

Class05: Data Viz with GGPLOT

Li Ling (A15092789)

Today we are playing with plotting graphics in R.

There are lots of ways to make cool figures in R. There is “base” R graphics (`plot()`,`hist()`, `boxplot()`, etc.)

There is also add-on packages, like `ggplot`,

```
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
```

```
head(cars)
```

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

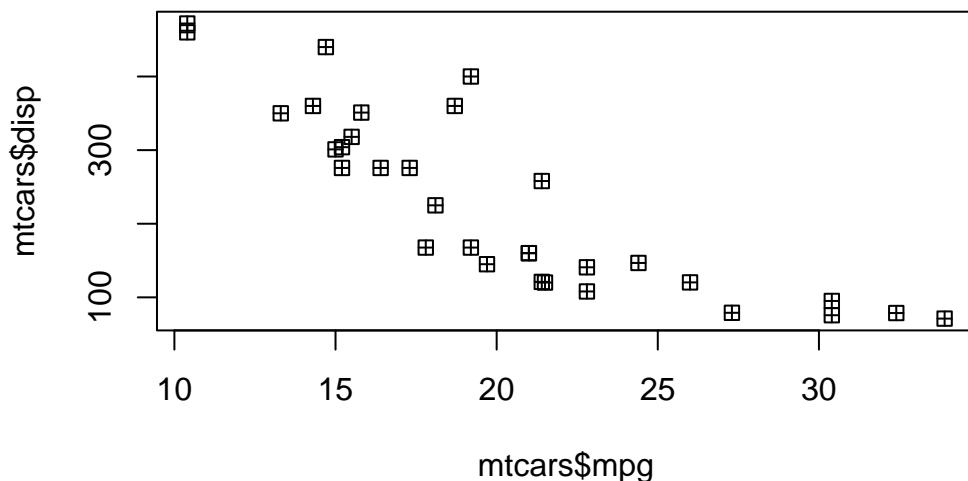
Let's plot this with "base" R:

```
head(mtcars)
```

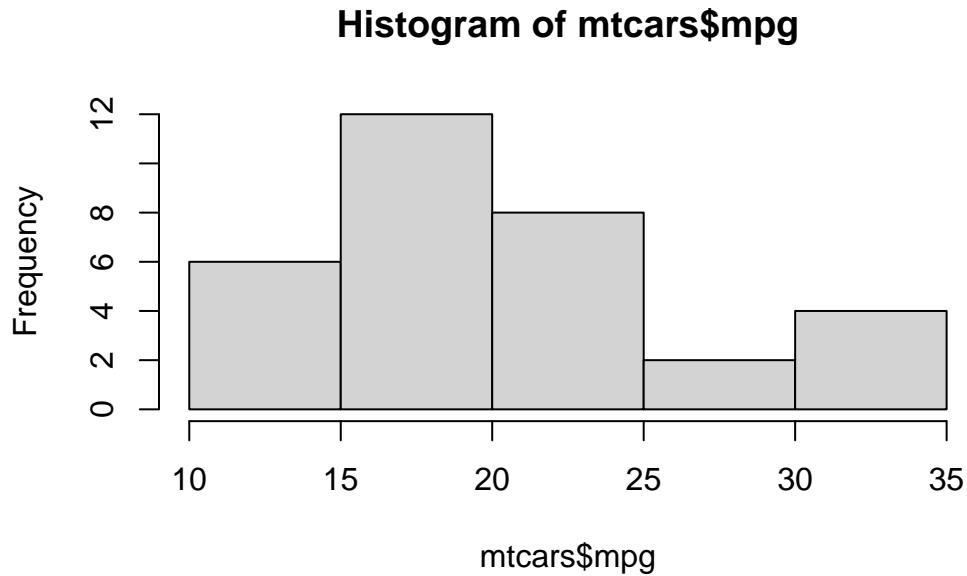
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

Let's plot mpg vs disp

```
plot(mtcars$mpg, mtcars$disp, pch=12)
```



```
hist(mtcars$mpg)
```

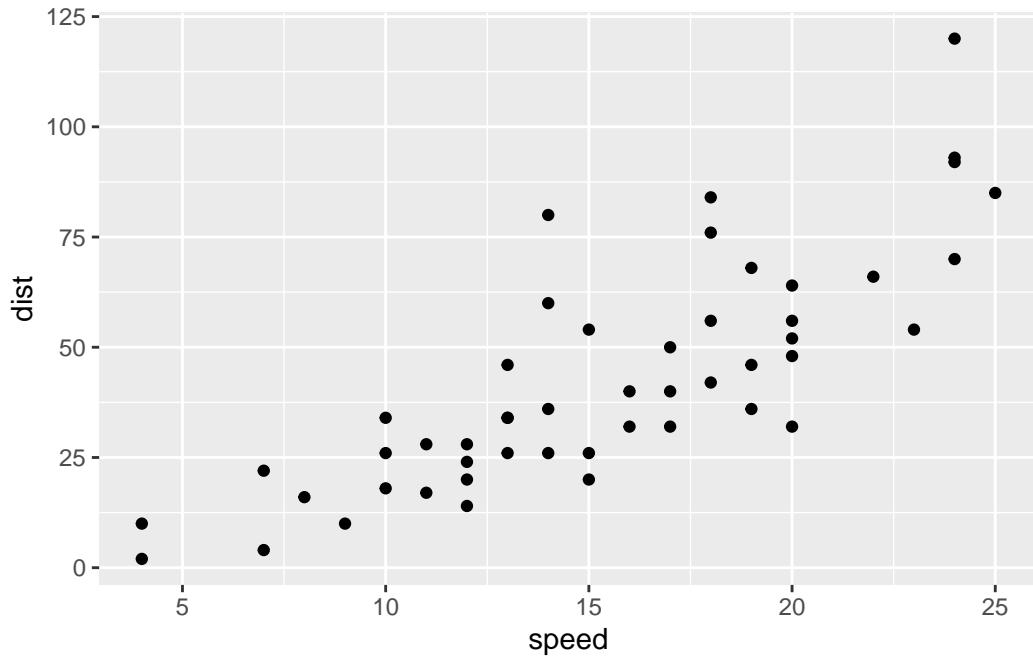


GGPLOT

The main function in the ggplot2 package is `ggplot`. First I need to install the `ggplot2` package. I can install any package with the function `install.packages()`

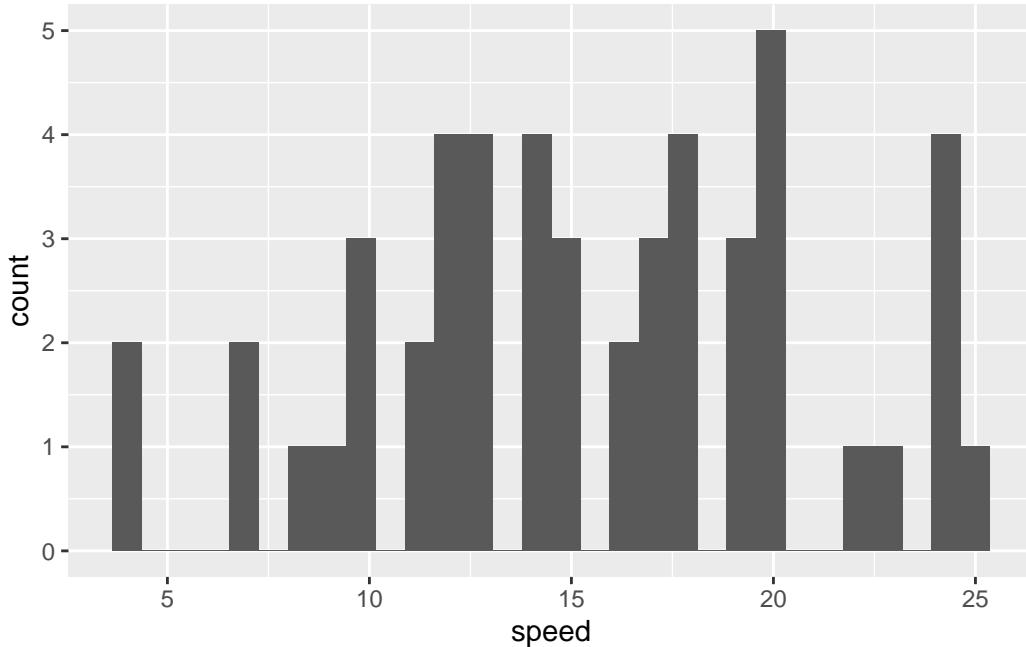
N.B. I never want to run `install.packages()` in my quarto source document!!!

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



```
ggplot(cars) +  
  aes(speed) +  
  geom_histogram()
```

`stat_bin()` using `bins = 30`. Pick better value `binwidth`.



Every ggplot needs at least 3 things:

- The **data** (given with `ggplot(cars)`)
- The **aesthetic** mapping (given with `aes(x=, y=)`)
- The **geom** (given by `geom_point()`)

For simple canned graphs “base” R is nearly always faster

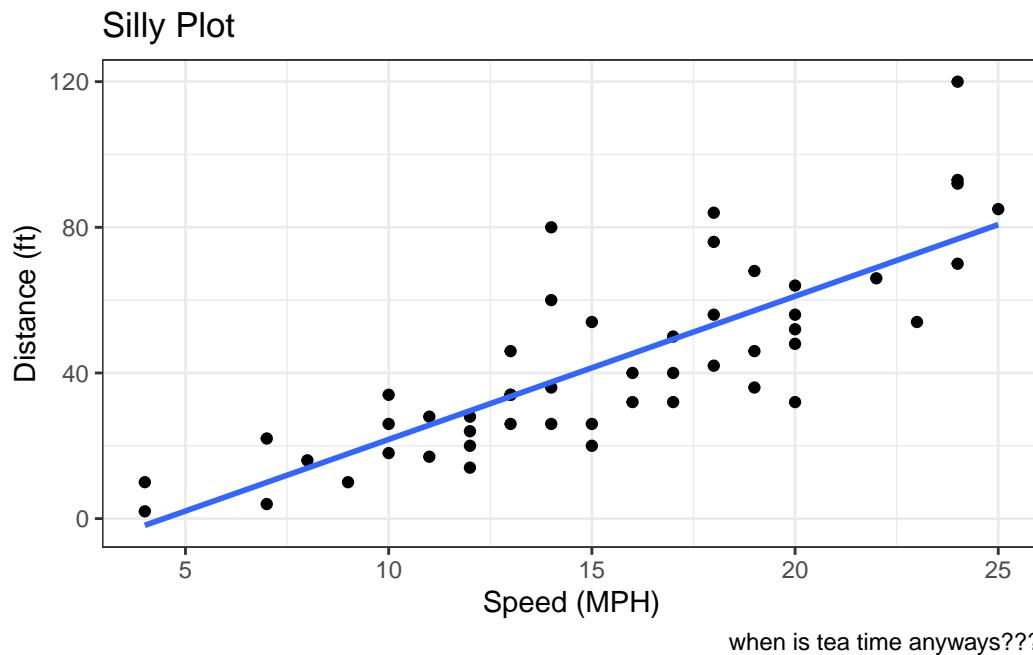
Adding more layers

Let’s add a line, a trend line given `geom_smooth()`, if you want a straight line, given `geom_smooth(method="lm")`, then given `geom_smooth(method="lm", se=FALSE)` which hides the confidence interval shading, showing only the line, let’s add subtitle and caption as well as custom axis labels.

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm", se = FALSE) +
  labs(
    title = "Silly Plot",
    x = "Speed (MPH)",
    y = "Distance (ft)",
```

```
caption = "when is tea time anyways???" +  
theme_bw()
```

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



Plot some expression data

Read data file from online URL

```
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

Q1. How many genes are in this dataset?

There are 5196 genes in this dataset.

Q2. How many “up” regulated genes are there? A==B is A equal B, A!=B is A not equal B, A>B is A greater than B, A>=B is A greater than or equal to B, A<B is A less than B, A<=B is A less than or equal to B.

```
#genes$State == "up" -->This will make the result a logical response  
sum(genes$State == "up")
```

[1] 127

```
#This will give the number of "up" regulated genes
```

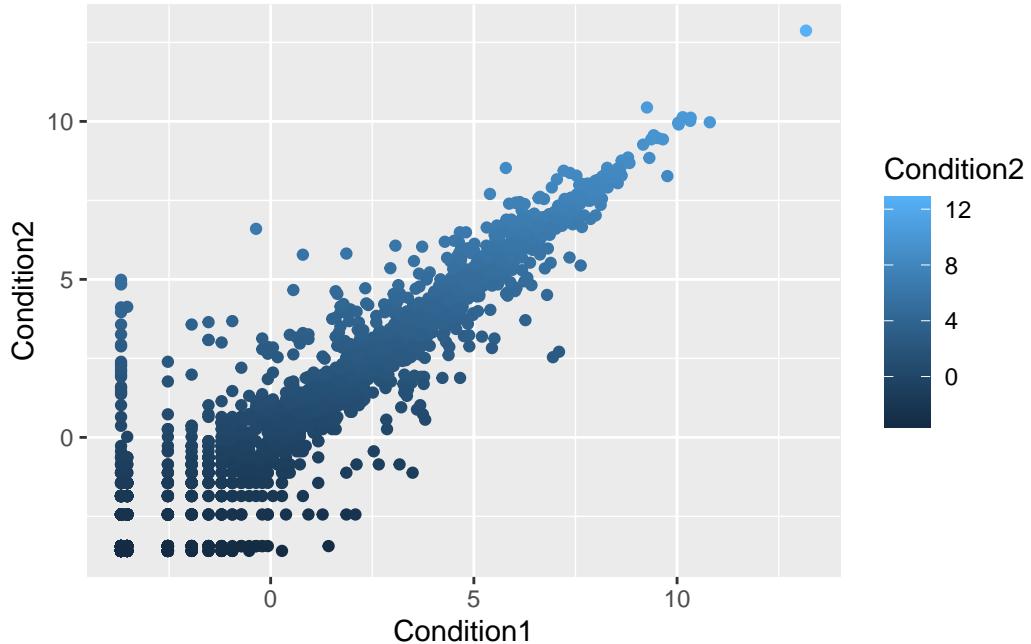
There are 127 “up” regulated genes.

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

	down	unchanging	up
	72	4997	127

Q3. Make a scatter plot of Condition1 vs Condition2, color the points by State, add a title and axis labels, make it a black and white theme.

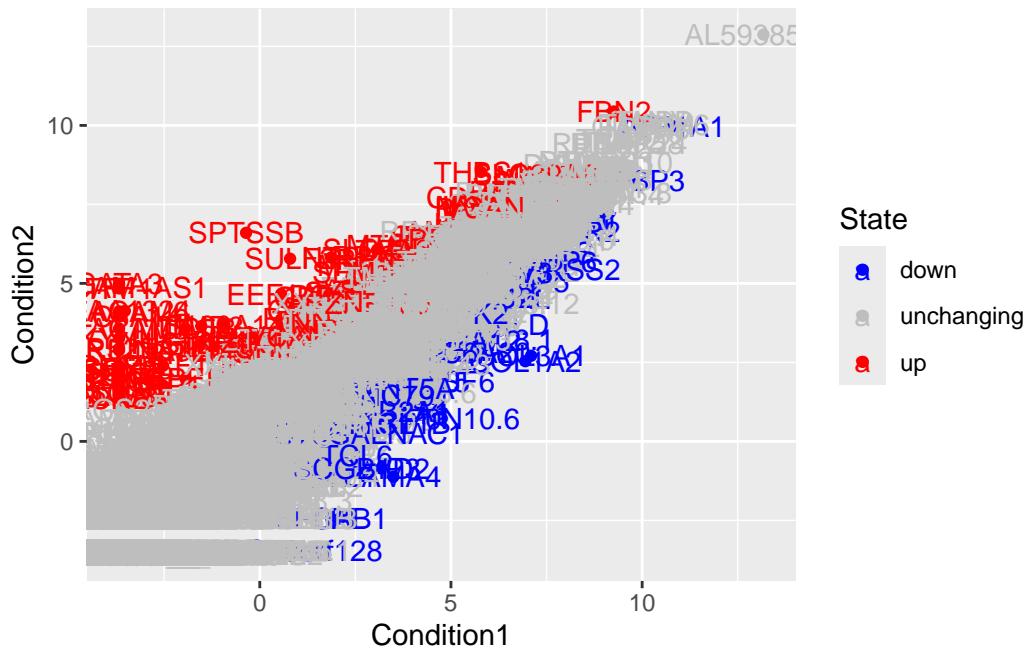
```
ggplot(genes) +  
  aes(Condition1, Condition2, color=Condition2) +  
  geom_point()
```



```
p <- ggplot(genes) +
  aes(Condition1, Condition2, color=State) +
  geom_point() +
  scale_color_manual(values=c("blue","grey","red"))
```

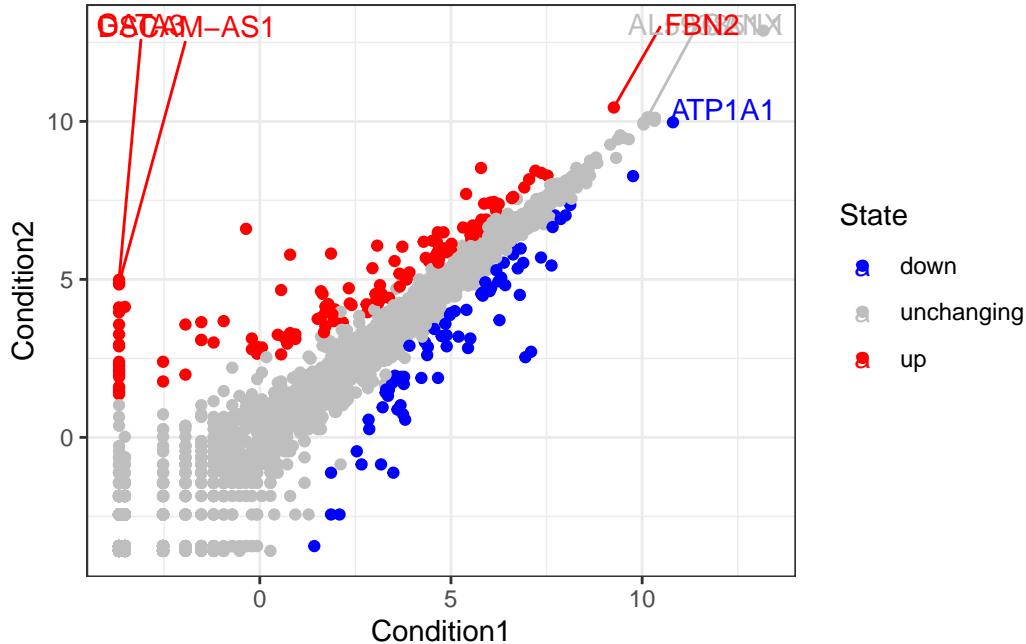
Silly Example of adding labels

```
ggplot(genes) +
  aes(Condition1, Condition2, color=State, label=Gene) +
  geom_point() +
  scale_color_manual(values=c("blue","grey","red")) +
  geom_text()
```



```
library(ggrepel)
ggplot(genes) +
  aes(Condition1, Condition2, color=State, label=Gene) +
  geom_point() +
  scale_color_manual(values=c("blue","grey","red")) +
  geom_text_repel(max.overlaps = 60) +
  theme_bw()
```

Warning: ggrepel: 5190 unlabeled data points (too many overlaps). Consider increasing max.overlaps



Going further

Play with some different layers and the gapminder datasets...

```
# File location online
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.csv"

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

```
#show the first 6 data points
head(gapminder)
```

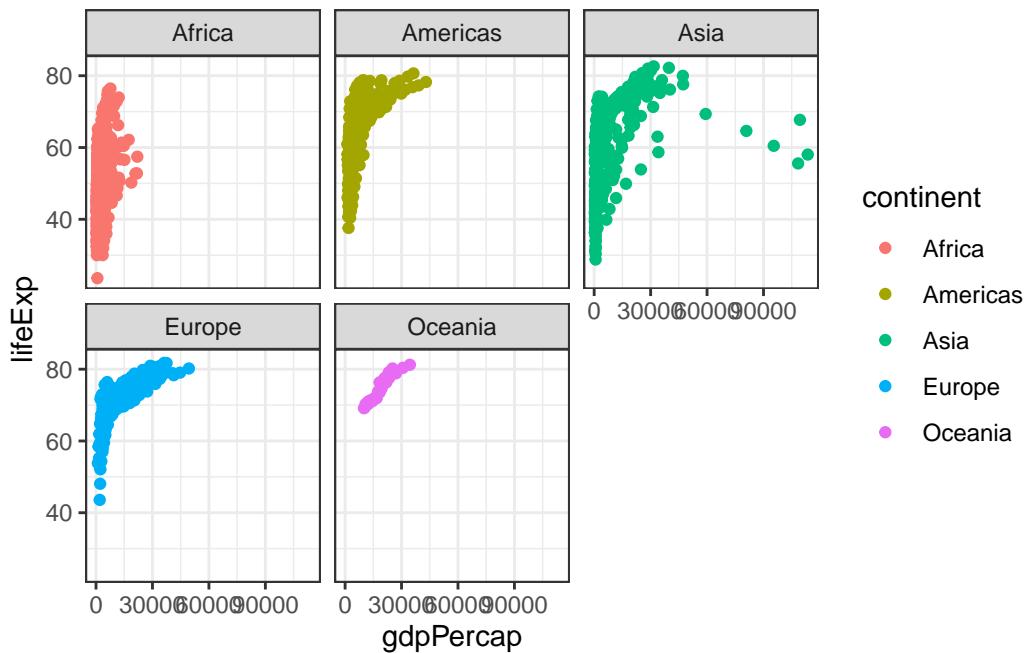
	country	continent	year	lifeExp	pop	gdpPerCap
1	Afghanistan	Asia	1952	28.801	8425333	779.4453
2	Afghanistan	Asia	1957	30.332	9240934	820.8530
3	Afghanistan	Asia	1962	31.997	10267083	853.1007
4	Afghanistan	Asia	1967	34.020	11537966	836.1971
5	Afghanistan	Asia	1972	36.088	13079460	739.9811
6	Afghanistan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

```
#show the last 6 data points
tail(gapminder)
```

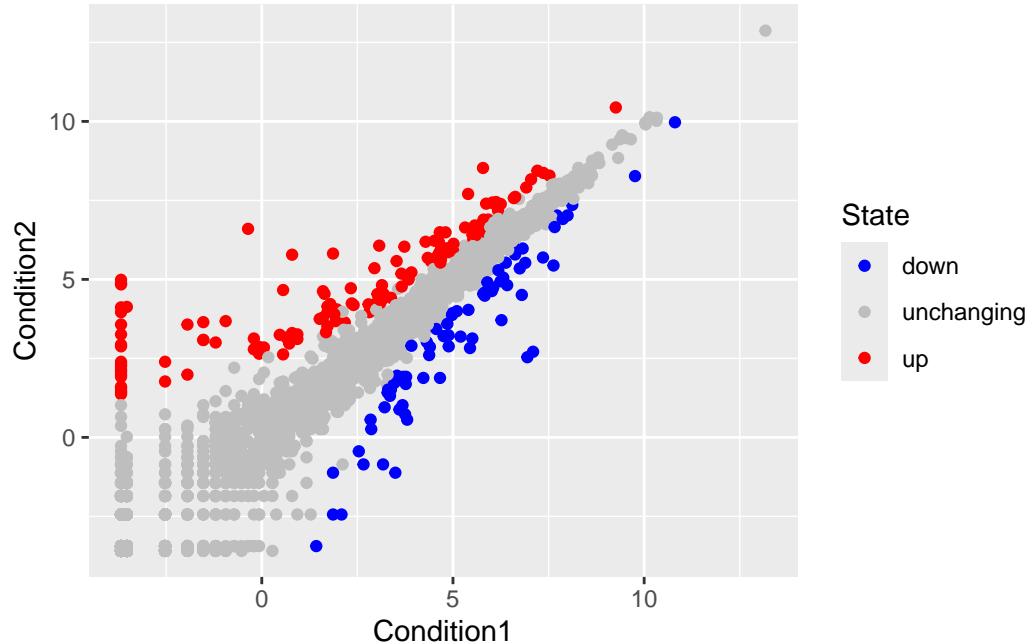
	country	continent	year	lifeExp	pop	gdpPerCap
1699	Zimbabwe	Africa	1982	60.363	7636524	788.8550
1700	Zimbabwe	Africa	1987	62.351	9216418	706.1573
1701	Zimbabwe	Africa	1992	60.377	10704340	693.4208
1702	Zimbabwe	Africa	1997	46.809	11404948	792.4500
1703	Zimbabwe	Africa	2002	39.989	11926563	672.0386
1704	Zimbabwe	Africa	2007	43.487	12311143	469.7093

A first Plot

```
ggplot(gapminder) +
  aes(gdpPerCap, lifeExp, col=continent) +
  geom_point() +
  facet_wrap(~continent) +
  theme_bw()
```

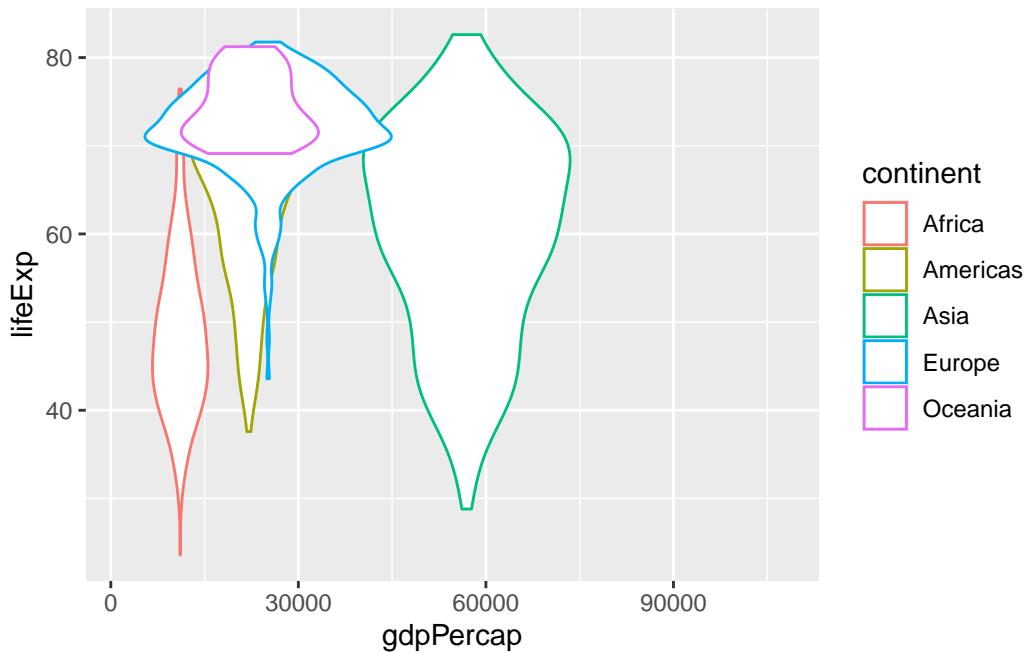


p



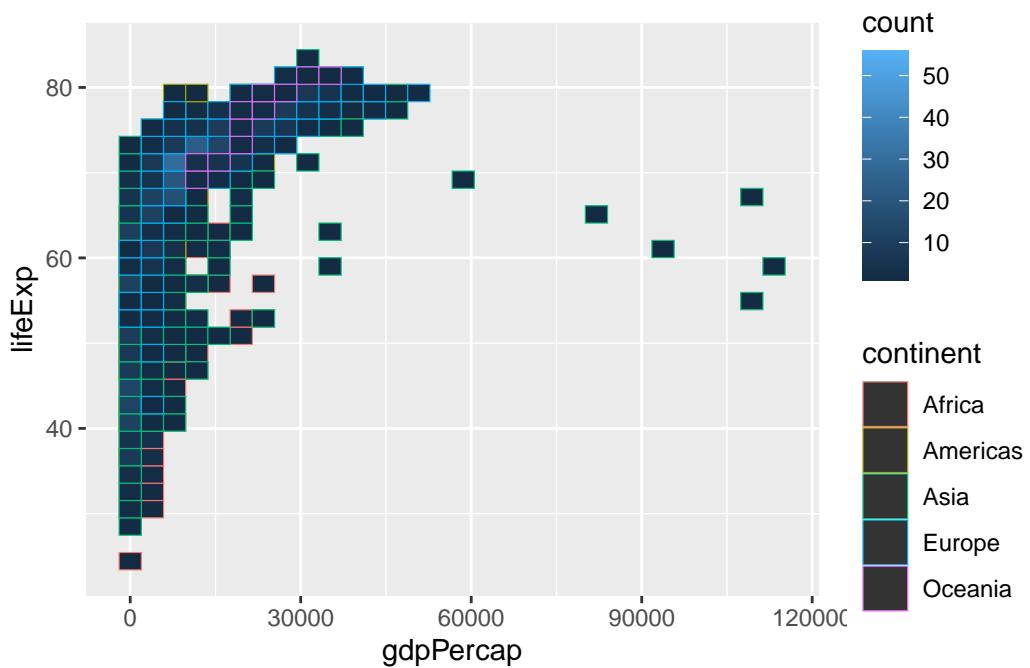
```
ggplot(gapminder) +  
  aes(gdpPerCap, lifeExp, col=continent) +  
  geom_violin()
```

Warning: `position_dodge()` requires non-overlapping x intervals.

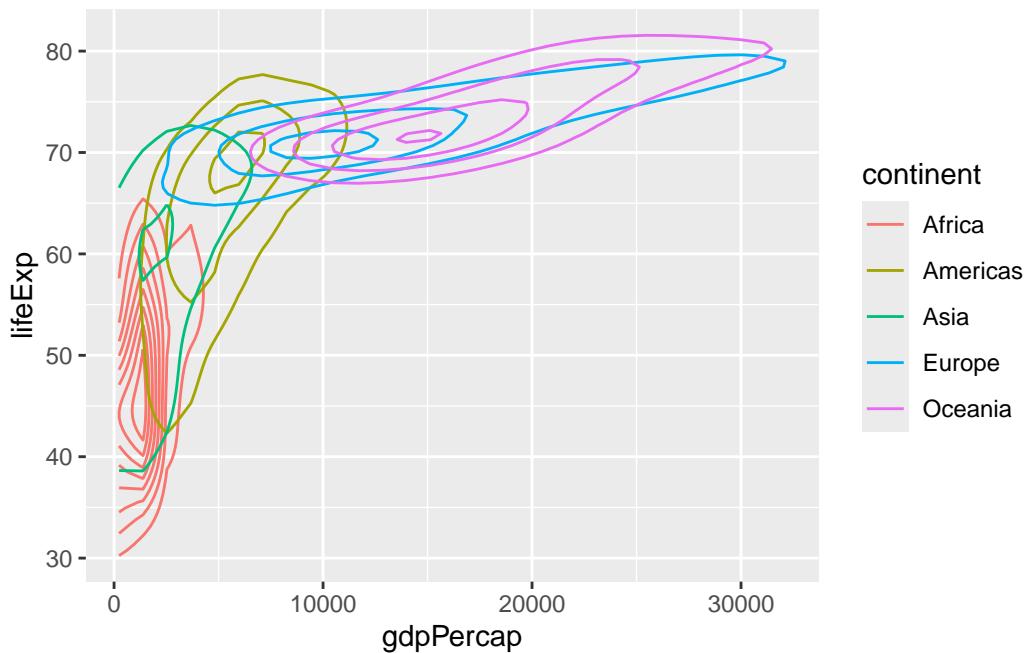


```
ggplot(gapminder) +
  aes(gdpPercap, lifeExp, col=continent) +
  geom_bin2d()
```

`stat_bin2d()` using `bins = 30`. Pick better value `binwidth`.



```
ggplot(gapminder) +
  aes(gdpPercap, lifeExp, col=continent) +
  geom_density2d()
```



```
ggplot(gapminder) +  
  aes(gdpPercap, lifeExp, col=continent) +  
  geom_smooth() +  
  scale_x_log10()
```

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

