

# lab5.0

Ilyas Darif A16577084

2025-01-21

## Intro to ggplot

there are many graphics systems in R (ways to make plots and figures). These include “base” R plots. today we will focus mostly on **ggplot2** package.

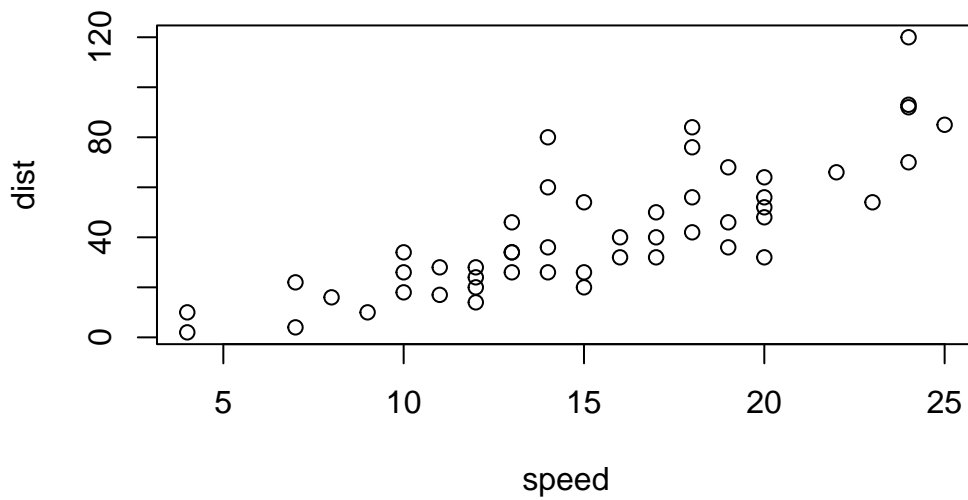
lets start with a plot of a simple in-built dataset called **cars**.

```
cars
```

	speed	dist
1	4	2
2	4	10
3	7	4
4	7	22
5	8	16
6	9	10
7	10	18
8	10	26
9	10	34
10	11	17
11	11	28
12	12	14
13	12	20
14	12	24
15	12	28
16	13	26
17	13	34
18	13	34
19	13	46
20	14	26
21	14	36

22	14	60
23	14	80
24	15	20
25	15	26
26	15	54
27	16	32
28	16	40
29	17	32
30	17	40
31	17	50
32	18	42
33	18	56
34	18	76
35	18	84
36	19	36
37	19	46
38	19	68
39	20	32
40	20	48
41	20	52
42	20	56
43	20	64
44	22	66
45	23	54
46	24	70
47	24	92
48	24	93
49	24	120
50	25	85

```
plot(cars)
```

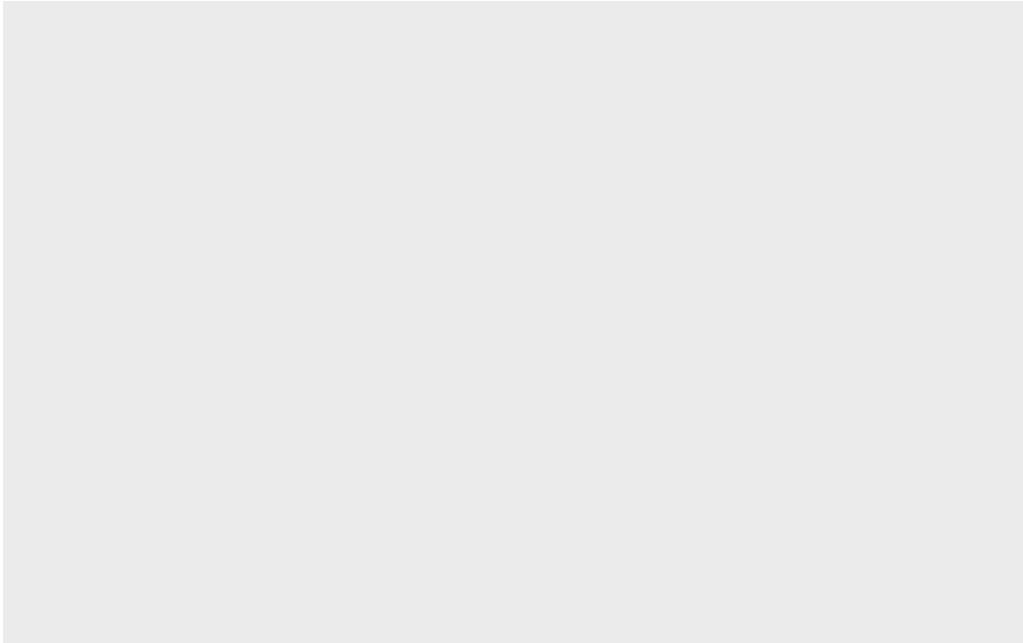


lest see how we can make this figure using **ggplot** install first by using `install.packages`

i will wun `install.packages(ggplot2)` in the R console not in the doc

before i can use a func froim a package i need to load the package from “`library()`” by doing `library(ggplot2)` to call it

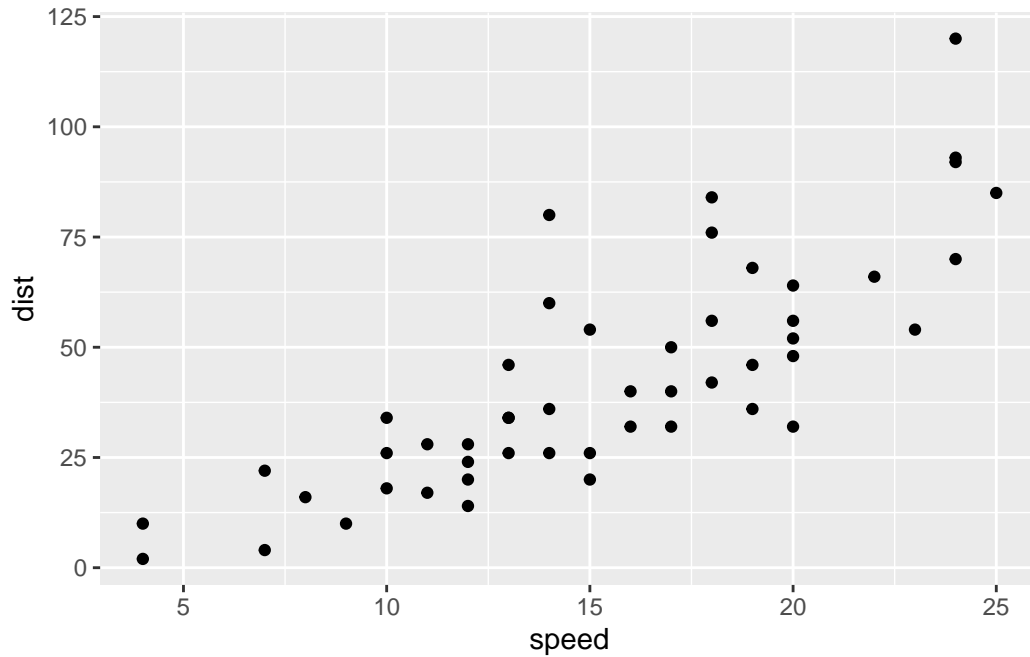
```
library(ggplot2)
ggplot(cars)
```



all ggplot figures have at least 3 things (called layers) these include:

- **data** (the input data set i want plot)
- **aes** (the aesthetic mapping of the data to my plot)
- **geoms** ( the geom\_point(), geom\_line)

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

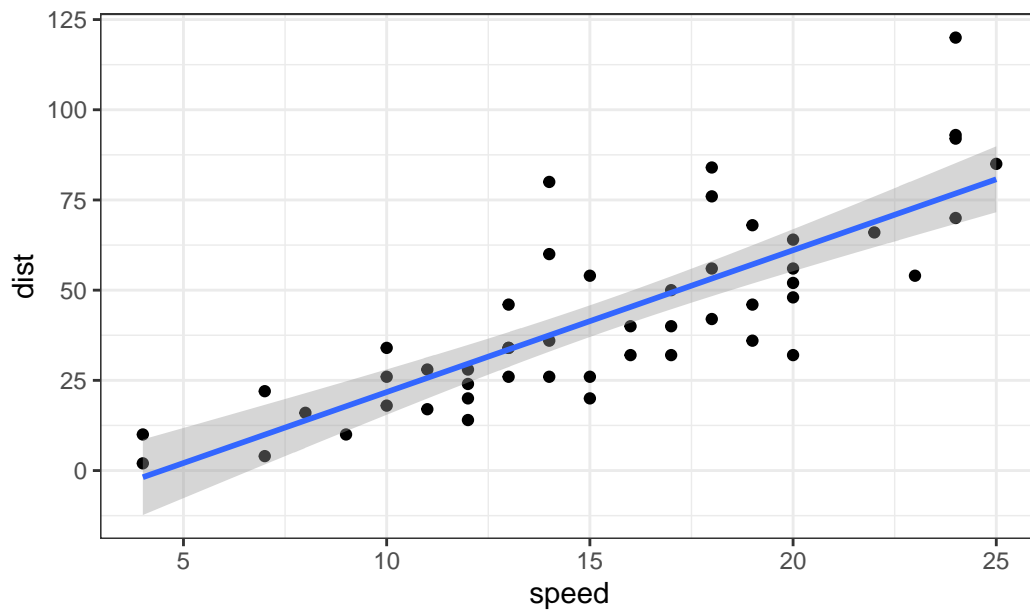


lets add a line to tshow the relation

```
ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point() +  
  geom_smooth(method="lm") +  
  theme_bw() +  
  labs(title = "My first GGLOT")
```

`geom\_smooth()` using formula = 'y ~ x'

## My first GGPLOT



```
## used to make the line
```

the code to read a dataset

```
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

How many genes are in this dataset

```
nrow(genes)
```

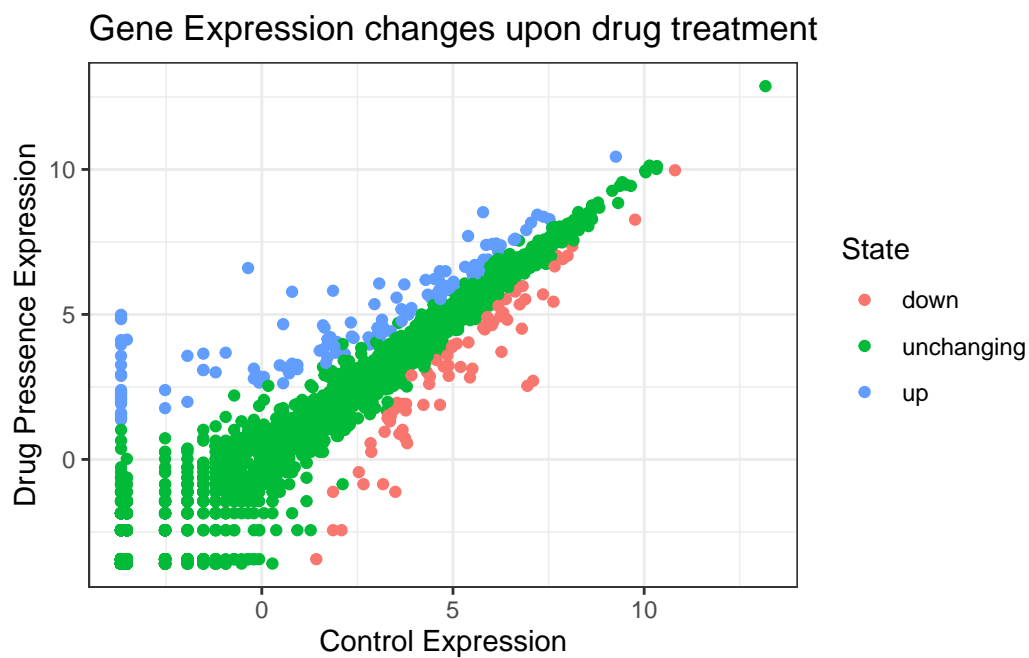
```
[1] 5196
```

```
ncol(genes)
```

```
[1] 4
```

A first plot of this dataset

```
ggplot(genes) +  
  aes(x=Condition1, y=Condition2, col=State) +  
  geom_point() +  
  theme_bw() +  
  labs(title = "Gene Expression changes upon drug treatment",  
        x="Control Expression",  
        y="Drug Presence Expression")
```



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

```
<ggproto object: Class ScaleDiscrete, Scale, gg>  
  aesthetics: colour  
  axis_order: function  
  break_info: function
```

```

break_positions: function
breaks: waiver
call: call
clone: function
dimension: function
drop: TRUE
expand: waiver
get_breaks: function
get_breaks_minor: function
get_labels: function
get_limits: function
get_transformation: function
guide: legend
is_discrete: function
is_empty: function
labels: waiver
limits: NULL
make_sec_title: function
make_title: function
map: function
map_df: function
n.breaks.cache: NULL
na.translate: TRUE
na.value: grey50
name: waiver
palette: function
palette.cache: NULL
position: left
range: environment
rescale: function
reset: function
train: function
train_df: function
transform: function
transform_df: function
super: <ggproto object: Class ScaleDiscrete, Scale, gg>

```

```

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

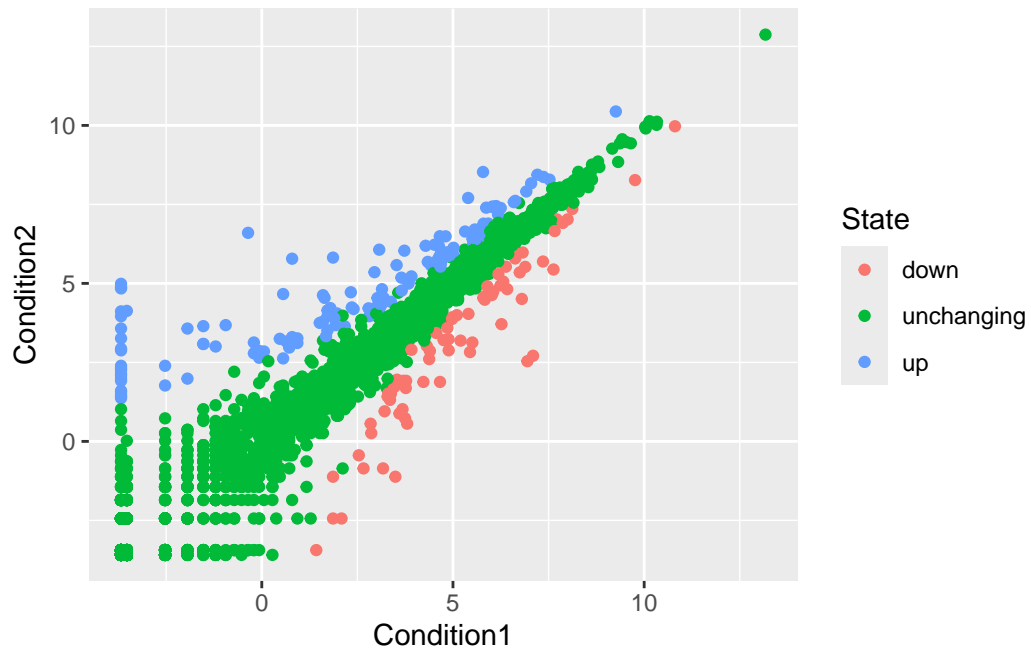
```

```

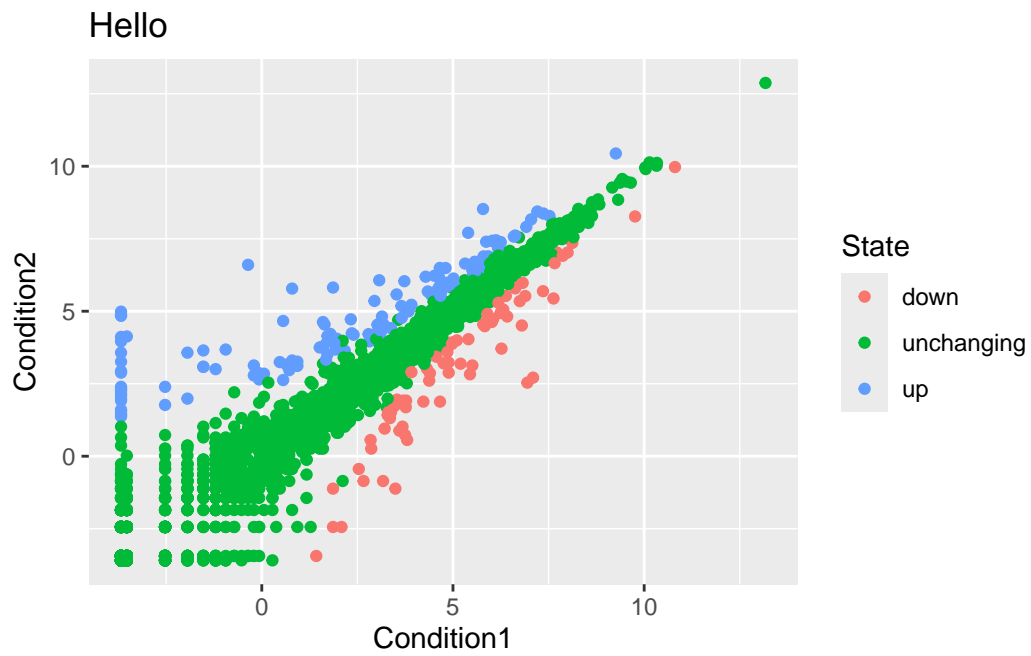
p

```

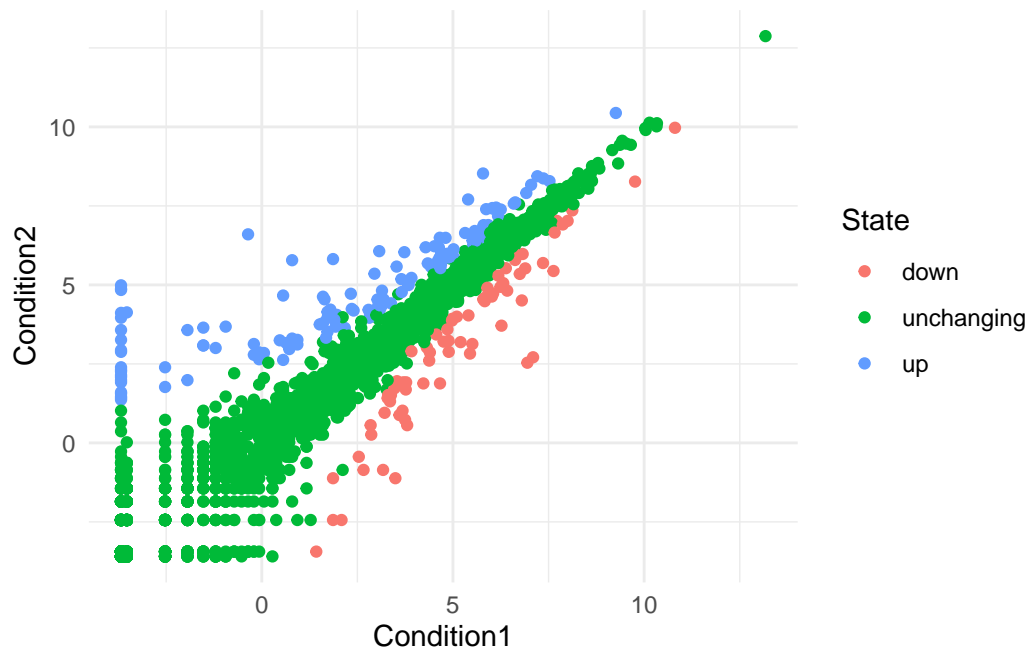




```
p + labs(title = "Hello")
```



```
p + theme_minimal()
```



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

down	unchanging	up
72	4997	127

```
##GapMinder
```

```
library(gapminder)  
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

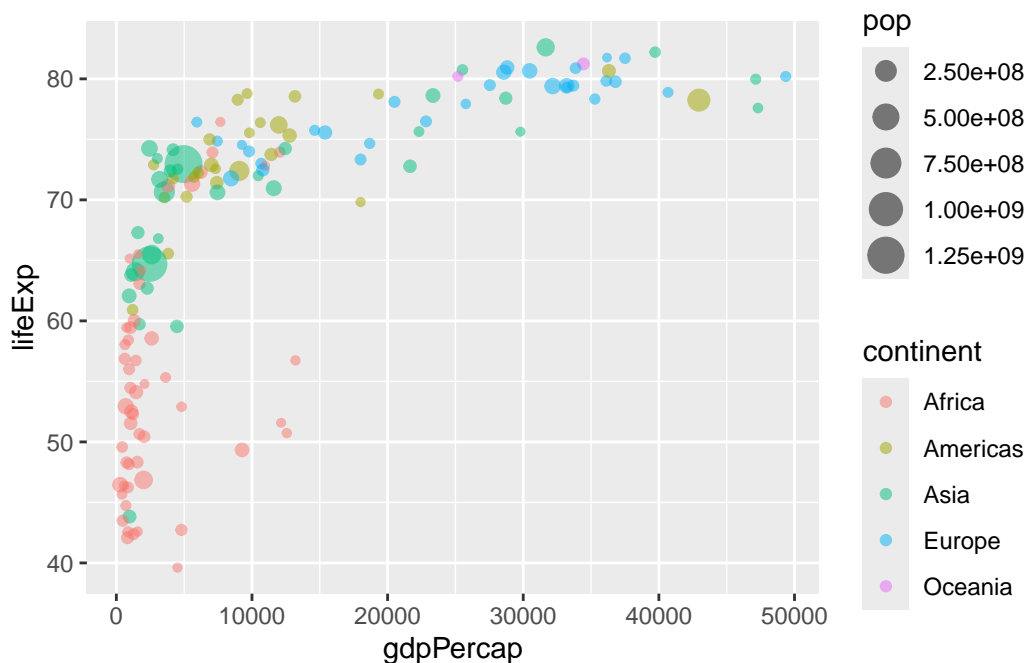
```
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.tsv"
```

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

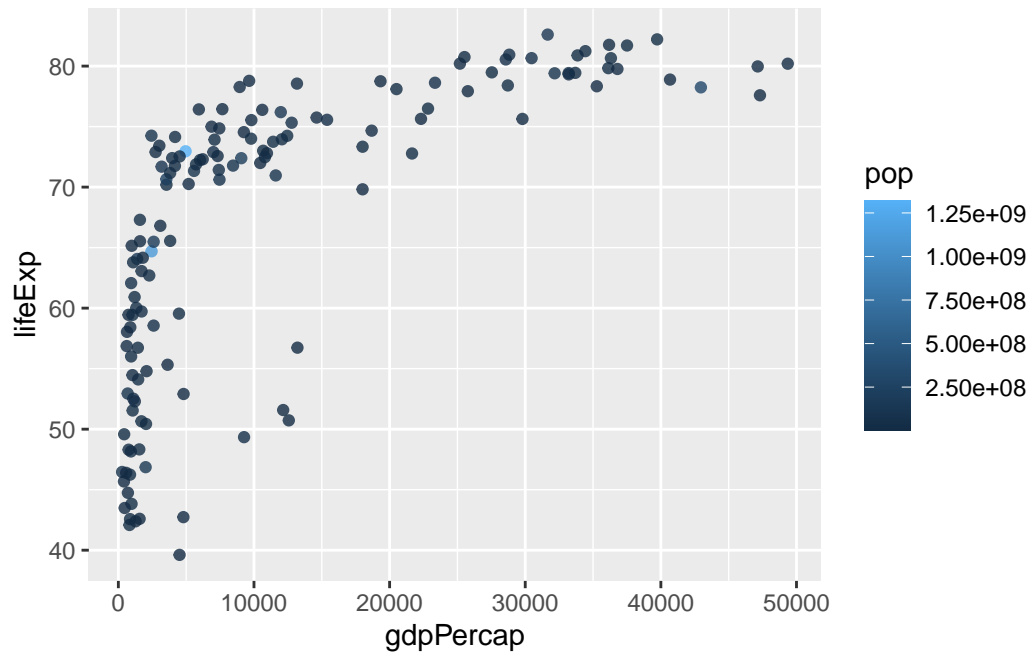
	country	continent	year	lifeExp	pop	gdpPercap
1	Afghanistan	Asia	1952	28.801	8425333	779.4453
2	Afghanistan	Asia	1957	30.332	9240934	820.8530
3	Afghanistan	Asia	1962	31.997	10267083	853.1007
4	Afghanistan	Asia	1967	34.020	11537966	836.1971
5	Afghanistan	Asia	1972	36.088	13079460	739.9811
6	Afghanistan	Asia	1977	38.438	14880372	786.1134

```
gapminder_2007 <- gapminder %>% filter(year==2007)
gapminder_1957 <- gapminder %>% filter(year==1957)

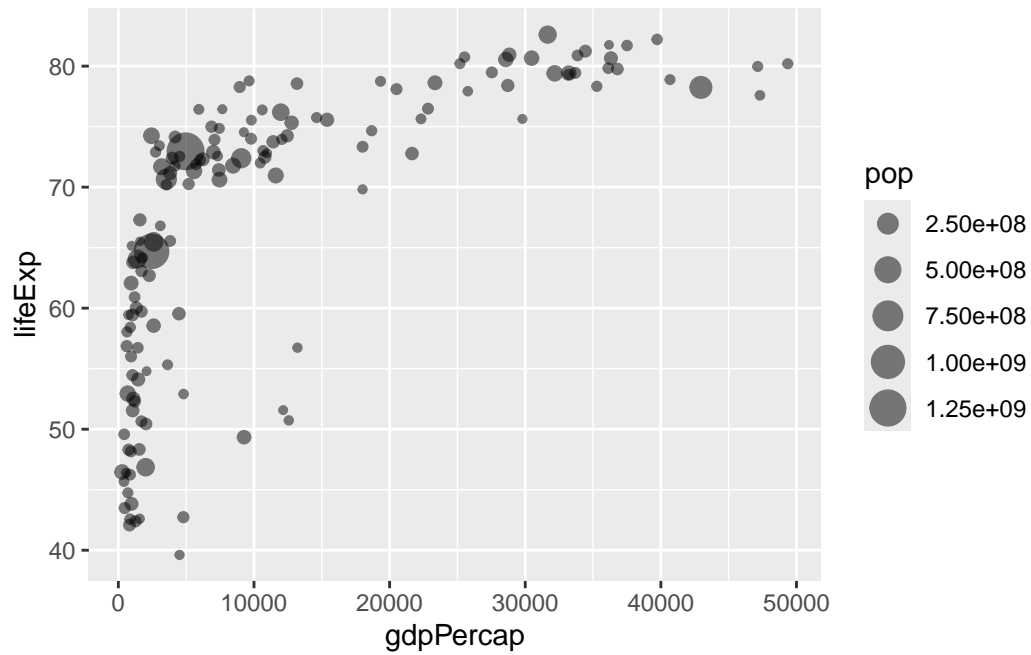
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y= lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5)
```



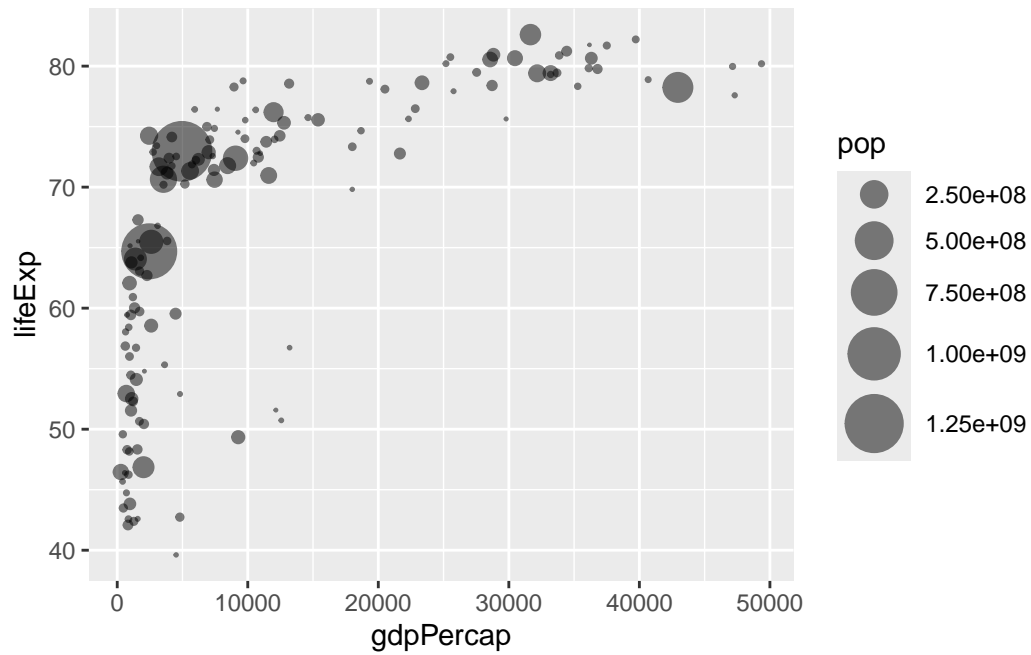
```
ggplot(gapminder_2007) +  
  aes(x=gdpPercap, y= lifeExp, color=pop) +  
  geom_point(alpha=0.8)
```



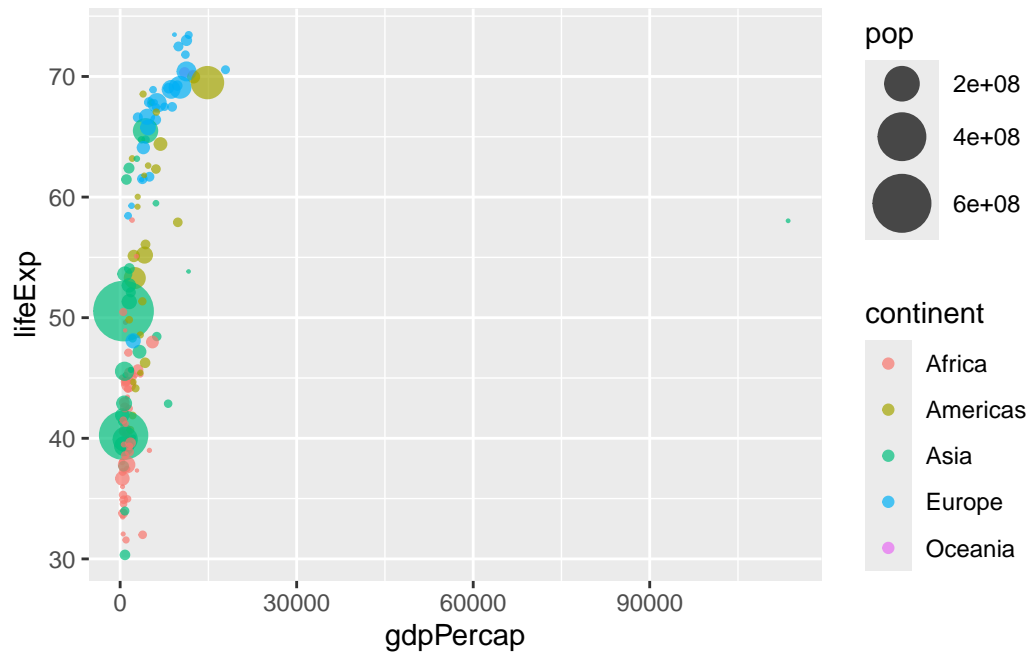
```
ggplot(gapminder_2007) +  
  aes(x=gdpPercap, y= lifeExp, size=pop) +  
  geom_point(alpha=0.5)
```



```
ggplot(gapminder_2007) +  
  geom_point(aes(x = gdpPerCap, y = lifeExp,  
                 size = pop), alpha=0.5) +  
  scale_size_area(max_size = 10)
```

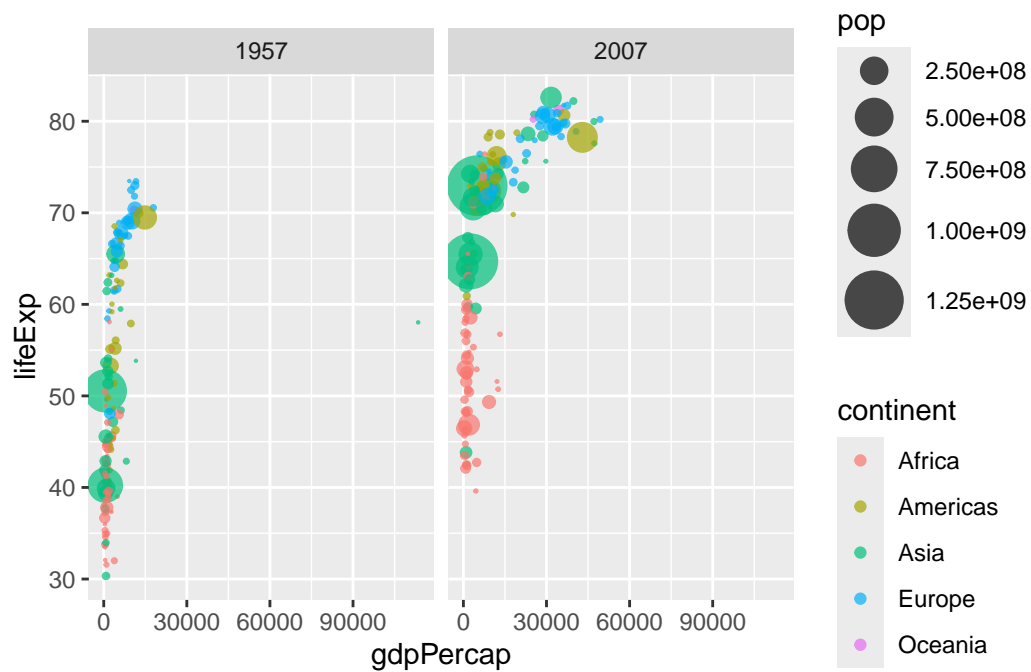


```
ggplot(gapminder_1957) +  
  aes(x = gdpPercap, y = lifeExp, color=continent,  
       size = pop) +  
  geom_point(alpha=0.7) +  
  scale_size_area(max_size = 10)
```



```
gapminder_1957 <- gapminder %>% filter(year==1957 | year==2007)

ggplot(gapminder_1957) +
  geom_point(aes(x = gdpPercap, y = lifeExp, color=continent,
                 size = pop), alpha=0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)
```



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

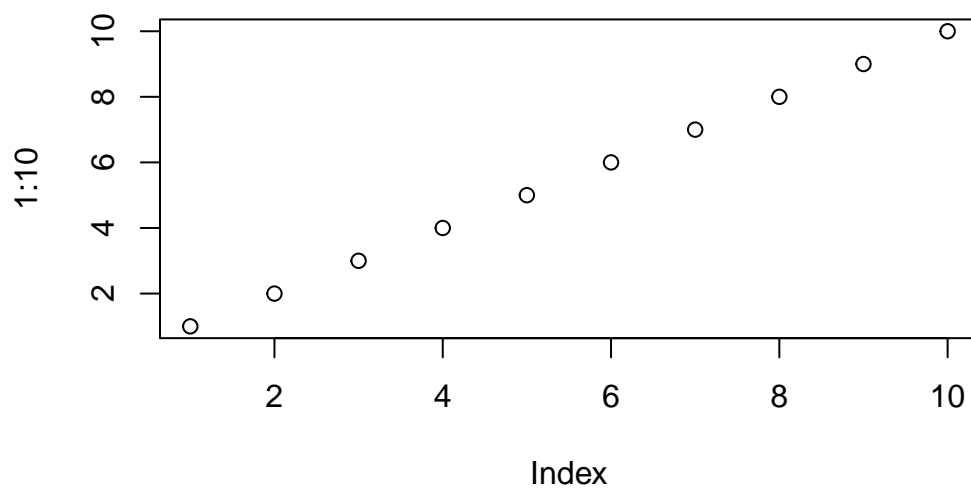
```
round(table(genes$State)/nrow(genes), 4)
```

down	unchanging	up
0.0139	0.9617	0.0244

You can add options to executable code like this

```
plot(1:10)
```





```
n.tot <- nrow(genes)
vals <- table(genes$State)

vals.percent <- vals/n.tot * 100
round(vals.percent, 2)
```

down	unchanging	up
1.39	96.17	2.44

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