

# Class 5: Data Visualization

Danika

## Plotting in R

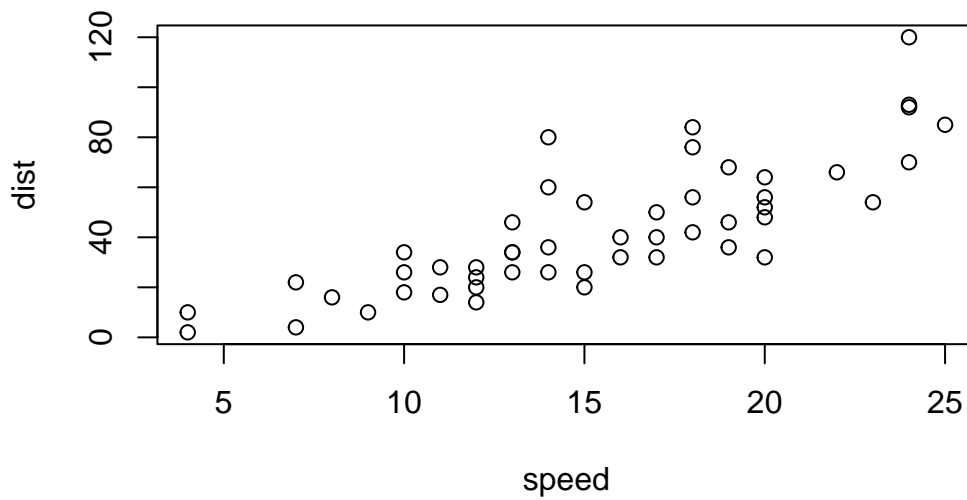
R has multiple plotting and graphic systems. The most popular of which is **ggplot2**.

We have already played with “base” R graphics. This comes along with R “out of the box”.

```
head(cars)
```

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

```
plot(cars)
```



Compared to base R plots ggplot is much more verbos - I need to write more code to get simple plots like the above.

To use ggplot I need to first install the ggplot2 package. To install any package in R I use the 'install.packages()' command along with the package name.

The install is a one time only requirement. The package is now on our computer. I don't need to re-install it.

However, I can't use it without loading it up with a 'library()' call.

```
library(ggplot2)
```

```
ggplot(cars)
```



All ggplot figures need at least 3 things:

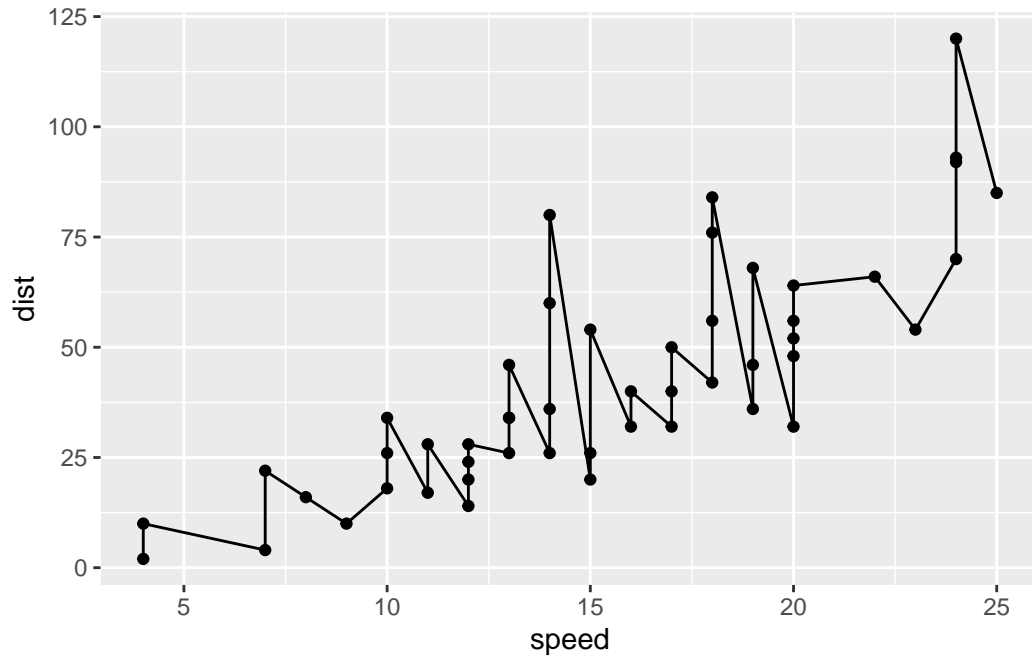
- data (this is the data.frame with our numbers)
- aesthetics (“aes”, how our data maps to the plot)
- geoms (do want lines, points, columns, etc...)

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



I want a trend line to show the relationship between dist and speed.

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

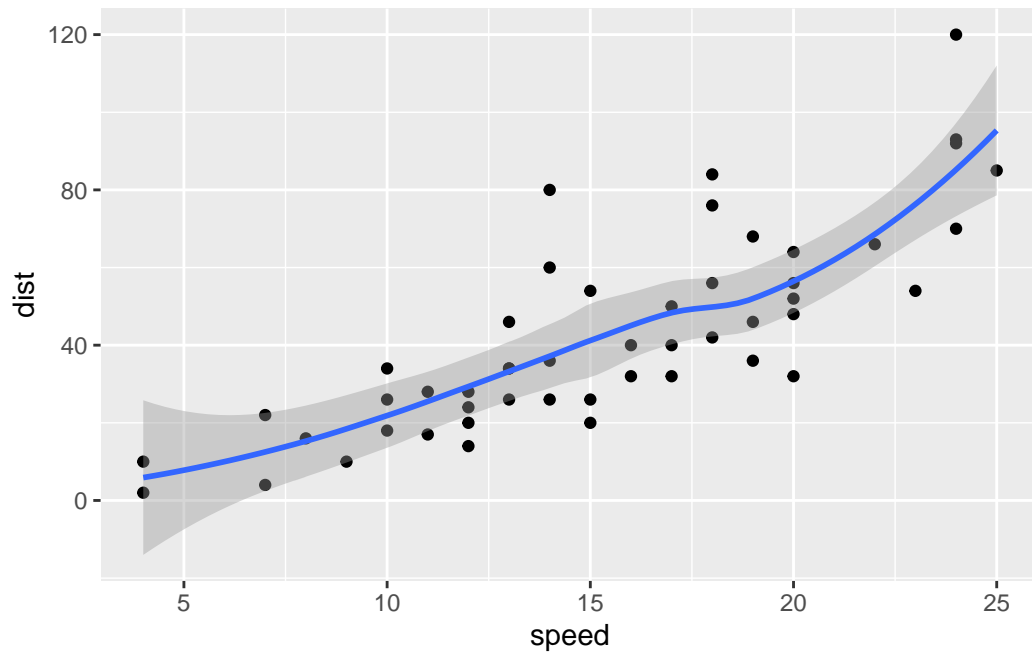


That is not what we want.

```
bb <- ggplot(data=cars)+  
  aes(x=speed, y=dist)+  
  geom_point()
```

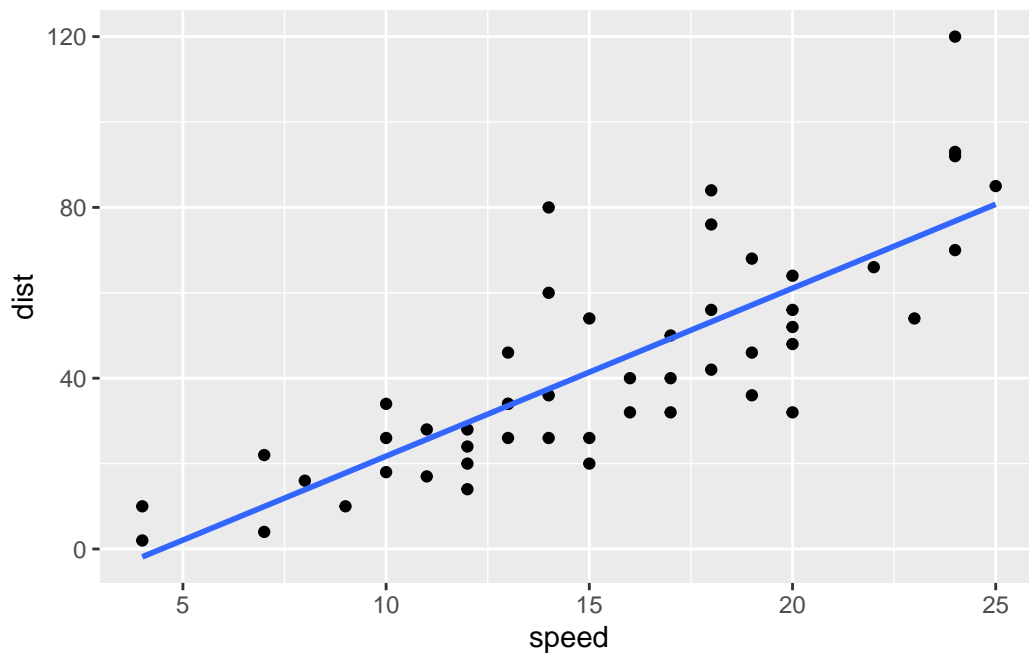
```
bb + geom_smooth()
```

`geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



```
bb + geom_smooth(method="lm", se=FALSE)
```

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



## Gene expression example

Read the dataset from online:

```
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
genes <- read.delim(url)
head(genes, 10)
```

	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
7	ABCA7	3.4484220	3.8266509	unchanging
8	ABCA9-AS1	-3.6808610	-3.5921390	unchanging
9	ABCC11	-3.5288580	-1.8551732	unchanging
10	ABCC3	0.9305738	3.2603040	up

The 'head()' function will print out the first few rows (6 by default).

```
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

```
nrow(genes)
```

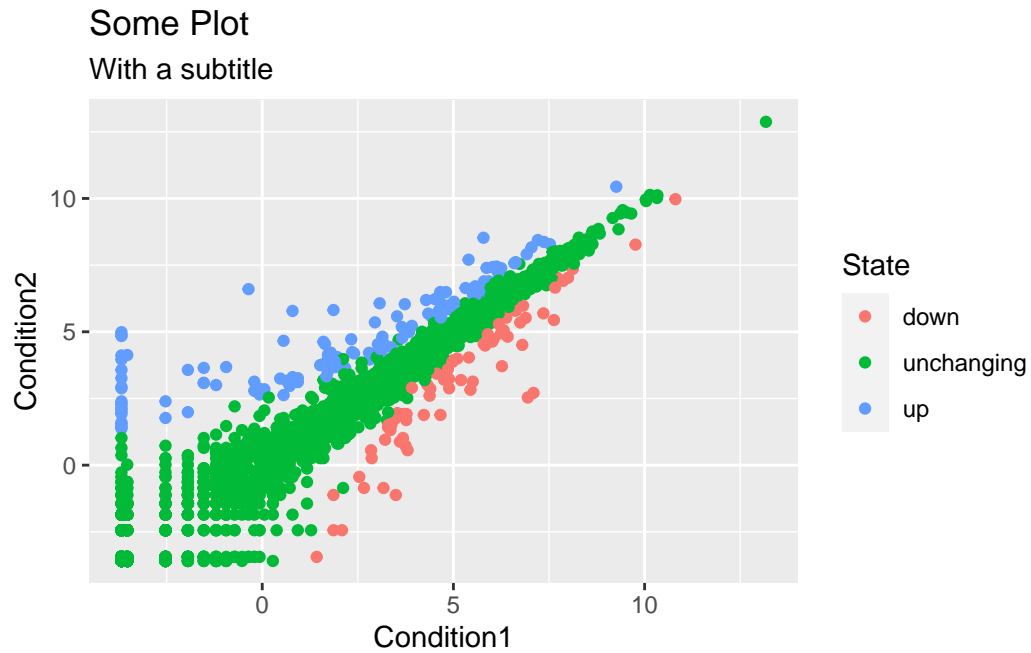
```
[1] 5196
```

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

down	unchanging	up
72	4997	127

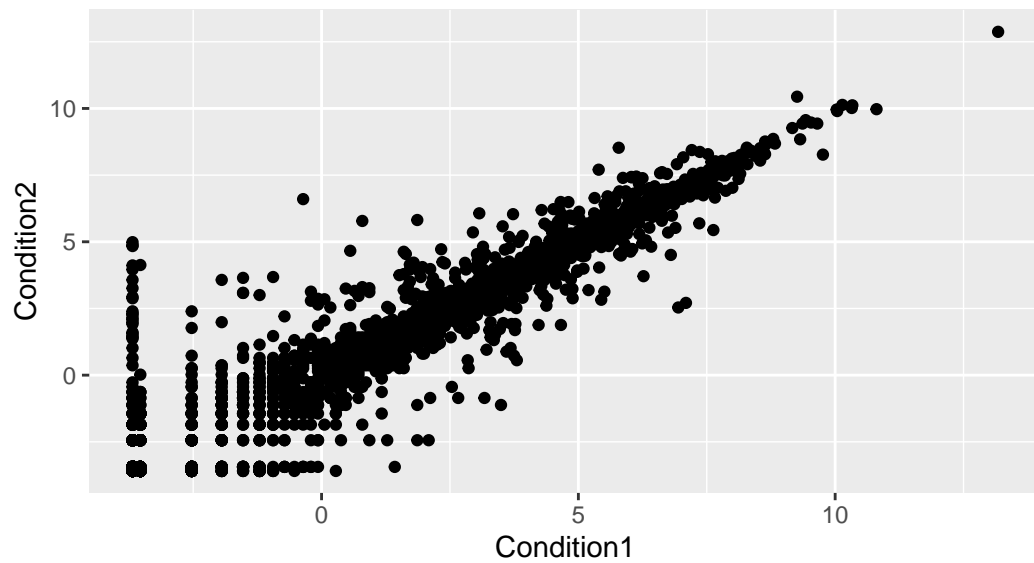
```
ggplot(genes)+  
  aes(Condition1, Condition2, color=State)+  
  geom_point()+  
  labs(title="Some Plot",  
        subtitle="With a subtitle")
```



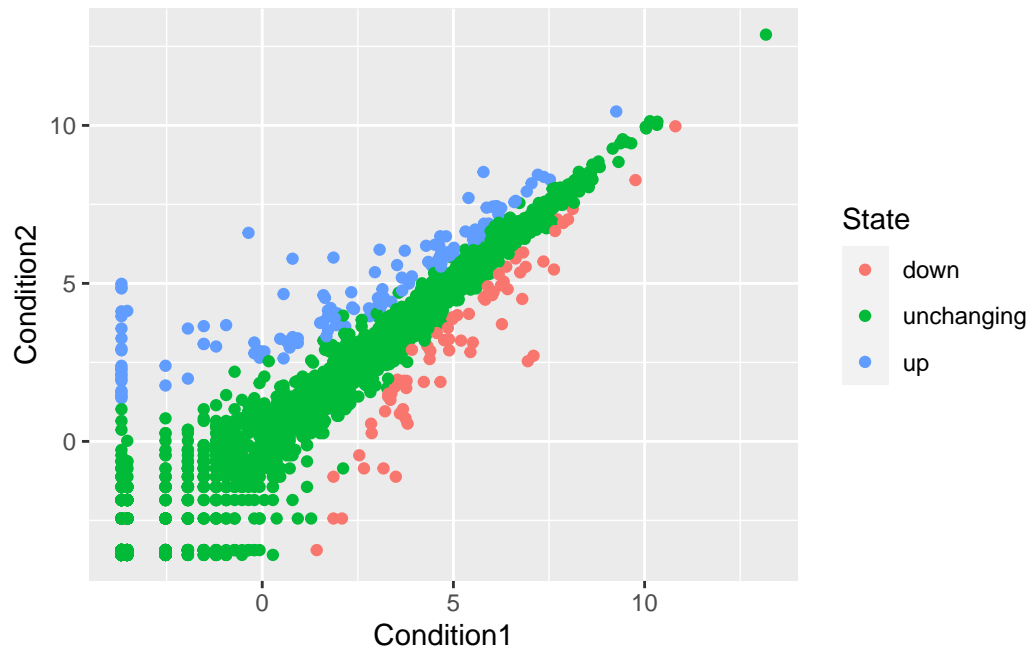


```
ggplot(genes)+  
  aes(Condition1, Condition2)+  
  geom_point()+  
  labs(title="Some Plot",  
        subtitle="With a subtitle")
```

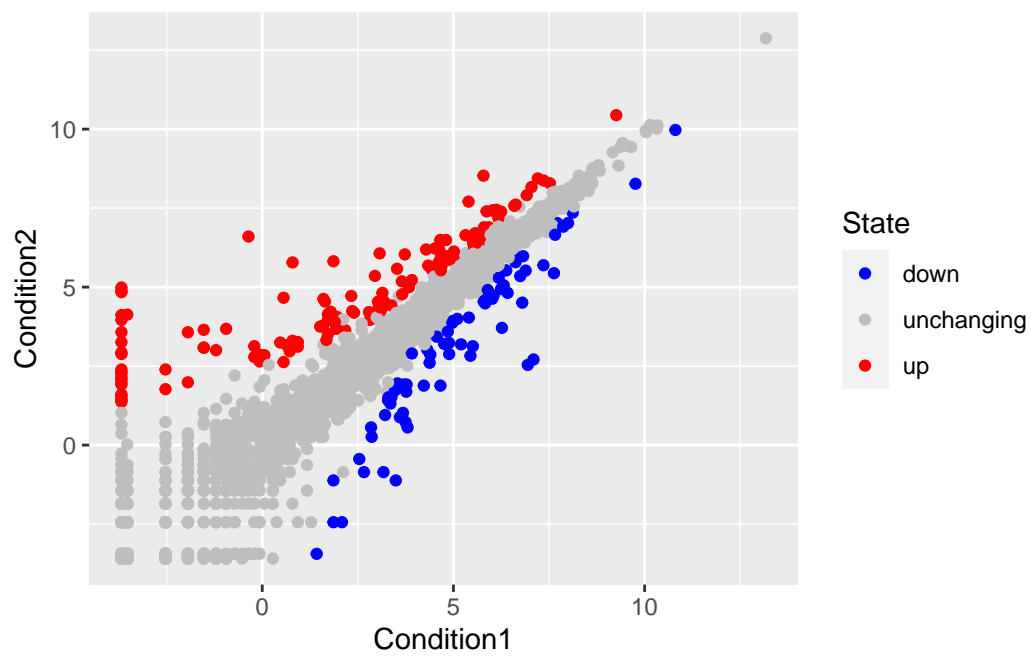
Some Plot  
With a subtitle



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



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



I write some text I want **bold** or *italic*