Week 7

Abhishek Solanki

1/4/2021

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.0 ──
```

```
## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.4     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gapminder)
library(socviz)

##Loading gss_sm data
data("gss_sm")

# Showing metadata from the gss_sm dataframe
gss_sm
```

```
## # A tibble: 2,867 x 32
##     year    id ballot   age childs sibs  degree race  sex   region
income16
##    <dbl> <dbl> <labe> <dbl>  <dbl> <lab> <fct>  <fct> <fct> <fct>  <fct>
##  1  2016     1 1         47      3 2     Bache… White Male  New E…
$170000…
##  2  2016     2 2         61      0 3     High … White Male  New E… $50000
…
##  3  2016     3 3         72      2 3     Bache… White Male  New E… $75000
…
##  4  2016     4 1         43      4 3     High … White Fema… New E…
$170000…
##  5  2016     5 3         55      2 2     Gradu… White Fema… New E…
$170000…
##  6  2016     6 2         53      2 2     Junio… White Fema… New E… $60000
…
##  7  2016     7 1         50      2 2     High … White Male  New E…
$170000…
```

```
##  8  2016     8 3         23      3 6     High … Other Fema… Middl… $30000
…
##  9  2016     9 1         45      3 5     High … Black Male  Middl… $60000
…
## 10  2016    10 3         71      4 1     Junio… White Male  Middl… $60000
…
## # … with 2,857 more rows, and 21 more variables: relig <fct>, marital
<fct>,
##   padeg <fct>, madeg <fct>, partyid <fct>, polviews <fct>, happy <fct>,
##   partners <fct>, grass <fct>, zodiac <fct>, pres12 <labelled>,
##   wtssall <dbl>, income_rc <fct>, agegrp <fct>, ageq <fct>, siblings
<fct>,
##   kids <fct>, religion <fct>, bigregion <fct>, partners_rc <fct>, obama
<dbl>
```

```
## using organdata, create a table of summary statistics by country called
by_country (show the mean of donors, gdp, health, roads, cerebvas, and the
standard deviation of donors)
by_country <- organdata %>% group_by(consent_law, country) %>%
summarize_if(is.numeric, funs(mean, sd), na.rm = TRUE) %>% ungroup()
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

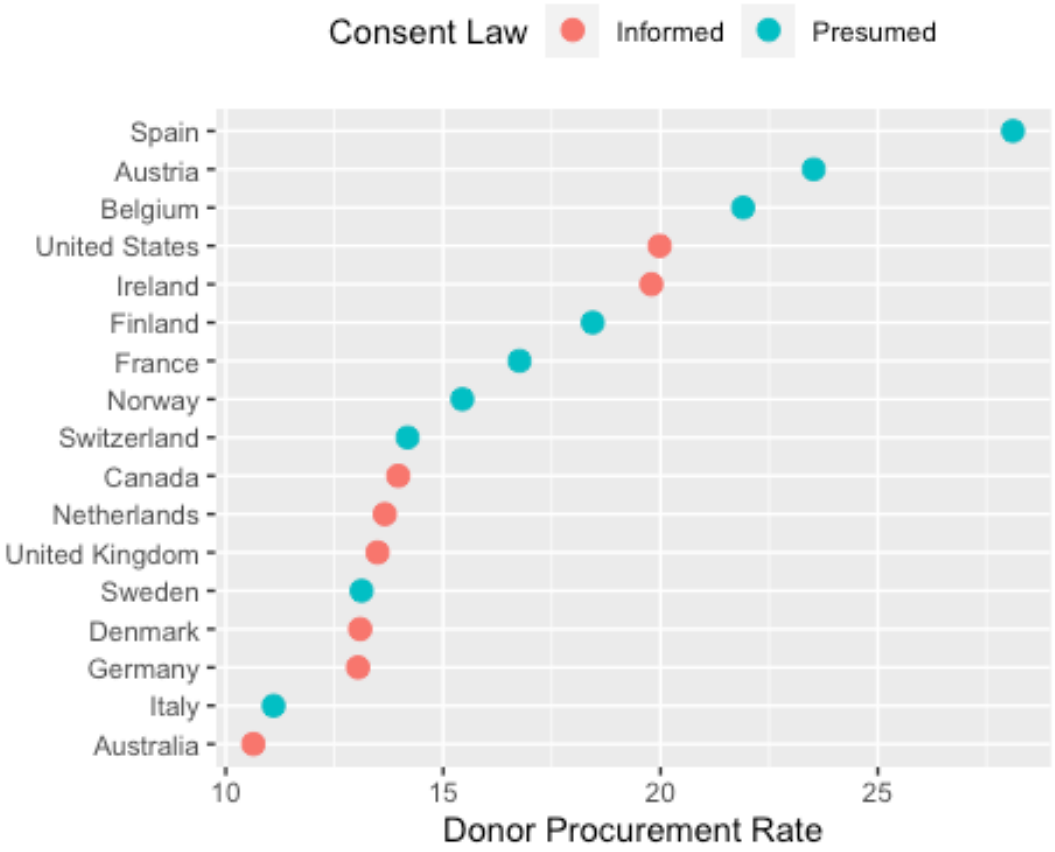
```
by_country <- organdata %>% group_by(consent_law, country) %>%
summarize(donors_mean= mean(donors, na.rm = TRUE), donors_sd = sd(donors,
na.rm = TRUE), gdp_mean = mean(gdp, na.rm = TRUE), health_mean = mean(health,
na.rm = TRUE), roads_mean = mean(roads, na.rm = TRUE), cerebvas_mean =
mean(cerebvas, na.rm = TRUE))
```

```
## `summarise()` regrouping output by 'consent_law' (override with `.groups`
argument)
```

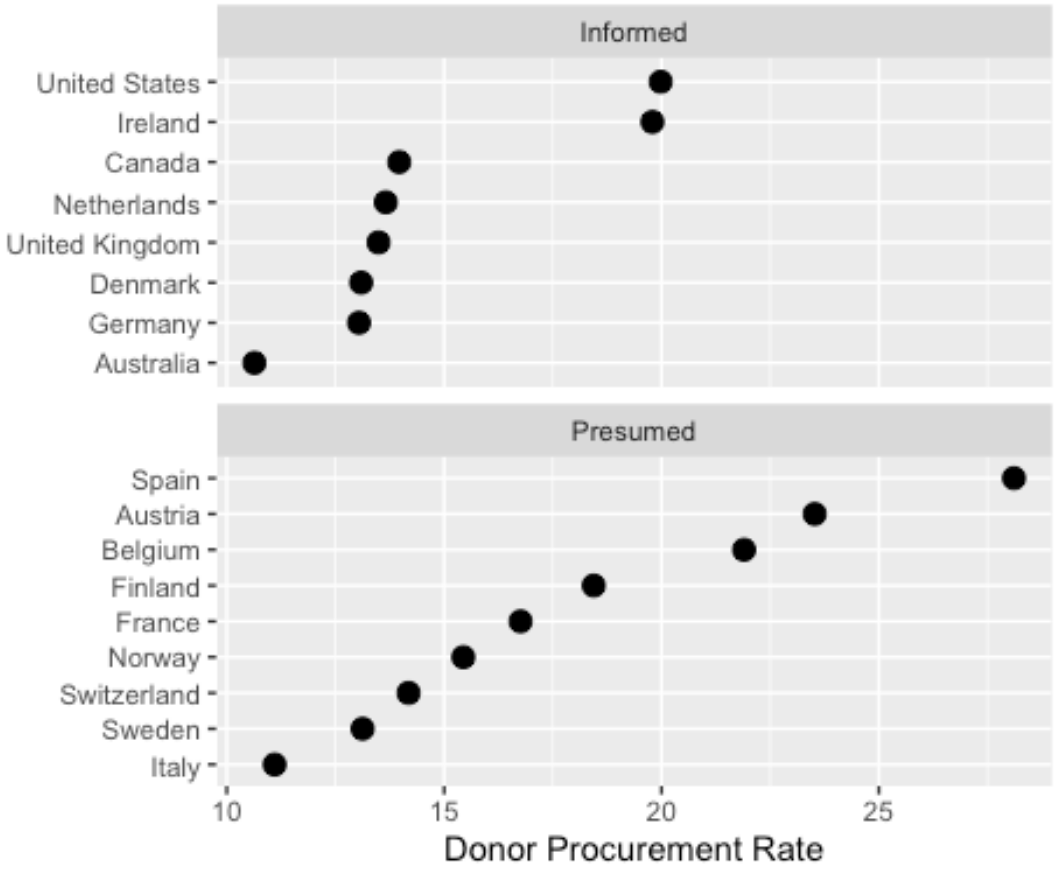
```
by_country
```

```
## # A tibble: 17 x 8
## # Groups:   consent_law [2]
##   consent_law country donors_mean donors_sd gdp_mean health_mean
roads_mean
##   <chr>       <chr>         <dbl>     <dbl>    <dbl>       <dbl>
```

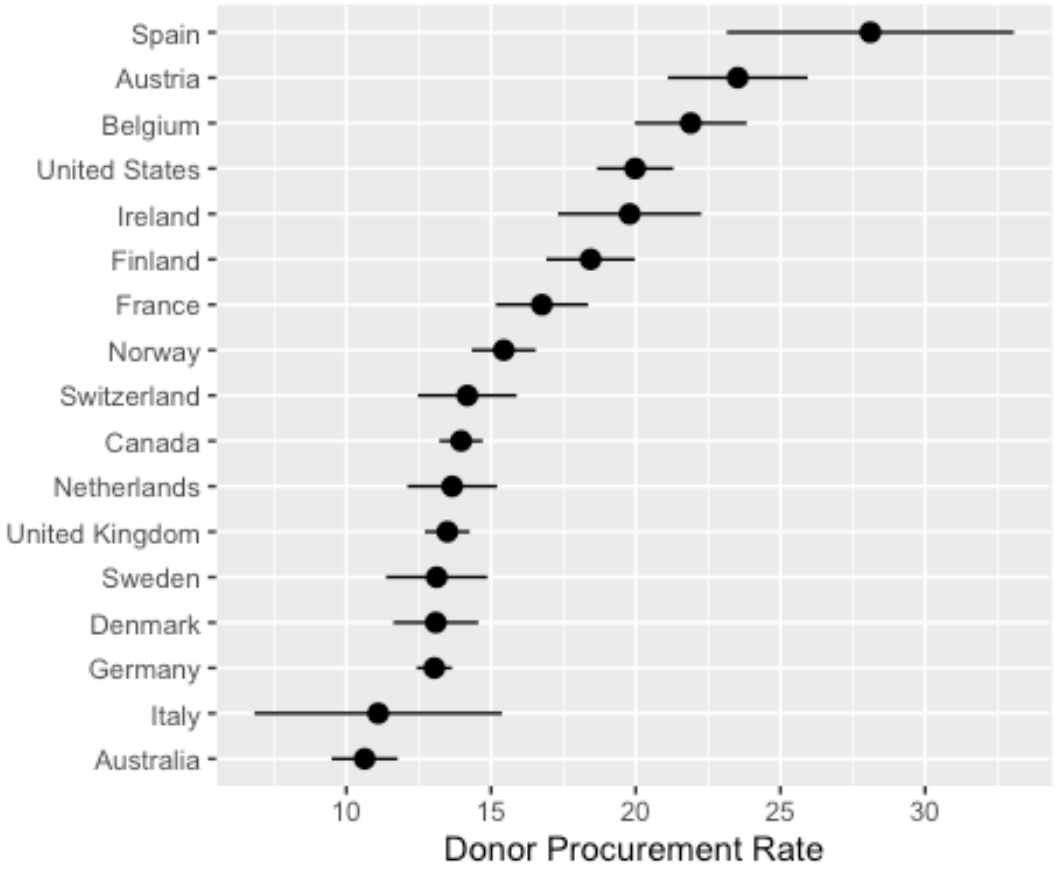
```
<dbl>
##  1 Informed    Austra…        10.6     1.14    22179.       1958.
105.
##  2 Informed    Canada         14.0     0.751   23711.       2272.
109.
##  3 Informed    Denmark        13.1     1.47    23722.       2054.
102.
##  4 Informed    Germany        13.0     0.611   22163.       2349.
113.
##  5 Informed    Ireland        19.8     2.48    20824.       1480.
118.
##  6 Informed    Nether…        13.7     1.55    23013.       1993.
76.1
##  7 Informed    United…        13.5     0.775   21359.       1561.
67.9
##  8 Informed    United…        20.0     1.33    29212.       3988.
155.
##  9 Presumed    Austria        23.5     2.42    23876.       1875.
150.
## 10 Presumed    Belgium        21.9     1.94    22500.       1958.
155.
## 11 Presumed    Finland        18.4     1.53    21019.       1615.
93.6
## 12 Presumed    France         16.8     1.60    22603.       2160.
156.
## 13 Presumed    Italy          11.1     4.28    21554.       1757
122.
## 14 Presumed    Norway         15.4     1.11    26448.       2217.
70.0
## 15 Presumed    Spain          28.1     4.96    16933        1289.
161.
## 16 Presumed    Sweden         13.1     1.75    22415.       1951.
72.3
## 17 Presumed    Switze…        14.2     1.71    27233        2776.
96.4
## # … with 1 more variable: cerebvas_mean <dbl>

## Producing a scatterplot of the by_country data with the points colored by
consent_law
p <- ggplot(data = by_country, mapping = aes(x = donors_mean, y =
reorder(country, donors_mean), color = consent_law))
p + geom_point(size=3) + labs(x = "Donor Procurement Rate", y = "", color =
"Consent Law") + theme(legend.position="top")
```



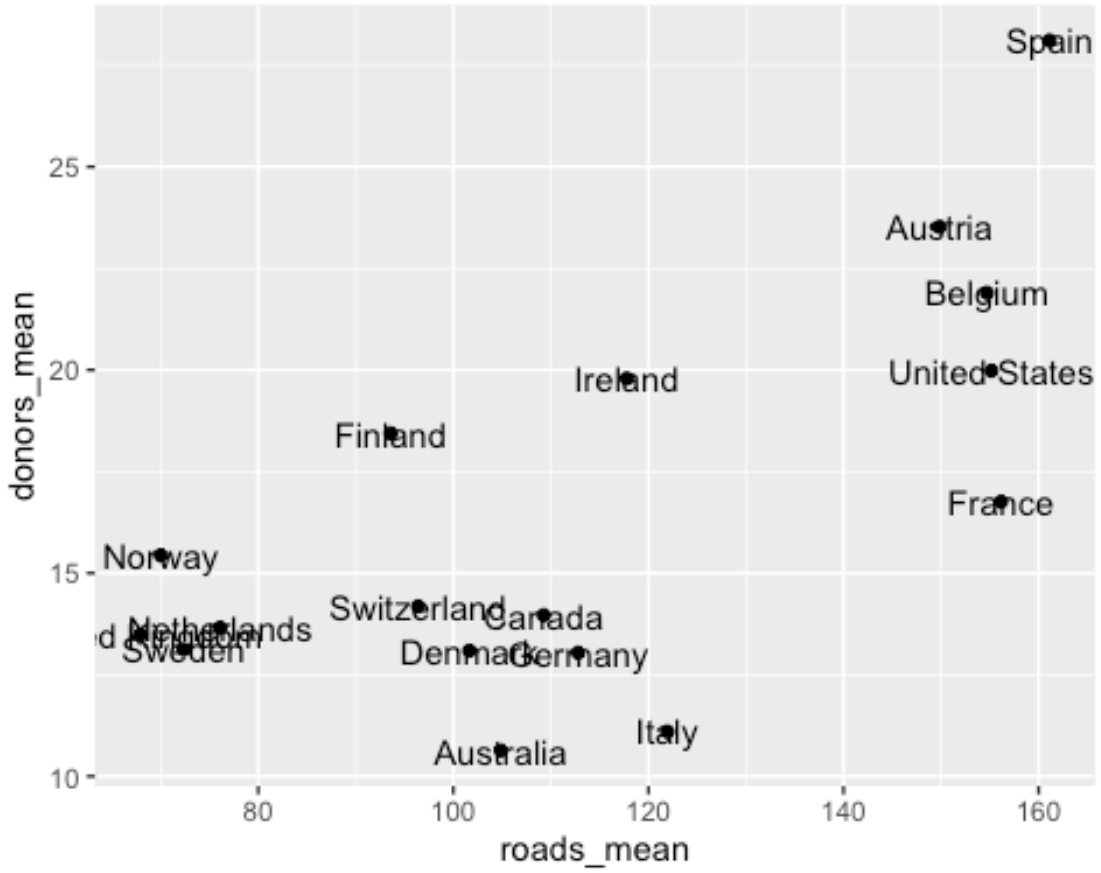

```
##Using facet_wrap() split the consent_law variable into two panels and rank
the countries by donation rate within the panels.

p2 <-  ggplot(data = by_country, mapping = aes(x = donors_mean, y =
reorder(country, donors_mean)))

p2 + geom_point(size=3) + facet_wrap(~ consent_law, scales = "free_y", ncol =
1) + labs(x= "Donor Procurement Rate", y= "")
```



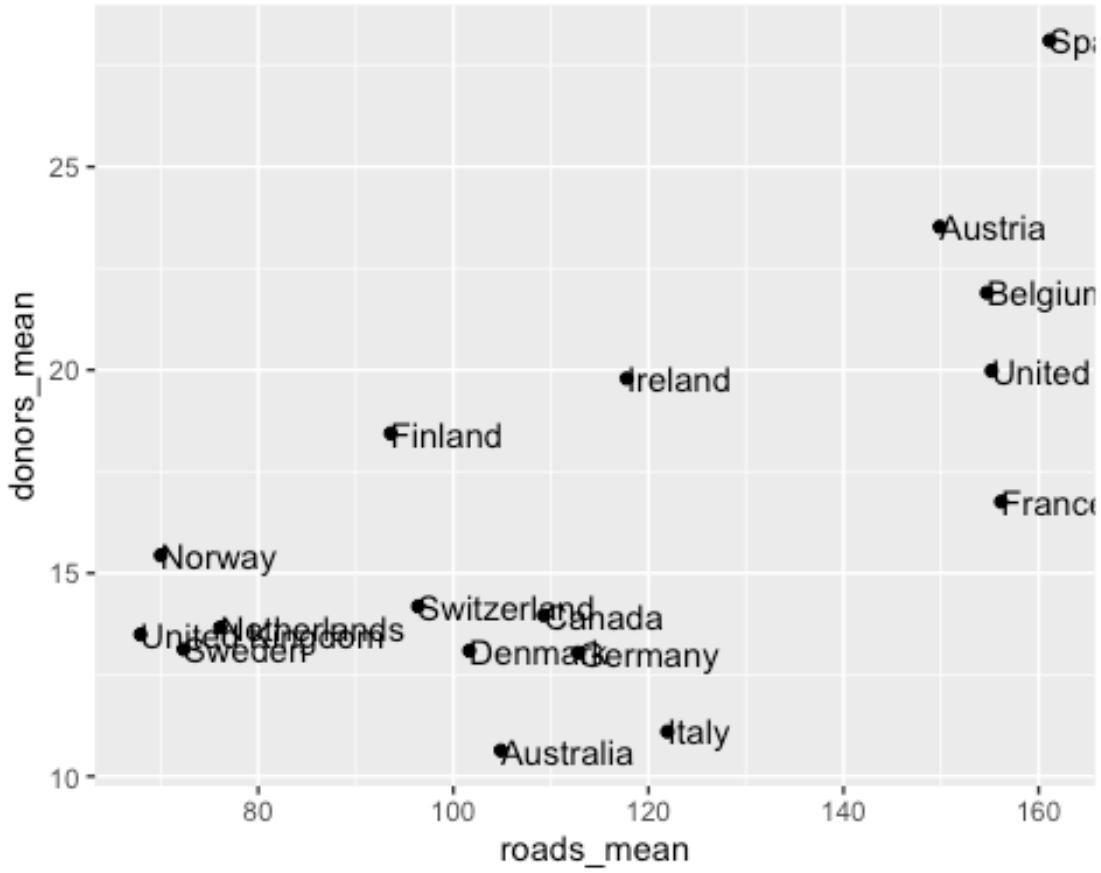

```
## Using geom_pointrange() to create a dot and whisker plot showing the mean
of donors and a confidence interval.

p3 <- ggplot(data = by_country, mapping = aes(x = reorder(country,
donors_mean), y = donors_mean))
p3 + geom_pointrange(mapping = aes(ymin = donors_mean - donors_sd, ymax =
donors_mean + donors_sd)) + labs(x= "", y= "Donor Procurement Rate") +
coord_flip()
```




```
##Creating a scatterplot of roads_mean v. donors_mean with the labels
identifying the country sitting to the right or left of the point.

p4 <- ggplot(data = by_country, mapping = aes(x = roads_mean, y =
donors_mean))
p4 + geom_point() + geom_text(mapping = aes(label = country))
```



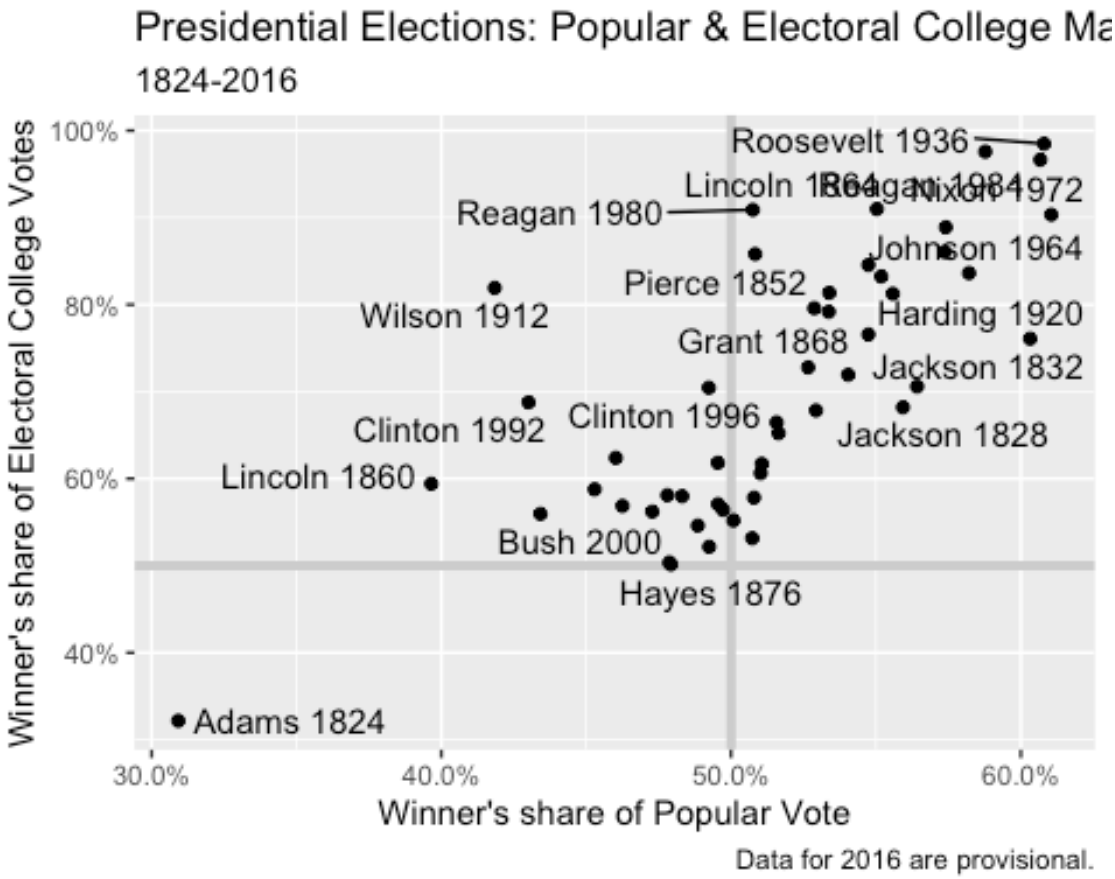

```
p5 <- ggplot(data = by_country, mapping = aes(x = roads_mean, y =
donors_mean))
p5 + geom_point() + geom_text(mapping = aes(label = country), hjust = 0)
```



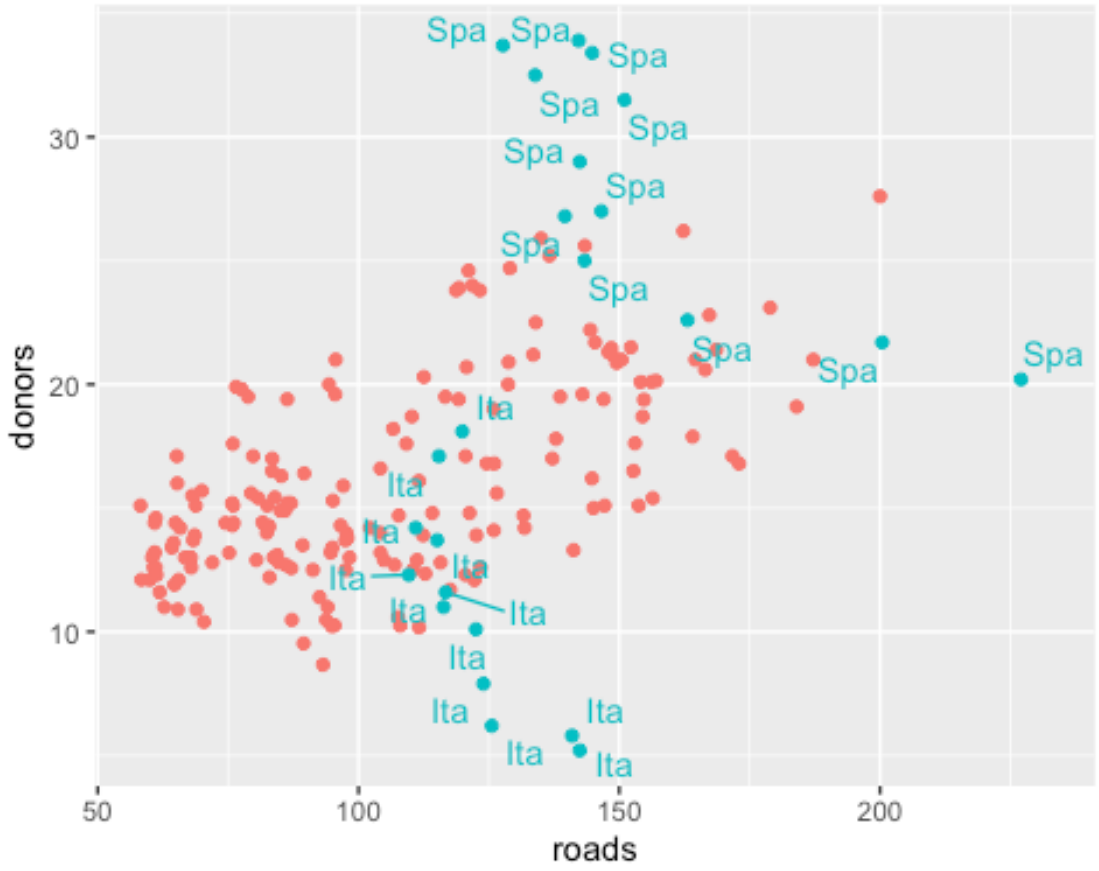

```
##loading the ggrepel() library
library(ggrepel)
elections_historic %>% select(2:7)
```

```
## # A tibble: 49 x 6
##     year winner                win_party ec_pct popular_pct
popular_margin
##    <int> <chr>                 <chr>      <dbl>       <dbl>
<dbl>
##  1  1824 John Quincy Adams     D.-R.      0.322       0.309        -0.104
##  2  1828 Andrew Jackson        Dem.       0.682       0.559         0.122
##  3  1832 Andrew Jackson        Dem.       0.766       0.547         0.178
##  4  1836 Martin Van Buren      Dem.       0.578       0.508         0.142
##  5  1840 William Henry Harrison Whig      0.796       0.529
0.0605
##  6  1844 James Polk            Dem.       0.618       0.495
0.0145
##  7  1848 Zachary Taylor        Whig       0.562       0.473
0.0479
##  8  1852 Franklin Pierce       Dem.       0.858       0.508
0.0695
##  9  1856 James Buchanan        Dem.       0.588       0.453         0.122
```

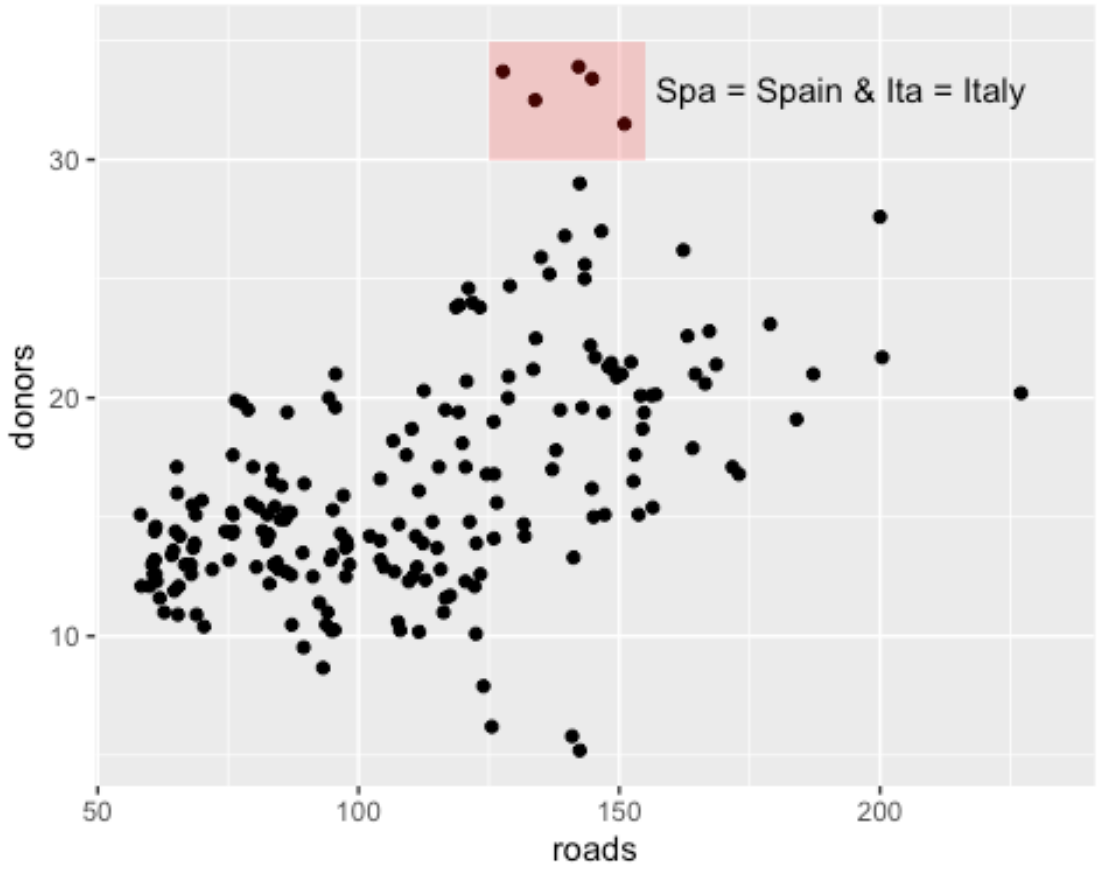
```
## 10  1860 Abraham Lincoln        Rep.       0.594       0.396         0.101
## # … with 39 more rows

## using the elections_historic data, plot the presidents popular
## vote percentage v electoral college vote percentage.
## draw axes at 50% for each attribute and use geom_text_repel()
## to keep the labels from obscuring the points.

p_title <- "Presidential Elections: Popular & Electoral College Margins"
p_subtitle <- "1824-2016"
p_caption <-  "Data for 2016 are provisional."

x_label <- "Winner's share of Popular Vote"
y_label <- "Winner's share of Electoral College Votes"
p6 <- ggplot(elections_historic, aes(x = popular_pct, y = ec_pct, label =
winner_label))
p6 + geom_hline(yintercept = 0.5, size = 1.4, color = "gray80") +
  geom_vline(xintercept = 0.5, size = 1.4, color = "gray80") +
  geom_point() +
  geom_text_repel() +
  scale_x_continuous(labels = scales::percent) +
  scale_y_continuous(labels = scales::percent) +
  labs(x = x_label, y = y_label, title = p_title, subtitle = p_subtitle,
caption = p_caption)

## Warning: ggrepel: 31 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
## What is the electoral college?
## The Electoral College is widely known as "Winner take all" system because
the winner of the popular vote in each state gets all of the state's
electoral votes.

## creating a new binary value column in organdata called 'ind' populated by
determining whether the ccode is "Spa" or "Ita" and the year is after than
1998.

organdata$ind <- organdata$ccode %in% c("Ita", "Spa") &
  organdata$year > 1998

p7 <- ggplot(data = organdata,
             mapping = aes(x = roads,
                           y = donors, color = ind))
p7 + geom_point() +
  geom_text_repel(data = subset(organdata, ind),
                  mapping = aes(label = ccode)) +
  guides(label = FALSE, color = FALSE)

## Warning: Removed 34 rows containing missing values (geom_point).
```




```
## creating an organdata plot of Roads v. Donors and map the ind attribute to
the color aesthetic.
## Label those points with the ccode and suppress the legends.

p8 <- ggplot(data = organdata, mapping = aes(x = roads, y = donors))
p8 + geom_point() +
  annotate(geom = "rect", xmin = 125, xmax = 155, ymin = 30, ymax = 35, fill
= "red", alpha = 0.2) +
  annotate(geom = "text", x = 157, y = 33, label = "Spa = Spain & Ita = Italy
", hjust = 0) +
  guides(label = FALSE, color = FALSE) ## 10. Added a label in a rectangle to
the previous plot that says

## Warning: Removed 34 rows containing missing values (geom_point).
```




```
"Spa = Spain & Ita = Italy"
```

```
## [1] "Spa = Spain & Ita = Italy"
```