Case Study 3

AKSTA Statistical Computing

Konstantinos Vakalopoulos 12223236

2023-06-01

Introduction

Load the data set you exported in the final Task of Case Study 2. Eliminate all observations with missing values in the development status variable. As a reminder, this data contains 2020 information on

- median age
- youth unemployment rate

for most world entities. Additional information related to the region, sub-region and development status is also provided for the entities.

First we set the directory where our files are and then we introduce the libraries dplyr and ggplot2.

```
setwd("C:/Users/vaka1/Desktop/Case Study 3")
library(ggplot2)
library(dplyr)
```

Afterwards, we read the csv file with the 2020 information on median age youth unemployment rate using the separator ";", replace the dots in the data with NAs and remove all observations with missing values in the development status variable using the filter command from dplyr package. Finally, we get an idea on the final data.

```
data <- read.csv("file_out.csv", sep = ";")
data[data == "."] <- NA
data <- data %>%
  filter(!is.na(Developed...Developing.Countries))
str(data)
```

```
## 'data.frame':    224 obs. of  10 variables:
##  $ country                          : chr  "Afghanistan" "Albania" "Algeria" "American Samoa" ...
##  $ ISO.3166.2                       : chr  "AF" "AL" "DZ" "AS" ...
##  $ ISO.3166.3                       : chr  "AFG" "ALB" "DZA" "ASM" ...
##  $ Region.Name                      : chr  "Asia" "Europe" "Africa" "Oceania" ...
##  $ Sub.region.Name                  : chr  "Southern Asia" "Southern Europe" "Northern Africa" "Polyne
##  $ Developed...Developing.Countries: chr  "Developing" "Developed" "Developing" "Developing" ...
##  $ median_age                       : chr  "19,5" "34,3" "28,9" "27,2" ...
##  $ youth_unempl_rate                : chr  "17,6" "31,9" "39,3" NA ...
##  $ above_average_median_age         : chr  "no" "no" "yes" "no" ...
##  $ above_average_yu                 : chr  "yes" "yes" "yes" NA ...
```

According to the data, it is observed that the columns median_age and youth_unempl_rate are characters. Also, the numbers contain the "," instead of ".". Thus, the command chartr() is used in order to replace the "," with ".". Finally, the columns are transformed into numeric.

```
data$median_age <- chartr(",", ".", data$median_age)
data$youth_unempl_rate <- chartr(",", ".", data$youth_unempl_rate)
str(data)
```

```
## 'data.frame':    224 obs. of  10 variables:
##  $ country                          : chr  "Afghanistan" "Albania" "Algeria" "American Samoa" ...
##  $ ISO.3166.2                       : chr  "AF" "AL" "DZ" "AS" ...
##  $ ISO.3166.3                       : chr  "AFG" "ALB" "DZA" "ASM" ...
##  $ Region.Name                      : chr  "Asia" "Europe" "Africa" "Oceania" ...
##  $ Sub.region.Name                  : chr  "Southern Asia" "Southern Europe" "Northern Africa" "Polyne
##  $ Developed...Developing.Countries: chr  "Developing" "Developed" "Developing" "Developing" ...
##  $ median_age                       : chr  "19.5" "34.3" "28.9" "27.2" ...
##  $ youth_unempl_rate                : chr  "17.6" "31.9" "39.3" NA ...
##  $ above_average_median_age         : chr  "no" "no" "yes" "no" ...
##  $ above_average_yu                 : chr  "yes" "yes" "yes" NA ...
```

```
data$median_age <- as.numeric(data$median_age)
data$youth_unempl_rate <- as.numeric(data$youth_unempl_rate)
str(data)
```

```
## 'data.frame':    224 obs. of  10 variables:
##  $ country                          : chr  "Afghanistan" "Albania" "Algeria" "American Samoa" ...
##  $ ISO.3166.2                       : chr  "AF" "AL" "DZ" "AS" ...
##  $ ISO.3166.3                       : chr  "AFG" "ALB" "DZA" "ASM" ...
##  $ Region.Name                      : chr  "Asia" "Europe" "Africa" "Oceania" ...
##  $ Sub.region.Name                  : chr  "Southern Asia" "Southern Europe" "Northern Africa" "Polyne
##  $ Developed...Developing.Countries: chr  "Developing" "Developed" "Developing" "Developing" ...
##  $ median_age                       : num  19.5 34.3 28.9 27.2 46.2 15.9 35.7 32.7 32.4 36.6 ...
##  $ youth_unempl_rate                : num  17.6 31.9 39.3 NA NA 39.4 NA NA 23.7 36.3 ...
##  $ above_average_median_age         : chr  "no" "no" "yes" "no" ...
##  $ above_average_yu                 : chr  "yes" "yes" "yes" NA ...
```

Task 1

Using ggplot2, create a density plot of the median age in the developing countries and another superimposed density plot of the median age in the developed countries.

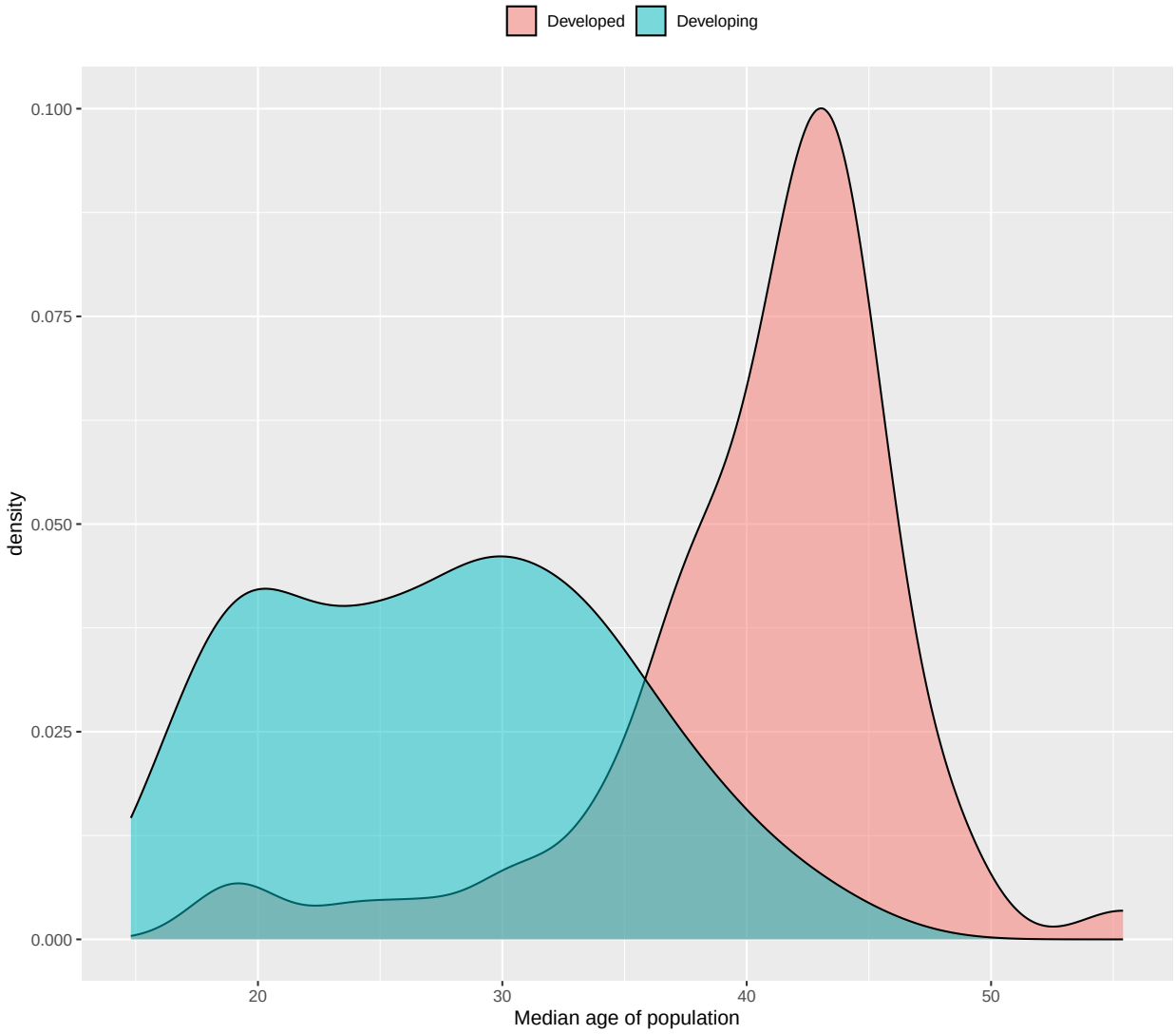
- The color of the density lines is black.
- The area under the density curve should be colored differently among developed vs. developing countries.
- For the colors, choose a transparency level of 0.5 for better visibility.
- Position the legend at the top center of the plot and give it no title (hint: use element_blank()).
- Rename the x axis as "Median age of population"

Comment briefly on the plot.

To create the density plot of the median age, the ggplot2 is used. First, for the density lines, the geom_density() is used with color parameter equals to black and alpha parameter equals to 0.5 for the color

line and transparency level, respectively. Regarding the area under the density curve, in the aesthetics the fill equals to Developed. . . Developing.Countries is used. For the position the legend at the top center of the plot and give it no title, the legend.position = "top" and legend.title = element_blank() is used, respectively. Finally, fir renaming the axis as "Median age of population", the xlab("Median age of population") is used.

```
ggplot(data, aes(x=median_age, fill = Developed...Developing.Countries)) +
  geom_density(color = "black", alpha = 0.5)+
  theme(legend.position = "top",
        legend.title = element_blank())+
  xlab("Median age of population")
```



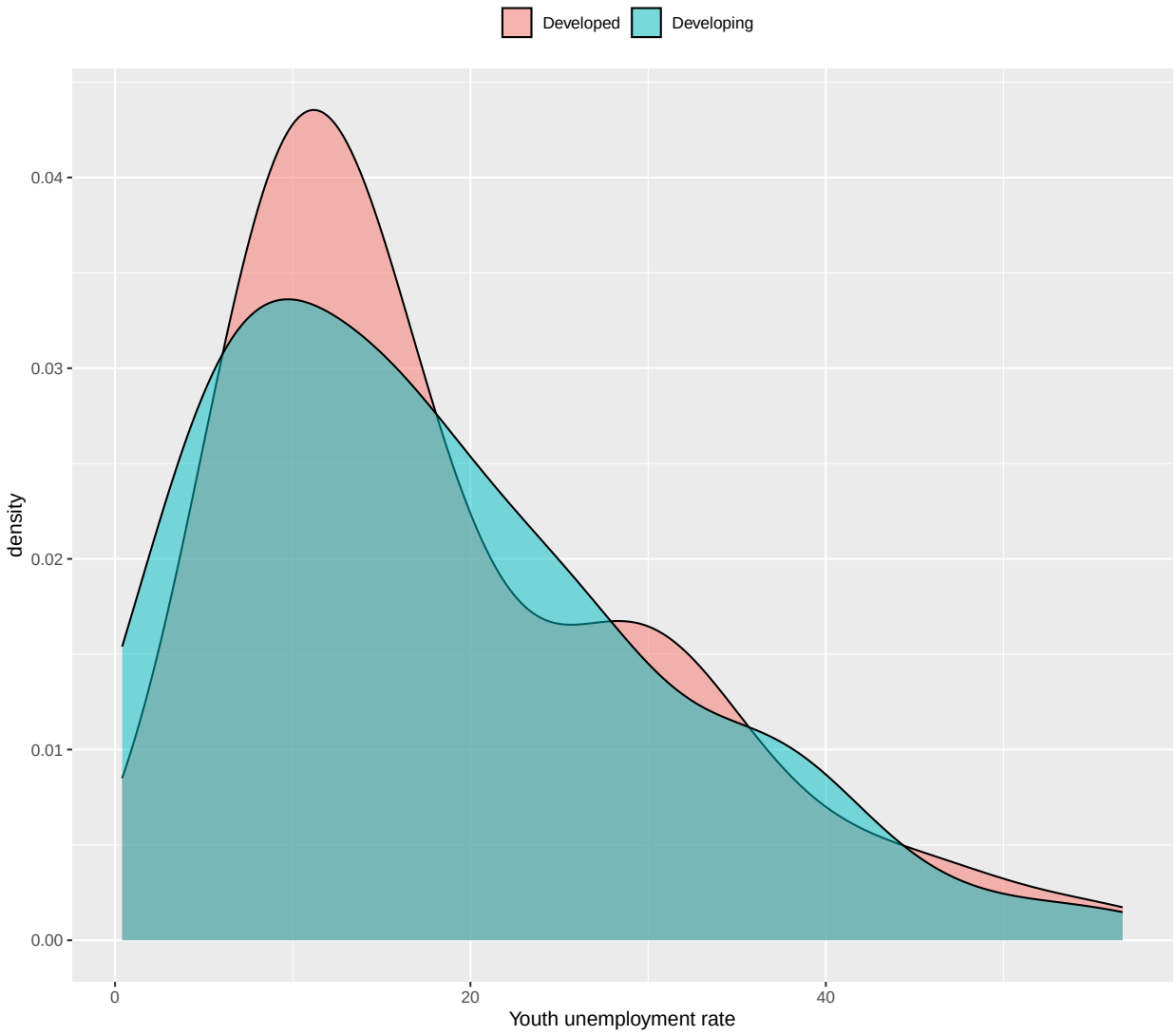
Based on the distribution of the 2 plots, can be seen that the plots are skewed. More specifically, the distribution of the developed countries is right skewed. On the other hand, the distribution of the developing countries is left skewed. Therefore, it is proven that the median age of the developed countries is higher than the developing countries.

Task 2

Using ggplot2, create a plot as in task 1 for the youth unemployment variable. Comment briefly on the plot.

The same thing applies for the youth unemployment rate. The things that change are the x variable in the aesthetics and the xlab(). Also, the youth_unempl_rate variable contains 43 NAs values which are excluded from the plot.

```
ggplot(data, aes(x= youth_unempl_rate, fill = Developed...Developing.Countries)) +
  geom_density(color = "black", alpha = 0.5)+
  theme(legend.position = "top",
        legend.title = element_blank())+
  xlab("Youth unemployment rate")
```



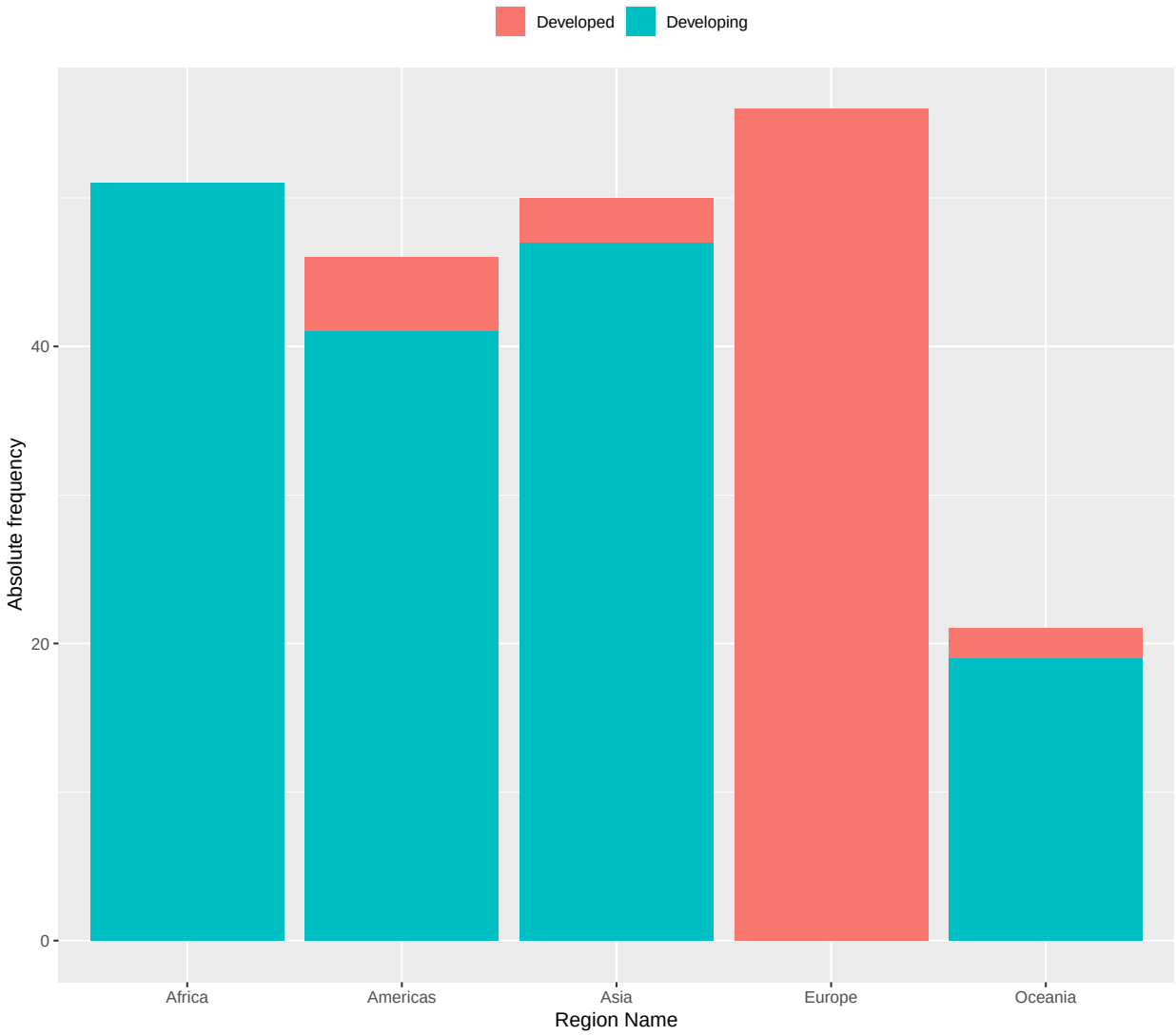
Compared to the previous plot, the youth unemployment rate of developed and developing countries to large extent is similar. This is proven by the plot, where both distribution are left skewed. Therefore, the development status of a countries is unrelated with the unemployment rate.

Task 3

Using ggplot2, create a stacked barplot of absolute frequencies showing how the entities are split into regions and development status. Create another stacked barplot of relative frequencies (height of the bars should be one). Comment briefly on the plots.

Concerning the first plot, in the ggplot() and more specific in the aesthetics the x variable (x axis) is equal to Region name and fill = Developed...Developing.Countries. The geom_bar() command is used in order to create a barplot.

```
ggplot(data, aes(x = Region.Name, fill = Developed...Developing.Countries))+
  geom_bar()+
  theme(legend.position = "top",
        legend.title = element_blank())+
  labs(x = "Region Name", y = "Absolute frequency")
```



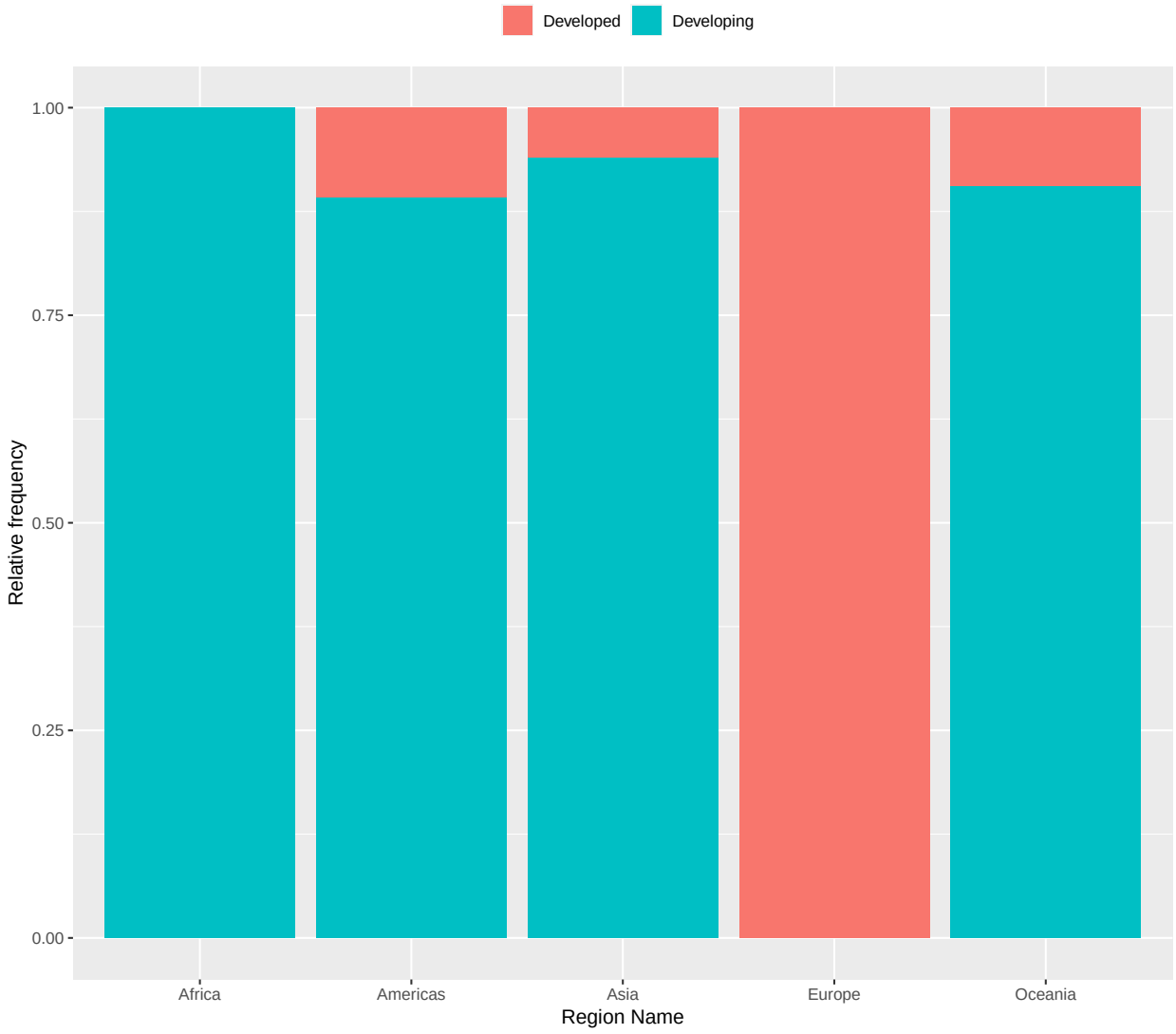
As is obvious from the plot, there are not developing countries in Europe and simultaneously there are not developed countries in Africa. Also, in the rest of the regions (America, Asia and Oceania) most of the

countries' status is developing. Finally, it is clear that the number of developing countries is higher than the developed countries.

However, a major disadvantage of stacked barplot is that it is difficult to accurately compare the individual values within each category. For this reason, another stacked barplot of relative frequencies (height of the bars should be one) is created.

In terms of creating the new stacked barplot, the command group=interaction(Developed...Developing.Countries, Region.Name) is used. This command allows us to plot the relative frequency including the development status of a country and also the region name.

```
ggplot(data, aes(x = Region.Name, group=interaction(Developed...Developing.Countries,
                                                    Region.Name)
                 ,fill = Developed...Developing.Countries))+
  geom_bar(position = "fill")+
  theme(legend.position = "top",
        legend.title = element_blank())+
  labs(x = "Region Name", y = "Relative frequency")
```



According to the plot, the values of each category are now easily comparable, in contrast to the previous plot. Additionally, what applies to the previous diagram also applies to this one, with the only difference being that it is more easily distinguishable.

Task 4

Using ggplot2, create a plot showing the relationship between median age and youth unemployment rate.

- Color the geoms based on the development status.
- Add a regression line for each development status.

First, in the aesthetics of ggplot the x variable (x axis) is the median age, the y variable (y axis) is the youth unemployment rate and the color is equal to development status. In order to show the the relationship between median age and youth unemployment rate, a scatterplot is used (geom_point()). To add a regression line for each development status, the command geom_smooth() is used, including the method = "lm", which indicates the least squares regression line, and se = FALSE, which means that the confidence interval around the regression line is not enabled.

```
ggplot(data,aes(x = median_age, y = youth_unempl_rate,
                color = Developed...Developing.Countries))+
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)+
  theme(legend.position = "top",
        legend.title = element_blank()) +
  labs(x = "Median Age", y = "Youth Unemployment Rate")
```

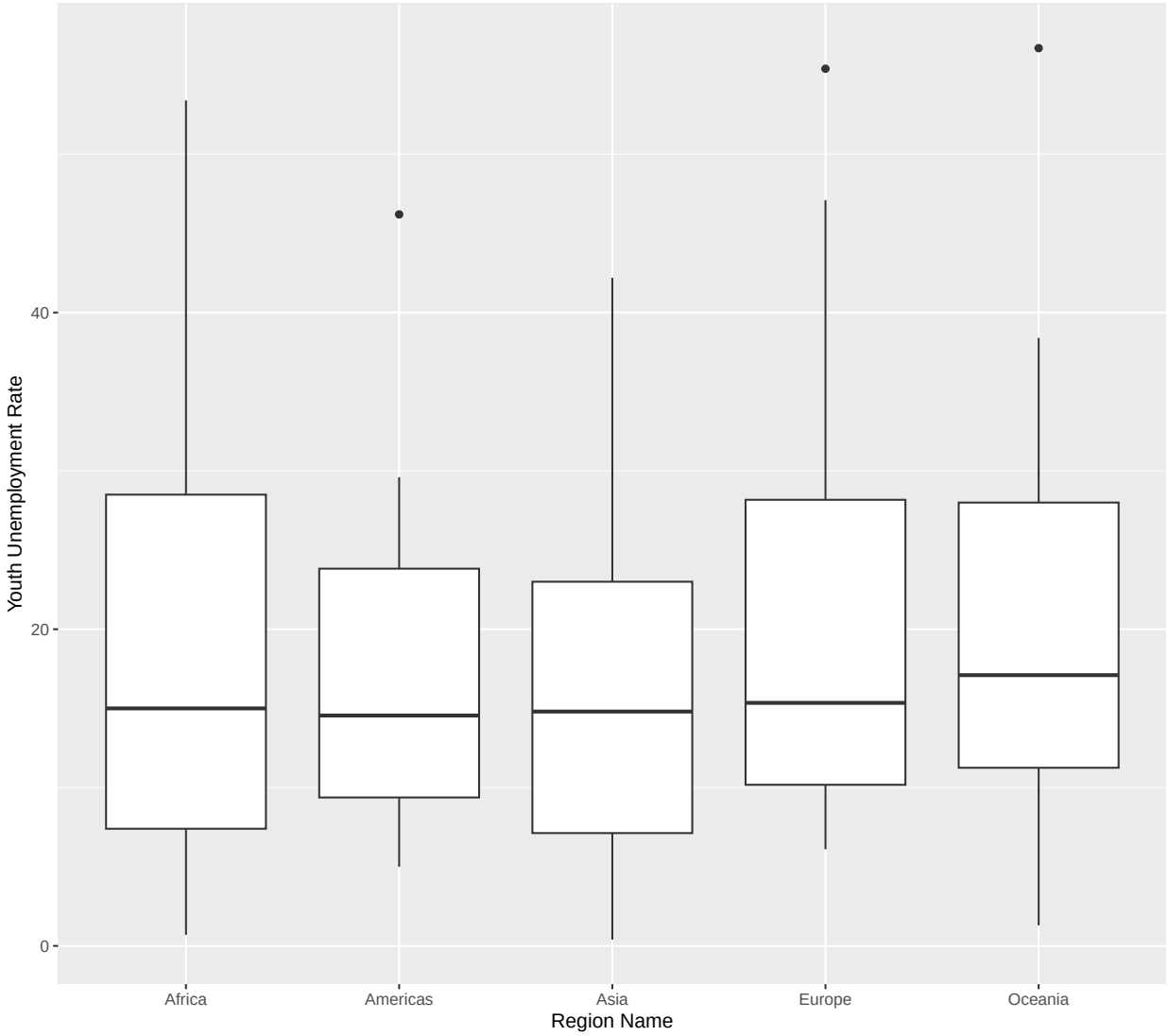
A major advantage of a scatterplot is that the relation of two variables can be observed. According to the plot, it appears that there is no correlation between median age and youth unemployment rate because all the points are unstructured. More specifically, regarding the developing countries, the regression line is closely parallel to the x axis which indicates that there is no correlation between the two variables. On the other hand, based on the regression line for the developed countries, there is a small negative correlation. This indicates that, while the median age increases, the youth unemployment rate decreases.

Task 5

Using base R or ggplot2 create parallel boxplots of the youth unemployment variables for each region. Do you see any striking differences?

For the creation of the boxplot, firstly in the aesthetics, the x variable is equal to region name and the y variable is equal to youth unemployment rate and secondly the geom_boxplot() is used.

```
ggplot(data, aes(x = Region.Name, y = youth_unempl_rate))+
  geom_boxplot()+
  labs(x = "Region Name", y = "Youth Unemployment Rate")
```



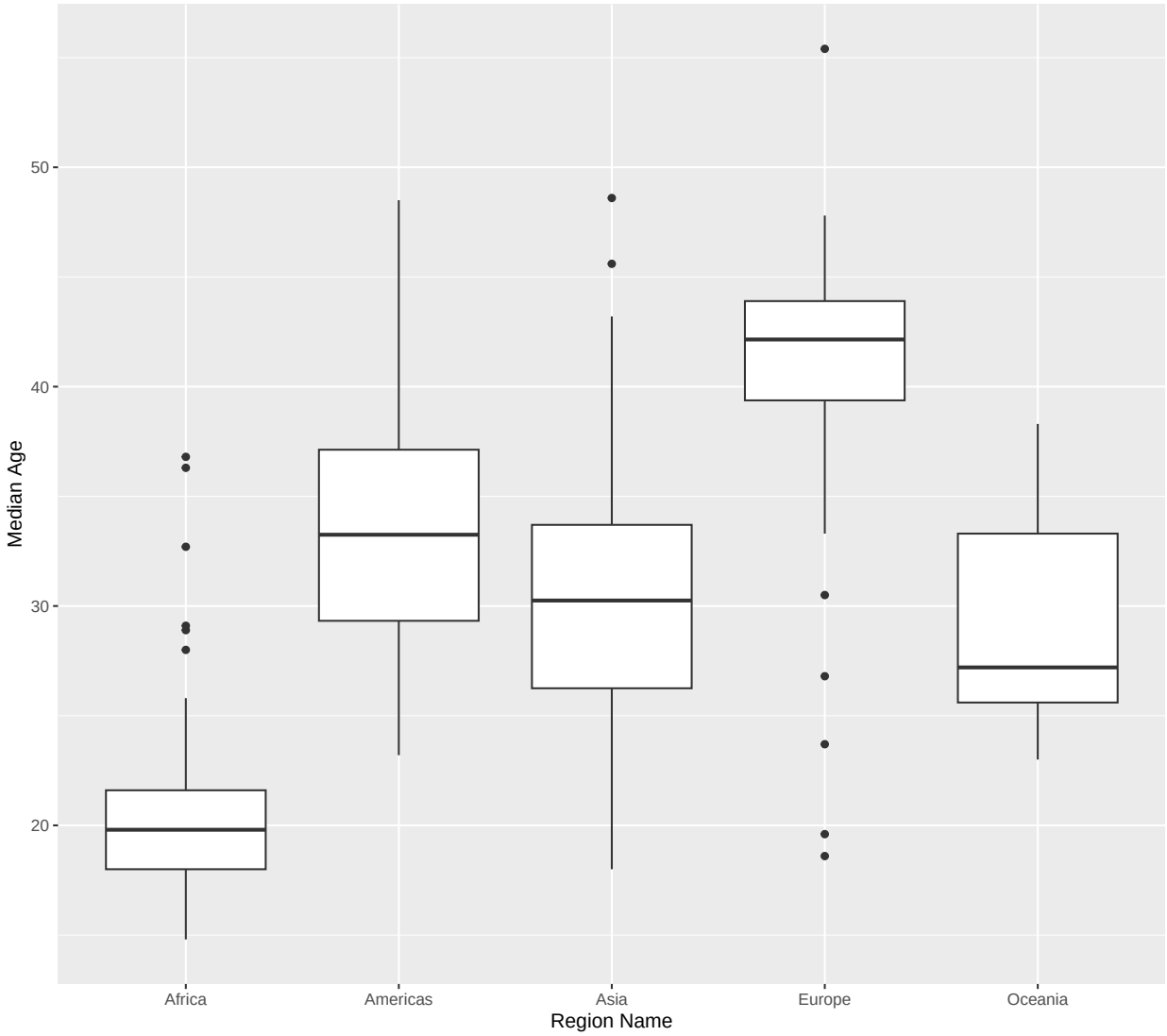
According to the plot, there are not any striking differences for each region. There are only a few differences between the confidence intervals and additionally in America, Europe and Oceania there is one outlier for each one.

Task 6

Create a plot as in Task 5 but for the median population age. Comment briefly.

The same thing is done in Task 6, but this the median age population is used.

```
ggplot(data, aes(x = Region.Name, y = median_age))+
  geom_boxplot()+
  labs(x = "Region Name", y = "Median Age")
```



According to the box plot, it is observed that Europe has the highest median age and Africa the lowest. However, in those two regions there some decisive outliers. The rest of the regions, the median age fluctuates within similar ranges.

It is worth mentioning that according to the plots from the previous tasks, the median age of population in the developed countries is high (density plot from task 1 and scatterplot from task 4). At the same time, most of the developed countries are in Europe (task 3). Therefore, it is reasonable the median age to be high in Europe and this is, also, proven by the box plot above.

Task 7

For each sub-region, calculate the median unemployment rate. Then create a plot which contains the sub-regions on the y-axis and the median unemployment rate on the x-axis.

- As geoms use points.
- Color the points by continent (i.e., Region) - use a colorblind friendly palette (see http://www.cookbook-r.com/Graphs/Colors_(ggplot2)/).

- Rename the axes.
- Using fct_reorder from the forcats package, arrange the levels of subregion such that in the plot the lowest (bottom) region contains the lowest youth unemployment rate and the upper most region contains the highest youth unemployment rate.

The library forcats is introduced.

```
#Introduce the library forcats
library(forcats)
```

Subsequently, a new data frame is created, called median_unempl_rate_region, which contains the median youth unemployment rate, the region name and the subregion name. The data frame is created in the following manner:

1. The NAs observations, according to the youth unemployment variable, are removed using the filter command from the dplyr package.
2. The observations are grouped by the subregion and region name.
3. The median is calculated using the summarize command.
4. The mutate command is used and inside the mutate, the fct_reorder() is used. The reason why we use mutate is to change the values of the Sub.region.Name variable. Also, the fct_reorder() is used to reorder the levels of the sub region variable based on the values of median youth unemployment rate. This is done in order to achieve the lowest (bottom) region to contain the lowest youth unemployment rate and the upper most region to contain the highest youth unemployment rate.

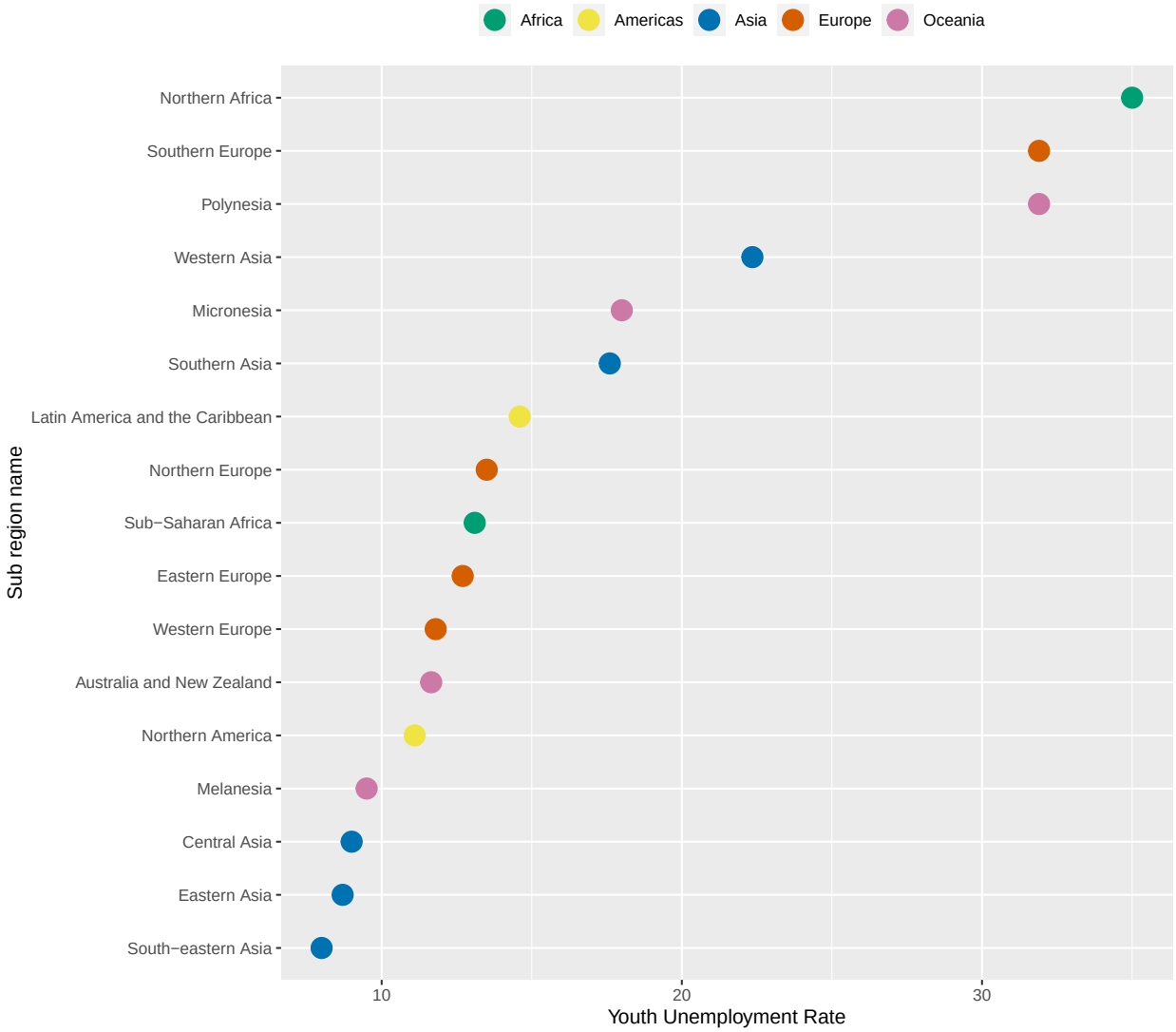
```
#From data we create the median_unempl_rate_region dataset
#Note that the NAs observations were removed
#We group by Sub.region.Name and Region.Name
#Finally we reorder the Sub.region.Name based on the median
median_unempl_rate_region <- data %>%
  filter(!is.na(youth_unempl_rate)) %>%
  group_by(Sub.region.Name, Region.Name) %>%
  summarise(median = median(youth_unempl_rate), .groups = "drop") %>%
  mutate(Sub.region.Name = fct_reorder(Sub.region.Name, median))
```

Based on http://www.cookbook-r.com/Graphs/Colors_(ggplot2)/, the palette for the colorblind friendly is initialized. The choice of colors is random and five colors are selected due to five regions (Africa, America, Asia, Europe and Oceania).

```
#Create the palette for the colorblind based on the exercise
my_palette <- c("#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
```

Finally, we plot the median youth unemployment rate (x axis) for each sub region (y axis) using the palette colors for each region. For the palette, the command scale_colour_manual() is used.

```
ggplot(median_unempl_rate_region, aes(median, Sub.region.Name, color = Region.Name))+
  geom_point(size = 5)+
  scale_colour_manual(values=my_palette)+
  labs(x = "Youth Unemployment Rate", y = "Sub region name")+
  theme(legend.position = "top",
        legend.title = element_blank())
```



According to the plot, northern Africa has the highest median youth unemployment rate and surprisingly the lowest youth unemployment rate can be found in Central, Eastern and South eastern Asia.

Task 8

Go online and find a data set which contains the 2020 population for the countries of the world together with ISO codes. Download this data and merge it to the dataset you are working on in this case study using a left join (A possible source: https://data.worldbank.org/indicator/SP.POP.TOTL?end=2020&start=2020)

From the source provided by the exercise, we download a zip file called API_SP.POP.TOTL_DS2_en_csv_v2_5436324. Inside the zip, the API_SP.POP.TOTL_DS2_en_csv_v2_5436324.csv is included. Using the command read.csv, we read the csv file but we skip the first 4 rows. Finally, we use the command str() to get an idea about the data.

```
#We ignore the first 4 rows of the population csv from the internet
pop_online <- read.csv("API_SP.POP.TOTL_DS2_en_csv_v2_5436324.csv", skip = 4)
```

```
#Get an idea of the data
str(pop_online)
```

```
## 'data.frame':    266 obs. of  67 variables:
##  $ Country.Name  : chr  "Aruba" "Africa Eastern and Southern" "Afghanistan" "Africa Western and Centr
##  $ Country.Code  : chr  "ABW" "AFE" "AFG" "AFW" ...
##  $ Indicator.Name: chr  "Population, total" "Population, total" "Population, total" "Population, tota
##  $ Indicator.Code: chr  "SP.POP.TOTL" "SP.POP.TOTL" "SP.POP.TOTL" "SP.POP.TOTL" ...
##  $ X1960         : num  5.46e+04 1.31e+08 8.62e+06 9.73e+07 5.36e+06 ...
##  $ X1961         : num  5.58e+04 1.34e+08 8.79e+06 9.93e+07 5.44e+06 ...
##  $ X1962         : num  5.67e+04 1.38e+08 8.97e+06 1.01e+08 5.52e+06 ...
##  $ X1963         : num  5.75e+04 1.42e+08 9.16e+06 1.04e+08 5.60e+06 ...
##  $ X1964         : num  5.82e+04 1.46e+08 9.36e+06 1.06e+08 5.67e+06 ...
##  $ X1965         : num  5.88e+04 1.50e+08 9.57e+06 1.08e+08 5.74e+06 ...
##  $ X1966         : num  5.93e+04 1.54e+08 9.78e+06 1.11e+08 5.79e+06 ...
##  $ X1967         : num  5.95e+04 1.58e+08 1.00e+07 1.13e+08 5.83e+06 ...
##  $ X1968         : num  5.95e+04 1.63e+08 1.02e+07 1.16e+08 5.87e+06 ...
##  $ X1969         : num  5.93e+04 1.68e+08 1.05e+07 1.19e+08 5.93e+06 ...
##  $ X1970         : num  5.91e+04 1.72e+08 1.08e+07 1.21e+08 6.03e+06 ...
##  $ X1971         : num  5.88e+04 1.78e+08 1.10e+07 1.24e+08 6.18e+06 ...
##  $ X1972         : num  5.89e+04 1.83e+08 1.13e+07 1.27e+08 6.36e+06 ...
##  $ X1973         : num  5.94e+04 1.88e+08 1.16e+07 1.31e+08 6.58e+06 ...
##  $ X1974         : num  6.00e+04 1.94e+08 1.19e+07 1.34e+08 6.80e+06 ...
##  $ X1975         : num  6.07e+04 1.99e+08 1.22e+07 1.38e+08 7.03e+06 ...
##  $ X1976         : num  6.12e+04 2.05e+08 1.24e+07 1.41e+08 7.27e+06 ...
##  $ X1977         : num  6.15e+04 2.11e+08 1.27e+07 1.45e+08 7.51e+06 ...
##  $ X1978         : num  6.17e+04 2.17e+08 1.29e+07 1.49e+08 7.77e+06 ...
##  $ X1979         : num  6.20e+04 2.24e+08 1.30e+07 1.53e+08 8.04e+06 ...
##  $ X1980         : num  6.23e+04 2.31e+08 1.25e+07 1.58e+08 8.33e+06 ...
##  $ X1981         : num  6.26e+04 2.38e+08 1.12e+07 1.62e+08 8.63e+06 ...
##  $ X1982         : num  6.31e+04 2.45e+08 1.01e+07 1.67e+08 8.95e+06 ...
##  $ X1983         : num  6.37e+04 2.53e+08 9.95e+06 1.72e+08 9.28e+06 ...
##  $ X1984         : num  6.42e+04 2.60e+08 1.02e+07 1.76e+08 9.62e+06 ...
##  $ X1985         : num  6.45e+04 2.68e+08 1.05e+07 1.81e+08 9.97e+06 ...
##  $ X1986         : num  6.46e+04 2.76e+08 1.04e+07 1.86e+08 1.03e+07 ...
##  $ X1987         : num  6.44e+04 2.84e+08 1.03e+07 1.91e+08 1.07e+07 ...
##  $ X1988         : num  6.43e+04 2.93e+08 1.04e+07 1.96e+08 1.11e+07 ...
##  $ X1989         : num  6.46e+04 3.01e+08 1.07e+07 2.01e+08 1.14e+07 ...
##  $ X1990         : num  6.57e+04 3.10e+08 1.07e+07 2.07e+08 1.18e+07 ...
##  $ X1991         : num  6.79e+04 3.19e+08 1.07e+07 2.12e+08 1.22e+07 ...
##  $ X1992         : num  7.02e+04 3.27e+08 1.21e+07 2.18e+08 1.26e+07 ...
##  $ X1993         : num  7.24e+04 3.36e+08 1.40e+07 2.24e+08 1.30e+07 ...
##  $ X1994         : num  7.47e+04 3.44e+08 1.55e+07 2.30e+08 1.35e+07 ...
##  $ X1995         : num  7.70e+04 3.53e+08 1.64e+07 2.36e+08 1.39e+07 ...
##  $ X1996         : num  7.94e+04 3.63e+08 1.71e+07 2.42e+08 1.44e+07 ...
##  $ X1997         : num  8.19e+04 3.72e+08 1.78e+07 2.49e+08 1.49e+07 ...
##  $ X1998         : num  8.44e+04 3.82e+08 1.85e+07 2.55e+08 1.54e+07 ...
##  $ X1999         : num  8.69e+04 3.91e+08 1.93e+07 2.62e+08 1.59e+07 ...
##  $ X2000         : num  8.91e+04 4.02e+08 1.95e+07 2.70e+08 1.64e+07 ...
##  $ X2001         : num  9.07e+04 4.12e+08 1.97e+07 2.77e+08 1.69e+07 ...
##  $ X2002         : num  9.18e+04 4.23e+08 2.10e+07 2.85e+08 1.75e+07 ...
##  $ X2003         : num  9.27e+04 4.34e+08 2.26e+07 2.93e+08 1.81e+07 ...
##  $ X2004         : num  9.35e+04 4.45e+08 2.36e+07 3.01e+08 1.88e+07 ...
```

```
##  $ X2005         : num  9.45e+04 4.57e+08 2.44e+07 3.10e+08 1.95e+07 ...
##  $ X2006         : num  9.56e+04 4.70e+08 2.54e+07 3.19e+08 2.02e+07 ...
##  $ X2007         : num  9.68e+04 4.82e+08 2.59e+07 3.28e+08 2.09e+07 ...
##  $ X2008         : num  9.80e+04 4.96e+08 2.64e+07 3.37e+08 2.17e+07 ...
##  $ X2009         : num  9.92e+04 5.09e+08 2.74e+07 3.46e+08 2.25e+07 ...
##  $ X2010         : num  1.00e+05 5.23e+08 2.82e+07 3.56e+08 2.34e+07 ...
##  $ X2011         : num  1.01e+05 5.38e+08 2.92e+07 3.66e+08 2.43e+07 ...
##  $ X2012         : num  1.02e+05 5.53e+08 3.05e+07 3.77e+08 2.52e+07 ...
##  $ X2013         : num  1.03e+05 5.68e+08 3.15e+07 3.87e+08 2.61e+07 ...
##  $ X2014         : num  1.04e+05 5.84e+08 3.27e+07 3.98e+08 2.71e+07 ...
##  $ X2015         : num  1.04e+05 6.00e+08 3.38e+07 4.09e+08 2.81e+07 ...
##  $ X2016         : num  1.05e+05 6.16e+08 3.46e+07 4.20e+08 2.92e+07 ...
##  $ X2017         : num  1.05e+05 6.33e+08 3.56e+07 4.31e+08 3.02e+07 ...
##  $ X2018         : num  1.06e+05 6.50e+08 3.67e+07 4.43e+08 3.13e+07 ...
##  $ X2019         : num  1.06e+05 6.67e+08 3.78e+07 4.54e+08 3.24e+07 ...
##  $ X2020         : num  1.07e+05 6.85e+08 3.90e+07 4.66e+08 3.34e+07 ...
##  $ X2021         : num  1.07e+05 7.03e+08 4.01e+07 4.78e+08 3.45e+07 ...
##  $ X             : logi  NA NA NA NA NA NA ...
```

Based on the instructions of the exercise, we need the 2020 population. Thus, from all the variables, the country code (i.e. the ISO code) the X2020 variables are selected using the command select. Also, we rename the column X2020 to Population.2020. Finally, we merge our current data set with the new one using left join and as key the ISO codes.

```
#Keep the country code (aka ISO code) column and the X2020 column for the population of 2020
#Change the name of the column X2020 to Year.2020 using the Rename command
pop_online_2020 <- pop_online %>%
  select(Country.Code, X2020) %>%
  rename("Population.2020" = "X2020")

#Merge the 2 data sets (data and pop_online_2020) with left join based on the ISO code
merged_dataset <- data %>% left_join(pop_online_2020, by=c('ISO.3166.3'='Country.Code'))
str(merged_dataset)
```

```
## 'data.frame':    224 obs. of  11 variables:
##  $ country                        : chr  "Afghanistan" "Albania" "Algeria" "American Samoa" ...
##  $ ISO.3166.2                     : chr  "AF" "AL" "DZ" "AS" ...
##  $ ISO.3166.3                     : chr  "AFG" "ALB" "DZA" "ASM" ...
##  $ Region.Name                    : chr  "Asia" "Europe" "Africa" "Oceania" ...
##  $ Sub.region.Name                : chr  "Southern Asia" "Southern Europe" "Northern Africa" "Polyne
##  $ Developed...Developing.Countries: chr  "Developing" "Developed" "Developing" "Developing" ...
##  $ median_age                     : num  19.5 34.3 28.9 27.2 46.2 15.9 35.7 32.7 32.4 36.6 ...
##  $ youth_unempl_rate              : num  17.6 31.9 39.3 NA NA 39.4 NA NA 23.7 36.3 ...
##  $ above_average_median_age       : chr  "no" "no" "yes" "no" ...
##  $ above_average_yu               : chr  "yes" "yes" "yes" NA ...
##  $ Population.2020                : num  38972230 2837849 43451666 46189 77700 ...
```

Task 9

On the merged data set from Task 8, using function ggplotly from package plotly re-create the scatterplot in Task 4 (without the regression lines), but this time make the size of the points proportional to the population. When hovering over the points the name of the country, the values for the age and the youth unemployment

and population should be shown. (Hint: use the aesthetic text = Country. In ggplotly use the argument tooltip = c("text", "x", "y", "size")).

First we installed the plotly package and then we introduce it. Also, the packages install.packages("webshot") and webshot::install_phantomjs() were installed in order to print the interactive plot to the pdf.

```
#Install plotly package
#install.packages("plotly")

#Introduce the library plotly
library(plotly)

#install.packages("webshot")
#webshot::install_phantomjs()
```

The plot from task is assigned to the variable task.4.plot without the without the regression lines, but this time the size of the points are proportional to the population (size = Population.2020). Furthermore, we add the text equals to country in order each point to be assigned with the country name.

```
task.4.plot <- ggplot(merged_dataset,aes(x = median_age, y = youth_unempl_rate,
                                         color = Developed...Developing.Countries))+
  geom_point(aes(text = country, size = Population.2020)) +
  theme(legend.position = "top",
        legend.title = element_blank()) +
  labs(x = "Median Age", y = "Youth Unemployment Rate")
```

The interactive plot using ggploty and tooltip = c("text", "x", "y", "size") is created. The specific tooltip is used in order when hovering over the points the name of the country, the values for the age and the youth unemployment and population should be shown

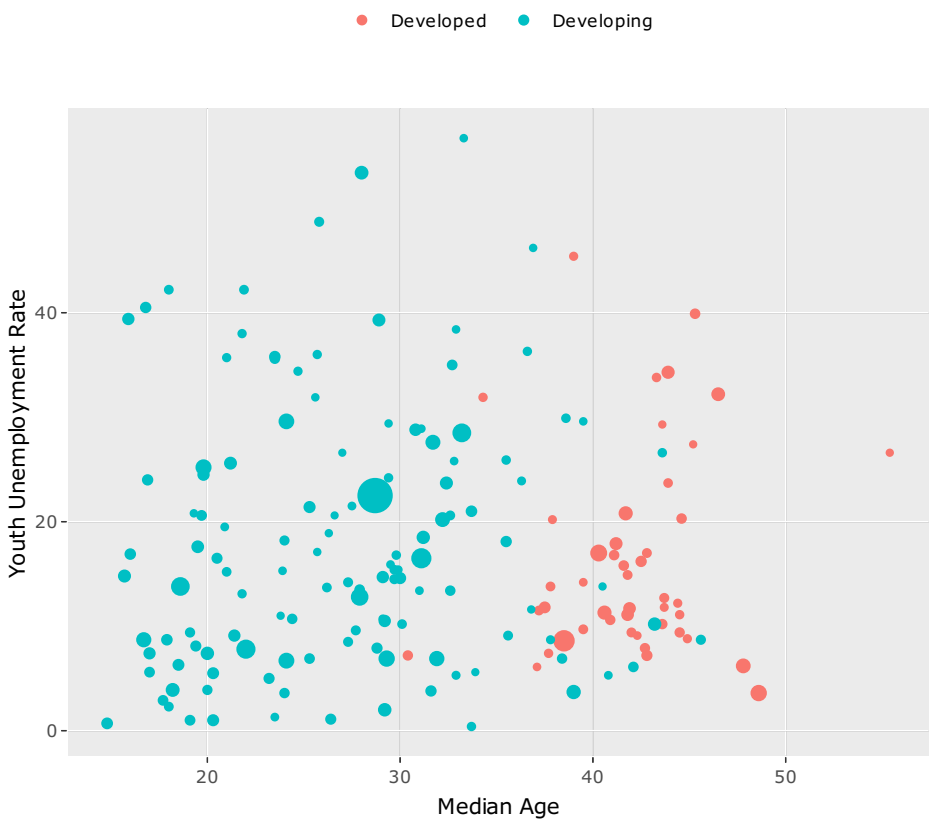
```
#create the interactive plot using ggploty
interactive.plot <- ggplotly(task.4.plot, tooltip = c("text", "x", "y", "size"))
```

Subsequently, we do some modifications on the layout of the plot regarding the legend and the title using the layout() command. We, basically, move the legend to the center and also remove the title of the legend.

```
#Move the legend to the top of the plot and also remove the title
interactive.plot <- layout(interactive.plot, legend = list(orientation = "h",
                                                           x = 0.5,
                                                           y = 1.1,
                                                           xanchor = "center",
                                                           yanchor = "bottom",
                                                           title = ""))
```

Finally, we plot the interactive plot. The main problem here is that the output of the Rmarkdown is a pdf and not a html document. Therefore, we lose the interactivity of the map.

```
interactive.plot
```


Developed
Developing
Youth Unemployment Rate
0
20
40
20
30
40
50
Median Age