Class 5: Data Viz & ggplot

Claire Lua A16922295

Table of contents

Background

There are many graphics systems available in R. These include "base" R and tons of add on packages like **ggplot2**.

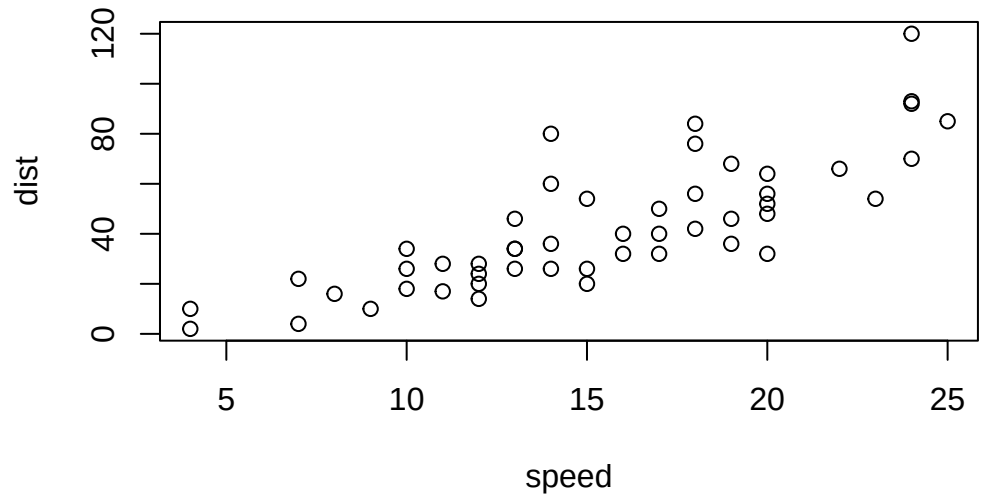
Let's compare "base" and **ggplot2** briefly. We can use some example data that is built-in with R called `cars`:

```
head(cars)
```

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

In base R I can just call `plot()`

```
plot(cars)
```



How can we do this with **ggplot2**

First we need to install the package. We do this `install.packages("ggplot2")`. I only need to do this once and then it will be available on my computer from then on.

> **Key point**: I only install packages in the R console not within quarto docs or Rscripts

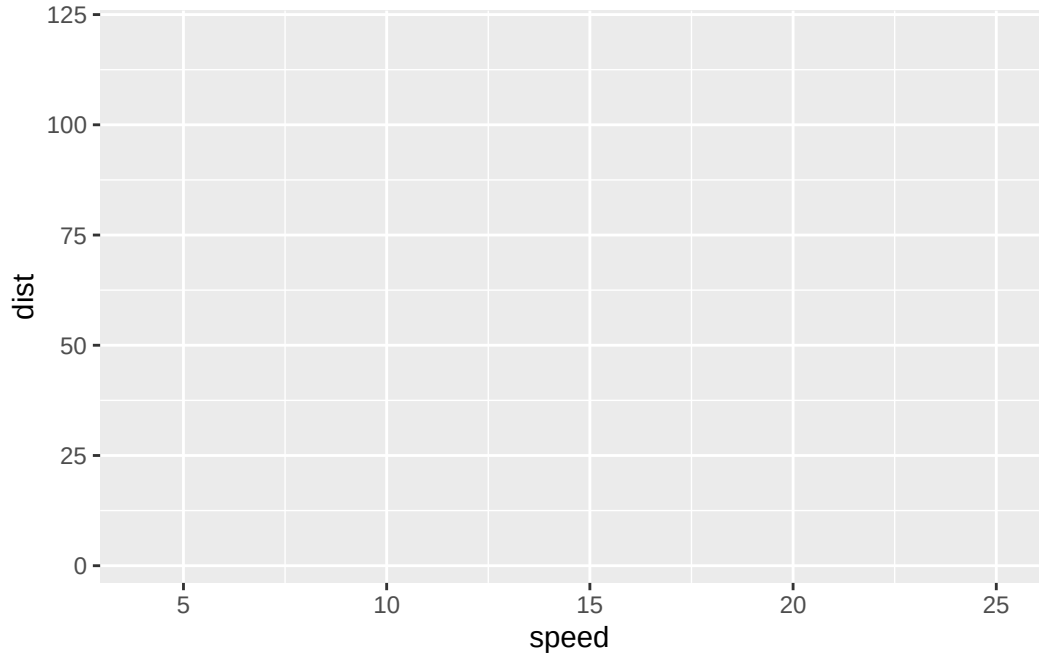
Before I use any add-on package I must load it up with a call to `library()`

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

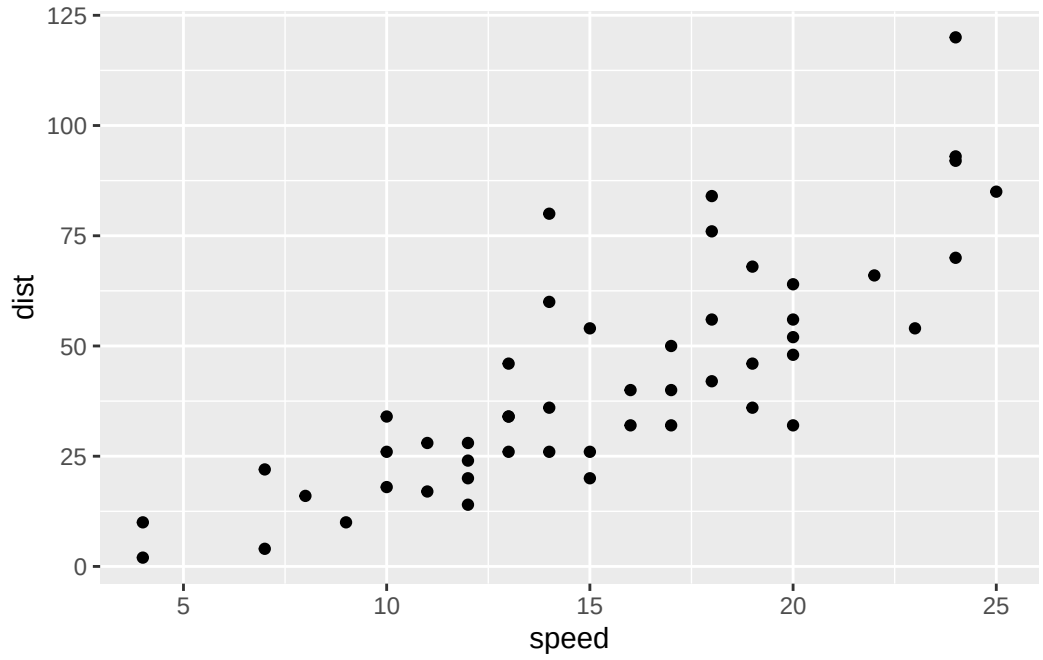
Every ggplot has at least 3 things:

- the **data** (in our case `cars`)
- the **aes**thetics (how the data map to the plot)
- the **geom**s that determine how the plot is drawn (lines, points, columns, etc.)

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




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



For "simple" plots ggplot is much more verbose than base R but the defaults are nicer and for complicated plots it becomes much more efficient and structured.

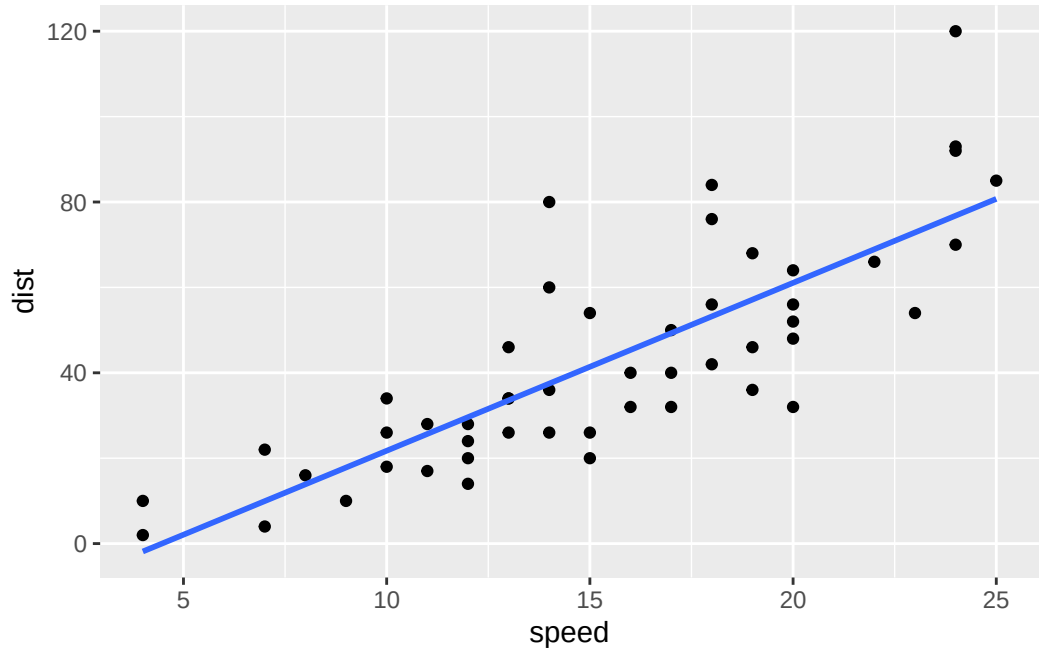
> Q. Add a line to show the relationship of speed to stopping distance (i.e. add another "layer")

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

I can always save any ggplot object (i.e. plot) and use it later for adding more layers.

```
p
```

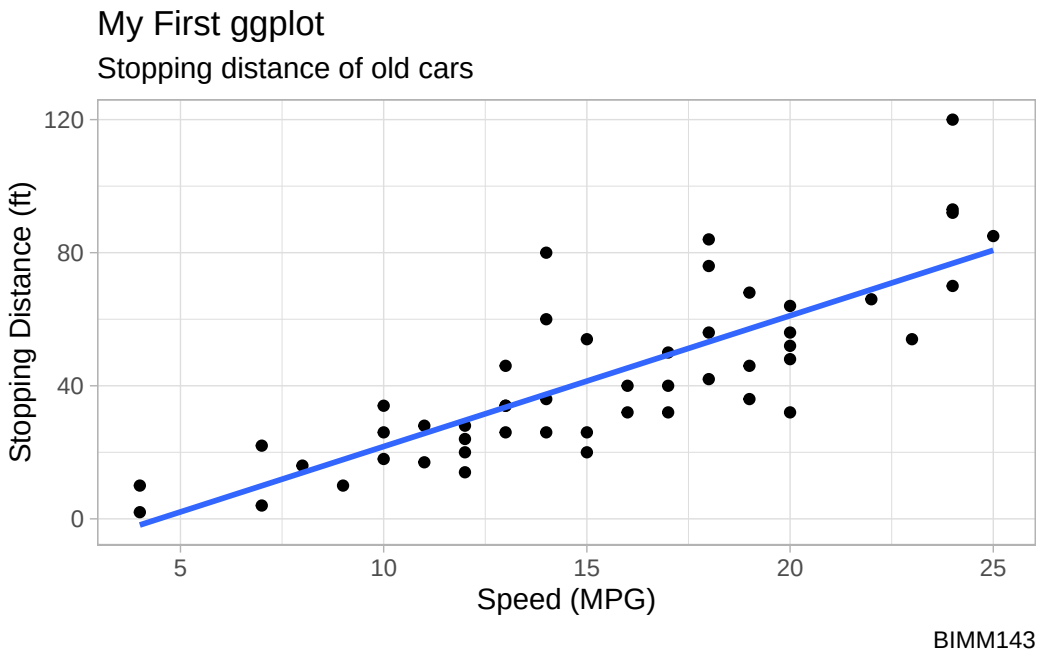
```
`geom_smooth()` using formula = 'y ~ x'
```



> Q. Add a title and suntitle to the plot

```
p + labs(title = "My First ggplot", subtitle = "Stopping distance of old cars", caption = "Bl
  theme_light()
```

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



Gene Expression 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 wee dataset?

```
nrow(genes)
```

```
[1] 5196
```

Q. How many columns are there? What are the names?

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

Q. How many “up” and “down” regulated genes are there?

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

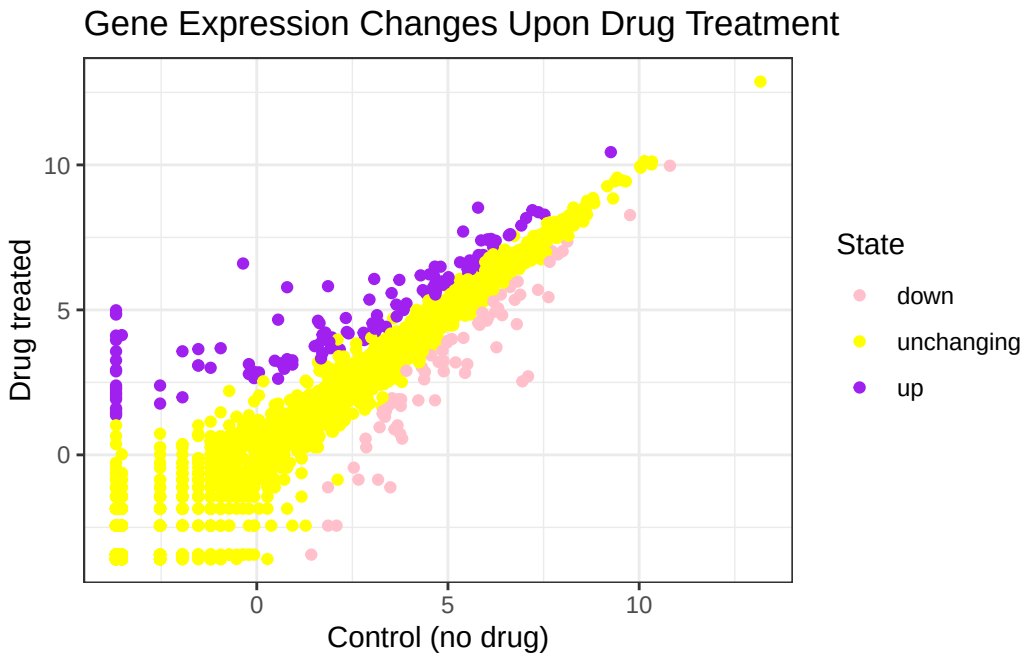
```
      down unchanging         up
        72       4997        127
```

Q. What fraction of total genes is up-regulated in the dataset?

Custom Color Plot

Q. Make a first plot of this data

```
ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State) +
  scale_color_manual(values=c("pink", "yellow","purple")) +
  geom_point() +
  labs(title="Gene Expression Changes Upon Drug Treatment", x="Control (no drug)", y="Drug tr
  theme_bw()
```



Car Type Plot

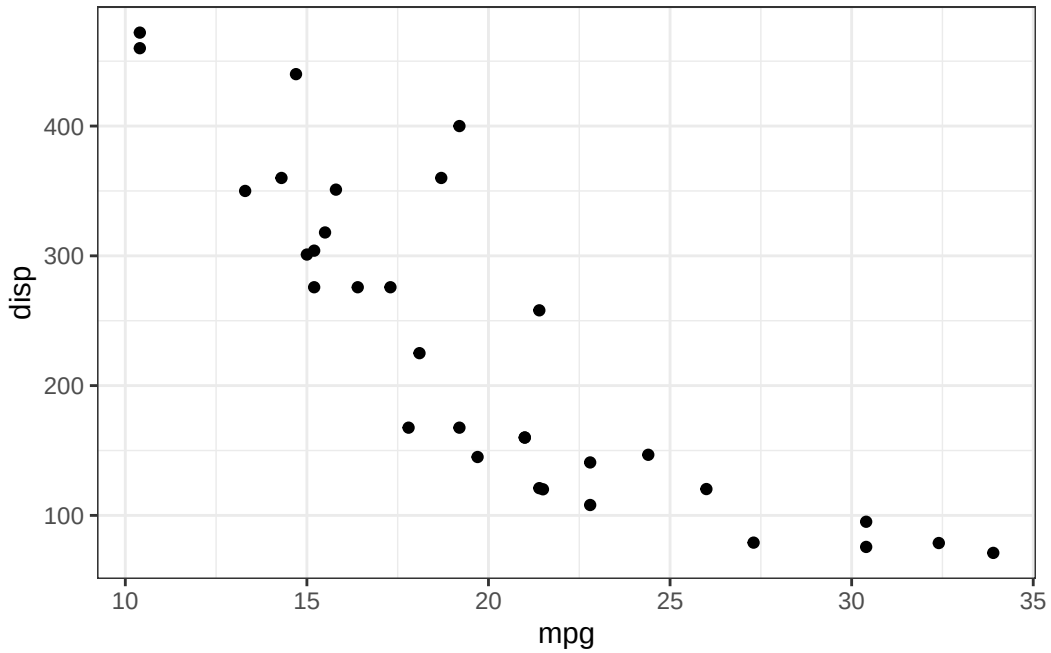
Let's plot some aspects of the in-built `mtcars` dataset

```
head(mtcars)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

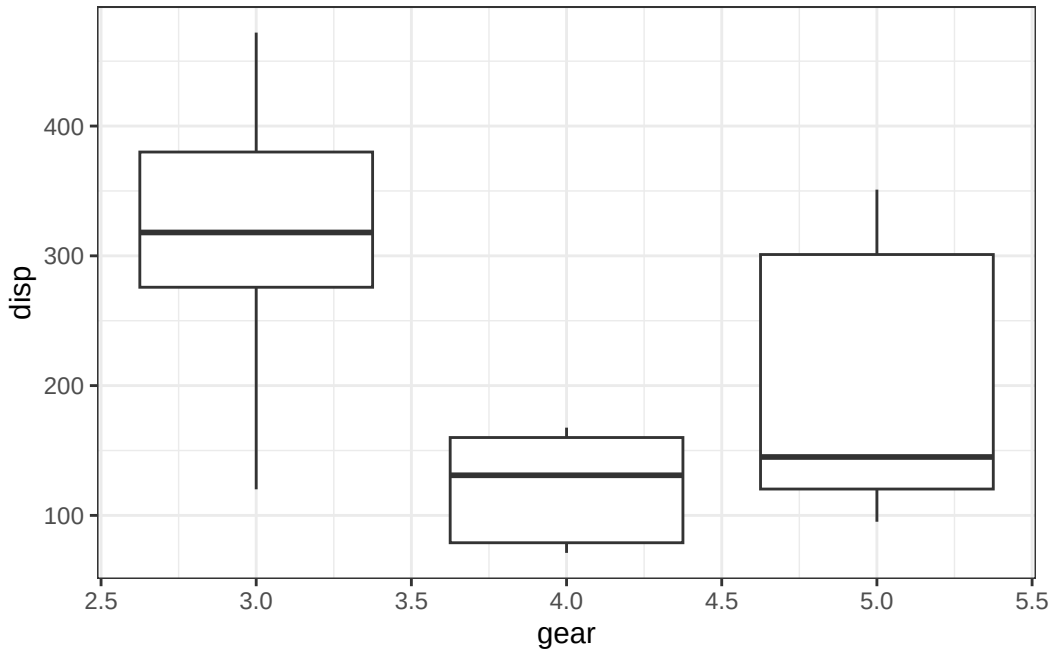
Q. Scatter plot of `mpg` vs `disp`

```
p1 = ggplot(mtcars) +
  aes(mpg, disp) +
  geom_point()+
  theme_bw()

p1
```



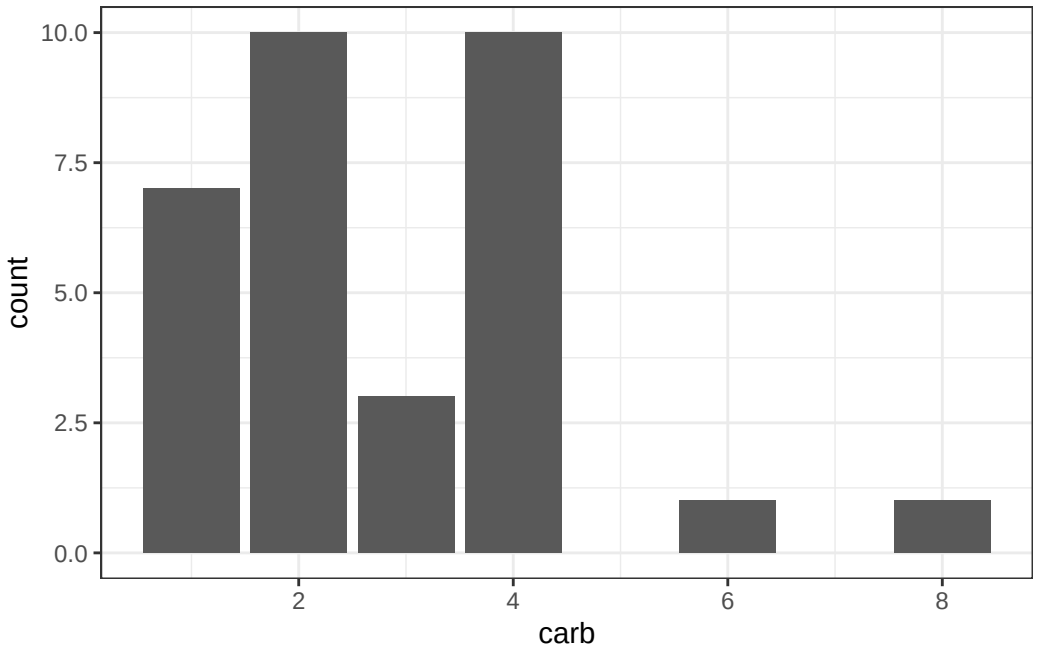
Q. Box plot of `gear` vs `disp`

```
p2 = ggplot(mtcars) +
  aes(x=gear, y=disp, group = gear) +
  geom_boxplot() +
  theme_bw()

p2
```



Q. Bar plot of `carb`

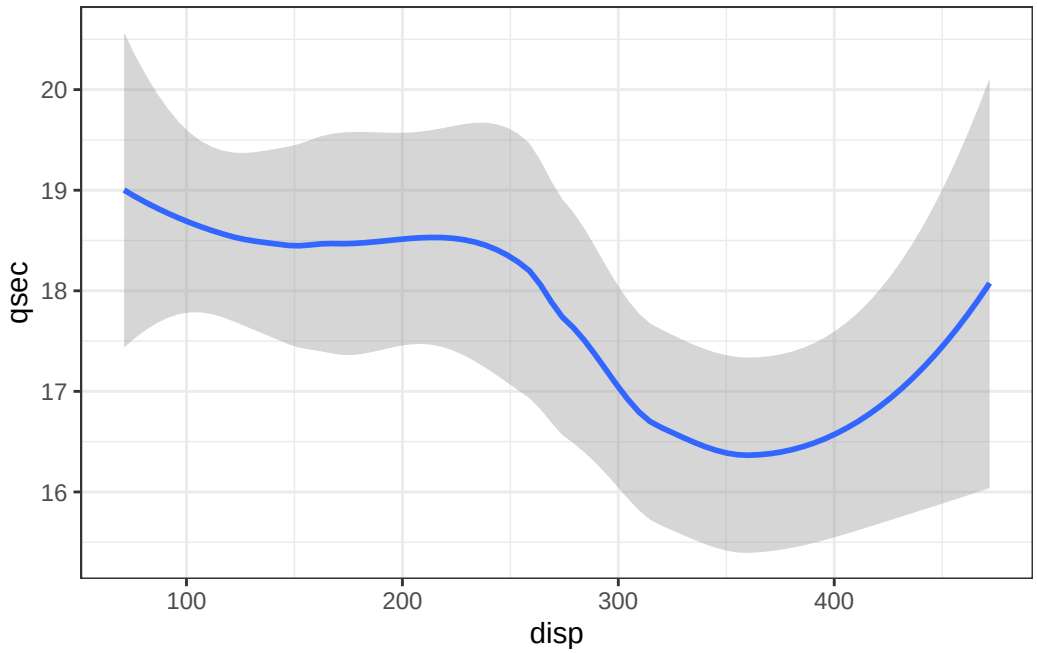
```
p3 = ggplot(mtcars) +
  aes(carb) +
  geom_bar() +
  theme_bw()

p3
```



Q. Smooth of `disp` vs `qsec`

```
p4 = ggplot(mtcars) +
  aes(disp, qsec) +
  geom_smooth() +
  theme_bw()

p4
```

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

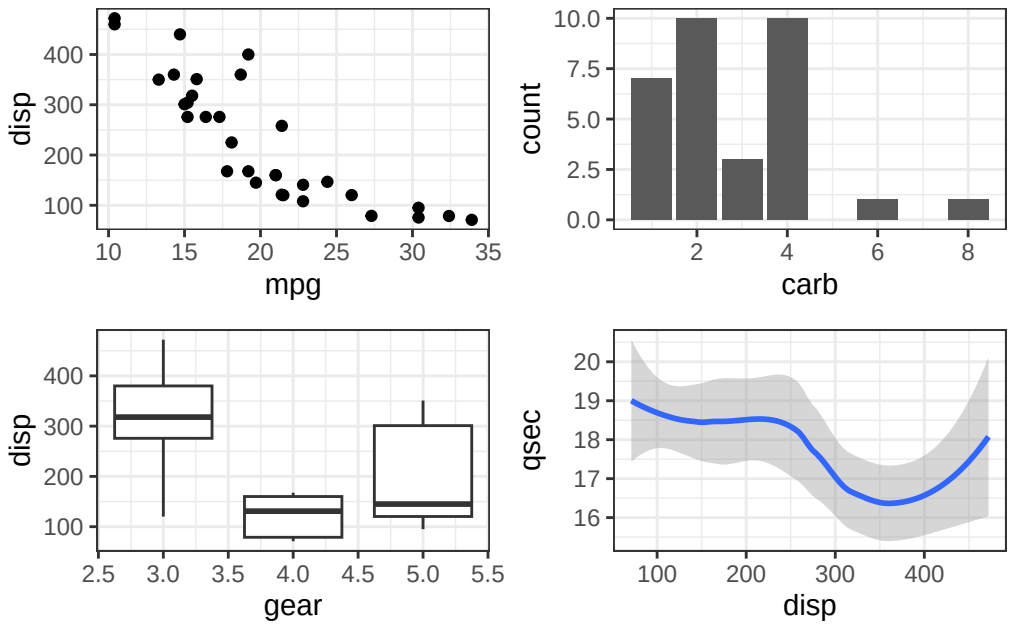


I want to combine all these plots into one figure with multiple pannels.

We can use the **patchwork** package to do this

```
library(patchwork)

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

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




```
ggsave(filename="myplot.png", width=5, height=3)
```

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

#Country Population Data Plot

And a wee peak

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

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
```

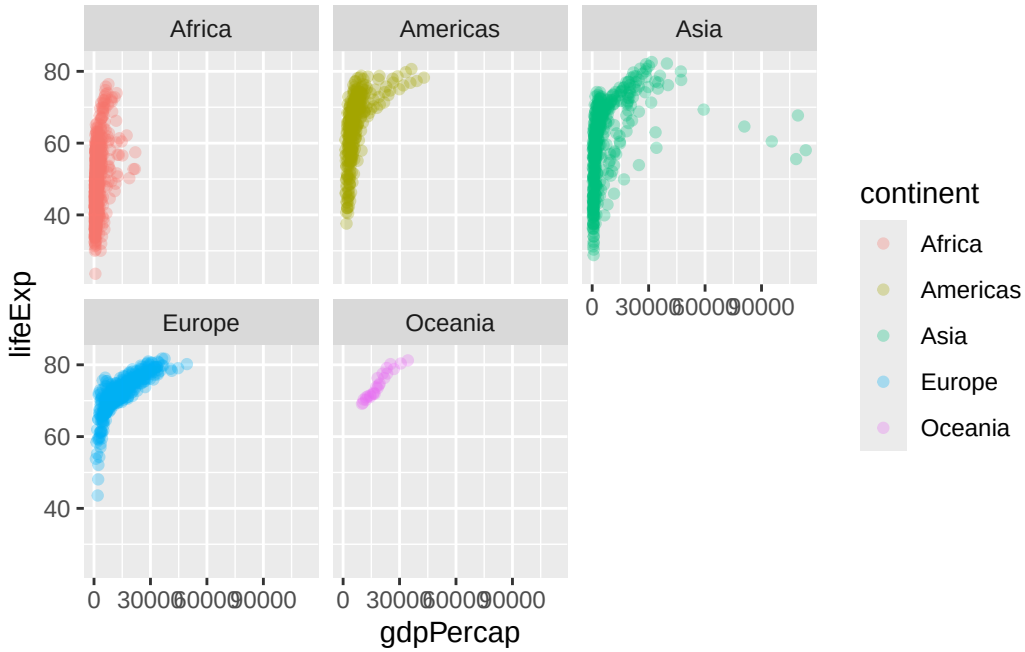
Q. How many countries are in this dataset?

```
length(table(gapminder$country))
```

```
[1] 142
```

Q. Plot gdpPercap vs lifeExp color by continent

```
ggplot(gapminder) +
  aes(gdpPercap, lifeExp, colour = continent) +
  geom_point(alpha=0.3) +
  facet_wrap(~continent)
```




```
  theme_bw()
```

```
List of 136
 $ line                            :List of 6
  ..$ colour       : chr "black"
  ..$ linewidth    : num 0.5
  ..$ linetype     : num 1
  ..$ lineend      : chr "butt"
```

```
 ..$ arrow        : logi FALSE
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_line" "element"
$ rect                           :List of 5
 ..$ fill         : chr "white"
 ..$ colour       : chr "black"
 ..$ linewidth    : num 0.5
 ..$ linetype     : num 1
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_rect" "element"
$ text                           :List of 11
 ..$ family       : chr ""
 ..$ face         : chr "plain"
 ..$ colour       : chr "black"
 ..$ size         : num 11
 ..$ hjust        : num 0.5
 ..$ vjust        : num 0.5
 ..$ angle        : num 0
 ..$ lineheight   : num 0.9
 ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : logi FALSE
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ title                          : NULL
$ aspect.ratio                   : NULL
$ axis.title                     : NULL
$ axis.title.x                   :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.title.x.top               :List of 11
 ..$ family       : NULL
```

```
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 0
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.title.x.bottom             : NULL
$ axis.title.y                    :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : num 90
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.title.y.left               : NULL
$ axis.title.y.right              :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : num -90
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text                       :List of 11
```

```
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : chr "grey30"
 ..$ size         : 'rel' num 0.8
 ..$ hjust        : NULL
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : NULL
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.x                    :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.x.top                :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 0
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.x.bottom             : NULL
$ axis.text.y                    :List of 11
 ..$ family       : NULL
```

```
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 1
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.y.left                : NULL
$ axis.text.y.right               :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 0
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.theta                 : NULL
$ axis.text.r                     :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 0.5
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 2.2points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.ticks                      :List of 6
```

```
 ..$ colour       : chr "grey20"
 ..$ linewidth    : NULL
 ..$ linetype     : NULL
 ..$ lineend      : NULL
 ..$ arrow        : logi FALSE
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_line" "element"
$ axis.ticks.x                    : NULL
$ axis.ticks.x.top                : NULL
$ axis.ticks.x.bottom             : NULL
$ axis.ticks.y                    : NULL
$ axis.ticks.y.left               : NULL
$ axis.ticks.y.right              : NULL
$ axis.ticks.theta                : NULL
$ axis.ticks.r                    : NULL
$ axis.minor.ticks.x.top          : NULL
$ axis.minor.ticks.x.bottom       : NULL
$ axis.minor.ticks.y.left         : NULL
$ axis.minor.ticks.y.right        : NULL
$ axis.minor.ticks.theta          : NULL
$ axis.minor.ticks.r              : NULL
$ axis.ticks.length               : 'simpleUnit' num 2.75points
 ..- attr(*, "unit")= int 8
$ axis.ticks.length.x             : NULL
$ axis.ticks.length.x.top         : NULL
$ axis.ticks.length.x.bottom      : NULL
$ axis.ticks.length.y             : NULL
$ axis.ticks.length.y.left        : NULL
$ axis.ticks.length.y.right       : NULL
$ axis.ticks.length.theta         : NULL
$ axis.ticks.length.r             : NULL
$ axis.minor.ticks.length         : 'rel' num 0.75
$ axis.minor.ticks.length.x       : NULL
$ axis.minor.ticks.length.x.top   : NULL
$ axis.minor.ticks.length.x.bottom: NULL
$ axis.minor.ticks.length.y       : NULL
$ axis.minor.ticks.length.y.left  : NULL
$ axis.minor.ticks.length.y.right : NULL
$ axis.minor.ticks.length.theta   : NULL
$ axis.minor.ticks.length.r       : NULL
$ axis.line                       : list()
 ..- attr(*, "class")= chr [1:2] "element_blank" "element"
$ axis.line.x                     : NULL
```

```
$ axis.line.x.top                 : NULL
$ axis.line.x.bottom              : NULL
$ axis.line.y                     : NULL
$ axis.line.y.left                : NULL
$ axis.line.y.right               : NULL
$ axis.line.theta                 : NULL
$ axis.line.r                     : NULL
$ legend.background               :List of 5
 ..$ fill         : NULL
 ..$ colour       : logi NA
 ..$ linewidth    : NULL
 ..$ linetype     : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_rect" "element"
$ legend.margin                   : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5poi
 ..- attr(*, "unit")= int 8
$ legend.spacing                  : 'simpleUnit' num 11points
 ..- attr(*, "unit")= int 8
$ legend.spacing.x                : NULL
$ legend.spacing.y                : NULL
$ legend.key                      : NULL
$ legend.key.size                 : 'simpleUnit' num 1.2lines
 ..- attr(*, "unit")= int 3
$ legend.key.height               : NULL
$ legend.key.width                : NULL
$ legend.key.spacing              : 'simpleUnit' num 5.5points
 ..- attr(*, "unit")= int 8
$ legend.key.spacing.x            : NULL
$ legend.key.spacing.y            : NULL
$ legend.frame                    : NULL
$ legend.ticks                    : NULL
$ legend.ticks.length             : 'rel' num 0.2
$ legend.axis.line                : NULL
$ legend.text                     :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : 'rel' num 0.8
 ..$ hjust        : NULL
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : NULL
```

```
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ legend.text.position            : NULL
$ legend.title                    :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 0
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : NULL
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ legend.title.position           : NULL
$ legend.position                 : chr "right"
$ legend.position.inside          : NULL
$ legend.direction                : NULL
$ legend.byrow                    : NULL
$ legend.justification            : chr "center"
$ legend.justification.top        : NULL
$ legend.justification.bottom     : NULL
$ legend.justification.left       : NULL
$ legend.justification.right      : NULL
$ legend.justification.inside     : NULL
$ legend.location                 : NULL
$ legend.box                      : NULL
$ legend.box.just                 : NULL
$ legend.box.margin               : 'margin' num [1:4] 0cm 0cm 0cm 0cm
 ..- attr(*, "unit")= int 1
$ legend.box.background           : list()
 ..- attr(*, "class")= chr [1:2] "element_blank" "element"
$ legend.box.spacing              : 'simpleUnit' num 11points
 ..- attr(*, "unit")= int 8
 [list output truncated]
- attr(*, "class")= chr [1:2] "theme" "gg"
- attr(*, "complete")= logi TRUE
- attr(*, "validate")= logi TRUE
```