Class 5: Data Visualization with ggplot

Jessica Le (PID: A17321021)

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 **ggplot2** package.

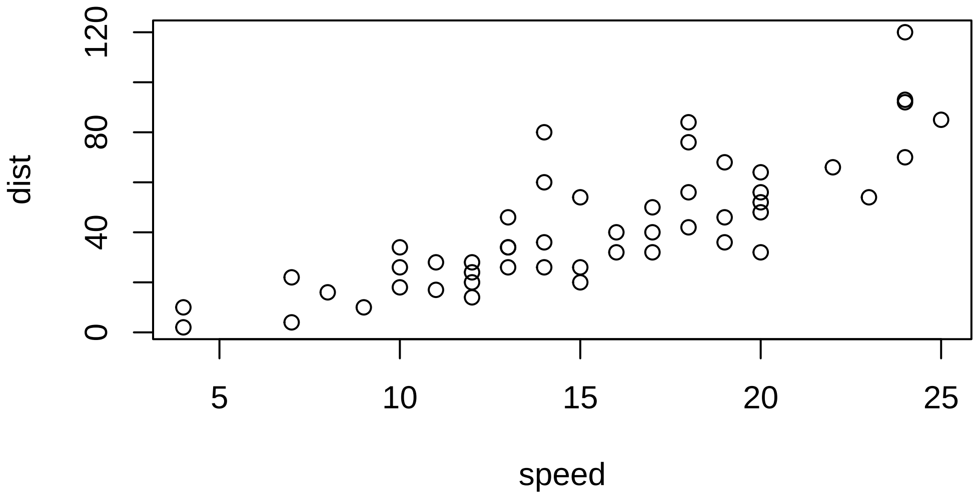
Let's start with a plot of a simple in-build dataset called 'cars'.

```
cars
```

```
   speed dist
1      4    2
2      4   10
3      7    4
4      7   22
5      8   16
6      9   10
7     10   18
8     10   26
9     10   34
10    11   17
11    11   28
12    12   14
13    12   20
14    12   24
15    12   28
16    13   26
17    13   34
18    13   34
19    13   46
20    14   26
21    14   36
22    14   60
```

23	14	80
24	15	20
25	15	26
26	15	54
27	16	32
28	16	40
29	17	32
30	17	40
31	17	50
32	18	42
33	18	56
34	18	76
35	18	84
36	19	36
37	19	46
38	19	68
39	20	32
40	20	48
41	20	52
42	20	56
43	20	64
44	22	66
45	23	54
46	24	70
47	24	92
48	24	93
49	24	120
50	25	85

```r
plot(cars)
```

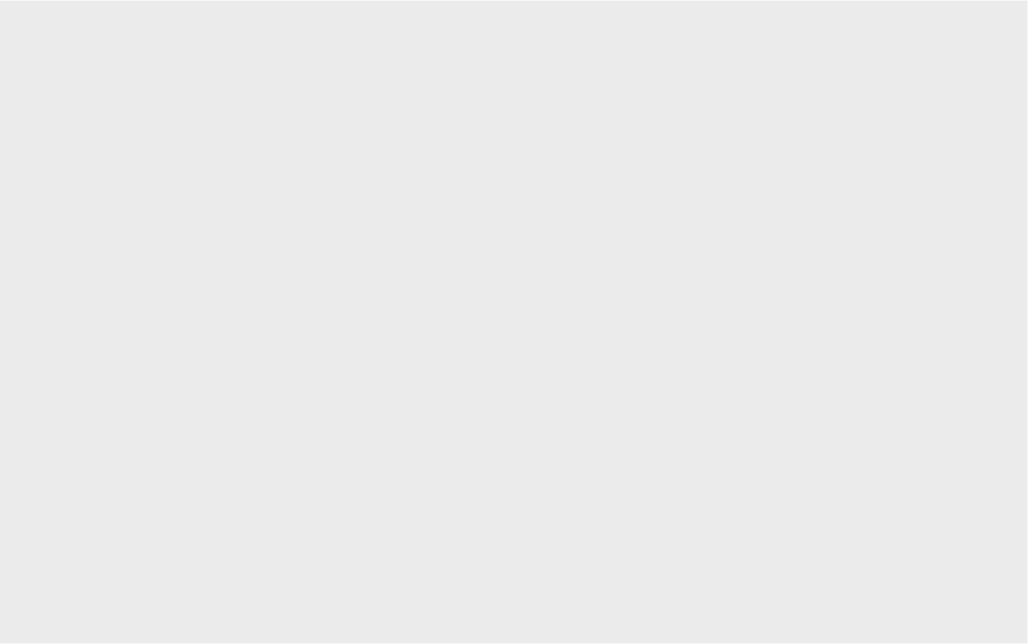


Let’s see how we can make this figure using **ggplot**. First I need to install this package on my computer. To install any R package, I use the function ‘install.packages()’.

> I will run ’install.packages(“ggplot2”) in my R console. Not on this quarto document because I don’t want it to reinstall each time I open the document.

Before I can use any functions from add on packages, I need to load the pacage from my “library()” with the ‘library(ggplot2)’ call.

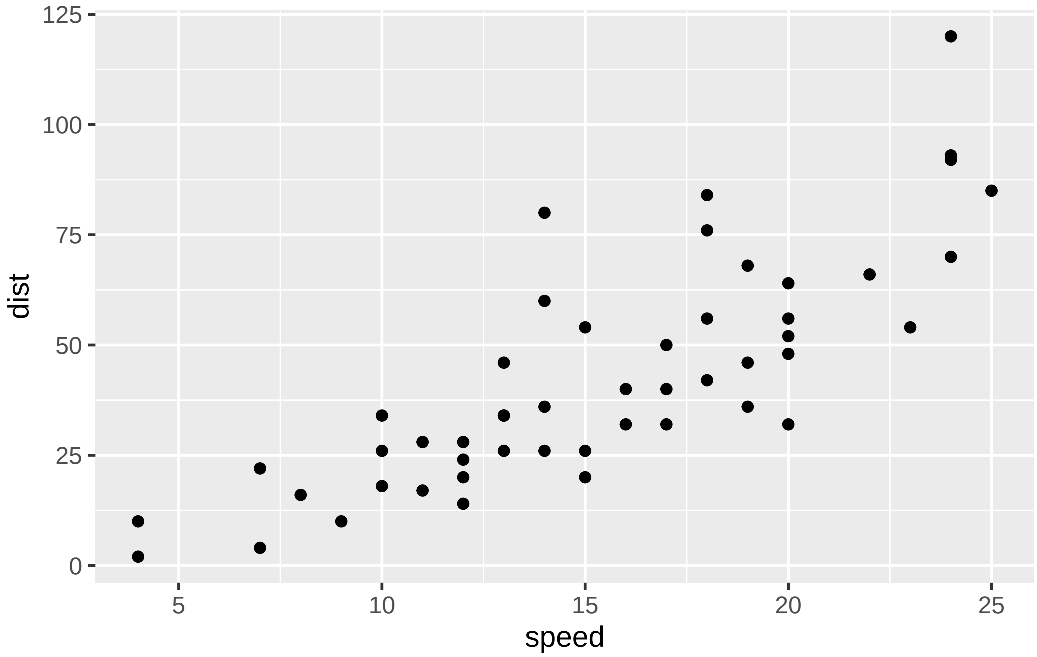
```r
library(ggplot2)
ggplot(cars)
```



All ggplot figures have at least 3 things (called layers). These include:

- **data** (the input dataset I want to plot from),
- **aes** (the aesthetic mapping of the data to my plot),
- **geoms** (the geom_point(), geom_line(), etc. that I want to draw)

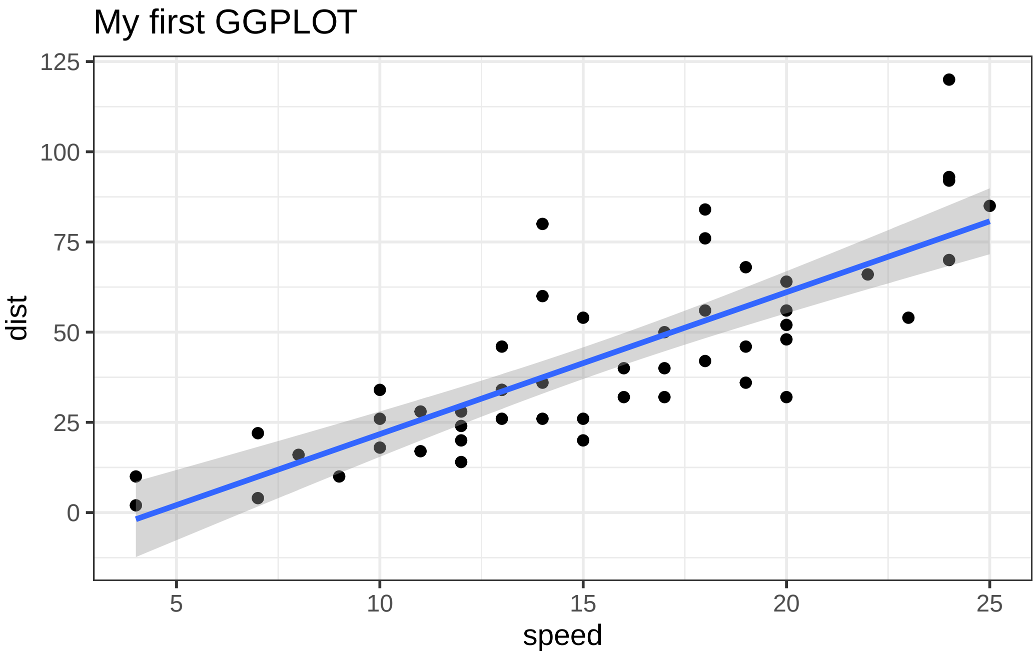
```r
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



Let’s add a line to show the relationship here:

```
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'
```



Q1 Which geometric layer should be used to create scatter plots in ggplot2?

geom_point()

Gene expression figure

The 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. How many genes are in this dataset?

```r
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?

```r
colnames(genes)
```

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

> 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?

```r
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?

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

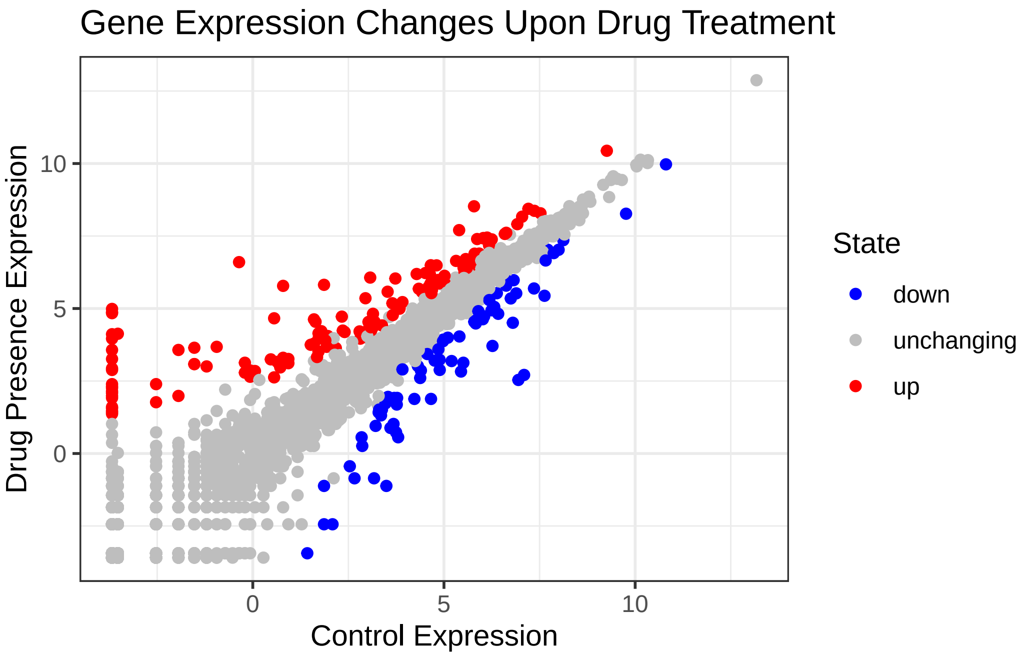
```

      down unchanging         up 
      0.01       0.96       0.02 
```

```r
n.tot <- nrow(genes)
vals <- table(genes$State)

vals.percent <- vals/n.tot *100
round (vals.percent, 2)
```

```

      down unchanging         up 
      1.39      96.17       2.44 
```

The first plot of this dataset.

```
ggplot(genes) +
  aes (x=Condition1, y=Condition2, col=State) +
  geom_point() +
  theme_bw() +
  labs(title="Gene Expression Changes Upon Drug Treatment",
       x="Control Expression",
       y="Drug Presence Expression") +
    scale_color_manual(values=c("blue", "gray", "red"))
```



Going Further

The file is located only.

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

Filter out only data from the year 2007.

First, I will run ’install.packages(“dplyr”) in my R console.

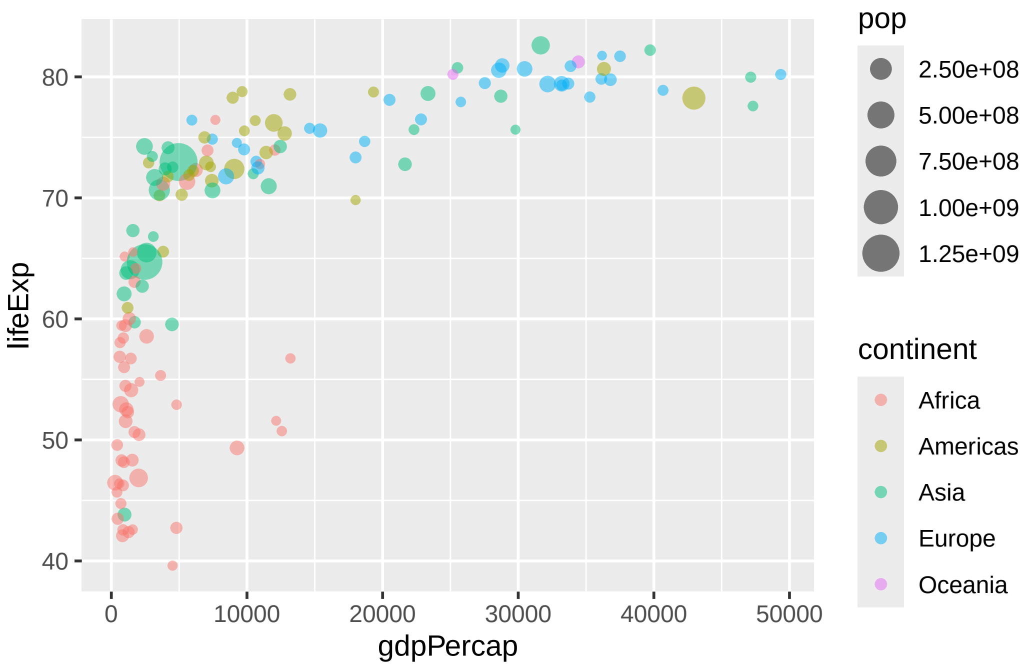
```
library(dplyr)
```

```
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)
```

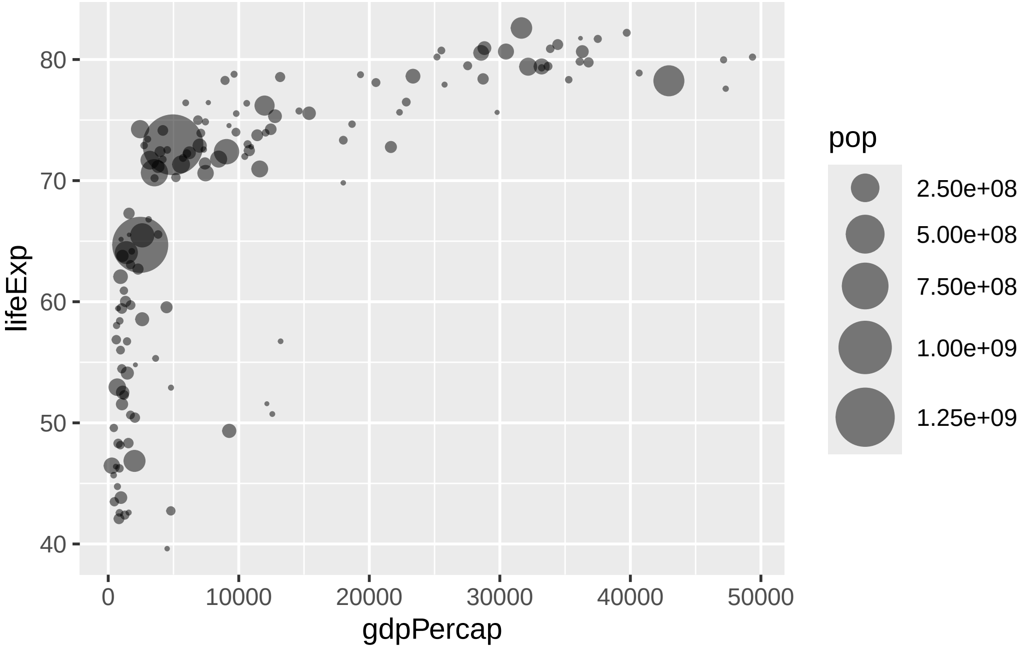
The gapminder_2007 set contains the variable GDP per capita **gdpPercap** and life expectancy **lifeExp** for 142 countries in the year 2007. The first basic scatterplot of this dataset is.

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



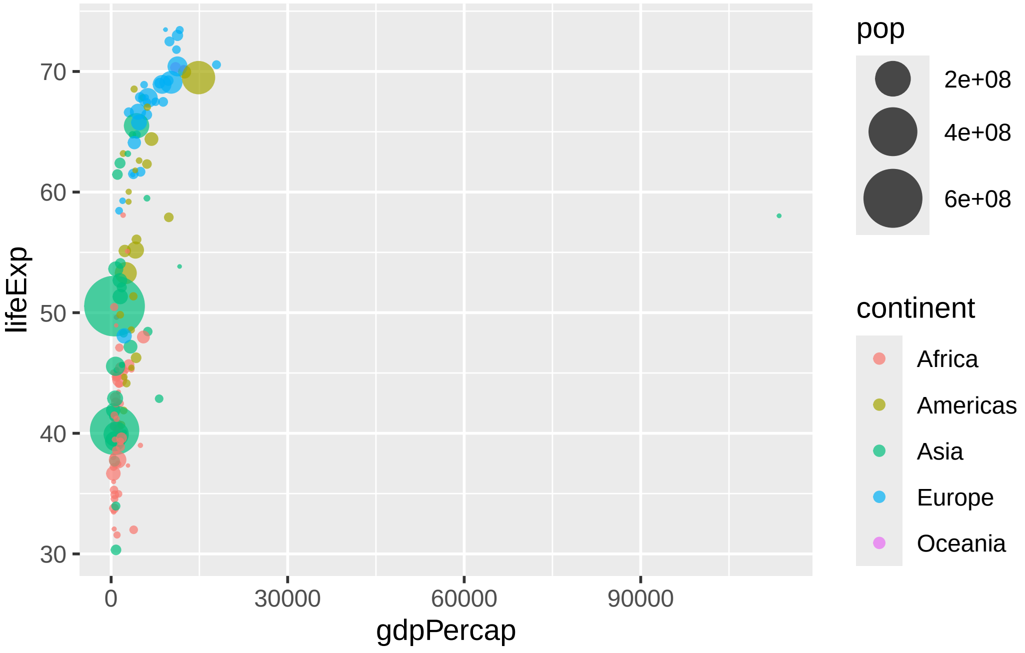
To ensure that the actual population size is reflected by the point size, use the function scale_size_area().

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



The gapminder dataset contains economic and demographic data about various countries since 1952. This is the scatterplot for the year 1957.

```
library(dplyr)
gapminder_1957 <- gapminder %>% filter(year==1957)
ggplot(gapminder_1957) +
  geom_point(aes(x=gdpPercap, y=lifeExp, color=continent, size=pop), alpha=0.7) +
  scale_size_area(max_size=10)
```

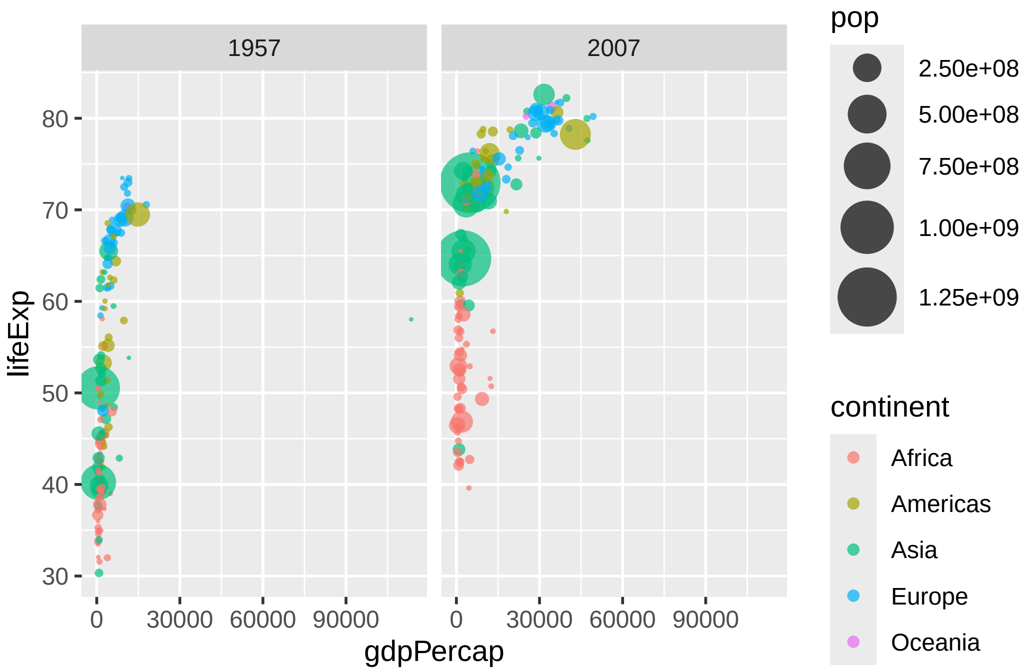


> Q. What do you notice about the 1957 plot, and is it easy to compare with the one for 2007?

The 1957 plot appears to have a direct relationship between gdpPercap and life expectancy. The 2007 plot shows that life expectancy increases as gdpPercap increases until the life expectacy is around 80 where it begins to remain constant regardless of the increase in gdpPercap. Yes, it is easy to compare the two different datasets since they have the same layers included to make their scatterplots.

To compare economic and demographic data about various countries in 1957 and 2007.

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

ggplot(gapminder_1957.2007) +
  geom_point(aes(x = gdpPercap, y = lifeExp, color=continent,
                 size = pop), alpha=0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)
```



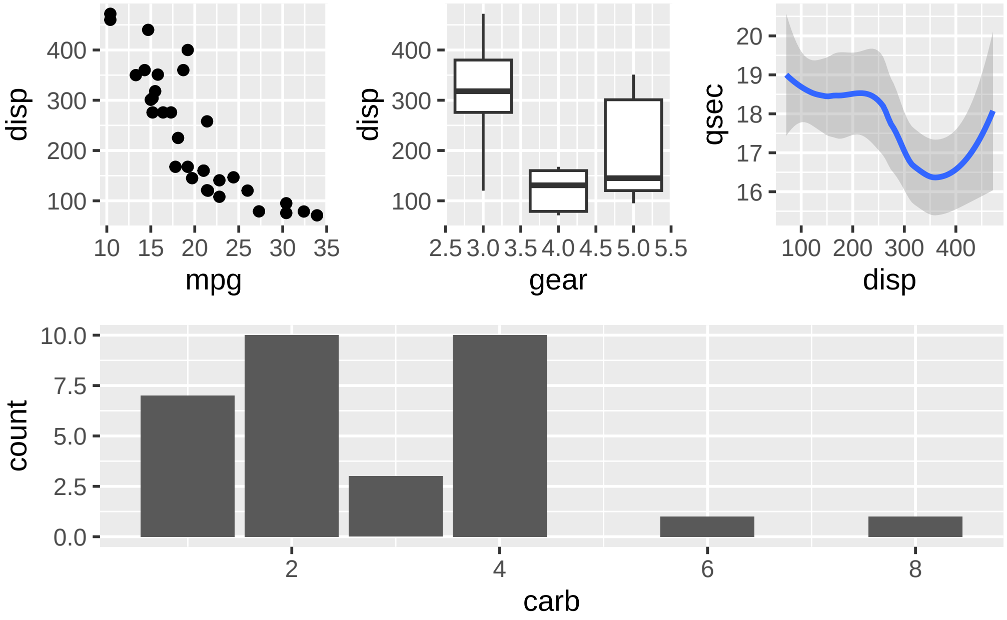
Patchwork Figures

Patchwork is useful for combining plots to make an all-in-one multi-panel figure. An example is shown below.

First, I will run ’install.packages(“patchwork”) in my R console.

```
library(patchwork)
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
400
300
200
100
10
15
20
25
30
35
mpg
disp
400
300
200
100
2.5
3.0
3.5
4.0
4.5
5.0
5.5
gear
qsec
20
19
18
17
16
100
200
300
400
disp
count
10.0
7.5
5.0
2.5
0.0
2
4
6
8
carb