

Class 05: Data Visualization with GGPLOT

Diego Diaz (PID: A17328629)

Table of contents

Background	1
Gene Expression Plot	5
Going Further	8
Custom Plots	10

Background

There are lots of ways to make figures in R. These include so-called “base R” graphics (e.g. `plot()`) and tons of add-on packages like `ggplot2`.

For example here we make the same plot with both:

```
head(cars)
```

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

```
plot(cars)
```



First I need to install the package with the command `install.packages()`.

N.B. We never run an install in a quarto code chunk or we will end up re-installing packages many, many times - which is not what we want!

Everytime we want to use of these “add-on” packages we need to load it up in R with the `library()` function:

```
library(ggplot2)
```

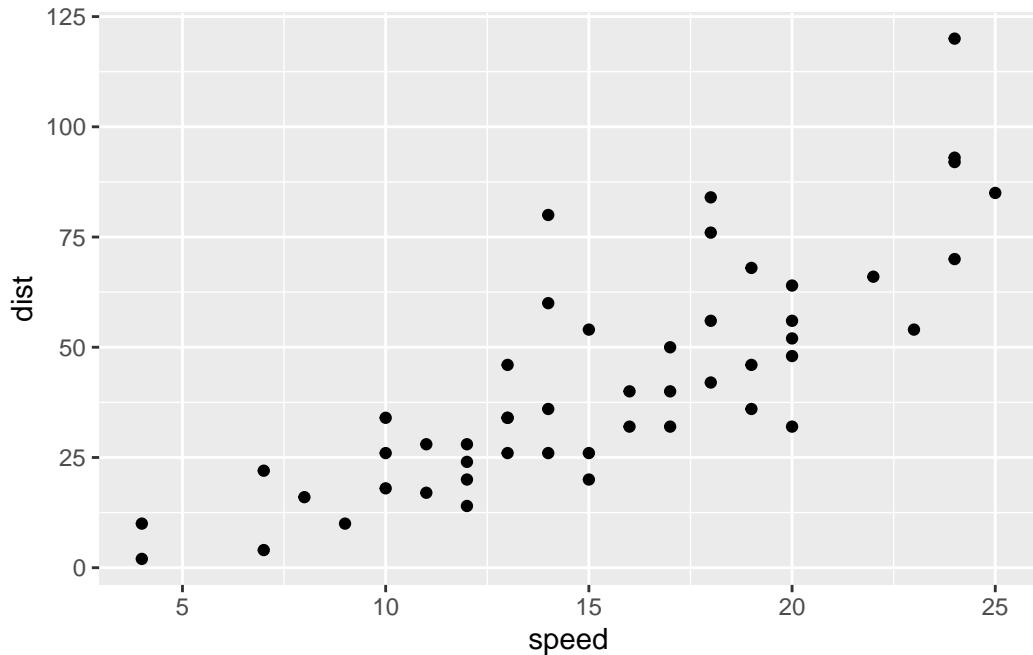
```
ggplot(cars)
```



Every ggplot needs at least 3 things:

- The **data**, the stuff you want plotted.
- The **aesthetics**, how data is mapped to the plot.
- The **geometry**, the type of plot.

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



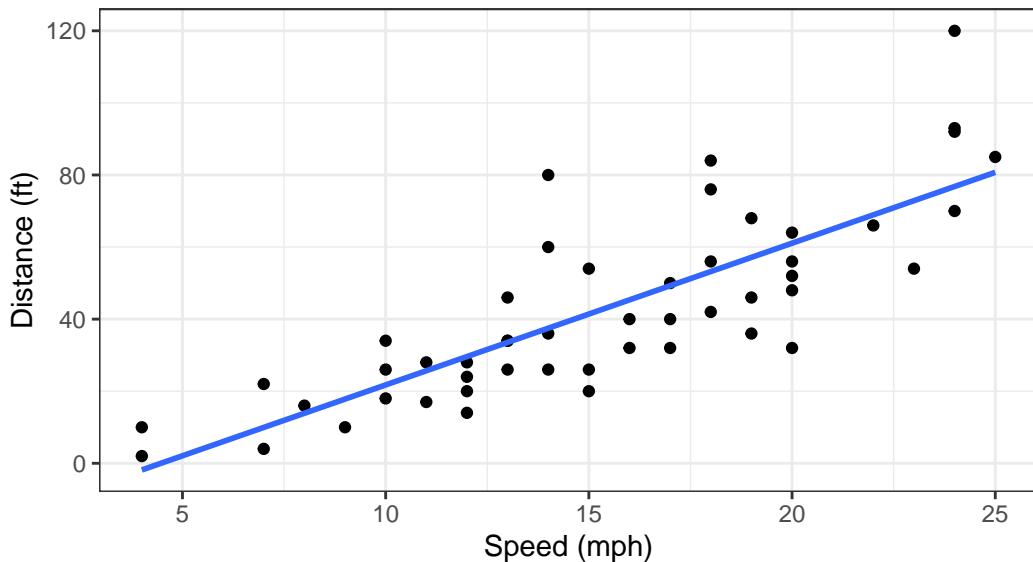
```
p <- ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method='lm', se=FALSE) +
  labs(title="Stopping Distance of Old Cars",
       subtitle = "Data from the `cars` object",
       x = "Speed (mph)",
       y = "Distance (ft)")
```

render it out

```
p + theme_bw()
`geom_smooth()` using formula = 'y ~ x'
```

Stopping Distance of Old Cars

Data from the `cars` object



Gene Expression Plot

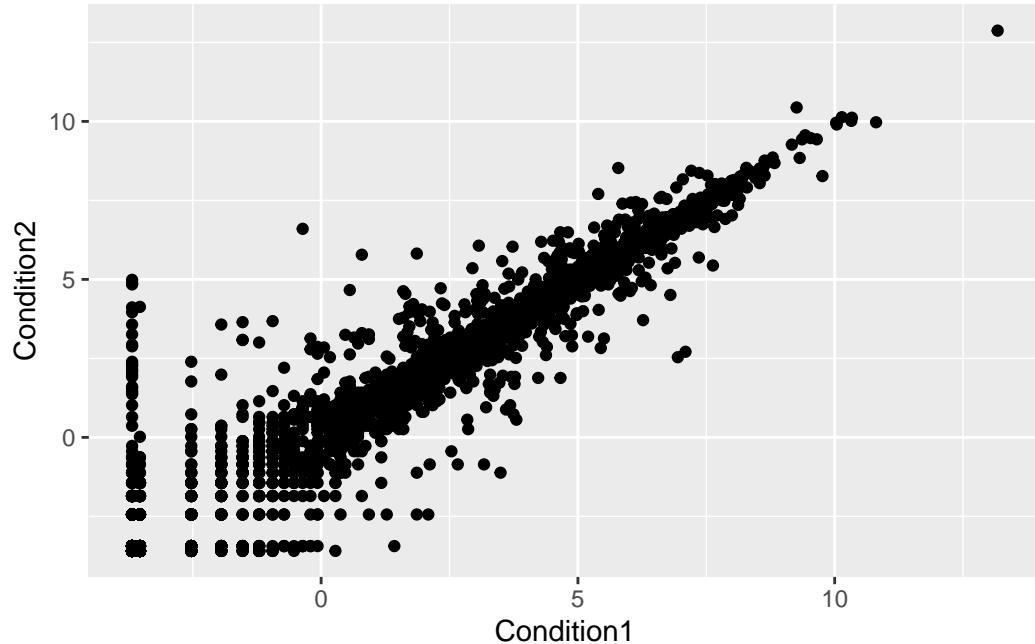
We can read the input data from the class website.

```
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

A first version plot:

```
ggplot(genes) +
  aes(Condition1, Condition2) +
  geom_point()
```

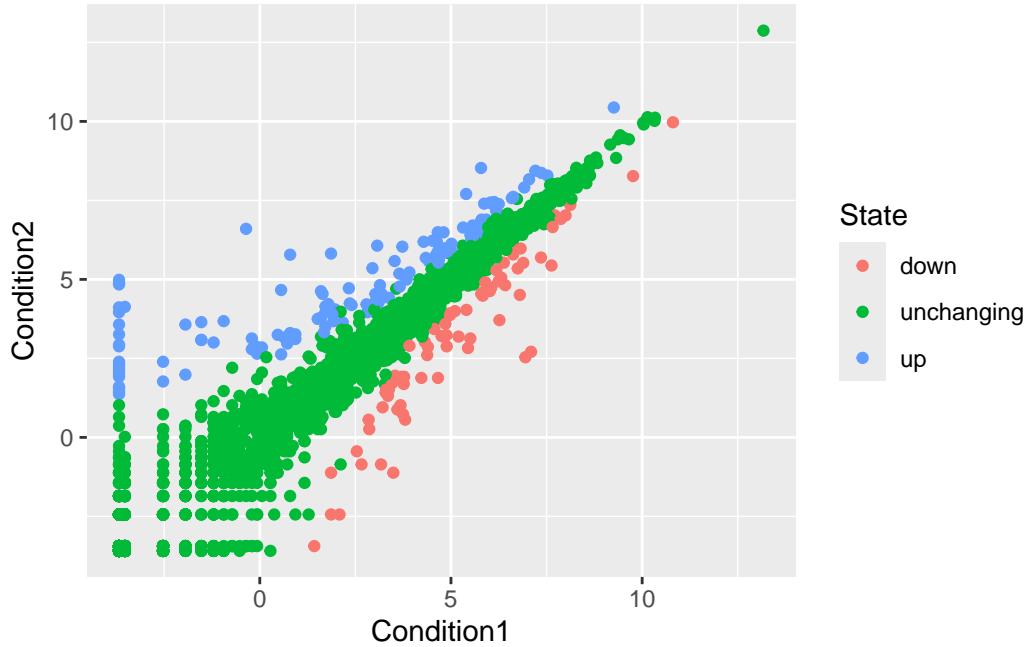


```
table(genes$State)
```

down	unchanging	up
72	4997	127

Version 2: Let's color by `State`, so we can compare the "up" and "down" significant genes compared to all the "unchanging" genes.

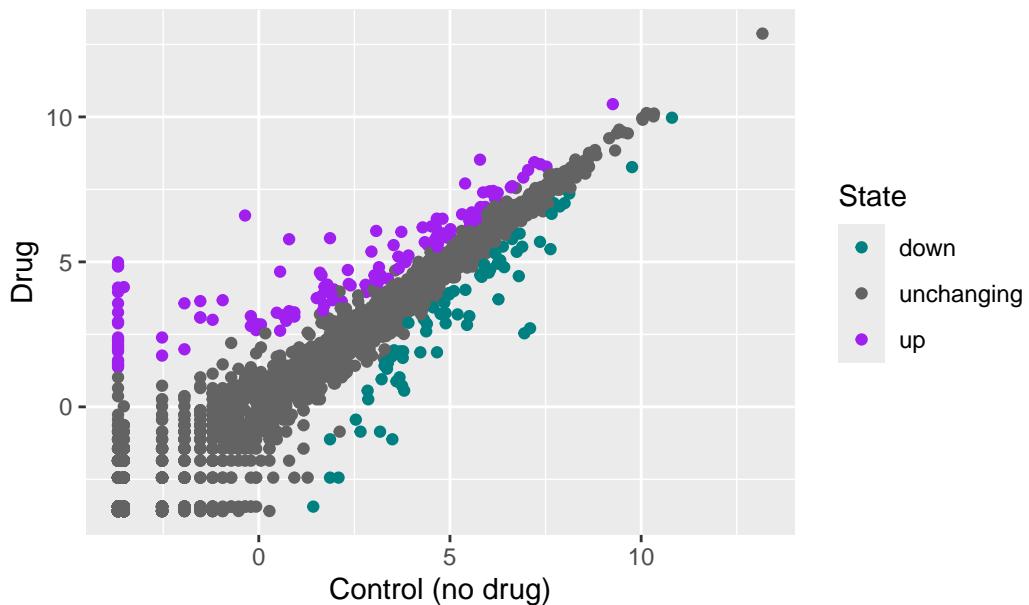
```
ggplot(genes) +
  aes(Condition1, Condition2, col=State) +
  geom_point()
```



Version 3: Let's modify the default colors to something we like.

```
ggplot(genes) +
  aes(Condition1, Condition2, col=State) +
  geom_point() +
  scale_color_manual(values= c("#008080", "#636363", "purple")) +
  labs(x="Control (no drug)",
       y="Drug",
       title="Gene Expression Changes upon GLP-1 drug")
```

Gene Expression Changes upon GLP–1 drug



Going Further

Let's have a look at the famous `gapminder` data set.

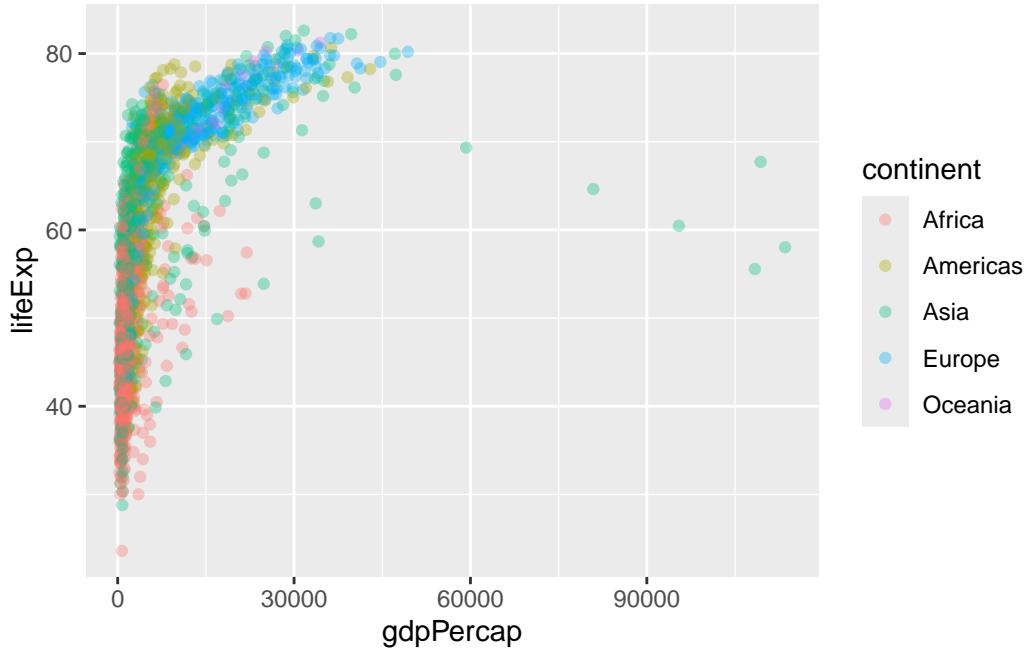
```
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.ts"

gapminder <- read.delim(url)

head(gapminder, 3)
```

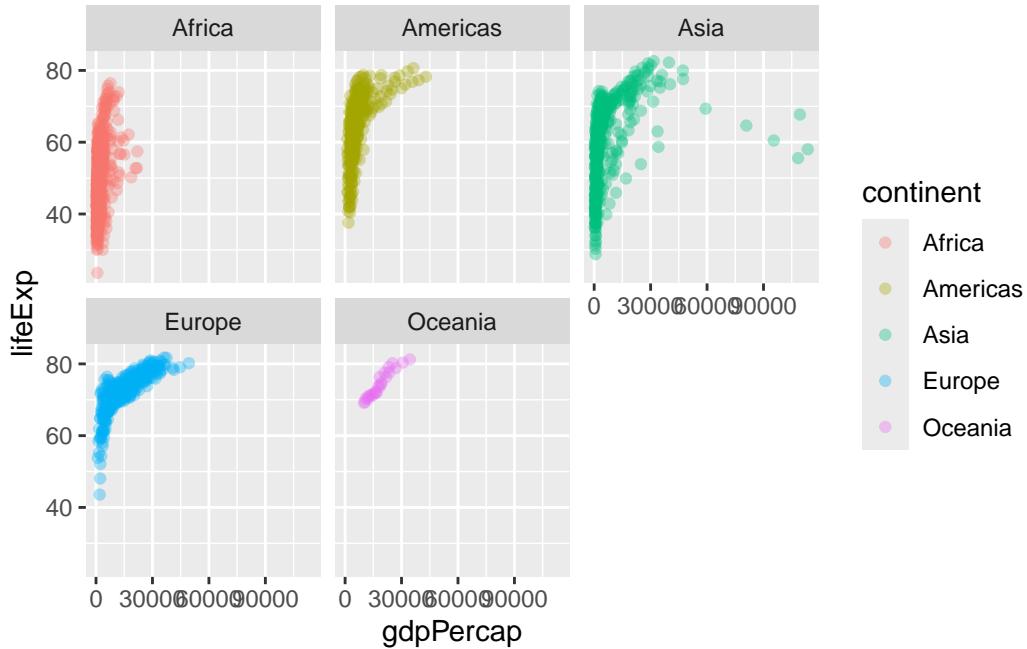
	country	continent	year	lifeExp	pop	gdpPerCap
1	Afghanistan	Asia	1952	28.801	8425333	779.4453
2	Afghanistan	Asia	1957	30.332	9240934	820.8530
3	Afghanistan	Asia	1962	31.997	10267083	853.1007

```
ggplot(gapminder) +
  aes(gdpPerCap, lifeExp, col=continent) +
  geom_point(alpha=(0.35))
```



Let's "facet" (i.e. make a separate plot) by continents rather than the big hot mess above.

```
ggplot(gapminder) +  
  aes(gdpPercap, lifeExp, col=continent) +  
  geom_point(alpha=0.35) +  
  facet_wrap(~continent)
```



Custom Plots

How big is this gapminder dataset?

```
nrow(gapminder)
```

```
[1] 1704
```

I want to “filter” down to a subset of this data. I will use the **dplyr** package to help me.

First I need to install it and load it up. `install.packages("dplyr")` and:

```
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
```

```
gapminder2007 <- filter(gapminder, year == 2007)  
head(gapminder2007)
```

	country	continent	year	lifeExp	pop	gdpPercap
1	Afghanistan	Asia	2007	43.828	31889923	974.5803
2	Albania	Europe	2007	76.423	3600523	5937.0295
3	Algeria	Africa	2007	72.301	33333216	6223.3675
4	Angola	Africa	2007	42.731	12420476	4797.2313
5	Argentina	Americas	2007	75.320	40301927	12779.3796
6	Australia	Oceania	2007	81.235	20434176	34435.3674

```
filter(gapminder2007, country == "Ireland")
```

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

```
filter(gapminder, year == 1977, country == "United States")
```

	country	continent	year	lifeExp	pop	gdpPercap
1	United States	Americas	1977	73.38	220239000	24072.63

Q. Make a plot comparing 1977 and 2007 for all countries

```
gapminderplot <- filter(gapminder, year %in% c(1977, 2007))  
ggplot(gapminderplot) +  
  aes(gdpPercap, lifeExp, col = continent) +  
  geom_point() +  
  facet_wrap(~year) +  
  labs(x="GDP per Capita",  
       y="Life Expectancy")
```

