

Lab5:DataViz with ggplot

Matthew Chan (A18130675)

Table of contents

Background	1
Gene Expression Plot	4
Going further with gapmider	6
First look at the dplyr package	8

Background

There are lots of ways to make plots in R. These include so-called “base R” (like the `plot()`) and add on packages like **ggplot2**

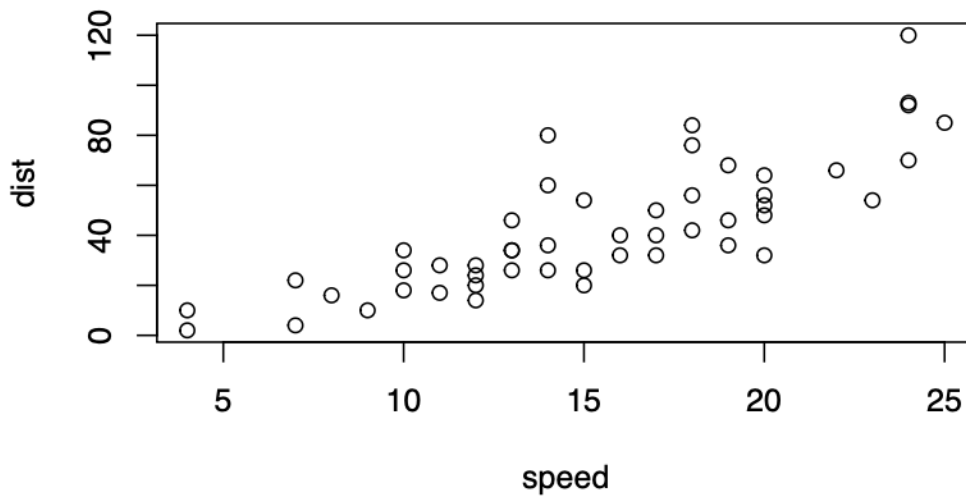
Let’s make the same plot with these two graphics system. We can use the inbuilt `cars` dataset:

```
head(cars)
```

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

With “base R” we can simply:

```
plot(cars)
```

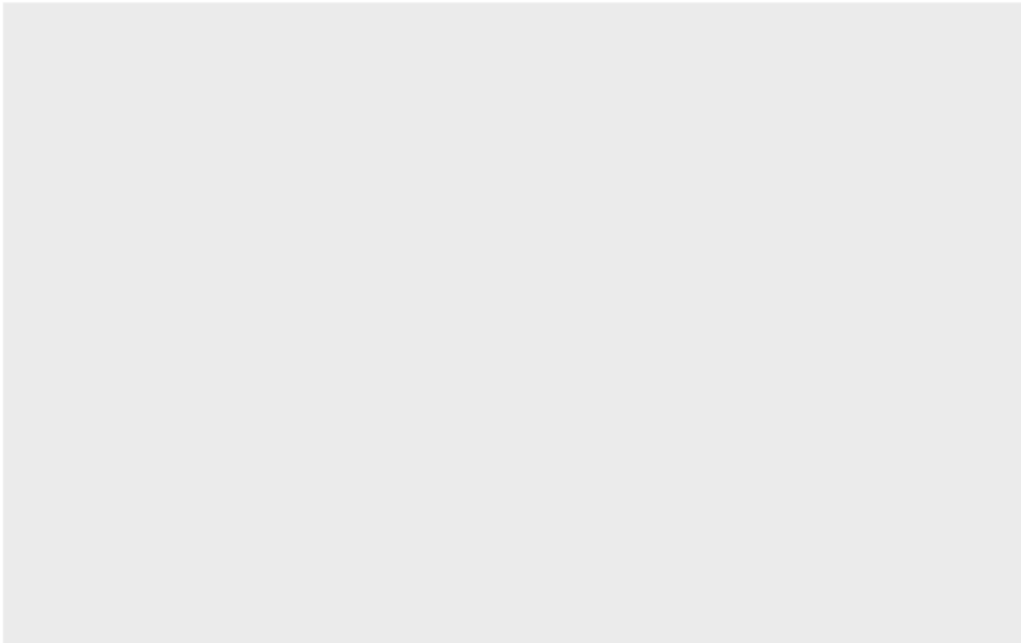


Now lets try ggplot. First ggplot needs to be installed with `'install.packages("ggplot2")'`.

N.B. we never run an `install.packages()` in a code chunk otherwise we will reinstall needlessly everytime the document is rendered

Everytime we want to use an addon package we must load it with a call to `library()`

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

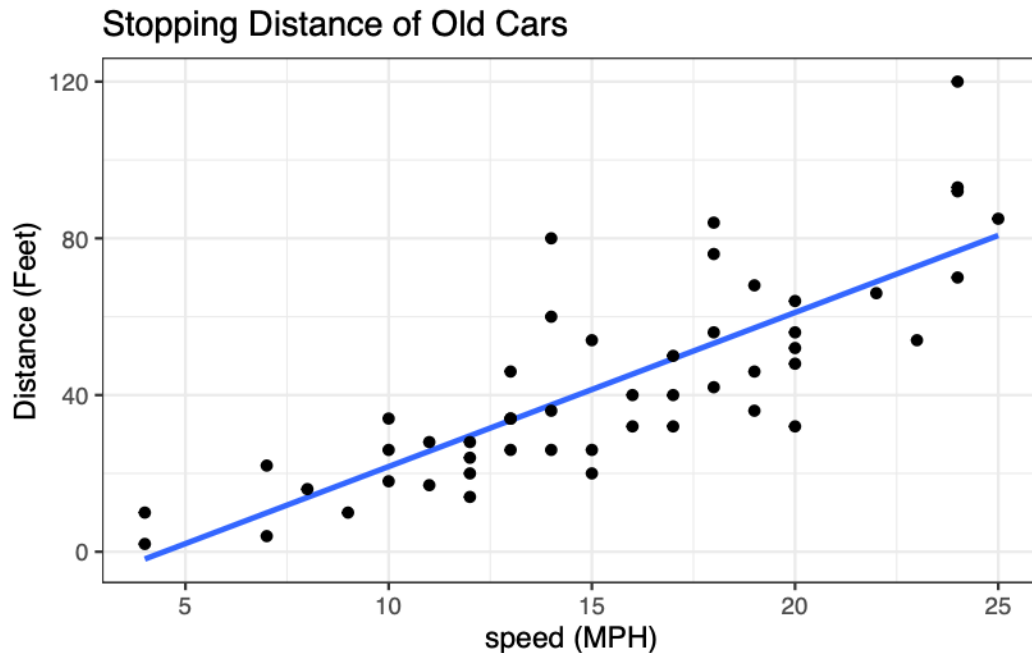


Every ggplot needs at least 3 things:

1. The **data** i.e. stuff to plot as a data.frame
2. The **aes** or aesthetics that map the ddata to the plot
3. The **geom_** or geometry i.e. the plot type such as points, lines, etc

```
ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_smooth(method = "lm", se = FALSE) +  
  geom_point() +  
  labs(x="speed (MPH)",  
        y="Distance (Feet)",  
        title="Stopping Distance of Old Cars") +  
  theme_bw()
```

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



Gene Expression Plot

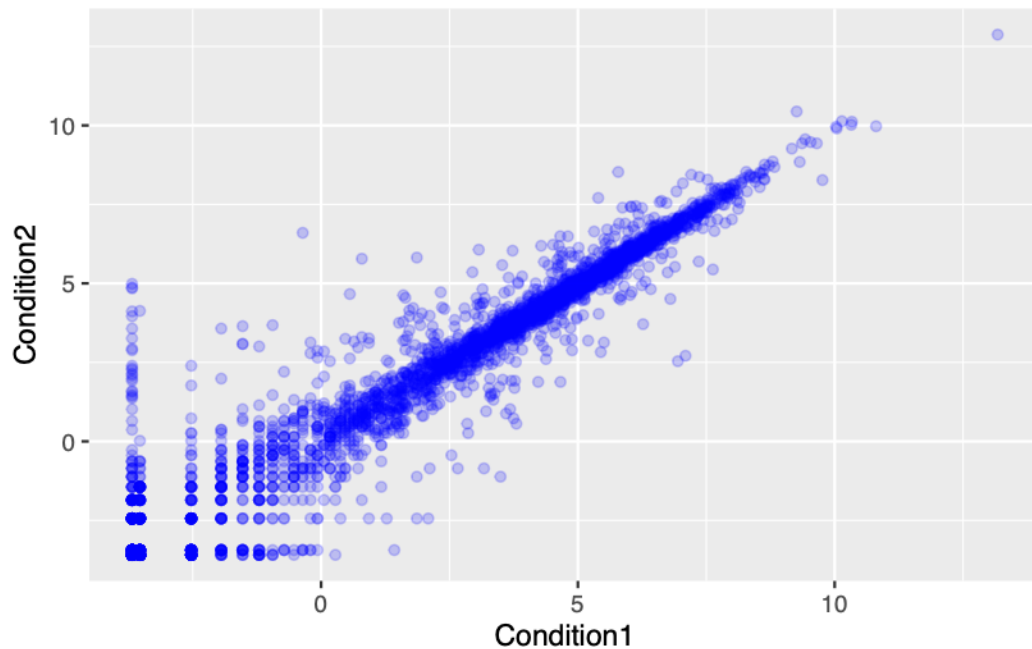
Read some data in the effects of GLP-1 inhibitor (drug) on gene expression values:

```
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

Version 1 plot - start simple by getting some ink on the page.

```
ggplot(genes) +
  aes(Condition1, Condition2) +
  geom_point(col = "blue", alpha = 0.2)
```



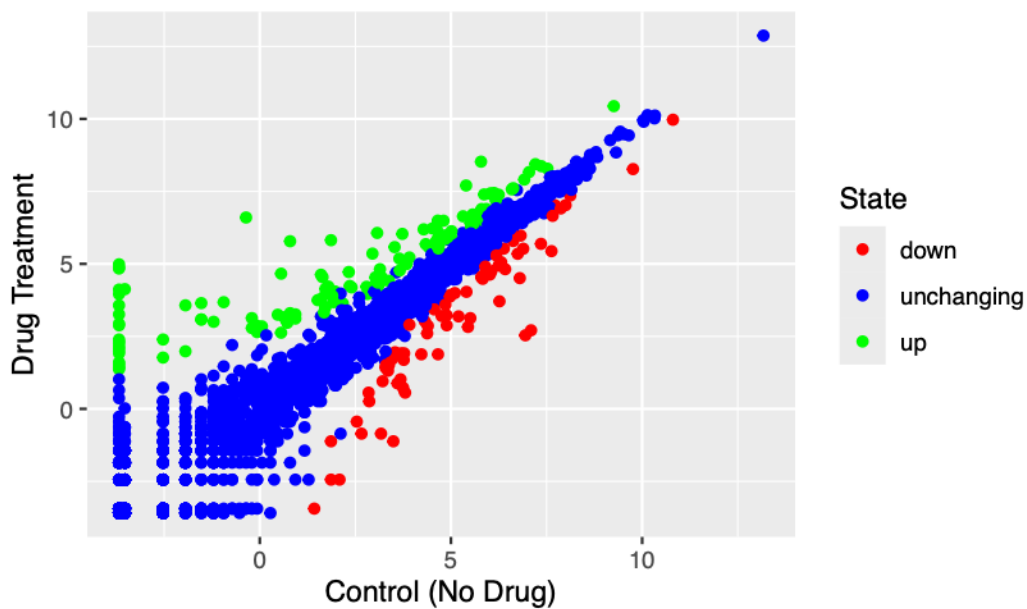
Lets color by `state` up, down or no change.

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

down	unchanging	up
72	4997	127

```
ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State) +
  geom_point() +
  scale_color_manual(values = c( "red", "blue", "green")) +
  labs(x="Control (No Drug)",
       y= "Drug Treatment",
       title = "Gene Expression Changes Upon Drug Treatment")
```

Gene Expression Changes Upon Drug Treatment



Going further with gapminder

Here we explore the famous `gapminder` dataset with some custom plots.

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

How many rows does this dataset have?

```
nrow(gapminder)
```

[1] 1704

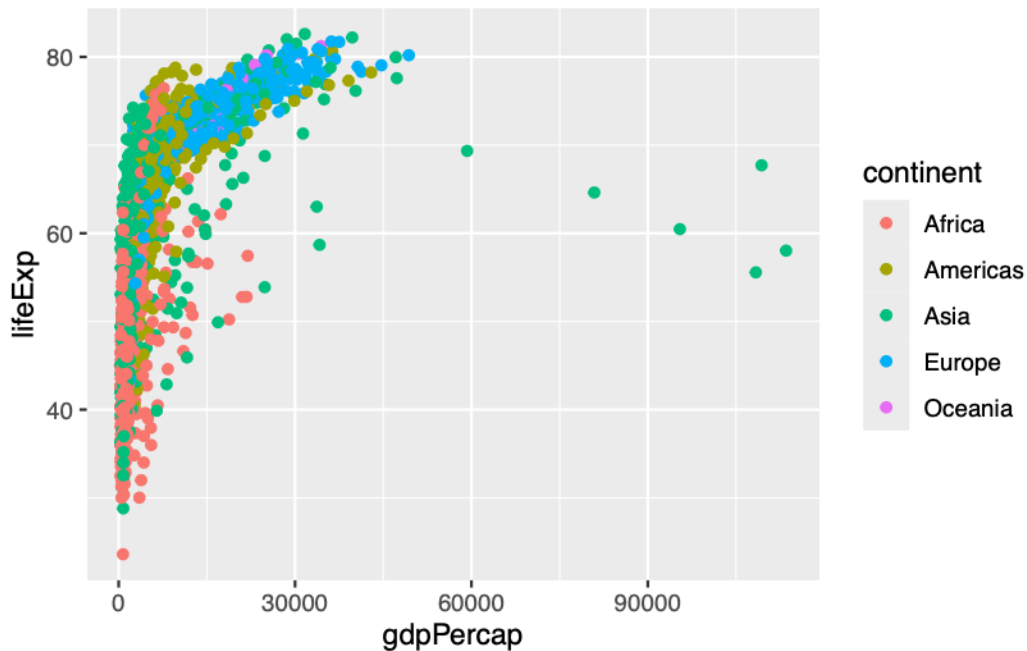
How many different continents are in this dataset?

```
table(gapminder$continent)
```

Africa	Americas	Asia	Europe	Oceania
624	300	396	360	24

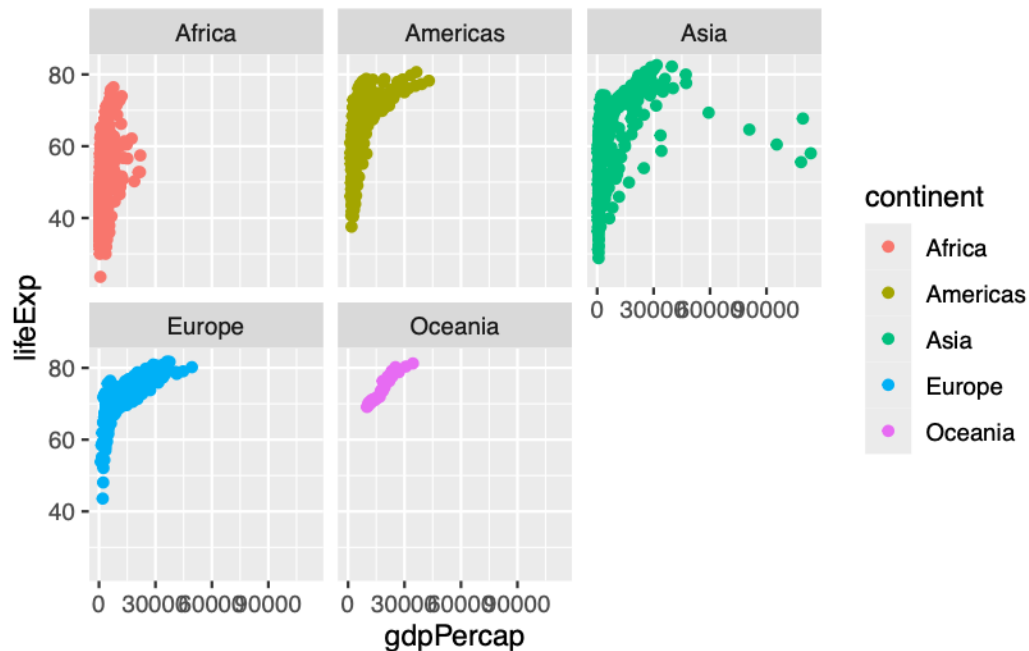
Version 1 plot gdpPercap vs LifeExp for all rows

```
ggplot(gapminder) +  
  aes(gdpPercap, lifeExp, col = continent)+  
  geom_point()
```



I want to see a plot for each continent - in ggplot lingo this is called “faceting”

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



First look at the dplyr package

Another add-on package with a function called `filter()` that we want to use install the dplyr code with “`install.packages("dplyr")`” call it with “`library(dplyr)`”

```
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`

```
filter(gapminder, year == 2007, country == "Ireland")
```


	country	continent	year	lifeExp	pop	gdpPercap
1	Ireland	Europe	2007	78.885	4109086	40676

```
input <- filter (gapminder, year == 2007 | year == 1977)
```

```
ggplot(input) +
  aes(gdpPercap, lifeExp, col = continent)+
  geom_point() +
  facet_wrap(~year)
```

