Groups, facets, and stats

Data Visualisation mini-course

Gavin L. Simpson

Institute of Environmental Change and Society

22/04/2019

Start

Chris Knight

Catch-up

```
library('gapminder')
library('ggplot2')

## load some new data
load(url('http://bit.ly/gss_data'))
ls()
```

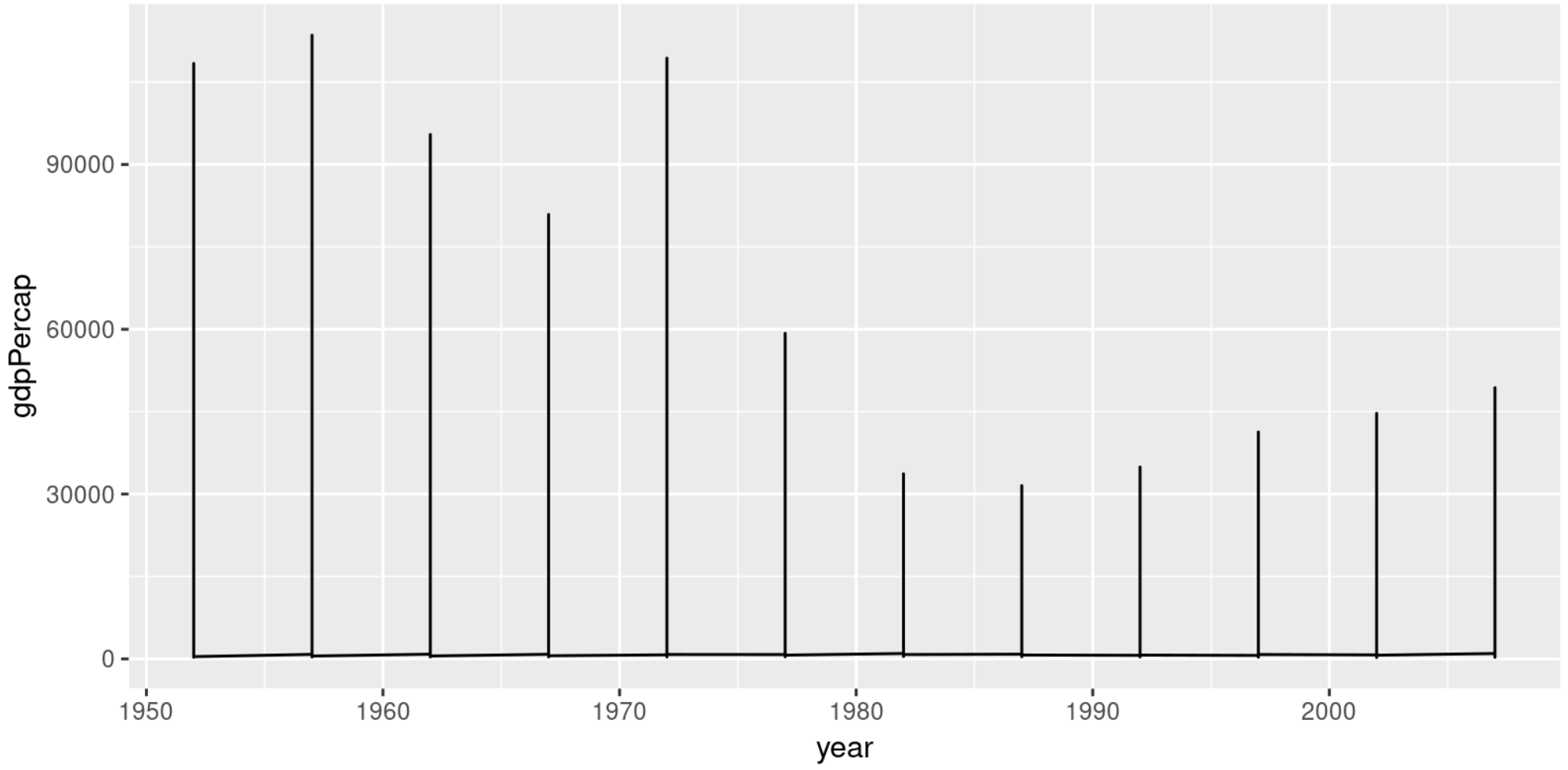
```
## [1] "gss_sm"
```

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

Grouping

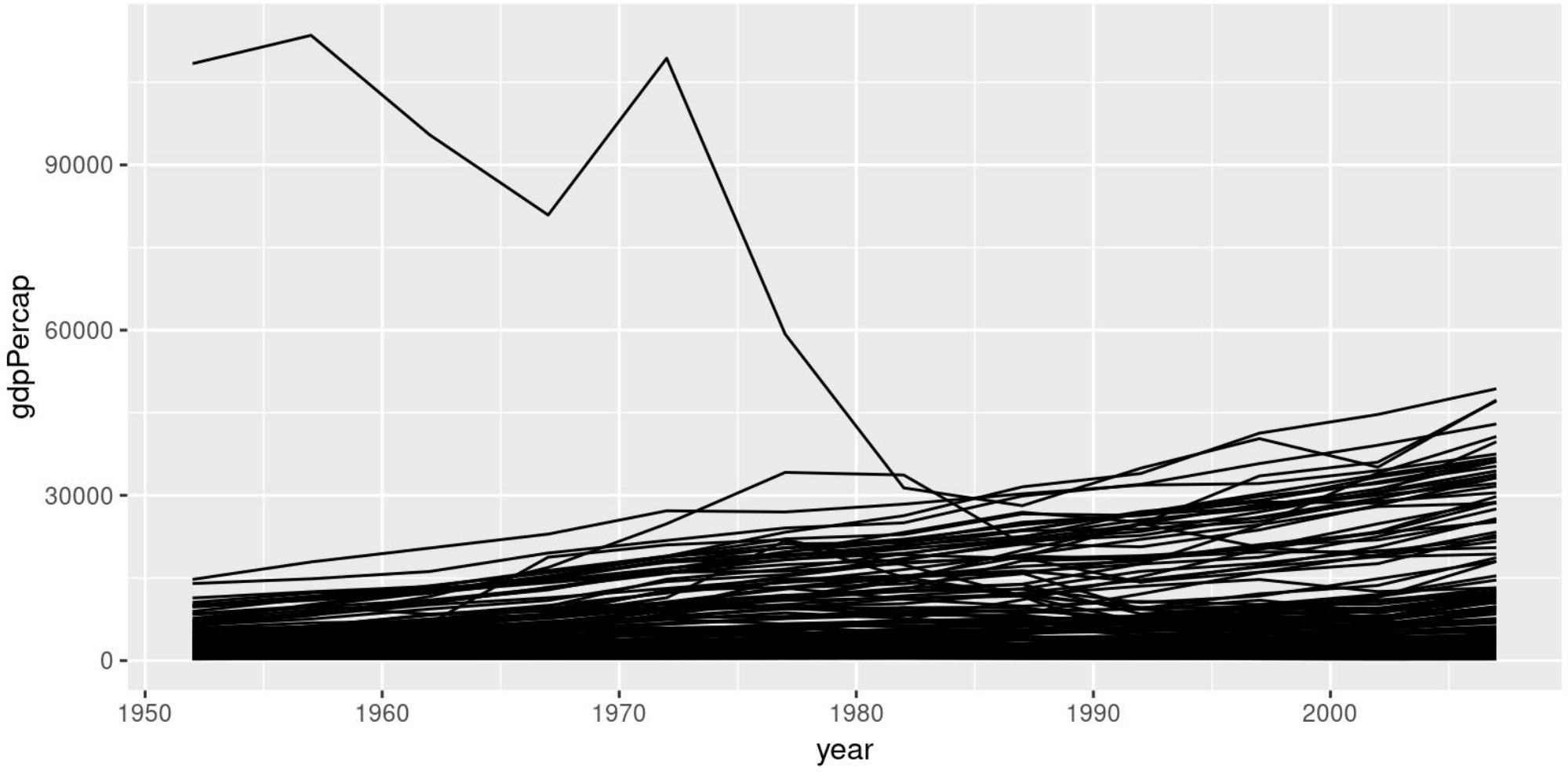
What went wrong?

```
ggplot(gapminder, aes(x = year, y = gdpPercap)) + geom_line()
```



Tell `ggplot` about structure in the data

```
ggplot(gapminder, aes(x = year, y = gdpPercap)) + geom_line(aes(group = country))
```



The `group` aesthetic

Use the `group` aesthetic when structure in the data isn't already mapped to an aesthetic

Facets — small multiples

JOSHUA COLEMAN

Small multiples

A *small multiple* is a series of similar plots using the same scale and axes

Multiple plots show different partitions of the data

In *ggplot*, small multiple plots are created by *facetting*

- `facet_wrap()`

  One or more categorical variables used to partition the data

  Individual plots are arrange in sequence over a number of rows and columns

- `facet_grid()`

  Data are partitioned by two sets of varables & arranged into a grid

  One set of variables forms partitions for the rows

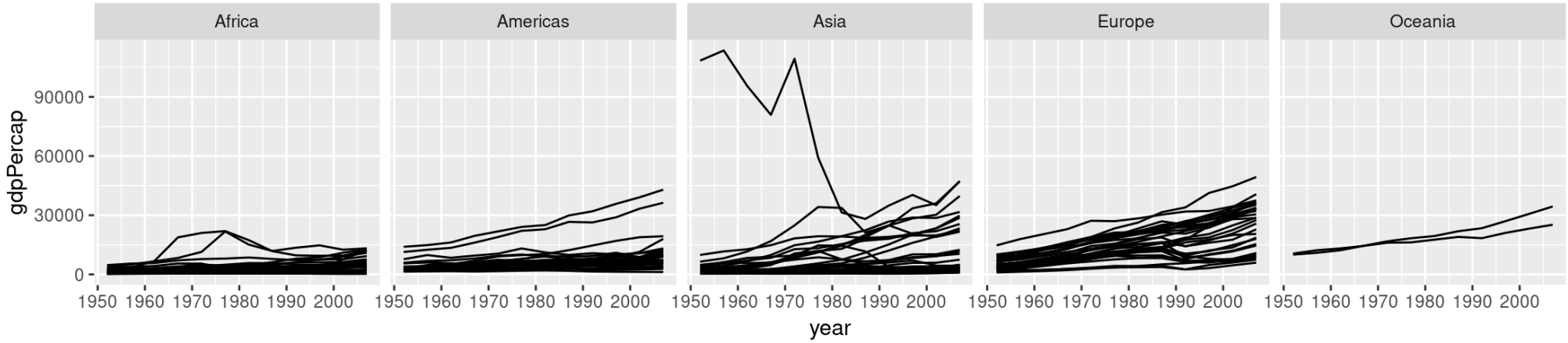
  Second set of variables forms partitions for the columns

Small multiples — `facet_wrap()`

The partition is specified using a formula: `~ f1 + f2`

Use the `nrow` and `ncol` arguments to set the required dimensions

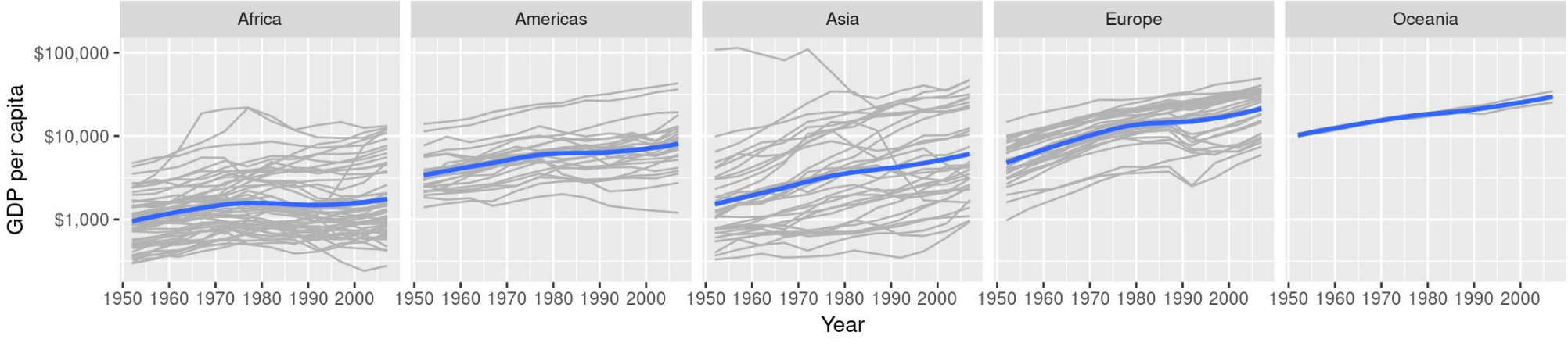
```
p +
  geom_line(aes(group = country)) +
  facet_wrap(~ continent, nrow = 1)
```



Most commonly used with a single partitioning variable

Small multiples — `facet_wrap()`

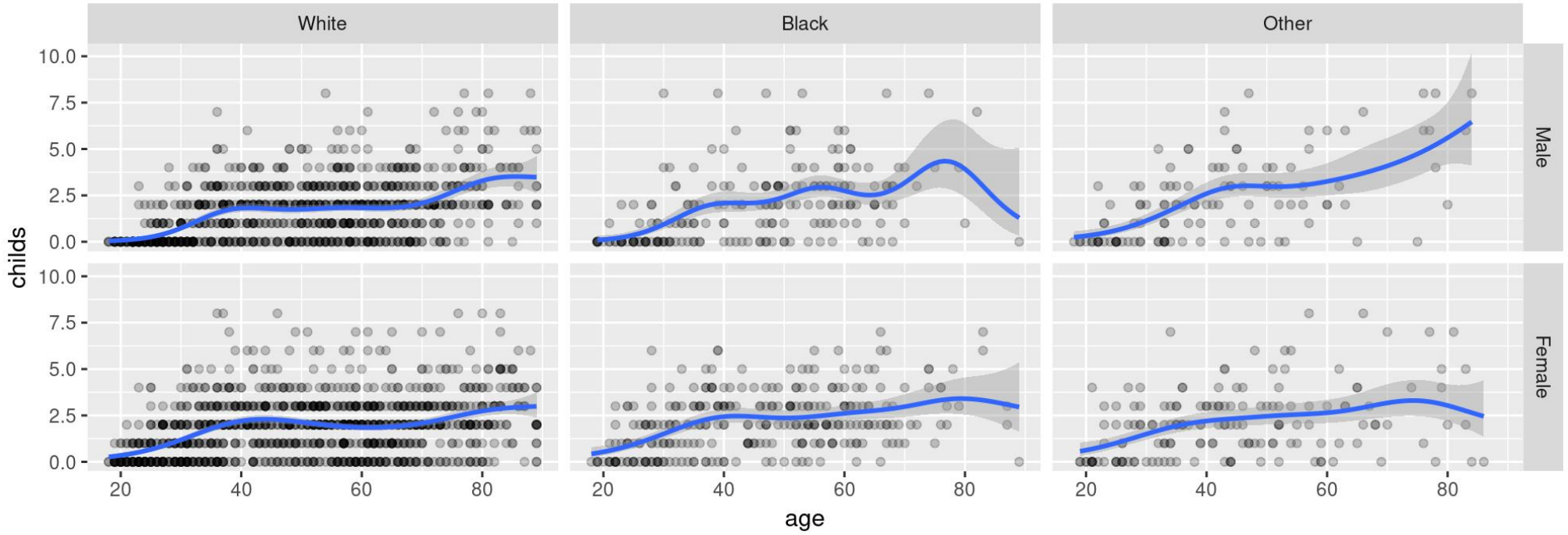
```
p +
  geom_line(colour = 'grey70', aes(group = country)) +
  geom_smooth(size = 1.1, method = 'loess', se = FALSE) +
  scale_y_log10(labels = scales::dollar) +
  facet_wrap(~ continent, nrow = 1) +
  labs(x = 'Year', y = 'GDP per capita')
```



Small multiples — `facet_grid()`

Partition is specified using a formula: `f1 ~ f2`

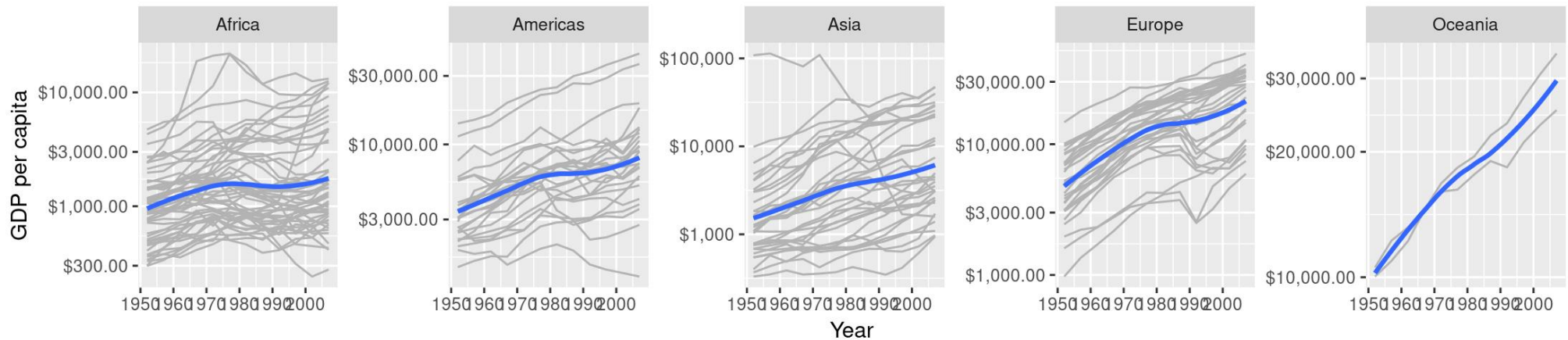
```
ggplot(gss_sm, aes(x = age, y = childs)) + geom_point(alpha = 0.2) +
  geom_smooth(method = 'gam', formula = y ~ s(x),
              method.args = list(family = poisson)) +
  facet_grid(sex ~ race)
```



Small multiples — `scales`

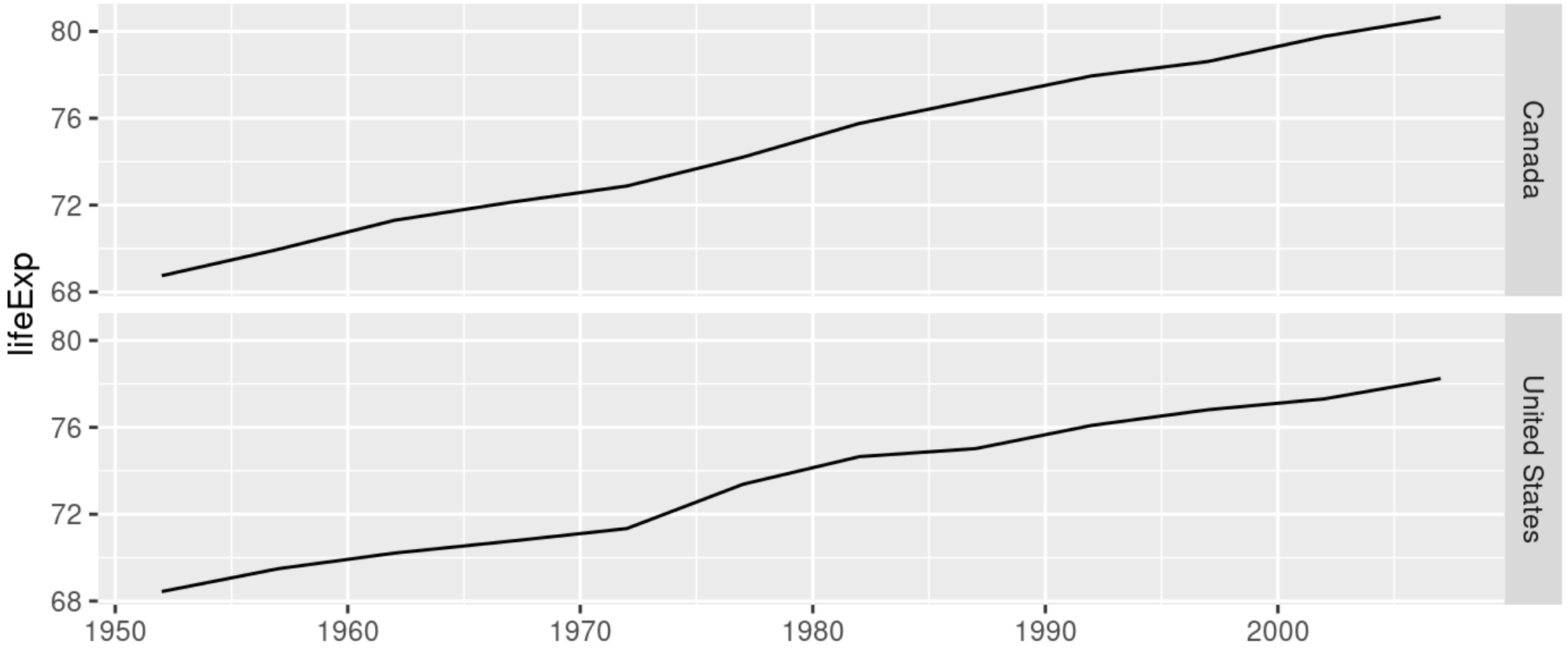
`scales` can be any of `"free_x"`, `"free_y"`, or `"free"`

```
p +
  geom_line(colour = 'grey70', aes(group = country)) +
  geom_smooth(size = 1.1, method = 'loess', se = FALSE) +
  scale_y_log10(labels = scales::dollar) +
  facet_wrap(~ continent, nrow = 1, scales = 'free_y') +
  labs(x = 'Year', y = 'GDP per capita')
```



Faceting time series

```
ggplot(subset(gapminder, subset = country %in% c('United States','Canada')),
       aes(x = year, y = lifeExp)) + geom_line() + labs(x = NULL) +
  facet_grid(country ~ .)
```



Stats

Ashraf Ali

Stats

Some geoms plot the data directly, other geoms manipulate the data before plotting

The manipulation is done by a `stat_xxx()` function, or *stat*

- Each *geom* has a default *stat*
- Each *stat* has a default *geom*

General Social Survey 2016

The GSS is a long-running survey of American adults

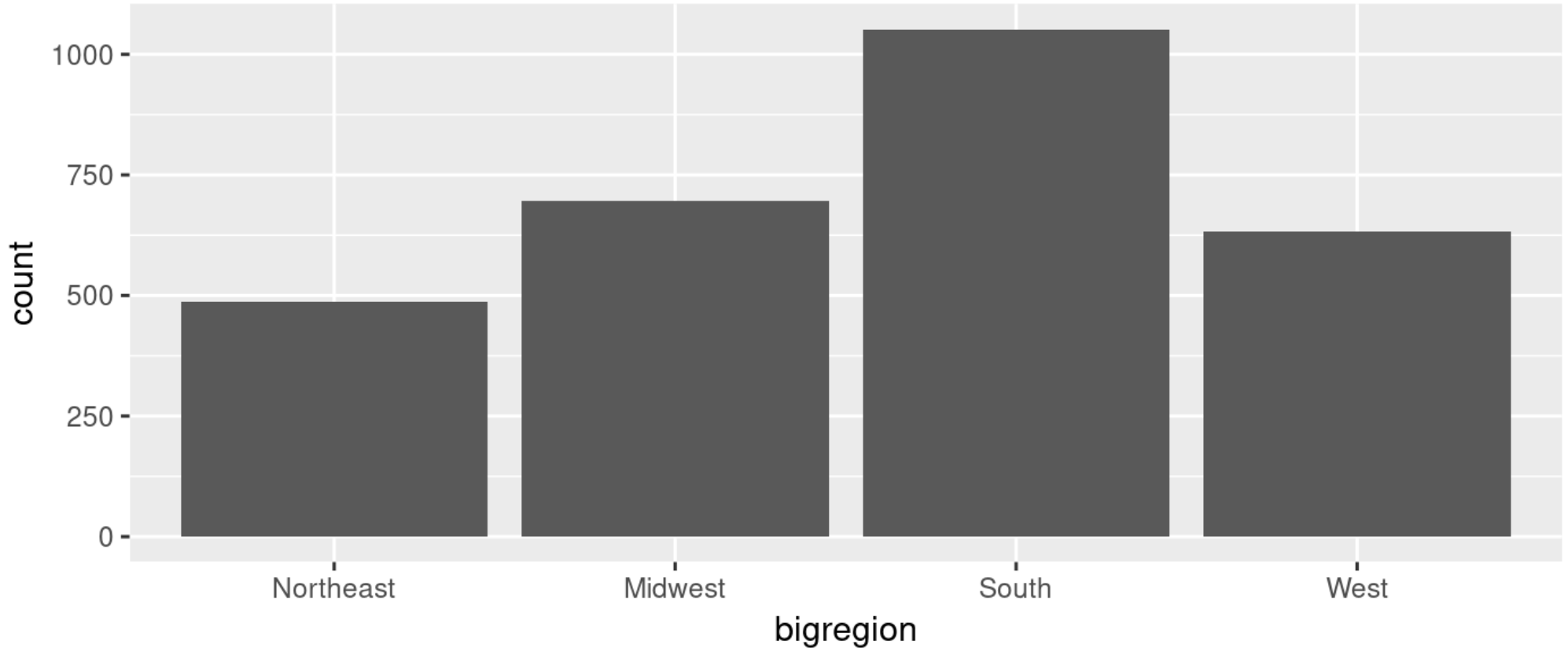
`gss_sm` is a small subset of the full GSS for 2016

```
## # A tibble: 6 x 32
##    year    id ballot   age childs  sibs degree race  sex   region income16
##   <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl> <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 …
## # … with 21 more variables: relig <fct>, marital <fct>, padeg <fct>,
## #   madeg <fct>, 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>
```

Also try `glimpse(gss_sm)`

Stats — what happened here?

```
ggplot(gss_sm, aes(x = bigregion)) + geom_bar()
```



`stat_count()`

The default stat for `geom_bar()` is `stat_count()`

It counts the number of observations in each group

Stats create temporary variables that we can use — this is where `count` came from

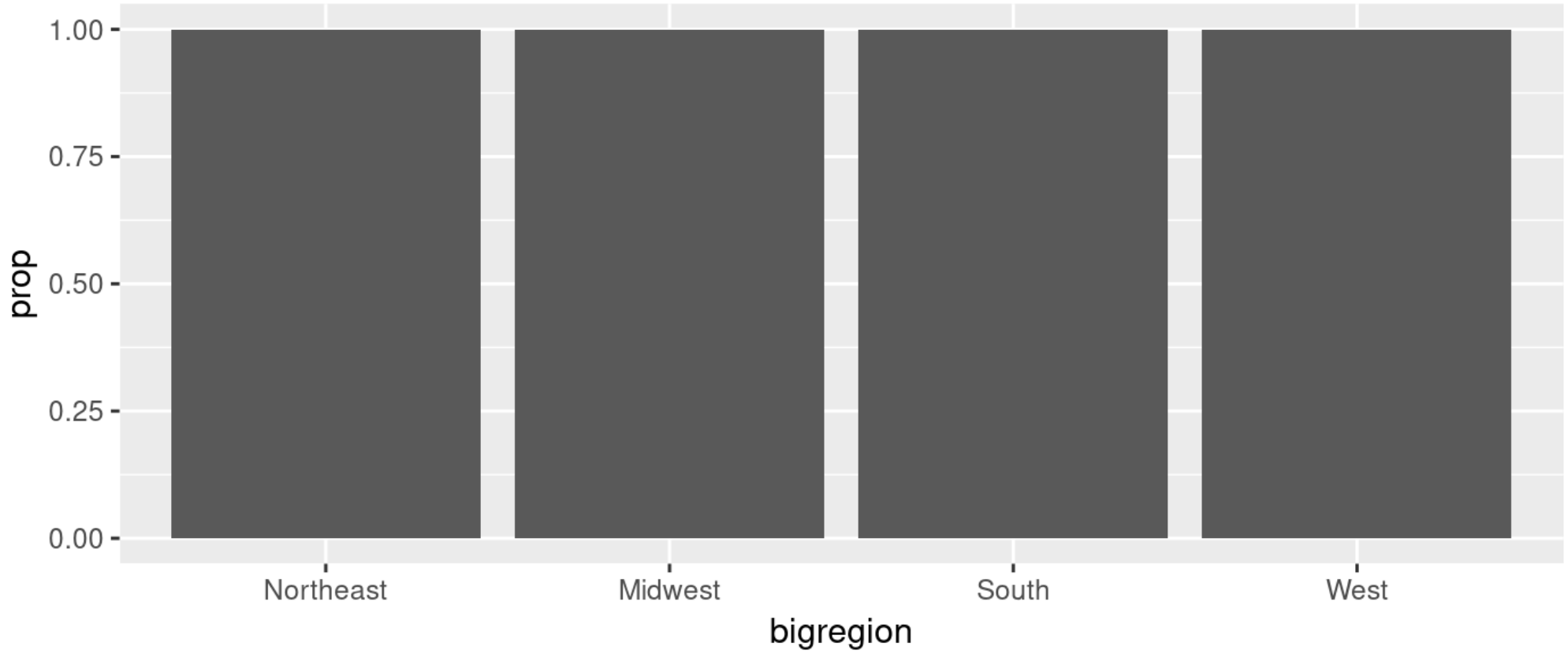
Temporary variables are named `..name..`

`stat_count()` creates:

- `..count..`
- `..prop..`

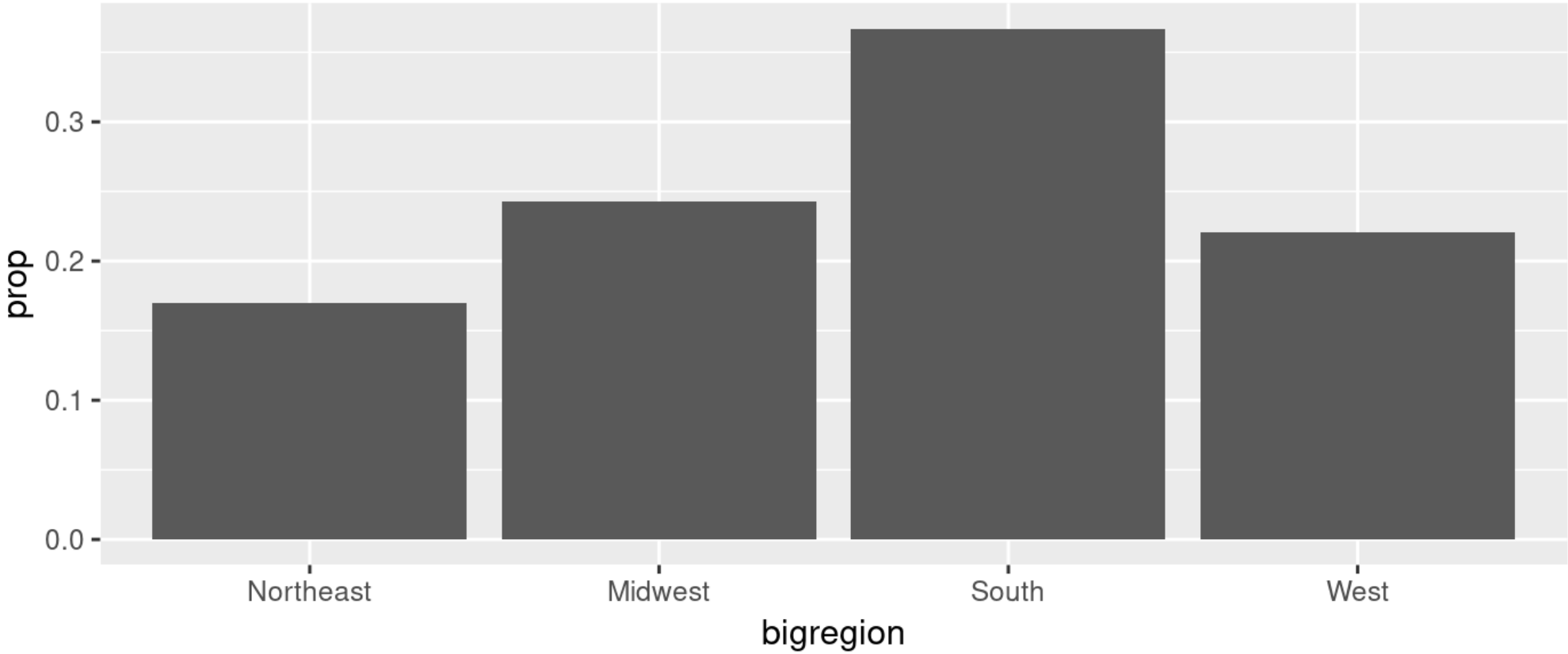
Stats — what happened here?

```
ggplot(gss_sm, aes(x = bigregion)) + geom_bar(mapping = aes(y = ..prop..))
```

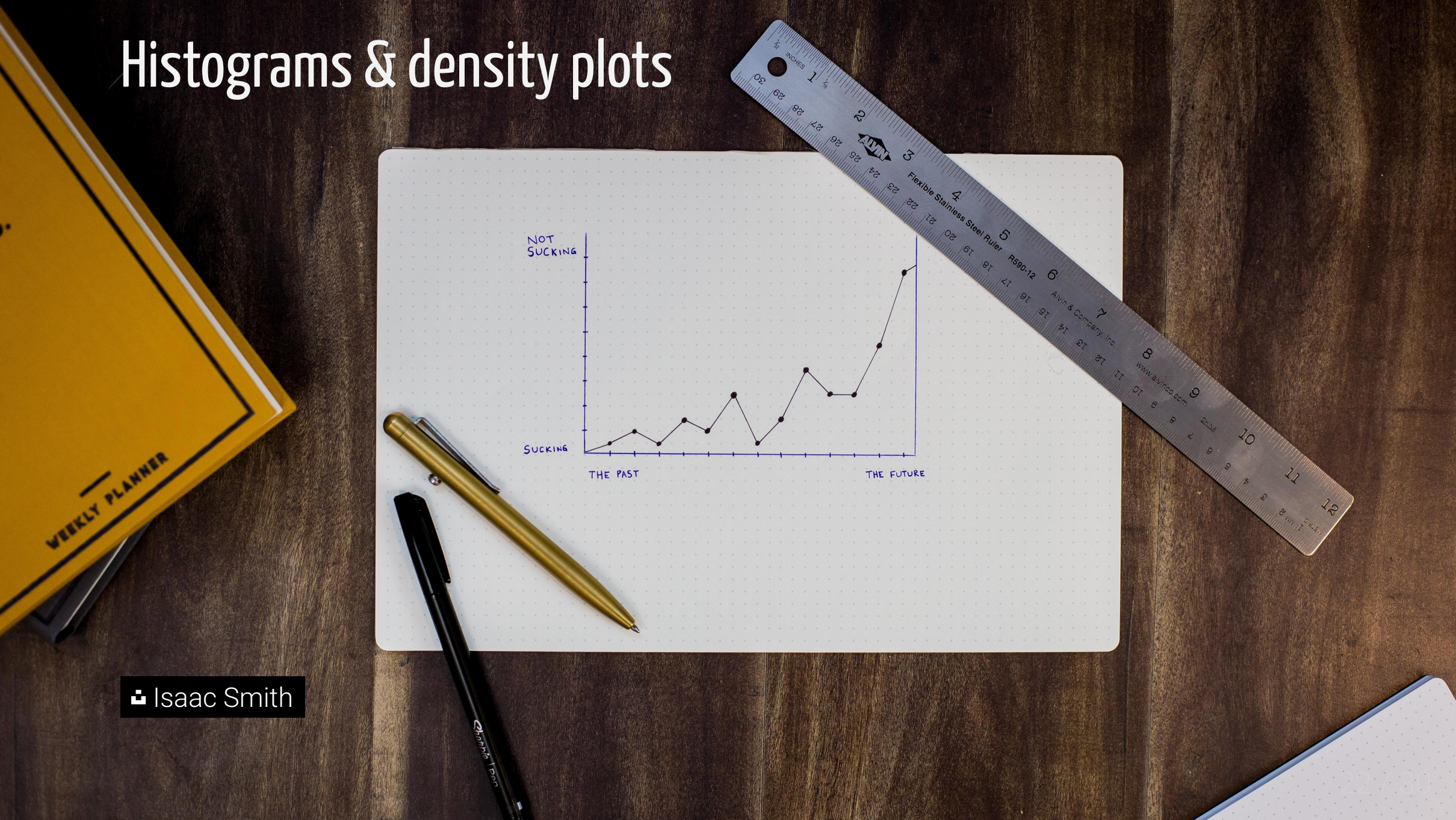


Stats — grouping

```
ggplot(gss_sm, aes(x = bigregion)) +
  geom_bar(mapping = aes(y = ..prop.., group = 1))
```



Histograms & density plots



Isaac Smith

midwest

Demographic data on midwest counties of the USA

```
## # A tibble: 437 x 28
##      PID county state  area poptotal popdensity popwhite popblack
##    <int> <chr>  <chr> <dbl>    <int>      <dbl>    <int>    <int>
##  1   561 ADAMS  IL    0.052    66090      1271.    63917     1702
##  2   562 ALEXA… IL    0.014    10626       759      7054     3496
##  3   563 BOND   IL    0.022    14991       681.    14477      429
##  4   564 BOONE  IL    0.017    30806      1812.    29344      127
##  5   565 BROWN  IL    0.018     5836       324.     5264      547
##  6   566 BUREAU IL    0.05     35688       714.    35157       50
##  7   567 CALHO… IL    0.017     5322       313.     5298        1
##  8   568 CARRO… IL    0.027    16805       622.    16519      111
##  9   569 CASS   IL    0.024    13437       560.    13384       16
## 10   570 CHAMP… IL    0.058   173025      2983.   146506    16559
## # … with 427 more rows, and 20 more variables: popamerindian <int>,
## #   popasian <int>, popother <int>, percwhite <dbl>, 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>
```

Also try `glimpse(midwest)`

Histograms

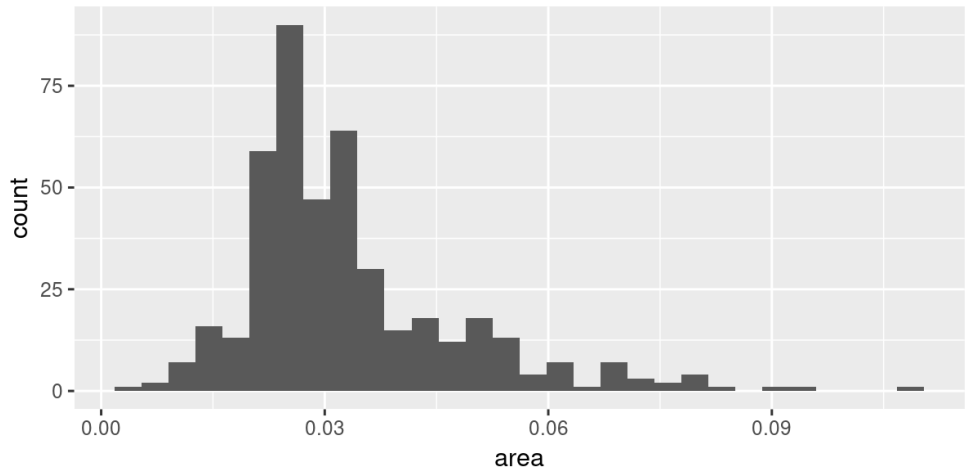
Histograms chop the data into segments known as *bins*

Observations within each bin are counted and possibly converted to a *density*

A histogram is a series of bars showing the count or density in each bin

`geom_histogram()`

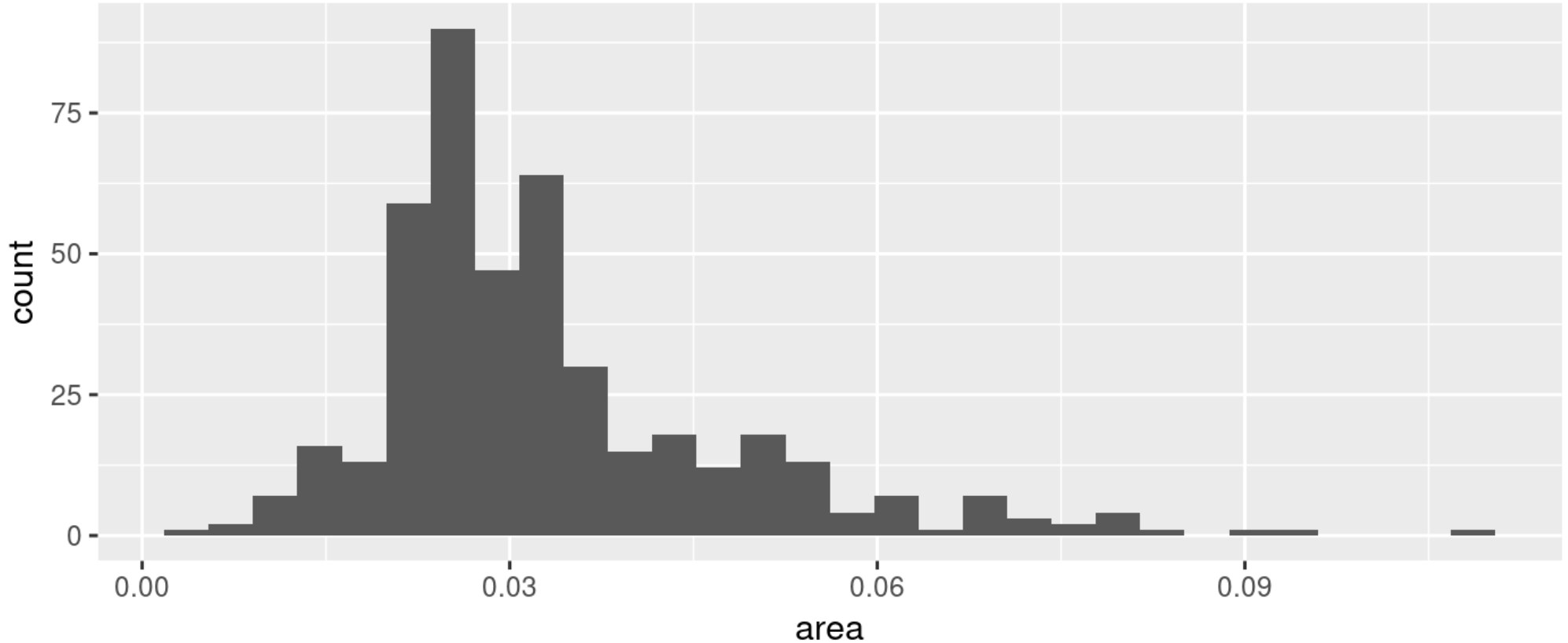
```
ggplot(midwest, aes(x = area)) +
  geom_histogram()
```



Histograms

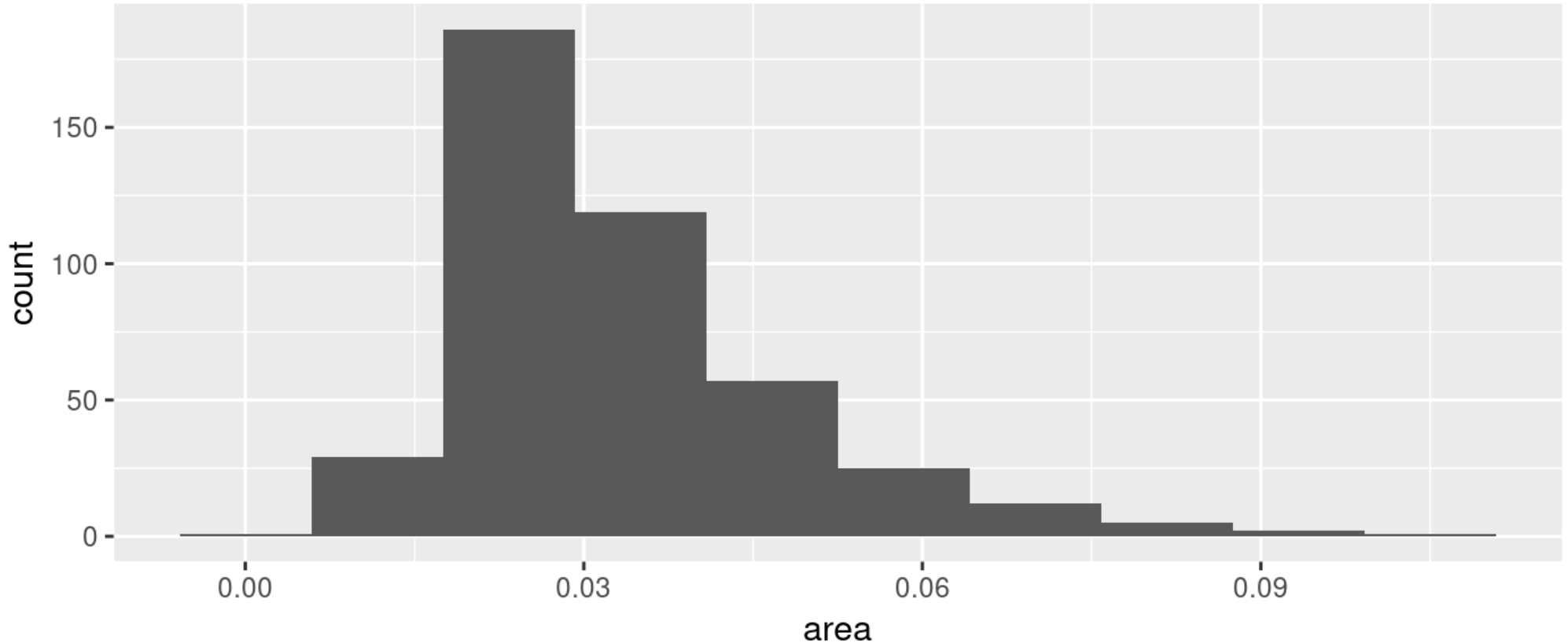
```
ggplot(midwest, aes(x = area)) + geom_histogram()
```

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



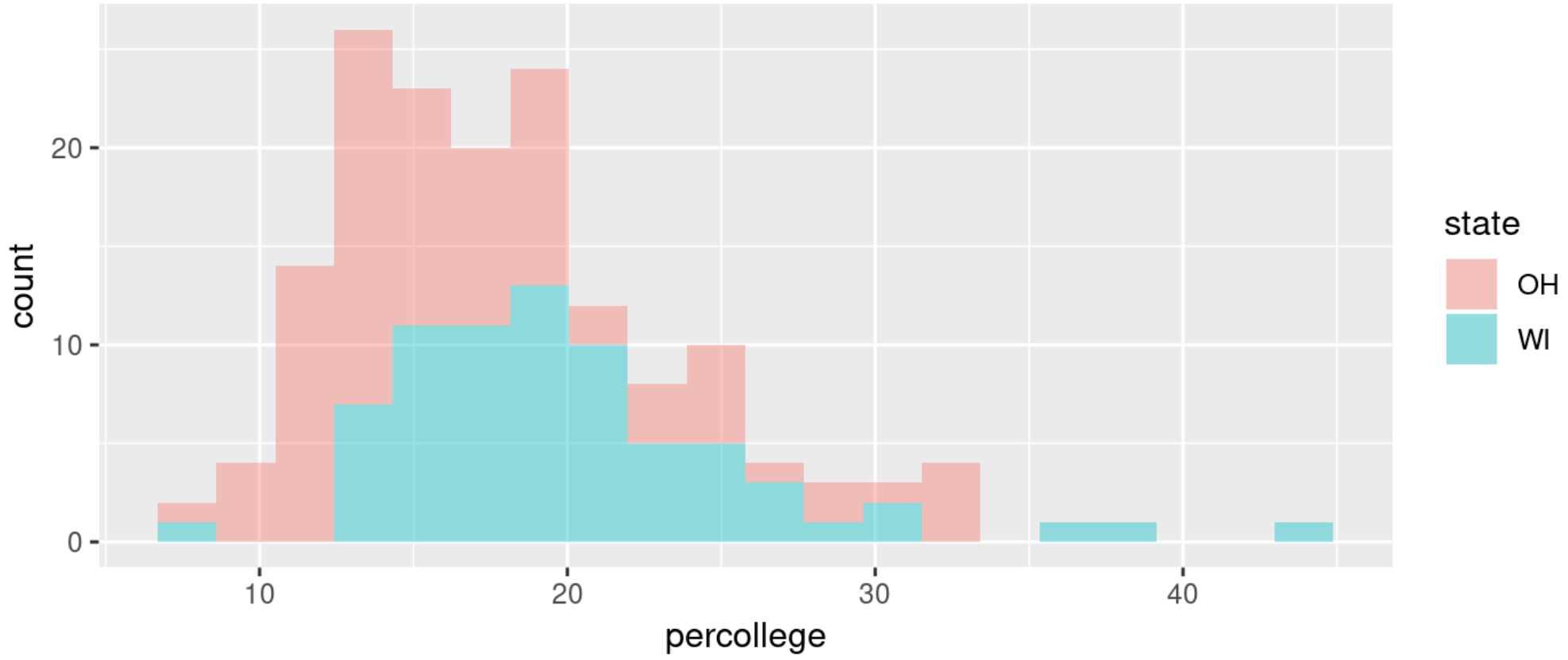
Histograms

```
nbin <- with(midwest, nclass.Sturges(area)) ## 10
ggplot(midwest, aes(x = area)) + geom_histogram(bins = nbin)
```



Histograms

```
ggplot(subset(midwest, subset = state %in% c('OH','WI')),
       aes(x = percollege, fill = state)) +
  geom_histogram(bins = 20, alpha = 0.4)
```



Density plots

Density plots are a smooth form of histogram

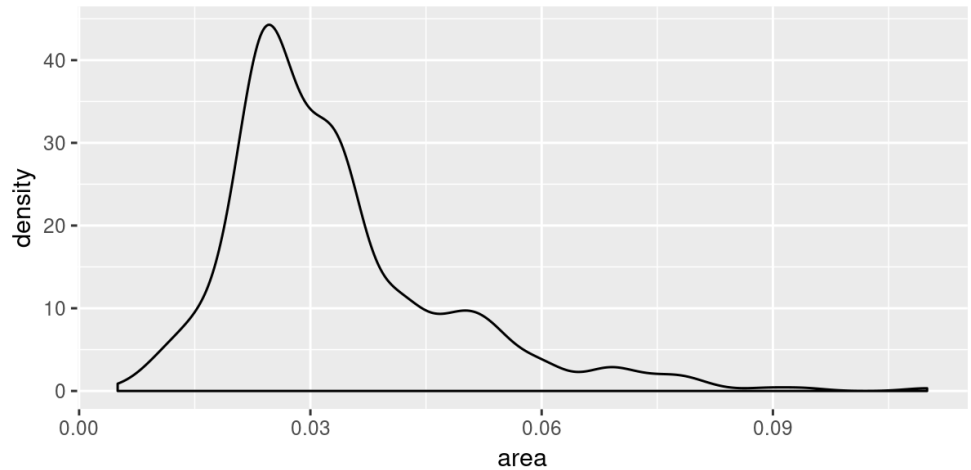
Density is estimated via a *kernel density estimator*

`geom_density()`

Default base geom is `geom_area()`

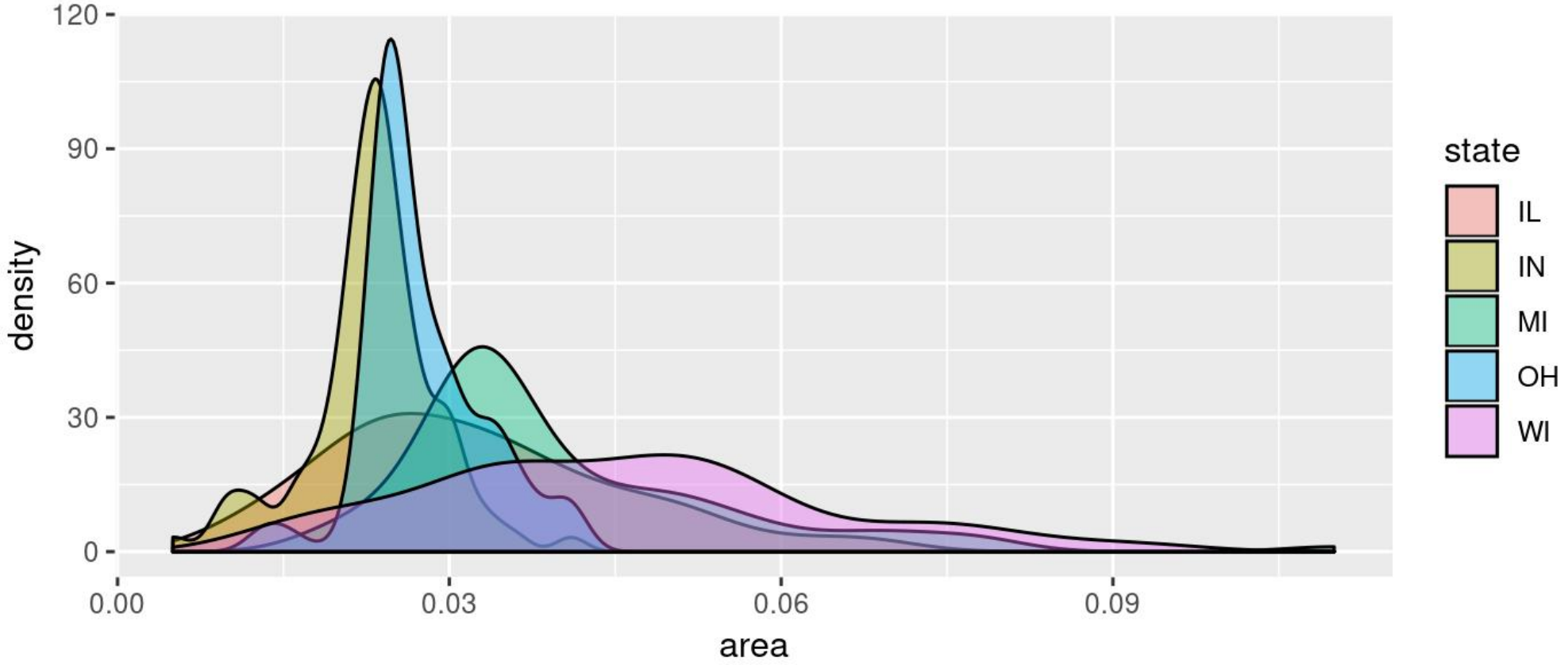
`fill` and `colour` aesthetics

```
ggplot(midwest, aes(x = area)) +
  geom_density()
```



Density plots

```
ggplot(midwest, aes(x = area, fill = state)) +
  geom_density(alpha = 0.4)
```



Density plots — alternative

```
ggplot(midwest, aes(x = area)) +
  geom_line(stat = 'density')
```