SPEC REU R Resources: Visualizing Regression Results with dot-and-whisker Plots – Homework

Yeiyoung Choo and Claudia Salas Gimenez

Summer 2024

In this homework, we will practice generating dot-and-whisker plots using regression results from tidy dataframes. These plots will visually represent the coefficient estimates and confidence intervals from our regression models.

As the final module of our online training, this assignment also aims to refine your skills in data management, regression analysis, and data visualization, equipping you with the tools needed for publication-ready quantitative analysis.

Initial Setup

Begin by setting your working directory to the location of your data files, and load the necessary libraries and dataset. We'll be working with the `.rds` file you saved in the groupwork assignment (`reg_aslaksen2010.rds`).

Note that for this module will be using the `dotwhisker` package, and as of April 2024, we need to install its dependencies `prediction` and `margins` from GitHub in order to load the `dotwhisker` package.

```r
# Set working directory
#setwd("YourFolderPath")

# Load required libraries
library(tidyverse)
library(ggplot2)
library(broom)

# As of April 2024, "dotwhisker" is no longer on CRAN and must be installed with its
# dependencies "prediction" and "margins" from GitHub:
# remotes::install_github("leeper/prediction", force = TRUE)
# remotes::install_github("leeper/margins", force = TRUE)
# remotes::install_github("fsolt/dotwhisker", force = TRUE)

library(dotwhisker)

# Load the data
as2010reg <- readRDS("reg_aslaksen2010.rds")
```

Exercise 1: Run Linear Regressions

Let's run the regression models previously specified in the groupwork assignment and store them as objects (`n1`, `n2` and `n3`).

- **Model 1 (n1):** This model predicts political rights in the upcoming year (pr_lead) using the predictors pr (political rights) and oilshare (share of oil in income).
- **Model 2 (n2):** Expanding on Model n1, this multiple linear regression includes additional predictors: lrgdppc (logged GDP per capita), lpop (logged population), and educ (education level).
- **Model 3 (n3):** Further extending Model n2, this model also incorporates open (openness to trade) to assess a broader range of factors influencing political rights.

Check the results for each model using the summary() function.

```r
# Run regression models
n1 <- lm(pr_lead ~ pr + oilshare, data = as2010reg)
n2 <- lm(pr_lead ~ pr + oilshare + lrgdppc + lpop + educ, data = as2010reg)
n3 <- lm(pr_lead ~ pr + oilshare + lrgdppc + lpop + educ + open, data = as2010reg)

# Display summary statistics
summary(n1)
```

```
##
## Call:
## lm(formula = pr_lead ~ pr + oilshare, data = as2010reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98518 -0.01966 -0.00455  0.01476  0.81061
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0265457  0.0027657   9.598  < 2e-16 ***
## pr           0.9586945  0.0042674 224.657  < 2e-16 ***
## oilshare    -0.0003052  0.0001051  -2.903  0.00371 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1006 on 4394 degrees of freedom
##   (1238 observations deleted due to missingness)
## Multiple R-squared:  0.9243, Adjusted R-squared:  0.9243
## F-statistic: 2.684e+04 on 2 and 4394 DF,  p-value: < 2.2e-16
```

```r
summary(n2)
```

```
##
## Call:
## lm(formula = pr_lead ~ pr + oilshare + lrgdppc + lpop + educ,
##     data = as2010reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97566 -0.02452 -0.00315  0.01593  0.78783
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.652e-02  3.105e-02  -0.854 0.392980
## pr           9.102e-01  7.147e-03 127.359  < 2e-16 ***
## oilshare    -5.976e-04  1.685e-04  -3.546 0.000397 ***
## lrgdppc      7.761e-03  3.135e-03   2.475 0.013360 *
## lpop        -8.157e-05  1.242e-03  -0.066 0.947655
## educ         4.612e-03  1.128e-03   4.089 4.43e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1025 on 3144 degrees of freedom
##   (2485 observations deleted due to missingness)
## Multiple R-squared:  0.9187, Adjusted R-squared:  0.9185
## F-statistic:  7102 on 5 and 3144 DF,  p-value: < 2.2e-16
```

```r
summary(n3)
```

```
##
## Call:
## lm(formula = pr_lead ~ pr + oilshare + lrgdppc + lpop + educ +
##     open, data = as2010reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97980 -0.02469 -0.00333  0.01592  0.78150
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.290e-03  3.348e-02   0.098 0.921727
## pr           9.076e-01  7.223e-03 125.648  < 2e-16 ***
## oilshare    -5.974e-04  1.684e-04  -3.547 0.000395 ***
## lrgdppc      8.368e-03  3.143e-03   2.662 0.007804 **
## lpop        -1.754e-03  1.429e-03  -1.228 0.219599
## educ         4.818e-03  1.130e-03   4.263 2.08e-05 ***
## open        -1.119e-04  4.727e-05  -2.366 0.018030 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1024 on 3143 degrees of freedom
##   (2485 observations deleted due to missingness)
## Multiple R-squared:  0.9188, Adjusted R-squared:  0.9186
## F-statistic:  5928 on 6 and 3143 DF,  p-value: < 2.2e-16
```

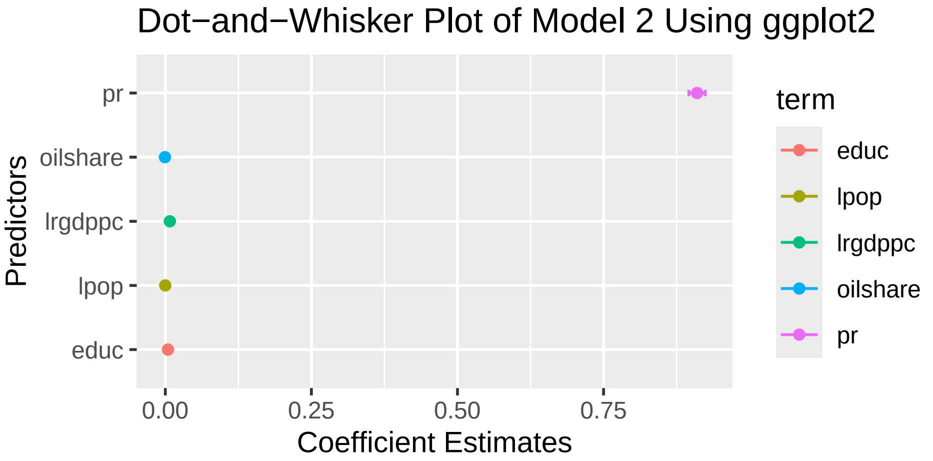
Exercise 2: Generate a Dot-and-Whisker Plot Using `ggplot2`

Use object `n2` to create a dot-and-whisker plot using the `ggplot2` package. Before you start plotting, make sure to create a dataframe to store the regression results and calculate the 95% confidence interval for each coefficient.

```r
# Display regression results and save them as 'summary_n2'
summary_n2 <- summary(n2)

# Create a dataframe for plotting based on the results stored in 'summary_n2'
results_df_n2 <- data.frame(
  term = rownames(summary_n2$coefficients)[-1],
  estimate = summary_n2$coefficients[-1, "Estimate"],
  std.error = summary_n2$coefficients[-1, "Std. Error"],
  conf.low = summary_n2$coefficients[-1, "Estimate"] -
    1.96 * summary_n2$coefficients[-1, "Std. Error"],
  ## Calculate the lower bound of the 95% confidence interval for each coefficient.
  conf.high = summary_n2$coefficients[-1, "Estimate"] +
    1.96 * summary_n2$coefficients[-1, "Std. Error"])
  ## Calculate the upper bound of the 95% confidence interval for each coefficient.

# Create a dot-and-whisker plot using ggplot2
ggplot(results_df_n2, aes(x = estimate, y = term, color = term)) +
  geom_point() +
  geom_errorbar(aes(xmin = conf.low, xmax = conf.high), width = 0.1) +
  ## Sets the horizontal span of the error bars based on the confidence intervals
  ## calculated, showing the estimate's uncertainty.
  labs(title = "Dot-and-Whisker Plot of Model 2 Using ggplot2",
       x = "Coefficient Estimates",
       y = "Predictors")
```

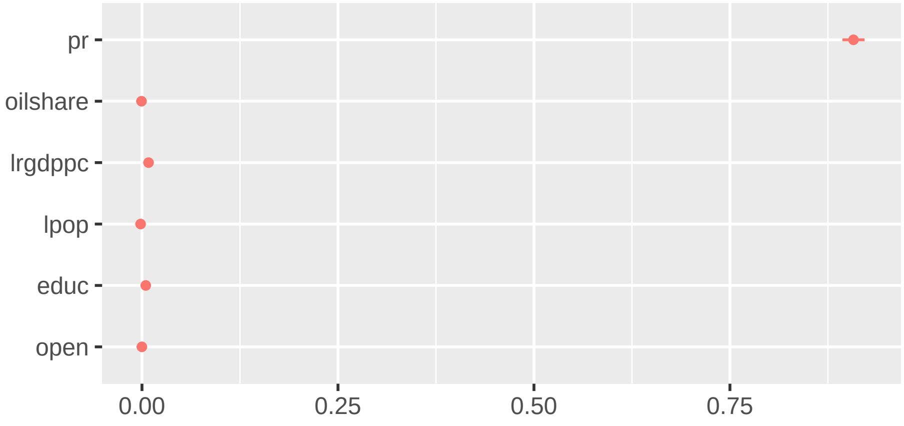


Exercise 3: Generate a Dot-and-Whisker Plot Using `dwplot()`

Now, create of a dot-and-whisker plot using the `dotwhisker` package with `n3` as input. Remember to tidy the dataframe before generating the dot-and-whisker plot.

```
# Convert regression results into tidy dataframe
n3df <- tidy(n3)

# Create a dot-and-whisker plot using dotwhisker package
dwplot(n3df)
```



Exercise 4: Create a Dot-and-Whisker Plot for Three Models with `dwplot()`

For this last exercise, create a single dot-and-whisker plot to visualize results from multiple regression models (`n1`, `n2` and `n3`), specifically focusing on the variables `oilshare`, `lpop`, and `educ`. Remember to tidy up the models `n1`, `n2`, and `n3`, filter for the variables `oilshare`, `lpop`, and `educ`, and label each model with a new column called `Model (model number)` before combining the dataframes with `rbind()`. Then, combine these dataframes using `rbind()`. Also, customize the plot to generate a clear and visually appealing graph. Save your final plot as a `.png` file using `ggsave()`.

Helpful Hint: The `ggsave()` structure is: `ggsave("your filepath/your filename.png"...)`

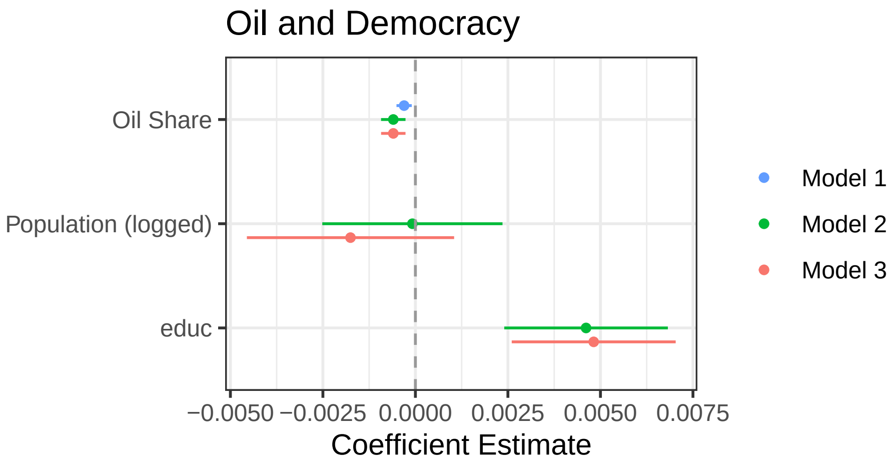
```r
# Prepare the tidy dataframe for Model 1
n1df <- tidy(n1) %>%
        filter(term == "oilshare") %>%
        mutate(model = "Model 1")

# Prepare the tidy dataframe for Model 2
n2df <- tidy(n2) %>%
        filter(term == "oilshare" | term == "lpop" | term == "educ") %>%
        mutate(model = "Model 2")

# Prepare the tidy dataframe for Model 3
n3df <- tidy(n3) %>%
        filter(term == "oilshare" | term == "lpop" | term == "educ") %>%
        mutate(model = "Model 3")

# Combine the prepared dataframes
models <- rbind(n1df, n2df, n3df)

# Customizing the plot
dwplot(models,
       model_order = c("Model 1","Model 2","Model 3"),
       # Specify the order of models in the plot
       vars_order = c("oilshare","lpop","educ")) %>%
       # Define the order of variables to be displayed in the plot
  relabel_predictors(c(oilshare = "Oil Share",
                       lpop = "Population (logged)",
                       open = "Education")) +
                       # Rename the variables in the plot for better readability
  theme_bw() +
  # Applies a minimalistic black and white theme
  xlab("Coefficient Estimate") +
  ylab("") +
  geom_vline(xintercept = 0,
             colour = "grey60",
             linetype = 2) +
  # Adjust visual layout
  ggtitle("Oil and Democracy") +
  # Set the title of the plot
  theme(legend.title=element_blank())
```



```r
# Save the plot
ggsave("DotWhiskerPlotModel.png", width = 10, height = 8)
```