

Class 5: Data Visualization with ggplot

Allen (PID: A16897142)

Table of contents

Background	1
Gene expression plot	6
Custom Color Plot	7
Using Different Geoms	8
More Aesthetic Features!	10

Background

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

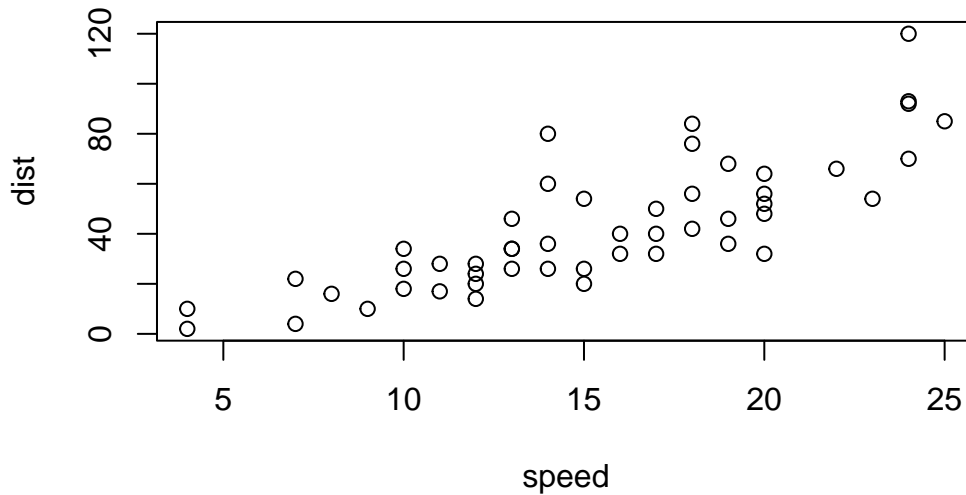
Let’s compare “base” and **ggplot2** briefly. We can use some example data that is built-in with R called **cars**:

```
head(cars)
```

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

I can just call `plot()` in base R.

```
plot(cars)
```



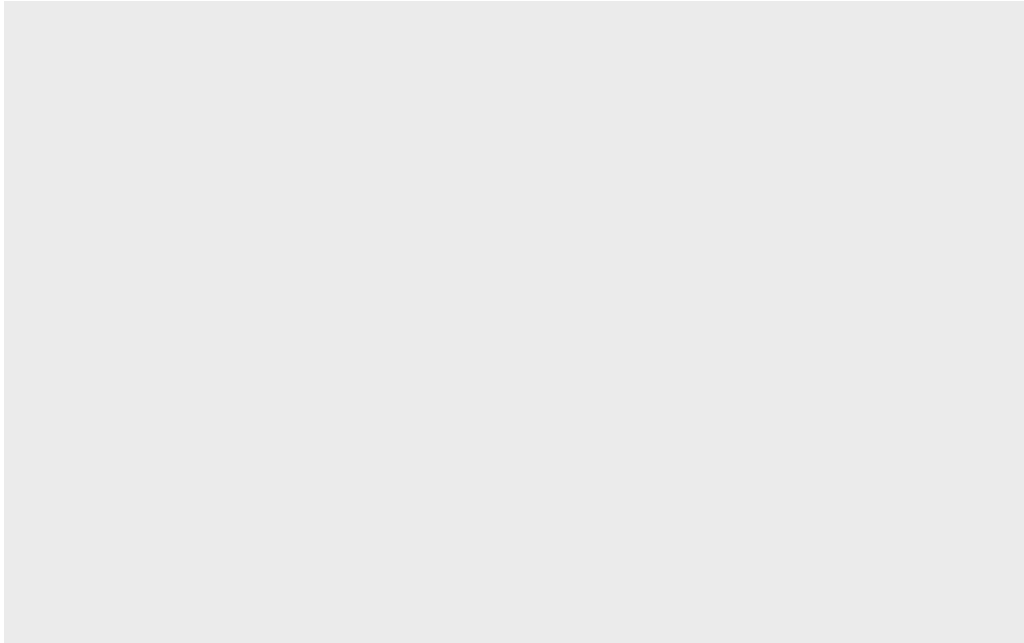
We can now make it pretty and we do this with ggplot! **ggplot2**

First we install ggplot. We do this with the code `install.packages("ggplot2")`. This only need to do this once and then it will be available on my computer any time I need to access ggplot.

Key Point: Packages can only be installed in the R console. Do not do it in quarto docs or R scripts.

To use any add-on packages you have installed, you need to load it up with `library ()`

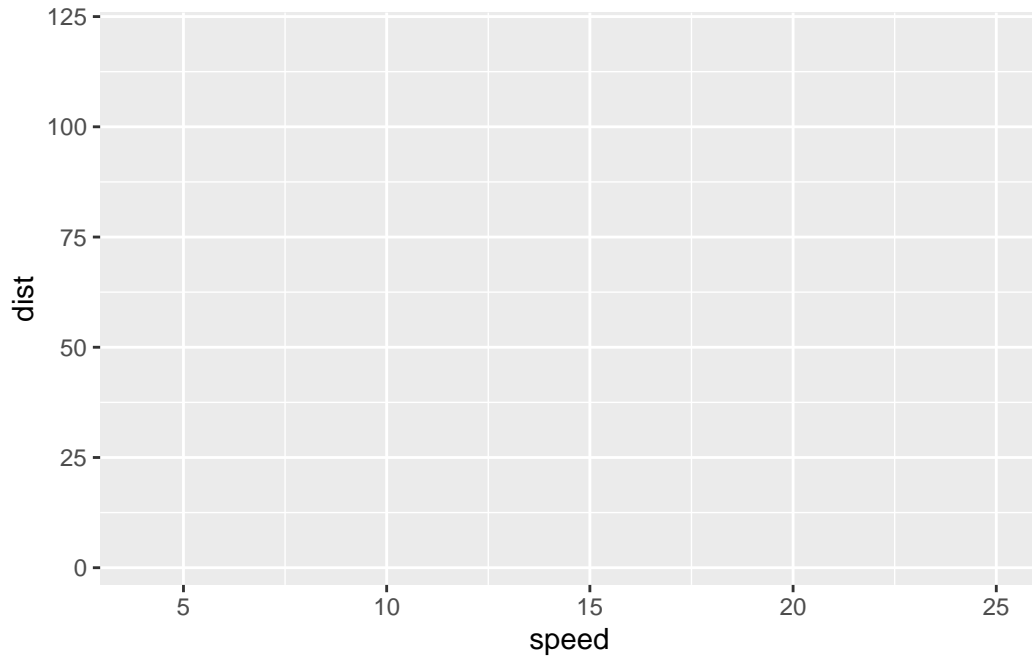
```
library(ggplot2)
ggplot(cars)
```



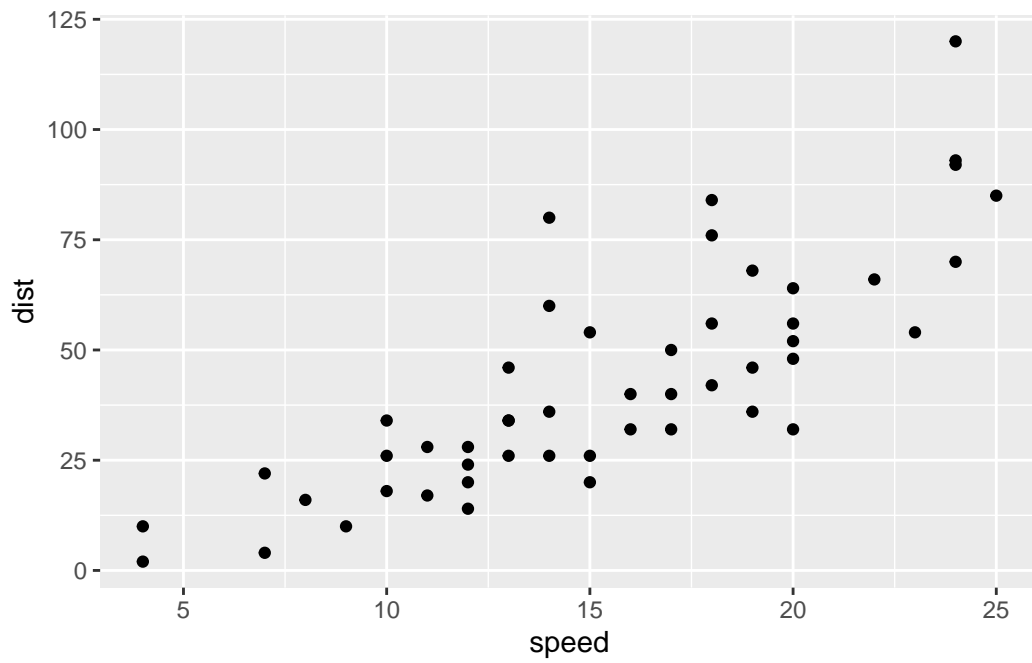
Every ggplot needs three different layers to make it show up:

- the **data** (and in our case it is **cars**)
- the **aesthetics** (how the data looks when mapped on the plot. Such as color and axis titles)
- the **geometry**(what the lines/plots/data is drawn on the plot such as line width)

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



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

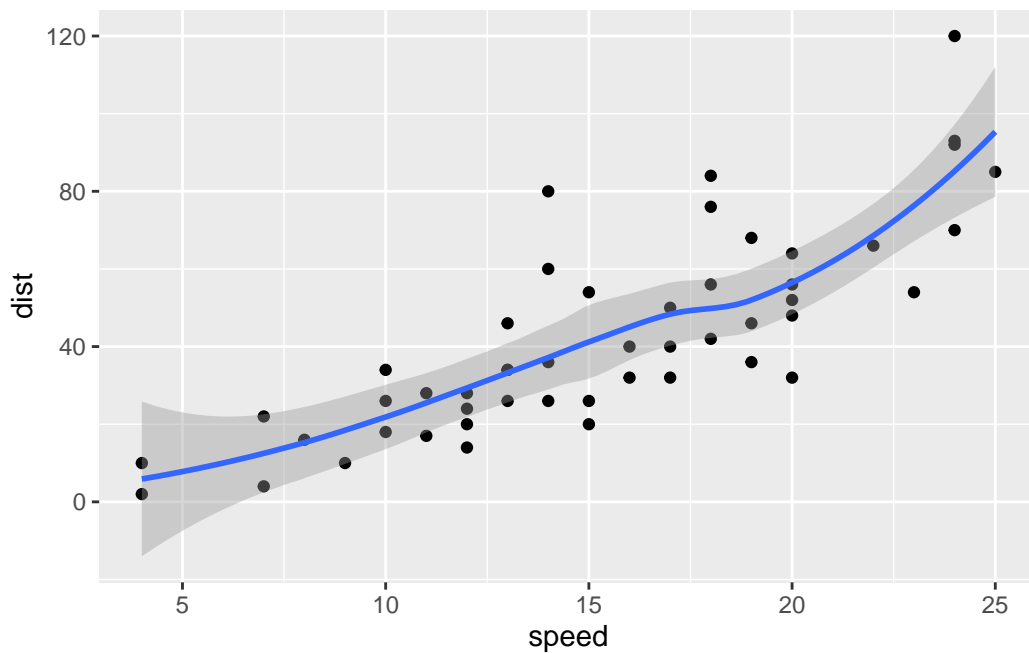


For more “simpler” plots, ggplot requires more words/code than base R. However, the defaults in ggplot are nicer and it is more efficient for complicated plots.

Try adding a line to show the relationship between speed and stopping distance.
Tip: by adding another layer

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

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



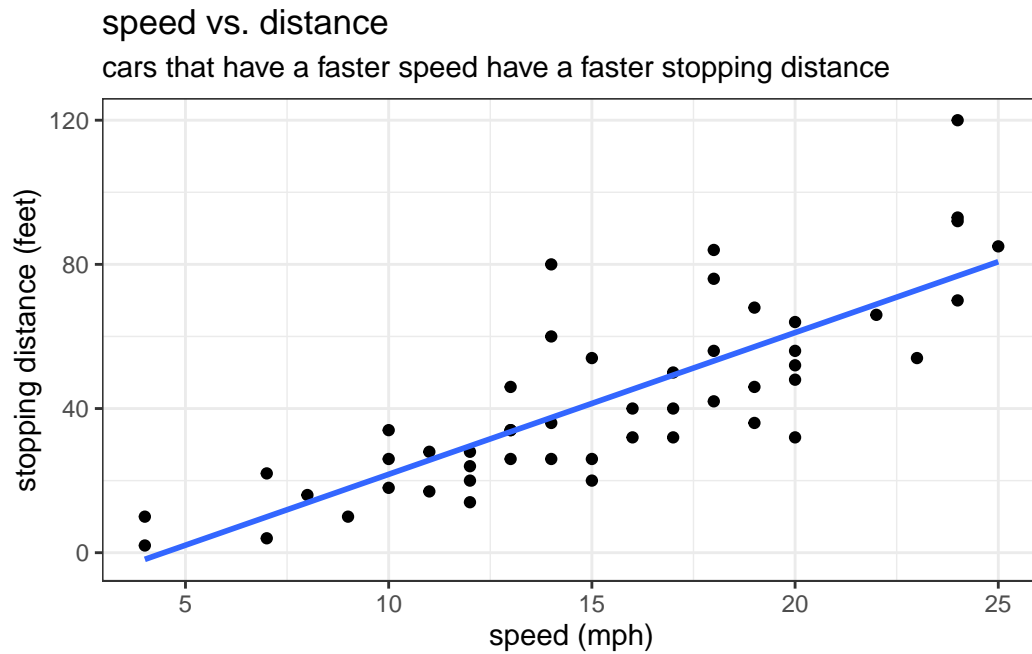
```
p <- ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point()+  
  geom_smooth(se=FALSE,method="lm")
```

I can save an object from ggplot for future use using `x <- x`

Try adding axis titles

```
p + labs(
  title="speed vs. distance",
  subtitle="cars that have a faster speed have a faster stopping distance",
  x="speed (mph)",
  y="stopping distance (feet)"
)+
theme_bw()
```

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



Gene expression plot

Read input data

```
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

```

How many genes are in this dataset? How many columns and what are their names?

```
nrow(genes)
```

```
[1] 5196
```

```
ncol(genes)
```

```
[1] 4
```

```
colnames(genes)
```

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

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

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

```

down unchanging      up
   72      4997     127

```

Custom Color Plot

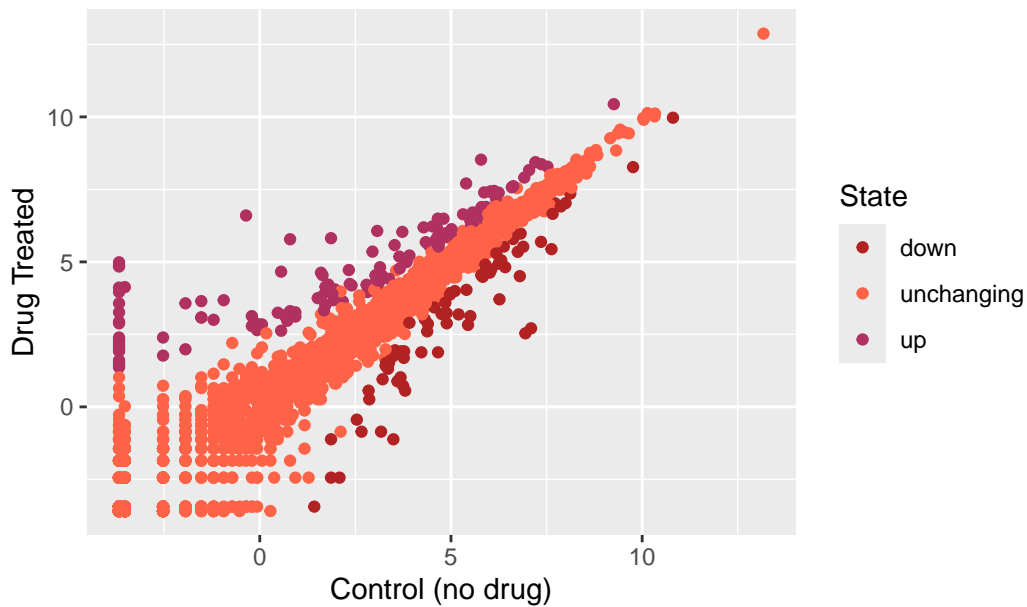
Make a plot with the data

```

ggplot(genes)+
  aes(x=Condition1, y=Condition2, col=State)+
  scale_color_manual(
    values=c("firebrick","tomato","maroon")
  )+
  geom_point()+
  labs(title="Changes of gene expression based on drug treatments", x="Control (no drug)", y="

```

Changes of gene expression based on drug treatments



Using Different Geoms

This will be plotted using the built-in `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

Scatterplot of mpg vs disp

```
p1 <- ggplot(mtcars)+  
  aes(x=disp,y=mpg)+  
  geom_point() +  
  labs(title="Miles per Gallon versus Displacement",  
        x="Displacement (feet)",  
        y="Miles per Gallon")
```


Boxplot of gear vs disp

```
p2 <- ggplot(mtcars)+  
  aes(gear, disp, group=gear)+  
  geom_boxplot()+  
  labs (title="Gear vs Displacement",  
        x="gear",  
        y="displacement(feet)")
```

Barplot of carb

```
p3 <- ggplot(mtcars)+  
  aes(carb)+  
  geom_bar()+  
  labs (title="Carbon Output of Cars",  
        x="carbon output",  
        y="units")
```

Smooth of disp vs qsec

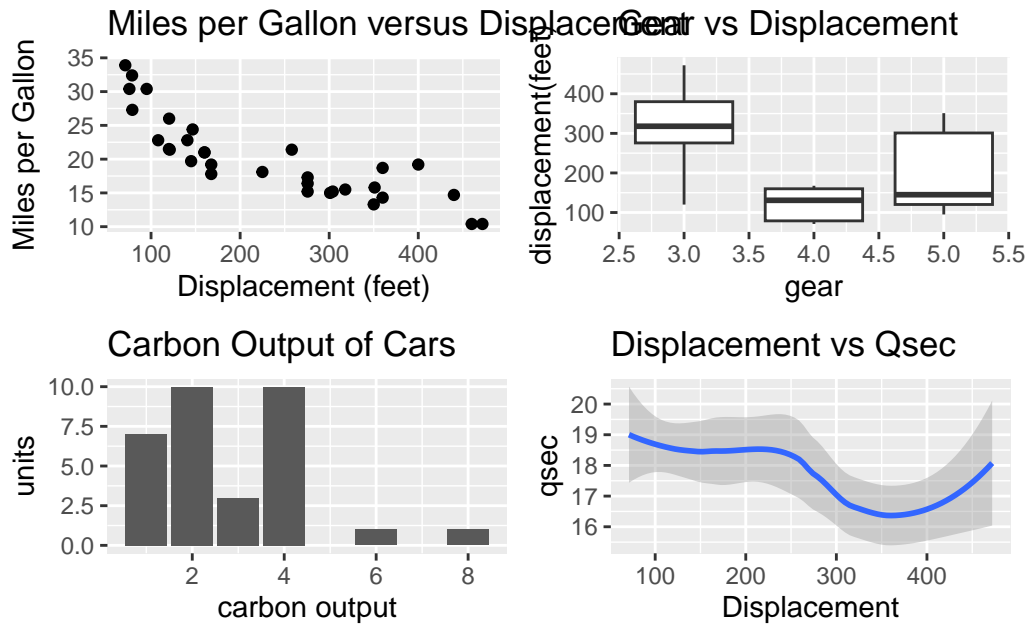
```
p4 <- ggplot(mtcars)+  
  aes(displacement, qsec)+  
  geom_smooth()+  
  labs (title="Displacement vs Qsec",  
        x="Displacement",  
        y="qsec")
```

I want to now combine all of these plots into one figure with a panel!

Use the code `library(patchwork)` to get this.

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

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



```
ggsave(filename="allmyplots.png", width=10, height=10)
```

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

More Aesthetic Features!

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

To just see the table

```
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

How many countries are in this dataset?

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

```
[1] 142
```

Plot GDP vs Life Expectancy and color by continent

```
ggplot(gapminder)+
  aes(x=gdpPercap, y=lifeExp, col=continent)+
  geom_point(alpha=0.3)+
  facet_wrap(~continent)+
  theme_bw()
```

