Gov 50: 3. Data Visualization

Matthew Blackwell

Harvard University

Roadmap

1. Building plots by layers
2. Histograms and boxplots
3. Grouped data

1/ Building plots by layers

Midwest data

```r
midwest
```

```
## # A tibble: 437 x 28
##      PID county    state  area popto~1 popde~2 popwh~3 popbl~4 popam~5
##    <int> <chr>     <chr> <dbl>   <int>   <dbl>   <int>   <int>   <int>
##  1   561 ADAMS     IL    0.052   66090   1271.   63917    1702      98
##  2   562 ALEXANDER IL    0.014   10626    759     7054    3496      19
##  3   563 BOND      IL    0.022   14991    681.   14477     429      35
##  4   564 BOONE     IL    0.017   30806   1812.   29344     127      46
##  5   565 BROWN     IL    0.018    5836    324.    5264     547      14
##  6   566 BUREAU    IL    0.05    35688    714.   35157      50      65
##  7   567 CALHOUN   IL    0.017    5322    313.    5298       1       8
##  8   568 CARROLL   IL    0.027   16805    622.   16519     111      30
##  9   569 CASS      IL    0.024   13437    560.   13384      16       8
## 10   570 CHAMPAIGN IL    0.058  173025   2983.  146506   16559     331
## # ... with 427 more rows, 19 more variables: 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>, ...
```

Building up a graph in pieces

Create ggplot object and direct it to the correct data:

```
p <- ggplot(data = midwest)
```

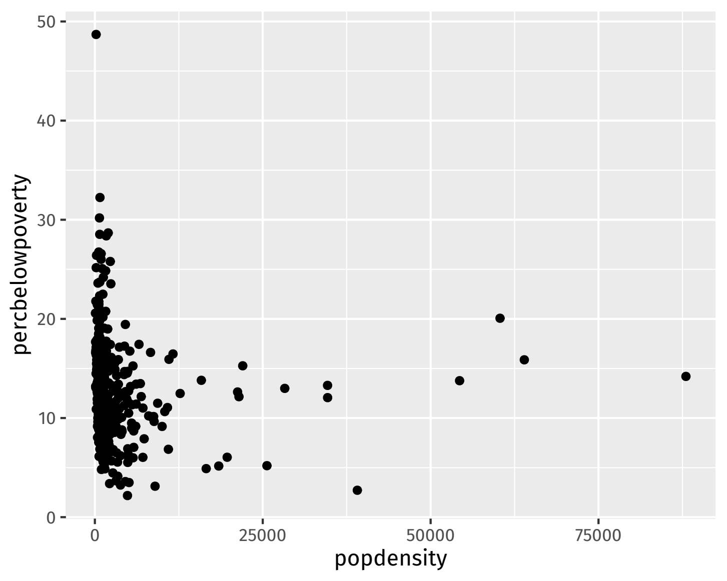
Mapping: tell ggplot what visual aesthetics correspond to which variables

```
p <- ggplot(data = midwest,
            mapping = aes(x = popdensity,
                          y = percbelowpoverty))
```

Other aesthetic mappings: `color`, `shape`, `size`, etc.

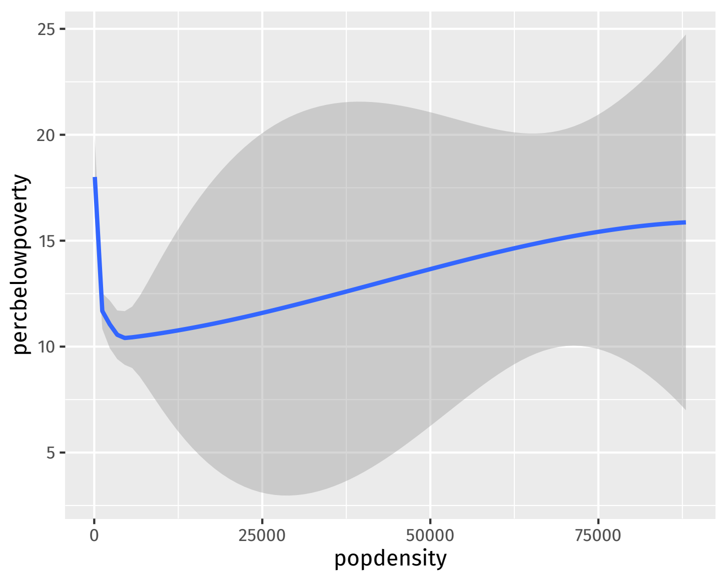
Adding a geom layer

```r
ggplot(data = midwest,
       mapping = aes(x = popdensity,
                     y = percbelowpoverty)) +
  geom_point()
```


50
40
30
20
10
0
percbelowpoverty
0
25000
50000
75000
popdensity

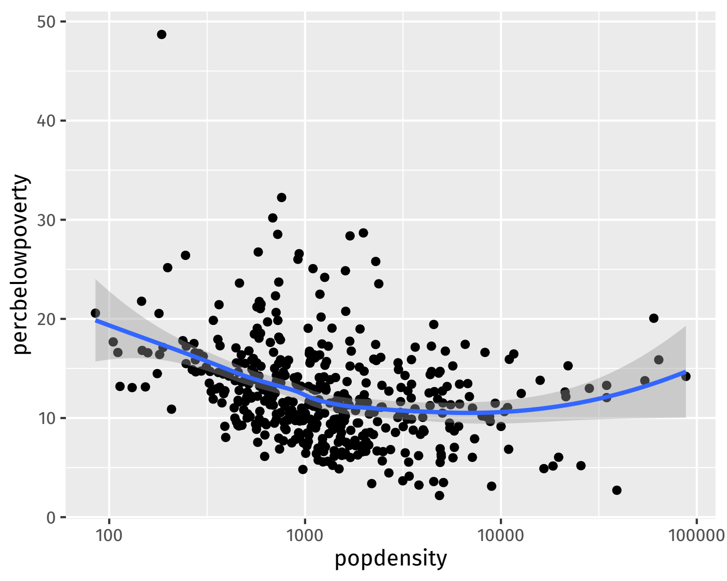
Trying a new geom

```
ggplot(data = midwest,
       mapping = aes(x = popdensity,
                     y = percbelowpoverty)) +
  geom_smooth()
```


25
20
15
10
5
percbelowpoverty
0
25000
50000
75000
popdensity

Layering geoms is additive

```r
ggplot(data = midwest,
       mapping = aes(x = popdensity,
                     y = percbelowpoverty)) +
  geom_point() +
  geom_smooth() +
  scale_x_log10()
```

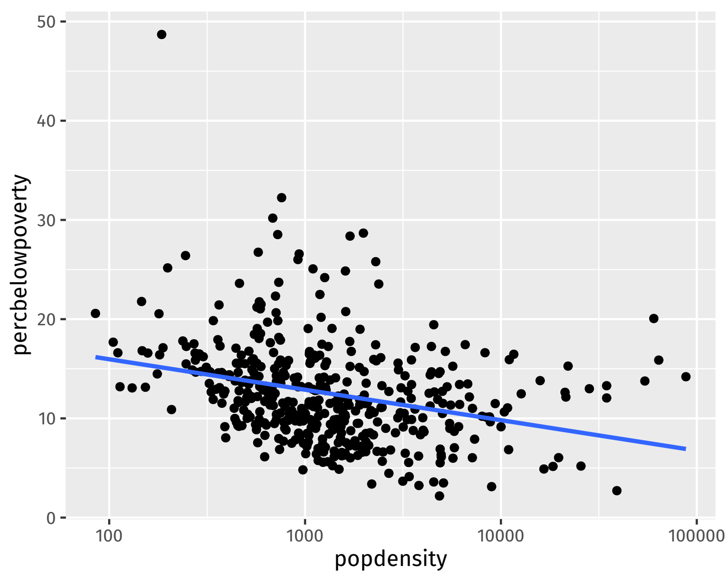

50
40
30
20
10
0
percbelowpoverty
100
1000
10000
100000
popdensity

Geoms are functions

Geoms can take arguments:

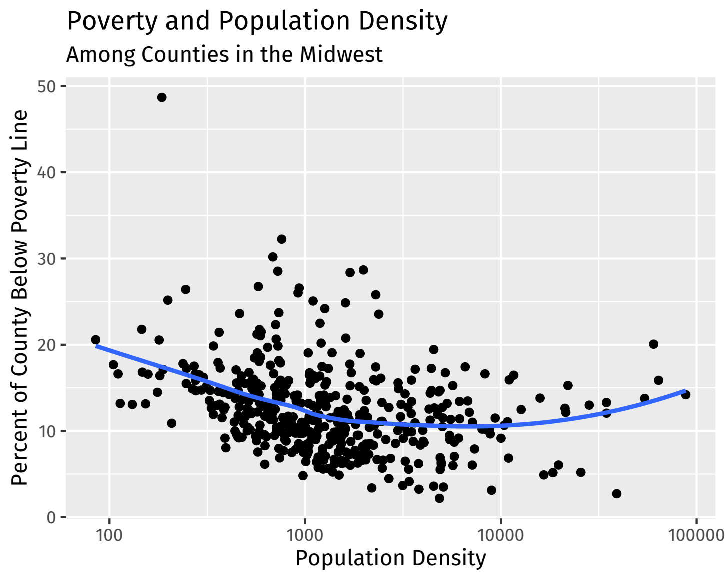
```
ggplot(data = midwest,
       mapping = aes(x = popdensity,
                     y = percbelowpoverty)) +
  geom_point() +
  geom_smooth(method = ”lm”, se = FALSE) +
  scale_x_log10()
```

Tells geom_smooth to do a linear fit with no error region


50
40
30
20
10
0
percbelowpoverty
100
1000
10000
100000
popdensity

Adding informative labels

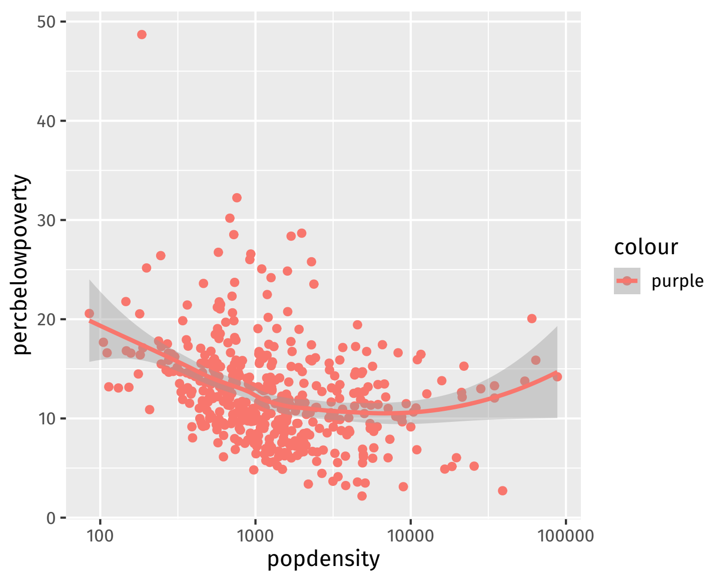
```
ggplot(data = midwest,
       mapping = aes(x = popdensity,
                     y = percbelowpoverty)) +
  geom_point() +
  geom_smooth(method = ”loess”, se = FALSE) +
  scale_x_log10() +
  labs(x = ”Population Density”,
       y = ”Percent of County Below Poverty Line”,
       title = ”Poverty and Population Density”,
       subtitle = ”Among Counties in the Midwest”,
       source = ”US Census, 2000”)
```


Poverty and Population Density
Among Counties in the Midwest
Percent of County Below Poverty Line
0
10
20
30
40
50
100
1000
10000
100000
Population Density

Mapping vs setting aesthetics

```
ggplot(data = midwest,
       mapping = aes(x = popdensity,
                     y = percbelowpoverty,
                     color = ”purple”)) +
  geom_point() +
  geom_smooth() +
  scale_x_log10()
```

Wait what?



50
40
30
20
10
0
100
1000
10000
100000
popdensity
percbelowpoverty
colour
purple

Mapping always refers to variables

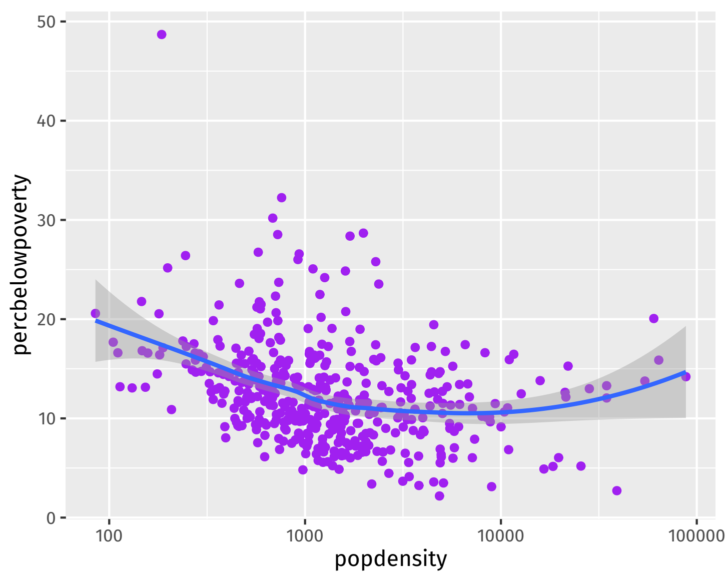
If passed a value other than a variable name, ggplot will implicitly create a variable with that value (in this case ”purple” that is constant)

```
ggplot(data = midwest,
       mapping = aes(x = popdensity,
                     y = percbelowpoverty,
                     color = ”purple”)) +
  geom_point() +
  geom_smooth() +
  scale_x_log10()
```

Setting aesthetics

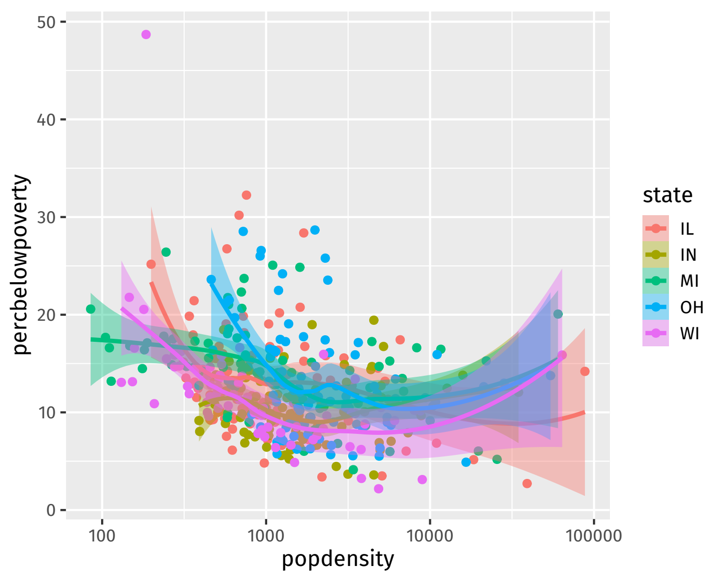
Set the color outside the `mapping = aes()` format.

```
ggplot(data = midwest,
       mapping = aes(x = popdensity,
                     y = percbelowpoverty)) +
  geom_point(color = ”purple”) +
  geom_smooth() +
  scale_x_log10()
```


50
40
30
20
10
0
100
1000
10000
100000
percbelowpoverty
popdensity

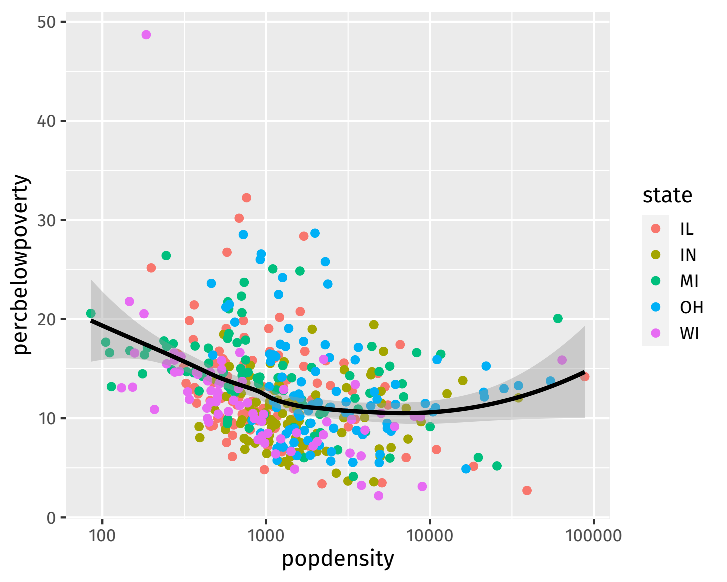
Mapping more aesthetics

```r
ggplot(data = midwest,
       mapping = aes(x = popdensity,
                     y = percbelowpoverty,
                     color = state,
                     fill = state)) +
  geom_point() +
  geom_smooth() +
  scale_x_log10()
```


50
40
30
20
10
0
percbelowpoverty
100
1000
10000
100000
popdensity
state
IL
IN
MI
OH
WI

Mappings can be done on a per geom basis

```
ggplot(data = midwest,
       mapping = aes(x = popdensity,
                     y = percbelowpoverty)) +
  geom_point(mapping = aes(color = state)) +
  geom_smooth(color = ”black”) +
  scale_x_log10()
```


50
40
30
20
10
0
percbelowpoverty
100
1000
10000
100000
popdensity
state
IL
IN
MI
OH
WI

2/ Histograms and boxplots

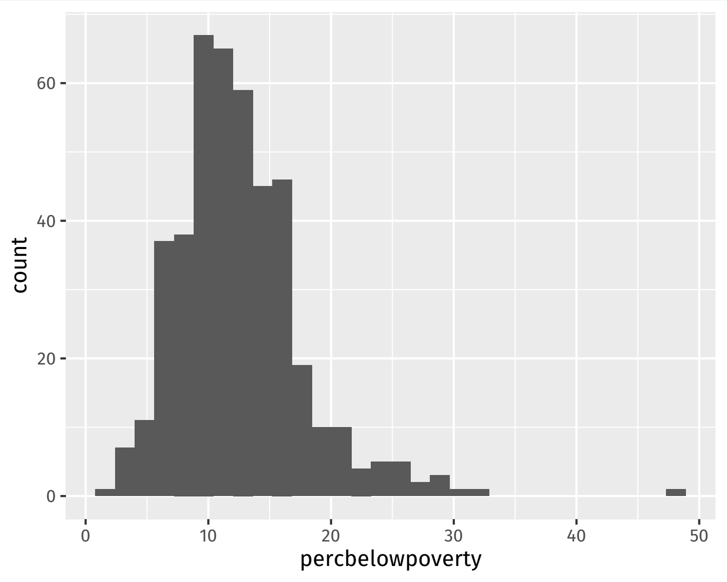
Histograms

Histograms show where there are more or fewer observations of a numeric variable.

```
ggplot(data = midwest,
       mapping = aes(x = percbelowpoverty)) +
  geom_histogram()
```

Split up range of variable into bins, count how many are in each bin.

y aesthetic calculated automatically.

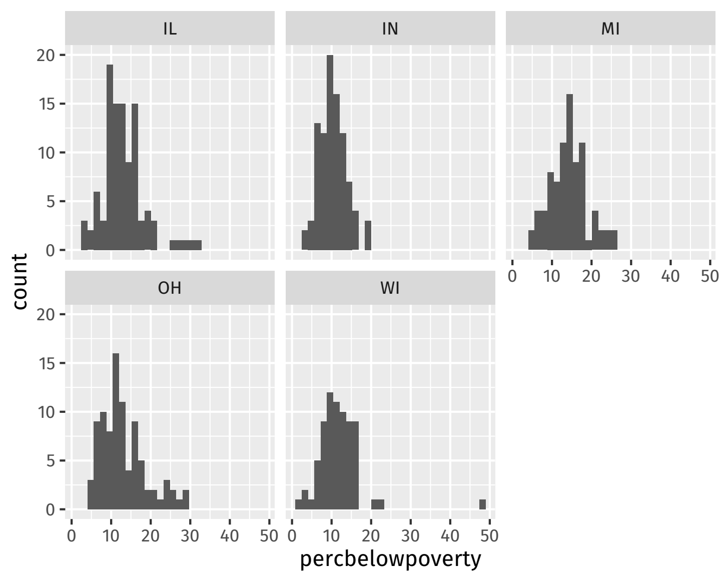

count
0
20
40
60
0
10
20
30
40
50
percbelowpoverty

Creating small multiples with facets

Small multiples: a series of similar graphs with the same scale/axes to help with comparing different partitions of a dataset.

```r
ggplot(data = midwest,
       mapping = aes(x = percbelowpoverty)) +
  geom_histogram() +
  facet_wrap(~ state)
```

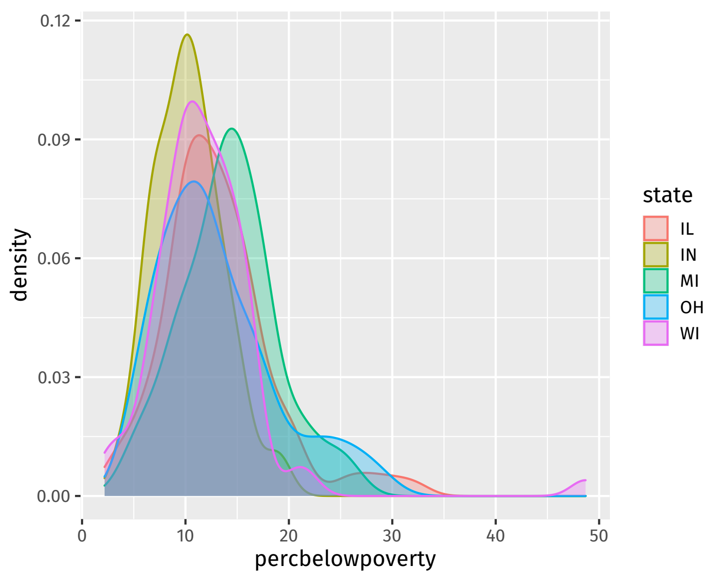
We'll see more of the `~ variable` syntax (called a formula).


IL
IN
MI
OH
WI
count
0
5
10
15
20
0
10
20
30
40
50
percbelowpoverty

Density as alternative to histograms

A **kernel density** plot is a smoothed version of a histogram and slightly easier to overlay.

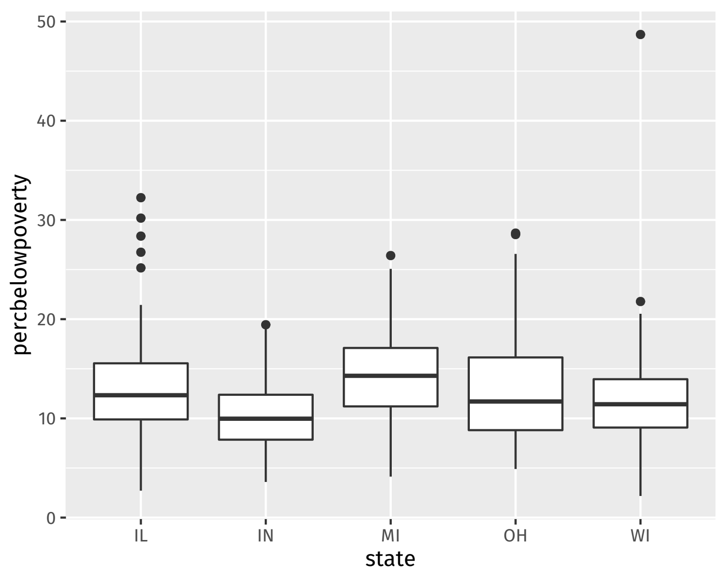
```
ggplot(data = midwest,
       mapping = aes(x = percbelowpoverty,
                     fill = state, color = state)) +
  geom_density(alpha = 0.3)
```


0.12
0.09
0.06
0.03
0.00
density
0
10
20
30
40
50
percbelowpoverty
state
IL
IN
MI
OH
WI

Boxplots

Boxplots are another way to compare distributions across discrete groups.

```r
ggplot(data = midwest,
       mapping = aes(x = state,
                     y = percbelowpoverty)) +
  geom_boxplot()
```


percbelowpoverty
50
40
30
20
10
0
IL
IN
MI
OH
WI
state

Boxplots in R

- “Box” represents middle 50% of the data.
  - 25% of the data above the box, 25% below
  - Width of the box is called the inter quartile range (IQR)
- Horizontal line in the box is the median
  - 50% of the data above the median, 50% below
- “Whiskers” represents either:
  - 1.5 × IQR or max/min of the data, whichever is smaller.
  - Points beyond whiskers are outliers.

3/ Grouped data

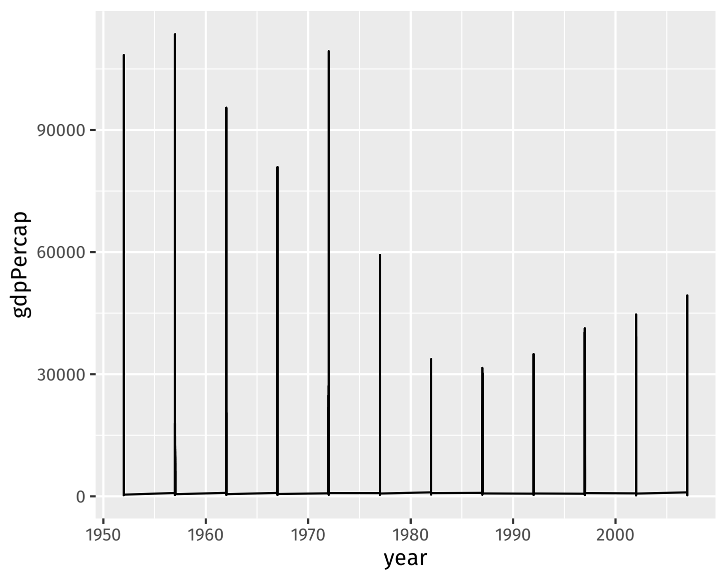
Back to the gapminder data

```r
glimpse(gapminder)
```

```
## Rows: 1,704
## Columns: 6
## $ country   <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afgh~
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, As~
## $ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 19~
## $ lifeExp   <dbl> 28.8, 30.3, 32.0, 34.0, 36.1, 38.4, 39.9, 40.8, 41~
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14~
## $ gdpPercap <dbl> 779, 821, 853, 836, 740, 786, 978, 852, 649, 635, ~
```

Let’s plot the trend in income

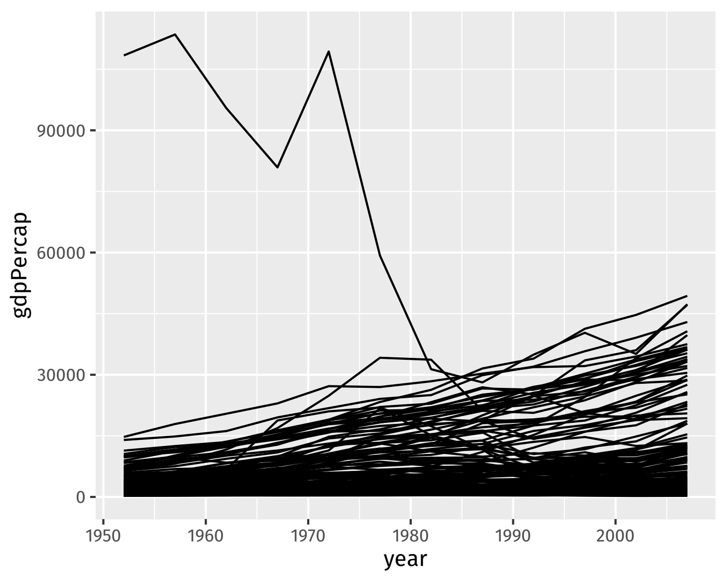
```r
ggplot(data = gapminder,
       mapping = aes(x = year,
                     y = gdpPercap)) +
  geom_line()
```



`geom_line` connects points from different countries in the same year.

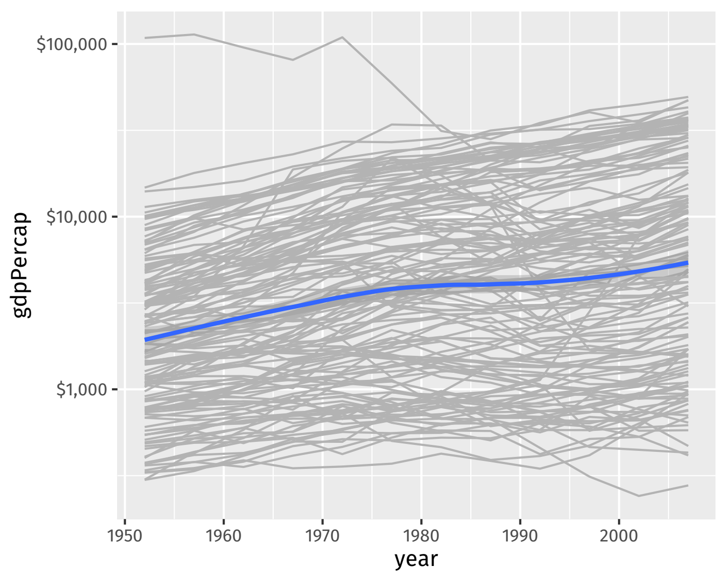
Tell geom_line how to group the lines

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


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

Scales

```
ggplot(data = gapminder,
       mapping = aes(x = year,
                     y = gdpPercap)) +
  geom_line(mapping = aes(group = country), color = ”grey70”) +
  geom_smooth(method = ”loess”) +
  scale_y_log10(labels = scales::dollar)
```


$100,000
$10,000
$1,000
gdpPercap
1950
1960
1970
1980
1990
2000
year