Class 5 - Data Visualization with ggplot2

Edward Vo (A16215241)

Using GGPLOT

The ggplot2 package needs to installed as it does not come with R "out of the box".

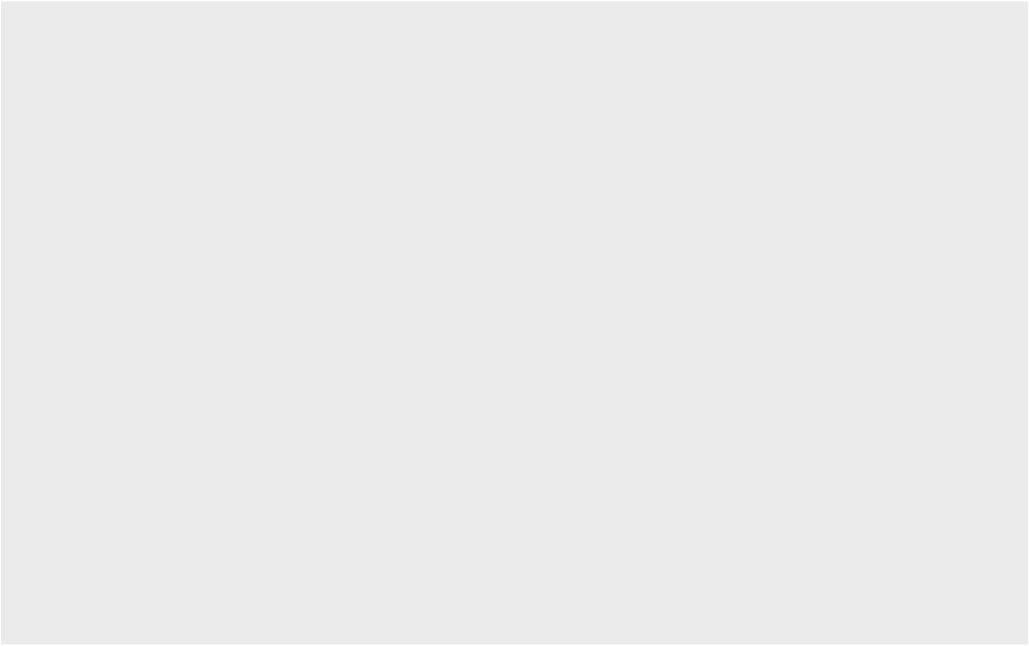
We use the `install.packages()` function to do this.

```
head(cars)
```

```
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
```

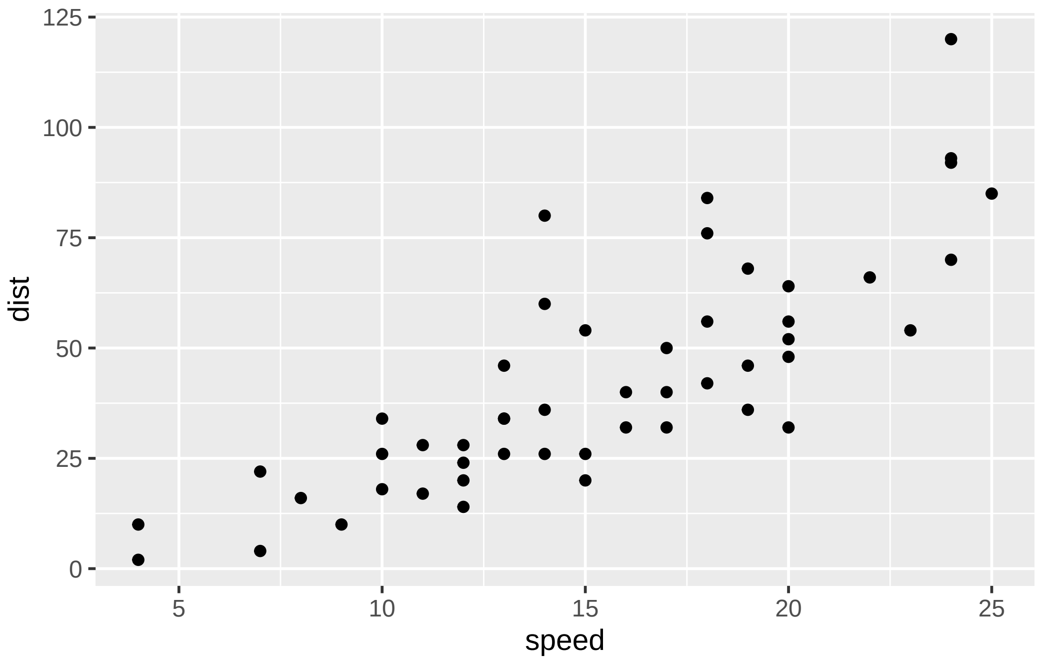
To use ggplot I need to load it up before I can calla ny of the functions in the package. I do this with the `library()` function.

```
library(ggplot2)
ggplot()
```



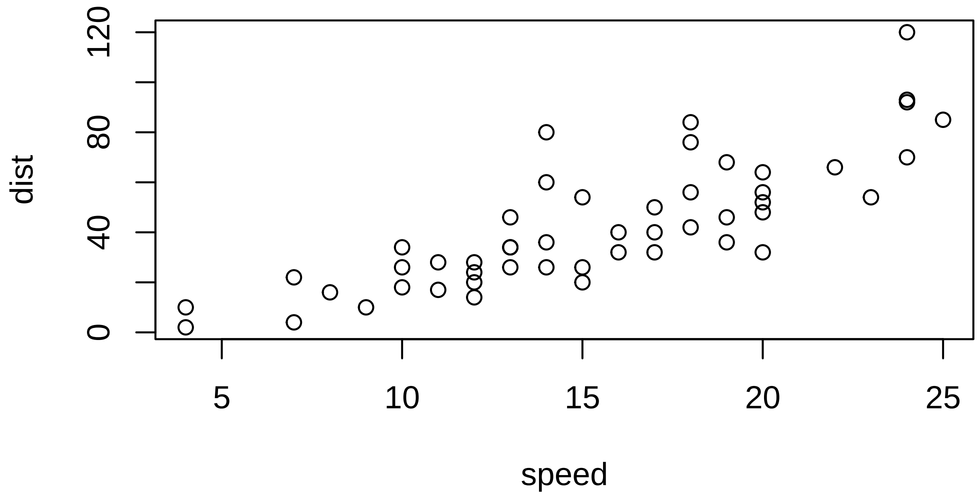
All ggplot figures have at least 3 things: - data (the stuff we want to plot) - aesthetic mapping (aes values) - geoms

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



ggplot is not the only graphing system in R there are lots of others. There is even “base R” graphics.

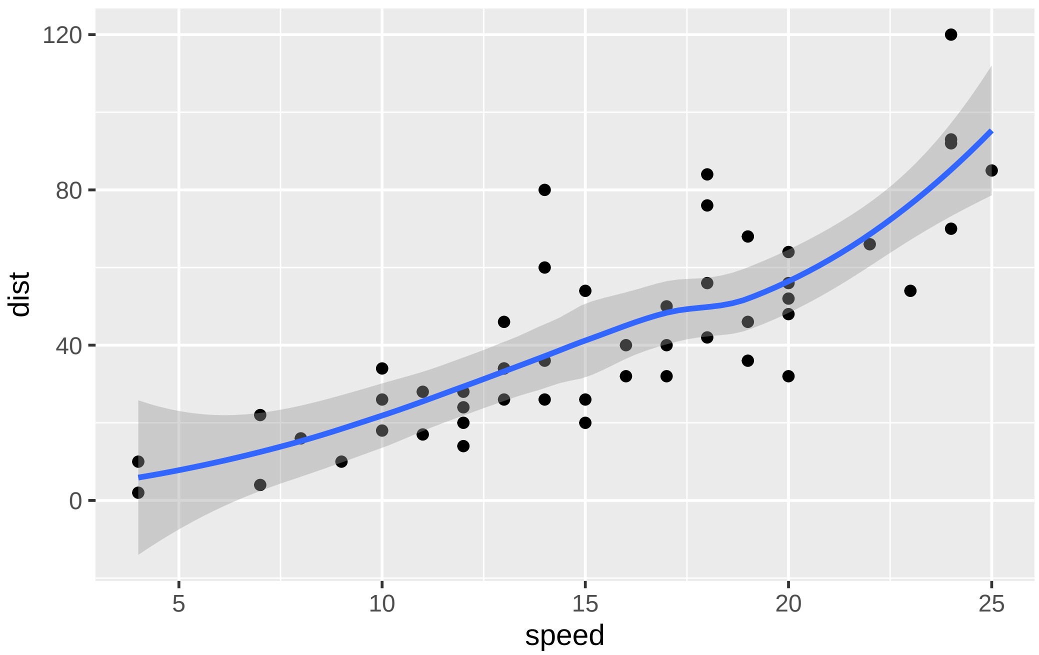
```
plot(cars)
```



Show relationship with smooth

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth()
```

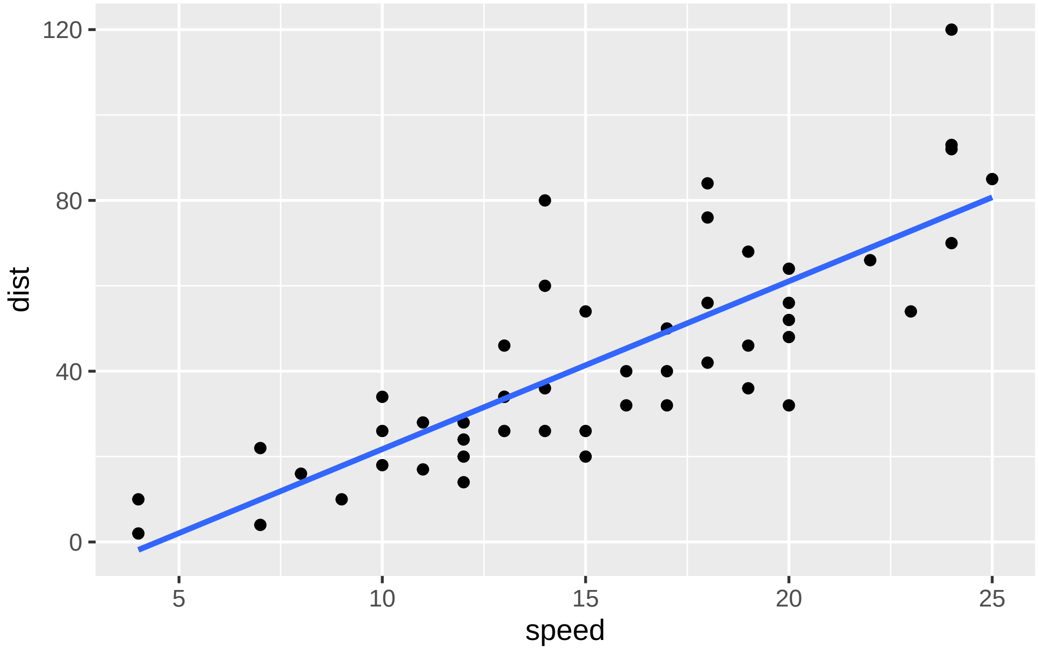
```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Straight line, no shade

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)
```

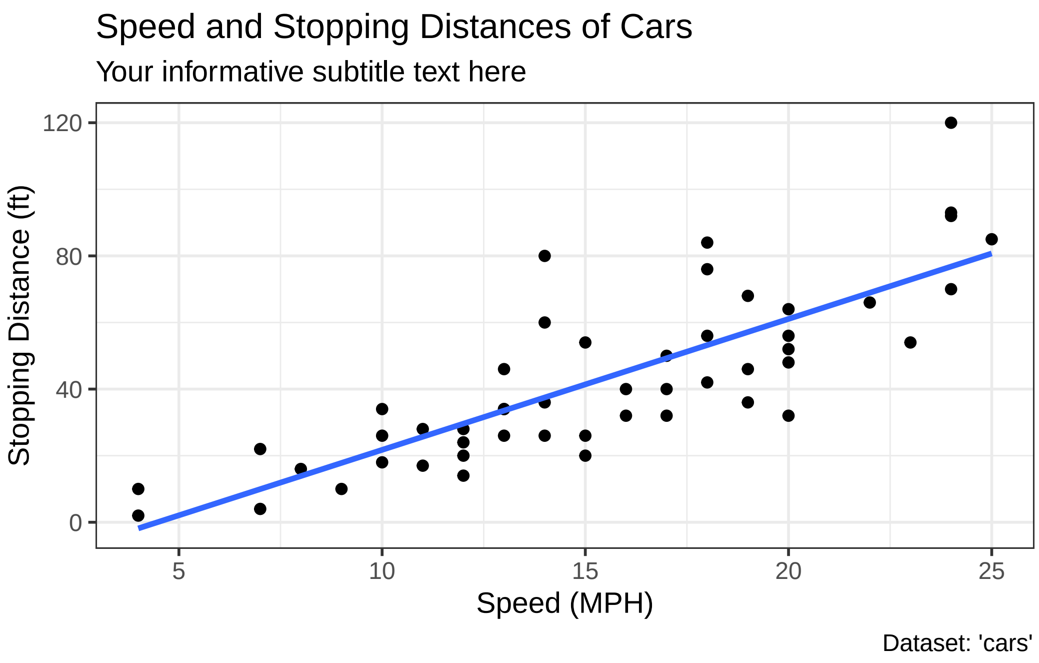
```
`geom_smooth()` using formula = 'y ~ x'
```



Label & change theme

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  labs(title="Speed and Stopping Distances of Cars",
       x="Speed (MPH)",
       y="Stopping Distance (ft)",
       subtitle = "Your informative subtitle text here",
       caption="Dataset: 'cars'") +
  geom_smooth(method="lm", se=FALSE) +
  theme_bw()
```

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



Section 6

Load data

```
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
genes <- read.delim(url)
head(genes)
```

```
        Gene Condition1 Condition2      State
1      A4GNT -3.6808610 -3.4401355 unchanging
2       AAAS  4.5479580  4.3864126 unchanging
3      AASDH  3.7190695  3.4787276 unchanging
4       AATF  5.0784720  5.0151916 unchanging
5       AATK  0.4711421  0.5598642 unchanging
6 AB015752.4 -3.6808610 -3.5921390 unchanging
```

Check # of rows and columns

```
nrow(genes)
```

```
[1] 5196
```

```
ncol(genes)
```

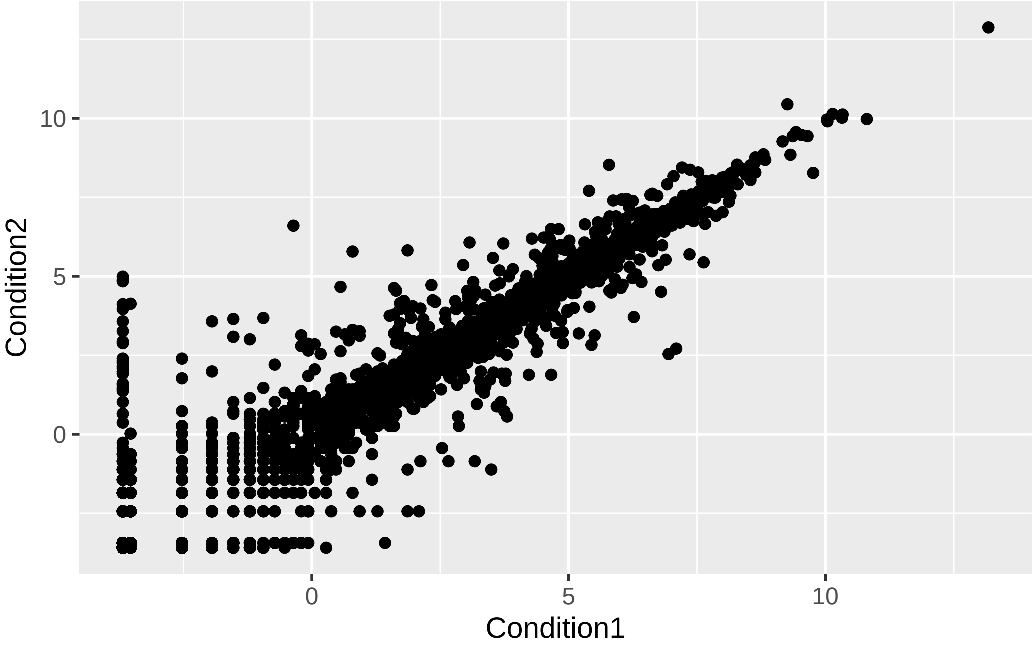
```
[1] 4
```

```
127/5196
```

```
[1] 0.02444188
```

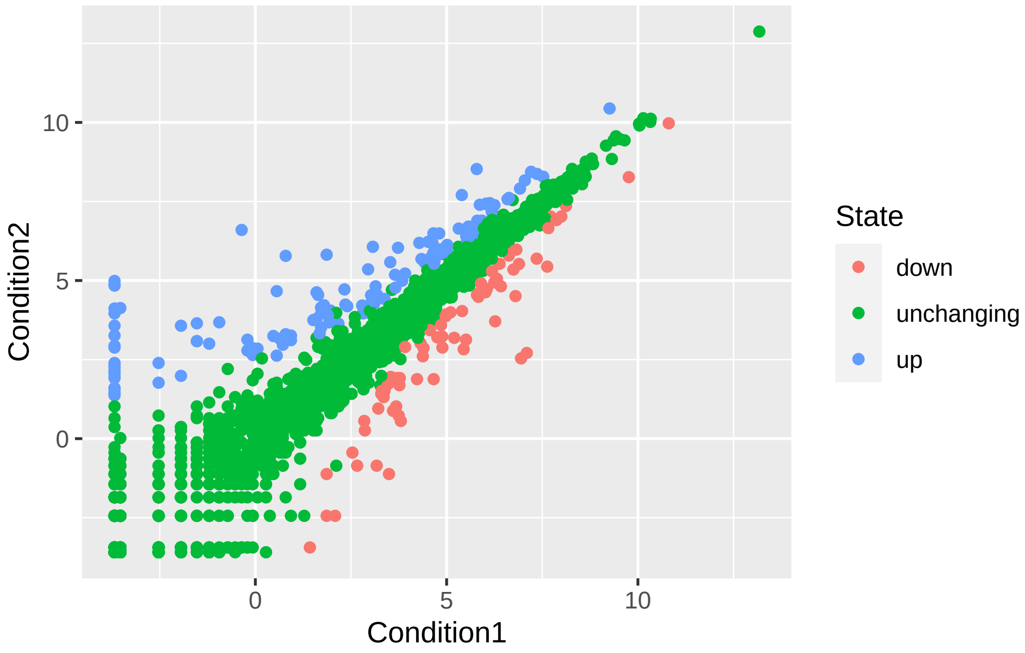
Set up genes scatter plot

```
ggplot(genes) +
    aes(x=Condition1, y=Condition2) +
    geom_point()
```



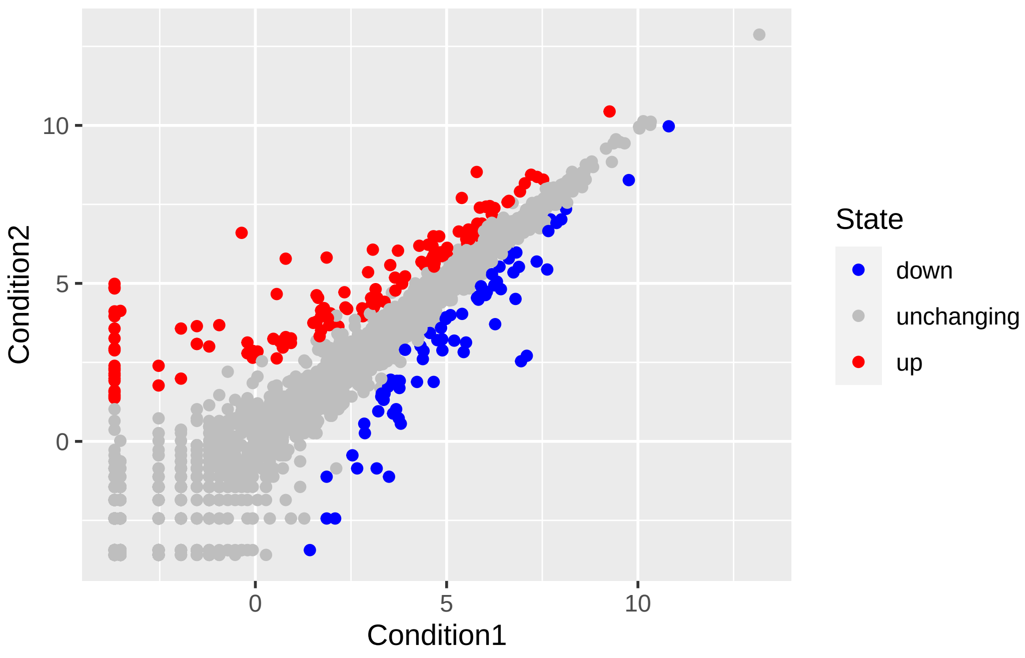
Add state as metric

```
p <- ggplot(genes) +
    aes(x=Condition1, y=Condition2, col=State) +
    geom_point()
p
```



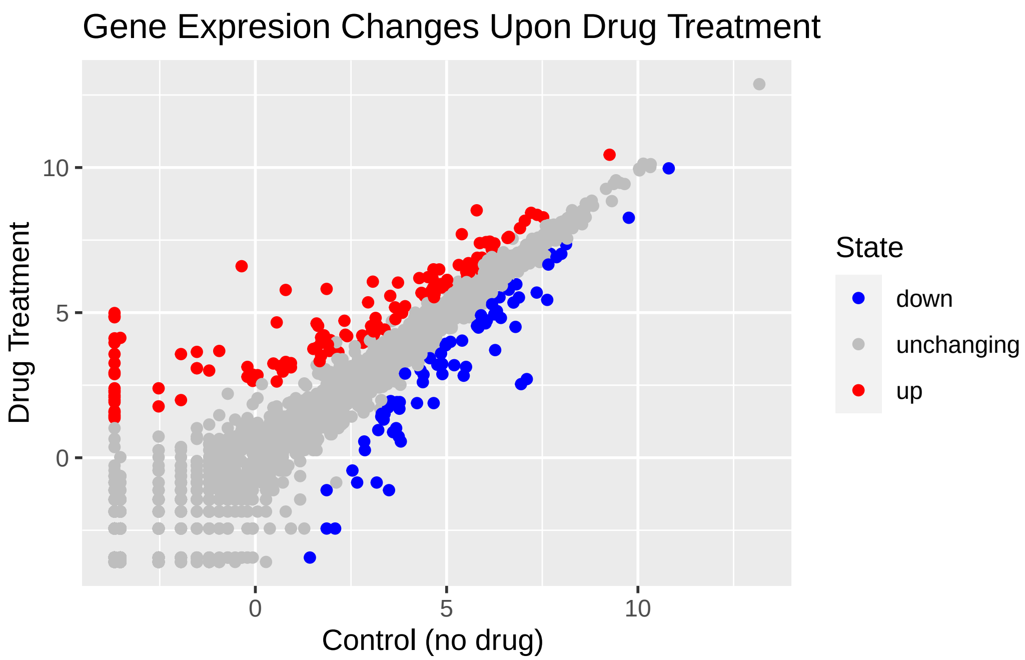
Change color scheme

```
p + scale_colour_manual( values=c("blue","gray","red") )
```



Add labels

```
p + scale_colour_manual(values=c("blue","gray","red")) +
    labs(title="Gene Expresion Changes Upon Drug Treatment",
         x="Control (no drug) ",
         y="Drug Treatment")
```



Section 7

Installation install.packages("gapminder")

```
library(gapminder)
```

```
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.

gapminder <- read.delim(url)
```

Installing dplyr install.packages("dplyr")

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag
```

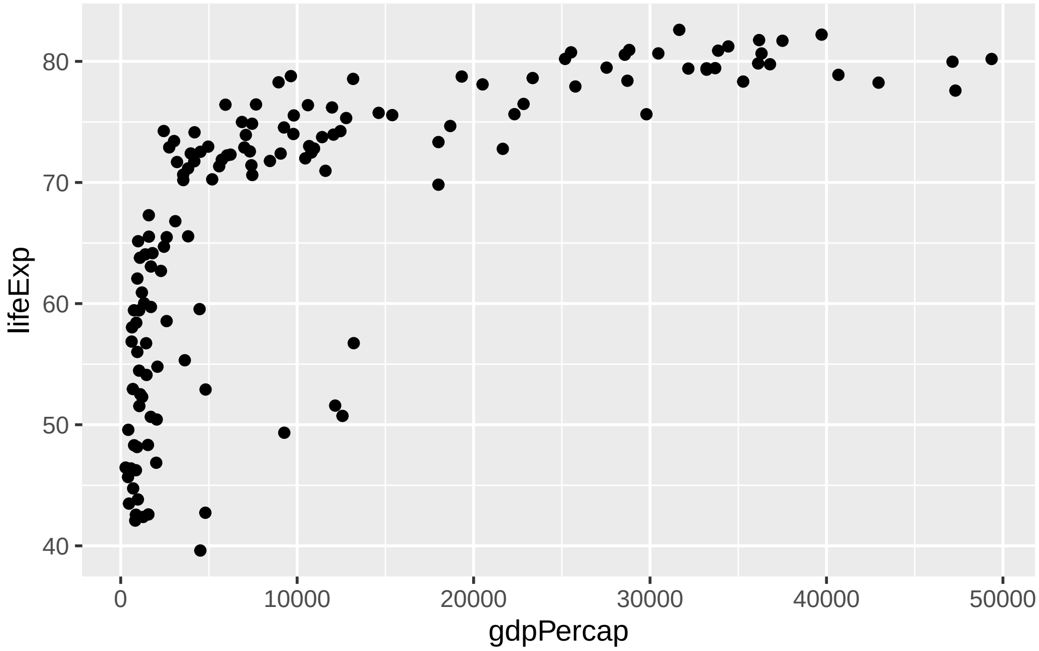
```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

Filter rows with only year 2007

```r
gapminder_2007 <- gapminder %>% filter(year==2007)
```

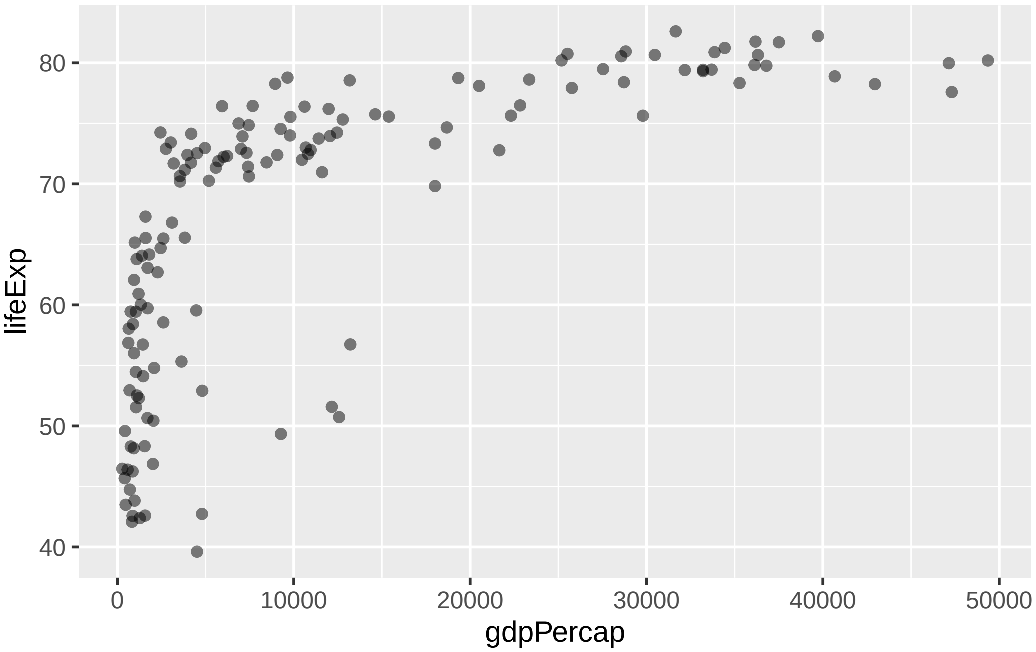
Basic scatter plot

```r
library(ggplot2)

ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point()
```



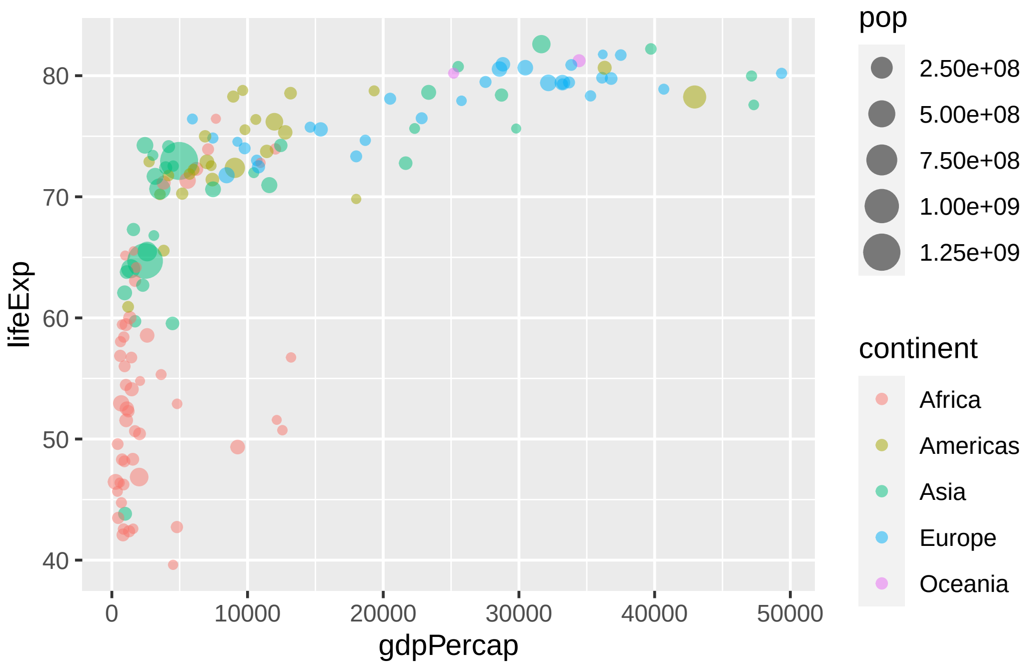
Make points transparent

```r
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point(alpha=0.5)
```



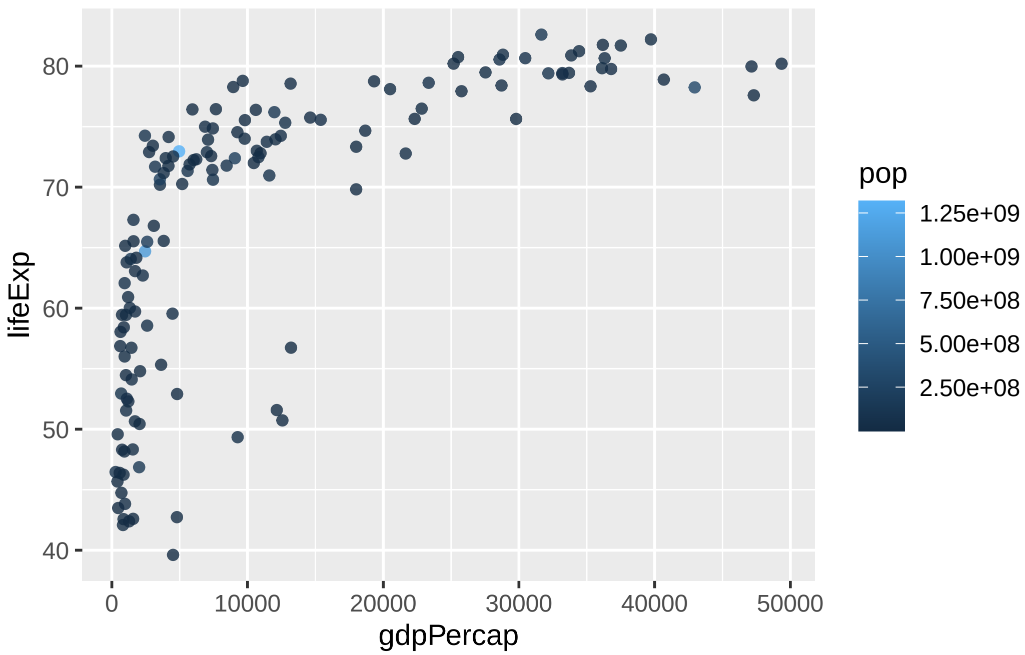
Add more variables (colors for contininent, size proportional to population)

```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5)
```



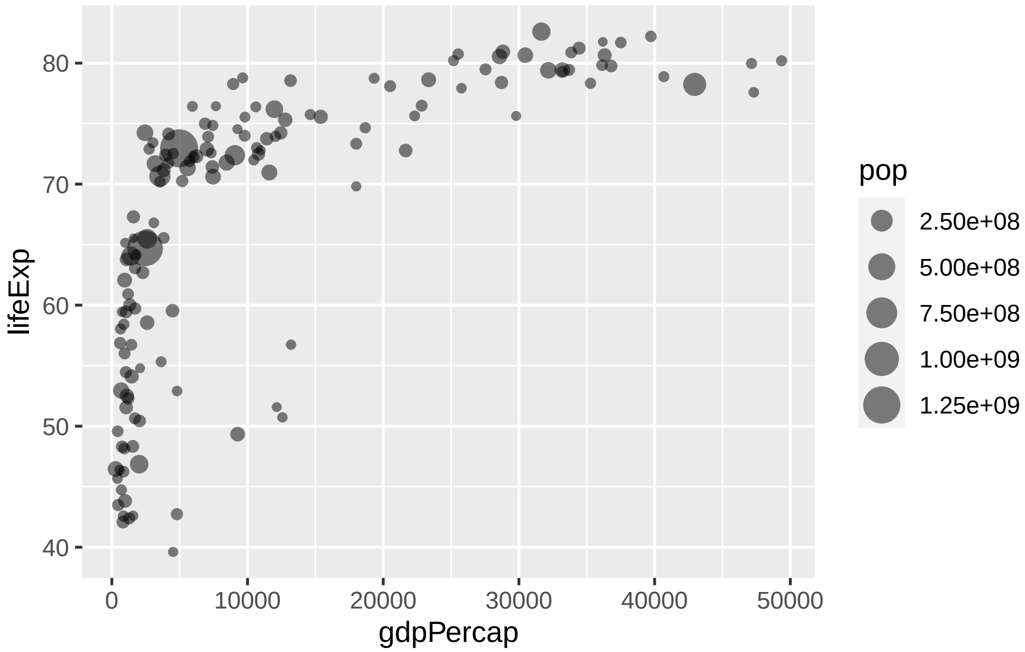
Gradient population

```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```



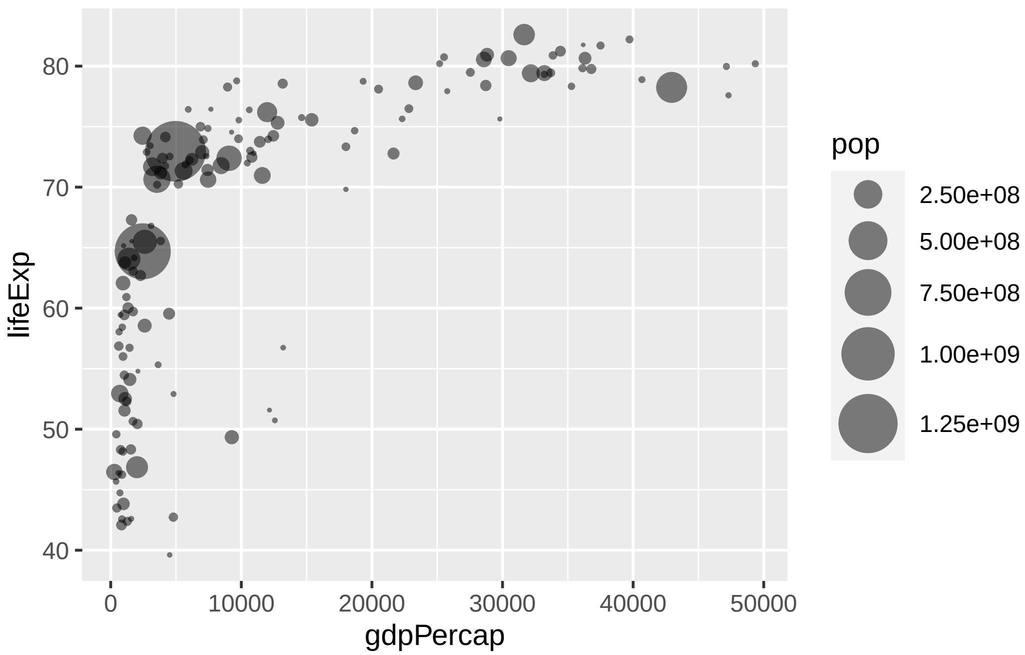
Adjust point size

```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, size = pop) +
  geom_point(alpha=0.5)
```



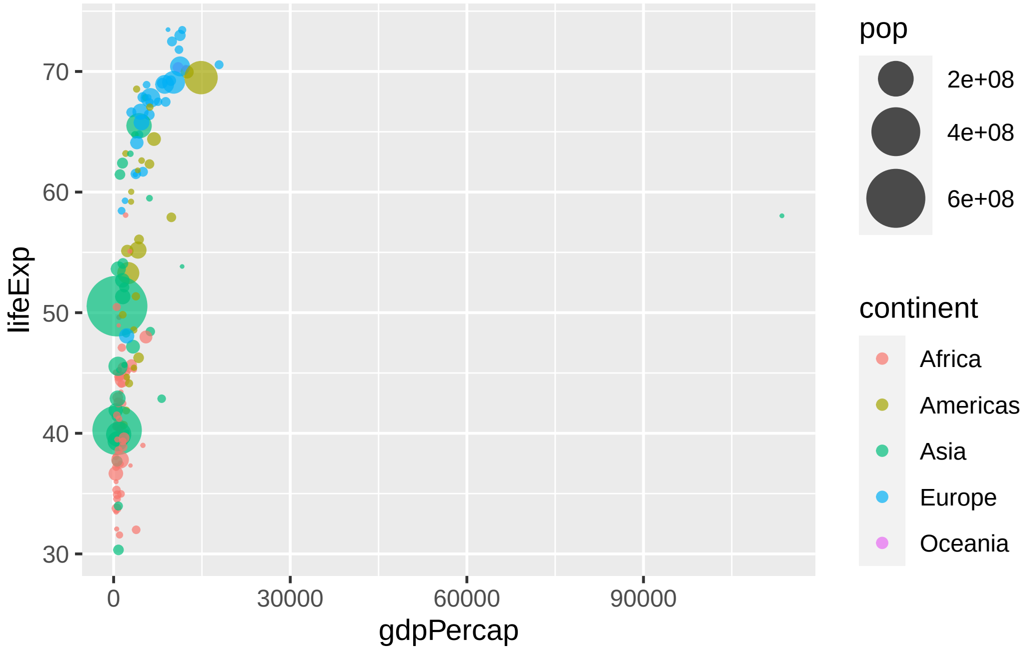
New scale

```
ggplot(gapminder_2007) +
  geom_point(aes(x = gdpPercap, y = lifeExp,
                 size = pop), alpha=0.5) +
  scale_size_area(max_size = 10)
```



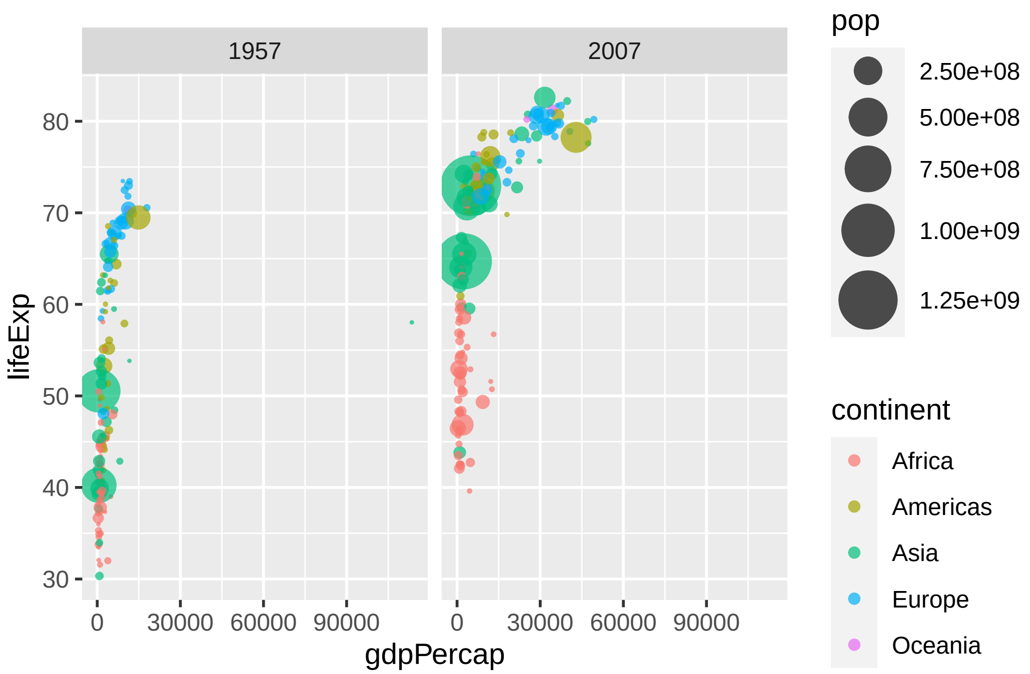
Scaled to population and colored for 1957

```
gapminder_1957 <- gapminder %>% filter(year==1957)

ggplot(gapminder_1957) +
  aes(x = gdpPercap, y = lifeExp, color=continent,
                 size = pop) +
  geom_point(alpha=0.7) +
  scale_size_area(max_size = 10)
```



Two in one code

```
gapminder_1957 <- gapminder %>% filter(year==1957 | year==2007)

ggplot(gapminder_1957) +
  geom_point(aes(x = gdpPercap, y = lifeExp, color=continent,
                 size = pop), alpha=0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)
```



Section 8

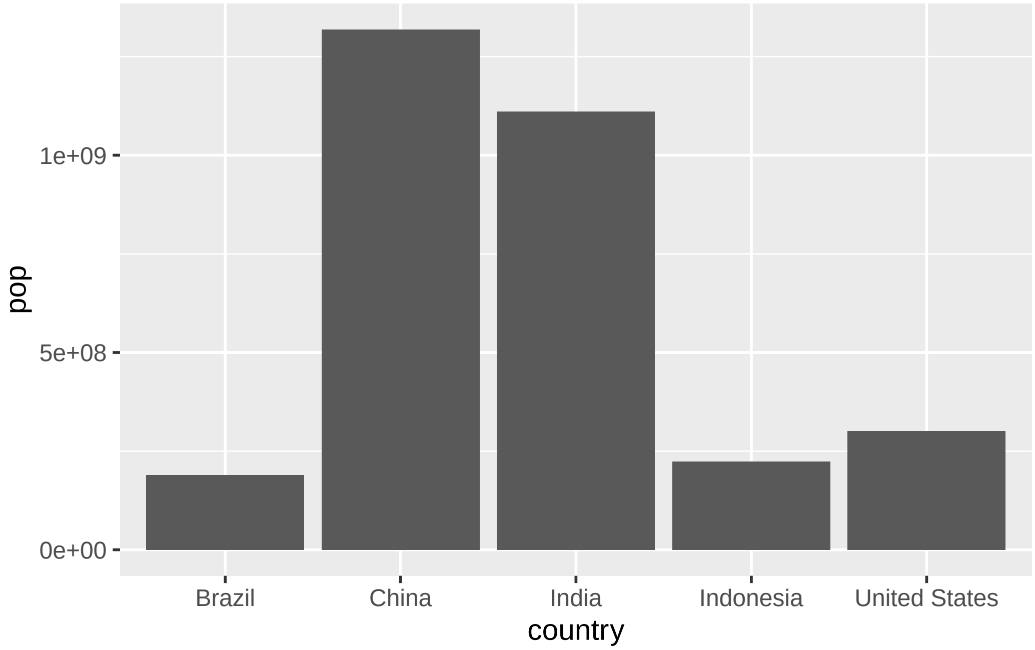
Load, filter, arrange data

```
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)

gapminder_top5
```

```
        country continent year lifeExp        pop gdpPercap
1         China      Asia 2007  72.961 1318683096  4959.115
2         India      Asia 2007  64.698 1110396331  2452.210
3 United States  Americas 2007  78.242  301139947 42951.653
4     Indonesia      Asia 2007  70.650  223547000  3540.652
5        Brazil  Americas 2007  72.390  190010647  9065.801
```

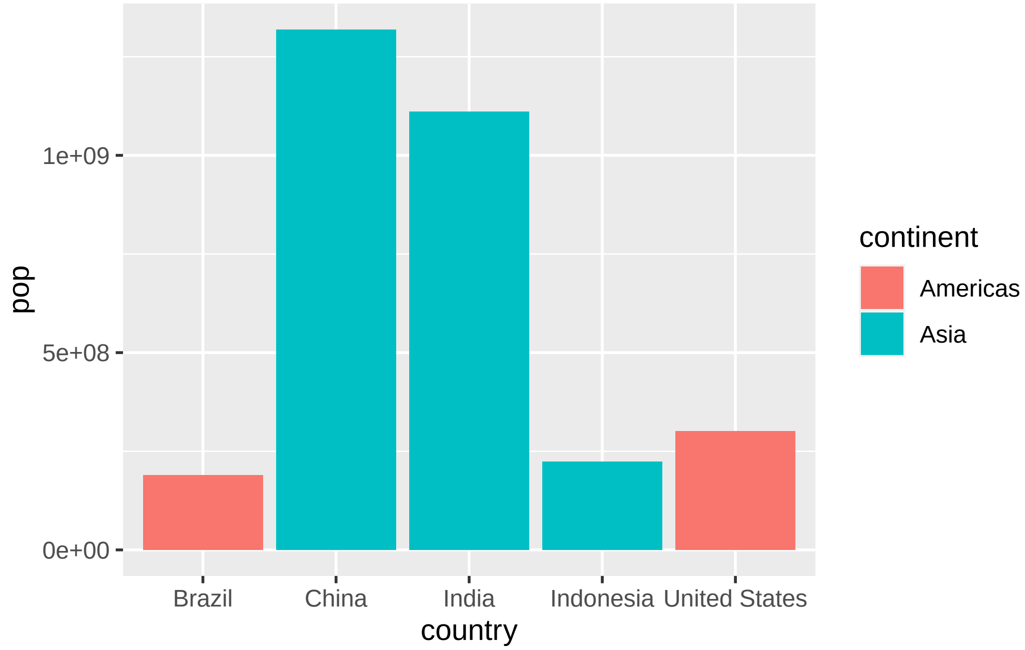
Simple bar chart

```
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = pop))
```



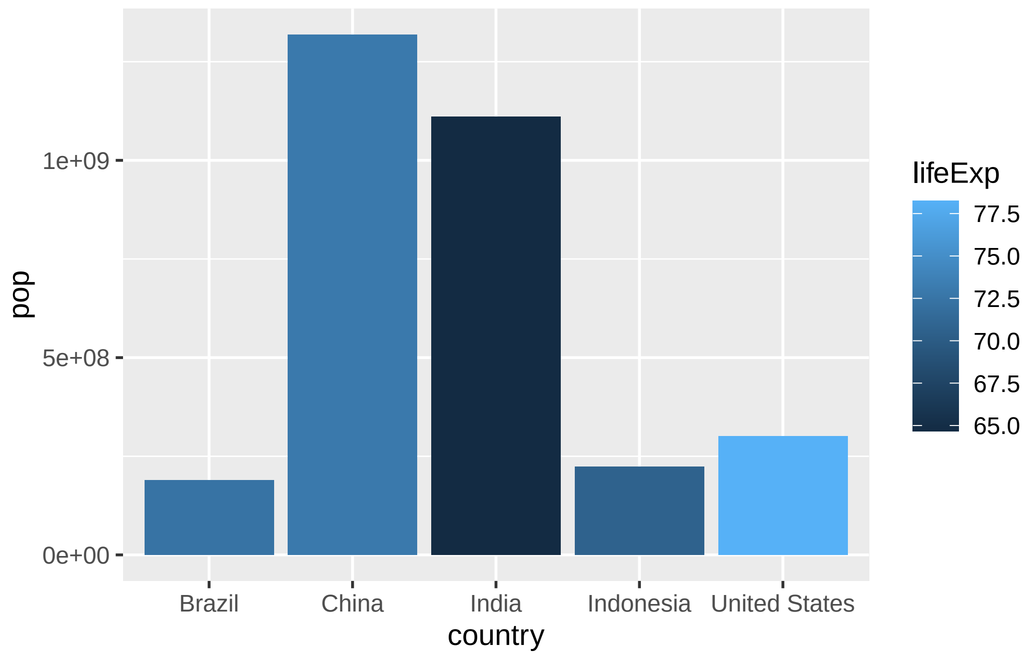
Color bars

```
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = pop, fill = continent))
```



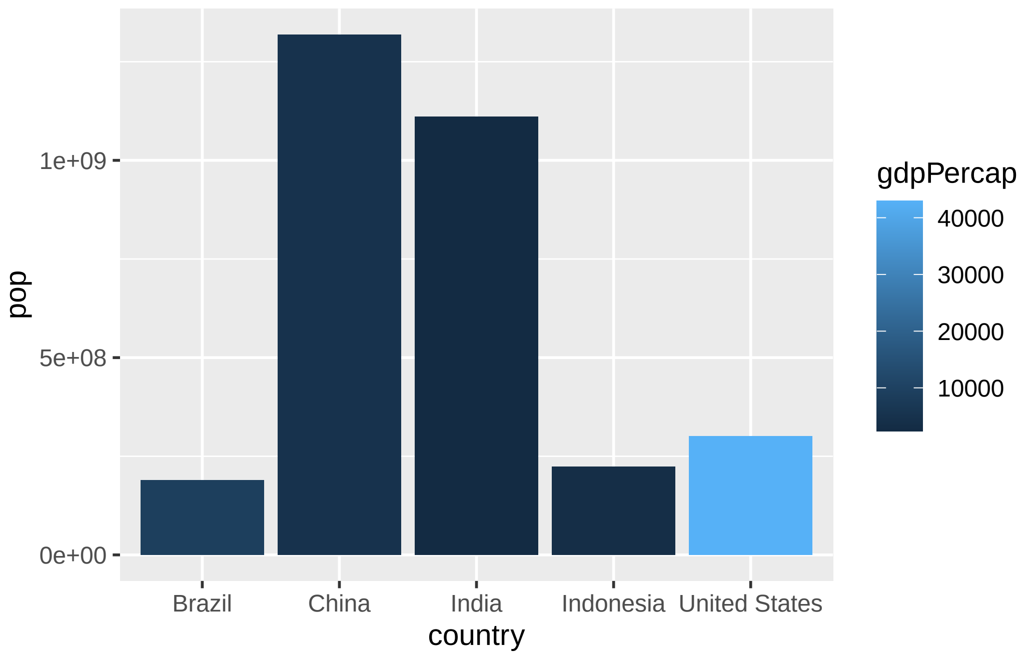
Life expectancy gradient

```
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = pop, fill = lifeExp))
```



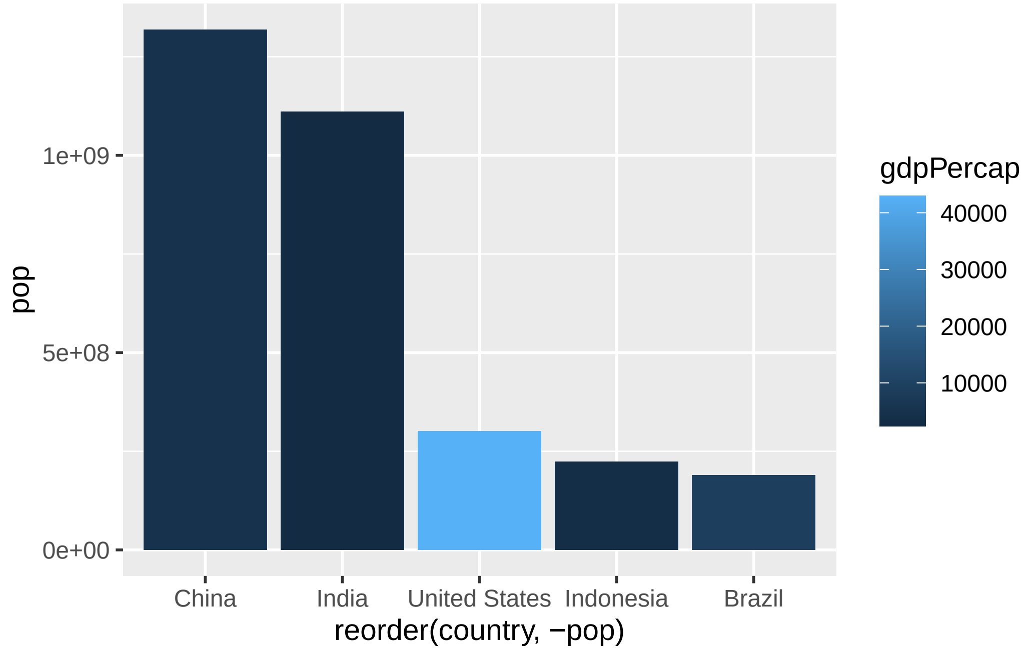
GDP gradient

```
ggplot(gapminder_top5) +
  aes(x=country, y=pop, fill=gdpPercap) +
  geom_col()
```



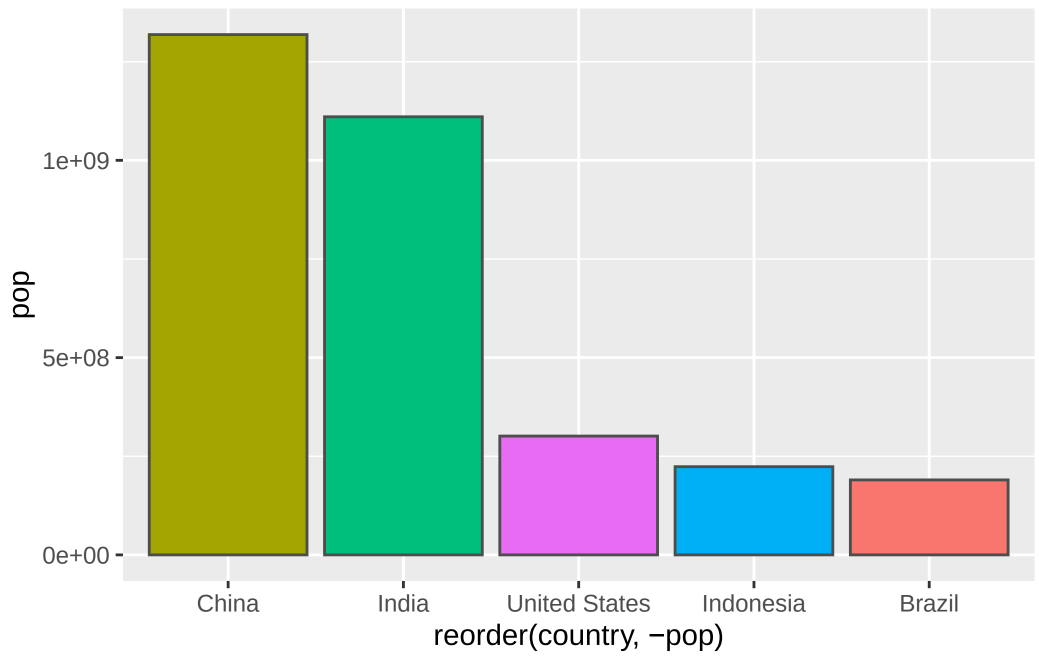
Ordered GDP gradient

```
ggplot(gapminder_top5) +
  aes(x=reorder(country, -pop), y=pop, fill=gdpPercap) +
  geom_col()
```



Fill by country

```
ggplot(gapminder_top5) +
  aes(x=reorder(country, -pop), y=pop, fill=country) +
  geom_col(col="gray30") +
  guides(fill="none")
```



Flipping bar charts

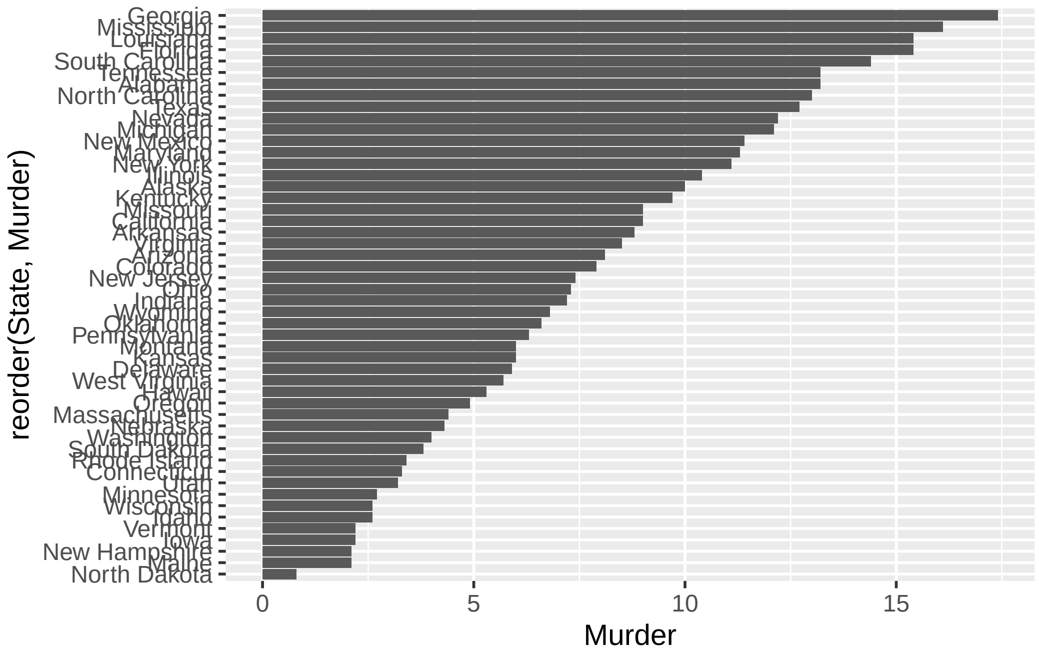
Load data

```
head(USArrests)
```

```
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
Colorado      7.9     204       78 38.7
```

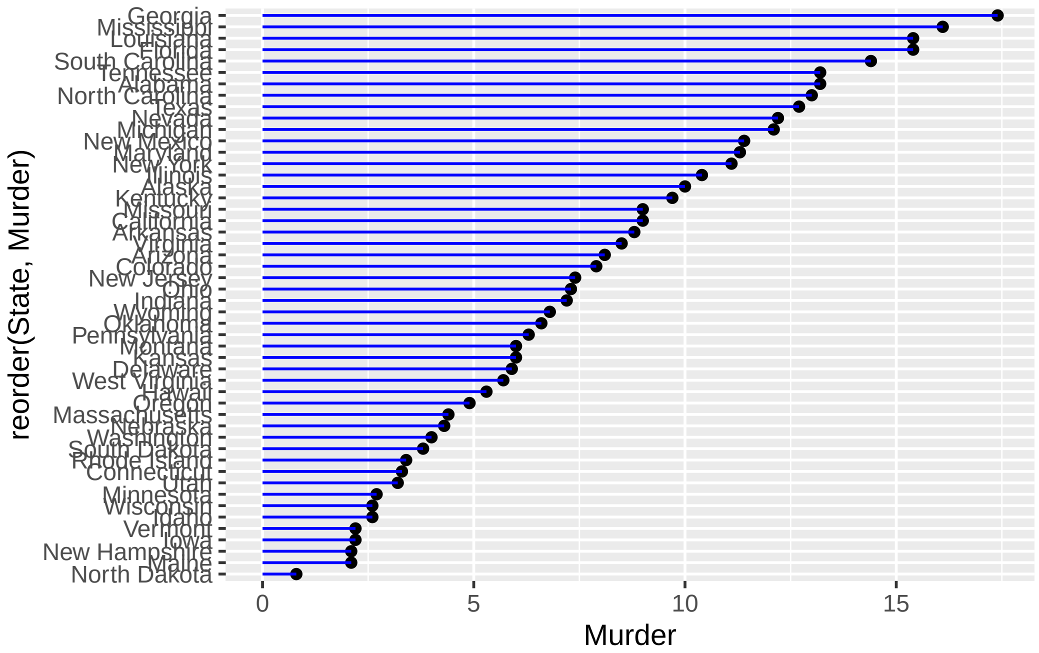
Plot by state `coord_flip` reverses X and Y axis

```
USArrests$State <- rownames(USArrests)
ggplot(USArrests) +
  aes(x=reorder(State,Murder), y=Murder) +
  geom_col() +
  coord_flip()
```



Custom scaling and aesthetics

```
ggplot(USArrests) +
  aes(x=reorder(State,Murder), y=Murder) +
  geom_point() +
  geom_segment(aes(x=State,
                   xend=State,
                   y=0,
                   yend=Murder), color="blue") +
  coord_flip()
```



Section 9

install.packages("gganimate") install.packages("gifski")

library(gapminder) library(gganimate) library(ggplot2)

ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) + geom_point(alpha = 0.7, show.legend = FALSE) + scale_colour_manual(values = country_colors) + scale_size(range = c(2, 12)) + scale_x_log10() + facet_wrap(~continent) + labs(title = 'Year: {frame_time}', x = 'GDP per capita', y = 'life expectancy') + transition_time(year) + shadow_wake(wake_length = 0.1, alpha = FALSE)

Section 10

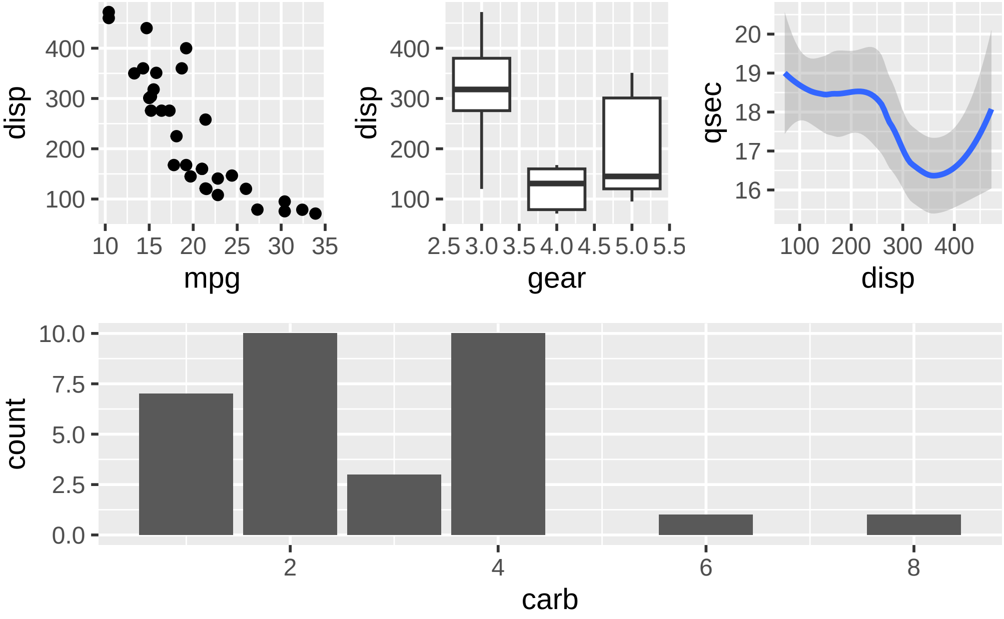
Combine many panels

install.packages("patchwork")

```
library(patchwork)

# Setup some example plots
```

```
p1 <- ggplot(mtcars) + geom_point(aes(mpg, disp))
p2 <- ggplot(mtcars) + geom_boxplot(aes(gear, disp, group = gear))
p3 <- ggplot(mtcars) + geom_smooth(aes(disp, qsec))
p4 <- ggplot(mtcars) + geom_bar(aes(carb))

# Use patchwork to combine them here:
(p1 | p2 | p3) /
      p4
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```


disp
mpg
disp
gear
qsec
disp
count
carb