Class 05: Data Visualization

AUTHOR
Alex Cagle (PID: A15661779)

```
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.2.3
```

```
# install.packages("dplyr")  ## un-comment to install if needed
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
```

```
# installing/loading the package:
# if(!require(installr)) {
#  install.packages("installr");
#  require(installr)
# } #load / install+load installr

# using the package:
#updateR()
```

Base R Graphics vs. ggplot2

There are many graphics systems available in R, including so-called "base" R graphics and the very popular **ggplot2** package.

To compare these, let's play with the inbuilt `cars` dataset.

```
head(cars)
```

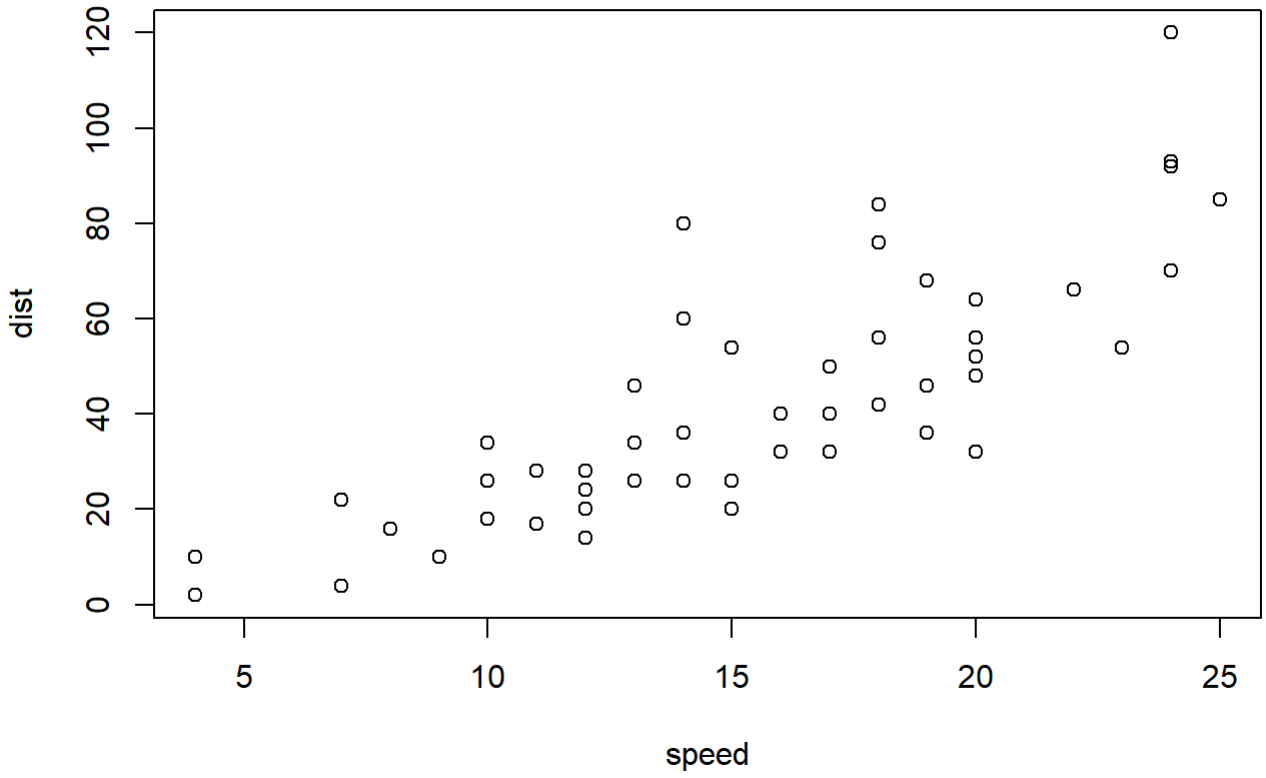
```
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
```

```
5     8   16
6     9   10
```

```
# head(cars, 10)
```

To use "base" R, I can simply call the `plot()` function:

```
plot(cars)
```

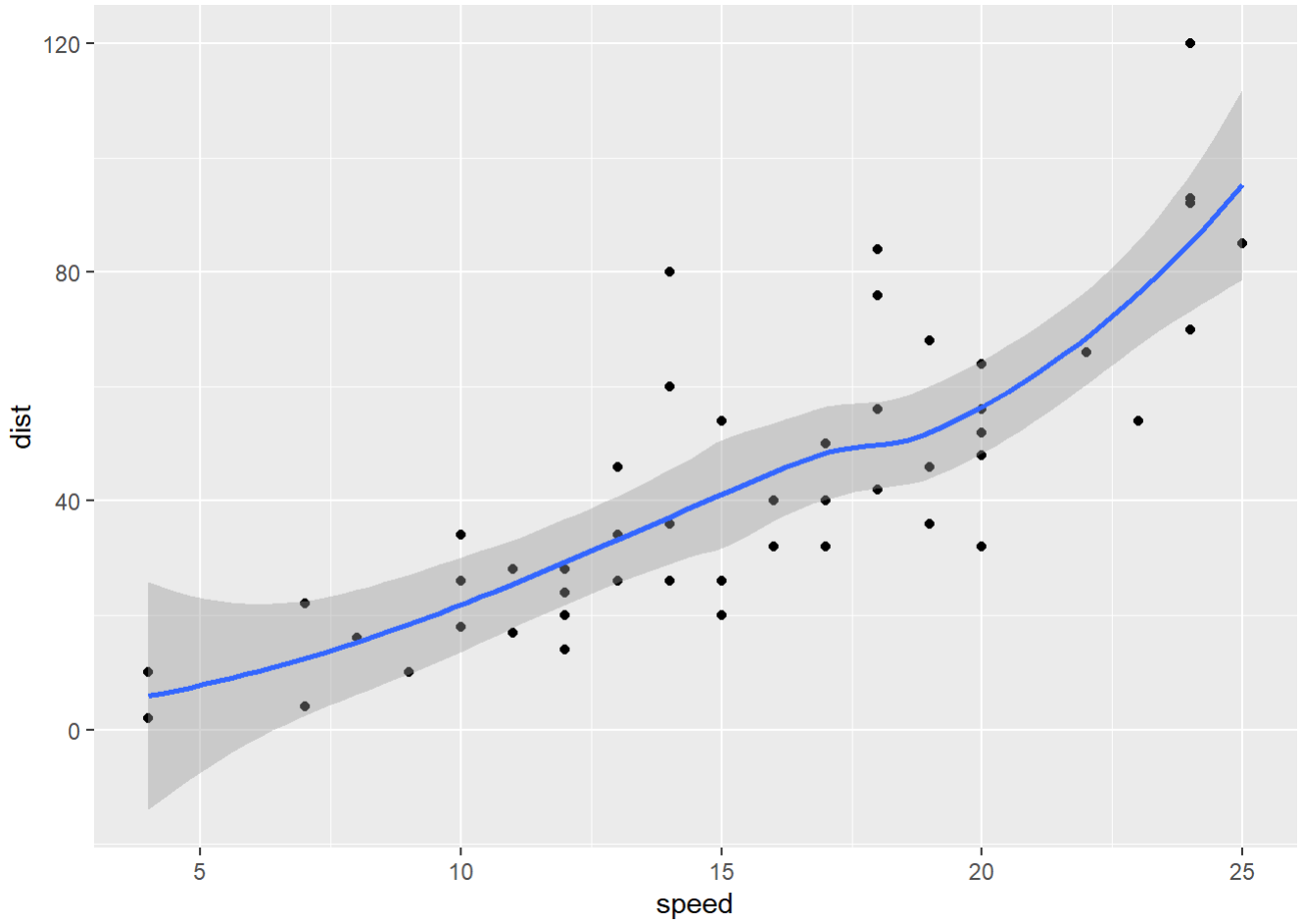


To use `ggplot2` package, I first need to install it with the function `install.package("ggplot2")`.

I will run this in my R console (i.e. the R brain) as I do not want to re-install it every time I render my report.

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

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



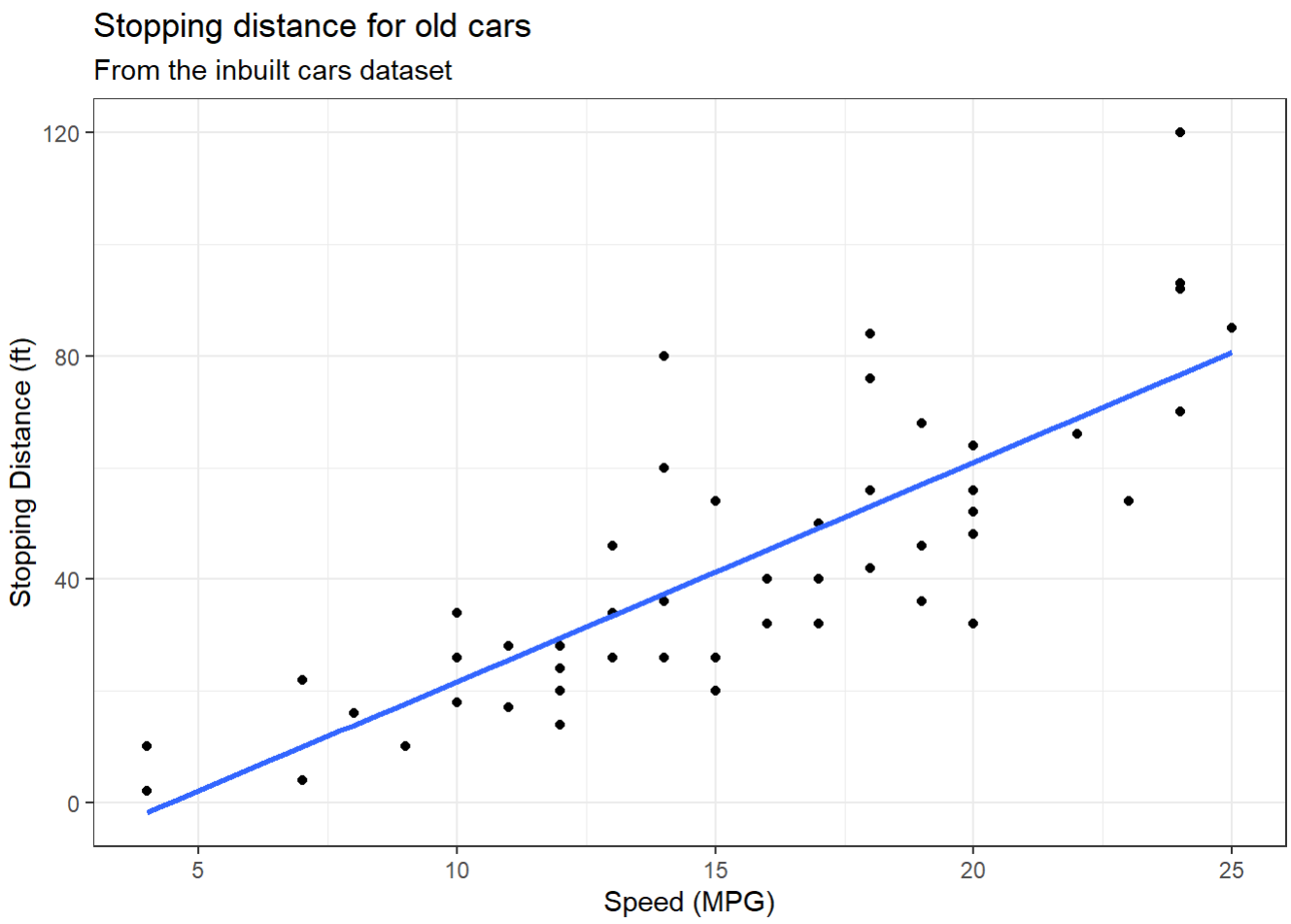
To make a figure with ggplot, I always need at least 3 things:

- data (i.e. what I want to plot)
- aesthetics (i.e. how the plot looks, aesthetic mapping of the data to the plot)
- the geoms (i.e. how I want to plot the data with different geometries)

GGplot is much more verbose than base R plots for standard plots, but it has a consistent layer system that I can use to make just about any plot.

```
ggplot(data=cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE) +
  labs(
       title="Stopping distance for old cars",
       x = "Speed (MPG)",
       y = "Stopping Distance (ft)",
       subtitle = "From the inbuilt cars dataset"
       ) +
  theme_bw()
```

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



A More Complicated Plot

```
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
```

> Q: How many genes are in this dataset?

```
nrow(genes)
```

```
[1] 5196
```

> Q: How can we summarize that last column - the "State" column?

```
table(genes$State)
```

```
      down unchanging         up
        72       4997        127
```

```
colnames(genes)
```

```
[1] "Gene"       "Condition1" "Condition2" "State"
```

```
ncol(genes)
```

```
[1] 4
```

> **Q.** Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset?

```
# Ratio of up-regulated genes to total genes
127 / 5196
```

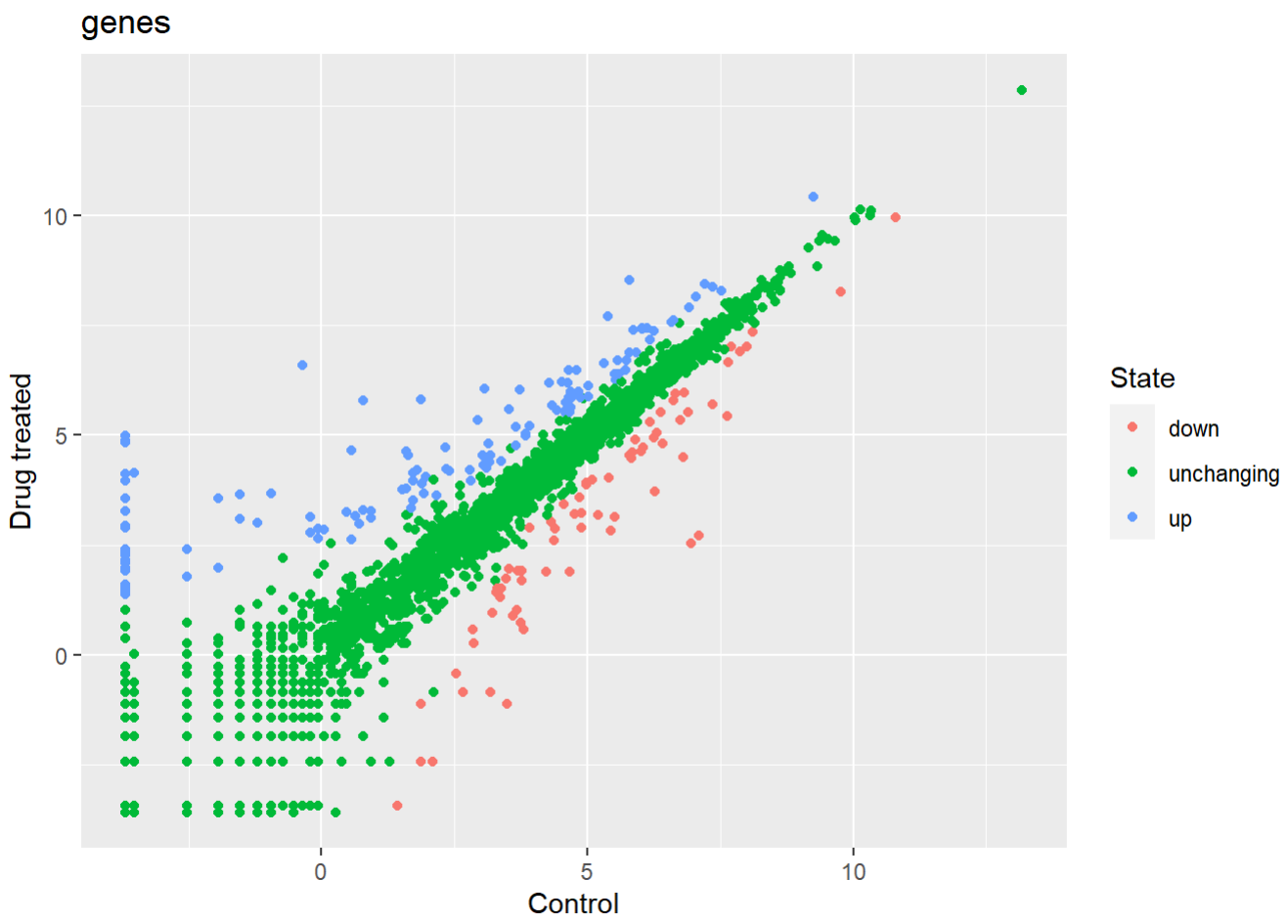
```
[1] 0.02444188
```

```
answer <- 0.024
```

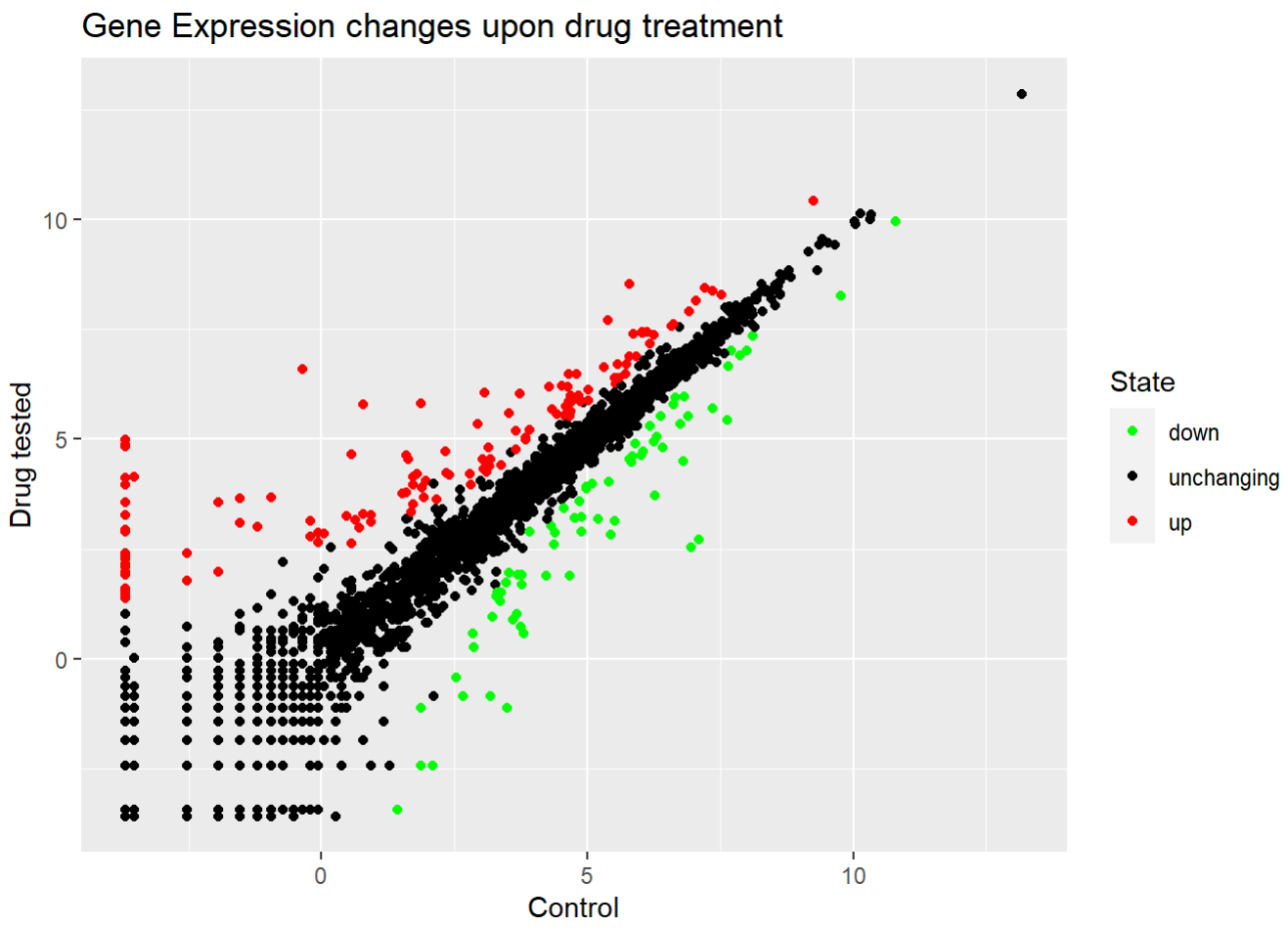
```
plt <- ggplot(data=genes) +
  aes(x=Condition1, y=Condition2, col=State) +
  geom_point() +
  xlab("Control") +
  ylab("Drug treated")
```

I can now just call `plt` when I want to plot or add to it.

```
plt + ggtitle("genes")
```




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

p + labs(title = "Gene Expression changes upon drug treatment",
         x = "Control",
         y = "Drug tested") +
  scale_color_manual(values = c("green", "black", "red"))
```



Going further

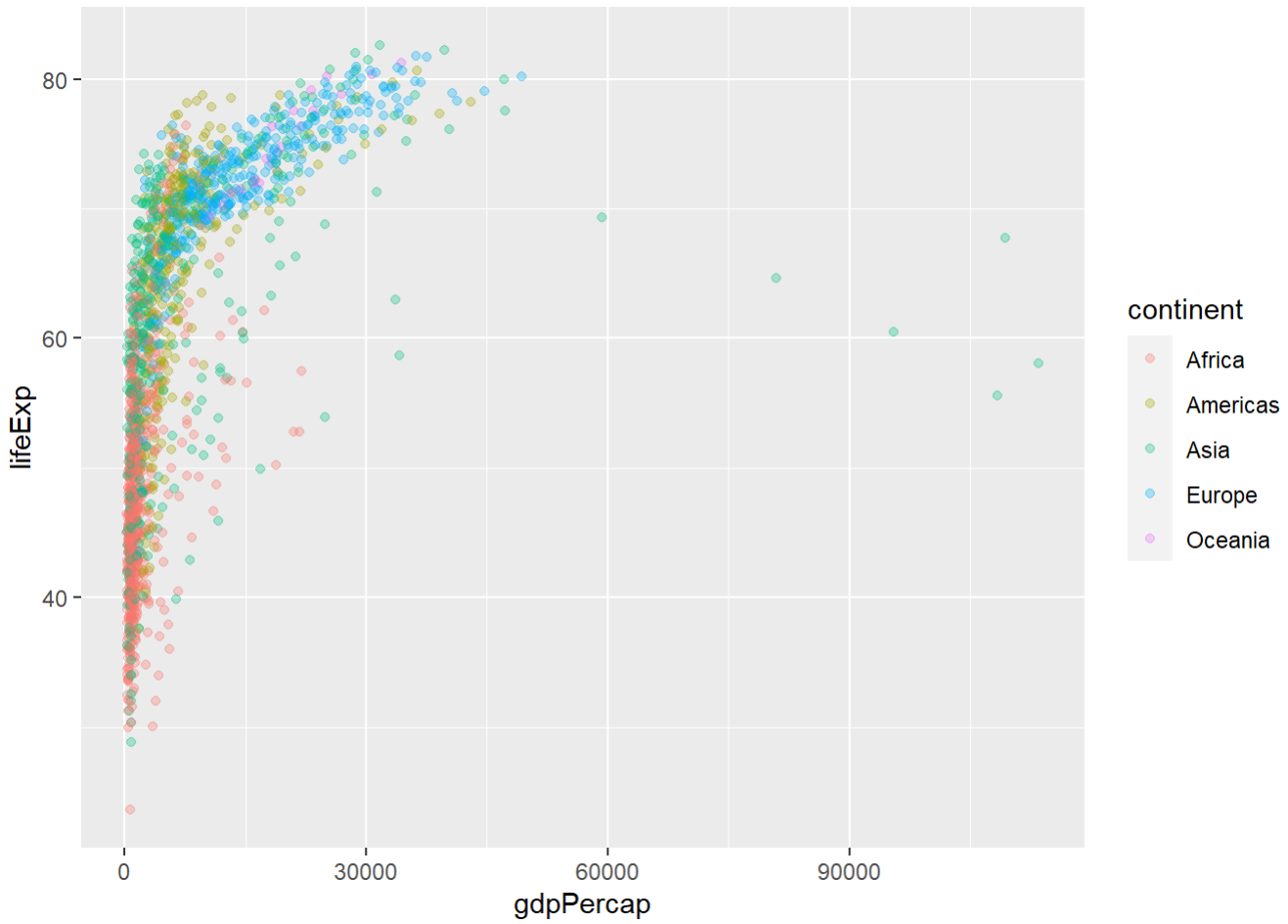
Here I read a slightly larger dataset

```
# File location online
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.tsv"

gapminder <- read.delim(url)

head(gapminder)
```

```
      country continent year lifeExp      pop gdpPercap
1 Afghanistan      Asia 1952  28.801  8425333  779.4453
2 Afghanistan      Asia 1957  30.332  9240934  820.8530
3 Afghanistan      Asia 1962  31.997 10267083  853.1007
4 Afghanistan      Asia 1967  34.020 11537966  836.1971
5 Afghanistan      Asia 1972  36.088 13079460  739.9811
6 Afghanistan      Asia 1977  38.438 14880372  786.1134
```

```
ggplot(data = gapminder) +
  aes(x = gdpPercap, y = lifeExp, col = continent) +
```

```
  geom_point(alpha = 0.3)
```



A very useful layer to add sometimes is for "faceting."

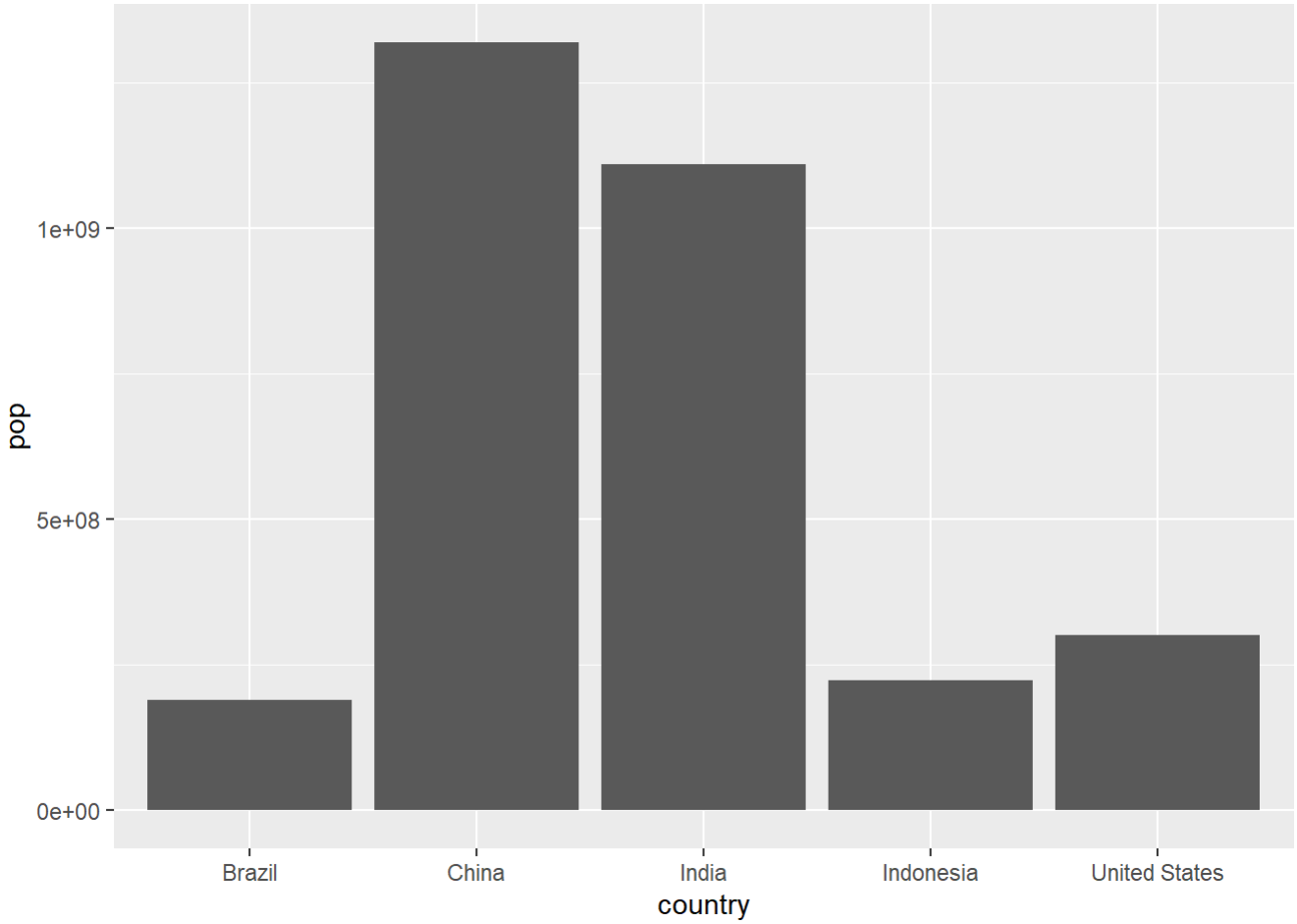
```
{ggplot(data = gapminder) +}   aes(x = gdpPercap, y = lifeExp, col = continent, size = pop) +
geom_point(alpha = 0.3) +   facet_wrap(~continent)
```

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

```
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
```

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



Lab 5 Questions

> **Q1.** For which phases is data visualization important in our scientific workflows?

```
ans1 <- "All of the above"
```

> **Q2.** True or False? The ggplot2 package comes already installed with R?

```
ans2 <- FALSE
```

> **Q3.** Which plot types are typically NOT used to compare distributions of numeric variables?

```
ans3 <- "Network graphs"
```

> **Q4.** Which statement about data visualization with ggplot2 is incorrect?

```
ans4 <- "It is incorrect to say that ggplot2 is the only way to create plots in R, since we can a
```

> **Q5.** Which geometric layer should be used to create scatter plots in ggplot2?

```
ans5 <- geom_point()
```

> **Q6.** Use the `nrow()` function to find out how many genes are in this dataset. What is your answer?

```
ans6 <- 5196
```

> **Q7.** Use the `colnames()` function and the `ncol()` function on the `genes` data frame to find out what the column names are (we will need these later) and how many columns there are. How many columns did you find?

```
ans7 <- 4
```

> **Q8.** Use the `table()` function on the `State` column of this data.frame to find out how many 'up' regulated genes there are. What is your answer?

```
ans8 <- 127
```

> **Q9.** Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset?

```
ans9 <- 0.024
```