

**M2-FA1.1 Data Structures and Visualization (Chapter Project)****I. Installing Packages and Loading Libraries**

To plot the charts and diagrams that will visualize the data, the **ggplot2** library was installed, in addition to the **dplyr** package that was used to manipulate the data after it had been stored in a data frame. To install these packages, the code snippets below were run.

```
install.packages("ggplot2")
install.packages("dplyr")

library(ggplot2)
library(dplyr)
```

Installing and loading the packages above should produce the results in the snippet below.

```
> install.packages("ggplot2")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Anton/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/ggplot2_3.4.1.zip'
Content type 'application/zip' length 4223652 bytes (4.0 MB)
downloaded 4.0 MB
package 'ggplot2' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:/Users/Anton/AppData/Local/Temp/RtmpAh6KQF/downloaded_packages
> install.packages("dplyr")
Error in install.packages : Updating loaded packages
> install.packages("dplyr")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Anton/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/dplyr_1.1.0.zip'
Content type 'application/zip' length 1548314 bytes (1.5 MB)
downloaded 1.5 MB
package 'dplyr' successfully unpacked and MD5 sums checked
warning in install.packages :
cannot remove prior installation of package 'dplyr'
warning in install.packages :
problem copying C:/Users/Anton/AppData/Local/R/win-library/4.2/00LOCKdplyr/lib/x64/dplyr.dll to C:/Users/Anton/AppData/Local/R/win-library/4.2/dplyr/lib/x64/dplyr.dll: Permission denied
warning in install.packages :
restored 'dplyr'

The downloaded binary packages are in
C:/Users/Anton/AppData/Local/Temp/RtmpAh6KQF/downloaded_packages
> library(ggplot2)
warning message:
package 'ggplot2' was built under R version 4.2.3
> 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
```

Since the data that will be used will come from a separate file, a working directory was set to allow the programmers to reference the data set from a specified working directory.

```
> # Set a working directory to store all the related data
> setwd("C:/Users/Anton/Documents/3Q2223/CS174/Module 2/M2 - FA1.1 Data Structures and Visualization (Chapter Project)")
```

**II. Importing the Data in RStudio and Creating a Data Frame to Store the Data**

Now that a proper working directory had been set, the *gapminderDataFiveYear.csv* dataset was imported using the **read.csv()** function and stored into the **countries.df** data frame by running the following code:

```
> # Import data using the read.csv() function and store into a data frame
> countries.df <- read.csv("gapminderDataFiveYear.csv")
```

The figure below shows the data frame created in RStudio after storing the data set into *countries.df*.

	country	year	pop	continent	lifeExp	gdpPercap
1	Afghanistan	1952	8425333	Asia	28.801	779.4453
2	Afghanistan	1957	9240934	Asia	30.332	820.8530
3	Afghanistan	1962	10267083	Asia	31.997	853.1007
4	Afghanistan	1967	11537966	Asia	34.020	836.1971
5	Afghanistan	1972	13079460	Asia	36.088	739.9811
6	Afghanistan	1977	14880372	Asia	38.438	786.1134
7	Afghanistan	1982	12881816	Asia	39.854	978.0114
8	Afghanistan	1987	13867957	Asia	40.822	852.3959
9	Afghanistan	1992	16317921	Asia	41.674	649.3414
10	Afghanistan	1997	22227415	Asia	41.763	635.3414
11	Afghanistan	2002	25268405	Asia	42.129	726.7341
12	Afghanistan	2007	31889923	Asia	43.828	974.5803
13	Albania	1952	1282697	Europe	55.230	1601.0561
14	Albania	1957	1476505	Europe	59.280	1942.2842
15	Albania	1962	1728137	Europe	64.820	2312.8890
16	Albania	1967	1984060	Europe	66.220	2760.1969
17	Albania	1972	2263554	Europe	67.690	3313.4222
18	Albania	1977	2509048	Europe	68.930	3533.0039
19	Albania	1982	2780097	Europe	70.420	3630.8807
20	Albania	1987	3075321	Europe	72.000	3738.9327
21	Albania	1992	3326498	Europe	71.581	2497.4379
22	Albania	1997	3428038	Europe	72.950	3193.0546
23	Albania	2002	3508512	Europe	75.651	4604.2117
24	Albania	2007	3600523	Europe	76.423	5937.0295
25	Algeria	1952	9279525	Africa	43.077	2449.0082
26	Algeria	1957	10677086	Africa	45.685	3013.0760

Showing 1 to 26 of 1,704 entries, 6 total columns

### III. Checking the Structure of the Data Frame

Then, to check the structure of the data frame, the `str()` function passed *countries.df* as its parameter.

```
> # Check the structure of the data frame
> str(countries.df)
'data.frame': 1704 obs. of 6 variables:
 $ country : chr "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
 $ year : int 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
 $ pop : num 8425333 9240934 10267083 11537966 13079460 ...
 $ continent: chr "Asia" "Asia" "Asia" "Asia" ...
 $ lifeExp : num 28.8 30.3 32 34 36.1 ...
 $ gdpPercap: num 779 821 853 836 740 ...
```

The function's output reveals that the data frame contains six attributes, namely *country*, *year*, *pop*, *continent*, *lifeExp*, and *gdpPercap*. The attributes *pop*, *lifeExp*, and *gdpPercap* are *num* data types, while *country* and *continent* are *chr* data types. These attributes' data types will enable an easier analysis of the data as the *num* attributes are compared in terms of the *chr* attributes later.

## IV. Summarizing the Statistics of the Data Frame

A more detailed approach to understanding the data is through the implementation of the `summary()` function, whose output can be seen below. The summary of the data frame reveals helpful statistics that could be used to determine which attributes to compare during the data visualization process. For example, the correlation between attributes *lifeExp* and *gdpPercap* in terms of the attribute *continent* was easily portrayed using a scatter plot because the two former attributes were numerical values. Other data visualizations were also portrayed and will be seen in the succeeding sections of the document.

```
> # Provide summary statistics
> summary(countries.df)
```

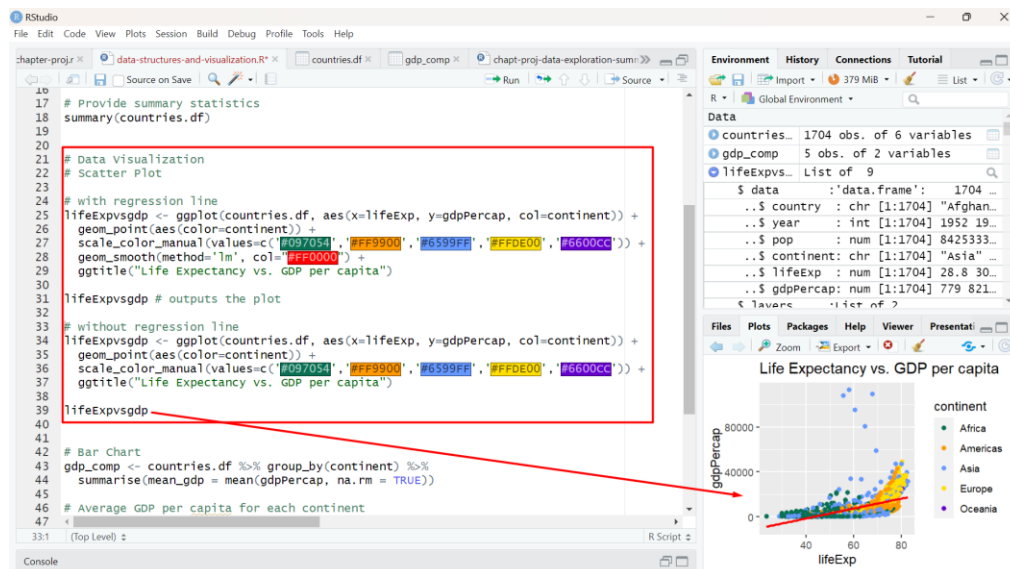
country	year	pop	continent	lifeExp	gdpPercap
Length:1704	Min. :1952	Min. :6.001e+04	Length:1704	Min. :23.60	Min. : 241.2
Class :character	1st Qu.:1966	1st Qu.:2.794e+06	Class :character	1st Qu.:48.20	1st Qu.: 1202.1
Mode :character	Median :1980	Median :7.024e+06	Mode :character	Median :60.71	Median : 3531.8
	Mean :1980	Mean :2.960e+07		Mean :59.47	Mean : 7215.3
	3rd Qu.:1993	3rd Qu.:1.959e+07		3rd Qu.:70.85	3rd Qu.: 9325.5
	Max. :2007	Max. :1.319e+09		Max. :82.60	Max. :113523.1

## V. Visualizing the Data

After the researchers had analyzed and manipulated the data, they found correlations between the attributes and portrayed them through a scatter plot, a bar chart, and a box plot. An analysis of each data visualization is included in the succeeding sections.

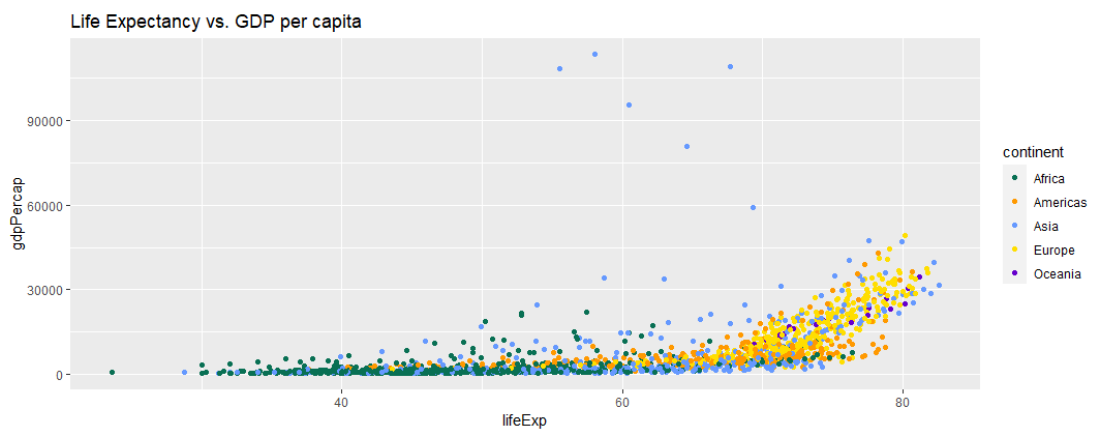
### a. Scatter Plot (Life Expectancy vs. GDP per capita)

The first tool used from the *ggplot2* package is the scatter plot, where functions *ggplot()*, *aes()*, *geom\_point()*, *scale\_color\_manual()*, and *geom\_smooth()* were utilized to portray the relationship between *lifeExp* and *gdpPercap* in terms of each *continent* from the data. A color scale was incorporated into the scatter plot to allow an easier visualizing of which continent had the highest or lowest life expectancy depending on their GDP per capita. The code snippet that produced the scatter plot outputs is enclosed in the figure below.

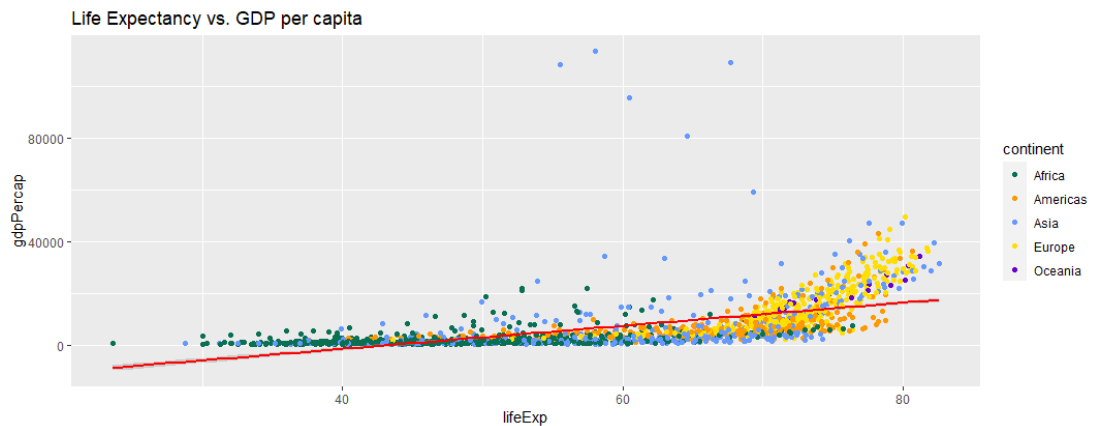


The list below provides a brief discussion of the functions that were used to generate the scatter plots.

- `ggplot()` – parameters include the referenced data frame, attributes *lifeExp* and *gdpPercap* as the *x*- and *y*-values that will be compared and *continent* as the column (legend)
- `geom_point()` – specifies the *continent* as the colored data value
- `scale_color_manual()` – specifies the color for each continent present in the attribute
- `geom_smooth()` - specifies the method and color of the regression line
- `ggtitle()` – parameter contains the title of the plot



**Figure 5a.1** Scatter Plot with no regression line



**Figure 5a.2** Scatter Plot with a regression line

The upward sloping straight regression line on the scatter plot indicates a **positive linear correlation** between attributes *lifeExp* and *gdpPercap*. Hence, as the life expectancy rate in the continents increases, the GDP per capita also increases. These results indicate that continents that have a higher life expectancy rate reach a higher GDP per capita than continents that have a lower life expectancy rate. This statement

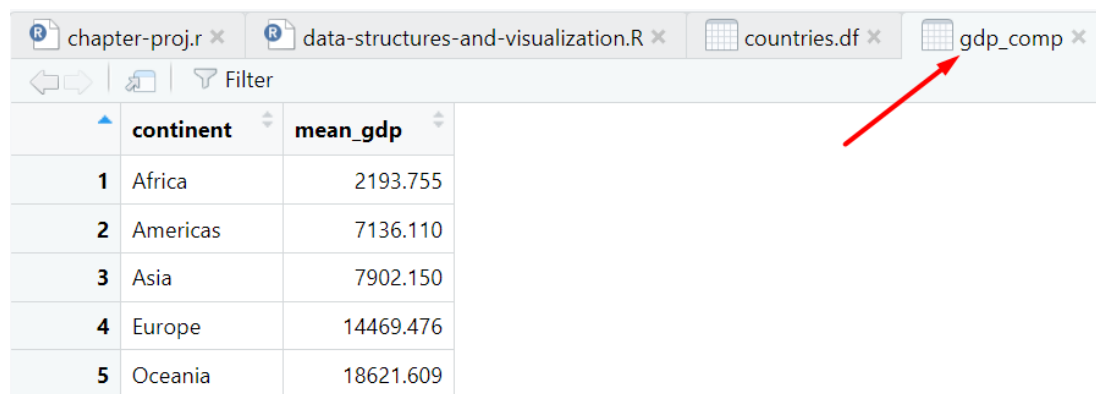
makes sense because a continent is more likely to develop its GDP per capita if people stayed around longer to produce or contribute to the gross value of the continent's economy. A closer look shows another interpretation of the results. The scatter plot reveals that majority of Africans have life expectancy rates ranging from 30 to 58 with GDP per capita that barely moves above the 0 *gdpPercap* line. On the other hand, majority of Europeans have a higher life expectancy ranging from 70 to 80 years with GDP per capita ranging from 10,000 to 40,000. According to these results, people who live in Europe will have a higher life expectancy rate of 70 to 80 years and a higher income, as evidenced by Europe's GDP per capita, than those who live in Africa. Therefore, if people who lived in Africa wanted to live longer and earn more than they earn in Africa, they could migrate to Europe for better healthcare and job opportunities.

#### b. Bar Chart (Average GDP per capita for each continent)

The next correlation to be visualized was that between the average GDP per capita and each continent present in the data. To compute for the average GDP per capita, the following code snippet was run:

```
> gdp_comp <- countries.df %>% group_by(continent) %>%  
+ summarise(mean_gdp = mean(gdpPercap, na.rm = TRUE))
```

The mean value of the attribute *gdpPercap* in terms of the attribute *continent* was calculated using the **mean()** function and stored in the *gdp\_comp* data frame. The *gdp\_comp* data frame contains the continents and their respective mean GDP per capita.

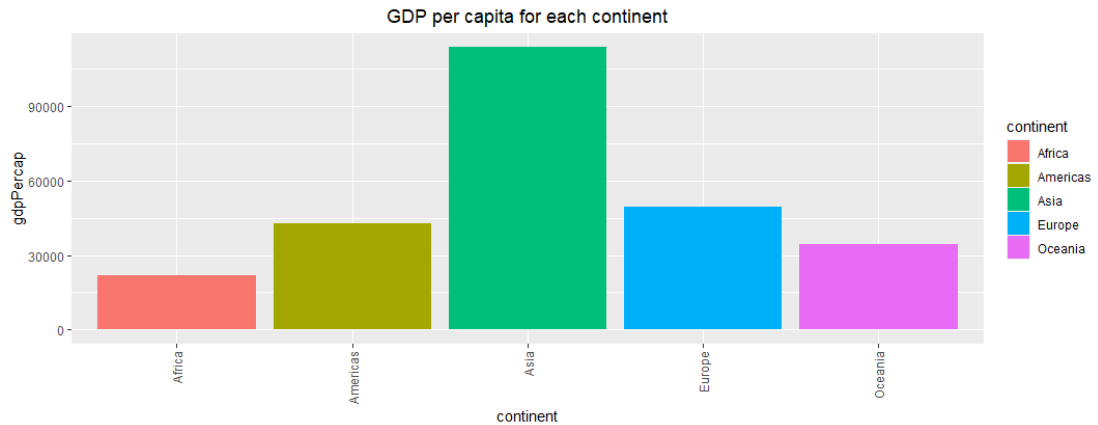


	continent	mean_gdp
1	Africa	2193.755
2	Americas	7136.110
3	Asia	7902.150
4	Europe	14469.476
5	Oceania	18621.609

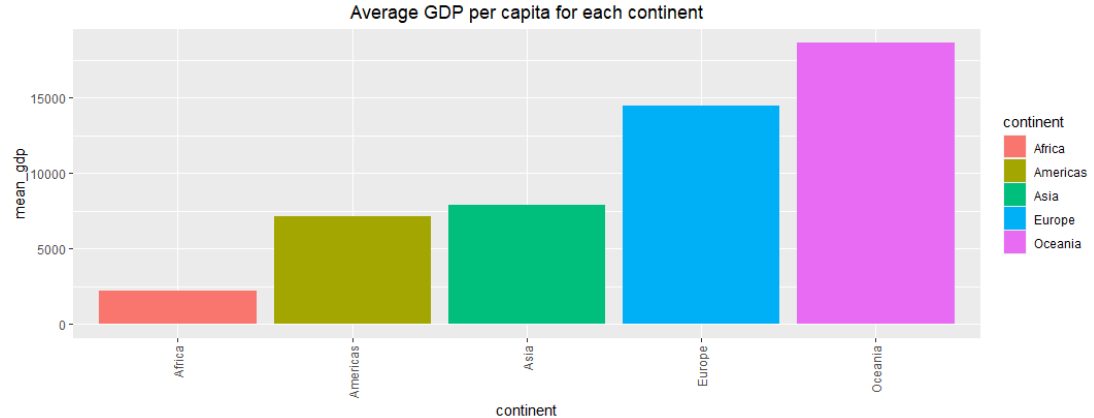
After the average GDP had been calculated and stored into a separate data frame, attributes *continent* and *mean\_gdp* were utilized to create the *Average GDP per capita for each continent bar chart*.

The list below provides a brief discussion of the necessary functions that were used to generate the bar chart.

- `ggplot()` – parameters include the referenced data frame, attributes *continent* and *mean\_gdp* as the *x*- and *y*-values that will be compared and *continent* as the column (legend)
- `ggtitle()` – parameter contains the title of the plot



**Figure 5b.1 Cumulative GDP per capita for each continent**



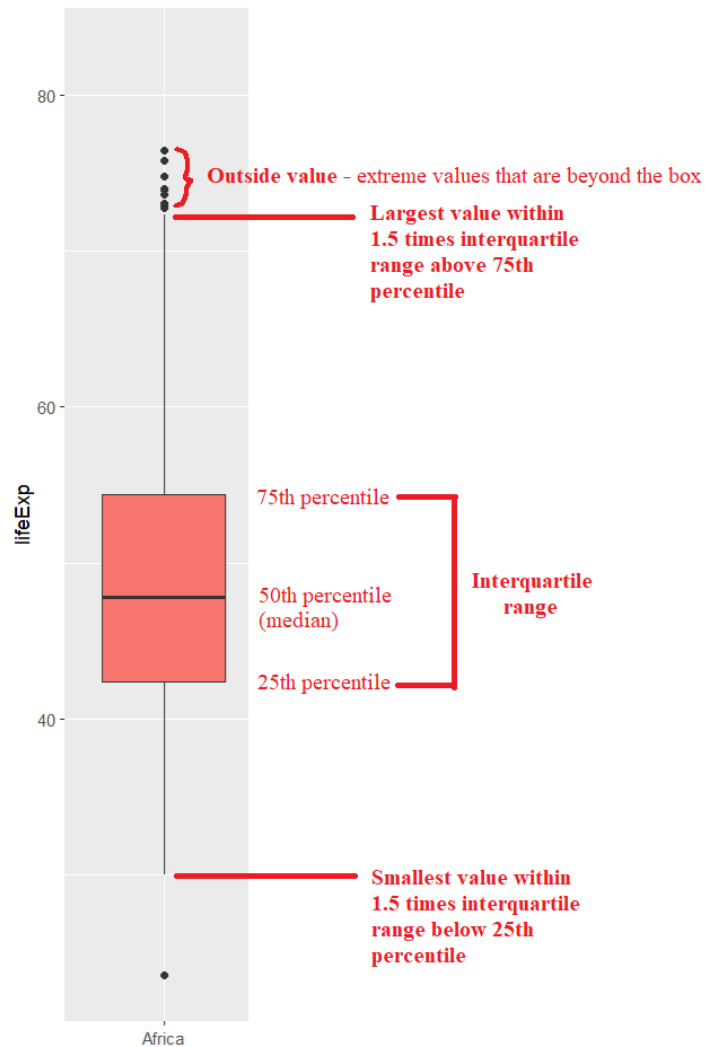
**Figure 5b.2 Average GDP per capita for each continent**

According to Figure 5b.1, the continent with the lowest cumulative GDP per capita is Africa, and the continent with the highest cumulative GDP per capita is Asia. However, when the average GDP values were compared, Oceania became the continent with the highest average GDP per capita, while Africa remained as the continent with the lowest average GDP. These results could indicate that the cost of living in Africa is a lot lower than the other continents as its producers earn barely as much as the other continents. The opposite could be said about Oceania, where the cost of living could be higher as the average gross value of its producers reach nearly 20,000.

### c. Box Plot (Life Expectancy within each Continent)

The last tool used from the *ggplot2* package is the boxplot using *ggplot()*, *geom\_boxplot()*, *ggtitle()*, and *theme()*. As its name suggests, *geom\_boxplot()* is used for creating a box plot in R. Looking at the code snippet below, the box plot was created showing the distribution of life expectancy in each continent.

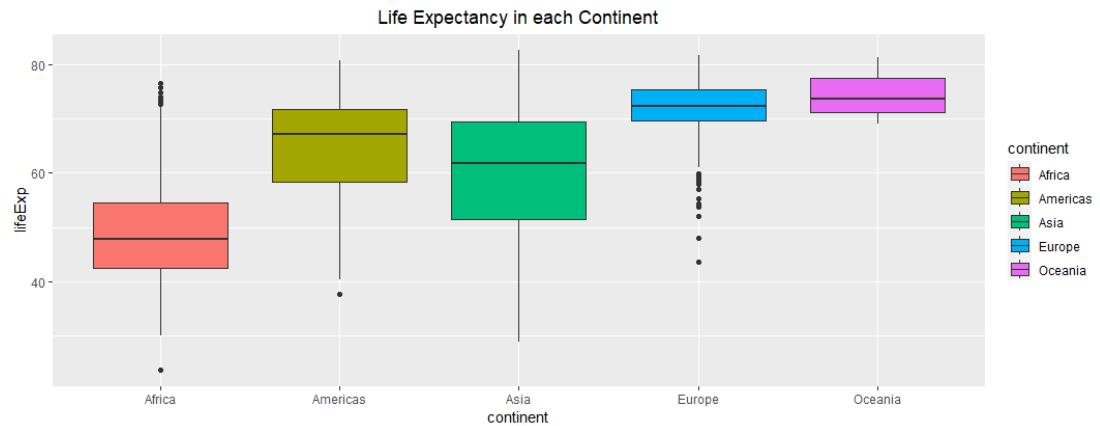
```
# Box Plot
# Life expectancy for each continent
ggplot(countries.df, aes(x = continent, y = lifeExp, fill = continent)) + geom_boxplot() +
  ggtitle("Life Expectancy in each Continent") + theme(plot.title = element_text(hjust = 0.5))
```



**Figure 5c.1 Explanation of the Box Plot**

As shown in the image above, the box in the Box Plot extends from the 25th percentile to the 75th percentile. The 25th percentile is the point where one-quarter of the values are below it. Likewise, the 75th percentile is the point where three-quarters are below it. The difference between the 75th and 25th percentile is called the

interquartile range. The horizontal lines, known as "whiskers," are complicated to explain, but many people define it as extending far enough that anything outside these lines is an extreme value.



**Figure 5c.2** *Life Expectancy in each Continent*

The power of the box plot is that many distributions can be compared at once. Here, the continents are ordered alphabetically. Africa has the lowest average life expectancy, as indicated by its median. However, a few extreme values (values outside the range of the horizontal lines or "whiskers") managed to reach nearly 80 years of age. The second lowest average after Africa is from Asia, whose average life expectancy barely reached 60. The third highest average is from the Americas, followed by Europe, with an average above 70 years of age. Notice that Europe has extreme values ranging from 40 to 60 years. The highest average comes from Oceania, with its upper quartile almost reaching 80.



## References

- Contributors, D. C. (n.d.). *Aggregating and analyzing data with dplyr*.  
<https://datacarpentry.org/R-genomics/04-dplyr.html#:~:text=dplyr%20is%20a%20package%20for,you%20access%20to%20more%20functions.>
- DeCicco, L. (2023, January 9). *Exploring ggplot2 boxplots - Defining limits and adjusting style*. Water Data for the Nation Blog. <https://waterdata.usgs.gov/blog/boxplots/>
- GeeksforGeeks. (2022, December 26). *How to Color Scatter Plot Points in R*.  
<https://www.geeksforgeeks.org/how-to-color-scatter-plot-points-in-r/>
- Glossary DataBank. (n.d.). <https://databank.worldbank.org/metadataglossary/statistical-capacity-indicators/series/5.51.01.10.gdp#:~:text=Long%20definition-.GDP%20per%20capita%20is%20the%20sum%20of%20gross%20value%20added,GDP%20data%20in%20local%20currency.>
- Group by one or more variables — group\_by. (n.d.).  
[https://dplyr.tidyverse.org/reference/group\\_by.html](https://dplyr.tidyverse.org/reference/group_by.html)
- N. (2022, December 20). *Calculate Mean or Average in R*. Spark by {Examples}.  
[https://sparkbyexamples.com/r-programming/calculate-mean-or-average-in-r/#:~:text=The%20mean\(\)%20is%20a,average%20as%20a%20numeric%20value.](https://sparkbyexamples.com/r-programming/calculate-mean-or-average-in-r/#:~:text=The%20mean()%20is%20a,average%20as%20a%20numeric%20value.)
- Numeracy, Maths and Statistics - Academic Skills Kit. (n.d.).  
<https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/types-of-correlation.html>
- Team, D. (2020, September 25). *Box Plot in R Tutorial*.  
<https://www.datacamp.com/tutorial/boxplot-in-r>
- What is str function in R? -. (2022, August 2). ProjectPro.  
<https://www.projectpro.io/recipes/what-is-str-function-r#:~:text=str%20displays%20structures%20of%20R,the%20data%20set%20is%20huge.>