R Notebook

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

plot(cars)



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

# Load the libraries  
library(gapminder) # Contains the Gapminder dataset  
library(ggplot2) # For creating static plots  
library(gganimate) # For adding animation to the plots  
library(gifski) # For rendering the animation as a GIF  
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 the Gapminder dataset  
data(gapminder)  
  
# Display the first few rows of the dataset  
head(gapminder)

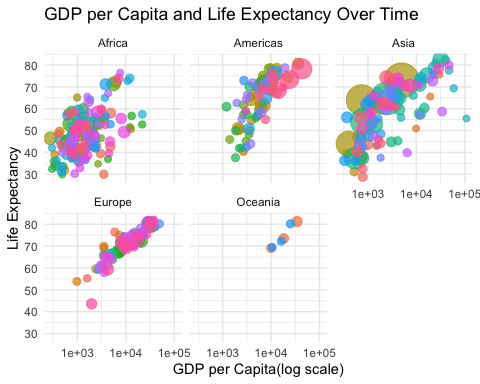
## # A tibble: 6 × 6  
## country continent year lifeExp pop gdpPercap  
## <fct> <fct> <int> <dbl> <int> <dbl>  
## 1 Afghanistan Asia 1952 28.8 8425333 779.  
## 2 Afghanistan Asia 1957 30.3 9240934 821.  
## 3 Afghanistan Asia 1962 32.0 10267083 853.  
## 4 Afghanistan Asia 1967 34.0 11537966 836.  
## 5 Afghanistan Asia 1972 36.1 13079460 740.  
## 6 Afghanistan Asia 1977 38.4 14880372 786.

# Summary of the dataset  
summary(gapminder)

## country continent year lifeExp   
## Afghanistan: 12 Africa :624 Min. :1952 Min. :23.60   
## Albania : 12 Americas:300 1st Qu.:1966 1st Qu.:48.20   
## Algeria : 12 Asia :396 Median :1980 Median :60.71   
## Angola : 12 Europe :360 Mean :1980 Mean :59.47   
## Argentina : 12 Oceania : 24 3rd Qu.:1993 3rd Qu.:70.85   
## Australia : 12 Max. :2007 Max. :82.60   
## (Other) :1632   
## pop gdpPercap   
## Min. :6.001e+04 Min. : 241.2   
## 1st Qu.:2.794e+06 1st Qu.: 1202.1   
## Median :7.024e+06 Median : 3531.8   
## Mean :2.960e+07 Mean : 7215.3   
## 3rd Qu.:1.959e+07 3rd Qu.: 9325.5   
## Max. :1.319e+09 Max. :113523.1   
##

# Create a static scatter plot  
gapminder1 <- gapminder %>% filter(as.numeric(as.character(year)) %in% c(1952, 1977, 2007))

static\_plot <- ggplot(gapminder1, aes(x = gdpPercap, y = lifeExp, size = pop, colour = country)) +  
 # ggplot is used with aes mapping x-axis to gdpPerCap, y-axis to lifeExp, size population and coloring with continent  
 geom\_point(alpha = 0.7, show.legend = FALSE) + # Add points with transparency  
 # `alpha = 0.7`: Sets the transparency of the points (0 = fully transparent, 1 = fully opaque).  
  
 scale\_size(range = c(2, 12)) +   
 # Set the size range of the points  
 scale\_x\_log10() +   
 # Apply log10 transformation to the X-axis  
 facet\_wrap(~continent) +  
 #Split the plot into sub-plots by continent using `facet\_wrap()`.  
 # Create sub-plots by continent  
 labs(title = 'GDP per Capita and Life Expectancy Over Time', x = 'GDP per Capita(log scale)', y = 'Life Expectancy') + # Add titles and # labels the x and the y axes  
 theme\_minimal()   
# Use a minimal theme for the plot  
# Display the static plot  
static\_plot

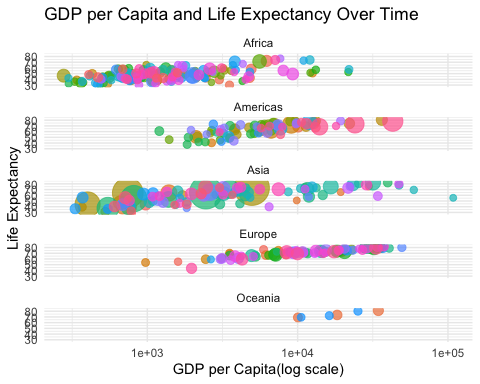


# Create a new animated plot with population on the X-axis  
plot <- ggplot(gapminder1, aes(x = pop, y = lifeExp, size = pop, colour = country)) +  
 # ggplot is used with aes mapping x-axis to Population, y-axis to lifeExp, size population and coloring with continent  
 geom\_point(alpha = 0.7, show.legend = FALSE) + # Add points with transparency  
 # `alpha = 0.7`: Sets the transparency of the points (0 = fully transparent, 1 = fully opaque).  
 scale\_size(range = c(2, 12)) +   
 # Set the size range of the points  
 scale\_x\_log10() +   
 # Apply log10 transformation to the X-axis  
 facet\_wrap(~continent) +  
 #Split the plot into sub-plots by continent using `facet\_wrap()`.  
 # Create sub-plots by continent  
 labs(title = 'Population and Life Expectancy Over Time', x = 'Population(log scale)', y = 'Life Expectancy') + # Add titles and # labels the x and the y axes  
 theme\_minimal() +  
 transition\_time(year) +  
 ease\_aes('linear')  
  
# Render and save the new animation  
#animate(plot, width = 800, height = 800)

# Change the layout to 1 row and 5 columns  
layout\_change <- ggplot(gapminder1, aes(x = gdpPercap, y = lifeExp, size = pop, colour = country)) +  
 # ggplot is used with aes mapping x-axis to gdpPerCap, y-axis to lifeExp, size population and coloring with continent  
 geom\_point(alpha = 0.7, show.legend = FALSE) + # Add points with transparency  
 # `alpha = 0.7`: Sets the transparency of the points (0 = fully transparent, 1 = fully opaque).  
  
 scale\_size(range = c(2, 12)) +   
 # Set the size range of the points  
 scale\_x\_log10() +   
 # Apply log10 transformation to the X-axis  
 facet\_wrap(~continent) +  
 #Split the plot into sub-plots by continent using `facet\_wrap()`.  
 # Create sub-plots by continent  
 labs(title = 'GDP per Capita and Life Expectancy Over Time', x = 'GDP per Capita(log scale)', y = 'Life Expectancy') + # Add titles and # labels the x and the y axes  
 theme\_minimal() +  
   
 transition\_time(year) +  
 ease\_aes('linear')  
  
# Render and save the new animation  
#animate(layout\_change, width = 1600, height = 400)

# Adjust font sizes  
font\_plot <- ggplot(gapminder1, aes(x = gdpPercap, y = lifeExp, size = pop, colour = country)) +  
 # ggplot is used with aes mapping x-axis to gdpPerCap, y-axis to lifeExp, size population and coloring with continent  
 geom\_point(alpha = 0.7, show.legend = FALSE) + # Add points with transparency  
 # `alpha = 0.7`: Sets the transparency of the points (0 = fully transparent, 1 = fully opaque).  
 scale\_size(range = c(2, 12)) +   
 # Set the size range of the points  
 scale\_x\_log10() +   
 # Apply log10 transformation to the X-axis  
 facet\_wrap(~continent) +  
 #Split the plot into sub-plots by continent using `facet\_wrap()`.  
 # Create sub-plots by continent  
 labs(title = 'GDP per Capita and Life Expectancy Over Time', x = 'GDP per Capita(log scale)', y = 'Life Expectancy') + # Add titles and # labels the x and the y axes  
 theme\_minimal()+  
  
 theme(  
 plot.title = element\_text(size = 20, face = "bold", color = "black"),   
 # Adjusts the font size, face to bold, and color to black for title of the plot  
 axis.title.x = element\_text(size = 15), # Adjust X-axis title font  
 axis.title.y = element\_text(size = 15), # Adjust Y-axis title font  
 axis.text.x = element\_text(size = 12), # Adjust X-axis tick labels  
 axis.text.y = element\_text(size = 12), # Adjust Y-axis tick labels  
 strip.text.x = element\_text(size = 14) # Adjust facet labels  
 ) +  
 transition\_time(year) +  
 ease\_aes('linear')  
# Render and save the new animation  
#animate(font\_plot, width = 800, height = 800)

static\_plot <- ggplot(gapminder1, aes(x = gdpPercap, y = lifeExp, size = pop, colour = country)) +  
 # ggplot is used with aes mapping x-axis to gdpPerCap, y-axis to lifeExp, size population and coloring with continent  
 geom\_point(alpha = 0.7, show.legend = FALSE) + # Add points with transparency  
 # `alpha = 0.7`: Sets the transparency of the points (0 = fully transparent, 1 = fully opaque).  
 scale\_size(range = c(2, 12)) +   
 # Set the size range of the points  
 scale\_x\_log10() +   
 # Apply log10 transformation to the X-axis  
 facet\_wrap(~continent,ncol=1) +  
 #Split the plot into sub-plots by continent using `facet\_wrap()`.  
 # Create sub-plots by continent  
 labs(title = 'GDP per Capita and Life Expectancy Over Time', x = 'GDP per Capita(log scale)', y = 'Life Expectancy') + # Add titles and # labels the x and the y axes  
 theme\_minimal()   
# Use a minimal theme for the plot  
# Display the static plot  
static\_plot



# Create the animated scatter plot by adding animation to the static plot.  
animated\_plot <- static\_plot +  
 transition\_time(year) +  
 ease\_aes('linear')   
#animate(animated\_plot, width = 800, height = 1200)