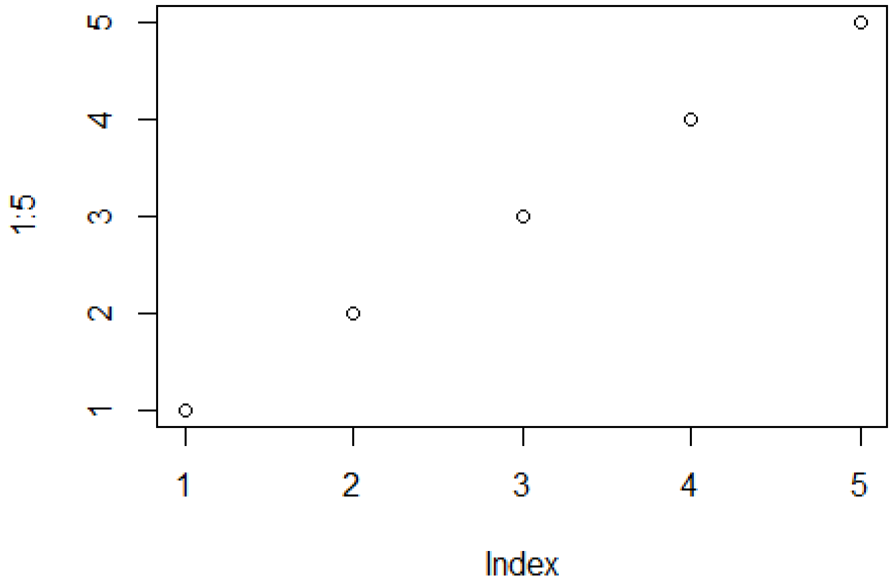
Week 5 Wednesday - Bioinformatics Class

Mirte Kuijpers

February 2nd, 2022

```
#In person class, Wednesday 2nd Feb - Data Visualization

plot(1:5)

#ggplot

##set-up
library("ggplot2")
```

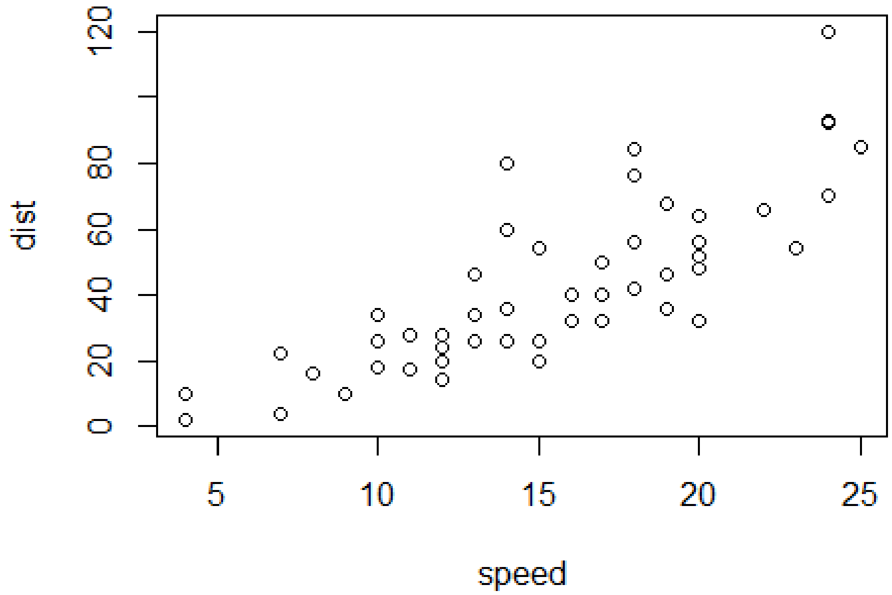


```
## Use cars data
###########################

###investigate data
head(cars)

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

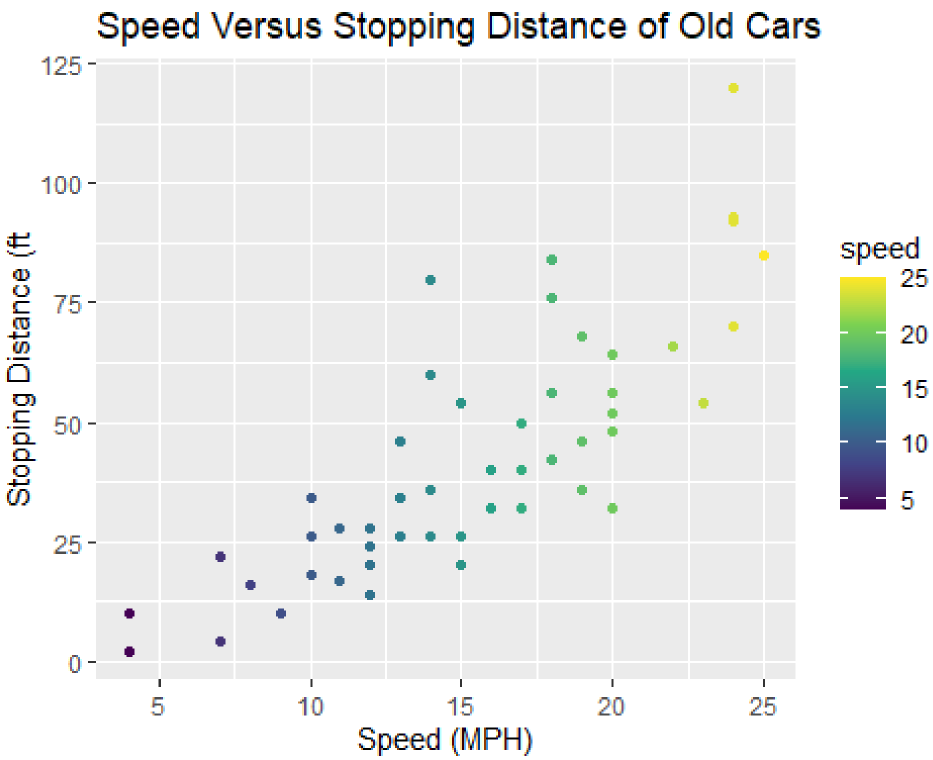
```
str(cars)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
##  $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
```

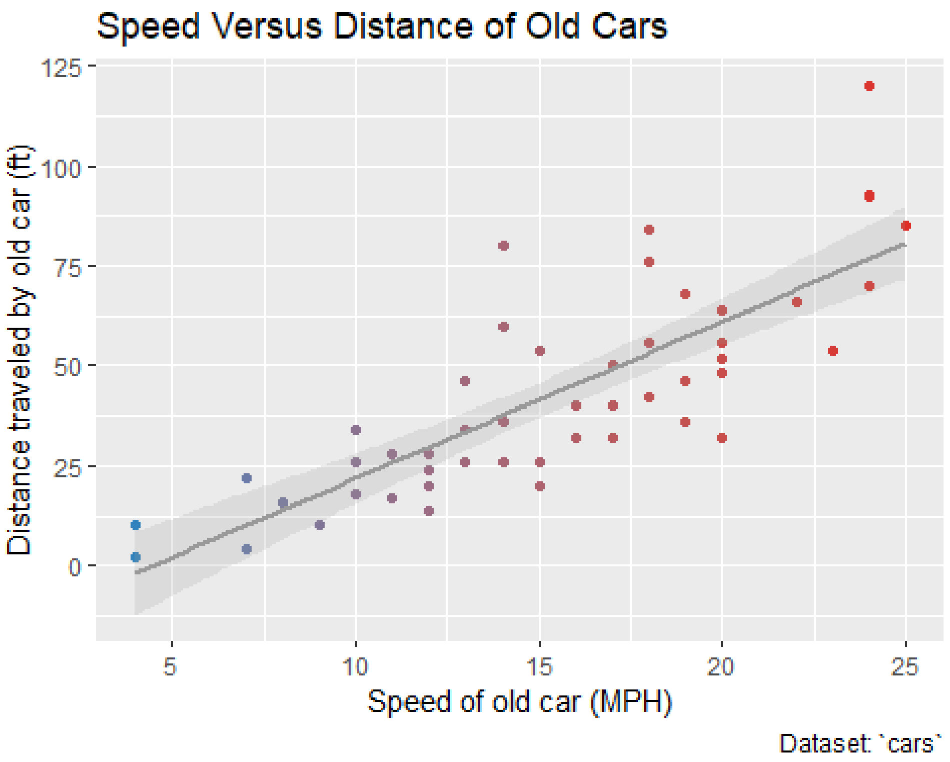
```
###plot data
plot(cars)
```



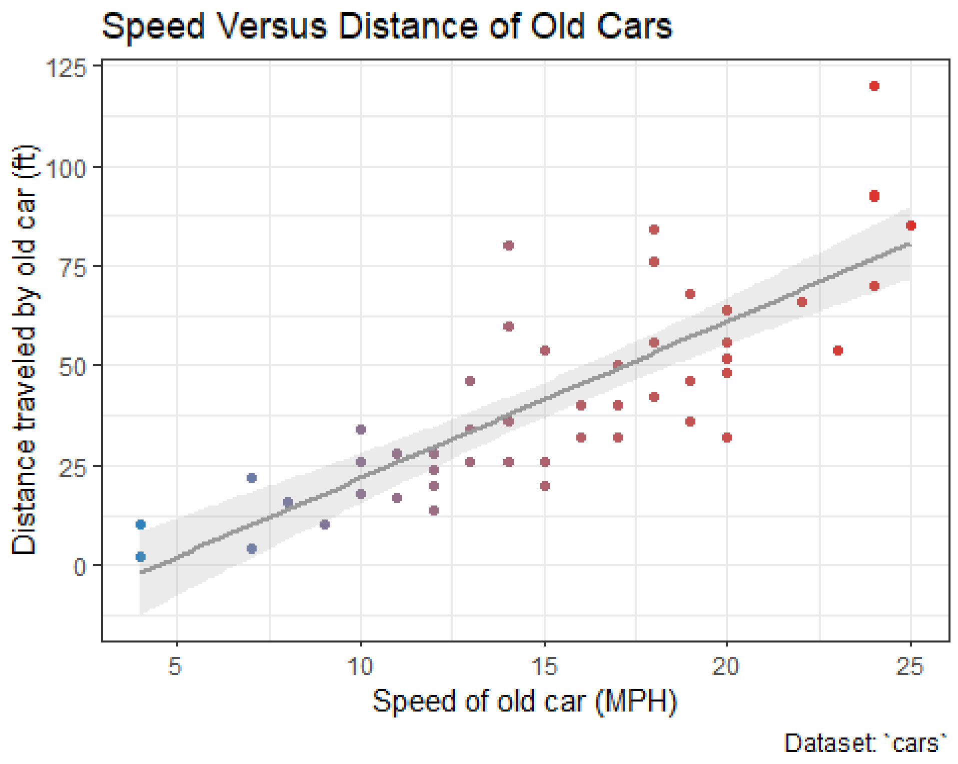
```
####first attempt
ggplot(cars, aes(speed, dist)) +
  geom_point(aes(colour = speed)) +
  labs(title="Speed Versus Stopping Distance of Old Cars", x="Speed (MPH)",
y="Stopping Distance (ft") +
  scale_color_continuous(type = "viridis")
```



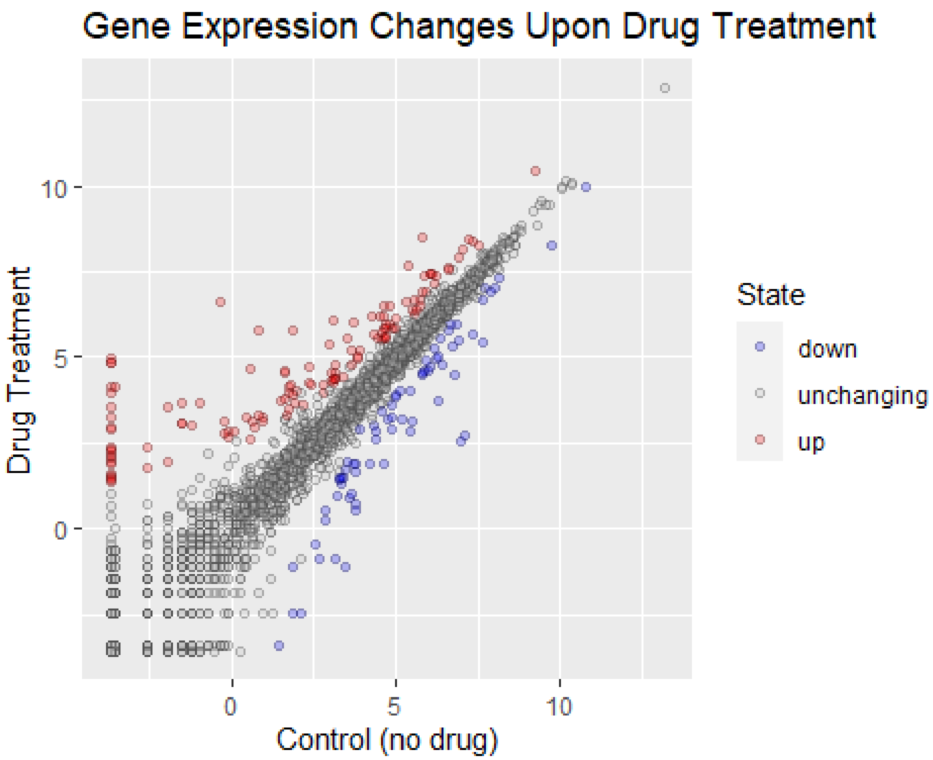
```
  #best to put what you can into the aes() of ggplot rather than into the
geom_'s -> more consistency, only put things specific to different geom_'s
into those

####improve colour scale
basic <- ggplot(cars, aes(speed, dist)) +
  geom_point(aes(colour = speed), show.legend = FALSE) +
  labs(title="Speed Versus Distance of Old Cars", x="Speed of old car (MPH)",
y="Distance traveled by old car (ft)", caption="Dataset: `cars`") +
  scale_colour_gradient(name="Speed", low = "#3182bd", high = "#de2d26",
space = "Lab", na.value = "grey50", guide = "colourbar", aesthetics =
"colour")

####add a trend line
basic + geom_smooth(method="lm", colour = "grey60", alpha=0.2)

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



```
####use the black and white theme as suggested in the tutorial
basic + geom_smooth(method="lm", colour = "grey60", alpha=0.2) + theme_bw()

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

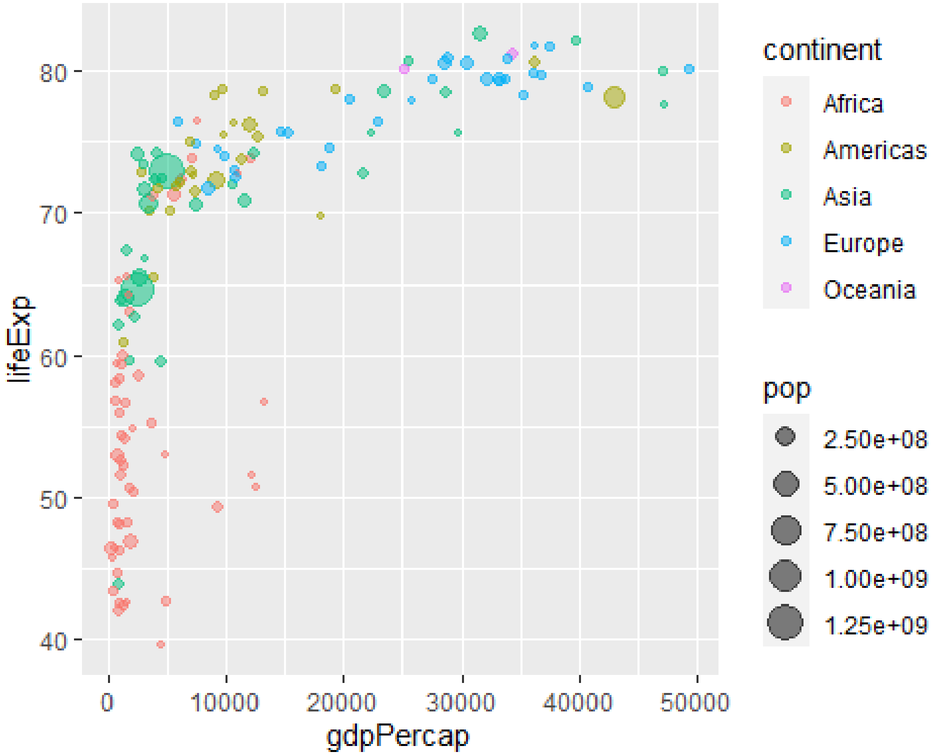


```
###########################
#Gene data (next section)
###########################

##get data
url <- "https://bioboot.github.io/bimm143_S20/class-
material/up_down_expression.txt"
genes <- read.delim(url)

##observe data
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

str(genes)

## 'data.frame':    5196 obs. of  4 variables:
##  $ Gene      : chr  "A4GNT" "AAAS" "AASDH" "AATF" ...
##  $ Condition1: num  -3.681 4.548 3.719 5.078 0.471 ...
##  $ Condition2: num  -3.44 4.39 3.48 5.02 0.56 ...
##  $ State      : chr  "unchanging" "unchanging" "unchanging" "unchanging"
...

nrow(genes) ###number of genes = 5196

## [1] 5196

ncol(genes)

## [1] 4

colnames(genes)

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

table(genes$State)

## 
##       down unchanging         up 
##         72       4997        127

### percentage of genes in each state
round( table(genes$State)/nrow(genes) * 100, 2 )

## 
##       down unchanging         up 
##       1.39      96.17       2.44

##plot data
ggplot(genes, aes(Condition1, Condition2, fill=State)) +
  geom_point(pch=21, alpha=0.25) +
  scale_fill_manual( values=c("blue","gray","red") ) +
  labs(title="Gene Expression Changes Upon Drug Treatment", x="Control (no
drug)", y="Drug Treatment")
```

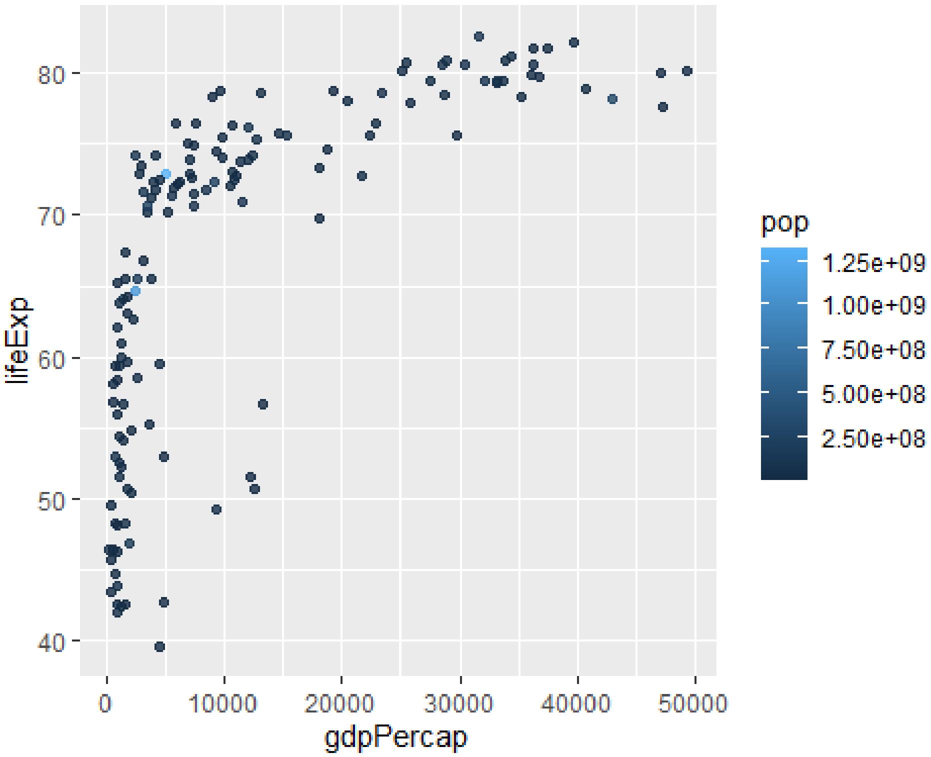


```
#Section 6 (optional)

##further setup for this section, install commented out as only had to be
done the first time
#install.packages("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

## Load data
url <-
"https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapm
inder.tsv"

gapminder <- read.delim(url) #read.delim is a function which reads a file in
table format and creates a data frame from it
```

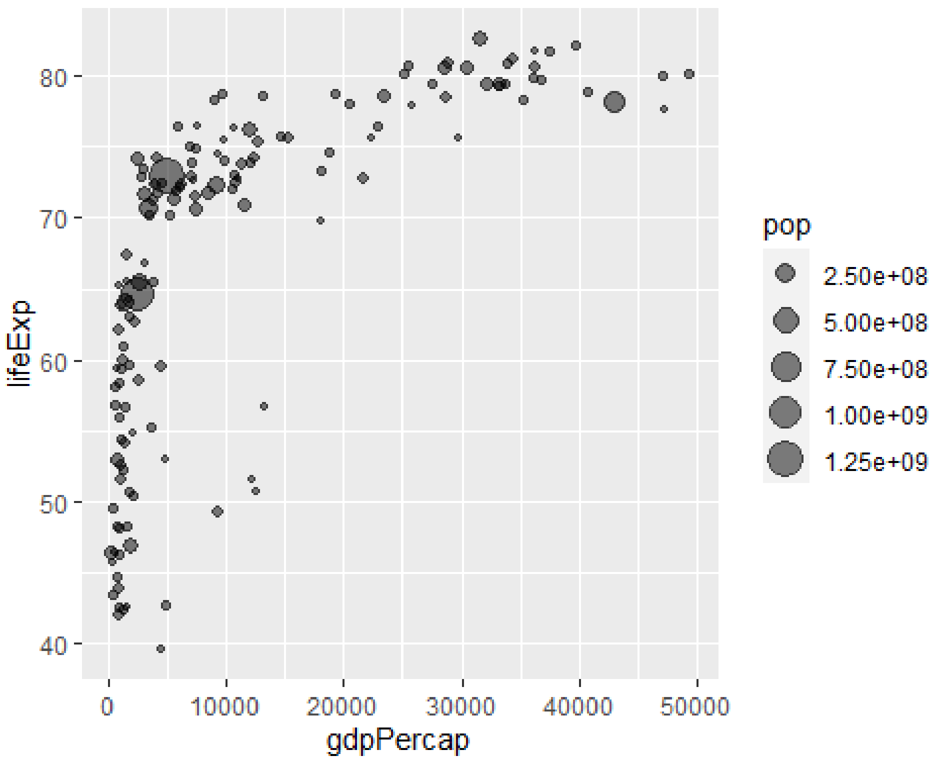
```
###get only the data for 2007
gapminder_2007 <- gapminder %>% filter(year==2007) #%>% passes the left hand side of the operator to the first argument of the right hand side of the operator

##Begin to plot data
ggplot(gapminder_2007, aes(gdpPercap, lifeExp, color=continent, size=pop)) +
  geom_point(alpha=0.5)
```



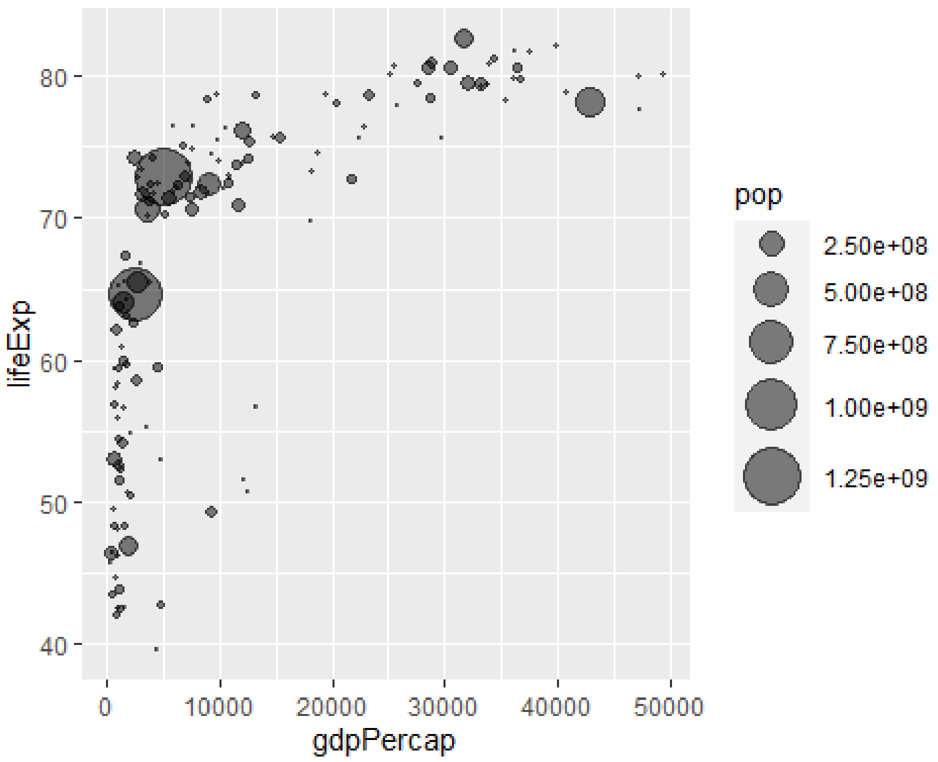
```
###alternative - colour by pop size
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```



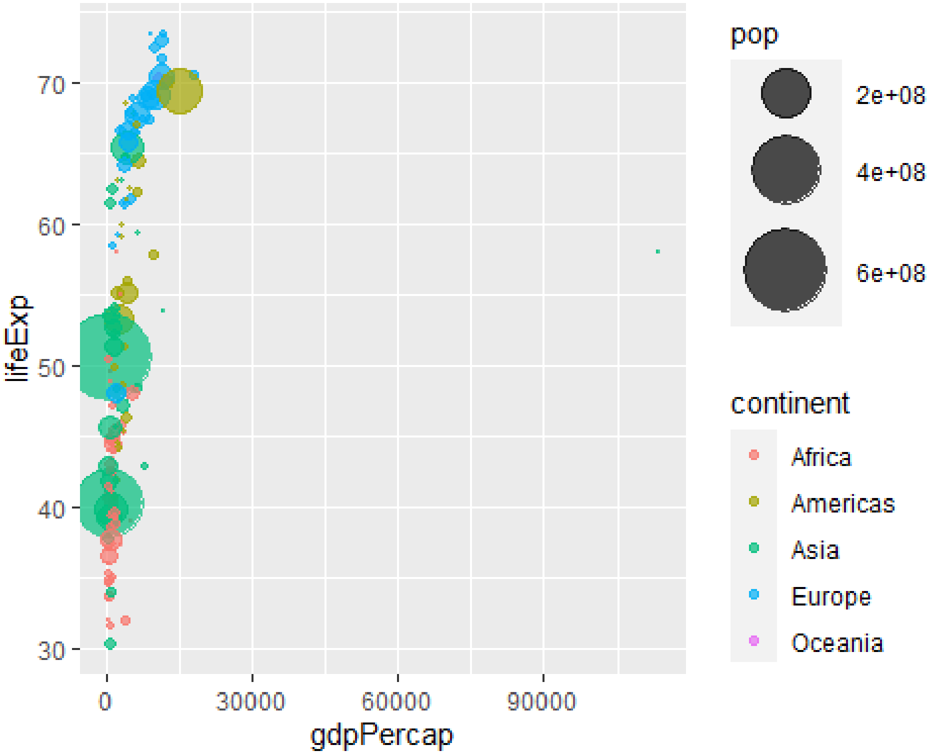
```
###or do size by pop size
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, size = pop) +
  geom_point(alpha=0.5)
```



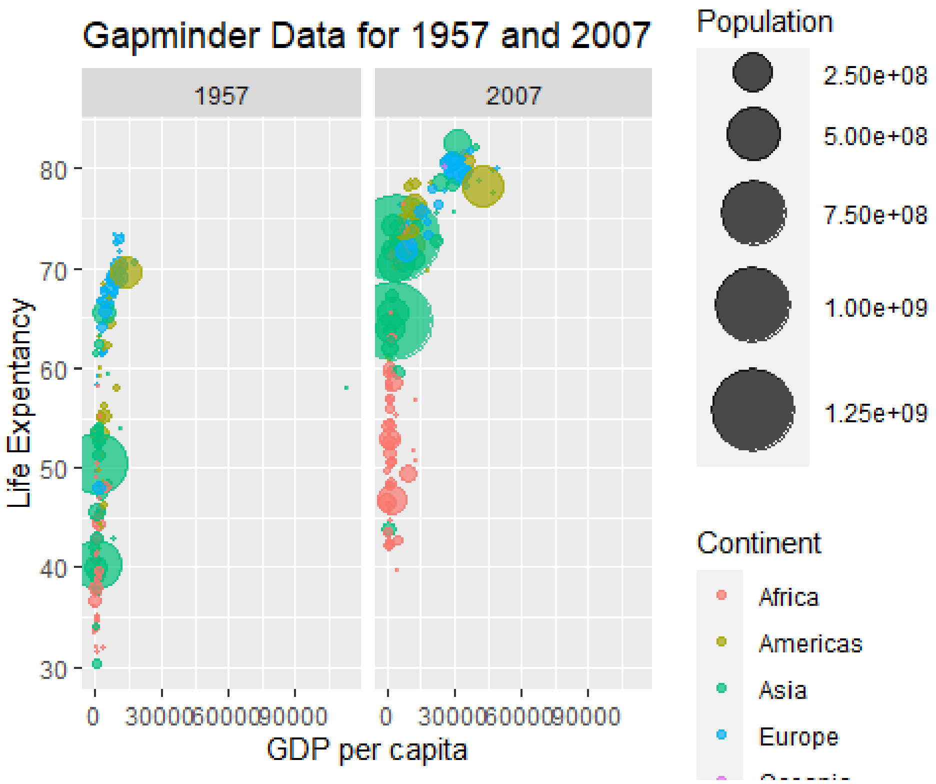
```
####but that doesn't work properly, area doesn't scale as it should so use
below instead
ggplot(gapminder_2007) +
  geom_point(aes(x = gdpPercap, y = lifeExp,
                 size = pop), alpha=0.5) +
  scale_size_area(max_size = 10)
```



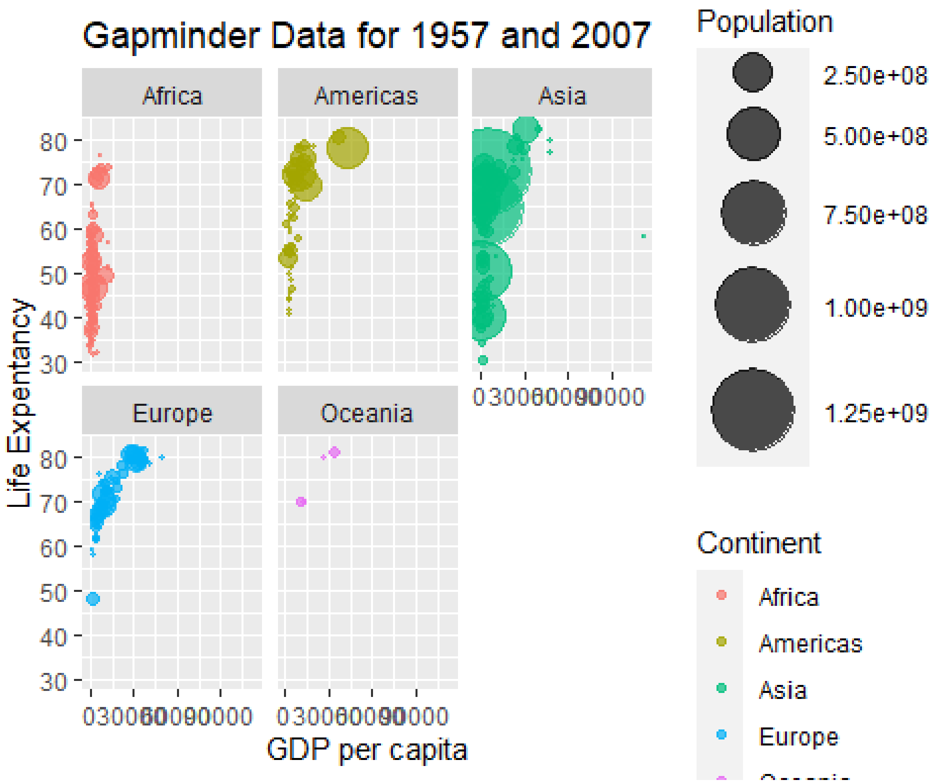
```r
##Actual question
###get 1957 as well
gapminder_1957 <- gapminder %>% filter(year==1957)

##plot
ggplot(gapminder_1957) +
  geom_point(aes(x = gdpPercap, y = lifeExp,
                 size = pop, colour=continent), alpha=0.7) +
  scale_size_area(max_size = 15)
```



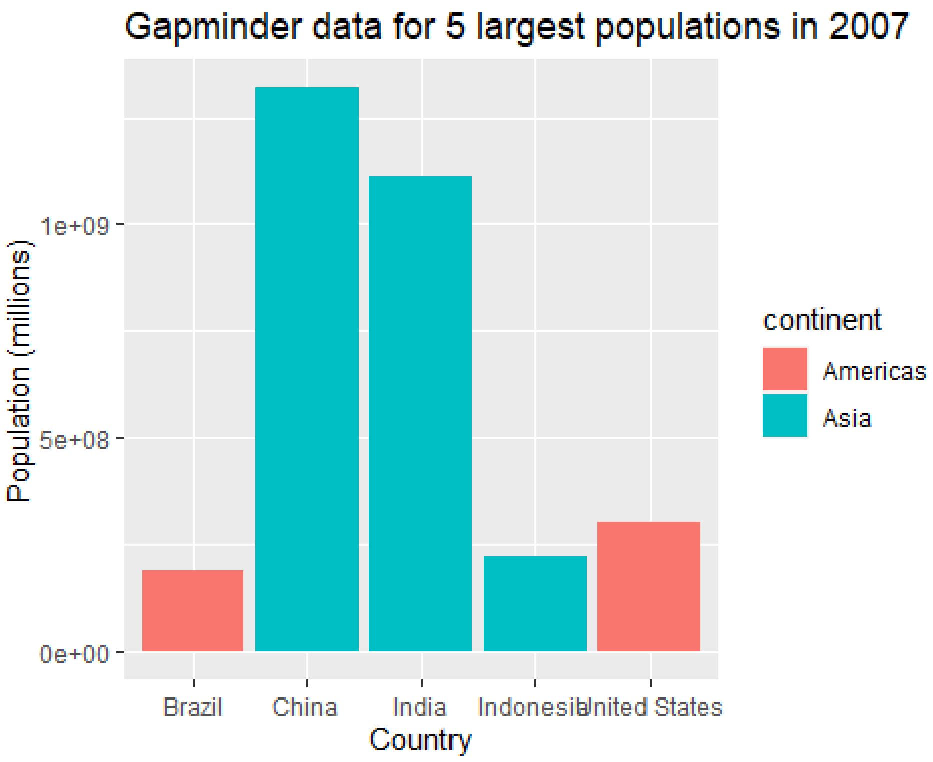
```
##1957 and 2007
gapminder_merge <- rbind(gapminder_1957, gapminder_2007)

ggplot(gapminder_merge) +
  geom_point(aes(x = gdpPercap, y = lifeExp,
                 size = pop, colour=continent), alpha=0.7) +
  scale_size_area(max_size = 15) +
  facet_wrap(~year) +
  labs(title="Gapminder Data for 1957 and 2007", x="GDP per capita", y="Life
Expentancy", colour="Continent", size="Population")
```



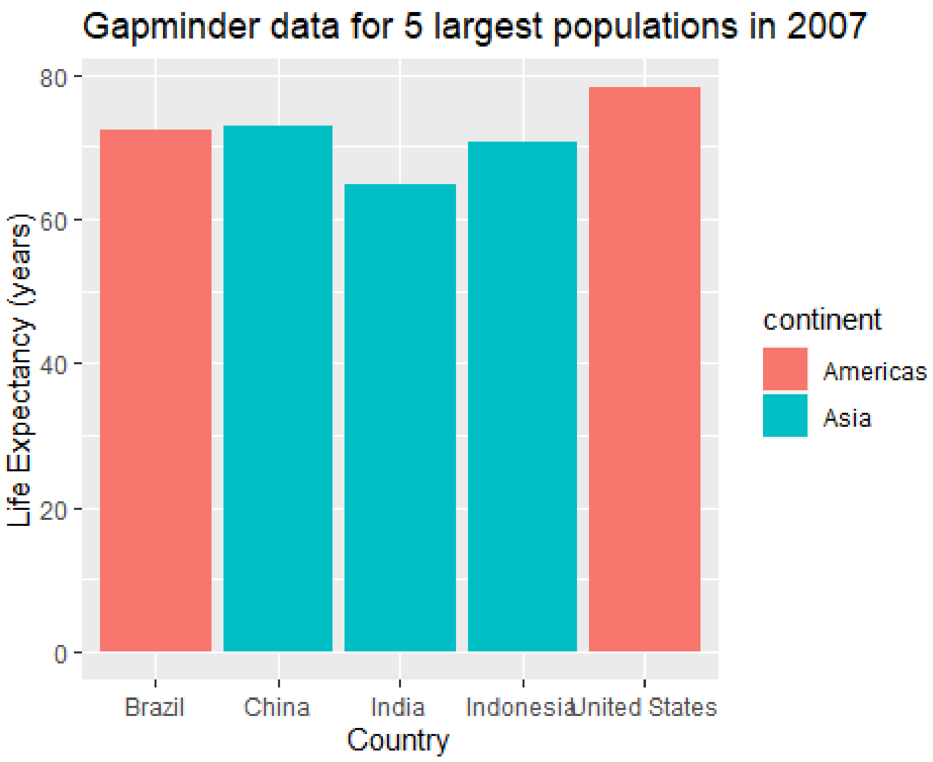
```
ggplot(gapminder_merge) +
  geom_point(aes(x = gdpPercap, y = lifeExp,
                 size = pop, colour=continent), alpha=0.7) +
  scale_size_area(max_size = 15) +
  facet_wrap(~continent) +
  labs(title="Gapminder Data for 1957 and 2007", x="GDP per capita", y="Life
Expentancy", colour="Continent", size="Population")
```



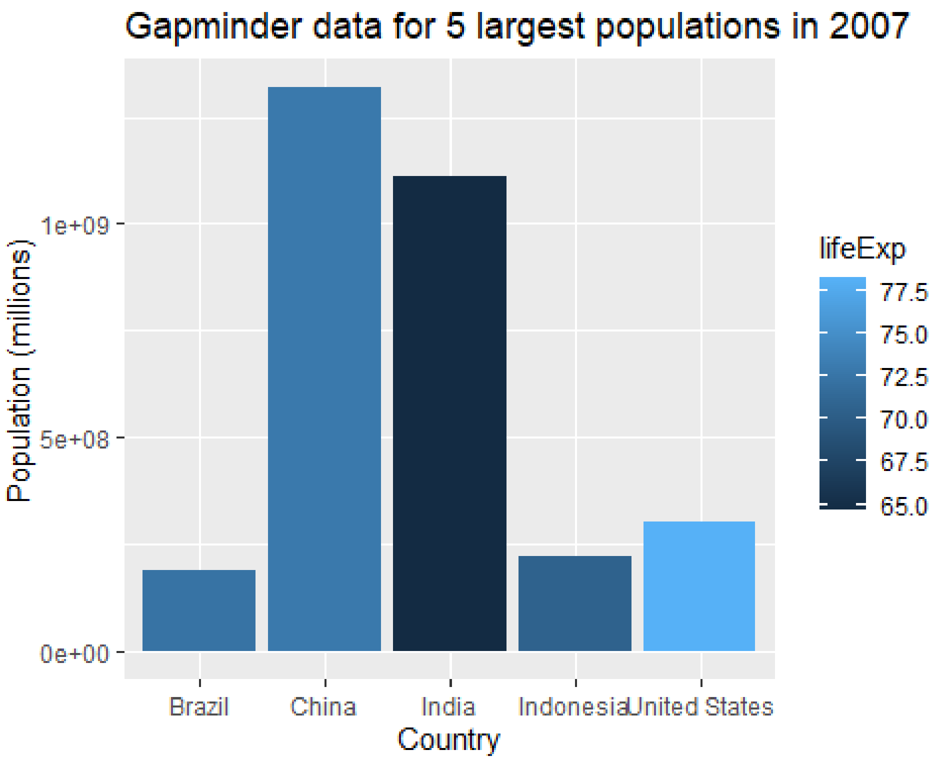
```
#Section 7 - (Optional)

##organise data
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)

gapminder_top5

##          country continent year lifeExp        pop gdpPercap
## 1          China      Asia 2007  72.961 1318683096  4959.115
## 2          India      Asia 2007  64.698 1110396331  2452.210
## 3 United States  Americas 2007  78.242  301139947 42951.653
## 4      Indonesia      Asia 2007  70.650  223547000  3540.652
## 5         Brazil  Americas 2007  72.390  190010647  9065.801

##plot data - country vs population
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = pop, fill = continent)) +
  labs(title = "Gapminder data for 5 largest populations in 2007", x =
"Country", y = "Population (millions)")
```



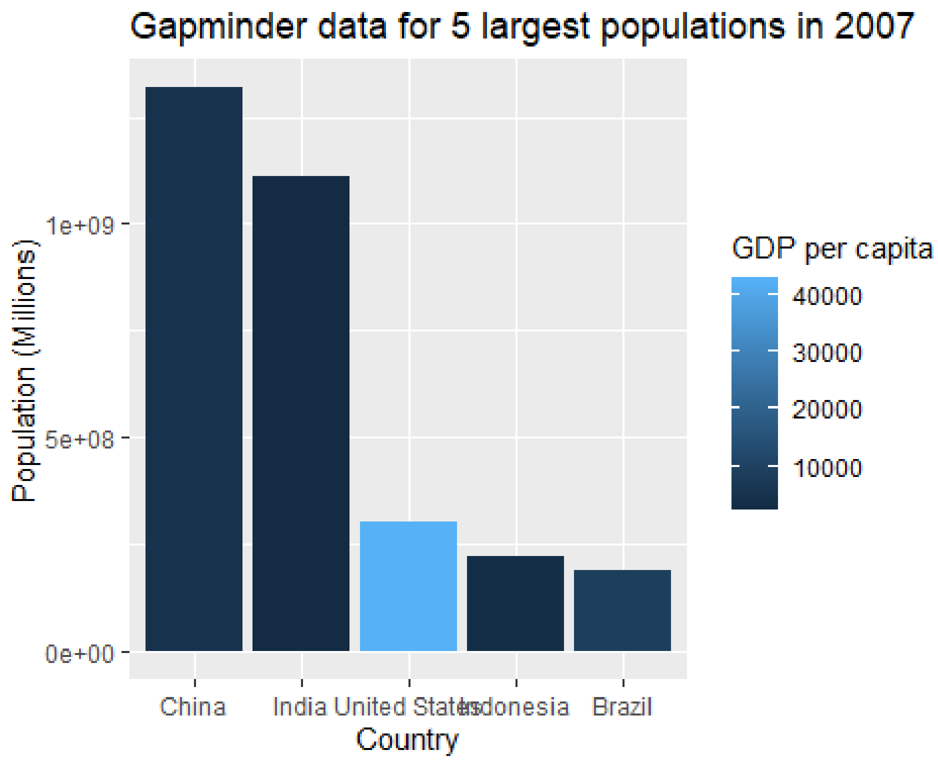
```
###Q plot
ggplot(gapminder_top5) +
  geom_col(aes(country, lifeExp, fill = continent)) +
  labs(title = "Gapminder data for 5 largest populations in 2007", x =
"Country", y = "Life Expectancy (years)")
```



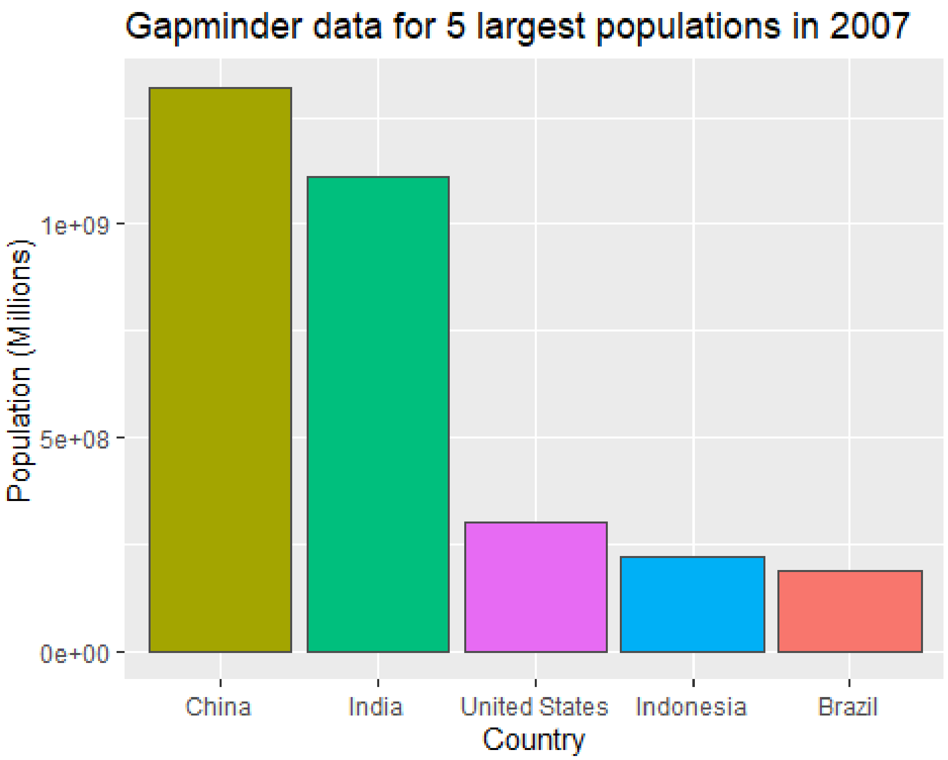
```
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = pop, fill = lifeExp)) +
  labs(title = "Gapminder data for 5 largest populations in 2007", x =
"Country", y = "Population (millions)")
```



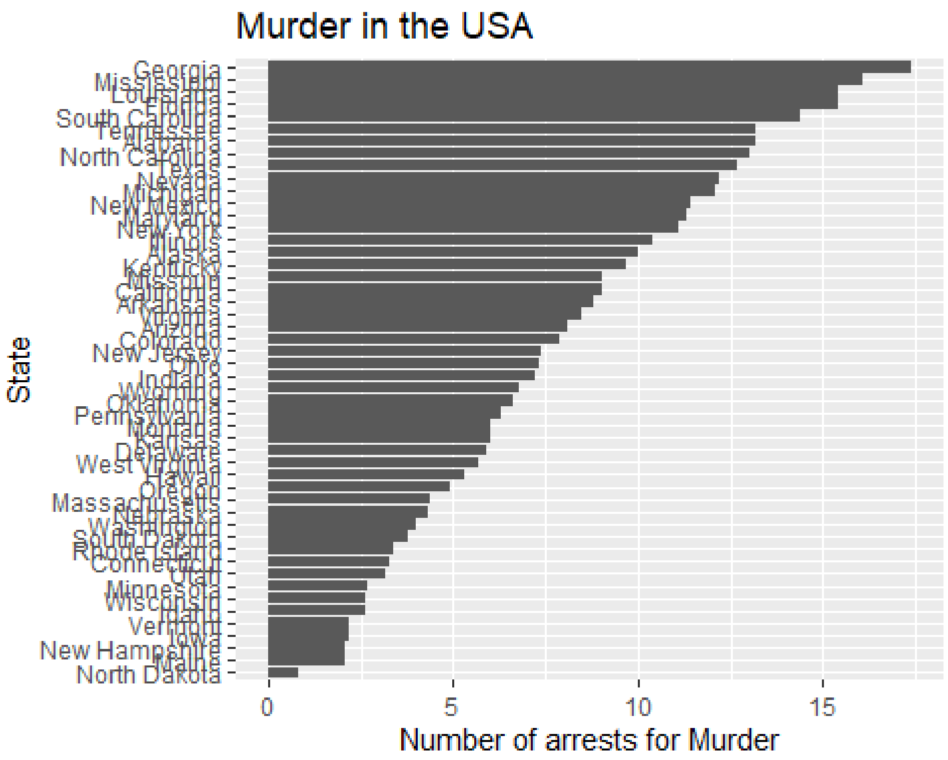
```
###Q plot
ggplot(gapminder_top5) +
  geom_col(aes(reorder(country, -pop), pop, fill = gdpPercap)) +
  labs(title = "Gapminder data for 5 largest populations in 2007", x =
"Country", y = "Population (Millions)", fill = "GDP per capita")
```



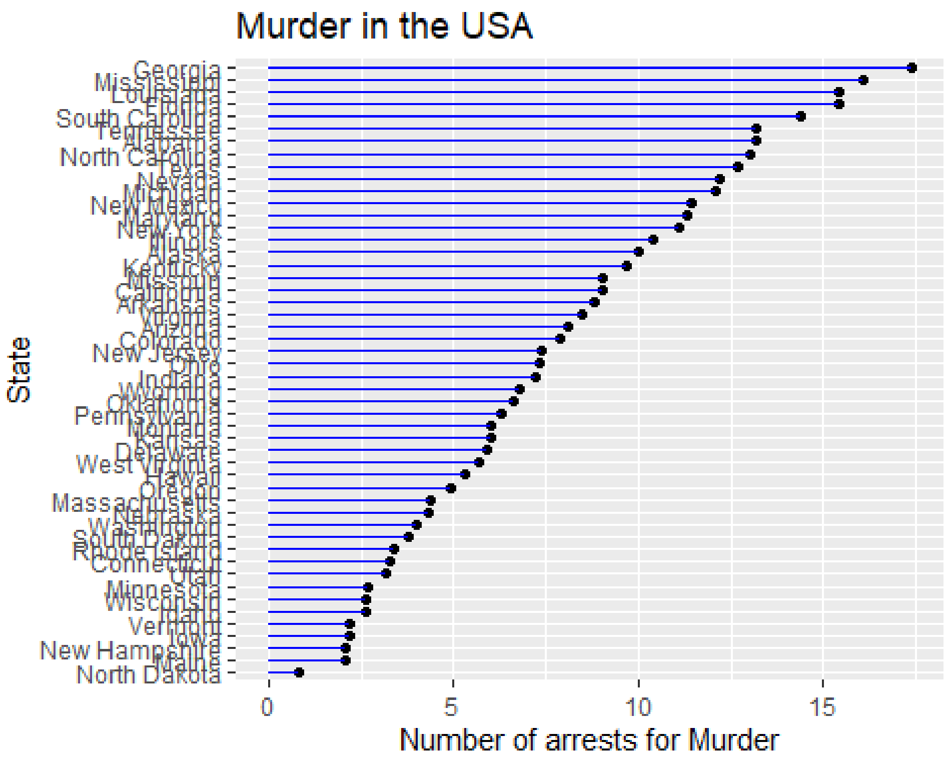
```
ggplot(gapminder_top5) +
  geom_col(aes(reorder(country, -pop), pop, fill = country), show.legend =
FALSE, col="gray30") +
  labs(title = "Gapminder data for 5 largest populations in 2007", x =
"Country", y = "Population (Millions)")
```



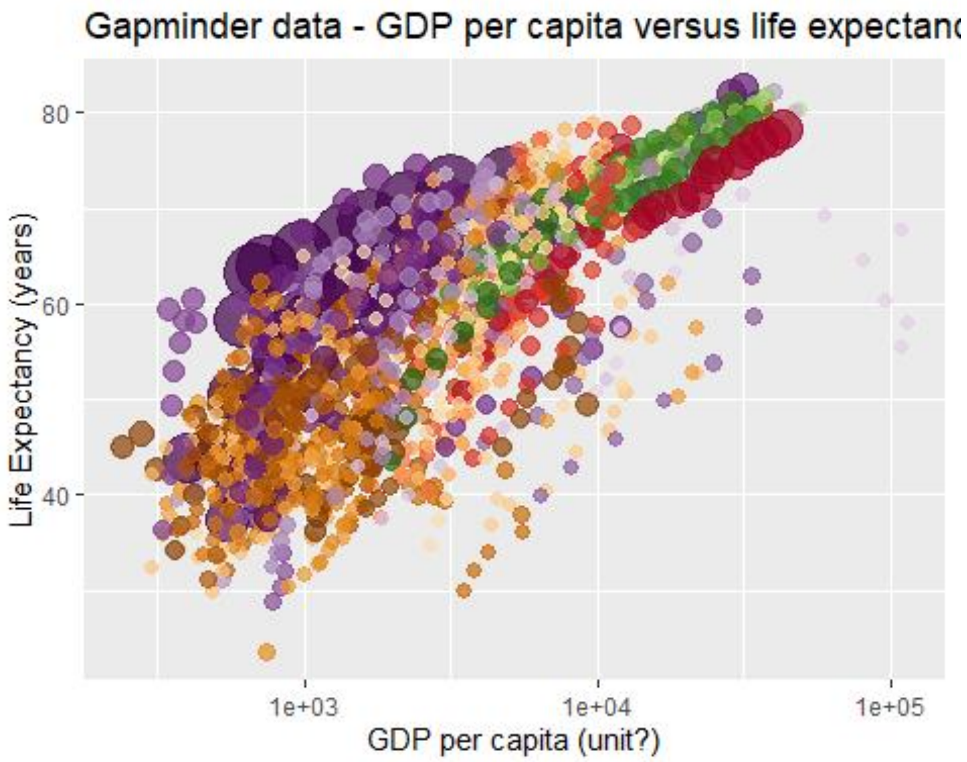
```
###Flipping bars
str(USArrests)

## 'data.frame':    50 obs. of  4 variables:
##  $ Murder  : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
##  $ Assault : int  236 263 294 190 276 204 110 238 335 211 ...
##  $ UrbanPop: int  58 48 80 50 91 78 77 72 80 60 ...
##  $ Rape    : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...

USArrests$State <- rownames(USArrests)

ggplot(USArrests) +
  geom_col(aes(reorder(State,Murder), Murder)) +
  coord_flip() +
  labs(title = "Murder in the USA", y = "Number of arrests for Murder", x =
"State")
```



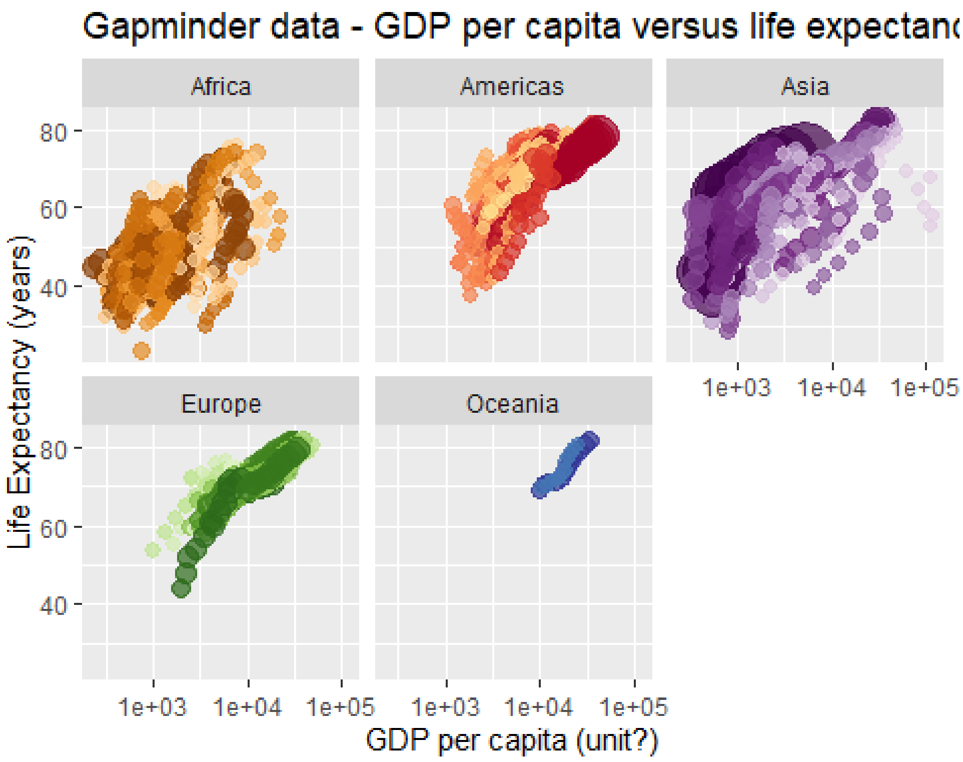
```
ggplot(USArrests) +
  aes(reorder(State,Murder), Murder) +
  geom_point() +
  geom_segment(aes(x=State,
                   xend=State,
                   y=0,
                   yend=Murder), color="blue") +
  coord_flip() +
  labs(title = "Murder in the USA", y = "Number of arrests for Murder", x =
"State")
```



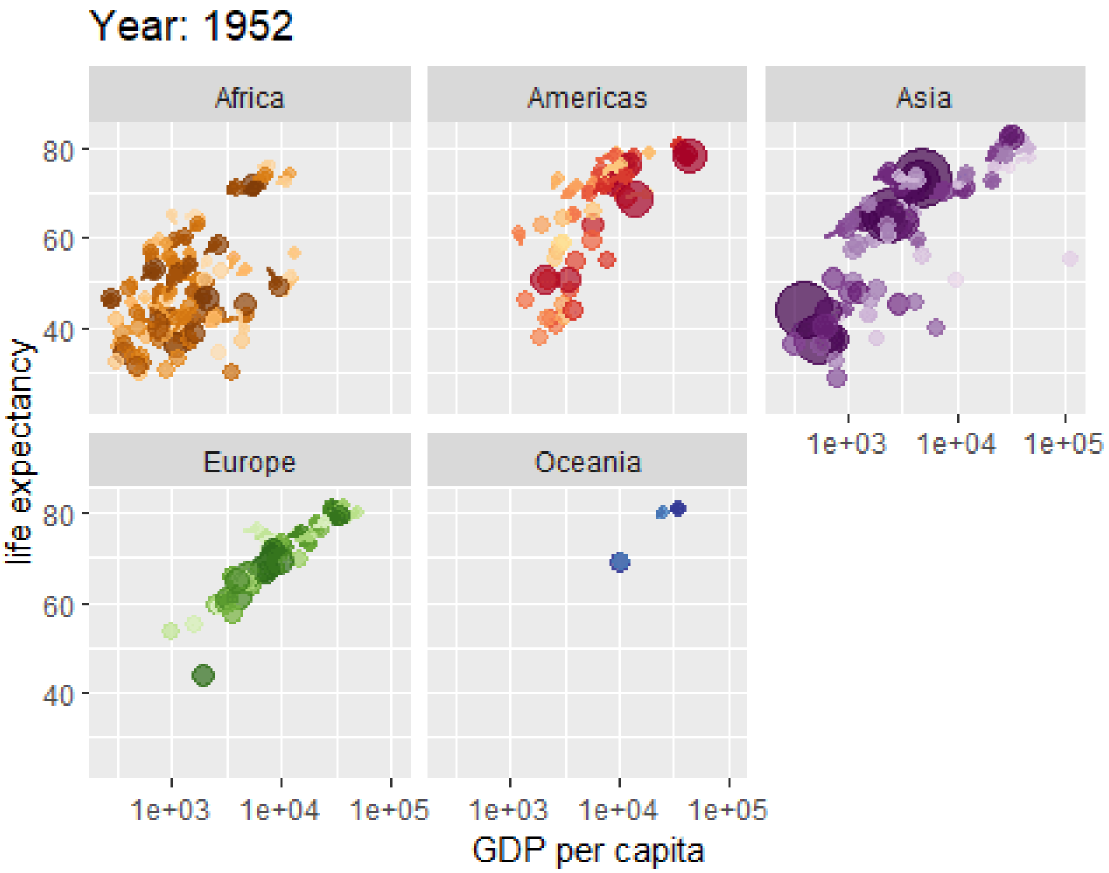
```
#Section 8 - optional
## first installed gifski and gganimate
## set up
library(gapminder)
library(gifski)
library(gganimate)

##intial plot -> plot GDP per capita against life expentency with size set by
population (and correctly scaled with scale_size) and colour set by country
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  labs(title = "Gapminder data - GDP per capita versus life expectancy for
various countries", x = "GDP per capita (unit?)", y = "Life Expectancy
(years)")
```



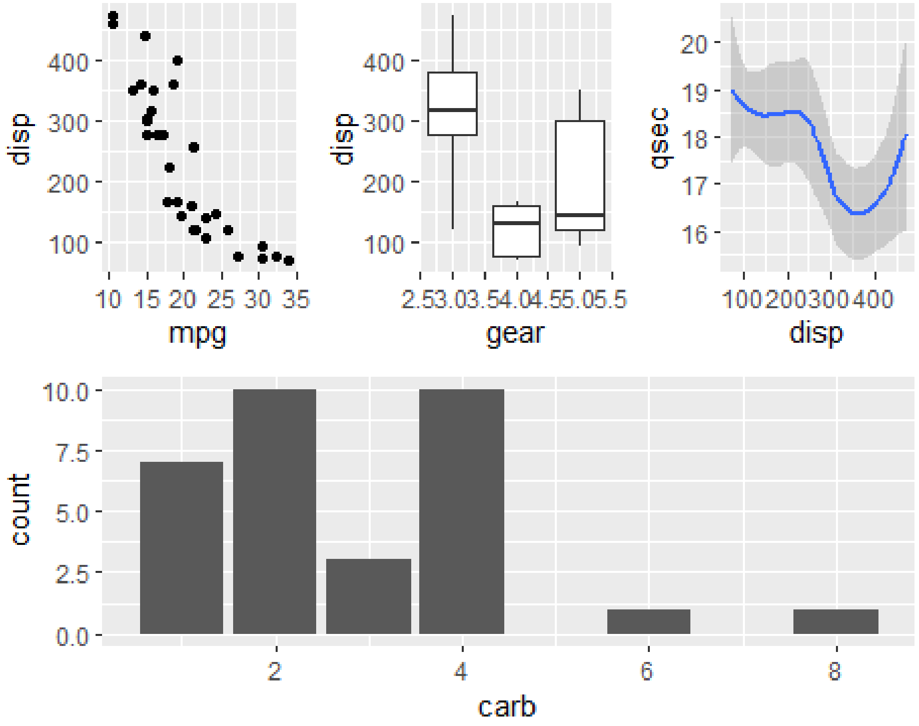
```
##same plot faceted by continent
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  labs(title = "Gapminder data - GDP per capita versus life expectancy for
various countries", x = "GDP per capita (unit?)", y = "Life Expectancy
(years)") +
  facet_wrap(~continent)
```



```r
#final plot
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  #labs(title = "Gapminder data - GDP per capita versus life expectancy for
various countries", x = "GDP per capita (unit?)", y = "Life Expectancy
(years)") +
  facet_wrap(~continent) +
  #gganimate specific lines
  labs(title = 'Year: {frame_time}', x = 'GDP per capita', y = 'life
expectancy') +
  transition_time(year) +
  shadow_wake(wake_length = 0.1, alpha = FALSE)
```



```
#Section 9 - Combining plots (optional)

library(patchwork)

# Setup some example plots
p1 <- ggplot(mtcars) + geom_point(aes(mpg, disp))
p2 <- ggplot(mtcars) + geom_boxplot(aes(gear, disp, group = gear))
p3 <- ggplot(mtcars) + geom_smooth(aes(disp, qsec))
p4 <- ggplot(mtcars) + geom_bar(aes(carb))

# Use patchwork to combine them here:
(p1 | p2 | p3) /
  p4

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



```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
## 
## Matrix products: default
## 
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252 
## [2] LC_CTYPE=English_United Kingdom.1252   
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C                           
## [5] LC_TIME=English_United Kingdom.1252    
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
## [1] patchwork_1.1.1 gganimate_1.0.7 gifski_1.4.3-1  dplyr_1.0.7    
## [5] gapminder_0.3.0 ggplot2_3.3.5  
## 
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8        plyr_1.8.6        pillar_1.7.0      compiler_4.1.2   
##  [5] highr_0.9         prettyunits_1.1.1 progress_1.2.2    tools_4.1.2      
##  [9] digest_0.6.29     lattice_0.20-45   nlme_3.1-153      evaluate_0.14    
## [13] lifecycle_1.0.1   tibble_3.1.6      gtable_0.3.0
```

```
viridisLite_0.4.0
## [17] mgcv_1.8-38       pkgconfig_2.0.3   rlang_1.0.0       Matrix_1.3-4
## [21] cli_3.1.1         yaml_2.2.2        xfun_0.29         fastmap_1.1.0
## [25] withr_2.4.3       stringr_1.4.0     knitr_1.37        hms_1.1.1
## [29] generics_0.1.2    vctrs_0.3.8       grid_4.1.2
tidyselect_1.1.1
## [33] glue_1.6.1        R6_2.5.1          fansi_1.0.2       rmarkdown_2.11
## [37] tweenr_1.0.2      purrr_0.3.4       farver_2.1.0      magrittr_2.0.2
## [41] splines_4.1.2     scales_1.1.1      ellipsis_0.3.2    htmltools_0.5.2
## [45] colorspace_2.0-2  labeling_0.4.2    utf8_1.2.2        stringi_1.7.6
## [49] munsell_0.5.0     crayon_1.4.2
```