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

Yeiyoung Choo and Claudia Salas Gimenez

March 2025

Welcome to the final module on Data Visualization! So far, we've focused on running regression analyses and displaying the results in professional-quality tables. However, a crucial next step in the research process is visualizing the regression results. While simple plots can show trends in the data, they often fail to capture the uncertainty surrounding the estimates. Dot-and-whisker plots, by displaying both point estimates and their 95% confidence intervals, offer a clear representation of effect sizes and precision.

For this assignment, we'll plot dot-and-whisker plots using the `ggcoefstats` function from the `ggstatsplot` package to visualize regression results exploring how post-conflict justice institutions—specifically, truth commissions—affect foreign direct investment (FDI) inflows. Additionally, we'll show you how to extract regression results into a tidy dataframe using the `broom` package, which prepares the data for any personalized visualization you might need beyond dot-and-whisker plots.

Initial Setup

Start by setting your working directory, loading the required libraries, and importing the al2012.rds dataset. For reference, this dataset was introduced in the research paper Appel & Loyle (2012) "The Economic Benefits of Justice: Post-conflict justice and foreign direct investment".

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

# Load required libraries
library(tidyverse)
library(ggplot2)
library(broom)
library(ggstatsplot)
## Install ggcoefstats from CRAN if not already installed

# Load the data
al2012 <- readRDS("al2012.rds")
```

Exploring the Relationship Between Post-Conflict Justice Institutions and FDI Inflows

In this assignment, we'll explore how post-conflict justice institutions—specifically, truth commissions—affect foreign direct investment (FDI) inflows. The idea is that by ensuring justice after a conflict, these institutions can help rebuild trust and attract foreign investments. Therefore, our prediction is that countries that implement truth commissions will experience higher FDI inflows compared to those that do not.

To test our prediction, we'll run a regression analysis and then visualize the results using dot-and-whisker

plots. This type of plot is especially useful because it displays both the estimated effect (the dot) and the uncertainty around that estimate (the whiskers). If you're new to statistics, think of the confidence interval (95% is the default) as the range in which we believe the true effect will fall 95% of the time if the study were repeated. Remember, this course is designed as a math-free introduction to complement, not replace, a thorough econometrics course.

Estimating Multivariate Regression Model

Let's begin by estimating the effect of post-conflict justice institutions on FDI inflows using a multivariate regression model. In this model, the dependent variable is FDI inflows (`fdiflow`) and our key predictor is `truthvictim`, a binary indicator that equals 1 when a country implements a post-conflict justice institution and 0 otherwise. We'll also include additional control variables such as `damage`, `duration`, `peaceagr`, and `victory` to account for different features of the conflicts.

```
# Estimate multivariate regression model
m1 <-lm(fdiflow ~ truthvictim + damage + duration + peaceagr + victory,
        data = al2012)

# Display regression results
summary(m1)
```

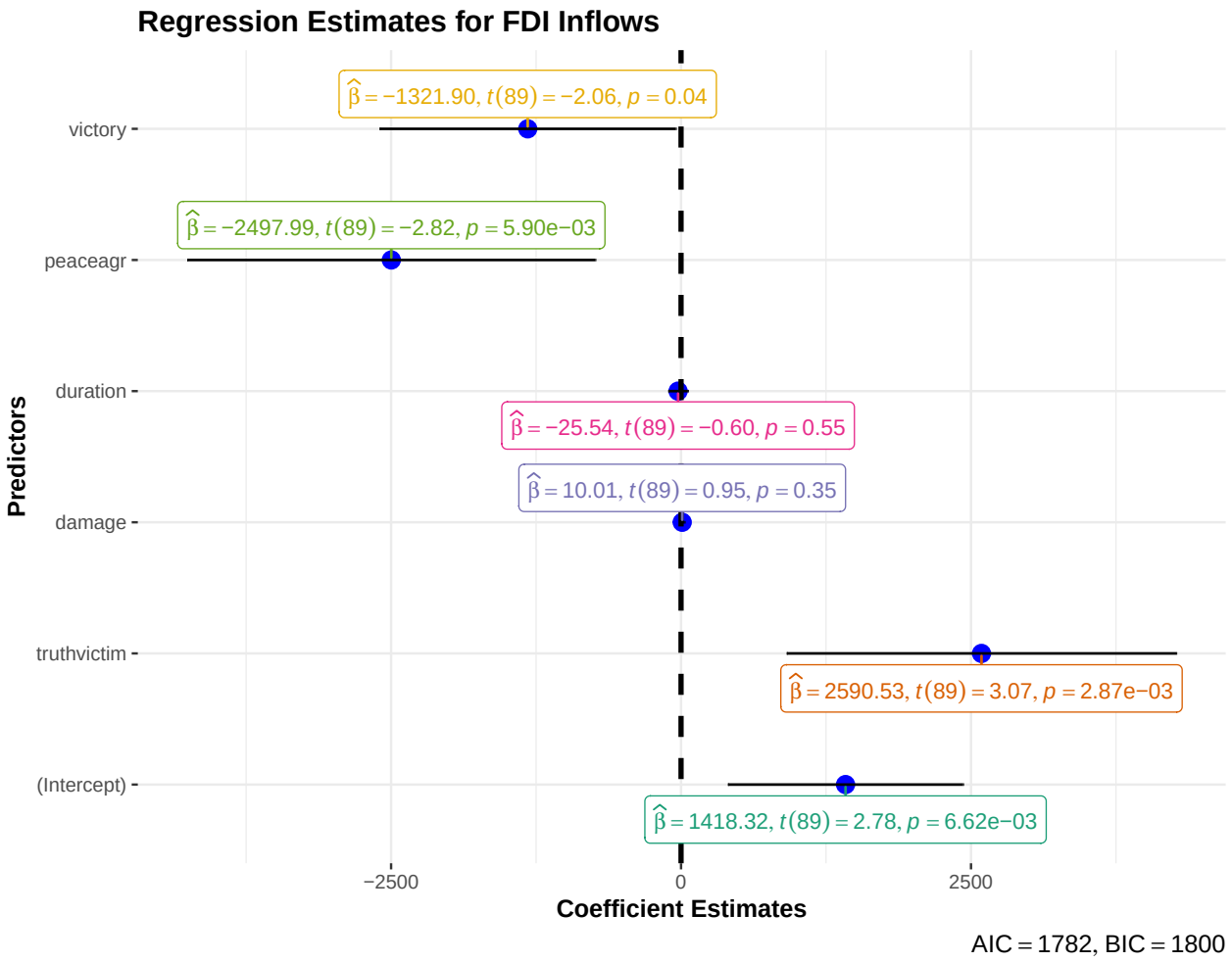
```
##
## Call:
## lm(formula = fdiflow ~ truthvictim + damage + duration + peaceagr +
##     victory, data = al2012)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3907.0 -1163.3  -166.3   270.7 20834.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1418.32     510.04   2.781  0.00662 **
## truthvictim  2590.53     844.96   3.066  0.00287 **
## damage         10.01      10.59   0.946  0.34685
## duration      -25.54      42.52  -0.601  0.54959
## peaceagr    -2497.99     885.38  -2.821  0.00590 **
## victory     -1321.90     642.40  -2.058  0.04254 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2753 on 89 degrees of freedom
## Multiple R-squared:  0.1529, Adjusted R-squared:  0.1053
## F-statistic: 3.213 on 5 and 89 DF,  p-value: 0.01031
```

Visualizing Regression Results with ggcoefstats

To visualize these results, we'll use the `ggcoefstats` function to create a dot-and-whisker plot. This function converts your regression results into a tidy dataframe with key information—such as coefficient estimates, standard errors, and confidence intervals—and then plots each estimate as a dot with whiskers. The result is a `ggplot2` object that you can easily customize with additional `ggplot2` functions.

```
# Generate a dot-and-whisker plot using ggcoefstats
ggcoefstats(m1,
            title = "Regression Estimates for FDI Inflows",
```

```
        xlab = "Coefficient Estimates",
        ylab = "Predictors")
```



Upgrading the Dot-and-Whisker Plot

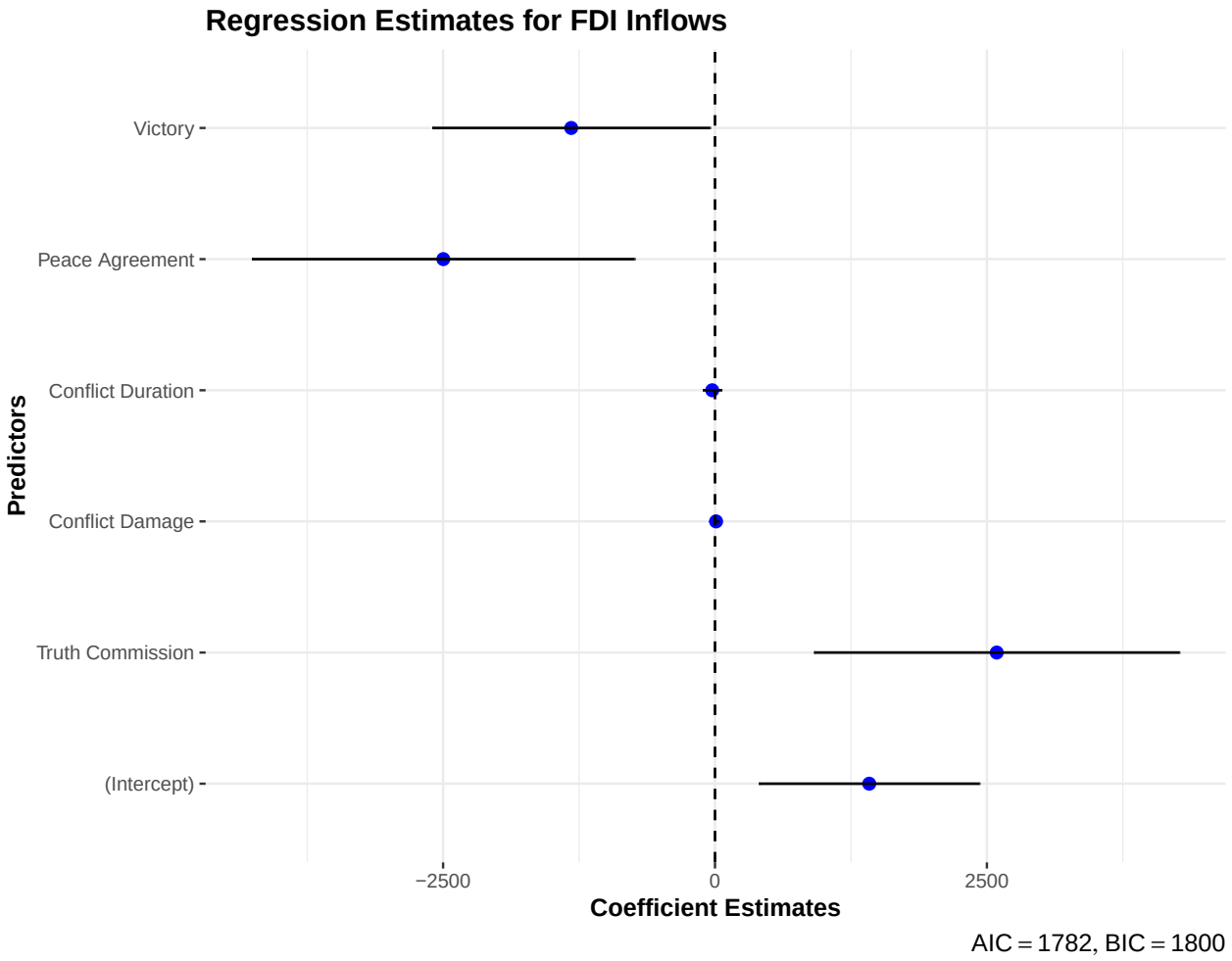
The resulting plot shows each predictor's estimated effect on FDI inflows along with the uncertainty around these estimates. However, we can upgrade this plot to make it even clearer. In particular, you can rename variables to be more descriptive and remove the text labels that display coefficients, t-statistics, and p-values to reduce clutter, allowing us to focus on the overall direction and magnitude of each effect.

For a complete list of all features you can modify in `ggcoefstats`, please refer to the ggcoefstats GitHub page. Additionally, if you're interested in the types of regression models that `ggcoefstats` supports for future analysis, see this comprehensive overview

Let's create a streamlined dot-and-whisker plot:

```
# Upgrade the dot-and-whisker plot
ggcoefstats(
  x = m1,
  stats.labels = FALSE,
  ## Removes text labels (coefficient, t-stat, p-value)
  point.args = list(size = 2.0, color = "blue", na.rm = TRUE),
  ## Set the size and color of the dots (coefficient estimates)
  vline.args = list(linewidth = 0.5, linetype = "dashed"),
```

```
  ## Set the line type and width of the vertical reference line
  xlab = "Coefficient Estimates",
  ylab = "Predictors",
  title = "Regression Estimates for FDI Inflows"
) +
  scale_y_discrete(labels = c(
    "truthvictim" = "Truth Commission",
    "damage"      = "Conflict Damage",
    "duration"    = "Conflict Duration",
    "peaceagr"    = "Peace Agreement",
    "victory"     = "Victory"
  ))
```




```
  ## Rename variable to more descriptive label
```

Comparing Regression Models Using a Combined Dot-and-Whisker Plot

To deepen our analysis, it is useful to compare different model specifications to better understand the effect of the independent variable on the outcome. Let's compare our initial model (`m1`), which includes only conflict-related controls, with a simple model (`m2`) that regresses only `truthvictim` on `fdiflow`, and an extended model (`m3`) that adds both conflict and political controls. By combining the dot-and-whisker plots for these three models into a single figure, we can directly observe how the inclusion of control variables affects the estimated impact of truth commissions on FDI inflows.

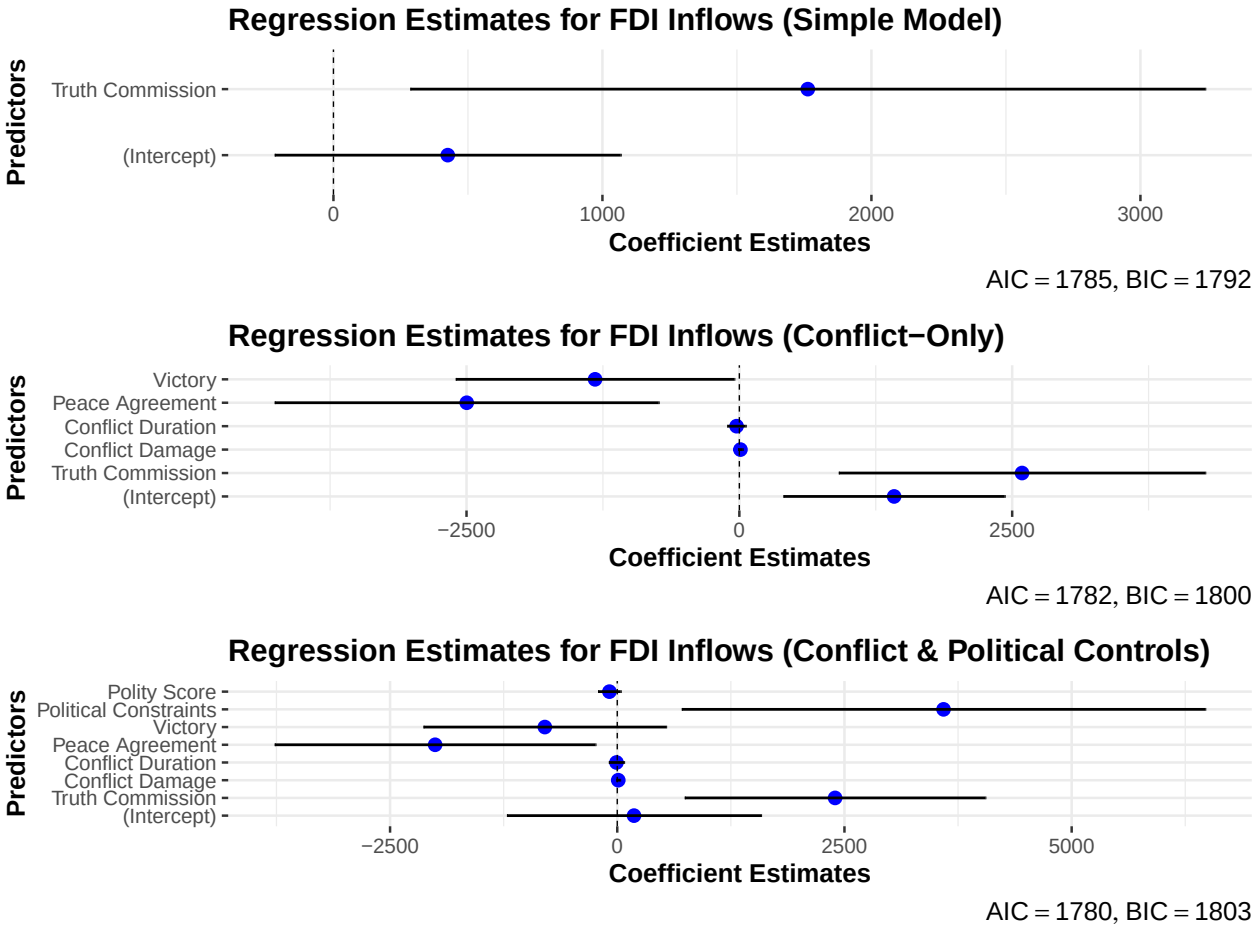
```r
# Estimate the two additional regression model
m2 <- lm(fdiflow ~ truthvictim,
         data = al2012)

m3 <- lm(fdiflow ~ truthvictim + damage + duration + peaceagr + victory + polcon +
           polity2, data = al2012)

# Combine the dot-and-whisker plots for models m1 and m2
combine_plots(
  plotlist = list(
    # Plot for model m2: Simple Model
    ggcoefstats(
      x = m2,
      stats.labels = FALSE,
      ## Removes text labels (coefficient, t-stat, p-value)
      point.args = list(size = 2.0, color = "blue", na.rm = TRUE),
      ## Set the size and color of the dots (coefficient estimates)
      vline.args = list(linewidth = 0.25, linetype = "dashed"),
      ## Set the line type and width of the vertical reference line
      xlab = "Coefficient Estimates",
      ylab = "Predictors",
      title = "Regression Estimates for FDI Inflows (Simple Model)"
    ) +
      scale_y_discrete(labels = c(
        "truthvictim" = "Truth Commission",
        "damage"      = "Conflict Damage",
        "duration"    = "Conflict Duration",
        "peaceagr"    = "Peace Agreement",
        "victory"     = "Victory"
      )),
    # Plot for model m1: Conflict-Only Controls
    ggcoefstats(
      x = m1,
      stats.labels = FALSE,
      ## Removes text labels (coefficient, t-stat, p-value)
      point.args = list(size = 2.0, color = "blue", na.rm = TRUE),
      ## Set the size and color of the dots (coefficient estimates)
      vline.args = list(linewidth = 0.25, linetype = "dashed"),
      ## Set the line type and width of the vertical reference line
      xlab = "Coefficient Estimates",
      ylab = "Predictors",
      title = "Regression Estimates for FDI Inflows (Conflict-Only)"
    ) +
      scale_y_discrete(labels = c(
        "truthvictim" = "Truth Commission",
        "damage"      = "Conflict Damage",
        "duration"    = "Conflict Duration",
        "peaceagr"    = "Peace Agreement",
        "victory"     = "Victory"
      )),
    # Plot for model m3: Conflict & Political Controls
    ggcoefstats(
      x = m3,
```

```
      stats.labels = FALSE,
      ## Removes text labels (coefficient, t-stat, p-value)
      point.args = list(size = 2.0, color = "blue", na.rm = TRUE),
      ## Set the size and color of the dots (coefficient estimates)
      vline.args = list(linewidth = 0.25, linetype = "dashed"),
      ## Set the line type and width of the vertical reference line
      xlab = "Coefficient Estimates",
      ylab = "Predictors",
      title = "Regression Estimates for FDI Inflows (Conflict & Political Controls)"
    ) +
      scale_y_discrete(labels = c(
        "truthvictim" = "Truth Commission",
        "damage"      = "Conflict Damage",
        "duration"    = "Conflict Duration",
        "peaceagr"    = "Peace Agreement",
        "victory"     = "Victory",
        "polcon"      = "Political Constraints",
        "polity2"     = "Polity Score"
      ))
  ),
  plotgrid.args = list(nrow = 3),
  annotation.args = list(title = "Comparing Regression Estimates")
)
```

Comparing Regression Estimates



Extracting and Customizing Tidy Regression Results

While `ggcoefstats` provides a clear visualization of regression results, you might want to further customize the display of your results or create other personalized visualizations beyond dot-and-whisker plots. This can be achieved by extracting the model results into a tidy dataframe using the broom package. To extract tidy regression results, use the `tidy()` function from `broom` package to obtain a dataframe containing the regression coefficients, standard errors, and p-values.

```
# Extract tidy regression results and create tidy dataframes
m1df <- tidy(m1)
m2df <- tidy(m2)
m3df <- tidy(m3)
```

Once you have this tidy dataframe, you can create any custom visualizations that meet your specific needs. This approach allows you to adjust labels, themes, and other plot aesthetics, giving you full control over how your results are presented.

Conclusion

In this assignment, we've explored how post-conflict justice institutions—specifically, truth commissions—affect FDI inflows by running regression analyses and visualizing the results with dot-and-whisker plots generated by the `ggcoefstats` function from the `ggstatsplot` package.

Moving forward, we'll continue to build on these concepts in both group work and homework assignments, providing additional opportunities to practice and refine your data visualization skills for displaying regression results.