

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Course

**Introduction To Data Science[B]**

Submitted To:

**AKINUL ISLAM JONY**

**Project Overview:**

The project is about the richest athletes or sportsmen in the world. From this project, we will be able to tell the rank of the sportsman who has the highest net worth. We have mainly worked with the age variable so that we can recognize which category aged people are mostly rich. Here, we first gain a dataset by web scrapping and then implement data preprocessing techniques into it. We then organize the data and describe the statistics. Data visualization is done at the end for a better understanding of the data.

To prepare a clean dataset, the following tasks of data pre-processing are to be performed using R language:

1. Data cleaning:

a. Smooth Noisy Data

b. Handling Missing Data

c. Data Wrangling or Munging

2. Data Integration

3. Data Transformation

4. Data Reduction

5. Data Discretization

The following steps were taken for data pre-processing:

**Project Solution Design:**

To collect data the following steps were taken:

1. Collect data from the website “therichest.com”
2. Store the data in a CSV file.

To prepare a clean dataset, the following steps were taken:

1. Data cleaning:
   1. Smooth Noisy Data
   2. Handling Missing Data
   3. Data Wrangling or Munging
2. Data Integration
3. Data Transformation

**The steps are described below:**

**Web Scrapping:**

1. For Web scrapping, we first install some libraries and for that, we have installed a package named “rvest”

Code:

* install.packages("rvest")
* library(rvest)
* library(stringr)
* library(tibble)

1. Then we create a loop to scrap multiple page

Top\_athlete<-data.frame()

i=1

while(i<3){

if(i<2){

url<-"https://www.therichest.com/top-lists/top-100-richest-athletes/"

}

else{

url<-"https://www.therichest.com/top-lists/top-100-richest-athletes/page/2/"

}

top\_athletes<-read\_html(url)

1. Extracting names

athletes\_name<-top\_athletes %>%

html\_nodes("tbody tr td.name")%>%

html\_text()

athlete\_name<-athletes\_name%>%

str\_trim()%>%

str\_split("\n")

lapply(athlete\_name, function(name){

name[1]

})%>%

unlist()%>%

str\_trim()

1. Extracting net worth

athletes\_networth<-top\_athletes %>%

html\_nodes("tbody tr td.networth")%>%

html\_text()

athlete\_networth<-athletes\_networth%>%

str\_trim()%>%

str\_split("\n")

lapply(athlete\_networth, function(networth){

networth[1]

})%>%

unlist()%>%

str\_trim()

1. Extracting Age

athletes\_age<-top\_athletes %>%

html\_nodes("tbody tr td.age")%>%

html\_text()

athlete\_age<-athletes\_age%>%

str\_trim()%>%

str\_split("\n")

lapply(athlete\_age, function(age){

age[1]

})%>%

unlist()%>%

str\_trim()%>%

as.numeric() #convert the character value to numeric

1. Extracting Country

athletes\_country<-top\_athletes %>%

html\_nodes("tbody tr td.country")%>%

html\_text()

athlete\_country<-athletes\_country%>%

str\_trim()%>%

str\_split("\n")

lapply(athlete\_country, function(country){

country[1]

})%>%

unlist()%>%

str\_trim()

1. Extracting Rank

Ranks<-top\_athletes %>%

html\_nodes("tbody tr td.rank")%>%

html\_text()

Rank<-Ranks%>%

str\_trim()%>%

str\_split("\n")

lapply(Rank, function(rank){

rank[1]

})%>%

unlist()%>%

str\_trim()%>%

as.numeric()

1. Organizing

top\_athelete\_nibble<-tibble(

Rank=Rank,

Name=athlete\_name,

Networth=athlete\_networth,

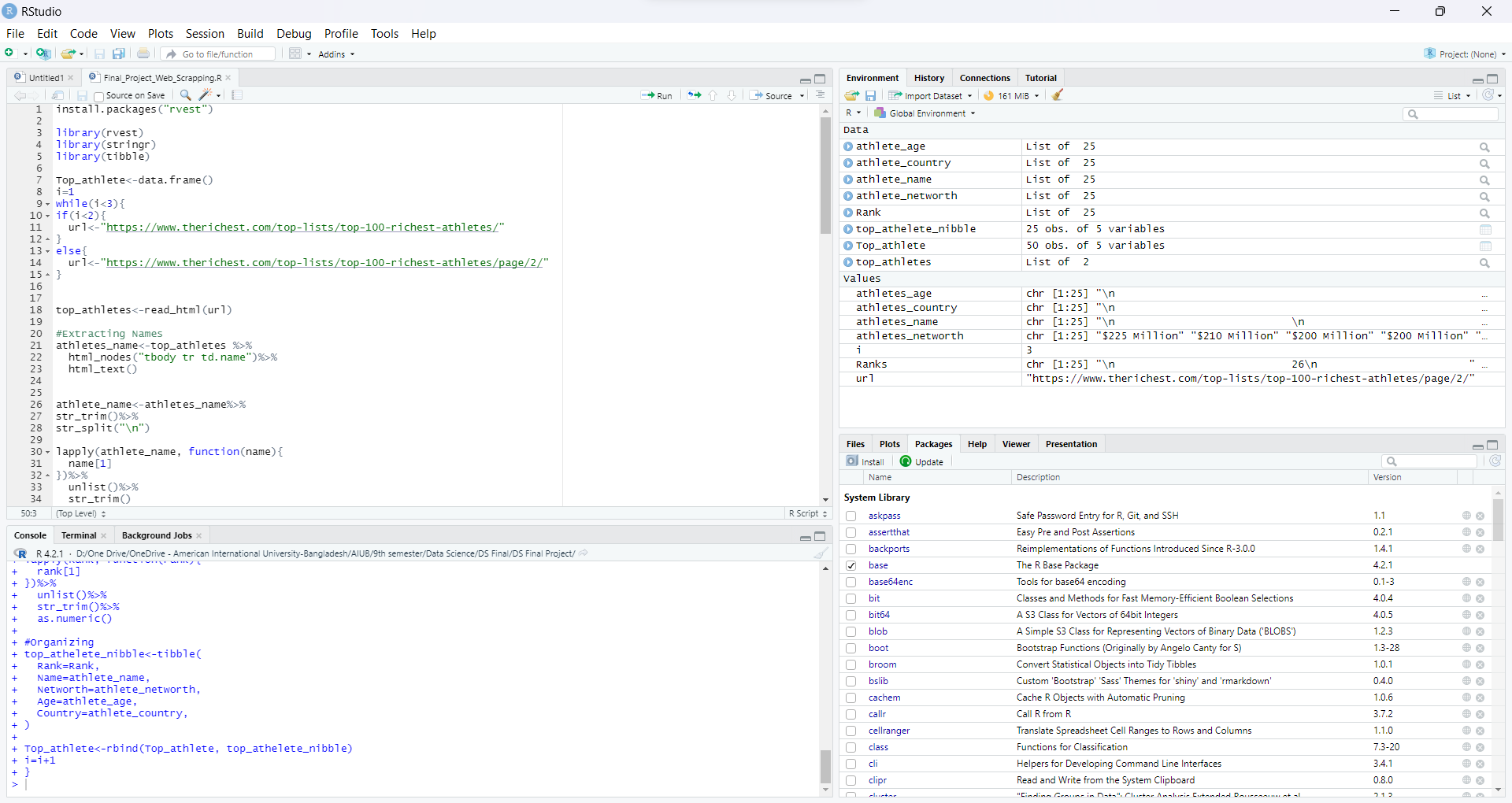
Age=athlete\_age,

Country=athlete\_country,

)

Top\_athlete<-rbind(Top\_athlete, top\_athelete\_nibble)

i=i+1

}

**Importing the Dataset**

Data pre-processing techniques were supposed to be applied in the provided dataset which was provided in a table. The data was then converted into a CSV file. The data was imported into RStudio for performing further operations.

Code for importing the dataset:

* dataset <- read.csv('Top\_athlete.csv') #reading data from csv file
* dataset

Table

Description automatically generated

Fig: Initial dataset

**Data pre-processing steps: -**

1. **Data Cleaning:**

* any(is.na(dataset))
* dataset <- dataset[dataset$Country != 'N/A', ]
* print(dataset)



It was found that no data was missing from the provided dataset.

1. **Data Transformation:**

In the data transformation part, data is transformed so that it is consistent and readable by the system. The numerical values which had unnecessary decimal numbers were rounded to integer format by using the round function.

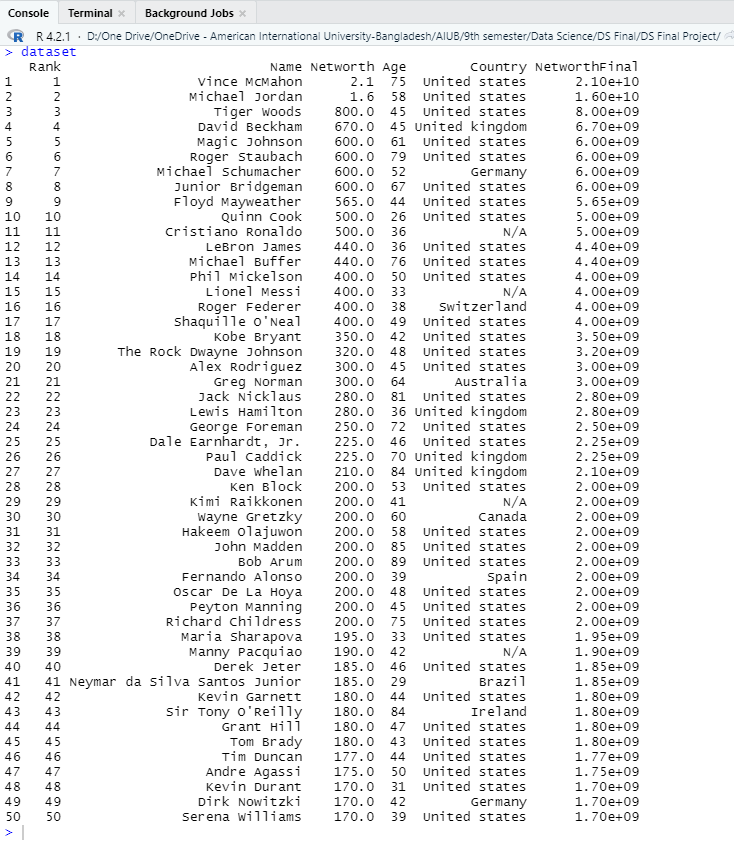
Code for formatting the numerical values:

dataset$Age<- round(dataset$Age)

We have also transformed the data of the Networth from character to numeric value.

Text

Description automatically generated

****

**Fig:** After conversion.

1. **Data Integration:**

Data integration is used to integrate multiple datasets or to add dimension to the original dataset. A column named ‘Age\_Type’ was integrated with the provided dataset. The column was based on the age variable. The data was converted into types having different ranges. Young (<50), Middle-Aged (>=50 & <60), and Older (60 and above).

The dataset was copied into a new variable so that the original dataset does not change. The new dataset was the same as the previous one only with an added column named Age-Type which had the same data as Age….

Code for integrating a new column:

dataset <- cbind(dataset, Age\_Type=NA) #adding a new empty column in the dataframe

for (i in 1:nrow(dataset)){ #assigning values to the column 'Type' according the given conditions

if(dataset$Age[i]<50){

dataset$Age\_Type[i] = 'Young'

}else if(dataset$Age[i]>=50 & dataset$Age[i]<60){

dataset$Age\_Type [i]= 'Middle-Aged'

}else{

dataset$Age\_Type[i] = 'Older'

}

}

df <- dataset

df

Table

Description automatically generated Fig: Dataset after integrating the Age-Type column.

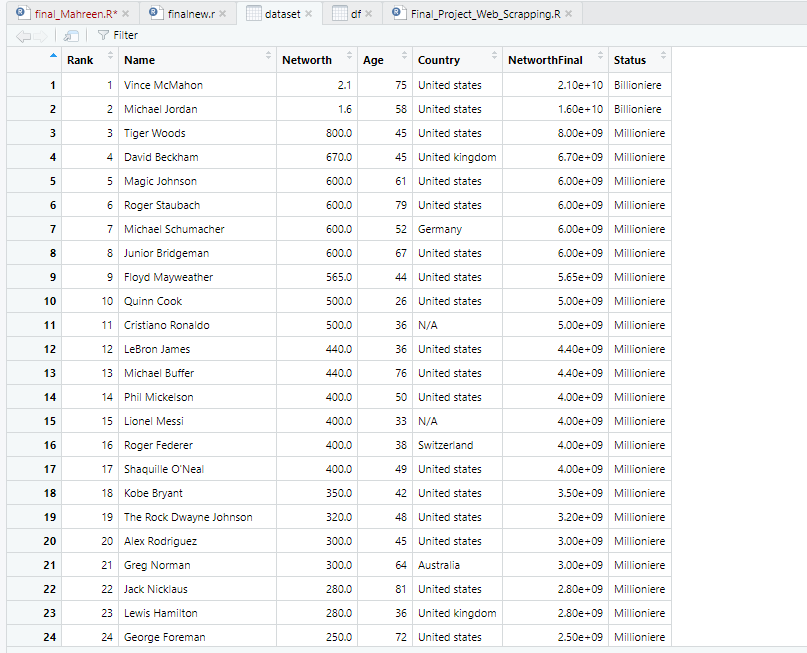


Fig: After integrating the column status.

We have integrated this new column to know the status of the money of the athletes. We have classified this into Billionaire and Millionaire.

1. **Data Reduction:**

Data reduction is the process where the dataset is represented in a reduced version so that it does not affect the output. We have not reduced any data from our dataset.

1. **Data Discretization:**

Dealing with data that are collected from processes that are continuous, such as temperature, ambient light, and a company’s stock price. Data discretization is applied to the data that are continuous to make them more manageable. The provided dataset did not have any continuous value for which data discretization was not necessary.

**Descriptive statistics:**

1. **Mean:**

The mean is the mathematical average of a set of two or more numbers. We use the R mean() function to find the mean.

* mean(df$Age)



1. **Median:**

The median is the value separating the higher half from the lower half of a data sample. We use the R median() function to find the median.

* median(df$Age)

A picture containing text

Description automatically generated

1. **Range:**

The range is calculated by subtracting the lowest value from the highest value.

* max(yr) - min(yr)

A picture containing graphical user interface

Description automatically generated

1. **Variance:**

The variance is a measure of dispersion. It compares each number to the mean. To calculate the variance in R, we use the var() function.

* var(df$Age)

Text

Description automatically generated

1. **Standard Deviation:**

The standard deviation is simply the square root of the variance. It is the measure of how dispersed the data is in relation to the mean. We use the R sd() function to find the sample standard deviation.

* sd(df$Age)

Text

Description automatically generated with low confidence

1. **Quartiles:**

Quartiles are values that separate the data into four equal parts. We use the R quantile() function to find the Quartiles of the values.

* quantile(yr)

Text

Description automatically generated with medium confidence

1. **Percentiles:**

Percentiles are values that separate the data into 100 equal parts. To find the 45th and 65th percentile:

* quantile(yr, c(0.45, 0.65))

Graphical user interface, text, application

Description automatically generated

1. **Interquartile Range:**

The interquartile range is the difference between the first and third quartiles (Q1 and Q3). We use the R IQR() function to find the interquartile range of the values.

* IQR(yr)

Graphical user interface, text, application

Description automatically generated

**Data Visualization:**

For data visualization, we have used the following code,

library(lattice)

library(datasets)

library(tidyverse)

str(df)

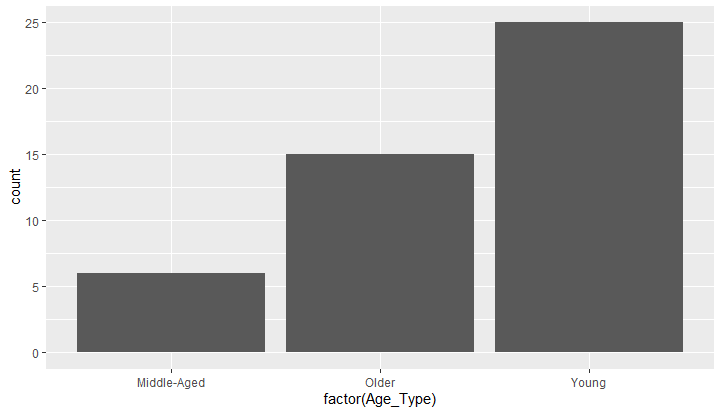
histogram(~Age, data=df)

bwplot(~Age, data=df)

view(df)

library(ggplot2)

ggplot(df, aes(x=factor(Age\_Type))) + geom\_bar()

From this, we have generated a Bar chart for Age-Type. 

Firg: Bar Chart

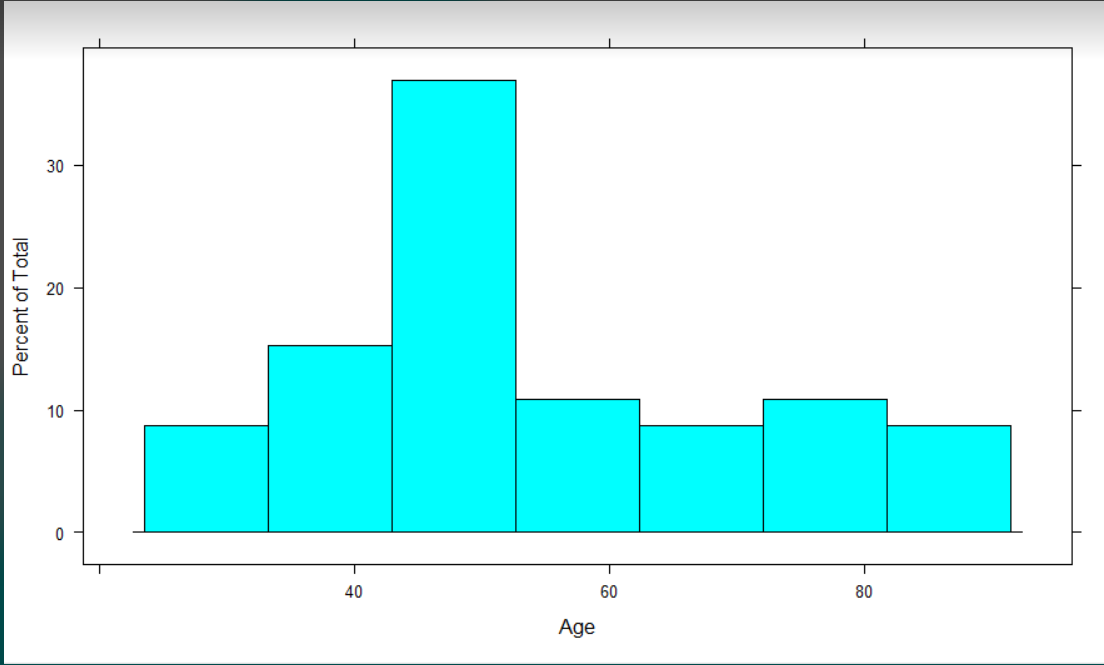
The above graph shows the categories into which the age column have been divided. Here we can see that most of the athletes are from the younger generation, Older category id the second highest and Middle-aged athletes are lowest in number.

**Chart, box and whisker chart

Description automatically generated**

Fig: Box Plot

Here we have checked if there are any outliers and the box plot shows that there are no outliers.

****

**Fig:** Histogram of different age’s richest atheletes.

**Discussion and Conclusion:**

We have gained a dataset by web scrapping using the libraries of the package ‘rvest’. We have checked if the dataset is clean or not in the data pre-processing part. It was mostly a clean dataset so there were no outliers or noisy data. We have also checked if any data was unavailable, and we have found a ‘false’ for that as well. We evaluated the data through descriptive statistics. Lastly, we have visualized the data using graphs. The graphs were plotted using R code.