Visualizing Regression Results with dotwhisker: Homework

Yeiyoung Choo

Spring 2023

For homework, we will practice how to generate dot-and-whisker plots with regression results stored in tidy dataframes. This is the second approach we learned in class. Throughout this exercise, we will use tidy dataframes as input for `dwplot()`.

Load Packages

```
library(tidyverse)
library(dotwhisker)
library(broom)  #for tidying results
```

Load Data

Load the RDS file you saved in groupwork (`reg_asklaksen2010.rds`).

```
as2010reg <- readRDS("reg_aslaksen2010.rds")
```

1 Running Linear Regression

Referring to your group work, run all three regression models (Models 1-3) and store them as objects (`n1`, `n2` and `n3`). Again, we use `pr_lead` as the outcome variable. Check regression results using `summary()`.

2 Tidying Regression Results

Using `tidy()` in *broom* package, store your regression results as tidy dataframes. Let's call them `n1df`, `n2df` and `n3df`.

3 Generating a Dot-and-Whisker Plot

a) Generate a dot-and-whisker plot for Model 1, using the tidy dataframe as input.

b) Display regression results from all three models in one plot. Make sure to add a column called `model` to label your tidy dataframes as `Model 1`, `Model 2`, and `Model 3`. When merged, this column will help identify each model. Merge the dataframes into one, calling it `models`. Use it as input.

4 Customizing the Plot

a) Generate a dot-and-whisker plot that shows coefficient estimates from all models, for the following variables: `oilshare`, `lpop`, `educ`. Starting again with `tidy()` and regression objects, prep your tidy dataframes first. Merge the dataframes, and then generate the plot.

b) Referring to the PNG file you saved in group work, customize your plot to create a polished version of it. Relabel predictors as follows: Oil Share, Population (logged), Education.

5 Saving Your Work

Save your plot as a PNG file using `ggsave`, setting dimensions as needed.