
```

```
## [1] 8.362149
```

```
# former approach: linking the school with its planning area
schools_plan[schools_plan$schoolid == 270, ]$welfare_chld
```

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

We'll next perform these calculations for all schools. First, we find the school inside each Voronoi region using `st_join` as before:

```
school_vor <- st_join(st_as_sf(bln_vor), schools)
head(school_vor)
```

```
## Simple feature collection with 6 features and 6 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 370000.7 ymin: 5805751 xmax: 377946.9 ymax: 5823264
## Projected CRS: ETRS89 / UTM zone 33N
##   schoolid traeger priv_schule_typ
## 1      140    oeff
## 2      155    oeff
## 3      143    oeff
## 4      145    oeff
## 5      161    oeff
## 6      369    priv         Waldorf
##                                        name                        ort
## 1                     Linden-Grundschule             Berlin Spandau
## 2                  Mary-Poppins-Grundschule             Berlin Spandau
## 3                Astrid-Lindgren-Grundschule             Berlin Spandau
## 4                  Grundschule am Ritterfeld             Berlin Spandau
## 5                Conrad-Schule (Grundschule) Berlin Steglitz-Zehlendorf
## 6 Freie Waldorfschule Havelhöhe - Eugen Kolisko             Berlin Spandau
##     plz                       geometry
## 1 13591 POLYGON ((372875.8 5823264,...
## 2 14089 POLYGON ((373855.6 5817213,...
## 3 13591 POLYGON ((373717.9 5820410,...
## 4 14089 POLYGON ((371755 5813750, 3...
## 5 14109 POLYGON ((371780.3 5810871,...
## 6 14089 POLYGON ((373736.3 5814453,...
```

Next we calculate the planning area intersections, their areas and weighted average of the welfare rate for each school's Voronoi region 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' Voronoi region dataset:

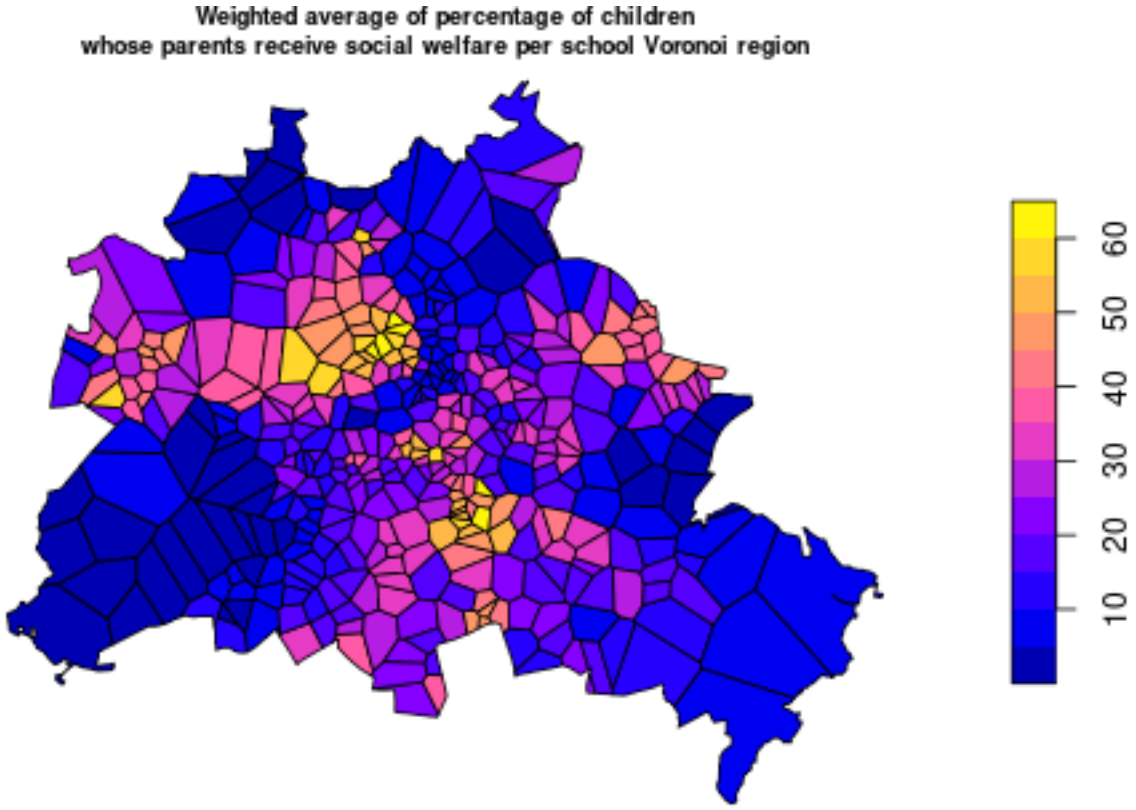
```
school_vor$welfare_chld <- sapply(school_vor$geometry, function(vor) {
  # the Voronoi polygon "vor" loses the CRS during sapply -> set it here again
  vor <- st_sfc(vor, crs = st_crs(bln))
```

```
  areas <- st_intersection(bln, vor)
  weighted.mean(areas$welfare_chld, as.numeric(st_area(areas)))
})

select(school_vor, schoolid, traeger, name, welfare_chld) %>% head()
```

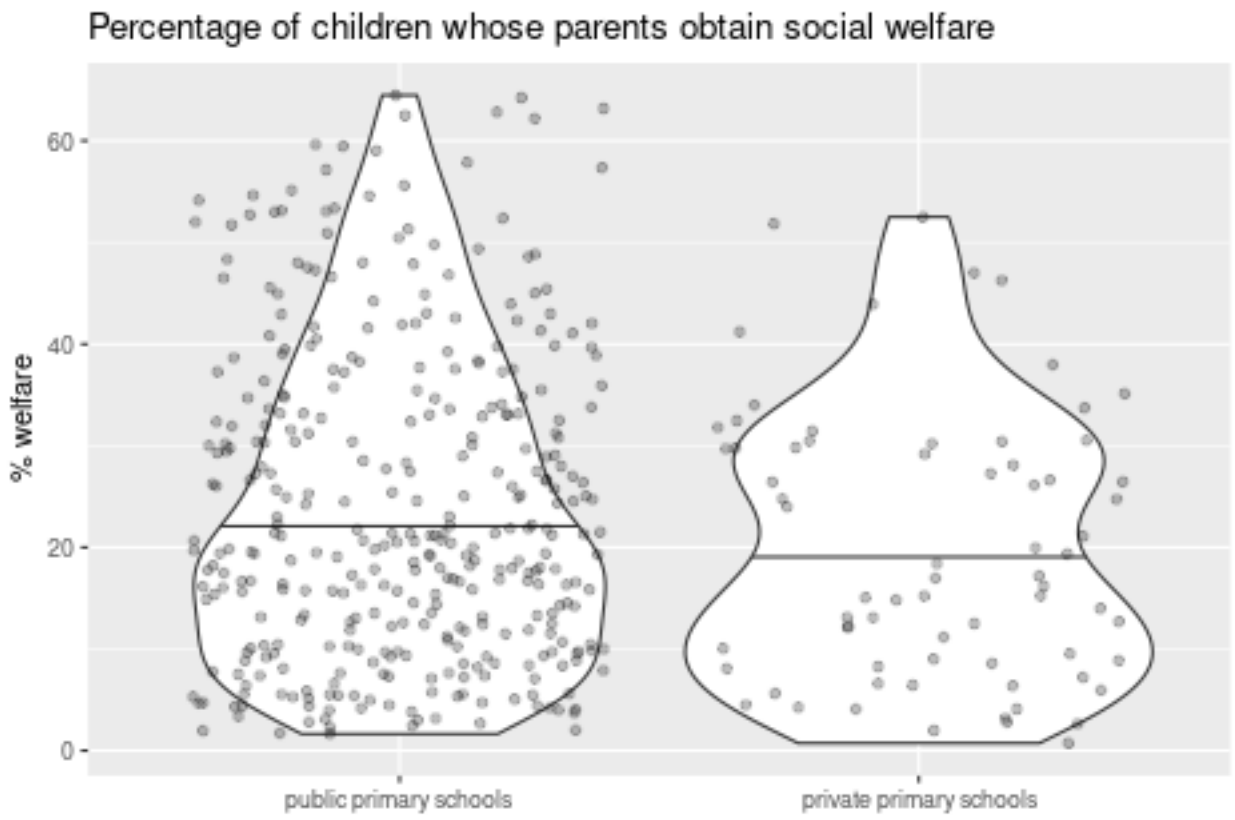
```
## Simple feature collection with 6 features and 4 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 370000.7 ymin: 5805751 xmax: 377946.9 ymax: 5823264
## Projected CRS: ETRS89 / UTM zone 33N
##   schoolid traeger                                        name welfare_chld
## 1      140    oeff                   Linden-Grundschule    17.248205
## 2      155    oeff                 Mary-Poppins-Grundschule     3.720224
## 3      143    oeff               Astrid-Lindgren-Grundschule    42.571484
## 4      145    oeff                 Grundschule am Ritterfeld     3.136074
## 5      161    oeff               Conrad-Schule (Grundschule)     4.300000
## 6      369    priv Freie Waldorfschule Havelhöhe - Eugen Kolisko     4.497095
##                         geometry
## 1 POLYGON ((372875.8 5823264,...
## 2 POLYGON ((373855.6 5817213,...
## 3 POLYGON ((373717.9 5820410,...
## 4 POLYGON ((371755 5813750, 3...
## 5 POLYGON ((371780.3 5810871,...
## 6 POLYGON ((373736.3 5814453,...
```

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



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

```
ggplot(school_vor) +
  geom_violin(aes(x = traeger, y = welfare_chld), draw_quantiles = c(0.5)) +
  geom_jitter(aes(x = traeger, y = welfare_chld), alpha = 0.25) +
  scale_x_discrete(labels = c('oeff' = '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 four instead of six percent.

```
st_drop_geometry(school_vor) %>%
  group_by(traeger) %>%
  summarise(median_welfare_chld = median(welfare_chld))
```

```
## # A tibble: 2 x 2
##   traeger median_welfare_chld
##   <chr>                 <dbl>
## 1 oeff                   20.9
## 2 priv                   17.0
```

Our updated approach led to a difference that is about one third smaller. This difference is not very large because of the very small Voronoi regions 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.

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.