

Assignment1_Mar18

Garrett Bullivant

2023-03-18

```
# observe our dataset
```

```
library(gapminder)
```

```
gapminder
```

```
## # A tibble: 1,704 x 6
```

```
##   country      continent  year lifeExp      pop gdpPercap
##   <fct>        <fct>    <int>  <dbl>    <int>   <dbl>
## 1 Afghanistan Asia      1952   28.8  8425333    779.
## 2 Afghanistan Asia      1957   30.3  9240934    821.
## 3 Afghanistan Asia      1962   32.0 10267083    853.
## 4 Afghanistan Asia      1967   34.0 11537966    836.
## 5 Afghanistan Asia      1972   36.1 13079460    740.
## 6 Afghanistan Asia      1977   38.4 14880372    786.
## 7 Afghanistan Asia      1982   39.9 12881816    978.
## 8 Afghanistan Asia      1987   40.8 13867957    852.
## 9 Afghanistan Asia      1992   41.7 16317921    649.
## 10 Afghanistan Asia      1997   41.8 22227415    635.
## # ... with 1,694 more rows
```

```
# load relevant libraries
```

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
```

```
## v ggplot2 3.4.1      v purrr  0.3.5
```

```
## v tibble  3.1.8      v dplyr  1.1.0
```

```
## v tidyr   1.2.1      v stringr 1.5.0
```

```
## v readr   2.1.3      v forcats 0.5.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()     masks stats::lag()
```

```
library(ggplot2)
```

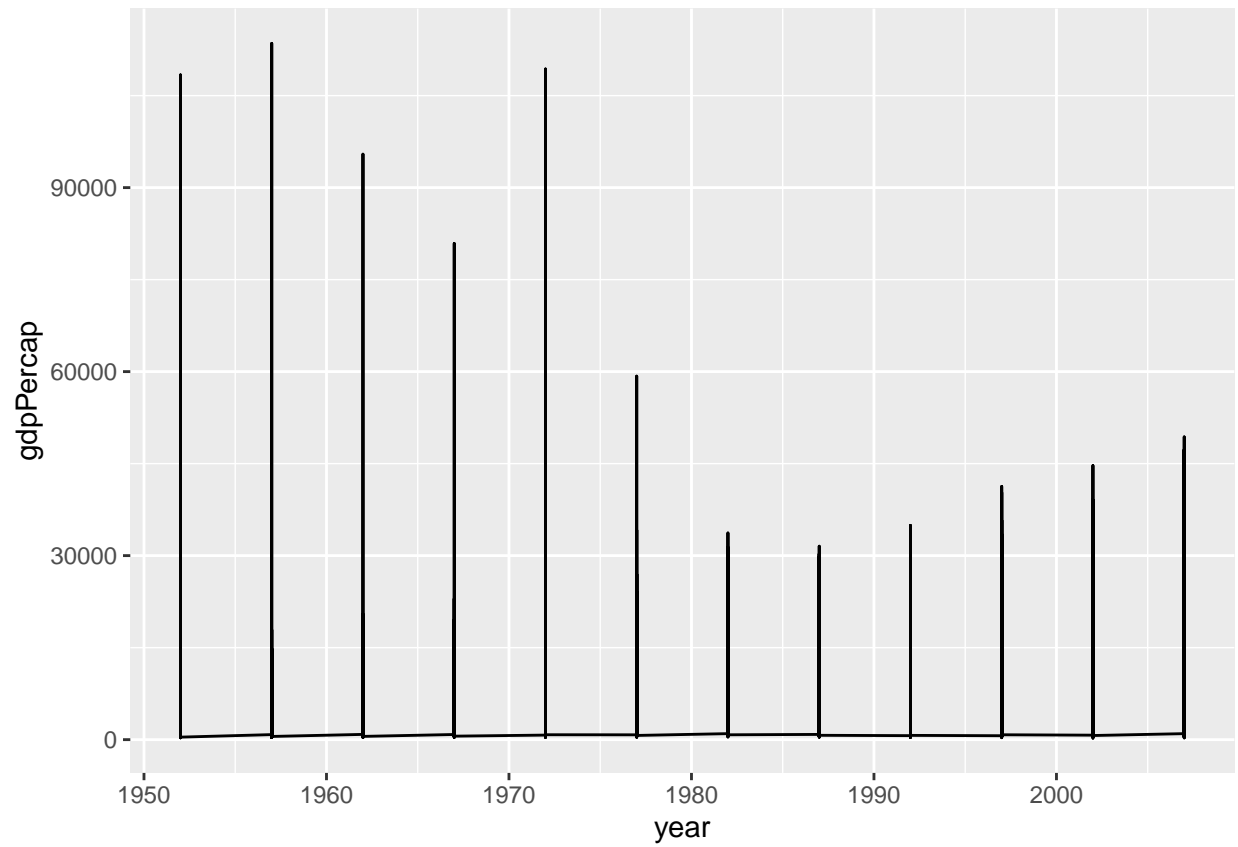
```
library(socviz)
```

```
library(gapminder)
```

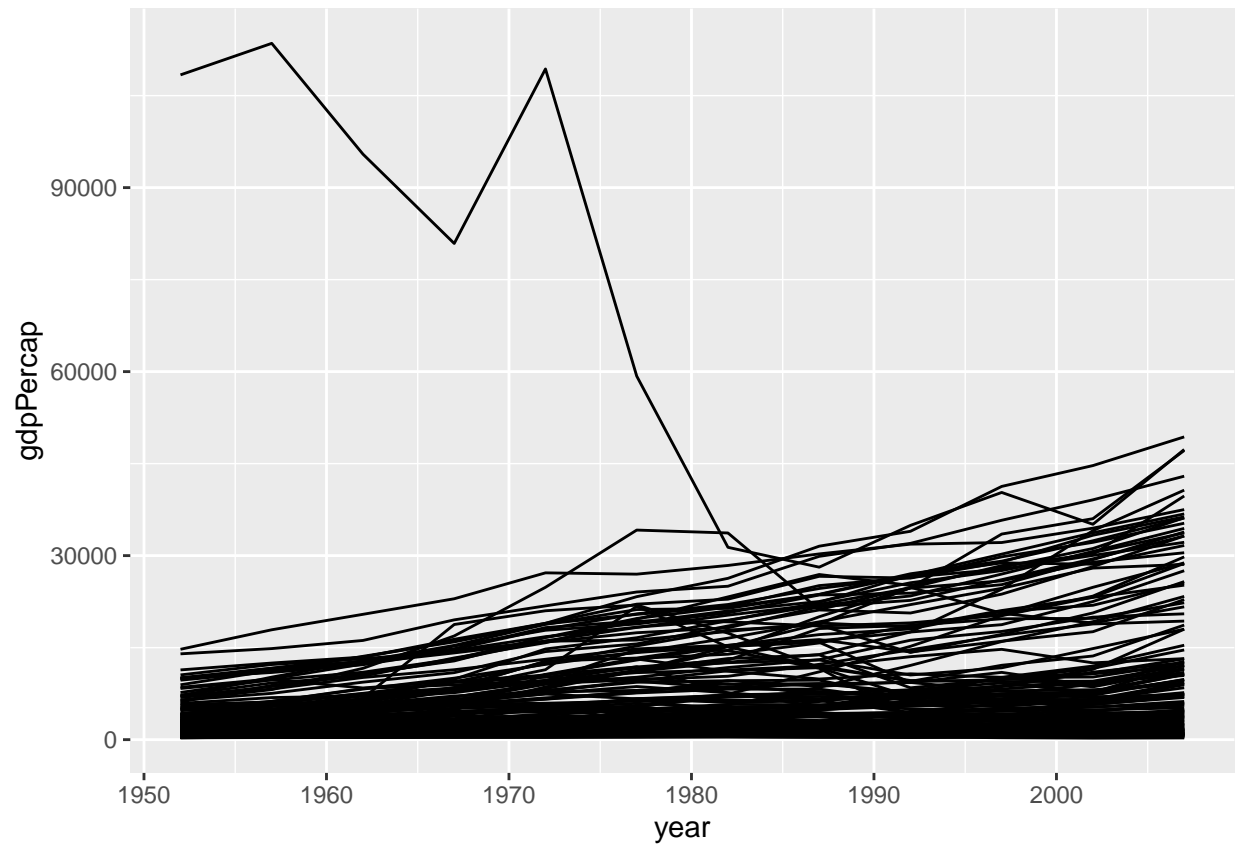
```
# try plotting our gdp data by year
```

```
p <- ggplot(data = gapminder,
            mapping = aes(x = year, y = gdpPercap))
```

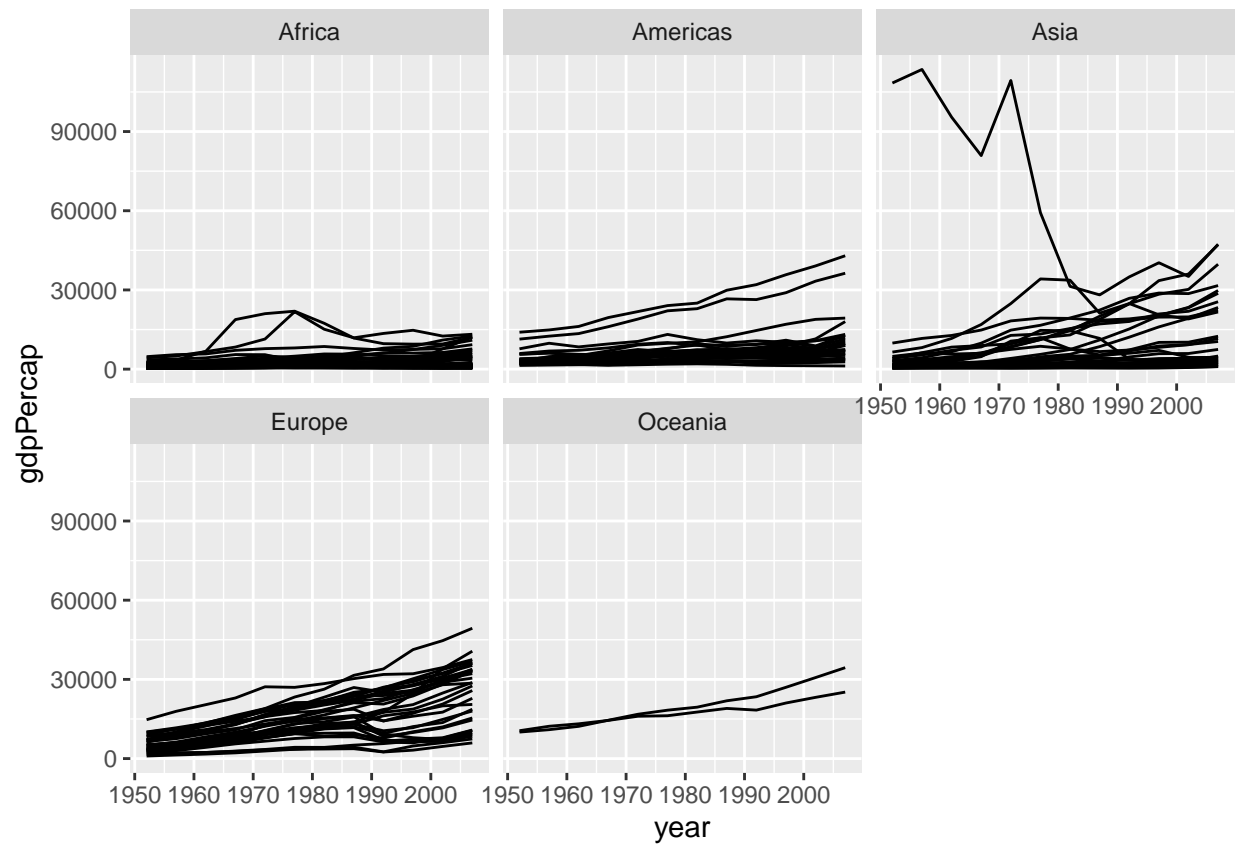
```
p + geom_line()
```



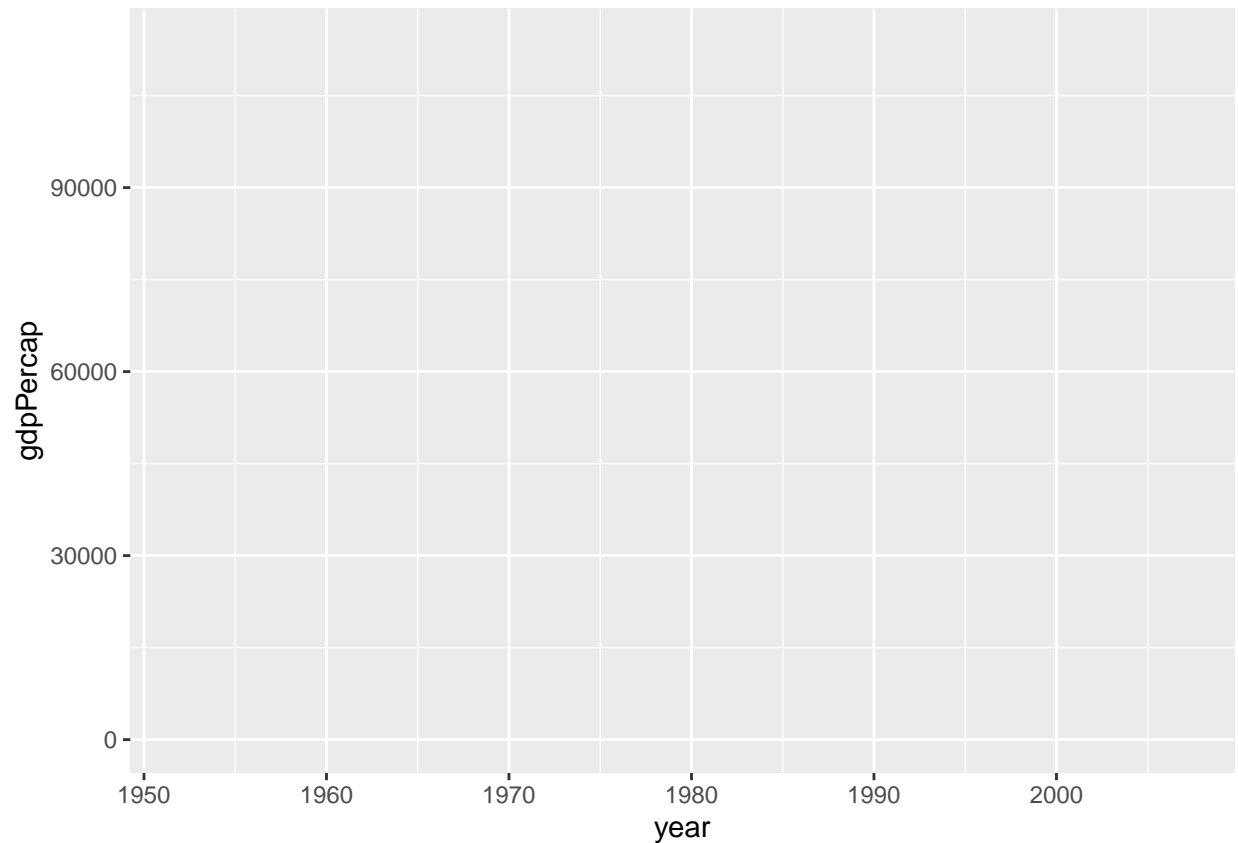
```
# try grouping our data first by country  
p + geom_line(aes(group = country))
```



```
# try breaking up our data according to continents by using the facet  
p + geom_line(aes(group = country)) + facet_wrap(~continent)
```



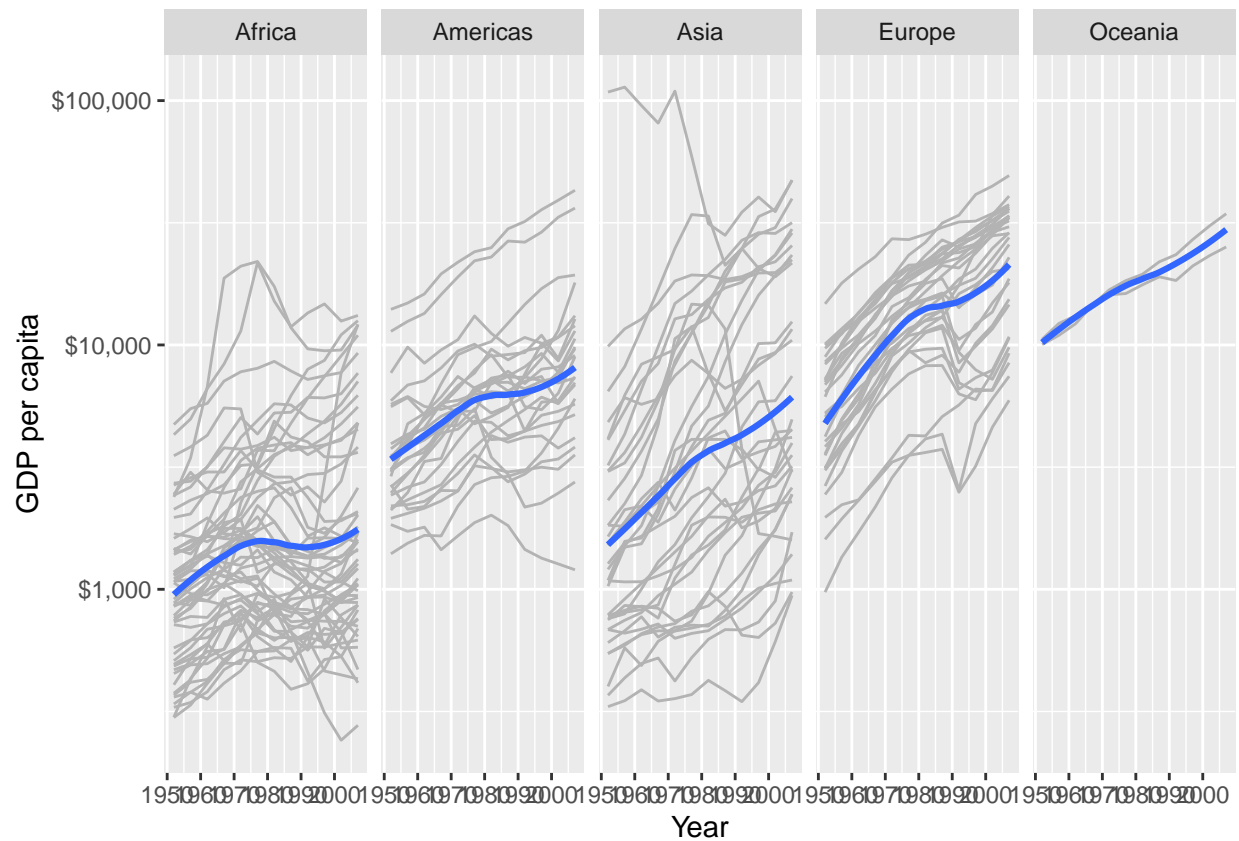
```
# improving our plot
ggplot(data=gapminder, mapping=aes(x=year, y=gdpPercap))
```



```
# adding gray colour to lines, adding trend line, make y axis logarithmic, change scale to dollars, add
p + geom_line(color='gray70', aes(group = country)) +
  geom_smooth(size=1.1,method='loess',se=FALSE) +
  scale_y_log10(labels=scales::dollar) +
  facet_wrap(~continent,ncol=5) +
  labs(x = 'Year',
       y = 'GDP per capita',
       Title = 'GDP per capita on Five Continents')
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

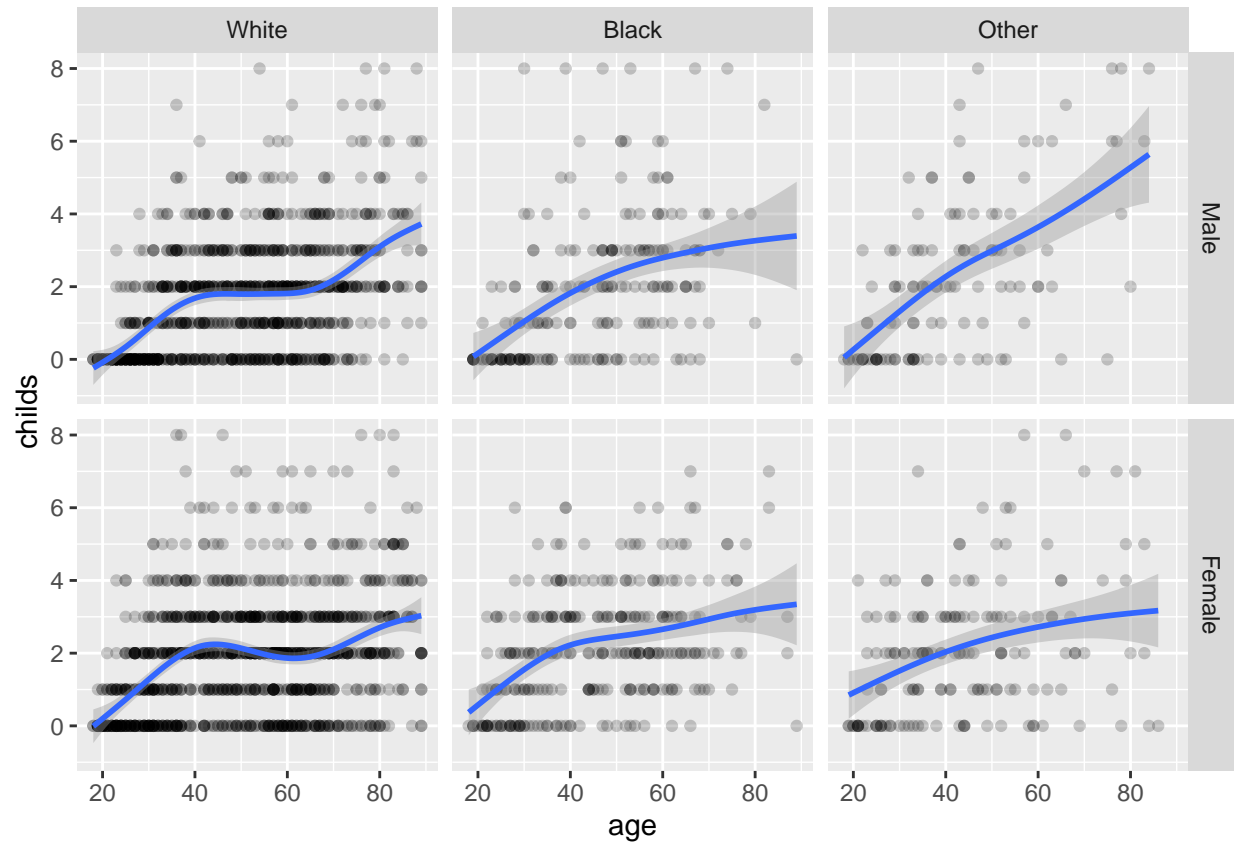


```
# using facet grid to split data by 2 variables
p <- ggplot(data=gss_sm, mapping = aes(x=age, y=childs))
p + geom_point(alpha=0.2) + geom_smooth() + facet_grid(sex~race)
```

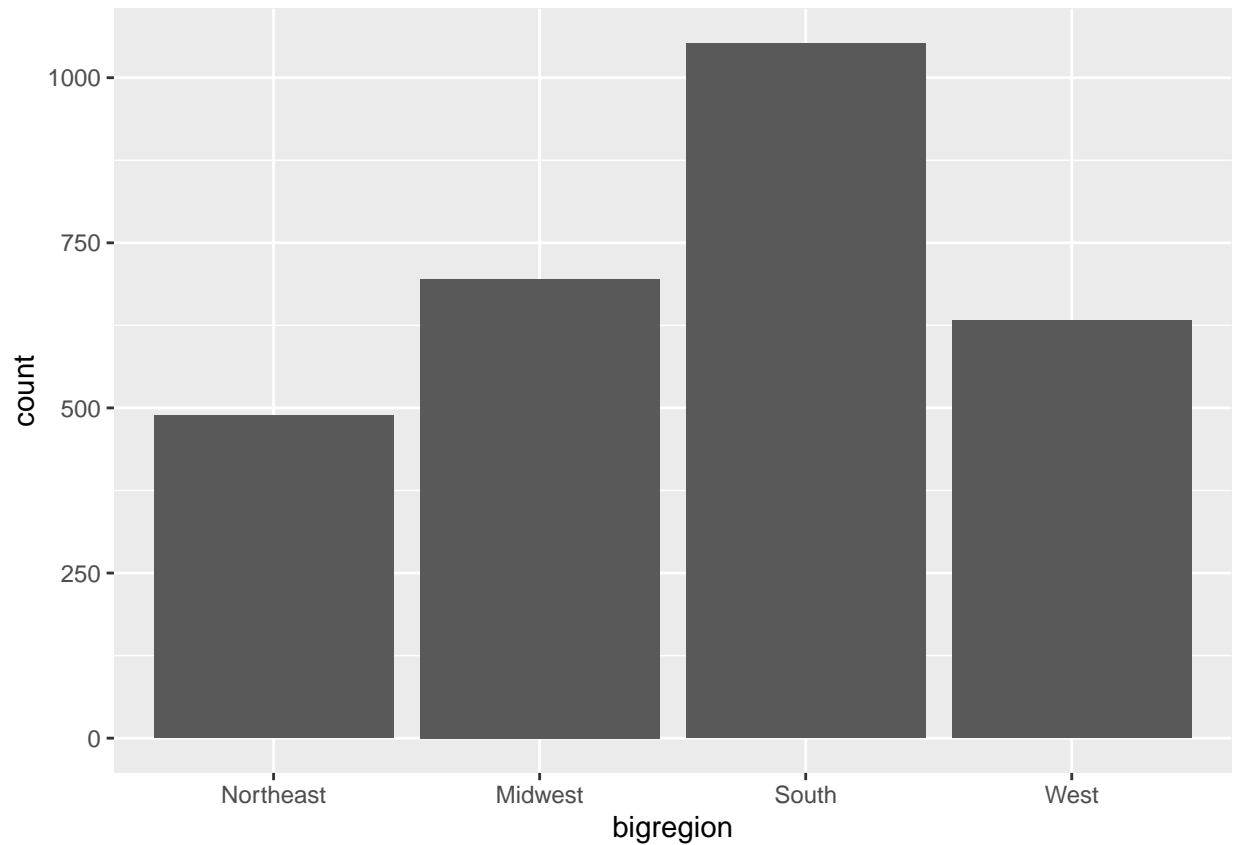
```
## 'geom_smooth()' using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 18 rows containing non-finite values ('stat_smooth()').
```

```
## Warning: Removed 18 rows containing missing values ('geom_point()').
```

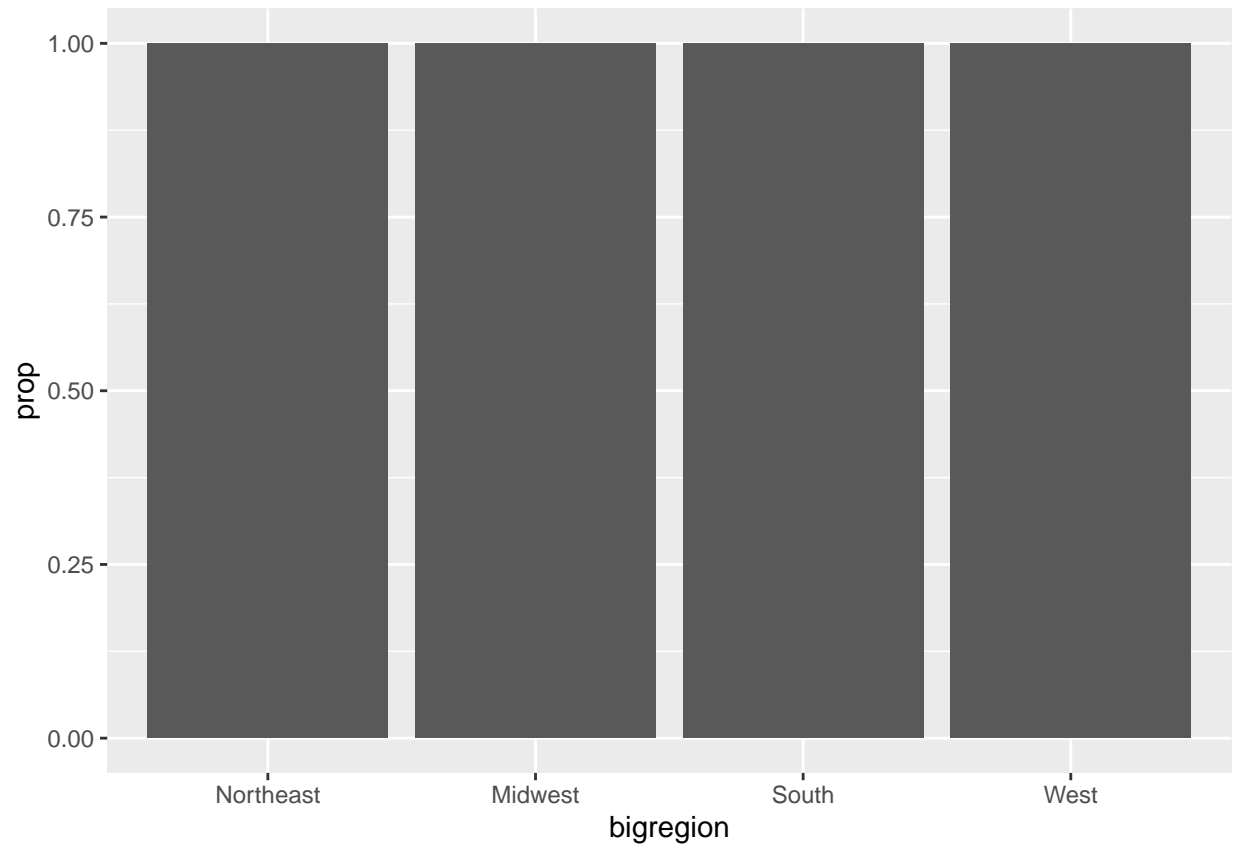


```
# exploring stat functions - default stat is count
p <- ggplot(data=gss_sm, mapping = aes(x = bigregion))
p + geom_bar()
```

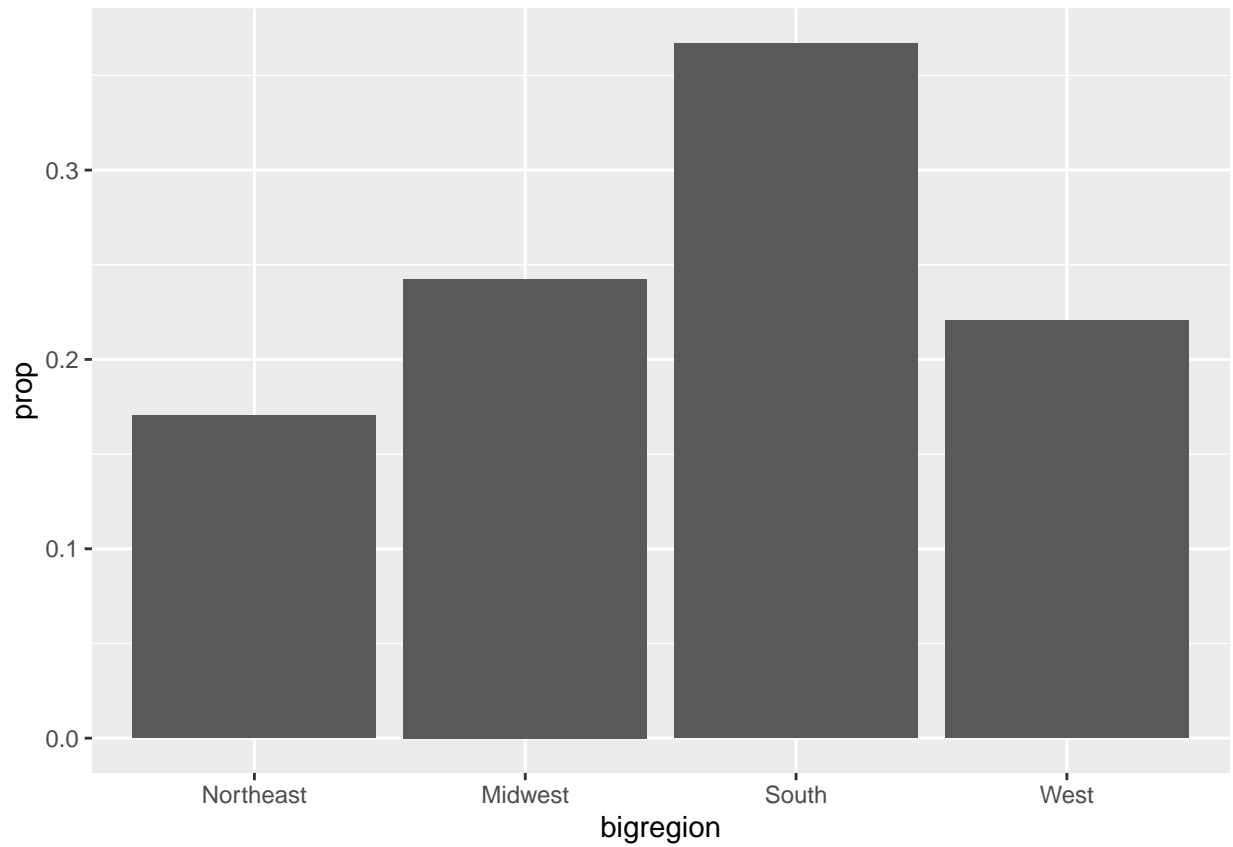


```
# try using another stat - proportion  
p + geom_bar(mapping = aes(y = ..prop..))
```

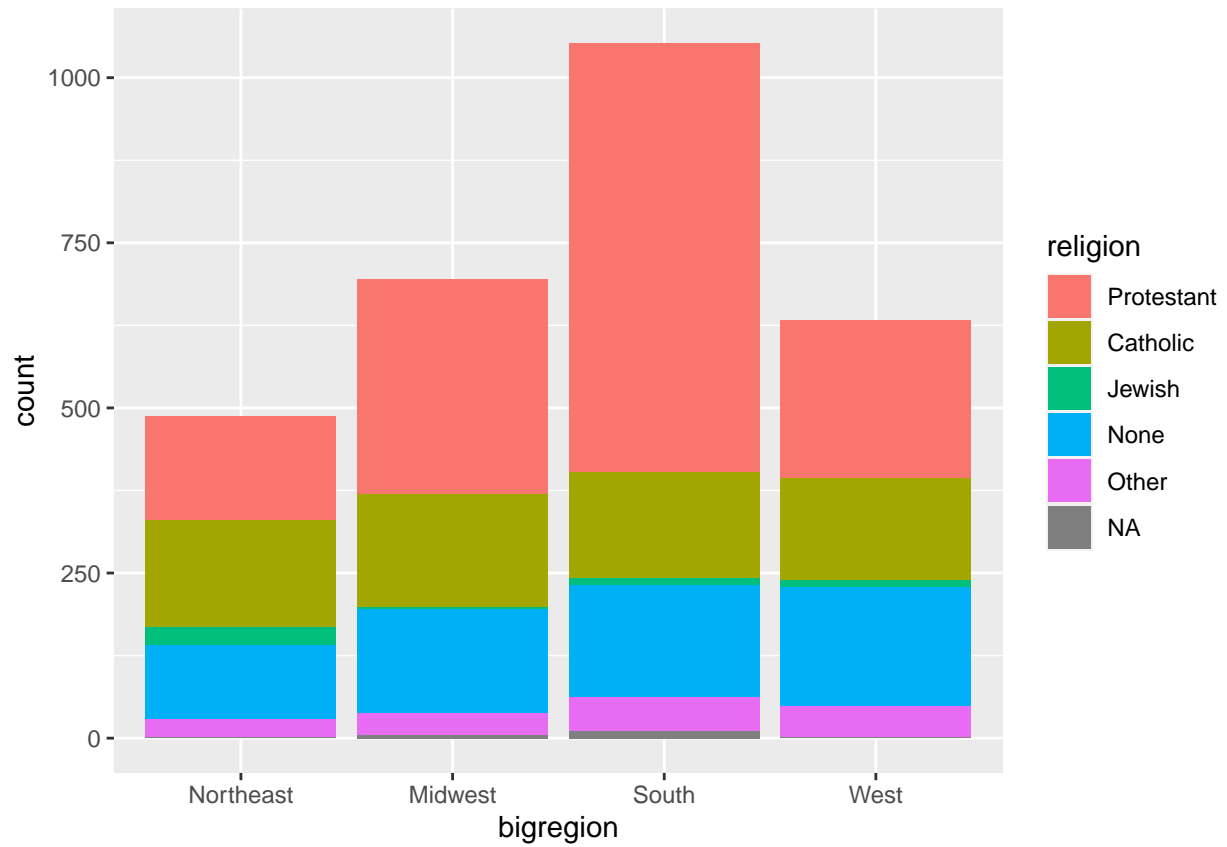
```
## Warning: The dot-dot notation ('..prop..') was deprecated in ggplot2 3.4.0.  
## i Please use 'after_stat(prop)' instead.
```

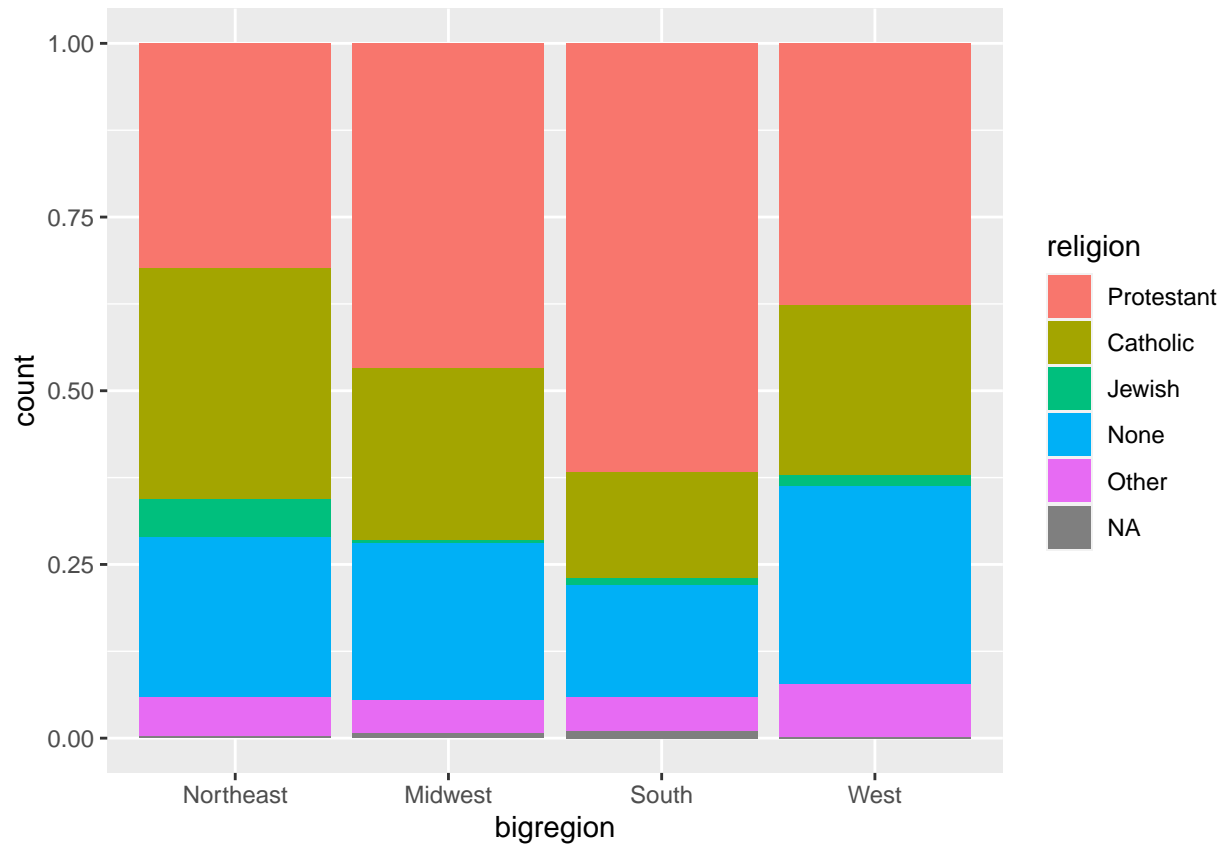
```
# need to create dummy group for grouping.  
p + geom_bar(mapping = aes(y = ..prop.., group = 1))
```



```
# frequency plots  
p <- ggplot(data = gss_sm, mapping = aes(x = bigregion, fill = religion))  
p + geom_bar()
```

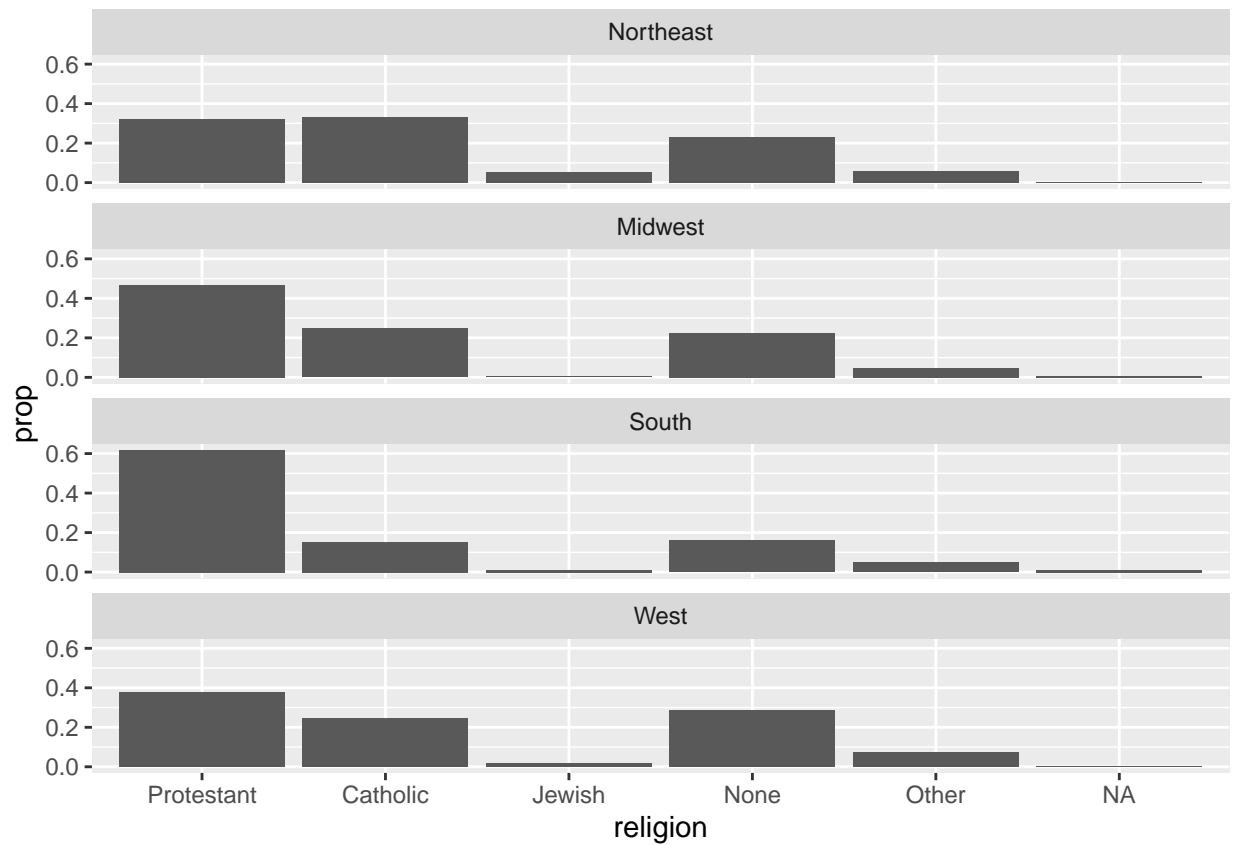


```
# want to compare proportions across groups  
p + geom_bar(position = 'fill')
```



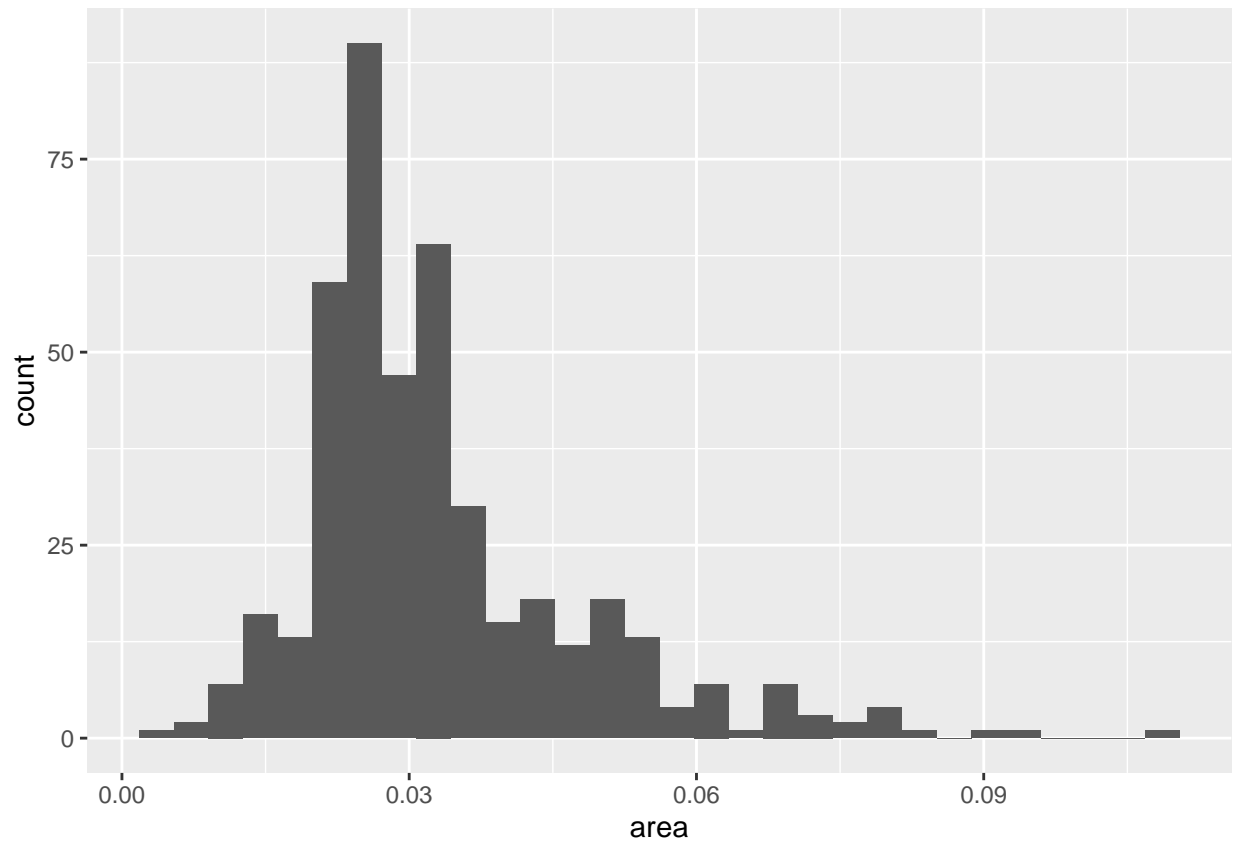
```
# now we can compare proportions across groups, but we can't see relative size respect to overall total
# faceting allows us to see proportional breakdown by region
p <- ggplot(data = gss_sm, mapping = aes(x = religion))

p + geom_bar(position = 'dodge', mapping = aes(y = ..prop.., group =
bigregion)) + facet_wrap(~bigregion, ncol = 1)
```

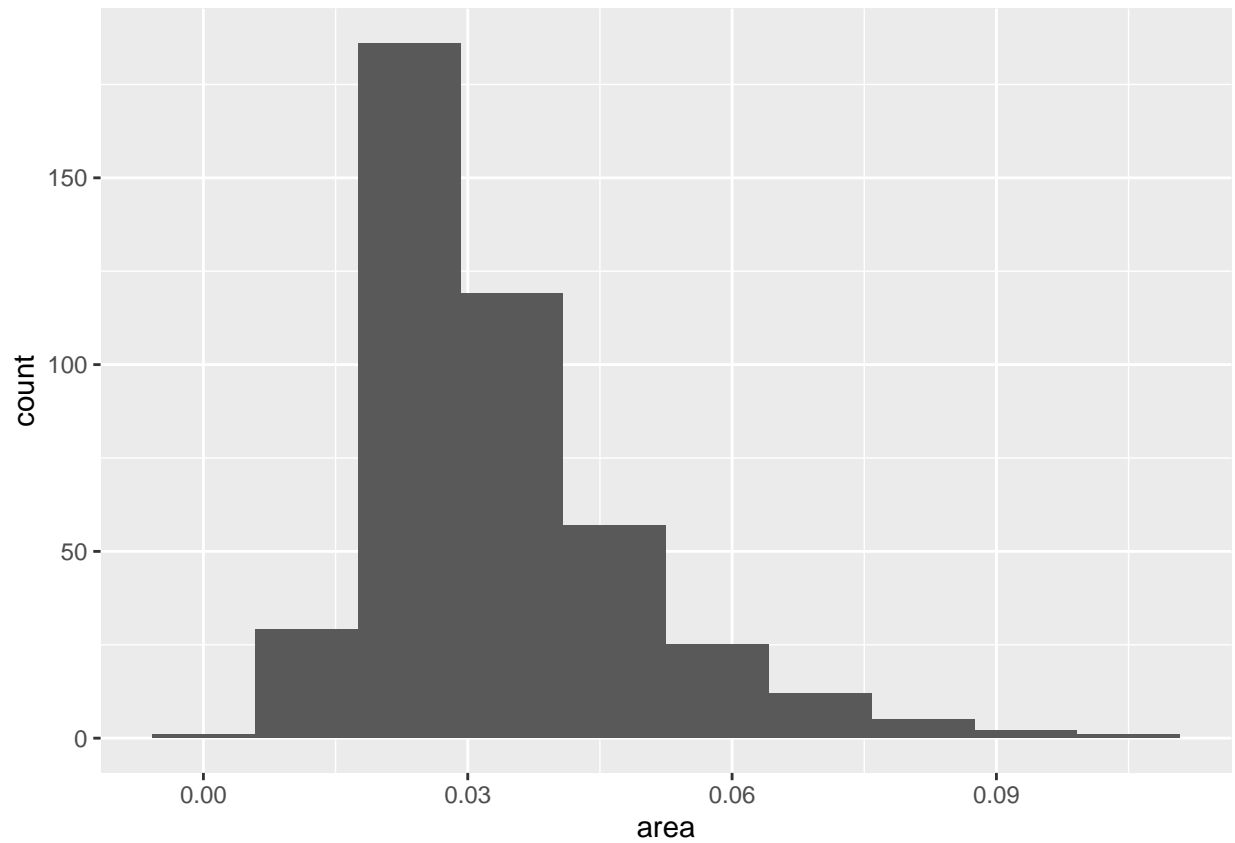


```
# explore histograms
p <- ggplot(data = midwest, mapping = aes(x = area))
p + geom_histogram()
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



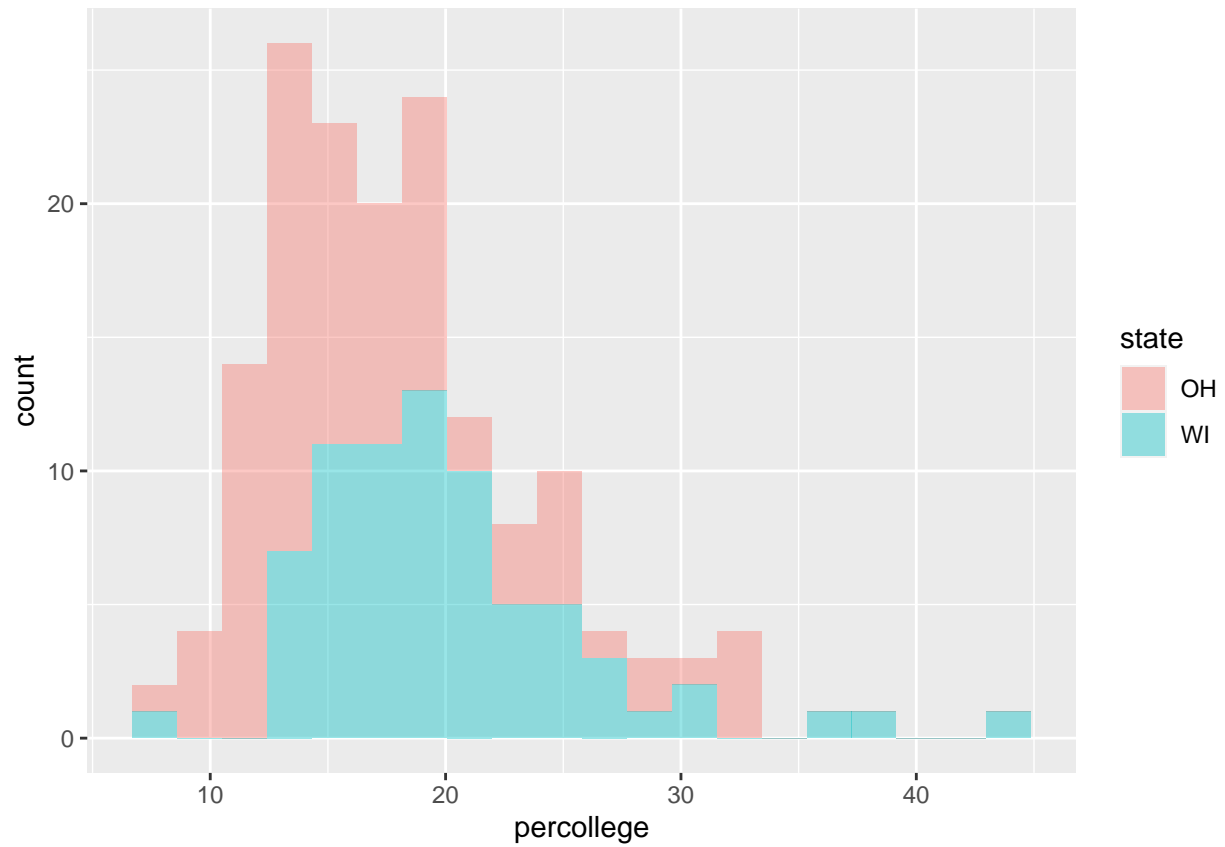
```
# we can also manually set the number of bins  
p <- ggplot(data = midwest, mapping = aes(x = area))  
p + geom_histogram(bins = 10)
```



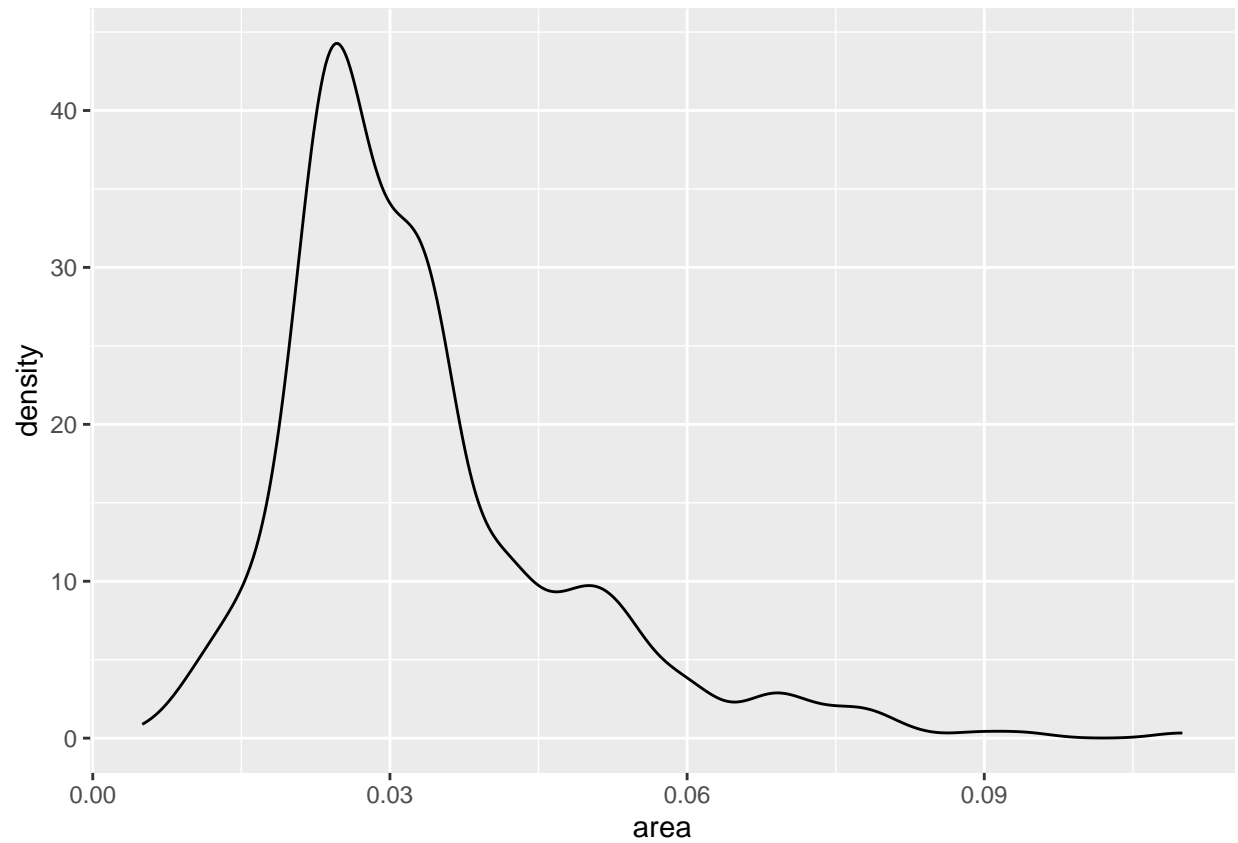
```
# subset our data by using a character vector of 2 states
oh_wi <- c("OH", "WI")

p <- ggplot(data = subset(midwest, subset = state %in% oh_wi),
mapping = aes(x = percollege, fill = state))

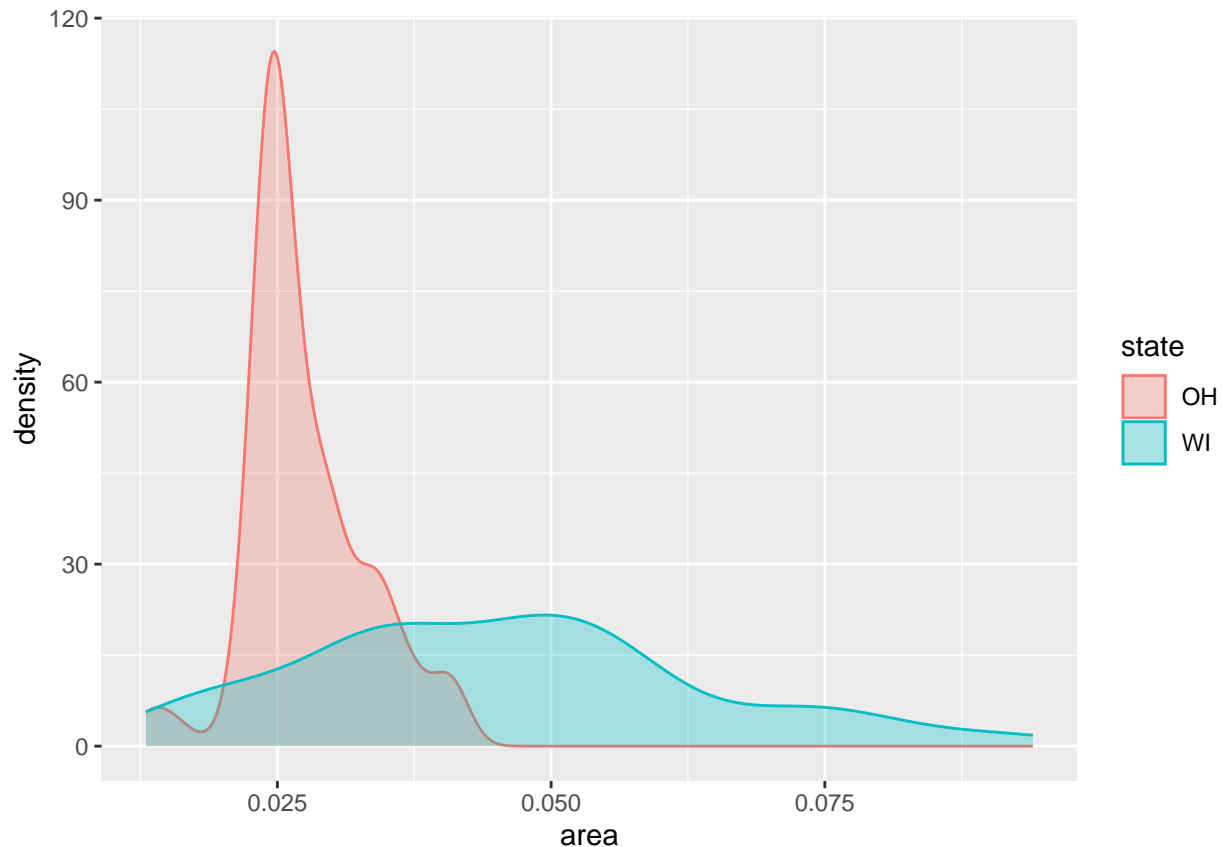
p + geom_histogram(alpha = 0.4, bins = 20)
```



```
# using a density plot for continuous variables  
p <- ggplot(data = midwest, mapping = aes(x = area))  
p + geom_density()
```

```
# subset the data for our density plot
p <- ggplot(data = subset(midwest, subset = state %in% oh_wi),
            mapping = aes(x = area, fill = state, color = state))
p + geom_density(alpha = 0.3)
```



- What information can we learn from this visualization?
- Is this an example of objective, neutral data visualization? Why or why not?
- second visual uses less emotional language and is just static and still
- second visual is easier to interpret objectively
- first visual is not neutral, but is any data visualization neutral? No.
- Data visualizations can be factual without being neutral

```
# explore organ dataset
organdata
```

```
## # A tibble: 238 x 21
##   country year   donors  pop pop_d~1  gdp gdp_lag health healt~2 pubhe~3
##   <chr>   <date>   <dbl> <int>  <dbl> <int>  <int>  <dbl>  <dbl>  <dbl>
## 1 Austral~ NA      NA    17065  0.220 16774  16591  1300   1224    4.8
## 2 Austral~ 1991-01-01 12.1  17284  0.223 17171  16774  1379   1300    5.4
## 3 Austral~ 1992-01-01 12.4  17495  0.226 17914  17171  1455   1379    5.4
## 4 Austral~ 1993-01-01 12.5  17667  0.228 18883  17914  1540   1455    5.4
## 5 Austral~ 1994-01-01 10.2  17855  0.231 19849  18883  1626   1540    5.4
## 6 Austral~ 1995-01-01 10.2  18072  0.233 21079  19849  1737   1626    5.5
## 7 Austral~ 1996-01-01 10.6  18311  0.237 21923  21079  1846   1737    5.6
## 8 Austral~ 1997-01-01 10.3  18518  0.239 22961  21923  1948   1846    5.7
## 9 Austral~ 1998-01-01 10.5  18711  0.242 24148  22961  2077   1948    5.9
```

```
## 10 Austral~ 1999-01-01 8.67 18926 0.244 25445 24148 2231 2077 6.1
## # ... with 228 more rows, 11 more variables: roads <dbl>, cerebvas <int>,
## # assault <int>, external <int>, txp_pop <dbl>, world <chr>, opt <chr>,
## # consent_law <chr>, consent_practice <chr>, consistent <chr>, ccode <chr>,
## # and abbreviated variable names 1: pop_dens, 2: health_lag, 3: pubhealth
```

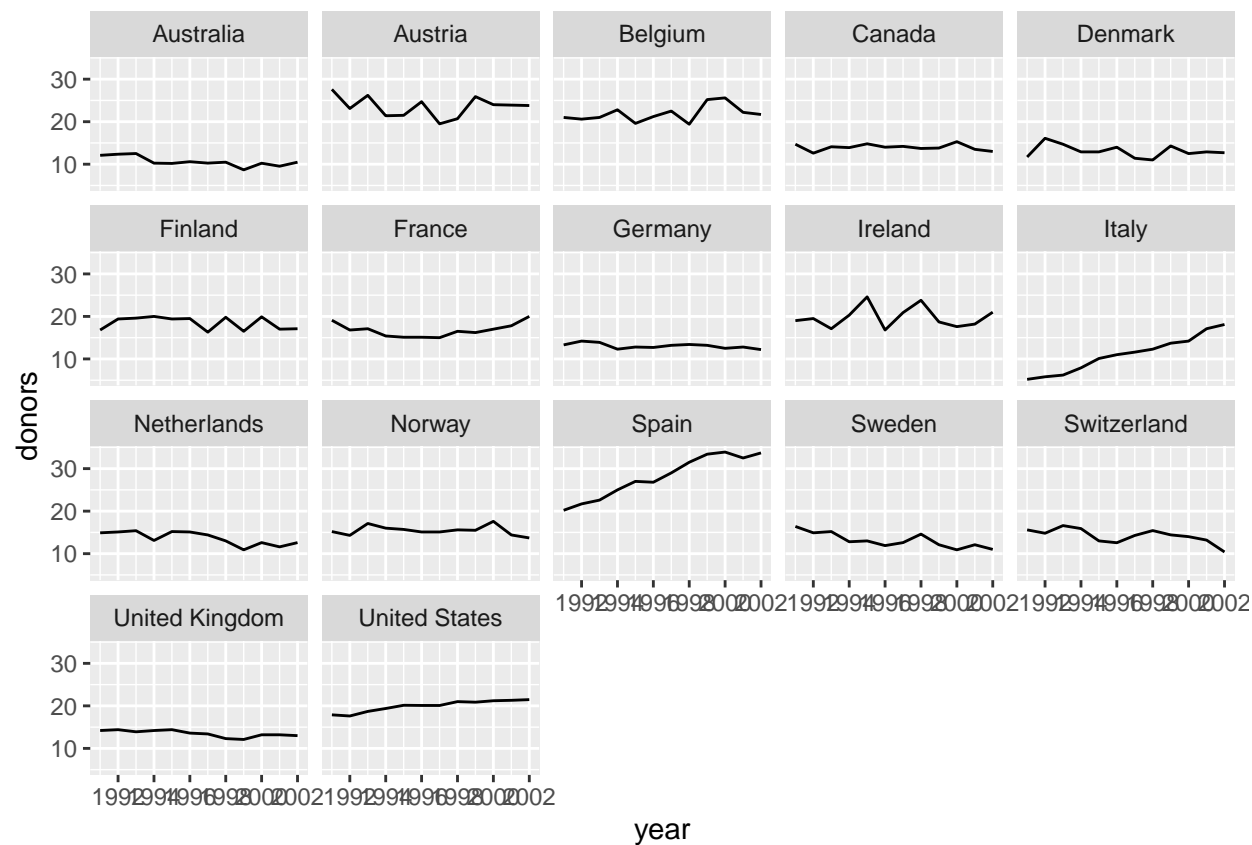
```
# we can pipe data
organdata %>% select(1:6) |> sample_n(size = 10)
```

```
## # A tibble: 10 x 6
##   country      year      donors    pop pop_dens    gdp
##   <chr>      <date>      <dbl>  <int>    <dbl> <int>
## 1 Austria    2001-01-01    23.9   8030     9.58  28457
## 2 Sweden     2001-01-01    12.1   8896     1.98  26902
## 3 Austria    1991-01-01    27.6   7755     9.25  19860
## 4 Belgium    1995-01-01    19.6  10137    30.6   21679
## 5 United States 1997-01-01    20.1 272647     2.83  30283
## 6 France      NA           NA    56709    10.3   18162
## 7 Belgium    1996-01-01    21.2  10157    30.7   22152
## 8 Canada     2001-01-01    13.5  31111     0.312 29235
## 9 France      NA           NA      NA     NA      NA
## 10 Spain     1994-01-01     25   39166     7.74  15024
```

```
# recall groups and facets. we can look at number of donors by years by country
p <- ggplot(data = organdata, mapping = aes(x = year, y = donors))

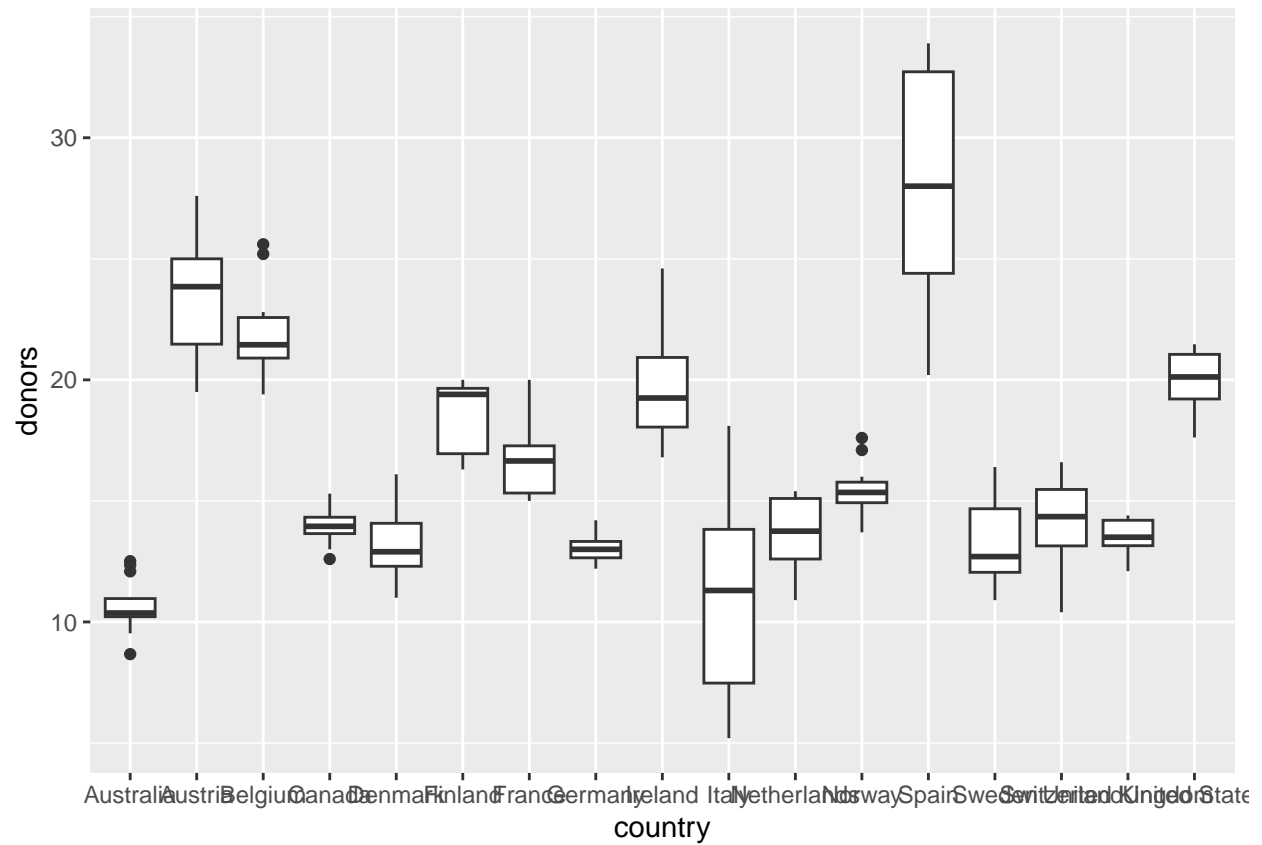
p + geom_line(aes(group = country)) + facet_wrap(~country)
```

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



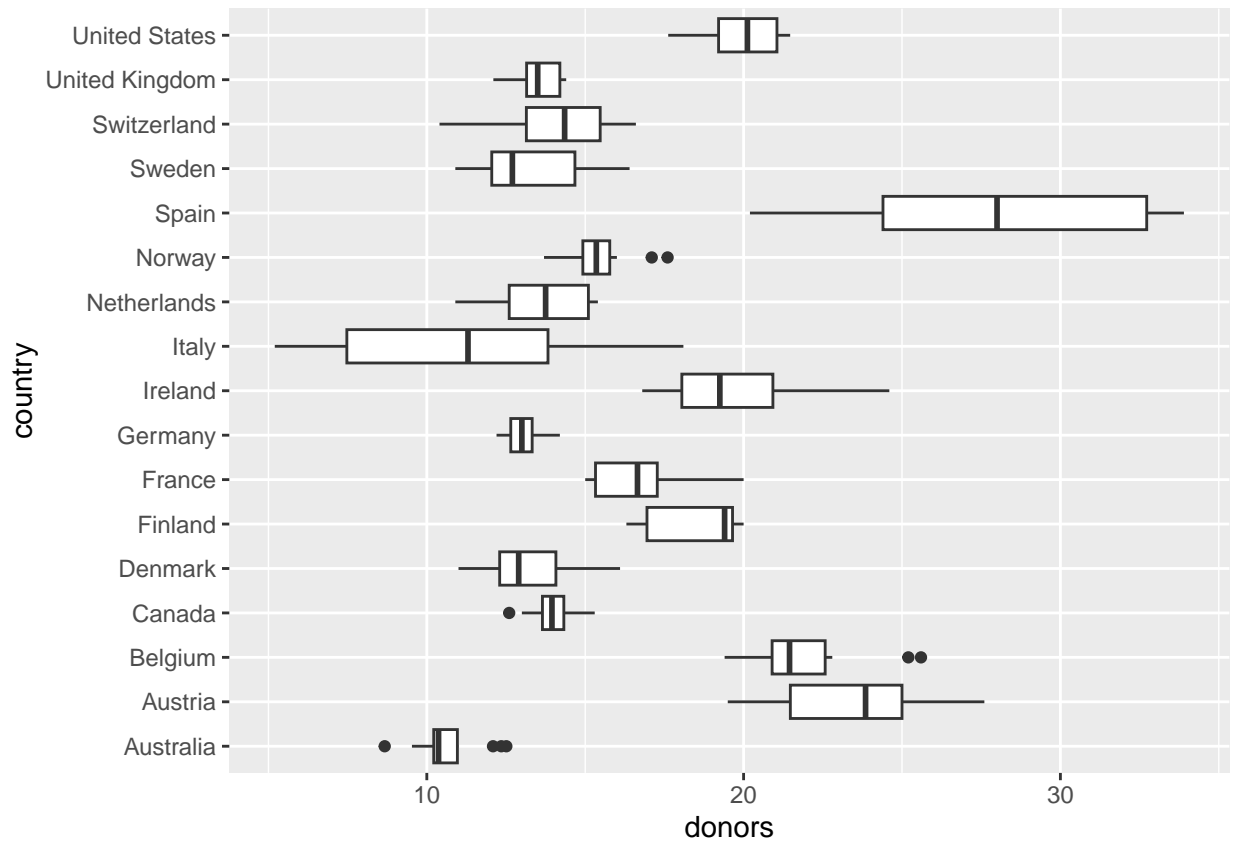
```
# make a boxplot
p <- ggplot(data = organdata, mapping = aes(x = country , y = donors))
p + geom_boxplot()
```

```
## Warning: Removed 34 rows containing non-finite values ('stat_boxplot()').
```



```
# improve our boxplot with coord_flip
p + geom_boxplot() + coord_flip()
```

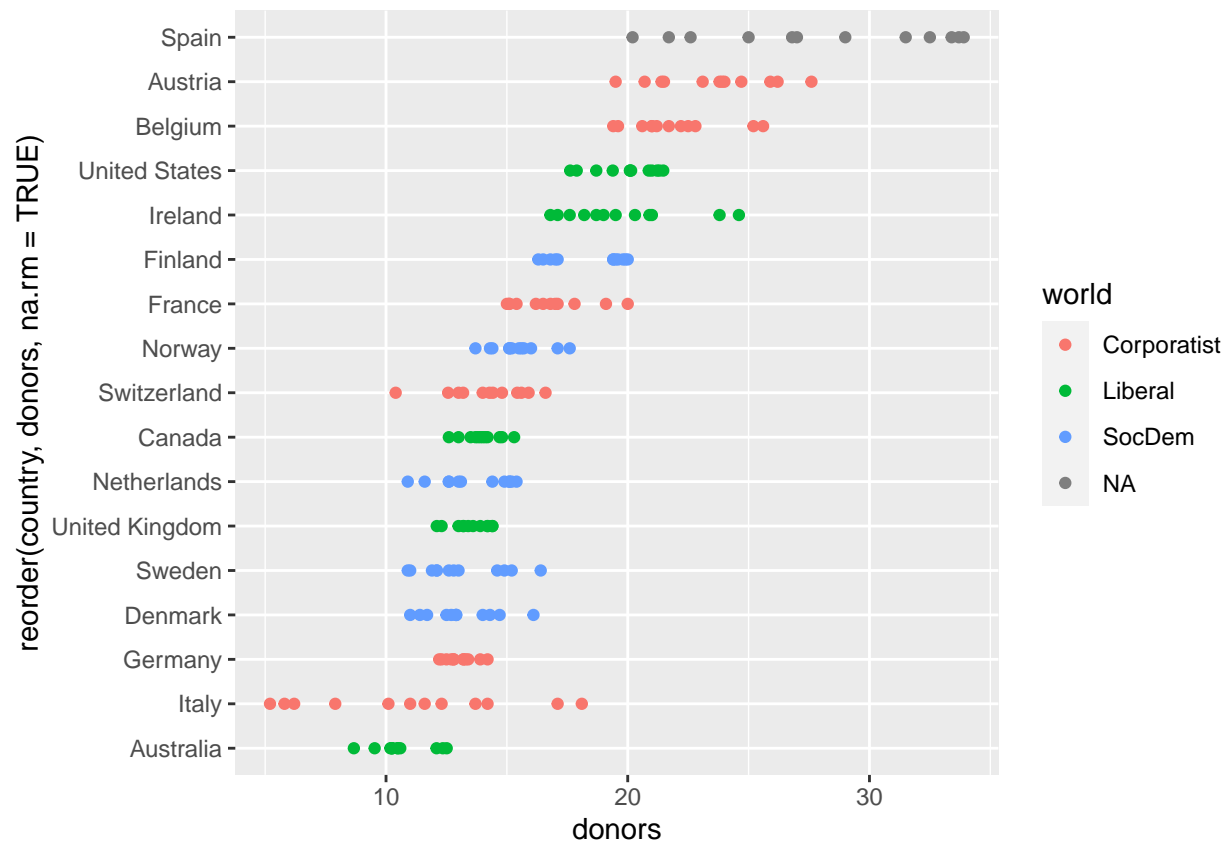
```
## Warning: Removed 34 rows containing non-finite values ('stat_boxplot()').
```



```
# improve our plot further by listing the countries from high to low donor
p <- ggplot(data = organdata, mapping = aes(x = reorder(country,
                                                    donors, na.rm = TRUE ), y = donors))

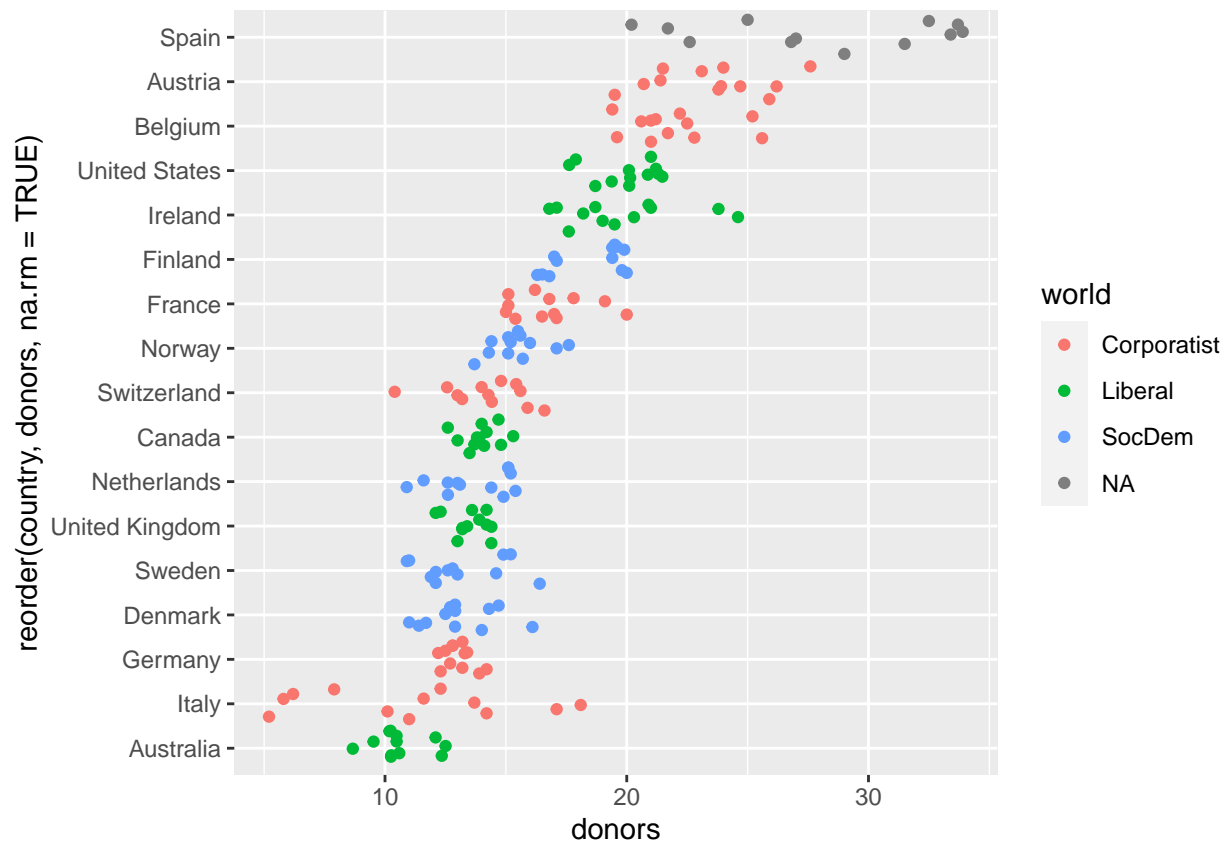
# if we have fewer data observations, it can be helpful to show individual observations
p <- ggplot(data = organdata, mapping = aes(x = reorder(country, donors,
                                                    na.rm=TRUE), y = donors, color = world))
p + geom_point() + coord_flip()
```

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



```
# we can use geom_jitter o randomly and slightly nudge each observation for better visualization
p + geom_jitter() + coord_flip()
```

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



```
# processing data with a pipeline
```

```
by_country <- organdata |>
  group_by(consent_law, country) |>
  summarize_if(is.numeric, funs(mean, sd), na.rm = TRUE) |>
  ungroup()
```

```
## Warning: 'funs()' was deprecated in dplyr 0.8.0.
```

```
## i 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)
```

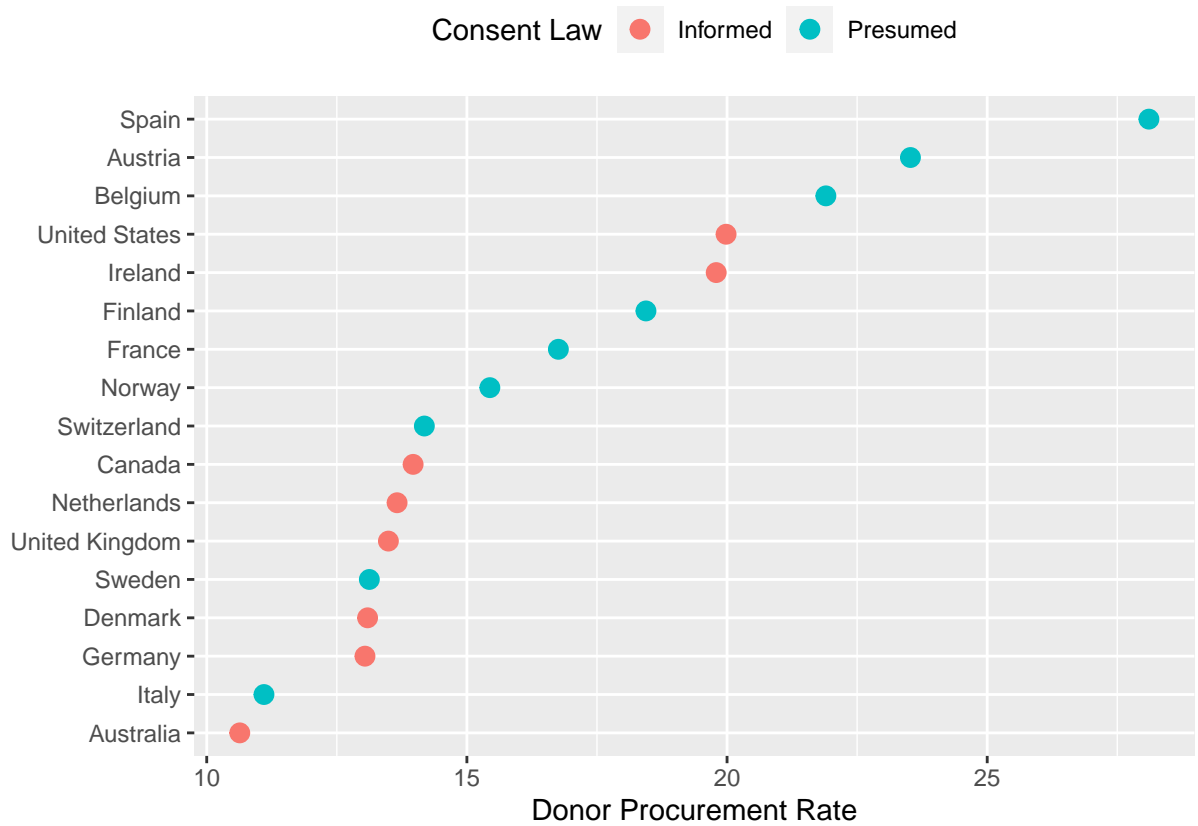
```
##
```

```
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```
# make a cleveland dot plot for categorical data
```

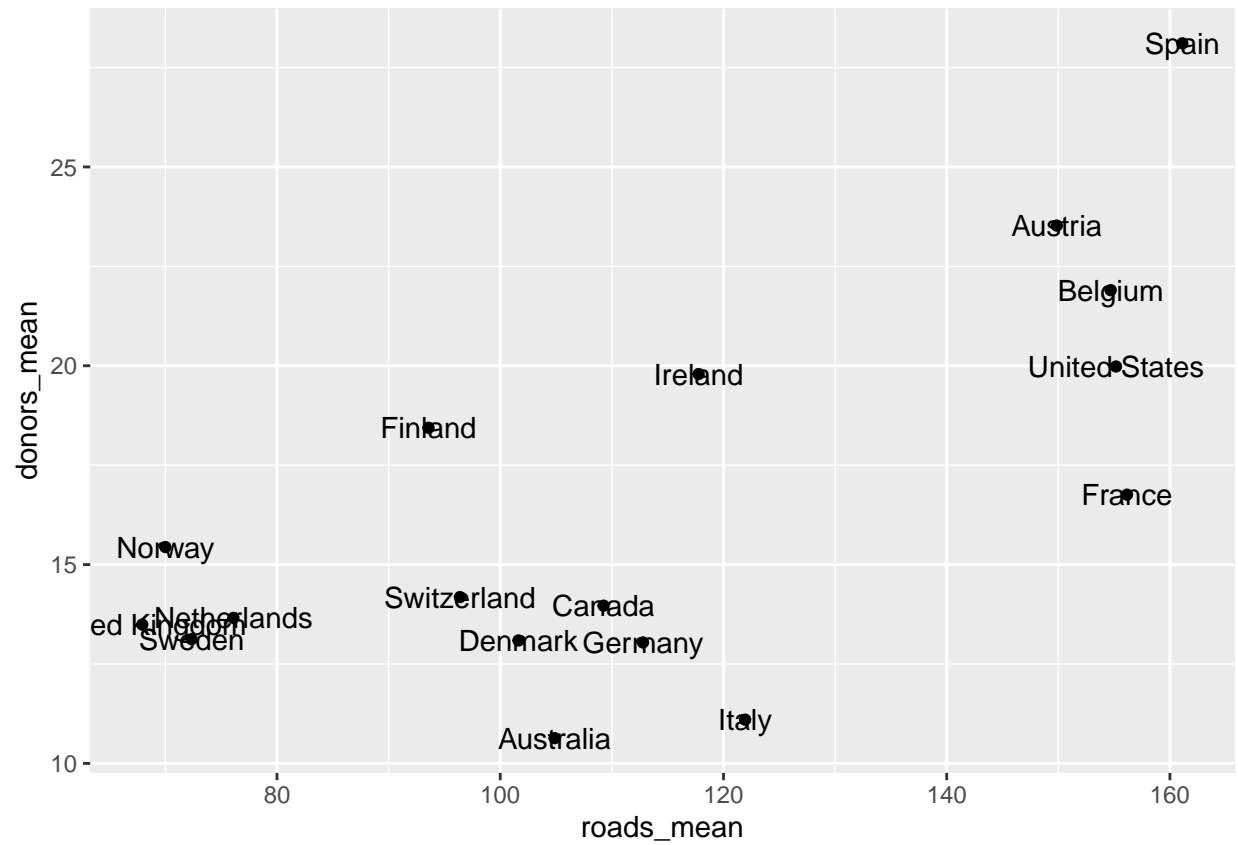
```
p <- ggplot(data = by_country, mapping = aes(x = donors_mean,
                                              y = reorder(country, donors_mean), color = consent_law))
```

```
p + geom_point(size=3) + labs(x = 'Donor Procurement Rate', y = '', color = 'Consent Law') + theme(legend.position = 'right')
```

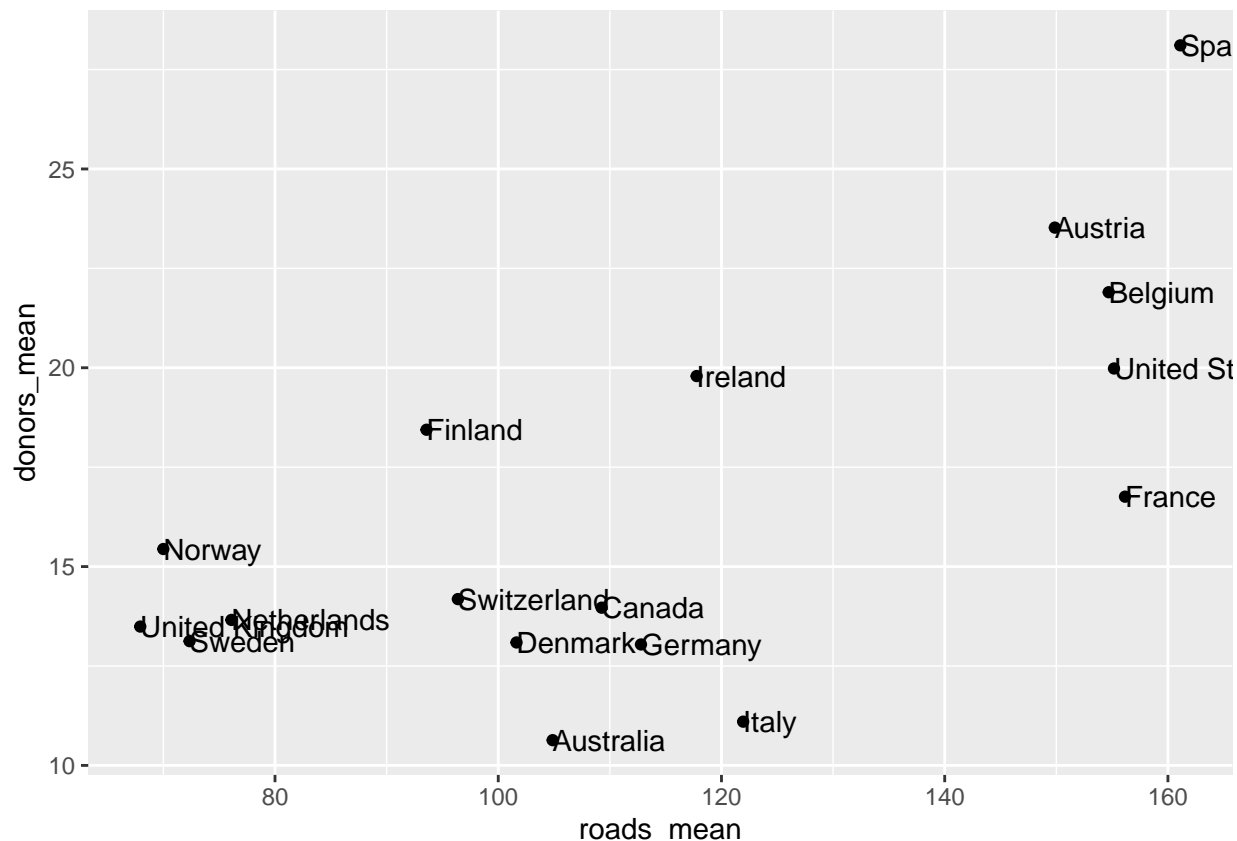



```
# plotting text
# adding data labels; we can add text to our plots with geom_text
p <- ggplot(data = by_country, mapping = aes(x = roads_mean,
                                              y = donors_mean))

p + geom_point() + geom_text(mapping = aes(label = country))
```



```
# we can adjust the position of our text
p + geom_point() + geom_text(mapping = aes(label = country), hjust = 0)
```



```
# for more flexible text formatting, we can use ggrepel
# load libraries
# install.packages('ggrepel')
library(ggrepel)

# we will use a new dataset about historical united states presidents
p <- ggplot(elections_historic, aes(x = popular_pct, y = ec_pct, label = winner_label))
p + 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 = "Winner's share of Popular Vote", y = "Winner's share of Electoral
College Votes", title = "Presidential Elections: Popular & Electoral
College Margins", subtitle = "1824 2016")
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbsToSbcs': dot
## substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbsToSbcs': dot
## substituted for <80>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
```

```

## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <99>

```

```

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <e2>

```

```

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning: ggrepel: 20 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Popular Vote' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <e2>

```

```

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Winner's share of Electoral' in 'mbcsToSbcs': dot
## substituted for <99>

```

Presidential Elections: Popular & Electoral College Margins

1824 2016

