

Class 05: Data Visualization with GGPLOT

AUTHOR

Laura Biggs

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
```

The `echo: false` option disables the printing of code (only output is displayed).

Our first ggplot

Install ggplot2 w/install.packages('ggplot2') Render in package w/library(ggplot2)

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

Warning: package 'ggplot2' was built under R version 4.1.3

Cars plots from Q6

```
# Load in cars data
head(cars)
```

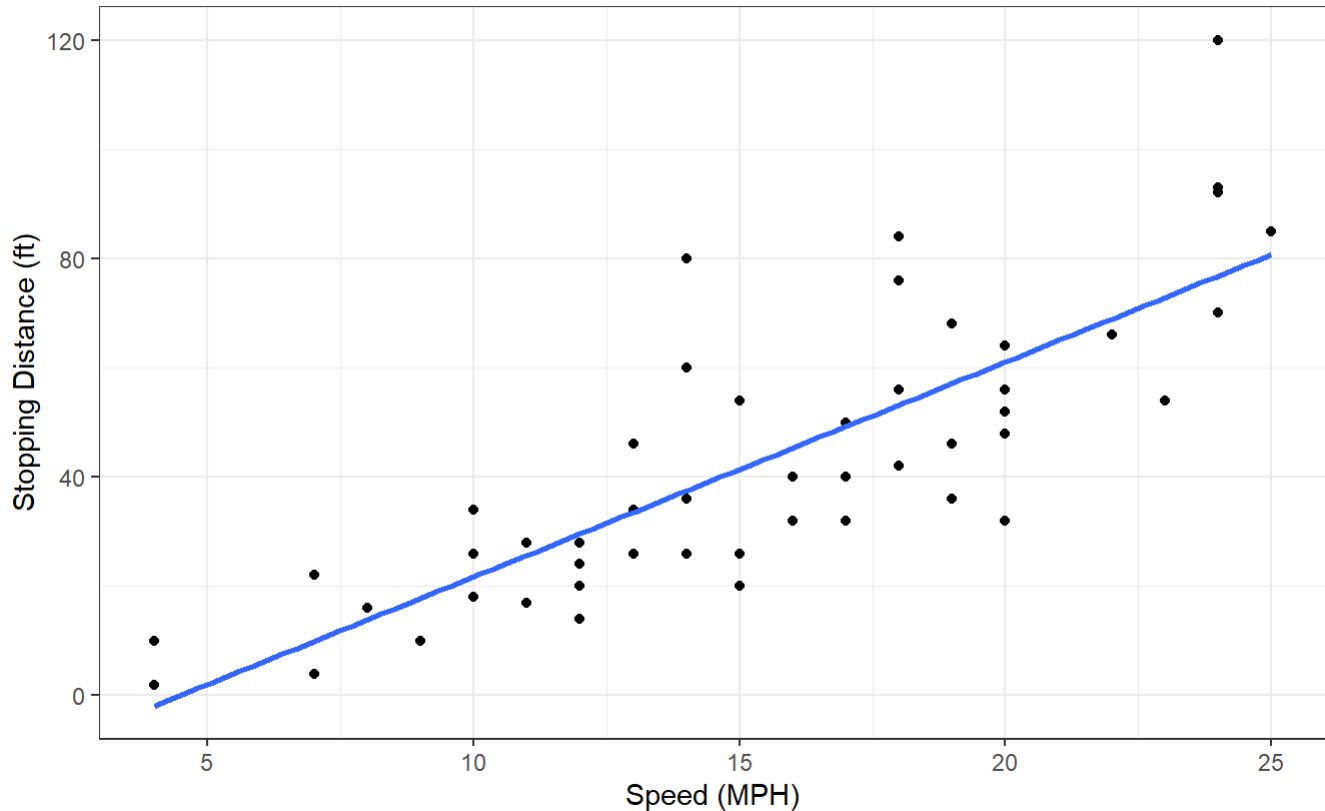
	speed	dist
1	4	2
2	4	10
3	7	4
4	7	22
5	8	16
6	9	10

```
# Create cars ggplot w/ ggplot, aes, geom_point
# geom_smooth creates thin trendline w/o SD
# Add labels with labs function
# Theme_bw created black and white theme
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method = 'lm', se=FALSE) +
  labs(title="Speed and Stopping Distances of Cars",
       x="Speed (MPH)",
       y="Stopping Distance (ft)",
       subtitle="Cars Dataset",
       caption = "Dataset: cars") +
  theme_bw()
```

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

Speed and Stopping Distances of Cars

Cars Dataset



Dataset: cars

Genes plot from Q6

```
# Load in data
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expressio
genes <- read.delim(url)

# First 6 genes in the dataframe
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

```
#How many rows in the df  
nrow(genes)
```

```
[1] 5196
```

```
#How many columns in the df  
ncol(genes)
```

```
[1] 4
```

```
colnames(genes)
```

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

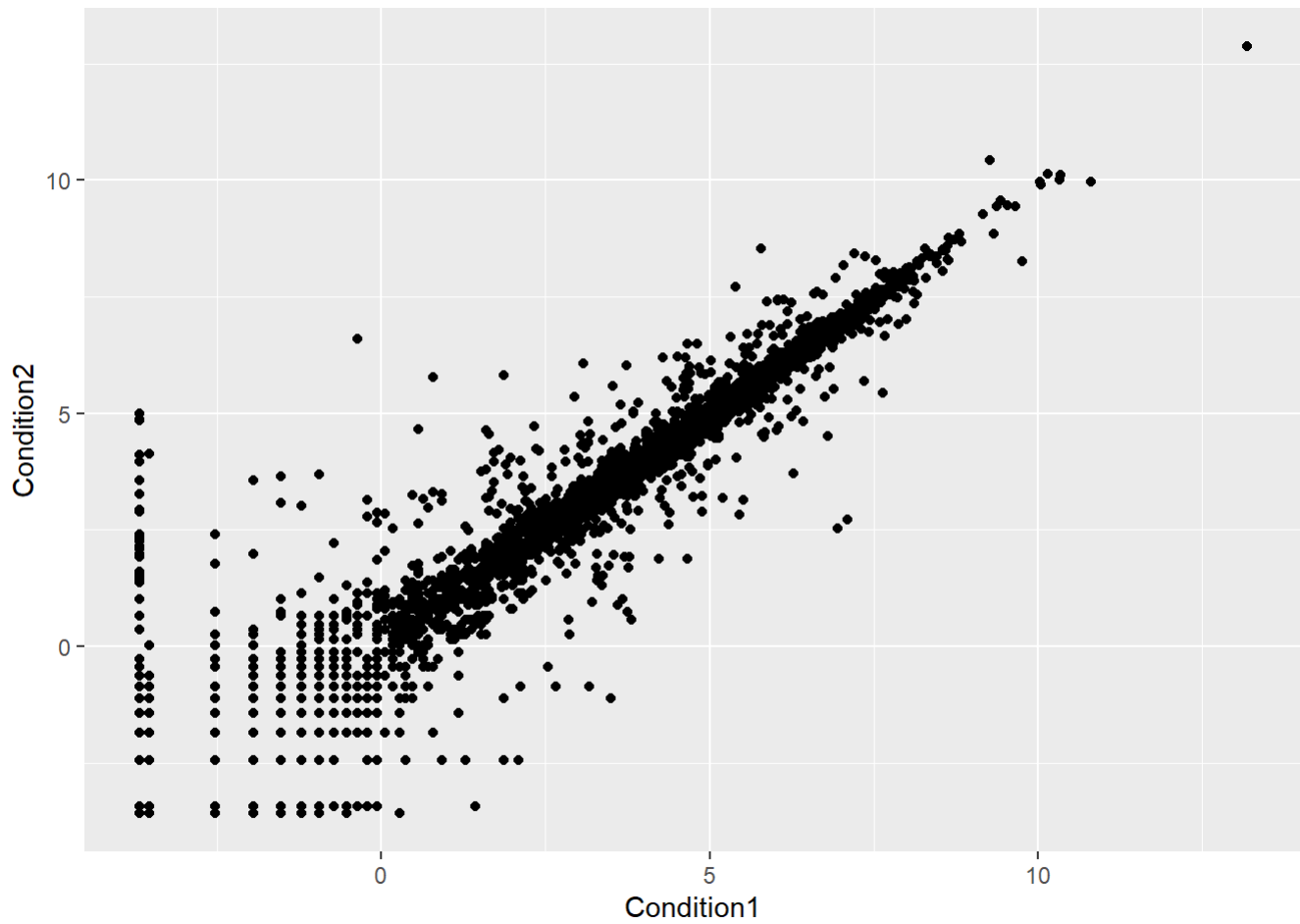
```
#Stratify the table by levels of regulation  
#Remember the $ sign isolates the selected column  
table(genes$State)
```

down	unchanging	up
72	4997	127

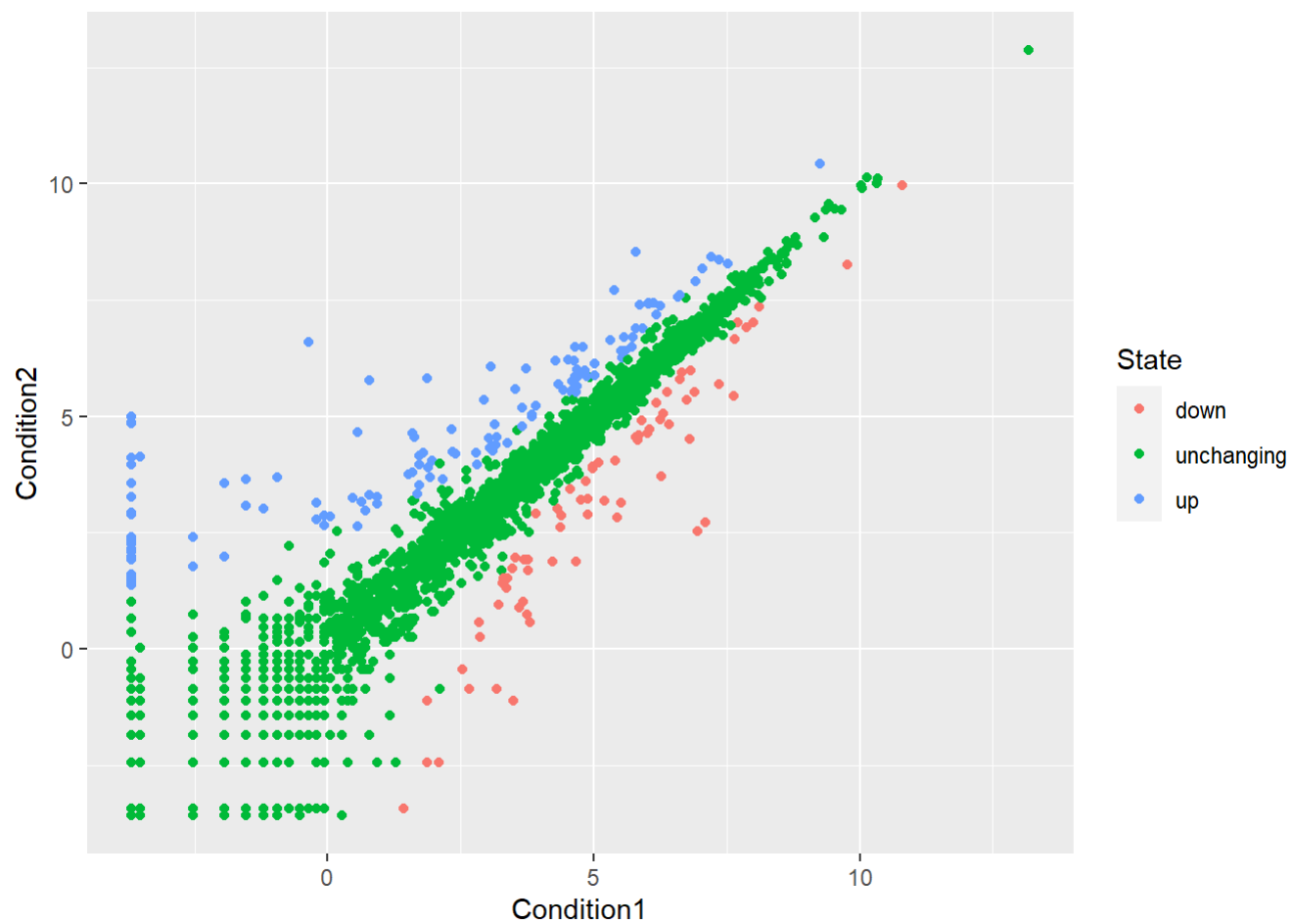
```
#What fraction of total genes are upregulated (to the 2nd decimal point)?  
round(table(genes$State)/nrow(genes) * 100, 2)
```

down	unchanging	up
1.39	96.17	2.44

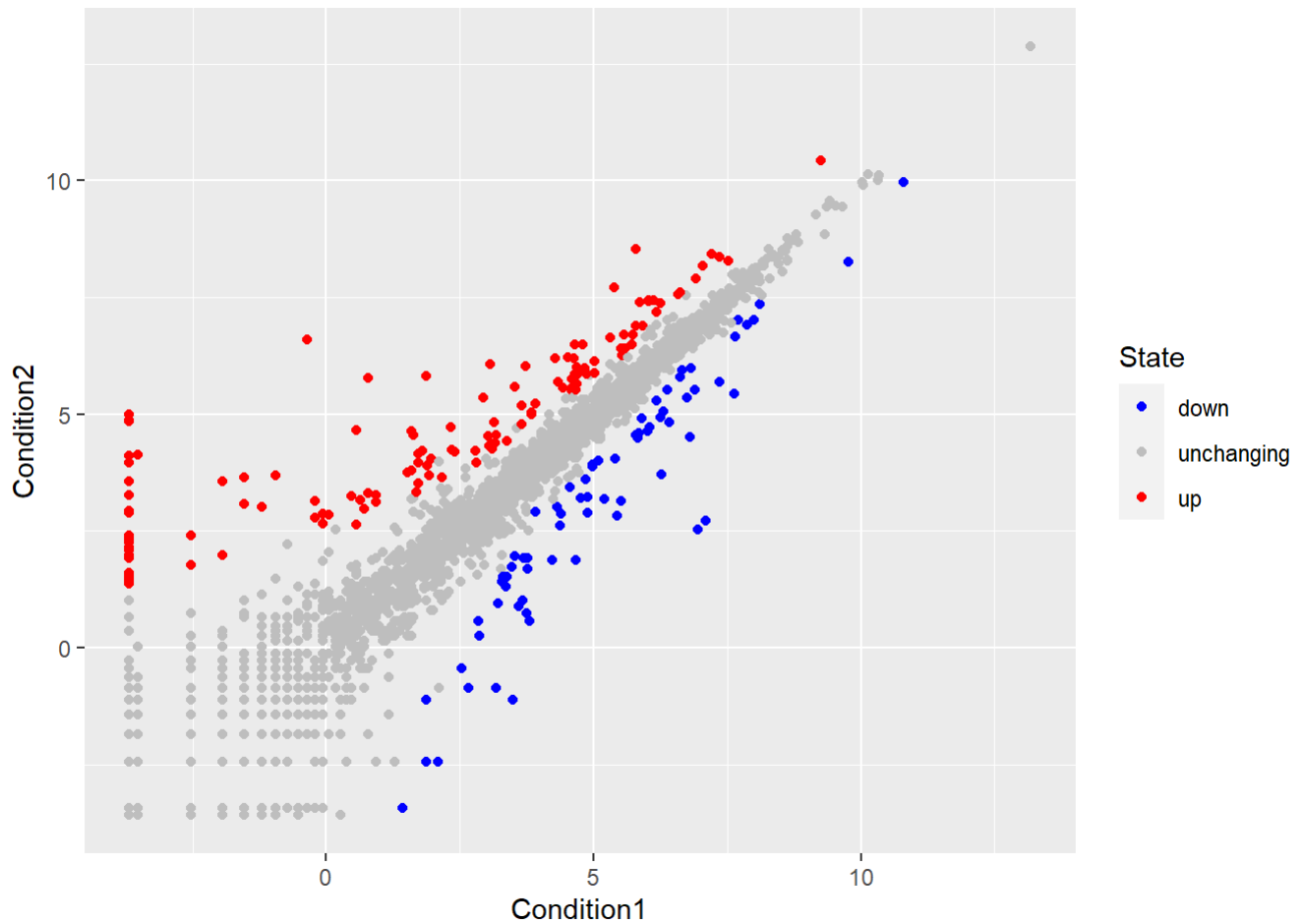
```
#Make ggplot w/condition 1 and 2  
#x and y are NOT in string format; no spaces between characters allowed  
ggplot(genes) +  
  aes(x=Condition1, y=Condition2) +  
  geom_point()
```



```
#Assign as p and add color according to the state  
ggplot(genes) +  
  aes(x=Condition1, y=Condition2, col=State) +  
  geom_point()
```



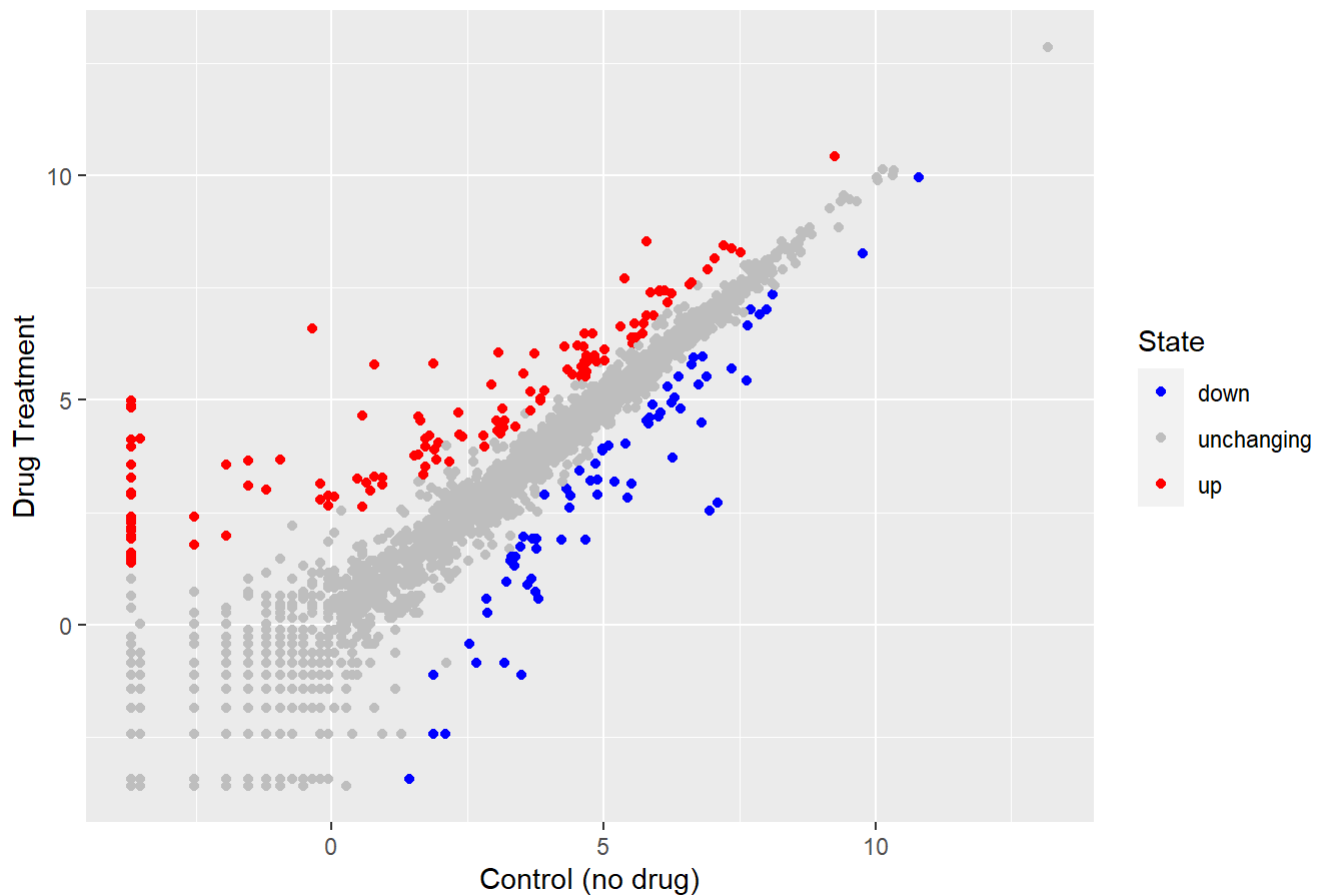
```
#Change the color to blue, grey, and red  
ggplot(genes) +  
  aes(x=Condition1, y=Condition2, col=State) +  
  geom_point() +  
  scale_color_manual(values=c("blue", "grey", "red"))
```



```
#Add plot labels; x and y titles are STRINGS
p <- ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State) +
  geom_point() +
  scale_color_manual(values=c("blue", "grey", "red")) +
  labs(title = 'Gene Expression Changes Upon Drug Treatment',
       x='Control (no drug)',
       y='Drug Treatment')
```

p

Gene Expression Changes Upon Drug Treatment



Gapminder Extension (Partially completed)

```
#Install gapminder url
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdat

gapminder <- read.delim(url)
```

```
#Install dplyr
#install.packages("dplyr")
library(dplyr)
```

Warning: package 'dplyr' was built under R version 4.1.3

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

```
library(ggplot2)

#Filter on the year 2007
gapminder_2007 <- gapminder %>% filter(year==2007)

#Make ggplot with GDP and Life expectancy
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point()
```

