

# Class 5: Data Viz with ggplot

Yiyu

## Graphs and plots in R

R has tons of different graphics systems. These include “**base R**” (e.g. the `plot()` function) and add on packages like **ggplot2**.

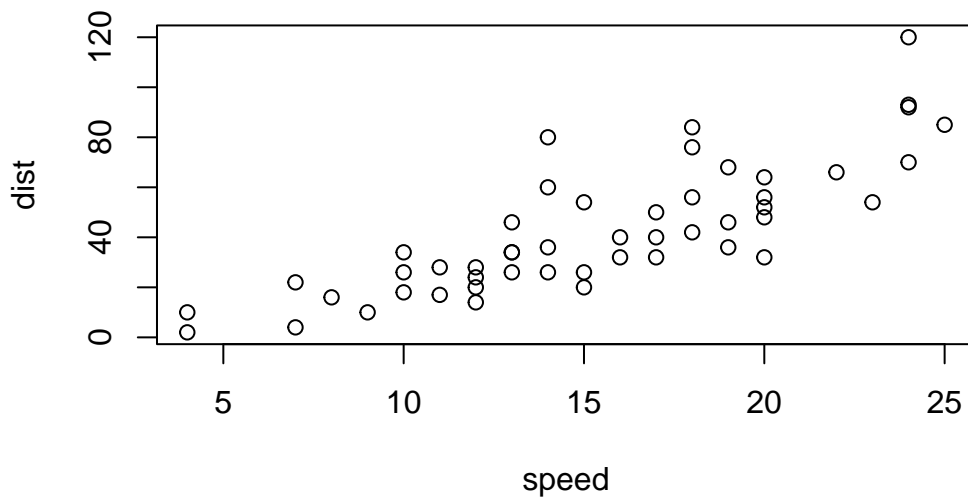
Let’s start with plotting a simple data set in “base R” and then ggplot2 to see how they differ.

```
head(cars)
```

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

To plot this in “base R”, I just use `plot()`

```
plot(cars)
```

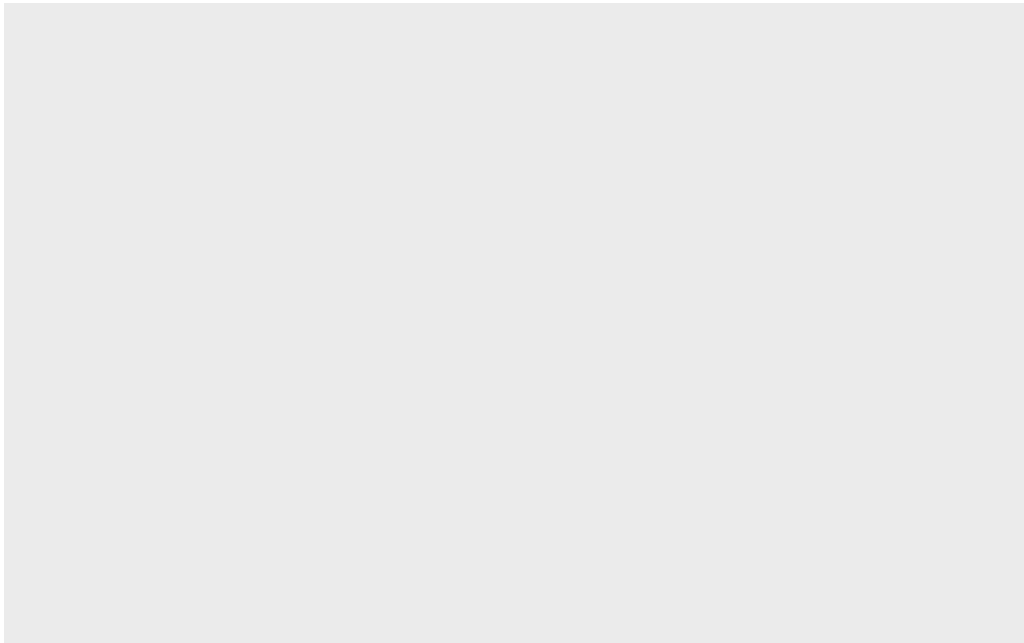


First to use ggplot2, I need to install the package. For this I use the `install.packages()` function

I don't want to run `install.packages()` in my quarto document as this would re-install the package every time I render the document.

The main function in the ggplot2 package is `ggplot()`, before I can use this function I need to load the package with a `library()` call.

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

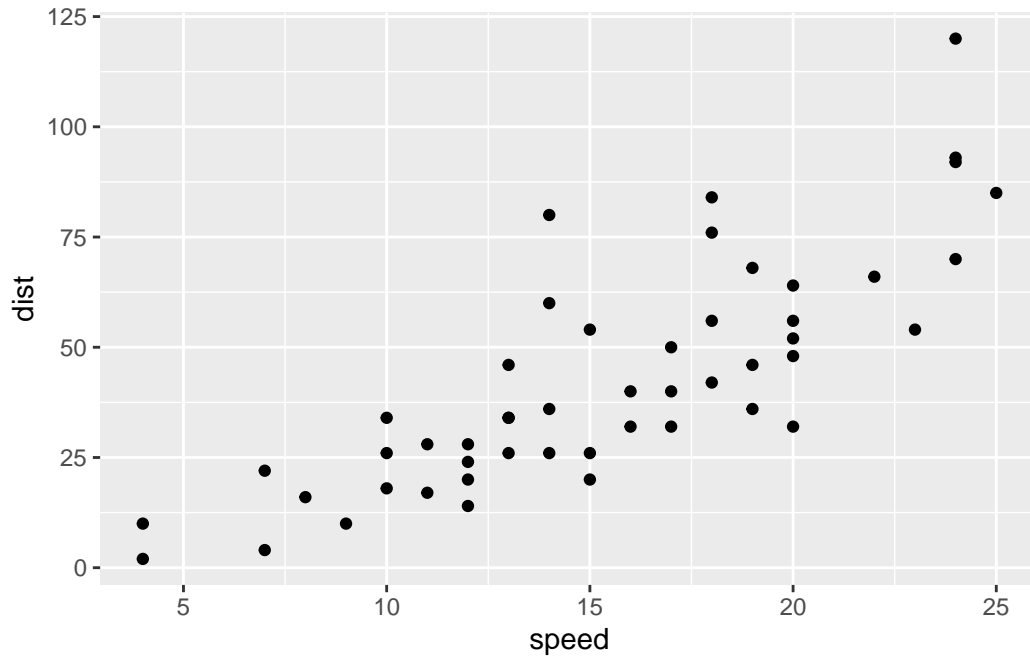


Blank because it's missing layers

There are at least 3 things that every ggplot needs:

- the **data** (the data I want to plot)
- the **aesthetics** (how the data maps to my plot)
- the **geoms** or geometries (the type of plot)

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



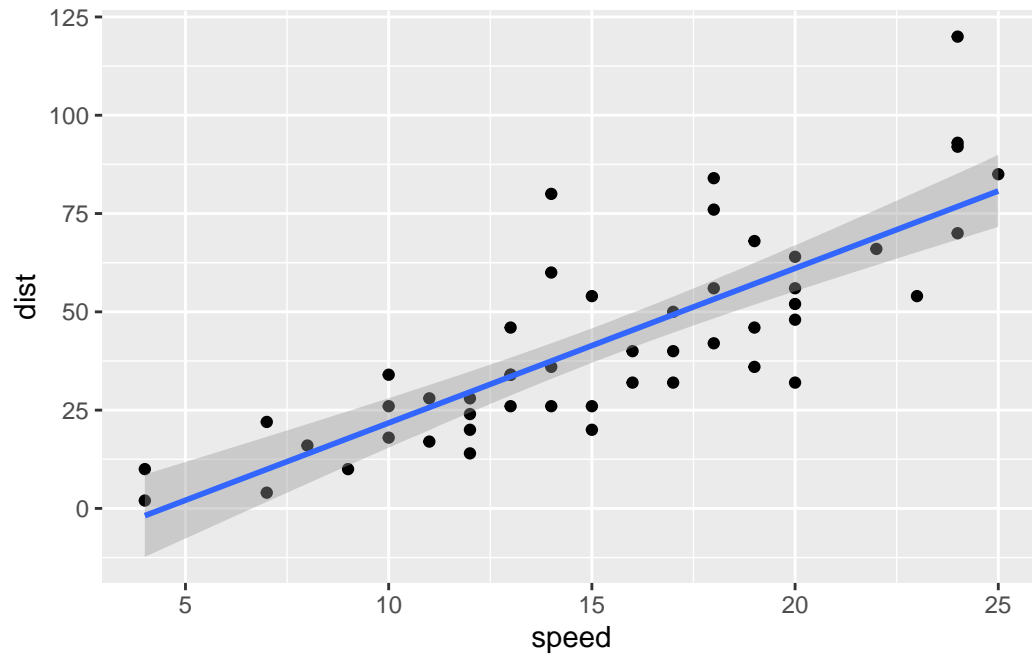
Can add even more layers

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

can save it to p, then just add on to p

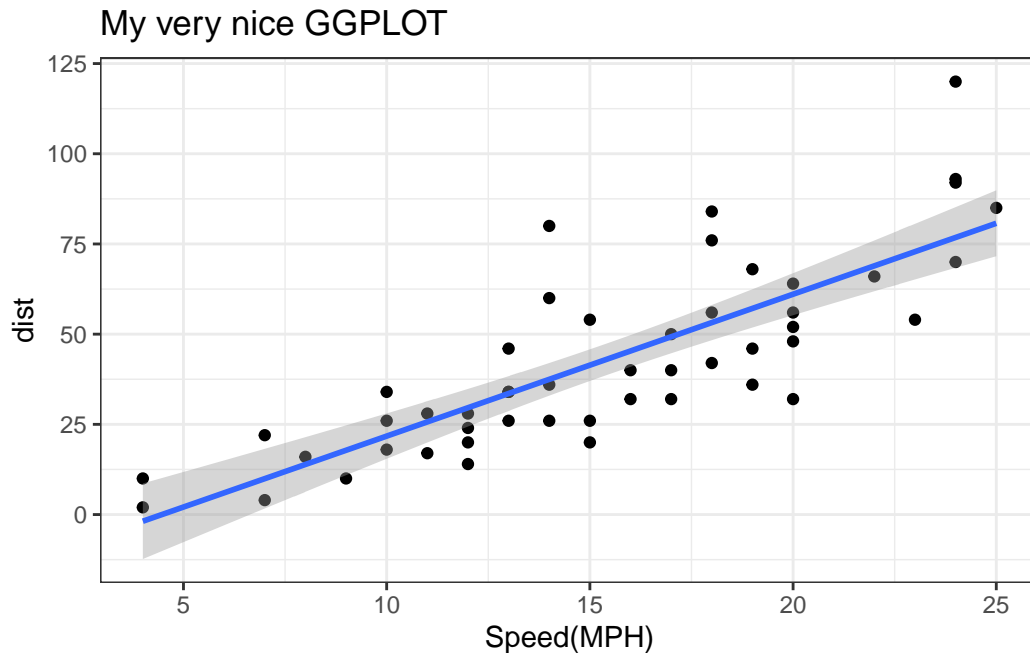
```
p
```

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



```
p + labs(title="My very nice GGPlot", x= "Speed(MPH)") +  
  theme_bw()
```

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



##RNSeq plot with more aes() values

```
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?

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?

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?

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

```
nrow(genes)
```

```
[1] 5196
```

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

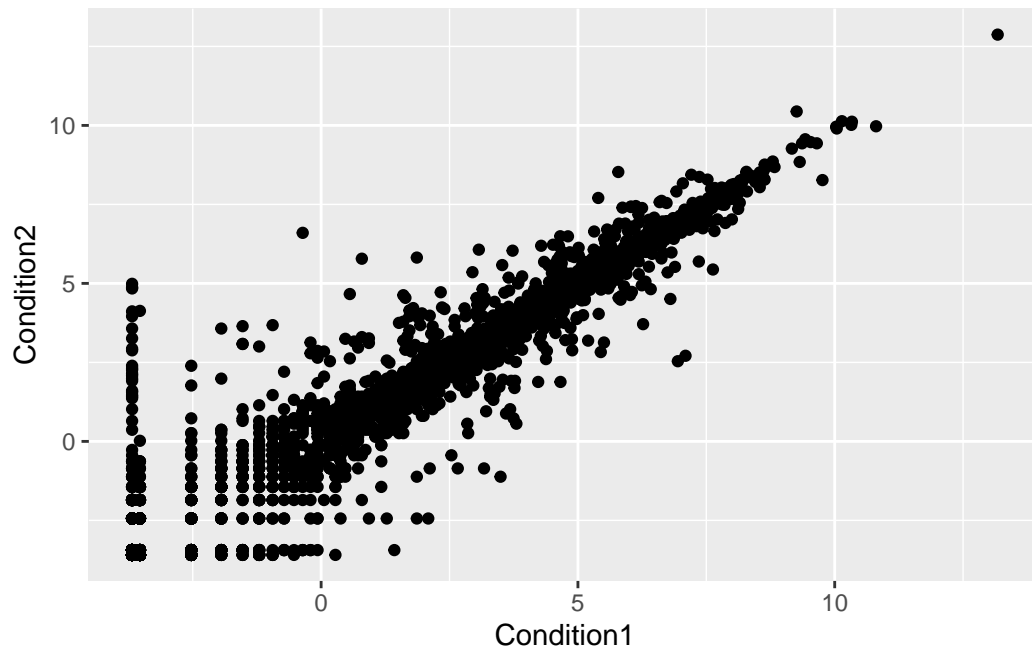
```
table(genes$State)
```

down	unchanging	up
72	4997	127

```
table(genes$State) / nrow(genes) * 100
```

down	unchanging	up
1.385681	96.170131	2.444188

```
ggplot(genes) +aes(x=Condition1, y=Condition2)+geom_point()
```

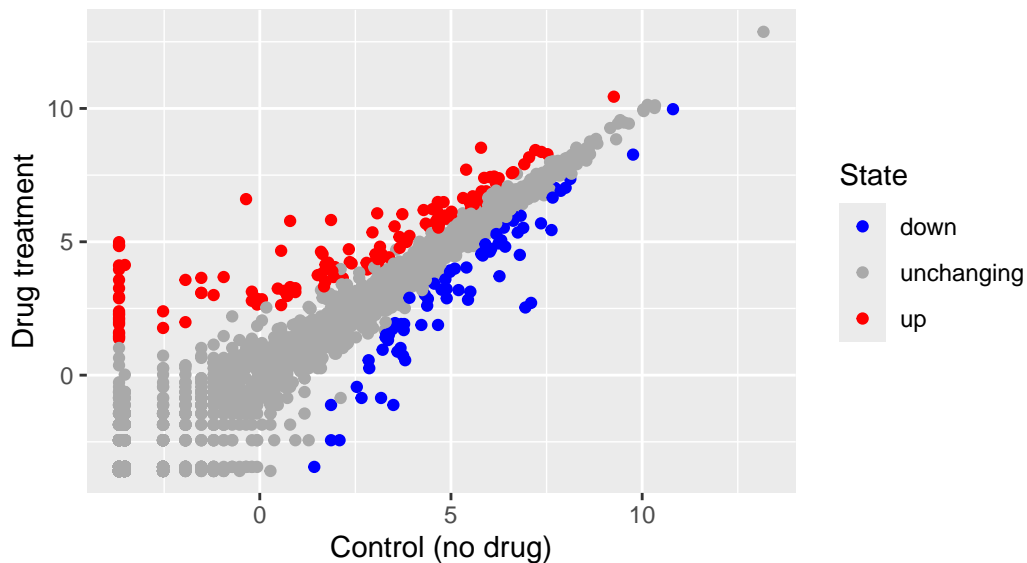


```
ggplot(genes) + aes(x=Condition1, y=Condition2, col=State)+ geom_point() +  
  scale_color_manual(values=c("blue","darkgray","red")) +  
  labs(title= "Expression changes upon drug treatment", subtitle= "RNASeq Data",  
        x= "Control (no drug)", y= "Drug treatment")
```



## Expression changes upon drug treatment

### RNASeq Data



##Gapminder dataset plots

We can get exposure to setting more `aes()` parameters with datasets that include more columns of useful data. For example the **gapminder** dataset on GDP and life expectancy for different countries over time.

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

Have a look at the first 6 rows

```
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 rows does this dataset have?

```
nrow(gapminder)
```

```
[1] 1704
```

How many unique contries are there in this dataset?

```
length(unique(gapminder$country))
```

```
[1] 142
```

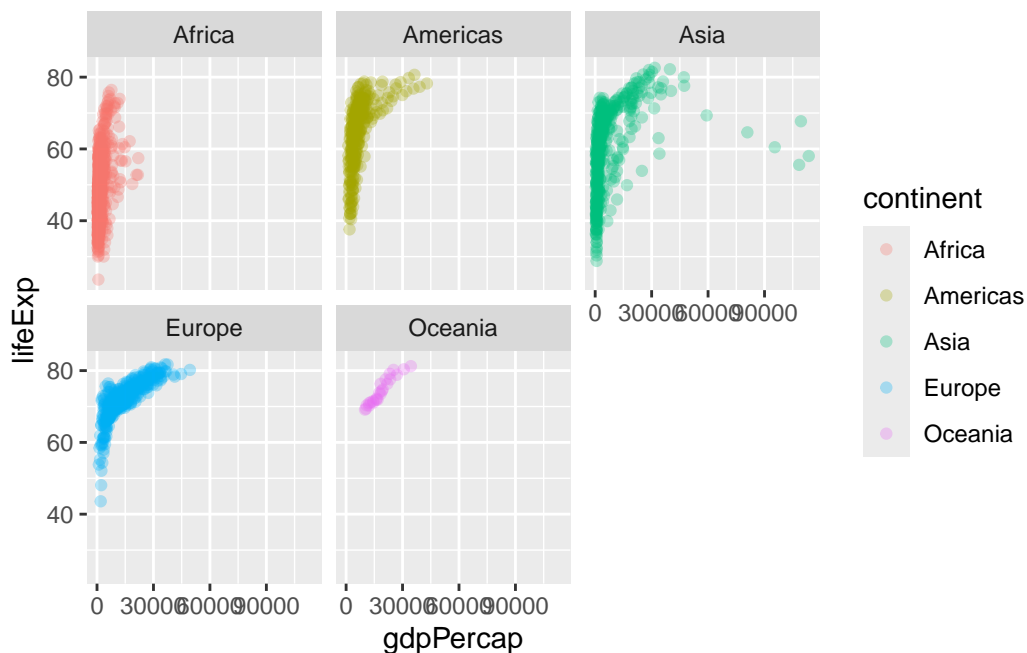
What years does this dataset cover?

```
unique(gapminder$year)
```

```
[1] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 2002 2007
```

Key functions that will be useful in our R journey include: - `nrow()` - `ncol()` - `length()` - `unique()` - `table()`

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



```
library(patchwork)
# Setup some example plots
p1 <- ggplot(mtcars) + geom_point(aes(mpg, disp))
p2 <- ggplot(mtcars) + geom_boxplot(aes(gear, disp, group = gear))
p3 <- ggplot(mtcars) + geom_smooth(aes(dis, qsec))
p4 <- ggplot(mtcars) + geom_bar(aes(carb))

# Use patchwork to combine them here:
(p1 | p2 | p3) /
  (p4 | p)
```

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

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

