# Session 13
# Data Visualization 1

## R for Social Data Science

Jeffrey Ziegler, PhD

Assistant Professor in Political Science & Data Science
Trinity College Dublin

Fall 2022

# Road map for today

Last week:

- Data input and output
- Data frames and alternatives
- ‘tidyverse‘ packages
- Working with tabular data
- Summary statistics

This time:

- Effectively tell a story with visuals
- Plotting in base R

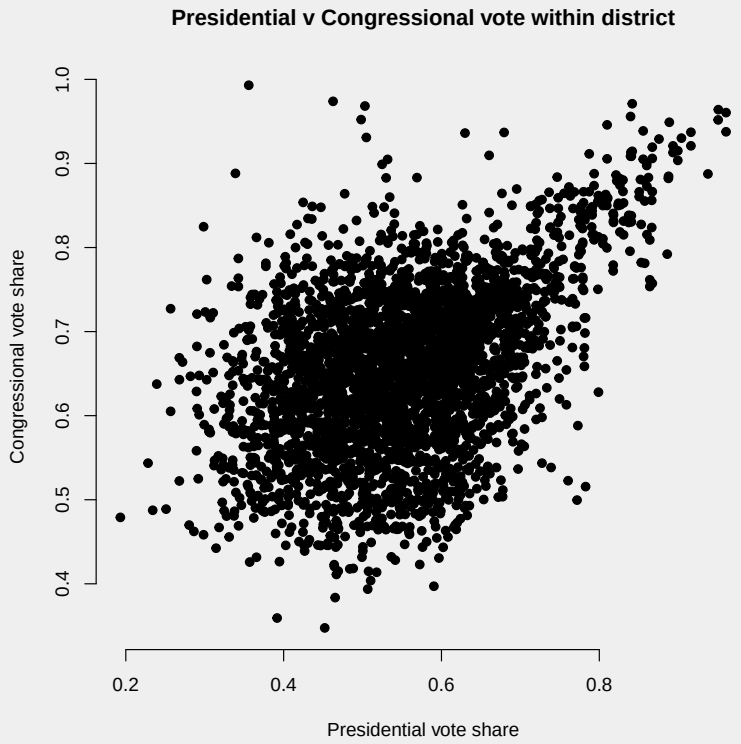
# Principals of "good" data vis - Tufte

- "Show" data
- Think about substance rather than about methodology, graphic design, technology, etc.
- Avoid distorting data
- Present many numbers (info) in a small space
- Make large data sets coherent
- Encourage eye to compare different pieces of data
- Reveal data at several levels of detail
- Integrate statistical and verbal descriptions of a data set

# Scatterplots

- Scatter plot can be created using `plot(x, y)`

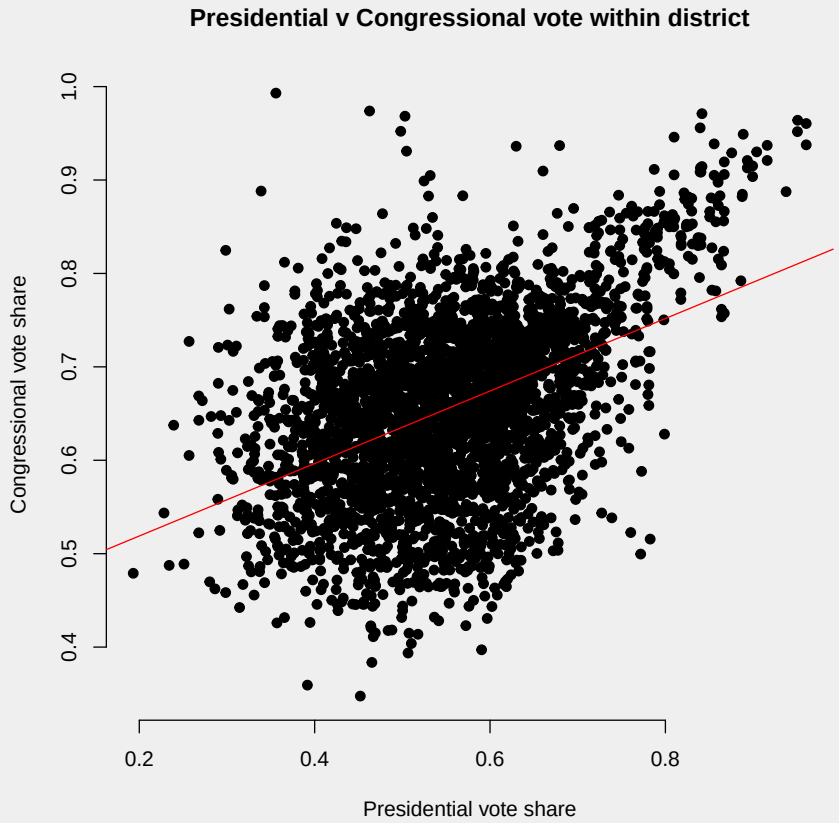
```
# Plot with main and axis titles
# Change point shape (pch = 19) and remove frame
plot(inc_local$presvote, inc_local$voteshare,
     main = "Presidential v Congressional vote within district",
     xlab = "Presidential vote share", ylab = "Congressional vote
    share", pch = 19, frame = F)
```



# Scatterplots

- `lm()` can be used to fit linear models between y and x
- Regression line can be added on the plot using `abline()`, which takes output of `lm()` as an argument

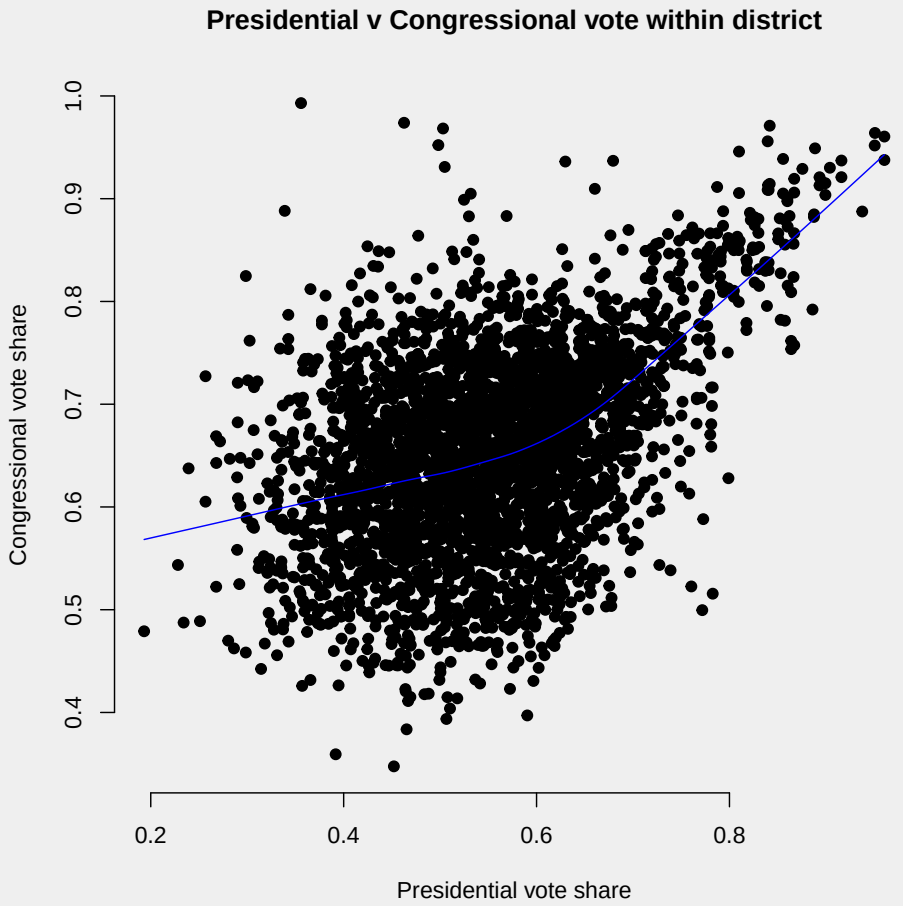
```
1 abline(lm(voteshare ~ presvote, data = inc_local), col = "red")
```



# Scatterplots

- Can also add a smoothing line using `loess()`

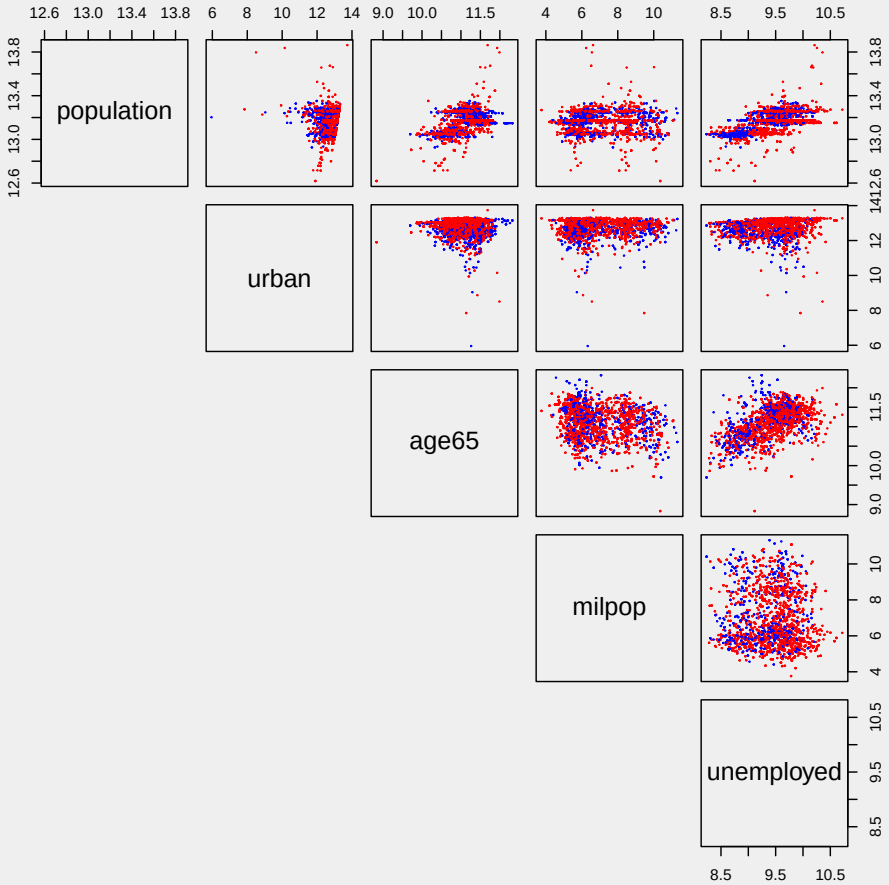
```
1  lines(lowess(inc_local$presvote, inc_local$voteshare), col = "blue")
```



# Scatterplots of many variables

- Can produce a matrix of scatter plots with `pairs()`

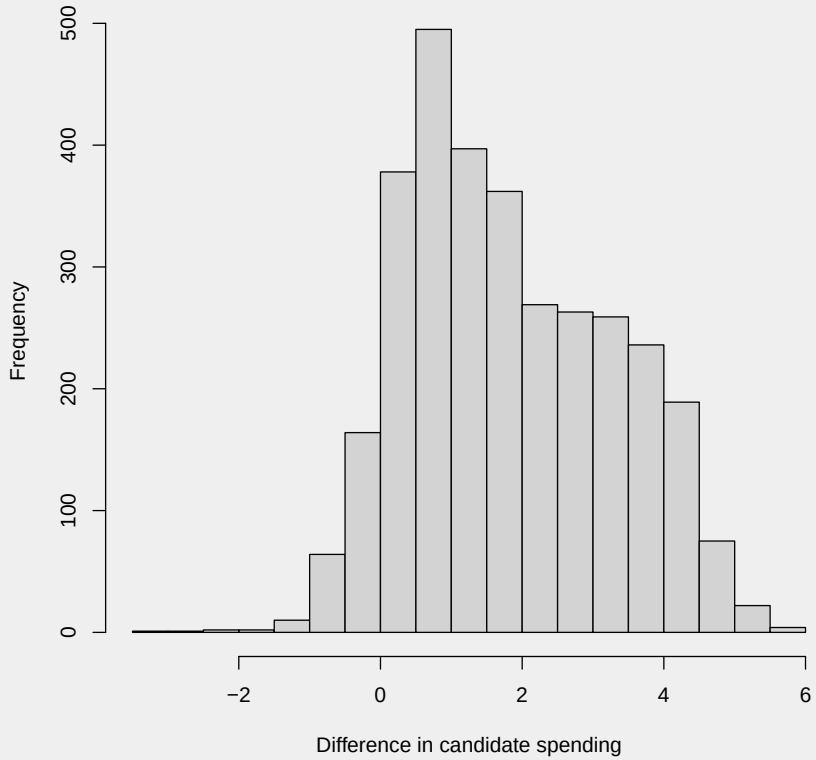
```
my_cols <- c("red", "blue")
pairs(inc_local[,15:ncol(inc_local)], pch = 19,  cex = 0.15,
      col = my_cols[as.factor(inc_local$south)],
      lower.panel=NULL)
```



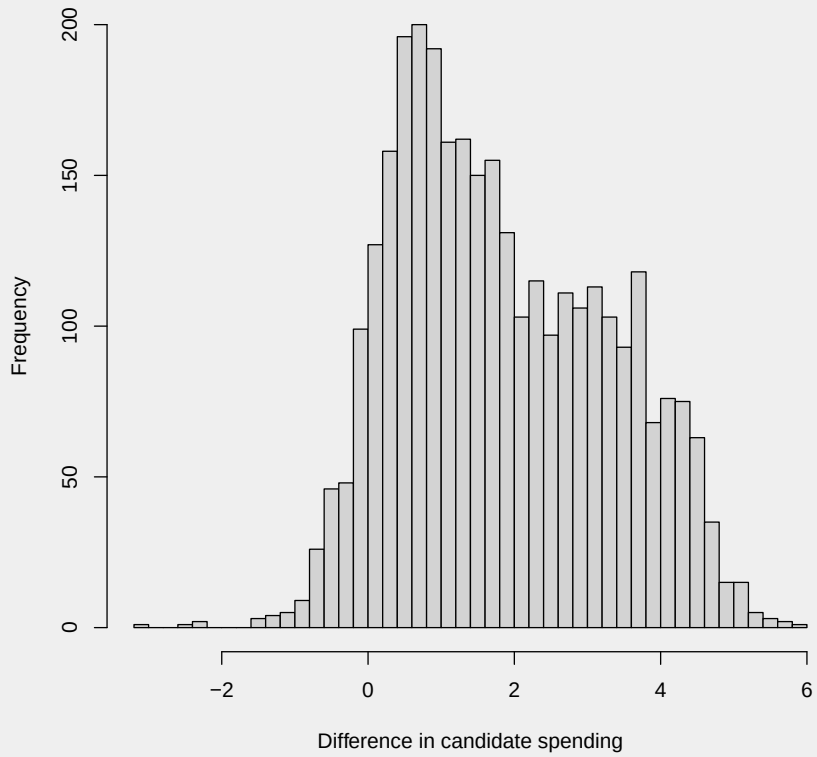
# Histograms

## Summarize distribution of variable as count

```
1 hist(inc_local$difflog, main = "", xlab = "
      Difference in candidate spending")
```



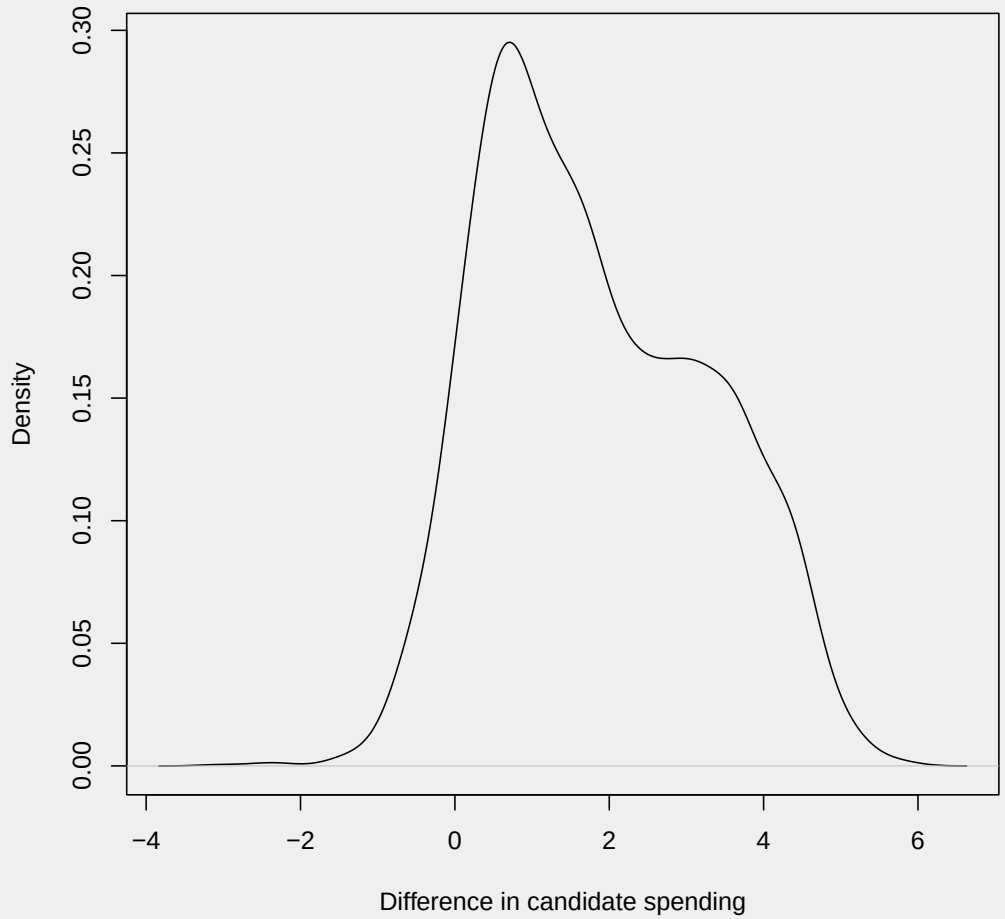

```
1 hist(inc_local$difflog, main = "", xlab = "
      Difference in candidate spending",
      breaks=50)
```



# Density plot

Summarize distribution of variable as proportion

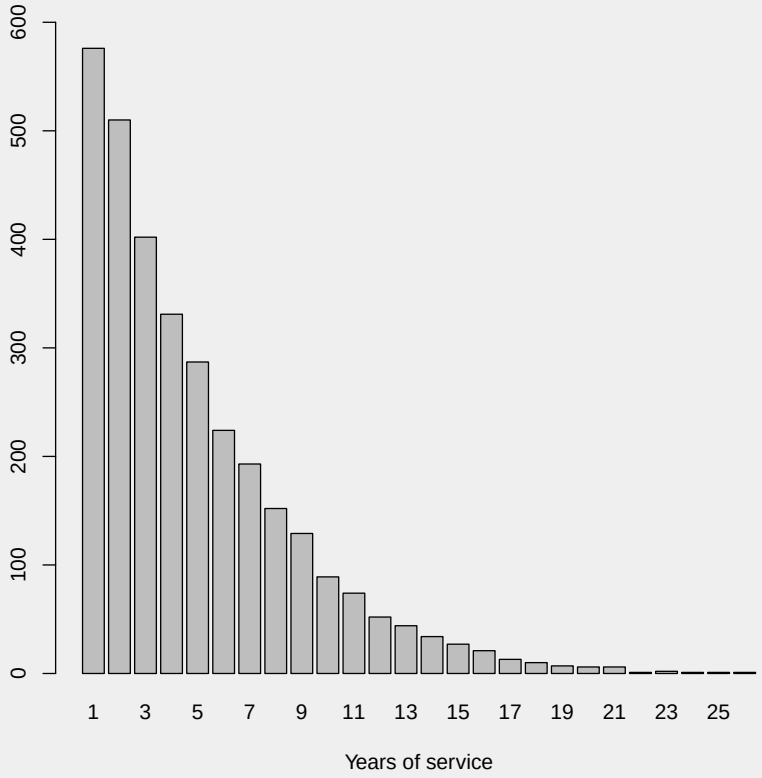
```
1 plot(density(inc_local$difflog), main = "", xlab = "Difference in
    candidate spending")
```



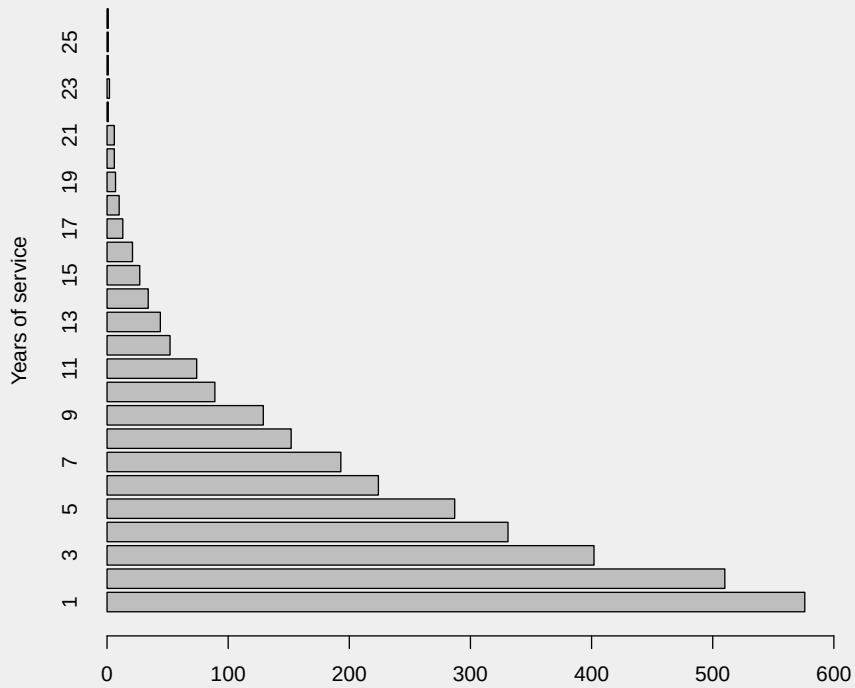
# Bar Chart

Summarize count or proportion that you've already calculated

```
barplot(table(inc_local$seniority), ylim=c
    (0, 600), xlab="Years of service")
```



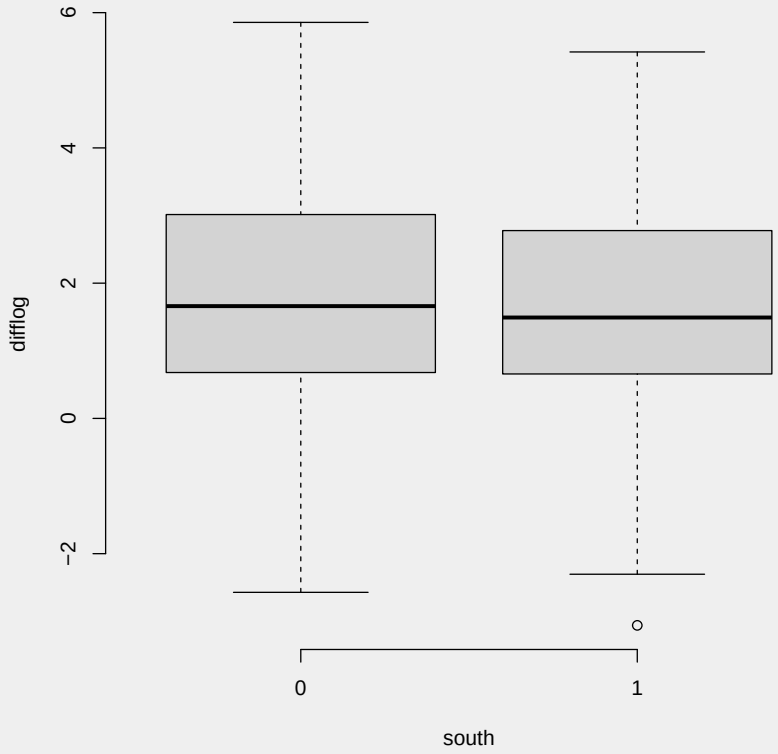

```
barplot(table(inc_local$seniority), xlim=c
    (0, 600), ylab="Years of service",
    horiz = T)
```



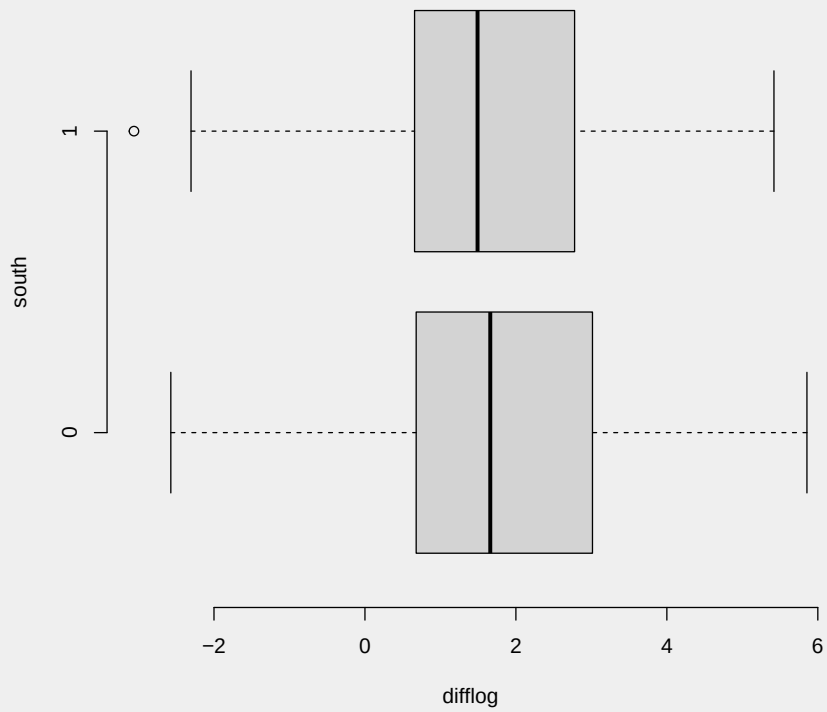
# Box plot

Summarize distribution (mean, quartiles, outliers) of variable by group

```
1 boxplot(difflog ~ south, data = inc_local,
      frame = F)
```




```
1 boxplot(difflog ~ south, data = inc_local,
      frame = F, horizontal = T)
```



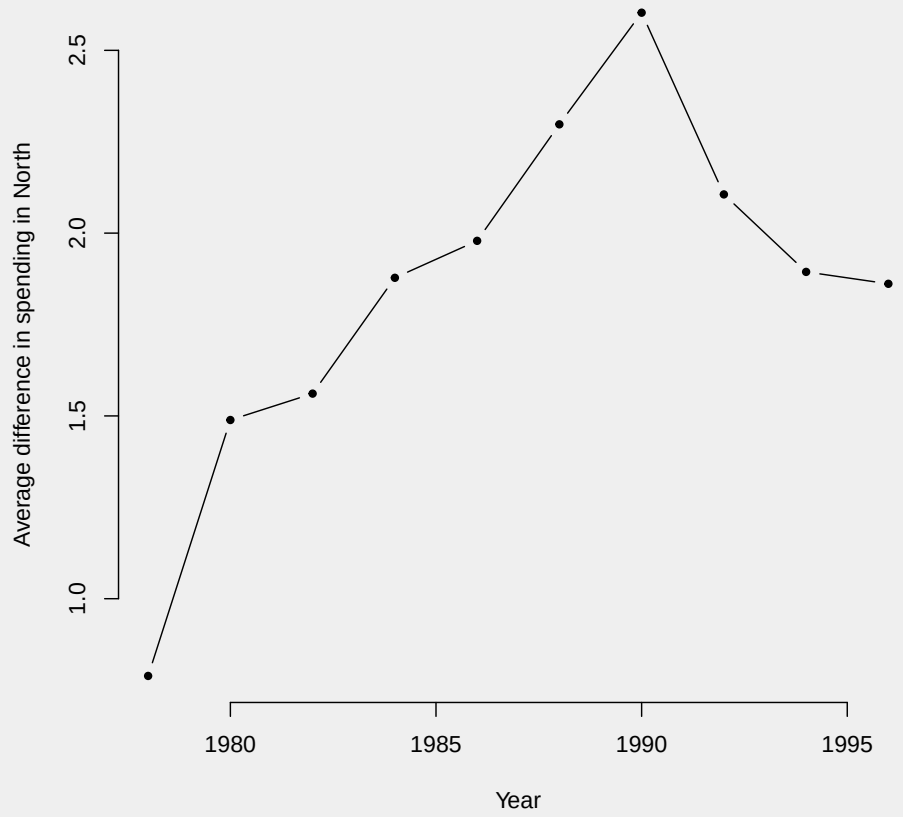
# Line chart

Type: character indicating type of plotting

- “p” for points
- “l” for lines
- “b” for both points and lines
- “c” for empty points joined by lines
- “o” for overplotted points and lines
- “s” and “S” for stair steps
- “n” does not produce any points or lines
- lty: line types
  - Line types can either be specified as
    - Integer (0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash), or
    - Character strings “blank”, “solid”, “dashed”, “dotted”, “dotdash”, “longdash”, or “twodash”

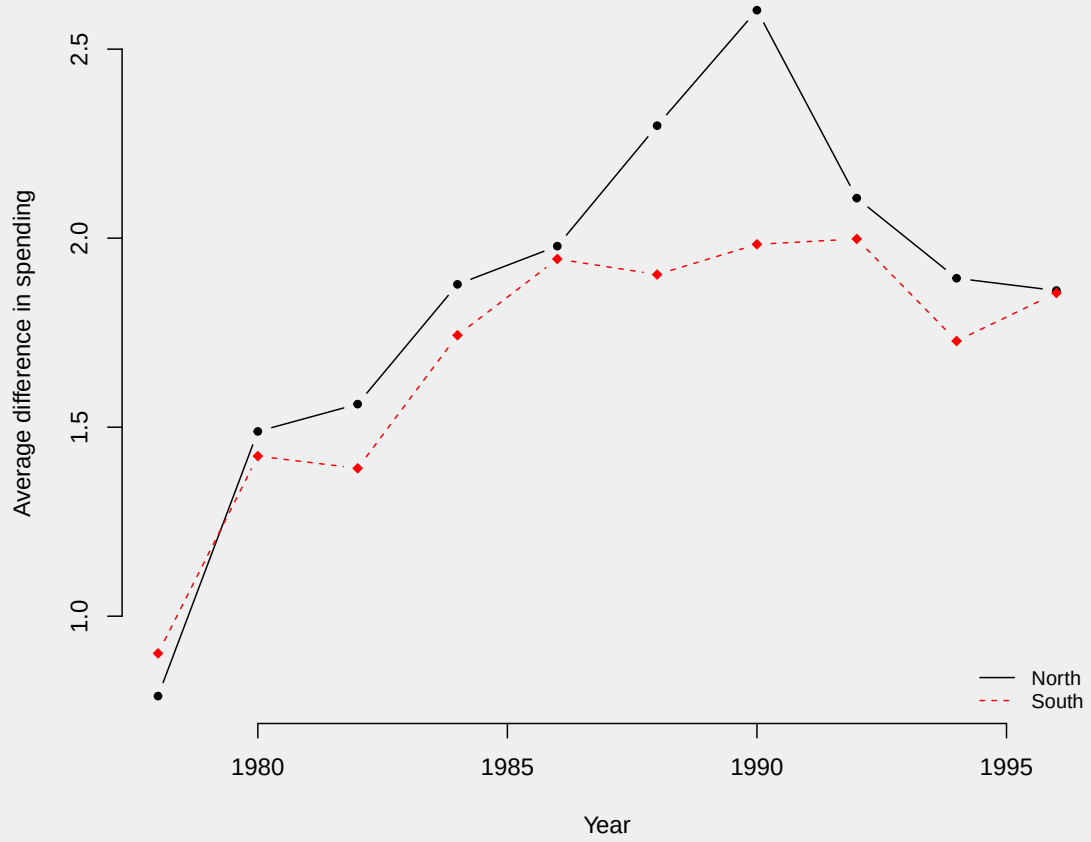
# Line chart

```r
south_year_avg <- inc_local %>%
  group_by(south, year) %>%
  summarize(avg = mean(difflog))
plot(south_year_avg[1:10,]$year, south_year_avg[1:10,]$avg, type = "b", frame = F, pch =
     20, xlab = "Year", ylab = "Average difference in spending in North")
```



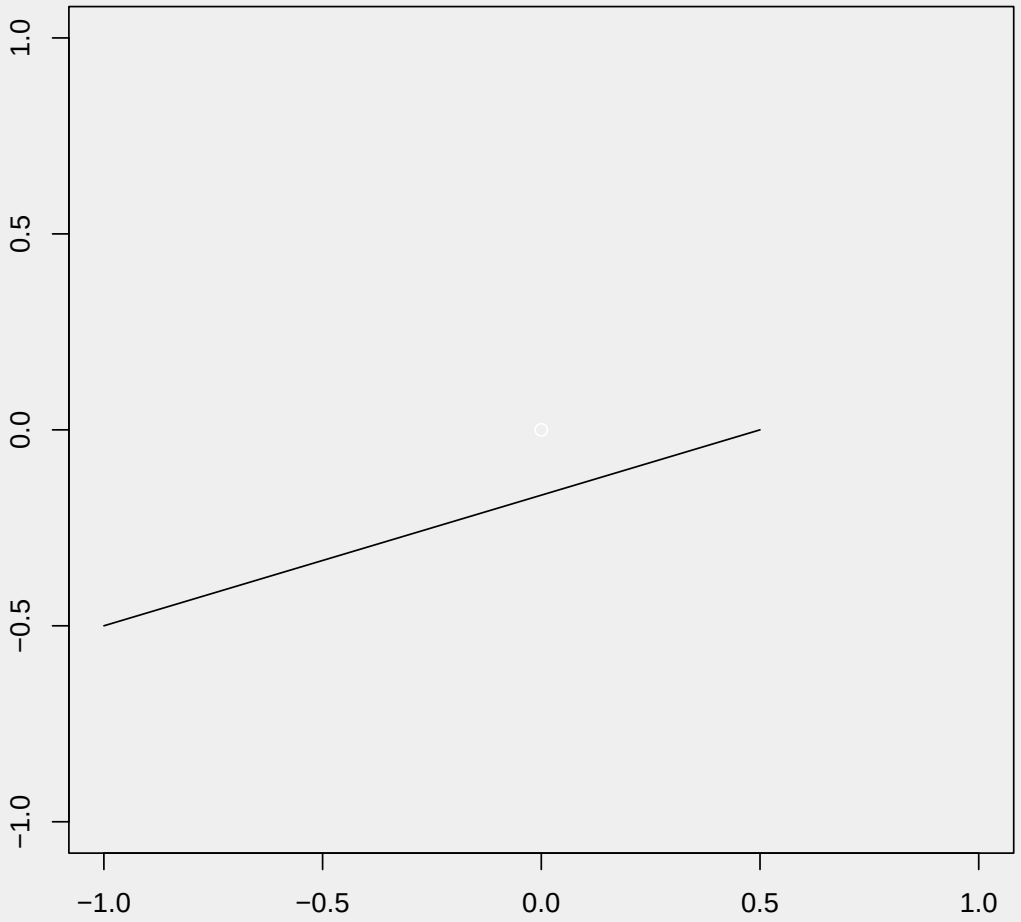
# Line chart

```
lines(south_year_avg[11:20,]$year, south_year_avg[11:20,]$avg, pch = 18, col = "red",
      type = "b", lty = 2)
# Add a legend to the plot
legend("bottomright", legend=c("North", "South"),
       col=c("black", "red"), lty = 1:2, cex=0.8, box.lty=0)
```



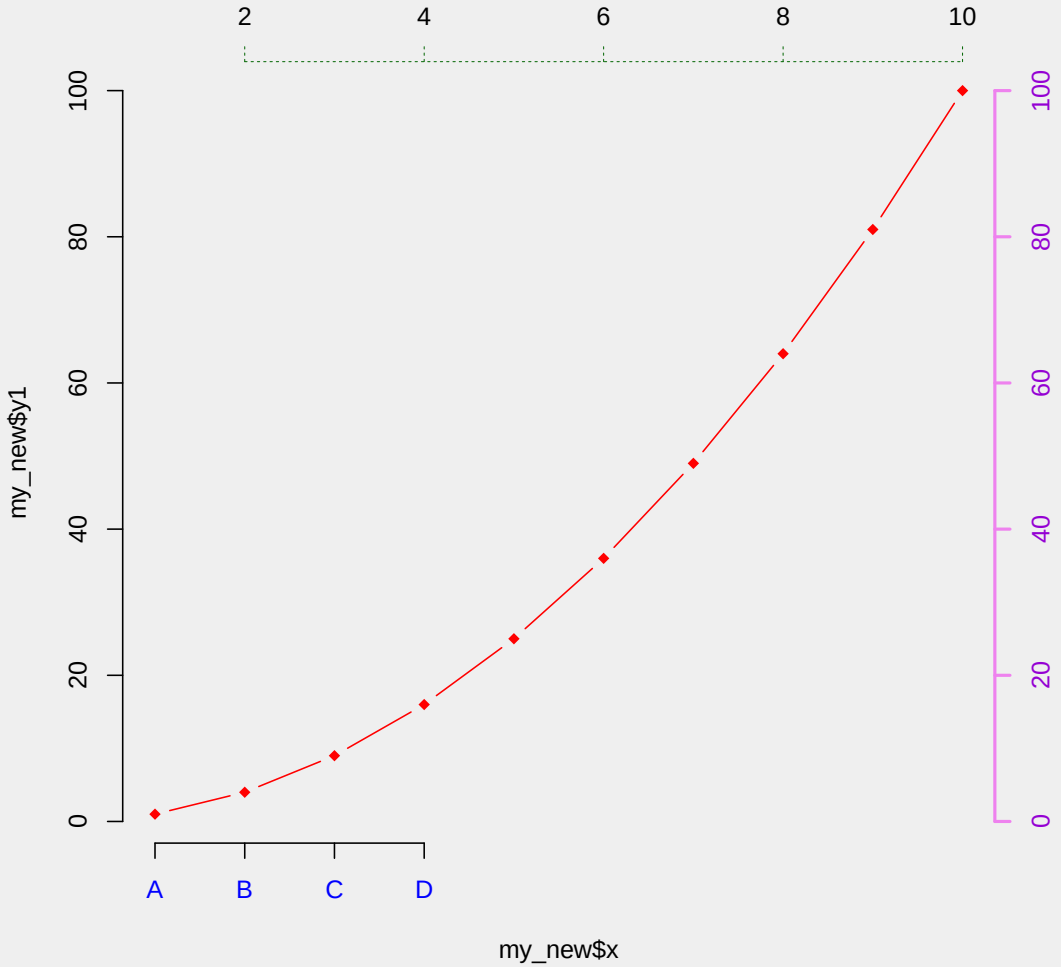
# Segments

```
1  plot(0, 0, col = "white", xlab = "", ylab = "")
2  segments(x0 = - 1, y0 = - 0.5, x1 = 0.5, y1 = 0)
```



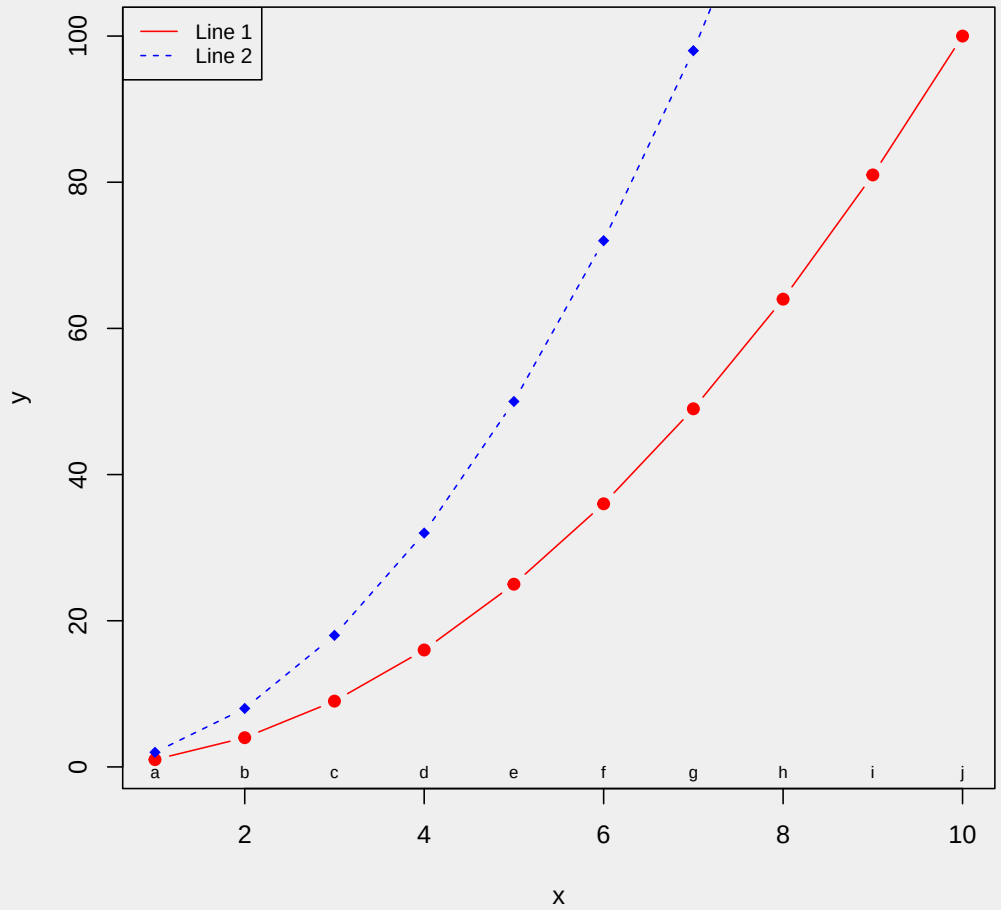
# X AND Y AXES

```r
plot(my_new$x, my_new$y1, pch=18, col="red", type="b",
     frame=FALSE, xaxt="n") # Remove x axis
axis(1, 1:4, LETTERS[1:4], col.axis="blue")
axis(3, col = "darkgreen", lty = 2, lwd = 0.5)
axis(4, col = "violet", col.axis = "dark violet", lwd = 2)
```



# Plot Labels

```
1  text(my_new$x, 2, my_new$x2,
2       cex=0.65, pos=1,col="black")
```



# Save Plots

There are two ways to think about saving your plots:

- If you're working in RMarkdown, just "knit" your file and your plots will show up as part of your HTML, Word, or PDF document
- If need to save an individual plot for some other purpose (e.g. putting it in a report created in Latex, Powerpoint, Word), use `pdf()` and `dev.off()` functions

```
1 pdf("../graphics/histogram.pdf")
2 hist(inc_local$difflog, main = "", xlab = "Difference in candidate spending")
3 dev.off()
```

# Tutorial - Plotting groups differences

We're interested in whether women promote different policies than men?

- Load in this dataset on drinking water facilities and a randomized policy experiment in India, where since the mid-1990s, 13 of village council heads have been randomly reserved for women
- Estimate the effect of the reservation policy on the number of new or repaired drinking water facilities in the villages
  - Hint: t.test(y  x, alternative="two.sided")
- Effectively plot this difference

# Overview

This time:

- Plotting in base R

Next time:

- Plotting in `ggplot`