Configuration Manual

Identification and Prediction of Factors Impact America Health Insurance Premium

Jun Jun Sun
Student ID: X17162238

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Jun Jun Sun ……………………………………………………………………………………………

Student ID: X17162238 …………………………………………………………………………………………………….

Programme: Data Analytics …………………………………… **Year:** …2020…………….

Module: MSc Data Analytics Research Project………………………………………..………

Lecturer: Dr. Catherine Mulwa ……………………………………………………………….………

Submission Due Date: 17/08/2020…………………………………………………………………………….………

Project Title: Identification and Prediction of Factors Impact America Health Insurance Premium…………………………………………………………………….………

Word Count: 2107……………………………… **Page Count:** 32………………………….……..………

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: JunJunSun ……………………………………………………………………………………………………

Date: 17/08/2020……………………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Jun Jun Sun
Student ID: x17162238

1 Introduction

The configuration manual determines from the begin setup stage till the result of the whole process. The purpose of this project is to define a best performance model to achieve the objective and address the research question. There are several models generated and conducted to identify the result. This configuration manual includes an explanation of hardware and software properties and installation used, with an implementation of each step of the process which includes data preparation, model code generated and results.

The structures of this configuration manual report are as follows:

Chapter 2: Discover the environment specification and configuration

Chapter 3: Explain the data preparation

Chapter 4: Discover model implementation and steps of each output generated

Chapter 5: Walk-through appendix

2 Environment Specification and Configuration

The environment specification and configuration deliver specifics of what systems are required to develop and implemented for this project, hardware and software are the key elements to implement this project. This chapter majority is to discover the integrated configuration environments that were used.

2.1 Hardware Configurations

This section will discuss the details of hardware, figure 1 shows the machine used for the implementation of this project. Windows 10 systems running on a Lenovo laptop named LAPTOP-3HR9A551 with 64-bit operating system, 2.7GHz processor and 8GM RAM was used.

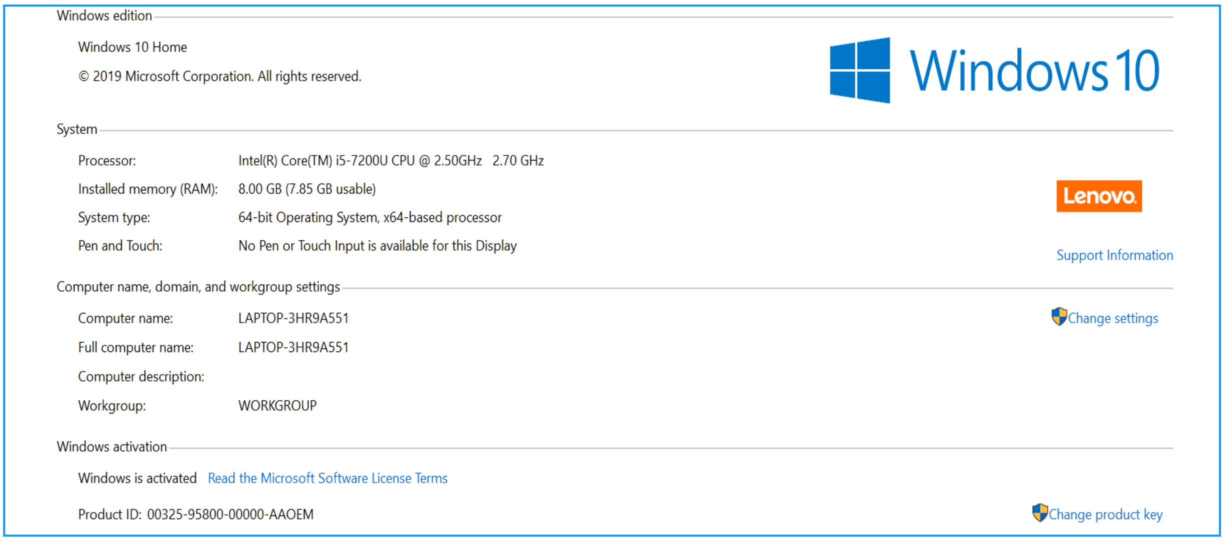

Figure 1 Hardware Configuration

2.2 Software Configurations

This section will discuss the details of software implementation and installation.

2.2.1 RStudio

The RStudio was downloaded from here[1].

The RStudio used version 1.3.1056 to implement for this project shows in figure 2.

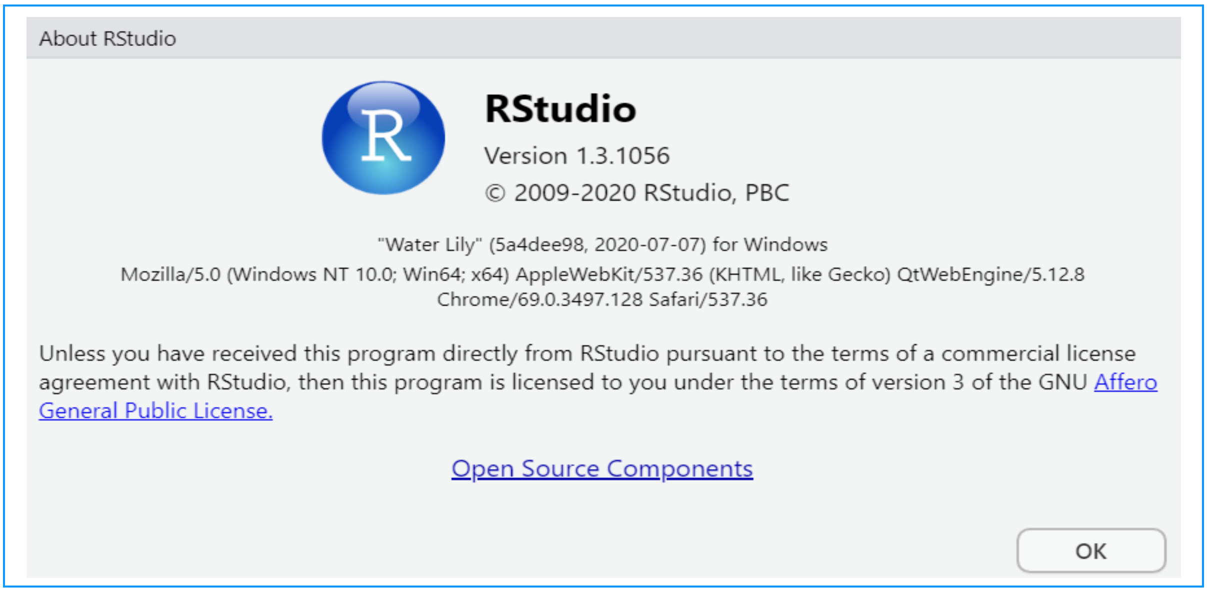

Figure 2 RStudio Version Used for Implementation

The RStudio properties (Figure 3) shown it has been created on 10/04/2018 from the following computer name LAPTOP-3HR9A551.

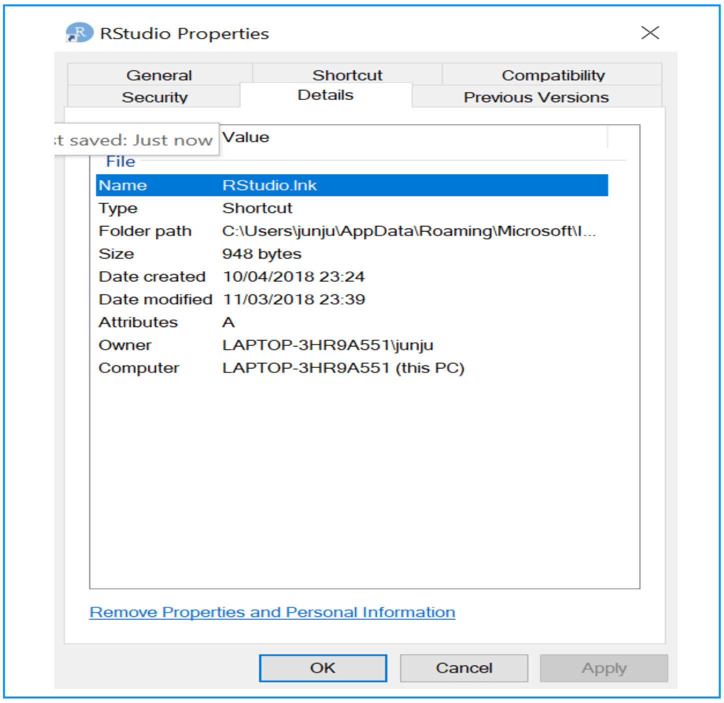

Figure 3 RStudio Properties

[1] https://rstudio.com/products/rstudio/download/

2.2.2 Tableau

The Tableau was downloaded from here[2].

The version of Tableau Desktop professional edition 2020 was used for this project shown in figure 4.

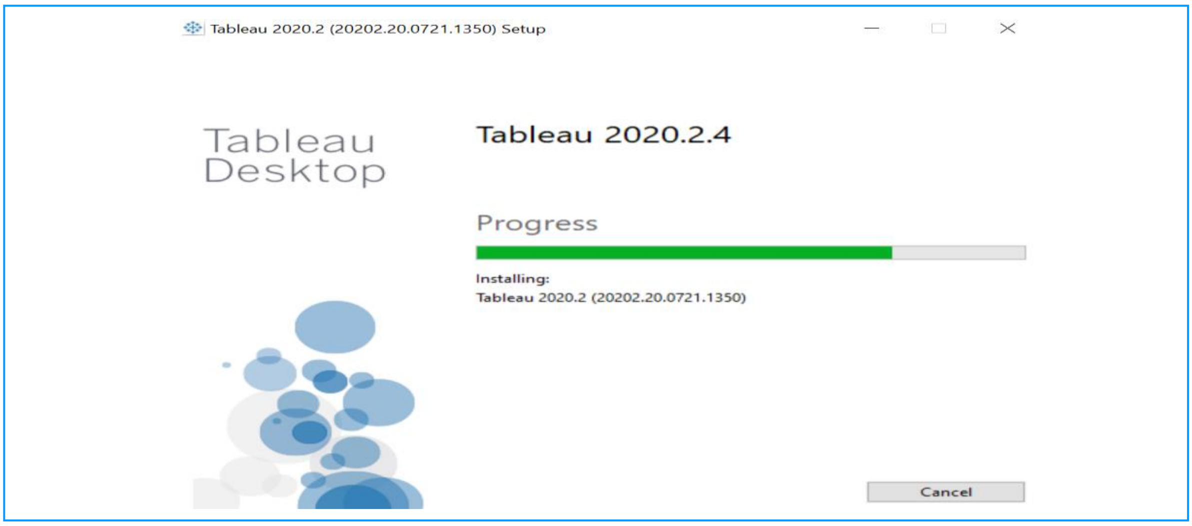

Figure 4 Tableau Version

The Tableau was created on 22/10/2019, which is also installed on this computer name LAPTOP-3HR9A551 shown at left in Figure 5. But the error appeared when I tried to modify graphs it shown update request that the Tableau, therefore, the update version 2020 was installed for modification details as figure 5 right.

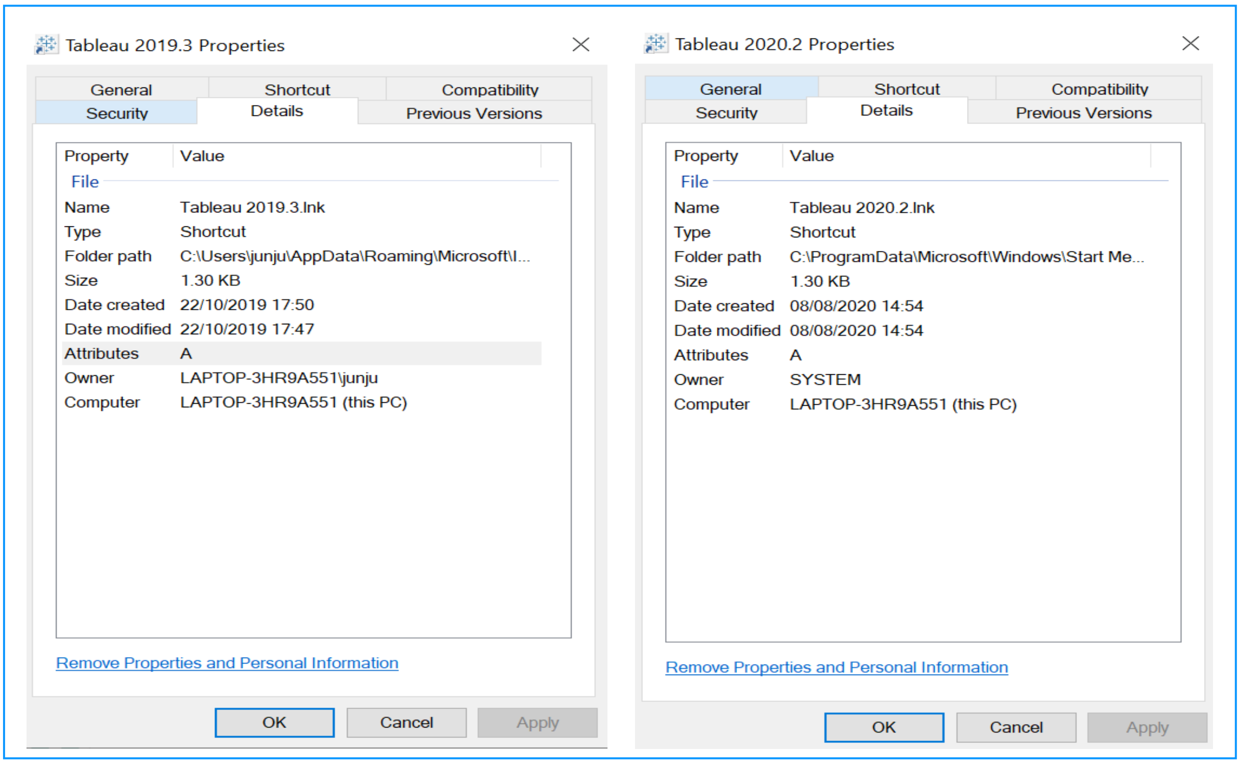

Figure 5 Tableau Properties

2 https://www.tableau.com/products/desktop/download?signin=academic

2.2.3 IMB SPSS Statistics

IBM SPSS Statistics version 26 (Figure 6) was downloaded from here[3].

SPSS was used to implement Statistic model.

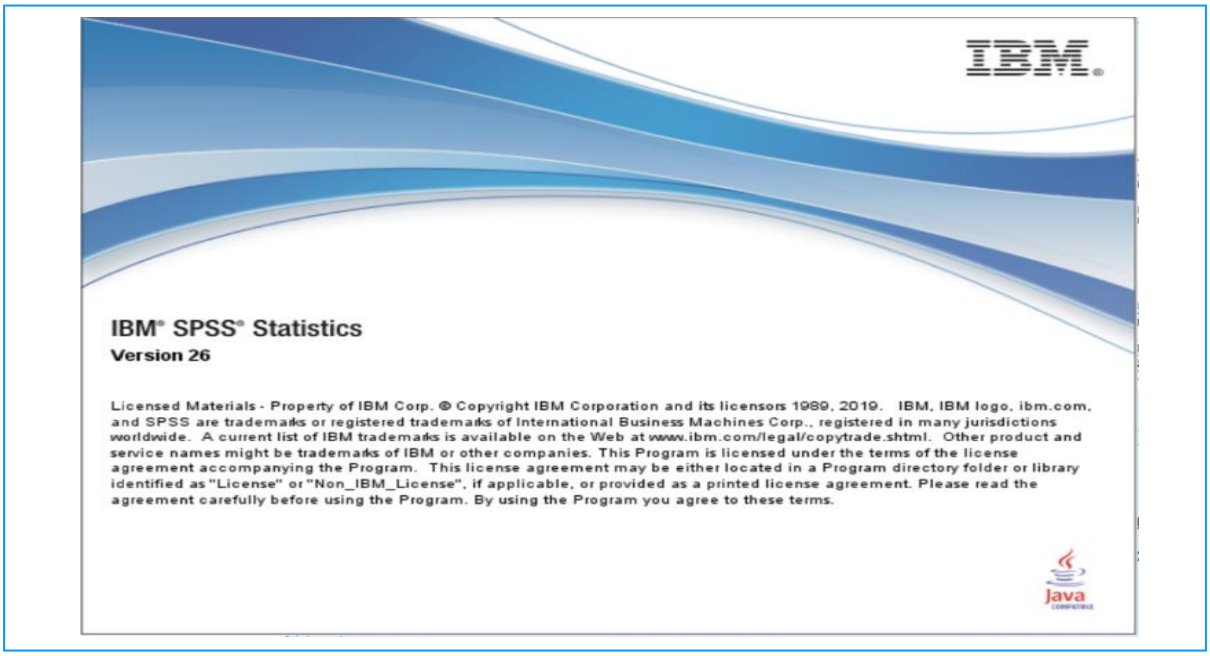

Figure 6 SPSS Statistics Version

From the IBM SPSS statistics 26 properties (Figure 7) shown it was created on 02/10/2019, installed on my machine name PC LAPTOP-3HR9A551.

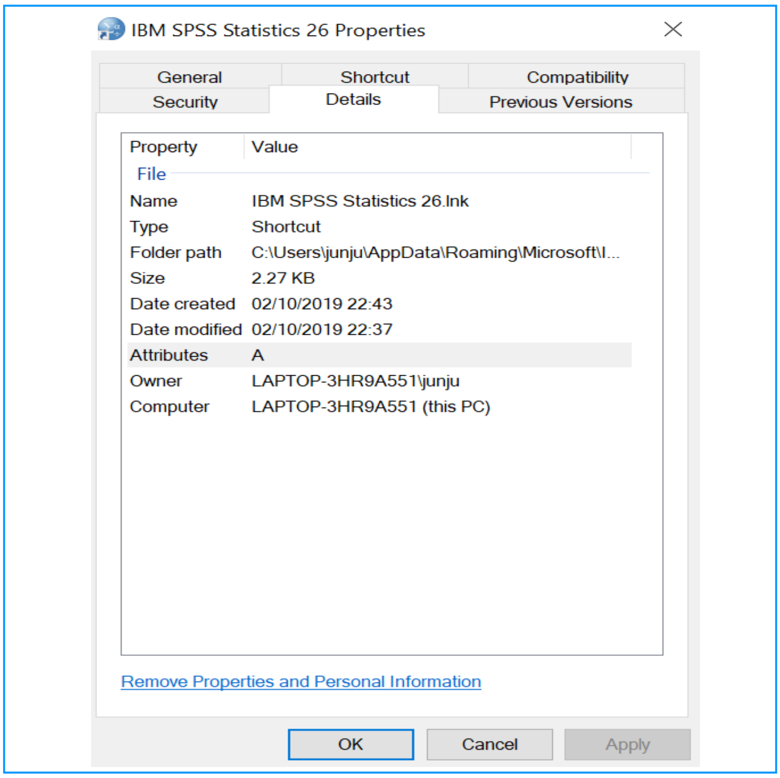

Figure 7 SPSS Statistics Properties

[3] https://itsupport.ncirl.ie/hc/en-ie/articles/360014035839-How-do-I-install-SPSS-

3 Data Preparation

Data was extracted from Kaggle and Data.World website, datasets have been already in csv format, there were five csv datasets used. Each one was cleaned, formatted and prepared into one final dataset. The original five datasets can be located from these links[4].

3.1 Data Preparation used RStudio

3.1.1 Package Install in RStudio

All the below modelling (Table 1) functions used split data, training and testing. This was divided up into 70% training data and the remaining 30% for testing data. Both data were saved into csv files to be used on each model for reusability to save time, duplication and complexity.

Table 1 RStudio Package Installed

Name	Model	Package
Data Preparation		library(dplyr)
Multiple Linear Regression	lm	library(ggplot2)
Random Forest	randomForest	library(randomForest)
Naïve Bayes	naiveBayes	library(e1071)
Logistic Regression	glm	N/A No packet needs for model
Support Vector Machine	svm	library(e1071)
Decision Tree	rpart	library(party) library(rpart)
K-Nearest Neighbour	knn	library(class) Used for the model
Accuracy Check		library(caret) library(class) library(gmodels)

3.1.2 Data Clean and Encode

```
#-Read the data set file
insurance1DS <- read.csv("datasets_26475_38092_insurance2.csv", header=TRUE, stringsAsFactors=FALSE, fileEncoding="latin1")
insurance1 <- select(insurance1DS, age, sex, bmi, children, smoker, region, charges)
insurance1$smoker <- ifelse(insurance1DS$smoker=="1", "yes", "no")
insurance1$sex <- ifelse(insurance1DS$sex=="1", "Male", "Female")


#-Read the data set file
insurance2DS  <- read.csv("datasets_26475_38092_insurance3r2.csv", header=TRUE, stringsAsFactors=FALSE, fileEncoding="latin1")
insurance2 <- select(insurance2DS, age, sex, bmi ,children, smoker, region, charges)
insurance2$smoker <- ifelse(insurance2DS$smoker=="1", "yes", "no")
insurance2$sex <- ifelse(insurance2DS$sex=="1", "Male", "Female")
```

Figure 8 Clean First Two Datasets

There are five csv datasets were selected for this project, figure 8 shows the reading of first and second csv files, selecting 7 variables, converting both smoker and sex data values from numeric to factor value.

4 https://www.kaggle.com/easonlai/sample-insurance-claim-prediction-dataset
https://www.kaggle.com/hiralpandhi/healthcaredataset?select=test_2v.csv
https://data.world/healthdatany/gaf8-ac33

```r
#-Merged Data sets
mergedInsurance <- rbind(insurance1, insurance2)

#-Convert numerical values to description values
for(ns in 1:nrow(mergedInsurance)) {
  if(mergedInsurance$region[ns] == 0){
    mergedInsurance$region[ns] <- "northeast"
  }else if(mergedInsurance$region[ns] == 1){
    mergedInsurance$region[ns] <- "northwest"
  }else if(mergedInsurance$region[ns] == 2){
    mergedInsurance$region[ns] <- "southeast"
  }else (mergedInsurance$region[ns] <- "southwest")
}
```

Figure 9 Merge First Two Datasets

Figure 9 shows the merging of the first and second csv file into one dataset (mergeInsurance) after the merged has been completed the process loops through each mergeInsurance row and assign each region value to a descriptive value.

```r
#-Read the data set file
train_2v  <- read.csv("train_2v.csv", header=TRUE, stringsAsFactors=FALSE, fileEncoding="latin1")
newTrain <- select(train_2v, age, gender, bmi, smoking_status)
newTrain[newTrain==""]<-NA
newTrain <- na.omit(newTrain)
newTrain$smoking_status <- ifelse(newTrain$smoking_status=="never smoked", "no", "yes")
newTrain <- subset(newTrain, age >= 18)
newTrain <- subset(newTrain, bmi >= 40 & bmi <= 70)

#-Read the data set file
test_2v <- read.csv("test_2v.csv", header=TRUE, stringsAsFactors=FALSE, fileEncoding="latin1")
newTest <- select(test_2v, age, gender, bmi, smoking_status)
newTest[newTest==""]<-NA
newTest <- na.omit(newTest)
newTest$smoking_status <- ifelse(newTest$smoking_status=="never smoked", "no", "yes")
newTest <- subset(newTest, age >= 18)
newTest <- subset(newTest, bmi >= 40 & bmi <= 70)
```

Figure 10 Clean Other Two Datasets

The above figure 10 reads third and fourth dataset, both datasets were select 4 variables and assign NA into empty value row to allow them to be removed. And anything else into smoker value. The selection was also done by age group which over or equal to18 and bmi group between 40 to 70.

```r
#-Merged Data sets
mergedTestTrain <- rbind(newTest, newTrain)

#-Naming the columns
colnames(mergedTestTrain)=c("age","sex", "bmi", "smoker")

#-Check for NA values in the merged Data sets
any(is.na(mergedTestTrain))
```

Figure 11 Merged Other Two Datasets

The figure 11 shows the merging of the third and the fourth dataset into one dataset called mergeTestTrain, then modifies all the four column names to match to first two datasets, put these variables into mergeTestTrain data, and check any NA value.

```
#-Read the data set file
chargesData <- read.csv("Inpatient_Prospective_Payment_System__IPPS__Provider_Summary_for_the_Top_100_Diagnosis-Related_Groups__DRG__-_FY2011.csv",
                        header=TRUE, stringsAsFactors=FALSE, fileEncoding="latin1")
chargeData <- select(chargesData, Average.Total.Payments)
chargeData1 <- subset(chargeData, Average.Total.Payments >= 40000 & Average.Total.Payments <= 70000)
chargeData1 <- unique(chargeData1)


#-Add one region values to each mergedTestTrain Data sets row
regions <- c("northeast","northwest", "southeast", "southwest")
mergedTestTrain$region <- sample(regions, size = nrow(mergedTestTrain), replace = TRUE)

#-Add number of children to each row in mergedTestTrain Data sets
children <- c(0,1,2,3,4,5)
mergedTestTrain$children <- sample(children, size = nrow(mergedTestTrain), replace = TRUE)

#-Add charges to each row in mergedTestTrain Data sets
mergedTestTrain$charges <- sample(chargeData1$Average.Total.Payments, size = nrow(mergedTestTrain), replace = TRUE)

#-Merged both data sets into insuranceData
insuranceData <- rbind(mergedInsurance, mergedTestTrain)

#-Check for NA values in insuranceData set
any(is.na(insuranceData))
```

Figure 12 Clean Fifth Dataset Then Merge Final Datasets

This figure 12 reads the fifth csv file, and then select charges value between 40000 to 70000 with unique value. Create two new columns region and children, then added 4 different regions and number of children (0-5) to its own variable to be added into MergeTestTrain dataset. All five datasets were added into master dataset called insuranceData.

3.1.3 Presenting Data

```
#-Insurance Data Boxplot Displaying Age & BMI
insurance_boxplot <- insuranceData %>%
  select(c(1,3)) %>%
  tidyr::gather()
boxplot <- ggplot(insurance_boxplot, aes(x = key, y = value)) +
  labs(x = "variable", title = "Insurance Data Boxplot") +
  geom_boxplot(outlier.colour = "red", fill="white",outlier.shape = 2)
boxplot
```

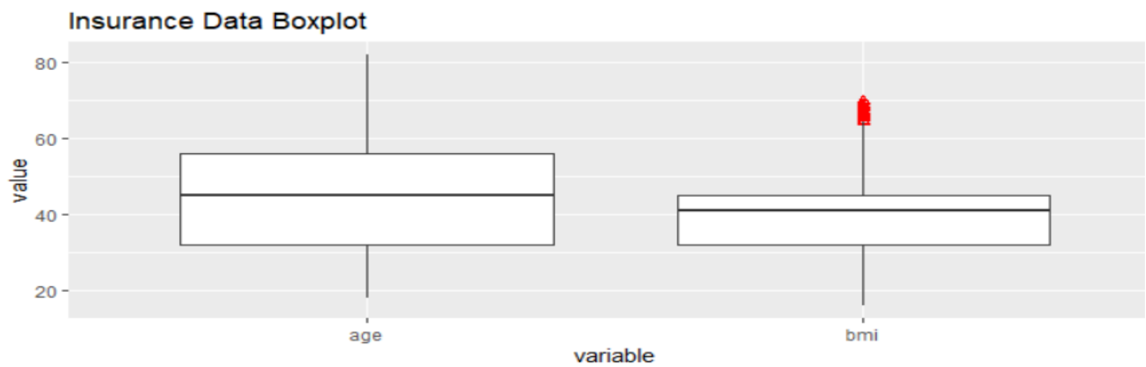

Figure 13 Data Presentation of age and bmi Distribution

Figure 13 shows the data presentation using boxplot generate age and bmi, showing the outlier of the data in red.

```r
#-Insurance Data Histogram Chart Displaying Age, BMI & Charges
insurance_hist <- insuranceData %>%
  select(c(1,3,7)) %>%
  tidyr::gather()
hist <- ggplot(data = insurance_hist, mapping = aes(x = value)) +
  geom_histogram(bins = 10, color = "black", fill="blue") +
  facet_wrap(~key, scales = 'free_x')
hist
```

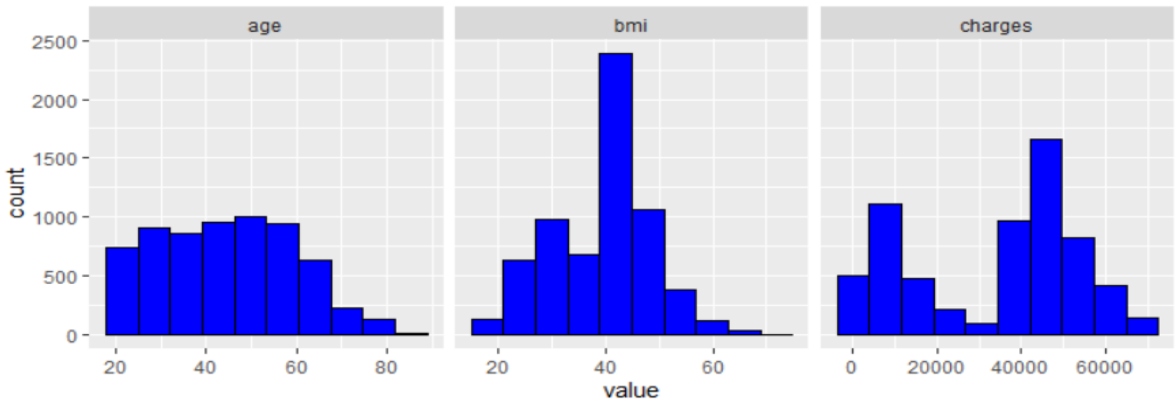

Figure 14 Data Presentation for age, bmi, charges Distribution

The figure 14 presents data distribution from generated histogram, used age, bmi and charges variables shows the count and the number of values related each group.

```r
#-Insurance Data Bar Chart Displaying Sex, Children, Smoker & Region
insurance_bar <- insuranceData %>%
  select(c(2,4:6 )) %>%
  tidyr::gather()
barchart <- ggplot(data = insurance_bar, mapping = aes(x= value),colorspaces) +
  geom_bar(colour = "black", fill= "lightblue") +
  facet_wrap(~key, scales = 'free_x')
barchart
```

Figure 15 Data Presentation children, sex, region, smoker

The figure 15 using bar charts generate the data presentation, which demonstrating four aspect groups for children, sex, smoker and region distribution.

3.1.4 Split into Training and Testing Dataset

```
#-Set data types
insuranceData$charges <- as.numeric(insuranceData$charges)
insuranceData$age <- as.integer(insuranceData$age)
insuranceData$sex <- as.factor(insuranceData$sex)
insuranceData$children <- as.integer(insuranceData$children)
insuranceData$smoker <- as.factor(insuranceData$smoker)
insuranceData$region <- as.factor(insuranceData$region)
str(insuranceData)

#-Create insurance Data file
write.csv(insuranceData, "insuranceData.csv", row.names=FALSE)

#Split the data 70/30
percentData <- round(0.7 * nrow(insuranceData))
sampleData <- sample(1:nrow(insuranceData), percentData)
trainInsuranceData <- insuranceData[sampleData, ]
testInsuranceData <- insuranceData[-sampleData, ]
```

```
#-Create a Train and Test files
write.csv(trainInsuranceData, "trainInsuranceData.csv", row.names=FALSE)
write.csv(testInsuranceData, "testInsuranceData.csv", row.names=FALSE)

#-Read the data set files and store in variables
train <- read.csv("trainInsuranceData.csv", stringsAsFactors = TRUE, header=TRUE, fileEncoding="latin1")
test <- read.csv("testInsuranceData.csv", stringsAsFactors = TRUE ,header=TRUE, fileEncoding="latin1")

#-Check for NA values in train & test data sets
any(is.na(train))
any(is.na(test))
```

```
> #-Set data types
> insuranceData$charges <- as.numeric(insuranceData$charges)
> insuranceData$age <- as.integer(insuranceData$age)
> insuranceData$sex <- as.factor(insuranceData$sex)
> insuranceData$children <- as.integer(insuranceData$children)
> insuranceData$smoker <- as.factor(insuranceData$smoker)
> insuranceData$region <- as.factor(insuranceData$region)
> str(insuranceData)
'data.frame':	6406 obs. of  7 variables:
 $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
 $ sex     : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 1 1 1 2 1 ...
 $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
 $ children: int  0 1 3 0 0 0 1 3 2 0 ...
 $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
 $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
 $ charges : num  16885 1726 4449 21984 3867 ...
```

Figure 16 Split to Training and Testing Data

The code in figure 16 shows the setting to data types to meet the model requirement and save the insuranceData file, the data is then split into 70% training and 30% testing and saved both datasets to be kept secure. Both train and test files are read and store in its own variable to be used in each model.

3.2 Data Preparation uses SPSS

The first step was imported csv file into SPSS software (Figure 17).

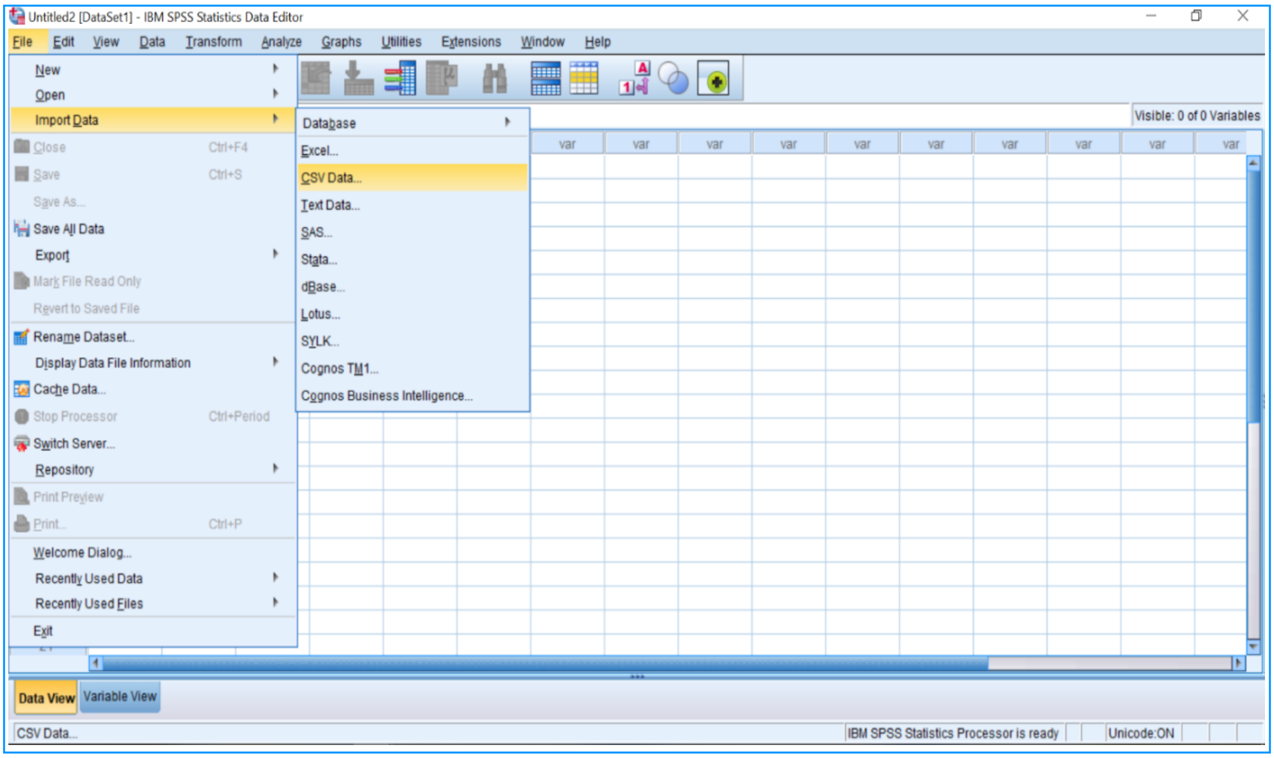

Figure 17 SPSS Data Import

The second step was encoded data used to transform and selected record into same variables (Figure 18).

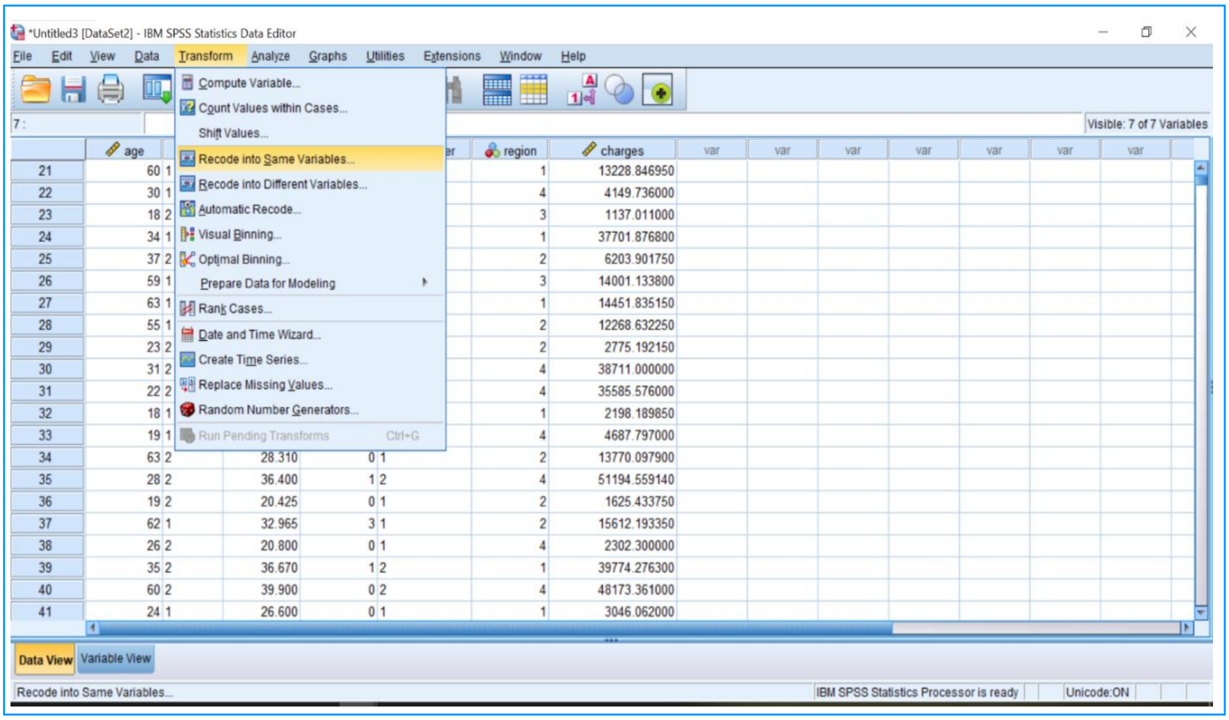

Figure 18 Data Encode

The third step below was modified sex variable into a numeric value (Figure 19).

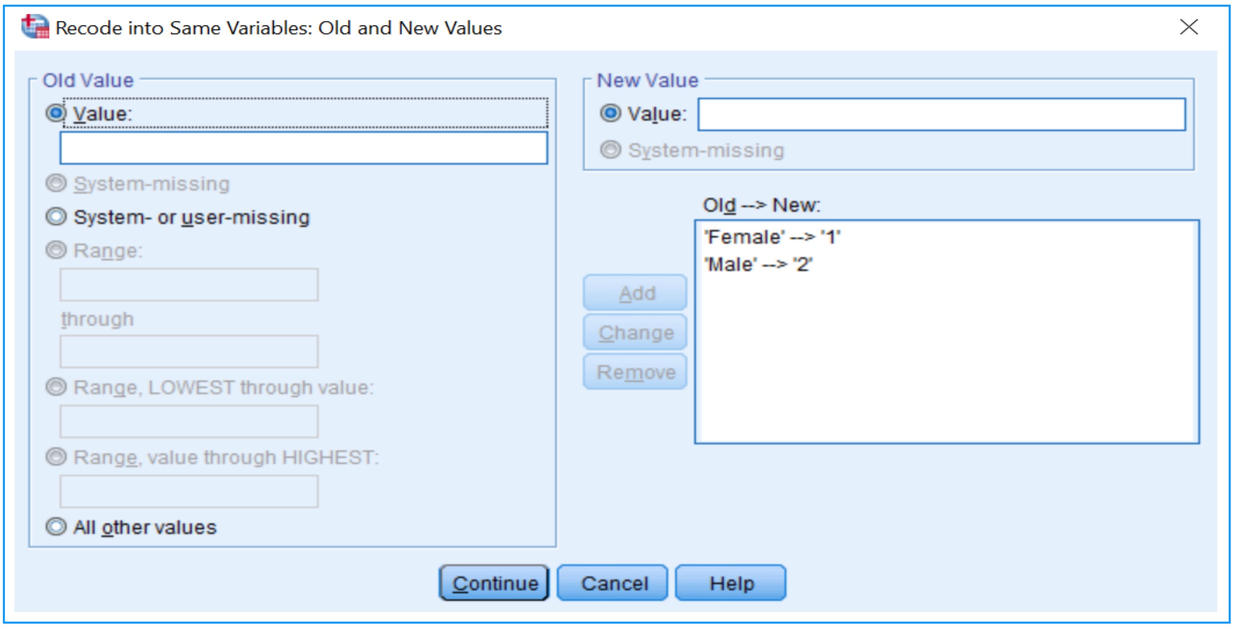

Figure 19 Change Variable sex to Number

This step was same as previous step change variable smoker to a numeric value (Figure 20).

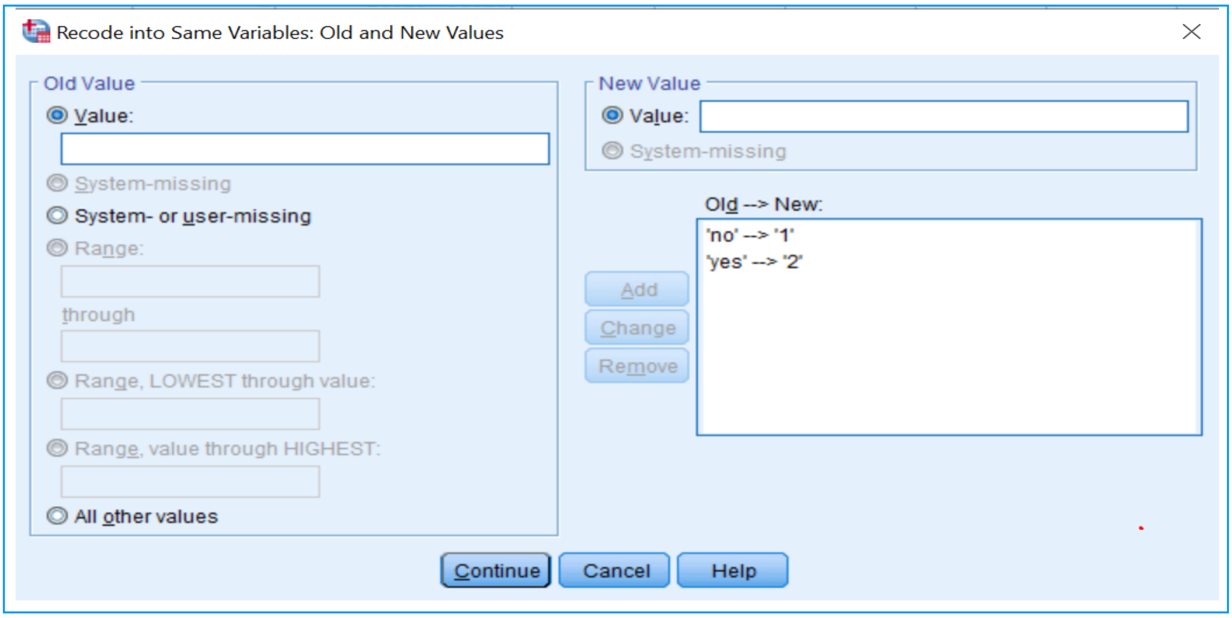

Figure 20 Change Variable Smoker to Number

Figure 21 below is shown the Encode result.

```
RECODE sex ('Female'='1') ('Male'='2').
EXECUTE.
RECODE smoker ('no'='1') ('yes'='2').
EXECUTE.
```

Figure 21 Encode Description

After encoding two variables, final step changed string variable to a numeric value (Figure 22).

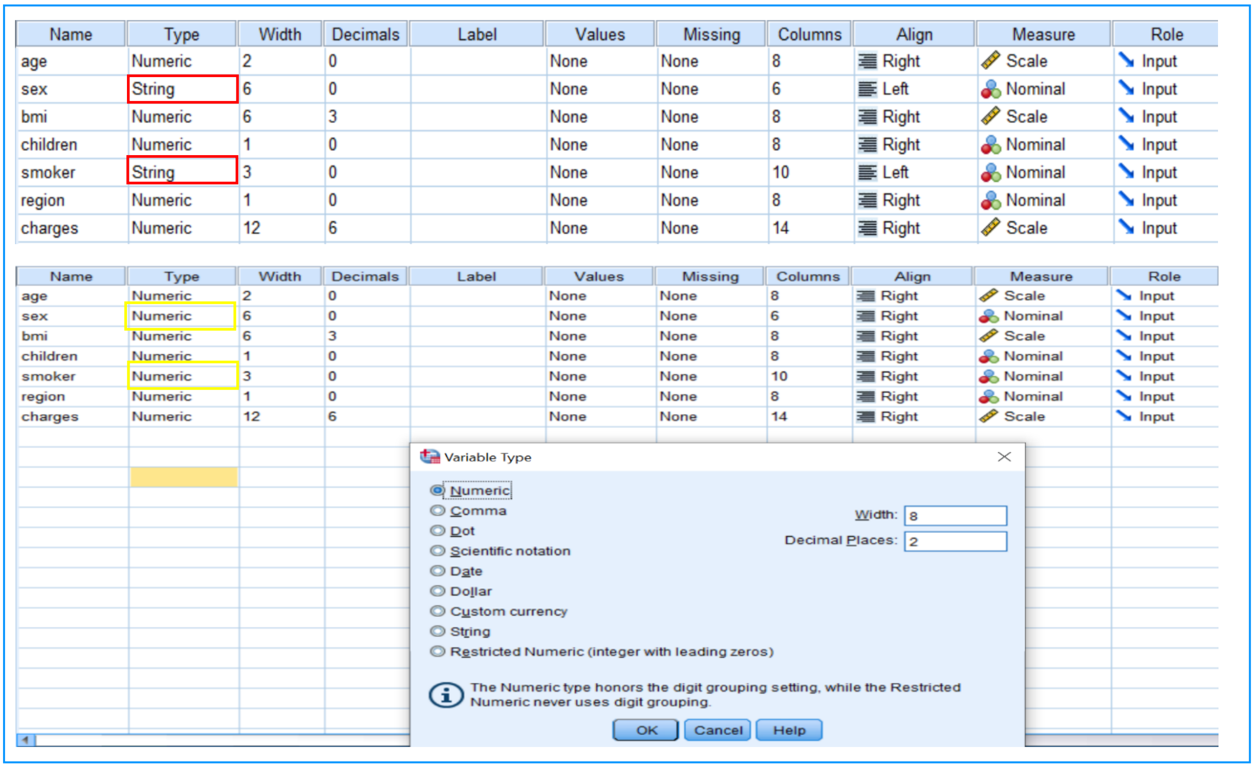

Figure 22 Change Data Type from String to Numeric

3.3 Tableau Data Imported

Figure 23 shows Tableau imported csv dataset for analyses and used visualisation for data presentation.

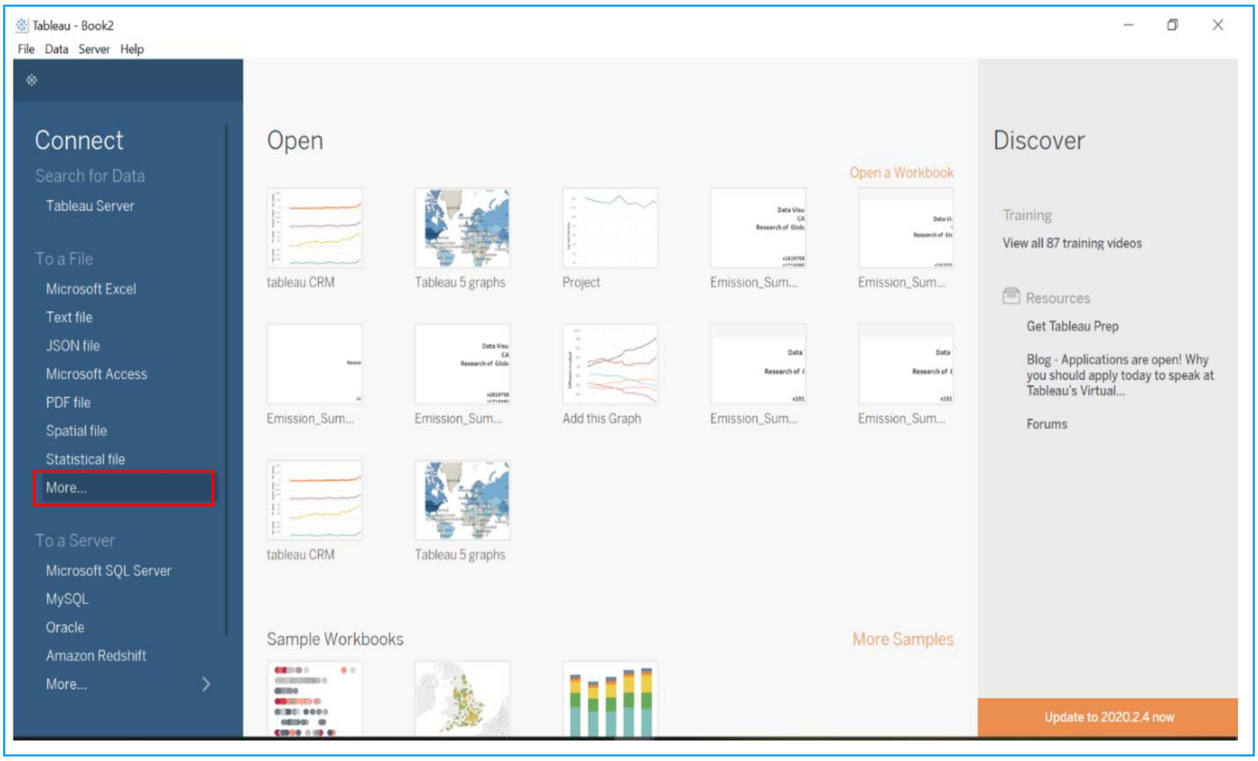

Figure 23 Tableau Imported Data

Figure 24 shows data were successfully imported into Tableau, then the change data type from string variable to a numeric variable, this is for analyses required.

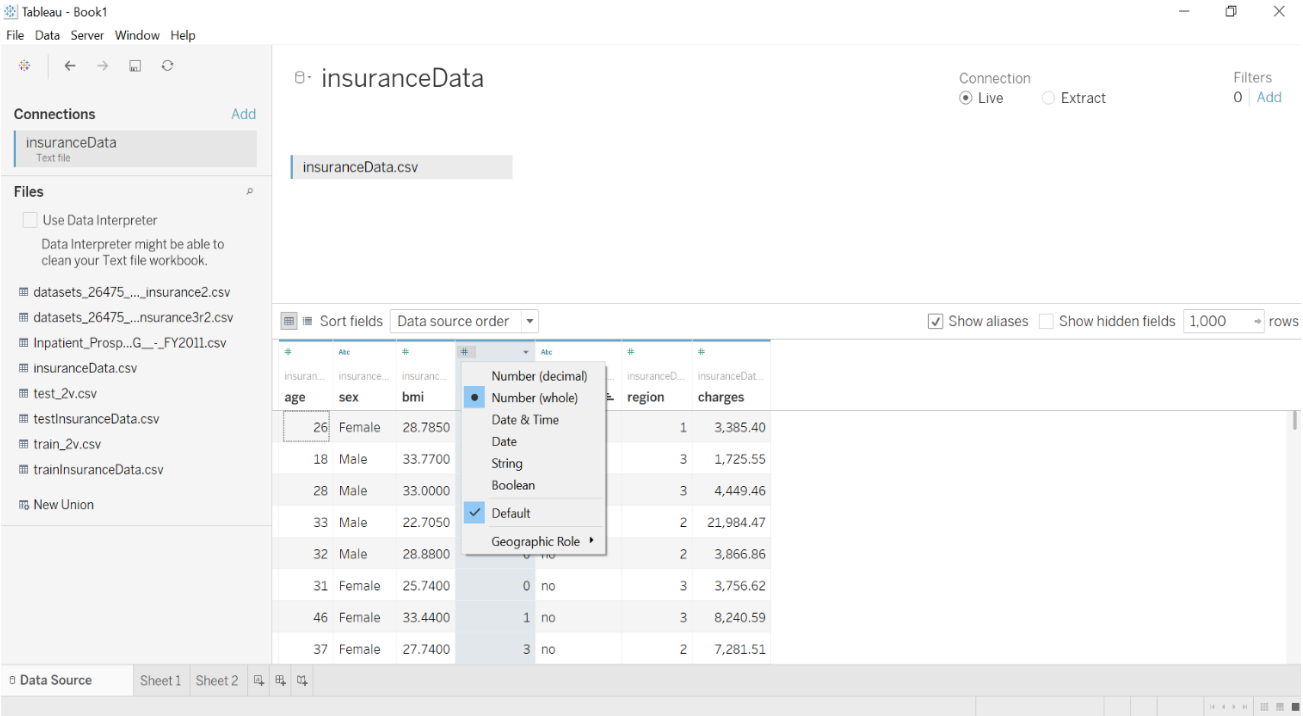

Figure 24 Dataset Preparation in Tableau

4 Implementation and Result Generated Steps

The implementation steps delivered into feature selection and machine learning model generation. There were 7 features selected and applied into 7 machine learning models in order to achieve the project objective. The features selected used RStudio generated Correlation Coefficient function and used SPSS generate a correlation table, both correlation result represent and evaluated the features was expected to conduct the implementation. After feature selection, there were several suitable libraries were installed in RStudio for generating the models, machine learning models used Multiple Linear Regression, Random Forest (Regression and Classification), Support Vector Machine (Regression and Classification), Naïve Bayes, Decision Tree, Logistic Regression and K-Nearest Neighbor to implemented, the output of models implementation used for evaluated by R-square (Regression model generated) and accuracy(Classification model generated).

4.1 RStudio – Models Generated

This project uses random report from code generated, the result of data might be varied as each time the output result from code generated was machine random selected value.

4.1.1 Multiple Linear Regression (Regression)

```
#------------------------Multiple Linear Regression---------------

#-Create LR Model
model_lm <- lm(charges ~., data = train)
summary(model_lm)

#-Create the prediction from the LR Model
pred <- predict(model_lm, newdata = test)

#-Create ggplot
ggplot(test, aes(x = pred, y = charges)) +
  geom_point(color = "blue", alpha = 0.7) +
  geom_abline(color = "red") +
  ggtitle("Prediction vs. Real values")
```

```
> #-Create LR Model
> model_lm <- lm(charges ~., data = train)
> summary(model_lm)

Call:
lm(formula = charges ~ ., data = train)

Residuals:
   Min     1Q Median     3Q    Max
-37213  -8301   -445   7755  41557

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     -31034.79     943.92 -32.879  < 2e-16 ***
age                207.35      12.07  17.186  < 2e-16 ***
sexMale          -2106.29     370.90  -5.679 1.44e-08 ***
bmi               1292.33      21.70  59.558  < 2e-16 ***
children          1436.91     113.55  12.654  < 2e-16 ***
smokeryes        10412.95     380.94  27.335  < 2e-16 ***
regionnorthwest   -791.47     507.95  -1.558    0.119
regionsoutheast  -1957.82     499.75  -3.918 9.08e-05 ***
regionsouthwest   -910.20     504.03  -1.806    0.071 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11860 on 4475 degrees of freedom
Multiple R-squared:  0.6453,    Adjusted R-squared:  0.6447
F-statistic:  1018 on 8 and 4475 DF,  p-value: < 2.2e-16
```

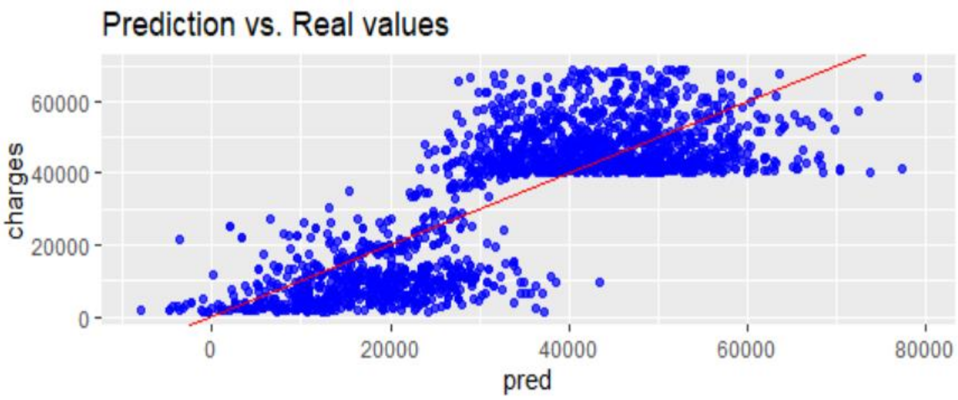

Figure 25 Multiple Linear Regression Model

Multiple Linear Regression model generated shows in figure 25, the result shows the R-square value and P-value, and the linear ggplot shows the model visualisation.

4.1.2 Random Forest (Regression)

```
#-------------------------Random Forest-------------------------

#-Create RF Model
model_rf <- randomForest(charges ~ ., data = train)
print(model_rf)
importance(model_rf)

#-Create the prediction from the RF Model
pred = predict(model_rf, newdata=test[-7])

#https://www.datatechnotes.com/2018/10/lasso-least-absolute-shrinkage-and.html
#-Show the Accuracy Checks
mse = mean(test$charges^2)
mae = MAE(test$charges, pred)
rmse = RMSE(test$charges, pred)
r2 = R2(test$charges, pred, form = "traditional")
cat(" MAE:", mae, "\n",  "MSE:", mse, "\n",
    "RMSE:", rmse, "\n", "R-squared:", r2)

#https://rstudio-pubs-static.s3.amazonaws.com/22159_77fe3f8ed4cb4ab7b3647a8d5c628ca4.html
#-Create Variable Importance & Error vs No. of trees Plots
par(mfrow=c(1,2))
varImpPlot(model_rf,main='Variable Importance Plot', pch=16,col='blue')
plot(model_rf,main='Error vs No. of trees plot', col='blue')
```

```
> print(model_rf)

Call:
 randomForest(formula = charges ~ ., data = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 58769461
                    % Var explained: 85.16
> importance(model_rf)
         IncNodePurity
age       1.363009e+11
sex       2.003386e+10
bmi       1.094058e+12
children  1.041816e+11
smoker    2.065845e+11
region    2.271364e+10
> #-Create the prediction from the RF Model
> pred = predict(model_rf, newdata=test[-7])
> #-Show the Accuracy Checks
> mse = mean(test$charges^2)
> mae = MAE(test$charges, pred)
> rmse = RMSE(test$charges, pred)
> r2 = R2(test$charges, pred, form = "traditional")
> cat(" MAE:", mae, "\n",  "MSE:", mse, "\n",
+     "RMSE:", rmse, "\n", "R-squared:", r2)
 MAE: 5229.505
 MSE: 1548509550
 RMSE: 7774.7
 R-squared: 0.7986904
```

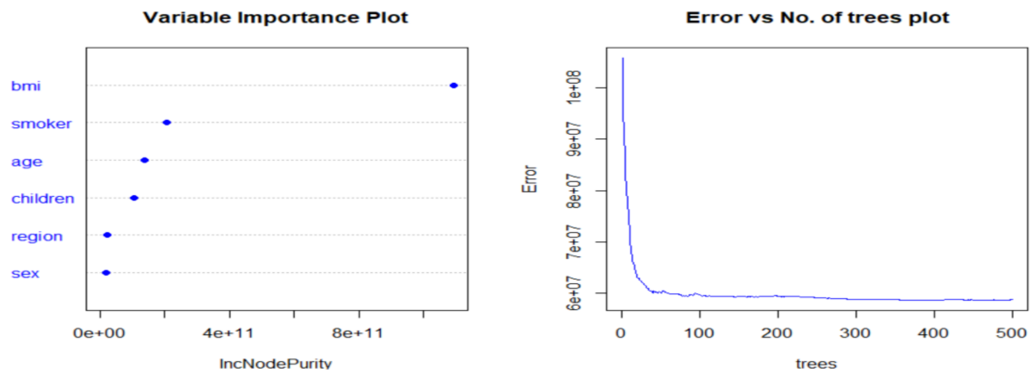

Figure 26 Random Forest Regression Model

Figure 26 shows the Random Forest Regression Model, which used charge as dependent variable generated R-square value for comparison with other models. Source code for accuracy generated was from DataTechNotes (Website, 2018), and source code for plot chart was by RStudio-pubs-static (Website, n.d.).

4.1.3 Support Vector Machine (Regression)

```
#-----------------------Support Vector Machine--------------------

#-Create SVM Model
svr.model <- svm(charges ~., data = train, type = "eps-regression",kernel = "radial")

#-Create the prediction from the SVM model
pred = predict(svr.model, test)

#https://www.datatechnotes.com/2018/10/lasso-least-absolute-shrinkage-and.html
#-Show the Accuracy Checks
mse = mean(test$charges^2)
mae = MAE(test$charges, pred)
rmse = RMSE(test$charges, pred)
r2 = R2(test$charges, pred, form = "traditional")
cat(" MAE:", mae, "\n",  "MSE:", mse, "\n",
    "RMSE:", rmse, "\n", "R-squared:", r2)

#https://www.datatechnotes.com/2019/09/support-vector-regression-example-with.html
#-Create Prediction Charges VS Real Charges
x = 1:length(test$charges)
plot(x, test$charges, main= "Prediction Charges VS Real Charges", pch=18, col="red")
lines(x, pred, lwd="1", col="blue")
```

```
> #-Create the prediction from the SVM model
> pred = predict(svr.model, test)
> #-Show the Accuracy Checks
> mse = mean(test$charges^2)
> mae = MAE(test$charges, pred)
> rmse = RMSE(test$charges, pred)
> r2 = R2(test$charges, pred, form = "traditional")
> cat(" MAE:", mae, "\n",  "MSE:", mse, "\n",
+     "RMSE:", rmse, "\n", "R-squared:", r2)
 MAE: 6896.876
 MSE: 1548509550
 RMSE: 9712.771
 R-squared: 0.6733704
```

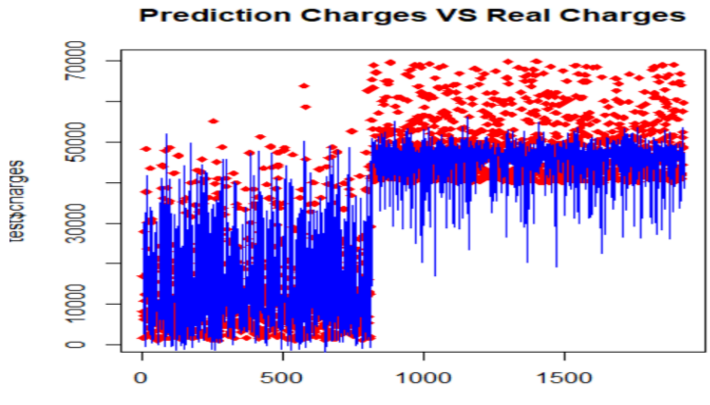

Figure 27 Support Vector Machine Regression Model

Support Vector Machine Regression Model generated in figure 27, it is generated R-square result for comparison with other models. The source code for accuracy generated was from DataTechNotes (Website, 2018), the source code for plot generated was by DataTechNotes (Website, 2019).

4.1.4 Naïve Bayes (Classification)

```
#------------------------------Naive Bayes----------------------

#-Create NB Model
model_nb <- naiveBayes(smoker ~ ., data = train)

#-Create the prediction from the NB Model
pred1 <- predict(model_nb, test, type="class")

#Create Confusion Matrix
confusionMatrix(pred1, test$smoker)
```

```
> model_nb <- naiveBayes(smoker ~ ., data = train)
> #-Create the prediction from the NB Model
> pred1 <- predict(model_nb, test, type="class")
> #Create Confusion Matrix
> confusionMatrix(pred1, test$smoker)
Confusion Matrix and Statistics

          Reference
Prediction  no yes
       no  654 164
       yes 543 561

               Accuracy : 0.6322
                 95% CI : (0.6101, 0.6538)
    No Information Rate : 0.6228
    P-Value [Acc > NIR] : 0.2053

                  Kappa : 0.2902

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.5464
            Specificity : 0.7738
         Pos Pred Value : 0.7995
         Neg Pred Value : 0.5082
             Prevalence : 0.6228
         Detection Rate : 0.3403
   Detection Prevalence : 0.4256
      Balanced Accuracy : 0.6601

       'Positive' Class : no
```

Figure 28 Naive Bayes Classification Model

Figure 28 shows the Naïve Bayes model generated and result represented accuracy and P-value compared other models.

4.1.5 Decision Tree (Classification)

```
#------------------------Decision Tree----------------------------

#-Create DT Model
model_dt  <- rpart(smoker ~ .,data=train,method = "class",parms = list(prior = c(0.3, 0.7)))

#-Create the prediction from the DT model
pred <- predict(model_dt,test,type="class")

#Create Confusion Matrix
confusionMatrix(pred,test$smoker)

#-Create Conditional Inference Tree
output.tree <- ctree(smoker ~ ., data = train)
plot(output.tree, main="Conditional Inference Tree with Insurance Data", gp = gpar(fontsize = 6))
```

```
> model_dt  <- rpart(smoker ~ .,data=train,method = "class",parms = list(prior = c(0.
3, 0.7)))
> #-Create the prediction from the DT model
> pred <- predict(model_dt,test,type="class")
> #Create Confusion Matrix
> confusionMatrix(pred,test$smoker)
Confusion Matrix and Statistics

          Reference
Prediction  no yes
       no  574   0
       yes 623 725

               Accuracy : 0.6759
                 95% CI : (0.6544, 0.6968)
    No Information Rate : 0.6228
    P-Value [Acc > NIR] : 7.016e-07

                  Kappa : 0.4101

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.4795
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.5378
             Prevalence : 0.6228
         Detection Rate : 0.2986
   Detection Prevalence : 0.2986
      Balanced Accuracy : 0.7398

       'Positive' Class : no
```

Figure 29 Decision Tree Classification Model

Figure 29 shows the Decision Tree model generation, which includes accuracy and P-value. Next figure 30 is the tree graphs represent the model visualisation.

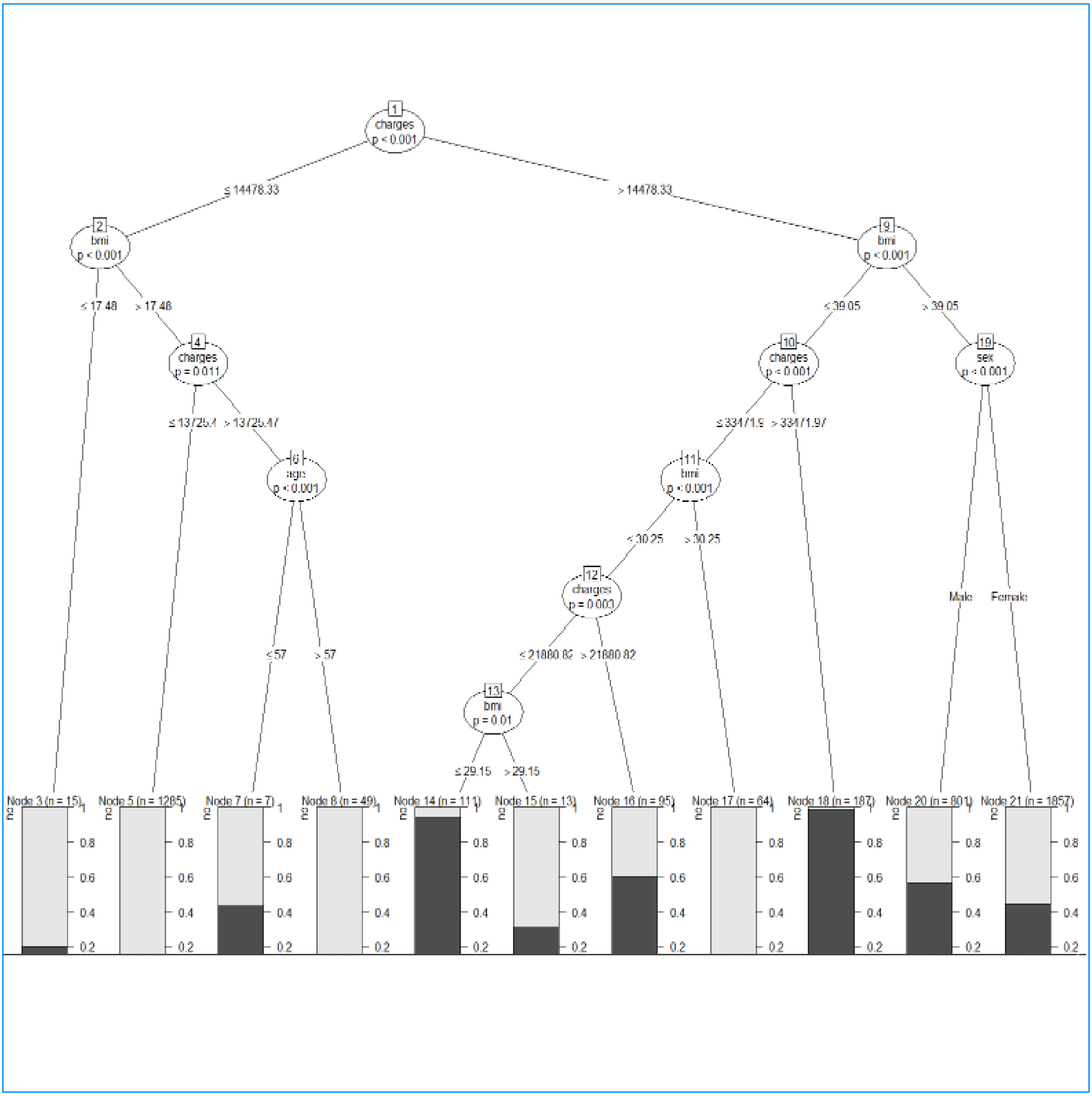

Figure 30 Decision Tree Graph

Decision Tree graph (Figure 30) shows smoker behaviour insurers paid higher health insurance premium than non-smoker behaviour insurers.

4.1.6 Logistic Regression (Classification)

```r
#-----------------------Logistic Regression----------------------

#-Create General Linear Model
glmModel <- train(smoker ~ .,data = train, method="glm",family = "binomial")
summary(glmModel)
confusionMatrix(glmModel)

#-Create the prediction from the GLM
pred <- predict(glmModel,test)

#Create Confusion Matrix
confusionMatrix(pred, test$smoker)
```

```
> glmModel <- train(smoker ~ .,data = train, method="glm",family = "binomial")
> summary(glmModel)

Call:
NULL

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1662  -0.9302  -0.4427   1.0914   1.8904

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -1.322e+00  2.142e-01  -6.171 6.79e-10 ***
age             -8.470e-03  2.405e-03  -3.522 0.000428 ***
sexMale          5.050e-01  7.383e-02   6.840 7.93e-12 ***
bmi             -4.523e-02  5.879e-03  -7.695 1.42e-14 ***
children        -2.284e-02  2.179e-02  -1.048 0.294682
regionnorthwest  9.796e-02  9.956e-02   0.984 0.325178
regionsoutheast  1.706e-01  9.783e-02   1.744 0.081118 .
regionsouthwest -1.233e-02  9.944e-02  -0.124 0.901289
charges          7.396e-05  3.169e-06  23.338  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5851.5  on 4483  degrees of freedom
Residual deviance: 4882.8  on 4475  degrees of freedom
AIC: 4900.8

Number of Fisher Scoring iterations: 4
```

```
> confusionMatrix(glmModel)
Bootstrapped (25 reps) Confusion Matrix

(entries are percentual average cell counts across resamples)

          Reference
Prediction   no  yes
       no  51.2 20.7
       yes 12.9 15.3

 Accuracy (average) : 0.6645

> #-Create the prediction from the GLM
> pred <- predict(glmModel,test)
> #Create Confusion Matrix
> confusionMatrix(pred, test$smoker)
Confusion Matrix and Statistics

          Reference
Prediction  no yes
       no  959 408
       yes 238 317

               Accuracy : 0.6639
                 95% CI : (0.6423, 0.685)
    No Information Rate : 0.6228
    P-Value [Acc > NIR] : 9.912e-05

                  Kappa : 0.25

 Mcnemar's Test P-Value : 2.947e-11

            Sensitivity : 0.8012
            Specificity : 0.4372
         Pos Pred Value : 0.7015
         Neg Pred Value : 0.5712
             Prevalence : 0.6228
         Detection Rate : 0.4990
   Detection Prevalence : 0.7112
      Balanced Accuracy : 0.6192

       'Positive' Class : no
```

Figure 31 Logistic Regression Classification Model

Figure 31 shows the Logistic Regression model generated used confusion matrix and summary to illustrate the model. The accuracy and P-value were used for compression result.

4.1.7 K-Nearest Neighbour (Classification)

```
#------------------------K-Nearest Neighbour------------------------

#-Store Smoker label Value
train_label <- train[,5]
test_label <- test[,5]

#-Set Factors data types to Nummeric types
train$smoker<-as.numeric(train$smoker)
test$smoker<-as.numeric(test$smoker)
train$region<-as.numeric(train$region)
test$region<-as.numeric(test$region)
train$sex<-as.numeric(train$sex)
test$sex<-as.numeric(test$sex)

#https://www.analyticsvidhya.com/blog/2015/08/learning-concept-knn-algorithms-programming/
#-Create KNN Model
model_knn <- knn(train = train, test = test,cl = train_label, k = 5, prob = TRUE)

#-check the accuracy of the predicted values
CrossTable(x = test_label, y = model_knn)

#-Create table count of Prediction VS Actual
table <- table(model_knn, test_label,dnn=c("Prediction","Actual"))

#-Get table accuracy
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(table)


> model_knn <- knn(train = train, test = test,cl = train_label, k = 5, prob = TRUE)
> #-check the accuracy of the predicted values
> CrossTable(x = test_label, y = model_knn)


   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  1922


             | model_knn
  test_label |        no |       yes | Row Total |
-------------|-----------|-----------|-----------|
          no |       896 |       301 |      1197 |
             |    26.918 |    45.746 |           |
             |     0.749 |     0.251 |     0.623 |
             |     0.740 |     0.423 |           |
             |     0.466 |     0.157 |           |
-------------|-----------|-----------|-----------|
         yes |       314 |       411 |       725 |
             |    44.443 |    75.529 |           |
             |     0.433 |     0.567 |     0.377 |
             |     0.260 |     0.577 |           |
             |     0.163 |     0.214 |           |
-------------|-----------|-----------|-----------|
Column Total |      1210 |       712 |      1922 |
             |     0.630 |     0.370 |           |
-------------|-----------|-----------|-----------|


> #-Create table count of Prediction VS Actual
> table <- table(model_knn, test_label,dnn=c("Prediction","Actual"))
> #-Get table accuracy
> accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
> accuracy(table)
[1] 68.00208
```

Figure 32 K-Nearest Neighbor Classification Model

Figure 32 shows the K-Nearest Neighbour model generated using cross table to get accuracy for comparison. Source code was from a website (Choudhury, 2015).

4.1.8 Support Vector Machine (Classification)

```
#--------------Support Vector Machine Classification--------------

#-Set Nummeric data types to Factors types
train$smoker<-as.factor(train$smoker)
test$smoker<-as.factor(test$smoker)
train$region<-as.factor(train$region)
test$region<-as.factor(test$region)
train$sex<-as.factor(train$sex)
test$sex<-as.factor(test$sex)

#-Create SVM Classification Model
model_svmc <- svm(smoker ~ . , train)

#-Create the prediction from the SVMC
pred <- predict(model_svmc, test, type="class")

#Create Confusion Matrix
confusionMatrix(pred, test$smoker)
```

```
> model_svmc <- svm(smoker ~ . , train)
> #-Create the prediction from the SVMC
> pred <- predict(model_svmc, test, type="class")
> #Create Confusion Matrix
> confusionMatrix(pred, test$smoker)
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1005  353
         2  192  372

               Accuracy : 0.7164
                 95% CI : (0.6957, 0.7365)
    No Information Rate : 0.6228
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3689

 Mcnemar's Test P-Value : 7.199e-12

            Sensitivity : 0.8396
            Specificity : 0.5131
         Pos Pred Value : 0.7401
         Neg Pred Value : 0.6596
             Prevalence : 0.6228
         Detection Rate : 0.5229
   Detection Prevalence : 0.7066
      Balanced Accuracy : 0.6764

       'Positive' Class : 1
```

Figure 33 Support Vector Machine Classification Model

Figure 33 shows the Support Vector Machine model generated used confusion matrix to get output from accuracy and P-value for comparison.

4.1.9 Random Forest (Classification)

```
#-----------------Random Forest Classification--------------------

#-Create RF Classification Model
model_rfc <- randomForest(smoker ~ ., data = train)

#-Create the prediction from the RFC
pred = predict(model_rfc, newdata=test[-5])

#Create Confusion Matrix
confusionMatrix(pred, test$smoker)
```

```
> model_rfc <- randomForest(smoker ~ ., data = train)
> #-Create the prediction from the RFC
> pred = predict(model_rfc, newdata=test[-5])
> #Create Confusion Matrix
> confusionMatrix(pred, test$smoker)
Confusion Matrix and Statistics

          Reference
Prediction   1   2
         1 950 314
         2 247 411

               Accuracy : 0.7081
                 95% CI : (0.6872, 0.7284)
    No Information Rate : 0.6228
    P-Value [Acc > NIR] : 2.389e-15

                  Kappa : 0.3672

 Mcnemar's Test P-Value : 0.005328

            Sensitivity : 0.7937
            Specificity : 0.5669
         Pos Pred Value : 0.7516
         Neg Pred Value : 0.6246
             Prevalence : 0.6228
         Detection Rate : 0.4943
   Detection Prevalence : 0.6576
      Balanced Accuracy : 0.6803

       'Positive' Class : 1
```

Figure 34 Random Forest Classification Model

Figure 34 shows Random Forest model generation, it used confusion matrix generated accuracy and P-value for comparison.

4.2 SPSS – Analysis & Models Generated

4.2.1 Correlation Table Generated

Click SPSS-Analyze-Correlate-Bivariate (Figure 35).

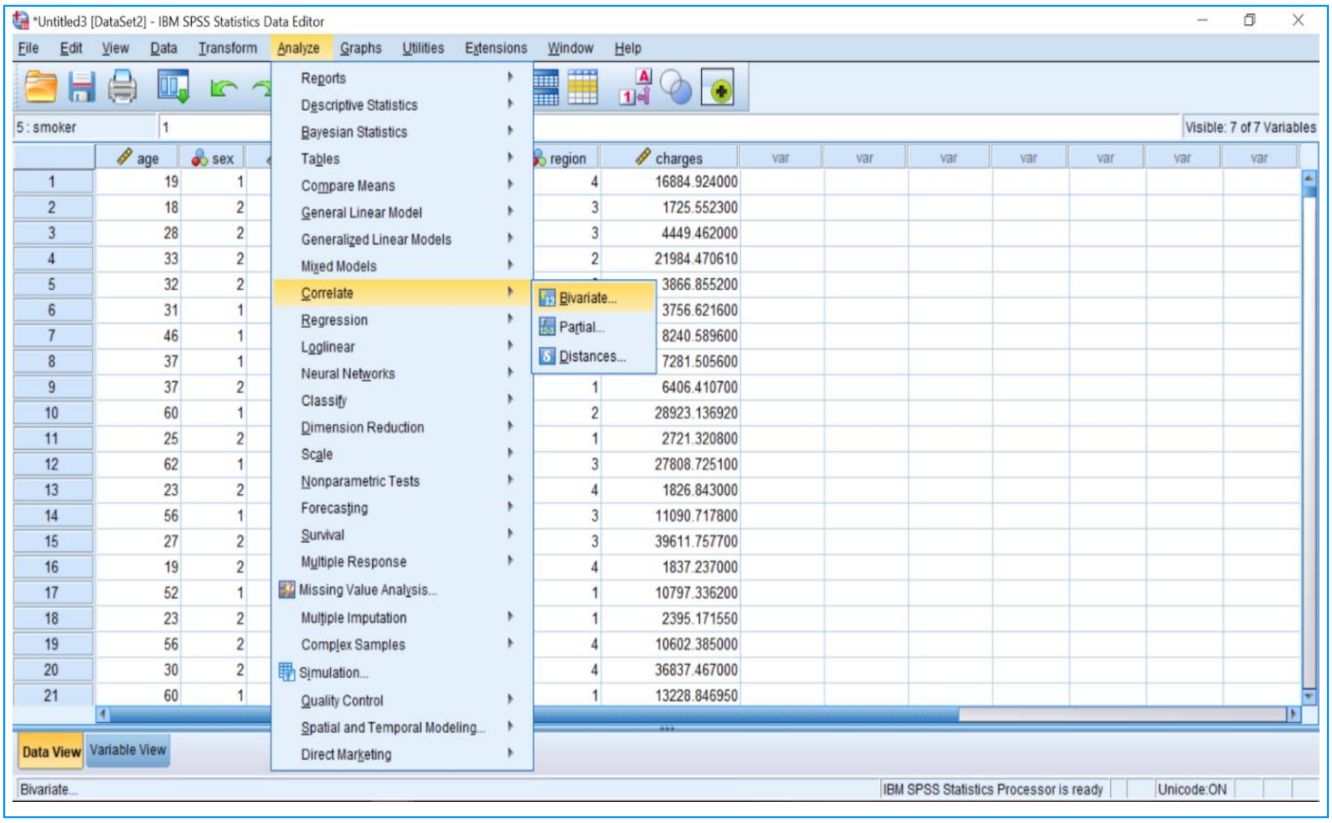

Figure 35 Generated Correlation

After the previous step, this step selected all variables into Pearson for correlation analyses (Figure 36).

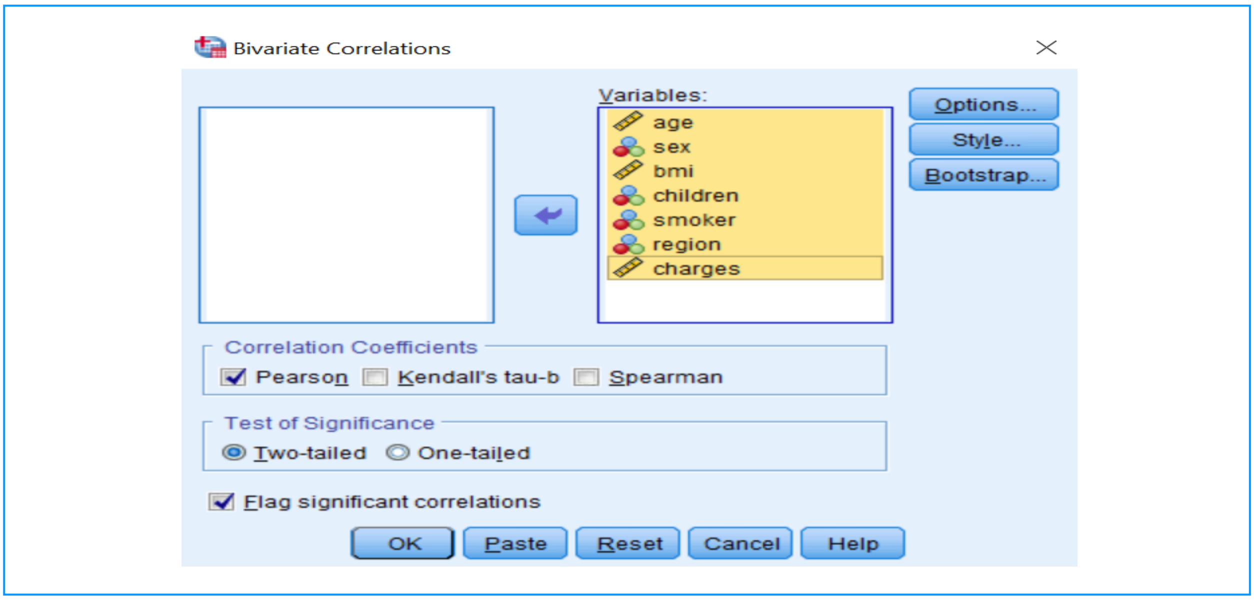

Figure 36 Select Variables

The correlation table generated show figure 37.

Correlations

		age	sex	bmi	children	smoker	region	charges
age	Pearson Correlation	1	-.062**	.235**	.132**	.082**	-.001	.333**
	Sig. (2-tailed)		.000	.000	.000	.000	.908	.000
	N	6406	6406	6406	6406	6406	6406	6406
sex	Pearson Correlation	-.062**	1	-.165**	-.080**	.029*	-.015	-.169**
	Sig. (2-tailed)	.000		.000	.000	.022	.224	.000
	N	6406	6406	6406	6406	6406	6406	6406
bmi	Pearson Correlation	.235**	-.165**	1	.333**	.219**	.031*	.730**
	Sig. (2-tailed)	.000	.000		.000	.000	.014	.000
	N	6406	6406	6406	6406	6406	6406	6406
children	Pearson Correlation	.132**	-.080**	.333**	1	.110**	.001	.369**
	Sig. (2-tailed)	.000	.000	.000		.000	.907	.000
	N	6406	6406	6406	6406	6406	6406	6406
smoker	Pearson Correlation	.082**	.029*	.219**	.110**	1	.008	.412**
	Sig. (2-tailed)	.000	.022	.000	.000		.504	.000
	N	6406	6406	6406	6406	6406	6406	6406
region	Pearson Correlation	-.001	-.015	.031*	.001	.008	1	-.005
	Sig. (2-tailed)	.908	.224	.014	.907	.504		.674
	N	6406	6406	6406	6406	6406	6406	6406
charges	Pearson Correlation	.333**	-.169**	.730**	.369**	.412**	-.005	1
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.674	
	N	6406	6406	6406	6406	6406	6406	6406

Figure 37 Correlation Table

4.2.2 ANOVE Model Generated

SPSS-Analyze-Regression-Linear (Figure 38).

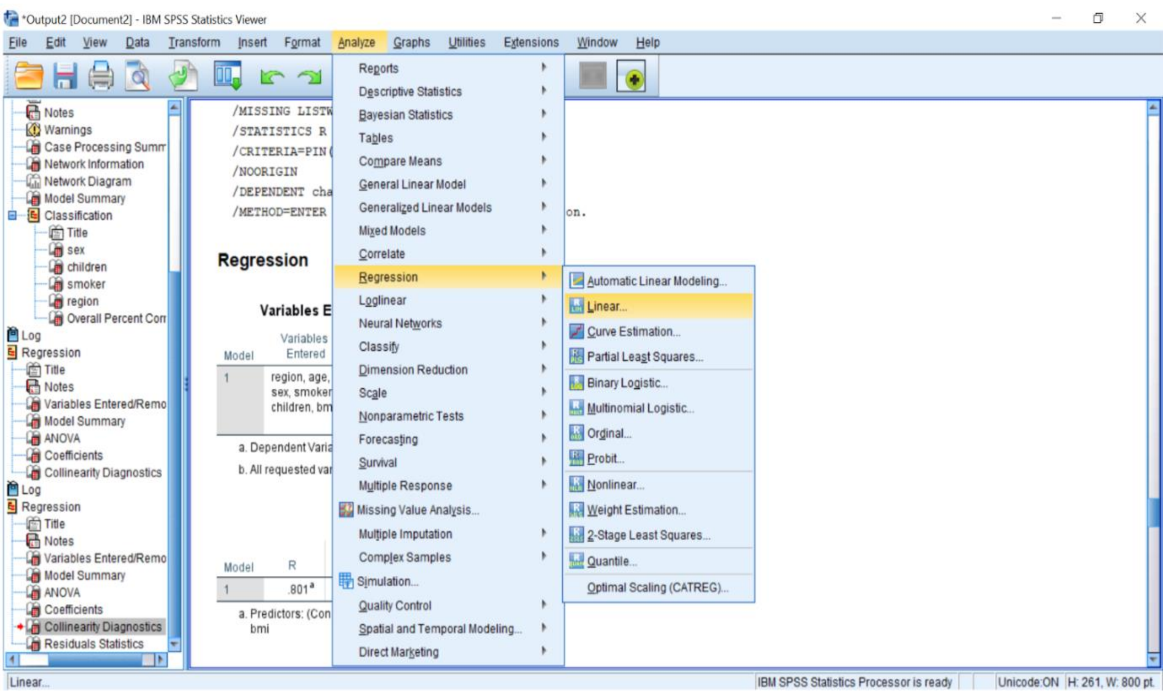

Figure 38 ANOVA Generated

This step input all the variables in dependent and independent block (Figure 39).

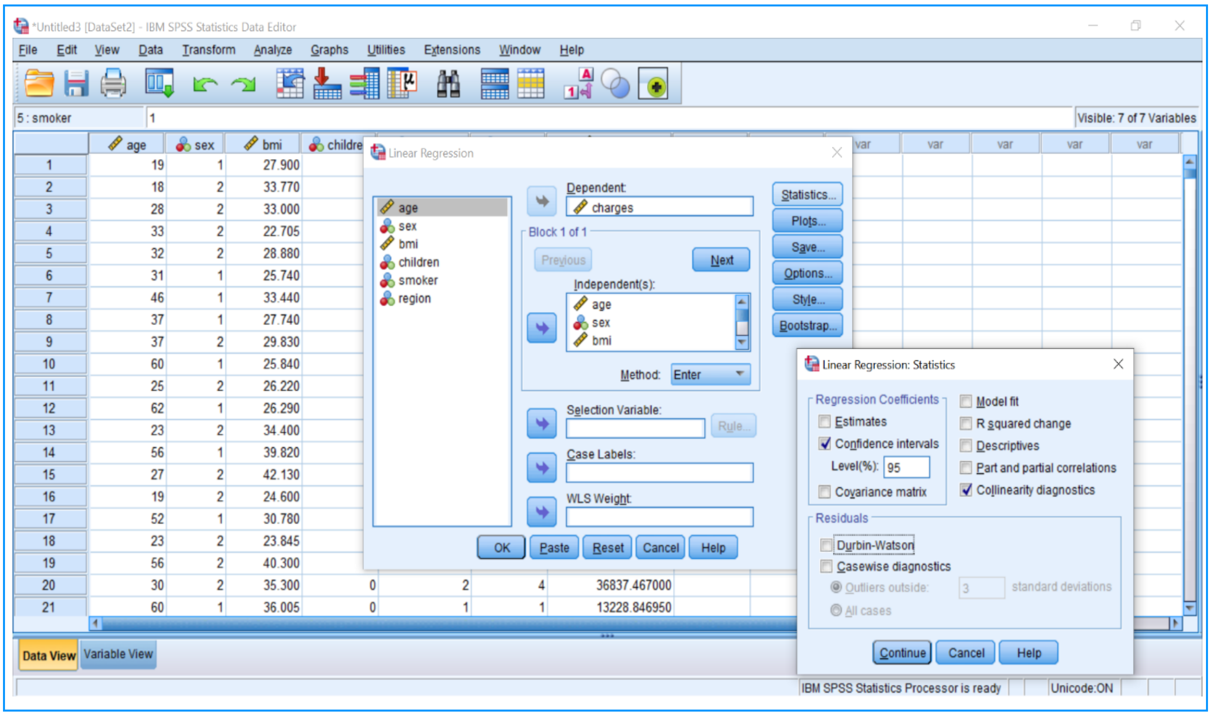

Figure 39 Select Variables

ANOVA model generated with model summary (Figure 40). R-Square value and sig (P-value) are used to identify the model performance for this project.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.801[a]	.642	.641	11950.46168	1.724

a. Predictors: (Constant), region, age, sex, smoker, children, bmi

b. Dependent Variable: charges

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1.636E+12	6	2.726E+11	1909.060	.000[b]
	Residual	9.139E+11	6399	142813534.4		
	Total	2.550E+12	6405			

a. Dependent Variable: charges

b. Predictors: (Constant), region, age, sex, smoker, children, bmi

Figure 40 ANOVA Model

4.3 Tableau Visualisation Generated

Figure 41 selected variables in column and rows, this graph used Sex, Smoker and Charges variables demonstrate the female user charges more insurance premium than male user. Also smoker users paid higher health insurance premium than non-smoker users.

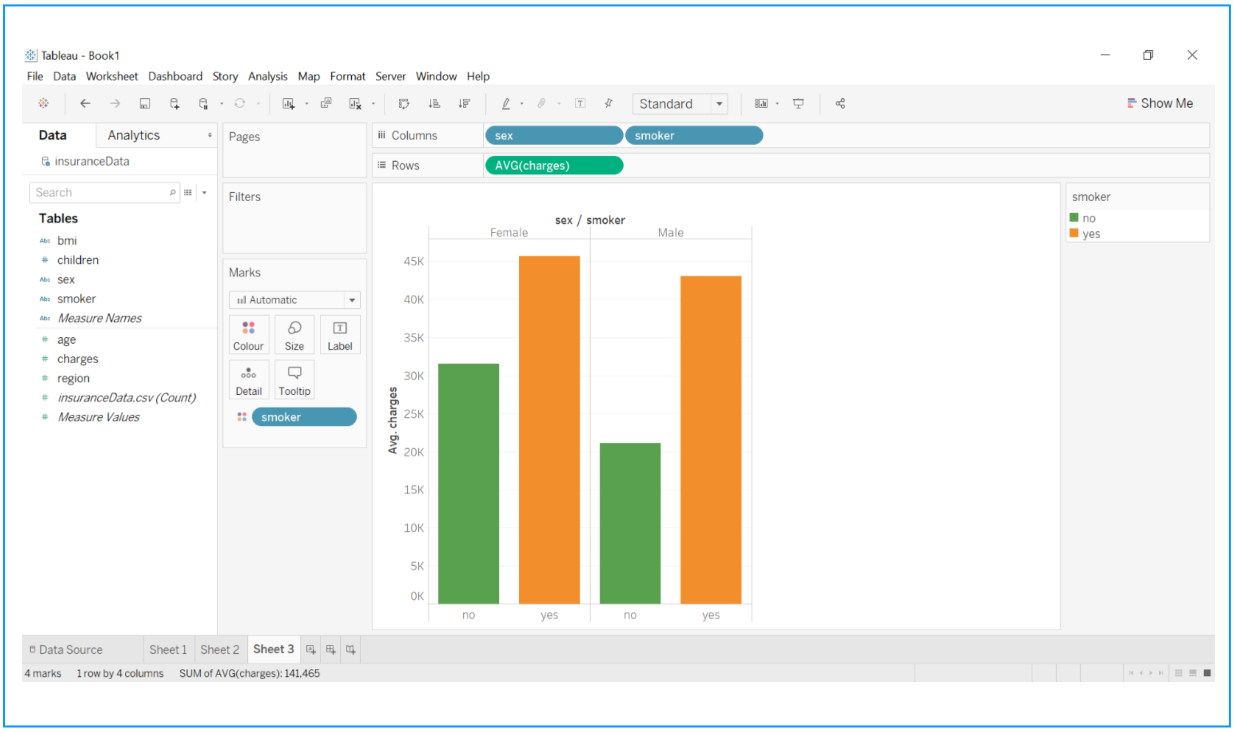

Figure 41 Charge Distribution with Smoker, Sex

The second visualisation (Figure 42) used Children and Charges variables to show how users with different amount children would impact health insurance charge.

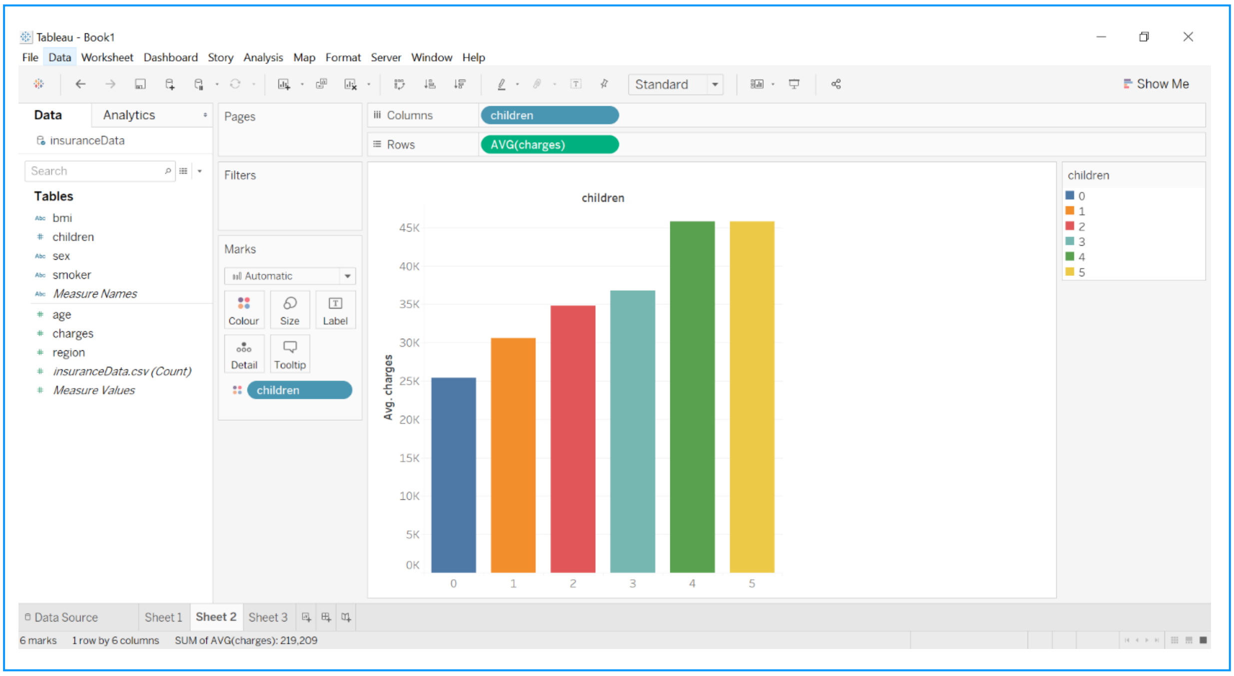

Figure 42 Charges Distribution with children

5 Appendix

There are some appendix works tried through this project to achieve the research and sub-research objectives.

Neutral Network generated from RStudio and SPSS.

5.1 Neutral Network Generated Used SPSS

Used SPSS to generate Neutral Network model (Figure 43)

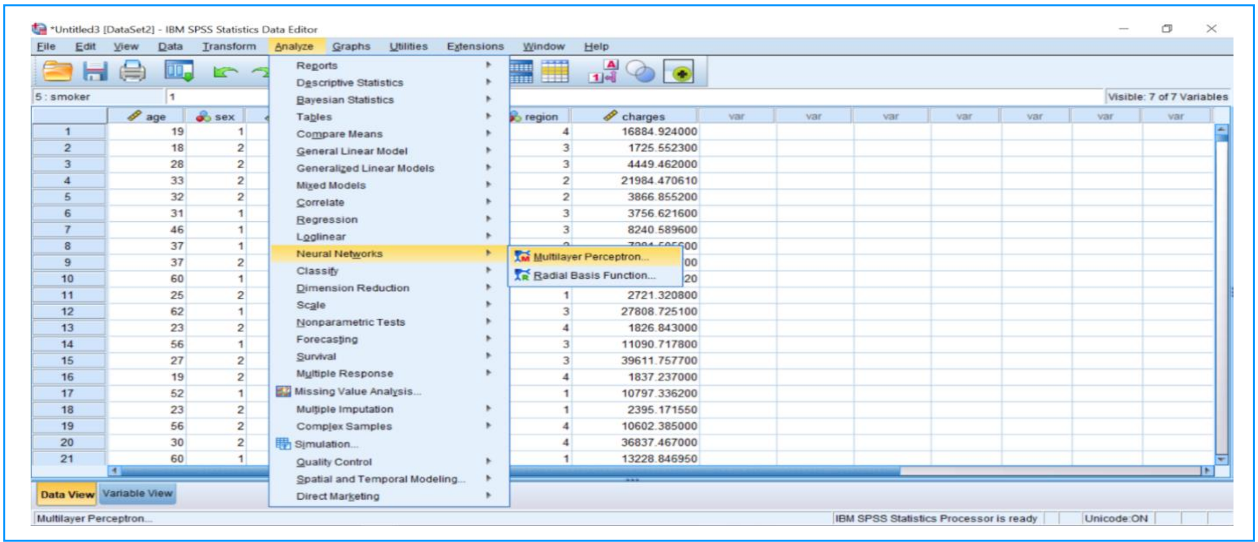

Figure 43 Neutral Network Model Generated

Figure 44 shows the Neutral Network Information of the model. Which including the output layer with six dependent variables and 1 input layer.

Network Information

Input Layer	Factors	1	charges
	Number of Units[a]		2600
Hidden Layer(s)	Number of Hidden Layers		1
	Number of Units in Hidden Layer 1[a]		1
	Activation Function		Hyperbolic tangent
Output Layer	Dependent Variables	1	age
		2	sex
		3	bmi
		4	children
		5	smoker
		6	region
	Number of Units		16
	Rescaling Method for Scale Dependents		Standardized
	Activation Function		Identity
	Error Function		Sum of Squares

a. Excluding the bias unit

Figure 44 Network Information

Figure 45 was the Neutral Network model summary and overall result.

Model Summary

Training	Sum of Squares Error		10036.234
	Average Overall Relative Error		1.000
	Percent Incorrect Predictions for Categorical Dependents	sex	38.5%
		children	72.1%
		smoker	36.7%
		region	73.3%
	Relative Error for Scale Dependents	age	1.000
		bmi	1.000
	Stopping Rule Used		1 consecutive step(s) with no decrease in error[a]
	Training Time		0:00:04.14
Testing	Sum of Squares Error		3087.418
	Average Overall Relative Error		1.001
	Percent Incorrect Predictions for Categorical Dependents	sex	38.2%
		children	72.6%
		smoker	36.7%
		region	72.3%
	Relative Error for Scale Dependents	age	1.000
		bmi	1.000

a. Error computations are based on the testing sample.

Overall Percent Correct

Sample	Overall Percent Correct
Training	44.9%
Testing	45.0%

Figure 45 Model Summary and Result

Figure 46 is Neutral Network visualisation.

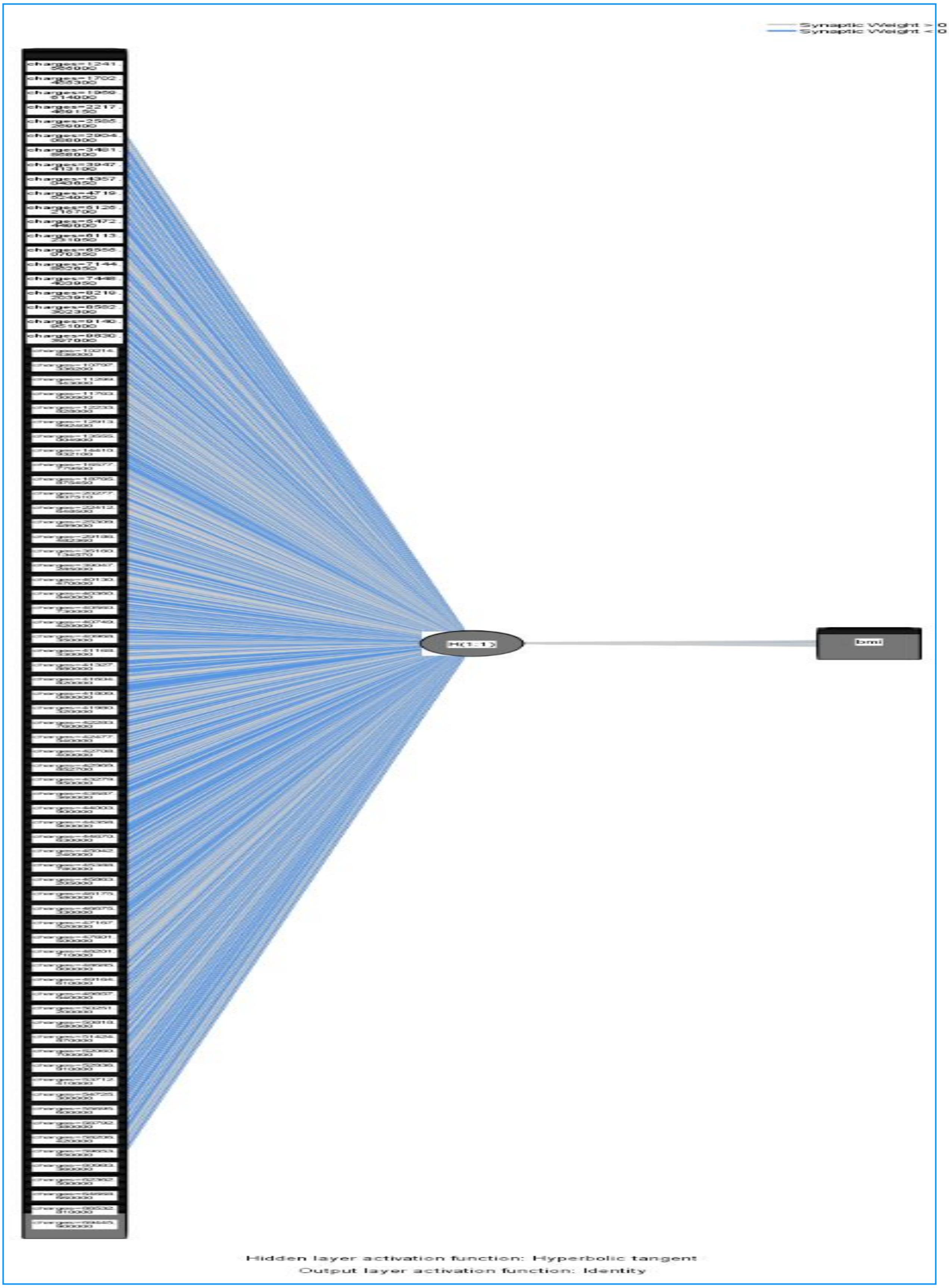

Figure 46 Neutral Network Model Graph

5.2 BMI value compared charge

The figure 47 used the bmi value to compare the charges, the health insurance cost related to high bmi value from the visualization represented.

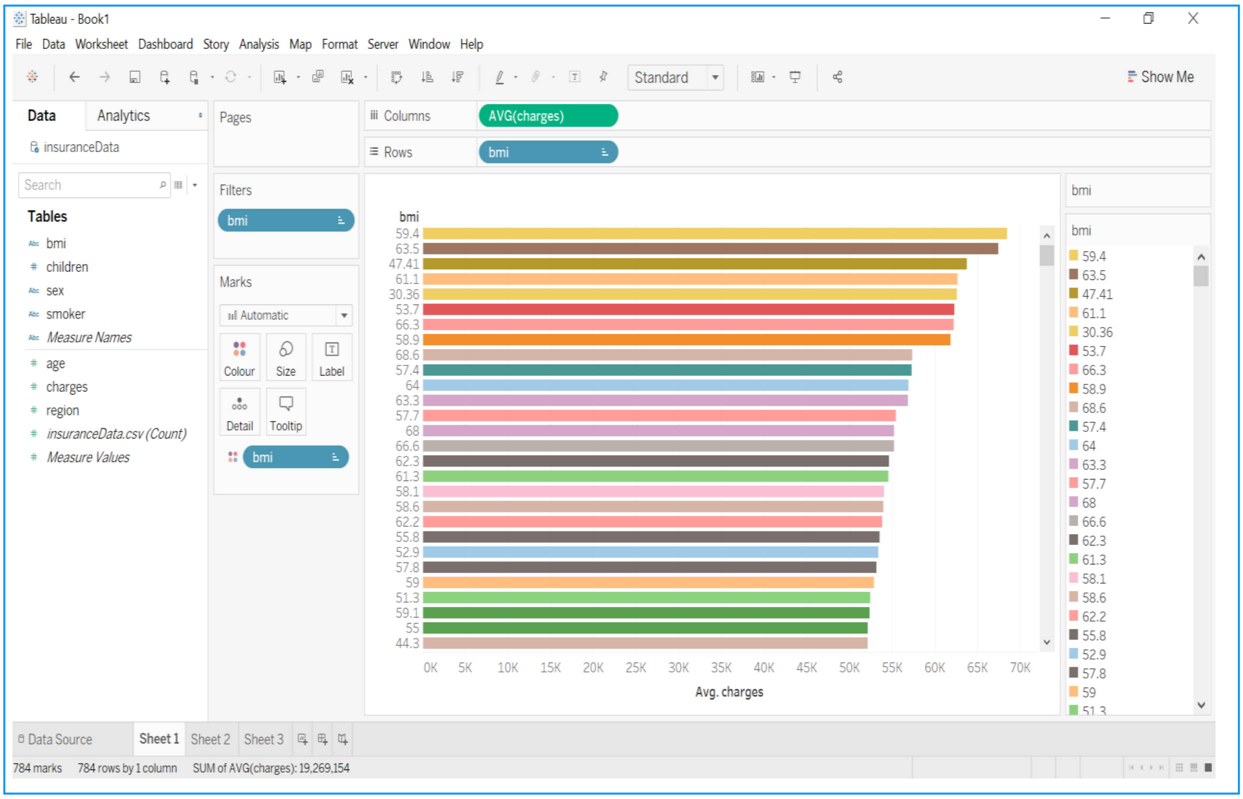

Figure 47 bmi with Charges

5.3 Regions compared Charges

Figure 48 shown the four regions health insurance charge distribution is quite consistent.

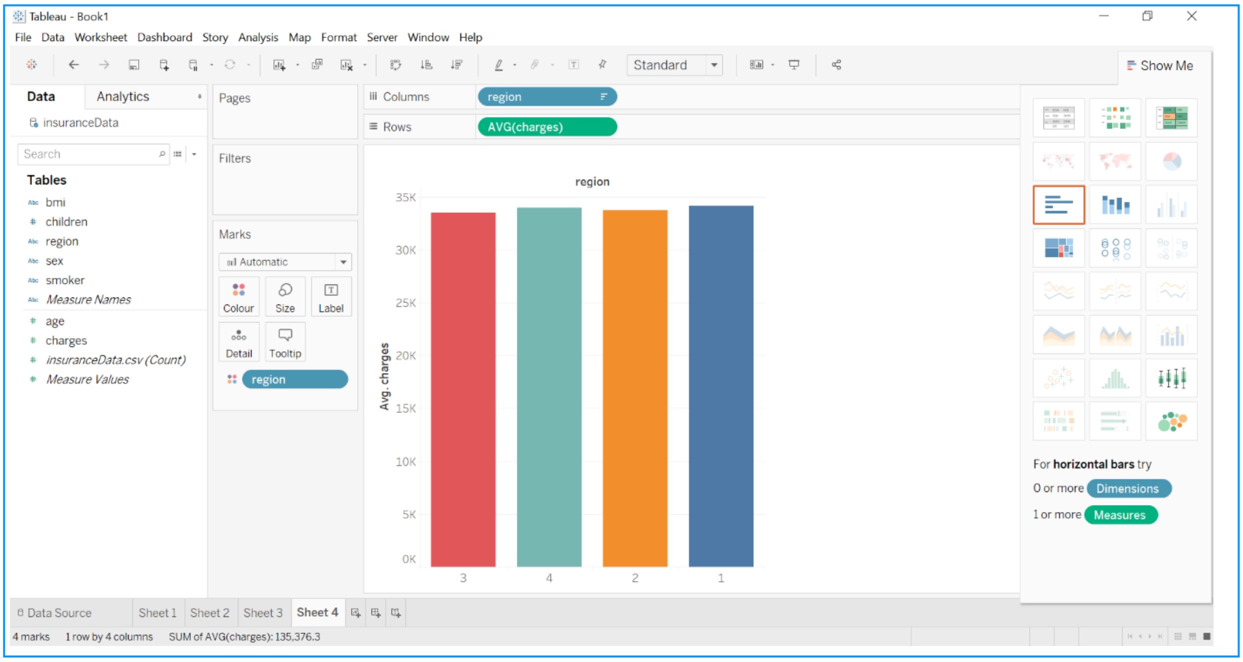

Figure 48 Region Compared Charges

References

Choudhury, P. R., 2015. *analyticsvidhya.* [Online]
Available at: https://www.analyticsvidhya.com/blog/2015/08/learning-concept-knn-algorithms-programming/

Website, 2018. *DataTechNotes.* [Online]
Available at: https://www.datatechnotes.com/2018/10/lasso-least-absolute-shrinkage-and.html

Website, 2019. [Online]
Available at: https://www.datatechnotes.com/2019/09/support-vector-regression-example-with.html

Website, n.d. *Random Forest : Walk Through.* [Online]
Available at: https://rstudio-pubs-static.s3.amazonaws.com/22159_77fe3f8ed4cb4ab7b3647a8d5c628ca4.html