Capstone Portafolio

Oscar Pacheco

Contents

1 Overview

Sports are a big part of any culture, in the United States, the sport considered number 1 is football, with its biggest event the Super Bowl happening every year. Followed by Basketball, Baseball, and Soccer. Project One will dive into salaries in the MLS (Major League Soccer), as previously mentioned, soccer is not the main sport in the United States, however this year, there was a major event in the soccer community around the world, the arrival of the superstar Lionel Messi to the MLS. In this project, we will see the salaries rules, as Major League Soccer has very particular rules when it comes to salaries and contracts in comparison to other leagues around the world. We will see the impact that players like Lione Messi, David Beckham, and Andrea Pirlo, among other big names, have in the revenue system for their teams and the mediatic impact. I'll utilize data visualization to observe the distribution of salaries among the teams in the MLS, as well as the distribution of salaries among the position of players, to answer the question of, which player position earns the most money.

For project number two, we will explore data from the World Happiness Report 2023. With the implementation of data analysis and data science, we will perform a linear regression analysis in this data set. The motivation for this project comes from when I first analyzed the data of the Happiness Report 2021 when I wanted to explore which variables contributed the most to the score, especially during the pandemic COVID 19. The project will now focus on the data set for the year 2023, aiming to obtain different results and pinpoint the variables that contribute the most to the Happiness Score post-pandemic in the year 2023. For the analysis, we will use economic variables like the GDP of the countries, combined with social variables like Social Support, Perception of corruption, and some health-related variables like Healthy Life Expectancy. Could the economic variables should have the highest impact on this score? or would it be the social ones?

In September of 2023, I suffered from an Achilles Tendon Rupture. A very painful and stressful situation, I was at the beginning of my last semester for my master's degree, I was in the first week of my first professional internship, and I was an international student. This situation was new for me, the health insurance process, the clinic appointment, and the injury itself. I spent two days trying to communicate with my insurance, my first language is Spanish and sometimes I have a hard time translating specific words into English, especially when it comes to scientific and medical terms. That's why in Project 3, we will explore how an Injury Diagnostic Classification model can be of use and its real-world applications.

All these projects tell a story of my personal background and experiences, as an international student I'm always interested in things happening in different countries, as an ex-soccer player I always keep track of my favorite sport, and my current injury gave me the motivation to utilize data science and analysis to create solutions for real-world problems.

2 Project: MLS Salaries

2.1 Introduction

Let's explore the beginnings of the MLS and key changes that have made the MLS the fourth most-watched sport in the United States. The various pay scales of professional soccer players in the MLS, rules, and exceptions. We will be looking more into detail key information about player salaries, player roles, and team locations. So, if your kids, family, or you dream of a soccer career, you'll know just where to start looking.

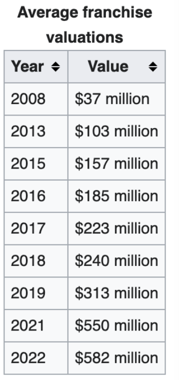
2.2 MLS (Major League Soccer) - Humble Beginnings

The MLS (Major League Soccer) is the highest level for professional soccer players in the United States. It was founded in 1996, it now has 29 teams, of which 26 are in the U.S. and 3 in Canada. MLS experienced financial problems in its first years, losing millions of dollars and having a hard time attracting fans. After 2009, the MLS went into a development stage and the began with expansion of soccer stadiums (they used to play in baseball and football fields that they would adapt to soccer) and academies to create a fanbase and produce American soccer players.

2.3 Game Changer

Around 2007 - 2008 the MLS implemented the rule of the "Designated Player" known also as the **"Beckham Rule"**, after the Los Angeles Galaxy signed a world-class soccer player like David Beckham.[quote]. Some world-class players that came after Beckham were: Cuauhtémoc Blanco, Thierry Henry, Robbie Keane, Tim Cahill, Zlatan Ibrahimovic, Kaká, Andrea Pirlo, Frank Lampard, Steven Gerrard, Didier Drogba, David Villa, Sebastian Giovinco, among other superstars, the newest and most crucial to the MLS profits Lionel Messi. This rule allowed teams to sign star players, which attracted national TV contracts, increased the fan base, and finally made the MLS profitable.With an average attendance of over 20,000 per game, MLS has the **fourth-highest** average attendance of any major professional sports league in the United States and Canada after the NFL, MLB, and Canadian Football League (CFL).(2023b)

.



Average franchise valuations

Year	Value
2008	$37 million
2013	$103 million
2015	$157 million
2016	$185 million
2017	$223 million
2018	$240 million
2019	$313 million
2021	$550 million
2022	$582 million

2.3.1 Salary Cap in the MLS

Each soccer team can have a roster of up to 30 players. For official soccer games, the team has to present 18 players. How does the MLS prevent wealthier teams from just buying 18 world-class players and completely dominating the league? They have a salary "cap" or regulations.(2023c)

Salary Cap for Senior Roster Up to 20 players, occupying roster slots 1-20, count against the club's 2023 Salary Budget of **$5,210,000** and are referred to collectively as the club's Senior Roster. Clubs are not required to fill roster slots 19 and 20, and clubs may spread their entire Salary Budget across 18 Senior

Roster Players. A minimum Salary Budget Charge will be imputed against a club's Salary Budget for each unfilled Senior Roster slot below 18. A club may have no more than 20 players on its Senior Roster, subject to the Season-Ending Injury, Injured List, and Loan exceptions. The Maximum Salary Budget Charge for a single player is $651,250. (See the Allocation Money section below for details on buying down a player's Salary Budget Charge.)

Salary Cap for Supplemental Roster The salaries of players on the Supplemental Roster (slots 21-30) do not count toward a club's Salary Budget. A club may have no more than ten players on its Supplemental Roster, subject to the Season-Ending Injury, Injured List, and Loan exceptions. All Generation Adidas players are Supplemental Roster players during the initial guaranteed term of their contract.(2023a)

Salary Cap for "Slots 21-24" Slots 21-24 may be filled with Senior Minimum Salary Players ($85,444), which may include Homegrown Players, Generation Adidas Players, any specifically designated players eligible for the MLS SuperDraft; or (iv) Homegrown Players earning more than the Senior Minimum Salary subject to the Homegrown Player Subsidy. All players in slots 21-24 must be paid a base salary that is at least the Senior Minimum Salary ($85,444).

Salary Cap for "Slots 25-28" Slots 25-28 may be filled with players earning the Reserve Minimum Salary ($67,360), which may include Homegrown Players, Homegrown Players earning more than the Reserve Minimum Salary subject to the Homegrown Player Subsidy. Reserve Minimum Salary Players must be 24 years or younger during the League Year. These slots may not be filled with Senior Minimum Salary Players (unless they are Homegrown Players subject to the Homegrown Player Subsidy). All players in slots 25-28 must be paid a base salary that is at least the Reserve Minimum Salary ($67,360).

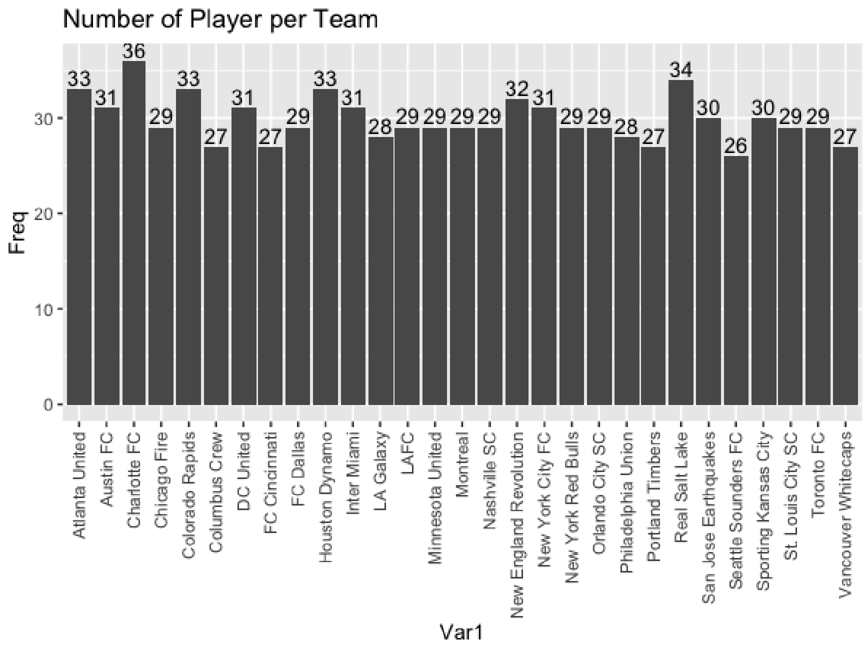
Salary Cap for "Slots 29-30" Slots 29 and 30 must be filled with Homegrown Players earning the Reserve Minimum Salary ($67,360) or earning more than the Reserve Minimum Salary subject to the Homegrown Player Subsidy. All Players in roster slots 29-30 must be paid a base salary which is at least the Reserve Minimum Salary.

2.4 MLS 2022-2023 Data Exploration

For our project, we will be looking into the 2022-2023 MLS salaries data set, which contains 865 observations and six variables in total. Four Categorical variables such as: *"Team Contract"*, *"Last Name"*, *"First Name"* and *"General Position"*. and Two numerical variables such as: *"Base Salary"* and *"Calculated Guaranteed Compensation"*

```
## tibble [865 x 6] (S3: tbl_df/tbl/data.frame)
##  $ Player: Team Contract             : chr [1:865] "Atlanta United" "Atlanta United" "Atlanta United
##  $ Player: Last Name                 : chr [1:865] "McFadden" "Fortune" "Sejdic" "Gutman" ...
##  $ Player: First Name                : chr [1:865] "Aiden" "Ajani" "Amar" "Andrew" ...
##  $ Player: General Position          : chr [1:865] "D" "M" "D" "D" ...
##  $ Base Salary                       : num [1:865] 85444 67360 85444 350000 600000 ...
##  $ Calculated Guaranteed Compensation: num [1:865] 85444 67360 92111 381250 612500 ...
```

We can see that the maximum amount of players is 36 and the least is 27. This number varies as explained before, there are rules for homegrown players, international players, senior roster, etc. This information is helpful as it allows us to move on to more specific analysis and have confidence that our results won't be biased or skewed.



2.5 MLS 2022-2023 - Salary

As you noticed, in our data set we have two different categories for "Salary", the first one is **"Base Salary"**, and the second one is **"Calculated Guaranteed Compensation"**. The Base Salary is the quantity stated in each player's contract. The Calculated Guaranteed Compensation includes a player's base salary and all signing and guaranteed bonuses annualized over the term of the player's contract, including option years.

Calculated Guaranteed Compensation can be defined as follows:

$$GuaranteedCompensation = Salary + (Bonus/Contract)$$

For example, a player earning an annual base salary of $500,000, whose contract has an initial term of two years with two one-year options and received a $100,000 signing bonus, his average annual guaranteed compensation would be $525,000 (base salary plus signing bonus ($100,000), with the signing bonus divided by the number of years covered by the contract (4)).

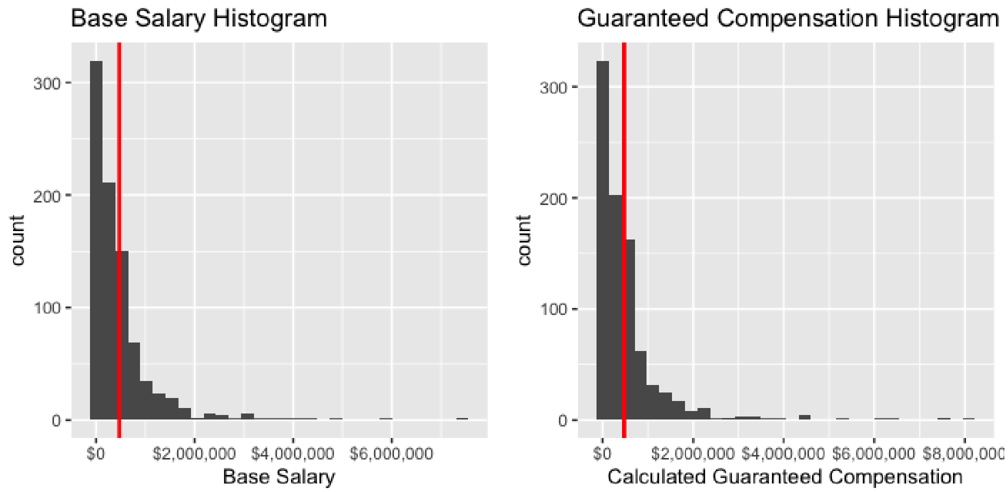
$$GuaranteedCompensation = 500,000 + (100,000/4)$$

$$GuaranteedCompensation = 500,000 + 25,000$$

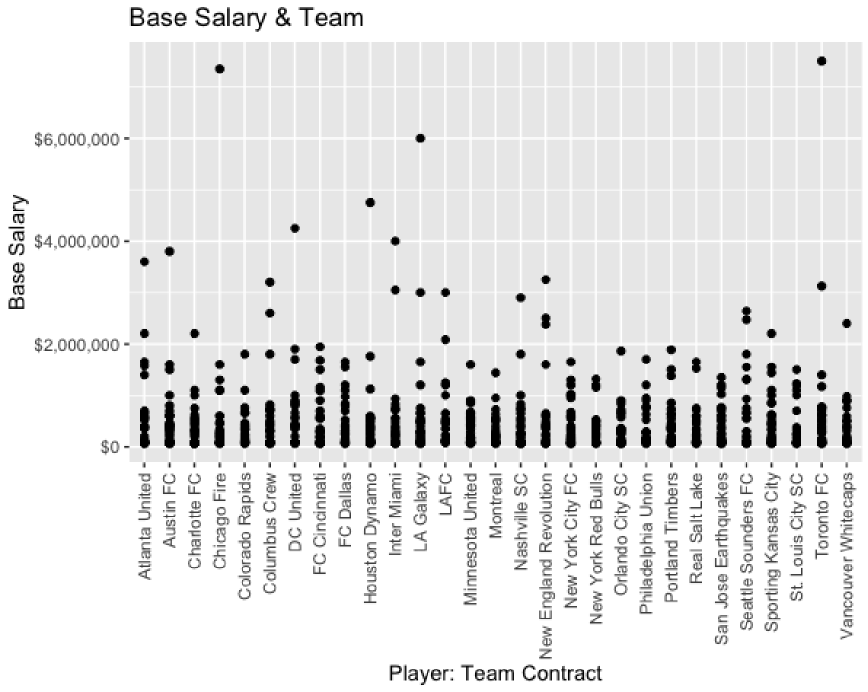
$$GuaranteedCompensation = 525,000$$

The Average Annual Guaranteed Compensation figure also includes any marketing bonus and any agent's fees, both annualized over the term of the contract. The Average Annual Guaranteed Compensation figure does not include Performance Bonuses because there is no guarantee that the player will hit those bonuses.

These figures include compensation from each player's contract with MLS. They do not include any compensation from any contracts with individual teams or their affiliates.



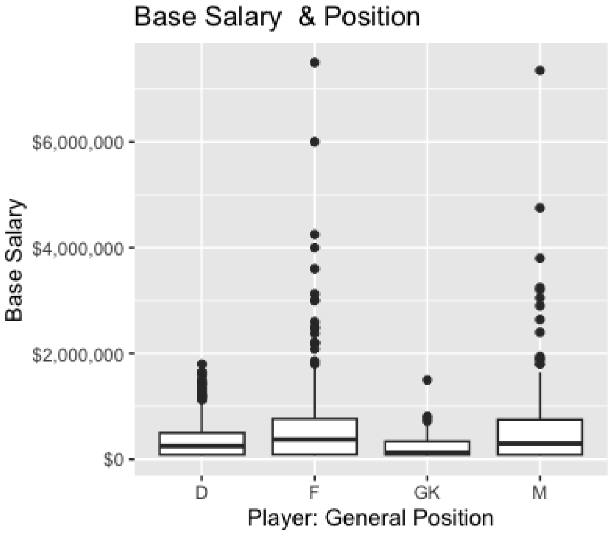
Our Histograms above show us that there is not a drastic change between the average **Base Salary ($473,475)** and the average ***Guaranteed Compensation($530,467)**. However, something to pay attention to in our histogram for the guaranteed compensation, is that a few variables appear on the far right side, meaning a few players reaching salaries of $7,000,000 and $8,000,000, something that doesn't appear in our histogram for the base salary. This is an important observation, especially for the club to be able to be on the right terms with the league, following the rules of salaries, and understanding how can they negotiate contracts with world-class players, such is the case of Lionel Messi, where is guaranteed compensation is way more than his base salary.



The plot above shows us a great visual representation of which teams spend the most, which teams spend the least, and the real application of the "salary cap" regulations. This image is great because we can quickly identify the clusters right on the line of the average salaries, the **designated players** are our outliers, located on the top part, and in the middle of the plot.

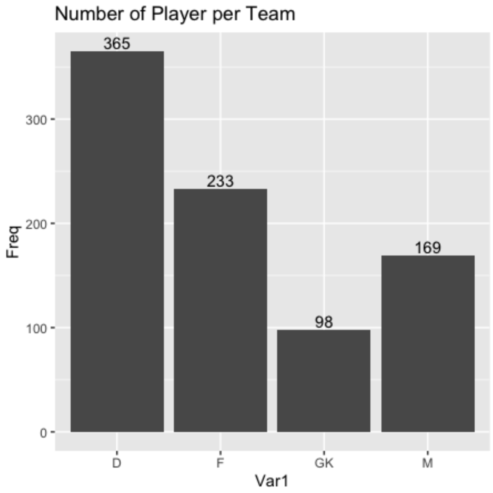
2.6 MLS 2022-2023 - Player by position

There are four major positions in soccer: Goalkeeper , Defender, Midfielder and Forward(Striker). As mentioned before MLS teams have rosters of 30 players on average, for official games they can only use 18, of which only 11 are starters. In the 2022-2023 season, there are around 865 professional soccer players. How can we know which position pays the most?, which position is saturated?, which one has more openings? we are about the find it out.



With our boxplot from above, we can quickly identify the top positions when it comes to salary, **forward** and **midfielder**, followed by defender and in the last spot goalkeeper. How can we define this by looking at the boxplot? the line we see in the middle of each box, tells us the median value of the category. We also obtain our outliers which are the dots on the upper side. In soccer you win the game by scoring more goals than your opponent, it makes sense that players who are closer to the goal and in more offensive areas of the field earn more. Another important fact is that most of the designated players are either forwards or midfielders, making them outliers. Each position is fundamental to having a competitive and strong team, you can't just have a team full of strikers and offensive players, you need a good balance and solid amendments.

We will now explore the distribution of players by positions



With our bar graph, we can identify the distribution of the soccer player in the MLS by position.

Goalkeepers (98), in this position, typically have a senior goalkeeper and two backup goalkeepers, who are commonly homegrown players, teams don't usually spend as much money on this position. Top goalkeepers can make close to $2,000,000 according to our data.

Midfielders (169), a very competitive position, with a high salary but not too many opportunities, this position is one of the most popular because you can be an offensive or a defensive player, and teams tend to have 6-8 midfielders on the squad, of which 3 - 4 will be starters, the team does tend to spend good money off those specifically designated midfielders that can give a competitive advantage. Top midfielders can make up to $7,000,000 according to our data.

Forwards (233), the highest paid position and the most competitive one, The team tends to have 4-6 forwards, of which 2 - 3 will start, the team does spend lots of money trying to have deadly wingers and awesome strikers, so when pursuing a career in this position, the player will be fighting for a spot against international and designated players. Top forwards can make up to $8,000,000 according to our data

Defenders (365), the position with the highest number of players in the MLS, which can be translated to higher opportunities, teams tend to have 8-10 defenders of which 4 - 5 are starters, the ones with better odds. A top defender can make up to $2,000,000 according to our data.

2.7 Conclusion

The MLS has a very unique format and set of rules, The unique structure of MLS sets up an interesting dilemma for the league and teams; individual teams are likely focused on their own revenues and the relationship between winning and attendance, while the league is likely to be highly focused on revenues(Coates, Frick, and Jewell 2016).

However we have to admit that MLS is in one of the best stages at the moment, our data set didn't include the newest world-class player, **Lionel Messi**, who has created a revolution, just like David Beckham when he arrived in 2007. With Messi, the MLS has landed deals with Adidas, and Apple, and increased tremendously the fanbase and outreach to people from the United States and around the world. Things just look bright for the United States and Soccer, as the World Cup, the largest and biggest sports tournament in the world, will be held in the United States in the year 2026, with very competitive soccer national teams for both male and female disciplines, it's looking like the right time to guide and support friends, relative, our ourselves in the chase for the dream of becoming professional soccer player in the United States.

3 Project: World Happiness Score

3.1 Introduction

Several psychologists agree that the main goal of rational human beings is to find happiness irrespective of their geographical locations, the time they live, or other demographic variables. Happiness is defined differently in each community, hence, there is no common ground to describe happiness (McMahon 2008). However, some entities go deep in trying to answer and assess the happiness of the world, by taking worldwide surveys, and utilizing distinct factors, to measure happiness.

The main purpose of this project is to show which variables have a higher impact on *"Happiness Score"*(dependent variable) according to data obtained from the World Happiness Report. We will also determine which variables like GDP per capita, social support, healthy life expectancy, freedom to make life choices, and generosity (independent variables) have more influence on the happiness score, with the help of a linear regression model.

Lastly, we will be able to see the differences between the years 2021 and 2023. As one of the main motivations for this project is the work I started in 2022 during my spring semester, I realized an applied regression analysis of the World Happiness Report of the year 2021. At that time we were still in the middle of the world pandemic of Coronavirus COVID-19. My goal at that time was to capture how the world would measure happiness during those dark times, and accurately respond if social features or economic features would have the highest impact on the happiness score of each country. Now, past 2 years of my initial analysis, I wanted to retake the same data type and compare the results from the year 2021 vs the ones for 2023.

I aim to provide a distinct perspective to our readers, as often, we define happiness with material things, and we do not measure it with other variables that are important in our lives, like peace, freedom of choice, generosity, etc.

3.2 The Data

3.2.1 Data Source

The first World Happiness Report was released on April 1, 2012. The first report outlined the state of world happiness, causes of happiness and misery, and policy implications highlighted by case studies. In 2013, the second World Happiness Report was issued, and in 2015 the third. Since 2016, it has been issued on an annual basis on the 20th of March, to coincide with the UN's International Day of Happiness.

We will use the World Happiness Report from the year 2023, a publication of the Sustainable Development Solutions Network, powered by the Gallup World Poll data (Authors 2023).

The rankings of national happiness are based on a happiness measurement survey undertaken worldwide by the polling company Gallup, Inc. National representative samples of respondents are asked to think of a ladder, with the best possible life for them being a 10, and the worst possible life being a 0. They are then asked to rate their own current lives on a 0 to 10 scale. The report correlates the life evaluation results with various life factors.

This is a poll that continually surveys citizens in 160 countries, representing more than 98% of the world's adult population. The poll has over 100 global questions and region-specific items. Gallup also works with organizations, cities, governments, and countries to create custom items and indexes to gather information to provide a score for the "happiness" of each country.

The World Happiness Report 2023 contains 137 observations and 9 variables. 8 numerical variables:

- **Rank.**
- **Ladder score (Happiness score):** National representative samples of respondents are asked to think of a ladder, with the best possible life for them being a 10, and the worst possible life being a 0. They are then asked to rate their own current lives on a 0 to 10 scale.
- **Logged GDP per capita:** Gross Domestic Product, or how much each country produces, divided by the number of people in the country. GDP per capita gives information about the size of the economy and how the economy is performing.
- **Social Support:**: Social support, or having someone to count on in times of trouble. “If you were in trouble, do you have relatives or friends you can count on to help you whenever you need them, or not?”
- **Healthy life expectancy:** More than life expectancy, how is your physical and mental health? Mental health is a key component of subjective well-being and is also a risk factor for future physical health and longevity. Mental health influences and drives a number of individual choices, behaviors, and outcomes.
- **Freedom to make life choices:** “Are you satisfied or dissatisfied with your freedom to choose what you do with your life?” This also includes Human Rights. Inherent to all human beings, regardless of race, sex, nationality, ethnicity, language, religion, or any other status. Human rights include the right to life and liberty, freedom from slavery and torture, freedom of opinion and expression, the right to work and education, and many more. Everyone is entitled to these rights without discrimination.
- **Generosity:** “Have you donated money to a charity in the past month?” A clear marker for a sense of positive community engagement and a central way that humans connect with each other. Research shows that in all cultures, starting in early childhood, people are drawn to behaviors that benefit other people.
- **Perceptions of corruption:** “Is corruption widespread throughout the government or not” and “Is corruption widespread within businesses or not?” Do people trust their governments and have trust in the benevolence of others?

1 nominal variable:

- **Country.**

```
tibble [137 x 9] (S3: tbl_df/tbl/data.frame)
 $ RANK                        : num [1:137] 1 2 3 4 5 6 7 8 9 10 ...
 $ Country name                : chr [1:137] "Finland" "Denmark" "Iceland" "Israel" ...
 $ Ladder score                : num [1:137] 7.8 7.59 7.53 7.47 7.4 ...
 $ Logged GDP per capita       : num [1:137] 10.8 11 10.9 10.6 10.9 ...
 $ Social support              : num [1:137] 0.969 0.954 0.983 0.943 0.93 ...
 $ Healthy life expectancy     : num [1:137] 71.1 71.3 72.1 72.7 71.6 ...
 $ Freedom to make life choices: num [1:137] 0.961 0.934 0.936 0.809 0.887 ...
 $ Generosity                  : num [1:137] -0.0188 0.1342 0.211 -0.0231 0.2127 ...
 $ Perceptions of corruption   : num [1:137] 0.182 0.196 0.668 0.708 0.379 ...
```

3.2.2 Data Exploration

Summary

Before moving into our regression analysis, we will start with data exploration, as it is very important to obtain a good sense of the data, variables’ behavior, and their distributions, one quick way of doing it, is to obtain a “summary” of our numeric variables

```
  Ladder score   Logged GDP per capita Social support   Healthy life expectancy
 Min.   :1.859   Min.   : 5.527        Min.   :0.3413   Min.   :51.53  
 1st Qu.:4.724   1st Qu.: 8.591        1st Qu.:0.7220   1st Qu.:60.65  
 Median :5.684   Median : 9.567        Median :0.8271   Median :65.84  
 Mean   :5.540   Mean   : 9.450        Mean   :0.7990   Mean   :64.97  
 3rd Qu.:6.334   3rd Qu.:10.540        3rd Qu.:0.8960   3rd Qu.:69.41  
 Max.   :7.804   Max.   :11.660        Max.   :0.9825   Max.   :77.28  
                                                        NA's   :1      
 Freedom to make life choices   Generosity        Perceptions of corruption
 Min.   :0.3816               Min.   :-0.254276   Min.   :0.1461           
 1st Qu.:0.7239               1st Qu.:-0.073543   1st Qu.:0.6678           
 Median :0.8005               Median : 0.001419   Median :0.7736           
 Mean   :0.7874               Mean   : 0.022444   Mean   :0.7254           
 3rd Qu.:0.8745               3rd Qu.: 0.117025   3rd Qu.:0.8459           
 Max.   :0.9614               Max.   : 0.531386   Max.   :0.9291           
```

With the summary, we can quickly glance across the numerical variables in our data, it allows us to "pre-visualize" the distribution of each variable, at least in a numeric way, with the important numbers, such as the ones located on the: "Min" (as minimum value), "Max"(as maximum value), and "Mean"(μ of the variable)

Missing Values

We will ensure our data has no missing values

```r
##Summary of missing values
#Create an empty summary table
num_of_Records <- nrow(hapiness2023)
num_of_Features <- ncol(hapiness2023)
data.summary <- as.data.frame(matrix(nrow=num_of_Features,ncol=3))

#Add table headings
colnames(data.summary) <- c("Features", "Missing Values","% of Missing Values")
#Add feature names in the first column
data.summary[,1] <- colnames(hapiness2023)


#A. Compute missing values per feature
for (i in 1:num_of_Features){
  data.summary[i,2] <- sum(is.na(hapiness2023[,i]))
  data.summary[i,3] <- (sum(is.na(hapiness2023[,i]))/num_of_Records)*100
}
```

We have determined there's only one missing value in our data set, making it less than 1% of our data set, which is great for further processes in our analysis. We know our missing value is on the **"Healthy life expectancy"** variable, we will now locate the country of the missing value to determine the action we will take to solve this singular missing value.

```r
hapiness2023$`Country name`[is.na(hapiness2023$`Healthy life expectancy`)]
```

```
[1] "State of Palestine"
```

The state of Palestine is a country that has no "Healthy life expectancy" value, for this case, we will use the "Healthy life expectancy" score of Palestine in 2021, which is of 62.25

Table 1: Missing Values Summary

Features	Missing Values	% of Missing Values
RANK	0	0.000000
Country name	0	0.000000
Ladder score	0	0.000000
Logged GDP per capita	0	0.000000
Social support	0	0.000000
Healthy life expectancy	1	0.729927
Freedom to make life choices	0	0.000000
Generosity	0	0.000000
Perceptions of corruption	0	0.000000

```r
hapiness2023$`Healthy life expectancy`[is.na(
  hapiness2023$`Healthy life expectancy`)] <- 62.25
hapiness2023[hapiness2023$`Country name` == "State of Palestine", c(
  "Country name", "Healthy life expectancy")]
```

```
# A tibble: 1 x 2
  `Country name`     `Healthy life expectancy`
  <chr>                                  <dbl>
1 State of Palestine                      62.2
```

Data Transformations

Since we are talking about the world's happiness score, we would like to present the happiest continents in the world, however, we do not have any region indicator variable in our data set. We decided to classify each country by its continent, with the help of the package **"country code"**, which can be converted from several different country coding schemes or into standardized country names in several languages. It can create variables with the name of the continent and/or several regional groupings to which each country belongs. (Arel-Bundock, Enevoldsen, and Yetman 2018)

```r
#add continents column
hapiness2023$`Regional indicator` <- countrycode(
  sourcevar = hapiness2023$`Country name`,
  origin = "country.name",
  destination = "continent")
table(hapiness2023$`Regional indicator`)
```

```

  Africa Americas     Asia   Europe  Oceania 
      37       21       38       38        2 
```

```
# A tibble: 5 x 3
   RANK `Country name` `Regional indicator`
  <dbl> <chr>          <chr>               
1     1 Finland        Europe              
2     2 Denmark        Europe              
3     3 Iceland        Europe              
4     4 Israel         Asia                
5     5 Netherlands    Europe              
```

```
# A tibble: 5 x 3
   RANK ‘Country name‘   ‘Regional indicator‘
  <dbl> <chr>            <chr>
1   133 Congo (Kinshasa) Africa
2   134 Zimbabwe         Africa
3   135 Sierra Leone     Africa
4   136 Lebanon          Asia
5   137 Afghanistan      Asia
```

With our “Regional indicator” variable is easier to detect the continents on each country. It also allows us to understand the data and see the distribution of each country by continent. as we could see in our previous tables, the top 5 countries were in their majority in Europe.

Now we will add one more variable by factorizing our new variable “Regional indicator” to give a numerical value to each continent, in this way our “Regional indicator” variable will be available to use in our regression analysis, and we can test if the continent of the country has an impact on their happiness score.

```r
#factor 'Regional Indicator'
hapiness2023$`Regional indicator` <- as.factor(hapiness2023$`Regional indicator`)
#Add new 'numeric' column for regional indicator
hapiness2023$Regionnum <- as.numeric(hapiness2023$`Regional indicator`)
hapiness2023[hapiness2023$`Country name` == "United States",
             c("RANK","Country name","Regional indicator","Regionnum")]
```

```
# A tibble: 1 x 4
   RANK ‘Country name‘ ‘Regional indicator‘ Regionnum
  <dbl> <chr>          <fct>                    <dbl>
1    15 United States  Americas                     2
```

3.2.3 Data Visualization

Scatterplot Matrix

With “ggpairs” a function from the “GGally” package(Schloerke, Crowley, and Cook 2018) allows us to build a great scatter plot matrix. It creates scatter plots of each pair visualized on the left side of the plot and **Pearson correlation value**

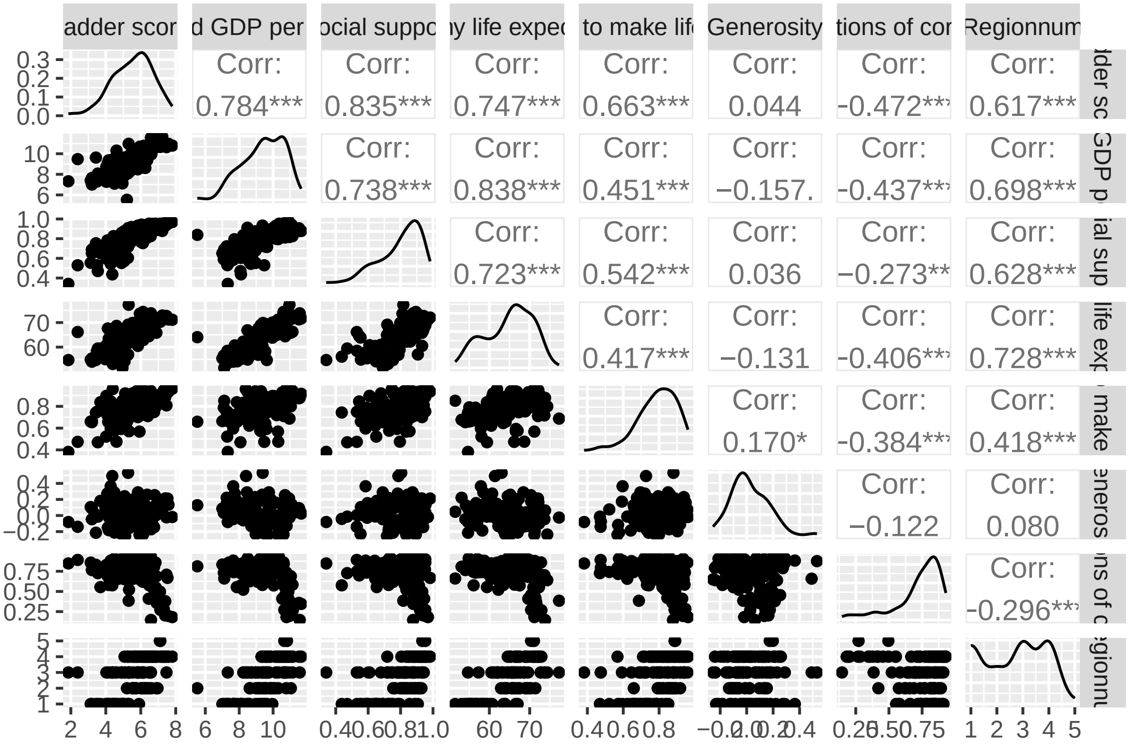
$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

and its significance is displayed on the right side. Better than the usual matrix that we obtain with the “plot” function, as it provides score and visualization of the distribution of data in each of the variables. We can see that the variables that have the highest correlation with the “Ladder score” are:

- **Logged GDP per capita**
- **Social support**
- **Healthy life expectancy**
- **Freedom to make life choices**
- **Regionnum**

When these variables increase, so does the ladder score. We can still see the correlation between the variable “Generosity” is positive but not nearly as high as the variables mentioned before. Lastly the variable “Perceptions of corruption” shows a negative correlation, meaning, as this variable decreases the ladder score increases, this variable has a good negative correlation number.

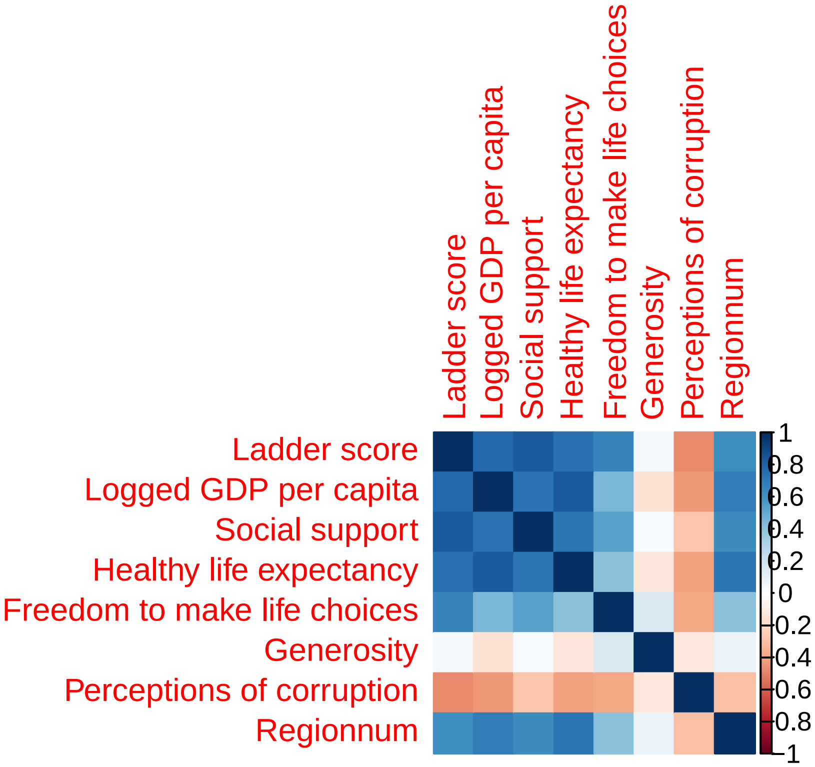
```
ggpairs(hapiness2023[4:11])
```



Heat Map

Our heat map with the "corrplot" function, from the package "corrplot"(Wei and Simko 2021), provides a visual exploratory tool on correlation matrix that supports automatic variable reordering to help detect hidden patterns among variables. Our heat map supports our previous observations, as positive correlations are displayed in blue and negative correlations in red. The color intensity and the size of the are proportional to the correlation coefficients.

```r
#Heat Map
#Positive correlations are displayed in blue and negative correlations in red.
#Color intensity are proportional to the correlation coefficients.
corrplot(cor(hapiness2023[4:11]), method ="color")
```



Thanks to our two visualizations, we were able to check the distribution of each of the variables, most of them present a normal distribution. Thanks to the representation and the Pearson correlation value, we know which variables we should focus on, and on which ones should we base our different tests on the regression analysis.

3.3 Regression analysis

Regression analysis is a reliable method of identifying which variables have an impact on a topic of interest, in our case our “Ladder.score”. The process of performing a regression allows you to confidently determine which factors matter most, which factors can be ignored, and how these factors influence each other.

Our goal with our regression analysis will be to pick the best-fitted model to predict the Ladder.score. We will perform different models, using simple linear regression, which is a model that describes the relationship between one dependent and one independent variable. We will also test multiple linear regression, which is a statistical technique that uses several explanatory variables to predict the outcome of a response variable.

3.3.1 Simple Linear Regression

The first model we will explore will be simple linear regression, which is a mathematical model that describes the relationship between two variables, the independent variable (predictor), and the dependent variable (outcome).

$$Y = a + bX + \varepsilon$$

where:

- Y represents the dependent variable (the outcome you want to predict).
- X represents the independent variable (the predictor).
- a represents the intercept, which is the point where the regression line crosses the Y-axis when X=0.
- b represents the slope of the regression line, which measures how the dependent variable changes for a one-unit change in the independent variable.
- ε represents the error term, which accounts for the random variability or noise in the relationship.

Model 1

For model 1 we will use as **independent variable** the “Social support” (highest Pearson correlation score), and our **dependent variable** the “ladder.score”.

```
Call:
lm(formula = ‘Ladder score‘ ~ ‘Social support‘, data = hapiness2023)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.76765 -0.37092  0.00851  0.46636  1.49965 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -0.3447     0.3386  -1.018     0.31    
‘Social support‘   7.3644     0.4183  17.605   <2e-16 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 0.6302 on 135 degrees of freedom
Multiple R-squared:  0.6966,    Adjusted R-squared:  0.6943 
F-statistic:   310 on 1 and 135 DF,  p-value: < 2.2e-16
```

With the summary of our linear regression model, we could prove that the variable “Social support” is statistically significant to our dependent variable. With a p-value less than 0.05, and a high R-squared score that explains the variance of 69.66%, we can see the high correlation between these variables.

We will explore two models more and we will compare our summary results to pick the best model.

3.3.2 Multiple Linear Regression

The second model we will explore will be the multiple linear regression, which extends the simple linear regression model to handle multiple independent variables instead of just one. It models the relationship between a dependent variable and two or more independent variables.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots + \beta_n X_n + \varepsilon$$

where:

- Y represents the dependent variable.
- X_1, X_2, X_3, …, X_n represent the independent variables or predictors.
- β_0 represents the intercept.
- $beta_1, beta_2, beta_3$, …, $beta_n$ represent the regression coefficients for each independent variable.
- ε represents the error term.

Model 2

For model 2 we will use as our independent variables all the numeric variables, and our dependent variable will stay the same, “ladder score”.

```
Call:
lm(formula = ‘Ladder score‘ ~ ., data = hapiness2023[4:11])

Residuals:
     Min       1Q   Median       3Q      Max 
-1.54074 -0.22678  0.04347  0.32017  1.06291 

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)    
(Intercept)                    -2.53813    0.83354  -3.045  0.00282 ** 
‘Logged GDP per capita‘         0.22329    0.07289   3.063  0.00267 ** 
‘Social support‘                3.87961    0.55056   7.047 9.85e-11 ***
‘Healthy life expectancy‘       0.02557    0.01499   1.706  0.09045 .  
‘Freedom to make life choices‘  2.39760    0.46693   5.135 1.02e-06 ***
Generosity                      0.25295    0.33105   0.764  0.44621    
‘Perceptions of corruption‘    -0.76062    0.28221  -2.695  0.00797 ** 
Regionnum                      -0.05130    0.05473  -0.937  0.35027    
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 0.484 on 129 degrees of freedom
Multiple R-squared:  0.829, Adjusted R-squared:  0.8197 
F-statistic: 89.34 on 7 and 129 DF,  p-value: < 2.2e-16
```

Model 2 summary allows us to identify that the are variables that are more statistically significant than others, identified by “*“, and that have more influence in our modeling. We can also see a low p-value and that our R-squared value increased, it now explains the variance of 82.9%.

We will conduct another model using only the variables that are more statistically significant to our dependent variable.

Model 3

For model 3 we will use as our independent variables: Logged GDP per capita, Social support, Freedom to make life choices, and Perceptions of corruption, and our dependent variable will stay the same, “ladder score”.

```
Call:
lm(formula = ‘Ladder score‘ ~ ‘Logged GDP per capita‘ + ‘Social support‘ +
    ‘Freedom to make life choices‘ + ‘Perceptions of corruption‘,
    data = hapiness2023)

Residuals:
     Min       1Q   Median       3Q      Max
-1.76318 -0.22459  0.07006  0.29787  1.03060

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                    -1.44500    0.54874  -2.633  0.00947 **
‘Logged GDP per capita‘         0.25830    0.05473   4.720 5.94e-06 ***
‘Social support‘                4.14287    0.51622   8.025 4.86e-13 ***
‘Freedom to make life choices‘  2.36695    0.45989   5.147 9.38e-07 ***
‘Perceptions of corruption‘    -0.86891    0.27263  -3.187  0.00179 **
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 0.4843 on 132 degrees of freedom
Multiple R-squared:  0.8248,    Adjusted R-squared:  0.8195
F-statistic: 155.3 on 4 and 132 DF,  p-value: < 2.2e-16
```

In Model 3 we can identify that 3 of the 4 variables are more statistically significant, identified by “*“, and they have more influence on our modeling. We can also see a low p-value and that our R-squared value stayed relatively the same as model 2, it now explains the variance of 82.48%.

We will conduct another model using only the variables that are more statistically significant to our dependent variable.

Model 4

For model 3 we will use as our independent variables: Logged GDP per capita, Social support, and Freedom to make life choices, and our dependent variable will stay the same, “ladder score”.

```
Call:
lm(formula = ‘Ladder score‘ ~ ‘Logged GDP per capita‘ + ‘Social support‘ +
    ‘Freedom to make life choices‘, data = hapiness2023)

Residuals:
     Min       1Q   Median       3Q      Max
-1.86044 -0.25456 -0.00379  0.35966  1.13015

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                    -2.74537    0.37936  -7.237 3.29e-11 ***
‘Logged GDP per capita‘         0.31994    0.05293   6.045 1.42e-08 ***
‘Social support‘                3.84962    0.52516   7.330 2.00e-11 ***
‘Freedom to make life choices‘  2.77580    0.45659   6.079 1.20e-08 ***
```

```
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 0.5007 on 133 degrees of freedom
Multiple R-squared:  0.8113,	Adjusted R-squared:  0.807 
F-statistic: 190.6 on 3 and 133 DF,  p-value: < 2.2e-16
```

After reviewing all 4 different models we will pick **model 4** as our decided regression model. All of our 3 multiple linear regression models had an increase in our R-squared value compared with the simple linear regression. Model 2,3,4 had similar scores, with the difference being that model 4 has fewer variables (all of them are statistically significant to the dependent variable), and its multiple R-squared and P-value relatively on the same level as model 2,3, which had more independent variables.

In Model 4 our regression formula looks like this:

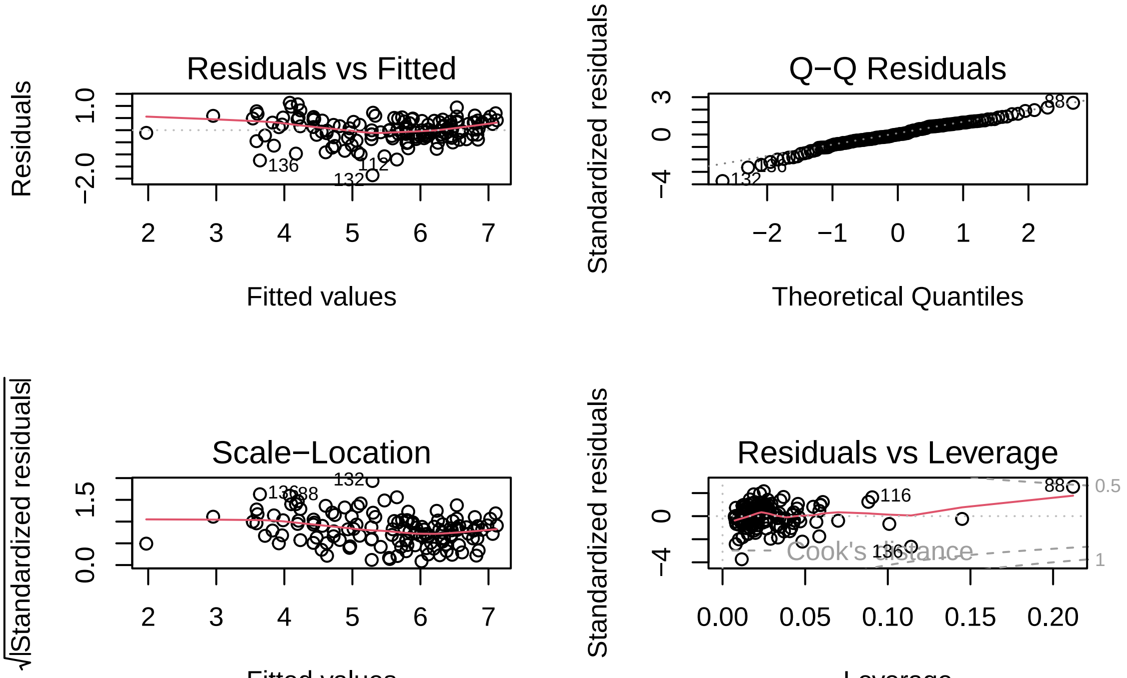
$Ladderscore = -2.74537 + 0.31994 * LoggedGDPpercapita + 3.84962 * Socialsupport + 2.77580 * Freedomtomakelifechoices$

- The Ladder score of a country is expected to be -2.74537, when the Logged GDP per capita = 0, Social support = 0, = Freedom to make life choices = 0
- For every one percentage increase in Logged GDP per capita, the score of a country is expected to increase by 0.31994
- For every one united increase in Social support, the score of a country is expected to increase by 3.84962
- For every one unit increase in Freedom to make life choices, the score of a country is expected to increase by 2.77580

We will move on to diagnostic plots to strengthen our decision on the model.

3.4 Model Diagnostic

Regression diagnostic plots are graphical tools used to assess the validity of linear regression models and they will help us identify potential issues or violations of the model assumptions. These plots help analysts understand the quality of the regression model for their data set.(Sheather 2009) We will see Residuals vs Fitted, Normal Q-Q, Scale Location, and Residuals vs Leverage plots



Residuals vs Fitted

This plot shows us if we have non-linear patterns. We can find equally spread residuals around a horizontal line without distinct patterns, this is a good indicator that we don't have non-linear relationships.

Normal Q-Q

This plot shows if residuals are normally distributed. We got a good residual distribution as the points follow the straight dashed line. just a few outliers on the lower left side, with the observation 132 could be worth exploring as a potential future issue. Overall the results are great.

Scale Location

This plot is very useful as shows if residuals are spread equally along the ranges of predictors, and we can check the assumption of equal variance, well known as **homoscedasticity**. We can see our model form a horizontal line and our residuals are randomly spread among it.

Residuals vs Leverage

This plot helps us to find influential cases. Not all outliers are influential in linear regression. Even though data have extreme values, they might not be influential in determining a regression line This shows there are few extreme values that would affect our regression, However, none of them pull or shift our regression line to any extreme

3.5 Predictions

Our multiple linear regression formula is:

$$Ladderscore = -2.74537 + 0.31994 * LoggedGDPpercapita + 3.84962 * Socialsupport + 2.77580 * Freedomtomakelifechoices$$

Now we will perform prediction with our model and some "dummy" data with values from our original data set, we will use our summary values like min, max, and mean.

- **Country 1**: Has the ***Min*** Logged GDP (5.527), has ***Max*** Social Support (0.9825), and ***Max*** Freedom to make life choices (0.9614).
- **Country 2**: Has the ***Mean*** Logged GDP (9.567), has ***Min*** Social Support (0.3413), and ***Min*** Freedom to make life choices (0.3816).
- **Country 3**: Has the ***Max*** Logged GDP (11.660), has ***Mean*** Social Support (0.7990), and ***Mean*** Freedom to make life choices (0.7874).

```
new_countries <- data.frame(
  "Logged GDP per capita" = c(5.527, 9.567, 11.660),
  "Social support" = c(0.9825, 0.3413, 0.7990),
  "Freedom to make life choices" = c(0.9614, 0.3816,0.7874)
)

# Replace dots with blank spaces in column names
colnames(new_countries) <- gsub("\\.", " ", colnames(new_countries))

#use the fitted model to predict the rating for the new country
predict(model_4_mlr, newdata=new_countries)
```

```
       1        2        3 
5.473844 2.688622 6.246646 
```

Before we analyse our results I want to recall the summary of all the Ladder scores:

```
summary(hapiness2023$`Ladder score`)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.859   4.724   5.684   5.540   6.334   7.804 
```

- **Country 1**: Has the ***Min*** Logged GDP (5.527), has ***Max*** Social Support (0.9825), and ***Max*** Freedom to make life choices (0.9614).

Obtained a Ladder Score of **5.4738**, a score that can be found right at the mean of the ladder scores for the year 2023. This "country" had the minimum value for logged GDP, the maximum value for Social Support, and the maximum value for Freedom to make life choices. This score shows us how important is the social aspect when it comes to measuring the happiness of a country.

- **Country 2**: Has the ***Mean*** Logged GDP (9.567), has ***Min*** Social Support (0.3413), and ***Min*** Freedom to make life choices (0.3816).

Obtained a Ladder Score of **2.6886**, a score that can be found very close to the minimum (1.859) value of the ladder scores for the year 2023. This "Country 2" had the mean value for Logged GDP, the minimum value for Social Support, and the minimum value for Freedom to make life choices. This scores, once again, show how important were the social features to measure the happiness score of countries in the year 2023.

- **Country 3**: Has the ***Max*** Logged GDP (11.660), has ***Mean*** Social Support (0.7990), and ***Mean*** Freedom to make life choices (0.7874).

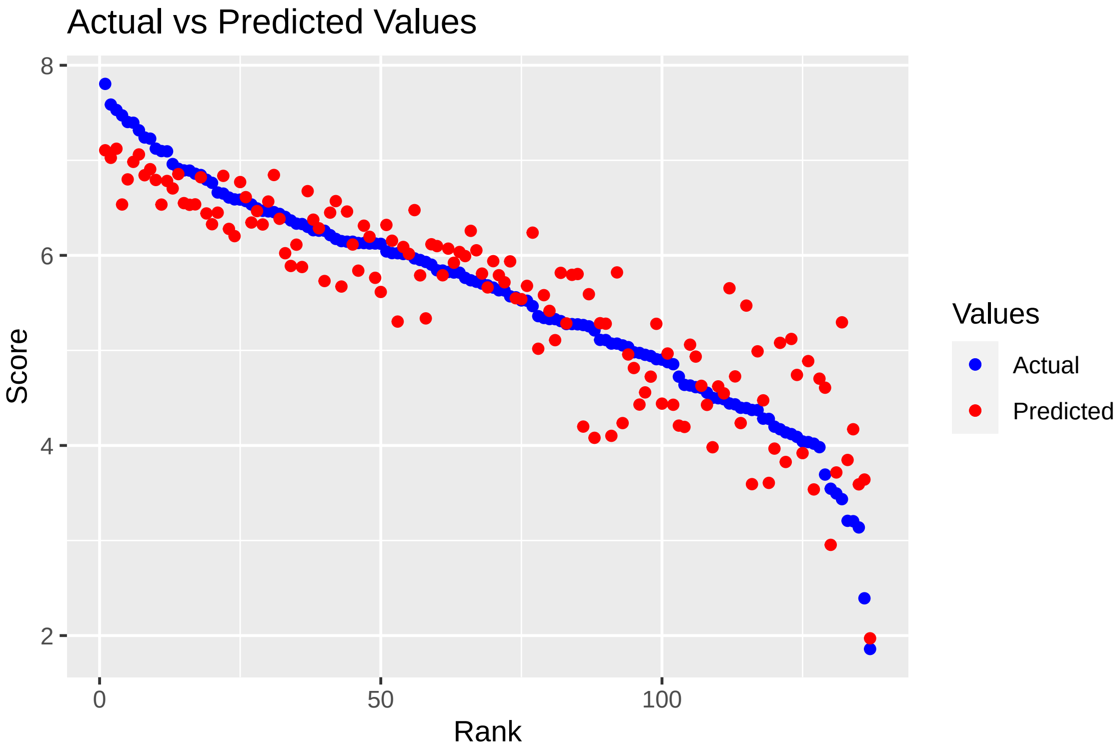
Obtained a Ladder Score of **6.2466**, a score that can be found in between the mean value (5.540) and the maximum value (7.804) of the Ladder scores for the year 2023. This "Country 3" had the maximum value for Logged GDP, the mean value for Social support, and the mean value for Freedom to make life choices. Of the three "countries" we tested with our best prediction model, "Country 3" obtained the best results. We can say these results were obtained thanks to a balance between the economic features (Logged GDP) and the social features (Social support, Freedom to make life choices).

These results show us that countries should not only focus on growing money, according to our data set, a country can have the lowest score when it comes to logged GDP, the maximum scores for Social support and Freedom to make life choices, and obtain a score that's very close to the mean of the World, when it comes to happiness. The ideal thing would be a great socio-economical mix, quite admirable is that the scores of 2023 showed that people valued more the social aspects, in comparison to results from 2021, where the values were more correlated with the economic side.

Predicted vs actual values

Lastly, we will see how our best model performs against the real Ladder score values from our World Happiness Report 2023. For that, we have created a scatterplot to see each of the Ladder Scores represented as dots in the same place, the color code is Blue for the actual Ladder score, and red for the predicted ladder scores.

The results are great, we can see both Ladder scores follow the same trend. The blue dots for our actual values form an almost perfect trend line and have very small variation, our predicted values are more scattered around, due to decimal points, however, all the results are not far from the actual Ladder score. This visualization allows us to understand the results from our previous summary reports. focusing more on a visual aspect to understand the performance of the prediction results in our model.



4 Project: Injury Diagnostic Classification

4.1 Introduction

Injuries to the lower leg, such as sprains and fractures, can be both painful and debilitating. Accurate and timely diagnosis of these injuries is crucial for effective treatment and recovery. This project was born from a personal experience. I suffered from an Achilles tendon rupture in late September 2023. I was playing a friendly pick-up soccer game with my friends, 10 minutes into the game, I made a motion to change pace and I felt like somebody kicked me under my calf. I heard a sound similar to a pop of a balloon and fell to the ground. I looked around me to see who hit me, there was no player nearby, and that's when I knew something bad happened. My injury happened around 10:00 pm, on a Monday, I visited a Doctor until Wednesday. As an international student, my first reaction was fear, and uncertainty, even though I played soccer since I was 5 years old, played for my country at the under-17 level, college level, and fourth division in the United States, I had never experienced an injury like this one. I was overthinking and checking my memories if I had something similar in my past years as a player, I used Monday night, and the whole day of Tuesday, to read, call my insurance, watch videos, and talk with my parents, all of this because I was uncertain of my diagnostic. Before my injury, I always thought Achilles tendon ruptures were rare and occasioned by impact collisions, or in athletes performing under high levels of pressure and intensity. When the reality is that of all the large tendon ruptures, 1 in 5 will be an Achilles tendon rupture. Achilles tendon rupture is estimated to occur in a little over 1 per 10,000 people per year. Males are also over 2 times more likely to develop an Achilles tendon rupture as opposed to women. Achilles tendon rupture tends to occur most frequently between the ages of 25-40 and over 60 years of age. Sports and high-impact activity are the most common cause of rupture in younger people, whereas sudden rupture from chronic tendon damage is more common in older people.(Wikipedia contributors, n.d.) This project aims to address the challenge of finding a diagnostic for these types of injuries, by leveraging the power of machine learning, specifically the Naive Bayes model , to classify leg injuries while practicing sports, into two distinct categories: "Sprain" and "Fracture."

4.2 The Data

The data set we will use comes from the NEISS(U.S. Consumer Product Safety Commission, n.d.), which is the National Electronic Injury Surveillance System, a statistically valid, injury surveillance system operated by the U.S. Consumer Product Safety Commission (CPSC).

NEISS injury data is gathered from the emergency departments (ED) of approximately 100 hospitals selected as a probability sample of all 5,000+ U.S. hospitals with emergency departments. The system's foundation rests on emergency department surveillance data, but the system also has the flexibility to gather additional data at either the surveillance or the investigation level.

4.2.1 Data Source

The data collection process begins when a patient is admitted to the emergency department of a NEISS hospital with an injury. An emergency department staff member elicits critical information about how the injury occurred and enters that information into the patient's medical record. The victim's age, gender, race, ethnicity, injury diagnosis, affected body parts, and incident locale are among other data variables coded. A brief narrative description of the incident is also included.

The NEISS has a NEISS Estimates Query Builder[Link], where we can specify dates, body parts, activeties/locations, age, sex, diagnostic, and the disposition, of the injuries. For our project, we built out a data set with the most recent 5 years (2018-2022), "Sports and Recreation Equipment" as activities/locations, "lower leg" for the body part, and "Sprain/Strain" and "Fracture" and our diagnosis.

4.2.2 Data Exploration

The first thing we will do is to display the structure of the data set, as it will provide a quick summary of our variables, dimensions, names, data types, etc.

```
tibble [2,813 x 25] (S3: tbl_df/tbl/data.frame)
 $ CPSC_Case_Number : num [1:2813] 1.8e+08 1.8e+08 1.8e+08 1.8e+08 1.8e+08 ...
 $ Treatment_Date   : POSIXct[1:2813], format: "2018-01-01" "2018-01-02" ...
 $ Age              : num [1:2813] 11 8 57 60 15 8 13 28 6 13 ...
 $ Sex              : num [1:2813] 2 2 2 2 1 2 1 2 1 1 ...
 $ Race             : num [1:2813] 0 1 0 0 1 1 3 1 1 1 ...
 $ Other_Race       : chr [1:2813] NA NA NA NA ...
 $ Hispanic         : logi [1:2813] NA NA NA NA NA NA ...
 $ Body_Part        : num [1:2813] 36 36 36 36 36 36 36 36 36 36 ...
 $ Diagnosis        : num [1:2813] 57 57 57 57 57 57 57 64 57 57 ...
 $ Other_Diagnosis  : logi [1:2813] NA NA NA NA NA NA ...
 $ Body_Part_2      : logi [1:2813] NA NA NA NA NA NA ...
 $ Diagnosis_2      : logi [1:2813] NA NA NA NA NA NA ...
 $ Other_Diagnosis_2: logi [1:2813] NA NA NA NA NA NA ...
 $ Disposition      : num [1:2813] 4 1 1 1 1 4 1 1 1 4 ...
 $ Location         : num [1:2813] 9 9 9 9 9 0 9 1 9 0 ...
 $ Fire_Involvement : num [1:2813] 0 0 0 0 0 0 0 0 0 0 ...
 $ Alcohol          : logi [1:2813] NA NA NA NA NA NA ...
 $ Drug             : logi [1:2813] NA NA NA NA NA NA ...
 $ Product_1        : num [1:2813] 3255 3297 3299 3283 1205 ...
 $ Product_2        : num [1:2813] 0 0 0 0 0 0 0 0 0 0 ...
 $ Product_3        : num [1:2813] 0 0 0 0 0 0 0 0 0 0 ...
 $ Narrative        : chr [1:2813] "11YOF FELL WHILE ICE SKATING AND FRACTURED LOW"..
 $ Stratum          : chr [1:2813] "S" "C" "V" "S" ...
 $ PSU              : num [1:2813] 73 31 21 7 68 31 31 85 5 5 ...
 $ Weight           : num [1:2813] 70.97 5.64 17.51 70.97 78.38 ...
```

This is the display of the structure of the data just for the year 2018, all our other data sets for year 2019-2022, have the same structure.

Our data has been previously filtered by the query builder, and has only results of injuries in the "lower leg area" (36), with "sprain/strain"(64) and "fracture"(57) as diagnostic. However, the data set has 25 variables, most of them containing multiple NA values.

For our diagnostic classifier, we will only need to use 3 variables:

- **Treatment_Date**
- **Diagnosis**
- **Narrative**

Using the structure-function str(), we see that the "lower leg" data frame includes 14,649 total injuries, three features: Treatment_Date (POSIXct date and time), Diagnosis (num), and Narrative (chr). The injury diagnostic has been coded as either **strain/sprain (64)** or **fracture(57)** in a numeric way, and the chr variable stores the full raw narrative text.

```
tibble [14,649 x 3] (S3: tbl_df/tbl/data.frame)
 $ Treatment_Date: POSIXct[1:14649], format: "2018-01-01" "2018-01-02" ...
 $ Diagnosis     : num [1:14649] 57 57 57 57 57 57 57 64 57 57 ...
 $ Narrative     : chr [1:14649] "11YOF FELL WHILE ICE SKATING AND FRACTURED LOWER"..
```

This is the display of the structure of the data for the years 2018-2022.

4.2.3 Missing Values

Now that we have the variables we want to work with, we need to ensure we don't have NA values

```
        Features Missing Values % of Missing Values
1 Treatment_Date              0                   0
2      Diagnosis              0                   0
3      Narrative              0                   0
```

4.3 Data Transformations

Our "Diagnosis" variable is currently a character type. Since we will be working with text and diagnostic classification, it will be better to convert it to a factor type. After converting it to factor we can see that 3369 (about 30% percent) of the "Narrative" in our data are classified as "Strain/Sprain"(64), while the the remainder were labeled "Fracture"(57). This proportion of the diagnosis is important as we continue with our analysis. We will ensure the distribution of Strain/Sprain and Fracture, is consistent at a 70% to 30% rate overall in training sets. Depending on the results obtained in the test phase, this is one of the things we can change and aim for a 50% to 50% distribution of the diagnosis, to seek for improvement in our model.

```r
lower_leg$Diagnosis<- factor(lower_leg$Diagnosis)
table(lower_leg$Diagnosis)
```

```

   57    64
11280  3369
```

Narrative Variable Clean-Up

The first step is to "Unicode" our Narrative variable, as we will be performing text mining and cleaning. we will use the iconv() function for character encoding conversion. It allows you to convert text from one character encoding to another, in this case, "UTF-8" is a common character encoding used to represent text data. This step will allow us to perform our cleaning functions with no problem.

Creating a Corpus

The next step is creating a corpus, which is a large and structured set of texts (collection of text documents), in this case, the phrases in our "Narrative" variable With the help of the function tm_map we will continue transforming our corpus. We will also use the "VCorpus" version instead of the "PCorpus", as the "VCorpus" creates an object that is volatile as it is stored in RAM, while the "PCorpus" creates an object that is permanent as it is stored in a disk(Lantz 2019). These are small but very efficient tips when it comes to text cleaning, especially if working with large data sets. Utilizing corpus as a comparison and modifying our narrative text with other packages like "stringr", makes a big difference in the speed, and storage used. We should always strive to process that optimizes the performance of the analysis overall and save some processing power for more complex tasks like model training and testing.

```r
lowerleg_corpus <- VCorpus(VectorSource(lower_leg$Narrative))
#inspect first two rows of the corpus
as.character(lowerleg_corpus[[2]])
```

```
[1] "8 YOF FELL WHILE ***.  DX FIBULA FX"
```

Tex transformation

With the help of the function tm_map we will continue transforming our corpus, We saw earlier that our phrases in “Narrative” are in capital letters, and contain numbers and symbols. We will now convert all the phrases to lower case we will eliminate any numbers present

```
lowerleg_corpus_clean <- tm_map(lowerleg_corpus, content_transformer(tolower))
lowerleg_corpus_clean <- tm_map(lowerleg_corpus_clean, removeNumbers)
as.character(lowerleg_corpus_clean[[2]])
```

```
[1] " yof fell while ***.  dx fibula fx"
```

In data mining, it’s common to remove what is called “filler” words, such as, “to”,“or”, and “but”. We will proceed to utilize the function “stopwords” which contains a list of common stopwords. Stop words are basically, the reason why we want to remove the stop words, because if we remove the words that are very commonly used in a given language, we can focus on the important words instead. and we will clean our corpus again. After, we will proceed to delete punctuation symbols as well, with the removePunctuation function.

```
lowerleg_corpus_clean <- tm_map(lowerleg_corpus_clean, removeWords, stopwords())
lowerleg_corpus_clean <- tm_map(lowerleg_corpus_clean, removePunctuation)
as.character(lowerleg_corpus_clean[[2]])
```

```
[1] " yof fell    dx fibula fx"
```

In this step, we will reduce the words to their root form in a process called stemming. Which is a natural language processing technique that lowers inflection in words to their root forms, hence aiding in the preprocessing of text, words, and documents for text normalization. Followed by the removal of the additional white space existing between our words. For these two steps, we used the functions stemDocument and stripWhitespace.

```
lowerleg_corpus_clean <- tm_map(lowerleg_corpus_clean, stemDocument)
lowerleg_corpus_clean <- tm_map(lowerleg_corpus_clean, stripWhitespace)
as.character(lowerleg_corpus_clean[[2]])
```

```
[1] "yof fell dx fibula fx"
```

Now we can check the difference between before and after the data cleaning by showing the first three rows of the raw data set versus the first three rows of the corpus clean set:

```
lapply(lowerleg_corpus[1:3], as.character)
```

```
$‘1‘
[1] "11YOF FELL WHILE ICE SKATING AND FRACTURED LOWER LEG"

$‘2‘
[1] "8 YOF FELL WHILE ***.  DX FIBULA FX"

$‘3‘
[1] "57YF WORKING OUT DOING AEROBICS WHEN ACC INVERTED ANKLE                >>FIB FX"
```

```
lapply(lowerleg_corpus_clean[1:3], as.character)
```

```
$‘1‘
[1] "yof fell ice skate fractur lower leg"

$‘2‘
[1] "yof fell dx fibula fx"

$‘3‘
[1] "yf work aerob acc invert ankl fib fx"
```

4.3.1 Text Tokenization

Tokenization, in machine learning, refers to the process of converting a sequence of text into smaller parts, known as tokens. These tokens can be as small as characters or as long as words. The primary reason this process matters is that it helps machines understand human language by breaking it down into bite-sized pieces, which are easier to analyze.

We will create a Document Term Matrix, where we will have the phrases in the Narrative variable, split into the words. The document term matrix will store a number that indicates how many times the word repeats.

```
lowerleg_dtm <- DocumentTermMatrix(lowerleg_corpus_clean)
```

4.3.2 Training and Test Datasets

For our diagnostic classifier to evaluate data it had not seen previously. We’ll divide the data set into two portions: 75% for training and 25% for testing.

```
#Create training and test datasets
#training 75% and test 25%
lowerleg_dtm_train <- lowerleg_dtm[1:10987, ]
lowerleg_dtm_test  <- lowerleg_dtm[10988:14649, ]
```

Let’s save the labels of the diagnostic in our training set, and our test set. This step will be very useful when we validate our model and see its accuracy.

```
#labels
lowerleg_train_labels <- lower_leg[1:10987, ]$Diagnosis
lowerleg_test_labels  <- lower_leg[10988:14649, ]$Diagnosis
```

We can check our distributions are similar in the training set and the test set.

```
#check that the proportion of spam is similar
prop.table(table(lowerleg_train_labels))
```

```
lowerleg_train_labels
       57        64
0.7721853 0.2278147
```

```
prop.table(table(lowerleg_test_labels))
```

```
lowerleg_test_labels
       57        64
0.7635172 0.2364828
```

4.3.3 Data Visualization

One way to visualize text data is to use a **word cloud** , which visualizes the frequency at which words appear in text data. Words appearing more often in the text are shown larger size, and less common terms are shown in smaller size.

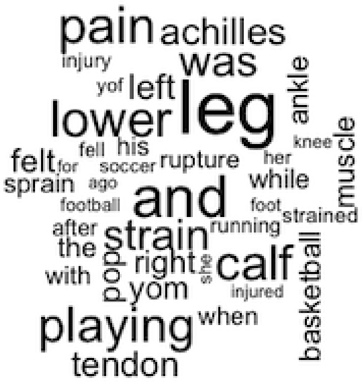


Figure 1: Strain/Sprain Word Cloud

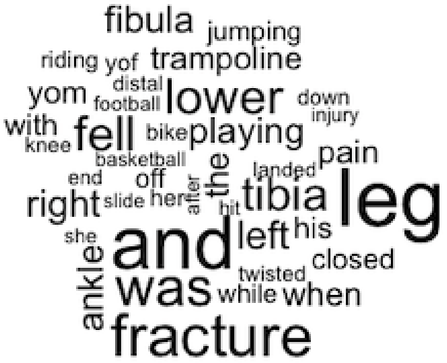


Figure 2: Fracture Word Cloud

The Sprain/Strain cloud is on the left. It include words such as **calf**, **pop**, **running**, **Achilles**, **tendon**; these terms do not appear in the Fracture cloud, instead we see words like **tibia**, **fibula**, and **landed**. These differences suggest that our naive Bayes model will have some key words to identify the difference between the classes.

4.4 Naive Bayes Text Clasification Analysis

4.4.1 Data Term Matrix transformation

The final step in the data preparation before we can start using our model is to transform the sparse matrix into a data structure that can be used to train a naive Bayes classifier. Our sparse matrix includes over 5,000 features where every word appears in at least one Narrative. To reduce the number of features, and make the process more efficient we will eliminate any words that appear less than 5 times in the Narrative or less than about 0.1 percent of records in the training data.

We will use the findFreqTerms() function. This function takes a document term matrix and returns a character vector containing the words appearing at least a specified number of times. In our case, it will display a character vector of the words appearing at least 5 times in the lowerleg_dtm_train matrix:

```
#Frequent words
lowerleg_freq_words <- findFreqTerms(lowerleg_dtm_train, 5)
```

With the frequent word identified, we reduced the number from over 5,000 words to a little bit more than 1,000 words. We will now proceed to create a Data Term Matrix, training, and testing, sets.

```
#Create Data Term Matrix only with frecuent words
lowerleg_dtm_freq_train <- lowerleg_dtm_train[ , lowerleg_freq_words]
lowerleg_dtm_freq_test <- lowerleg_dtm_test[ , lowerleg_freq_words]
```

Now, our frequent words training and testing Data Term Matrix are numeric. This now measures the number of times a word appears in the "Narrative". By giving the numerical values, we allowed our model to compute the calculation and the probabilities to decide whether that specific "Narrative" is diagnosed as a Sprain/Strain or a Fracture.

```
kbl(lowerleg_dtm_freq_test_table, caption = "Numerical Data Term Matrix Test",
    booktabs = T) %>%
kable_styling(latex_options = "hold_position")
```

Table 2: Numerical Data Term Matrix Test

	accid	accident	achil	achill	achilli	acl	across	activ	activitydx	acut
10993	0	0	0	1	0	0	0	0	0	0
10994	1	0	0	0	0	0	0	0	0	0
10995	0	0	0	0	0	0	0	0	0	0
10996	0	0	0	0	0	0	0	0	0	0
10997	0	0	0	1	0	0	0	0	0	0
10998	0	0	0	1	0	0	0	0	0	0
10999	0	0	0	0	0	0	0	0	0	0
11000	1	0	0	0	0	0	0	0	0	0
11001	0	0	0	0	0	0	0	0	0	0
11003	0	0	0	2	0	0	0	0	0	0

We need to change them to categorical variables since the naive Bayes classifier is typically trained on data with categorical features. We will change this to a factor variable that indicates yes or no depending on whether the word appears at all. The following code defines a convert_counts() function to convert counts to factors:

```r
#Create function to convert count to factors "Yes","No"
convert_counts <- function(x) {
  x <- ifelse(x > 0, "Yes", "No")
}

#apply() convert_counts() to columns of train/test data
lowerleg_train <- apply(lowerleg_dtm_freq_train, MARGIN = 2, convert_counts)
lowerleg_test  <- apply(lowerleg_dtm_freq_test,  MARGIN = 2, convert_counts)

kbl(as.data.frame(lowerleg_test[6:16,9:19]),caption = "Categorical Data Term Matrix Test" ,
    booktabs = T) %>%
kable_styling(latex_options = "hold_position")
```

Table 3: Categorical Data Term Matrix Test

	accid	accident	achil	achill	achilli	acl	across	activ	activitydx	acut	admit
10993	No	No	No	Yes	No	No	No	No	No	No	No
10994	Yes	No	No	No	No	No	No	No	No	No	No
10995	No	No	No	No	No	No	No	No	No	No	No
10996	No	No	No	No	No	No	No	No	No	No	No
10997	No	No	No	Yes	No	No	No	No	No	No	No
10998	No	No	No	Yes	No	No	No	No	No	No	No
10999	No	No	No	No	No	No	No	No	No	No	No
11000	Yes	No	No	No	No	No	No	No	No	No	No
11001	No	No	No	No	No	No	No	No	No	No	No
11002	No	No	No	No	No	No	No	No	No	No	No
11003	No	No	No	Yes	No	No	No	No	No	No	No

4.4.2 Train a model

With all the data transformation done, it is time to apply the naive Bayes, one of the most common probabilistic algorithms used for classification tasks, which is also a supervised non-linear classification algorithm.

Historically, this technique became popular with applications in email filtering, spam detection, document categorization, and weather prediction. This classification model has many advantages, it is simple and easy to implement and work well with small amount of data and with large amounts. The classification algorithm is also able to produce forecasting as it relies on principles of probabilities. One of the noticeable disadvantages is that it doesn't perform well with numerical data, an extra step of creating "bins" is necessary to obtain decent results, however, other models are superior to Naive Bayes when it comes to working with numerical features.

It's based on Bayes' theorem work of "Thomas Bayes" and is considered "naive" because it makes the assumption that all features are conditionally independent, given the class. in other words, it describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

The formula for this algorithm is:

$$P(C_k|x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n|C_k) \cdot P(C_k)}{P(x_1, x_2, \ldots, x_n)}$$

Where:

- $P(C_k|x_1, x_2, \ldots, x_n)$ is the posterior probability of class C_k given the observed features.
- $P(x_1, x_2, \ldots, x_n|C_k)$ is the likelihood of observing the features given class C_k
- $P(C_k)$ is the prior probability of class C_k
- $P(x_1, x_2, \ldots, x_n)$ is the marginal likelihood of the features

Our model will be able to classify the text in the Narrative variable and decide the diagnostic if it is a Sprain/Strain or a Fracture. We will use the package e1071.(Dimitriadou et al. 2009)

The lowerleg_classifier variable now contains a naive Bayes classifier an object that can be used to make predictions.

```
lowerleg_classifier <- naiveBayes(lowerleg_train,lowerleg_train_labels)
```

4.4.3 Evaluate model performance

With our lowerleg_classifier, we will now predict the diagnostic for the unseen test data. We will use the predict() function, and we will compare the results to their true values.

```
lowerleg_test_pred <- predict(lowerleg_classifier, lowerleg_test)
```

4.4.4 Cross table

To evaluate our model performance, we'll use the CrossTable() function in the gmodels package(Warnes et al. 2018). This time, we'll use our prediction results, versus the actual test labels. The cross table will help us visualize and identify which results were accurate and which ones were wrong.

```
results <- CrossTable(lowerleg_test_pred, lowerleg_test_labels,
           prop.chisq = FALSE, prop.t = FALSE, prop.r = FALSE,
           dnn = c('predicted', 'actual'))
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Col Total |
|-------------------------|


Total Observations in Table:  3662


             | actual
   predicted |        57 |        64 | Row Total |
-------------|-----------|-----------|-----------|
          57 |      2761 |        72 |      2833 |
             |     0.987 |     0.083 |           |
-------------|-----------|-----------|-----------|
          64 |        35 |       794 |       829 |
             |     0.013 |     0.917 |           |
-------------|-----------|-----------|-----------|
Column Total |      2796 |       866 |      3662 |
             |     0.764 |     0.236 |           |
-------------|-----------|-----------|-----------|
```

The parts of this cross table are

- **Predicted Labels**: The columns labeled "57"(Fracture) and "64"(Sprain/Strain) under "predicted" represents the diagnostics that the Naive Bayes model predicted.
- **Actual Labels**: The rows labeled "57(Fracture)" and "64(Sprain/Strain)" under "actual" represents the true labels.
- **Main Diagonal**(True Predictions): The numbers on the main diagonal (2761 for label 57 and 794 for label 64) represents the number of instances that were correctly classified by the model. For example, 2761 times the model correctly predicted the label 57(Fracture), and 794 times it correctly predicted label 64(Sprain/Strain).
- **Off-Diagonal**(Miss classifications): The numbers off the main diagonal (35 and 72) represent the instances that were incorrectly classified. For example, the model predicted label 57 instead of the true label 64 for 35 cases, and the model predicted label 64 instead of the true label 57(Fracture) for 72 cases.

- **Row Total**: This shows the total number of instances with the actual labels. There were 2833 cases with the true label 57(Fracture) and 829 instances with the true label 64(Sprain/Strain).
- **Column Total**: This shows the total number of predictions made by the model for each label. The model predicted label 57(Fracture) for 2796 instances and label 64(Sprain/Strain) for 866 instances.
- **Accuracy**: The diagonal cells also contain percentages (0.987 for label 57 and 0.917 for label 64), which represent the positive predictive value for each label. By dividing the number of true positives (main diagonal) by the total number of predicted labels (column total). This is how it's calculated:

$$
\begin{aligned}
Accuracy &= \frac{TruePositives57 + Truepositive64}{TotalInstances} * 100 &(1)\\
Accuracy &= \frac{2761 + 794}{3662} * 100 &(2)\\
Accuracy &= 97.08 &(3)\\
& &(4)
\end{aligned}
$$

- **Error Rate**: The off-diagonal percentages (0.083 for label 64 and 0.013 for label 57) represents the misclassification rate for each label, which is calculated by dividing the number of false predictions (off-diagonal) by the total number of actual labels (row total).
- **Proportion of Total**: The percentages at the bottom (0.764 for label 57 and 0.236 for label 64) represent the proportion of the total predictions made by the model for each label.

4.4.5 Improving Model Performance

We aim to improve our model performance, we will now use the Laplace estimator when training our model. This will allow words that appeared in zero Sprain/Strain or zero Fracture Narrative to have an indisputable say in the classification process. Just because the word "run" only appeared in the Fracture Narrative in the training data, it does not mean that every message with that word should be classified as Fracture.

```r
injuries_classifier2 <- naiveBayes(lowerleg_train, lowerleg_train_labels,
                                    laplace = 1)
injuries_test_pred2 <- predict(injuries_classifier2, lowerleg_test)
```

```r
CrossTable(injuries_test_pred2, lowerleg_test_labels,
             prop.chisq = FALSE, prop.t = FALSE, prop.r = FALSE,
             dnn = c('predicted', 'actual'))
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Col Total |
|-------------------------|


Total Observations in Table:  3662
```

```
             | actual 
   predicted |        57 |        64 | Row Total | 
-------------|-----------|-----------|-----------|
          57 |      2756 |        62 |      2818 | 
             |     0.986 |     0.072 |           | 
-------------|-----------|-----------|-----------|
          64 |        40 |       804 |       844 | 
             |     0.014 |     0.928 |           | 
-------------|-----------|-----------|-----------|
Column Total |      2796 |       866 |      3662 | 
             |     0.764 |     0.236 |           | 
-------------|-----------|-----------|-----------|
```

The parts of this cross table are

- **Predicted Labels**: The columns labeled “57”(Fracture) and “64”(Sprain/Strain) under “predicted” represents the diagnostics that the Naive Bayes model predicted.
- **Actual Labels**: The rows labeled “57(Fracture)” and “64(Sprain/Strain)” under “actual” represents the true labels.
- **Main Diagonal**(True Predictions): The numbers on the main diagonal (2756 for label 57 and 804 for label 64) represents the number of instances that were correctly classified by the model. For example, 2756 times the model correctly predicted the label 57(Fracture), and 804 times it correctly predicted label 64(Sprain/Strain).
- **Off-Diagonal**(Miss classifications): The numbers off the main diagonal (40 and 62) represent the instances that were incorrectly classified. For example, the model predicted label 57 instead of the true label 64 for 40 cases, and the model predicted label 64 instead of the true label 57(Fracture) for 62 cases.
- **Row Total**: This shows the total number of instances with the actual labels. There were 2818 cases with the true label 57(Fracture) and 844 instances with the true label 64(Sprain/Strain).
- **Column Total**: This shows the total number of predictions made by the model for each label. The model predicted label 57(Fracture) for 2796 instances and label 64(Sprain/Strain) for 866 instances.
- **Accuracy**: The diagonal cells also contain percentages (0.986 for label 57 and 0.928 for label 64), which represent the positive predictive value for each label. By dividing the number of true positives (main diagonal) by the total number of predicted labels (column total). This is how it’s calculated:

$$Accuracy = \frac{TruePositives57 + Truepositive64}{TotalInstances} * 100 \tag{5}$$

$$Accuracy = \frac{2756 + 804}{3662} * 100 \tag{6}$$

$$Accuracy = 97.21 \tag{7}$$

$$\tag{8}$$

- **Error Rate**: The off-diagonal percentages (0.072 for label 64 and 0.014 for label 57) represents the misclassification rate for each label, which is calculated by dividing the number of false predictions (off-diagonal) by the total number of actual labels (row total).

- **Proportion of Total**: The percentages at the bottom (0.489 for label 57 and 0.511 for label 64) represent the proportion of the total predictions made by the model for each label.

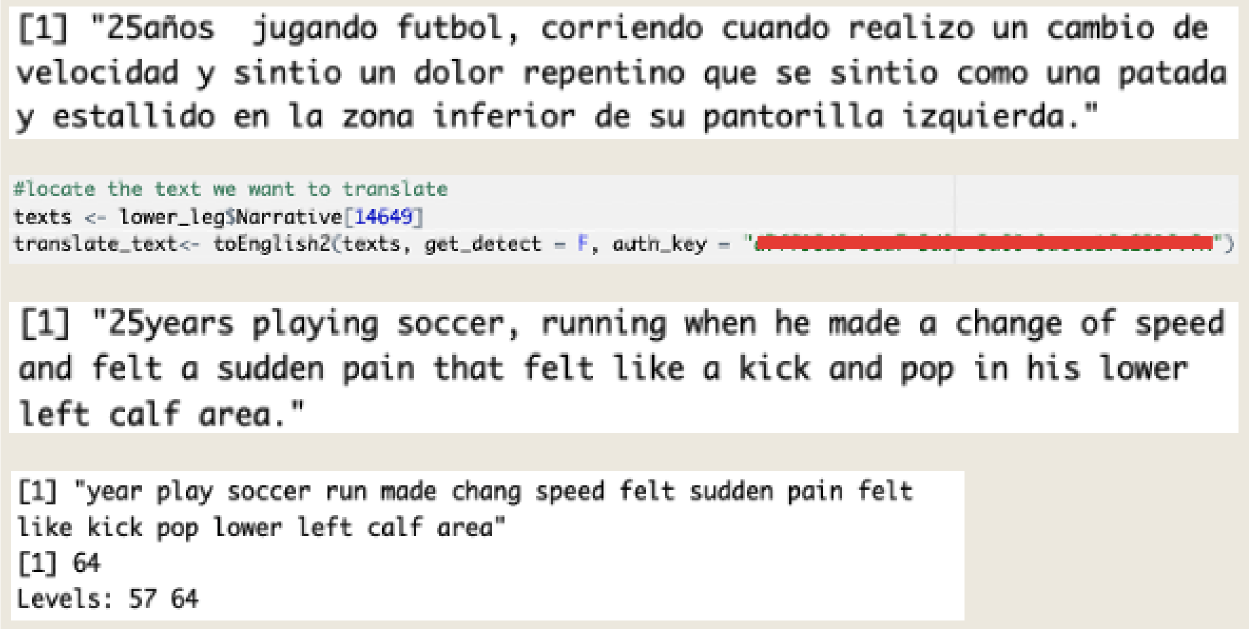
There was a increase of .13 in our model two using Laplace.

4.5 Language Translation

Recalling the motivation for this specific project, it was my personal story regarding my Achilles tendon rupture injury. I'm an International student from El Salvador, and my first language is Spanish. I have been in the United States for seven years, and I can say with confidence that I have advanced English, however in stressful times, new formal vocabulary, or very specific topics, I'll say I still rely on translating a few words from Spanish to English before I speak or while I try to understand the sentence and the context.

My injury was rather than painful, it was stressful, because of the insurance, the communication with several different clinics, and the communication with my parents back in my home country. All of these factors, plus it happened at the beginning of my last semester of graduate school and on the week I started my internship. The process of understanding what type of injury I had, regardless of the language, would have solved all these stressful situations.

That's why I decided to explore many different functions, and packages for R, that would help me translate text from Spanish to English. That's when I found the package *deeplr*(n.d.). A package that creates a link to the professional translation services of "DeepL Translator", which is a neural machine translation service that was launched in August 2017. The package allows a connection with, the web service for translating texts between different languages. A DeepL API developer account is required to use the service, using the API as a bridge and serving as the "log in" information to allow the specific requested texts to be translated to the desired language.



```
[1] "25años  jugando futbol, corriendo cuando realizo un cambio de
velocidad y sintio un dolor repentino que se sintio como una patada
y estallido en la zona inferior de su pantorilla izquierda."
```

```
#locate the text we want to translate
texts <- lower_leg$Narrative[14649]
translate_text<- toEnglish2(texts, get_detect = F, auth_key = "[redacted]")
```

```
[1] "25years playing soccer, running when he made a change of speed
and felt a sudden pain that felt like a kick and pop in his lower
left calf area."
```

```
[1] "year play soccer run made chang speed felt sudden pain felt
like kick pop lower left calf area"
[1] 64
Levels: 57 64
```

We can see how the function"toEnglish2" successfully located the row we wanted to translate, we can read the corpus clean version, and lastly, the classification our model gave to this specific event (my personal experience). Succesfully it classified it as a Sprain/Strain (64) rather than a fracture (57).

Bibliography

2023a. https://www.mlssoccer.com/news/2023-mls-roster-rules-and-regulations.

———. 2023c. https://mlsplayers.org/resources/salary-guide.

———. 2023b. https://en.wikipedia.org/wiki/Major_League_Soccer.

———. n.d. https://CRAN.R-project.org/package=deeplr .

Arel-Bundock, Vincent, Nils Enevoldsen, and CJ Yetman. 2018. "Countrycode: An r Package to Convert Country Names and Country Codes." *Journal of Open Source Software* 3 (28): 848. https://doi.org/10.21105/joss.00848.

Authors, Various. 2023. "World Happiness Report 2023." 2023. https://worldhappiness.report/ed/2023/.

Coates, Dennis, Bernd Frick, and Todd Jewell. 2016. "Superstar Salaries and Soccer Success: The Impact of Designated Players in Major League Soccer." *Journal of Sports Economics* 17 (7): 716–35.

Dimitriadou, Evgenia, Kurt Hornik, Friedrich Leisch, David Meyer, Andreas Weingessel, and Maintainer Friedrich Leisch. 2009. "Package 'E1071'." *R Software Package, Avaliable at Http://Cran. Rproject. Org/Web/Packages/E1071/Index. Html.*

Lantz, Brett. 2019. *Machine Learning with r: Expert Techniques for Predictive Modeling*. Packt publishing ltd.

McMahon, Darrin M. 2008. "The Pursuit of Happiness in History." *The Science of Subjective Well-Being*, 80–93.

Schloerke, Barret, Jason Crowley, and Di Cook. 2018. "Package 'GGally'." *Extension to 'Ggplot2.'See* 713.

Sheather, Simon. 2009. *A Modern Approach to Regression with r*. Springer Science & Business Media.

U.S. Consumer Product Safety Commission. n.d. "NEISS Injury Data." https://www.cpsc.gov/Research--Statistics/NEISS-Injury-Data.

Warnes, Gregory R, Ben Bolker, Thomas Lumley, Maintainer Gregory R Warnes, and MASS Imports. 2018. "Package 'Gmodels'." *Vienna: R Foundation for Statistical Computing.*

Wei, Taiyun, and Viliam Simko. 2021. *R Package 'Corrplot': Visualization of a Correlation Matrix*. https://github.com/taiyun/corrplot.

Wikipedia contributors. n.d. "Achilles Tendon Rupture - Wikipedia." https://en.wikipedia.org/wiki/Achilles_tendon_rupture#Differential_diagnosis.