Class 5: Data Viz with ggplot

Emily Ignatoff (A16732102)

Intro to ggplot

There are many graphics systems in R (ways to make plots and figures). These include "base" R plots. Today we will focus mostly on the **ggplot2** package.

> Q: Which plot types are typically NOT used to compare distributions of numeric variables? **Network graphs**

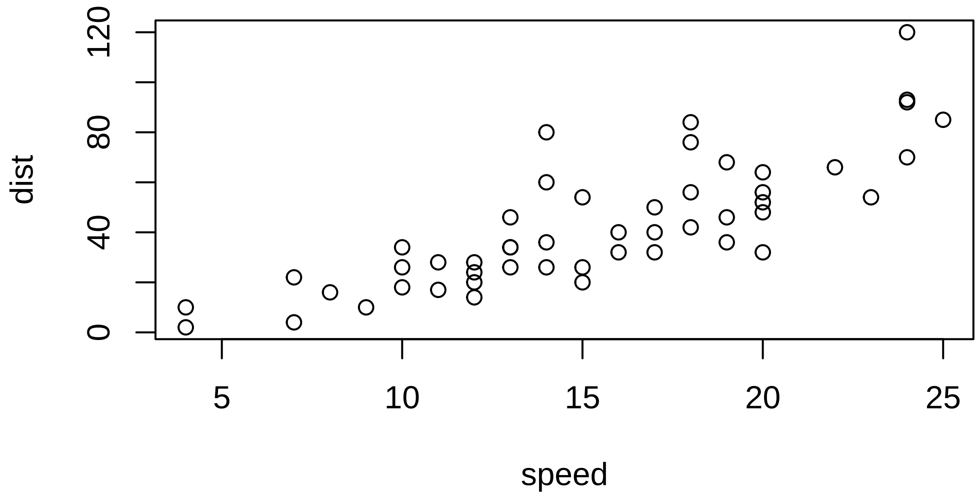
> Q: Which statement about data visualization with ggplot2 is incorrect? **ggplot is the only way to create plots in R**

Let's start with a plot of a built-in dataset called `cars`.

```r
head(cars)
```

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

```r
plot(cars)
```



Let's see how we can make this figure using **ggplot2**. For sake of clarity, I already have **ggplot2** installed, and thus have skipped the `install.package(ggplot2)` command. To install any package in R, I use the function `install_package()`.

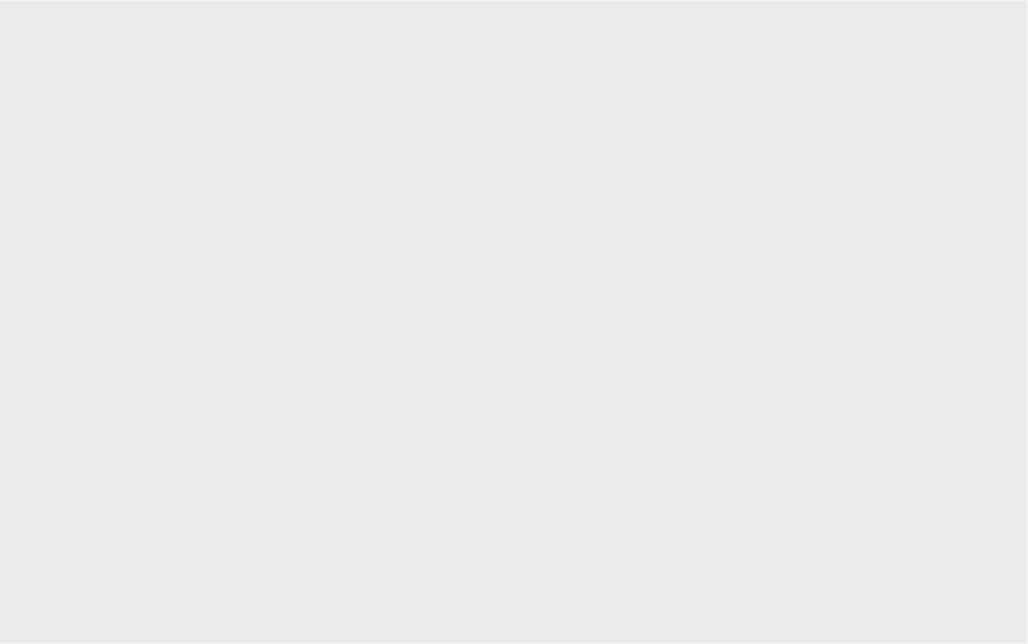
> Note: do NOT install packages inside the Quarto document, it is better to do this directly in the console.

Before I can use any functions from add on packages, I must load the package using the "library()" function, in this case `library(ggplot2)`

```r
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.3.3
```

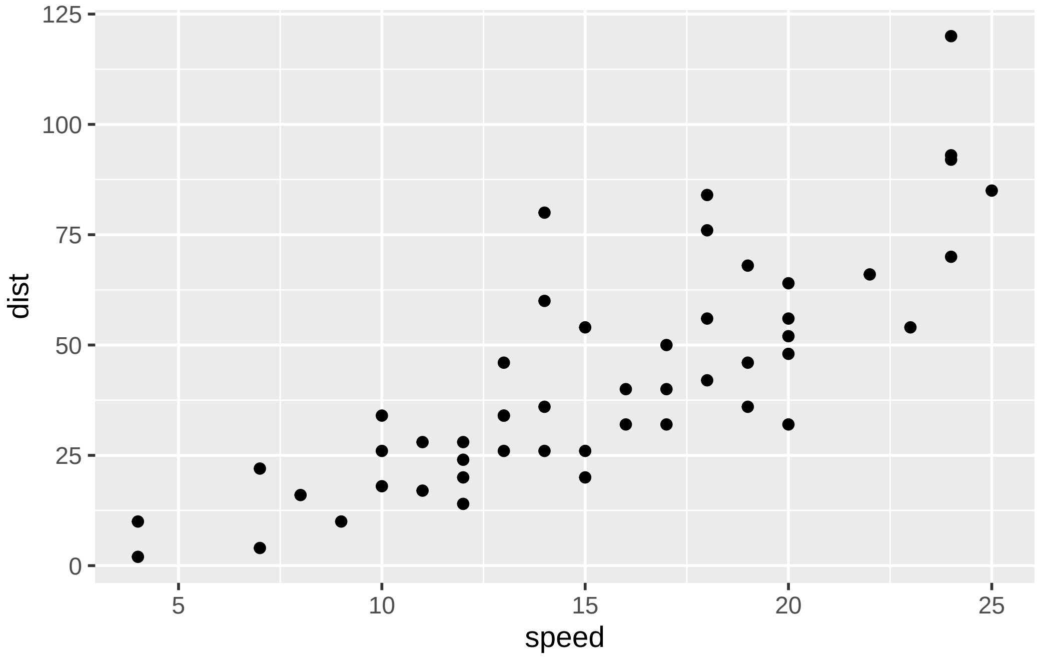
```r
ggplot(cars)
```



All ggplot figures have at least 3 layers. These are:

- **Data** (input dataset to plit)
- **Aesthetics** (aes) (aesthetic mapping of data on plot)
- **Geometry** (geo) (point, line, bar, etc that I want to draw)

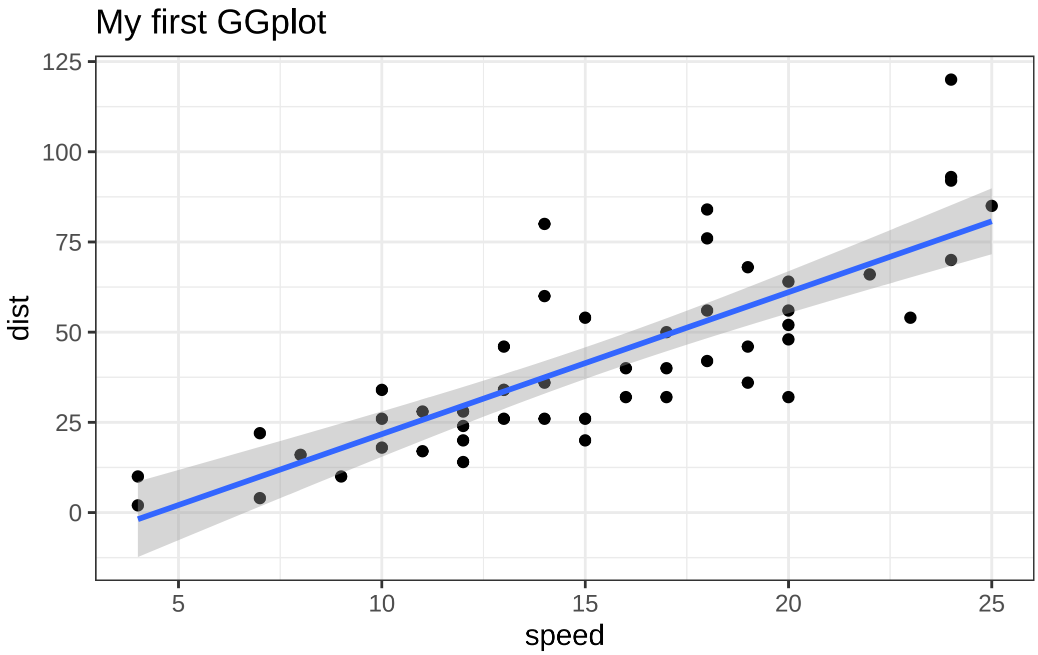
```r
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



Let’s add a line to show the relationship between distance and speed

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm") +
  theme_bw() +
  labs(title="My first GGplot")
```

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



Q: Which geometric layer should be used to create scatter plots in ggplot2? **geom_point()**

Code to read the dataset:

```
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: Use the nrow() function to find out how many genes are in this dataset. What is your answer? **5196 rows**

```
nrow(genes)
```

```
[1] 5196
```

Q: 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? **4 columns**

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

Q: 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? **127 genes**

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

```

      down unchanging         up 
        72       4997        127 
```

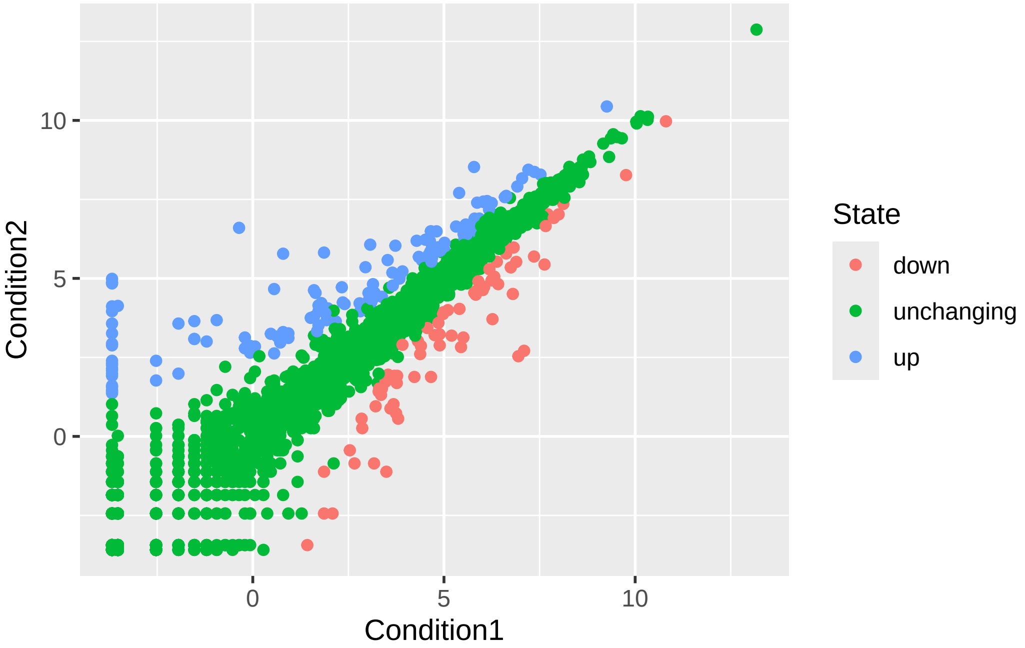
Q: Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset? **2.44 percent of genes**

```
round(table(genes$State)/nrow(genes), 4) * 100
```

```

      down unchanging         up 
      1.39      96.17       2.44 
```

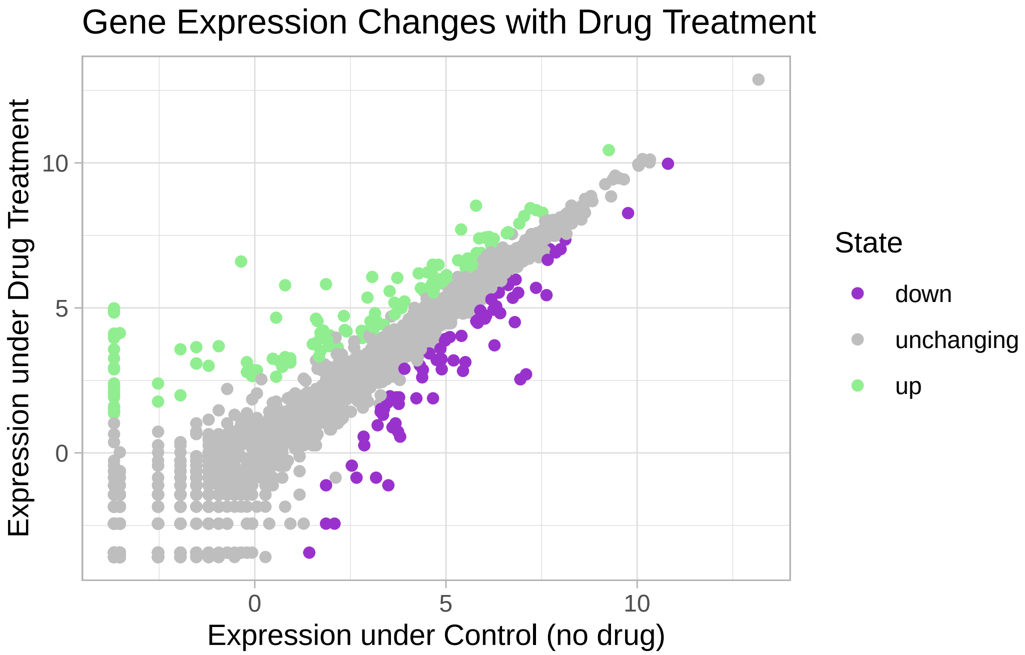
A first plot of this dataset:

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



Now let’s fix the color scheme and add titles/labels:

```
x + scale_color_manual(values= c("darkorchid","gray","lightgreen")) +
labs(title="Gene Expression Changes with Drug Treatment",
       x="Expression under Control (no drug)",
       y="Expression under Drug Treatment") +
  theme_light()
```



Let's explore more plots we can make using the ggplot2 pacakge!

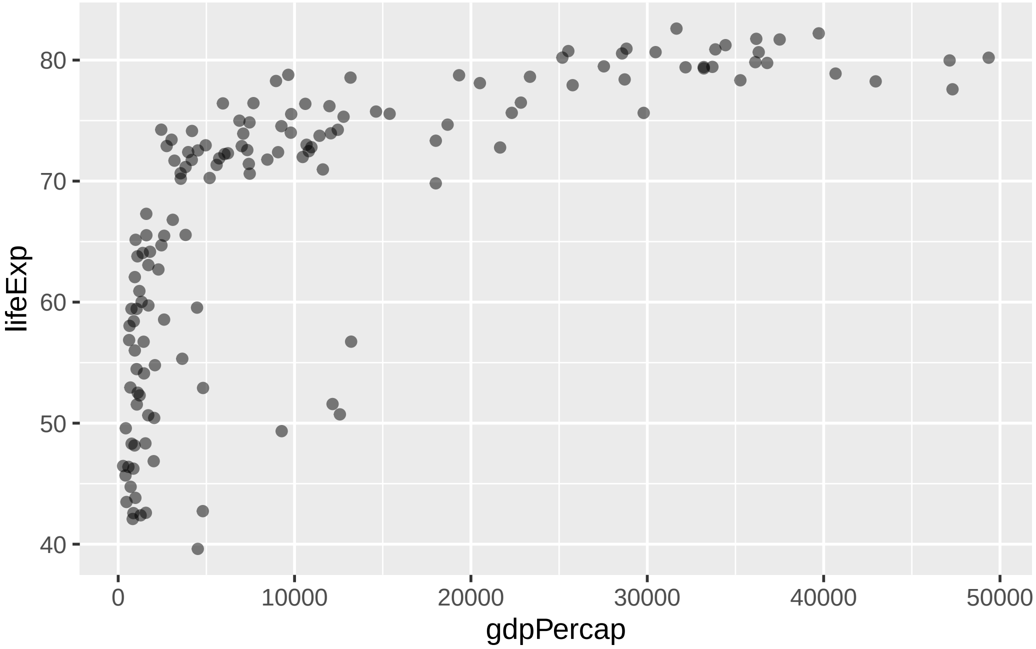
```
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.ts

gapminder <- read.delim(url)

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)
```

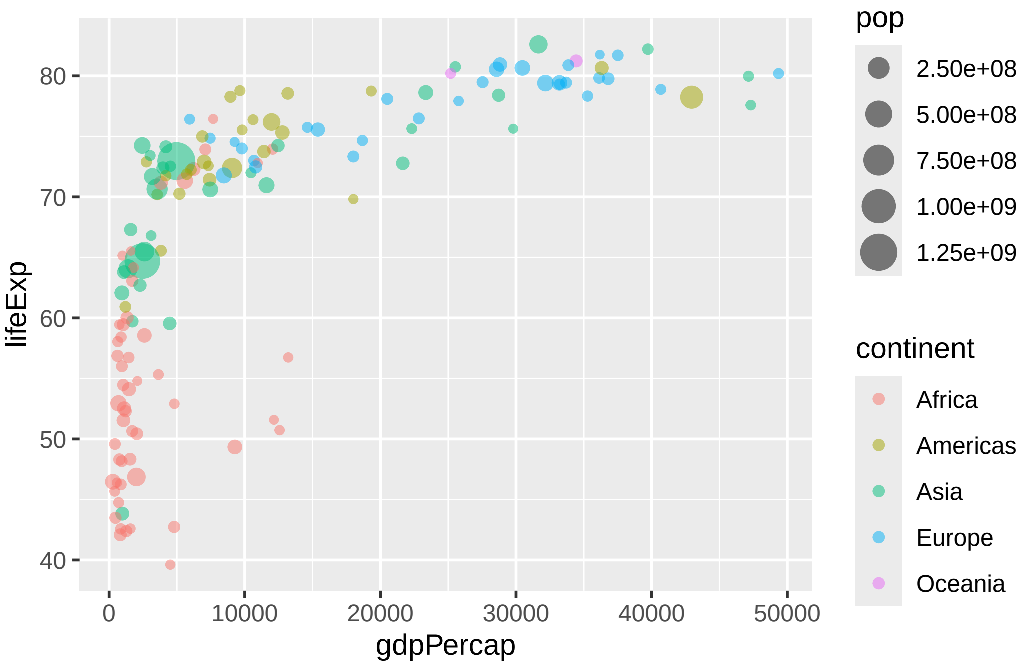
We have now filtered data from just 2007 in the gapminder dataset, I will now plot this:

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



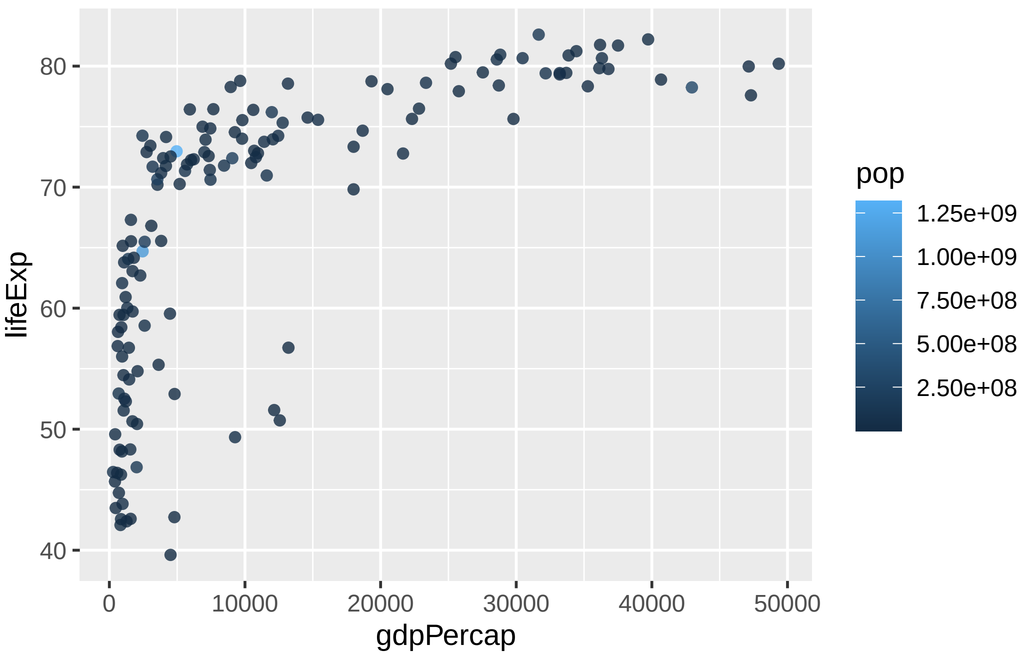
Let’s add some additional aesthetics:

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



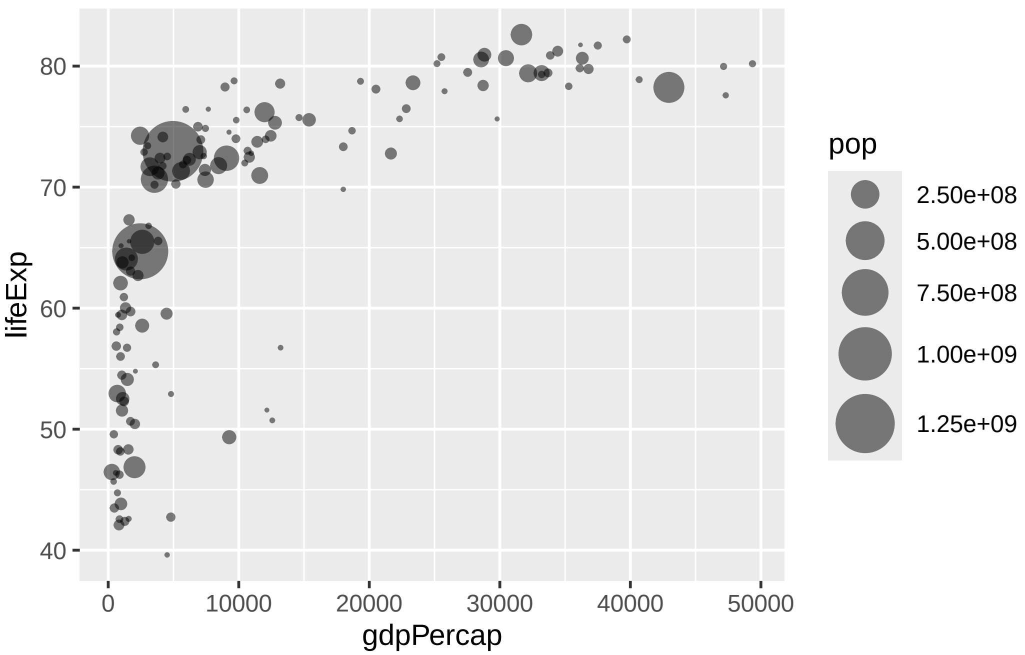
Let’s look at the continuous variation upon this graph:

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



Let's look at population by size instead of color:

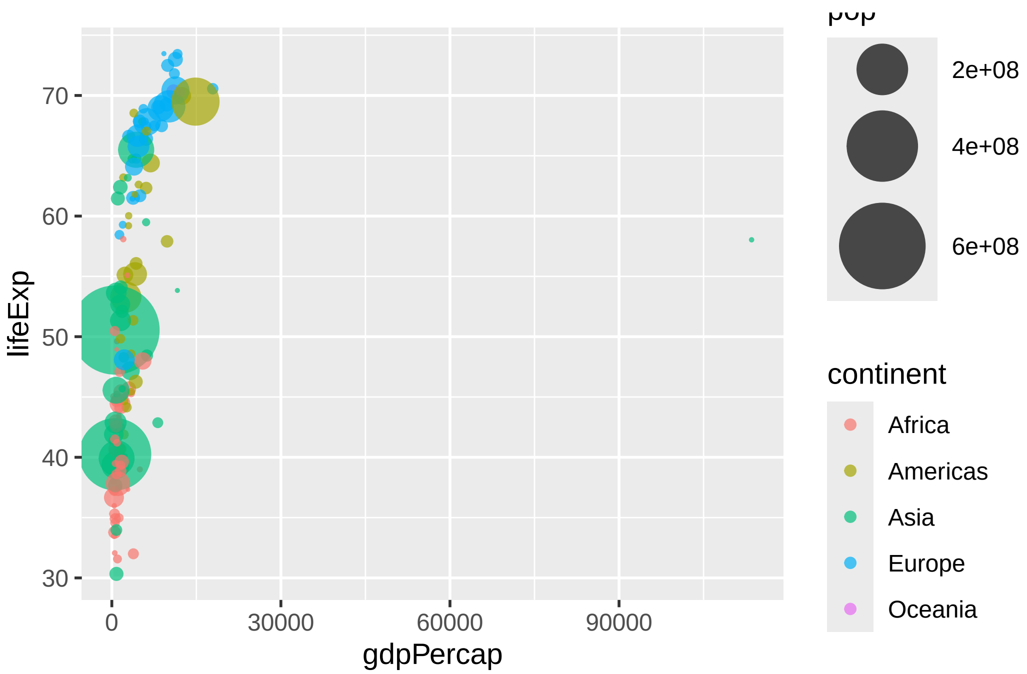
```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, size=pop)+
  geom_point(alpha=0.5) +
  scale_size_area(max_size = 10)
```



I will now do the same to data from 1957:

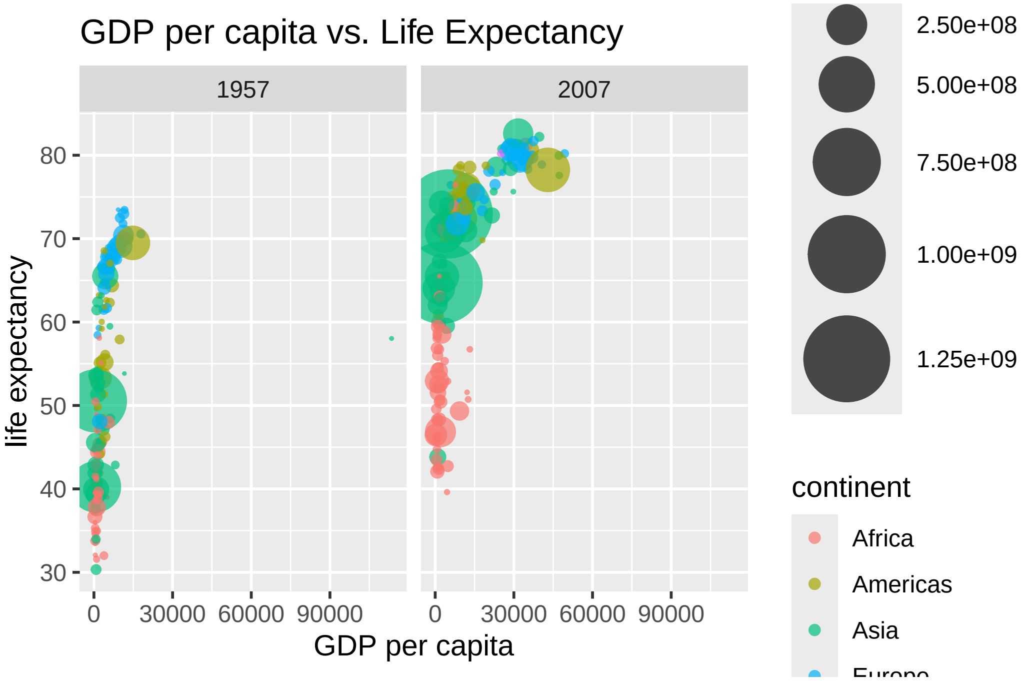
```
gapminder_1957 <- gapminder %>% filter(year==1957)
```

```
ggplot(gapminder_1957) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop)+
  geom_point(alpha=0.7)+
  scale_size_area(max_size=15)
```



I will now introduce both years:

```
gapminder_2years <- gapminder %>% filter(year==1957 | year==2007)

ggplot(gapminder_2years) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop)+
  geom_point(alpha=0.7)+
  scale_size_area(max_size=15)+
  facet_wrap(~year)+
  labs(title="GDP per capita vs. Life Expectancy",
       x="GDP per capita",
       y="life expectancy")
```

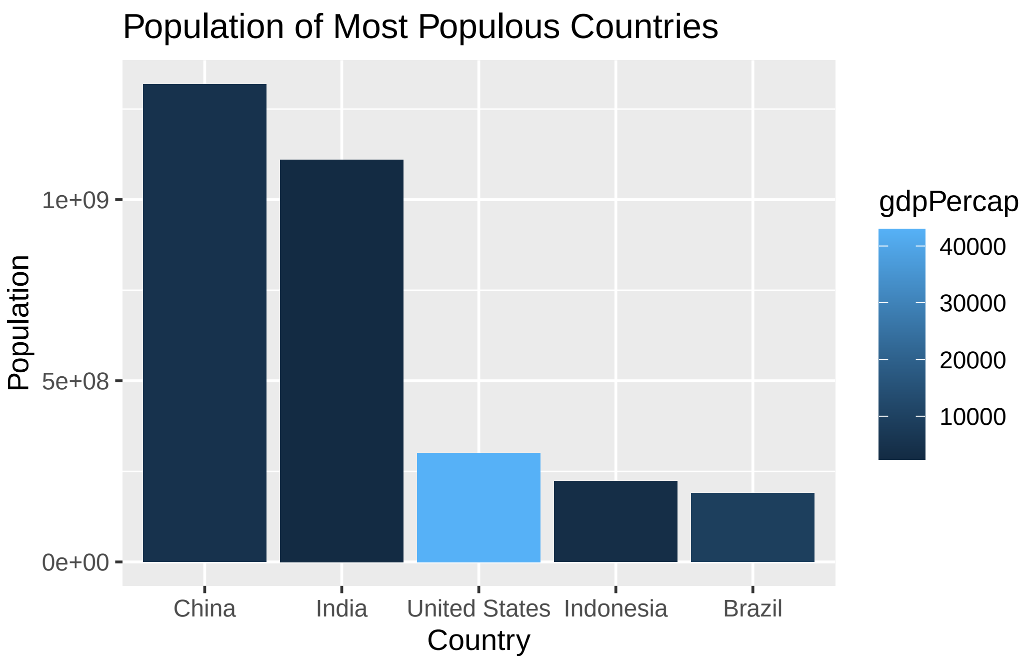


Let us now observe this data using boxplots:

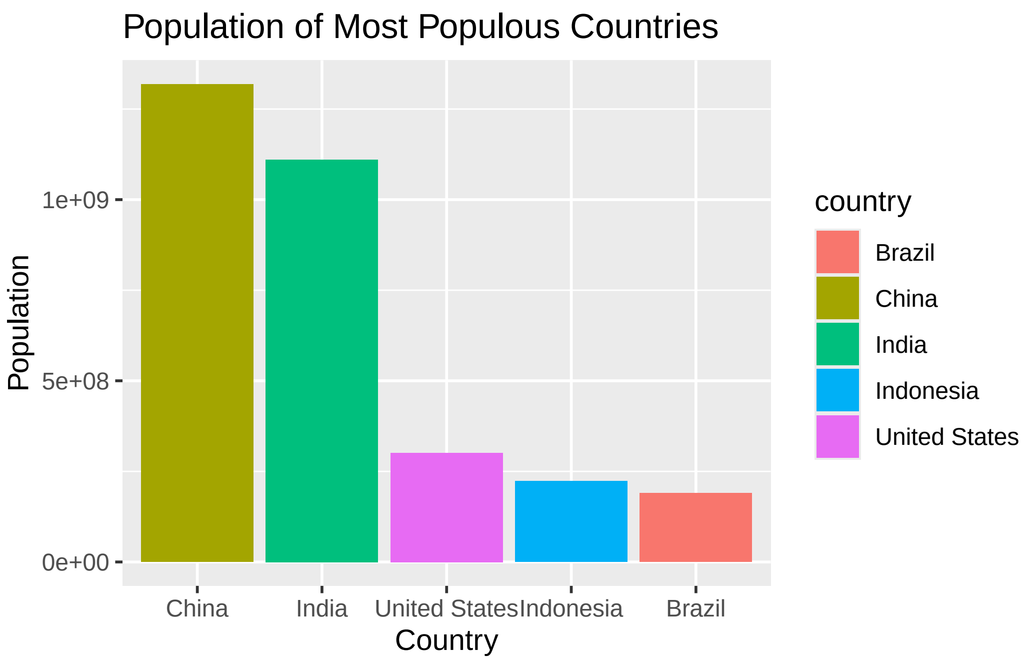
```
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)

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) +
  aes(x=reorder(country, -pop), y=pop, fill=gdpPercap)+
  geom_col() +
  labs(title="Population of Most Populous Countries",
      x="Country",
      y="Population")
```



```
ggplot(gapminder_top5) +
  aes(x=reorder(country, -pop), y=pop, fill=country)+
  geom_col() +
  labs(title="Population of Most Populous Countries",
      x="Country",
      y="Population")
```

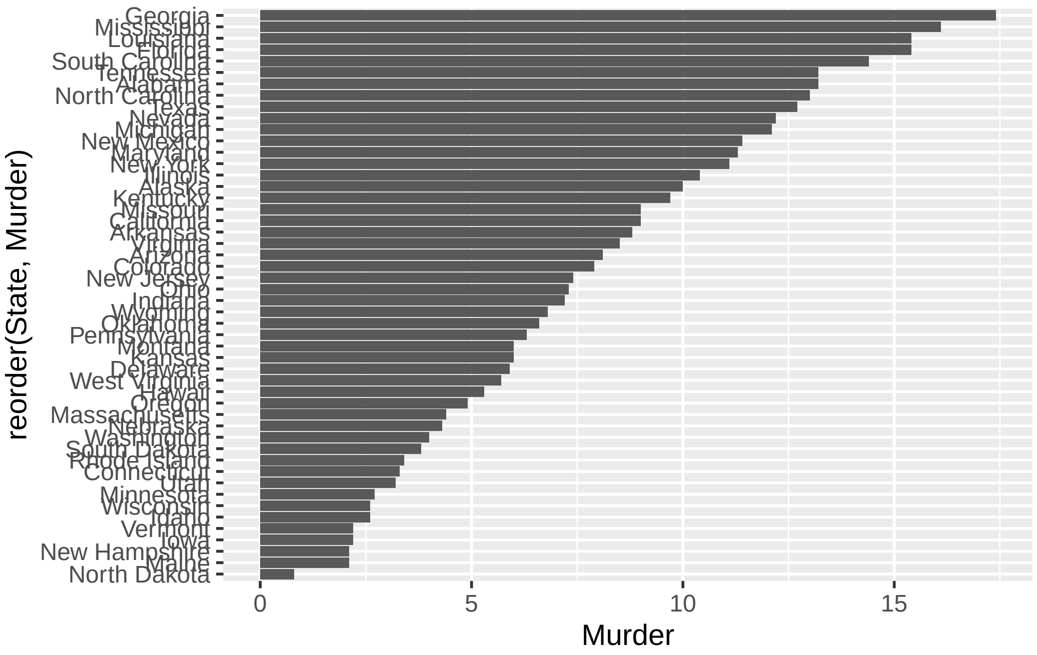


Flipping coordinates:

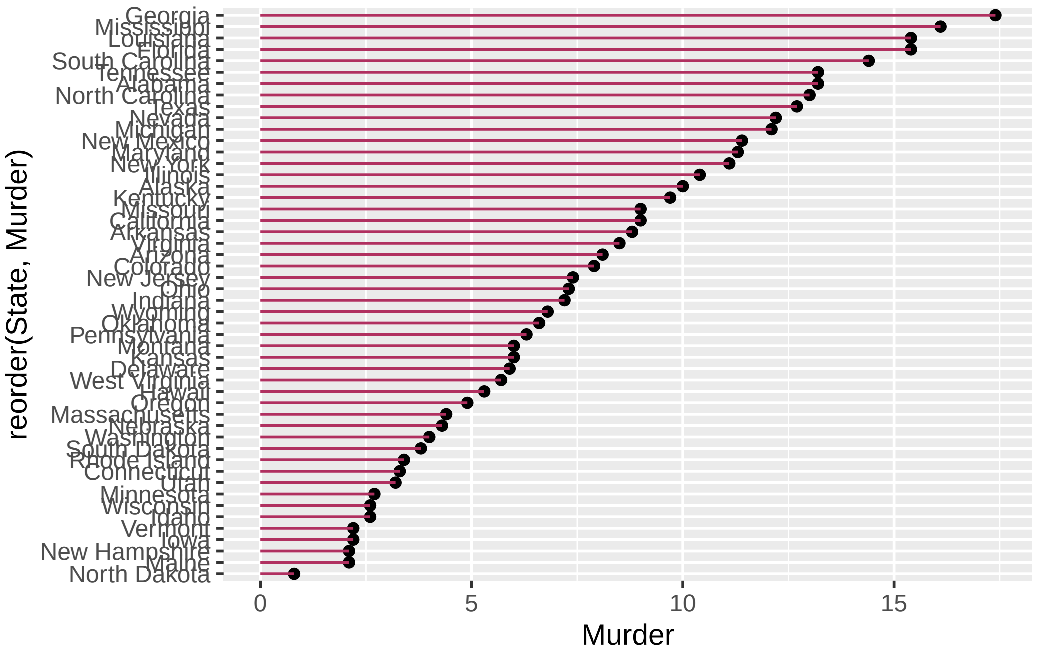
```
head(USArrests)
```

```
           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
```

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



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



Let's animate!

```
library(gifski)
```

```
Warning: package 'gifski' was built under R version 4.3.3
```

```
library(gganimate)
```

```
Warning: package 'gganimate' was built under R version 4.3.3
```

```
#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 by continent
  #facet_wrap(~continent) +
  # Here comes the gganimate specific bits
  #labs(title = 'Year: {frame_time}', x = 'GDP per capita', y = 'life expectancy') +
 # transition_time(year) +
 # shadow_wake(wake_length = 0.1, alpha = FALSE)
```

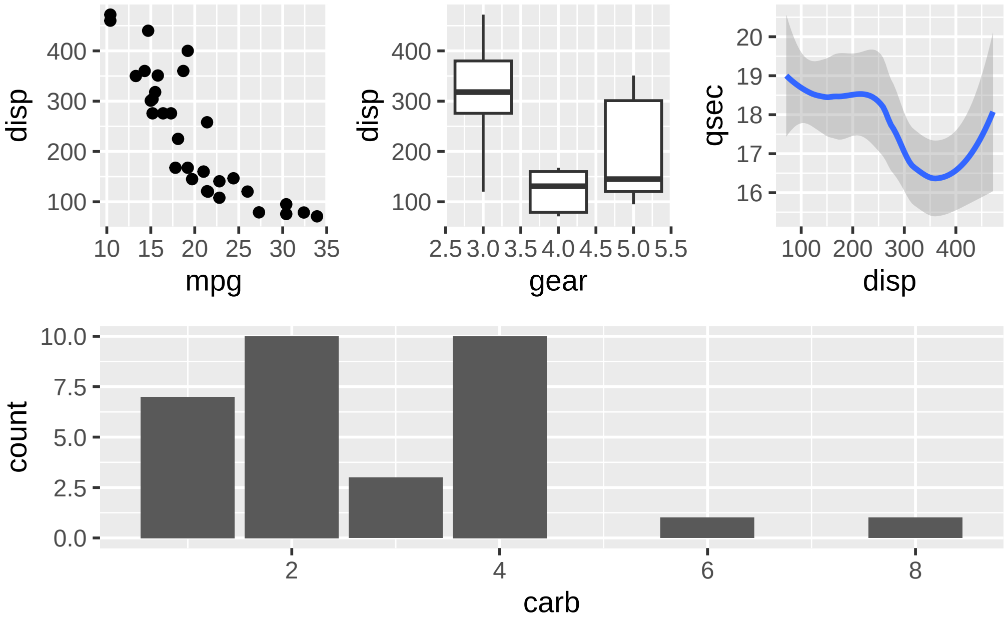
Finally, let’s make a multipanel figure:

```
library(patchwork)
```

```
Warning: package 'patchwork' was built under R version 4.3.3
```

```
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))

(p1|p2|p3) / p4
```

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


disp
mpg
disp
gear
qsec
disp
count
carb