

# Class\_5

Max Strul

## Table of contents

Running Code . . . . .	1
<b>Our First Plot</b>	<b>2</b>
<b>Gene Expression!</b>	<b>6</b>
<b>Demographic data from various countries</b>	<b>8</b>
<b>Animations!</b>	<b>14</b>
<b>Session Information</b>	<b>15</b>

## Running Code

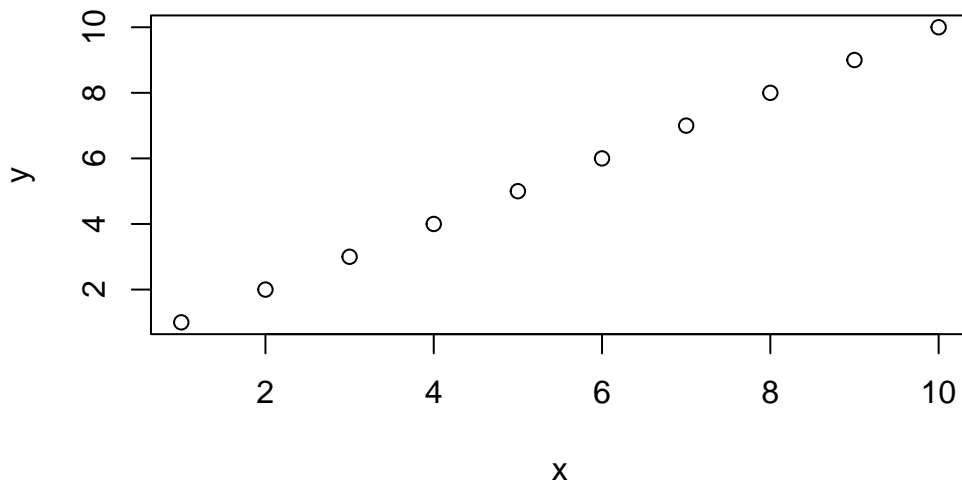
When you click the **Render** (note: Render in bold because of two “\*\*”) button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

[1] 2

You can add options to executable code like this

[1] 4

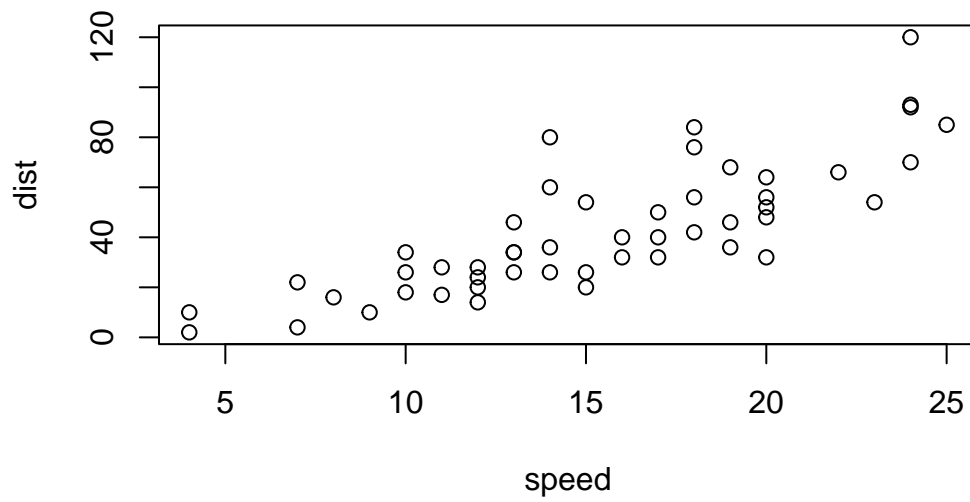


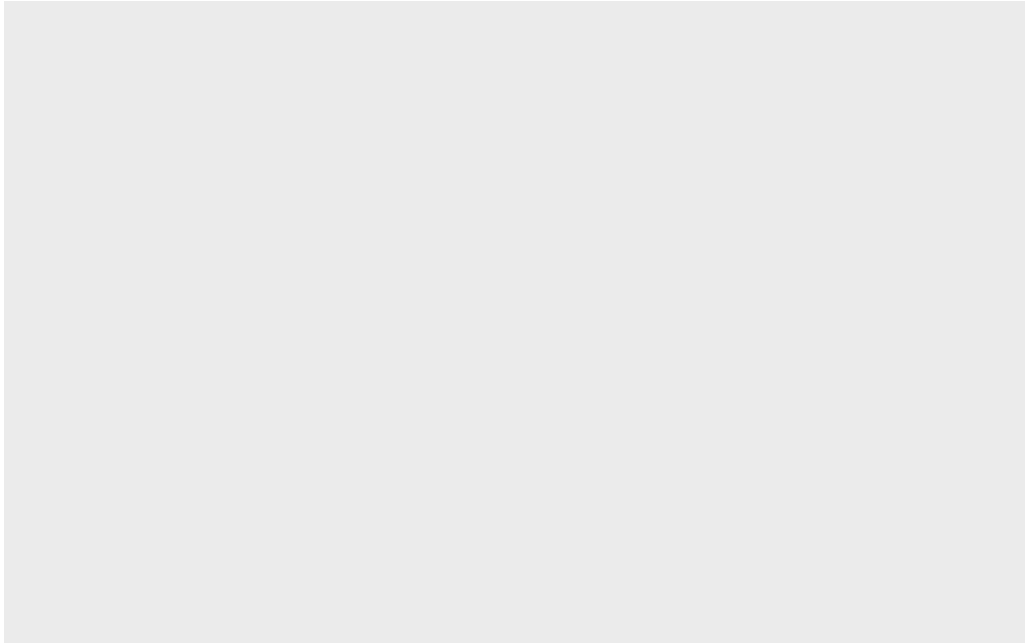
The `echo: false` option disables the printing of code (only output is displayed).

## Our First Plot

**Note:** you can make a new code chunk by pressing `cmd + option + i`

This graph is plotted with `plot()`

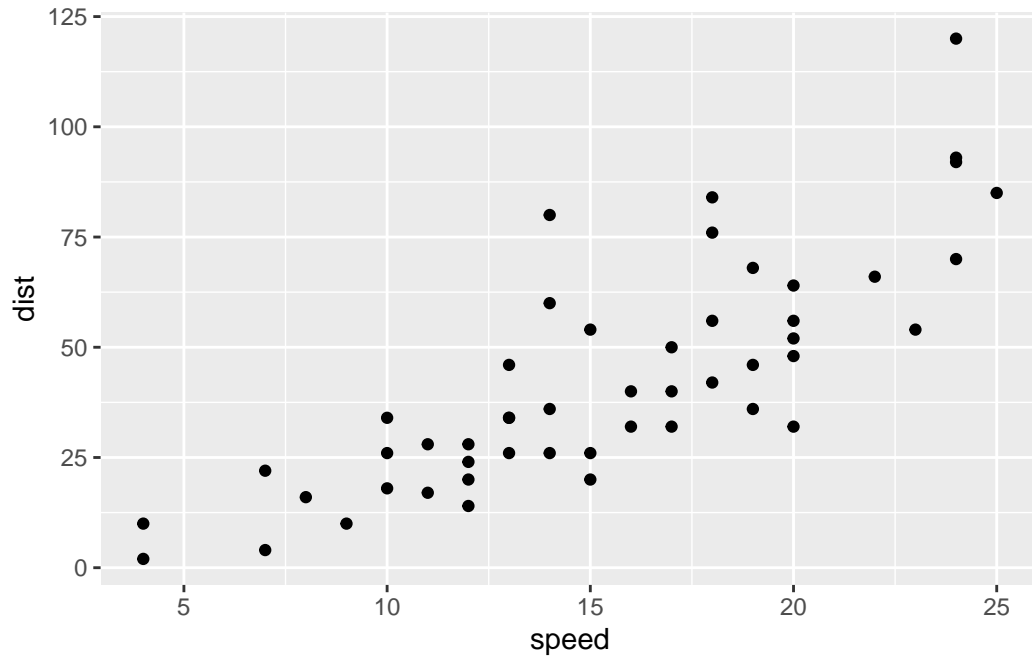




Note: `ggplot(cars)` did not achieve what we wanted. It was a blank canvas. This is because every `ggplot` needs 3 layers:

- Data** (i.e. the data.frame we have),
- Aes** (the aesthetic mapping of our data to what we want to plot),
- Geoms**(How we want to plot this stuff)

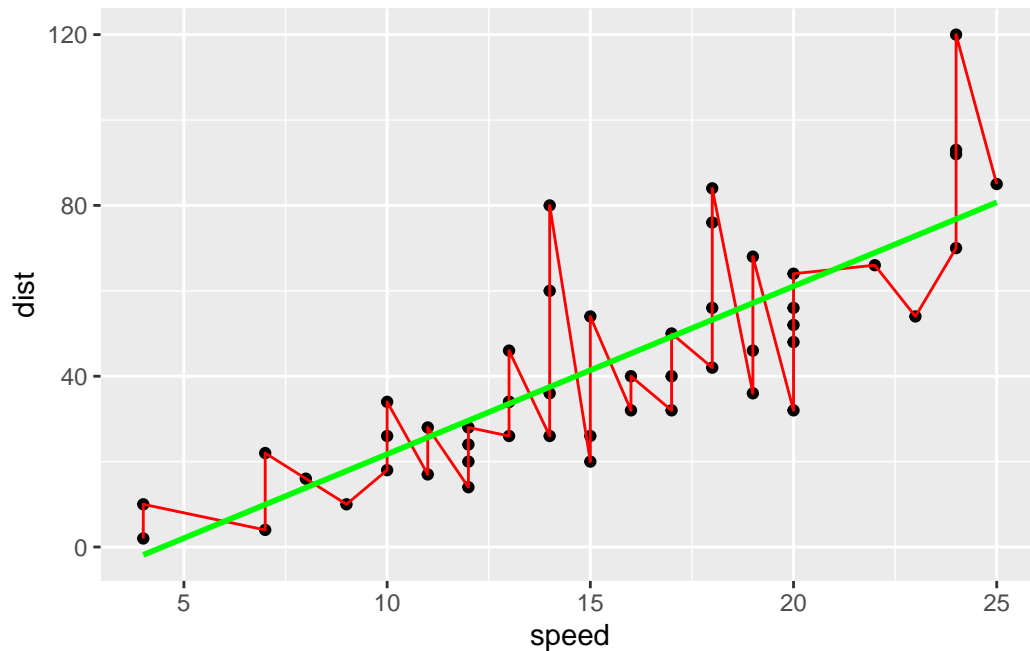
```
ggplot(data=cars)+  
  aes(x=speed,y=dist) +  
  geom_point()
```



Now we can add lines!

```
ggplot(data=cars)+  
  aes(x=speed,y=dist) +  
  geom_point()+  
  geom_line(colour="red")+  
  geom_smooth(method = lm , se = FALSE, colour = "green")
```

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



```
#formula= y~x, colour="blue")
#method = lm means linear
#you can also do formula = y~poly(x,2) for exponential
```

## Gene Expression!

```
library(RColorBrewer)
#library(RColorBrewer)
#display.brewer.all()
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

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

down	unchanging	up
72	4997	127

```
127/(72+4997+127)
```

```
[1] 0.02444188
```

```
g <- ggplot(data=genes)+  
  aes(x=Condition1,y=Condition2,col=State,names=Gene)+  
  geom_point() +  
  scale_color_manual(values=brewer.pal(3, "BrBG"))+  
  ggtitle("Gene Expression Changes Upon Drug Treatment")+  
  xlab("Control (no drug)")+  
  ylab("Drug Treatment")  
  
#library(plotly)  
  
#ggplotly(g, hoverinfo = "Gene")
```

There are 5196 genes in this data set.

The column names are: Gene, Condition1, Condition2, State

There are 4 columns

The complete information for this data frame is: 72, 4997, 127

The total up-regulated genes is: 127

The fraction of up-regulated genes is: 0.0244419

Now we will move onto the next project where we want to begin looking at economic and demographic data from various countries since 1952.

## Demographic data from various countries

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

```
library(gapminder)
```

Attaching package: 'gapminder'

The following object is masked \_by\_ '.GlobalEnv':

gapminder

```
library(patchwork)
```

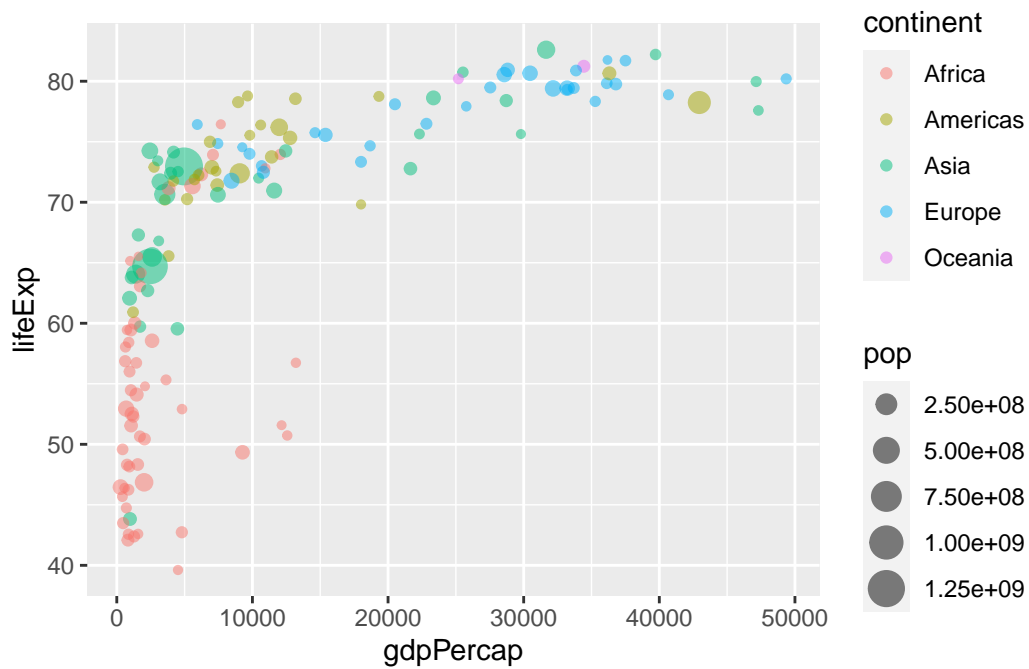
```
gapminder_2007 <- gapminder %>% filter(year==2007)
```

```
head(gapminder_2007)
```

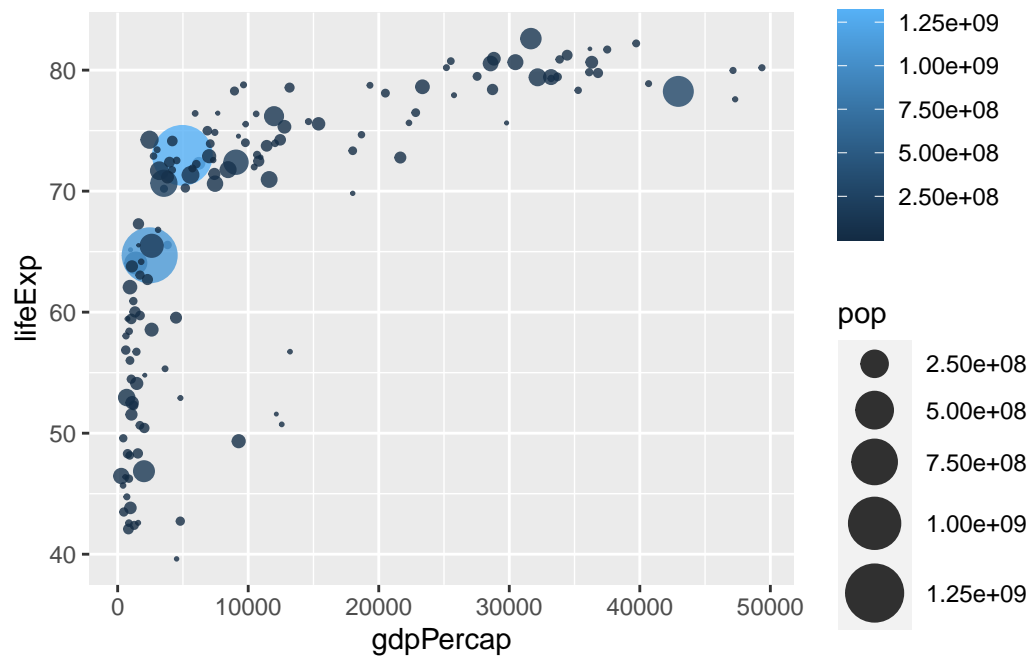
	country	continent	year	lifeExp	pop	gdpPercap
1	Afghanistan	Asia	2007	43.828	31889923	974.5803
2	Albania	Europe	2007	76.423	3600523	5937.0295
3	Algeria	Africa	2007	72.301	33333216	6223.3675
4	Angola	Africa	2007	42.731	12420476	4797.2313
5	Argentina	Americas	2007	75.320	40301927	12779.3796
6	Australia	Oceania	2007	81.235	20434176	34435.3674



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



```
z <- ggplot(gapminder_2007)+
  aes(x=gdpPercap,y=lifeExp,color=pop,size=pop)+
  scale_size_area(max_size=10)+
  geom_point(alpha=0.8)
z
```



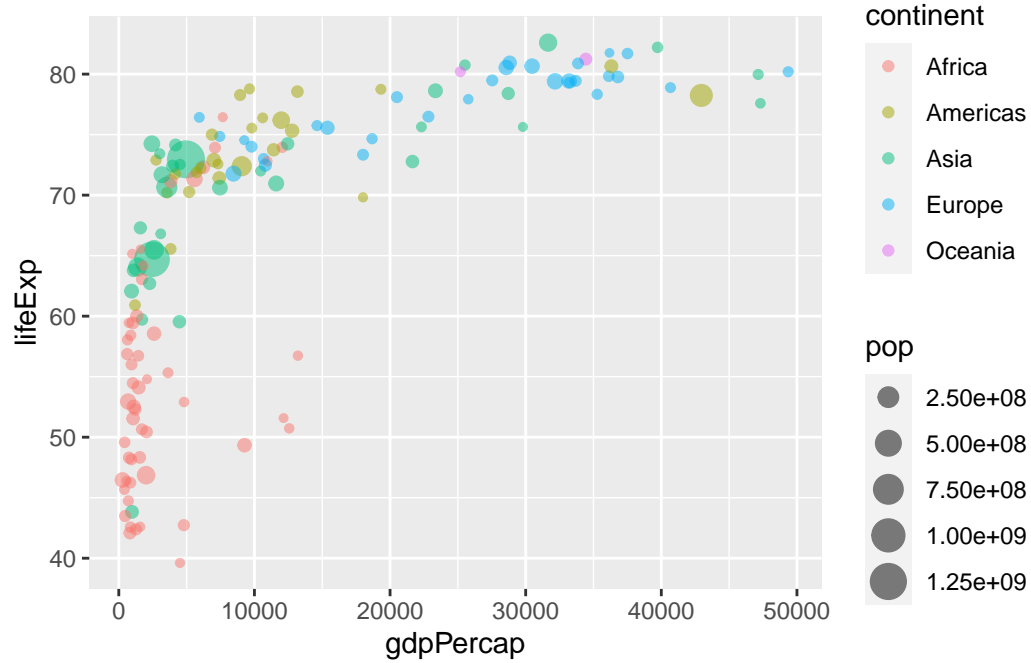
```
x <- ggplot(gapminder_2007)+
  aes(x=gdpPercap,y=lifeExp,color=continent,size=pop)+
  geom_point(alpha=0.5)
  scale_size_area(max_size=10)
```

<ScaleContinuous>

Range:

Limits: 0 -- 1

x

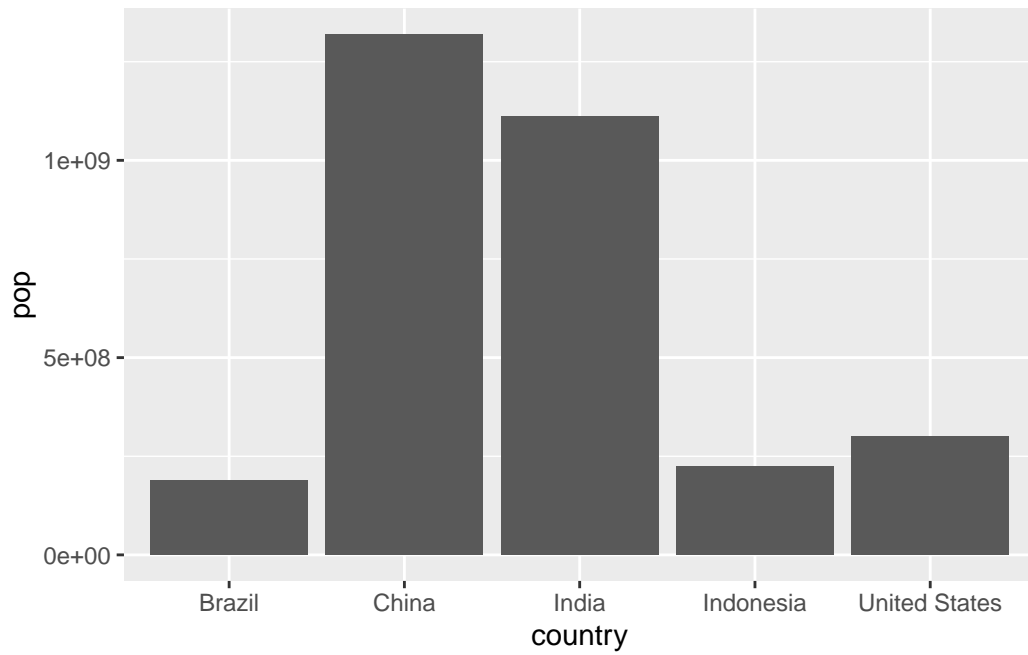


$(y \mid z) / x$

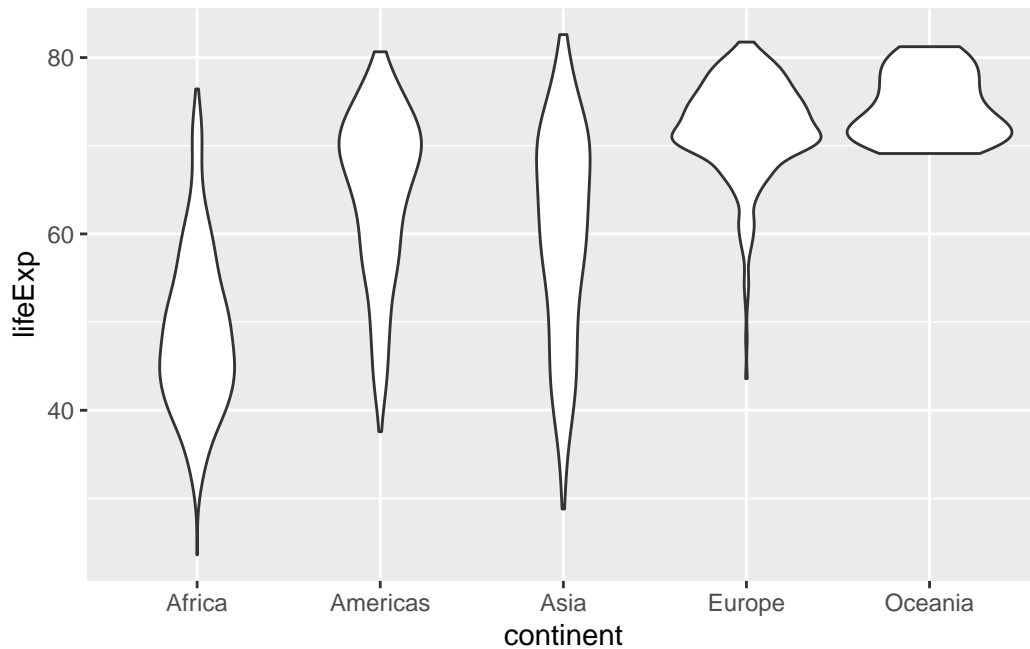


```
library(dplyr)
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)

ggplot(gapminder_top5)+
  geom_col(aes(x=country,y=pop))
```



```
ggplot(gapminder)+  
  aes(continent,lifeExp)+  
  geom_violin()
```



## Animations!

```
#!/ eval: false
#library(gapminder)
#library(gganimate)
#ggplot(gapminder)+
#  aes(gdpPercap, lifeExp, size= pop, col=country)+
#  geom_point(alpha = 0.7, show.legend = FALSE)+
#  scale_colour_manual( values = country_colors)+
#  scale_size(range=c(2,12))+
#  scale_x_log10()+
#  facet_wrap(~continent)+
#Animation
#  labs(title = 'Year: {frame_time}', x= 'GDP per capita', y='life expectancy')+
#  transition_time(year)+
#  shadow_wake(wake_length=0.1, alpha= FALSE)
```

## Session Information

```
sessionInfo()
```

R version 4.2.1 (2022-06-23)

Platform: aarch64-apple-darwin20 (64-bit)

Running under: macOS Monterey 12.5.1

Matrix products: default

BLAS: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib

LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/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

attached base packages:

[1] stats graphics grDevices utils datasets methods base

other attached packages:

[1] patchwork\_1.1.2 gapminder\_0.3.0 dplyr\_1.0.10 RColorBrewer\_1.1-3

[5] ggplot2\_3.3.6

loaded via a namespace (and not attached):

[1] pillar_1.8.1	compiler_4.2.1	tools_4.2.1	digest_0.6.29
[5] lattice_0.20-45	nlme_3.1-157	jsonlite_1.8.2	evaluate_0.17
[9] lifecycle_1.0.3	tibble_3.1.8	gtable_0.3.1	mgcv_1.8-40
[13] pkgconfig_2.0.3	rlang_1.0.6	Matrix_1.4-1	cli_3.4.1
[17] yaml_2.3.5	xfun_0.33	fastmap_1.1.0	withr_2.5.0
[21] stringr_1.4.1	knitr_1.40	generics_0.1.3	vctrs_0.4.2
[25] grid_4.2.1	tidyselect_1.2.0	glue_1.6.2	R6_2.5.1
[29] fansi_1.0.3	rmarkdown_2.16	farver_2.1.1	magrittr_2.0.3
[33] splines_4.2.1	scales_1.2.1	htmltools_0.5.3	colorspace_2.0-3
[37] labeling_0.4.2	utf8_1.2.2	stringi_1.7.8	munsell_0.5.0