

# class 5: Data visualization with ggplot

Yang Liu (A16629623)

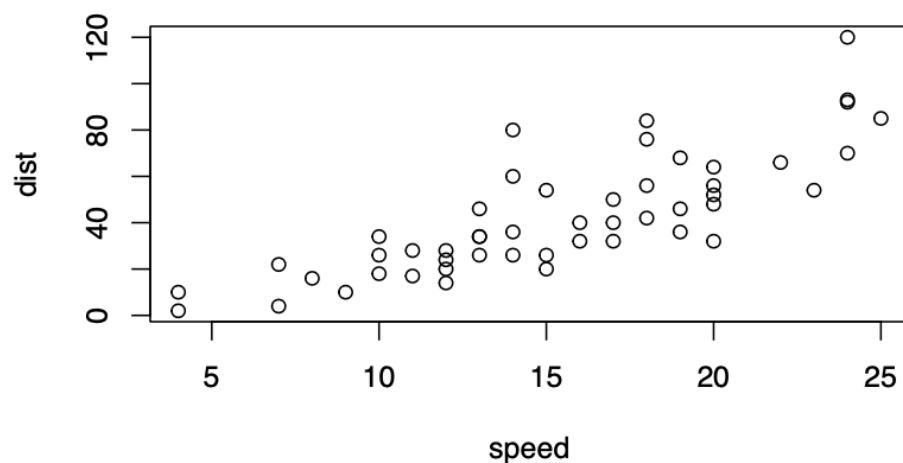
## Table of contents

A more complicated scatter plot . . . . .	4
Exploring their gapmider dataset . . . . .	17

Today we will have our first play with the **ggplot2** package - one of the most popular graphics packages on the planet.

There are many plotting systems in R. These include so called “*base*” plotting/graphics.

```
plot(cars)
```



Base plot is generally rather short code and somewhat dull plots - but it is always there for you and is fast for big datasets.

If i want to use **ggplot2** it takes some more work

```
# ggplot(cars)
```

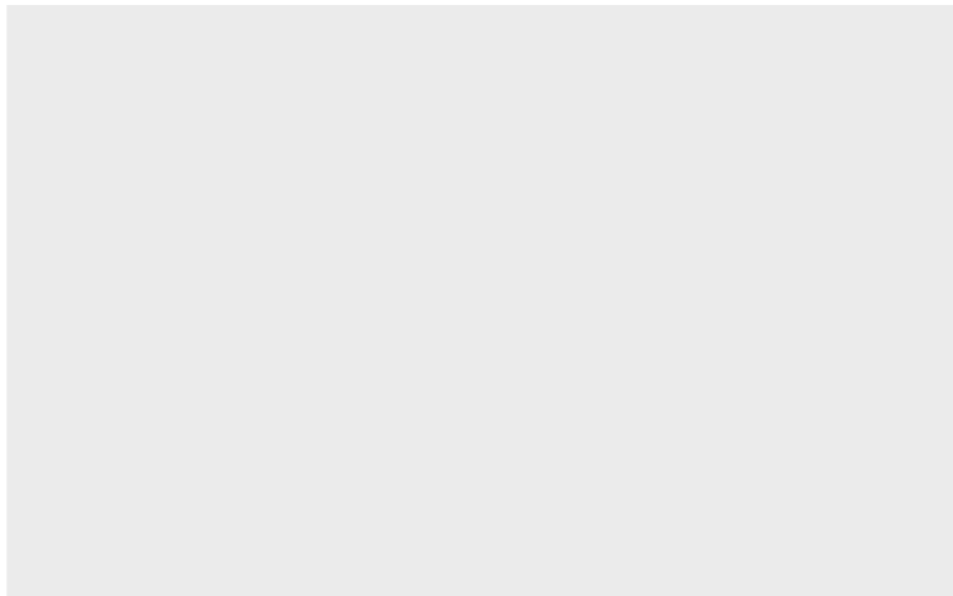
I need to install the package first. To do this I can use the function ‘install.packages(“ggplot2”)’

Every time I want to use a package I need to load it up with a ‘library()’ call.

```
library(ggplot2)
```

Now finally I can use ggplot

```
ggplot(cars)
```



Every ggplot has at least 3 things:

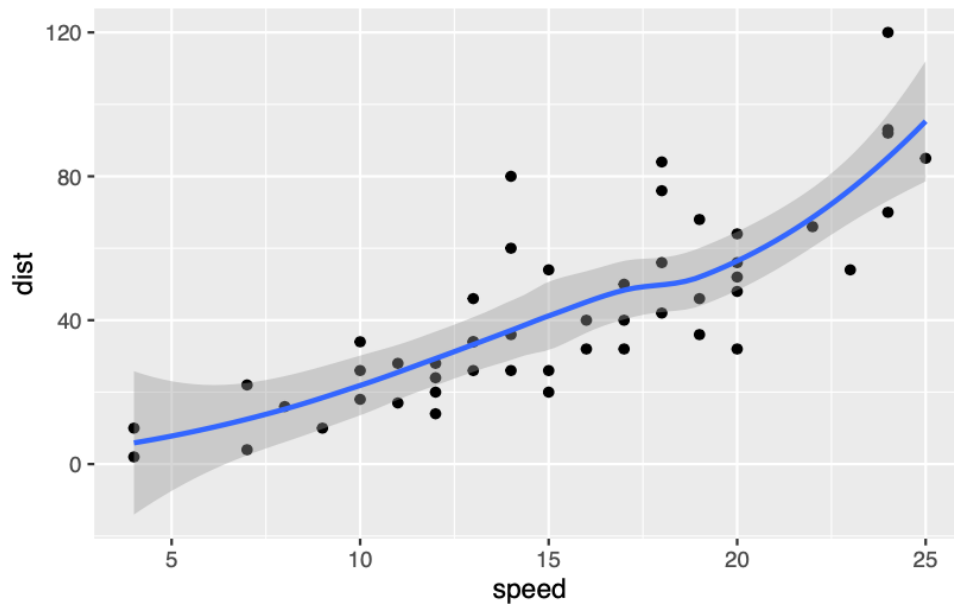
- **data** (the data.frame witht the data you want to plot)
- **aes** (the asthetic mapping of the data to the plot)
- **geom** (how do you want the plot to look, points, lines, columns)

```
head(cars)
```

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

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

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

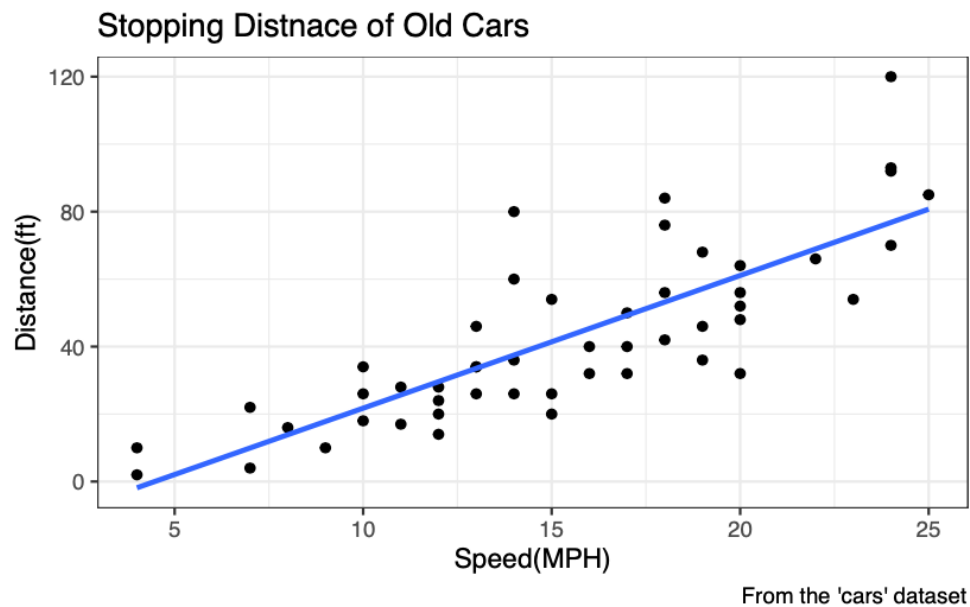


I want a linear model and no standard error bounds shown on my plot.

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

bp + geom_smooth(se=FALSE, method="lm") +
  labs(title = "Stopping Distnace of Old Cars",
       x="Speed(MPH)",
       y="Distance(ft)",
       caption="From the 'cars' dataset") +
  theme_bw()
```

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



## A more complicated scatter plot

Here we make a plot of gene expression data:

```
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

```
genes$State == "up"
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE
[13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



```

[4753] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4765] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4777] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4789] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4801]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4813] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4825] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4837] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4849] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4861] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4873] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
[4885] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4897] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4909] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4921] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4933] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4945] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4957] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4969] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4981] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[4993] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5005] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5017] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
[5029] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5041] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5053] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5065] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5077] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5089] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5101] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5113] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5125] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
[5137] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5149] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5161] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5173] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[5185] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

```

```

round( sum(genes$State == "up") / nrow(genes) * 100, 2)

```

```

[1] 2.44

```

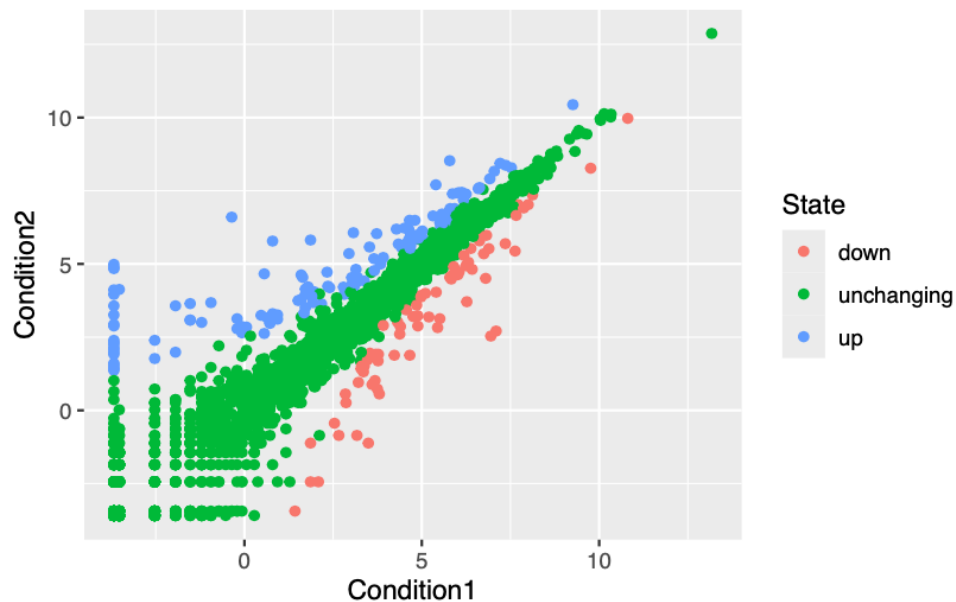
Another way to calculate percentage of total genes is up-regulated in this dataset?

```
n.gene <- nrow(genes)
n.up <- sum(genes$State == "up")
up.present <- n.up/n.gene * 100
round (up.present, 2)
```

```
[1] 2.44
```

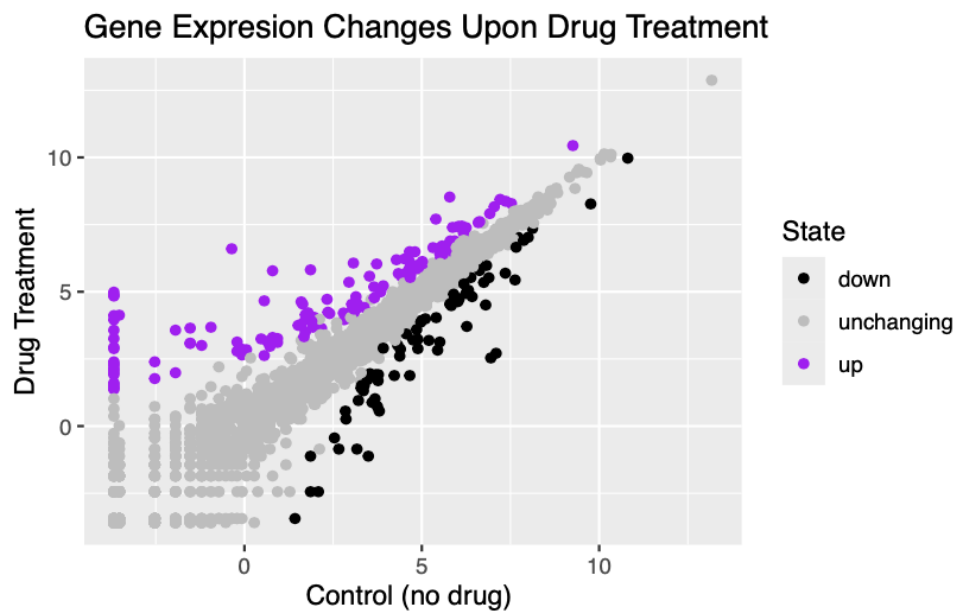
```
bp <- ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State) +
  geom_point()
```

```
bp
```



```
bp + scale_colour_manual( values=c("black","gray","purple") ) +
  labs(title="Gene Expression Changes Upon Drug Treatment",
       x="Control (no drug) ",
       y="Drug Treatment")
```





## Exploring their gapminder dataset

Here we will load up the gapminder dataset to get practice with different aes mappings.

```
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.  
gapminder <- read.delim(url)
```

How many rows and columns can be found in this dataframe?

```
row <- nrow(gapminder)  
column <- ncol(gapminder)  
  
row
```

```
[1] 1704
```

```
column
```

```
[1] 6
```

```
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

Q. How many countries are there in this data set?

```
table(gapminder$year)
```

1952	1957	1962	1967	1972	1977	1982	1987	1992	1997	2002	2007
142	142	142	142	142	142	142	142	142	142	142	142

```
table(gapminder$continent)
```

Africa	Americas	Asia	Europe	Oceania
624	300	396	360	24

I could use the 'unique()' function...

```
unique(gapminder$continent)
```

```
[1] "Asia"      "Europe"    "Africa"    "Americas"  "Oceania"
```

```
length(unique(gapminder$continent))
```

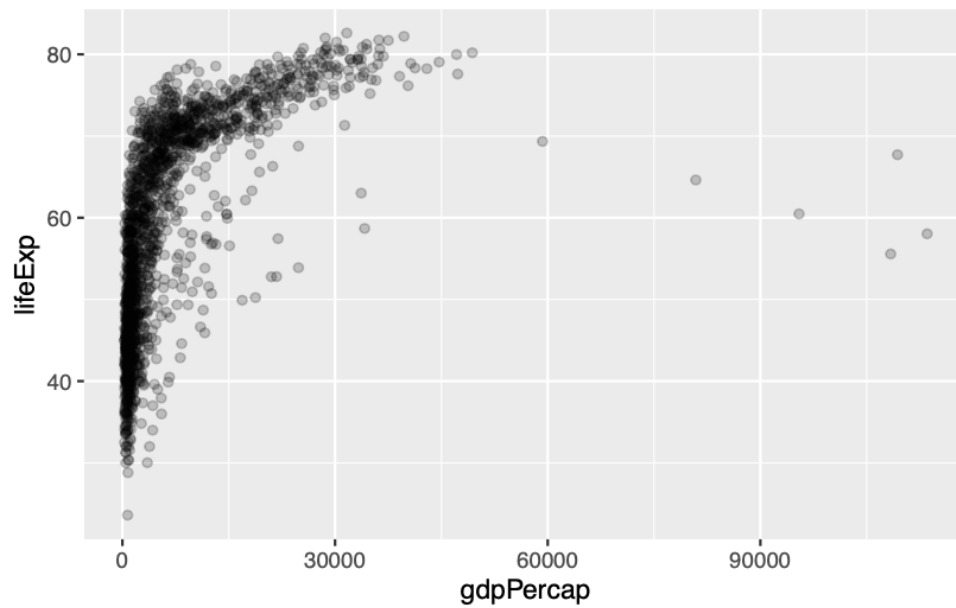
```
[1] 5
```

```
length(unique(gapminder$country))
```

```
[1] 142
```

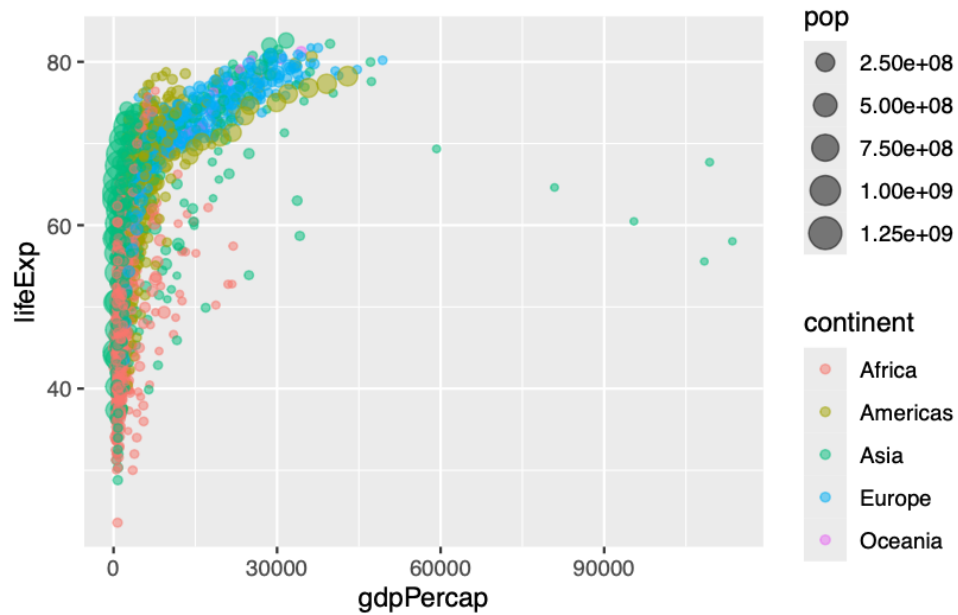
```
library(ggplot2)
```

```
ggplot(gapminder) +  
  aes(x=gdpPercap, y=lifeExp) +  
  geom_point(alpha=0.2)
```



Q

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



```
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

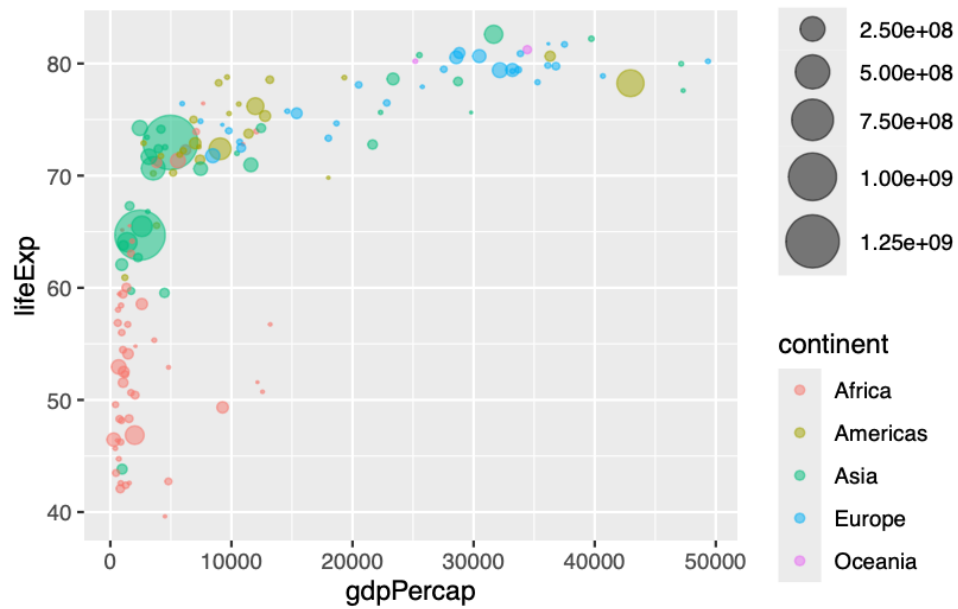
```
gapminder_2007 <- gapminder %>% filter(year==2007)
```

```
head(gapminder_2007)
```

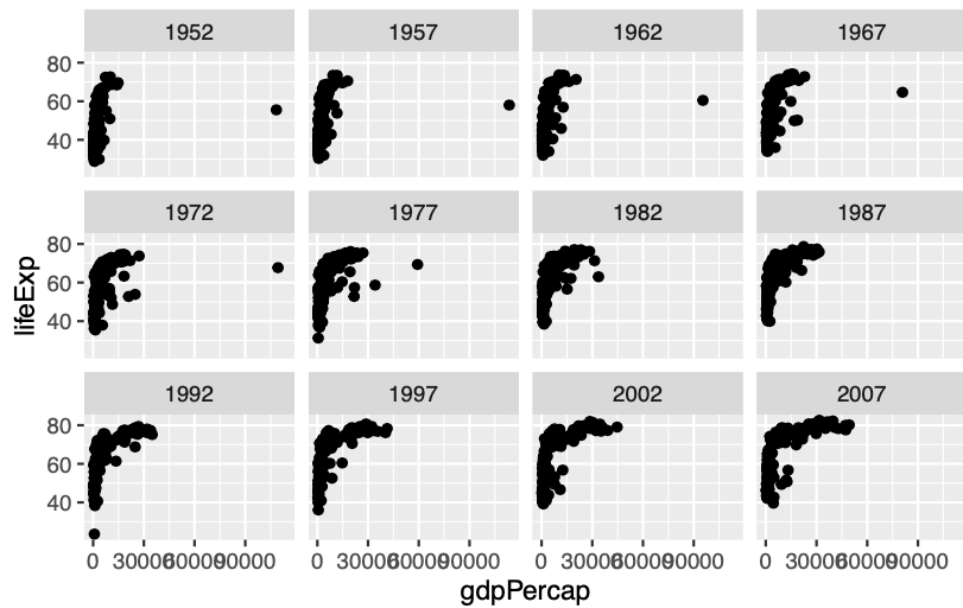
	country	continent	year	lifeExp	pop	gdpPerCap
1	Afghanistan	Asia	2007	43.828	31889923	974.5803
2	Albania	Europe	2007	76.423	3600523	5937.0295

3	Algeria	Africa	2007	72.301	33333216	6223.3675
4	Angola	Africa	2007	42.731	12420476	4797.2313
5	Argentina	Americas	2007	75.320	40301927	12779.3796
6	Australia	Oceania	2007	81.235	20434176	34435.3674

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

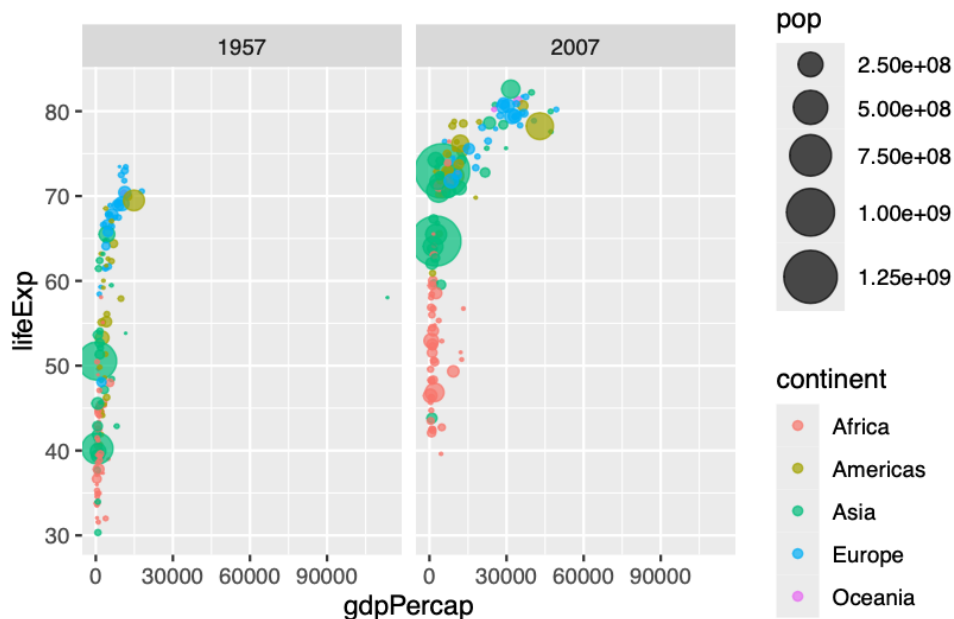


```
ggplot(gapminder) +
  aes(x = gdpPercap, y = lifeExp) +
  geom_point() +
  facet_wrap(~year)
```



```
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)
```



```
sessionInfo()
```

```
R version 4.3.3 (2024-02-29)
```

```
Platform: x86_64-apple-darwin20 (64-bit)
```

```
Running under: macOS Sonoma 14.0
```

```
Matrix products: default
```

```
BLAS: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
```

```
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;
```

```
locale:
```

```
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
```

```
time zone: America/Los_Angeles
```

```
tzcode source: internal
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] dplyr_1.1.4  ggplot2_3.5.0
```

loaded via a namespace (and not attached):

[1] vctrs_0.6.4	nlme_3.1-164	cli_3.6.1	knitr_1.45
[5] rlang_1.1.2	xfun_0.41	generics_0.1.3	jsonlite_1.8.7
[9] labeling_0.4.3	glue_1.6.2	colorspace_2.1-0	htmltools_0.5.7
[13] scales_1.3.0	fansi_1.0.5	rmarkdown_2.25	grid_4.3.3
[17] evaluate_0.23	munsell_0.5.0	tibble_3.2.1	fastmap_1.1.1
[21] yaml_2.3.7	lifecycle_1.0.4	compiler_4.3.3	pkgconfig_2.0.3
[25] mgcv_1.9-1	rstudioapi_0.15.0	farver_2.1.1	lattice_0.22-5
[29] digest_0.6.33	R6_2.5.1	tidyselect_1.2.0	utf8_1.2.4
[33] splines_4.3.3	pillar_1.9.0	magrittr_2.0.3	Matrix_1.6-5
[37] withr_2.5.2	tools_4.3.3	gtable_0.3.4	