Problem Set 3

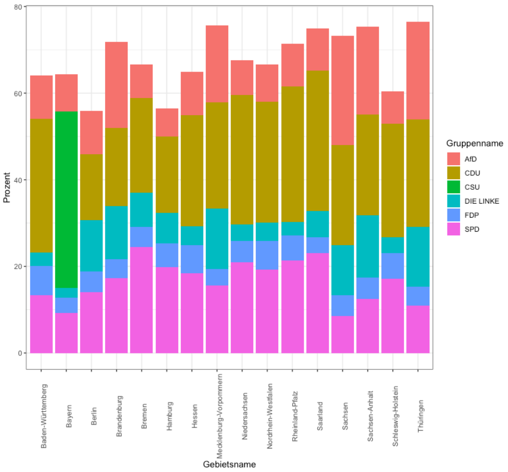
The third problem set focuses on visualization. Your solution should be composed of a well-structured R script which should provide the designated functions. Besides the functions, the code should be directly runnable or at least sufficiently well documented (working directory, path settings) to be executed.

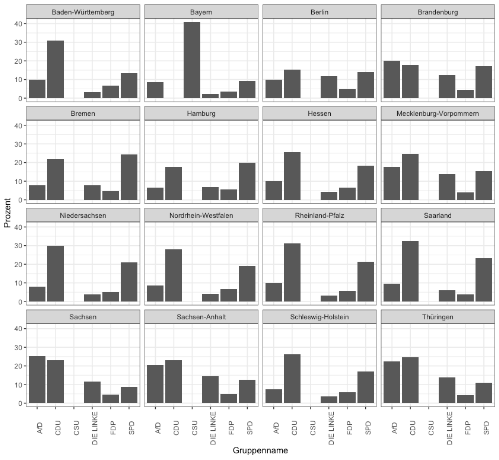
This problem set is **due on July 7th by 18.00** through the Wuecampus upload functionality.

1. The ninth election to the European Parliament in Germany was held on 26 May 2019, electing members of the national Germany constituency to the European Parliament. Your task is to visualize the results of the election which can be downloaded here: https://www.bundeswahlleiter.de/dam/jcr/5441f564-1f29-4971-9ae2-b8860c1724d1/ew19_kerg2.csv

    a. Load the data directly from the link using the read_csv2 function. Note that you have to skip the first 9 lines to receive the data frame correctly. Subsequently, filter the data set to data on a state ("Bundesland") and party level. Note that you have to handle Unicode formatting. (Hint: the final data set will have 640 rows and 18 columns)

    b. First, visualize the share of votes the largest parties received in each state ("CDU", "CSU", "SPD", "AfD", "DIE LINKE", "GRÜNE", "FDP"). Create one figure aggregating all the information in one plot (such as Figure 1) and one figure using facet_wrap (Figure 2).



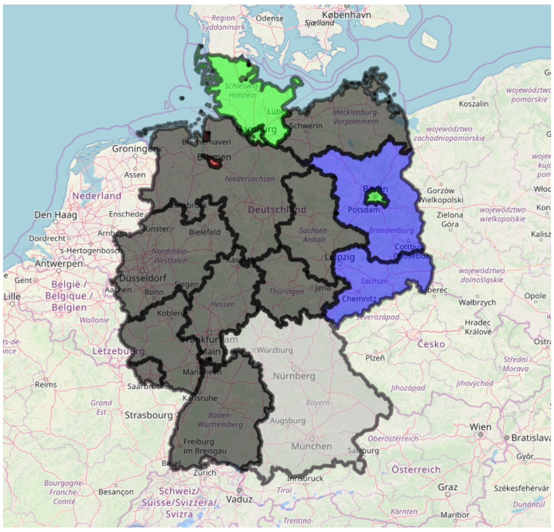


    c. Obviously, a party's performance largely depends on the state. Use a suited type of visualization to show the distribution of each of the 5 parties' performance across all states.

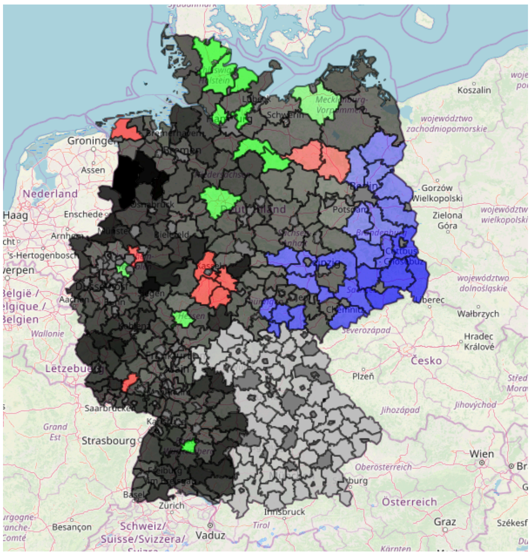
    d. Your next task is to leverage the leaflet package in order to visualize the election winners in each state. Therefore, you have to:

        i. Aggregate the data frame to find the winners of each state
        ii. Load the (level 1) SpatialPolygonsDataFrames (sp) for Germany
        iii. Join both data frames to map bounding boxes and winners
        iv. Define a color palette using the colorFactor function.
        v. Create the map using leaflet

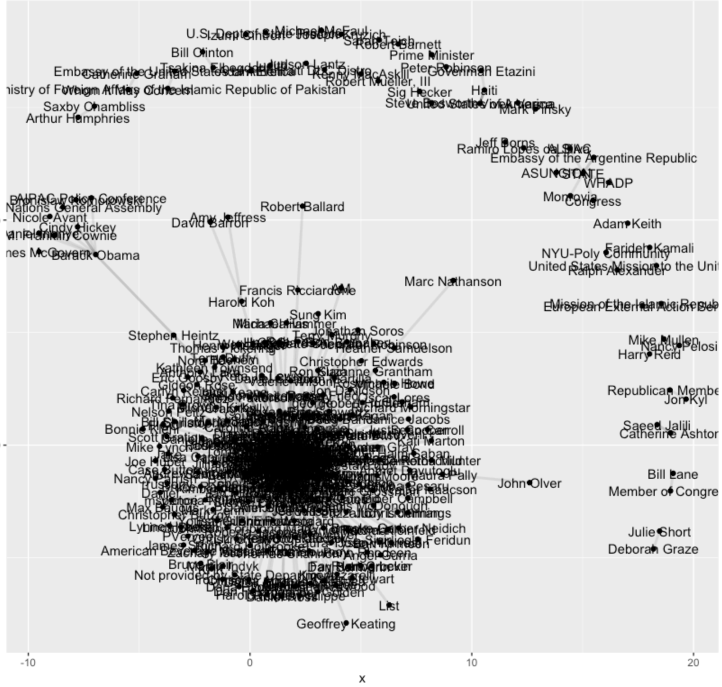
The final figure should look similar to the following:



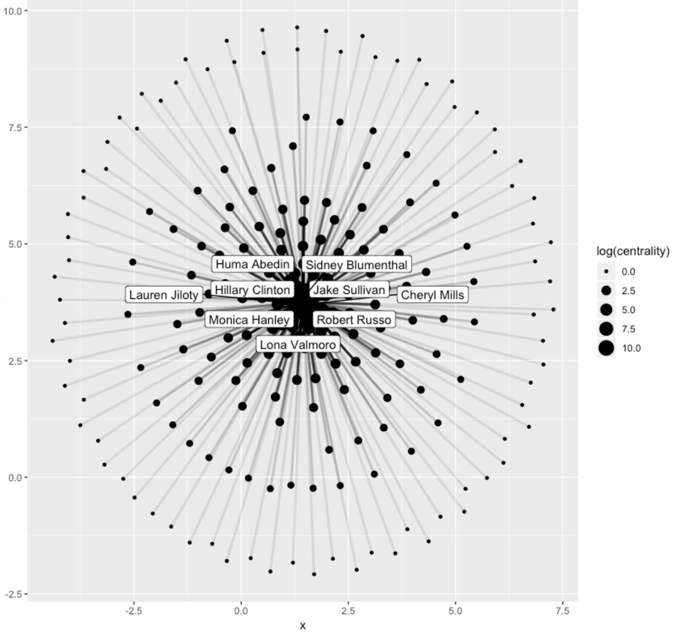
e. Repeat the steps above to create a similar map on a constituency level. Additionally, map the opacity of the constituencies to the share of votes of the winning party. The final result should look like the following:



2. Your second task is to analyze the infamous Hillary Clinton email data set. The code to load and clean the data is already in the submission script.
   a. Transform the data set into a table graph object and visualize the email network as following:



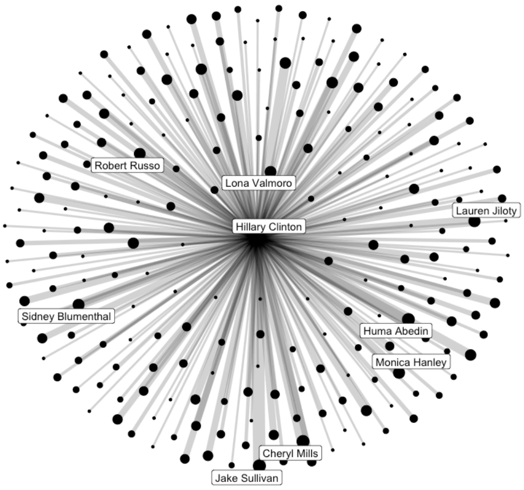
   b. Try to explain the small sub-graphs that are not connected to Hillary Clinton and clean the data set to remove them. Visualize the network again.
   c. Next, we want to improve the graph visualization by taking the centrality of the nodes into account. Specifically, you should size the nodes according to their logarithmic centrality degree. Additionally, add labels only to nodes with a centrality higher than 500. The resulting figure should look like this:



   d. Finally, we want to add additional information into our figure. More specifically, we want the edge-width to show how often the nodes interact with each other.

i. Leverage your data wrangling skills to count the number of interactions between each pair of nodes
ii. Join the new data frame to the email data frame
iii. Remove duplicate edges
iv. Plot the graph and remove all legends

The final plot should look like this:


Robert Russo
Lona Valmoro
Lauren Jiloty
Hillary Clinton
Sidney Blumenthal
Huma Abedin
Monica Hanley
Cheryl Mills
Jake Sullivan