Final_Markdown

AC

2025-11-20

Reading in the Data

We first read the eight different "World Happiness" data sets in using **read.csv()** as provided from Kaggle..

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## 
## 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
```

```r
happy2015 <- read.csv('Data/2015.csv', header = TRUE)
happy2016 <- read.csv('Data/2016.csv', header = TRUE)
happy2017 <- read.csv('Data/2017.csv', header = TRUE)
happy2018 <- read.csv('Data/2018.csv', header = TRUE)
happy2019 <- read.csv('Data/2019.csv', header = TRUE)
happy2020 <- read.csv('Data/2020.csv', header = TRUE)
happy2021 <- read.csv('Data/2021.csv', header = TRUE)
happy2022 <- read.csv('Data/2022.csv', header = TRUE)
```

Data Cleaning

Next, we cleaned each of the datasets individually.

```r
#cleaning the 2015 data set
happy2015 <- rename(happy2015,
                    'Score_2015' = 'Happiness.Score',
                    'GDP_2015' = 'Economy..GDP.per.Capita.',
                    'Social_Support_2015' = 'Family',
                    'Corruption_2015' = 'Trust..Government.Corruption.',
                    'Dystopia_Residual_2015' = 'Dystopia.Residual',
                    'Life_Expectancy_2015' = 'Health..Life.Expectancy.',
                    'Freedom_2015' = 'Freedom',
                    'Generosity_2015' = 'Generosity')
happy2015$Standard.Error <- NULL
happy2015$Happiness.Rank <- NULL

#Cleaning the 2016 dataset
happy2016 <- rename(happy2016,
                    'Score_2016' = 'Happiness.Score',
                    'GDP_2016' = 'Economy..GDP.per.Capita.',
                    'Social_Support_2016' = 'Family',
                    'Corruption_2016' = 'Trust..Government.Corruption.',
                    'Dystopia_Residual_2016' = 'Dystopia.Residual',
                    'Life_Expectancy_2016' = 'Health..Life.Expectancy.',
                    'Freedom_2016' = 'Freedom',
                    'Generosity_2016' = 'Generosity')
happy2016$Lower.Confidence.Interval <- NULL
happy2016$Upper.Confidence.Interval <- NULL
happy2016$Region <- NULL
happy2016$Happiness.Rank <- NULL

#Cleaning the 2017 dataset
happy2017 <- rename(happy2017,
                    'Score_2017' = 'Happiness.Score',
                    'GDP_2017' = 'Economy..GDP.per.Capita.',
                    'Social_Support_2017' = 'Family',
                    'Corruption_2017' = 'Trust..Government.Corruption.',
                    'Dystopia_Residual_2017' = 'Dystopia.Residual',
                    'Life_Expectancy_2017' = 'Health..Life.Expectancy.',
                    'Freedom_2017' = 'Freedom',
                    'Generosity_2017' = 'Generosity')
happy2017$Whisker.high <- NULL
happy2017$Whisker.low <- NULL
happy2017$Happiness.Rank <- NULL

#Cleaning the 2018 dataset
happy2018 <- rename(happy2018,
                    'Country' = 'Country.or.region',
                    'Score_2018' = 'Score',
                    'GDP_2018' = 'GDP.per.capita',
                    'Social_Support_2018' = 'Social.support',
                    'Life_Expectancy_2018' = 'Healthy.life.expectancy',
                    'Freedom_2018' = 'Freedom.to.make.life.choices',
                    'Generosity_2018' = 'Generosity',
                    'Corruption_2018' = 'Perceptions.of.corruption'
                    )

happy2018$Overall.rank <- NULL

#Cleaning the 2019 dataset
happy2019 <- rename(happy2019,
                    'Country' = 'Country.or.region',
                    'Score_2019' = 'Score',
                    'GDP_2019' = 'GDP.per.capita',
                    'Social_Support_2019' = 'Social.support',
                    'Life_Expectancy_2019' = 'Healthy.life.expectancy',
                    'Freedom_2019' = 'Freedom.to.make.life.choices',
                    'Generosity_2019' = 'Generosity',
                    'Corruption_2019' = 'Perceptions.of.corruption'
)
happy2019$Overall.rank <- NULL

#Cleaning the 2020 dataset
happy2020$Logged.GDP.per.capita <-NULL
happy2020$Social.support <-NULL
happy2020$Healthy.life.expectancy <-NULL
happy2020$Freedom.to.make.life.choices <-NULL
happy2020$Generosity <-NULL
happy2020$Perceptions.of.corruption <-NULL
happy2020$Ladder.score.in.Dystopia <-NULL
happy2020$Standard.error.of.ladder.score <-NULL
happy2020$upperwhisker <-NULL
happy2020$lowerwhisker <-NULL
happy2020$Regional.indicator <- NULL


happy2020 <- rename(happy2020,
                    'Country' = 'Country.name',
                    'Score_2020' = 'Ladder.score',
                    'GDP_2020' = 'Explained.by..Log.GDP.per.capita',
                    'Social_Support_2020' = 'Explained.by..Social.support',
                    'Life_Expectancy_2020' = 'Explained.by..Healthy.life.expectancy',
                    'Freedom_2020' = 'Explained.by..Freedom.to.make.life.choices',
                    'Generosity_2020' = 'Explained.by..Generosity',
                    'Corruption_2020' = 'Explained.by..Perceptions.of.corruption',
                    'Corruption_2020' = 'Explained.by..Perceptions.of.corruption',
                    'Dystopia_Residual_2020' = 'Dystopia...residual'
)

#Cleaning the 2021 dataset
happy2021$Logged.GDP.per.capita <-NULL
happy2021$Social.support <-NULL
happy2021$Healthy.life.expectancy <-NULL
happy2021$Freedom.to.make.life.choices <-NULL
happy2021$Generosity <-NULL
happy2021$Perceptions.of.corruption <-NULL
happy2021$Ladder.score.in.Dystopia <-NULL
happy2021$Standard.error.of.ladder.score <-NULL
happy2021$upperwhisker <-NULL
happy2021$lowerwhisker <-NULL
happy2021$Regional.indicator <- NULL

happy2021 <- rename(happy2021,
                    'Country' = 'Country.name',
                    'Score_2021' = 'Ladder.score',
                    'GDP_2021' = 'Explained.by..Log.GDP.per.capita',
                    'Social_Support_2021' = 'Explained.by..Social.support',
                    'Life_Expectancy_2021' = 'Explained.by..Healthy.life.expectancy',
                    'Freedom_2021' = 'Explained.by..Freedom.to.make.life.choices',
                    'Generosity_2021' = 'Explained.by..Generosity',
                    'Corruption_2021' = 'Explained.by..Perceptions.of.corruption',
                    'Dystopia_Residual_2021' = 'Dystopia...residual',
)

#Cleaning the 2022 dataset
happy2022$RANK <-NULL
happy2022$Whisker.high <-NULL
happy2022$Whisker.low <-NULL

happy2022 <- rename(happy2022,
                    'Score_2022' = 'Happiness.score',
                    'GDP_2022' = 'Explained.by..GDP.per.capita',
                    'Social_Support_2022' = 'Explained.by..Social.support',
                    'Life_Expectancy_2022' = 'Explained.by..Healthy.life.expectancy',
                    'Freedom_2022' = 'Explained.by..Freedom.to.make.life.choices',
                    'Generosity_2022' = 'Explained.by..Generosity',
                    'Corruption_2022' = 'Explained.by..Perceptions.of.corruption',
                    'Dystopia_Residual_2022' = 'Dystopia..1.83....residual',
)
```

Checking the Data

Here, we checked the data to make sure the country names were the same in spelling and label.

```
#setdiff(unique(happy2015$Country), unique(happy2016$Country))

#unique(happy2017$Country[grepl("(?=.*o)(?=.*m)(?=.*a)(?=.*l)(?=.*i)",happy2017$Country, ignore.case =
```

We, then, fixed the ones with the different spellings in order to make all of the data uniform.

```
happy2015$Country[happy2015$Country == 'Somaliland region'] <- 'Somaliland Region'
happy2015$Country[happy2015$Country == 'Macedonia'] <- 'North Macedonia'
happy2016$Country[happy2016$Country == 'Macedonia'] <- 'North Macedonia'
happy2017$Country[happy2017$Country == 'Macedonia'] <- 'North Macedonia'
happy2018$Country[happy2018$Country == 'Macedonia'] <- 'North Macedonia'
happy2020$Country[happy2020$Country == 'Macedonia'] <- 'North Macedonia'
happy2022$Country[happy2022$Country == 'Congo'] <- "Congo (Brazzaville)"

happy2022$Country[happy2022$Country == 'Lesotho*'] <- "Lesotho"
happy2022$Country[happy2022$Country == 'Palestinian Territories*'] <- "Palestinian Territories"
happy2022$Country[happy2022$Country == 'Yemen*'] <- "Yemen"
happy2022$Country[happy2022$Country == 'Turkmenistan*'] <- "Turkmenistan"
happy2022$Country[happy2022$Country == 'Rwanda*'] <- "Rwanda"
happy2022$Country[happy2022$Country == 'North Cyprus*'] <- "North Cyprus"
happy2022$Country[happy2022$Country == 'Niger*'] <- "Niger"
happy2022$Country[happy2022$Country == 'Mauritania*'] <- "Mauritania"
happy2022$Country[happy2022$Country == 'Guatemala*'] <- "Guatemala"
```

```
happy2022$Country[happy2022$Country == 'Gambia*'] <- "Gambia"
happy2022$Country[happy2022$Country == 'Comoros*'] <- "Comoros"
happy2022$Country[happy2022$Country == 'Chad*'] <- "Chad"
happy2022$Country[happy2022$Country == 'Botswana*'] <- "Botswana"
happy2022$Country[happy2022$Country == 'Belarus*'] <- "Belarus"
happy2022$Country[happy2022$Country == 'Azerbaijan*'] <- "Azerbaijan"
happy2022$Country[happy2022$Country == 'Eswatini, Kingdom of*'] <- "Swaziland"
happy2022$Country[happy2022$Country == 'Czechia'] <- "Czech Republic"

happy2017$Country[happy2017$Country == 'Hong Kong S.A.R., China'] <- "Hong Kong"
happy2020$Country[happy2020$Country == 'Hong Kong S.A.R. of China'] <- "Hong Kong"
happy2021$Country[happy2021$Country == 'Hong Kong S.A.R. of China'] <- "Hong Kong"
happy2022$Country[happy2022$Country == 'Hong Kong S.A.R. of China'] <- "Hong Kong"

happy2022$Country[happy2022$Country == 'Madagascar*'] <- "Madagascar"
happy2022$Country[happy2022$Country == 'Luxembourg*'] <- "Luxembourg"
happy2022$Country[happy2022$Country == 'Libya*'] <- "Libya"
happy2022$Country[happy2022$Country == 'Liberia*'] <- "Liberia"
happy2022$Country[happy2022$Country == 'Kuwait*'] <- "Kuwait"
happy2022$Country[happy2022$Country == 'Mauritania*'] <- "Mauritania"

happy2017$Country[happy2017$Country == 'Taiwan Province of China'] <- "Taiwan"
happy2020$Country[happy2020$Country == 'Taiwan Province of China'] <- "Taiwan"
happy2021$Country[happy2021$Country == 'Taiwan Province of China'] <- "Taiwan"
happy2022$Country[happy2022$Country == 'Taiwan Province of China'] <- "Taiwan"

happy2019$Country[happy2019$Country == 'Northern Cyprus'] <- "North Cyprus"
happy2018$Country[happy2018$Country == 'Northern Cyprus'] <- "North Cyprus"
happy2022$Country[happy2022$Country == 'Congo'] <- "Congo (Brazzaville)"

happy2019$Country[happy2019$Country == 'Trinidad & Tobago'] <- "Trinidad and Tobago"
happy2018$Country[happy2018$Country == 'Trinidad & Tobago'] <- "Trinidad and Tobago"
```

During this process, we also decided to cut out the variable "Dystopia Residual" as this was not a topic of interest for us.

```
happy2015$Dystopia_Residual_2015 <- NULL
happy2016$Dystopia_Residual_2016 <- NULL
happy2017$Dystopia_Residual_2017 <- NULL
happy2018$Dystopia_Residual_2018 <- NULL
happy2019$Dystopia_Residual_2019 <- NULL
happy2020$Dystopia_Residual_2020 <- NULL
happy2021$Dystopia_Residual_2021 <- NULL
happy2022$Dystopia_Residual_2022 <- NULL
```

Merging the Data

In order to look at only one data set, we merged all of the datasets by country using **merge()**.

```
merged_15_16 <- merge(happy2015, happy2016, by = "Country", all = TRUE)
merged_15_17 <- merge(merged_15_16, happy2017, by = "Country", all = TRUE)
merged_15_18 <- merge(merged_15_17, happy2018, by = "Country", all = TRUE)
```

```
merged_15_19 <- merge(merged_15_18, happy2019, by = "Country", all = TRUE)
merged_15_20 <- merge(merged_15_19, happy2020, by = "Country", all = TRUE)
merged_15_21 <- merge(merged_15_20, happy2021, by = "Country", all = TRUE)
happy_15to22 <- merge(merged_15_21, happy2022, by = "Country", all = TRUE)
```

And we did some more miscellaneous data cleaning to tidy things up.

```
happy_15to22 <- happy_15to22[happy_15to22$Country != "xx", ] # deleted an invalid xx entry
happy_15to22$Country[happy_15to22$Country == 'Swaziland'] <- 'Eswatini'
happy_15to22$Region[c(50, 101, 132, 136)] <- "Sub-Saharan Africa"
happy_15to22$Region[89] <- "Southern Asia"
happy_15to22$Region[c(14, 120)] <- "Latin America and Caribbean"
# manually assigning region to countries with NA for Region
```

Checking the Data Part 2

Because errors can abound in this, we checked our data again to make sure we were getting what we want. For this one, we sought to ensure that the scores in the dataset were all numerical and not character type using **as.numeric()**.

```
# ensuring scores are num and not chr
happy_15to22$Generosity_2022 <- as.numeric(sub(",", ".", happy_15to22$Generosity_2022))
happy_15to22$GDP_2022 <- as.numeric(sub(",", ".", happy_15to22$GDP_2022))
happy_15to22$Social_Support_2022 <- as.numeric(sub(",", ".", happy_15to22$Social_Support_2022))
happy_15to22$Corruption_2018 <- as.numeric(happy_15to22$Corruption_2018)
```

```
## Warning: NAs introduced by coercion
```

```
happy_15to22$Corruption_2022 <- as.numeric(sub(",", ".", happy_15to22$Corruption_2022))
happy_15to22$Life_Expectancy_2022 <- as.numeric(sub(",", ".", happy_15to22$Life_Expectancy_2022))
happy_15to22$Freedom_2022 <- as.numeric(sub(",", ".", happy_15to22$Freedom_2022))
```

Finalizing the Data Cleaning

Right now, we just have one huge dataset of the 8 datasets combined but we only really want the average scores of these years. So we created new columns of the means over the years for our desired variables: “GDP”, “Social Support”, “Generosity”, “Corruption”, “Life Expectancy”, and “Freedom” using **rowMeans()**.

```
happy_15to22$meanGDP <- rowMeans(happy_15to22[, c("GDP_2015","GDP_2016","GDP_2017","GDP_2018",
                                                  "GDP_2019","GDP_2020","GDP_2021","GDP_2022")],
                                 na.rm = TRUE)
happy_15to22$meanSocialSupport <- rowMeans(happy_15to22[,
                                    c("Social_Support_2015","Social_Support_2016","Social_Support_2017"
                                      "Social_Support_2018","Social_Support_2019","Social_Support_2020"
                                      "Social_Support_2021","Social_Support_2022")],
                                           na.rm = TRUE)
happy_15to22$meanGenerosity <-rowMeans(happy_15to22[,c("Generosity_2015","Generosity_2016",
                            "Generosity_2017","Generosity_2018","Generosity_2019",
                            "Generosity_2020","Generosity_2021", "Generosity_2022")],na.rm = TRUE)
happy_15to22$meanCorruption <-rowMeans(happy_15to22[,
                            c("Corruption_2015","Corruption_2016","Corruption_2017",
                            "Corruption_2018","Corruption_2019","Corruption_2020",
                            "Corruption_2021","Corruption_2022")],na.rm = TRUE)
happy_15to22$meanLifeExpectancy <- rowMeans(happy_15to22[,c("Life_Expectancy_2015","Life_Expectancy_201
                           "Life_Expectancy_2017","Life_Expectancy_2018","Life_Expectancy_2019",
                           "Life_Expectancy_2020", "Life_Expectancy_2021","Life_Expectancy_2022")],na.r
happy_15to22$meanFreedom <- rowMeans(happy_15to22[, c("Freedom_2015","Freedom_2016",
                                                      "Freedom_2017","Freedom_2018",
                                                    "Freedom_2019","Freedom_2020","Freedom_2021",
                                                    "Freedom_2022")], na.rm = TRUE)
```

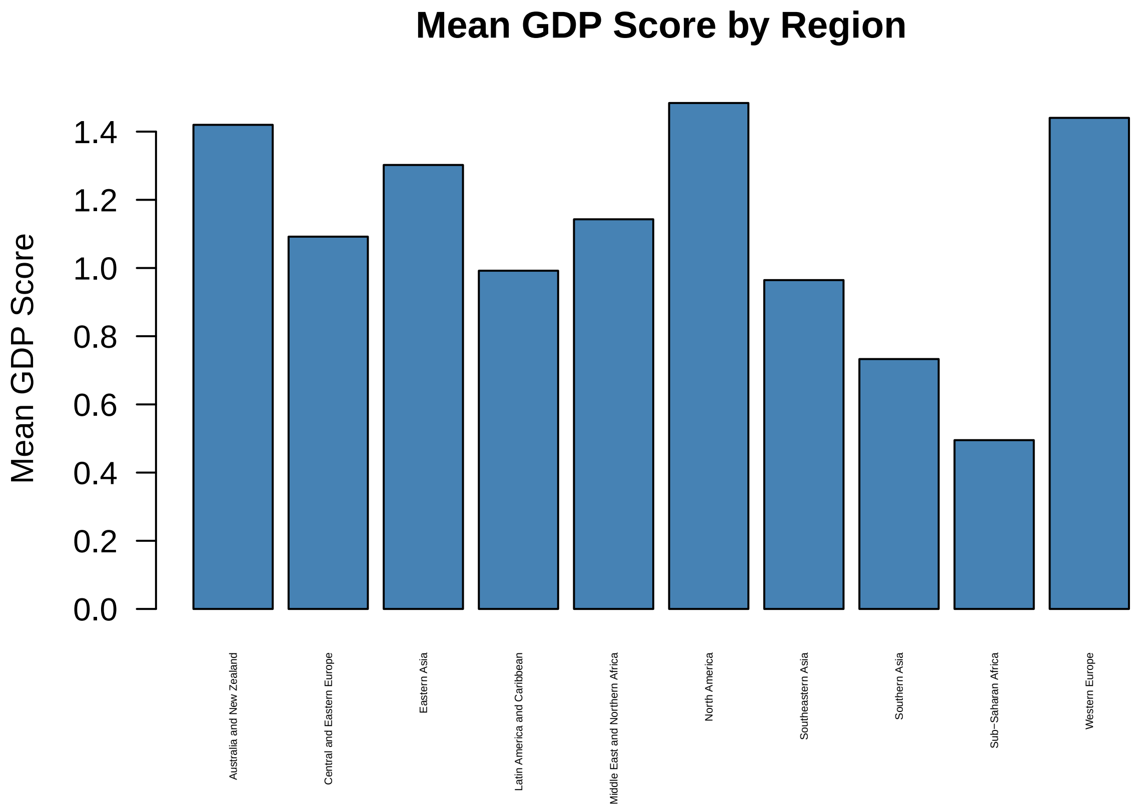
Finally, we created the actual dataset that we will be using to run analyses, which hold only the averaged scores of the desired variables that we created earlier.

```
happy_avg <- happy_15to22[, c("Country", "Region", "meanGDP", "meanSocialSupport", "meanGenerosity","mea

#renamed dataset
happyData <-happy_avg
```

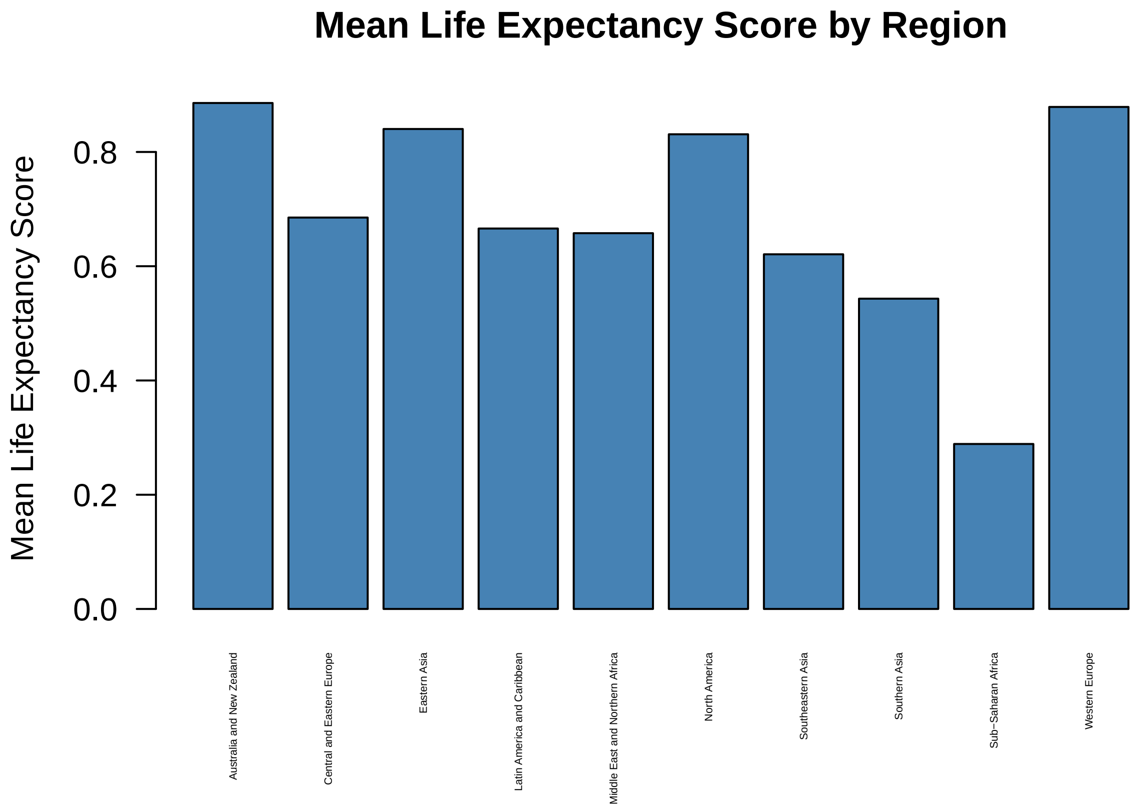
Analyses

First, we made bar charts of the means of each variable for every region. This was just to see if any of them looked like they differed greatly and to get an idea how the regions might differ for each variable. For example, it seems like Sub-Saharan Africa might have a drastically different mean score for GDP as compared to North American. It seems like Southern Asia might have a drastically different mean score for Life Expectancy as compared to Western Europe. And it seems that Central and Eastern Europe might have a drastically different mean score for Freedom as compared to Australia and New Zealand.

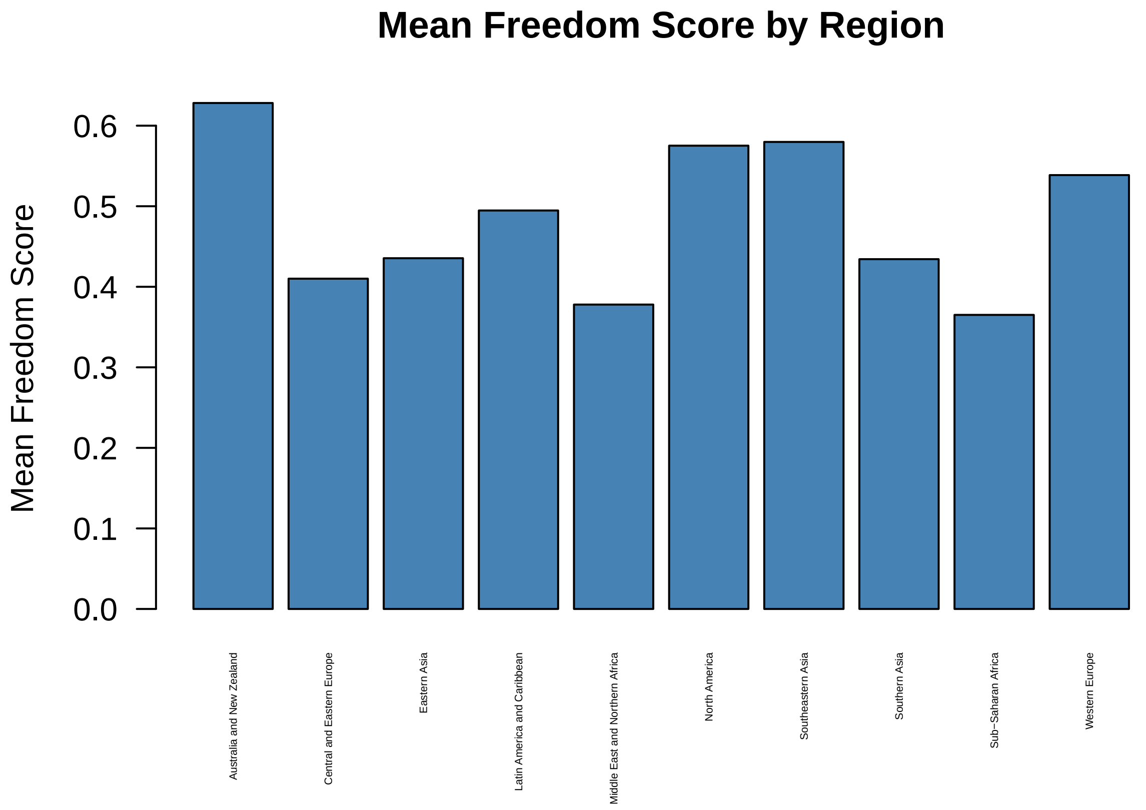
```
#for meanGDP
mean_meanGDP <-tapply(happyData$meanGDP, happyData$Region, mean)
barplot(
  mean_meanGDP,
  col = "steelblue",
  ylab = "Mean GDP Score",
  main = "Mean GDP Score by Region",
  las = 2,  # rotated region names 90 degrees
  cex.names = 0.3 #made region names smaller
)
```



```r
#for meanLifeExpectancy
mean_meanLifeExpectancy <-tapply(happyData$meanLifeExpectancy, happyData$Region, mean)
barplot(
  mean_meanLifeExpectancy,
  col = "steelblue",
  ylab = "Mean Life Expectancy Score",
  main = "Mean Life Expectancy Score by Region",
  las = 2, #rotated region names 90 degrees
  cex.names = 0.3 #made region names smaller
)
```



```r
#for meanFreedom
mean_meanFreedom <-tapply(happyData$meanFreedom, happyData$Region, mean)
barplot(
  mean_meanFreedom,
  col = "steelblue",
  ylab = "Mean Freedom Score",
  main = "Mean Freedom Score by Region",
  las = 2,  # rotated region names 90 degrees
  cex.names = 0.3 #made region names smaller
)
```



While, some of the mean scores do look very different from the other regions, are they actually that much different? To test this, we needed to run some t-tests to see if the means were they were statistically significant in their differences. But first, we needed to create two vectors. One vector assigned regions to numbers and the other was a vector of the variables we ran analyses on: “meanGDP”, “meanFreedom”, and “meanLifeExpectancy”.

```
#Vector that acts as a key assigning regions to a number
regionKey <- c(
  "Southern Asia","Central and Eastern Europe",
  "Middle East and Northern Africa", "Sub-Saharan Africa",
  "Latin America and Caribbean", "Australia and New Zealand",
  "Western Europe", "Southeastern Asia",
  "North America", "Eastern Asia"
)

# Vector of variables to analyze in happyData
vars <- c("meanGDP", "meanFreedom", "meanLifeExpectancy")
```

We, then, ran analyses to see if these mean scores actually did differ or if it was just by chance. In order to do this, however, we needed to create a function using **function()** that would run the t-tests to see if the difference in the scores of the regions was statistically significant and it would return its p-values. But, in order to see the p-values in an organized fashion, we had to create a matrix where the p-values would be put in using a *for loop*. We assigned the matrix’s row and column headers properly and then checked to make sure it was working by viewing it.

```r
# Function to compute t-test p-value
GetPValue <- function(i, j, happyFactor) {
  x <- happyData %>% filter(Region == regionKey[i]) %>% pull(happyFactor)
  y <- happyData %>% filter(Region == regionKey[j]) %>% pull(happyFactor)
  return(t.test(x, y)$p.value)
}

# Create p-value matrices for each variable
pMatrices <- list()

for (var in vars) { #loops factors
  pMat <- matrix(NA, nrow = 10, ncol = 10)#creates empty matrix where p-values will be stored
  for (i in 1:9) {    #loops rows
    for (j in (i+1):10) {  #loops columns
      pMat[i, j] <- GetPValue(i, j, var)
      #passes the two regions to conduct the t.test on for one specific happy factor
    }
  }

  #giving column/row headers w/ country names instead of their numbers using region keys
  rownames(pMat) <- regionKey
  colnames(pMat) <- regionKey

  #storing the completed p-value matrix into the larger list of matrices for every variable
  pMatrices[[var]] <- pMat
}

#View(pMatrices[["meanGDP"]]) #checking the matrix for one factor
```

For the p-values we received, we knew that there was a 5% possibility of getting a false positive so we adjusted it using the FDR method.

```r
adjusted_pMatrices <- lapply(pMatrices, function(mat) {

  # get upper-triangle p-values
  pvals <- mat[upper.tri(mat)]

  # FDR correction
  adj <- p.adjust(pvals, method = "fdr")

  # fill into matrix
  mat_adj <- mat
  mat_adj[upper.tri(mat_adj)] <- adj

  return(mat_adj)
})

# check results
names(adjusted_pMatrices)
```

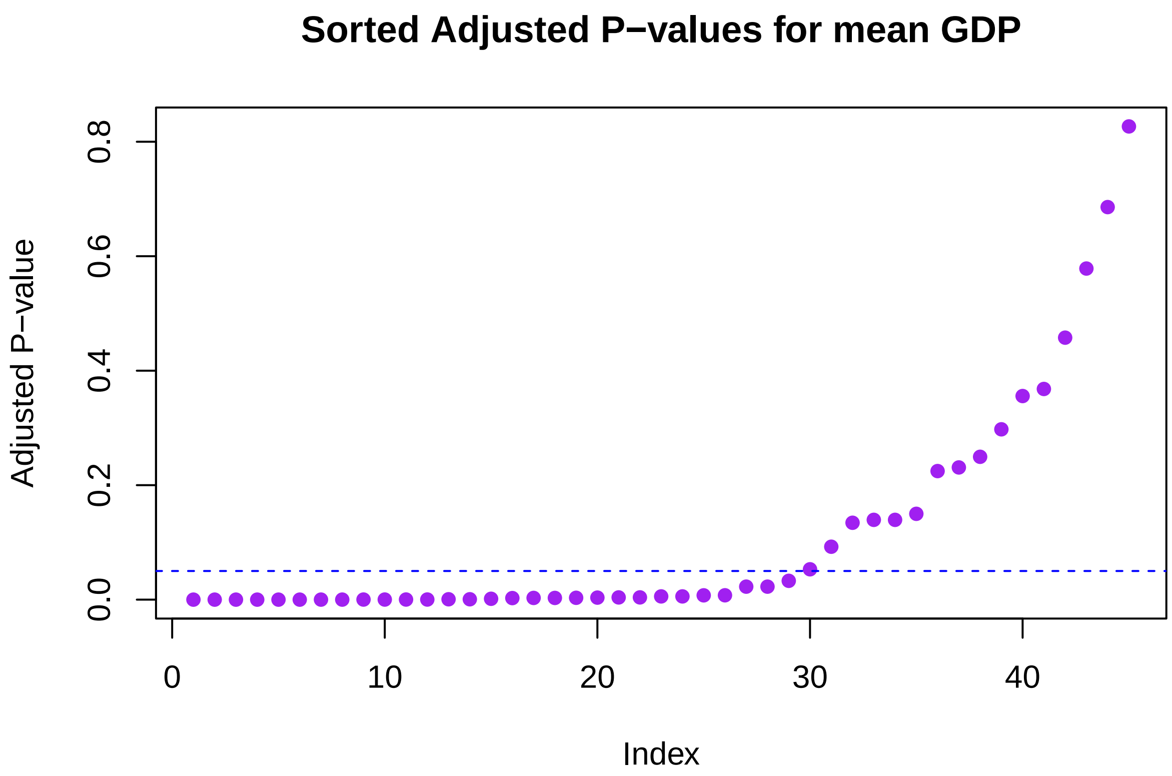
```
## [1] "meanGDP"            "meanFreedom"        "meanLifeExpectancy"
```

```r
#View(adjusted_pMatrices[["meanGDP"]])
#View(adjusted_pMatrices[["meanLifeExpectancy"]])
#View(adjusted_pMatrices[["meanFreedom"]])
```

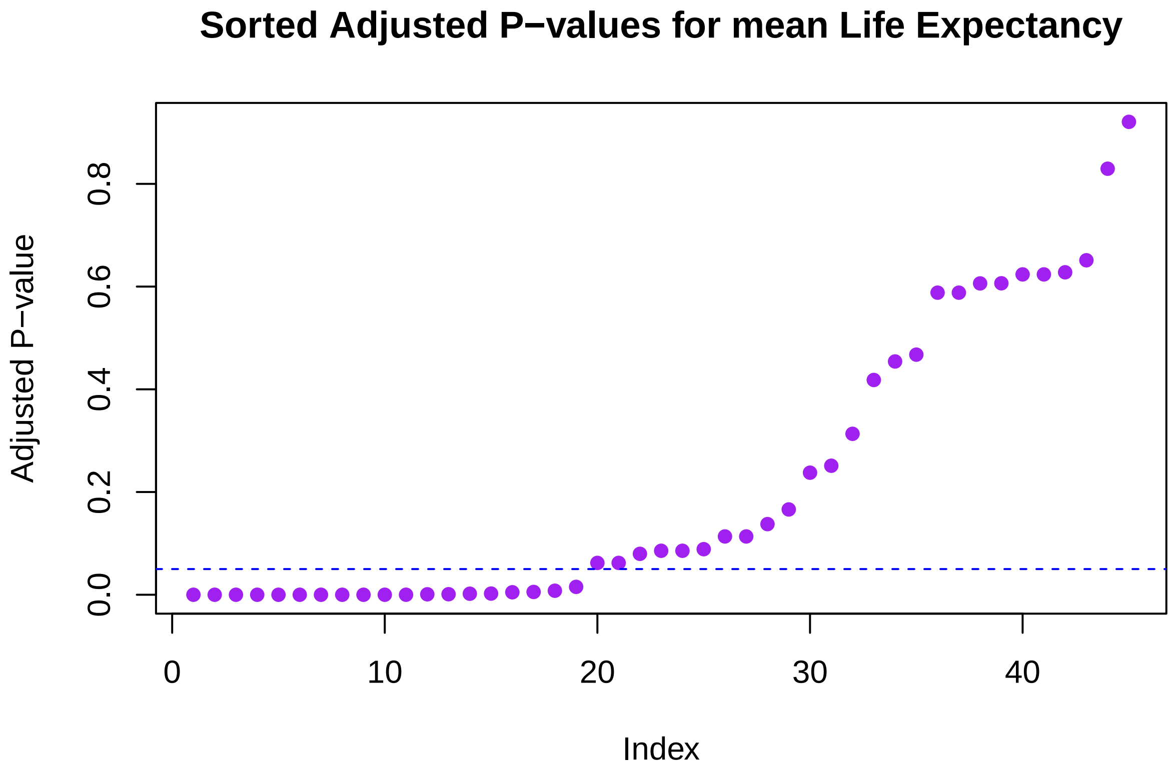
Plots

Now, we plotted the adjusted p-values and added a 0.05 cut off line to see if there were any statistically significant adjusted p-values and if so, how many values were statistically significant.It seems that 29 values are statistically significant for mean GDP score. It seems that 19 values are statistically significant for mean Life Expectancy score. And it seems that 18 values are statistically significant for mean Freedom scores.

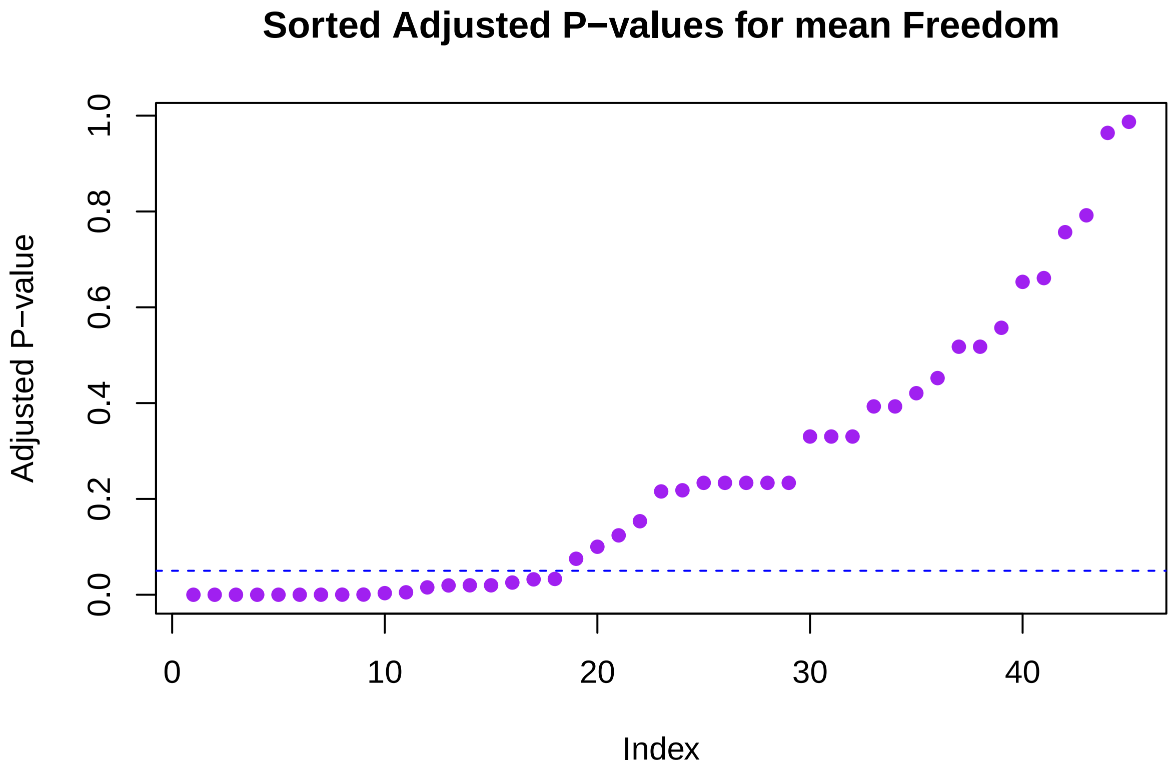
```r
#for mean GDP
adj_p_meanGDP <- adjusted_pMatrices[["meanGDP"]]

# Flattened to a vector
adj_p_vec_meanGDP <- as.vector(adj_p_meanGDP)

# Removed NAs
adj_p_vec_meanGDP <- adj_p_vec_meanGDP[!is.na(adj_p_vec_meanGDP)]

#sorted the values
sorted_adj_p_meanGDP <- sort(adj_p_vec_meanGDP)

#plotted the sorted values
plot(
  1:length(sorted_adj_p_meanGDP),
  sorted_adj_p_meanGDP,
  type = "p",
  pch = 16,
  col = "purple",
  xlab = "Index",
  ylab = "Adjusted P-value",
  main = "Sorted Adjusted P-values for mean GDP"
)

# Added horizontal significance threshold at 0.05
abline(h = 0.05, col = "blue", lty = 2)
```



```r
#for meanLifeExpectancy
adj_p_meanLifeExpectancy <- adjusted_pMatrices[["meanLifeExpectancy"]]

# Flattened to a vector
adj_p_vec_meanLifeExpectancy <- as.vector(adj_p_meanLifeExpectancy)

# Removed NAs
adj_p_vec_meanLifeExpectancy <- adj_p_vec_meanLifeExpectancy[!is.na(adj_p_vec_meanLifeExpectancy)]

#sorted the values
sorted_adj_p_meanLifeExpectancy <- sort(adj_p_vec_meanLifeExpectancy)

#plotted the sorted values
plot(
  1:length(sorted_adj_p_meanLifeExpectancy),
  sorted_adj_p_meanLifeExpectancy,
  type = "p",
  pch = 16,
  col = "purple",
  xlab = "Index",
  ylab = "Adjusted P-value",
  main = "Sorted Adjusted P-values for mean Life Expectancy"
)

# Added horizontal significance threshold at 0.05
abline(h = 0.05, col = "blue", lty = 2)
```



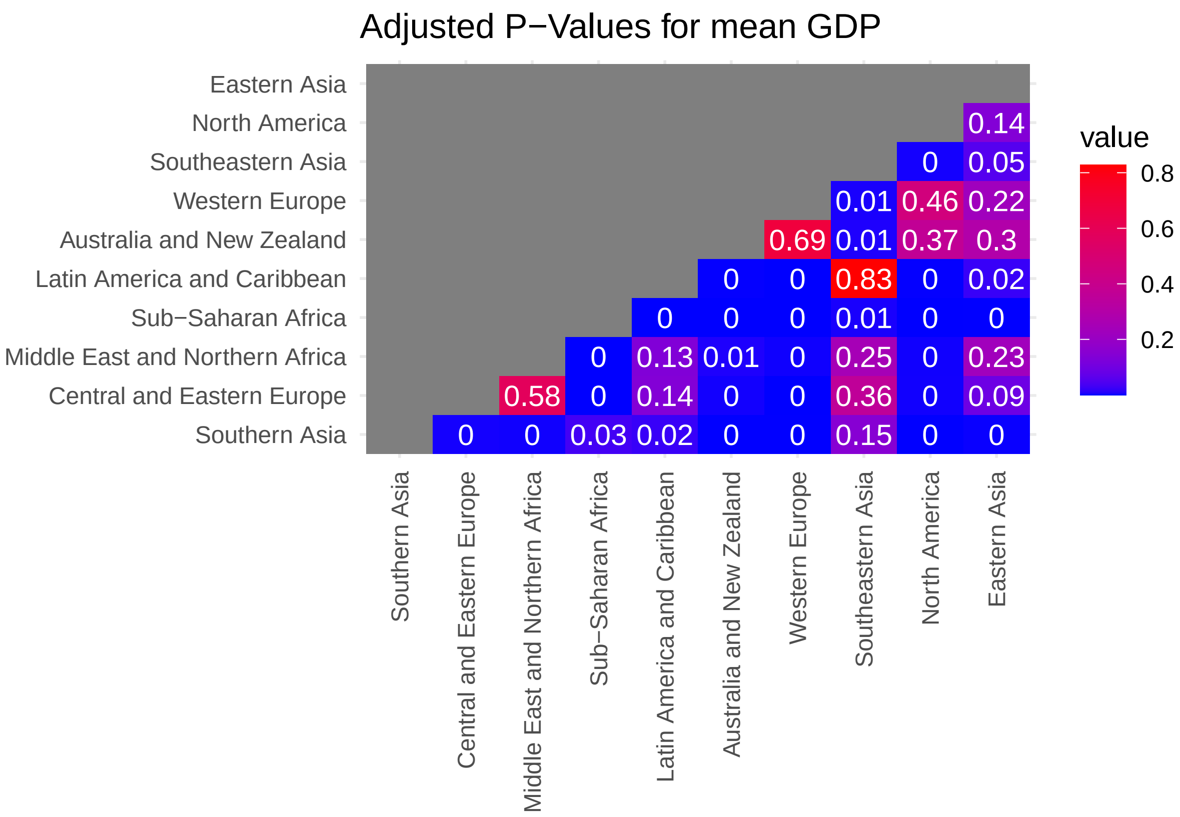
```r
##for mean FREEDOM
adj_p_meanFreedom <- adjusted_pMatrices[["meanFreedom"]]

# Flattened to a vector
adj_p_vec_meanFreedom <- as.vector(adj_p_meanFreedom)

# RemovedNAs
adj_p_vec_meanFreedom <- adj_p_vec_meanFreedom[!is.na(adj_p_vec_meanFreedom)]

#sorted the values
sorted_adj_p_meanFreedom <- sort(adj_p_vec_meanFreedom)

#plotted the sorted values
plot(
  1:length(sorted_adj_p_meanFreedom),
  sorted_adj_p_meanFreedom,
  type = "p",
  pch = 16,
  col = "purple",
  xlab = "Index",
  ylab = "Adjusted P-value",
  main = "Sorted Adjusted P-values for mean Freedom"
)

# Added horizontal significance threshold at 0.05
abline(h = 0.05, col = "blue", lty = 2)
```



But which regions are the statistically significant values for? For this, we created a plot of the adjusted P-values according to each region using **ggplot()**. And we did this for "meanGDP", "meanLifeExpectancy", and "meanFreedom".

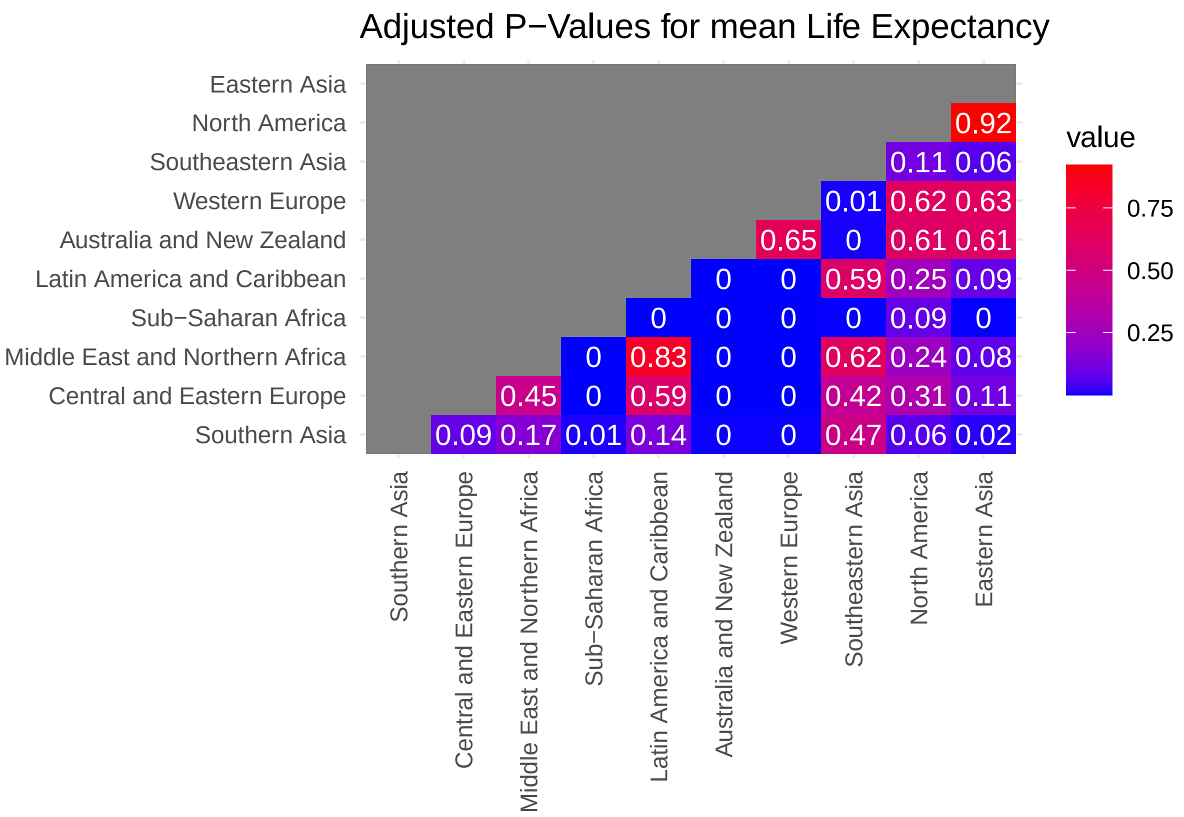
```r
library(ggplot2)
library(reshape2)

#had to melt data for meanGDP
melted_meanGDP <-melt(adjusted_pMatrices[["meanGDP"]])

#plotted adjusted p-values for meanGDP
ggplot(melted_meanGDP, aes(Var2, Var1, fill = value)) +
  geom_tile() +
  geom_text(aes(label = round(value, 2)), color = "white") +
  scale_fill_gradient(low = "blue", high = "red") +
  theme_minimal() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  xlab("") + ylab("") +
  ggtitle ("Adjusted P-Values for mean GDP")
```

```
## Warning: Removed 55 rows containing missing values or values outside the scale range
## (‘geom_text()‘).
```



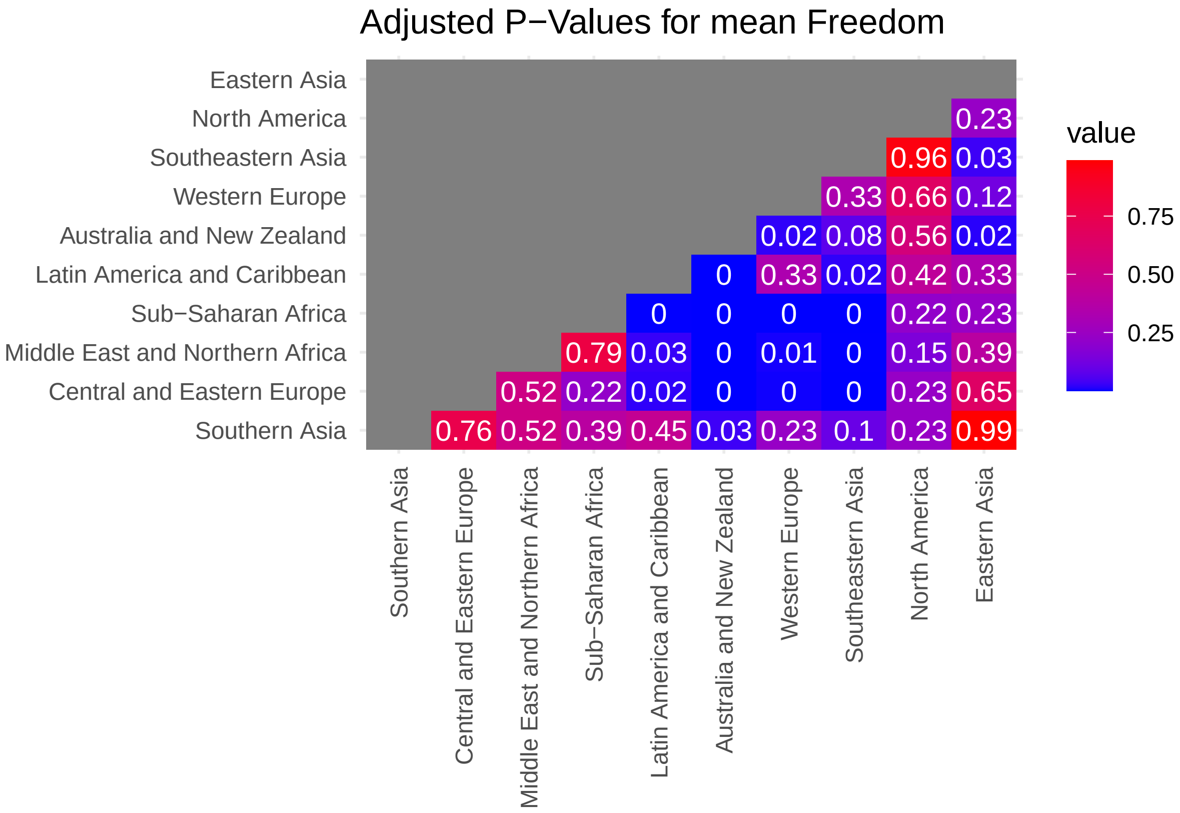
```
#for meanLifeExpectancy
melted_meanLifeExpectancy <- melt(adjusted_pMatrices[["meanLifeExpectancy"]])

ggplot(melted_meanLifeExpectancy, aes(Var2, Var1, fill = value)) +
  geom_tile() +
  geom_text(aes(label = round(value, 2)), color = "white") +
  scale_fill_gradient(low = "blue", high = "red") +
  theme_minimal() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  xlab("") + ylab("") +
  ggtitle ("Adjusted P-Values for mean Life Expectancy")
```

```
## Warning: Removed 55 rows containing missing values or values outside the scale range
## (‘geom_text()‘).
```



```
#for meanFreedom
melted_meanFreedom <- melt(adjusted_pMatrices[["meanFreedom"]])

ggplot(melted_meanFreedom, aes(Var2, Var1, fill = value)) +
  geom_tile() +
  geom_text(aes(label = round(value, 2)), color = "white") +
  scale_fill_gradient(low = "blue", high = "red") +
  theme_minimal() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  xlab("") + ylab("") +
  ggtitle ("Adjusted P-Values for mean Freedom")
```

```
## Warning: Removed 55 rows containing missing values or values outside the scale range
## (‘geom_text()‘).
```



From this, we can clearly see that the data is consistent with the previous plotting of p-values.