Show the Right Numbers

ggplot IMPLEMENTS A GRAMMAR OF GRAPHICS

The grammar is a set of rules for how produce graphics from data, taking **pieces of data** and **mapping** them to **geometric objects** (like points and lines) **that have aesthetic attributes** (like position, color and size), together with further rules for **transforming the data if needed**, adjusting **scales**, or projecting the results onto a **coordinate system**.


Like other rules of syntax, the grammar limits what you can validly say, but it doesn't make what you say sensible or meaningful.

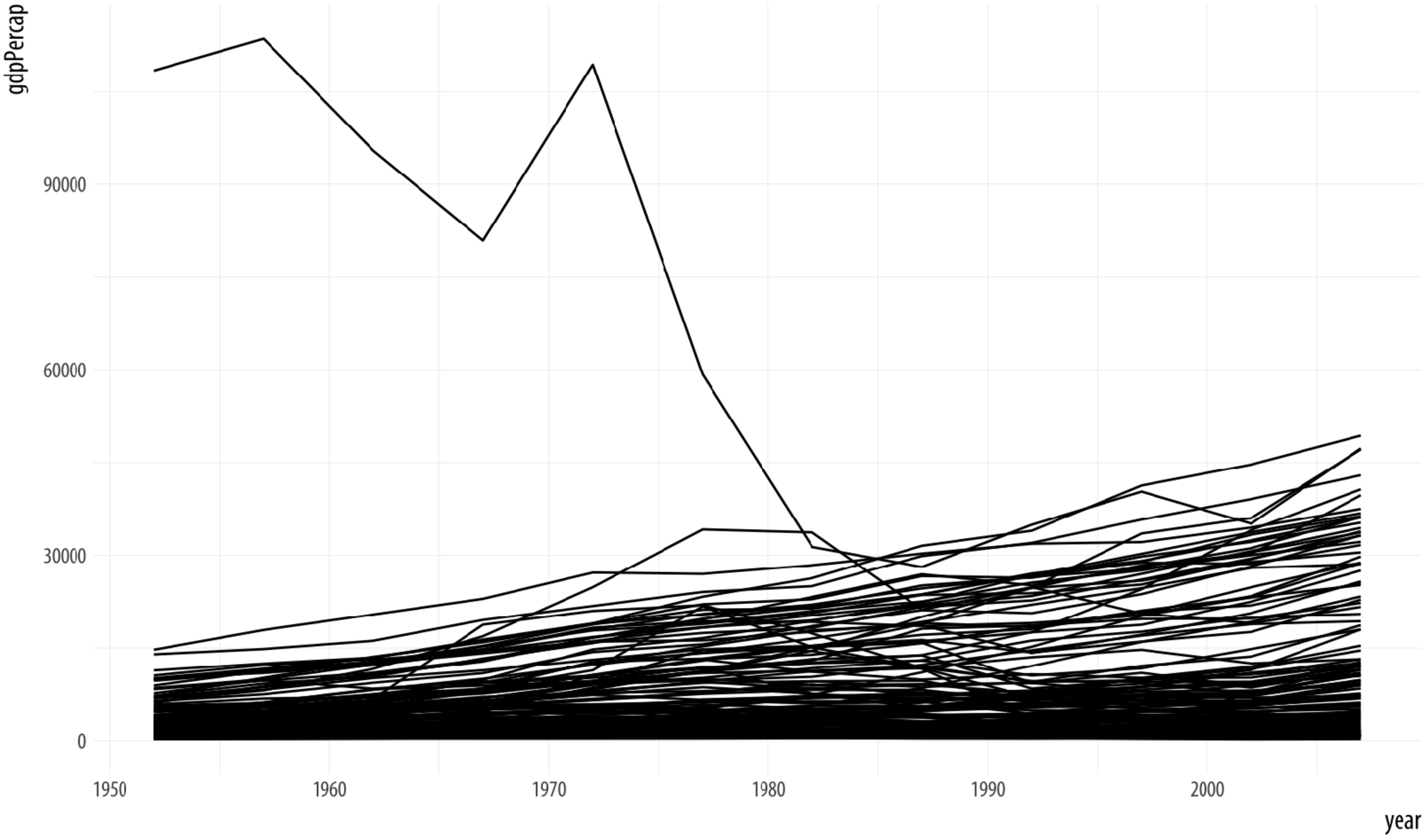
Grouped Data and the group **aesthetic**

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


gdpPercap
90000
60000
30000
0
1950
1960
1970
1980
1990
2000


0000
0
1950
1960
1970

```r
p <- ggplot(data = gapminder,
            mapping = aes(x = year,
                          y = gdpPercap))
p + geom_line(mapping = aes(group = country))
```

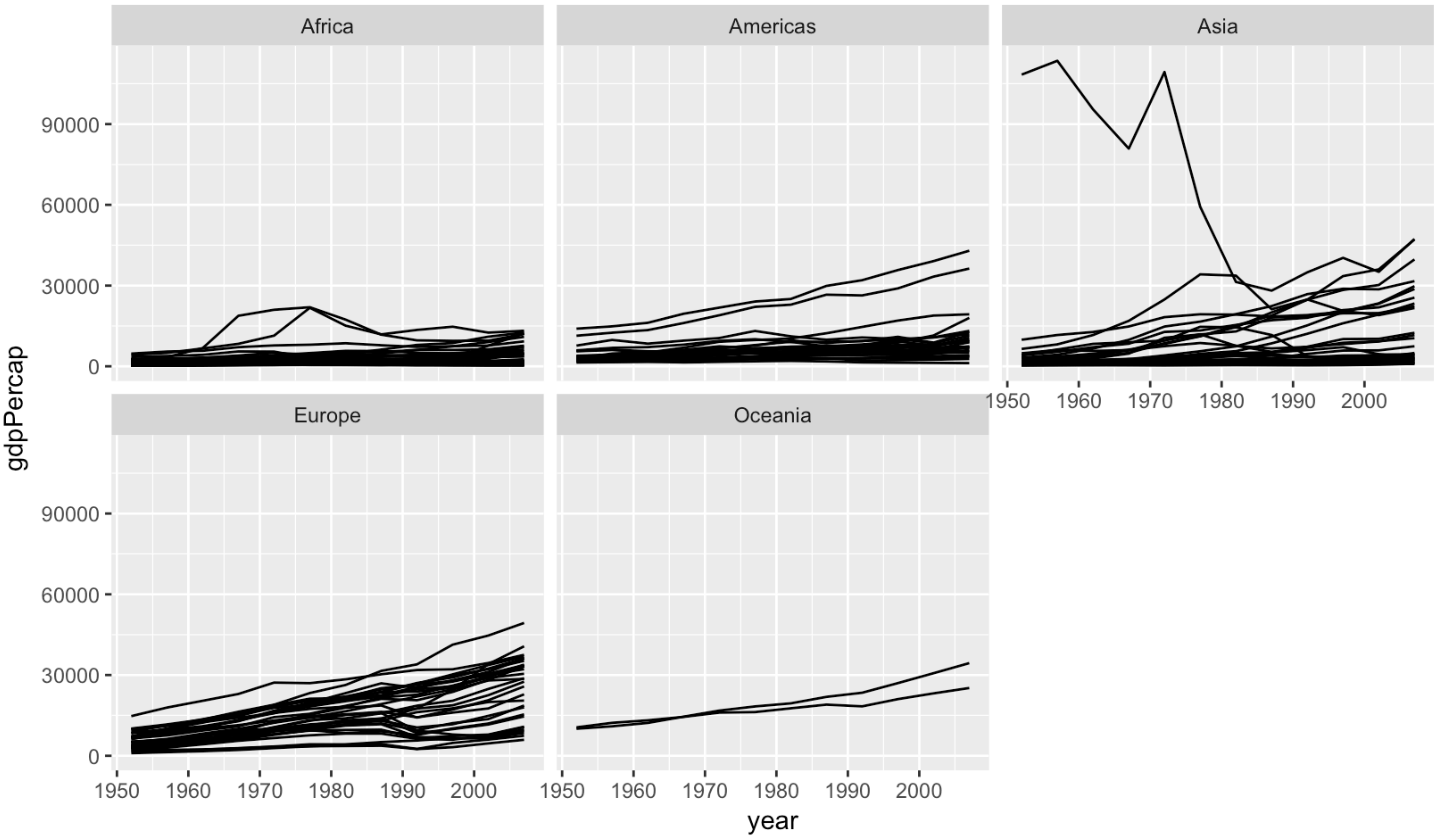

gdpPercap
90000
60000
30000
0
1950
1960
1970
1980
1990
2000
year

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

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

A facet is not a geom. It's a way of arranging geoms.

Facets use R's 'formula' syntax. Read the ~ as "on" or "by".


Africa
Americas
Asia
Europe
Oceania
gdpPercap
90000
60000
30000
0
1950
1960
1970
1980
1990
2000
year

```r
p + geom_line(color = "gray70",
              mapping = 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")
```

The labs() function lets you name labels, title, subtitle, etc.

GDP per capita on Five Continents
Africa
Americas
Asia
Europe
Oceania
$100,000
$10,000
$1,000
GDP per capita
1950 1960 1970 1980 1990 2000
Year

geoms CAN TRANSFORM DATA

gss_sm

A subset of General Social Survey Questions from 2016

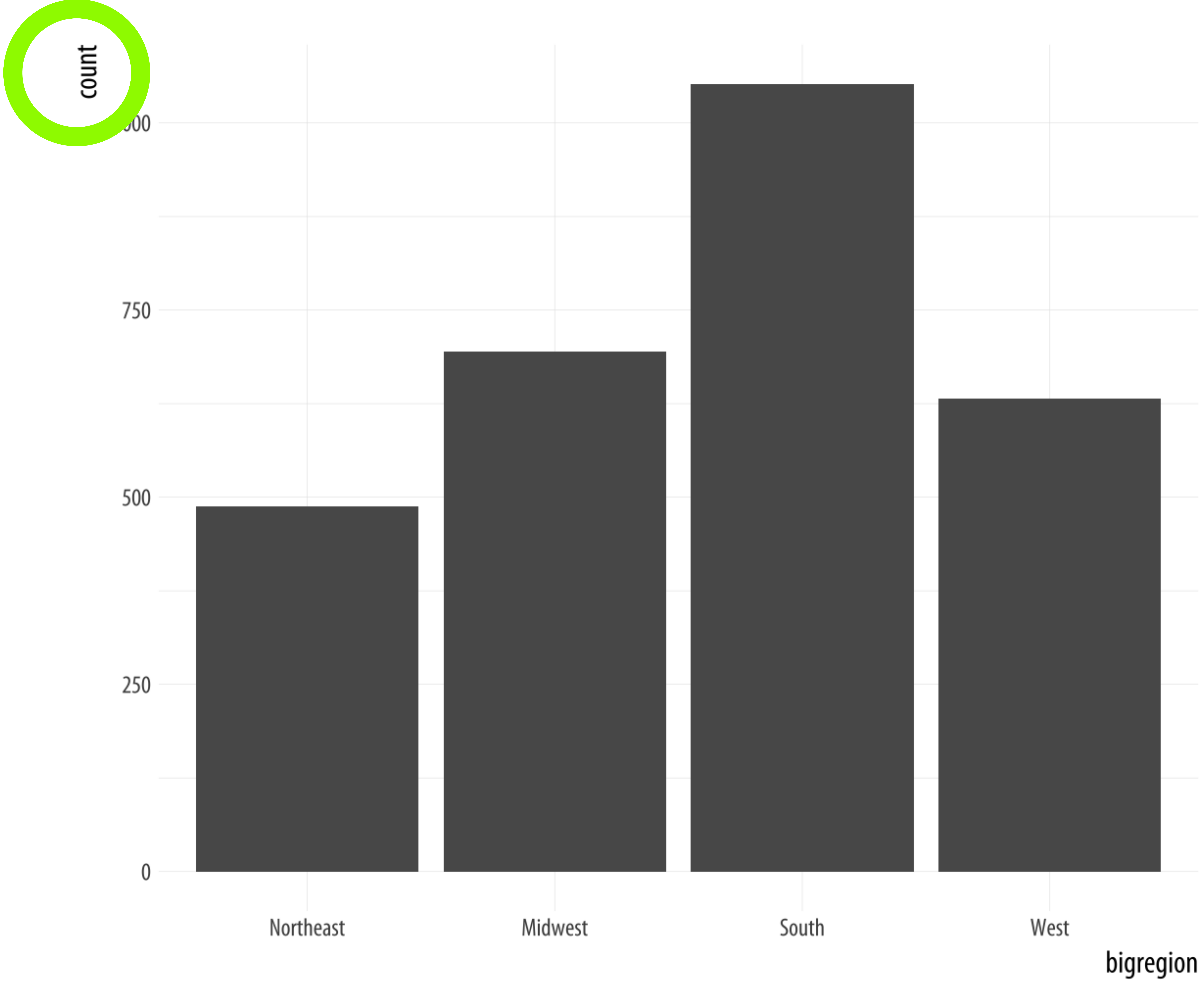
```
> gss_sm
# A tibble: 2,867 x 32
    year    id ballot   age childs  sibs degree race  sex   region income16 relig marital padeg madeg
   <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl> <fct>  <fct> <fct> <fct>  <fct>    <fct> <fct>   <fct> <fct>
 1  2016     1      1    47      3     2 Bache… White Male  New E… $170000… None  Married Grad… High…
 2  2016     2      2    61      0     3 High … White Male  New E… $50000 … None  Never … Lt H… High…
 3  2016     3      3    72      2     3 Bache… White Male  New E… $75000 … Cath… Married High… Lt H…
 4  2016     4      1    43      4     3 High … White Fema… New E… $170000… Cath… Married NA    High…
 5  2016     5      3    55      2     2 Gradu… White Fema… New E… $170000… None  Married Bach… High…
 6  2016     6      2    53      2     2 Junio… White Fema… New E… $60000 … None  Married NA    High…
 7  2016     7      1    50      2     2 High … White Male  New E… $170000… None  Married High… High…
 8  2016     8      3    23      3     6 High … Other Fema… Middl… $30000 … Cath… Married Lt H… Lt H…
 9  2016     9      1    45      3     5 High … Black Male  Middl… $60000 … Prot… Married Lt H… Lt H…
10  2016    10      3    71      4     1 Junio… White Male  Middl… $60000 … None  Divorc… High… High…
# … with 2,857 more rows, and 17 more variables: partyid <fct>, polviews <fct>, happy <fct>,
#   partners <fct>, grass <fct>, zodiac <fct>, pres12 <dbl>, wtssall <dbl>, income_rc <fct>, agegrp <fct>,
#   ageq <fct>, siblings <fct>, kids <fct>, religion <fct>, bigregion <fct>, partners_rc <fct>,
#   obama <dbl>
> 
```

```r
with(gss_sm, table(religion))
```

```
##
## Protestant    Catholic      Jewish        None       Other
##       1371         649          51         619         159
```

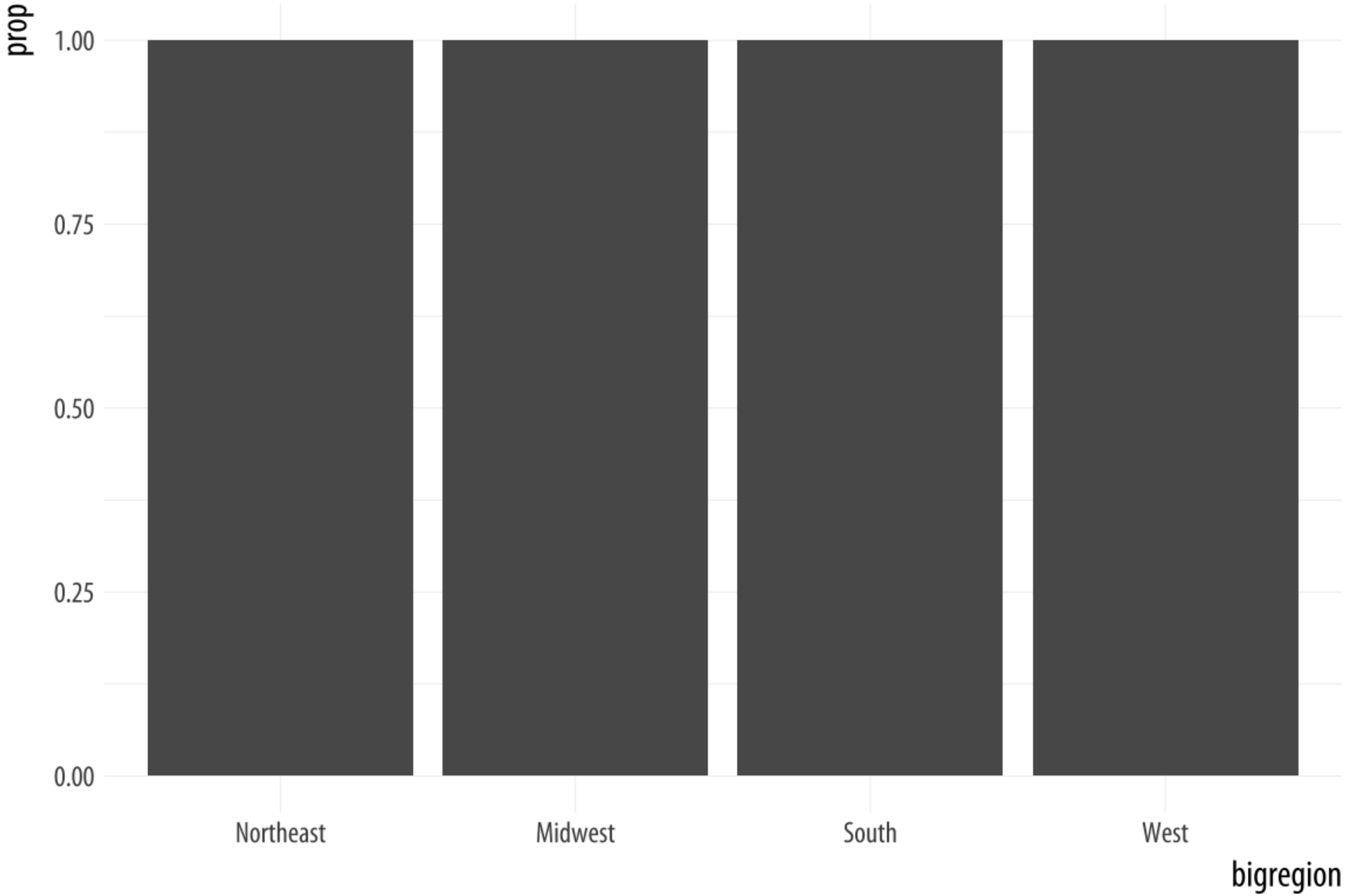
```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = bigregion))
p + geom_bar()
```

Just the one aesthetic mapping, to x.

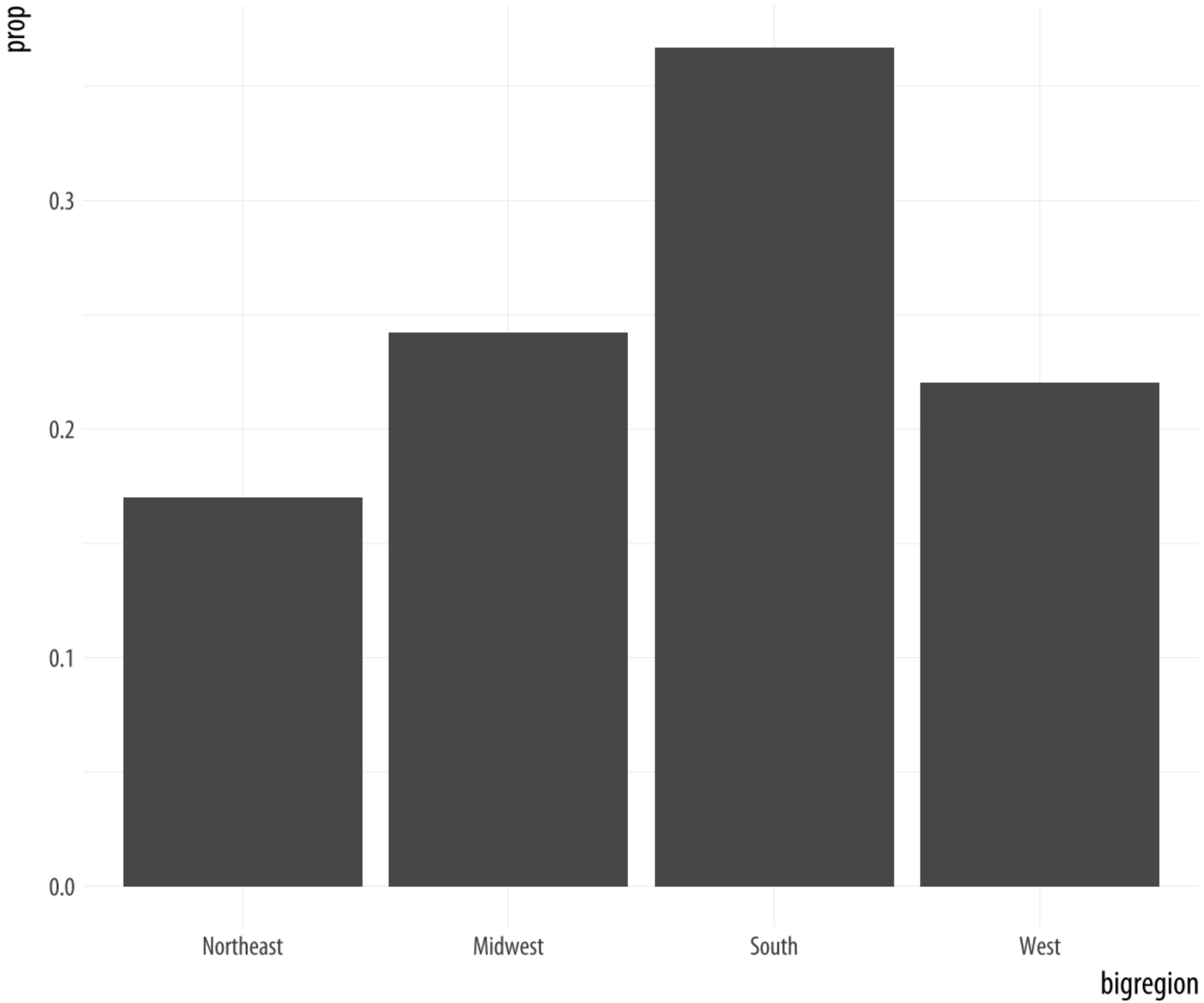

count
1000
750
500
250
0
Northeast
Midwest
South
West
bigregion

The y-axis variable, `count`, is not in the data. Instead, ggplot has calculated it for us. It does this using the default `stat_` function associated with `geom_bar()`, `stat_count().` This function can compute two new variables, `count`, and `prop` (short for **proportion**). The `count` statistic is the default one used.

```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = bigregion))
p + geom_bar(mapping = aes(y = ..prop..))
```


prop
1.00
0.75
0.50
0.25
0.00
Northeast
Midwest
South
West
bigregion

```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = bigregion))
p + geom_bar(mapping = aes(y = ..prop.., group = 1))
```


prop
0.3
0.2
0.1
0.0
Northeast
Midwest
South
West
bigregion

```r
p + geom_bar()
```

```r
p + stat_count()
```

geom_ functions call their default stat_ functions behind the scenes. (And vice versa)

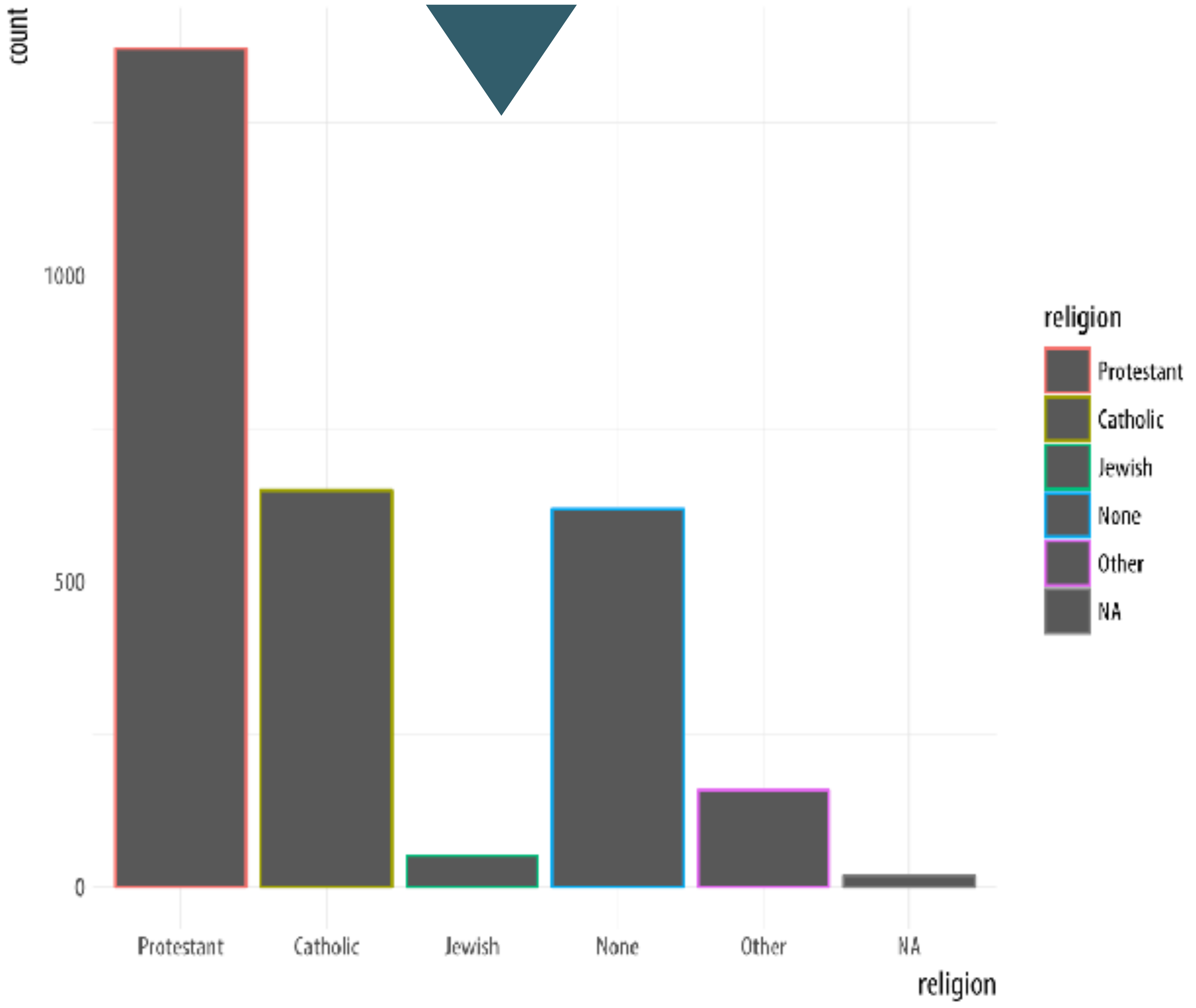
```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = religion))
p + geom_bar()
```

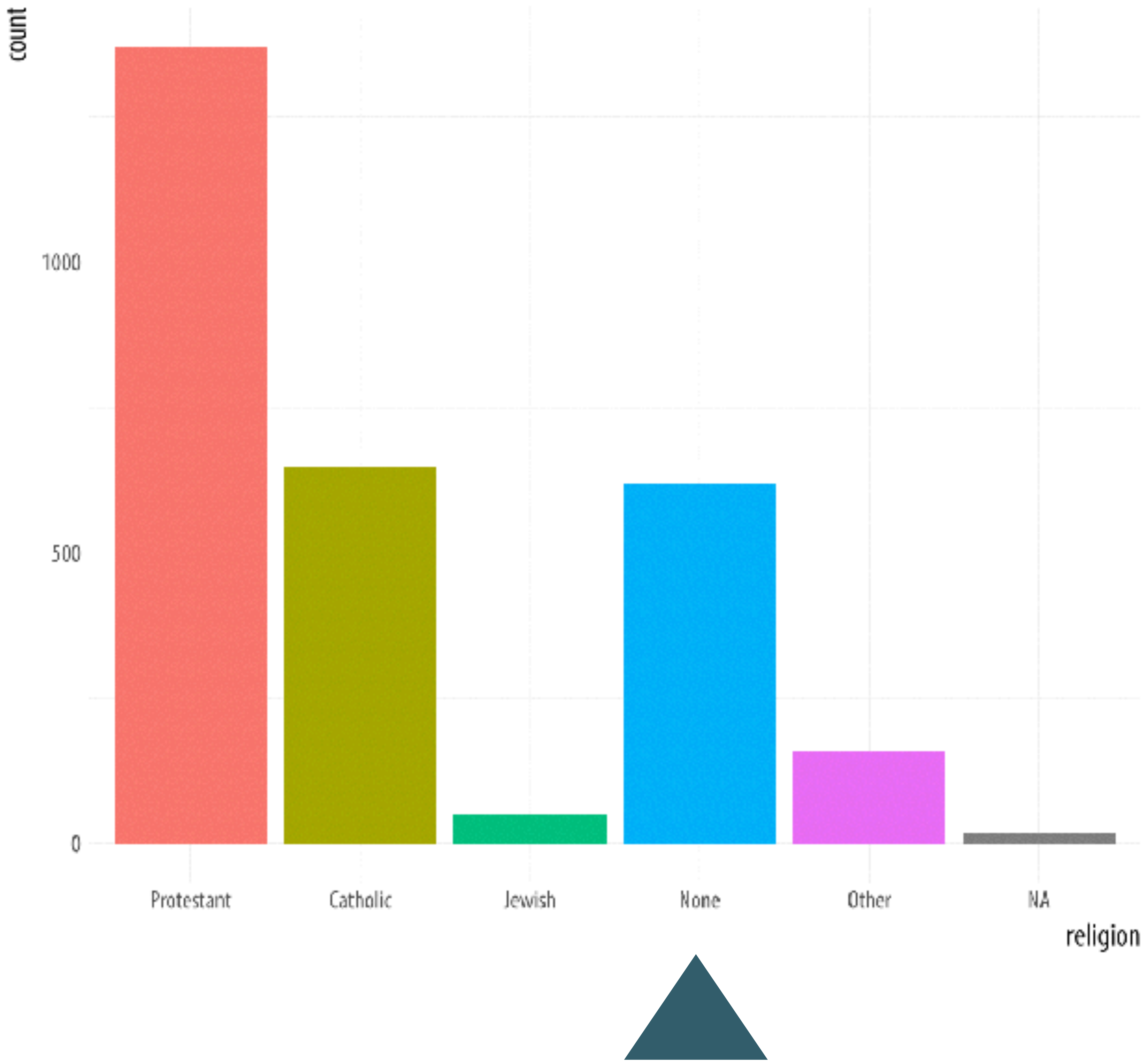
```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = religion, color = religion))
p + geom_bar()
```

```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = religion, fill = religion))
p + geom_bar()
```

```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = religion, fill = religion))
p + geom_bar() + guides(fill = FALSE)
```

```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = religion, color = religion))
p + geom_bar()
```






```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = religion, fill = religion))
p + geom_bar() + guides(fill = FALSE)
```

HISTOGRAMS & KERNEL DENSITIES

midwest

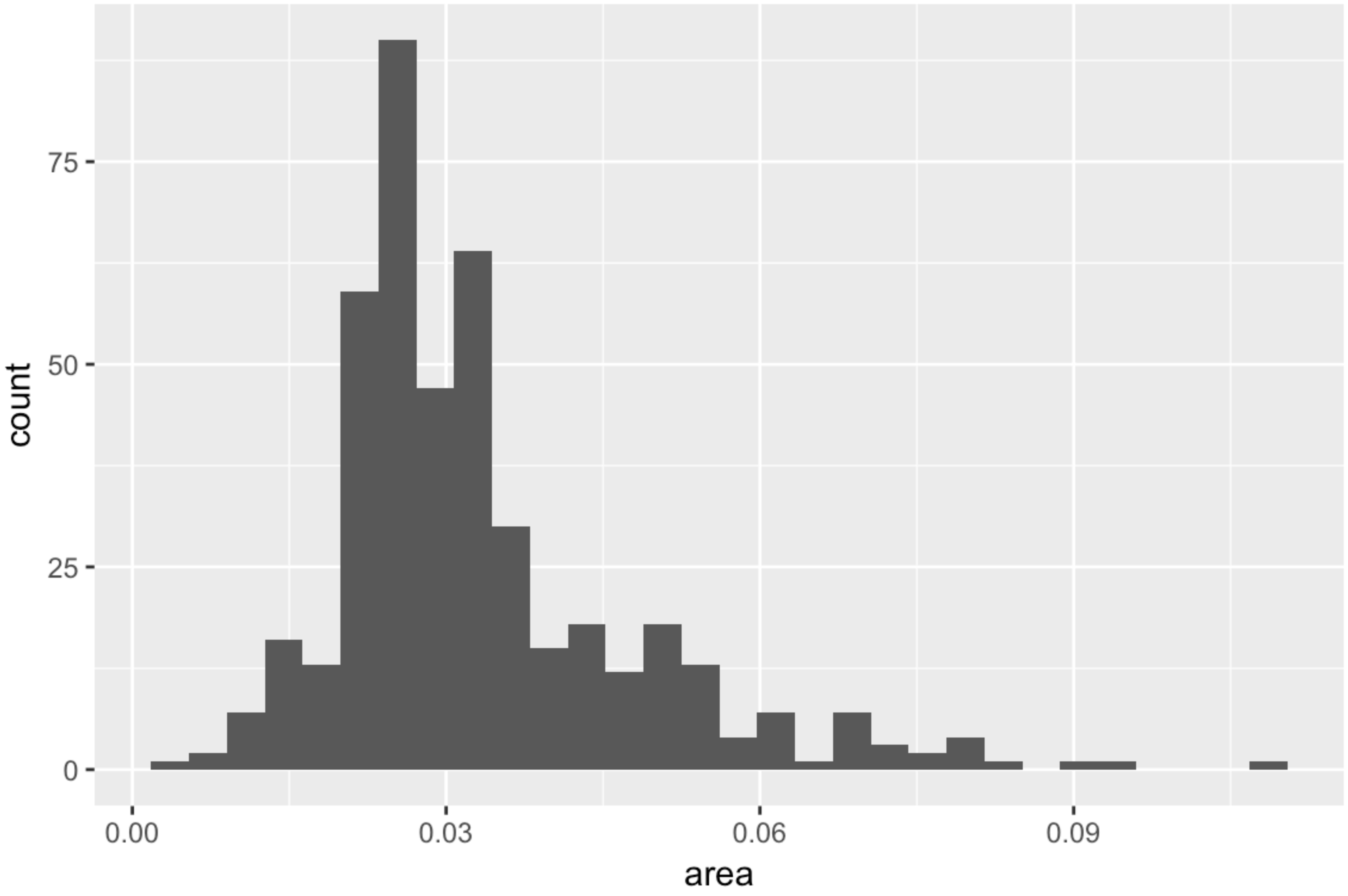
County-Level Census Data for Midwestern States

```
> midwest
# A tibble: 437 x 28
     PID county state  area poptotal popdensity popwhite popblack popamerindian popasian popother percwhite
   <int> <chr>  <chr> <dbl>    <int>      <dbl>    <int>    <int>         <int>    <int>    <int>     <dbl>
 1   561 ADAMS  IL    0.052    66090      1271.    63917     1702            98      249      124      96.7
 2   562 ALEXA… IL    0.014    10626       759      7054     3496            19       48        9      66.4
 3   563 BOND   IL    0.022    14991       681.    14477      429            35       16       34      96.6
 4   564 BOONE  IL    0.017    30806      1812.    29344      127            46      150     1139      95.3
 5   565 BROWN  IL    0.018     5836       324.     5264      547            14        5        6      90.2
 6   566 BUREAU IL    0.05     35688       714.    35157       50            65      195      221      98.5
 7   567 CALHO… IL    0.017     5322       313.     5298        1             8       15        0      99.5
 8   568 CARRO… IL    0.027    16805       622.    16519      111            30       61       84      98.3
 9   569 CASS   IL    0.024    13437       560.    13384       16             8       23        6      99.6
10   570 CHAMP… IL    0.058   173025      2983.   146506    16559           331     8033     1596      84.7
# … with 427 more rows, and 16 more variables: percblack <dbl>, percamerindan <dbl>, percasian <dbl>,
#   percother <dbl>, popadults <int>, perchsd <dbl>, percollege <dbl>, percprof <dbl>,
#   poppovertyknown <int>, percpovertyknown <dbl>, percbelowpoverty <dbl>, percchildbelowpovert <dbl>,
#   percadultpoverty <dbl>, percelderlypoverty <dbl>, inmetro <int>, category <chr>
>
```

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

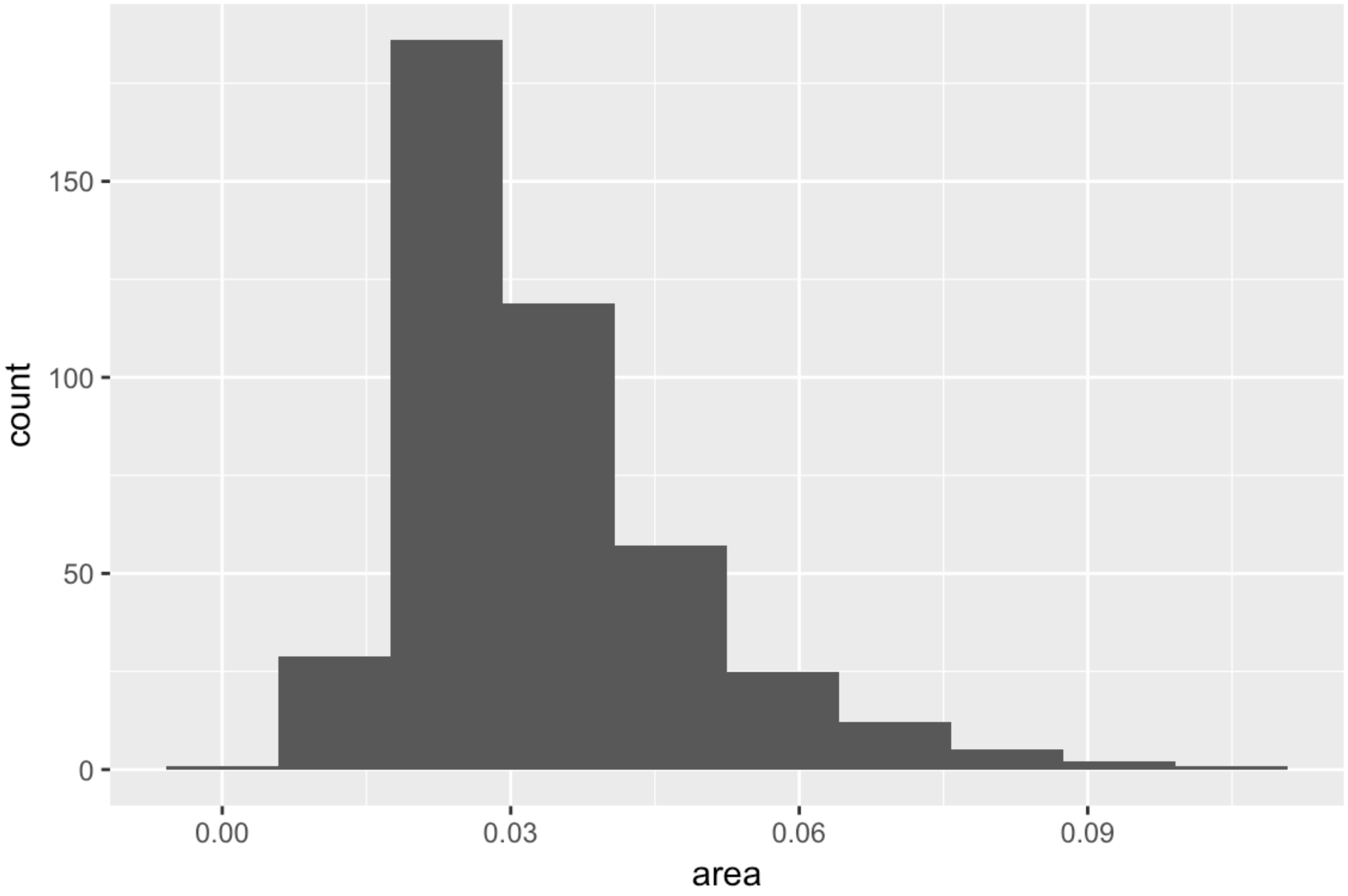
```
## `stat_bin()` using `bins = 30`.
## Pick better value with `binwidth`.
```

The default stat for this geom has to make a choice, and is letting us know we might want to override it.



count
75
50
25
0
0.00
0.03
0.06
0.09
area

```r
p <- ggplot(data = midwest,
            mapping = aes(x = area))
p + geom_histogram(bins = 10)
```


150
100
50
0
count
0.00
0.03
0.06
0.09
area

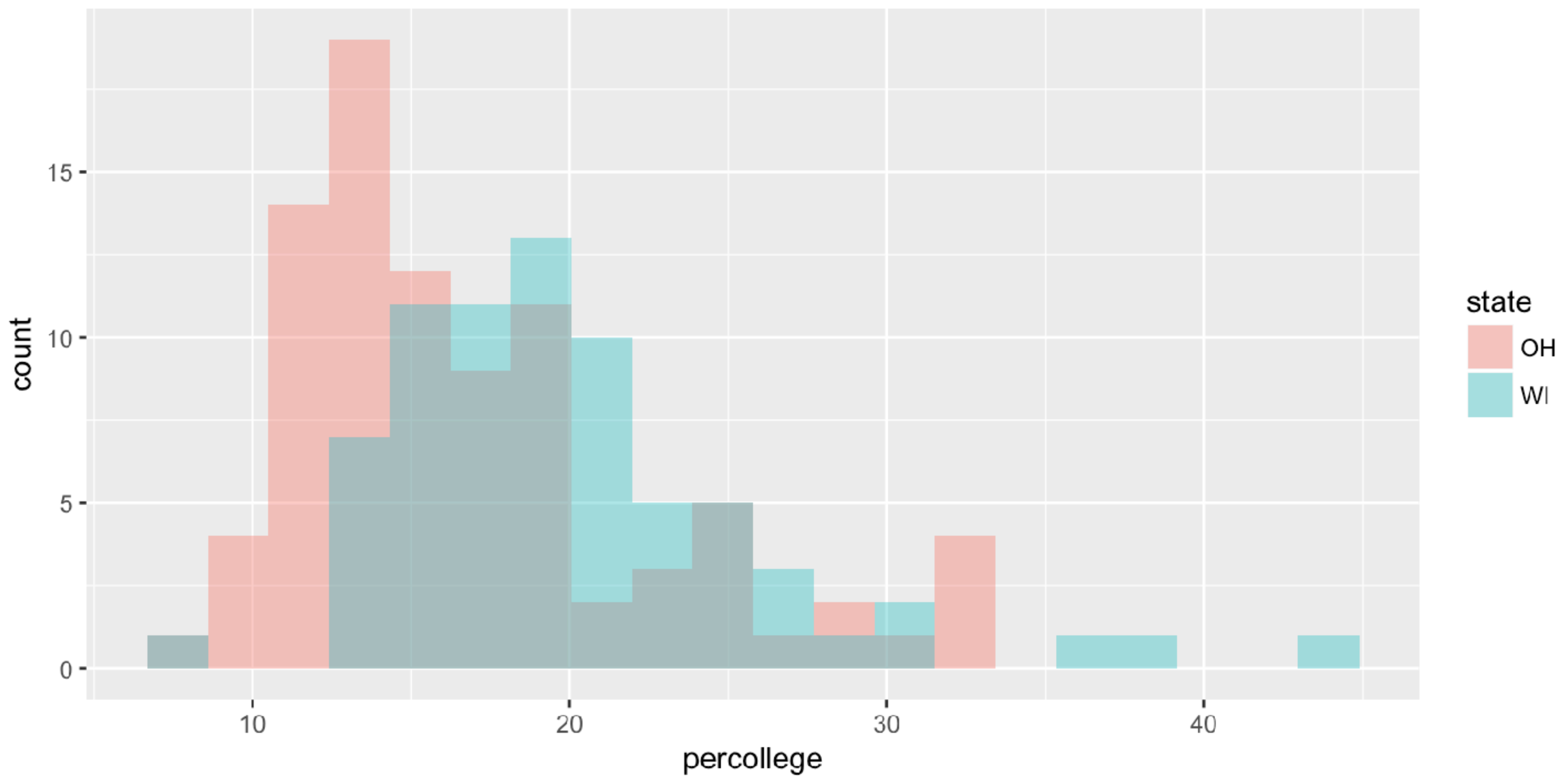
```r
oh_wi <- c("OH", "WI")
```

subset our data on the fly

a convenient, built-in operator

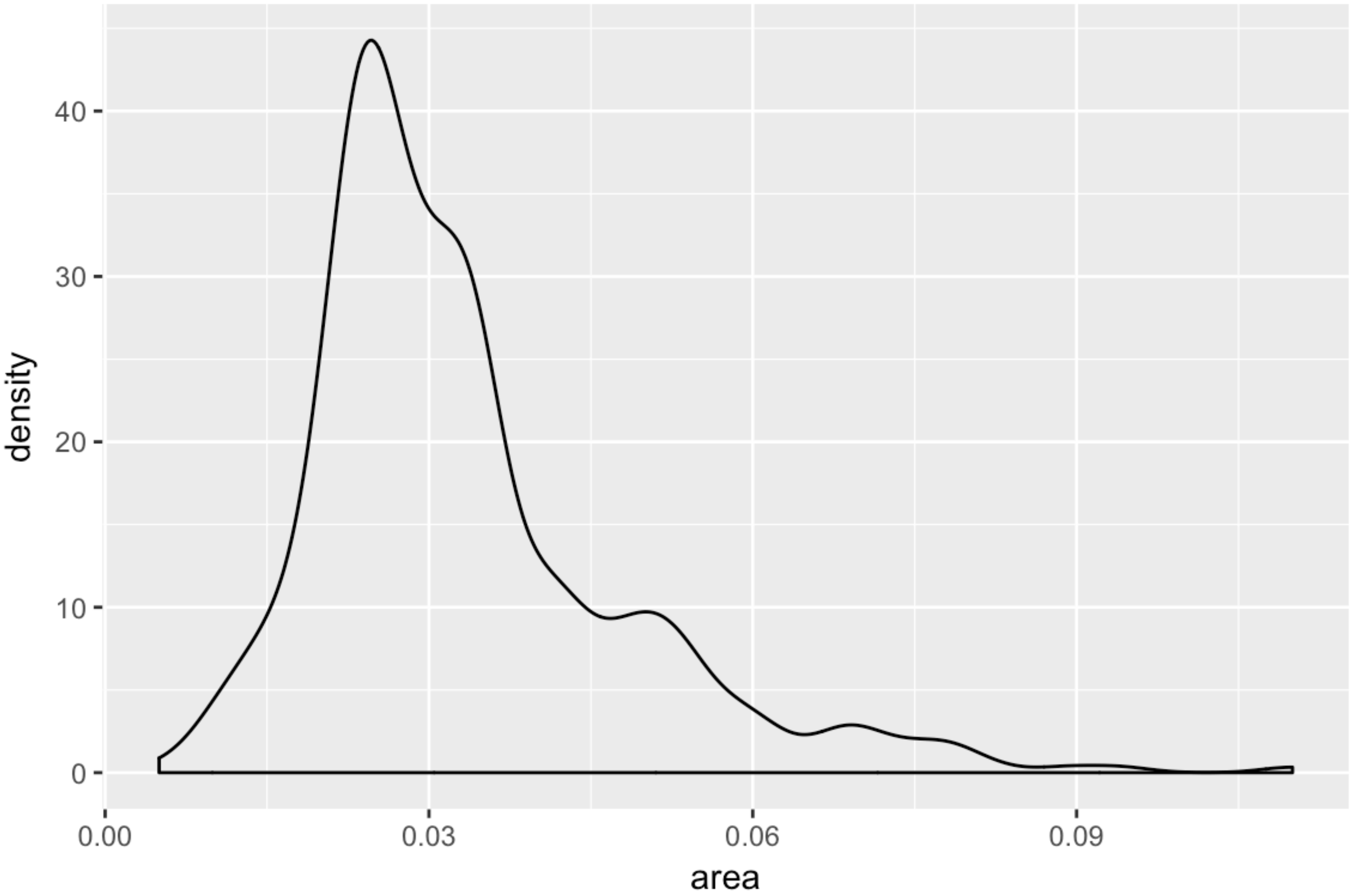
```r
p <- ggplot(data = subset(midwest, state %in% oh_wi),
            mapping = aes(x = percollege, fill = state))

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

Just plot x by its values on the scale, don't stack or dodge


count
15
10
5
0
10
20
30
40
percollege
state
OH
WI

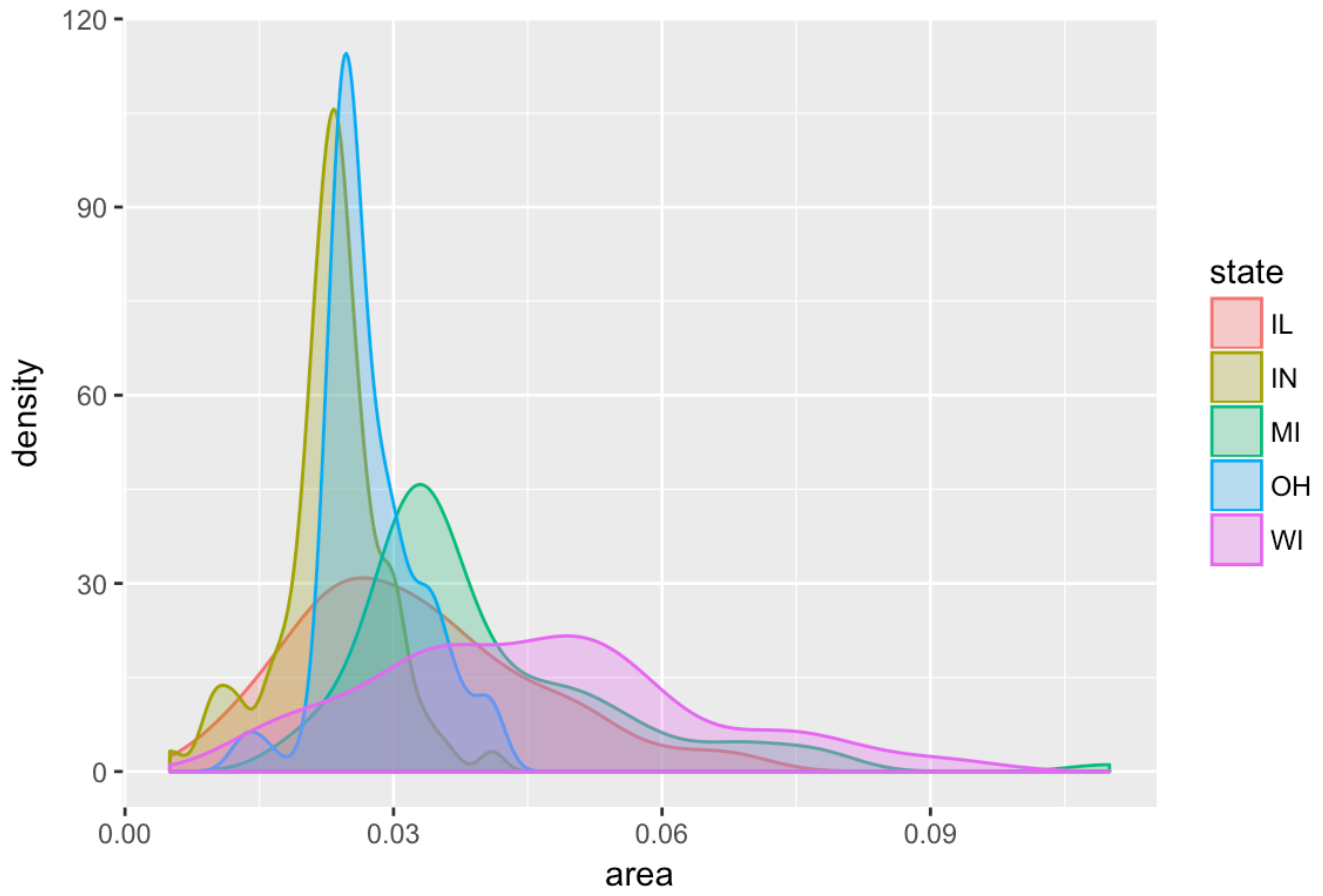
```r
p <- ggplot(data = midwest,
            mapping = aes(x = area))
p + geom_density()
```

geom_hist()'s continuous counterpart, geom_density()

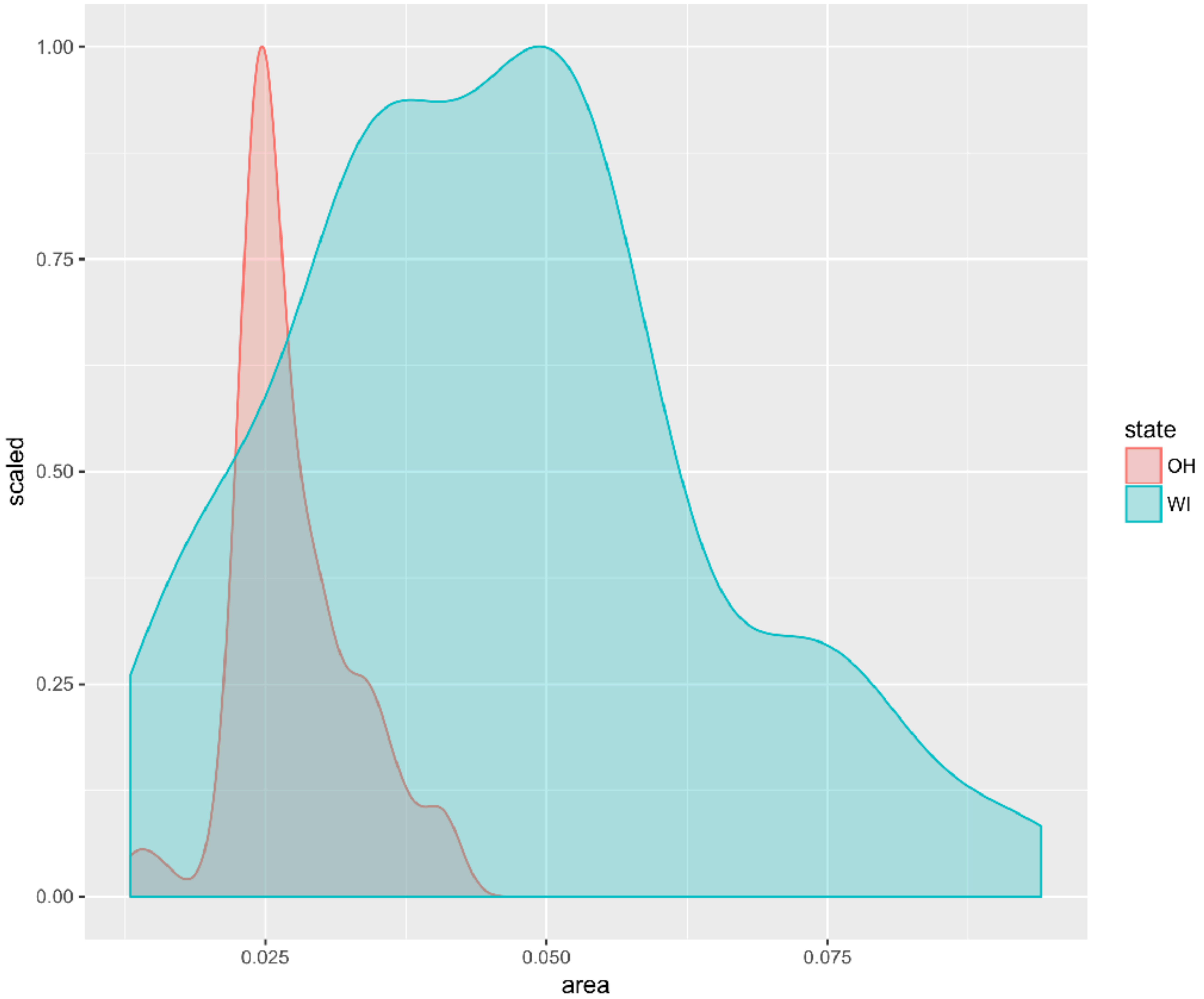


density
40
30
20
10
0
0.00
0.03
0.06
0.09
area

```r
p <- ggplot(data = midwest,
            mapping = aes(x = area,
                          fill = state,
                          color = state))

p + geom_density(alpha = 0.3)
```


120
90
60
30
0
density
0.00
0.03
0.06
0.09
area
state
IL
IN
MI
OH
WI

```r
p <- ggplot(data = subset(midwest, subset = state %in% OH_WI),
            mapping = aes(x = area, fill = state, color = state))

p + geom_density(alpha = 0.3, mapping = (aes(y = ..scaled..)))
```



AVOIDING TRANSFORMATIONS WHEN NECESSARY

```
> titanic
```

```
##       fate gender    n percent
## 1 perished   male 1364    62.0
## 2 perished female  126     5.7
## 3 survived   male  367    16.7
## 4 survived female  344    15.6
```

No counting up required?
Then stat = identity

```r
p <- ggplot(data = titanic,
            mapping = aes(x = fate,
                          y = percent,
                          fill = sex))
p + geom_bar(stat = "identity",
             position = "dodge") +
    theme(legend.position = "top")
```

The theme() function controls parts of the plot that don’t belong to its “grammatical” structure

```r
p <- ggplot(data = titanic,
            mapping = aes(x = fate,
                          y = percent,
                          fill = sex))
p + geom_col(position = "dodge") +
    theme(legend.position = "top")
```

Even better: for convenience when not counting up, just use geom_col()

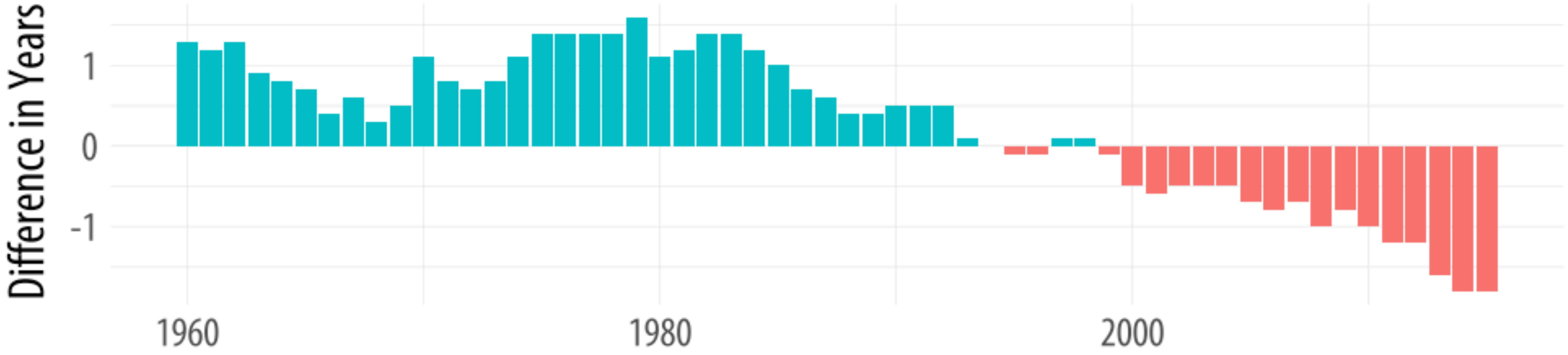

sex
female
male
percent
60
40
20
0
perished
survived
fate

```
oecd_sum

## # A tibble: 57 x 5
## # Groups:   year [57]
##     year other   usa  diff hi_lo
##    <int> <dbl> <dbl> <dbl> <chr>
##  1  1960  68.6  69.9 1.30  Below
##  2  1961  69.2  70.4 1.20  Below
##  3  1962  68.9  70.2 1.30  Below
##  4  1963  69.1  70.0 0.900 Below
##  5  1964  69.5  70.3 0.800 Below
##  6  1965  69.6  70.3 0.700 Below
##  7  1966  69.9  70.3 0.400 Below
##  8  1967  70.1  70.7 0.600 Below
##  9  1968  70.1  70.4 0.300 Below
## 10  1969  70.1  70.6 0.500 Below
## # ... with 47 more rows
```

```r
p <- ggplot(data = oecd_sum,
            mapping = aes(x = year, y = diff, fill = hi_lo))
p + geom_col() + guides(fill = FALSE) +
  labs(x = NULL, y = "Difference in Years",
       title = "The US Life Expectancy Gap",
       subtitle = "Difference between US and OECD
                   average life expectancies, 1960-2015",
       caption = "Data: OECD. After a chart by Christopher Ingraham,
                  Washington Post, December 27th 2017.")
```

The US Life Expectancy Gap

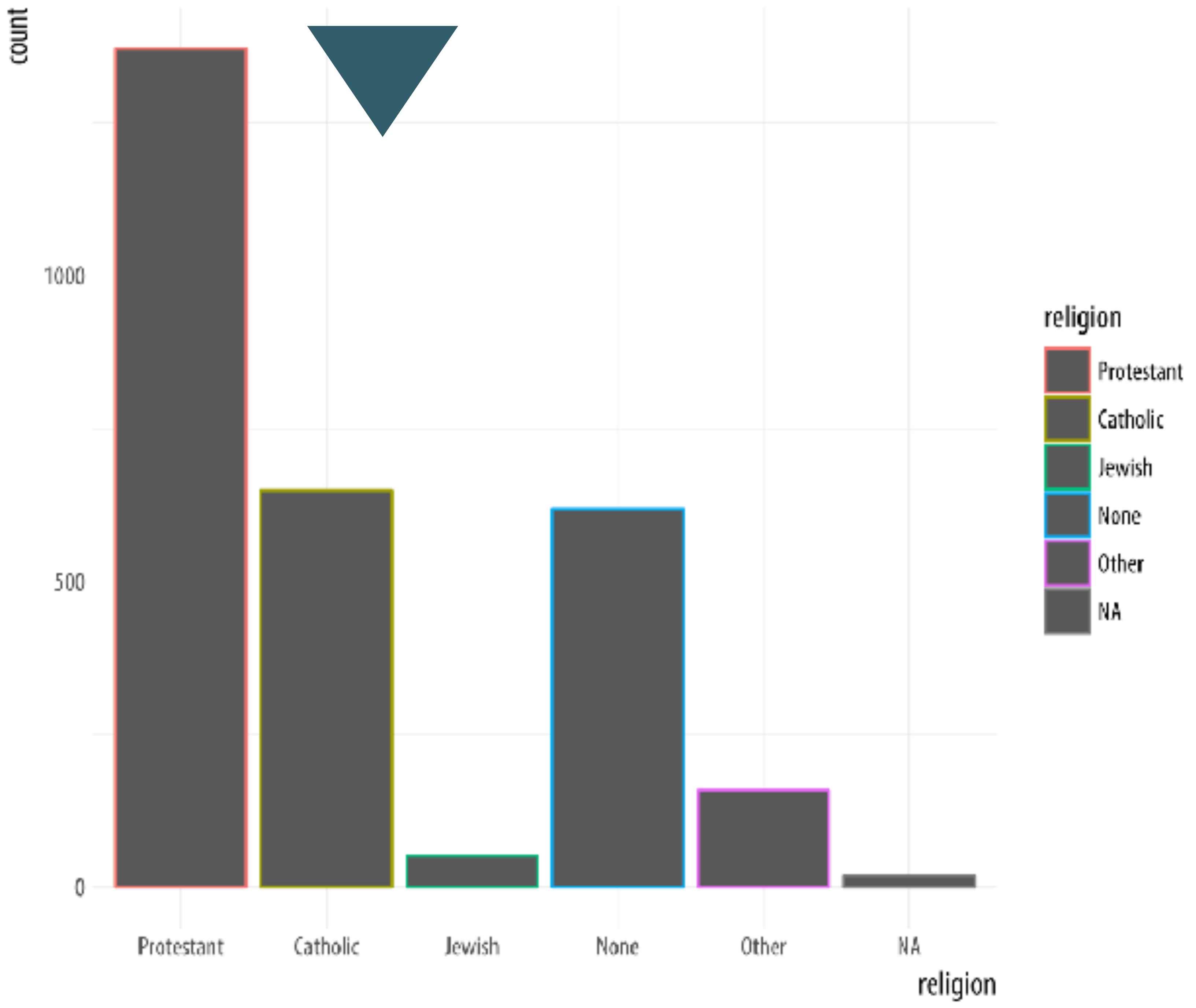
Difference between US and OECD average life expectancies, 1960-2015

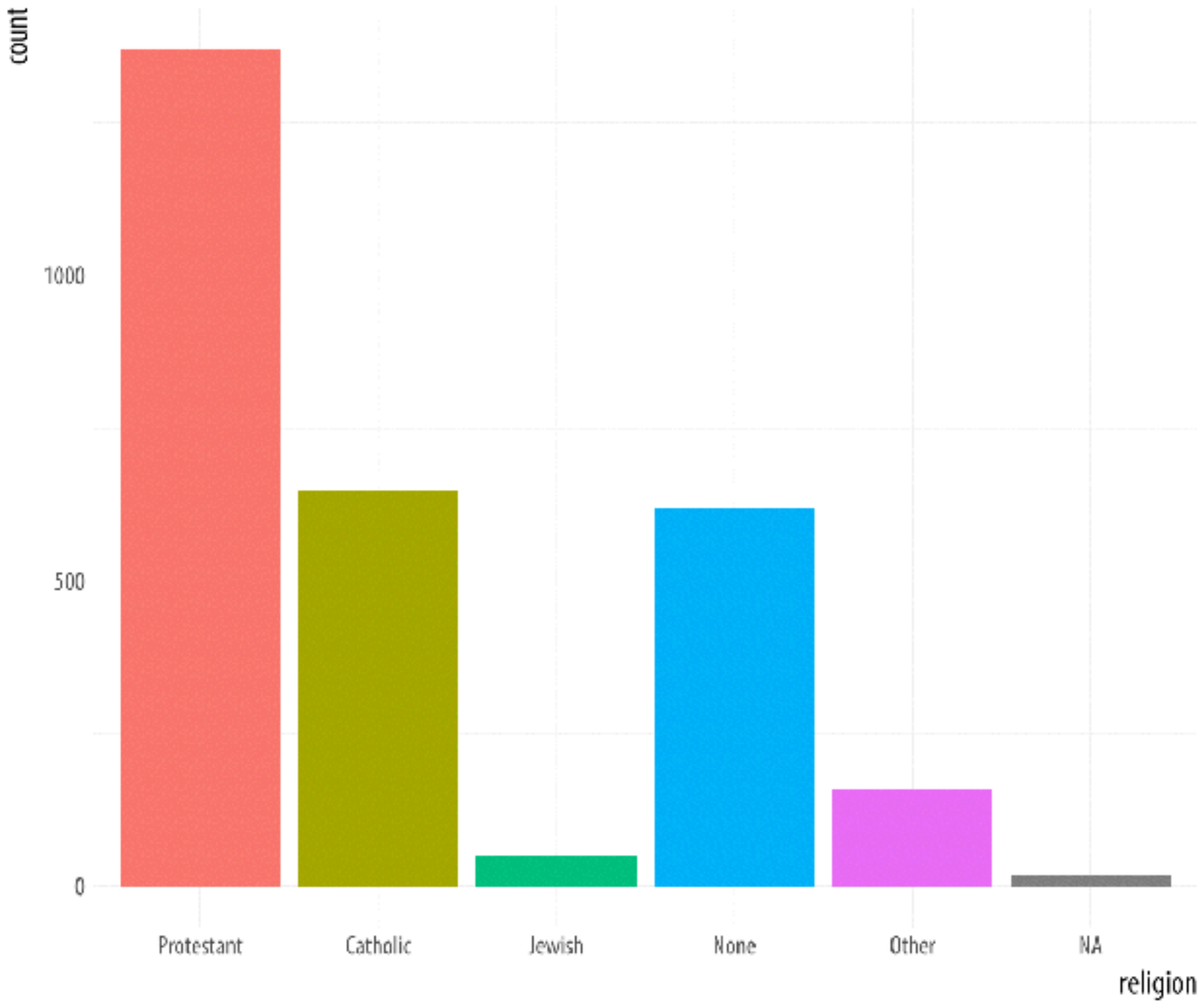


Data: OECD. After a chart by Christopher Ingraham,
Washington Post, December 27th 2017.

CROSSTABULATION THE AWKWARD WAY

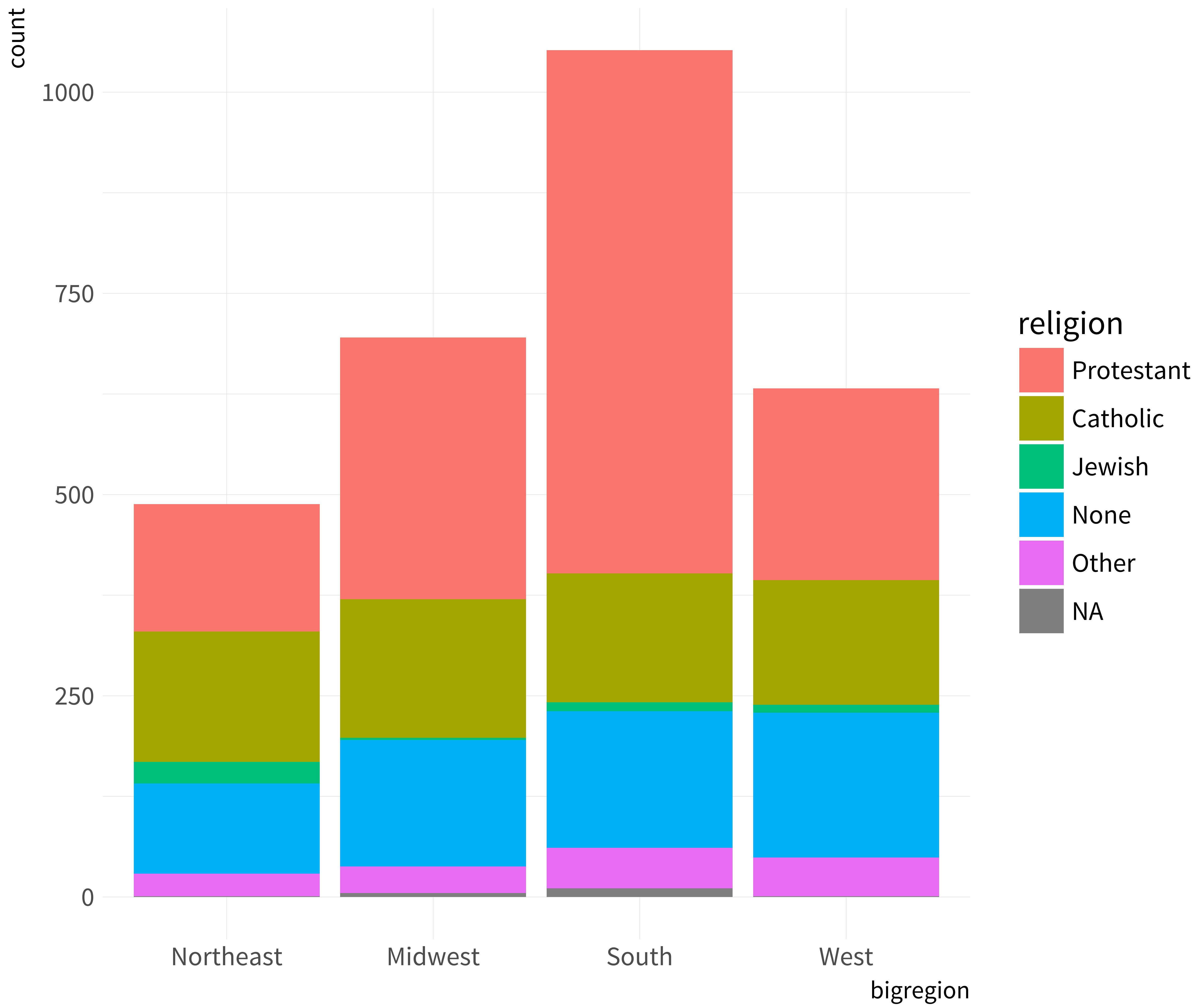
```
p <- ggplot(data = gss_sm,
            mapping = aes(x = religion, color = religion))
p + geom_bar()
```

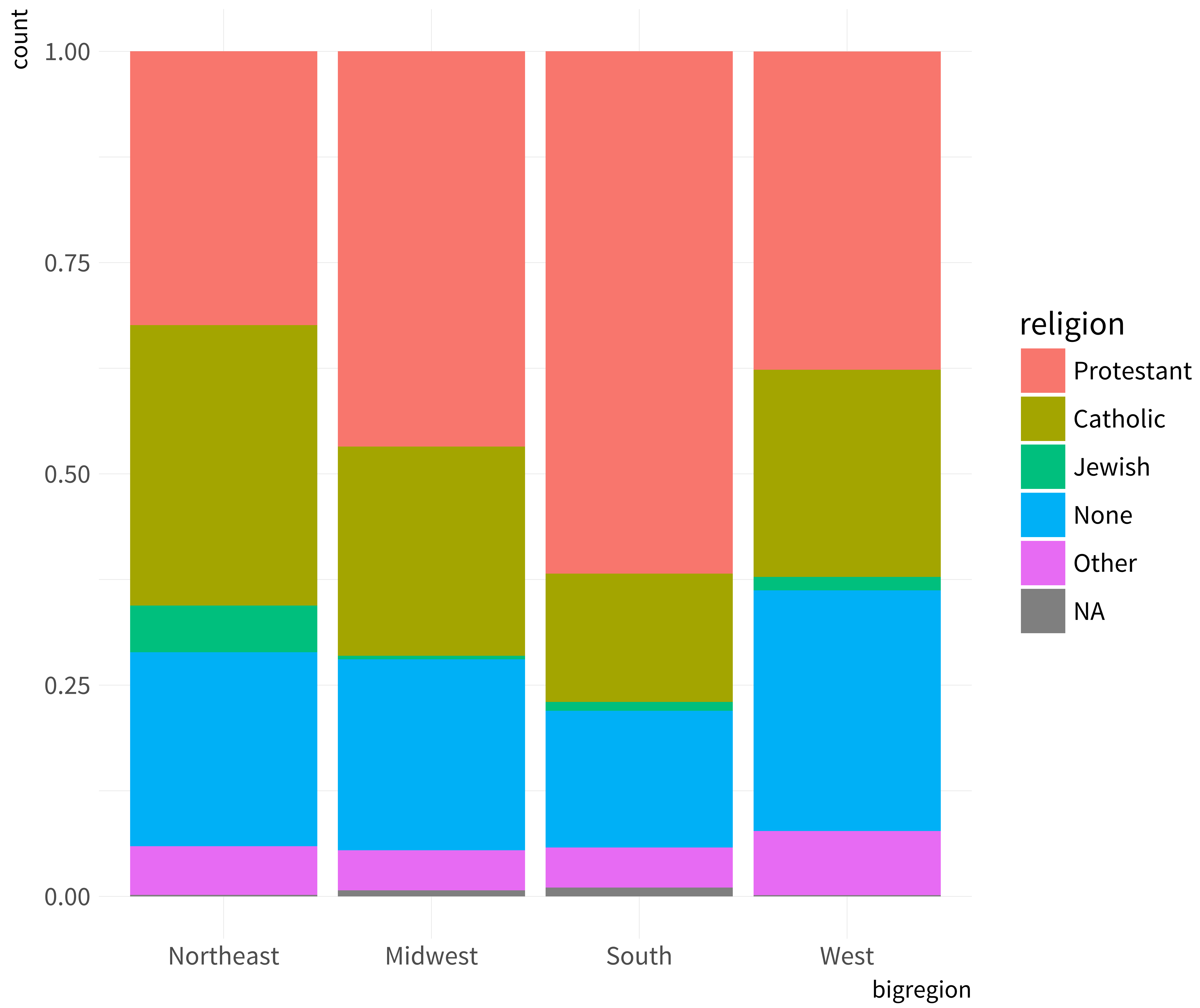





```
p <- ggplot(data = gss_sm,
            mapping = aes(x = religion, fill = religion))
p + geom_bar() + guides(fill = FALSE)
```

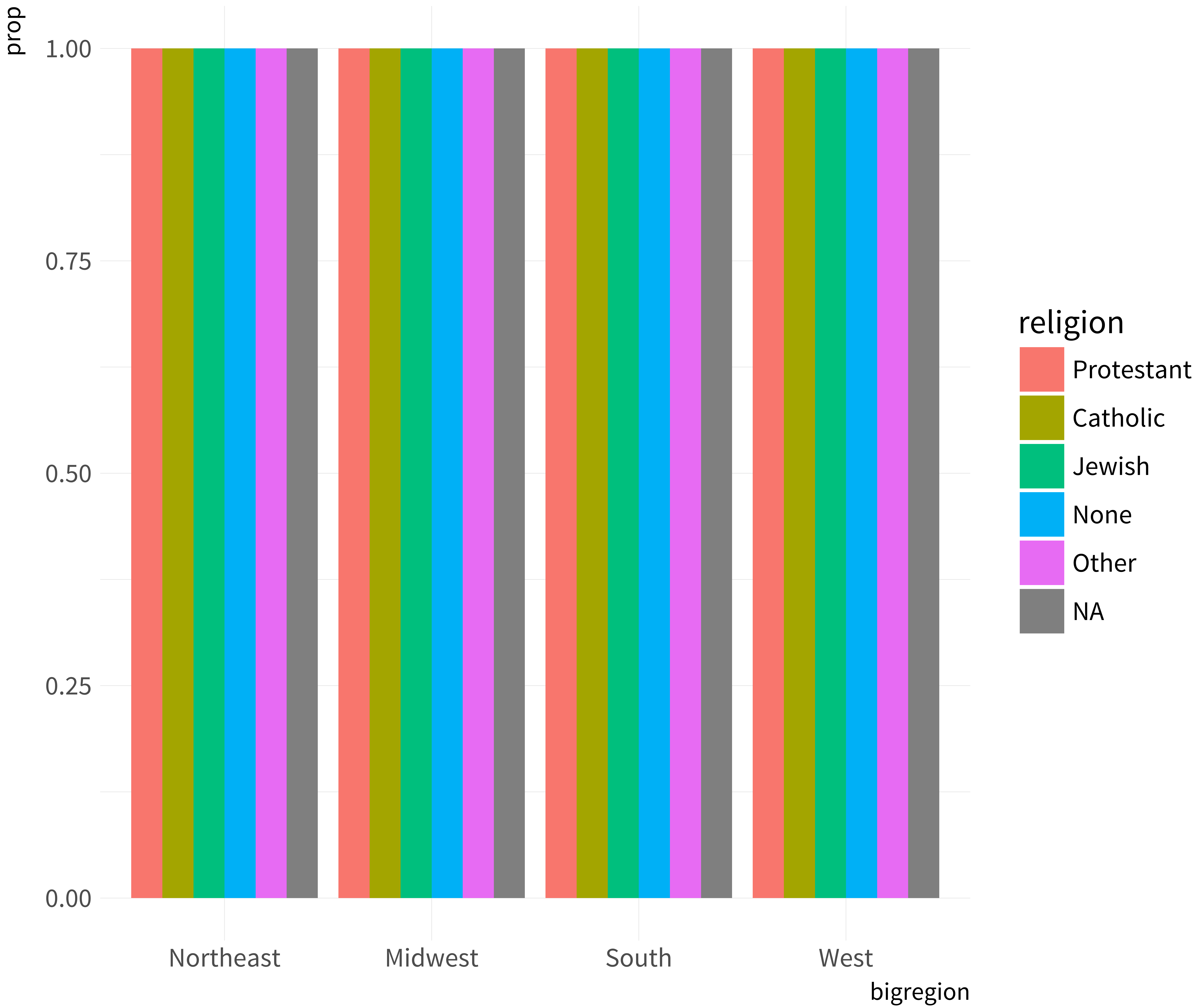
```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = bigregion,
                          fill = religion))

p + geom_bar()
```





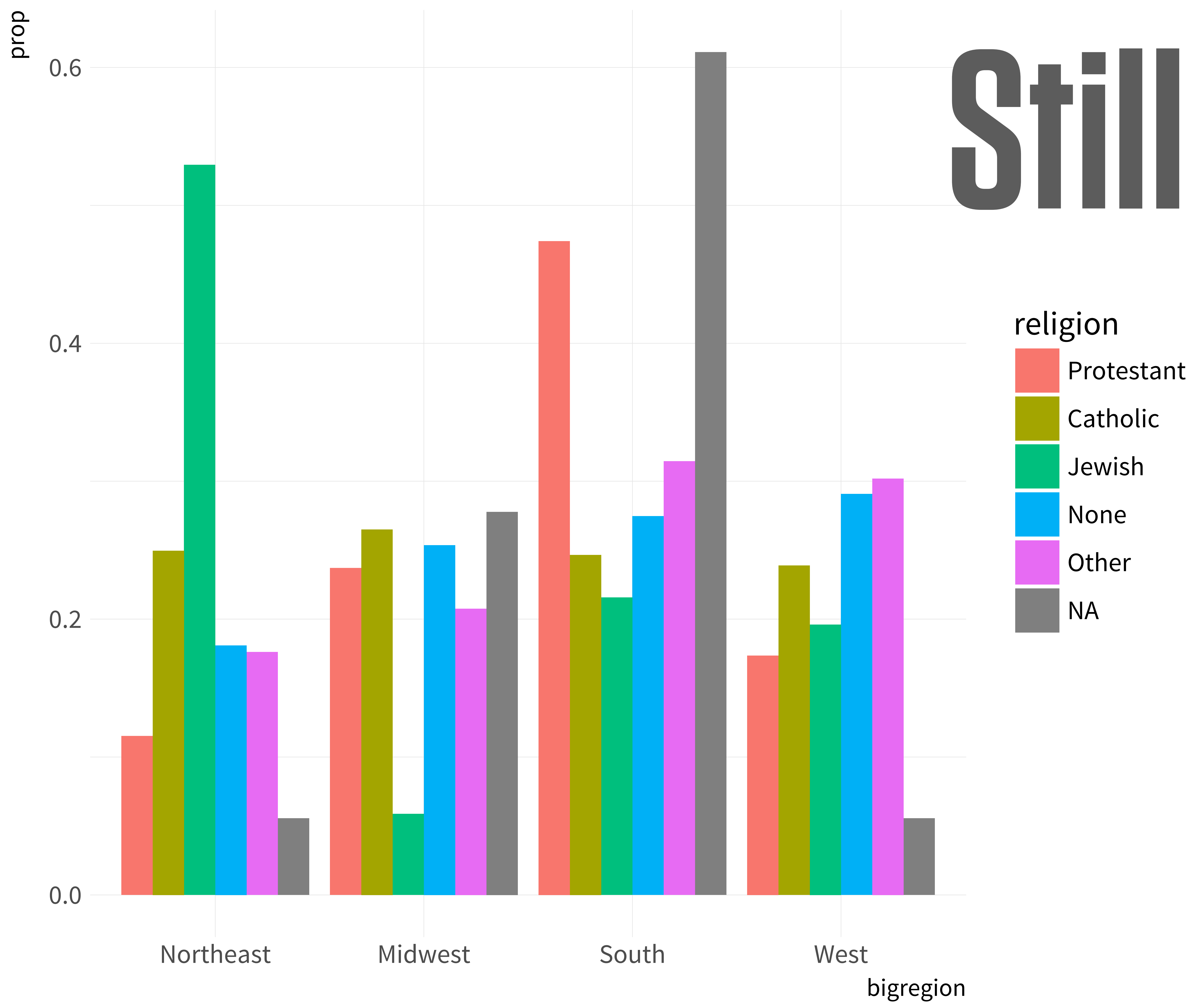

```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = bigregion,
                          fill = religion))
p + geom_bar(position = "fill")
```

```
p <- ggplot(data = gss_sm,
            mapping = aes(x = bigregion,
                          fill = religion))
p + geom_bar(position = "dodge",
             mapping = aes(y = ..prop..))
```


prop
1.00
0.75
0.50
0.25
0.00
Northeast
Midwest
South
West
bigregion
religion
Protestant
Catholic
Jewish
None
Other
NA

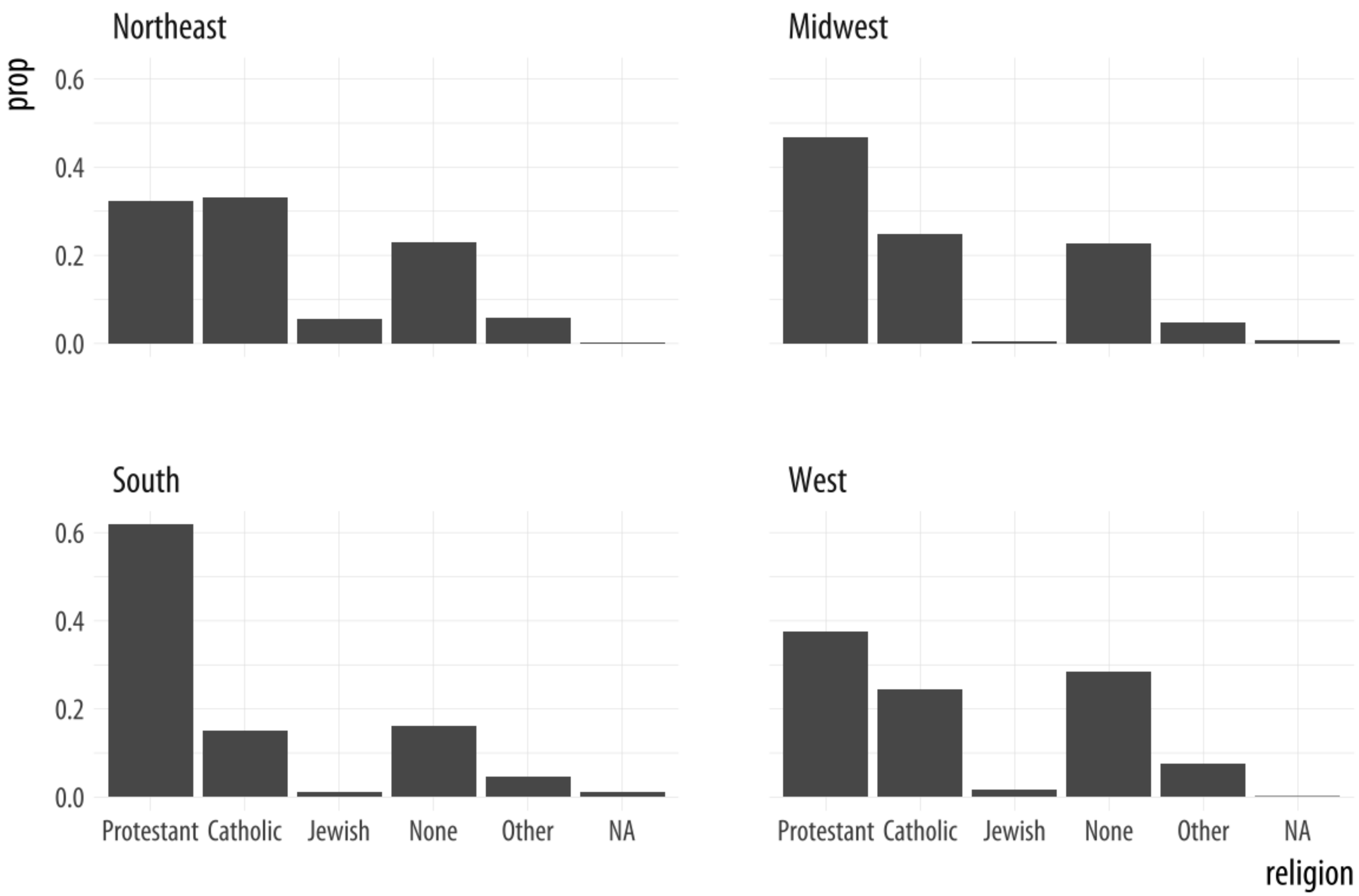
```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = bigregion,
                          fill = religion))
p + geom_bar(position = "dodge",
             mapping = aes(y = ..prop..,
                           group = religion))
```

Still not right!



Time to take a step back

```r
p <- ggplot(data = gss_sm,
            mapping = aes(x = religion))
p + geom_bar(position = "dodge",
             mapping = aes(y = ..prop..,
                           group = bigregion)) +
    facet_wrap(~ bigregion, ncol = 2)
```


Northeast
Midwest
South
West
prop
0.0
0.2
0.4
0.6
Protestant
Catholic
Jewish
None
Other
NA
religion

SURELY THINGS CAN BE EASIER THAN THIS?