ASSIGNMENT #1 - Tidyverse & Int. to R

Pooja Dilip Talikoti ,ptali1@unh.newhaven.edu

Use gapminder package and data set to answer the following questions. Please write your commands under each question.

1. Get the data for 2002. Assign a name to that data.

```
data_2002 <- gapminder %>% filter(year == 2002)
data_2002
```

```
## # A tibble: 142 × 6
##    country     continent  year lifeExp       pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>     <int>     <dbl>
##  1 Afghanistan Asia       2002    42.1  25268405      727.
##  2 Albania     Europe     2002    75.7   3508512     4604.
##  3 Algeria     Africa     2002    71.0  31287142     5288.
##  4 Angola      Africa     2002    41.0  10866106     2773.
##  5 Argentina   Americas   2002    74.3  38331121     8798.
##  6 Australia   Oceania    2002    80.4  19546792    30688.
##  7 Austria     Europe     2002    79.0   8148312    32418.
##  8 Bahrain     Asia       2002    74.8    656397    23404.
##  9 Bangladesh  Asia       2002    62.0 135656790     1136.
## 10 Belgium     Europe     2002    78.3  10311970    30486.
## # ℹ 132 more rows
```

2. Get the data for Germany in 2002.

```
germany_2002 <- gapminder %>% filter(country == "Germany", year == 2002)
germany_2002
```

```
## # A tibble: 1 × 6
##   country continent  year lifeExp      pop gdpPercap
##   <fct>   <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Germany Europe     2002    78.7 82350671    30036.
```

3. Find which country has the lowest lifeExp overall.

```
lowest_lifeExp_overall <- gapminder %>% arrange(lifeExp) %>% slice(1)
lowest_lifeExp_overall
```

```
## # A tibble: 1 × 6
##   country continent  year lifeExp     pop gdpPercap
##   <fct>   <fct>     <int>   <dbl>   <int>     <dbl>
## 1 Rwanda  Africa     1992    23.6 7290203      737.
```

4. Find which country has the lowest lifeExp in 2002.

```
lowest_lifeExp_2002 <- data_2002 %>% arrange(lifeExp) %>% slice(1)
lowest_lifeExp_2002
```

```
## # A tibble: 1 × 6
##   country continent  year lifeExp      pop gdpPercap
##   <fct>   <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Zambia  Africa     2002    39.2 10595811     1072.
```

5. Find the lifeExp in Germany in 2002.

```
germany_lifeExp_2002 <- germany_2002$lifeExp
germany_lifeExp_2002
```

```
## [1] 78.67
```

6. Find the countries whose lifeExp is higher than 80 in 2002.

```
countries_lifeExp_above_80 <- data_2002 %>% filter(lifeExp > 80) %>%
select(country, lifeExp)
countries_lifeExp_above_80
```

```
## # A tibble: 7 × 2
##   country          lifeExp
##   <fct>              <dbl>
## 1 Australia           80.4
## 2 Hong Kong, China    81.5
## 3 Iceland             80.5
## 4 Italy               80.2
## 5 Japan               82
## 6 Sweden              80.0
## 7 Switzerland         80.6
```

7. Find the countries whose lifeExp is more than 70 and less than 80

```
countries_lifeExp_70_to_80 <- data_2002 %>% filter(lifeExp > 70, lifeExp <
80) %>% select(country, lifeExp)
countries_lifeExp_70_to_80
```

```
## # A tibble: 68 × 2
##    country                lifeExp
##    <fct>                    <dbl>
##  1 Albania                   75.7
##  2 Algeria                   71.0
##  3 Argentina                 74.3
##  4 Austria                   79.0
```

```
##  5 Bahrain                   74.8
##  6 Belgium                   78.3
##  7 Bosnia and Herzegovina    74.1
##  8 Brazil                    71.0
##  9 Bulgaria                  72.1
## 10 Canada                    79.8
## # ℹ 58 more rows
```

8. Find the lifeExp in Europe across the years. Which year is the highest lifeExp in Europe?

```
gapminder %>%
  filter(continent == "Europe") %>%
  group_by(year) %>%
  summarise(avg_lifeExp = mean(lifeExp)) %>%
  arrange(desc(avg_lifeExp))
```

```
## # A tibble: 12 × 2
##     year avg_lifeExp
##    <int>       <dbl>
##  1  2007        77.6
##  2  2002        76.7
##  3  1997        75.5
##  4  1992        74.4
##  5  1987        73.6
##  6  1982        72.8
##  7  1977        71.9
##  8  1972        70.8
##  9  1967        69.7
## 10  1962        68.5
## 11  1957        66.7
## 12  1952        64.4
```

9. Define gdp as it is equal to to gdpPercap * pop/10000 . Find the gdp of Europe in 2002.

```
gdp_europe_2002 <- data_2002 %>%
  mutate(gdp = gdpPercap * pop / 10000) %>%
  filter(continent == "Europe") %>%
  summarize(total_gdp = sum(gdp))
gdp_europe_2002
```

```
## # A tibble: 1 × 1
##     total_gdp
##         <dbl>
## 1 1309346445.
```

10. Which country has the highest gdp in Europe in 2002 ?

```
highest_gdp_country <- data_2002 %>%
  mutate(gdp = gdpPercap * pop / 10000) %>%
  filter(continent == "Europe") %>%
  arrange(desc(gdp)) %>%
  slice(1)
highest_gdp_country
```

```
## # A tibble: 1 × 7
##   country continent  year lifeExp      pop gdpPercap        gdp
##   <fct>   <fct>     <int>   <dbl>    <int>     <dbl>      <dbl>
## 1 Germany Europe     2002    78.7 82350671    30036. 247346845.
```

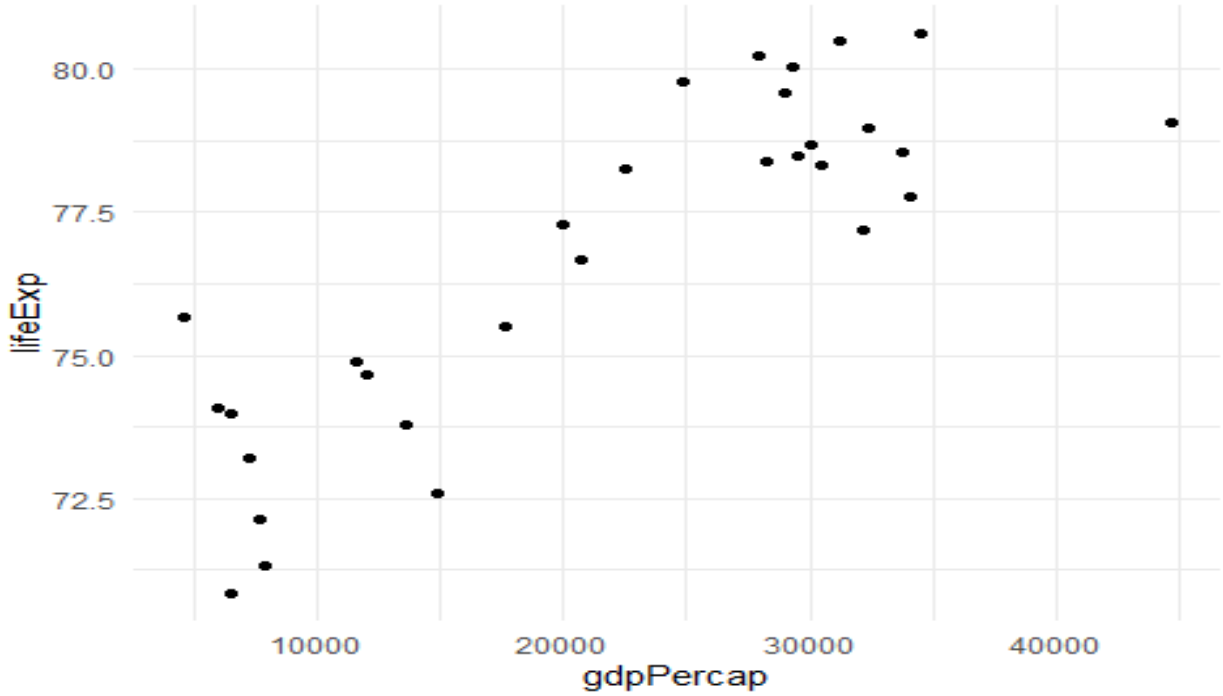
11. Save the data in 2002 in Europe. Call it data_2002.

```
data_Europe <- data_2002 %>% filter(continent == "Europe")
data_Europe
```

```
## # A tibble: 30 × 6
##    country                continent  year lifeExp      pop gdpPercap
##    <fct>                  <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Albania                Europe     2002    75.7  3508512     4604.
##  2 Austria                Europe     2002    79.0  8148312    32418.
##  3 Belgium                Europe     2002    78.3 10311970    30486.
##  4 Bosnia and Herzegovina Europe     2002    74.1  4165416     6019.
##  5 Bulgaria               Europe     2002    72.1  7661799     7697.
##  6 Croatia                Europe     2002    74.9  4481020    11628.
##  7 Czech Republic         Europe     2002    75.5 10256295    17596.
##  8 Denmark                Europe     2002    77.2  5374693    32167.
##  9 Finland                Europe     2002    78.4  5193039    28205.
## 10 France                 Europe     2002    79.6 59925035    28926.
## # ℹ 20 more rows
```

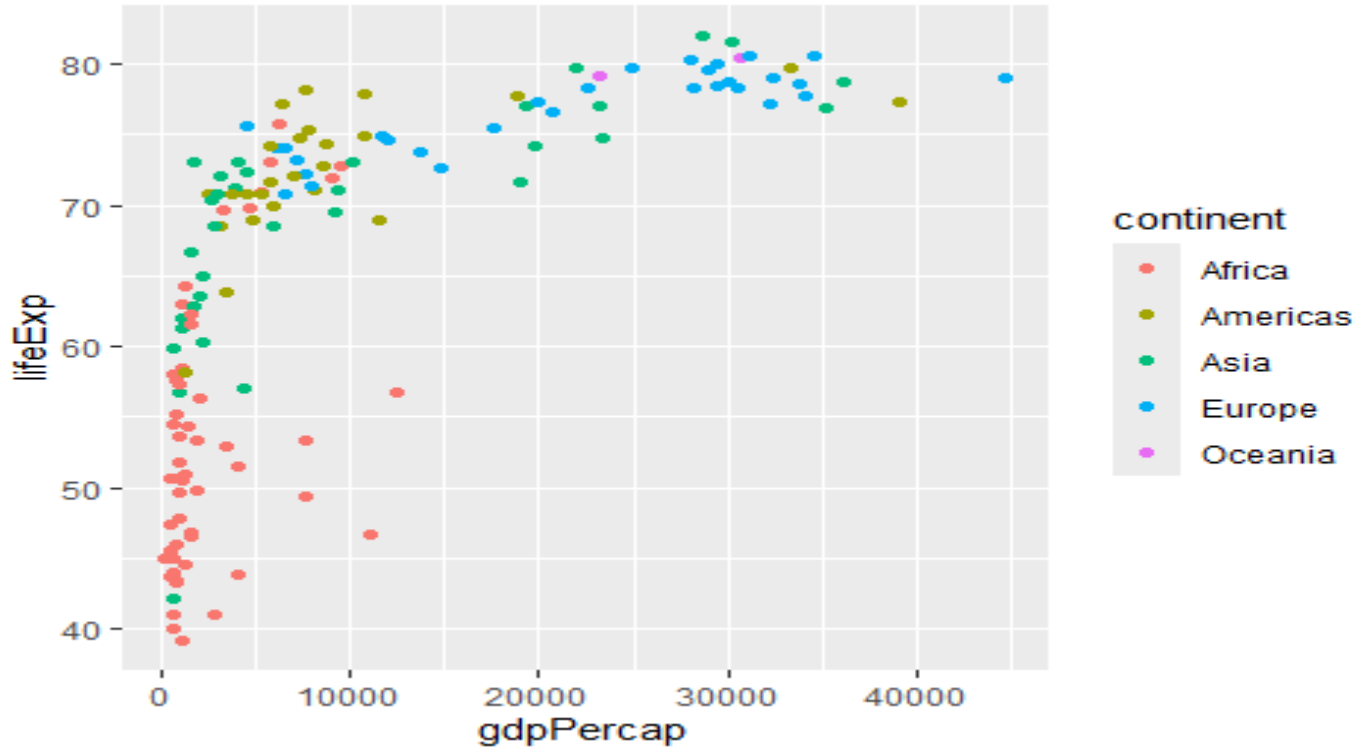
12. Use data_2002. Use ggplot. Plot gdpPercap vs lifeExp.

```
ggplot(data_Europe, aes(x = gdpPercap, y = lifeExp)) + geom_point() +
theme_minimal()
```



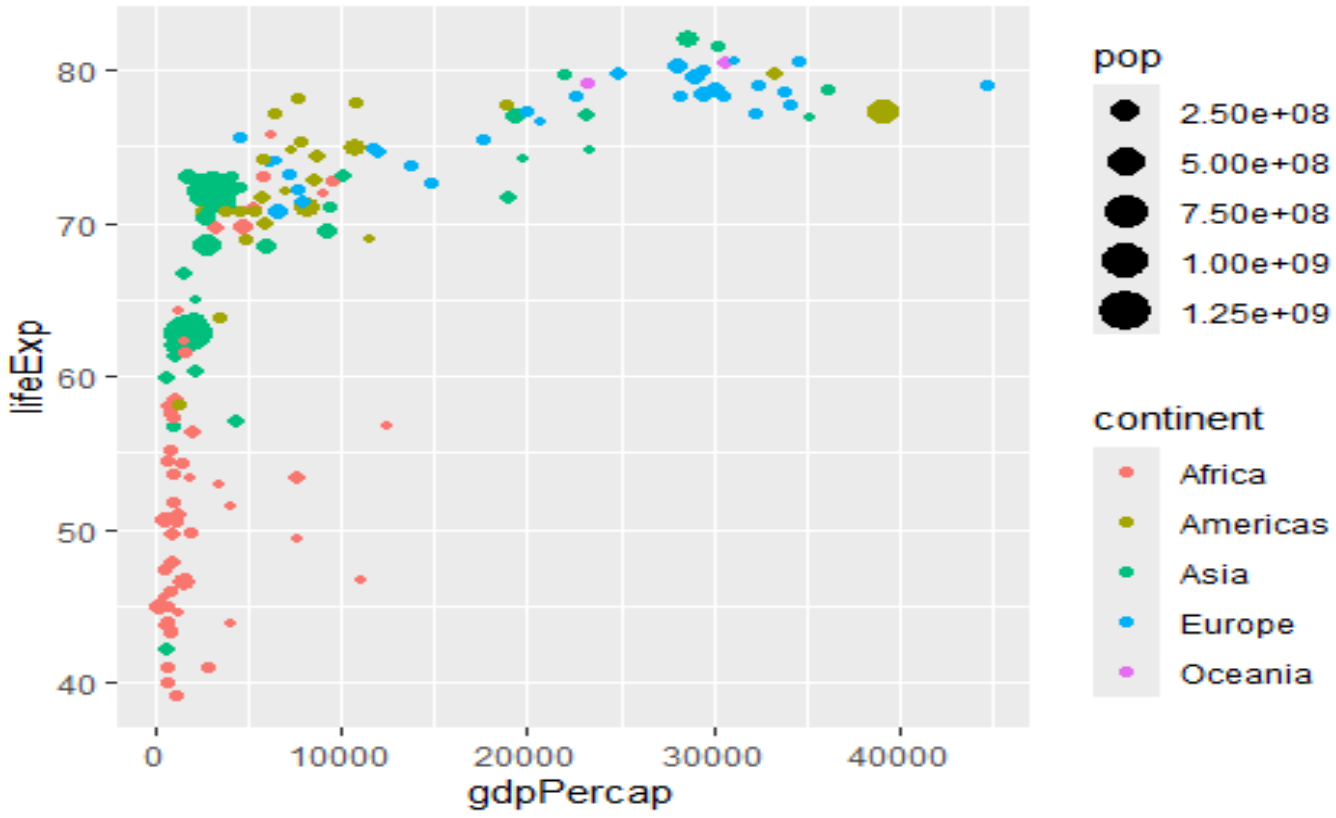
13. Use data_2002. Use ggplot. Plot gdpPercap vs lifeExp by continent (color)

```
ggplot(data_2002, aes(x = gdpPercap, y = lifeExp, color = continent)) +
geom_point()
```



14. Use data_2002. Use ggplot. Plot gdpPercap vs lifeExp by continent and pop (color and size)

```
ggplot(data_2002, aes(x = gdpPercap, y = lifeExp, color = continent, size =
pop)) +
geom_point()
```



15. Get data for Europe in 2002. Call it data_Europe

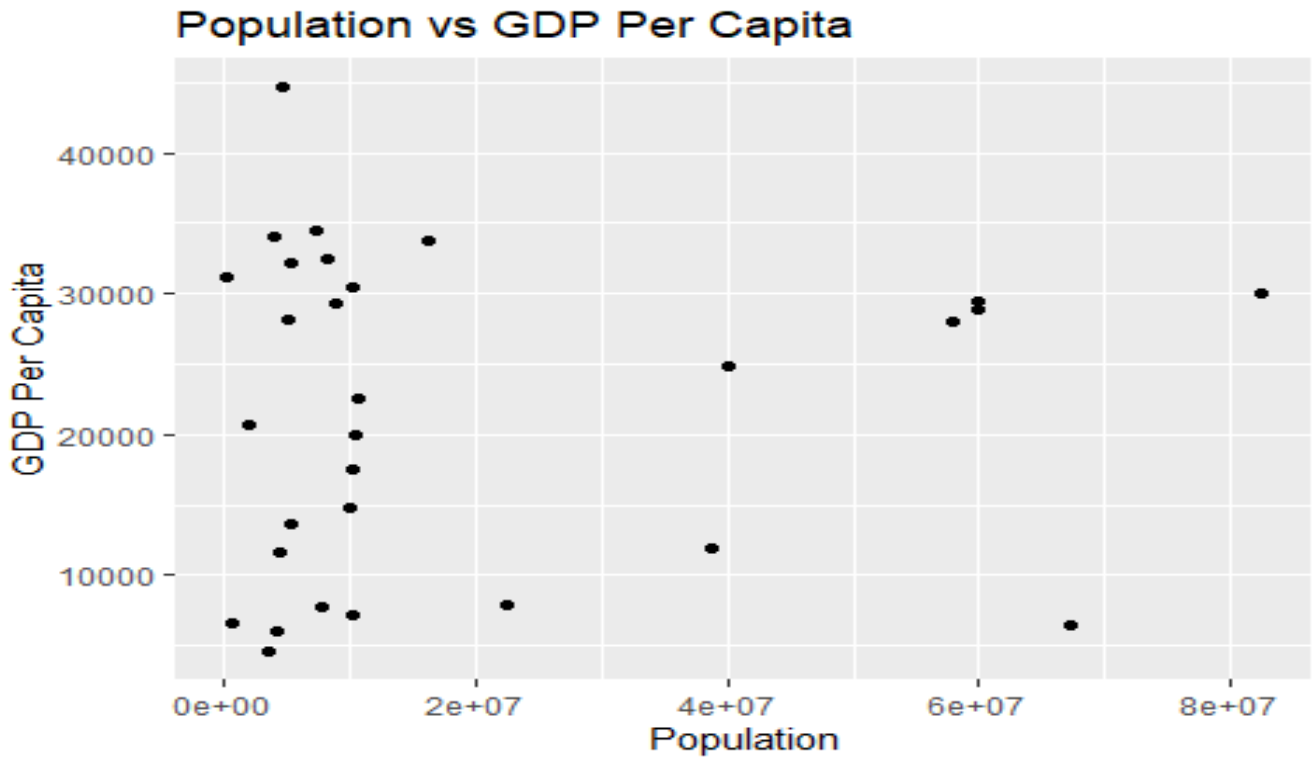
```
data_Europe <- gapminder %>% filter(year == 2002, continent == "Europe")
data_Europe
```

```
## # A tibble: 30 × 6
##    country                continent  year lifeExp      pop gdpPercap
##    <fct>                  <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Albania                Europe     2002    75.7  3508512     4604.
##  2 Austria                Europe     2002    79.0  8148312    32418.
##  3 Belgium                Europe     2002    78.3 10311970    30486.
##  4 Bosnia and Herzegovina Europe     2002    74.1  4165416     6019.
##  5 Bulgaria               Europe     2002    72.1  7661799     7697.
##  6 Croatia                Europe     2002    74.9  4481020    11628.
##  7 Czech Republic         Europe     2002    75.5 10256295    17596.
##  8 Denmark                Europe     2002    77.2  5374693    32167.
##  9 Finland                Europe     2002    78.4  5193039    28205.
```

```
## 10 France                  Europe     2002    79.6 59925035    28926.
## # ℹ 20 more rows
```

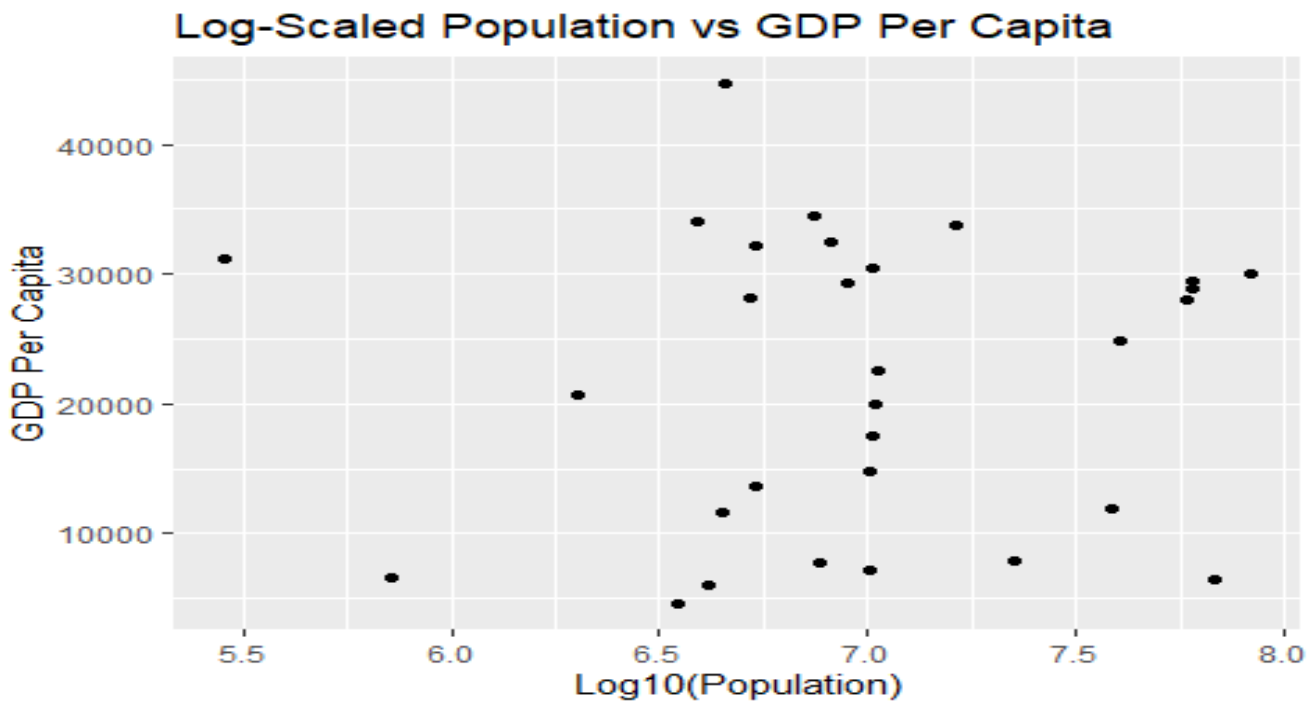
16. Use data_Europe. Use ggplot. Plot pop vs gdpPercap.

```
ggplot(data_Europe, aes(x = pop, y = gdpPercap)) +
  geom_point() +
  labs(title = "Population vs GDP Per Capita", x = "Population", y = "GDP Per
Capita")
```



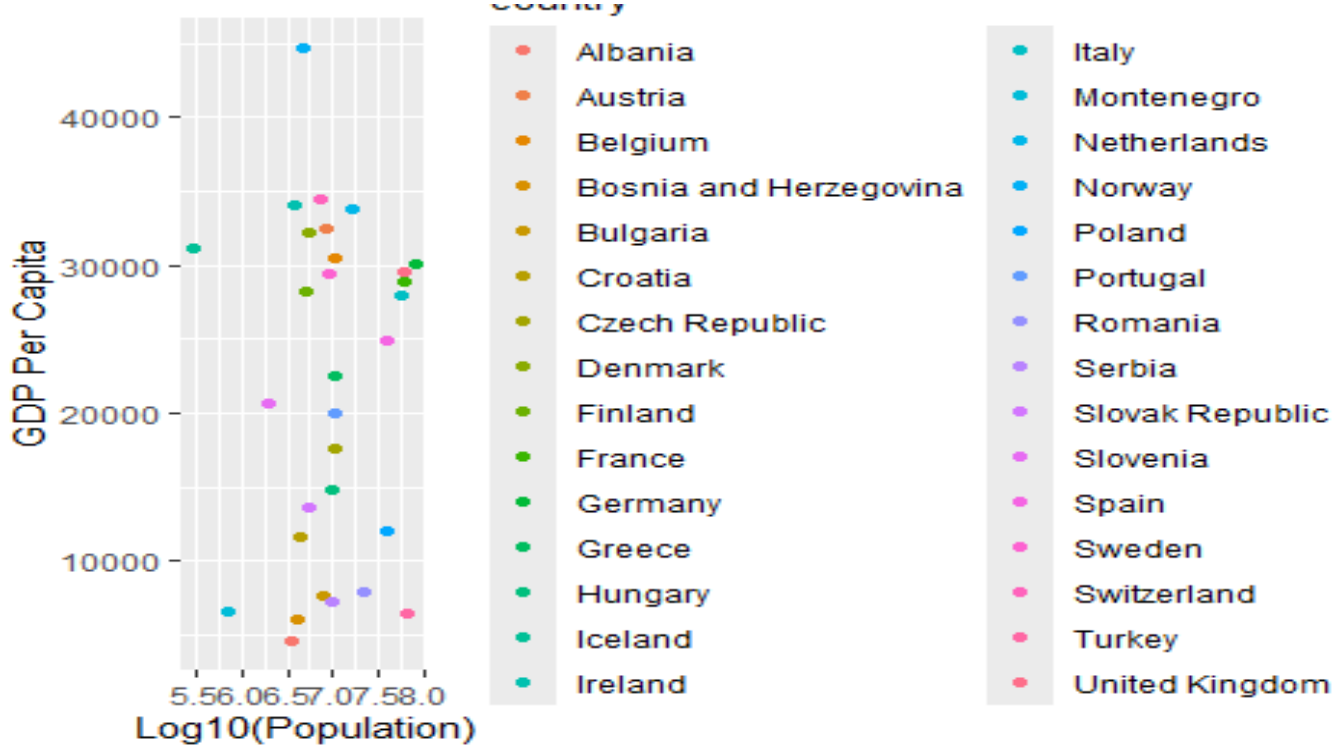
17. Use data_Europe. Use ggplot. Plot pop vs gdpPercap. Scale population by log10

```
ggplot(data_Europe, aes(x = log10(pop), y = gdpPercap)) +
  geom_point() +
  labs(title = "Log-Scaled Population vs GDP Per Capita", x =
"Log10(Population)", y = "GDP Per Capita")
```



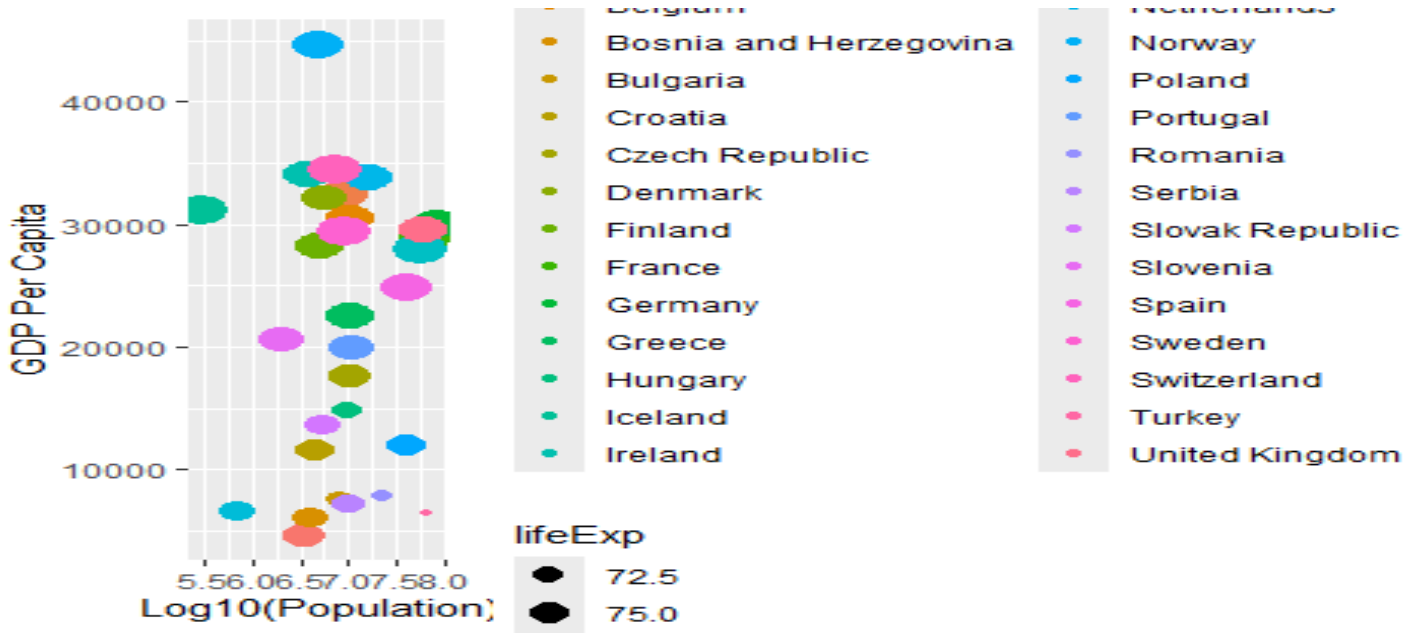
18. Use data_Europe. Use ggplot. Plot pop vs gdpPercap. Scale population by log10. Color the data by country.

```
ggplot(data_Europe, aes(x = log10(pop), y = gdpPercap, color = country)) +
  geom_point() +
  labs( x = "Log10(Population)", y = "GDP Per Capita")
```



19. Use data_Europe. Use ggplot. Plot pop vs gdpPercap. Scale population by log10. Color the data by country and size it by lifeExp.

```
ggplot(data_Europe, aes(x = log10(pop), y = gdpPercap, color = country, size
= lifeExp)) +
  geom_point() +
  labs( x = "Log10(Population)", y = "GDP Per Capita")
```



20. See the attached file in excel, namely,tourism.xls. Create a folder and give a name FORECASTING.

1) Save the tourism excel file in that FORECASTING directory.

2) Set your working directory as FORECASTING

3) Import tourism excel file into R-studio.

4) Assign a different name to this data, such as "mydata"

5) Check the structure of your dataset by str() function. Change Region column from character to factor. Use as.factor() function.

```
options(digits = 3, scipen = 9999, stringasFactors = FALSE)
# make sure characters are not factors. The 1st column, Quarter, needs to be
NOT factor.

library(readxl)
library(readxl)
setwd("C:/Users/pooja/OneDrive/Desktop/FORCASTING")
# Load dataset
mydata <- read_excel("tourism-3.xlsx")
str(mydata)
```

```
## tibble [24,320 × 5] (S3: tbl_df/tbl/data.frame)
##  $ Quarter: chr [1:24320] "1998-01-01" "1998-04-01" "1998-07-01" "1998-10-
01" ...
##  $ Region : chr [1:24320] "Adelaide" "Adelaide" "Adelaide" "Adelaide" ...
##  $ State  : chr [1:24320] "South Australia" "South Australia" "South
Australia" "South Australia" ...
##  $ Purpose: chr [1:24320] "Business" "Business" "Business" "Business" ...
##  $ Trips  : num [1:24320] 135 110 166 127 137 ...
```

```
# Convert Region column to factor
mydata$Region <- as.factor(mydata$Region)
str(mydata)
```

```
## tibble [24,320 × 5] (S3: tbl_df/tbl/data.frame)
##  $ Quarter: chr [1:24320] "1998-01-01" "1998-04-01" "1998-07-01" "1998-10-
01" ...
##  $ Region : Factor w/ 76 levels "Adelaide","Adelaide Hills",..: 1 1 1 1 1
1 1 1 1 1 ...
##  $ State  : chr [1:24320] "South Australia" "South Australia" "South
Australia" "South Australia" ...
##  $ Purpose: chr [1:24320] "Business" "Business" "Business" "Business" ...
##  $ Trips  : num [1:24320] 135 110 166 127 137 ...
```