Visual Data Science - Profile & Wrangle

Nikolaus Czernin

Data preparation

For this project we mainly used the data contained in the data yearbooks by Statistik Austria (2019-2024) (Austria 2019-2024). They come in ZIP-files, containing folders for different data domains (education, health, immigration, etc.).

Population data

The `bevölkerung` population data (Statistik Austria, 2024) (Austria 2019-2024), `bevoelkerung/tab_5.1.2_bevoelkerungsst` is a single dataset that encompasses the population data per district in Vienna the early 2000s until 2024. The original file is a CSV that includes a lot of informative meta data in the same sheet, above and below the actual data table, which we prune upon loading. The variable names are renamed to enable intuitive data transformations using the dplyr library. The original data saved each year as a column, which we pivoted to create one long dataset with one district and one year per row.

Hospitals

The `krankenhäuser` hospital data (Statistik Austria, 2019-2024) (Austria 2019-2024), `gesundheit/gesundheit_tab_6.1.2.cs` while coming fmro the same source, only ever contains the data for the year the data set was published, so in order to get timeline data for multiple years, we had to download and wrangle each year's data collection, laod and transform them individually and join them. This was challenging because it was no matter of copying a pasting the code, because the datasets were formatted differently over the years. Generally, wrangling steps included, again, omitting any title and metadata rows above and below the actual data table, changing the variables' names to enable intuitive transformations, filtering out summary rows (e.g. total over all districts), formatting the numeric character columns (replacing commas with dots), and, finally, comining each year's loaded data via `bind_rows()`.

Doctors

The `ärztinnen` data (Statistik Austria, 2019-2024) (Austria 2019-2024), `gesundheit/gesundheit_tab_6.1.4.csv`, which contains the counts of private practice doctors per district, was also loaded via one data set per year and included the same wrangling steps as the `krankenhäuser` data.

Geospatial data

To create choropleths with the data listed above for the Viennese districts, we used a simplified, non-government adaption of public geospatial data by Perlot (2011) (Perlot 2021). This dataset, `geodata`, was intuitive to use and required only minor transformations to create pretty district name displays.

Exploratory Analysis

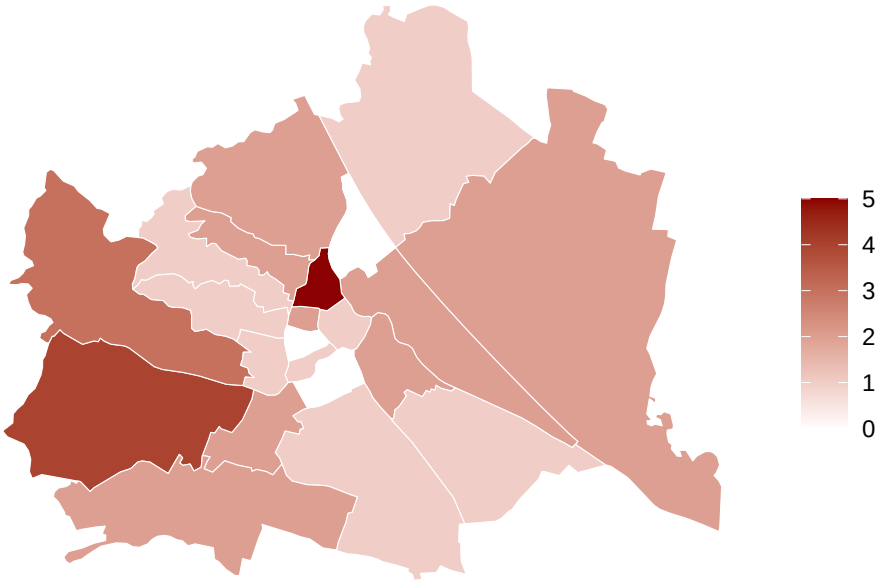
Insights

Insight 1: Hospitals & doctors

As a first step in the exploratory analysis, lets take a look at the absolute numbers of available hospitals and doctors per district.

First, we group the data points in the `krankenhäuser` data by the year and district to count the number of hospitals per year and district. By joining that with the `geodata`, we can create choropleths, effectively drawing the polygons that represent the simplified map shapes of the district, where each polygon's fill color darkness is mapped to the number of hospitals in that district.

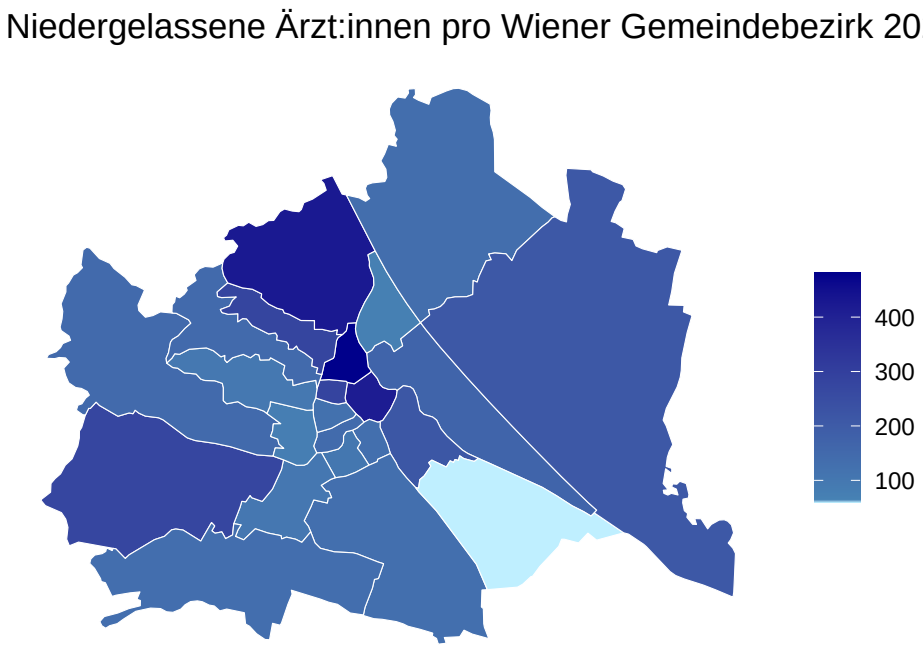
```
year = 2024
krankenhäuser %>%
  filter(Jahr == year) %>%
  group_by(Jahr, Bezirk_Nr) %>%
  count() %>%
  right_join(bezirke_wien) %>%
  mutate(n=n %>% replace_na(0)) %>%
  # print() %>%
  ggplot(aes(geometry=geometry)) +
    geom_sf(aes(fill = n), color = "white") +
    scale_fill_viridis_c(option = "plasma") +
    labs(
      title = paste("Anzahl Krankenanstalten pro Wiener Gemeindebezirk", year),
      fill = ""
         ) +
    theme_minimal() +
   # scale_fill_gradient(low = "white", high = "darkred") +
    scale_fill_gradientn(colors = c("white", "snow", "darkred"),
                         values = scales::rescale(c(0, 1, 80)),) +
   theme(
      axis.text = element_blank(),
      panel.grid = element_blank(),
      aspect.ratio = .8 # Plot Seitenverhältnisse ändern
      )
```



The red choropleth shows, that the districts Margareten, Neubau and Brigittenau appear not to have any hospitals in them. Alsergrund takes the lead with the most hospitals (5).

While those three districts feature no stationary doctors, there is a vast supply of private practice practitioners outside of hospitals. To visualize this, we perform the same aggregation and join as before on the `ärztinnen` data set and create the same type of choropleth with a blue hue.

```
year = 2024
ärztinnen %>%
  filter(Jahr == year) %>%
  group_by(Jahr, Bezirk_Nr) %>%
  summarise(n = sum(FachärztInnen, na.rm=T)) %>%
  right_join(bezirke_wien) %>%
  mutate(n=n %>% replace_na(0)) %>%
  # print() %>%
  ggplot(aes(geometry=geometry)) +
    geom_sf(aes(fill = n), color = "white") +
    scale_fill_viridis_c(option = "plasma") +
    labs(
      title = paste("Niedergelassene Ärzt:innen pro Wiener Gemeindebezirk", year),
      fill = ""
         ) +
    theme_minimal() +
  scale_fill_gradientn(colors = c("lightblue1", "steelblue",  "blue4"),
                       values = scales::rescale(c(0, 1, 80)),) +
   theme(
      axis.text = element_blank(),
      panel.grid = element_blank(),
      aspect.ratio = .8 # Plot Seitenverhältnisse ändern
      )
```



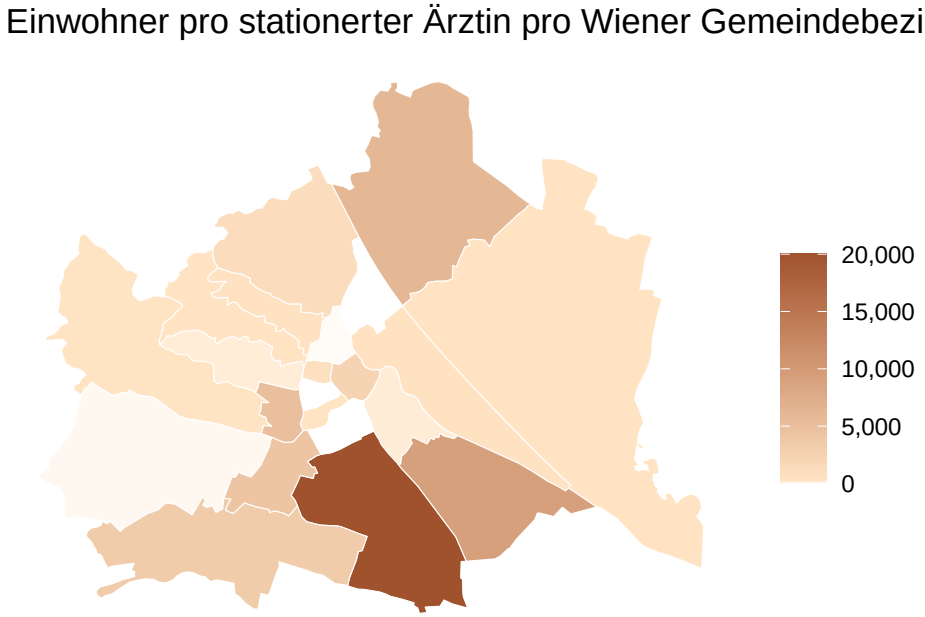
This choropleth shows that Döbling, Alsergrund and Innere Stadt, which contained a single hospital in the previous plot, take the lead with over 400 private practice doctors each, whereas only 60 settled in Simmering.

Insight 2: Population per hospital

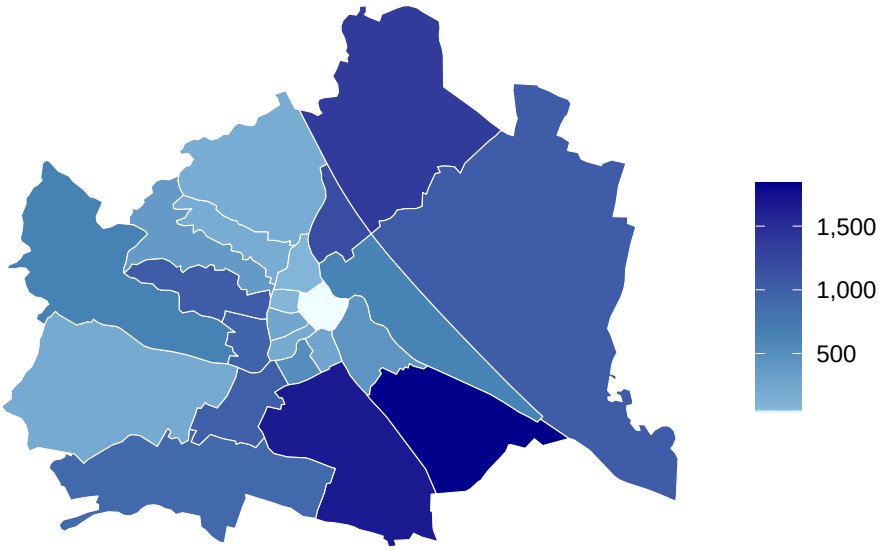
By joining the `bevölkerung` data set with the `geodata`, we can create another choropleth, visualizing the population for a certain year per district.

While no helpful insight can be gained here yet, we can join the resulting data set with the `krankenhäuser` and `ärztinnen` data sets to find out the number of residents per doctor stationed in a hospital and private practice doctor for each district. This shows a clearer picture of where doctors have a larger pool of possible patients.

```
# year = 2024
krankenhäuser %>%
  group_by(Jahr, Bezirk_Nr) %>%
  summarise(Ärztinnen.und.Ärzte = sum(`Ärztinnen.und.Ärzte`, na.rm=T)) %>%
  left_join(bevölkerung) %>%
  mutate(n = Bevölkerung / Ärztinnen.und.Ärzte) %>%
  filter(Jahr == year) %>%
  right_join(bezirke_wien) %>%
  mutate(n=n %>% replace_na(0)) %>%
  ggplot(aes(geometry=geometry)) +
    geom_sf(aes(fill = n), color = "white") +
    scale_fill_viridis_c(option = "plasma") +
    labs(
      title = paste("Einwohner pro stationerter Ärztin pro Wiener Gemeindebezirk", year),
      fill = ""
         ) +
    theme_minimal() +
  scale_fill_gradientn(colors = c("white", "bisque1", "sienna"),
                       labels = label_comma(),
                       values = scales::rescale(c(0, 1, 80)),) +
   theme(
      axis.text = element_blank(),
      panel.grid = element_blank(),
      aspect.ratio = .8 # Plot Seitenverhältnisse ändern
      )
```



```r
# year = 2024
ärztinnen %>%
  filter(Jahr == year) %>%
  left_join(bevölkerung, by=c("Bezirk_Nr", "Jahr")) %>%
  mutate(n = Bevölkerung / `FachärztInnen`) %>%
  right_join(bezirke_wien) %>%
  mutate(n=n %>% replace_na(0)) %>%
  ggplot(aes(geometry=geometry)) +
    geom_sf(aes(fill = n), color = "white") +
    scale_fill_viridis_c(option = "plasma") +
    labs(
      title = paste("Bevölkerung pro niedergelassener Fachärzt:innen\npro Wiener Gemeindebezirk", year)
      fill = ""
         ) +
    theme_minimal() +
  scale_fill_gradientn(colors = c("azure1", "lightblue1", "steelblue", "blue4"),
                       labels = label_comma(),
                       values = scales::rescale(c(0, 1, 80)),) +
   theme(
      axis.text = element_blank(),
      panel.grid = element_blank(),
      aspect.ratio = .8 # Plot Seitenverhältnisse ändern
      )
```



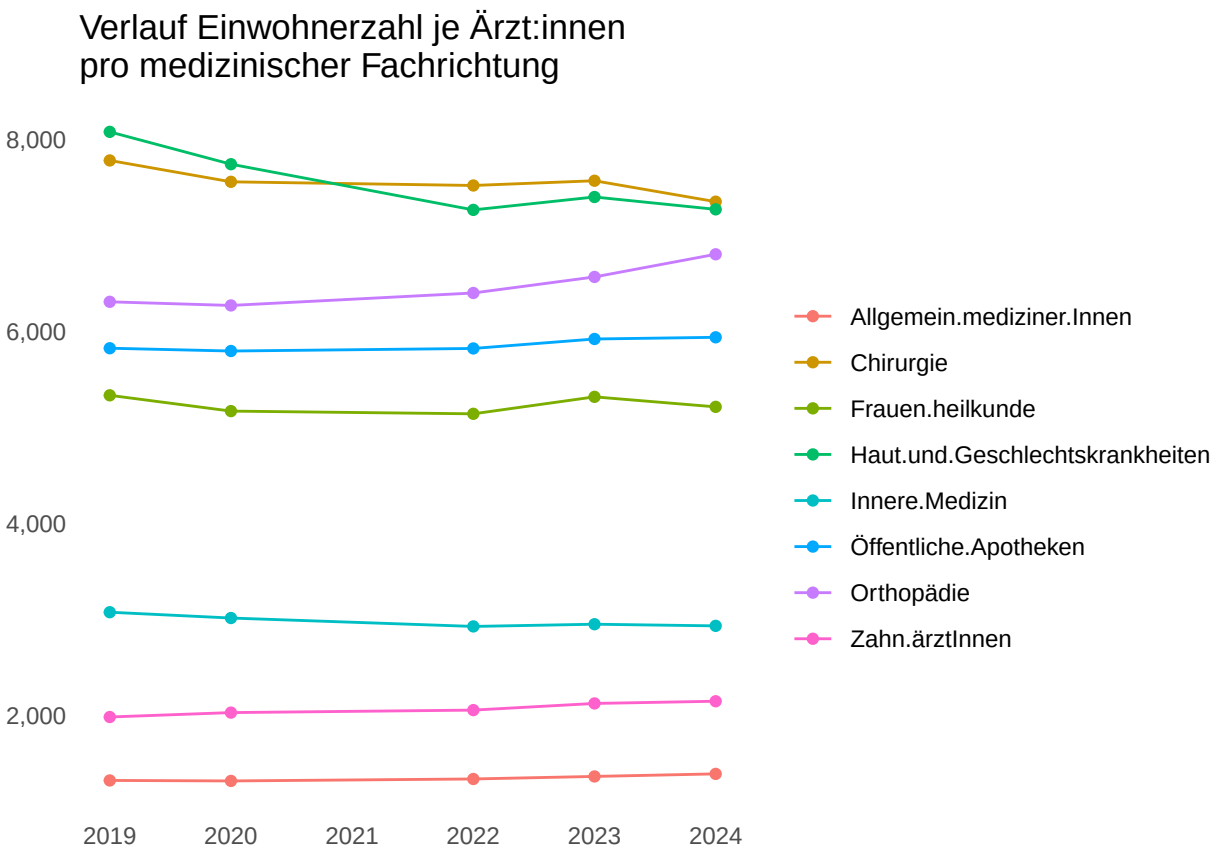
Insight 3: Scarcity of medical specialties

The previous insights focused on all kinds of private practitioners, however, some types of specialties are more common and some more scarce than others.

By grouping the `ärztinnen` data per year and summing the doctor counts, we get the number of specialized practitioners per year and can investigate, which specialists are more scarce than others. By, again, joining it with the `bevölkerung` data, we can get the relative population per specialist for each year.

```
bevölkerung_gesamt <- bevölkerung %>%
  group_by(Jahr) %>%
  summarise(Bevölkerung = sum(Bevölkerung, na.rm=T))

ärztinnen %>%
  select(-FachärztInnen) %>%
  pivot_longer(-c("Bezirk", "Bezirk_Nr", "Jahr"), names_to = "Fachrichtung", values_to = "n") %>%
  group_by(Fachrichtung, Jahr) %>%
  summarise(n = sum(n, na.rm=T)) %>%
  left_join(bevölkerung_gesamt) %>%
  mutate(n_rel = Bevölkerung / n) %>%
  ggplot(aes(x=Jahr, y=n_rel, color=Fachrichtung)) +
    geom_point() +
    geom_line() +
    scale_fill_viridis_c(option = "plasma") +
    labs(
      title = "Verlauf Einwohnerzahl je Ärzt:innen \npro medizinischer Fachrichtung",
      color = ""
         ) +
    theme_minimal() +
    # ylim(0, 8200) +
    scale_y_continuous(labels=label_comma()) +
    theme(
      axis.title = element_blank(),
      panel.grid = element_blank(),
    )
```



The line plot above shows that general practitioners are by far the most common, followed by dentists and internal specialists. The most scarce are surgeons and dermatologists, followed by orthopedists and obstetricians. Pharmacies are shown up there in the top scarce groups too, though they serve a different purpose altogether.

References

Austria, Statistik. 2019-2024. “Statistisches Jahrbuch Der Stadt Wien.” Statistik Austria. https://www.data.gv.at/katalog/datasets/a6b357c5-03c9-4743-9d3d-b6624294e7b9.

Perlot, Flooh. 2021. “GeoJSON/TopoJSON Austria (2016–2021).” GitHub. https://github.com/ginseng666/GeoJSON-TopoJSON-Austria.