

class05

Audrey Ting Zhu(A16898668)

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

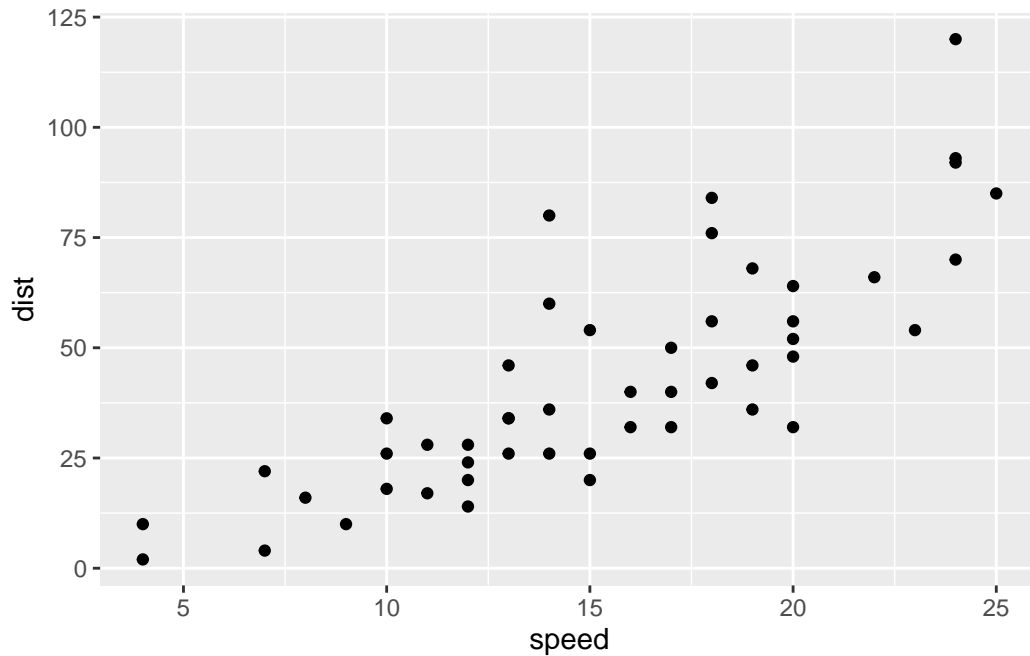
All ggplot figures have a least 3 layers: The `echo: false` option disables the printing of code (only output is displayed). `-data` -aes(how the data map to the plot) `-geoms`(how we draw the plot, lines, points,etc.)

Before I use any new package I need to download and install it with the `'install.packages()'` command.

I will never use `'install.package()'` within my quarto document otherwise I will insall the package over and over again.

```
#install.packages("ggplot2")
library(ggplot2)

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



#For which phases is data visualization important in our scientific workflows? ANS:All of the above

#True or False? The ggplot2 package comes already installed with R? ANS: FALSE # Which plot types are typically NOT used to compare distributions of numeric variables? ANS:Network graphs

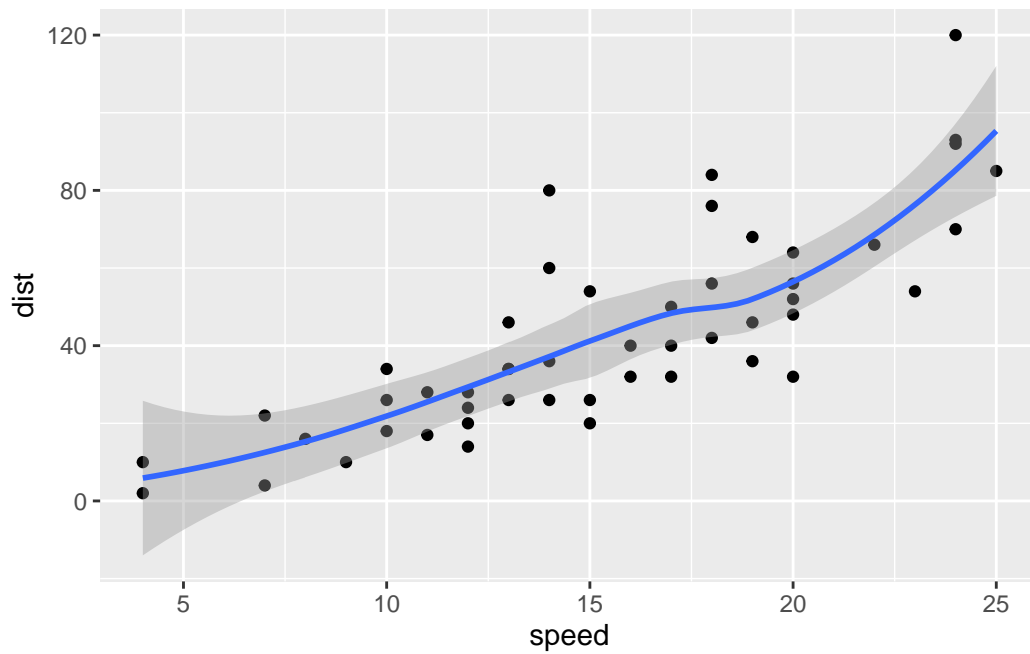
#Which statement about data visualization with ggplot2 is incorrect? ANS:ggplot2 is the only way to create plots in R

#Which geometric layer should be used to create scatter plots in ggplot2? ANS: geom_point()

#In your own RStudio can you add a trend line layer to help show the relationship between the plot variables with the geom_smooth() function?

```
library(ggplot2)
ggplot(cars)+
aes(x=speed, y=dist) + geom_point()+ geom_smooth()
```

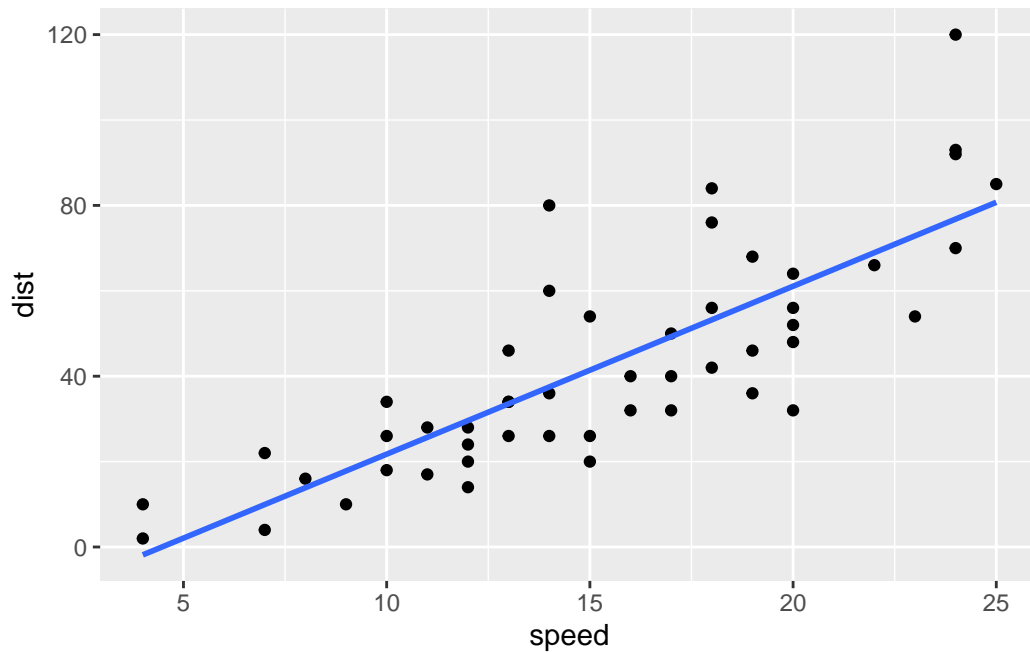
```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



#Argue with `geom_smooth()` to add a straight line from a linear model without the shaded standard error region?

```
library(ggplot2)
ggplot(cars)+
aes(x=speed, y=dist) + geom_point()+ geom_smooth(method="lm", se=FALSE)
```

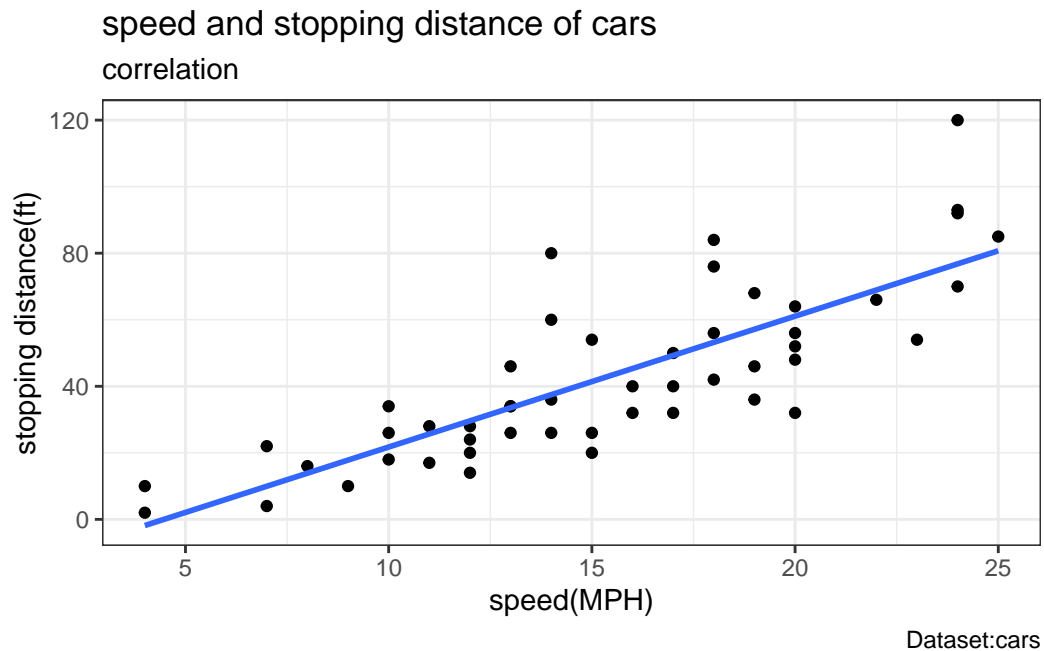
```
`geom_smooth()` using formula = 'y ~ x'
```



#Can you finish this plot by adding various label annotations with the `labs()` function and changing the plot look to a more conservative “black & white” theme by adding the `theme_bw()` function:

```
library(ggplot2)
ggplot(cars)+
  aes(x=speed, y=dist)+ geom_point()+labs(title= "speed and stopping distance of cars",x="speed")
```

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



```
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

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

```
nrow(genes)
```

```
[1] 5196
```

#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? ANS: The column names are "Gene" "Condition1" "Condition2" "State". The number of columns is 4.

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

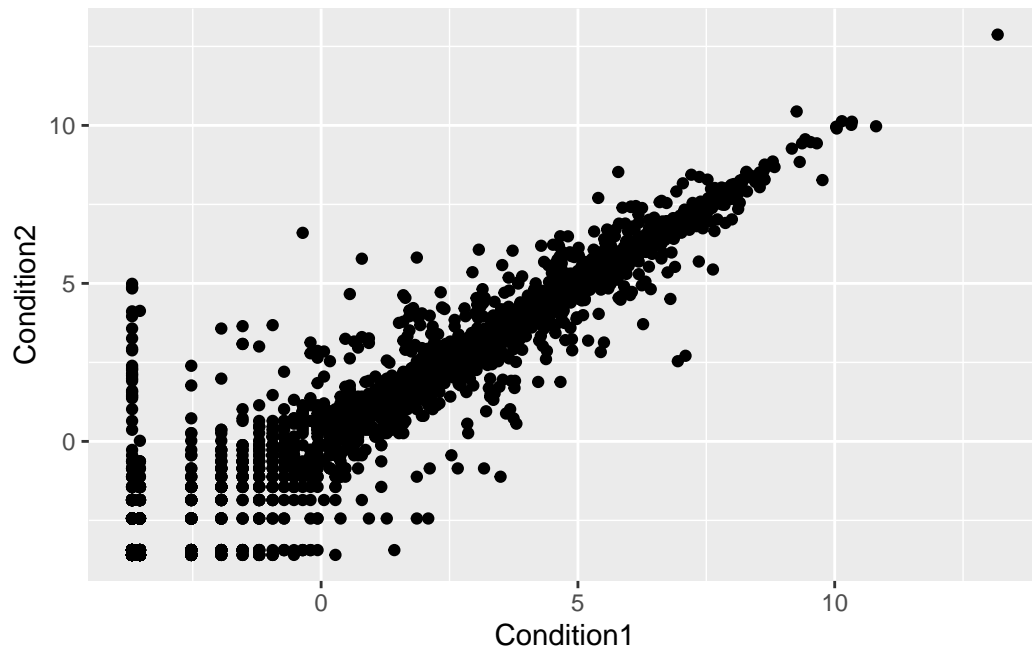
#Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset?ANS:2.44 upregulated

```
round( table(genes$State)/nrow(genes) * 100, 2 )
```

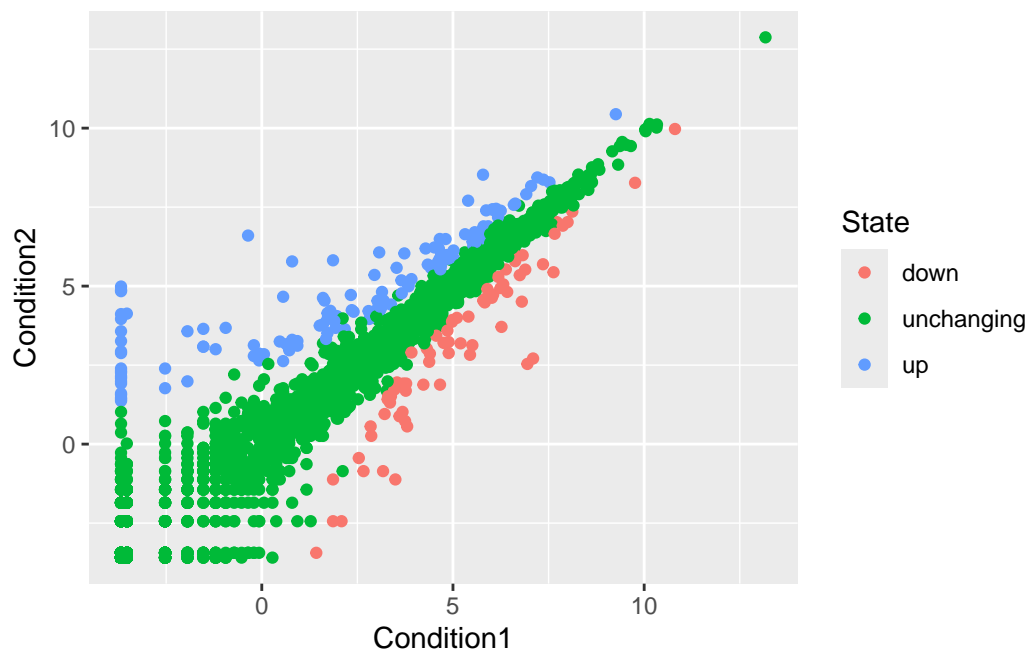
down	unchanging	up
1.39	96.17	2.44

#Complete the code below to produce the following plot

```
library(ggplot2)
ggplot(genes) +
  aes(x=Condition1, y=Condition2)+geom_point()
```



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



#Nice, now add some plot annotations to the p object with the labs() function so your plot looks like the following:

```
p<-ggplot(genes) + aes(x=Condition1,y=Condition2,col=State)+geom_point()  
p + scale_colour_manual(values=c("blue","gray","red")) +  
  labs(title="Gene Expression Changes Upon Drug Treatment",  
        x="Control (no drug) ",  
        y="Drug Treatment")
```

