

Class05: Data Vis with ggplot

Whitney (PID:A16781338)

Graphics system in R

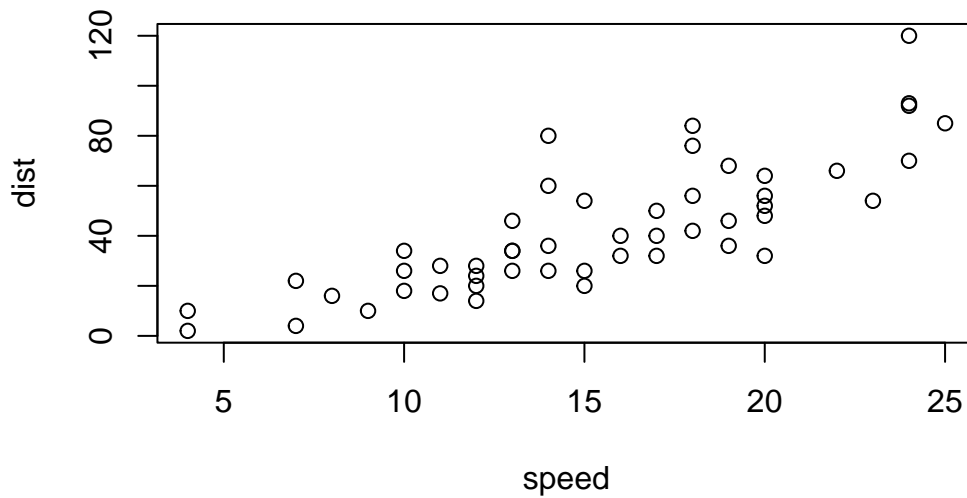
There are many graphics systems in R for making plots and figures.

We have already played a little with “**base R**” graphics and the `plot()` function.

Today we will start learning about a popular graphics package called `ggplot2()`.

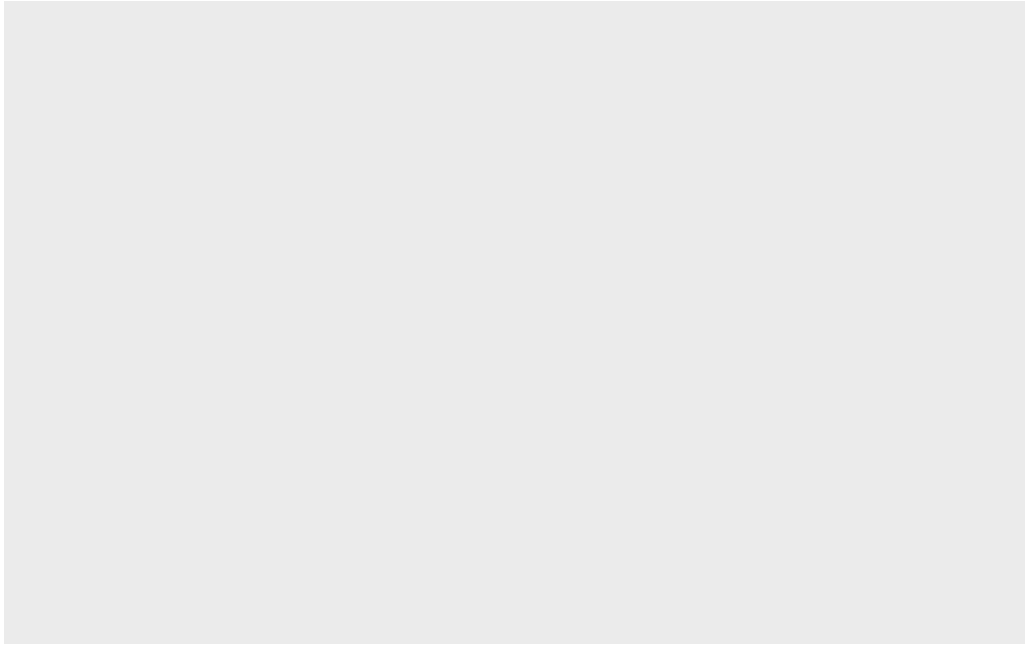
This is an add on package - i.w. we need to install it. I install it (like I install any package) with the `install.packages()` function.

```
plot(cars)
```



before I can use the functions from a package I have to load up the package from my “library”. We use the `library(ggplot2)` command to load it up.

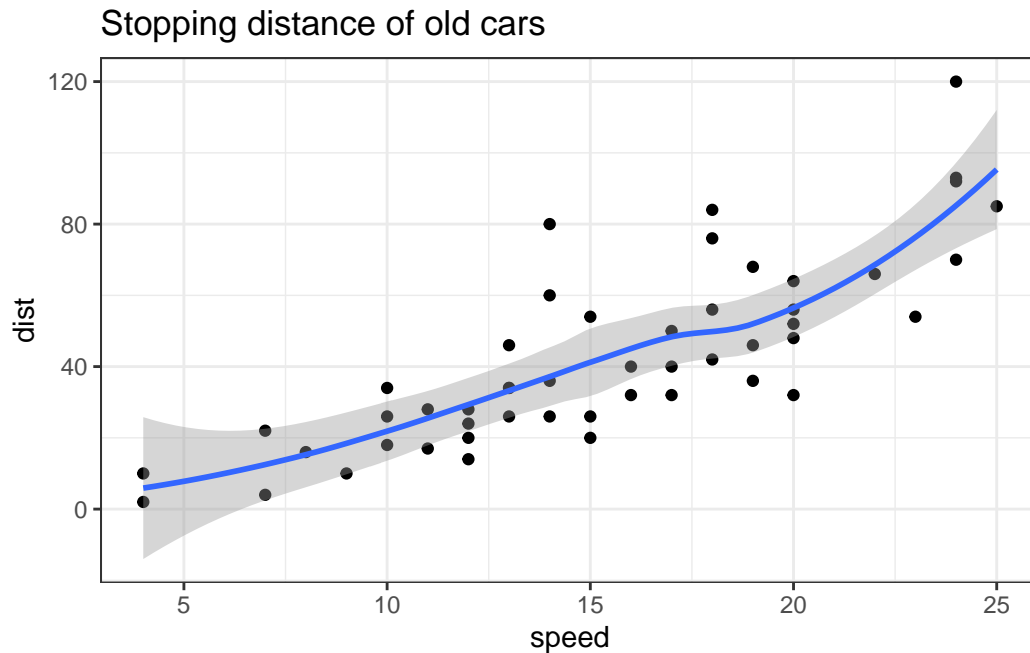
```
library(ggplot2)
ggplot(cars)
```



every ggplot is made up of at least 3 things: - data (the numbers etc. that will go into your plot) - aes (how the columns of data map to the plot aesthetics) - geoms (how the plot actually looks, points, bars,)

```
ggplot(cars) + aes(x=speed, y=dist) + geom_point() + geom_smooth() + labs(title="Stopping
```

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



for simple plots, ggplot is more verbose - it takes more code than base R plot.

```
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

```
nrow(genes)
```

```
[1] 5196
```

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

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

down	unchanging	up
72	4997	127

Q. Use the `nrow()` function to find out how many genes are in this dataset. What is your answer? 5196

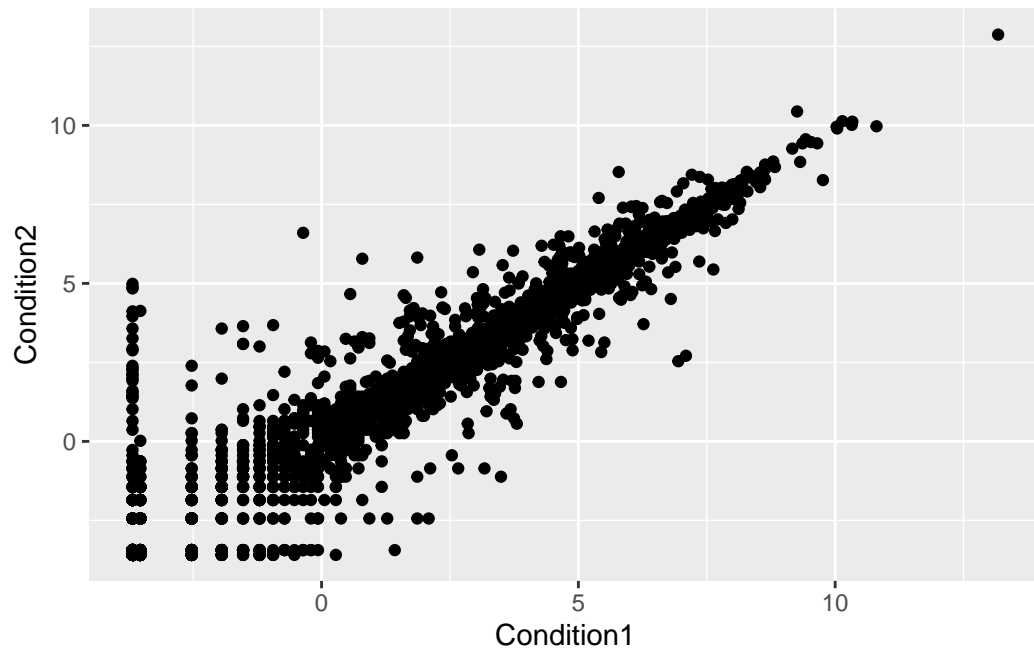
Q. Use the `colnames()` function and the `ncol()` function on the `genes` data frame to find out what the column names are (we will need these later) and how many columns there are. How many columns did you find? 4: Gene, Condition1, Condition2, State

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? 127

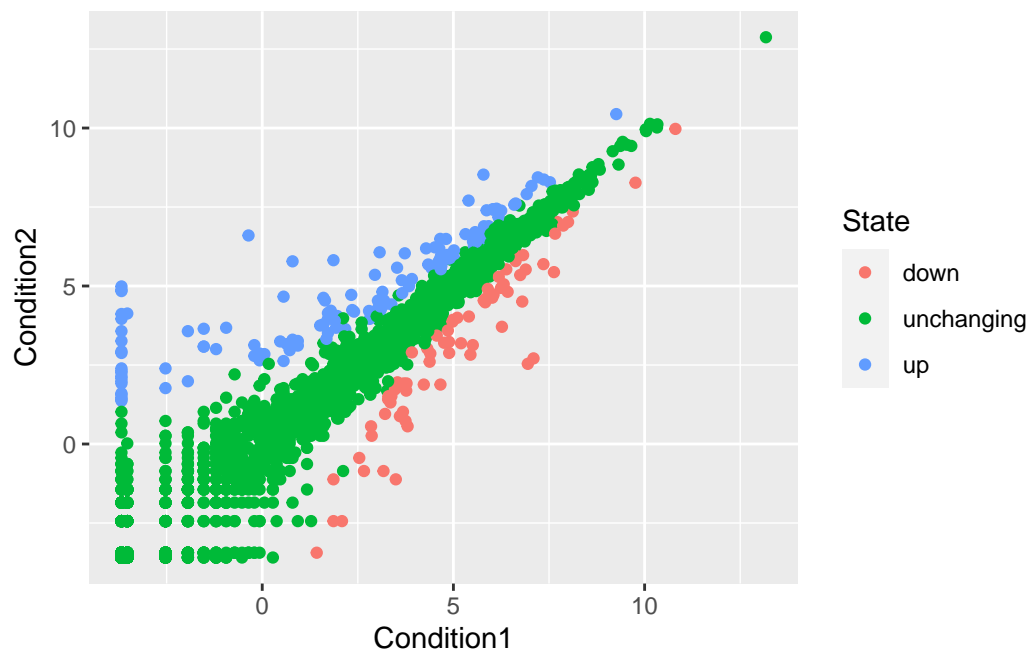
Q. Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset? 2.44

Q. Complete the code below to produce the following plot

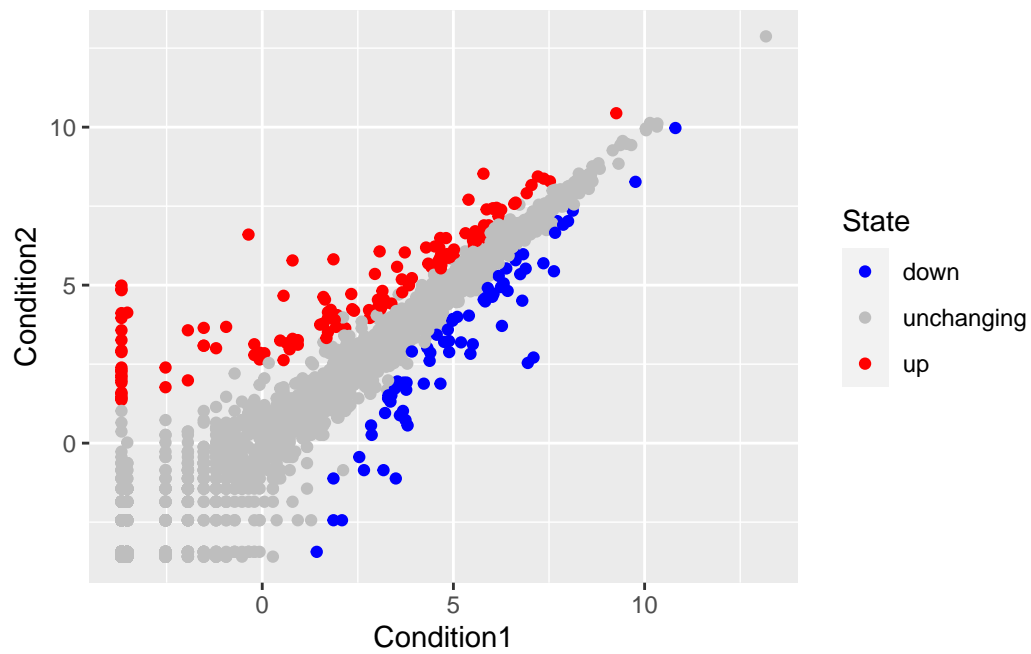
```
ggplot(genes)+aes(x=Condition1, y=Condition2)+geom_point()
```



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

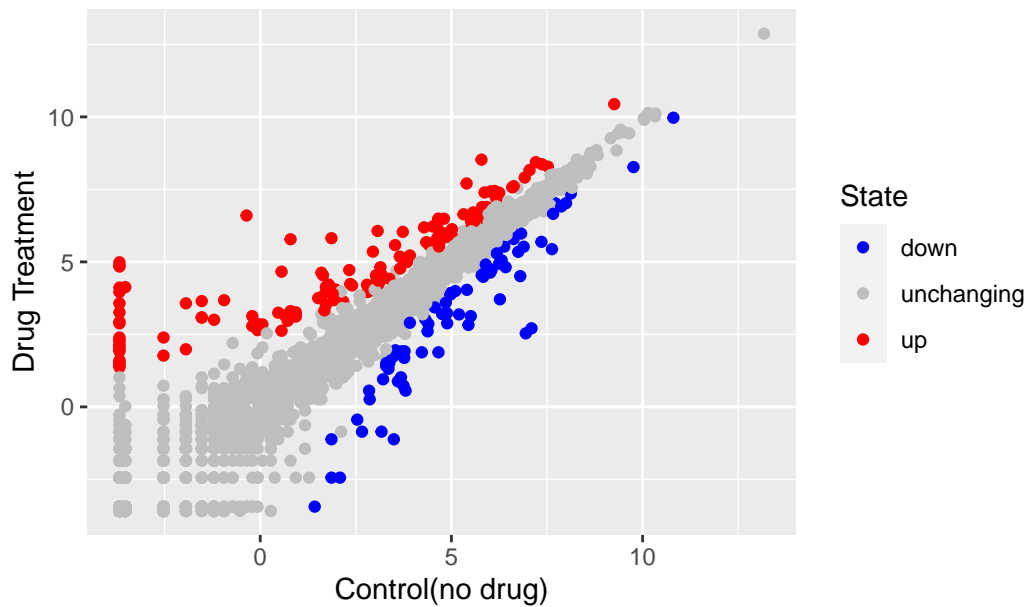


```
p+scale_colour_manual(values=c("blue", "gray", "red"))
```



```
p+scale_colour_manual(values=c("blue", "gray", "red"))+labs(title="Gene Expression Changes")
```

Gene Expression Changes Upon Drug Treatment



```
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.  
gapminder <- read.delim(url)  
  
# install.packages("dplyr") ## un-comment to install if needed  
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

```
gapminder_2007 <- gapminder %>% filter(year==2007)
```

```
ggplot(gapminder_2007)+aes(x="gdpPerCap",y="lifeExp")+geom_point()
```

