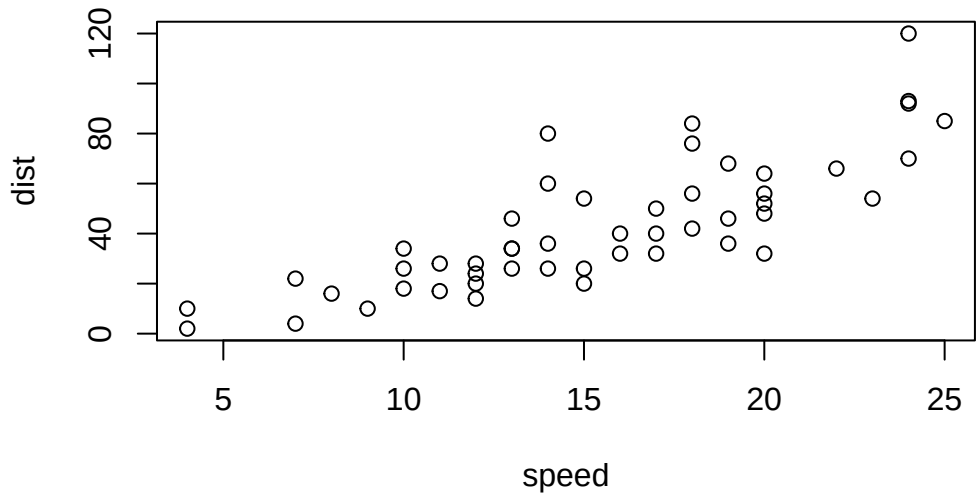
Class 5: Data Viz with ggplot

Woocheol (PID: A16998418)

Plotting in R

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org. R has lot's of ways to make plots and figures. This includes so-called **base** graphics and packages like **ggplot2**

```
plot(cars)
```



This is a **base** R plot of the in-bult `cars` dataset that has only two colums.

```
head(cars)
```

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

Q.How would we plot this wee dataset with **ggplot2**?

All ggplot figures have at least 3 layers:

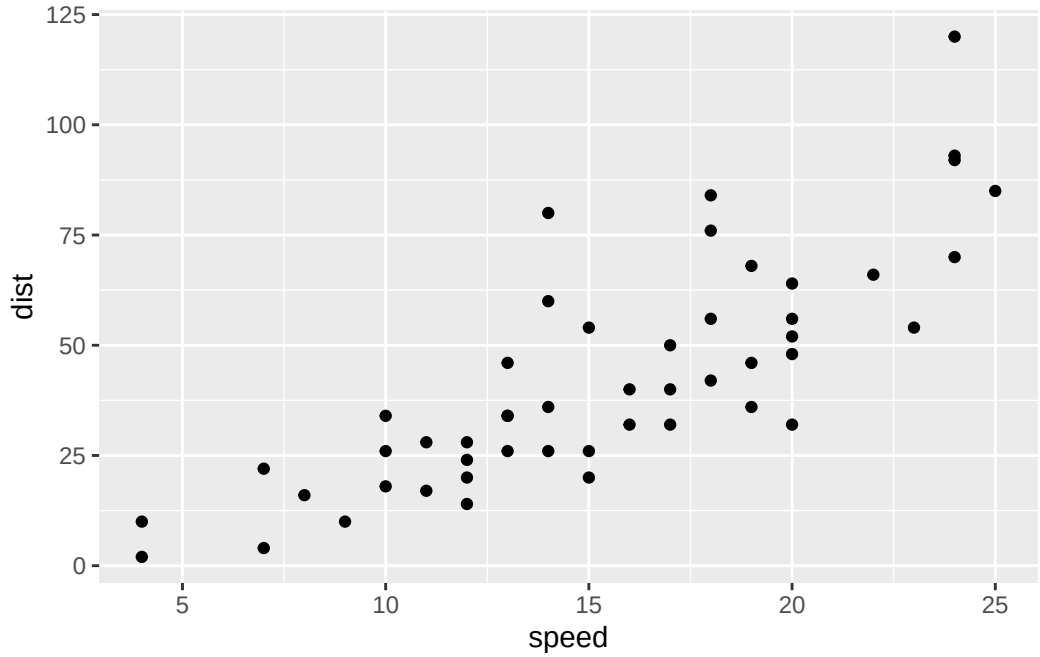
- **data**
- **aes** (how the data map to the plot)
- **geoms** (how we draw the plot, lines, points, etc)

Before I use any new package I need to download and install it with the `install.packages()` command.

I never use `install.packages()` within my quarto document otherwise I will install the package over and over and over again - which is silly!

Once a package is installed I can load it up with `library()` function.

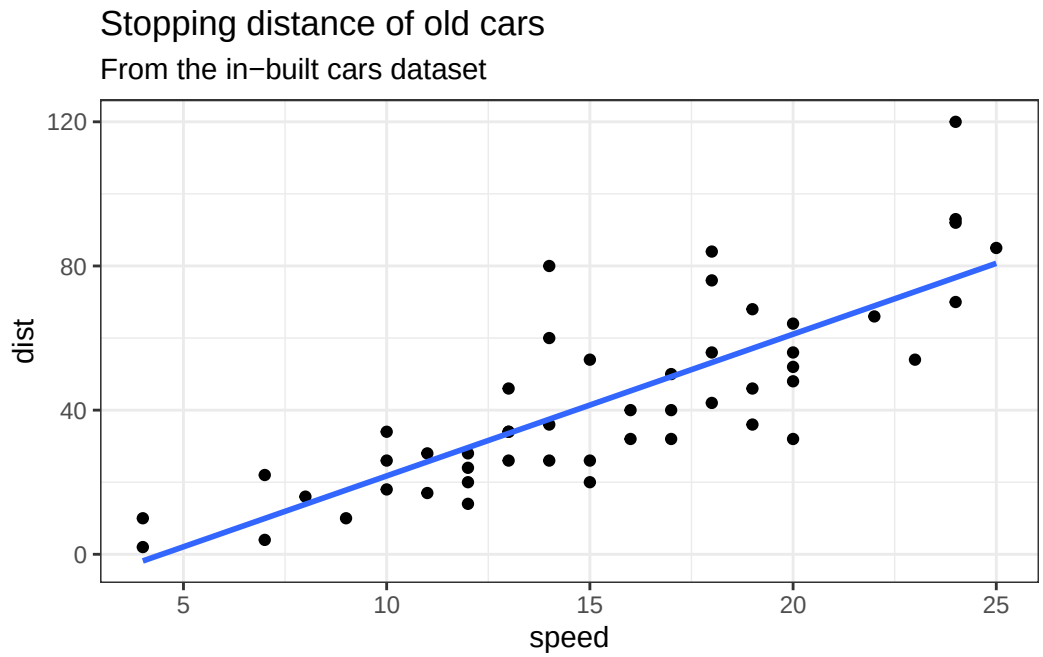
```
# install.packages("ggplot2")
library(ggplot2)
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



Key-point: For simple plots (like the one above) ggplot is more verbose (we need to do more typing) but as plots get more complicated ggplot starts to be more clear and simple than base R plot()

```
ggplot(cars) +
  aes(speed, dist) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE) +
  labs(title="Stopping distance of old cars",
       subtitle = "From the in-built cars dataset") +
  theme_bw()
```

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



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

All of the above

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

FALSE

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

Network graphs

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

ggplot2 is the only way to create plots in R

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

geom_point()

```
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
```

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

```
nrow(genes)
```

```
[1] 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?

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 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?

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

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

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

```
round( table(genes$State)/nrow(genes) * 100, 2)
```

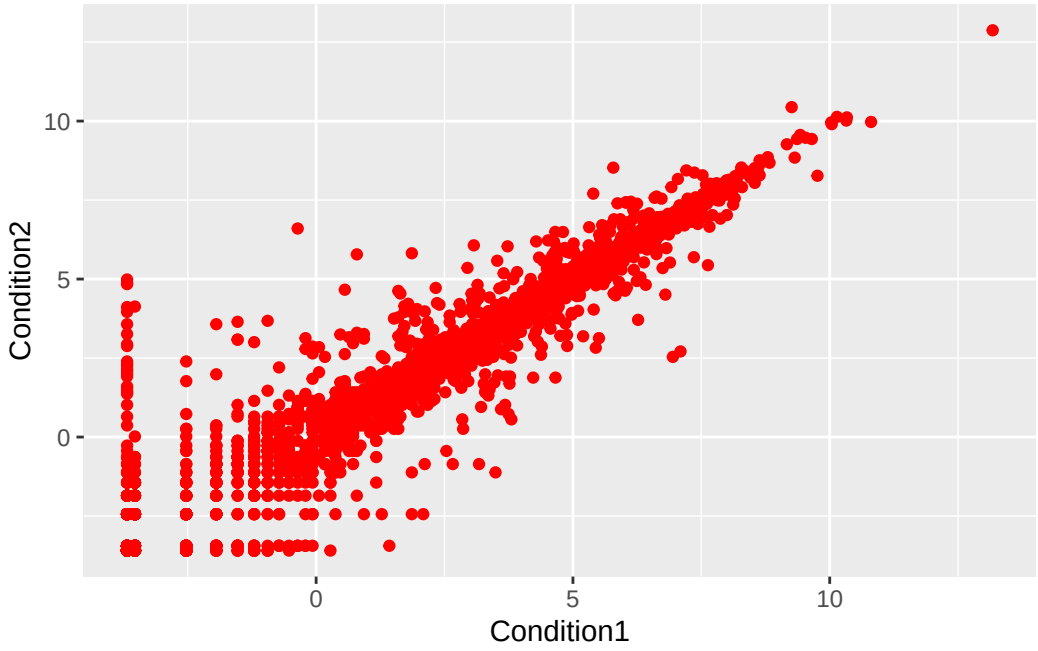
```
      down unchanging         up
      1.39      96.17       2.44
```

The key functions here where:

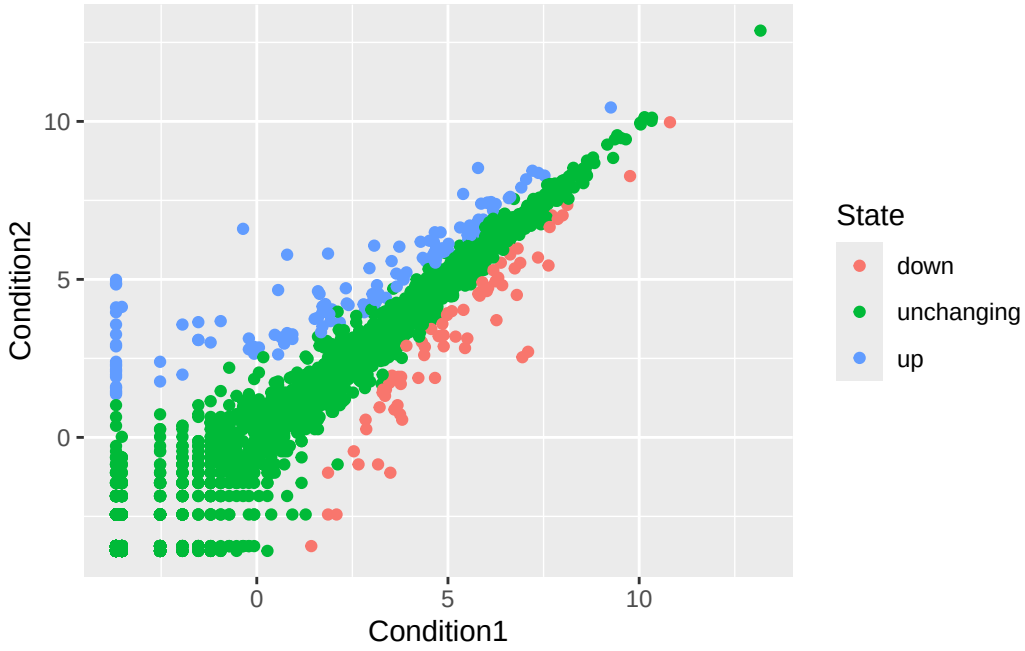
`nrow()` and `ncol()`

A first plot:

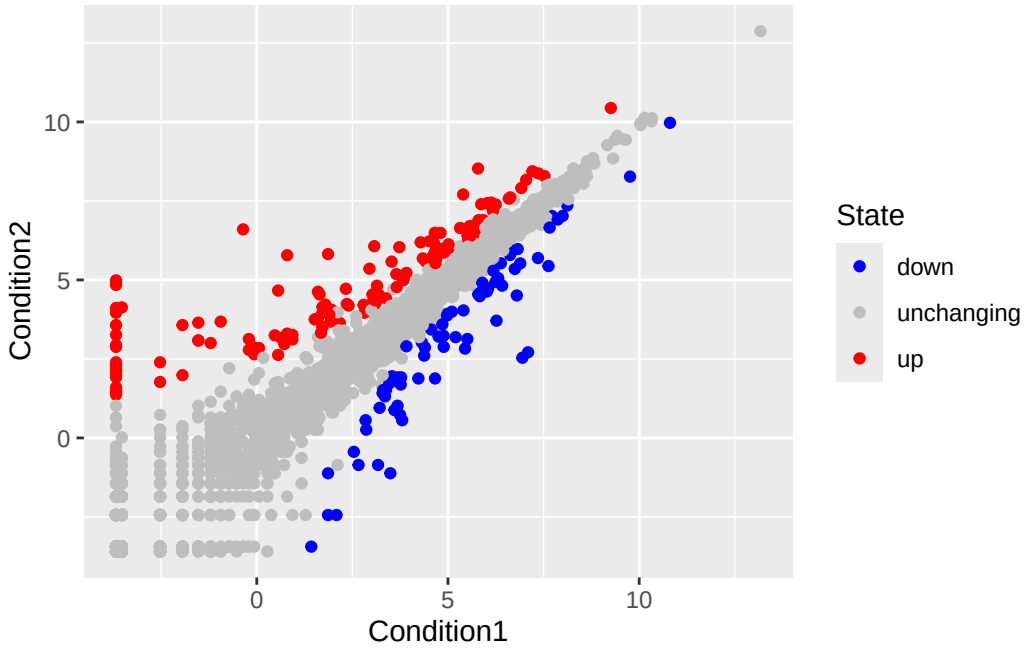
```
ggplot(genes) +
  aes(Condition1, Condition2) +
  geom_point(col='red')
```



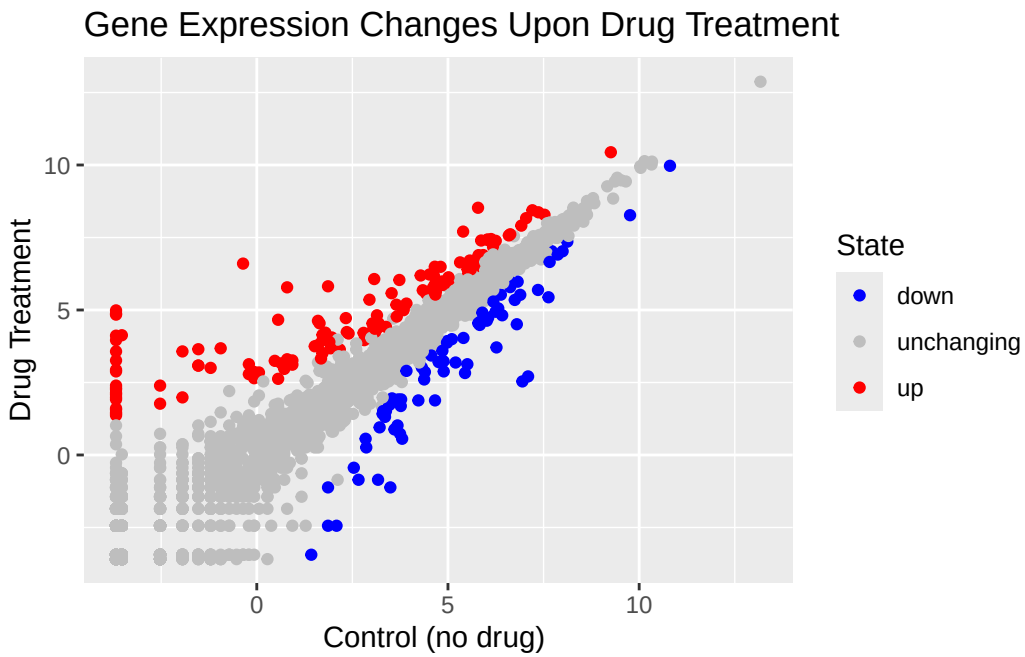

```
p <- ggplot(genes) +
    aes(x=Condition1, y=Condition2, col=State) +
    geom_point()
p
```




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



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



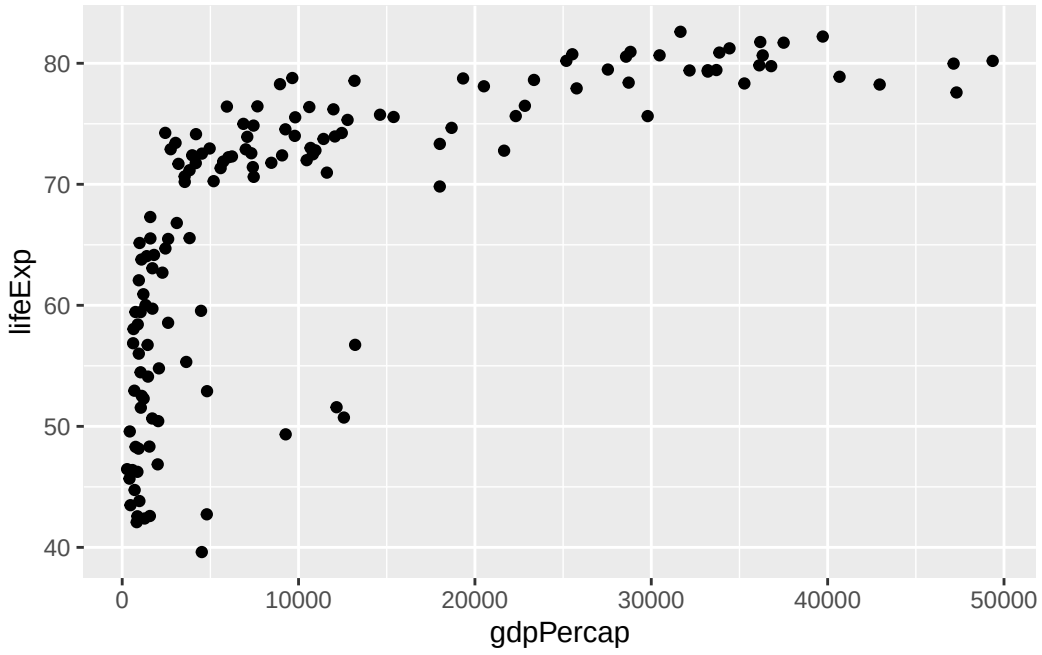

```
# install.packages("dplyr")  ## un-comment to install if needed
library(gapminder)
library(dplyr)
```

```
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
```

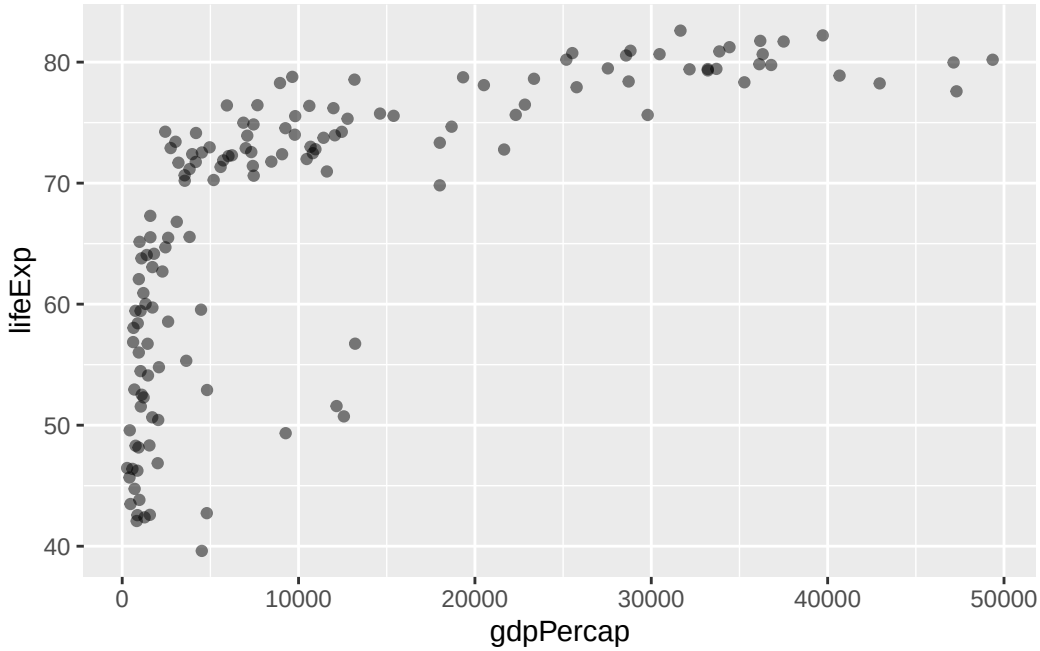
```
gapminder_2007 <- gapminder %>% filter(year==2007)
```

Q. Complete the code below to produce a first basic scater plot of this gapminder_2007 dataset:

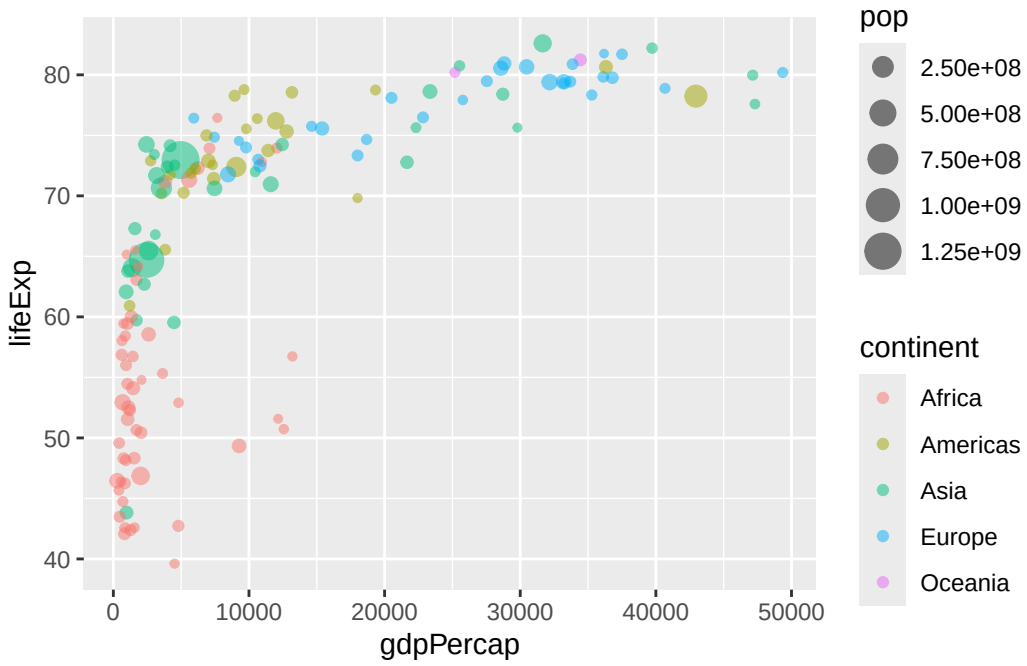
```
ggplot(gapminder_2007) +
  aes(x= gdpPercap, y= lifeExp) +
  geom_point()
```



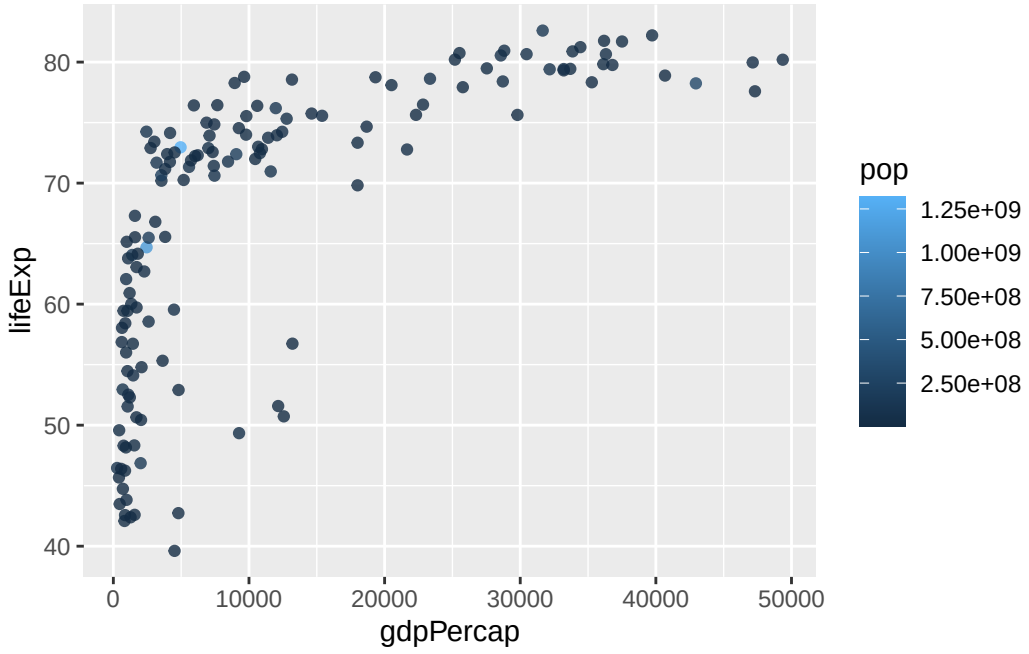

```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point(alpha=0.5)
```



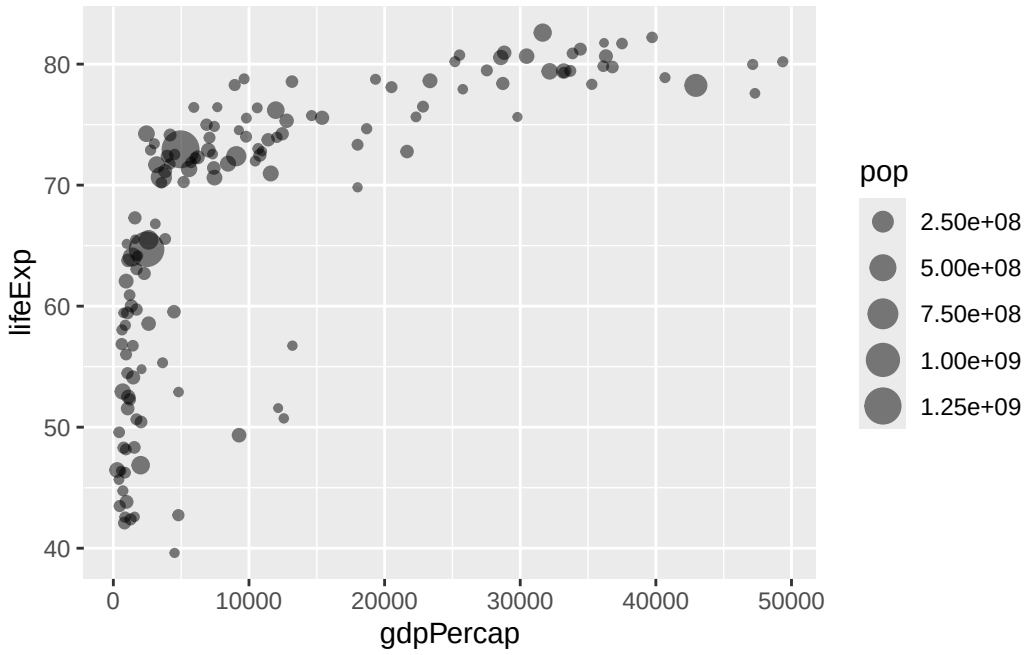

```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5)
```



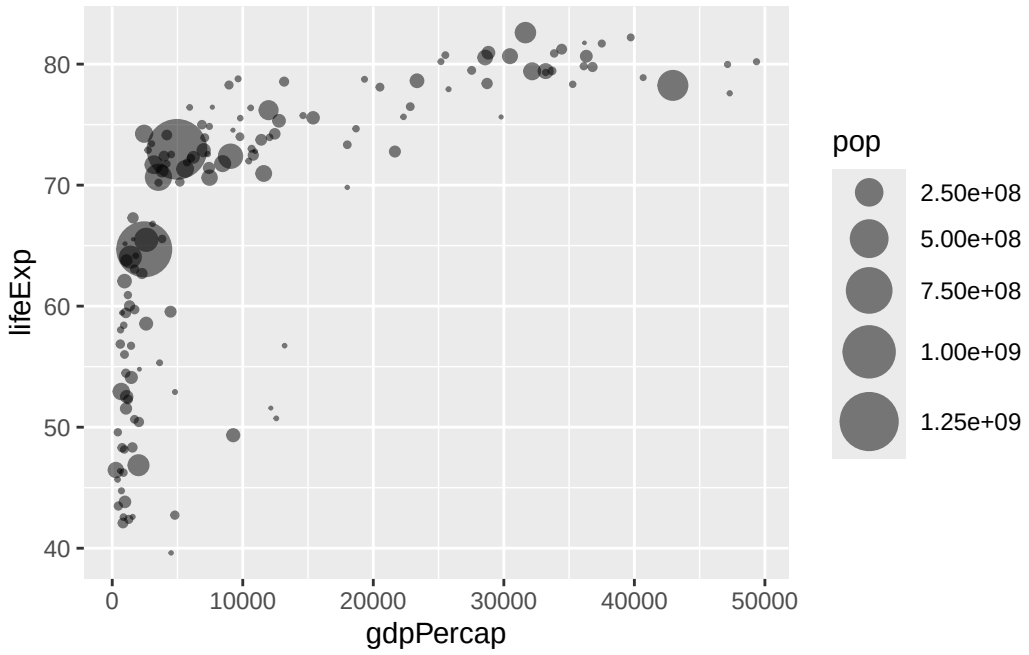
```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```



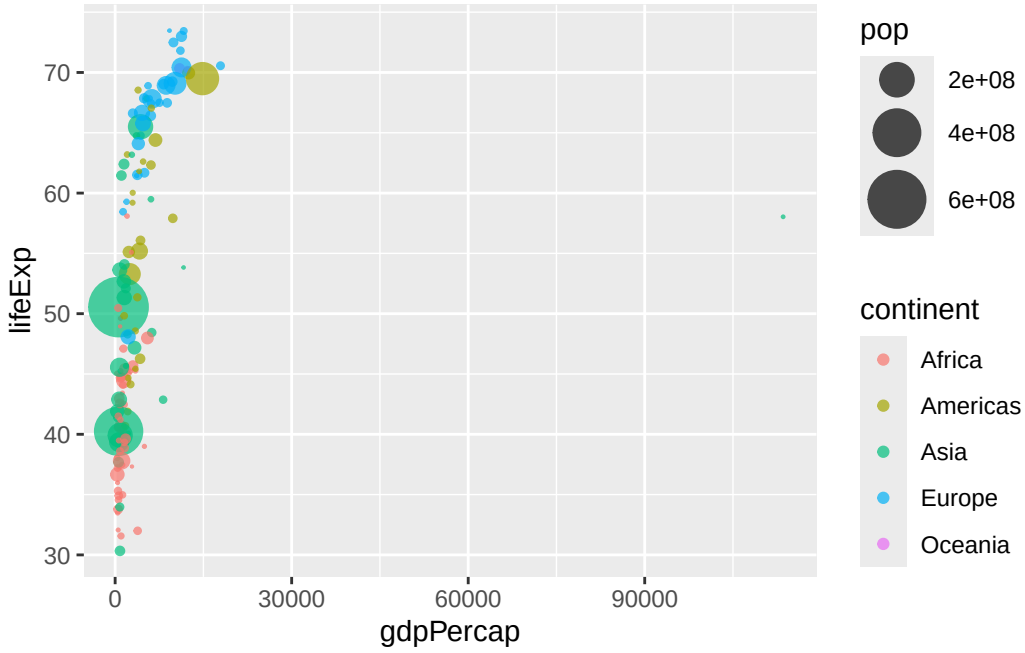

```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, size = pop) +
  geom_point(alpha=0.5)
```




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



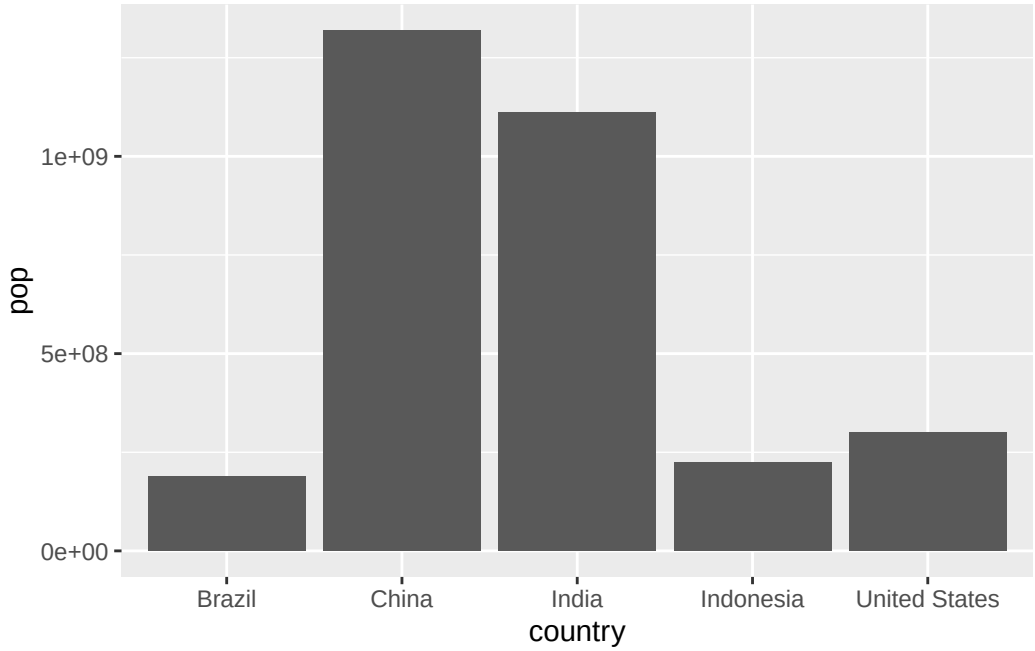

```
gapminder_1957 <- gapminder %>% filter(year==1957)

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




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

gapminder_top5
```

```
# A tibble: 5 x 6
  country       continent  year lifeExp        pop gdpPercap
  <fct>         <fct>     <int>   <dbl>      <int>     <dbl>
1 China         Asia       2007    73.0 1318683096     4959.
2 India         Asia       2007    64.7 1110396331     2452.
3 United States Americas   2007    78.2  301139947    42952.
4 Indonesia     Asia       2007    70.6  223547000     3541.
5 Brazil        Americas   2007    72.4  190010647     9066.
```

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



What was the population of Ireland in the last year we have data for?

```
filter(gapminder, country=="Ireland",
       year==2007)
```

```
# A tibble: 1 x 6
  country continent  year lifeExp     pop gdpPercap
  <fct>   <fct>     <int>   <dbl>   <int>     <dbl>
1 Ireland Europe     2007    78.9 4109086    40676.
```

Q. What countries in data set had pop smaller than Ireland in 2007?

- First limit/subset the dataset to the year 2007

```
gap07 <- filter(gapminder, year==2007)
```

- Then find the `pop` value for Ireland

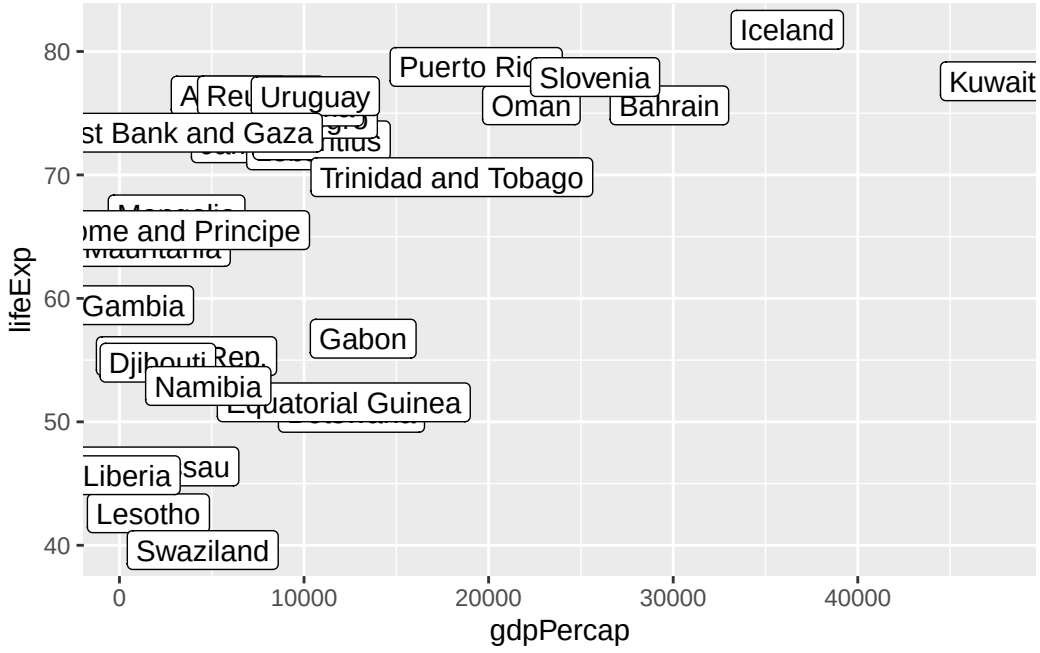
```
ire_pop <- filter(gap07, country=="Ireland")["pop"]
```

- Then extract all rows with the `pop` less than Ireland's

```
gap_small <- filter(gap07, pop < 4109086)
nrow(gap_small)
```

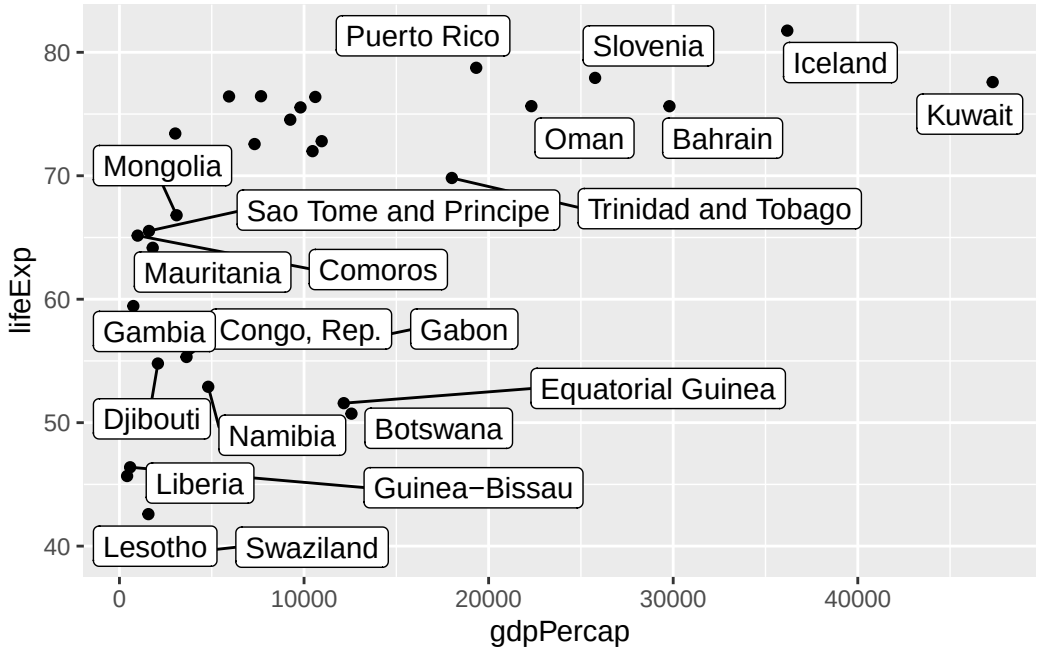
```
[1] 31
```

```
ggplot(gap_small) +
  aes(gdpPercap, lifeExp, label=country) +
  geom_point() +
  geom_label()
```




```
library(ggrepel)

ggplot(gap_small) +
  aes(gdpPercap, lifeExp, label=country) +
  geom_point() +
  geom_label_repel()
```

```
Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).