

|  |  |
| --- | --- |
| Exploratory Data Analysis  Dataset: pisa2009train.csv |  |
| Analyzing the dataset, finding the hidden pattern, and observing the trend in data, so that we can apply ML model for predictive analysis.  By Muhammad Awais  url: https://github.com/mawais38/Exploratory-analysis-and-Linear-Regression-in-R.git  Submitted to  *Dr. Ammar Raja* |  |

# Task Required

We are required to perform complete Exploratory Data Analysis (EDA) on given dataset and provide with the explanation of it. Dataset is given in csv format.

Open blank project in RStudio and setting the working directory that will help us importing dataset easily. Then we load the require library for analysis. After that we load the dataset using read\_csv () method in dataframe as df.

1. # set the Working Directory
2. setwd("E:/UMT(DS)/Business Analytics/session\_11")
4. # Loading Required library
5. library(DataExplorer)
6. library(ggplot2)
7. library(ggthemes)
8. library(corrplot)

 After loading library, we load the dataset, and cheek the dimension of dataset. Then we look at the features of dataset using names () method. Then we move to str () to cheek the basic statistics of dataset.

1. # Loading CSV datafile
2. df <- read.csv('pisa2009train.csv')
4. # Dimension of dataset
5. dim(df)
6. # names of featues
7. names(df)
9. # brief summary of dataset
10. str(df)
11. # statistic of each variable
12. summary(df)

 For the exploratory data analysis, we use DataExplorer library with many useful functions to explore our data and save our time of analysis. Automated data exploration process for analytic tasks and predictive modelling, so that users could focus on understanding data and extracting insights. The package scans and analyses each variable and visualizes them with typical graphical techniques. Common data processing methods are also available to treat and format data. By default, DataExplorer is not installed we need to manually install it.

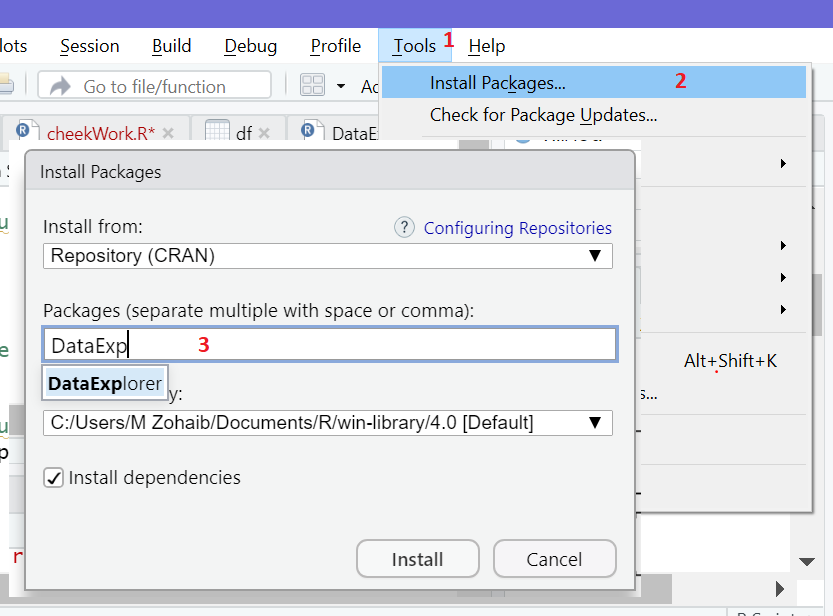


Figure Installing DataExplorer Package

# Features & Structure of Dataset

First, we cheek the documentation of introduce () method using? mark shorthand. It describes the function an input argument to the method. It describes basic information in input dataset. For example

* Number of rows and columns
* Number of continues columns and discrete
* Number of columns with everything missing and missing observation
* Number of rows without missing observation and total observation in dataset

1. # cheeking the docmentation
2. ?introduce
4. # basic information about dataset
5. introduce(df)



Figure Basic Information about the dataset

Then we move on plot\_str () method, this method Visualize data structure of dataset in D3 network graph. We can work with few argument to change the shape of graph form radial to diagonal. We have to type of graph in str() method either ‘diagonal’ or ‘radial’. But we go for diagonal.

1. # cheek the docomentation
2. ?plot\_str
4. # plot the structure of Dataset
5. plot\_str(df, type = "d")

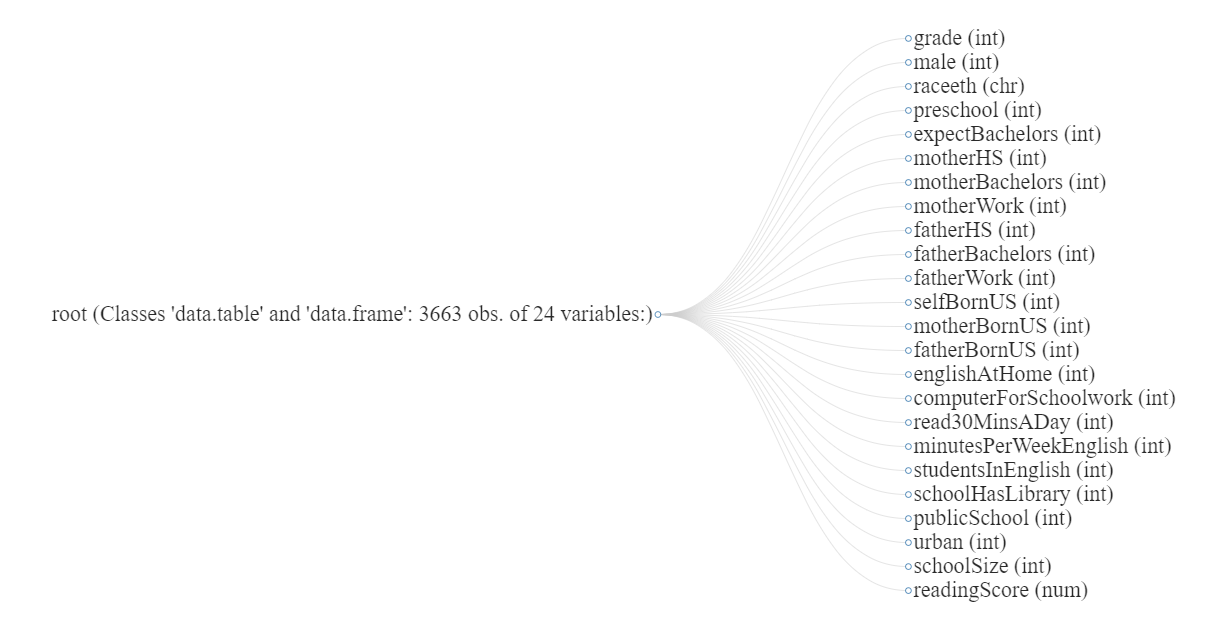


Figure Structure of dataset with variable names and types

# Missing Value Analysis

Now we move on the missing value analysis for that we use plot\_missing() method and reading is documentation helps us understanding the method and its use, there are few argument require to manipulate the graph or we can just pass our dataframe object to see it graph with missing columns.

1. # CHEEK THE MISSINING VALUE IN DATASET
3. # cheek the docmentation
4. ?plot\_missing
6. # plot frequency of missing vlaues
7. plot\_missing(df,
8. geom\_label\_args = list("size"=2.5,"label.padding"=unit(0.2,"lines")),
9. missing\_only=TRUE,
10. theme\_config = list(legend.position = c("right"))
11. )

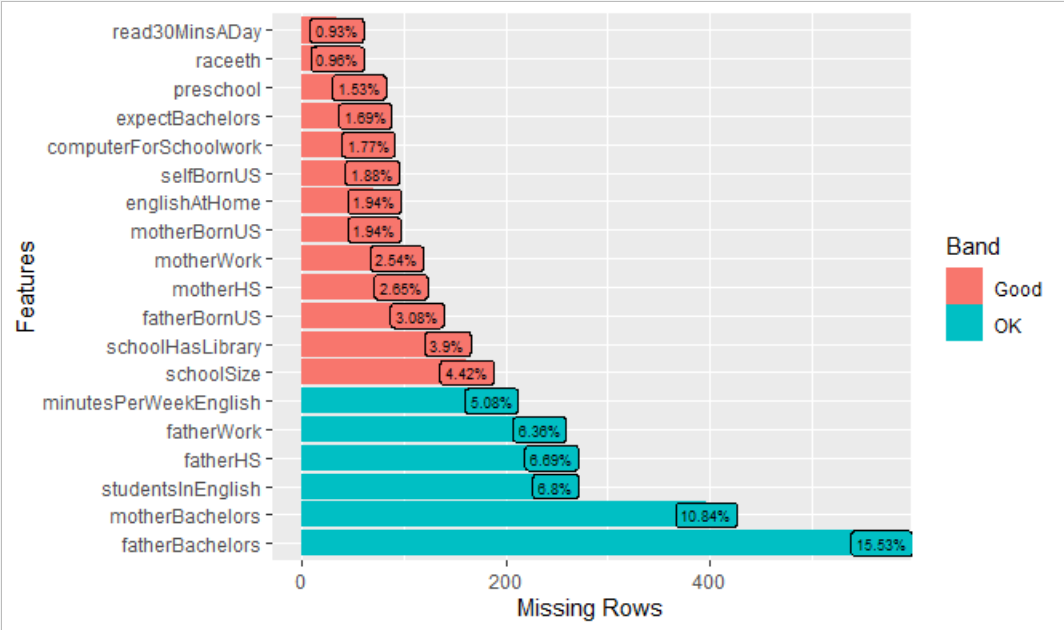


Figure Missing Values analysis of features

We see that column name with missing values in percentage and we can calculate the exact number of missing values in each column. We can either compute that missing values or we can replace with any values or we can remove that missing samples from the dataset. We can calculate missing values either by per columns or we can use summary () method over dataset to get the sum of missing values in each features.

1. # COUNTING MISSING VLAUES
3. sum(is.na(df$read30MinsADay))
4. # or we we can use summary()
5. summary(df)

# Distribution of Features

Now we move on to the distribution of features in our data set, distribution tells us about the how value is in our dataset. Values can follow normal distribution or not, values can be right skewed or left skewed. For that information we can plot the histogram or density plot of dataset to know their distribution of data. For distribution we have following options

* Histogram
* Density plot
* Bar chart

We can plot histogram of each columns or we can use library to plot the all the histogram of continues features. One good thing about plot\_histogram () is that , you pass the dataframe and it automatically detect the continues features in it and plot it histogram.

1. # DISTRIBUTION OF DATASET
3. # cheek the doc
4. ?plot\_histogram
6. # plot histogram
7. plot\_histogram(df,
8. geom\_histogram\_args = list(bins = 35L),
9. title = "Histogram",
10. ggtheme = theme\_minimal(),
11. nrow = 3L,
12. ncol = 3L)

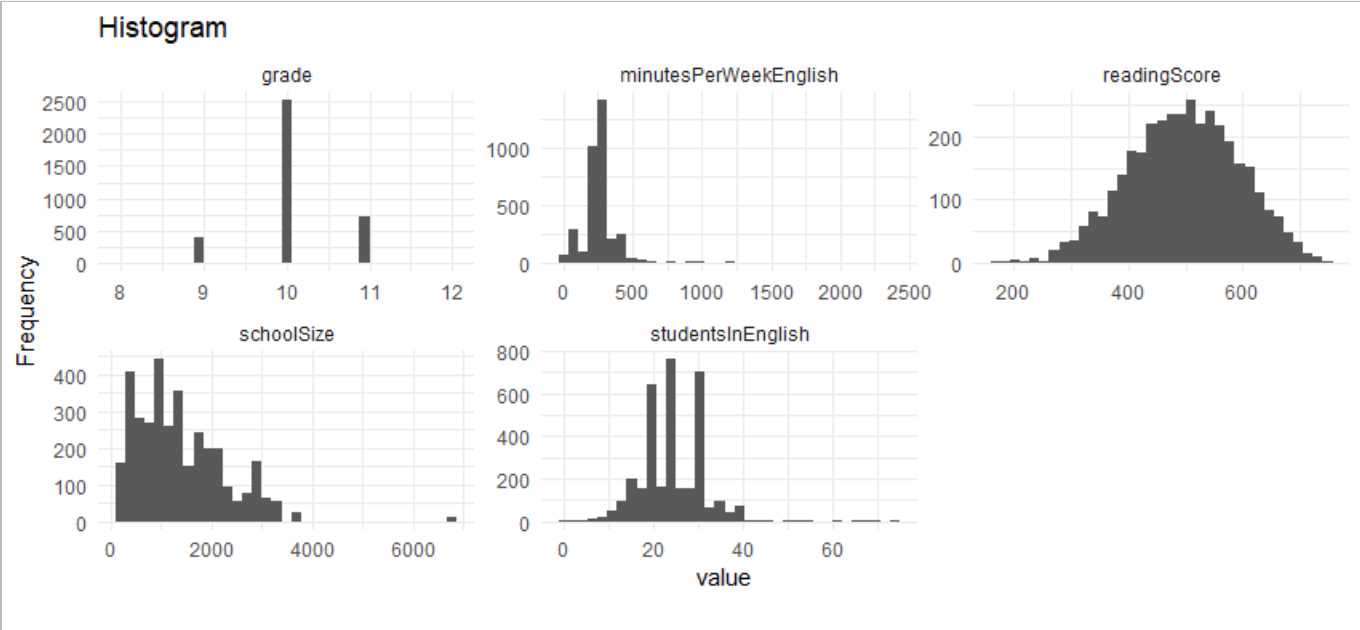


Figure Histogram of continues features

Why is displays only 5 graphs because it only considers the continues columns, and the rest of features contains binary values with 0, 1 and NA. to plot that we need to go for bar chart that help us visualize the data.

And only 1 columns name as readingScore is follows the normal distribution and rest are skewed either to left and right to make it look like normal distribution we need to transform that by apply some transformation e-g log transformation for skewed distribution. There are few other methods available for transformation of data.

# Density plot

Density plot is the representation of distribution of numeric columns. It is smooth version of histogram where we use infinite numbers of bins, to make the plot more smooth. We have handy function of plot\_density () available in DataExplorer library, checking it documentation quite handy to use and make plot for continues variable, as we see in previous graph that we have only 5 continues columns in our data set and rest are just with values 0 and 1. In density plot we will see the same 5 plots.

1. # DENSITY PLOT / SOMOOTH HISTOGRAM
3. ?plot\_density
5. # plot density graph
6. plot\_density(df,
7. title = "Density Plot",
8. ggtheme = theme\_minimal(),
9. nrow = 3L,
10. ncol = 3L
11. )

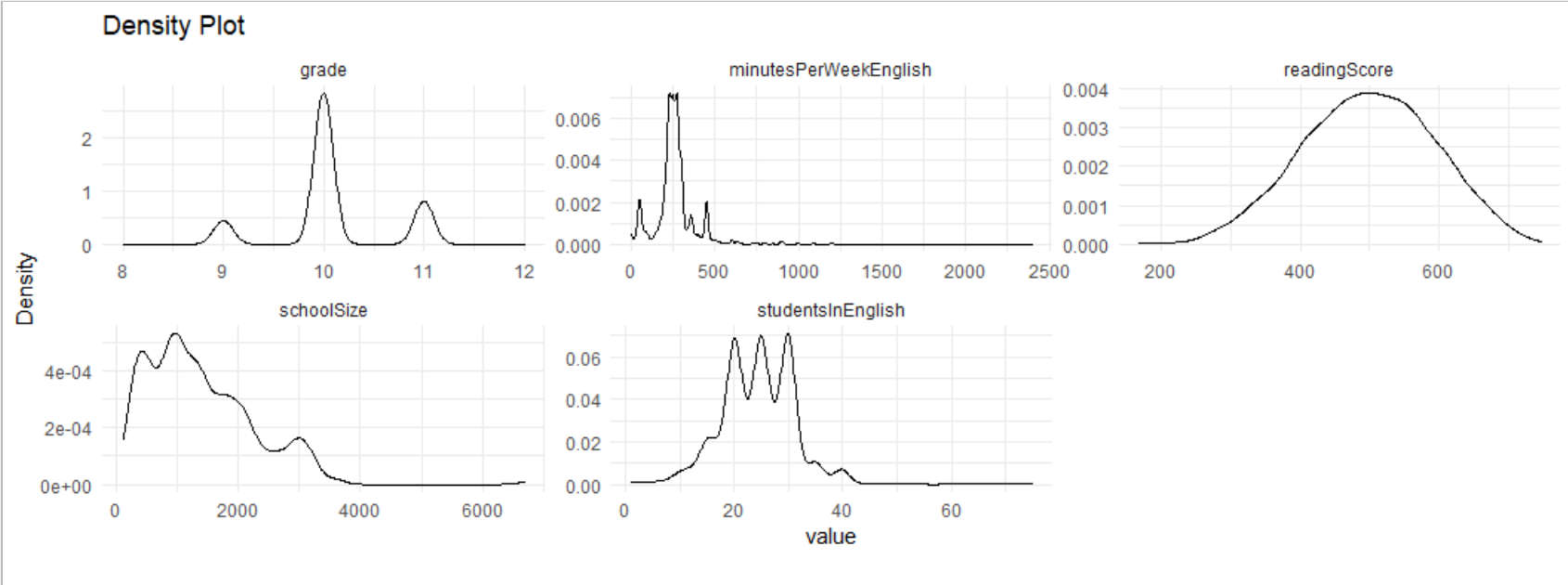


Figure Smooth Density plot for continues features

 Explanation for Density plot:

What is a density plot? A [density plot](http://serialmentor.com/dataviz/histograms-density-plots.html) is a smoothed, continuous version of a histogram estimated from the data. The most common form of estimation is known as [kernel density estimation](https://en.wikipedia.org/wiki/Kernel_density_estimation). In this method, a continuous curve (the kernel) is drawn at every individual data point and all these curves are then added together to make a single smooth density estimation. The kernel most often used is a Gaussian (which produces a Gaussian bell curve at each data point).

The x-axis is the value of the variable just like in a histogram, but [what exactly does the y-axis represent](https://stats.stackexchange.com/questions/48109/what-does-the-y-axis-in-a-kernel-density-plot-mean)? The y-axis in a density plot is the probability density function for the kernel density estimation. However, we need to be careful to specify this is a probability density and not a probability. The difference is the probability density is the probability per unit on the x-axis. To convert to an actual probability, we need to find the area under the curve for a specific interval on the x-axis. Somewhat confusingly, because this is a probability density and not a probability, the [y-axis can take values greater than one.](https://stackoverflow.com/questions/42661973/r-density-plot-y-axis-larger-than-1) The only requirement of the density plot is that the total area under the curve integrates to one.

# Bar chart for discrete variables

A bar chart or bar graph presents data with rectangular bars at heights or lengths proportional to the values they represent. The bars can be vertical or horizontal, though they are usually vertical. To analyze the categorical variable or discrete variable/features we can use bar chart, to visualizes the distribution of values in column. We use plot\_bar() method of DataExplorer. And we clearly see that we have most of columns with values 0,1 and some NA values in the distribution of data.

1. # BAR CHAR FOR CATAGORIAL VAIRALES / DISCRETE
3. # cheek the doc
4. ?plot\_bar
6. # bar chart for discrete variable
7. plot\_bar(df,
8. title = "Bar Chart",
9. ggtheme = theme\_minimal(),
10. nrow = 5L,
11. ncol = 5L,
12. )

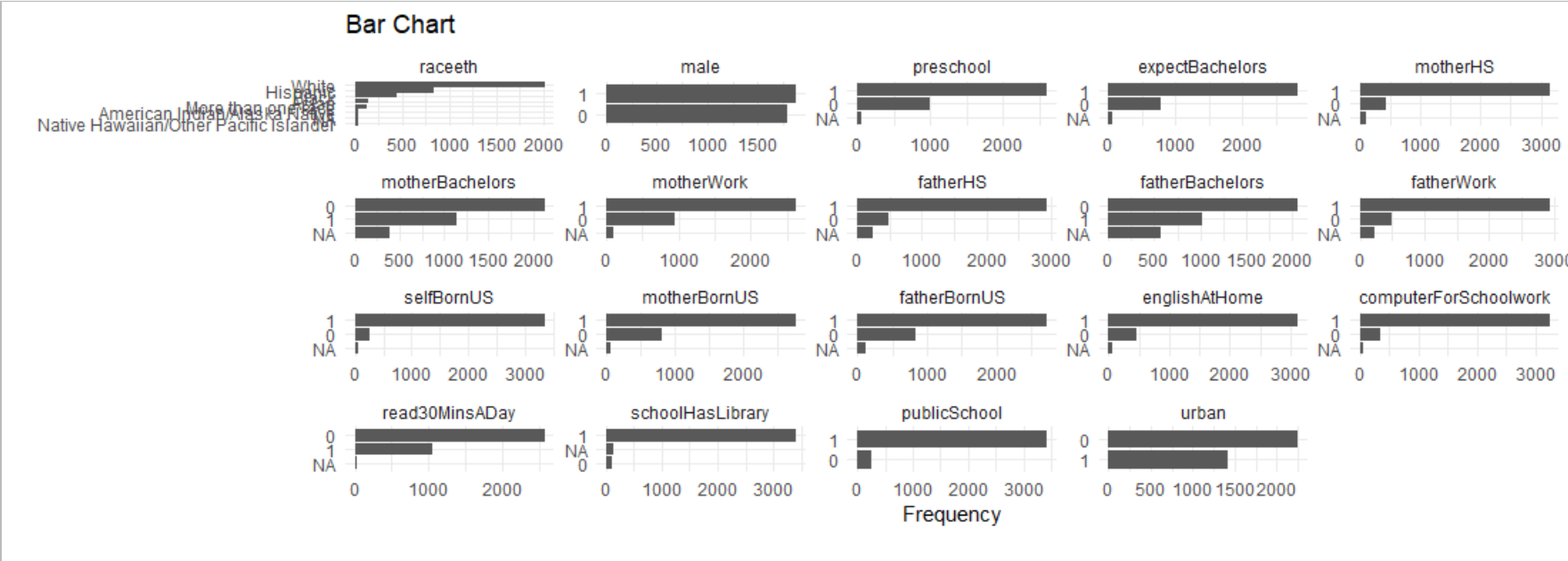
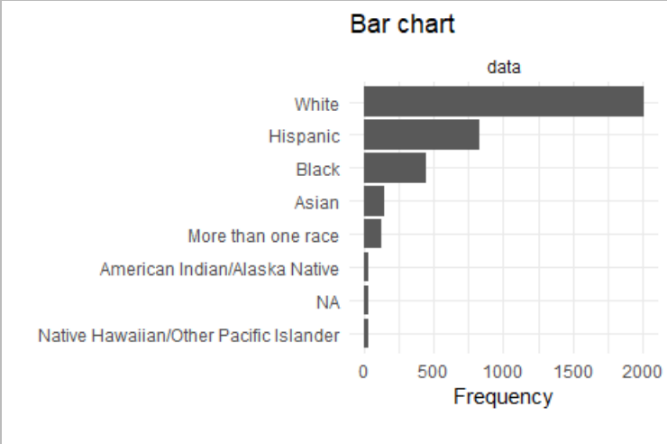


Figure Bar char for discrete features

For in depth analysis, we can plot for individual or group some columns to plot for clear result, when features or number of columns increase, we need to use library for bulk plot and get the ideas about the distribution of data.

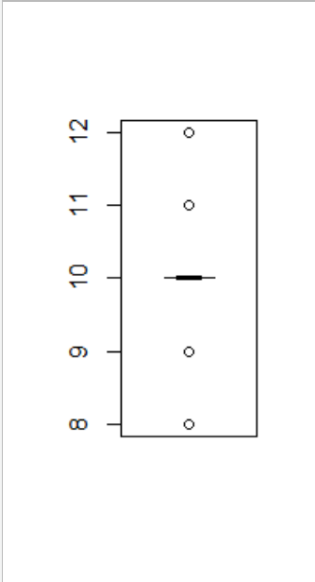
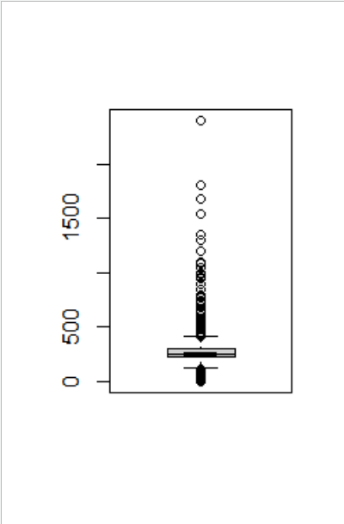
1. # ploting raceeth colums
2. plot\_bar(df$raceeth,
3. title = "Bar chart",
4. ggtheme = theme\_minimal(),
5. nrow = 1L,
6. ncol = 1L)

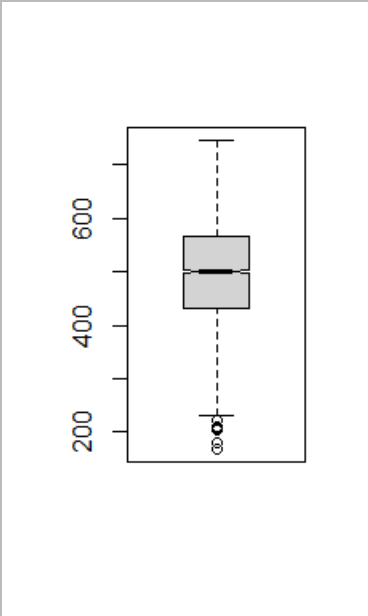
Figure 8 raceeth bar chart

# Boxplot for outlier’s detection

Boxplot is a descriptive method for analysis for data distribution and outlier detection, it is purely based on statistics, first calculate the quartile and then interquartile range and stuff like that, but we are concern with plot that visually display information about columns distribution and outliers in it. It helps us identification of outliers then we decide how we can deal with outliers in our dataset.

As in above graph, histogram and density plot we came to know that although we have 24 features in our dataset, there are 5 continues numeric columns and the rest are with binary values 0’s and 1’s.

1. # BOXPLOT FOR OUTLIER DETECTION
3. ?boxplot
4. boxplot(df$grade,notch = TRUE)
5. boxplot(df$minutesPerWeekEnglish,notch = TRUE)
6. boxplot(df$readingScore,notch = TRUE)
7. boxplot(df$schoolSize,notch = TRUE)
8. boxplot(df$studentsInEnglish)



We have only few Variable with continues value. We do have some outliers in our data distribution.

# Correlation of Features

Co-relation plays very important role in features selection for ML model. It tells us how features are related to each other or we can cheek how one variable effect the output or target variable that we are trying to predict. Co-relation values vary from -1 to 1. 1 means strong relation means that if one variable is increasing other variable is also increasing. And -1 correlation means if one variable is increasing other is decreasing. And 0 means that both variables have no co-relation.

To calculate correlation between 2 variable we have to make sure that both are numeric and have no NA values in it.

1. # CORELATION HEATMAP TO FEATUES VARIABLE
3. ?corrplot
4. library(corrplot)
6. smaple <- df[,15:20]
7. smaple <- na.omit(smaple)
8. m <- cor(smaple)
9. corrplot(m,
10. method = "number",
11. bg='white')

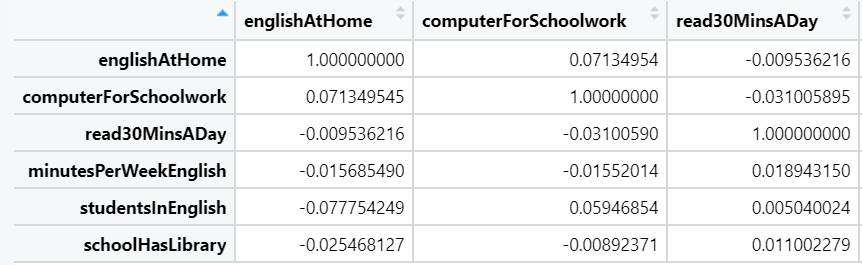


Figure correlation matrix of few variable

We can plot correlation in different format. As we can see the correlation is quite week.

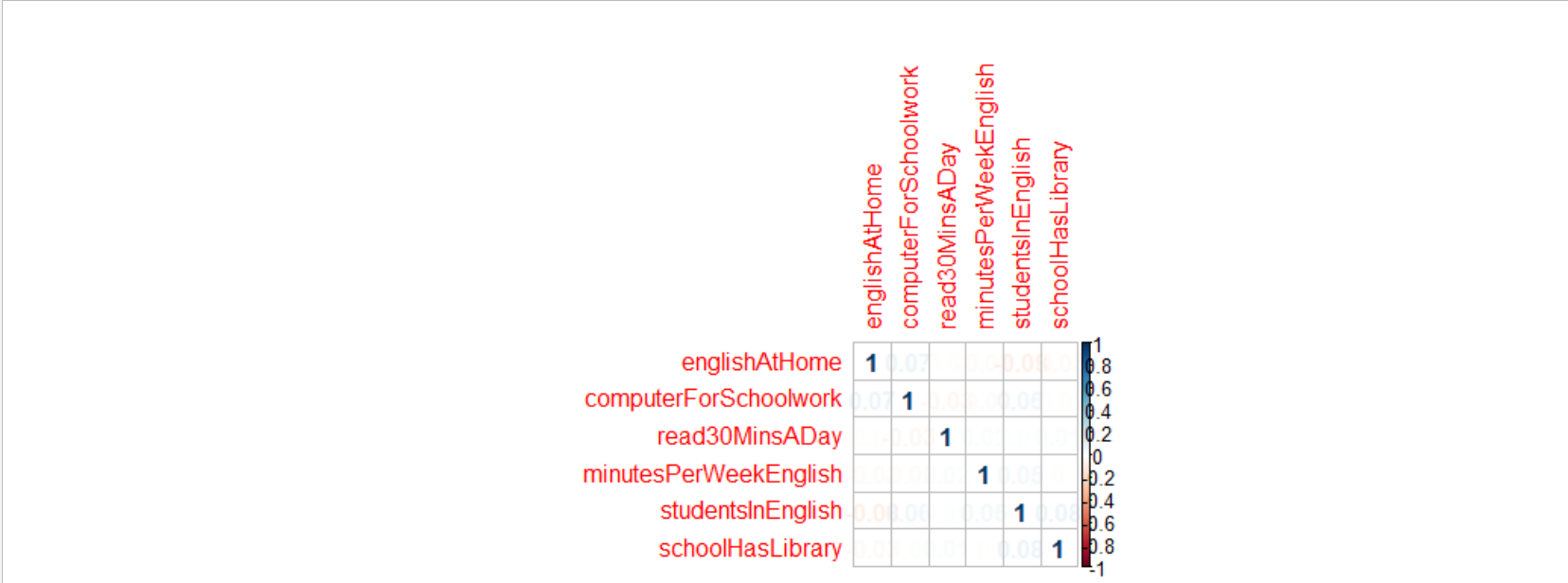


Figure 10 correlation matrix plot