Diabetes Analysis

Jonathan Flanagan (x18143890) & Neil Fitzgerald (x18149693)

Introduction

The purpose of this analysis is to see if certain health indicators can be used to predict the presence of diabetes in a person. Diabetes is categorized as serious chronic disease where individuals lose the ability to effectively regulate glucose levels in their blood, a result of this can be a reduced quality of life and/or life expectancy.

During the normal digestion process, foods are broken down into sugars and are released into the bloodstream. This in-turn, signals the pancreas to release insulin. Insulin helps cells within the body by enabling the use of sugars in the bloodstream as a form of energy. Diabetes is generally characterized by either the body not making enough insulin or not being able to use the insulin that is made effectively. Typically labeled Type 1 or Type 2 diabetes.

Health risks such as heart disease, vision loss, lower-limb amputation, and kidney disease are typically associated with sufferers of chronic diabetes. While there is no cure for diabetes, strategies like losing weight, eating healthily, being active, and receiving medical treatments can mitigate the harms of this disease in many patients.

Early diagnosis can lead to lifestyle changes and a more effective treatment. By making predictive models for diabetes risk, it can become one of many important tools for health officials.

Data

The data set for this analysis was downloaded from Kaggle link details about how this data set was originally collected are below:

"The Behavioral Risk Factor Surveillance System (BRFSS) is a health-related telephone survey that is collected annually by the CDC. Each year, the survey collects responses from over 400,000 Americans on health-related risk behaviors, chronic health conditions, and the use of preventative services. It has been conducted every year since 1984."

Also included is the *CDC Behavioral Risk Factor Surveillance System Codebook* which contains the column codes, questions asked and the possible responses.

The downloaded dataset has a 50/50 split of 70,692 diabetic and non-diabetic patients. Where diabetic is classified as being either type 1 or type 2 diabetic or being in a pre-diabetic condition, the the purposes of this analysis these three case types are labeled as diabetic.

A subset of 22 of the original 330 questions asked are used in this dataset and any answers where the respondent gave the answer *"Don't know"* or *"Refused to answer"* are removed.

Import Data & Initial Exploration

The data is imported from the downloaded csv. The columns are inspected, a basic summary is created for each column and the data types are viewed to see the initial shape and feature set of the data. A check for any missing values is also performed.

```
#----------------- IMPORT THE DATA FOR ANALYSIS

get_data <- function(){
  # import data
  data <- read.csv("./data/diabetes_binary_health_indicators.csv")
  # convert to dataframe
  data <- as.data.frame(data)

  return (data)
}

data <- get_data()

# print the number of rows and cols
dim(data)
```

```
## [1] 70692    22
```

The data set contains 70,692 rows (observations) from the CDC survey and 22 features. As this analysis will look at predicting the possibility of a diagnosis of diabetes, the **diabetes_binary** column will be a our dependent variable.

The first 5 rows of the data are viewed to get an idea of what is contained in the data set.

```
options(width = 100)

# print out first 6 rows and first 7 columns to fit on pdf output
head(data)
```

```
##   Diabetes_binary HighBP HighChol CholCheck BMI Smoker Stroke HeartDiseaseorAttack PhysActivity
## 1               0      1        0         1  26      0      0                    0            1
## 2               0      1        1         1  26      1      1                    0            0
## 3               0      0        0         1  26      0      0                    0            1
## 4               0      1        1         1  28      1      0                    0            1
## 5               0      0        0         1  29      1      0                    0            1
## 6               0      0        0         1  18      0      0                    0            1
##   Fruits Veggies HvyAlcoholConsump AnyHealthcare NoDocbcCost GenHlth MentHlth PhysHlth DiffWalk Sex
## 1      0       1                 0             1           0       3        5       30        0   1
## 2      1       0                 0             1           0       3        0        0        0   1
## 3      1       1                 0             1           0       1        0       10        0   1
## 4      1       1                 0             1           0       3        0        3        0   1
## 5      1       1                 0             1           0       2        0        0        0   0
## 6      1       1                 0             0           0       2        7        0        0   0
##   Age Education Income
## 1   4         6      8
## 2  12         6      8
## 3  13         6      8
## 4  11         6      8
## 5   8         5      8
## 6   1         4      7
```

All of the values appear to be numeric or numeric factors. This will be inspected further on for full clarifications. A summary of all the columns is then checked.

```r
options(width = 100)

# summmary of all rows
summary(data)
```

```
##  Diabetes_binary     HighBP          HighChol        CholCheck           BMI
##  Min.   :0.0     Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :12.00
##  1st Qu.:0.0     1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:25.00
##  Median :0.5     Median :1.0000   Median :1.0000   Median :1.0000   Median :29.00
##  Mean   :0.5     Mean   :0.5635   Mean   :0.5257   Mean   :0.9753   Mean   :29.86
##  3rd Qu.:1.0     3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:33.00
##  Max.   :1.0     Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :98.00
##      Smoker           Stroke        HeartDiseaseorAttack  PhysActivity       Fruits
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000       Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000       1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.00000   Median :0.0000       Median :1.000   Median :1.0000
##  Mean   :0.4753   Mean   :0.06217   Mean   :0.1478       Mean   :0.703   Mean   :0.6118
##  3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0.0000       3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.0000       Max.   :1.000   Max.   :1.0000
##     Veggies       HvyAlcoholConsump AnyHealthcare    NoDocbcCost          GenHlth
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.000   Min.   :0.00000   Min.   :1.000
##  1st Qu.:1.0000   1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.00000   1st Qu.:2.000
##  Median :1.0000   Median :0.00000   Median :1.000   Median :0.00000   Median :3.000
##  Mean   :0.7888   Mean   :0.04272   Mean   :0.955   Mean   :0.09391   Mean   :2.837
##  3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:1.000   3rd Qu.:0.00000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.000   Max.   :1.00000   Max.   :5.000
##     MentHlth         PhysHlth        DiffWalk           Sex              Age           Education
##  Min.   : 0.000   Min.   : 0.00   Min.   :0.0000   Min.   :0.000   Min.   : 1.000   Min.   :1.000
##  1st Qu.: 0.000   1st Qu.: 0.00   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.: 7.000   1st Qu.:4.000
##  Median : 0.000   Median : 0.00   Median :0.0000   Median :0.000   Median : 9.000   Median :5.000
##  Mean   : 3.752   Mean   : 5.81   Mean   :0.2527   Mean   :0.457   Mean   : 8.584   Mean   :4.921
##  3rd Qu.: 2.000   3rd Qu.: 6.00   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:11.000   3rd Qu.:6.000
##  Max.   :30.000   Max.   :30.00   Max.   :1.0000   Max.   :1.000   Max.   :13.000   Max.   :6.000
##      Income
##  Min.   :1.000
##  1st Qu.:4.000
##  Median :6.000
##  Mean   :5.698
##  3rd Qu.:8.000
##  Max.   :8.000
```

The summary shows very little initial information as most of the rows that should be factors are being read as numeric only.

We can view the data types to confirm this.

```
# view the data types
str(data)
```

```
## 'data.frame':    70692 obs. of  22 variables:
##  $ Diabetes_binary     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ HighBP              : num  1 1 0 1 0 0 0 0 0 0 ...
##  $ HighChol            : num  0 1 0 1 0 0 1 0 0 0 ...
##  $ CholCheck           : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ BMI                 : num  26 26 26 28 29 18 26 31 32 27 ...
##  $ Smoker              : num  0 1 0 1 1 0 1 1 0 1 ...
##  $ Stroke              : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ HeartDiseaseorAttack: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PhysActivity        : num  1 0 1 1 1 1 1 0 1 0 ...
##  $ Fruits              : num  0 1 1 1 1 1 1 1 1 1 ...
##  $ Veggies             : num  1 0 1 1 1 1 1 1 1 1 ...
##  $ HvyAlcoholConsump   : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ AnyHealthcare       : num  1 1 1 1 1 0 1 1 1 1 ...
##  $ NoDocbcCost         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ GenHlth             : num  3 3 1 3 2 2 1 4 3 3 ...
##  $ MentHlth            : num  5 0 0 0 0 7 0 0 0 0 ...
##  $ PhysHlth            : num  30 0 10 3 0 0 0 0 0 6 ...
##  $ DiffWalk            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sex                 : num  1 1 1 1 0 0 1 1 0 1 ...
##  $ Age                 : num  4 12 13 11 8 1 13 6 3 6 ...
##  $ Education           : num  6 6 6 6 5 4 5 4 6 4 ...
##  $ Income              : num  8 8 8 8 8 7 6 3 8 4 ...
```

A view of the data types for each column confirms that they have all been read in as numeric. A copy of the questionnaire and the meaning of each factor level has also been downloaded to check which question in the questionnaire each column header refers to and the possible answers that could have been given.

Note: **any answers which returned no information such as "don't know" or "prefer not to answer" were already removed to the data set prior to downloading**

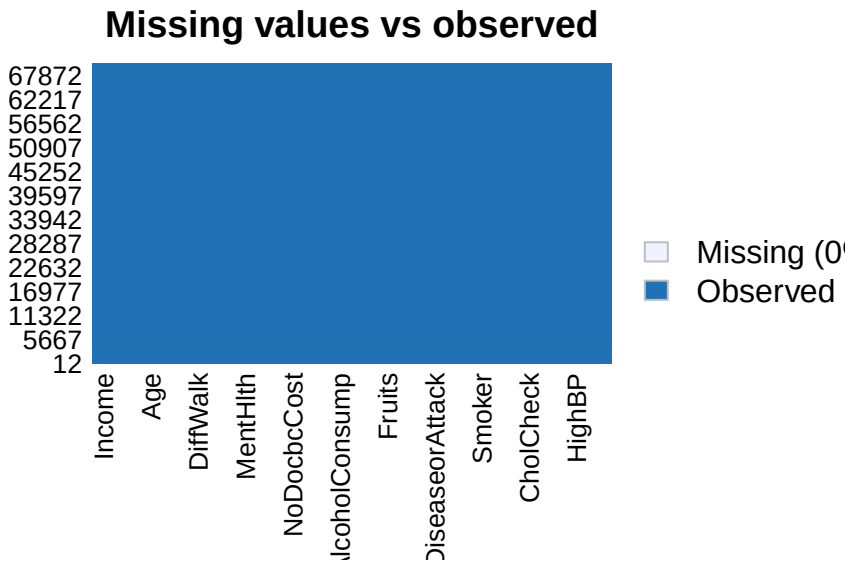
Next we can perform a boolean check to see if any columns contain NA values

```
# Check na's for each column
na_check <- mapply(anyNA, data)
na_check
```

```
##      Diabetes_binary               HighBP             HighChol            CholCheck
##                FALSE                FALSE                FALSE                FALSE
##                  BMI               Smoker               Stroke HeartDiseaseorAttack
##                FALSE                FALSE                FALSE                FALSE
##         PhysActivity               Fruits              Veggies    HvyAlcoholConsump
##                FALSE                FALSE                FALSE                FALSE
##        AnyHealthcare          NoDocbcCost              GenHlth             MentHlth
##                FALSE                FALSE                FALSE                FALSE
##             PhysHlth             DiffWalk                  Sex                  Age
##                FALSE                FALSE                FALSE                FALSE
##            Education               Income
##                FALSE                FALSE
```

The data can also be visually inspect using a data map for NaN's.

```
# visualize the missing data
missmap(data, main = "Missing values vs observed")
```



Data Types & Imputation

There are no signs of any missing data in any of the columns, but from exploring the data types and the data summary we can see some features that should be factors are imported as numbers.

The original survey conducted by the CDC has string value factors that are easier to read so we will change the factors back to their original string value for the time being to make a more human readable data set during the initial exploration.

Table of the Data Features, the original survey question and the data type.

Data Features	Survey Question	Data Type
Diabetes_binary	(Ever told) you have diabetes, Predictor Variable	BOOLEAN - (TRUE/FALSE)
HighBP	Adults who have been told they have high blood pressure by a doctor, nurse, or other health professional	BOOLEAN (TRUE/FALSE)
HighChol	Have you EVER been told by a doctor, nurse or other health professional that your blood cholesterol is high	BOOLEAN (TRUE/FALSE)
CholCheck	Cholesterol check within past five years	BOOLEAN (TRUE/FALSE)
BMI	Body Mass Index (BMI)	INT
Smoker	Have you smoked at least 100 cigarettes in your entire life?	FACTOR (Yes/No)
Stroke	(Ever told) you had a stroke	Factor (Yes/No)
HeartDiseaseorAttack	Have ever reported having coronary heart disease (CHD) or myocardial infarction (MI)	FACTOR (Yes/No)
PhysActivity	Adults who reported doing physical activity or exercise during the past 30 days other than their regular job	BOOLEAN (TRUE/FALSE)

Data Features	Survey Question	Data Type
Fruits	Consume Fruit 1 or more times per day	BOOLEAN (TRUE/FALSE)
Veggies	Consume Vegetables 1 or more times per day	BOOLEAN (TRUE/FALSE)
HvyAlcoholConsump	Heavy drinkers (adult men having more than 14 drinks per week and adult women having more than 7 drinks per week)	BOOLEAN (TRUE/FALSE)
AnyHealthcare	Do you have any kind of health care coverage	FACTOR (Yes/No)
NoDocbcCost	Was there a time in the past 12 months when you needed to see a doctor but could not because of cost?	FACTOR (Yes/No)
GenHlth	Would you say that in general your health is:	FACTOR (1:Excellent, 2:Very Good, 3:Good, 4:Fair, 5:Poor)
MentHlth	For how many days during the past 30 days was your mental health not good?	INT (0-30)
PhysHlth	for how many days during the past 30 days was your physical health not good?	INT (0-30)
DiffWalk	Do you have serious difficulty walking or climbing stairs?	FACTOR (Yes/No)
Sex	Indicate sex of respondent	FACTOR (1:Male, 2:Female)
Age	Fourteen-level age category	FACTOR (1:Age 18 to 24,2:Age 25 to 29,3:Age 30 to 34,4:Age 35 to 39,5:Age 40 to 44,6:Age 45 to 49,7:Age 50 to 54,8:Age 55 to 59,9:Age 60 to 64,10:Age 65 to 69,11:Age 70 to 74,12:Age 75 to 79,13:Age 80 or older)
Education	What is the highest grade or year of school you completed?	FACTOR(1:Never attended school or only kindergarten,2:Grades 1 through 8 (Elementary),3:Grades 9 through 11 (Some high school),4:Grade 12 or GED (High school graduate),5:College 1 year to 3 years (Some college or technical school),6:College 4 years or more (College graduate))

Data Features	Survey Question	Data Type
Income	What is your annual household income from all sources:	FACTOR (1: Less than $10,000,2: Less than $15,000 ($10,000 to less than $15,000),3: Less than $20,000 ($15,000 to less than $20,000),4: Less than $25,000 ($20,000 to less than $25,000),5: Less than $35,000 ($25,000 to less than $35,000),6: Less than $50,000 ($35,000 to less than $50,000),7: Less than $75,000 ($50,000 to less than $75,000),8: $75,000 or more)

Changing data back to original state

A function is created called remap_features that contains all the mappings in python dictionaries that are to be applied to certain types of columns. The function takes in the data frame, the column name and an option from 1 to 7 depending on which dictionary is to be used.

Note: *Python is used in following two code chunks as the operation was easier to apply for the team members who are more familiar with Python than R*

For future reference a "Code:" tag will be added before each chunk to identify which language is used in the code chunk

Code: *Python*

```
# import libraries needed
import pandas as pd

#create a python data frame of the imported data
df = pd.DataFrame(r.data)

# function to remap the Boolean and yes/no columns
def remap_feature(df, col, y):

  # dictionaries for Boolean or Yes/No factors
  bool_dict = {1: True, 0: False}
  factor_dict = {1: "Yes", 0: "No"}

  # dictionary for general health column
  GenHlth_dict = {1:'Excellent', 2:'Very Good', 3:'Good', 4:'Fair', 5:'Poor'}

  # dictionary for male and female factors
  Sex_dict = {1:'Male', 0:'Female'}

  # dictionary for level of educations
  edu_dict = {1:'Only kindergarten',
              2:'Grades 1 - 8',
              3:'Grades 9 - 11',
              4:'Grade 12 or GED',
              5:'College 1 - 3 years',
              6:'College 4 years or more'}

  # dictionary for level of household income
  income_dict = {1: '< $10k',
                 2: '> $10k, < $15k',
                 3: '> $15k, < $20k',
                 4: '> $20k, < $25k',
                 5: '> $25k, < $35k',
                 6: '> $35k, < $50k',
                 7: '> $50k, < $75k',
```

```
                8: '> $75k'}

  age_dict = {1:'18 to 24',
              2:'25 to 29',
              3:'30 to 34',
              4:'35 to 39',
              5:'40 to 44',
              6:'45 to 49',
              7:'50 to 54',
              8:'55 to 59',
              9:'60 to 64',
              10:'65 to 69',
              11:'70 to 74',
              12:'75 to 79',
              13:'80 or older'}

  if(y == 1):
    df[col] = df[col].map(bool_dict)
  elif(y == 2):
    df[col] = df[col].map(factor_dict)
  elif(y == 3):
    df[col] = df[col].map(GenHlth_dict)
  elif(y == 4):
    df[col] = df[col].map(Sex_dict)
  elif(y == 5):
    df[col] = df[col].map(edu_dict)
  elif(y == 6):
    df[col] = df[col].map(income_dict)
  elif(y == 7):
    df[col] = df[col].map(age_dict)
  else:
    df[col] = df[col]

  return df[col]
```

The mappings are then applied, returning the adjusted data in the columns and the first few rows of the new mappings are viewed to see the changes made.

Code: *Python*

```
#------- convert columns back to how they where answered in the survey

# Boolean features
df['HighBP'] = remap_feature(df,'HighBP',1 )
df['HighChol'] = remap_feature(df,'HighChol',1 )
df['CholCheck'] = remap_feature(df,'CholCheck',1 )
df['PhysActivity'] = remap_feature(df,'PhysActivity',1 )
df['Fruits'] = remap_feature(df,'Fruits',1 )
df['Veggies'] = remap_feature(df,'Veggies',1 )
df['HvyAlcoholConsump'] = remap_feature(df,'HvyAlcoholConsump',1 )

# Yes/No Factor Features
df['Smoker'] = remap_feature(df,'Smoker',2 )
df['Stroke'] = remap_feature(df,'Stroke',2 )
df['HeartDiseaseorAttack'] = remap_feature(df,'HeartDiseaseorAttack',2 )
df['AnyHealthcare'] = remap_feature(df,'AnyHealthcare',2 )
df['NoDocbcCost'] = remap_feature(df,'NoDocbcCost',2 )
df['DiffWalk'] = remap_feature(df,'DiffWalk',2 )
```

```
# General Health Feature
df['GenHlth'] = remap_feature(df,'GenHlth',3 )

# sex feature
df['Sex'] = remap_feature(df,'Sex', 4)

# education level feature
df['Education']= remap_feature(df, 'Education', 5)

# level of income feature
df['Income'] = remap_feature(df, 'Income', 6)

# Level change for Age
df['Age'] = remap_feature(df, 'Age', 7)

print(df.iloc[:,0:20])
```

```
##        Diabetes_binary  HighBP  HighChol  ...  DiffWalk     Sex          Age
## 0                  0.0    True     False  ...        No    Male     35 to 39
## 1                  0.0    True      True  ...        No    Male     75 to 79
## 2                  0.0   False     False  ...        No    Male  80 or older
## 3                  0.0    True      True  ...        No    Male     70 to 74
## 4                  0.0   False     False  ...        No  Female     55 to 59
## ...                ...     ...       ...  ...       ...     ...          ...
## 70687              1.0   False      True  ...        No  Female     45 to 49
## 70688              1.0   False      True  ...       Yes    Male     65 to 69
## 70689              1.0    True      True  ...       Yes  Female  80 or older
## 70690              1.0    True      True  ...       Yes  Female     70 to 74
## 70691              1.0    True      True  ...        No  Female     60 to 64
##
## [70692 rows x 20 columns]
```

Now that the survey data has been reversed to its original text state it can be loaded back into a new R data frame for inspection.

Data types

The data types are inspected and changed as needed to either factor, logical or numeric.

Code: *R*

```
# converting pandas dataframe back to R dataframe
data.new <-  py$df

# view first 5 rows to double check the data conversion
head(data.new)
```

```
##   Diabetes_binary HighBP HighChol CholCheck BMI Smoker Stroke HeartDiseaseorAttack PhysActivity
## 1               0   TRUE    FALSE      TRUE  26     No     No                    No         TRUE
## 2               0   TRUE     TRUE      TRUE  26    Yes    Yes                    No        FALSE
## 3               0  FALSE    FALSE      TRUE  26     No     No                    No         TRUE
## 4               0   TRUE     TRUE      TRUE  28    Yes     No                    No         TRUE
## 5               0  FALSE    FALSE      TRUE  29    Yes     No                    No         TRUE
## 6               0  FALSE    FALSE      TRUE  18     No     No                    No         TRUE
##   Fruits Veggies HvyAlcoholConsump AnyHealthcare NoDocbcCost   GenHlth MentHlth PhysHlth DiffWalk
## 1  FALSE    TRUE             FALSE           Yes          No      Good        5       30       No
## 2   TRUE   FALSE             FALSE           Yes          No      Good        0        0       No
## 3   TRUE    TRUE             FALSE           Yes          No Excellent        0       10       No
## 4   TRUE    TRUE             FALSE           Yes          No      Good        0        3       No
## 5   TRUE    TRUE             FALSE           Yes          No Very Good        0        0       No
## 6   TRUE    TRUE             FALSE            No          No Very Good        7        0       No
##      Sex         Age              Education         Income
## 1   Male    35 to 39 College 4 years or more         > $75k
## 2   Male    75 to 79 College 4 years or more         > $75k
## 3   Male 80 or older College 4 years or more         > $75k
## 4   Male    70 to 74 College 4 years or more         > $75k
## 5 Female    55 to 59     College 1 - 3 years         > $75k
## 6 Female    18 to 24         Grade 12 or GED > $50k, < $75k
```

With the data now in its text translation for the factors we can recheck the data types and adjust what is needed before we explore the data set for any readily available insights.

Code: *R*

```
# view data types
str(data.new)
```

```
## 'data.frame':    70692 obs. of  22 variables:
##  $ Diabetes_binary     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ HighBP              : logi  TRUE TRUE FALSE TRUE FALSE FALSE ...
##  $ HighChol            : logi  FALSE TRUE FALSE TRUE FALSE FALSE ...
##  $ CholCheck           : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ BMI                 : num  26 26 26 28 29 18 26 31 32 27 ...
##  $ Smoker              : chr  "No" "Yes" "No" "Yes" ...
##  $ Stroke              : chr  "No" "Yes" "No" "No" ...
##  $ HeartDiseaseorAttack: chr  "No" "No" "No" "No" ...
##  $ PhysActivity        : logi  TRUE FALSE TRUE TRUE TRUE TRUE ...
##  $ Fruits              : logi  FALSE TRUE TRUE TRUE TRUE TRUE ...
##  $ Veggies             : logi  TRUE FALSE TRUE TRUE TRUE TRUE ...
##  $ HvyAlcoholConsump   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ AnyHealthcare       : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ NoDocbcCost         : chr  "No" "No" "No" "No" ...
##  $ GenHlth             : chr  "Good" "Good" "Excellent" "Good" ...
```

```
##  $ MentHlth            : num  5 0 0 0 0 7 0 0 0 0 ...
##  $ PhysHlth            : num  30 0 10 3 0 0 0 0 0 6 ...
##  $ DiffWalk            : chr  "No" "No" "No" "No" ...
##  $ Sex                 : chr  "Male" "Male" "Male" "Male" ...
##  $ Age                 : chr  "35 to 39" "75 to 79" "80 or older" "70 to 74" ...
##  $ Education           : chr  "College 4 years or more" "College 4 years or more" "College 4 years or more" "Col
##  $ Income              : chr  "> $75k" "> $75k" "> $75k" "> $75k" ...
##  - attr(*, "pandas.index")=RangeIndex(start=0, stop=70692, step=1)
```

Some of the columns have been recognized correctly as logical values and numbers but the character columns will need to be converted to factors

Code: *R*

```r
# converting character columns to factors
factor_cols <- c('Diabetes_binary','Smoker', 'Stroke', 'HeartDiseaseorAttack', 'AnyHealthcare',
                 'NoDocbcCost', 'GenHlth', 'DiffWalk', 'Sex', 'Age', 'Education', 'Income')

data.new[factor_cols] <- lapply(data.new[factor_cols], factor)

# view the new data types
str(data.new)
```

```
## 'data.frame':    70692 obs. of  22 variables:
##  $ Diabetes_binary     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ HighBP              : logi  TRUE TRUE FALSE TRUE FALSE FALSE ...
##  $ HighChol            : logi  FALSE TRUE FALSE TRUE FALSE FALSE ...
##  $ CholCheck           : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ BMI                 : num  26 26 26 28 29 18 26 31 32 27 ...
##  $ Smoker              : Factor w/ 2 levels "No","Yes": 1 2 1 2 2 1 2 2 1 2 ...
##  $ Stroke              : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...
##  $ HeartDiseaseorAttack: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PhysActivity        : logi  TRUE FALSE TRUE TRUE TRUE TRUE ...
##  $ Fruits              : logi  FALSE TRUE TRUE TRUE TRUE TRUE ...
##  $ Veggies             : logi  TRUE FALSE TRUE TRUE TRUE TRUE ...
##  $ HvyAlcoholConsump   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ AnyHealthcare       : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 2 2 ...
##  $ NoDocbcCost         : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ GenHlth             : Factor w/ 5 levels "Excellent","Fair",..: 3 3 1 3 5 5 1 2 3 3 ...
##  $ MentHlth            : num  5 0 0 0 0 7 0 0 0 0 ...
##  $ PhysHlth            : num  30 0 10 3 0 0 0 0 0 6 ...
##  $ DiffWalk            : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Sex                 : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 1 2 2 1 2 ...
##  $ Age                 : Factor w/ 13 levels "18 to 24","25 to 29",..: 4 12 13 11 8 1 13 6 3 6 ...
##  $ Education           : Factor w/ 6 levels "College 1 - 3 years",..: 2 2 2 2 1 3 1 3 2 3 ...
##  $ Income              : Factor w/ 8 levels "< $10k","> $10k, < $15k",..: 8 8 8 8 8 7 6 3 8 4 ...
##  - attr(*, "pandas.index")=RangeIndex(start=0, stop=70692, step=1)
```

All columns have now been changed to the correct data type. We can review the data summary to get a better idea of what is present in the data set

Code: *R*

```
# view data summary
summary(data.new)
```

```
##  Diabetes_binary   HighBP         HighChol       CholCheck            BMI        Smoker
##  0:35346         Mode :logical   Mode :logical   Mode :logical   Min.   :12.00   No :37094
##  1:35346         FALSE:30860     FALSE:33529     FALSE:1749      1st Qu.:25.00   Yes:33598
##                  TRUE :39832     TRUE :37163     TRUE :68943     Median :29.00
##                                                                  Mean   :29.86
##                                                                  3rd Qu.:33.00
##                                                                  Max.   :98.00
##
##  Stroke      HeartDiseaseorAttack PhysActivity      Fruits         Veggies        HvyAlcoholConsump
##  No :66297   No :60243            Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  Yes: 4395   Yes:10449            FALSE:20993     FALSE:27443     FALSE:14932     FALSE:67672
##                                   TRUE :49699     TRUE :43249     TRUE :55760     TRUE :3020
##
##
##
##
##  AnyHealthcare NoDocbcCost      GenHlth         MentHlth         PhysHlth     DiffWalk
##  No : 3184     No :64053   Excellent: 8282   Min.   : 0.000   Min.   : 0.00   No :52826
##  Yes:67508     Yes: 6639   Fair     :13303   1st Qu.: 0.000   1st Qu.: 0.00   Yes:17866
##                            Good     :23427   Median : 0.000   Median : 0.00
##                            Poor     : 5808   Mean   : 3.752   Mean   : 5.81
##                            Very Good:19872   3rd Qu.: 2.000   3rd Qu.: 6.00
##                                              Max.   :30.000   Max.   :30.00
##
##      Sex                 Age                          Education                Income
##  Female:38386   65 to 69   :10856   College 1 - 3 years    :20030   > $75k        :20646
##  Male  :32306   60 to 64   :10112   College 4 years or more:26020   > $50k, < $75k:11425
##                 55 to 59   : 8603   Grade 12 or GED        :19473   > $35k, < $50k:10287
##                 70 to 74   : 8044   Grades 1 - 8           : 1647   > $25k, < $35k: 8010
##                 50 to 54   : 6872   Grades 9 - 11          : 3447   > $20k, < $25k: 6658
##                 80 or older: 5426   Only kindergarten      :   75   > $15k, < $20k: 5557
##                 (Other)    :20779                                   (Other)       : 8109
```

With the data set initially cleaned and ready to explore we can make a new data frame for the ready to use data set.

Code: *R*

```
# new cleaned data set
data.clean <- data.new

# view the first 5 rows
head(data.clean)
```

```
##   Diabetes_binary HighBP HighChol CholCheck BMI Smoker Stroke HeartDiseaseorAttack PhysActivity
## 1               0   TRUE    FALSE      TRUE  26     No     No                    No         TRUE
## 2               0   TRUE     TRUE      TRUE  26    Yes    Yes                    No        FALSE
## 3               0  FALSE    FALSE      TRUE  26     No     No                    No         TRUE
## 4               0   TRUE     TRUE      TRUE  28    Yes     No                    No         TRUE
## 5               0  FALSE    FALSE      TRUE  29    Yes     No                    No         TRUE
## 6               0  FALSE    FALSE      TRUE  18     No     No                    No         TRUE
##   Fruits Veggies HvyAlcoholConsump AnyHealthcare NoDocbcCost   GenHlth MentHlth PhysHlth DiffWalk
## 1  FALSE    TRUE             FALSE           Yes          No      Good        5       30       No
## 2   TRUE   FALSE             FALSE           Yes          No      Good        0        0       No
## 3   TRUE    TRUE             FALSE           Yes          No Excellent        0       10       No
## 4   TRUE    TRUE             FALSE           Yes          No      Good        0        3       No
## 5   TRUE    TRUE             FALSE           Yes          No Very Good        0        0       No
## 6   TRUE    TRUE             FALSE            No          No Very Good        7        0       No
##      Sex         Age               Education         Income
## 1   Male    35 to 39 College 4 years or more         > $75k
## 2   Male    75 to 79 College 4 years or more         > $75k
## 3   Male 80 or older College 4 years or more         > $75k
## 4   Male    70 to 74 College 4 years or more         > $75k
## 5 Female    55 to 59     College 1 - 3 years         > $75k
## 6 Female    18 to 24         Grade 12 or GED > $50k, < $75k
```

Dataset Exploration

First the numeric values will be explored, the numeric values in the data are represented by either days for **MentHlth** or **PhysHlth** and an integer range for **BMI**.

The first two may be turned into factors but the features can be explored in their numeric representation first. As turning them into factors would lead to having two features with 30 levels of factors.

BMI

Exploring the BMI feature and its individual relationship to the diabetes diagnosis. BMI stands for Body Mass Index and is a measure that uses your height and weight to work out if your weight is healthy.

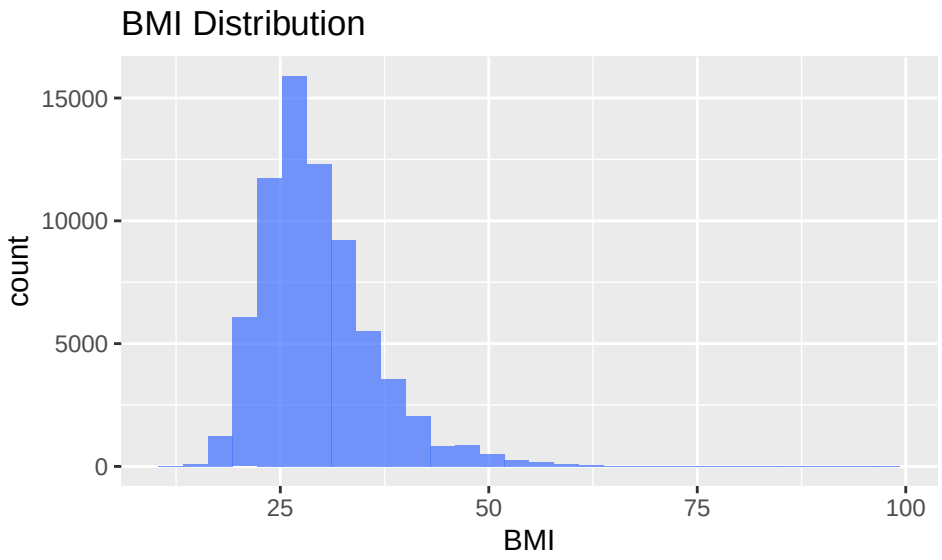
The BMI calculation divides an adult's weight in kilograms by their height in meters squared. For most adults, an ideal BMI is in the 18.5 to 24.9 range.

Distributions

First the distribution characteristics of the BMI feature will be explored, then the distributions of the BMI between the Diabetic and Non-Diabetic groups.

Code: *R*

```
# histogram distribution
ggplot(data.clean, aes(x = BMI)) +
  geom_histogram(fill='royalblue1', alpha = 0.75, bins=30)+
  ggtitle("BMI Distribution")
```



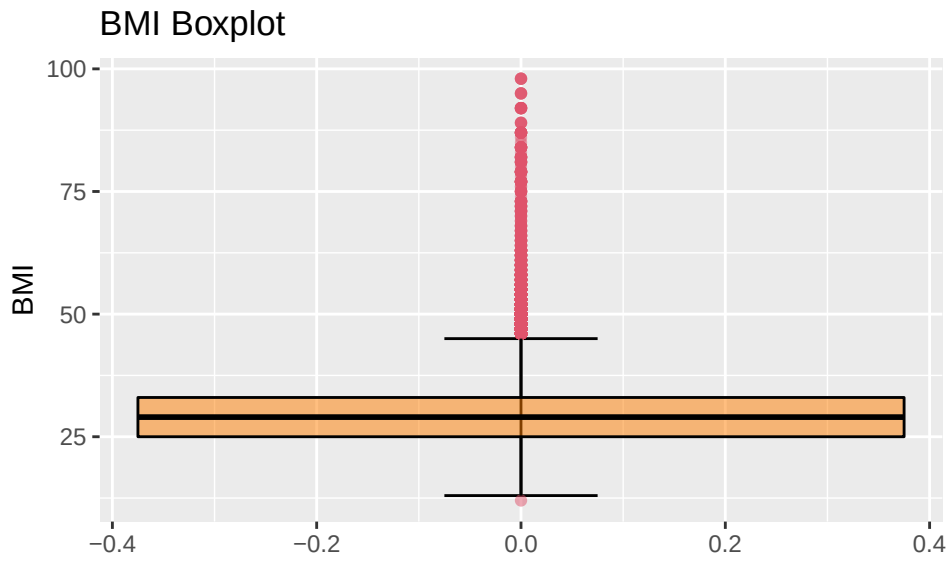

```
ggsave(path='./graphs',filename = '1_bmi_dist.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

The BMI doesn't appear to be normally distributed, it appears heavily skewed to the left with a long right tail. We can use a box plot to visualize the quartile ranges as well as see outliers in the data.

Code: *R*

```
# box plot of BMI range
ggplot(data.clean, aes(y = BMI)) +
  stat_boxplot(geom = "errorbar",
               width = 0.15,
               color = 1) +  # Error bar color
  geom_boxplot(fill = 'darkorange1',           # Box color
               alpha = 0.5,         # Transparency
               color = 1,           # Border color
               outlier.colour = 2)+
  ggtitle("BMI Boxplot")
```



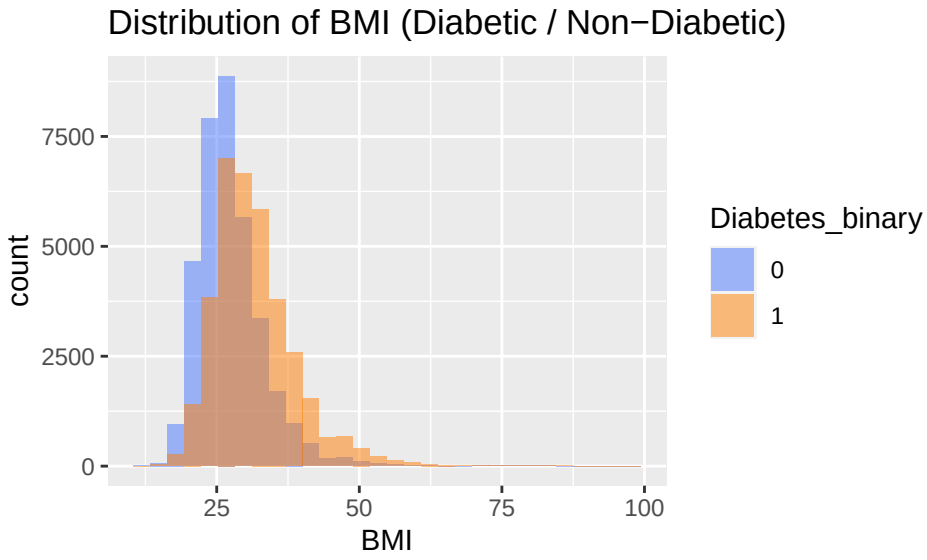

```
ggsave(path='./graphs',filename = '2_bmi_box.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

The box plot similarly shows there is a long string of outliers outside the 3rd quartile range. As Diabetes is the predictor in this analysis the same distributions can be grouped by their diabetic diagnosis to see if there are any initial differences between the two groups in BMI range.

Code: *R*

```
# grouped histogram of BMI by diabetic diagnosis
ggplot(data.clean, aes(x = BMI, fill = Diabetes_binary)) +
  geom_histogram(alpha = 0.5, position = "identity", bins=30)+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("Distribution of BMI (Diabetic / Non-Diabetic)")
```




```
ggsave(path='./graphs',filename = '3_bmi_both_dist.png', dpi = 300)
```

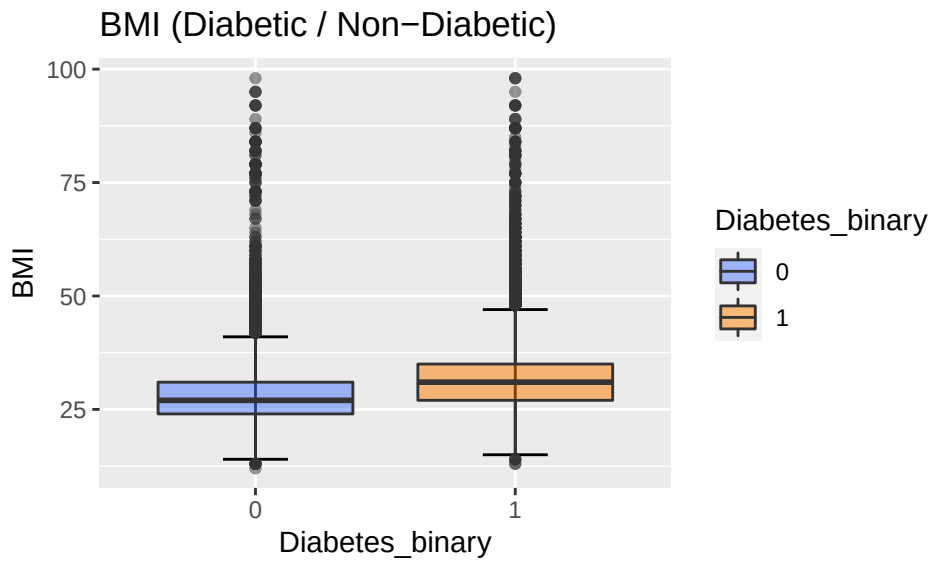
```
## Saving 5 x 3 in image
```

From the histogram distribution it does appear that there is a shift int BMI towards the right tail with respect to people with a diabetic condition.

A box plot of the groups can show if there is any visual differences in the distribution around the means.

Code: *R*

```
ggplot(data.clean, aes(x = Diabetes_binary, y = BMI, fill = Diabetes_binary)) +
  stat_boxplot(geom = "errorbar",
               width = 0.25) +
  geom_boxplot(alpha = 0.5)+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("BMI (Diabetic / Non-Diabetic)")
```




```
ggsave(path='./graphs',filename = '4_bmi_both_box.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

The box plot of the two groups (Diabetic / Non-Diabetic) indicates that there is a difference in the means between the two groups and there are a significant amount of values outside the 3rd quartile for each group.

Normality testing

Normality can be tested across the entire BMI feature as well as across the two Diabetic diagnosis groups. QQ plots will be used for visualizing the normality as well as normality testing where an alpha value of 0.05 is used to determine the distribution normality.

Code: *R*

```
#perform kolmogorov-smirnov test
ks.test(data.clean$BMI, 'pnorm')
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.clean$BMI
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

A Kolmogorov-Smirnov normality test on the BMI feature indicates that the data isn't normally distributed with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

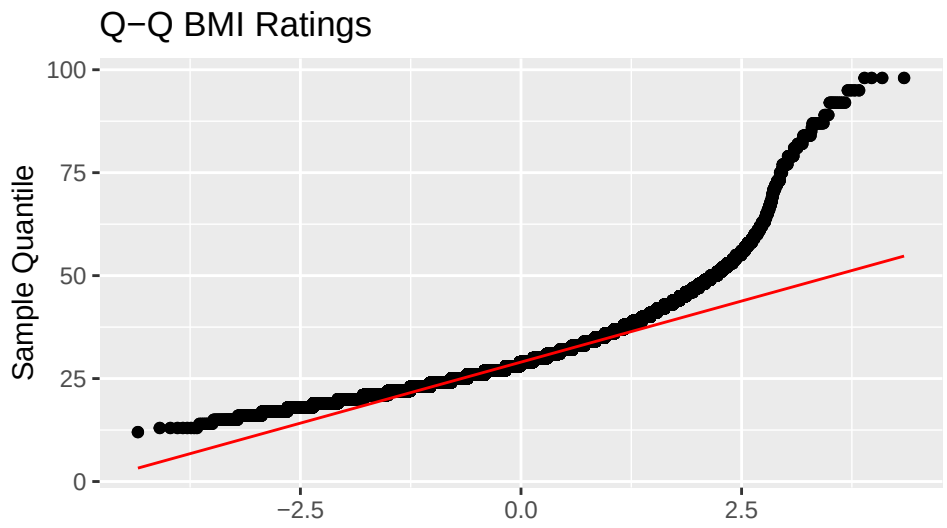
```
#perform shapiro-wilk test
lillie.test(data.clean$BMI)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.clean$BMI
## D = 0.10863, p-value < 2.2e-16
```

A second Kolmogorov-Smirnov normality test using the Lillie Fors correction on the BMI feature indicates that the data isn't normally distributed with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

```
# qq plot of BMI distribution
qplot(sample = BMI, data = data.clean)+
labs(title="Q-Q BMI Ratings",
       y = "Sample Quantile")+
  stat_qq() +
  stat_qq_line(colour = "red")
```




```
ggsave(path='./graphs',filename = '5_bmi_qq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Visually inspecting on the QQ plot also provides enough evidence that the BMI feature is not normally distributed and that parametric statistical tests would not be suitable on this data.

Code: *R*

```
#perform kolmogorov-smirnov test
ks.test(data.clean$BMI[data.clean$Diabetes_binary == 1], 'pnorm')
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.clean$BMI[data.clean$Diabetes_binary == 1]
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

A Kolmogorov-Smirnov normality test on the BMI feature when separated as only Diabetic diagnosed indicates that the data isn't normally distributed with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

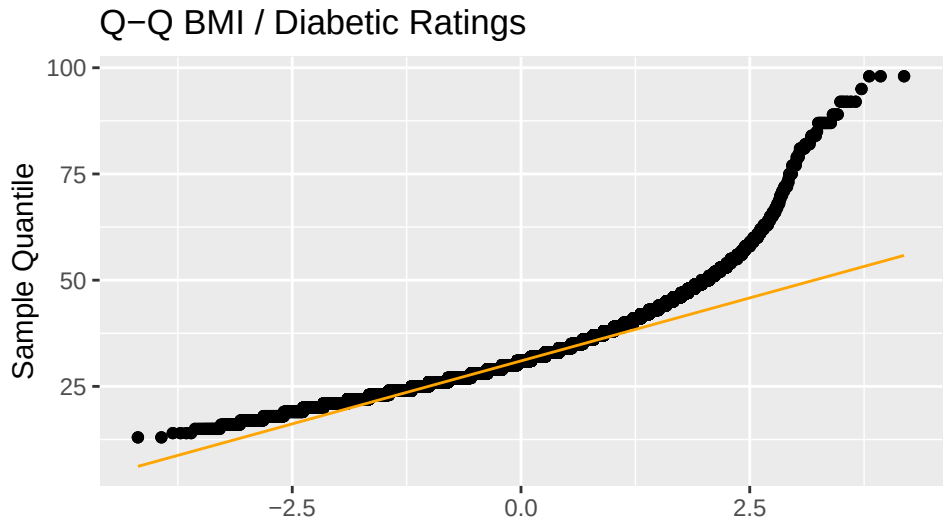
```
#perform shapiro-wilk test
lillie.test(data.clean$BMI[data.clean$Diabetes_binary == 1])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.clean$BMI[data.clean$Diabetes_binary == 1]
## D = 0.10453, p-value < 2.2e-16
```

A second Kolmogorov-Smirnov normality test using the Lillie Fors correction on the BMI feature when separated as only Diabetic diagnosed indicates that the data isn't normally distributed with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

```
# qq plot of BMI distribution for Diabetic diagnosis
qplot(sample = BMI, data = data.clean[data.clean$Diabetes_binary == 1,])+
labs(title="Q-Q BMI / Diabetic Ratings",
       y = "Sample Quantile")+
  stat_qq() +
  stat_qq_line(colour = "orange")
```



```r
ggsave(path='./graphs',filename = '6_bmi_dia_qq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Visually inspecting on the QQ plot also provides enough evidence that the BMI feature when separated as only Diabetic diagnosed is not normally distributed and that parametric statistical tests would not be suitable on this data.

Code: *R*

```r
#perform kolmogorov-smirnov test
ks.test(data.clean$BMI[data.clean$Diabetes_binary == 0], 'pnorm')
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.clean$BMI[data.clean$Diabetes_binary == 0]
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

A Kolmogorov-Smirnov normality test on the BMI feature when separated as not Diabetic diagnosed indicates that the data isn't normally distributed with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

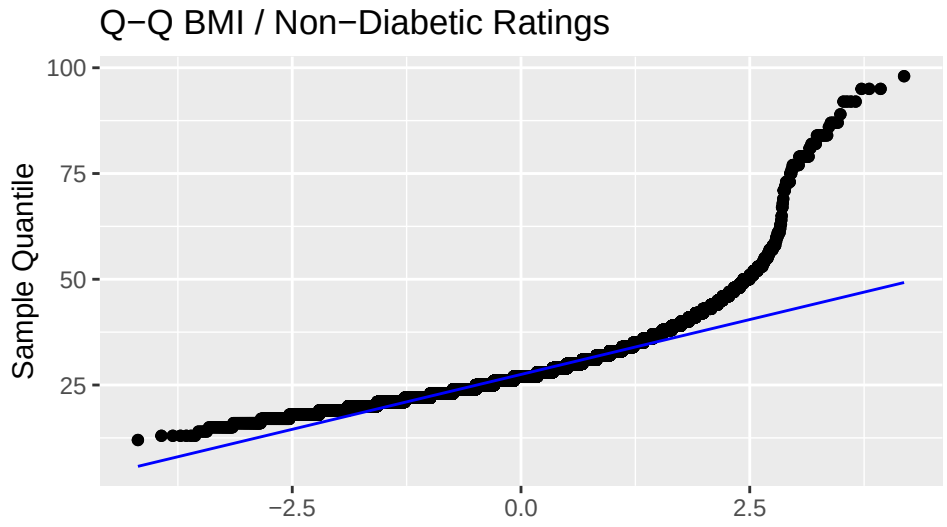
```r
#perform shapiro-wilk test
lillie.test(data.clean$BMI[data.clean$Diabetes_binary == 0])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.clean$BMI[data.clean$Diabetes_binary == 0]
## D = 0.12092, p-value < 2.2e-16
```

A second Kolmogorov-Smirnov normality test using the Lillie Fors correction on the BMI feature when separated as not Diabetic diagnosed indicates that the data isn't normally distributed with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

```
# qq plot of BMI distribution for Non-Diabetic diagnosis
qplot(sample = BMI, data = data.clean[data.clean$Diabetes_binary == 0,])+
labs(title="Q-Q BMI / Non-Diabetic Ratings",
       y = "Sample Quantile")+
  stat_qq() +
  stat_qq_line(colour = "blue")
```




```
ggsave(path='./graphs',filename = '7_bmi_non_dia_qq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Visually inspecting on the QQ plot also provides enough evidence that the BMI feature when separated as not Diabetic diagnosed is not normally distributed and that parametric statistical tests would not be suitable on this data.

Manual testing of the central tendencies can also be checked to confirm all the previous findings about the BMI feature and also to check the standard deviation and variations

Code: *R*

```
m <- mean(data.clean$BMI)
md <- median(data.clean$BMI)
s <- sd(data.clean$BMI)
v <- var(data.clean$BMI)

sprintf("Mean: %f Median: %f SD: %f Var: %f ",m,md,s, v)
```

```
## [1] "Mean: 29.856985 Median: 29.000000 SD: 7.113954 Var: 50.608339 "
```

Although the data is not normally distributed the mean and median values are close, with the standard deviation of 7.11 and a variance of 50.61

MentHlth

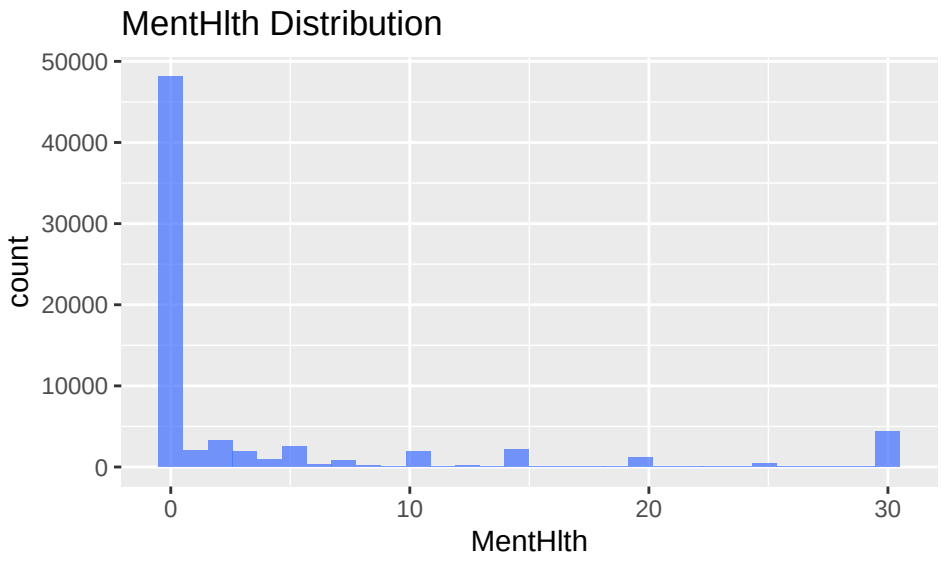
Exploring the MentHlth feature and its individual relationship to the diabetes diagnosis. In the CDC survey the values in the data represent how many days during the past 30 days was the persons mental health not good.

Distributions

First the distribution characteristics of the MentHlth feature will be explored, then the distributions of the MentHlth between the Diabetic and Non-Diabetic groups.

Code: *R*

```
# histogram distribution
ggplot(data.clean, aes(x = MentHlth)) +
  geom_histogram(fill='royalblue1', alpha = 0.75, bins=30)+
  ggtitle("MentHlth Distribution")
```




```
ggsave(path='./graphs',filename = '8_menthlth_dist.png', dpi = 300)
```

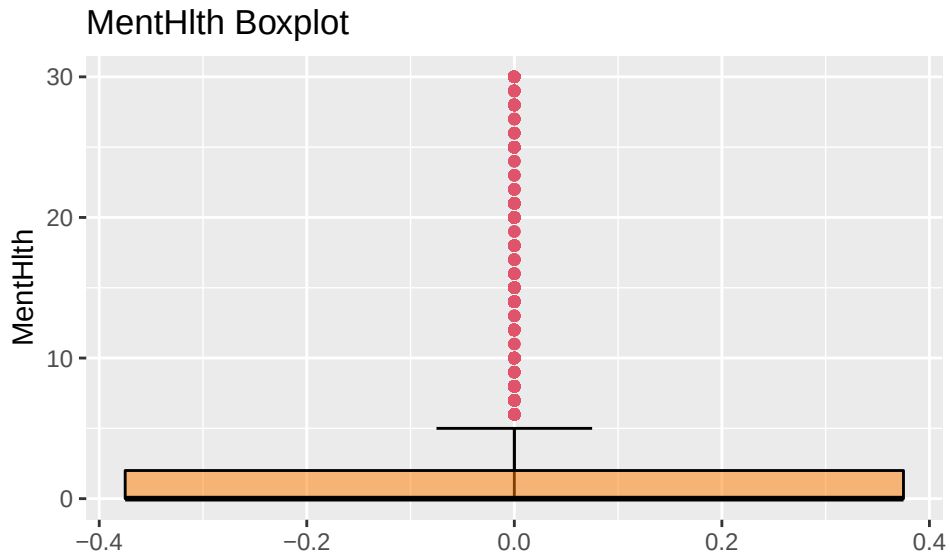
```
## Saving 5 x 3 in image
```

The MentHlth distribution is not normally distributed and is heavily present in the zero to 5 day range. This data may be better represented as a factor in this data but it can still be investigated as a numeric value for exploration purposes.

A box plot can be used to visually check the spread of data through the quartile ranges.

Code: *R*

```
# box plot of BMI range
ggplot(data.clean, aes(y = MentHlth)) +
  stat_boxplot(geom = "errorbar",
               width = 0.15,
               color = 1) +  # Error bar color
  geom_boxplot(fill = 'darkorange1',           # Box color
               alpha = 0.5,         # Transparency
               color = 1,           # Border color
               outlier.colour = 2)+
  ggtitle("MentHlth Boxplot")
```




```
ggsave(path='./graphs',filename = '9_menthlth_box.png', dpi = 300)
```

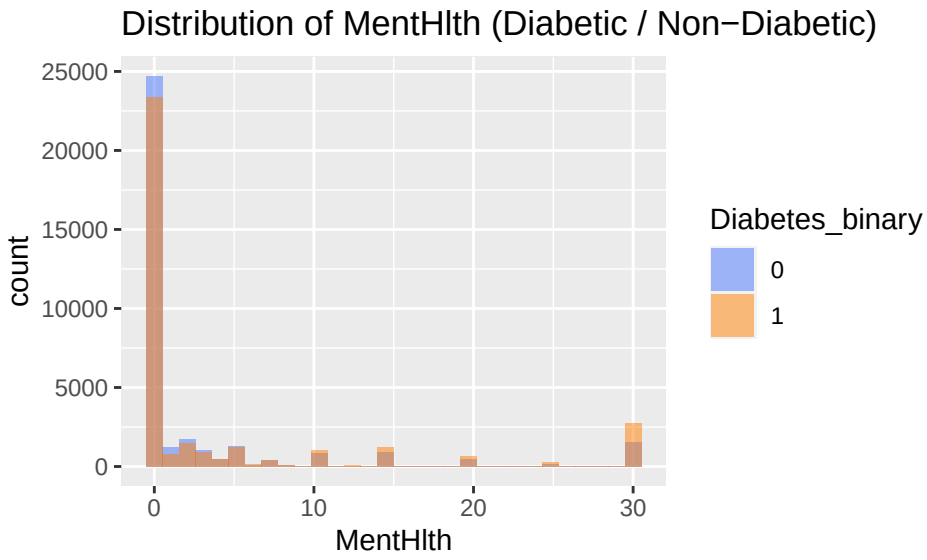
```
## Saving 5 x 3 in image
```

The box plot similarly shows there is a long string of outliers outside the 3rd quartile range and the data is heavily weighted in the 0 to 5 day range with significant outliers.

Diabetes being the predictor in the analysis the same distributions will be grouped by their diabetic diagnosis to see if there are other initial differences between the two groups in MentHlth days.

Code: *R*

```
# grouped histogram of BMI by diabetic diagnosis
ggplot(data.clean, aes(x = MentHlth, fill = Diabetes_binary)) +
  geom_histogram(alpha = 0.5, position = "identity", bins=30)+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("Distribution of MentHlth (Diabetic / Non-Diabetic)")
```



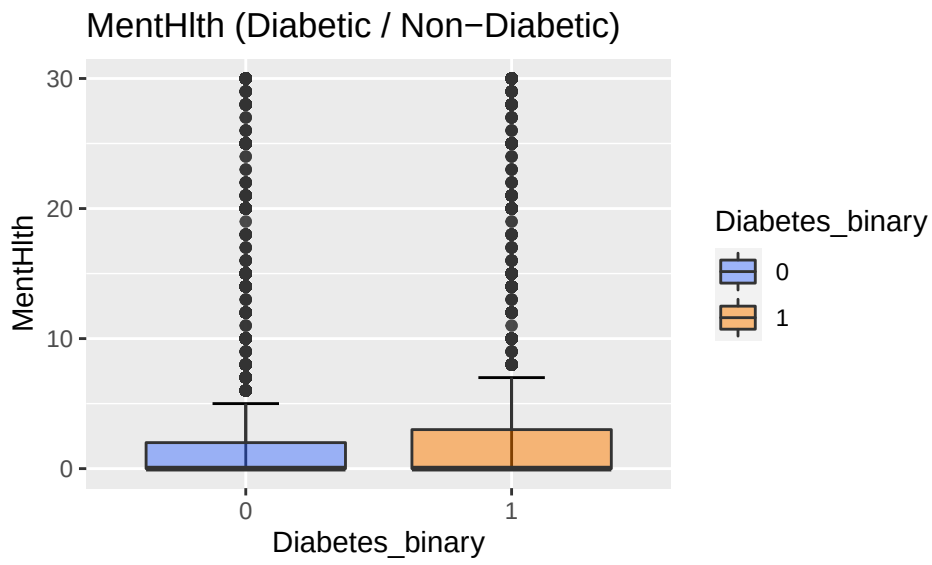

```
ggsave(path='./graphs',filename = '10_menthlth_both_dist.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

The grouped distributions show very little differences in the distributions, also gives more credence to use of this feature as factor in later tests.

Code: *R*

```
ggplot(data.clean, aes(x = Diabetes_binary, y = MentHlth, fill = Diabetes_binary)) +
  stat_boxplot(geom = "errorbar",
               width = 0.25) +
  geom_boxplot(alpha = 0.5)+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("MentHlth (Diabetic / Non-Diabetic)")
```




```
ggsave(path='./graphs',filename = '11_menthlth_both_box.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Box plots of the MentHlth feature again show that the data is heavy weighted in the 0 to 5 day range with a uniform spread of outliers, the grouped box plots show that the range for diabetic diagnosis shows a wider variance in the inter quartile range but with a similar mean.

Normality testing

Normality will be tested for the MentHlth feature as a whole as well as divided into the two Diabetic diagnosis groups.

QQ plots will be used for visualizing the normality as well as normality testing where an alpha value of 0.05 is used to determine the distribution normality.

Code: *R*

```
#perform kolmogorov-smirnov test
ks.test(data.clean$MentHlth, 'pnorm')
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.clean$MentHlth
## D = 0.5, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

The Kolmogorov-Smirnov normality test on the MentHlth feature indicates that the data isn't normally distributed with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

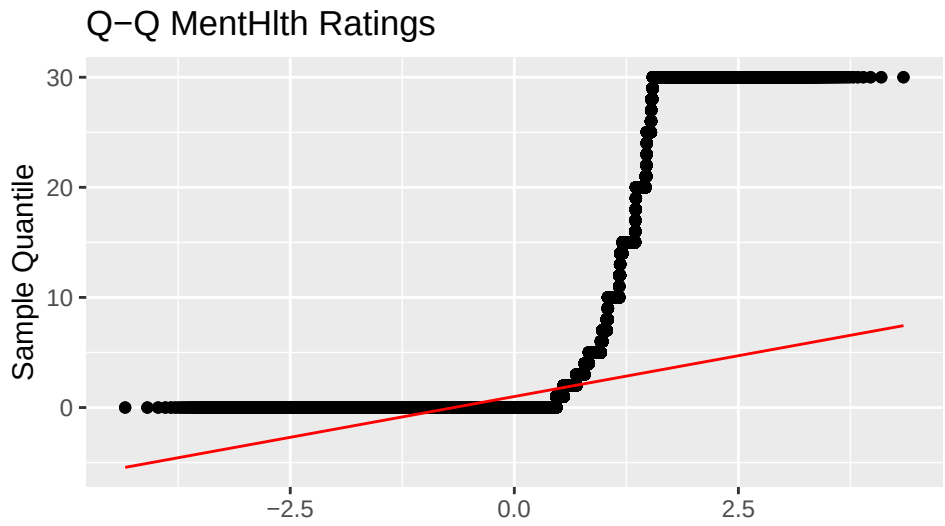
```
#perform shapiro-wilk test
lillie.test(data.clean$MentHlth)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.clean$MentHlth
## D = 0.35755, p-value < 2.2e-16
```

Also the Kolmogorov-Smirnov normality test using the Lillie Fors correction on the MentHlth feature indicates that the data isn't normally distributed with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

```
# qq plot of BMI distribution
qplot(sample = MentHlth, data = data.clean)+
labs(title="Q-Q MentHlth Ratings",
       y = "Sample Quantile")+
  stat_qq() +
  stat_qq_line(colour = "red")
```




```
ggsave(path='./graphs',filename = '12_mentlhlth_qq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Looking at the qq plot, the distortion of the MentHlth can be seen, the data is not normally distributed. Again this data feature could be retested as a categorical variable instead of numeric.

Code: *R*

```
#perform kolmogorov-smirnov test
ks.test(data.clean$MentHlth[data.clean$Diabetes_binary == 1], 'pnorm')
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.clean$MentHlth[data.clean$Diabetes_binary == 1]
## D = 0.5, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Separating the MentHlth column out by the Diabetic diagnosis does not show any change in the Kolmogorov-Smirnov normality test, with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

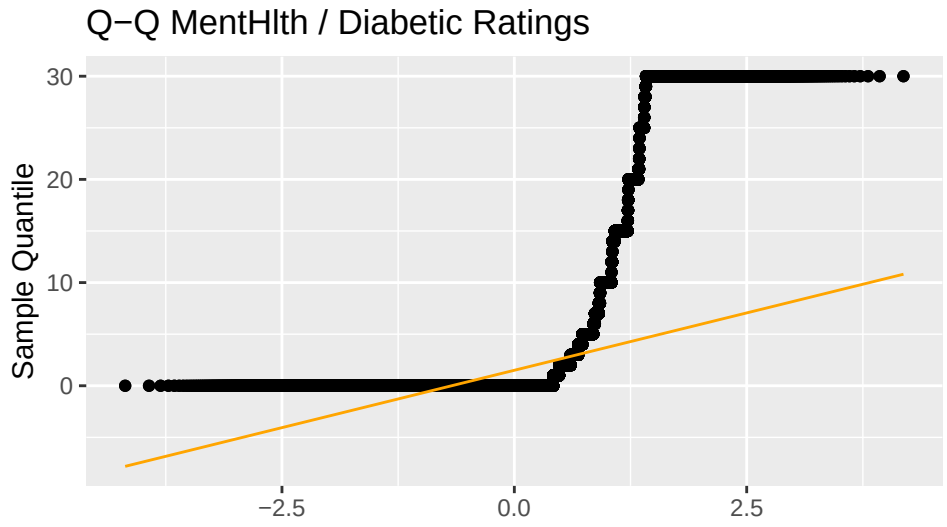
```
#perform shapiro-wilk test
lillie.test(data.clean$MentHlth[data.clean$Diabetes_binary == 1])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.clean$MentHlth[data.clean$Diabetes_binary == 1]
## D = 0.3531, p-value < 2.2e-16
```

Again using the Lillie Fors method, does not change the results of the Kolmogorov-Smirnov normality test, with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

```
# qq plot of BMI distribution for Diabetic diagnosis
qplot(sample = MentHlth, data = data.clean[data.clean$Diabetes_binary == 1,])+
labs(title="Q-Q MentHlth / Diabetic Ratings",
       y = "Sample Quantile")+
  stat_qq() +
  stat_qq_line(colour = "orange")
```




```
ggsave(path='./graphs',filename = '13_mentlhlth_dia_qq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

As expected the qq plot shows similar shape when separated by diagnosis to to the the MnthHlth column as a whole.

Code: *R*

```
#perform kolmogorov-smirnov test
ks.test(data.clean$MentHlth[data.clean$Diabetes_binary == 0], 'pnorm')
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.clean$MentHlth[data.clean$Diabetes_binary == 0]
## D = 0.5, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

The same results are seen for the people have not been diagnosed with diabetes with a p-value of less than the alpha 0.05 for the normality test.

Code: *R*

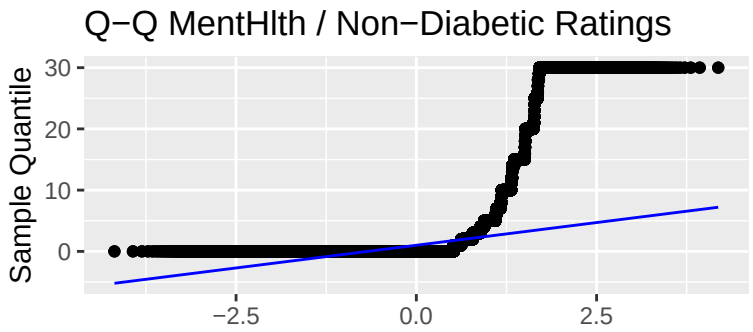
```
#perform shapiro-wilk test
lillie.test(data.clean$MentHlth[data.clean$Diabetes_binary == 0])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.clean$MentHlth[data.clean$Diabetes_binary == 0]
## D = 0.36197, p-value < 2.2e-16
```

The Lillie Fors method again doesn't change the p-value in any way for the non-Diabetic people.

Code: *R*

```
# qq plot of BMI distribution for Non-Diabetic diagnosis
qplot(sample = MentHlth, data = data.clean[data.clean$Diabetes_binary == 0,])+
labs(title="Q-Q MentHlth / Non-Diabetic Ratings",
       y = "Sample Quantile")+
  stat_qq() +
  stat_qq_line(colour = "blue")
```




```
ggsave(path='./graphs',filename = '14_mntlhlth_non_dia_qq.png', dpi = 300)
```

```
## Saving 4 x 2 in image
```

And the qq plot shows a similar trend to the previous qq plots for the MntHlth column with not normally distributed data.

Code: *R*

```
m <- mean(data.clean$MentHlth)
md <- median(data.clean$MentHlth)
s <- sd(data.clean$MentHlth)
v <- var(data.clean$MentHlth)

sprintf("Mean: %f Median: %f SD: %f Var: %f", m, md, s, v)
```

```
## [1] "Mean: 3.752037 Median: 0.000000 SD: 8.155627 Var: 66.514244"
```

The box plots and histograms previous indicated a high degree of values in the 0 to 5 day range and it can be seen from the mean of 3.75 and the median of 0 that this is the case, with a standard deviation of 8.15 and a variance of 66.514.

PhysHlth

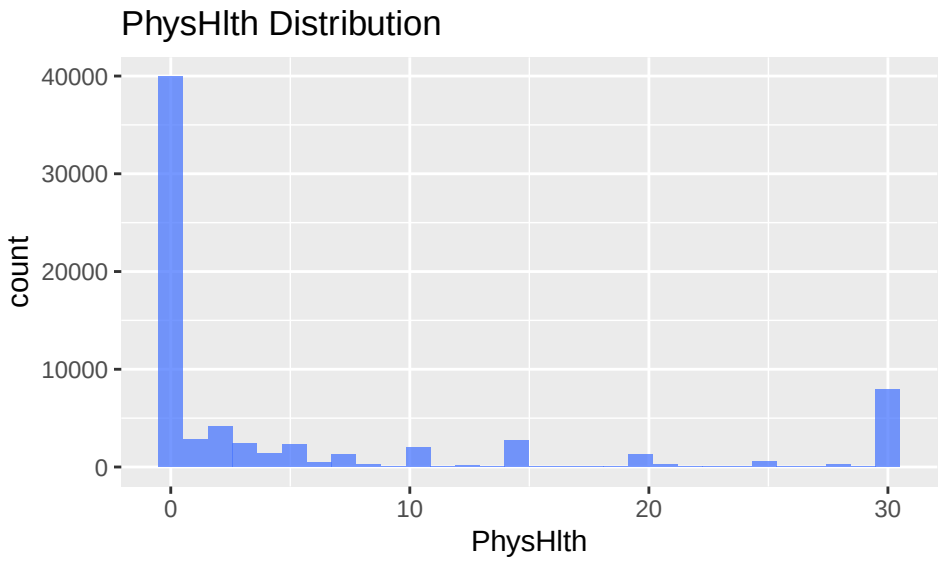
Exploring the MentHlth feature and its individual relationship to the diabetes diagnosis. In the CDC survey the values in the data represent how many days during the past 30 days the persons physical health not good.

Distributions

First the distribution characteristics of the PhysHlth feature will be explored, then the distributions of the PhysHlth between the Diabetic and Non-Diabetic groups.

Code: *R*

```
# histogram distribution
ggplot(data.clean, aes(x = PhysHlth)) +
  geom_histogram(fill='royalblue1', alpha = 0.75, bins=30)+
  ggtitle("PhysHlth Distribution")
```



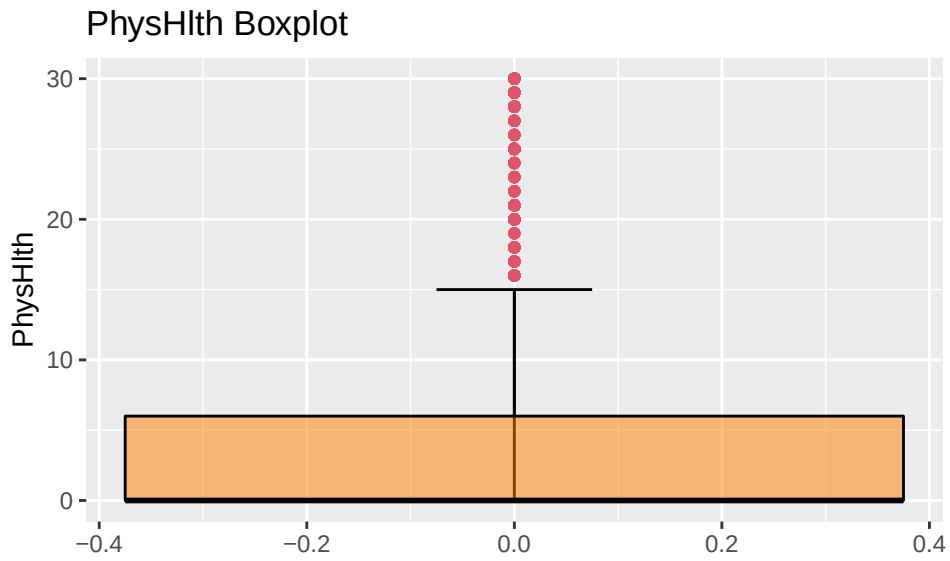

```
ggsave(path='./graphs',filename = '15_physhlth_dist.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

The distribution in the PhysHlth column has a similar shape to that of the MntHlth column with a higher frequency in the 0 day range.

Code: *R*

```
# box plot of BMI range
ggplot(data.clean, aes(y = PhysHlth)) +
  stat_boxplot(geom = "errorbar",
               width = 0.15,
               color = 1) +  # Error bar color
  geom_boxplot(fill = 'darkorange1',           # Box color
               alpha = 0.5,         # Transparency
               color = 1,           # Border color
               outlier.colour = 2)+
  ggtitle("PhysHlth Boxplot")
```



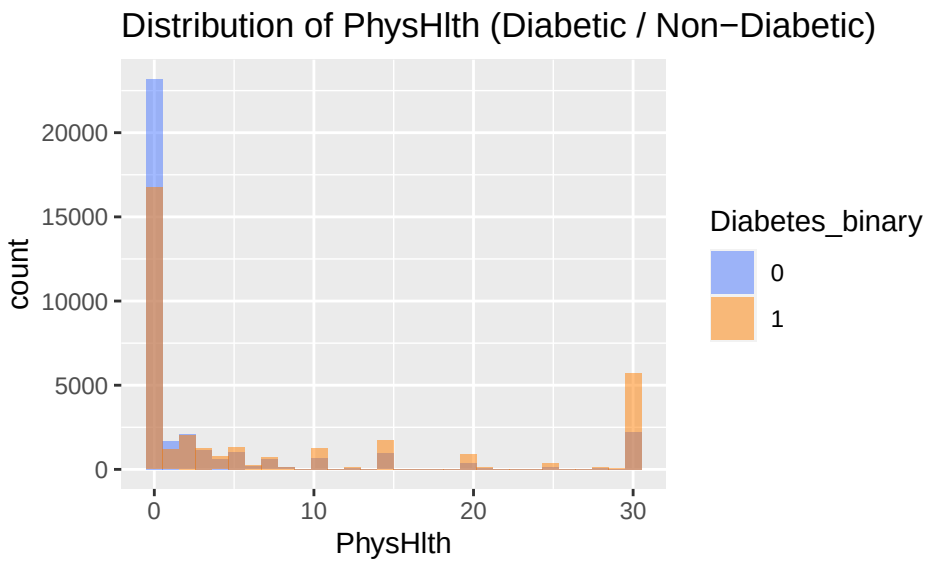

```
ggsave(path='./graphs',filename = '16_physhlth_box.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

The box plot of the PhysHlth column again shows a high weighting in the 0 to 5 day range with outliers out to the max day range of 30 days. Similar to the MntHlth column this could possible be interpreted as a factor.

Code: *R*

```
# grouped histogram of BMI by diabetic diagnosis
ggplot(data.clean, aes(x = PhysHlth, fill = Diabetes_binary)) +
  geom_histogram(alpha = 0.5, position = "identity", bins=30)+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("Distribution of PhysHlth (Diabetic / Non-Diabetic)")
```



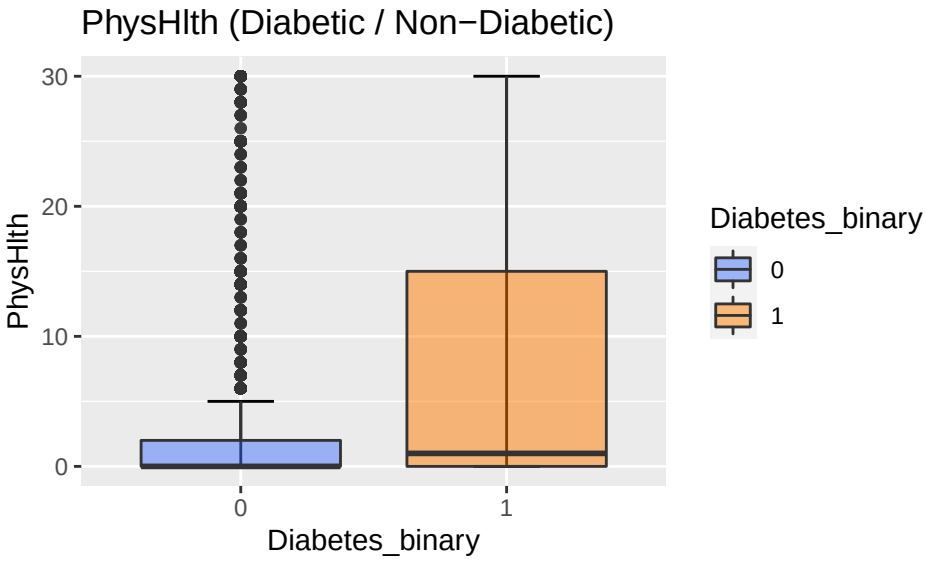

```
ggsave(path='./graphs',filename = '17_physhlth_both_dist.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Again the distribution when grouped by the Diabetes diagnosis follows a similar pattern with the exception that people reported as having Diabetes have a higher number of days reported as "not good" than those without diabetes. Futher testing will be carried out to see if this has a significant impact on diagnosis.

Code: *R*

```
ggplot(data.clean, aes(x = Diabetes_binary, y = PhysHlth, fill = Diabetes_binary)) +
  stat_boxplot(geom = "errorbar",
               width = 0.25) +
  geom_boxplot(alpha = 0.5)+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("PhysHlth (Diabetic / Non-Diabetic)")
```



```r
ggsave(path='./graphs',filename = '18_physhlth_both_box.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

The box plot shows that most of the outliers appear in the non-diabetic cohort while the diabetic cohort appears to have most of its values within the quartile range.

Normality testing

Normality will be tested for the PhysHlth feature and same as previous it will be tested as as a whole as well as divided into the two Diabetic diagnosis groups.

QQ plots will be used for visualizing the normality as well as normality testing where an alpha value of 0.05 is used to determine the distribution normality.

Code: *R*

```r
#perform kolmogorov-smirnov test
ks.test(data.clean$PhysHlth, 'pnorm')
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.clean$PhysHlth
## D = 0.5, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

The Kolmogorov-Smirnov normality test on the PhysHlth feature, as expected, indicates that the data isn't normally distributed with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

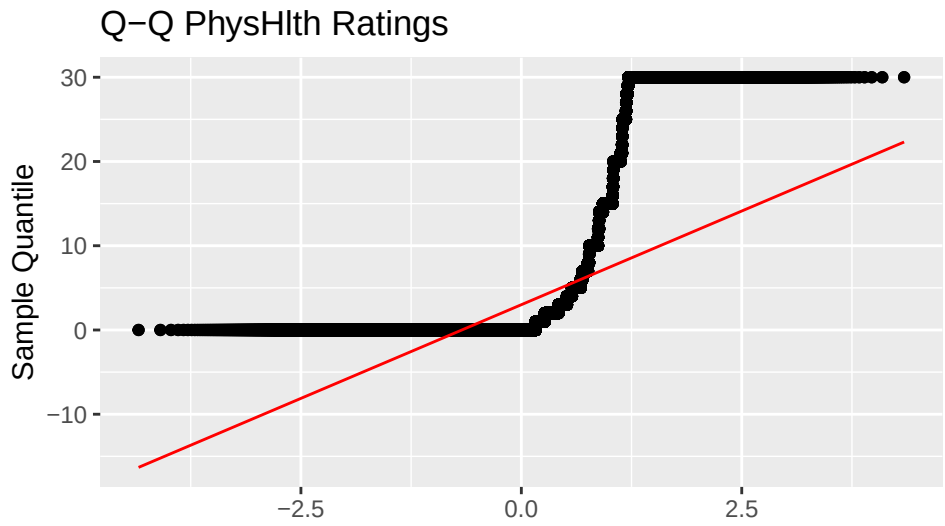
```r
#perform shapiro-wilk test
lillie.test(data.clean$PhysHlth)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.clean$PhysHlth
## D = 0.31056, p-value < 2.2e-16
```

Also as expected, the Kolmogorov-Smirnov normality test using the Lillie Fors correction returns a similar p-value of significantly less than the alpha value of 0.05.

Code: *R*

```r
# qq plot of BMI distribution
qplot(sample = PhysHlth, data = data.clean)+
labs(title="Q-Q PhysHlth Ratings",
       y = "Sample Quantile")+
  stat_qq() +
  stat_qq_line(colour = "red")
```




```
ggsave(path='./graphs',filename = '19_physhlth_qq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

A qq plot of the overall PhysHlth column returns a plot as expected similar to the MntHlth column and shows the data is not normally distributed

Code: *R*

```
#perform kolmogorov-smirnov test
ks.test(data.clean$PhysHlth[data.clean$Diabetes_binary == 1], 'pnorm')
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.clean$PhysHlth[data.clean$Diabetes_binary == 1]
## D = 0.5, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

The Kolmogorov-Smirnov normality test on the PhysHlth feature grouped by Diabetic people, again indicates that the data isn't normally distributed with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

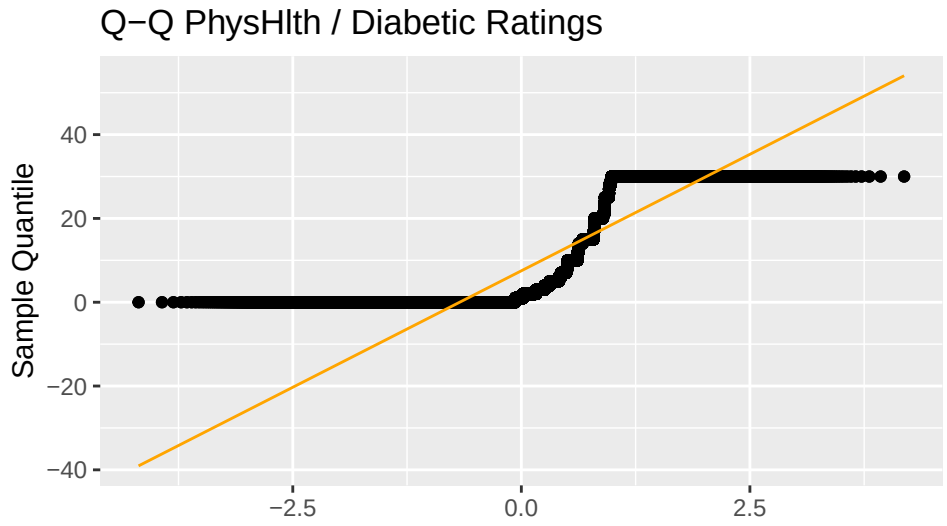
```
#perform shapiro-wilk test
lillie.test(data.clean$PhysHlth[data.clean$Diabetes_binary == 1])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.clean$PhysHlth[data.clean$Diabetes_binary == 1]
## D = 0.27051, p-value < 2.2e-16
```

And the Kolmogorov-Smirnov normality test using the Lillie Fors correction returns a similar p-value of significantly less than the alpha value of 0.05 when grouped by Diabetic people.

Code: *R*

```r
# qq plot of BMI distribution for Diabetic diagnosis
qplot(sample = PhysHlth, data = data.clean[data.clean$Diabetes_binary == 1,])+
labs(title="Q-Q PhysHlth / Diabetic Ratings",
       y = "Sample Quantile")+
  stat_qq() +
  stat_qq_line(colour = "orange")
```




```r
ggsave(path='./graphs',filename = '20_physhlth_dia_qq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

The qq plot of the grouped Diabetic people also shows a similar shape to the overall PhysHlth column qq plot previous.

Code: *R*

```r
#perform kolmogorov-smirnov test
ks.test(data.clean$PhysHlth[data.clean$Diabetes_binary == 0], 'pnorm')
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.clean$PhysHlth[data.clean$Diabetes_binary == 0]
## D = 0.5, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

The Kolmogorov-Smirnov normality test on the PhysHlth feature grouped by non-Diabetic people, again indicates that the data isn't normally distributed with a p-value of significantly less than the alpha value of 0.05.

Code: *R*

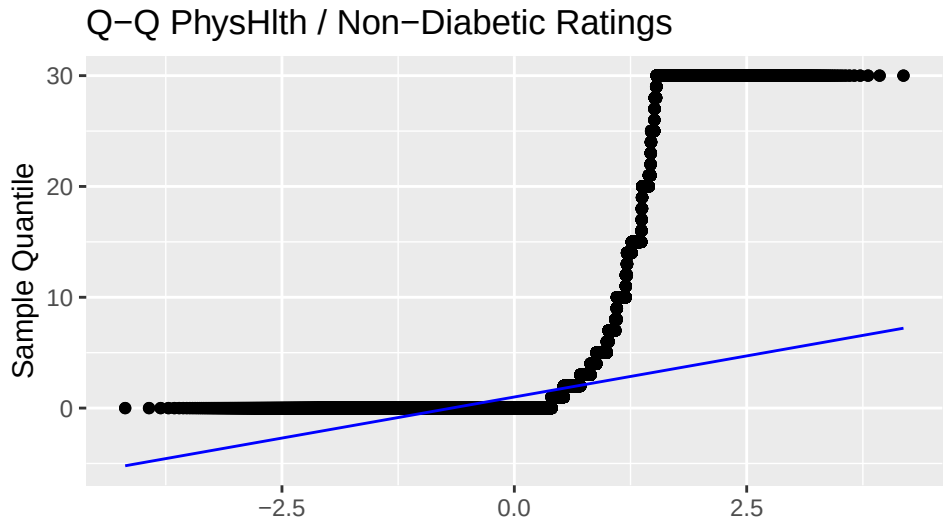
```r
#perform shapiro-wilk test
lillie.test(data.clean$PhysHlth[data.clean$Diabetes_binary == 0])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.clean$PhysHlth[data.clean$Diabetes_binary == 0]
## D = 0.34296, p-value < 2.2e-16
```

And the Kolmogorov-Smirnov normality test using the Lillie Fors correction returns a similar p-value of significantly less than the alpha value of 0.05 when grouped by non-Diabetic people.

Code: *R*

```
# qq plot of BMI distribution for Non-Diabetic diagnosis
qplot(sample = PhysHlth, data = data.clean[data.clean$Diabetes_binary == 0,])+
labs(title="Q-Q PhysHlth / Non-Diabetic Ratings",
       y = "Sample Quantile")+
  stat_qq() +
  stat_qq_line(colour = "blue")
```




```
ggsave(path='./graphs',filename = '21_physhlth_non_dia_qq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

The qq plot of the grouped non-Diabetic people also shows a similar shape to the overall PhysHlth column qq plot and the Diabetic groups previous.

Code: *R*

```
m <- mean(data.clean$PhysHlth)
md <- median(data.clean$PhysHlth)
s <- sd(data.clean$PhysHlth)
v <- var(data.clean$PhysHlth)

sprintf("Mean: %f Median:%f SD: %f Var: %f", m, md, s, v)
```

```
## [1] "Mean: 5.810417 Median:0.000000 SD: 10.062261 Var: 101.249087"
```

The mean days for PhysHlth is slightly higher than the MntHlth at 5.8 with a median of zero, the standard deviation is higher than the MnthHlth feature at 10 days with a variance of 101.24

HighBP - logical factor

The High Blood Pressure factor in the CDC survey is from the question, Adults who have been told they have high blood pressure by a doctor, nurse, or other health professional with a TRUE or FALSE value.

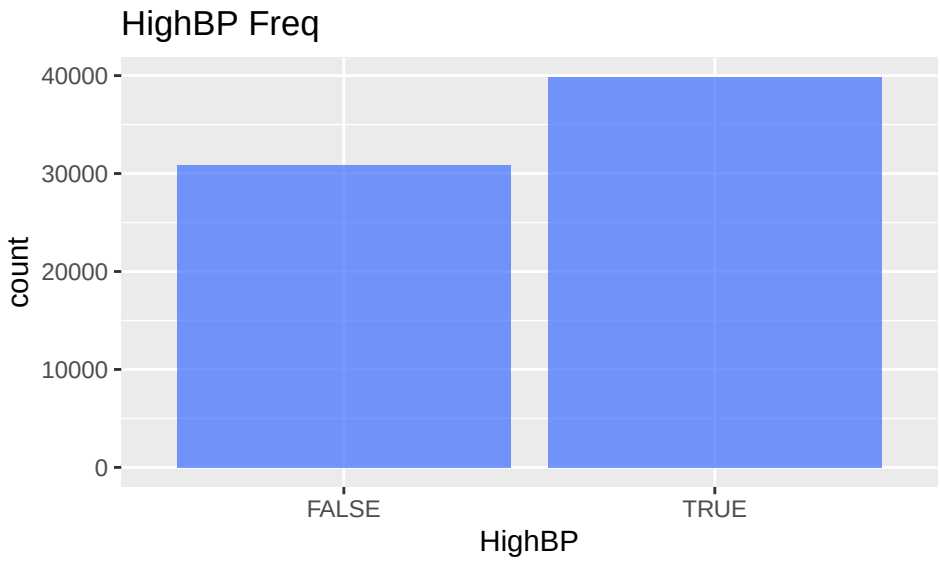
This can sometimes be an indicator or many different possible medical diagnosis not just diabetes but for the purposes of this analysis it will be checked how much of an impact this can have on predicting the possibility of Diabetes.

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(HighBP)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("HighBP Freq")
```



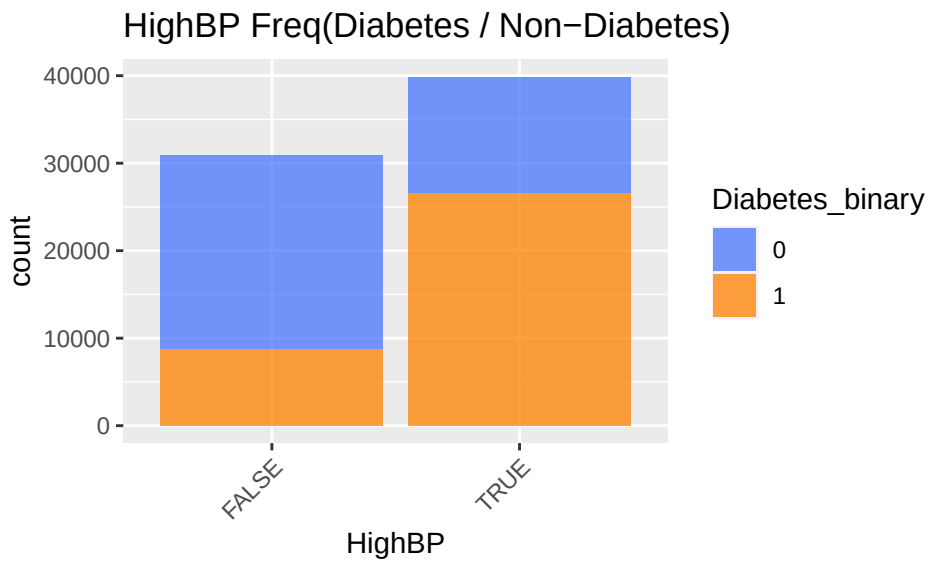

```
ggsave(path='./graphs',filename = '22_highBP_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

The data shows that from the survey there is a higher number of people with high blood pressure.

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = HighBP, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("HighBP Freq(Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '23_highBP_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

When adding the diabetic groups to the bar chart, there appears to be a higher percentage of people diagnised with diabetes that also have high blood pressure.

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$HighBP))*100)
names(table) <- c("HighBP", "Percent")
table
```

```
##   HighBP  Percent
## 1  FALSE 43.65416
## 2   TRUE 56.34584
```

Confirming the results of the blood pressure bar chart, 56.35% of people report having high blood pressure versus 43.65% of people not having high blood pressure.

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + HighBP, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary HighBP  Freq Percent
## 1               0  FALSE 22118   62.58
## 2               1  FALSE  8742   24.73
## 3               0   TRUE 13228   37.42
## 4               1   TRUE 26604   75.27
```

From the high blood pressure feature, 75.27% of people from the survey who are diagnosed with Diabetes also have high blood pressure, where as only 37.42% of the people that are diagnosed as not having Diabetes have high blood pressure.

HighChol - logical factor

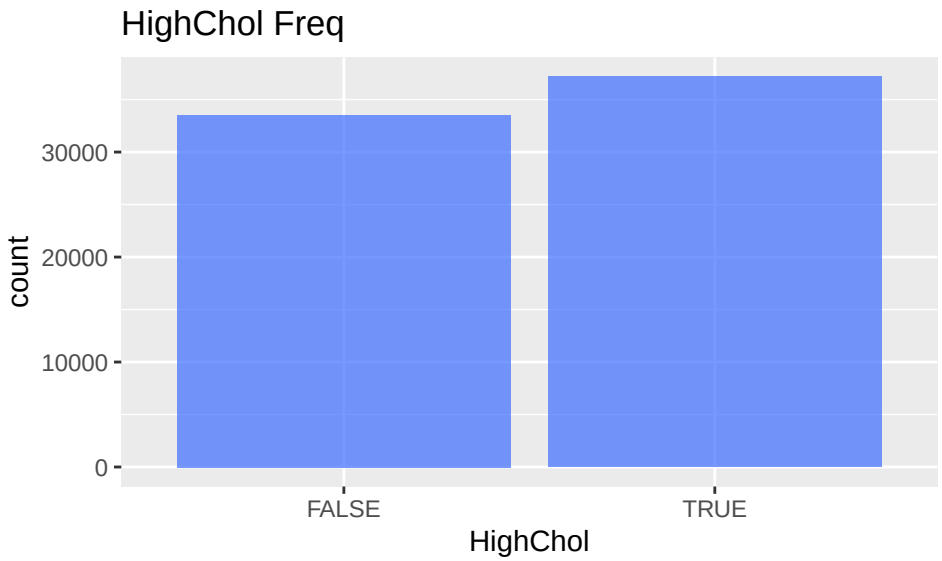
The question from the survey is - Have you EVER been told by a doctor, nurse or other health professional that your blood cholesterol is high.

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(HighChol)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("HighChol Freq")
```

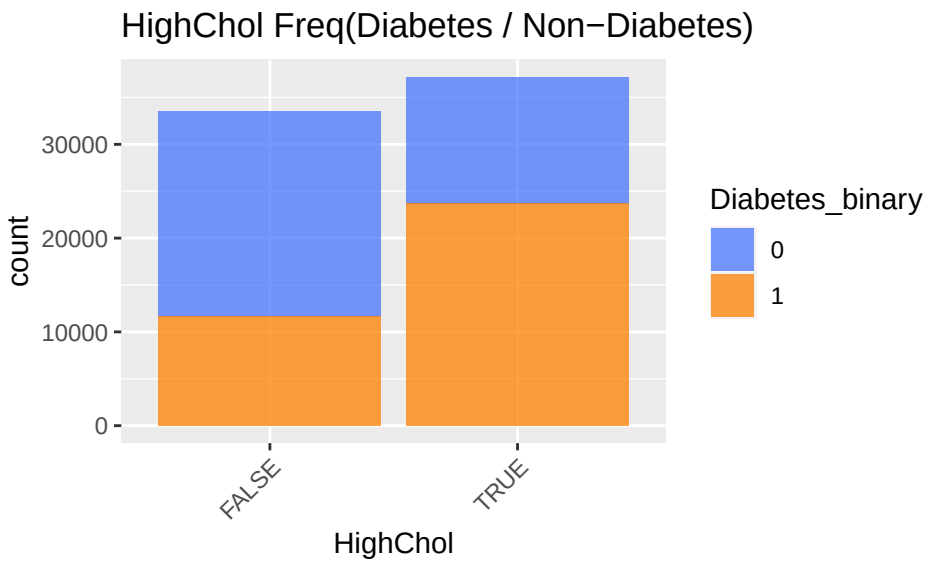



```
ggsave(path='./graphs',filename = '24_highChol_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = HighChol, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("HighChol Freq(Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '25_highChol_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$HighChol))*100)
names(table) <- c("HighChol", "Percent")
table
```

```
##   HighChol Percent
## 1    FALSE 47.4297
## 2     TRUE 52.5703
```

There is an approximate even split between people who have reported ever having high cholesterol versus people who have not with 52.57% True and 47.43% False.

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + HighChol, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary HighChol  Freq Percent
## 1               0    FALSE 21869   61.87
## 2               1    FALSE 11660   32.99
## 3               0     TRUE 13477   38.13
## 4               1     TRUE 23686   67.01
```

From the diabetic group there is 67.01% that have reported high cholesterol versus only 32.99% not having high cholesterol. This appears to be a significant difference and could be heavy determining factor in diagnosing diabetes.

CholCheck - logical factor

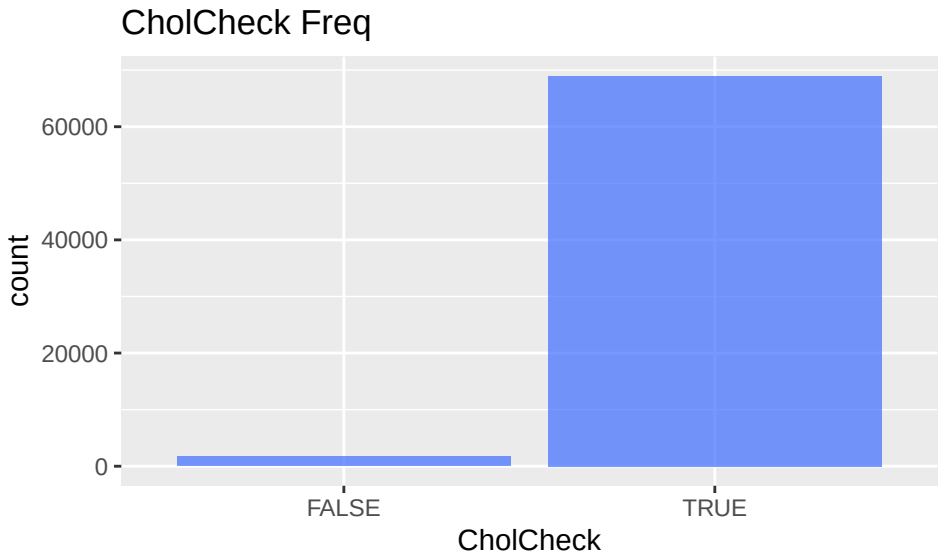
The question in the survey for this factor is - Cholesterol check within past five years. Having a high percentage of TRUE in this factor would lead to the assumption that the results from the previous column are accurate and noteworthy.

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(CholCheck)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("CholCheck Freq")
```

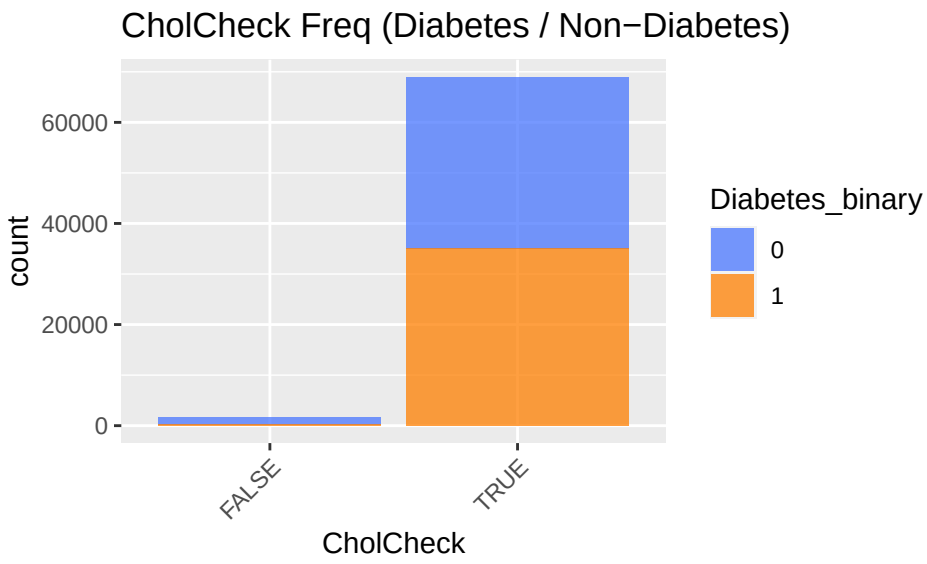



```
ggsave(path='./graphs',filename = '26_cholCheck_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = CholCheck, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("CholCheck Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '27_cholCheck_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$CholCheck))*100)
names(table) <- c("CholCheck", "Percent")
table
```

```
##   CholCheck   Percent
## 1     FALSE  2.474113
## 2      TRUE 97.525887
```

97.53% of the respondents have had there cholesterol checked in the previous five years.

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + CholCheck, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary CholCheck  Freq Percent
## 1               0     FALSE  1508    4.27
## 2               1     FALSE   241    0.68
## 3               0      TRUE 33838   95.73
## 4               1      TRUE 35105   99.32
```

Seen as 97.53% of respondents have had their cholesterol checked it is a good indicator that the high cholesterol column has valid data.

It also shows that both 99.32% of people with Diabetes and 95.73% of people without diabetes have had their cholesterol checked in the previous five years.

Smoker - 2 level factor

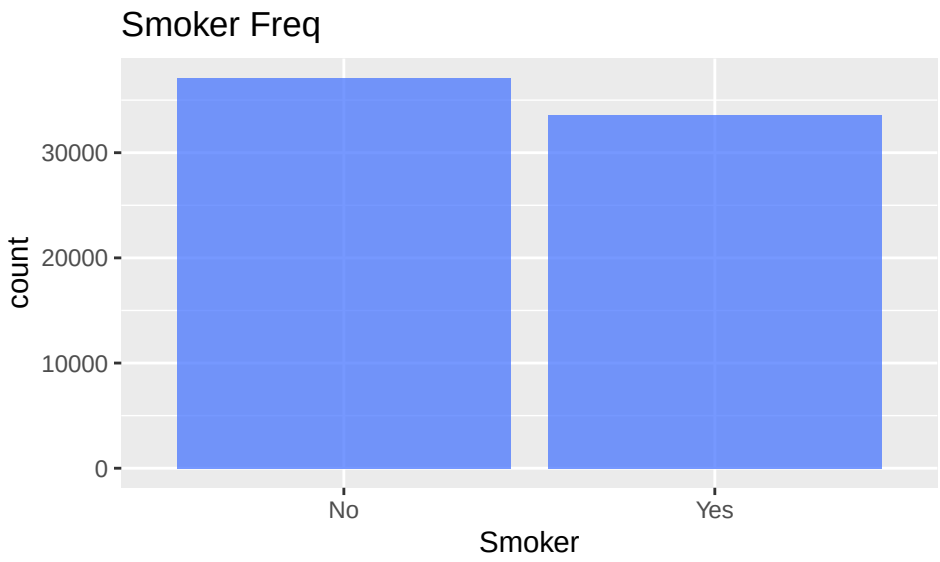
The question from the survey asked - Have you smoked at least 100 cigarettes in your entire life? this question doesn't take into account when someone smoked or if they did, how long ago did they quit, or how many over the 100 mark did they smoke.

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(Smoker)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("Smoker Freq")
```

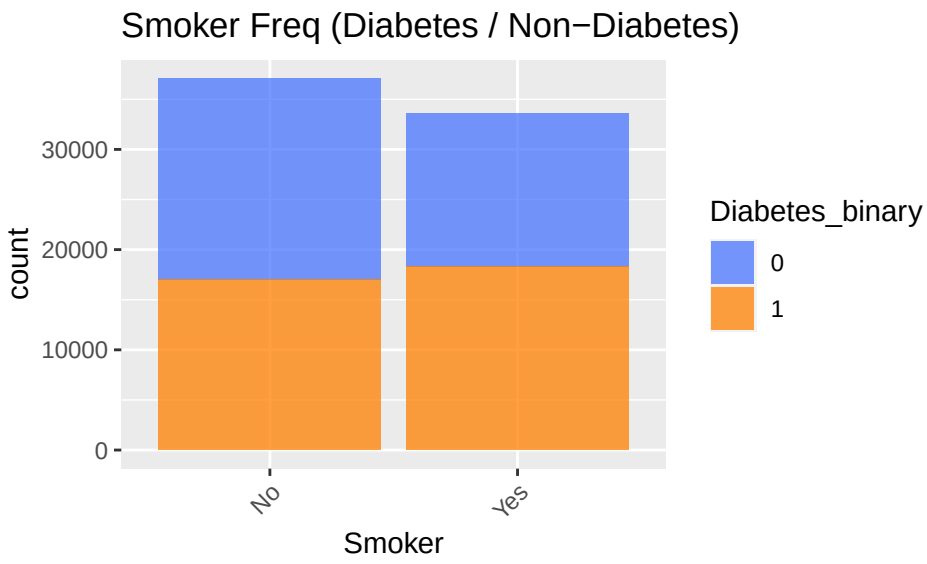



```
ggsave(path='./graphs',filename = '28_smoker_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = Smoker, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("Smoker Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '29_smoker_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$Smoker))*100)
names(table) <- c("Smoker", "Percent")
table
```

```
##   Smoker Percent
## 1     No 52.4727
## 2    Yes 47.5273
```

The appears an approximate even split in the survey of people who have or have no smoked with 54.47% not having smoked and 47.53% having smoked. it is yet to be determined if this split is statistically significant.

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + Smoker, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary Smoker  Freq Percent
## 1               0     No 20065   56.77
## 2               1     No 17029   48.18
## 3               0    Yes 15281   43.23
## 4               1    Yes 18317   51.82
```

For the cohort that has a diabetes diagnosis 51.82% of people have smoke and 48.18% of people haven't smoked more than 100 cigarettes.

Stroke - 2 level factor

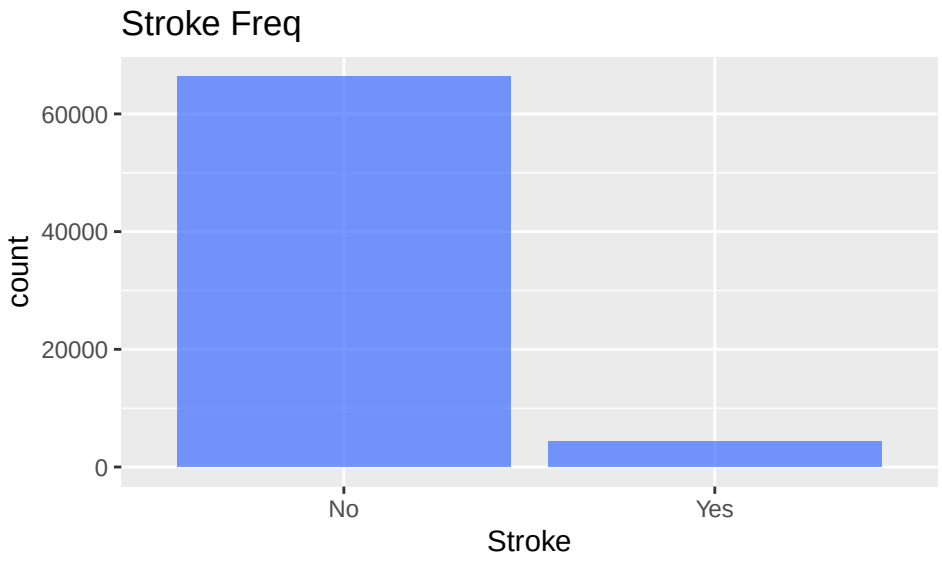
The survey question asks - (Ever told) you had a stroke. There has been studies to argue the hypothesis that diabetes can be a risk factor for stroke but not the other way around.link

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(Stroke)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("Stroke Freq")
```

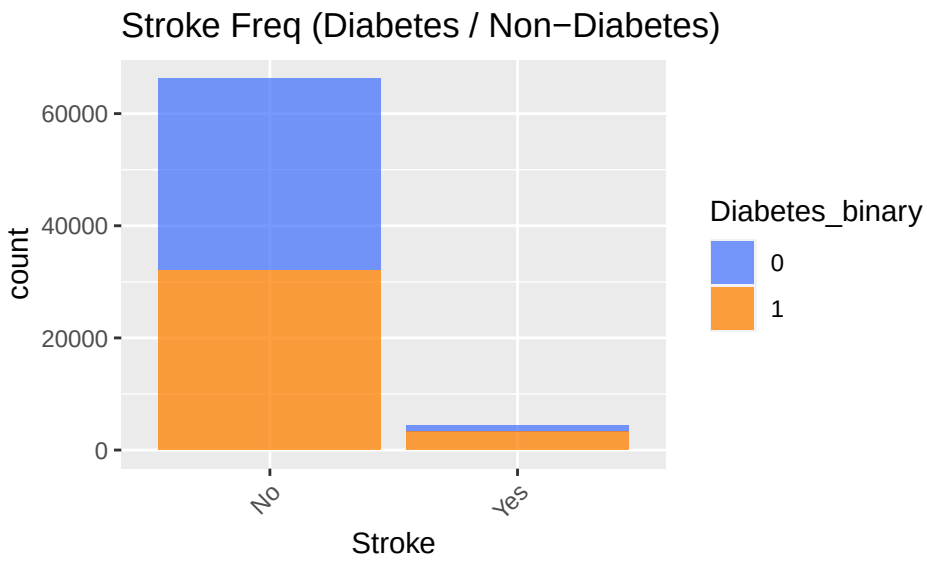



```
ggsave(path='./graphs',filename = '30_stroke_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = Stroke, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("Stroke Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '31_stroke_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$Stroke))*100)
names(table) <- c("Stroke", "Percent")
table
```

```
##   Stroke   Percent
## 1     No 93.782889
## 2    Yes  6.217111
```

Only 6.22% of the study respondents reported having a been told they have had a stroke.

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + Stroke, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary Stroke  Freq Percent
## 1               0     No 34219   96.81
## 2               1     No 32078   90.75
## 3               0    Yes  1127    3.19
## 4               1    Yes  3268    9.25
```

Of the people diagnosed with Diabetes in the study only 9.25% of people had been told they have had a stroke but from the number of people who have been told they had a stroke 74.36% of people where Diabetes sufferers.

There is no indication in the survey though when the patient had the stroke, if it was before or after Diabetes diagnosis or if Diabetes was a risk a factor in that stroke. The percent does indicate that there is some correlation between the two.

HeartDiseaseorAttack - 2 level factor

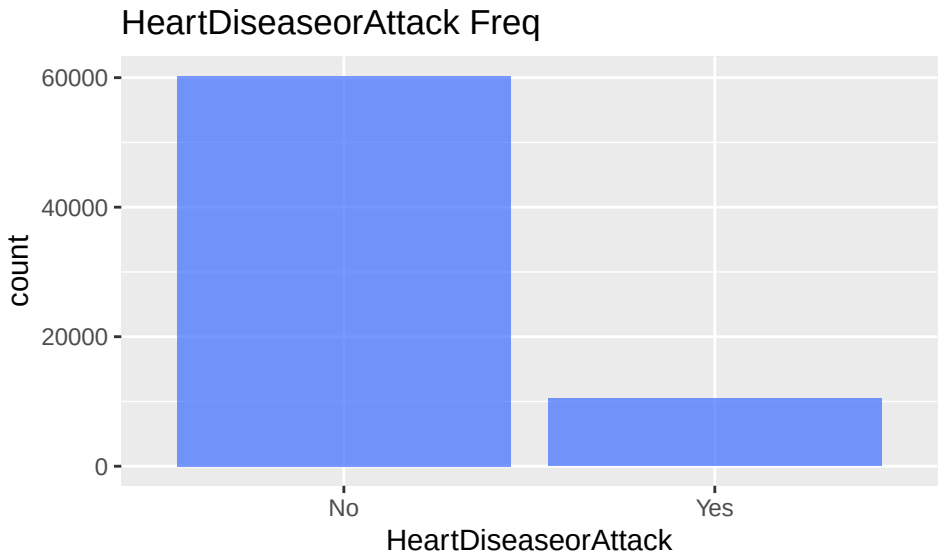
The survey question asked for this column was - Have ever reported having coronary heart disease (CHD) or myocardial infarction(MI). There are papers that show an increased risk of heart disease with diabetes link but can it show the reverse.

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(HeartDiseaseorAttack)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("HeartDiseaseorAttack Freq")
```

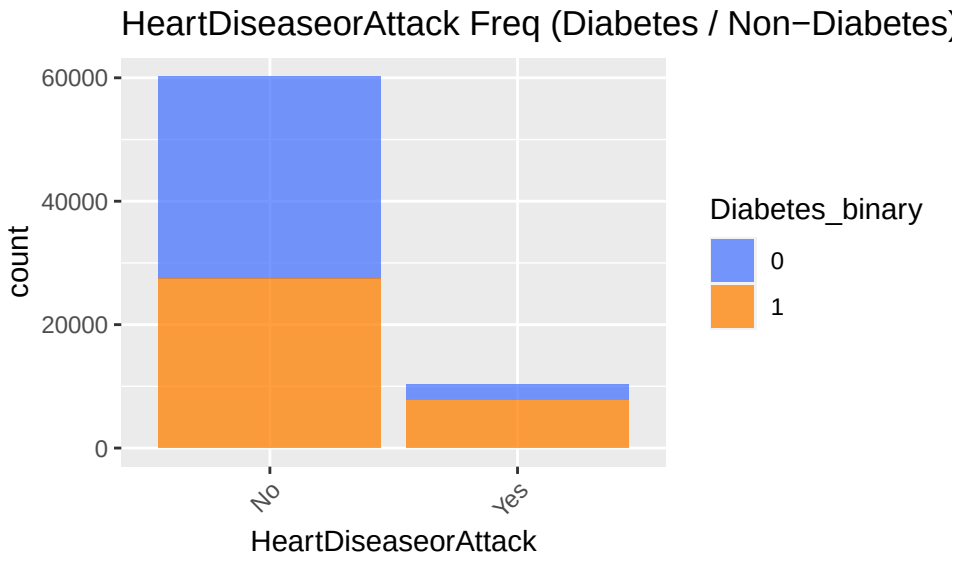



```
ggsave(path='./graphs',filename = '32_heartDis_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = HeartDiseaseorAttack, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("HeartDiseaseorAttack Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '33_heartDis_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$HeartDiseaseorAttack))*100)
names(table) <- c("HeartDiseaseorAttack", "Percent")
table
```

```
##   HeartDiseaseorAttack  Percent
## 1                   No 85.21898
## 2                  Yes 14.78102
```

14.78% of people in the survey reported having heart disease of some kind.

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + HeartDiseaseorAttack, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary HeartDiseaseorAttack  Freq Percent
## 1               0                   No 32775   92.73
## 2               1                   No 27468   77.71
## 3               0                  Yes  2571    7.27
## 4               1                  Yes  7878   22.29
```

From the 14.78% of people with some type of heart disease, 75.69% of those people have been diagnosed with Diabetes.

PhysActivity - logical factor

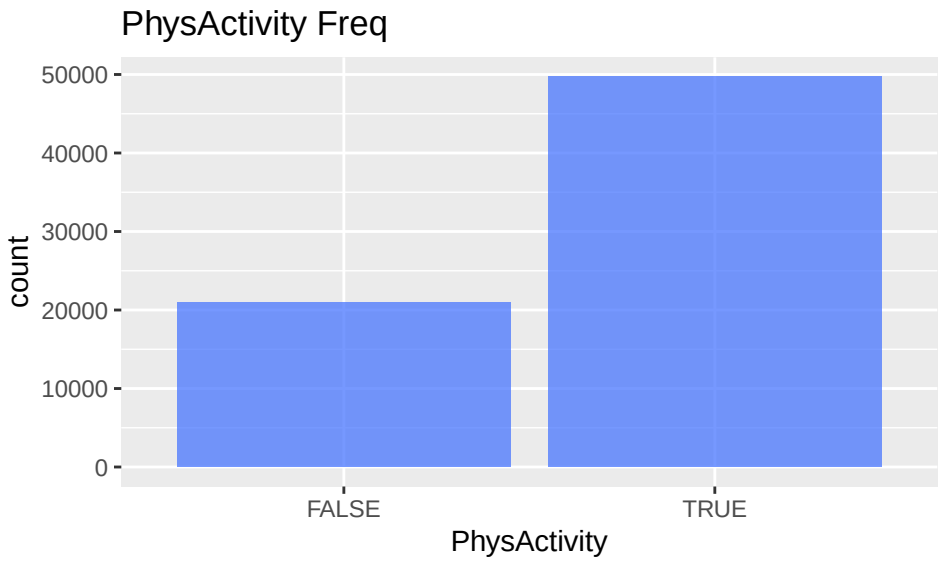
The survey question for this column was - Adults who reported doing physical activity or exercise during the past 30 days other than their regular job. Study link

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(PhysActivity)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("PhysActivity Freq")
```

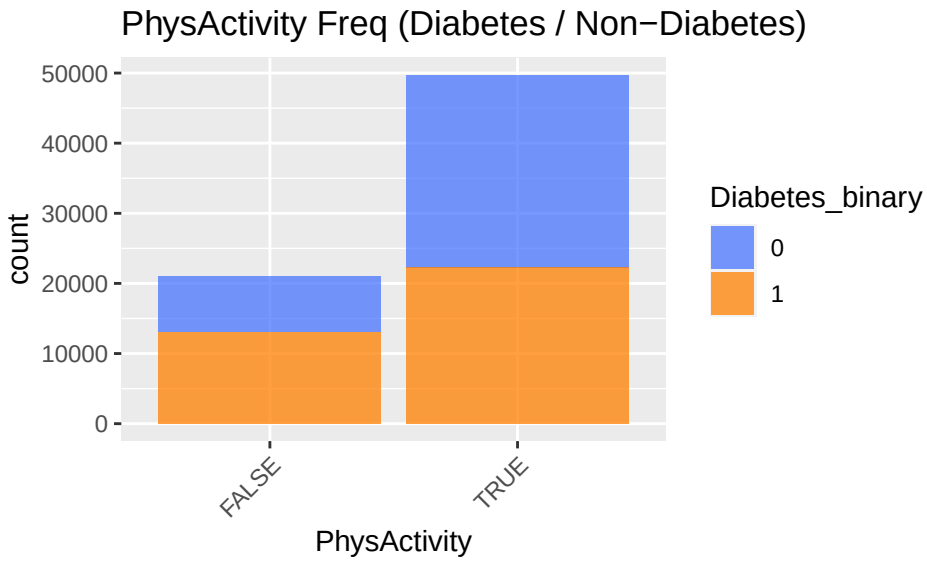



```
ggsave(path='./graphs',filename = '34_physactive_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = PhysActivity, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("PhysActivity Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '35_physactive_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$PhysActivity))*100)
names(table) <- c("PhysActivity", "Percent")
table
```

```
##   PhysActivity  Percent
## 1        FALSE 29.69643
## 2         TRUE 70.30357
```

29.70% of the people in the survey report not having done any physical activity outside their normal job in the previous 30 days.

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + PhysActivity, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary PhysActivity  Freq Percent
## 1               0        FALSE  7934   22.45
## 2               1        FALSE 13059   36.95
## 3               0         TRUE 27412   77.55
## 4               1         TRUE 22287   63.05
```

62.20% of those who have not done any physical activity are from the Diabetes diagnosed cohort. From the Diabetes cohort as a whole 63.05% had completed physical activity while 36.95% hadn't but for the non-diabetic people 77.55% had and only 22.45% hadn't. Al thought this measure could be biased dependent on a persons perspective of physical activity it does indicate that Diabetic people have a higher amount of inactivity than non-diabetic people.

Fruits - logical factor

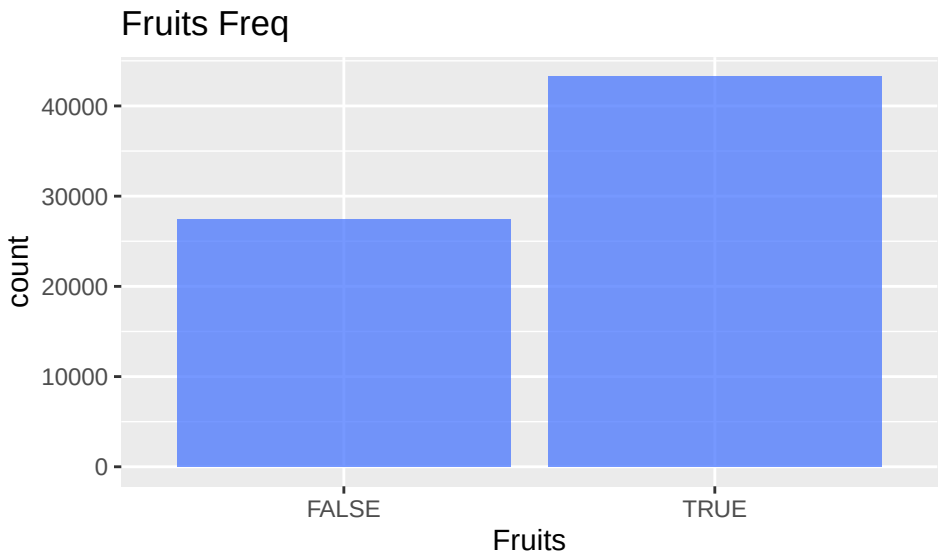
The study question asked - Consume Fruit 1 or more times per day. Study on Diet: link

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(Fruits)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("Fruits Freq")
```

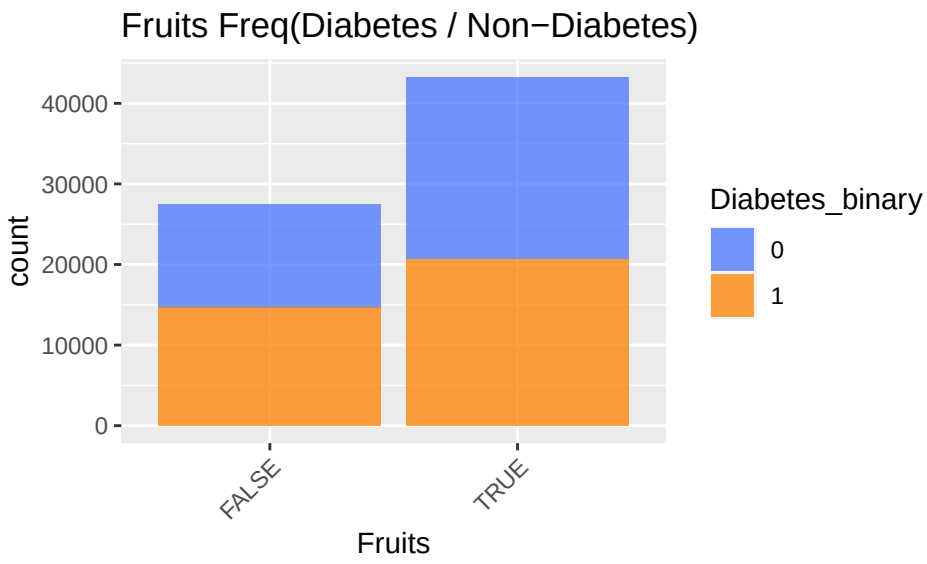



```
ggsave(path='./graphs',filename = '36_fruits_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = Fruits, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("Fruits Freq(Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '37_fruits_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$Fruits))*100)
names(table) <- c("Fruits", "Percent")
table
```

```
##   Fruits  Percent
## 1  FALSE 38.82052
## 2   TRUE 61.17948
```

61.17% of people reported eating 1 or more fruits per day

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + Fruits, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary Fruits  Freq Percent
## 1               0  FALSE 12790   36.19
## 2               1  FALSE 14653   41.46
## 3               0   TRUE 22556   63.81
## 4               1   TRUE 20693   58.54
```

For the diabetic group 58.54% reported eating fruit per day compared to the non diabetic group where 63.81% reported eating one or more fruits per day.

Veggies - logical factor

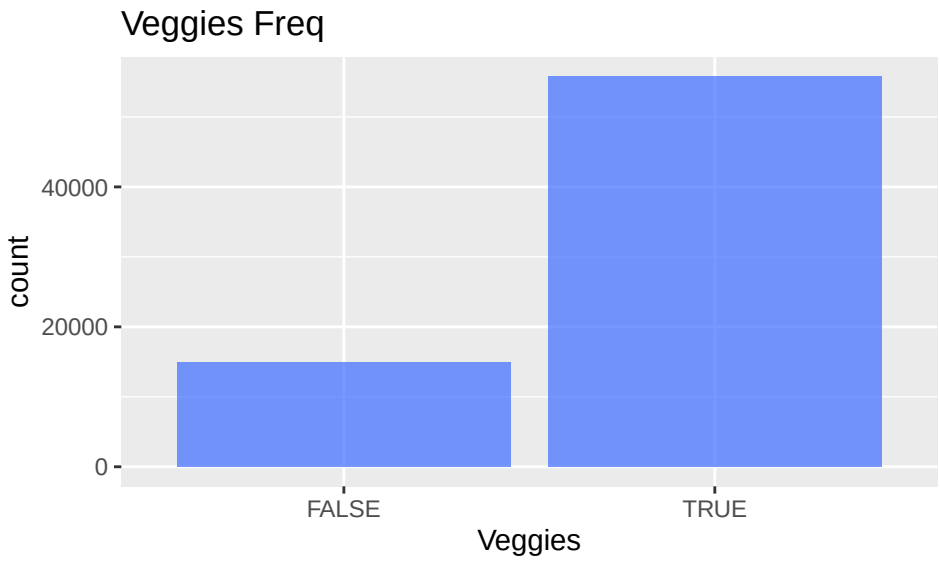
The study question for this column was - Consume Vegetables 1 or more times per day

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(Veggies)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("Veggies Freq")
```

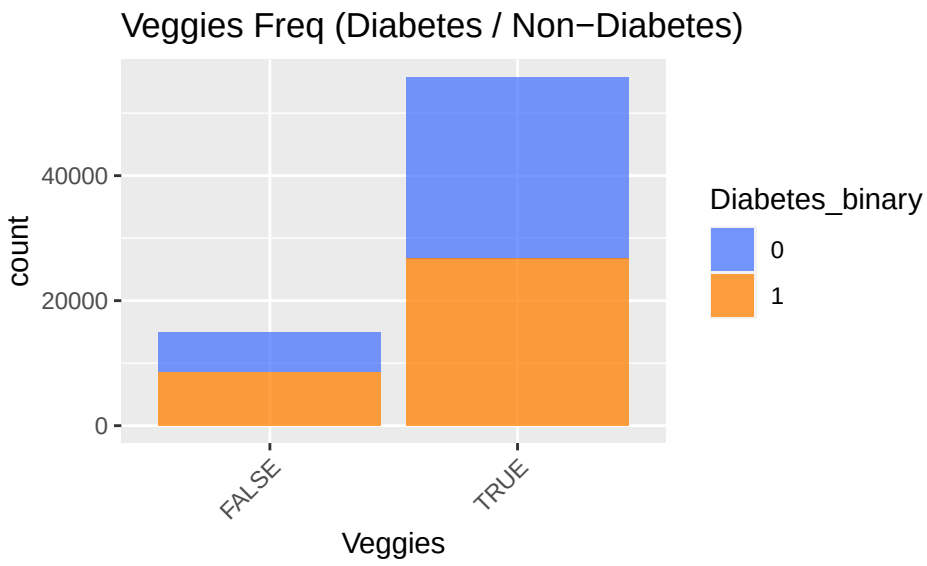



```
ggsave(path='./graphs',filename = '38_veggies_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = Veggies, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("Veggies Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '39_veggies_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$Veggies))*100)
names(table) <- c("Veggies", "Percent")
table
```

```
##   Veggies  Percent
## 1   FALSE 21.12262
## 2    TRUE 78.87738
```

78.87% of people reported eating 1 more vegatables a day compared to 21.13% not eating vegatables.

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + Veggies, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary Veggies  Freq Percent
## 1               0   FALSE  6322   17.89
## 2               1   FALSE  8610   24.36
## 3               0    TRUE 29024   82.11
## 4               1    TRUE 26736   75.64
```

Of the diabetic group 75.64% reported eating vegetables compared to 82.11% from the non diabetic, From the the group that reported to not eat 1 or more vegetables a day 57.66% of them where diabetic.

HvyAlcoholConsump - logical factor

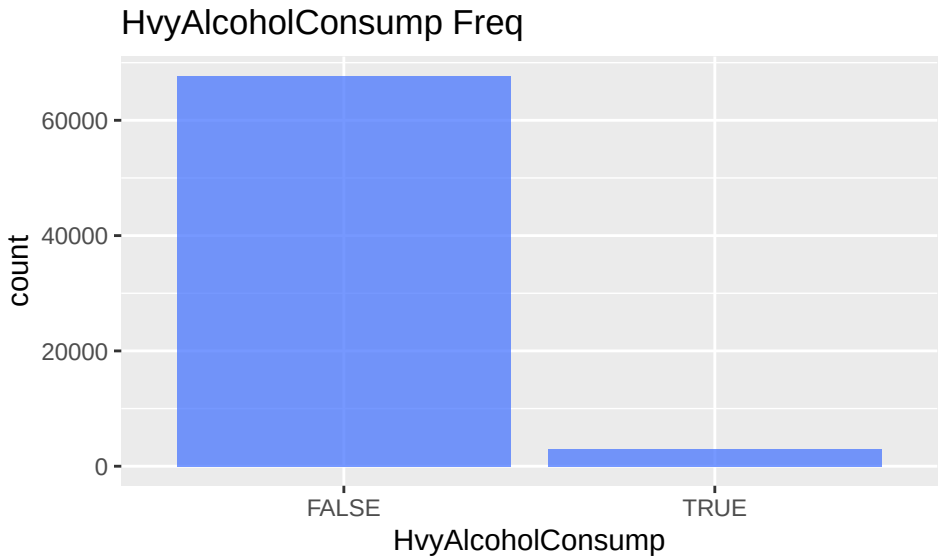
For this column the question in the survey was around the number of drinks per week from adults - Heavy drinkers (adult men having more than 14 drinks per week and adult women having more than 7 drinks per week), resulting in a boolean output.

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(HvyAlcoholConsump)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("HvyAlcoholConsump Freq")
```

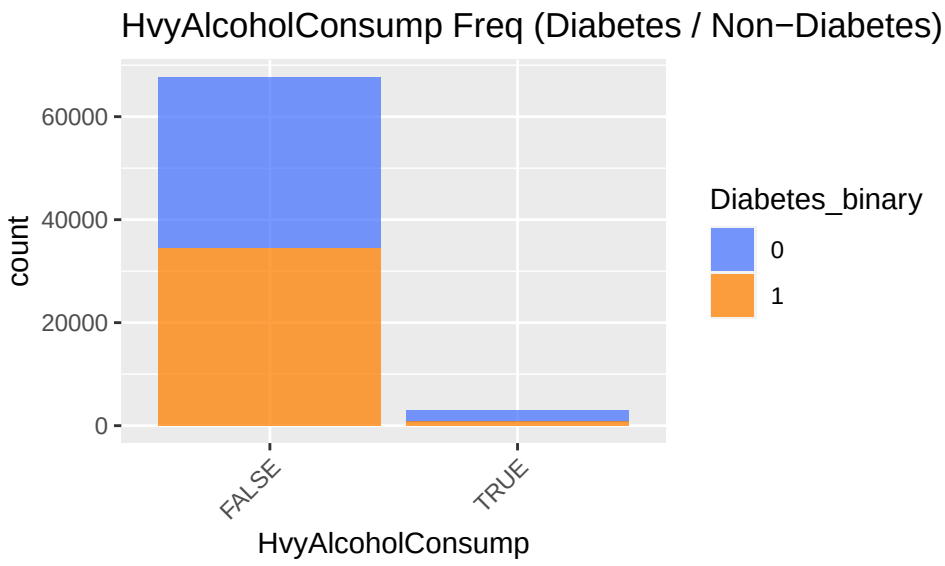



```
ggsave(path='./graphs',filename = '40_hvyAlcho_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = HvyAlcoholConsump, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("HvyAlcoholConsump Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '41_hvyAlcho_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$HvyAlcoholConsump))*100)
names(table) <- c("HvyAlcoholConsump", "Percent")
table
```

```
##   HvyAlcoholConsump   Percent
## 1             FALSE 95.727947
## 2              TRUE  4.272053
```

95.72% of repsondents had no signs of heavy alchohol drinking.

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + HvyAlcoholConsump, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary HvyAlcoholConsump  Freq Percent
## 1               0             FALSE 33158   93.81
## 2               1             FALSE 34514   97.65
## 3               0              TRUE  2188    6.19
## 4               1              TRUE   832    2.35
```

Heavy alcohol appears more prevalent in people without diabetes where 97.65% of the diabetic group did not show signs of heavy alcohol consumption compared to 93.81% of people without diabetes. This column may impact any modelling where a person who does not consume alcohol could indicate diabetes where it could be due to a diet restriction after being diagnosed as diabetic.

AnyHealthcare - 2 level factor

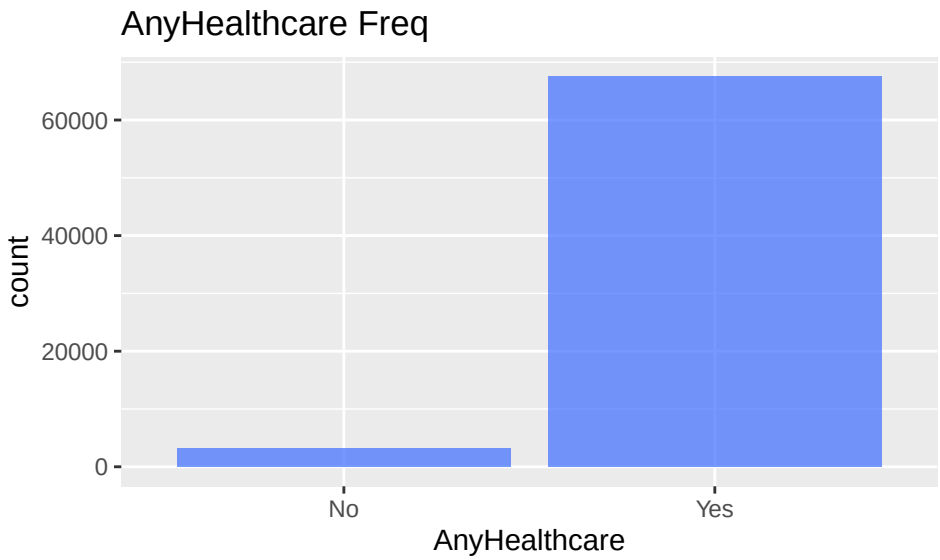
The question for this column in the survey was - Do you have any kind of health care coverage

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(AnyHealthcare)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("AnyHealthcare Freq")
```

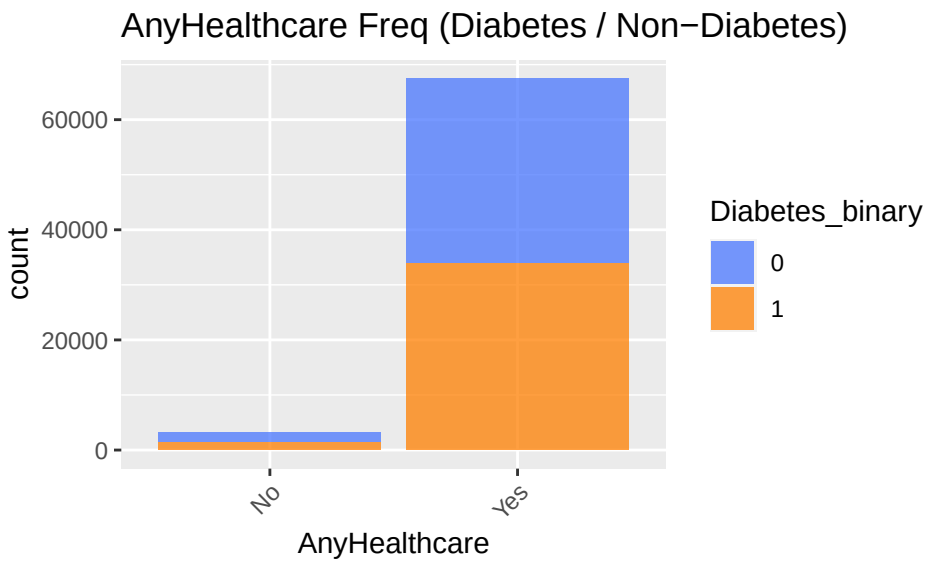



```
ggsave(path='./graphs',filename = '42_hlthCare_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = AnyHealthcare, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("AnyHealthcare Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '43_hvyAlcho_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$AnyHealthcare))*100)
names(table) <- c("AnyHealthcare", "Percent")
table
```

```
##   AnyHealthcare   Percent
## 1            No  4.504046
## 2           Yes 95.495954
```

95.49% of people reported havng some kind of healthcare

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + AnyHealthcare, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary AnyHealthcare  Freq Percent
## 1               0            No  1762    4.99
## 2               1            No  1422    4.02
## 3               0           Yes 33584   95.01
## 4               1           Yes 33924   95.98
```

There appears to be no major difference in the number of people with or without diabetes reporting to have some kind of healthcare, this could be a cultural aspect as this survey was conducted in the United States and may not have any real impact on any modelling.

NoDocbcCost - 2 level factor

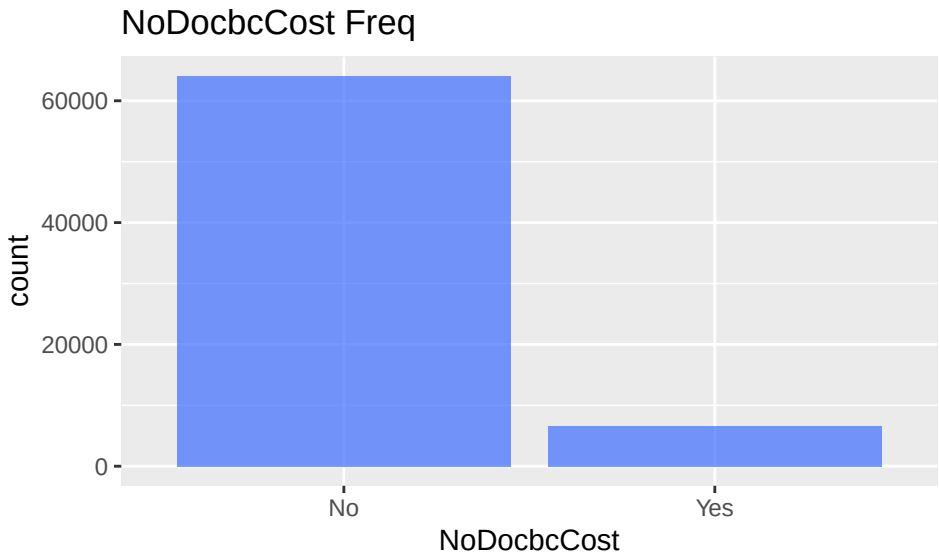
The survey question for this columns was - Was there a time in the past 12 months when you needed to see a doctor but could not because of cost?, Due to the high number of people with healthcare in the previous column it would be expected that the majority of people would answer no to this question.

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(NoDocbcCost)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("NoDocbcCost Freq")
```

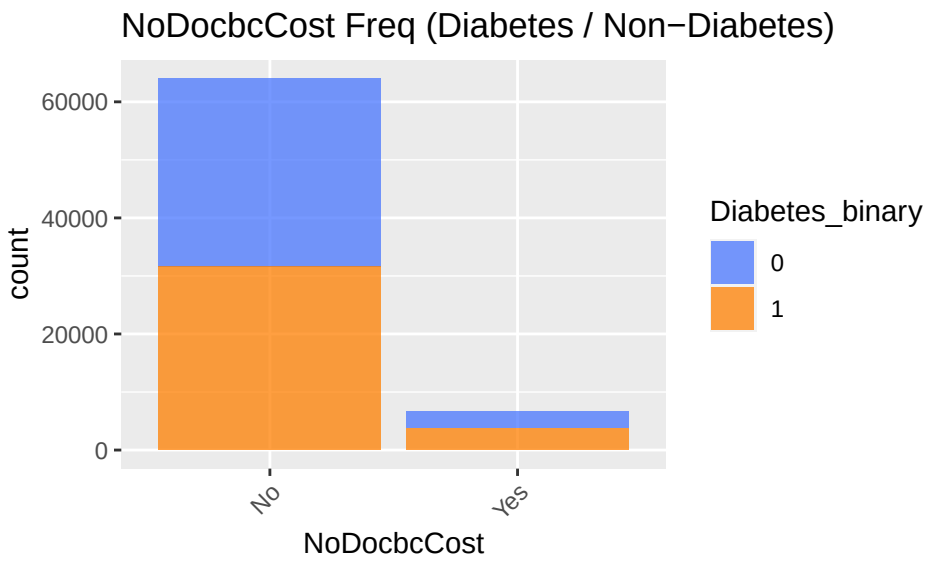



```
ggsave(path='./graphs',filename = '44_noDocCost_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = NoDocbcCost, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("NoDocbcCost Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '44_noDocCost_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$NoDocbcCost))*100)
names(table) <- c("NoDocbcCost", "Percent")
table
```

```
##   NoDocbcCost   Percent
## 1          No 90.608555
## 2         Yes  9.391445
```

As expected 90.60% of people reported no to this question

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + NoDocbcCost, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary NoDocbcCost  Freq Percent
## 1               0          No 32449   91.80
## 2               1          No 31604   89.41
## 3               0         Yes  2897    8.20
## 4               1         Yes  3742   10.59
```

Interestingly, from the group with Diabetes, 10.59% reported Yes where as only 8.20% reported Yes from the non-diabetic cohort.

GenHlth - 5 level factor

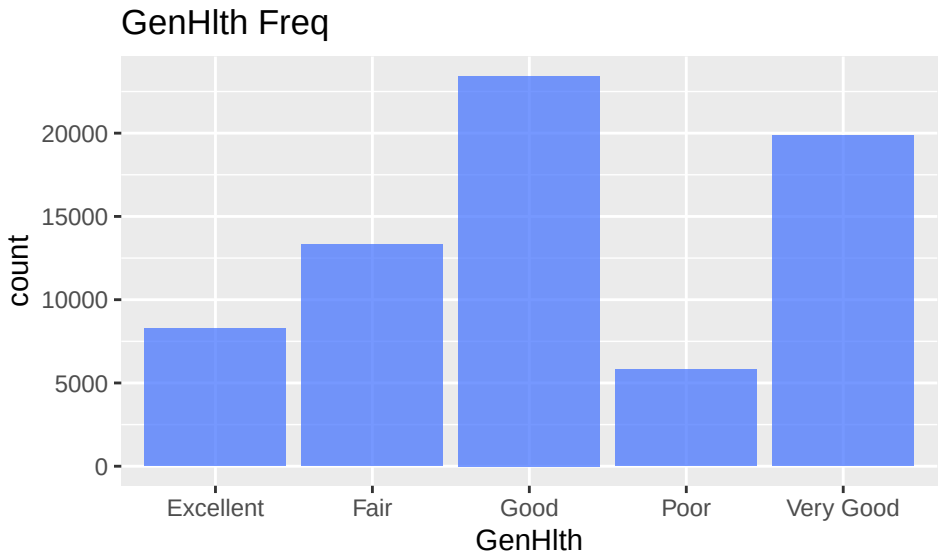
The survey question for this column was - Would you say that in general your health is: (asked to choose one of the options), this question like some previous questions could be biased as it depends on a persons perspective of their own health.

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(GenHlth)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("GenHlth Freq")
```

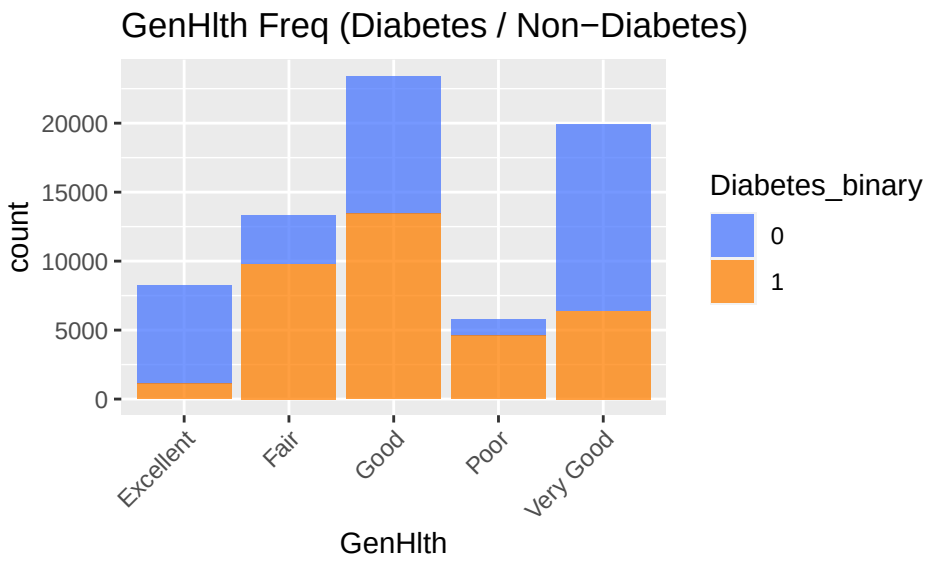



```
ggsave(path='./graphs',filename = '45_genHlth_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = GenHlth, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("GenHlth Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '46_genHlth_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$GenHlth))*100)
names(table) <- c("GenHlth", "Percent")
table
```

```
##      GenHlth   Percent
## 1 Excellent 11.715611
## 2      Fair 18.818254
## 3      Good 33.139535
## 4      Poor  8.215923
## 5 Very Good 28.110677
```

The results show that the largest groups where Good with 33.14% of the people and Very Good with 28.11% of the people.

Code: *R*

```
# creating x tab for python data
table2 <- xtabs(~ GenHlth + Diabetes_binary, data=data.clean)
table2 <- as.data.frame(table2)
data_py <- table2
```

Checking the percentages per group of Diabetic and non diabetic

Code: *Python*

```
import pandas as pd

df = pd.DataFrame(r.data_py)
```

```
# grouping by factor and getting percentages
df['Percentage'] = ((df['Freq'] / df.groupby(['GenHlth'])['Freq'].transform('sum'))*100).round(2)
df = df.sort_values('GenHlth', ascending=False)
df
```

```
##       GenHlth Diabetes_binary   Freq  Percentage
## 4   Very Good               0  13491       67.89
## 9   Very Good               1   6381       32.11
## 3        Poor               0   1230       21.18
## 8        Poor               1   4578       78.82
## 2        Good               0   9970       42.56
## 7        Good               1  13457       57.44
## 1        Fair               0   3513       26.41
## 6        Fair               1   9790       73.59
## 0   Excellent               0   7142       86.24
## 5   Excellent               1   1140       13.76
```

In the very good and excellent groups, diabetics represent 32.11% and 13.75% of the respondents where as in the good, fair and poor groups Diabetic people represent 57.44%, 73.59% and 78.82% of each group. This could be as a result of their diagnosis and could possibly be an indicator but could also lean toward a bias nature.

DiffWalk - 2 level factor

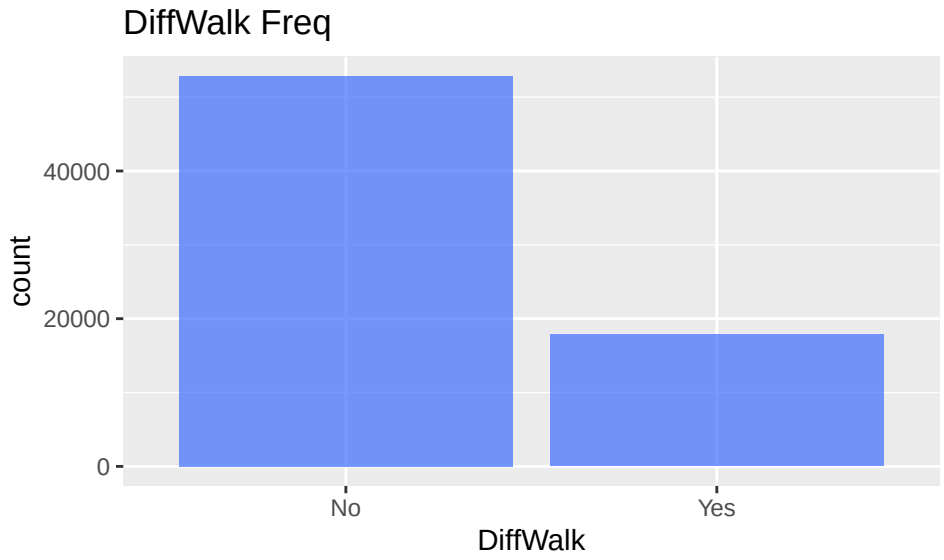
For this column the survey question asked - Do you have serious difficulty walking or climbing stairs?

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(DiffWalk)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("DiffWalk Freq")
```

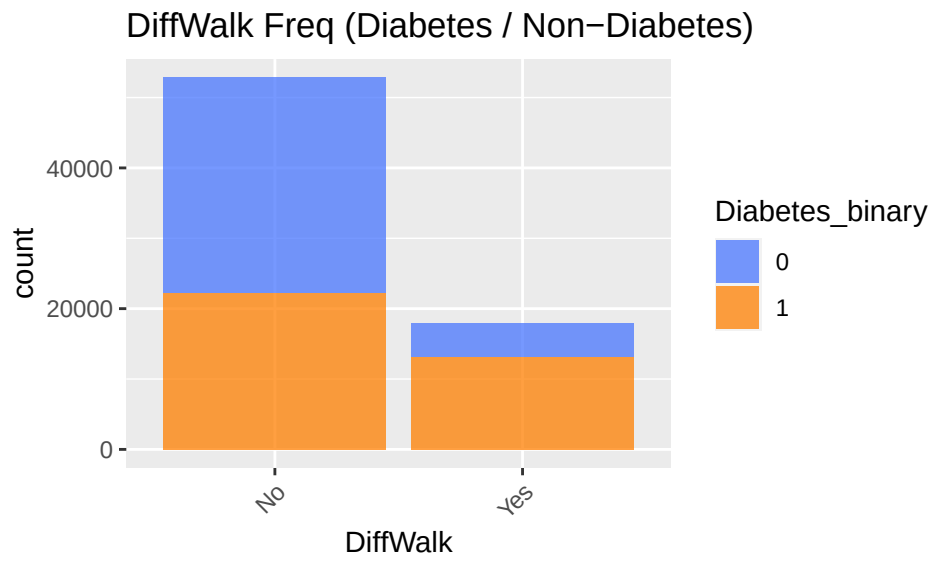


```
ggsave(path='./graphs',filename = '47_difWalk_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = DiffWalk, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("DiffWalk Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '48_difWalk_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$DiffWalk))*100)
names(table) <- c("DiffWalk", "Percent")
table
```

```
##   DiffWalk  Percent
## 1       No 74.72698
## 2      Yes 25.27302
```

74.72% of the people responded with no they have no difficulty, where 25.28% responded with yes.

Code: *R*

```r
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + DiffWalk, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary DiffWalk  Freq Percent
## 1               0       No 30601   86.58
## 2               1       No 22225   62.88
## 3               0      Yes  4745   13.42
## 4               1      Yes 13121   37.12
```

From the group that have a diagnosis of diabetes 37.12% reported having difficulty where only 13.42% of the non-diabetic people had difficulty. Again there is no initial indication whether this is a result of the diagnosis or not but the presence of a higher percentage with diabetes that do have difficulty could result in an indicator.

Sex - 2 level factor

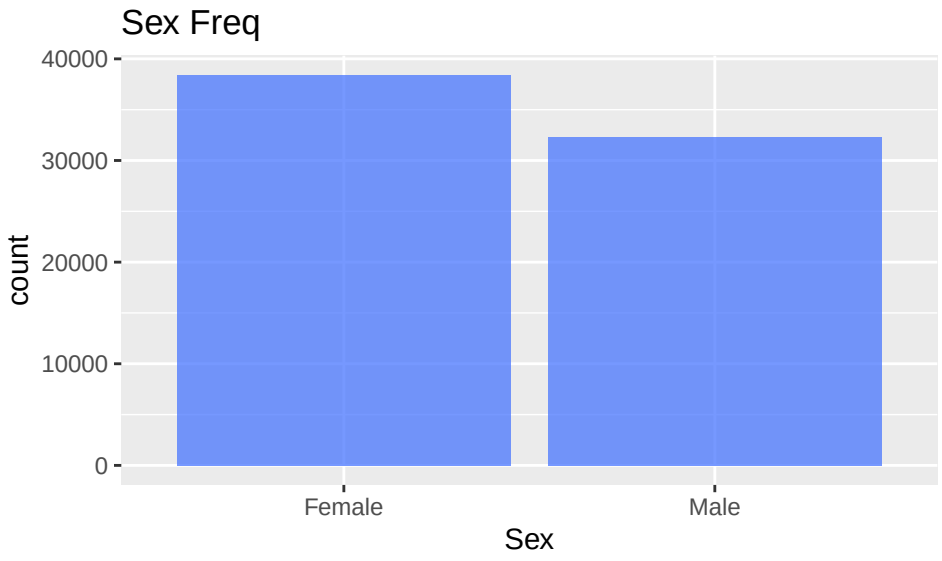
The survey question for this column simply asked the sex of the person.

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```r
# freq
ggplot(data.clean, aes(Sex)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("Sex Freq")
```

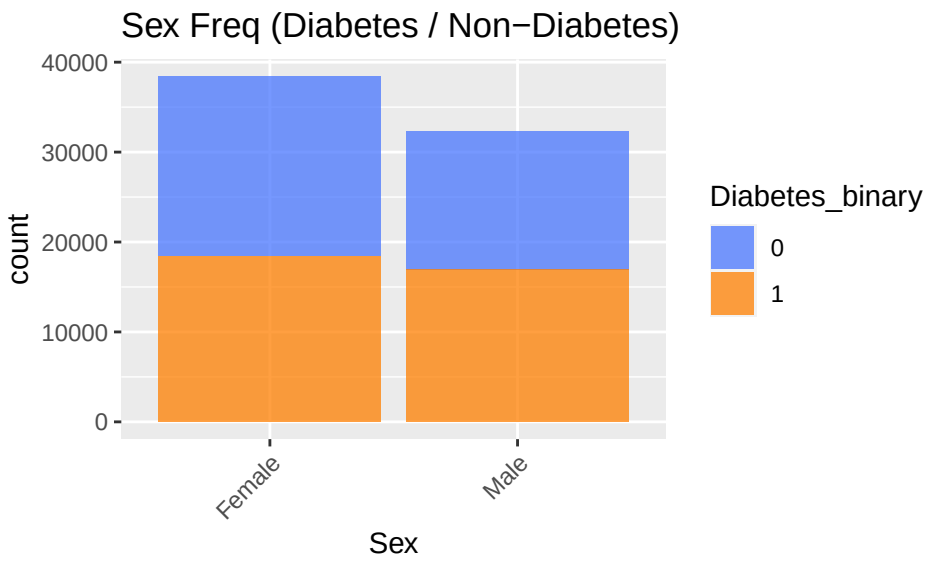



```r
ggsave(path='./graphs',filename = '49_sex_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = Sex, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("Sex Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '50_sex_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$Sex))*100)
names(table) <- c("Sex", "Percent")
table
```

```
##      Sex  Percent
## 1 Female 54.30035
## 2   Male 45.69965
```

54.30% of the people in the survey where female and 45.70% of the people where male.

Code: *R*

```
# checking percentages per group
table2 <- xtabs(~ Diabetes_binary + Sex, data=data.clean)
table2 <- as.data.frame(table2)
table2$Percent <- round(100*(table2$Freq/sum(table2$Freq[table2$Diabetes_binary==0])),digits=2)
table2
```

```
##   Diabetes_binary    Sex  Freq Percent
## 1               0 Female 19975   56.51
## 2               1 Female 18411   52.09
## 3               0   Male 15371   43.49
## 4               1   Male 16935   47.91
```

From the group of Diabetic patients 52.09% where female and 47.91% where male. This percentage could be due to the number of males and females in the study but does coincide with findings from this study link

Age - 13 level factor

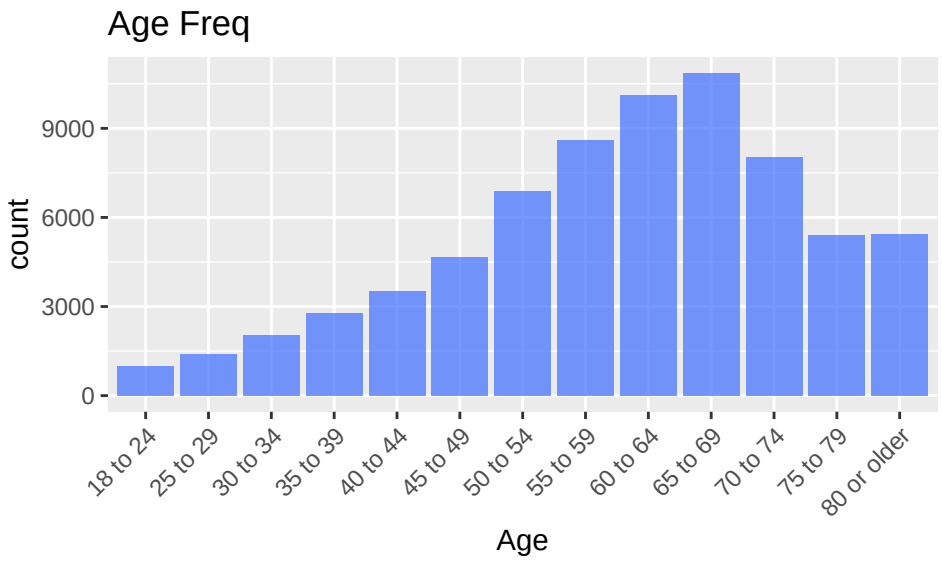
For this column people in the survey where asked their age and where put into one of 13 categories.1:Age 18 to 24,2:Age 25 to 29,3:Age 30 to 34,4:Age 35 to 39,5:Age 40 to 44,6:Age 45 to 49,7:Age 50 to 54,8:Age 55 to 59,9:Age 60 to 64,10:Age 65 to 69,11:Age 70 to 74,12:Age 75 to 79,13:Age 80 or older

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(Age)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  ggtitle("Age Freq")
```

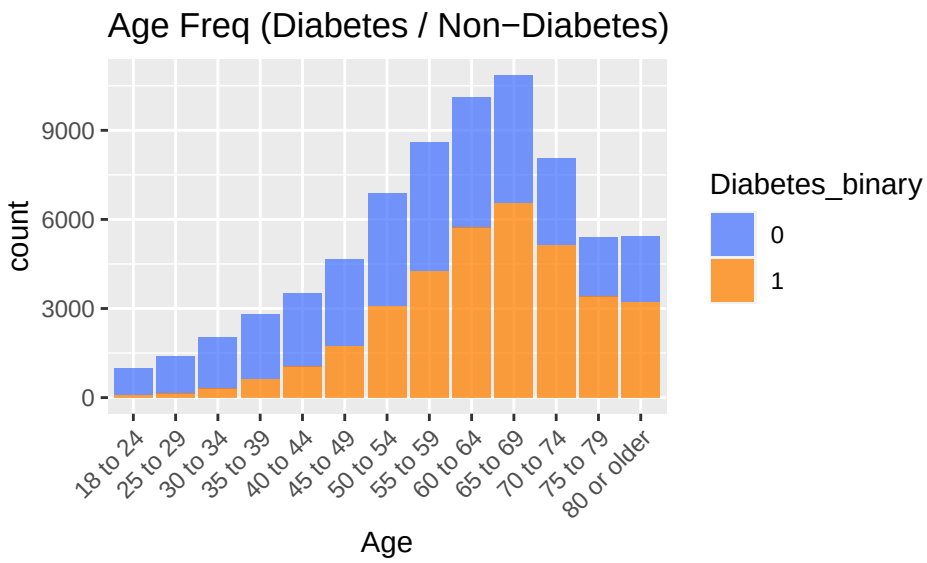



```
ggsave(path='./graphs',filename = '51_age_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```r
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = Age, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("Age Freq (Diabetes / Non-Diabetes)")
```




```r
ggsave(path='./graphs',filename = '51_age_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```r
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$Age))*100)
names(table) <- c("Age", "Percent")
table
```

```
##            Age   Percent
## 1     18 to 24  1.384881
## 2     25 to 29  1.974764
## 3     30 to 34  2.898489
## 4     35 to 39  3.950942
## 5     40 to 44  4.979347
## 6     45 to 49  6.575001
## 7     50 to 54  9.721043
## 8     55 to 59 12.169694
## 9     60 to 64 14.304306
## 10    65 to 69 15.356759
## 11    70 to 74 11.378940
## 12    75 to 79  7.630283
## 13 80 or older  7.675550
```

The majority of people in this study fell between the ages of 55 and 74. (53.21%)

Code: *R*

```r
# creating x tab for python data
table2 <- xtabs(~ Age + Diabetes_binary, data=data.clean)
table2 <- as.data.frame(table2)
data_py <- table2
```

Checking the percentages per group of Diabetic and non diabetic

Code: *Python*

```python
import pandas as pd

df = pd.DataFrame(r.data_py)

# grouping by factor and getting percentages
df['Percentage'] = ((df['Freq'] / df.groupby(['Age'])['Freq'].transform('sum'))*100).round(2)
df = df.sort_values('Age', ascending=False)
df
```

```
##             Age Diabetes_binary  Freq  Percentage
## 25  80 or older                1  3209       59.14
## 12  80 or older                0  2217       40.86
## 11     75 to 79                0  1991       36.91
## 24     75 to 79                1  3403       63.09
## 10     70 to 74                0  2903       36.09
## 23     70 to 74                1  5141       63.91
## 9      65 to 69                0  4298       39.59
## 22     65 to 69                1  6558       60.41
## 8      60 to 64                0  4379       43.30
## 21     60 to 64                1  5733       56.70
## 7      55 to 59                0  4340       50.45
## 20     55 to 59                1  4263       49.55
## 6      50 to 54                0  3784       55.06
## 19     50 to 54                1  3088       44.94
## 5      45 to 49                0  2906       62.52
## 18     45 to 49                1  1742       37.48
## 4      40 to 44                0  2469       70.14
## 17     40 to 44                1  1051       29.86
## 3      35 to 39                0  2167       77.59
## 16     35 to 39                1   626       22.41
## 2      30 to 34                0  1735       84.68
## 15     30 to 34                1   314       15.32
## 14     25 to 29                1   140       10.03
## 1      25 to 29                0  1256       89.97
## 0      18 to 24                0   901       92.03
## 13     18 to 24                1    78        7.97
```

Age groups with the higher percentage of diabetes prevalent are 70 to 74 (63.91%), 75 to 79 (63.09%), 65 to 69 (60.41%) and 60 to 64 (56.70%) this could be due to the number of people in each age category in the study.

Education - 6 level factor

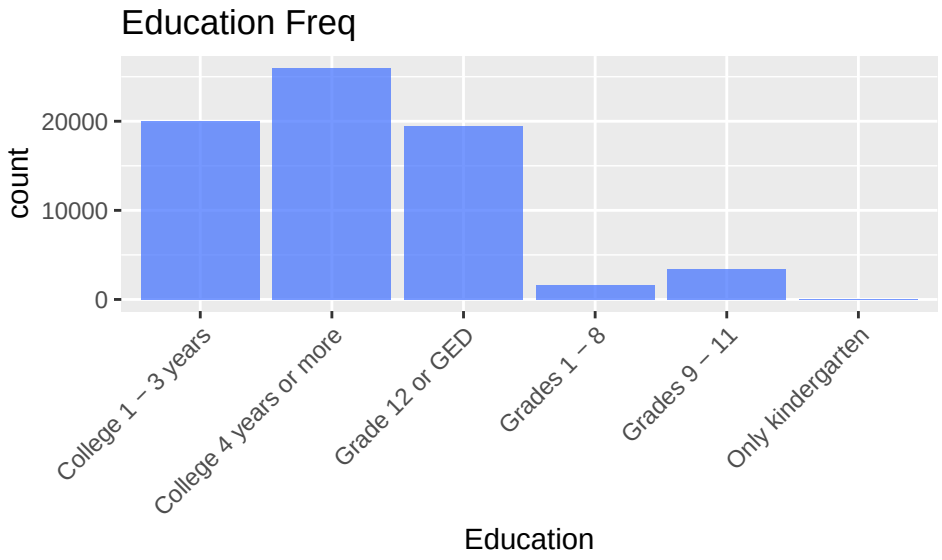
For this column in the data, people where asked - What is the highest grade or year of school you completed? and where placed into one of 6 categories. 1:Never attended school or only kindergarten,2:Grades 1 through 8 (Elementary),3:Grades 9 through 11 (Some high school),4:Grade 12 or GED (High school graduate),5:College 1 year to 3 years (Some college or technical school),6:College 4 years or more (College graduate)

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(Education)) +
  geom_bar(fill='royalblue1', alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  ggtitle("Education Freq")
```

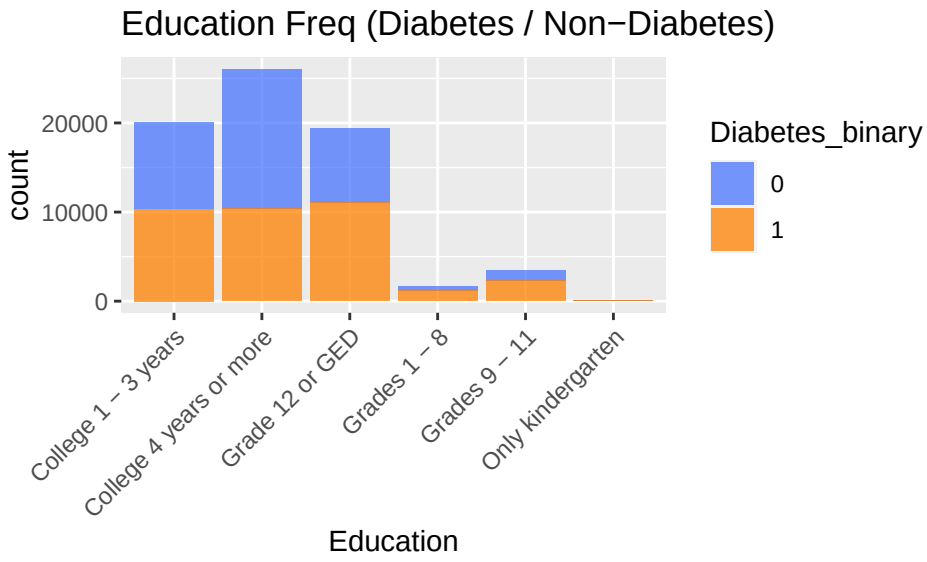



```
ggsave(path='./graphs',filename = '52_edu_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = Education, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("Education Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '53_edu_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$Education))*100)
names(table) <- c("Education", "Percent")
table
```

```
##                 Education   Percent
## 1     College 1 - 3 years 28.334182
## 2 College 4 years or more 36.807560
## 3         Grade 12 or GED 27.546257
## 4            Grades 1 - 8  2.329825
## 5           Grades 9 - 11  4.876082
## 6       Only kindergarten  0.106094
```

91% of people in the survey had achieved a level of education above grade 12 and over 60% attended college (65.14%)

Code: *R*

```
# creating x tab for python data
table2 <- xtabs(~ Education + Diabetes_binary, data=data.clean)
table2 <- as.data.frame(table2)
data_py <- table2
```

Checking the percentages per group of Diabetic and non diabetic

Code: *Python*

```
import pandas as pd

df = pd.DataFrame(r.data_py)
```

```
# grouping by factor and getting percentages
df['Percentage'] = ((df['Freq'] / df.groupby(['Education'])['Freq'].transform('sum'))*100).round(2)
df = df.sort_values('Education', ascending=False)
df
```

```
##                   Education Diabetes_binary   Freq  Percentage
## 5          Only kindergarten               0     28       37.33
## 11         Only kindergarten               1     47       62.67
## 4              Grades 9 - 11               0   1151       33.39
## 10             Grades 9 - 11               1   2296       66.61
## 3               Grades 1 - 8               0    464       28.17
## 9               Grades 1 - 8               1   1183       71.83
## 2            Grade 12 or GED               0   8407       43.17
## 8            Grade 12 or GED               1  11066       56.83
## 1   College 4 years or more               0  15620       60.03
## 7   College 4 years or more               1  10400       39.97
## 0       College 1 - 3 years               0   9676       48.31
## 6       College 1 - 3 years               1  10354       51.69
```

People that achieved a lower level of education had a higher percentage of diabetes with people achieving between grade 1 to 8 having the highest percentage of diabetes present with 71.83%

Income - 8 level factor

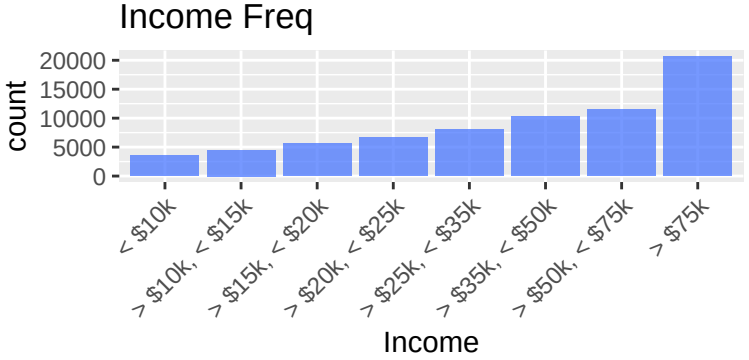
For this column, people where asked - What is your annual household income from all sources, and placed into one of eight categories.

Factor Frequency

The count of each value will be checked, both on its own and by the different predictor cohorts for the analysis to see if there are any insights that can be ascertained straight away.

Code: *R*

```
# freq
ggplot(data.clean, aes(Income)) +
  geom_bar(fill='royalblue1', alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  ggtitle("Income Freq")
```

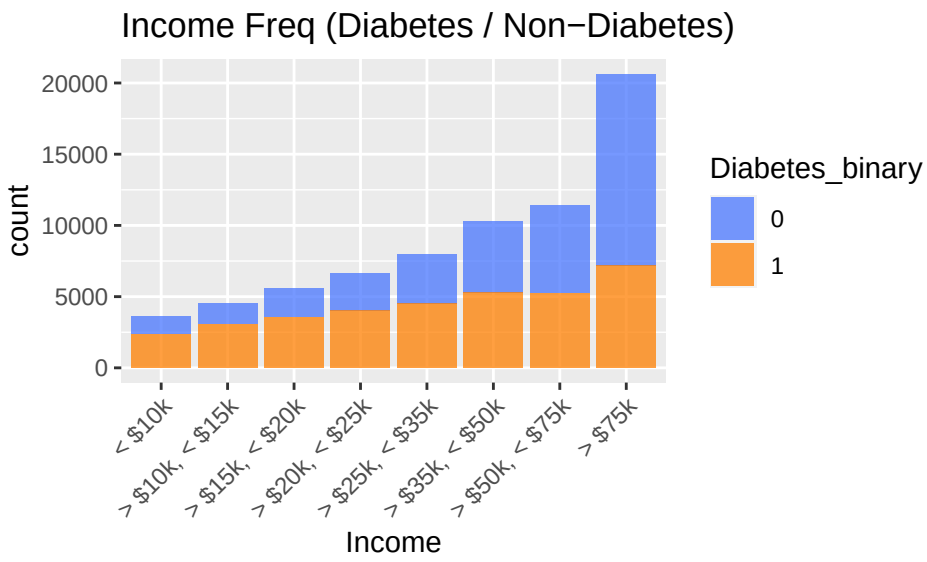


```
ggsave(path='./graphs',filename = '54_income_freq.png', dpi = 300)
```

```
## Saving 4 x 2 in image
```

Code: *R*

```
# freq with diabetes diagnosis
ggplot(data.clean, aes(x = Income, fill = Diabetes_binary)) +
  geom_bar(alpha = 0.75) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  scale_fill_manual(values = c("royalblue1", "darkorange1"))+
  ggtitle("Income Freq (Diabetes / Non-Diabetes)")
```




```
ggsave(path='./graphs',filename = '55_income_both_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Code: *R*

```
# percent of each factor
table <- as.data.frame(prop.table(table(data.clean$Income))*100)
names(table) <- c("Income", "Percent")
table
```

```
##           Income   Percent
## 1         < $10k  5.108074
## 2 > $10k, < $15k  6.362813
## 3 > $15k, < $20k  7.860861
## 4 > $20k, < $25k  9.418322
## 5 > $25k, < $35k 11.330844
## 6 > $35k, < $50k 14.551859
## 7 > $50k, < $75k 16.161659
## 8         > $75k 29.205568
```

59.92% of respondents had a total household income of greater than $35k with 29.21% of the people surveyed having a household income of greater than $75k.

Code: *R*

```
# creating x tab for python data
table2 <- xtabs(~ Income + Diabetes_binary, data=data.clean)
table2 <- as.data.frame(table2)
data_py <- table2
```

Checking the percentages per group of Diabetic and non diabetic

Code: *Python*

```
import pandas as pd

df = pd.DataFrame(r.data_py)

# grouping by factor and getting percentages
df['Percentage'] = ((df['Freq'] / df.groupby(['Income'])['Freq'].transform('sum'))*100).round(2)
df = df.sort_values('Income', ascending=False)
df
```

```
##              Income Diabetes_binary   Freq  Percentage
## 7           > $75k                0  13451       65.15
## 15          > $75k                1   7195       34.85
## 6   > $50k, < $75k                0   6160       53.92
## 14  > $50k, < $75k                1   5265       46.08
## 5   > $35k, < $50k                0   4996       48.57
## 13  > $35k, < $50k                1   5291       51.43
## 4   > $25k, < $35k                0   3506       43.77
## 12  > $25k, < $35k                1   4504       56.23
## 3   > $20k, < $25k                0   2604       39.11
## 11  > $20k, < $25k                1   4054       60.89
## 2   > $15k, < $20k                0   1989       35.79
## 10  > $15k, < $20k                1   3568       64.21
## 1   > $10k, < $15k                0   1412       31.39
## 9   > $10k, < $15k                1   3086       68.61
## 0           < $10k                0   1228       34.01
## 8           < $10k                1   2383       65.99
```

The survey indicates that the lower the household income bracket the higher percentage of people in that bracket with Diabetes.

Diabetes_binary - Predictor - 2 level factor

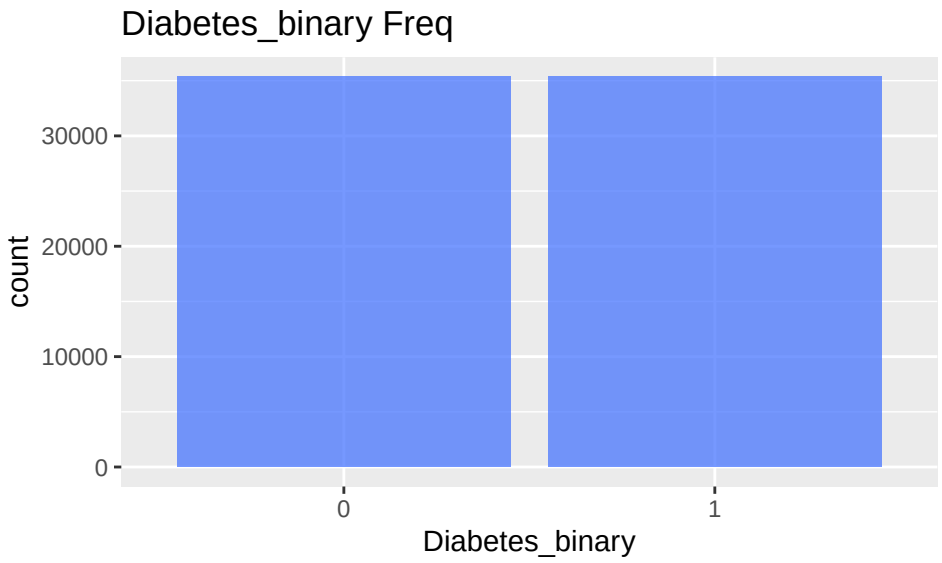
This is the dependent (predictor) variable for this analysis. Checking to see if the split is 50/50 between people with and without Diabetes.

Factor Frequency

The count of each value will be checked, but as this data set was downloaded as a 50/50 dataset between the groups it is expected to see a 50/50 split in the Diabetic versus non-Diabetic people.

Code: *R*

```
# freq
ggplot(data.clean, aes(Diabetes_binary)) +
  geom_bar(fill='royalblue1', alpha=0.75)+
  ggtitle("Diabetes_binary Freq")
```




```
ggsave(path='./graphs',filename = '56_diabetes_freq.png', dpi = 300)
```

```
## Saving 5 x 3 in image
```

Data Insights

Correlation Testing

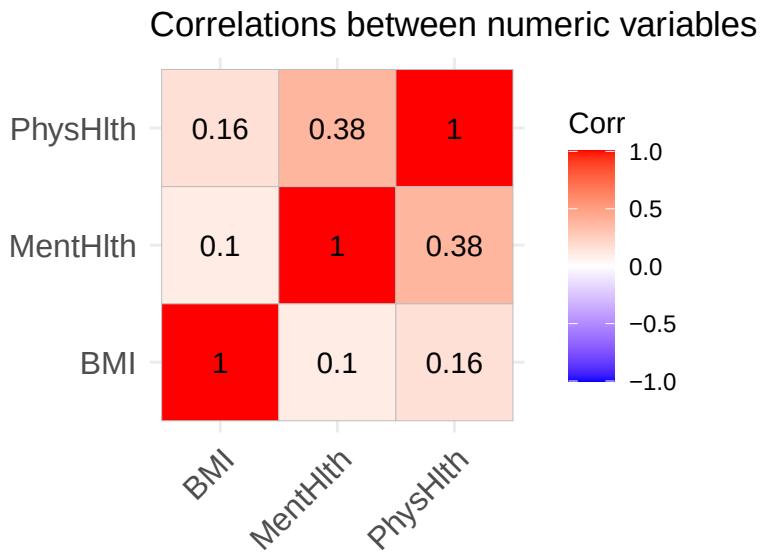
Correlations between numeric columns as well as chi squared tests of independence can be performed on the data set to see if there is any immediate correlations between columns or any dependence between factors.

Numeric Correlations

The correlations between the numeric variables can be tested with pearson correlation coefficient and visualized in a heat map and scatter plots.

Code: *R*

```
# checking correlations of numeric data
num_cols <- unlist(lapply(data.clean, is.numeric))


data_num <- data.clean[ , num_cols] # create Subset for correlation testing
# create correlation matrix for plotting
corr.mtrx<-round(cor(data_num),2)

# plot matrix
ggcorrplot(corr.mtrx, hc.order=TRUE, method ="square", lab=TRUE)+
  ggtitle("Correlations between numeric variables")
```



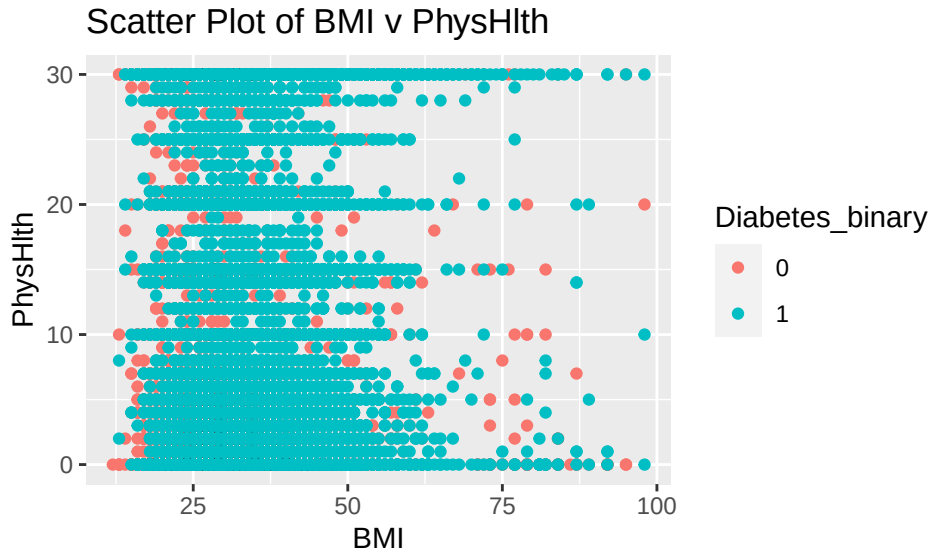
There appears to be no correlation between the numeric values in the dataset. MentHlth and PhysHlth have the strongest correlation with a low/medium correlation coefficient of 0.38.

WE can view these in scatter plots but is expected that due to the nature of the data in the PhysHlth and MentHlth columns where they could be classified as factors, the scatter plots will have no representaion of a correlation.

Code: *R*

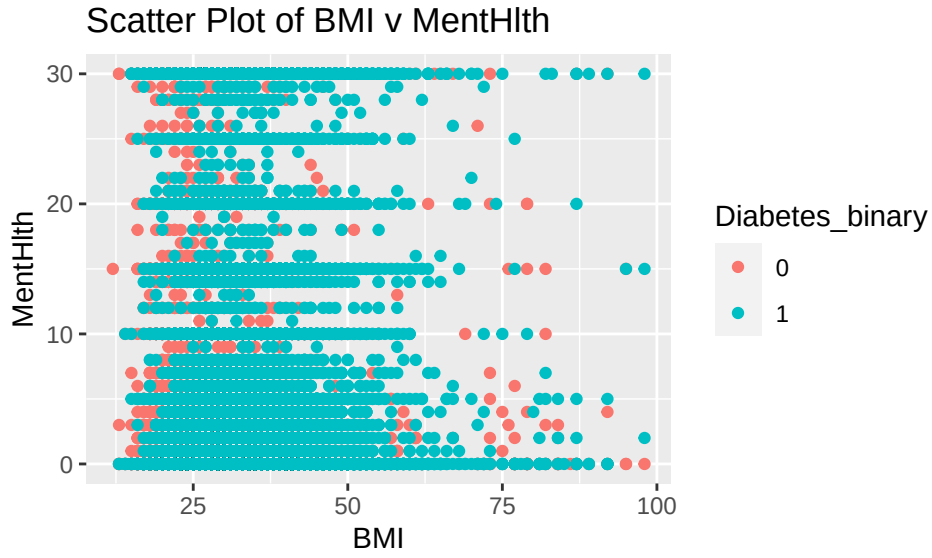
```
# Scatter plots for BMI and physhlth
ggplot(data.clean, aes(x=BMI, y=PhysHlth, colour=Diabetes_binary)) +
  geom_point()+
```

```
  ggtitle("Scatter Plot of BMI v PhysHlth")+
  xlab("BMI")+
  ylab("PhysHlth")
```



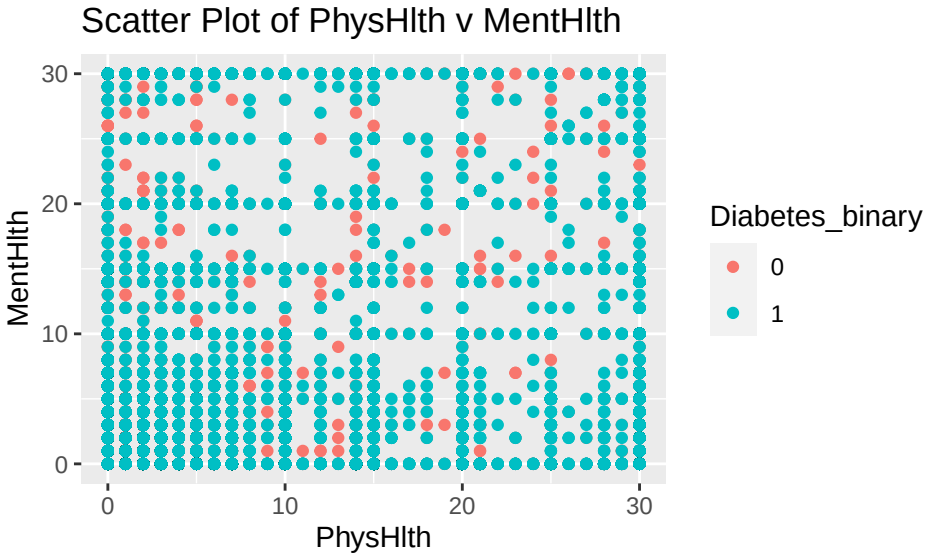
Code: *R*

```
# Scatter plots for BMI and menthlth
ggplot(data.clean, aes(x=BMI, y=MentHlth, colour=Diabetes_binary)) +
  geom_point()+
  ggtitle("Scatter Plot of BMI v MentHlth")+
  xlab("BMI")+
  ylab("MentHlth")
```



Code: *R*

```r
# Scatter plots for physhlth and menthlth
ggplot(data.clean, aes(x=PhysHlth, y=MentHlth, colour=Diabetes_binary)) +
  geom_point()+
  ggtitle("Scatter Plot of PhysHlth v MentHlth")+
  xlab("PhysHlth")+
  ylab("MentHlth")
```



As expected there is no value in the scatter plot representation of the data.

Chi tests of Independence (Complete DataFrame)

For the independence tests columns that showed no value in the data exploration will also be removed. Columns for testing will be: HighBP, HighChol, Smoker, Stroke, HeartDiseaseorAttack, PhysActivity, Fruits, Veggies, GenHlth, DiffWalk, Sex, Age, Education and Income. They will all be tested against the Diabetes factor to see if there is any dependence associated. Some columns will also be tested against the Sex column to see if they are dependent there also

H0 ($p < 0.05$): (null hypothesis) The two variables are independent (Diabetes ~ variable tested). H1 ($p > 0.05$): (alternative hypothesis) The two variables are not independent (Diabetes ~ variable tested).

Code: *R*

```r
# new copy of data
data.chi <- data.clean

# changing to factors
data.chi$PhysHlth <- as.factor(data.chi$PhysHlth)
data.chi$MentHlth <- as.factor(data.chi$MentHlth)

# selecting columns for the chi tests
data.chi <- data.chi %>% dplyr::select(Diabetes_binary, HighBP, HighChol, Smoker,
                                       Stroke, HeartDiseaseorAttack,PhysActivity,Fruits,Veggies,
                                       GenHlth,DiffWalk, Sex, Age, Education, Income)
```

HighBP

Performing chi tests of independence on HighBP and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~HighBP + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##        Diabetes_binary
## HighBP      0     1
##   FALSE 22118  8742
##   TRUE  13228 26604
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table3
## X-squared = 10288, df = 1, p-value < 2.2e-16
```

HighChol

Performing chi tests of independence on HighChol and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~HighChol + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##         Diabetes_binary
## HighChol     0     1
##    FALSE 21869 11660
##    TRUE  13477 23686
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table3
## X-squared = 5911.8, df = 1, p-value < 2.2e-16
```

Smoker

Performing chi tests of independence on Smoker and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~Smoker + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##       Diabetes_binary
## Smoker     0     1
##    No  20065 17029
##    Yes 15281 18317
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table3
## X-squared = 522.48, df = 1, p-value < 2.2e-16
```

Stroke

Performing chi tests of independence on Stroke and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~Stroke + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##       Diabetes_binary
## Stroke     0     1
##    No  34219 32078
##    Yes  1127  3268
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table3
## X-squared = 1111.1, df = 1, p-value < 2.2e-16
```

HeartDiseaseorAttack

Performing chi tests of independence on HeartDiseaseorAttack and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~HeartDiseaseorAttack + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##                     Diabetes_binary
## HeartDiseaseorAttack     0     1
##                  No  32775 27468
##                  Yes  2571  7878
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table3
## X-squared = 3161.7, df = 1, p-value < 2.2e-16
```

PhysActivity

Performing chi tests of independence on PhysActivity and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~PhysActivity + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##             Diabetes_binary
## PhysActivity     0     1
##        FALSE  7934 13059
##        TRUE  27412 22287
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table3
## X-squared = 1779, df = 1, p-value < 2.2e-16
```

Fruits

Performing chi tests of independence on Fruits and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~Fruits + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##        Diabetes_binary
## Fruits      0     1
##   FALSE 12790 14653
##   TRUE  22556 20693
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table3
## X-squared = 206.5, df = 1, p-value < 2.2e-16
```

Veggies

Performing chi tests of independence on Veggies and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~Veggies + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##        Diabetes_binary
## Veggies     0     1
##   FALSE  6322  8610
##   TRUE  29024 26736
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table3
## X-squared = 444.08, df = 1, p-value < 2.2e-16
```

GenHlth

Performing chi tests of independence on GenHlth and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~GenHlth + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##            Diabetes_binary
## GenHlth         0     1
##   Excellent  7142  1140
##   Fair       3513  9790
##   Good       9970 13457
##   Poor       1230  4578
##   Very Good 13491  6381
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table3
## X-squared = 12304, df = 4, p-value < 2.2e-16
```

DiffWalk

Performing chi tests of independence on DiffWalk and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~DiffWalk + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##         Diabetes_binary
## DiffWalk     0     1
##      No  30601 22225
##      Yes  4745 13121
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table3
## X-squared = 5253.7, df = 1, p-value < 2.2e-16
```

Sex

Performing chi tests of independence on the Sex and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~Sex + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##         Diabetes_binary
## Sex          0     1
##   Female 19975 18411
##   Male   15371 16935
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table3
## X-squared = 139.26, df = 1, p-value < 2.2e-16
```

Age

Performing chi tests of independence on Age and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~Age + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##              Diabetes_binary
## Age             0    1
##   18 to 24    901   78
##   25 to 29   1256  140
##   30 to 34   1735  314
##   35 to 39   2167  626
##   40 to 44   2469 1051
##   45 to 49   2906 1742
##   50 to 54   3784 3088
##   55 to 59   4340 4263
##   60 to 64   4379 5733
##   65 to 69   4298 6558
##   70 to 74   2903 5141
##   75 to 79   1991 3403
##   80 or older 2217 3209
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table3
## X-squared = 6179.1, df = 12, p-value < 2.2e-16
```

Education

Performing chi tests of independence on Education and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~Education + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##                          Diabetes_binary
## Education                     0     1
##   College 1 - 3 years      9676 10354
##   College 4 years or more 15620 10400
##   Grade 12 or GED          8407 11066
##   Grades 1 - 8              464  1183
##   Grades 9 - 11            1151  2296
##   Only kindergarten          28    47
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table3
## X-squared = 2132.3, df = 5, p-value < 2.2e-16
```

Income

Performing chi tests of independence on Income and Diabetes.

Code: *R*

```
# checking percentages per group
table3 <- xtabs(~Income + Diabetes_binary, data=data.chi)
table3 <- as.table(table3)
table3
```

```
##                 Diabetes_binary
## Income                0     1
##   < $10k           1228  2383
##   > $10k, < $15k   1412  3086
##   > $15k, < $20k   1989  3568
##   > $20k, < $25k   2604  4054
##   > $25k, < $35k   3506  4504
##   > $35k, < $50k   4996  5291
##   > $50k, < $75k   6160  5265
##   > $75k          13451  7195
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table3)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table3
## X-squared = 3855.5, df = 7, p-value < 2.2e-16
```

All of the chi squared tests returned a p-value of less than the alpha of 0.05, all returned at 2.2e-16 which would indicate the each variable is independent of the diabetes diagnosis. This is possibly due to the size of the population tested and the sensitivity or the test.

Chi test of Independence (Sample of Data)

Because all the values returned p-values of less than 2.2e-16 this could be because our data set is too large. To accommodate for this we can sample the dataset but also keeping the same distribution of all the factors and retest the sampled data to see if we get the same results.

H0 ($p < 0.05$): (null hypothesis) The two variables are independent (Diabetes ~ variable tested). H1 ($p > 0.05$): (alternative hypothesis) The two variables are not independent (Diabetes ~ variable tested).

The sample data set will have to reproduce the same distributions as the population dataset. The below code loops through each of the columns and tests whether the probabilities. For each test, if all the p-values are greater than 0.05, we can say that the sample is not biased and is a valid representation of the greater data set.

Code: *R*

```
# make a copy of the dataset
dataset <- data.chi

# set the sample size we want
sample_size = 353 # 0.5% of original dataset
```

```r
# set seed for reporducabilty
set.seed(1)

# parameters for the sample set
idxs = sample(1:nrow(dataset),sample_size,replace=F)

# create the sample set
subsample = dataset[idxs,]
# list for p-values
pvalues = list()

# loop through the dataset columns and test each column
# an alpha 0.05 is used and the p-value is used to select the
# the corresponding distribution in the sample set
for (col in names(dataset)) {
  if (class(dataset[,col]) %in% c("numeric","integer")) {
    # Numeric variable. Using Kolmogorov-Smirnov test
    pvalues[[col]] = ks.test(subsample[[col]],dataset[[col]])$p.value

  } else {
    # Categorical variable. Using Pearson's Chi-square test
    probs = table(dataset[[col]])/nrow(dataset)
    pvalues[[col]] = chisq.test(table(subsample[[col]]),p=probs)$p.value

  }
}

pvalues
```

```
## $Diabetes_binary
## [1] 0.4246561
##
## $HighBP
## [1] 0.9229867
##
## $HighChol
## [1] 0.703303
##
## $Smoker
## [1] 0.2314067
##
## $Stroke
## [1] 0.8347539
##
## $HeartDiseaseorAttack
## [1] 0.2817906
##
## $PhysActivity
## [1] 0.8002989
##
## $Fruits
## [1] 0.5097267
##
## $Veggies
## [1] 0.9414948
##
## $GenHlth
## [1] 0.630745
```

```
##
## $DiffWalk
## [1] 0.4466423
##
## $Sex
## [1] 0.4118774
##
## $Age
## [1] 0.749956
##
## $Education
## [1] 0.5048789
##
## $Income
## [1] 0.5144285
```

All of the p-values are greater than the 0.05 alpha therefore we can say that the sample set reproduces the data set in a smaller scale. The sample data set can now be retested with the chi test of independence.

HighBP - Sample test

Code: *R*

```r
# checking percentages per group
table4 <- xtabs(~HighBP + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##        Diabetes_binary
## HighBP    0   1
##   FALSE 109  46
##   TRUE   75 123
```

Code: *R*

```r
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table4
## X-squared = 35.383, df = 1, p-value = 2.708e-09
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical association between Diabetes and the tested variable

HighChol - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~HighChol + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##         Diabetes_binary
## HighChol   0   1
##    FALSE 115  56
##    TRUE   69 113
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table4
## X-squared = 29.247, df = 1, p-value = 6.371e-08
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical association between Diabetes and the tested variable

Smoker - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~Smoker + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##       Diabetes_binary
## Smoker  0  1
##    No  99 75
##    Yes 85 94
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table4
## X-squared = 2.7654, df = 1, p-value = 0.09632
```

Since the p-value of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is an association between Diabetes and the tested variable

Stroke - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~Stroke + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##       Diabetes_binary
## Stroke   0   1
##    No  178 154
##    Yes   6  15
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table4
## X-squared = 4.0108, df = 1, p-value = 0.04521
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical association between Diabetes and the tested variable

HeartDiseaseorAttack - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~HeartDiseaseorAttack + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##                     Diabetes_binary
## HeartDiseaseorAttack   0   1
##                  No  174 134
##                  Yes  10  35
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table4
## X-squared = 17.132, df = 1, p-value = 3.487e-05
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical association between Diabetes and the tested variable

PhysActivity - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~PhysActivity + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##             Diabetes_binary
## PhysActivity   0   1
##        FALSE  41  66
##        TRUE  143 103
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table4
## X-squared = 10.948, df = 1, p-value = 0.0009368
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical association between Diabetes and the tested variable

Fruits - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~Fruits + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##        Diabetes_binary
## Fruits    0   1
##   FALSE  63  68
##   TRUE  121 101
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table4
## X-squared = 1.1129, df = 1, p-value = 0.2915
```

Since the p-value of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is an association between Diabetes and the tested variable

Veggies - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~Veggies + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##        Diabetes_binary
## Veggies   0   1
##   FALSE  36  38
##   TRUE  148 131
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table4
## X-squared = 0.29421, df = 1, p-value = 0.5875
```

Since the p-value of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is an association between Diabetes and the tested variable

GenHlth - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~GenHlth + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##            Diabetes_binary
## GenHlth      0  1
##   Excellent 34  5
##   Fair      25 42
##   Good      59 68
##   Poor       7 25
##   Very Good 59 29
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table4
## X-squared = 46.314, df = 4, p-value = 2.119e-09
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical association between Diabetes and the tested variable

DiffWalk - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~DiffWalk + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##         Diabetes_binary
## DiffWalk   0   1
##      No  156 114
##      Yes  28  55
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table4
## X-squared = 13.758, df = 1, p-value = 0.0002079
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical association between Diabetes and the tested variable

Sex - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~Sex + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##         Diabetes_binary
## Sex       0  1
##   Female 95 89
##   Male   89 80
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table4
## X-squared = 0.0076226, df = 1, p-value = 0.9304
```

Since the p-value of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is an association between Diabetes and the tested variable

Age - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~Age + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##              Diabetes_binary
## Age            0  1
##   18 to 24     4  0
##   25 to 29     8  1
##   30 to 34    12  0
##   35 to 39     9  2
##   40 to 44    11  7
##   45 to 49    18  9
##   50 to 54    25 15
##   55 to 59    24 21
##   60 to 64    25 27
##   65 to 69    16 38
##   70 to 74    13 17
##   75 to 79    11 20
##   80 or older  8 12
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
## Warning in chisq.test(table4): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table4
## X-squared = 44.918, df = 12, p-value = 1.064e-05
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical association between Diabetes and the tested variable

Education - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~Education + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##                          Diabetes_binary
## Education                  0  1
##   College 1 - 3 years     59 47
##   College 4 years or more 69 47
##   Grade 12 or GED         44 55
##   Grades 1 - 8             3  8
##   Grades 9 - 11            8 12
##   Only kindergarten        1  0
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
## Warning in chisq.test(table4): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table4
## X-squared = 10.207, df = 5, p-value = 0.06958
```

Since the p-value of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is an association between Diabetes and the tested variable

Income - Sample test

Code: *R*

```
# checking percentages per group
table4 <- xtabs(~Income + Diabetes_binary, data=subsample)
table4 <- as.table(table4)
table4
```

```
##                 Diabetes_binary
## Income            0  1
##   < $10k          5 12
##   > $10k, < $15k 14 14
##   > $15k, < $20k 11 16
##   > $20k, < $25k 19 20
##   > $25k, < $35k 22 22
##   > $35k, < $50k 18 24
##   > $50k, < $75k 26 22
##   > $75k         69 39
```

Code: *R*

```
#Perform Chi-Square Test of Independence
chisq.test(table4)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table4
## X-squared = 12.743, df = 7, p-value = 0.07861
```

Since the p-value of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is an association between Diabetes and the tested variable

Results

From the sample size created the results are:

- HighBP

p-value of 2.708e-09 (REJECT) This means we have sufficient evidence to say that there is an association between HighBP and Diabetes

- HighChol

p-value of 6.371e-08 (REJECT) This means we have sufficient evidence to say that there is an association between HighBP and Diabetes

- Smoker

p-value of 0.09632 (FAIL to REJECT) This means we do not have sufficient evidence to say that there is an association between Smoker and Diabetes

- Stroke

p-value of 0.04521 (REJECT) This means we have sufficient evidence to say that there is an association between Stroke and Diabetes

- HeartDiseaseorAttack

p-value of 3.487e-05 (REJECT) This means we have sufficient evidence to say that there is an association between HeartDiseaseorAttack and Diabetes

- PhysActivity

p-value of 0.0009368 (REJECT) This means we have sufficient evidence to say that there is an association between PhysActivity and Diabetes

- Fruits

p-value of 0.2915 (FAIL to REJECT) This means we do not have sufficient evidence to say that there is an association between Fruits and Diabetes

- Veggies

p-value of 0.5875 (FAIL to REJECT) This means we do not have sufficient evidence to say that there is an association between Veggies and Diabetes

- GenHlth

p-value of 2.119e-09 (REJECT) This means we have sufficient evidence to say that there is an association between GenHlth and Diabetes

- DiffWalk

p-value of 0.0002079 (REJECT) This means we have sufficient evidence to say that there is an association between DiffWalk and Diabetes

- Sex

p-value of 0.9304 (FAIL to REJECT) This means we do not have sufficient evidence to say that there is an association between Sex and Diabetes

- Age

p-value of 1.064e-05 (REJECT) This means we have sufficient evidence to say that there is an association between Age and Diabetes

- Education

p-value of 0.06958 (FAIL to REJECT) This means we do not have sufficient evidence to say that there is an association between Education and Diabetes

- Income

p-value of 0.07861 (FAIL to REJECT) This means we do not have sufficient evidence to say that there is an association between Income and Diabetes

Chi Squared Test, Goodness of fit (expected / observed)

Where somebody has diabetes we can check the percentages to see of there is any statistical difference in the cohorts that might indicate a higher prevalence in one than the other.

For this test it will be performed on the binary factors (TRUE / FALSE) and (Yes / No) answers to see if they vary statistically from a 50/50 split

For this the non-diabetic people in the survey will be filtered out and only positive diabetic diagnosis will be used.

Code: *R*

```
# only keeping diabetic people
data_diabetic <- subset(data.chi, data.chi$Diabetes_binary == 1)

# check factor types and only keep columns with binary factors
str(data_diabetic)
```

```
## 'data.frame':    35346 obs. of  15 variables:
##  $ Diabetes_binary     : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ HighBP              : logi  TRUE FALSE TRUE FALSE TRUE TRUE ...
##  $ HighChol            : logi  TRUE FALSE TRUE FALSE FALSE TRUE ...
##  $ Smoker              : Factor w/ 2 levels "No","Yes": 2 2 1 2 1 2 2 2 2 2 ...
##  $ Stroke              : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 2 1 ...
##  $ HeartDiseaseorAttack: Factor w/ 2 levels "No","Yes": 2 1 1 1 1 2 2 1 1 1 ...
##  $ PhysActivity        : logi  FALSE TRUE FALSE TRUE TRUE FALSE ...
##  $ Fruits              : logi  TRUE TRUE FALSE FALSE TRUE FALSE ...
##  $ Veggies             : logi  TRUE TRUE TRUE FALSE TRUE TRUE ...
##  $ GenHlth             : Factor w/ 5 levels "Excellent","Fair",..: 4 3 2 5 1 4 2 2 2 5 ...
##  $ DiffWalk            : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 2 1 2 2 1 ...
##  $ Sex                 : Factor w/ 2 levels "Female","Male": 1 2 1 2 1 2 2 1 1 1 ...
##  $ Age                 : Factor w/ 13 levels "18 to 24","25 to 29",..: 9 13 11 7 13 10 12 8 9 12 ...
##  $ Education           : Factor w/ 6 levels "College 1 - 3 years",..: 1 2 3 1 1 2 4 3 1 5 ...
##  $ Income              : Factor w/ 8 levels "< $10k","> $10k, < $15k",..: 1 8 6 6 4 5 4 7 4 3 ...
```

Code: *R*

```
# selecting columns for the chi tests
data_diabetic <- data_diabetic %>% dplyr::select(HighBP, HighChol, Smoker,
                                      Stroke, HeartDiseaseorAttack,PhysActivity,Fruits,Veggies,
                                      DiffWalk, Sex)

str(data_diabetic)
```

```
## 'data.frame':    35346 obs. of  10 variables:
##  $ HighBP              : logi  TRUE FALSE TRUE FALSE TRUE TRUE ...
##  $ HighChol            : logi  TRUE FALSE TRUE FALSE FALSE TRUE ...
##  $ Smoker              : Factor w/ 2 levels "No","Yes": 2 2 1 2 1 2 2 2 2 2 ...
##  $ Stroke              : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 2 1 ...
##  $ HeartDiseaseorAttack: Factor w/ 2 levels "No","Yes": 2 1 1 1 1 2 2 1 1 1 ...
##  $ PhysActivity        : logi  FALSE TRUE FALSE TRUE TRUE FALSE ...
##  $ Fruits              : logi  TRUE TRUE FALSE FALSE TRUE FALSE ...
##  $ Veggies             : logi  TRUE TRUE TRUE FALSE TRUE TRUE ...
##  $ DiffWalk            : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 2 1 2 2 1 ...
##  $ Sex                 : Factor w/ 2 levels "Female","Male": 1 2 1 2 1 2 2 1 1 1 ...
```

Left with 10 columns to check

HighBP

Chi squared test on the percentage that do and don't have high blood pressure when having Diabetes

Code: *R*

```
# create percentage table for chi squared test
tbl = table(data_diabetic$HighBP)
tbl.perc = prop.table(tbl)
tbl.perc = round(tbl.perc*100,2)

# create observed versus expected
obs <- as.numeric(tbl.perc)
obs
```

```
## [1] 24.73 75.27
```

```
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 25.543, df = 1, p-value = 4.327e-07
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have Diabetes and the tested variable

HighChol

Chi squared test on the percentage that do and don't have high cholesterol when having Diabetes

Code: *R*

```
# create percentage table for chi squared test
tbl = table(data_diabetic$HighChol)
tbl.perc = prop.table(tbl)
tbl.perc = round(tbl.perc*100,2)

# create observed versus expected
obs <- as.numeric(tbl.perc)
obs
```

```
## [1] 32.99 67.01
```

```
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 11.574, df = 1, p-value = 0.0006689
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have Diabetes and the tested variable

Smoker

Chi squared test on the percentage that do and don't smoke when having Diabetes

Code: *R*

```
# create percentage table for chi squared test
tbl = table(data_diabetic$Smoker)
tbl.perc = prop.table(tbl)
tbl.perc = round(tbl.perc*100,2)

# create observed versus expected
obs <- as.numeric(tbl.perc)
obs
```

```
## [1] 48.18 51.82
```

```
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 0.1325, df = 1, p-value = 0.7159
```

Since the p-value of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is a statistical difference in the people that have Diabetes and the tested variable

Stroke

Chi squared test on the percentage that did and didn't have a stroke while having Diabetes

Code: *R*

```
# create percentage table for chi squared test
tbl = table(data_diabetic$Stroke)
tbl.perc = prop.table(tbl)
tbl.perc = round(tbl.perc*100,2)

# create observed versus expected
obs <- as.numeric(tbl.perc)
obs
```

```
## [1] 90.75  9.25
```

```
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 66.422, df = 1, p-value = 3.639e-16
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have Diabetes and the tested variable

HeartDiseaseorAttack

Chi squared test on the percentage that did and didn't have a diagnosis of some kind of heart disease while having Diabetes

Code: *R*

```
# create percentage table for chi squared test
tbl = table(data_diabetic$HeartDiseaseorAttack)
tbl.perc = prop.table(tbl)
tbl.perc = round(tbl.perc*100,2)

# create observed versus expected
obs <- as.numeric(tbl.perc)
obs
```

```
## [1] 77.71 22.29
```

```
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 30.714, df = 1, p-value = 2.99e-08
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have Diabetes and the tested variablee

PhysActivity

Chi squared test on the percentage that did and didn't report doing a physical activity in the last 30 days while having Diabetes

Code: *R*

```
# create percentage table for chi squared test
tbl = table(data_diabetic$PhysActivity)
tbl.perc = prop.table(tbl)
tbl.perc = round(tbl.perc*100,2)

# create observed versus expected
obs <- as.numeric(tbl.perc)
obs
```

```
## [1] 36.95 63.05
```

```
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 6.8121, df = 1, p-value = 0.009054
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have Diabetes and the tested variable

Fruits

Chi squared test on the percentage that did and didn't report eating at least one Fruit a day while having Diabetes

Code: *R*

```
# create percentage table for chi squared test
tbl = table(data_diabetic$Fruits)
tbl.perc = prop.table(tbl)
tbl.perc = round(tbl.perc*100,2)

# create observed versus expected
obs <- as.numeric(tbl.perc)
obs
```

```
## [1] 41.46 58.54
```

```
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 2.9173, df = 1, p-value = 0.08764
```

Since the p-value of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is a statistical difference in the people that have Diabetes and the tested variable

Veggies

Chi squared test on the percentage that did and didn't report eating at least one vegetable a day while having Diabetes

Code: *R*

```
# create percentage table for chi squared test
tbl = table(data_diabetic$Veggies)
tbl.perc = prop.table(tbl)
tbl.perc = round(tbl.perc*100,2)

# create observed versus expected
obs <- as.numeric(tbl.perc)
obs
```

```
## [1] 24.36 75.64
```

```
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 26.296, df = 1, p-value = 2.928e-07
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have Diabetes and the tested variable

DiffWalk

Chi squared test on the percentage that did and didn't report having serious difficulty walking or climbing stairs while having Diabetes

Code: *R*

```
# create percentage table for chi squared test
tbl = table(data_diabetic$DiffWalk)
tbl.perc = prop.table(tbl)
tbl.perc = round(tbl.perc*100,2)

# create observed versus expected
obs <- as.numeric(tbl.perc)
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 6.6358, df = 1, p-value = 0.009995
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have Diabetes and the tested variable

Sex

Chi squared test on the percentage that are Male versus Female while having Diabetes

Code: *R*

```
# create percentage table for chi squared test
tbl = table(data_diabetic$Sex)
tbl.perc = prop.table(tbl)
tbl.perc = round(tbl.perc*100,2)

# create observed versus expected
obs <- as.numeric(tbl.perc)
obs
```

```
## [1] 52.09 47.91
```

```
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 0.17472, df = 1, p-value = 0.6759
```

Since the p-value of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is a statistical difference in the people that have Diabetes and the tested variable

Results

From the chi squared goodness of fit test against people who do have diabetes and the observed percentage in that group with a particular ailment.

- HighBP

p-value of 4.327e-07 (REJECT)

This means that in the number of people who have diabetes, HighBP does not follow a hypothesized distribution of 50:50.

- HighChol

p-value of 0.0006689 (REJECT) This means that in the number of people who have diabetes, HighChol does not follow a hypothesized distribution of 50:50.

- Smoker

p-value of 0.7159 (FAIL to REJECT) This means that in the number of people who have diabetes, Smoker does approx. follow a hypothesized distribution of 50:50.

- Stroke

p-value of 3.639e-16 (REJECT) This means that in the number of people who have diabetes, Stroke does not follow a hypothesized distribution of 50:50.

- HeartDiseaseorAttack

p-value of 2.99e-08 (REJECT) This means that in the number of people who have diabetes, HeartDiseaseorAttack does not follow a hypothesized distribution of 50:50.

- PhysActivity

p-value of 0.009054 (REJECT) This means that in the number of people who have diabetes, PhysActivity does not follow a hypothesized distribution of 50:50.

- Fruits

p-value of 0.08764 (FAIL to REJECT) This means that in the number of people who have diabetes, Fruits does approx. follow a hypothesized distribution of 50:50

- Veggies

p-value of 2.928e-07 (REJECT) This means that in the number of people who have diabetes, Veggies does not follow a hypothesized distribution of 50:50.

- DiffWalk

p-value of 0.009995 (REJECT) This means that in the number of people who have diabetes, DiffWalk does not follow a hypothesized distribution of 50:50.

- Sex

p-value of 0.6759 (FAIL to REJECT) This means that in the number of people who have diabetes, Sex does approx. follow a hypothesized distribution of 50:50

Chi squared test (Goodness of Fit) - Part 2

For these tests, it will be tested the number of males and females in the survey with or without diabetes, people with high blood pressure, high cholesterol, low physical activity, difficulty waling, and have had a stroke or heart disease.

Males

Code: *R*

```
# only keeping males
data.chi.test <- subset(data.chi, data.chi$Sex == 'Male')

# selecting columns for the chi tests
data.chi.test <- data.chi.test %>% dplyr::select(Diabetes_binary)

tbl.chi <- table(data.chi.test)

tbl.perc.chi = prop.table(tbl.chi)
tbl.perc.chi = round(tbl.perc.chi*100,2)
tbl.perc.chi
```

```
## data.chi.test
##     0     1
## 47.58 52.42
```

```r
# create observed versus expected
obs <- as.numeric(tbl.perc.chi)
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 0.23426, df = 1, p-value = 0.6284
```

Since the p-value of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is a statistical difference in the people that have or have not got Diabetes and the tested variable

Females

Code: *R*

```r
# only keeping males
data.chi.test <- subset(data.chi, data.chi$Sex == 'Female')

# selecting columns for the chi tests
data.chi.test <- data.chi.test %>% dplyr::select(Diabetes_binary)

tbl.chi <- table(data.chi.test)

tbl.perc.chi = prop.table(tbl.chi)
tbl.perc.chi = round(tbl.perc.chi*100,2)
tbl.perc.chi
```

```
## data.chi.test
##     0     1
## 52.04 47.96
```

```r
# create observed versus expected
obs <- as.numeric(tbl.perc.chi)
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 0.16646, df = 1, p-value = 0.6833
```

Since the p-value of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is a statistical difference in the people that have or have not got Diabetes and the tested variable

High Blood Pressure

Code: *R*

```
# only keeping males
data.chi.test <- subset(data.chi, data.chi$HighBP == TRUE)

# selecting columns for the chi tests
data.chi.test <- data.chi.test %>% dplyr::select(Diabetes_binary)

tbl.chi <- table(data.chi.test)

tbl.perc.chi = prop.table(tbl.chi)
tbl.perc.chi = round(tbl.perc.chi*100,2)
tbl.perc.chi
```

```
## data.chi.test
##     0     1
## 33.21 66.79
```

```
# create observed versus expected
obs <- as.numeric(tbl.perc.chi)
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 11.276, df = 1, p-value = 0.0007851
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have or have not got Diabetes and the tested variable

High Cholesterol

Code: *R*

```
# only keeping males
data.chi.test <- subset(data.chi, data.chi$HighChol == TRUE)

# selecting columns for the chi tests
data.chi.test <- data.chi.test %>% dplyr::select(Diabetes_binary)

tbl.chi <- table(data.chi.test)

tbl.perc.chi = prop.table(tbl.chi)
tbl.perc.chi = round(tbl.perc.chi*100,2)
tbl.perc.chi
```

```
## data.chi.test
##     0     1
## 36.26 63.74
```

```r
# create observed versus expected
obs <- as.numeric(tbl.perc.chi)
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 7.5515, df = 1, p-value = 0.005996
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have or have not got Diabetes and the tested variable

Low Physical Activity

Code: *R*

```r
# only keeping males
data.chi.test <- subset(data.chi, data.chi$PhysActivity == FALSE)

# selecting columns for the chi tests
data.chi.test <- data.chi.test %>% dplyr::select(Diabetes_binary)

tbl.chi <- table(data.chi.test)

tbl.perc.chi = prop.table(tbl.chi)
tbl.perc.chi = round(tbl.perc.chi*100,2)
tbl.perc.chi
```

```
## data.chi.test
##     0     1
## 37.79 62.21
```

```r
# create observed versus expected
obs <- as.numeric(tbl.perc.chi)
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 5.9634, df = 1, p-value = 0.01461
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have or have not got Diabetes and the tested variable

Difficulty Walking

Code: *R*

```
# only keeping males
data.chi.test <- subset(data.chi, data.chi$DiffWalk == 'Yes')

# selecting columns for the chi tests
data.chi.test <- data.chi.test %>% dplyr::select(Diabetes_binary)

tbl.chi <- table(data.chi.test)

tbl.perc.chi = prop.table(tbl.chi)
tbl.perc.chi = round(tbl.perc.chi*100,2)
tbl.perc.chi
```

```
## data.chi.test
##     0     1
## 26.56 73.44
```

```
# create observed versus expected
obs <- as.numeric(tbl.perc.chi)
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 21.977, df = 1, p-value = 2.759e-06
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have or have not got Diabetes and the tested variable

Had a Stroke

Code: *R*

```
# only keeping males
data.chi.test <- subset(data.chi, data.chi$Stroke == 'Yes')

# selecting columns for the chi tests
data.chi.test <- data.chi.test %>% dplyr::select(Diabetes_binary)

tbl.chi <- table(data.chi.test)

tbl.perc.chi = prop.table(tbl.chi)
tbl.perc.chi = round(tbl.perc.chi*100,2)
tbl.perc.chi
```

```
## data.chi.test
##     0     1
## 25.64 74.36
```

```r
# create observed versus expected
obs <- as.numeric(tbl.perc.chi)
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 23.736, df = 1, p-value = 1.105e-06
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have or have not got Diabetes and the tested variable

Had/Have Heart Disease

Code: *R*

```r
# only keeping males
data.chi.test <- subset(data.chi, data.chi$HeartDiseaseorAttack == 'Yes')

# selecting columns for the chi tests
data.chi.test <- data.chi.test %>% dplyr::select(Diabetes_binary)

tbl.chi <- table(data.chi.test)

tbl.perc.chi = prop.table(tbl.chi)
tbl.perc.chi = round(tbl.perc.chi*100,2)
tbl.perc.chi
```

```
## data.chi.test
##     0     1
## 24.61 75.39
```

```r
# create observed versus expected
obs <- as.numeric(tbl.perc.chi)
exp <- c(.5, .5)

chisq.test(x = obs, p = exp )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 25.786, df = 1, p-value = 3.814e-07
```

Since the p-value of the test is less than 0.05, we reject the null hypothesis. This means there is sufficient evidence to say that there is a statistical difference in the people that have or have not got Diabetes and the tested variable

Results

From the chi squared goodness of fit test against people who do and do not have diabetes in a particular group.

- Males

p-value of 0.6284 (FAIL to REJECT)

This means that in the number of Males, Diabetes does approx. follow a hypothesized distribution of 50:50.

- Females

p-value of 0.6833 (FAIL to REJECT)

This means that in the number of Females, Diabetes does approx. follow a hypothesized distribution of 50:50.

- High Blood Pressure

p-value of 0.0007851 (REJECT) This means that in the number of people with High Blood Pressure, Diabetes does not follow a hypothesized distribution of 50:50.

- High Cholesterol

p-value of 0.005996 (REJECT) This means that in the number of people who have High Cholesterol, Diabetes does not follow a hypothesized distribution of 50:50.

- Low Physical Activity

p-value of 0.01461 (REJECT) This means that in the number of people who reported Low Physical Activity, Diabetes does not follow a hypothesized distribution of 50:50.

- Difficulty Walking

p-value of 2.759e-06 (REJECT) This means that in the number of people who reported difficulty walking, Diabetes does not follow a hypothesized distribution of 50:50.

- Had a Stroke

p-value of 1.105e-06 (REJECT) This means that in the number of people who have had a stroke, Diabetes does not approx. follow a hypothesized distribution of 50:50

- Had/Have Heart Disease

p-value of 3.814e-07 (REJECT) This means that in the number of people who have some type of reported heart disease, Diabetes does not follow a hypothesized distribution of 50:50.

Modeling

Logistic Regression

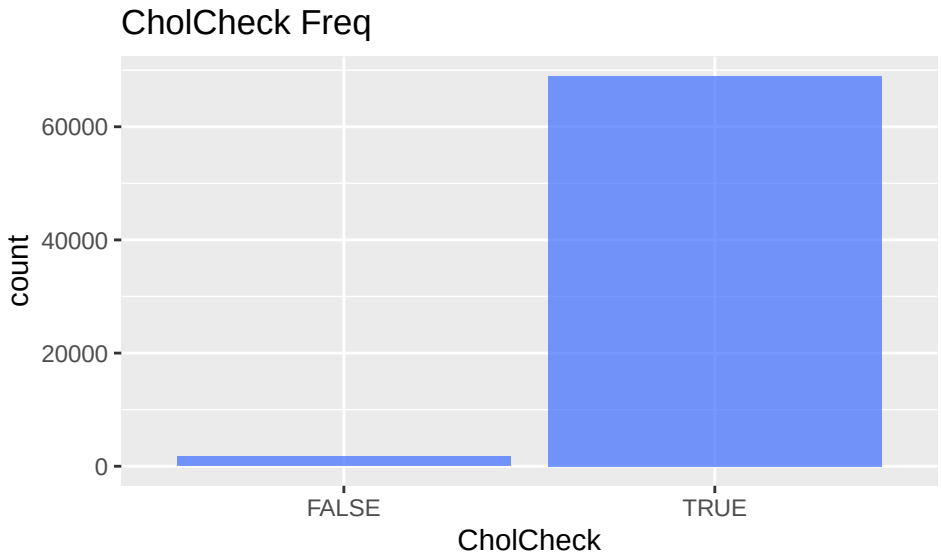
The data has a binomial predictor so logistic regression can be used to attempt to predict the presence of Diabetes in people. The regression will be run in a few different models and the tested for accuracy, specificity. miss-classification error and ROC.

Preparing Data for Logistic Regression

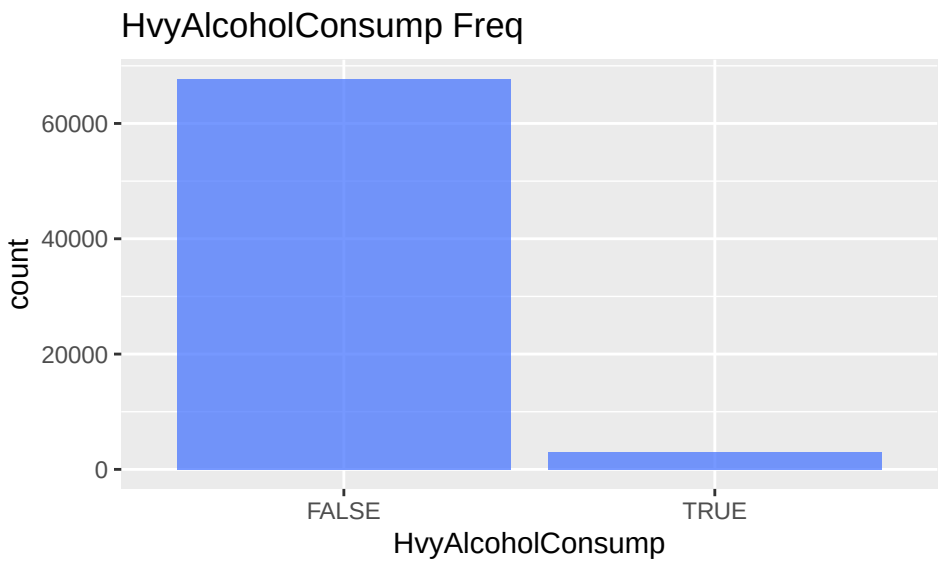
Some of the columns will not be used for the regression as they do not provide any additional information, such as **CholCheck**, **HvyAlcoholConsump**, **NoDocbcCost** and **AnyHealcare** as almost all of the surveyed people answered one way on each of these

Code: *R*

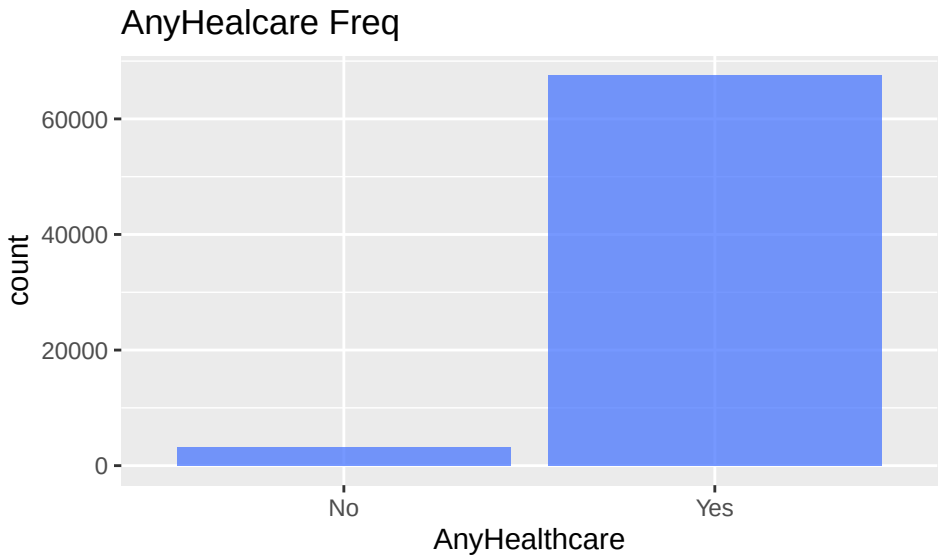
```
# freq
ggplot(data.clean, aes(CholCheck)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("CholCheck Freq")
```



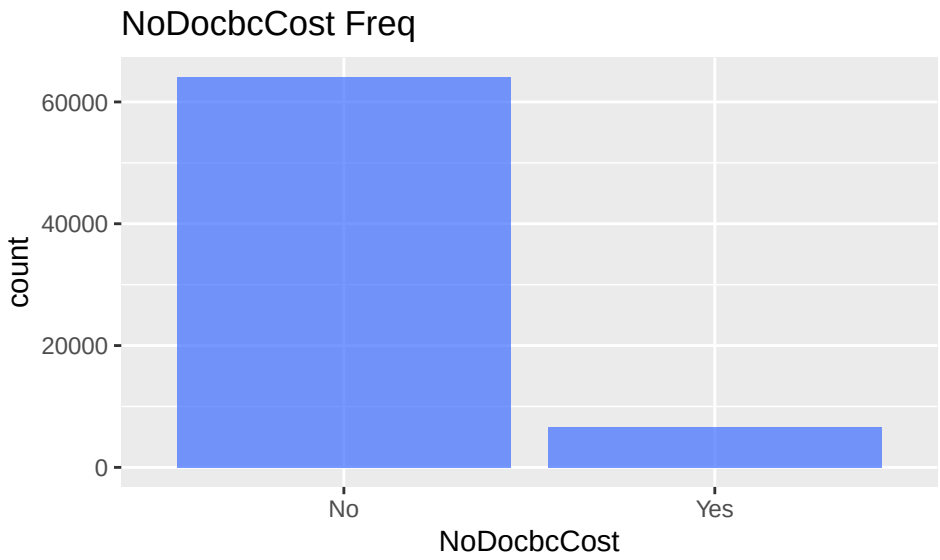

```
# freq
ggplot(data.clean, aes(HvyAlcoholConsump)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("HvyAlcoholConsump Freq")
```




```
# freq
ggplot(data.clean, aes(AnyHealthcare)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("AnyHealcare Freq")
```




```
# freq
ggplot(data.clean, aes(NoDocbcCost)) +
  geom_bar(fill='royalblue1', alpha = 0.75)+
  ggtitle("NoDocbcCost Freq")
```



Selecting the columns that will be used for regression testing

Code: *R*

```
# remove some columns with excess factor types of model
data.reg <- data.clean %>% dplyr::select(Diabetes_binary, HighBP, HighChol, BMI,
                                         Smoker, Stroke, HeartDiseaseorAttack, PhysActivity,
                                         Fruits, Veggies, GenHlth, MentHlth, PhysHlth, DiffWalk,
                                         Sex, Age, Education, Income )

str(data.reg)
```

```
## 'data.frame':    70692 obs. of  18 variables:
##  $ Diabetes_binary     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ HighBP              : logi  TRUE TRUE FALSE TRUE FALSE FALSE ...
##  $ HighChol            : logi  FALSE TRUE FALSE TRUE FALSE FALSE ...
##  $ BMI                 : num  26 26 26 28 29 18 26 31 32 27 ...
##  $ Smoker              : Factor w/ 2 levels "No","Yes": 1 2 1 2 2 1 2 2 1 2 ...
##  $ Stroke              : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...
##  $ HeartDiseaseorAttack: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PhysActivity        : logi  TRUE FALSE TRUE TRUE TRUE TRUE ...
##  $ Fruits              : logi  FALSE TRUE TRUE TRUE TRUE TRUE ...
##  $ Veggies             : logi  TRUE FALSE TRUE TRUE TRUE TRUE ...
##  $ GenHlth             : Factor w/ 5 levels "Excellent","Fair",..: 3 3 1 3 5 5 1 2 3 3 ...
##  $ MentHlth            : num  5 0 0 0 0 7 0 0 0 0 ...
##  $ PhysHlth            : num  30 0 10 3 0 0 0 0 0 6 ...
##  $ DiffWalk            : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Sex                 : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 1 2 2 1 2 ...
##  $ Age                 : Factor w/ 13 levels "18 to 24","25 to 29",..: 4 12 13 11 8 1 13 6 3 6 ...
##  $ Education           : Factor w/ 6 levels "College 1 - 3 years",..: 2 2 2 2 1 3 1 3 2 3 ...
##  $ Income              : Factor w/ 8 levels "< $10k","> $10k, < $15k",..: 8 8 8 8 8 7 6 3 8 4 ...
##  - attr(*, "pandas.index")=RangeIndex(start=0, stop=70692, step=1)
```

Creating a training and test set for the model, 80% will be used for testing and 20% will be the training set

Code: *R*

```
#Use 70% of data set as training set and remaining 30% as testing set
sample <- sample(c(TRUE, FALSE), nrow(data.reg), replace=TRUE, prob=c(0.8,0.2))
train <- data.reg[sample, ]
test <- data.reg[!sample, ]
```

Check that the training and test sets have each of the factors included

Code: *R*

```
#summary of training set
summary(train)
```

```
##  Diabetes_binary   HighBP         HighChol            BMI        Smoker      Stroke
##  0:28281         Mode :logical   Mode :logical   Min.   :12.00   No :29704   No :52973
##  1:28217         FALSE:24785     FALSE:26737     1st Qu.:25.00   Yes:26794   Yes: 3525
##                  TRUE :31713     TRUE :29761     Median :29.00
##                                                  Mean   :29.85
##                                                  3rd Qu.:33.00
##                                                  Max.   :98.00
##
##  HeartDiseaseorAttack PhysActivity      Fruits         Veggies             GenHlth
##  No :48139            Mode :logical   Mode :logical   Mode :logical   Excellent: 6627
##  Yes: 8359            FALSE:16829     FALSE:22004     FALSE:12003     Fair     :10709
##                       TRUE :39669     TRUE :34494     TRUE :44495     Good     :18714
##                                                                       Poor     : 4595
##                                                                       Very Good:15853
##
##
##     MentHlth         PhysHlth      DiffWalk        Sex                 Age
##  Min.   : 0.000   Min.   : 0.000   No :42293   Female:30728   65 to 69   : 8667
##  1st Qu.: 0.000   1st Qu.: 0.000   Yes:14205   Male  :25770   60 to 64   : 8089
##  Median : 0.000   Median : 0.000                              55 to 59   : 6863
##  Mean   : 3.797   Mean   : 5.821                              70 to 74   : 6382
##  3rd Qu.: 2.000   3rd Qu.: 5.000                              50 to 54   : 5528
##  Max.   :30.000   Max.   :30.000                              80 or older: 4316
##                                                               (Other)    :16653
##                    Education                Income
##  College 1 - 3 years    :16004   > $75k        :16445
##  College 4 years or more:20826   > $50k, < $75k: 9152
##  Grade 12 or GED        :15559   > $35k, < $50k: 8253
##  Grades 1 - 8           : 1299   > $25k, < $35k: 6467
##  Grades 9 - 11          : 2749   > $20k, < $25k: 5335
##  Only kindergarten      :   61   > $15k, < $20k: 4381
##                                  (Other)       : 6465
```

Code: *R*

```
#summary of test set
summary(test)
```

```
##  Diabetes_binary   HighBP         HighChol            BMI        Smoker     Stroke
##  0:7065          Mode :logical   Mode :logical   Min.   :13.00   No :7390   No :13324
##  1:7129          FALSE:6075      FALSE:6792      1st Qu.:25.00   Yes:6804   Yes:  870
##                  TRUE :8119      TRUE :7402      Median :29.00
##                                                  Mean   :29.88
##                                                  3rd Qu.:33.00
##                                                  Max.   :92.00
##
##  HeartDiseaseorAttack PhysActivity      Fruits         Veggies            GenHlth
##  No :12104            Mode :logical   Mode :logical   Mode :logical   Excellent:1655
##  Yes: 2090            FALSE:4164      FALSE:5439      FALSE:2929      Fair     :2594
##                       TRUE :10030     TRUE :8755      TRUE :11265     Good     :4713
##                                                                       Poor     :1213
##                                                                       Very Good:4019
##
##
##     MentHlth         PhysHlth      DiffWalk        Sex                Age
##  Min.   : 0.000   Min.   : 0.000   No :10533   Female:7658   65 to 69   :2189
##  1st Qu.: 0.000   1st Qu.: 0.000   Yes: 3661   Male  :6536   60 to 64   :2023
##  Median : 0.000   Median : 0.000                             55 to 59   :1740
##  Mean   : 3.573   Mean   : 5.768                             70 to 74   :1662
##  3rd Qu.: 2.000   3rd Qu.: 6.000                             50 to 54   :1344
##  Max.   :30.000   Max.   :30.000                             80 or older:1110
##                                                              (Other)    :4126
##                    Education                Income
##  College 1 - 3 years    :4026   > $75k         :4201
##  College 4 years or more:5194   > $50k, < $75k:2273
##  Grade 12 or GED        :3914   > $35k, < $50k:2034
##  Grades 1 - 8           : 348   > $25k, < $35k:1543
##  Grades 9 - 11          : 698   > $20k, < $25k:1323
##  Only kindergarten      :  14   > $15k, < $20k:1176
##                                 (Other)        :1644
```

The data in each set appears to have all factors in each row accounted for.

Logistic Regression Model 1

For the initial model all the columns will be used. It can then be refined in subsequent models using the columns that have the highest impact on the model.

Code: *R*

```
# fit the model initially with all variables
log.mod1 <- glm(Diabetes_binary ~ ., data=train,
                  family='binomial')

# view summary
summary(log.mod1)
```

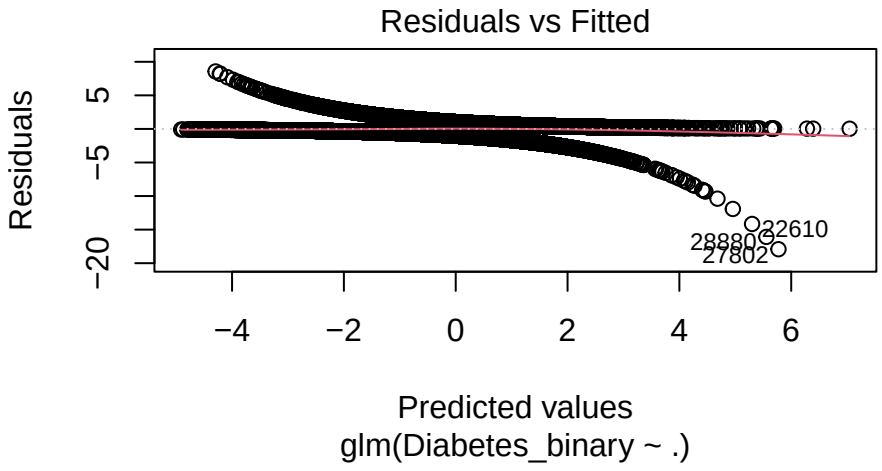
```
##
## Call:
## glm(formula = Diabetes_binary ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3985  -0.8079  -0.1442   0.8287   2.9353
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -5.538243   0.164182 -33.732  < 2e-16 ***
## HighBPTRUE                       0.701882   0.022085  31.781  < 2e-16 ***
## HighCholTRUE                     0.573149   0.021217  27.014  < 2e-16 ***
## BMI                              0.073117   0.001771  41.275  < 2e-16 ***
## SmokerYes                       -0.069634   0.021183  -3.287  0.00101 **
## StrokeYes                        0.177903   0.045235   3.933 8.39e-05 ***
## HeartDiseaseorAttackYes          0.298339   0.031766   9.392  < 2e-16 ***
## PhysActivityTRUE                -0.018069   0.023777  -0.760  0.44729
## FruitsTRUE                      -0.010255   0.021952  -0.467  0.64039
## VeggiesTRUE                     -0.064027   0.026046  -2.458  0.01396 *
## GenHlthFair                      1.896425   0.048737  38.911  < 2e-16 ***
## GenHlthGood                      1.451786   0.042463  34.189  < 2e-16 ***
## GenHlthPoor                      2.035180   0.064501  31.552  < 2e-16 ***
## GenHlthVery Good                 0.720539   0.042892  16.799  < 2e-16 ***
## MentHlth                        -0.003644   0.001408  -2.588  0.00966 **
## PhysHlth                        -0.003132   0.001375  -2.278  0.02271 *
## DiffWalkYes                      0.165308   0.028922   5.716 1.09e-08 ***
## SexMale                          0.278028   0.021490  12.937  < 2e-16 ***
## Age25 to 29                     -0.028558   0.179869  -0.159  0.87385
## Age30 to 34                      0.329191   0.161461   2.039  0.04147 *
## Age35 to 39                      0.694299   0.152502   4.553 5.30e-06 ***
## Age40 to 44                      1.009554   0.149159   6.768 1.30e-11 ***
## Age45 to 49                      1.194904   0.146741   8.143 3.86e-16 ***
## Age50 to 54                      1.412493   0.144903   9.748  < 2e-16 ***
## Age55 to 59                      1.522126   0.144220  10.554  < 2e-16 ***
## Age60 to 64                      1.764460   0.143896  12.262  < 2e-16 ***
## Age65 to 69                      1.925207   0.143931  13.376  < 2e-16 ***
## Age70 to 74                      2.064966   0.144852  14.256  < 2e-16 ***
## Age75 to 79                      2.011261   0.146309  13.747  < 2e-16 ***
## Age80 or older                   1.905747   0.146345  13.022  < 2e-16 ***
## EducationCollege 4 years or more -0.120395   0.026691  -4.511 6.46e-06 ***
## EducationGrade 12 or GED         -0.020320   0.027433  -0.741  0.45886
## EducationGrades 1 - 8             0.127753   0.074212   1.721  0.08517 .
## EducationGrades 9 - 11            0.036656   0.051934   0.706  0.48030
## EducationOnly kindergarten       -0.208906   0.304502  -0.686  0.49268
```

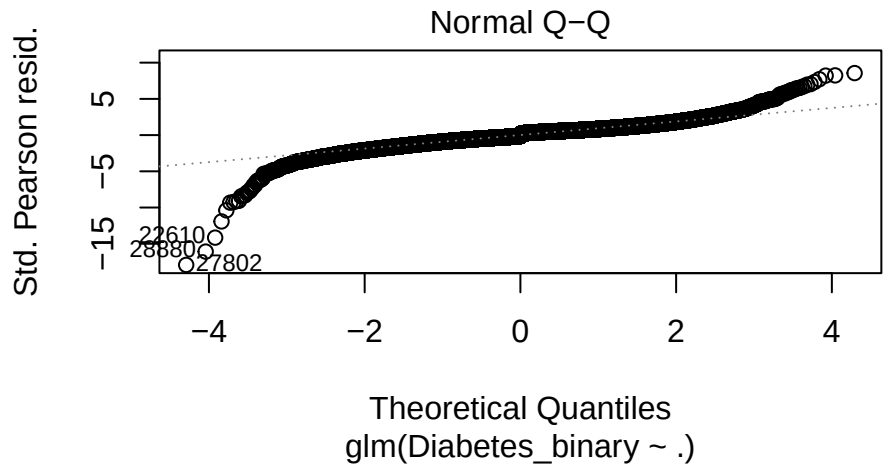
```
## Income> $10k, < $15k              0.006512   0.062552   0.104  0.91709
## Income> $15k, < $20k             -0.072476   0.059835  -1.211  0.22579
## Income> $20k, < $25k             -0.068903   0.057998  -1.188  0.23483
## Income> $25k, < $35k             -0.174497   0.056695  -3.078  0.00209 **
## Income> $35k, < $50k             -0.230611   0.055499  -4.155 3.25e-05 ***
## Income> $50k, < $75k             -0.255364   0.055689  -4.586 4.53e-06 ***
## Income> $75k                     -0.410161   0.054710  -7.497 6.53e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 78323  on 56497  degrees of freedom
## Residual deviance: 57804  on 56456  degrees of freedom
## AIC: 57888
##
## Number of Fisher Scoring iterations: 5
```

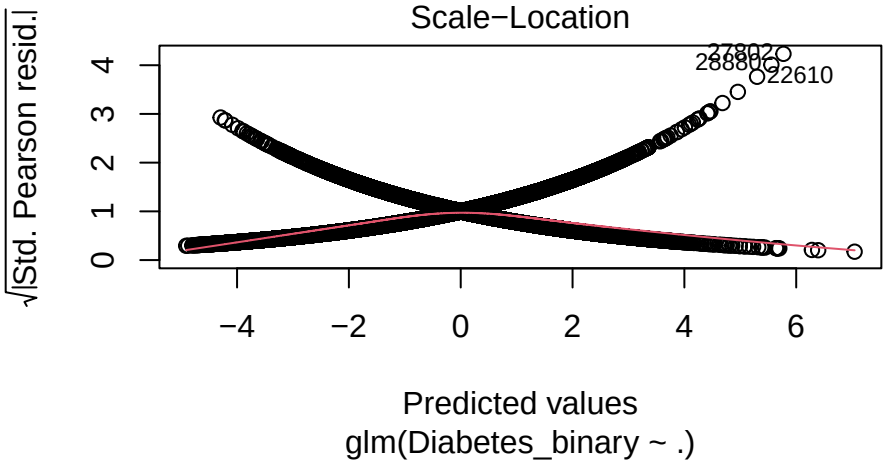
Plot the residuals to see

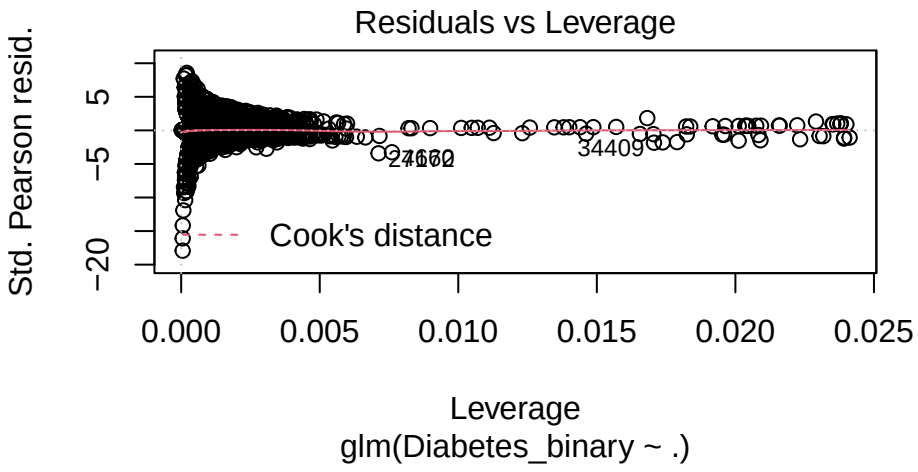
Code: *R*

```
# plotting the results
plot(log.mod1)
```




Normal Q−Q
Std. Pearson resid.
−15
−5
5
22610
28880
27802
−4
−2
0
2
4
Theoretical Quantiles
glm(Diabetes_binary ~ .)


Scale−Location
√|Std. Pearson resid.|
0
1
2
3
4
27802
28880
22610
−4
−2
0
2
4
6
Predicted values
glm(Diabetes_binary ~ .)



View the McFadden Pseudo R^2 Score

Code: *R*

```
# McFadden R2 Score
pscl::pR2(log.mod1)["McFadden"]
```

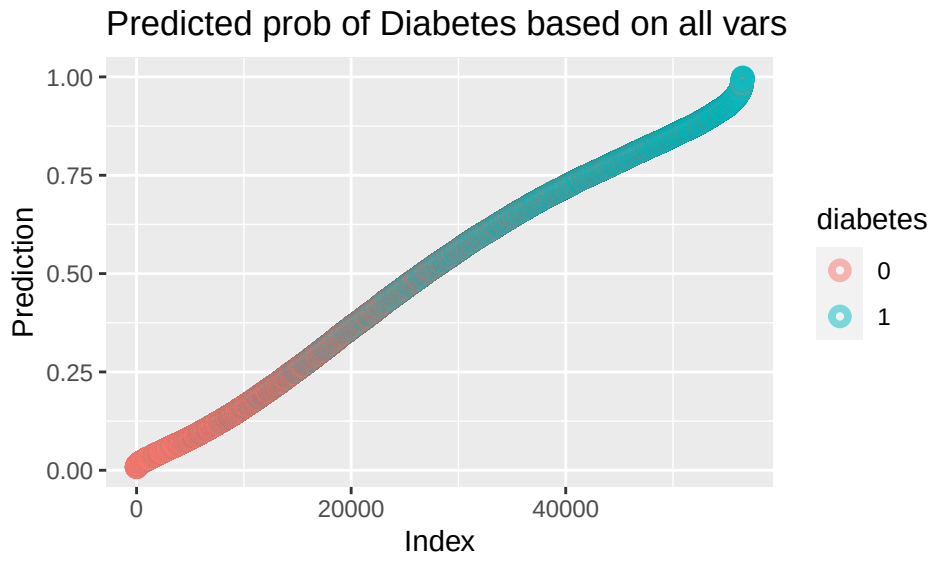
```
## fitting null model for pseudo-r2
```

```
##  McFadden
## 0.2619746
```

Graph the predictions in the training set.

Code: *R*

```
# load predictions to dataframe
pred.data.log1 <- data.frame(
  prob.of.diabetes = log.mod1$fitted.values,
  diabetes = train$Diabetes_binary)

# sort predictions
pred.data.log1 <- pred.data.log1[order(pred.data.log1$prob.of.diabetes, decreasing = FALSE),]

# add rank
pred.data.log1$rank <- 1:nrow(pred.data.log1)

# graph the results
ggplot(data=pred.data.log1, aes(x=rank, y=prob.of.diabetes))+
  geom_point(aes(color=diabetes), alpha=0.5, shape =1, stroke =2) +
  xlab("Index") +
  ylab("Prediction")+
  ggtitle("Predicted prob of Diabetes based on all vars")
```




```
#ggsave('LogReg_all_Vars.png', plot16, path = './graphs/')
```

Next, predictions can be made on the test set and an optimal cut off point found

Code: *R*

```
# make predictions
log.mod1.pred <- predict(log.mod1, test, type="response")

#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(test$Diabetes_binary, log.mod1.pred)[1]
optimal
```

```
## [1] 0.4569913
```

Create a confusion Matrix **Code**: *R*

```
# create confusion matrix of results
InformationValue::confusionMatrix(test$Diabetes_binary,log.mod1.pred)
```

```
##      0    1
## 0 5076 1660
## 1 1989 5469
```

Get the models sensitivity, specificity and miss-classification error rate

Code: *R*

```
#calculate sensitivity
s <- InformationValue::sensitivity(test$Diabetes_binary,log.mod1.pred)

#calculate specificity
sp <- InformationValue::specificity(test$Diabetes_binary,log.mod1.pred)

#calculate total misclassification error rate
me <- InformationValue::misClassError(test$Diabetes_binary,log.mod1.pred, threshold=optimal)

sprintf("The sensitivity of the model: %f", s)
```

```
## [1] "The sensitivity of the model: 0.767148"
```

```
sprintf("The specificity of the model: %f", sp)
```

```
## [1] "The specificity of the model: 0.718471"
```

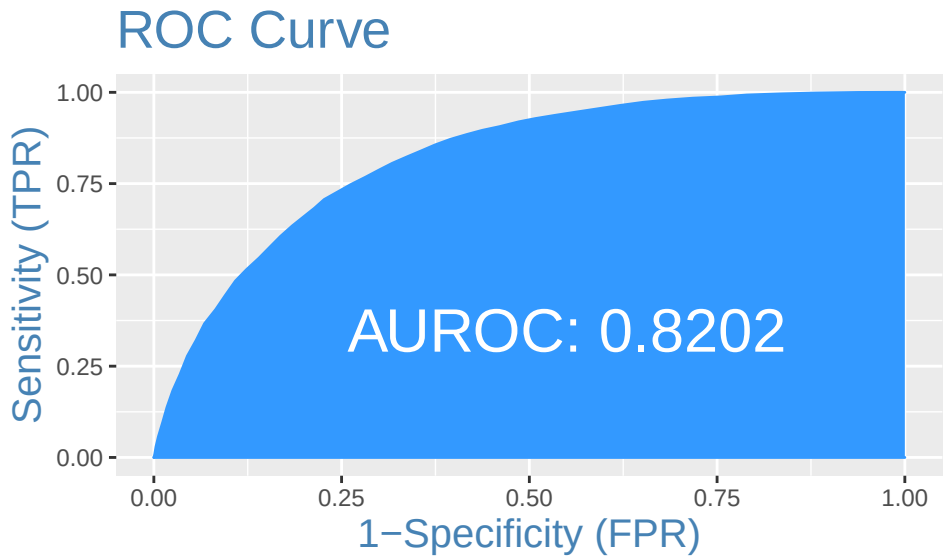
```
sprintf("The total misclassification error rate of the model: %f", me)
```

```
## [1] "The total misclassification error rate of the model: 0.255200"
```

Plot the ROC curve for final evaluation of the model, the graph displays the percentage of true positives predicted by the model as the prediction probability cutoff is lowered from 1 to 0. The higher the AUC (area under the curve), the more accurately our model is able to predict outcomes

Code: *R*

```
# plot the ROC Curve of the model
plotROC(test$Diabetes_binary,log.mod1.pred)
```



Looking at the variable importance and the VIF values of the model to get a better understanding of what is happening in the model.

Code: *R*

```
# variable importance
caret::varImp(log.mod1)
```

```
##                                Overall
## HighBPTRUE                  31.7810226
## HighCholTRUE                27.0136022
## BMI                         41.2751598
## SmokerYes                    3.2872392
## StrokeYes                    3.9328665
## HeartDiseaseorAttackYes      9.3917943
## PhysActivityTRUE             0.7599401
```

```
## FruitsTRUE                          0.4671596
## VeggiesTRUE                         2.4582167
## GenHlthFair                        38.9111945
## GenHlthGood                        34.1890624
## GenHlthPoor                        31.5524424
## GenHlthVery Good                   16.7990335
## MentHlth                            2.5877282
## PhysHlth                            2.2782723
## DiffWalkYes                         5.7156473
## SexMale                            12.9373274
## Age25 to 29                         0.1587723
## Age30 to 34                         2.0388272
## Age35 to 39                         4.5527214
## Age40 to 44                         6.7683162
## Age45 to 49                         8.1429496
## Age50 to 54                         9.7478595
## Age55 to 59                        10.5541979
## Age60 to 64                        12.2620136
## Age65 to 69                        13.3759362
## Age70 to 74                        14.2556660
## Age75 to 79                        13.7466552
## Age80 or older                     13.0223322
## EducationCollege 4 years or more    4.5107868
## EducationGrade 12 or GED            0.7407263
## EducationGrades 1 - 8               1.7214431
## EducationGrades 9 - 11              0.7058217
## EducationOnly kindergarten          0.6860574
## Income> $10k, < $15k                0.1041037
## Income> $15k, < $20k                1.2112667
## Income> $20k, < $25k                1.1880196
## Income> $25k, < $35k                3.0777976
## Income> $35k, < $50k                4.1552252
## Income> $50k, < $75k                4.5855270
## Income> $75k                        7.4969905
```

Code: *R*

```
# vif values
car::vif(log.mod1)
```

```
##                          GVIF Df GVIF^(1/(2*Df))
## HighBP               1.128006  1        1.062076
## HighChol             1.069278  1        1.034059
## BMI                  1.133615  1        1.064714
## Smoker               1.076486  1        1.037539
## Stroke               1.067729  1        1.033310
## HeartDiseaseorAttack 1.132194  1        1.064046
## PhysActivity         1.135049  1        1.065387
## Fruits               1.101592  1        1.049568
## Veggies              1.096760  1        1.047263
## GenHlth              1.875028  4        1.081748
## MentHlth             1.268095  1        1.126097
## PhysHlth             1.764408  1        1.328310
## DiffWalk             1.447522  1        1.203130
## Sex                  1.104062  1        1.050744
## Age                  1.364478 12        1.013033
## Education            1.373510  5        1.032246
## Income               1.585134  7        1.033452
```

Since none of the predictor variables in our models have a VIF over 5, we can assume that multicollinearity is not an issue in our model.

Logistic Regression Model 2

For the initial model all the columns were used. For this model the columns with the highest importance will be used to see if there are any improvements in the model.

Code: *R*

```
# fit the model initially with all high importance columns
log.mod2 <- glm(Diabetes_binary ~ HighBP + HighChol +   BMI +
                  Smoker + Stroke + HeartDiseaseorAttack +
                  GenHlth + Sex + Age,
                data=train,
                family='binomial')

# view summary
summary(log.mod2)
```

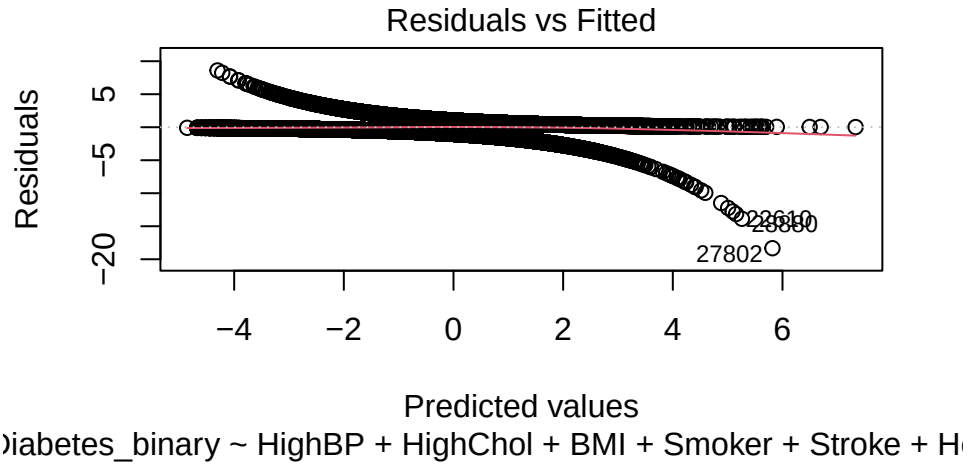
```
##
## Call:
## glm(formula = Diabetes_binary ~ HighBP + HighChol + BMI + Smoker +
##     Stroke + HeartDiseaseorAttack + GenHlth + Sex + Age, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4117  -0.8155  -0.1526   0.8346   2.9385
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -5.962709   0.154282 -38.648  < 2e-16 ***
## HighBPTRUE                0.723282   0.021972  32.919  < 2e-16 ***
## HighCholTRUE              0.563458   0.021095  26.711  < 2e-16 ***
## BMI                       0.075783   0.001743  43.483  < 2e-16 ***
## SmokerYes                -0.032875   0.020795  -1.581    0.114
## StrokeYes                 0.222556   0.044964   4.950 7.43e-07 ***
## HeartDiseaseorAttackYes   0.316673   0.031624  10.014  < 2e-16 ***
## GenHlthFair               2.044371   0.045888  44.551  < 2e-16 ***
## GenHlthGood               1.520028   0.041984  36.205  < 2e-16 ***
## GenHlthPoor               2.217301   0.056108  39.518  < 2e-16 ***
## GenHlthVery Good          0.726615   0.042679  17.025  < 2e-16 ***
## SexMale                   0.216999   0.020699  10.484  < 2e-16 ***
## Age25 to 29              -0.073911   0.180321  -0.410    0.682
## Age30 to 34               0.252461   0.161882   1.560    0.119
## Age35 to 39               0.606237   0.152777   3.968 7.24e-05 ***
## Age40 to 44               0.908610   0.149371   6.083 1.18e-09 ***
## Age45 to 49               1.109228   0.146984   7.547 4.47e-14 ***
## Age50 to 54               1.329204   0.145126   9.159  < 2e-16 ***
## Age55 to 59               1.444968   0.144448  10.003  < 2e-16 ***
## Age60 to 64               1.696744   0.144091  11.776  < 2e-16 ***
## Age65 to 69               1.870861   0.144078  12.985  < 2e-16 ***
## Age70 to 74               2.032437   0.144976  14.019  < 2e-16 ***
## Age75 to 79               2.009697   0.146383  13.729  < 2e-16 ***
## Age80 or older            1.932919   0.146259  13.216  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

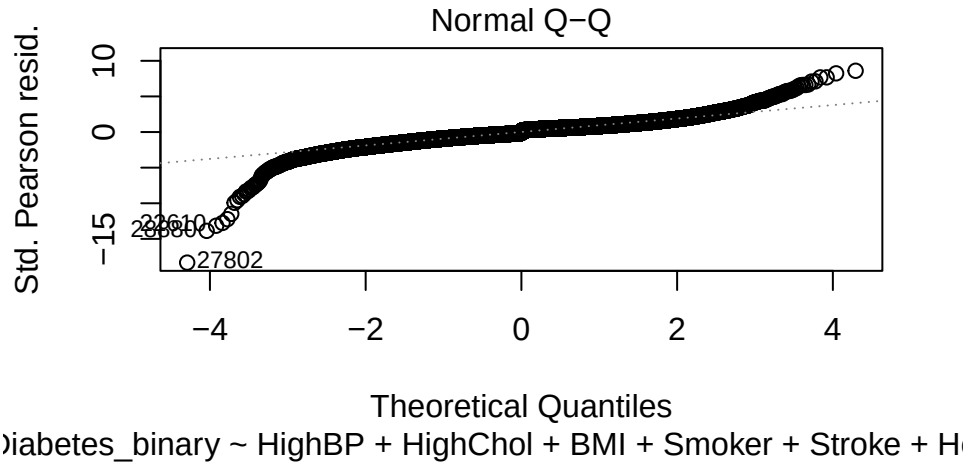
```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 78323  on 56497  degrees of freedom
## Residual deviance: 58130  on 56474  degrees of freedom
## AIC: 58178
##
## Number of Fisher Scoring iterations: 5
```

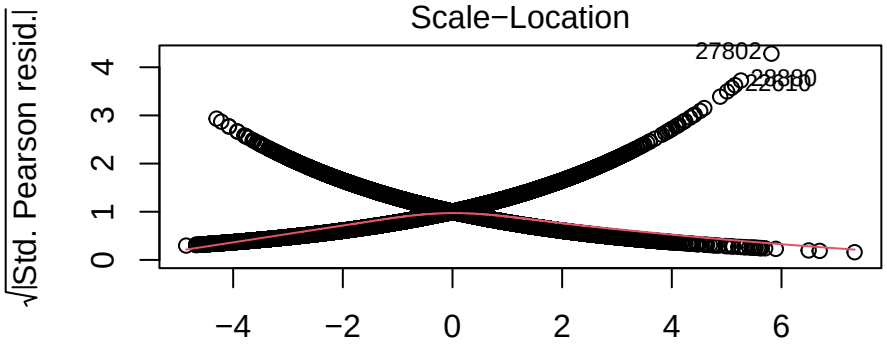
Plot the residuals to see

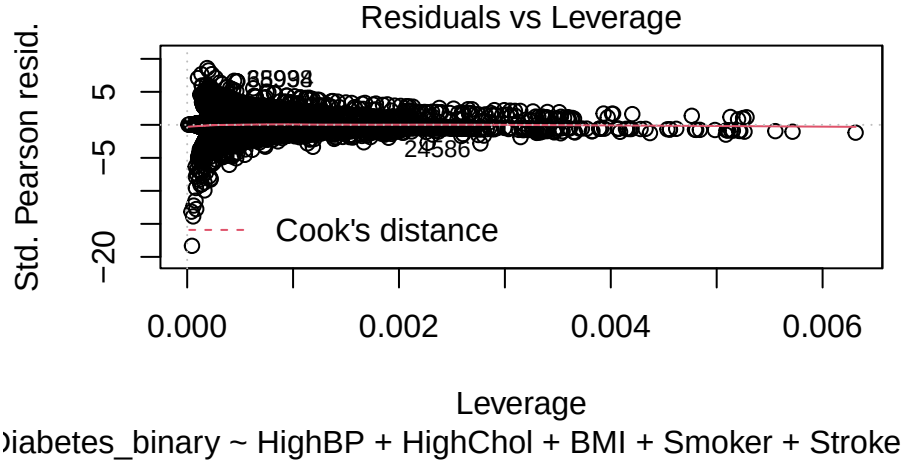
Code: *R*

```
# plotting the results
plot(log.mod2)
```






Scale−Location
√|Std. Pearson resid.|
27802
0 1 2 3 4
−4 −2 0 2 4 6
Predicted values
Diabetes_binary ~ HighBP + HighChol + BMI + Smoker + Stroke + H


Residuals vs Leverage
Std. Pearson resid.
24586
Cook's distance
5 −5 −20
0.000 0.002 0.004 0.006
Leverage
Diabetes_binary ~ HighBP + HighChol + BMI + Smoker + Stroke + H

View the McFadden Pseudo R^2 Score

Code: *R*

```
# McFadden R2 Score
pscl::pR2(log.mod2)["McFadden"]
```

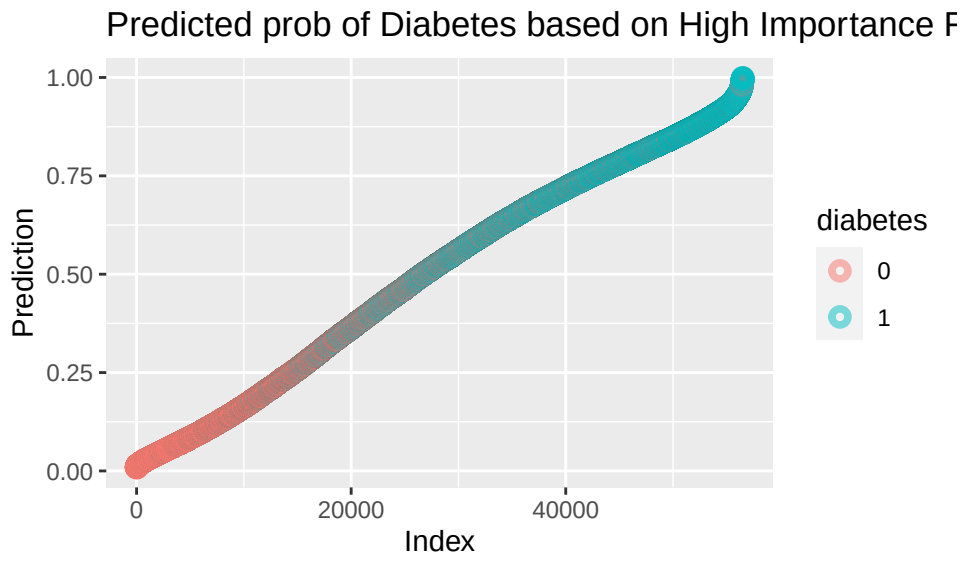
```
## fitting null model for pseudo-r2
```

```
##  McFadden
## 0.2578206
```

Graph the predictions in the training set.

Code: *R*

```
# load predictions to dataframe
pred.data.log2 <- data.frame(
  prob.of.diabetes = log.mod2$fitted.values,
  diabetes = train$Diabetes_binary)

# sort predictions
pred.data.log2 <- pred.data.log2[order(pred.data.log2$prob.of.diabetes, decreasing = FALSE),]

# add rank
pred.data.log2$rank <- 1:nrow(pred.data.log2)

# graph the results
ggplot(data=pred.data.log2, aes(x=rank, y=prob.of.diabetes))+
  geom_point(aes(color=diabetes), alpha=0.5, shape =1, stroke =2) +
  xlab("Index") +
  ylab("Prediction")+
  ggtitle("Predicted prob of Diabetes based on High Importance Factors")
```




```
#ggsave('LogReg_all_Vars.png', plot16, path = './graphs/')
```

Next, predictions can be made on the test set and an optimal cut off point found

Code: *R*

```
# make predictions
log.mod2.pred <- predict(log.mod2, test, type="response")


#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(test$Diabetes_binary, log.mod2.pred)[1]
optimal
```

```
## [1] 0.4969764
```

Create a confusion Matrix

Code: *R*

```
# create confusion matrix of results
InformationValue::confusionMatrix(test$Diabetes_binary,log.mod2.pred)
```

```
##      0    1
## 0 5055 1645
## 1 2010 5484
```

Get the models sensitivity, specificity and miss-classification error rate

Code: *R*

```
#calculate sensitivity
s <- InformationValue::sensitivity(test$Diabetes_binary,log.mod2.pred)

#calculate specificity
sp <- InformationValue::specificity(test$Diabetes_binary,log.mod2.pred)

#calculate total misclassification error rate
me <- InformationValue::misClassError(test$Diabetes_binary,log.mod2.pred, threshold=optimal)

sprintf("The sensitivity of the model: %f", s)
```

```
## [1] "The sensitivity of the model: 0.769252"
```

```
sprintf("The specificity of the model: %f", sp)
```

```
## [1] "The specificity of the model: 0.715499"
```

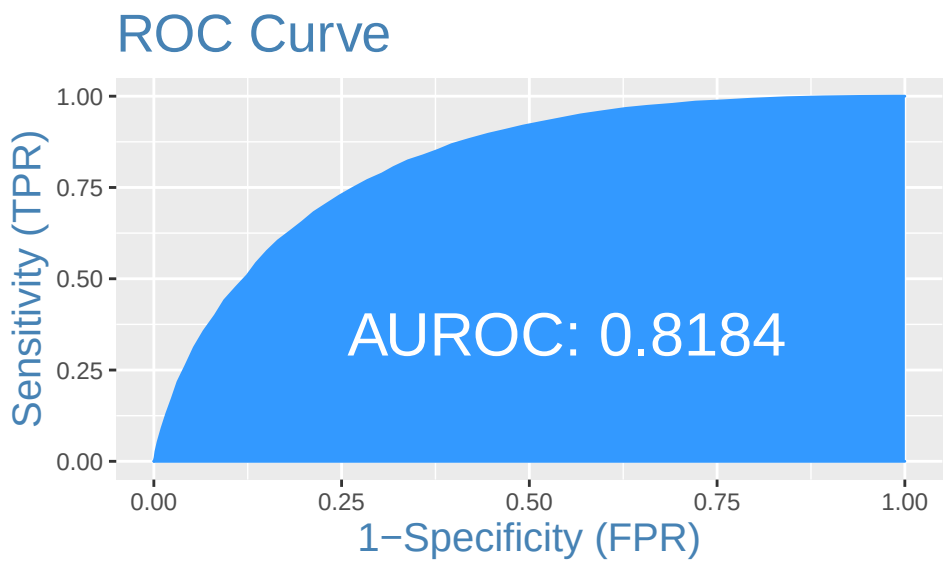
```
sprintf("The total misclassification error rate of the model: %f", me)
```

```
## [1] "The total misclassification error rate of the model: 0.256800"
```

Plot the ROC curve for final evaluation of the model, the graph displays the percentage of true positives predicted by the model as the prediction probability cutoff is lowered from 1 to 0. The higher the AUC (area under the curve), the more accurately our model is able to predict outcomes

Code: *R*

```
# plot the ROC Curve of the model
plotROC(test$Diabetes_binary,log.mod2.pred)
```



Looking at the variable importance and the VIF values of the model to get a better understanding of what is happening in the model.

Code: *R*

```
# variable importance
caret::varImp(log.mod2)
```

```
##                            Overall
## HighBPTRUE              32.9189099
## HighCholTRUE            26.7108343
## BMI                     43.4834258
## SmokerYes                1.5809598
## StrokeYes                4.9496679
## HeartDiseaseorAttackYes 10.0136488
## GenHlthFair             44.5509407
## GenHlthGood             36.2052935
## GenHlthPoor             39.5181214
## GenHlthVery Good        17.0251339
## SexMale                 10.4836393
## Age25 to 29              0.4098857
## Age30 to 34              1.5595315
## Age35 to 39              3.9681139
## Age40 to 44              6.0828960
## Age45 to 49              7.5466023
## Age50 to 54              9.1589438
## Age55 to 59             10.0033548
## Age60 to 64             11.7755316
```

```
## Age65 to 69             12.9850775
## Age70 to 74             14.0191183
## Age75 to 79             13.7289854
## Age80 or older          13.2157616
```

Code: *R*

```
# vif values
car::vif(log.mod2)
```

```
##                           GVIF Df GVIF^(1/(2*Df))
## HighBP                1.123957  1        1.060168
## HighChol              1.064071  1        1.031538
## BMI                   1.101553  1        1.049549
## Smoker                1.044255  1        1.021888
## Stroke                1.058377  1        1.028774
## HeartDiseaseorAttack 1.125943  1        1.061104
## GenHlth               1.116342  4        1.013852
## Sex                   1.030418  1        1.015095
## Age                   1.201012 12        1.007661
```

Since none of the predictor variables in our models have a VIF over 5, we can assume that multicollinearity is not an issue in our model.

Logistic Regression Model 3

For the initial model all the columns were used. For the second model all the columns with the highest importance were used to see if there are any improvements in the model, there was a decline in the performance. Now a subset of the highest importance columns will be used. **HighBP**, **HighChol**, **HeartDiseaseorAttack**, **Veggies**, **MenthHlth**, **DiffWalk**, **Sex**, **Age** , **Education**, **Income**, and **GenHlth**

Code: *R*

```
# fit the model initially with all high importance columns
log.mod3 <- glm(Diabetes_binary ~ HighBP + HighChol + HeartDiseaseorAttack +
                  Veggies + GenHlth + MentHlth + DiffWalk + Sex +
                  Age + Income + BMI + Education + Income,
                data=train,
                family='binomial')

# view summary
summary(log.mod3)
```

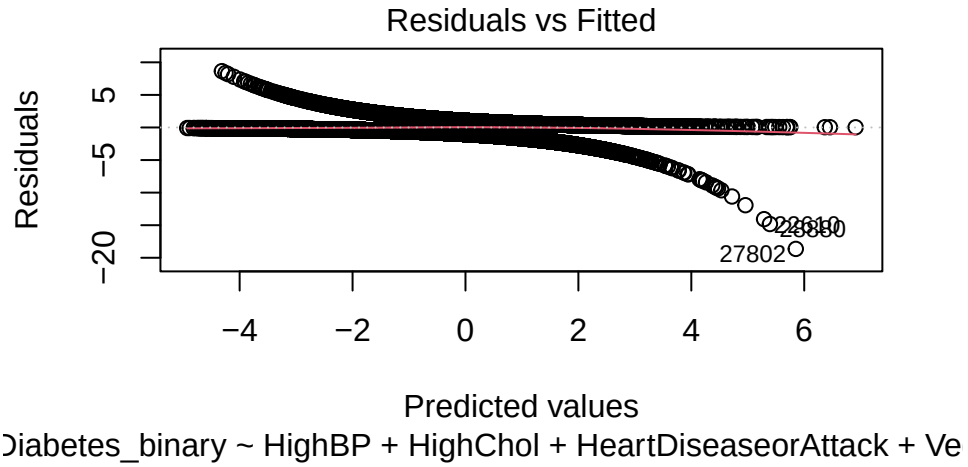
```
##
## Call:
## glm(formula = Diabetes_binary ~ HighBP + HighChol + HeartDiseaseorAttack +
##     Veggies + GenHlth + MentHlth + DiffWalk + Sex + Age + Income +
##     BMI + Education + Income, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4216  -0.8102  -0.1433   0.8295   2.9421
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -5.569449   0.162017 -34.376  < 2e-16 ***
```

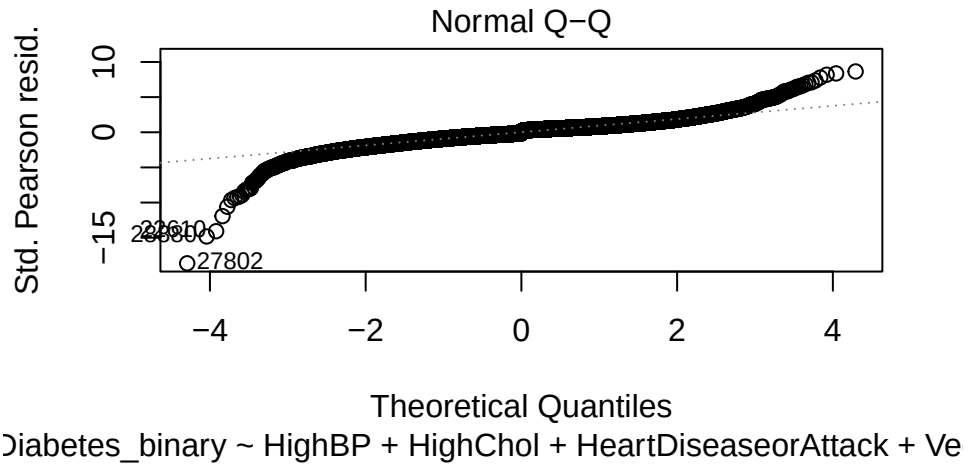
```
## HighBPTRUE                          0.705494   0.022064  31.975  < 2e-16 ***
## HighCholTRUE                        0.572384   0.021187  27.015  < 2e-16 ***
## HeartDiseaseorAttackYes             0.309160   0.031445   9.832  < 2e-16 ***
## VeggiesTRUE                        -0.071569   0.025315  -2.827  0.00470 **
## GenHlthFair                         1.876637   0.047607  39.419  < 2e-16 ***
## GenHlthGood                         1.447033   0.042366  34.156  < 2e-16 ***
## GenHlthPoor                         1.993208   0.060117  33.155  < 2e-16 ***
## GenHlthVery Good                    0.718530   0.042884  16.755  < 2e-16 ***
## MentHlth                           -0.004403   0.001378  -3.196  0.00139 **
## DiffWalkYes                         0.155376   0.027809   5.587 2.31e-08 ***
## SexMale                             0.269627   0.021216  12.709  < 2e-16 ***
## Age25 to 29                        -0.036904   0.179836  -0.205  0.83741
## Age30 to 34                         0.317225   0.161403   1.965  0.04937 *
## Age35 to 39                         0.681215   0.152389   4.470 7.81e-06 ***
## Age40 to 44                         0.998187   0.149078   6.696 2.15e-11 ***
## Age45 to 49                         1.181933   0.146651   8.059 7.66e-16 ***
## Age50 to 54                         1.397876   0.144779   9.655  < 2e-16 ***
## Age55 to 59                         1.507330   0.144095  10.461  < 2e-16 ***
## Age60 to 64                         1.750307   0.143759  12.175  < 2e-16 ***
## Age65 to 69                         1.909063   0.143763  13.279  < 2e-16 ***
## Age70 to 74                         2.048084   0.144656  14.158  < 2e-16 ***
## Age75 to 79                         1.999687   0.146112  13.686  < 2e-16 ***
## Age80 or older                      1.900949   0.146154  13.007  < 2e-16 ***
## Income> $10k, < $15k                0.003567   0.062463   0.057  0.95446
## Income> $15k, < $20k               -0.073299   0.059755  -1.227  0.21995
## Income> $20k, < $25k               -0.073310   0.057916  -1.266  0.20558
## Income> $25k, < $35k               -0.177372   0.056609  -3.133  0.00173 **
## Income> $35k, < $50k               -0.235047   0.055414  -4.242 2.22e-05 ***
## Income> $50k, < $75k               -0.259591   0.055606  -4.668 3.04e-06 ***
## Income> $75k                       -0.413995   0.054615  -7.580 3.45e-14 ***
## BMI                                 0.073319   0.001762  41.617  < 2e-16 ***
## EducationCollege 4 years or more -0.113132   0.026535  -4.264 2.01e-05 ***
## EducationGrade 12 or GED         -0.019387   0.027381  -0.708  0.47892
## EducationGrades 1 - 8             0.136370   0.074115   1.840  0.06577 .
## EducationGrades 9 - 11            0.037190   0.051820   0.718  0.47296
## EducationOnly kindergarten       -0.179424   0.304301  -0.590  0.55544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 78323  on 56497  degrees of freedom
## Residual deviance: 57836  on 56461  degrees of freedom
## AIC: 57910
##
## Number of Fisher Scoring iterations: 5
```

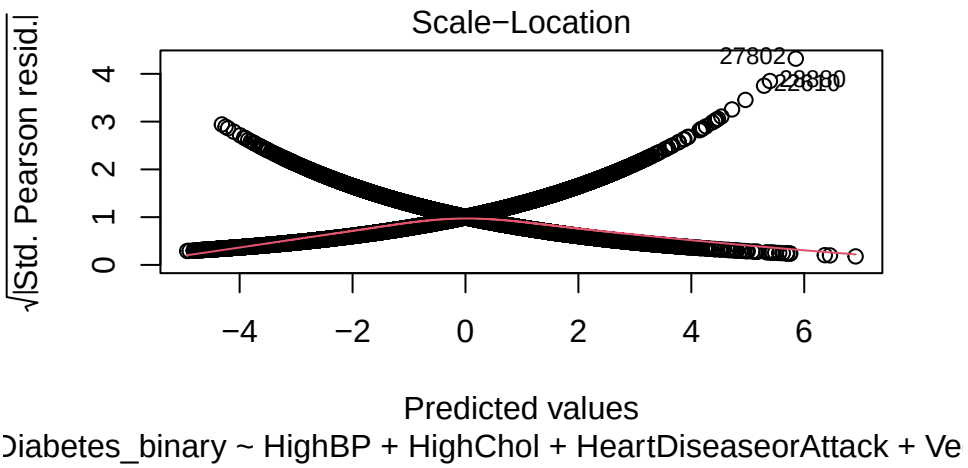
Plot the residuals to see

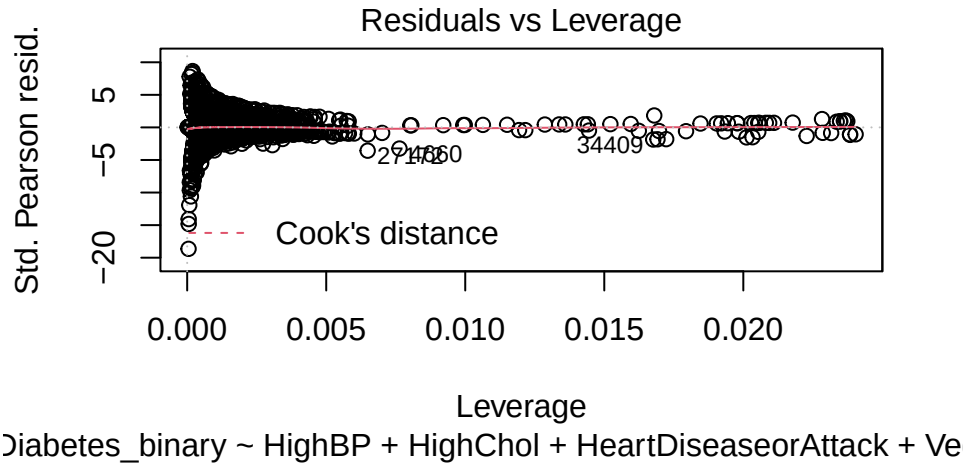
Code: *R*

```
# plotting the results
plot(log.mod3)
```









View the McFadden Pseudo R^2 Score

Code: *R*

```
# McFadden R2 Score
pscl::pR2(log.mod3)["McFadden"]
```

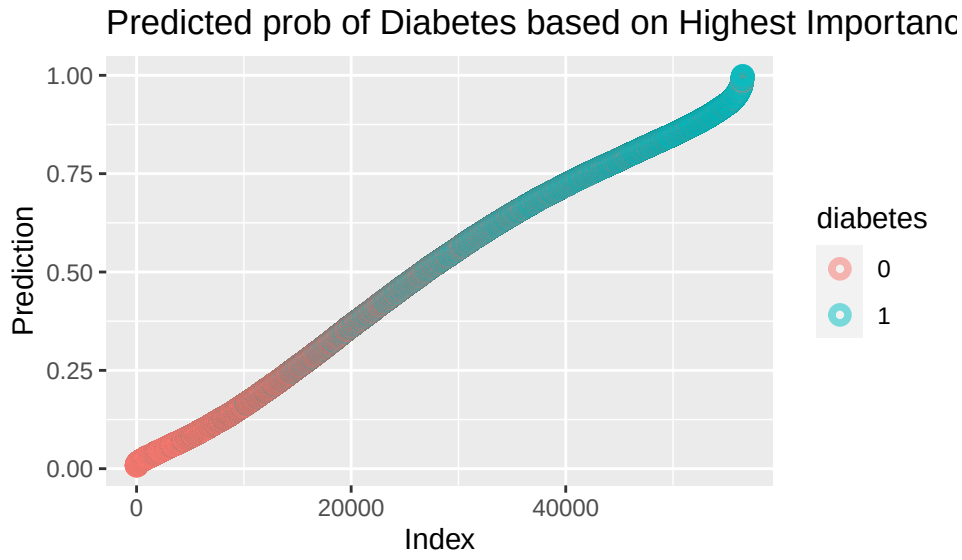
```
## fitting null model for pseudo-r2
```

```
##  McFadden
## 0.2615656
```

Graph the predictions in the training set.

Code: *R*

```
# load predictions to dataframe
pred.data.log3 <- data.frame(
  prob.of.diabetes = log.mod3$fitted.values,
  diabetes = train$Diabetes_binary)

# sort predictions
pred.data.log3 <- pred.data.log3[order(pred.data.log3$prob.of.diabetes, decreasing = FALSE),]

# add rank
pred.data.log3$rank <- 1:nrow(pred.data.log3)

# graph the results
ggplot(data=pred.data.log3, aes(x=rank, y=prob.of.diabetes))+
  geom_point(aes(color=diabetes), alpha=0.5, shape =1, stroke =2) +
  xlab("Index") +
  ylab("Prediction")+
  ggtitle("Predicted prob of Diabetes based on Highest Importance Factors")
```




```
#ggsave('LogReg_all_Vars.png', plot16, path = './graphs/')
```

Next, predictions can be made on the test set and an optimal cut off point found

Code: *R*

```
# make predictions
log.mod3.pred <- predict(log.mod3, test, type="response")


#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(test$Diabetes_binary, log.mod3.pred)[1]
optimal
```

```
## [1] 0.4471672
```

Create a confusion Matrix

Code: *R*

```
# create confusion matrix of results
InformationValue::confusionMatrix(test$Diabetes_binary,log.mod3.pred)
```

```
##      0    1
## 0 5068 1648
## 1 1997 5481
```

Get the models sensitivity, specificity and miss-classification error rate

Code: *R*

```
#calculate sensitivity
s <- InformationValue::sensitivity(test$Diabetes_binary,log.mod3.pred)

#calculate specificity
sp <- InformationValue::specificity(test$Diabetes_binary,log.mod3.pred)

#calculate total misclassification error rate
me <- InformationValue::misClassError(test$Diabetes_binary,log.mod3.pred, threshold=optimal)

sprintf("The sensitivity of the model: %f", s)
```

```
## [1] "The sensitivity of the model: 0.768832"
```

```
sprintf("The specificity of the model: %f", sp)
```

```
## [1] "The specificity of the model: 0.717339"
```

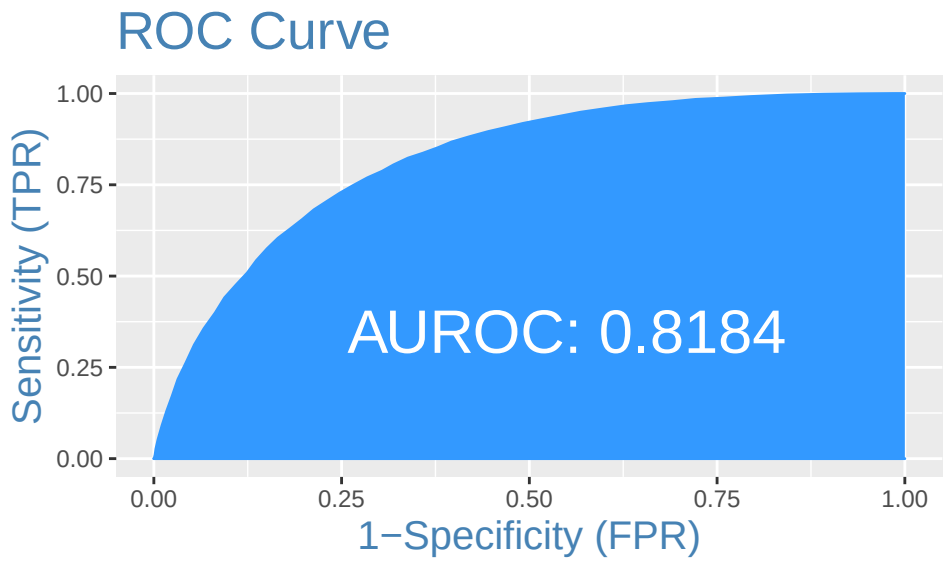
```
sprintf("The total misclassification error rate of the model: %f", me)
```

```
## [1] "The total misclassification error rate of the model: 0.255600"
```

Plot the ROC curve for final evaluation of the model, the graph displays the percentage of true positives predicted by the model as the prediction probability cutoff is lowered from 1 to 0. The higher the AUC (area under the curve), the more accurately our model is able to predict outcomes

Code: *R*

```
# plot the ROC Curve of the model
plotROC(test$Diabetes_binary,log.mod2.pred)
```



Best model from logistic regression is the first model that uses all the variables in the dataset to make a prediction. The overall AUROC score for the best model is: 0.8202 or 82.02%

Random Forest

Random Forest can be used to attempt to predict the presence of Diabetes in people. The random forest will be run in a default state with no parameters to begin with, then an attempt to tune the model will be performed and the tested for accuracy, specificity. miss-classification error and ROC.

The same training and test sets created for the logistic regression model can be used in this model while using the factors in their original state.

Code: *R*

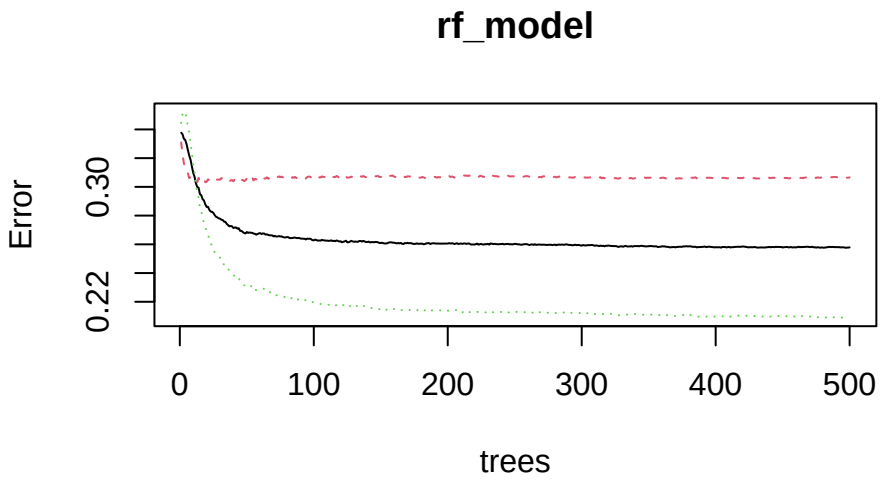
```
# fir the random forest model
rf_model <- randomForest(formula = Diabetes_binary~.,data=train)

# output the model
rf_model
```

```
##
## Call:
##  randomForest(formula = Diabetes_binary ~ ., data = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 25.79%
## Confusion matrix:
##       0     1 class.error
## 0 19611  8670   0.3065662
## 1  5899 22318   0.2090584
```

The rf used a classification type with 500 trees using a 4 random variables at each point

Code: *R*

```
# plot the model
plot(rf_model)
```

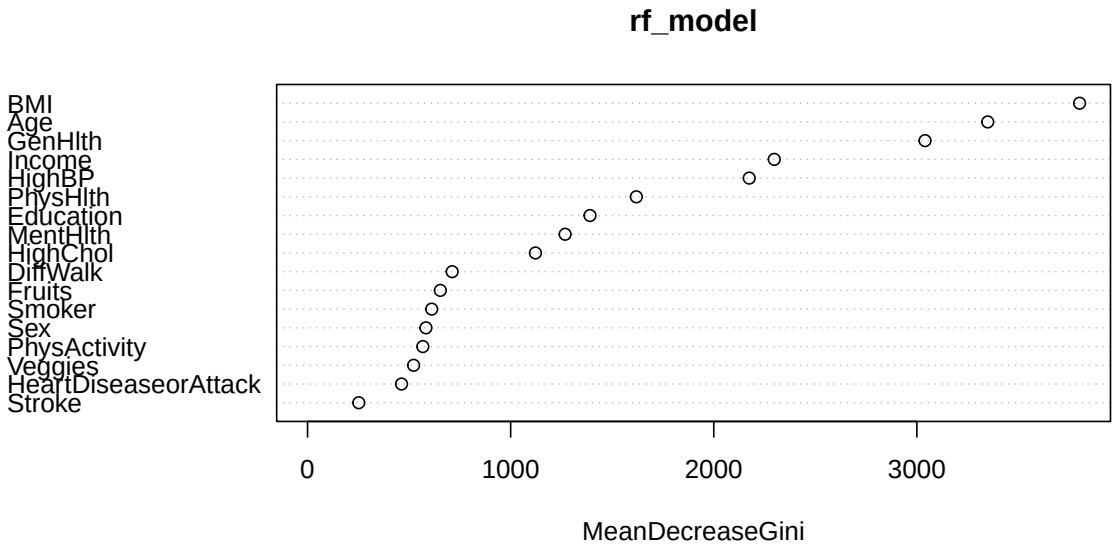


```
#varImpPlot(fit.rf)
```

The graph shows that after approx 100 trees the error rate only shows minor decline

Code: *R*

```
# plot the variabel importance in the model
varImpPlot(rf_model)
```

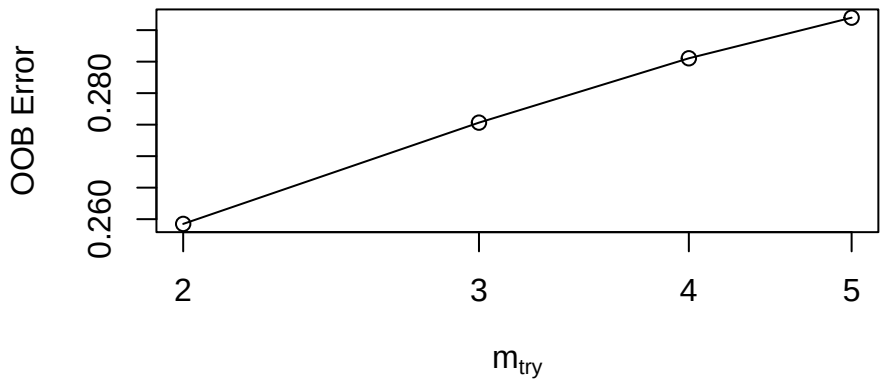


The graph shows that BMI, AGE, GenHlth, Income and HighBP have the highest weighting on the model.

Code: *R*

```
model_tuned <- tuneRF(
               x=train[c('BMI', 'Age', 'GenHlth','Income', 'HighBP')], #define predictor variables
               y=train$Diabetes_binary, #define response variable
               ntreeTry=500,
               mtryStart=4,
               stepFactor=1.5,
               improve=0.01,
               trace=FALSE #don't show real-time progress
               )
```

```
## 0.03576742 0.01
## 0.05837351 0.01
## -0.1262375 0.01
```



The graph shows that the nest OOB is achieved with 2 random features as per the default.

Code: *R*

```
rf_pred <- predict(rf_model,newdata=test,type="prob")[,2]

#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(test$Diabetes_binary, rf_pred)[1]
optimal
```

```
## [1] 0.53
```

Code: *R*

```
#calculate sensitivity
s <- InformationValue::sensitivity(test$Diabetes_binary,rf_pred)

#calculate specificity
sp <- InformationValue::specificity(test$Diabetes_binary,rf_pred)

#calculate total misclassification error rate
me <- InformationValue::misClassError(test$Diabetes_binary,rf_pred, threshold=optimal)

sprintf("The sensitivity of the model: %f", s)
```

```
## [1] "The sensitivity of the model: 0.792538"
```

```
sprintf("The specificity of the model: %f", sp)
```

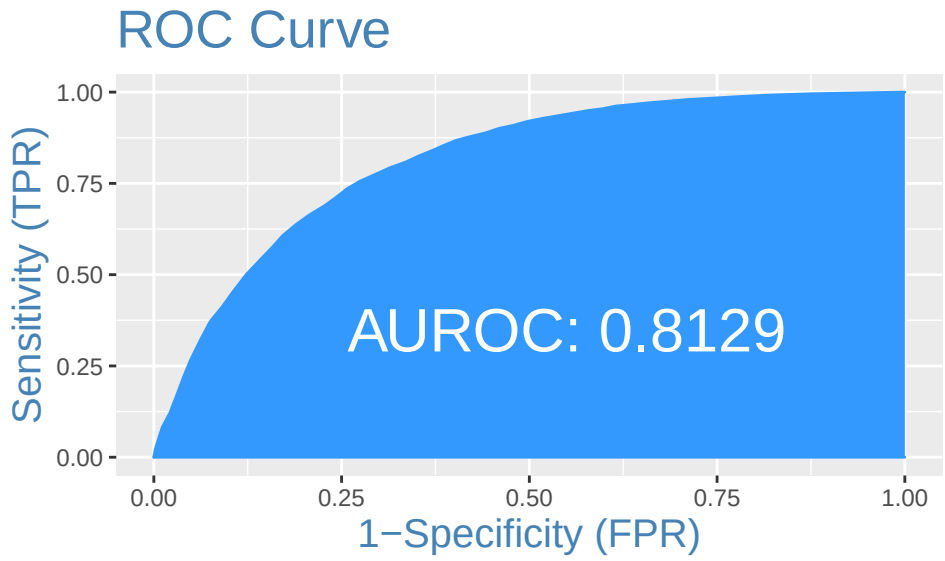
```
## [1] "The specificity of the model: 0.685775"
```

```
sprintf("The total misclassification error rate of the model: %f", me)
```

```
## [1] "The total misclassification error rate of the model: 0.259100"
```

Code: *R*

```
# plot the ROC Curve of the model
plotROC(test$Diabetes_binary,rf_pred)
```



The tuned random forest model returned an overall AUROC score for the tuned model is: 0.8128 or 81.28%

KNN

K-Nearest Neighbor will be tested as a possible model for predicting the classification of diabetes in a person. A new data will be copied from the original but using the same columns as the previous models. the original dat set is used as numeric values are needed for the algorithm. As most of the numeric values are 1 and 0 they will be tested both scaled and not scaled versions to see which best applies to the model.

Code: *R*

```
# copy of data
knn_data <- data

# remove some columns with excess factor types of model
knn_data <- knn_data %>% dplyr::select(Diabetes_binary, HighBP, HighChol, BMI,
                                       Smoker, Stroke, HeartDiseaseorAttack, PhysActivity,
                                       Fruits, Veggies, GenHlth, MentHlth, PhysHlth, DiffWalk,
                                       Sex, Age, Education, Income )

# change factor
knn_data$Diabetes_binary <- as.factor(knn_data$Diabetes_binary)

# check data types
str(knn_data)
```

```
## 'data.frame':    70692 obs. of  18 variables:
##  $ Diabetes_binary     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ HighBP              : num  1 1 0 1 0 0 0 0 0 0 ...
##  $ HighChol            : num  0 1 0 1 0 0 1 0 0 0 ...
##  $ BMI                 : num  26 26 26 28 29 18 26 31 32 27 ...
##  $ Smoker              : num  0 1 0 1 1 0 1 1 0 1 ...
##  $ Stroke              : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ HeartDiseaseorAttack: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PhysActivity        : num  1 0 1 1 1 1 1 0 1 0 ...
##  $ Fruits              : num  0 1 1 1 1 1 1 1 1 1 ...
##  $ Veggies             : num  1 0 1 1 1 1 1 1 1 1 ...
##  $ GenHlth             : num  3 3 1 3 2 2 1 4 3 3 ...
##  $ MentHlth            : num  5 0 0 0 0 7 0 0 0 0 ...
##  $ PhysHlth            : num  30 0 10 3 0 0 0 0 0 6 ...
##  $ DiffWalk            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sex                 : num  1 1 1 1 0 0 1 1 0 1 ...
##  $ Age                 : num  4 12 13 11 8 1 13 6 3 6 ...
##  $ Education           : num  6 6 6 6 5 4 5 4 6 4 ...
##  $ Income              : num  8 8 8 8 8 7 6 3 8 4 ...
```

The below function is used to down sample version of the data set, exact same approach as with the chi-squared tests previous.

Code: *R*

```
# make a copy of the dataset
samp_data <- knn_data

# set the sample size we want
sample_size = 20000 #

# set seed for reporducabilty
set.seed(1)

# parameters for the sample set
```

```
idxs = sample(1:nrow(samp_data),sample_size,replace=F)

# create the sample set
subsample = samp_data[idxs,]
# list for p-values
pvalues = list()

# loop through the dataset columns and test each column
# an alpha 0.05 is used and the p-value is used to select the
# the corresponding distribution in the sample set
for (col in names(samp_data)) {
  if (class(samp_data[,col]) %in% c("numeric","integer")) {
    # Numeric variable. Using Kolmogorov-Smirnov test
    pvalues[[col]] = ks.test(samp_data[[col]],samp_data[[col]])$p.value

  } else {
    # Categorical variable. Using Pearson's Chi-square test
    probs = table(samp_data[[col]])/nrow(samp_data)
    pvalues[[col]] = chisq.test(table(subsample[[col]]),p=probs)$p.value

  }
}

# convert the Diabetes diagnosis to a Yes / No
subsample <- subsample %>%
      mutate(Diabetes_binary = ifelse(Diabetes_binary == 1,"Yes","No"))
```

A training and test set are created and the values checked to make sure it has all worked. The data (except for the dependent variable) will be scaled at this point.

Code: *R*

```
knn_data1 <- knn_data %>%
      mutate(Diabetes_binary = ifelse(Diabetes_binary == 1,"Yes","No"))

#Use 70% of data set as training set and remaining 30% as testing set
sample <- sample(c(TRUE, FALSE), nrow(subsample), replace=TRUE, prob=c(0.8,0.2))
train_samp <- subsample[sample, ]
test_samp <- knn_data1[!sample, ]

# Scale the new numerical values
train_samp[-1] = scale(train_samp[-1])
test_samp[-1] = scale(test_samp[-1])

# show the data types
str(train_samp)
```

```
## 'data.frame':    15897 obs. of  18 variables:
##  $ Diabetes_binary     : chr  "No" "Yes" "Yes" "No" ...
##  $ HighBP              : num  0.881 0.881 0.881 0.881 -1.134 ...
##  $ HighChol            : num  -1.049 0.954 -1.049 0.954 -1.049 ...
##  $ BMI                 : num  -0.539 -1.529 0.875 -1.246 0.592 ...
##  $ Smoker              : num  -0.959 1.042 1.042 1.042 -0.959 ...
##  $ Stroke              : num  -0.26 3.85 -0.26 -0.26 -0.26 ...
##  $ HeartDiseaseorAttack: num  -0.418 2.39 -0.418 -0.418 -0.418 ...
##  $ PhysActivity        : num  0.643 0.643 0.643 0.643 -1.556 ...
##  $ Fruits              : num  0.791 0.791 0.791 0.791 -1.265 ...
##  $ Veggies             : num  0.513 0.513 -1.951 0.513 0.513 ...
```

```
##  $ GenHlth             : num  0.144 1.942 0.144 -0.755 0.144 ...
##  $ MentHlth            : num  -0.339 -0.462 -0.462 -0.462 -0.462 ...
##  $ PhysHlth            : num  -0.572 0.924 -0.273 -0.472 -0.173 ...
##  $ DiffWalk            : num  -0.582 1.718 1.718 -0.582 -0.582 ...
##  $ Sex                 : num  -0.912 1.096 1.096 1.096 1.096 ...
##  $ Age                 : num  0.145 0.145 0.843 1.192 -1.25 ...
##  $ Education           : num  1.046 -1.854 1.046 1.046 -0.887 ...
##  $ Income              : num  0.598 -0.774 1.056 -0.317 1.056 ...
```

Next step is to create and train the model.

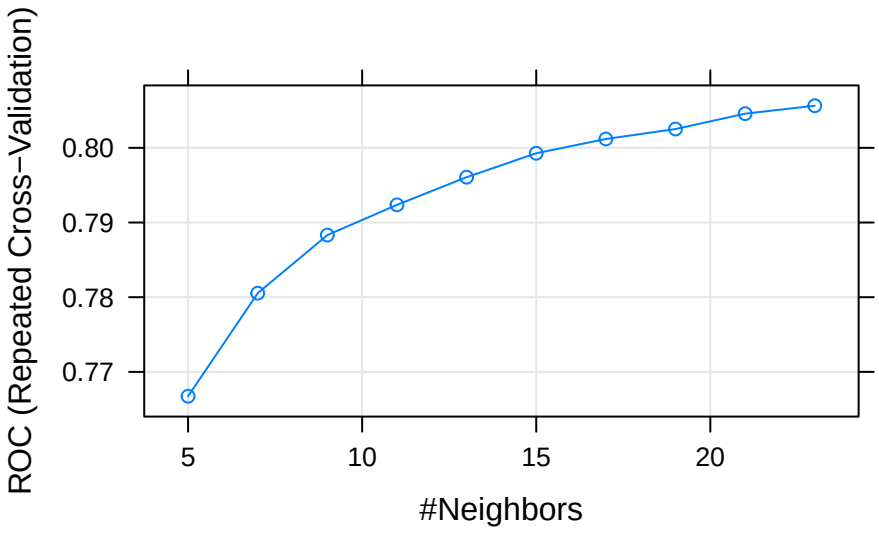
Code: *R*

```r
# Setting up train controls
repeats = 3
numbers = 10
tunel = 10

# set seed
set.seed(1)

# model parameters
x <- trainControl(method = "repeatedcv",
                  number = numbers,
                  repeats = repeats,
                  classProbs = TRUE,
                  summaryFunction = twoClassSummary)

# Run the kNN
model1 <- train(Diabetes_binary~. , data = train_samp, method = "knn",
                preProcess = c("center","scale"),
                trControl = x,
                metric = "ROC",
                tuneLength = tunel)

# Summary of model
model1
```

```
## k-Nearest Neighbors
##
## 15897 samples
##    17 predictor
##     2 classes: 'No', 'Yes'
##
## Pre-processing: centered (17), scaled (17)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 14308, 14307, 14308, 14306, 14307, 14307, ...
## Resampling results across tuning parameters:
##
##   k   ROC        Sens       Spec
##    5  0.7667386  0.6897832  0.7380560
##    7  0.7805412  0.6920113  0.7552128
##    9  0.7883173  0.6928105  0.7613227
##   11  0.7923657  0.6930628  0.7647116
##   13  0.7960702  0.6937347  0.7686445
##   15  0.7992756  0.6937763  0.7719920
##   17  0.8011852  0.6920110  0.7740021
##   19  0.8025203  0.6925995  0.7778107
##   21  0.8045785  0.6928520  0.7802377
```

```
##   23  0.8056383  0.6939029  0.7821202
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 23.
```

```
# plot the output
plot(model1)
```



make predictions against the test set

Code: *R*

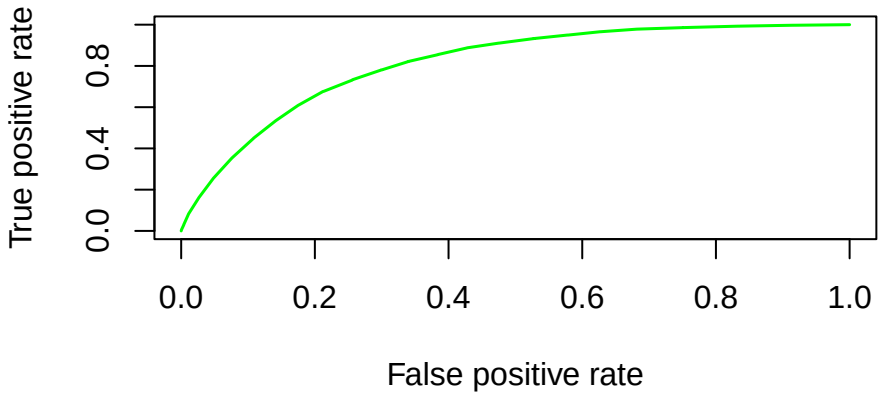
```
# make predictions on the test set
valid_pred <- predict(model1,test_samp, type = "prob")
```

Code: *R*

```
# get the prediction values and the test set values
pred_val <- prediction(valid_pred[,2],test_samp$Diabetes_binary)

# Calculating Area under Curve (AUC)
perf_val <- performance(pred_val,"auc")
perf_val
```

```
## A performance instance
##   'Area under the ROC curve'
```

```
# Plot AUC
perf_val <- performance(pred_val, "tpr", "fpr")
plot(perf_val, col = "green", lwd = 1.5)
```



ROC graph shows an approx 0.8 AU-ROC it can be checked by the same process as prev models with a more accurate graph and read out

Code: *R*

```
# get predictions again but output only the probabilities to list
valid_pred1 <- predict(model1,test_samp, type = "prob")[,2]

# make a copy of the test set
test1 <- test_samp

# change the diabetic vairable back to a 1 or 0
test1 <- test1 %>%
      mutate(Diabetes_binary = ifelse(Diabetes_binary == "Yes",1,0))

#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(test1$Diabetes_binary, valid_pred)[1]
optimal
```

```
## [1] 0.13
```

Code: *R*

```
#calculate sensitivity
s <- InformationValue::sensitivity(test1$Diabetes_binary,valid_pred1)

#calculate specificity
sp <- InformationValue::specificity(test1$Diabetes_binary,valid_pred1)

#calculate total misclassification error rate
me <- InformationValue::misClassError(test1$Diabetes_binary,valid_pred1, threshold=0.5)

sprintf("The sensitivity of the model: %f", s)
```

```
## [1] "The sensitivity of the model: 0.781951"
```

```r
sprintf("The specificity of the model: %f", sp)
```

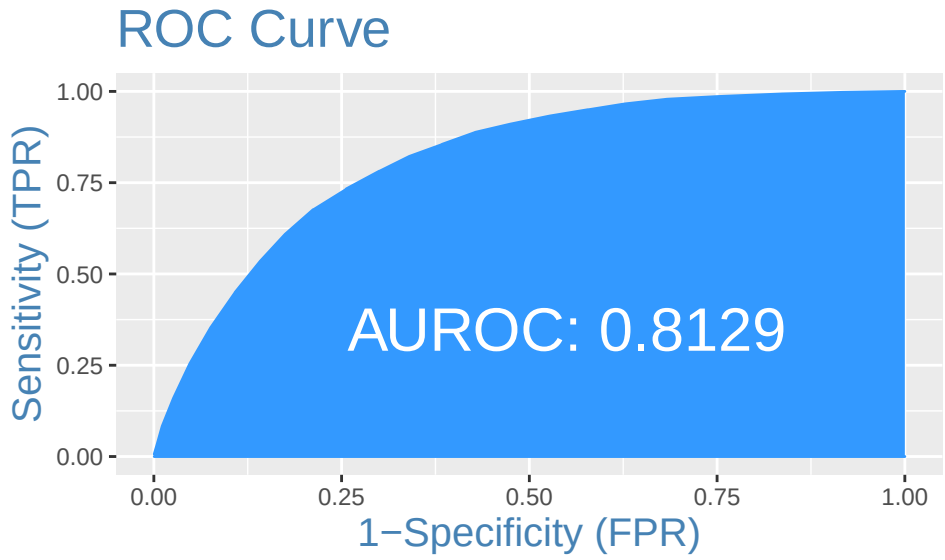
```
## [1] "The specificity of the model: 0.698799"
```

```r
sprintf("The total misclassification error rate of the model: %f", me)
```

```
## [1] "The total misclassification error rate of the model: 0.259500"
```

Code: *R*

```r
# plot the ROC Curve of the model
plotROC(test1$Diabetes_binary,valid_pred1)
```



The AU-ROC for the KNN model gives a value of 0.8129

Code: *R*

```r
# output auc with Proc package
auc(test1$Diabetes_binary,valid_pred1)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.813
```

The KNN model produced an AU ROC of 0.8129 or 81.29%, so far the first logistic regression model has performed the best. Then the random forest model slightly worse and then the KNN

Naive Bayes

Naive Bayes model will be used as the fourth and final model to test for predicting diabetes using the health indicators from the data set. The cleaned data set will be used with the original factor string answers. although a subset of the original data set will be sampled for processing reasons.

Code: *R*

```
# make a copy of the dataset
samp_data1 <- data.clean

# set the sample size we want
sample_size = 5000 #

# set seed for reporducabilty
set.seed(1)

# parameters for the sample set
idxs = sample(1:nrow(samp_data1),sample_size,replace=F)

# create the sample set
subsample1 = samp_data1[idxs,]
# list for p-values
pvalues = list()

# loop through the dataset columns and test each column
# an alpha 0.05 is used and the p-value is used to select the
# the corresponding distribution in the sample set
for (col in names(samp_data1)) {
  if (class(samp_data1[,col]) %in% c("numeric","integer")) {
    # Numeric variable. Using Kolmogorov-Smirnov test
    pvalues[[col]] = ks.test(samp_data1[[col]],samp_data1[[col]])$p.value

  } else {
    # Categorical variable. Using Pearson's Chi-square test
    probs = table(samp_data1[[col]])/nrow(samp_data1)
    pvalues[[col]] = chisq.test(table(subsample1[[col]]),p=probs)$p.value

  }
}

# convert the Diabetes diagnosis to a Yes / No
subsample1 <- subsample1 %>%
      mutate(Diabetes_binary = ifelse(Diabetes_binary == 1,"Yes","No"))

subsample1$Diabetes_binary <- as.factor(subsample1$Diabetes_binary)
```

A training and test set are created and the values checked to make sure it has all worked. The data (except for the dependent variable) will be scaled at this point.

Code: *R*

```
#Use 70% of data set as training set and remaining 30% as testing set
sample1 <- sample(c(TRUE, FALSE), nrow(subsample1), replace=TRUE, prob=c(0.8,0.2))
train_samp1 <- subsample1[sample1, ]
test_samp1 <- subsample1[!sample1, ]


# show the data types
str(train_samp1)
```

```
## 'data.frame':    4002 obs. of  22 variables:
##  $ Diabetes_binary     : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 1 1 1 1 2 ...
##  $ HighBP              : logi  TRUE TRUE TRUE TRUE FALSE FALSE ...
##  $ HighChol            : logi  FALSE TRUE FALSE FALSE FALSE FALSE ...
##  $ CholCheck           : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ BMI                 : num  26 19 36 43 34 20 27 27 22 39 ...
##  $ Smoker              : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 1 2 2 1 2 ...
##  $ Stroke              : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...
##  $ HeartDiseaseorAttack: Factor w/ 2 levels "No","Yes": 1 2 1 2 1 1 1 1 1 2 ...
##  $ PhysActivity        : logi  TRUE TRUE TRUE TRUE FALSE FALSE ...
##  $ Fruits              : logi  TRUE TRUE TRUE FALSE FALSE TRUE ...
##  $ Veggies             : logi  TRUE TRUE FALSE TRUE TRUE TRUE ...
##  $ HvyAlcoholConsump   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ AnyHealthcare       : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 1 2 2 ...
##  $ NoDocbcCost         : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...
##  $ GenHlth             : Factor w/ 5 levels "Excellent","Fair",..: 3 4 3 2 3 3 1 2 5 2 ...
##  $ MentHlth            : num  1 0 0 0 0 0 0 15 0 30 ...
##  $ PhysHlth            : num  0 15 3 5 4 0 0 20 1 30 ...
##  $ DiffWalk            : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 1 2 2 1 1 ...
##  $ Sex                 : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 1 2 1 1 2 ...
##  $ Age                 : Factor w/ 13 levels "18 to 24","25 to 29",..: 9 9 11 12 5 7 9 3 9 11 ...
##  $ Education           : Factor w/ 6 levels "College 1 - 3 years",..: 2 5 2 1 3 2 1 1 3 1 ...
##  $ Income              : Factor w/ 8 levels "< $10k","> $10k, < $15k",..: 7 4 8 6 8 3 6 4 8 5 ...
##  - attr(*, "pandas.index")=RangeIndex(start=0, stop=70692, step=1)
```

A quick summary of the test set can be checked to make sure all factors are accounted for in the sample

Code: *R*

```
summary(test_samp1)
```

```
##  Diabetes_binary   HighBP         HighChol       CholCheck            BMI     Smoker    Stroke
##  No :494         Mode :logical   Mode :logical   Mode :logical   Min.   :15   No :539   No :936
##  Yes:504         FALSE:446       FALSE:461       FALSE:24        1st Qu.:25   Yes:459   Yes: 62
##                  TRUE :552       TRUE :537       TRUE :974       Median :28
##                                                                  Mean   :30
##                                                                  3rd Qu.:33
##                                                                  Max.   :79
##
##  HeartDiseaseorAttack PhysActivity      Fruits         Veggies        HvyAlcoholConsump
##  No :840              Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  Yes:158              FALSE:293       FALSE:379       FALSE:217       FALSE:961
##                       TRUE :705       TRUE :619       TRUE :781       TRUE :37
##
##
##
##
##  AnyHealthcare NoDocbcCost      GenHlth       MentHlth         PhysHlth      DiffWalk      Sex
##  No : 54       No :888     Excellent:119   Min.   : 0.000   Min.   : 0.000   No :749   Female:581
##  Yes:944       Yes:110     Fair     :175   1st Qu.: 0.000   1st Qu.: 0.000   Yes:249   Male  :417
##                            Good     :328   Median : 0.000   Median : 0.000
##                            Poor     : 96   Mean   : 3.587   Mean   : 5.778
##                            Very Good:280   3rd Qu.: 2.000   3rd Qu.: 5.000
##                                            Max.   :30.000   Max.   :30.000
##
##        Age                         Education              Income
##  60 to 64:161   College 1 - 3 years    :293   > $75k        :317
##  65 to 69:140   College 4 years or more:364   > $50k, < $75k:143
```

```
##  55 to 59:128   Grade 12 or GED        :268   > $35k, < $50k:126
##  70 to 74:111   Grades 1 - 8           : 20   > $25k, < $35k:125
##  50 to 54:101   Grades 9 - 11          : 53   > $20k, < $25k: 88
##  75 to 79: 85   Only kindergarten      :  0   > $10k, < $15k: 73
##  (Other) :272                                 (Other)       :126
```

An X and Y variable are created, splitting the data set by the dependent and other independant variables.

Code: *R*

```
features <- setdiff(names(train_samp1), "Diabetes_binary")
x <- train_samp1[, features] # Set X as list of features
y <- train_samp1$Diabetes_binary # Set Y as Attrition
```

Training parameters for the model are set in the below chunks

Code: *R*

```
train_control <- trainControl(
method = "cv",
number = 10
)

search_grid <- expand.grid(
usekernel = c(TRUE, FALSE),
fL = 0:5,
adjust = seq(0, 5, by = 1)
)
```

The Naive Bayes model is trained.

Code: *R*

```
model.nb <- train(
x = x,
y = y,
method = "nb",
trControl = train_control,
tuneGrid = search_grid,
preProc = c("BoxCox", "center", "scale", "pca")
)
```

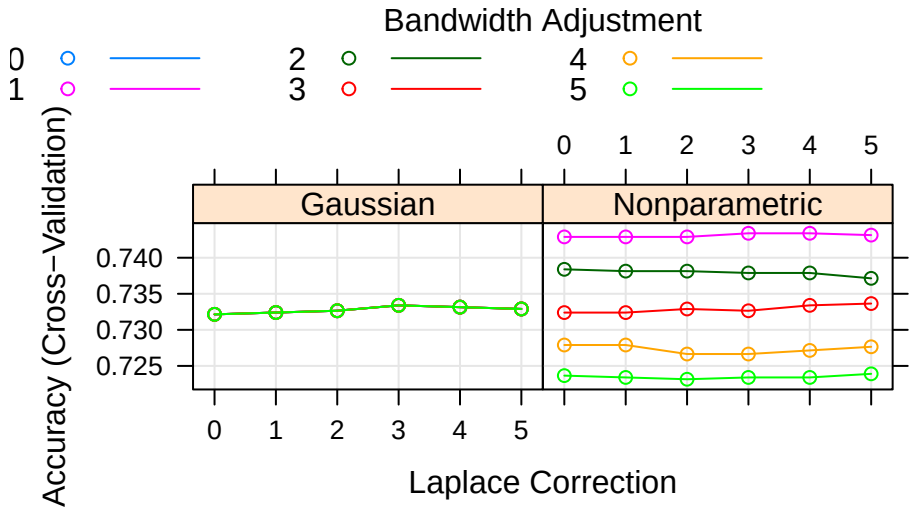
The top 5 models for accuracy are selected and plotted.

Code: *R*

```
model.nb$results %>%
top_n(5, wt = Accuracy) %>%
arrange(desc(Accuracy))
```

```
##   usekernel fL adjust  Accuracy     Kappa AccuracySD    KappaSD
## 1      TRUE  3      1 0.7433878 0.4867771 0.01384288 0.02766629
## 2      TRUE  4      1 0.7433878 0.4867770 0.01413921 0.02826153
## 3      TRUE  5      1 0.7431371 0.4862752 0.01374178 0.02746558
## 4      TRUE  0      1 0.7428884 0.4857778 0.01380426 0.02758825
## 5      TRUE  1      1 0.7428884 0.4857778 0.01380426 0.02758825
## 6      TRUE  2      1 0.7428884 0.4857778 0.01380426 0.02758825
```

```
# plot search grid results
plot(model.nb)
```



A confusion matrix of the best model is shown

Code: *R*

```
# results for best model
confusionMatrix(model.nb)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction   No  Yes
##        No  37.2 13.0
##        Yes 12.7 37.1
##
##  Accuracy (average) : 0.7434
```

Predictions are now made on the test data set created during the sampling.

Code: *R*

```
pred.nb <- predict(model.nb, newdata = test_samp1)
confusionMatrix(pred.nb, test_samp1$Diabetes_binary)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  369 152
##        Yes 125 352
##
##                Accuracy : 0.7224
```

```
##                  95% CI : (0.6935, 0.75)
##     No Information Rate : 0.505
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.4451
##
##  Mcnemar's Test P-Value : 0.1182
##
##             Sensitivity : 0.7470
##             Specificity : 0.6984
##          Pos Pred Value : 0.7083
##          Neg Pred Value : 0.7379
##              Prevalence : 0.4950
##          Detection Rate : 0.3697
##    Detection Prevalence : 0.5220
##       Balanced Accuracy : 0.7227
##
##        'Positive' Class : No
##
```

A balanced accuracy score of 72.27% is achieved by the model. The model can be tested again as per the other models giving a probability output.

Code: *R*

```
# get predictions again but output only the probabilities to list
valid_pred2 <- predict(model.nb, newdata = test_samp1, type = "prob")[,2]

# make a copy of the test set
test1 <- test_samp

# change the diabetic vairable back to a 1 or 0
test_samp1.1 <- test_samp1 %>%
      mutate(Diabetes_binary = ifelse(Diabetes_binary == "Yes",1,0))

#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(test_samp1.1$Diabetes_binary, valid_pred2)[1]
optimal
```

```
## [1] 0.4399501
```

Code: *R*

```
#calculate sensitivity
s <- InformationValue::sensitivity(test_samp1.1$Diabetes_binary,valid_pred2)

#calculate specificity
sp <- InformationValue::specificity(test_samp1.1$Diabetes_binary,valid_pred2)

#calculate total misclassification error rate
me <- InformationValue::misClassError(test_samp1.1$Diabetes_binary,valid_pred2, threshold=optimal)

sprintf("The sensitivity of the model: %f", s)
```

```
## [1] "The sensitivity of the model: 0.698413"
```

```r
sprintf("The specificity of the model: %f", sp)
```

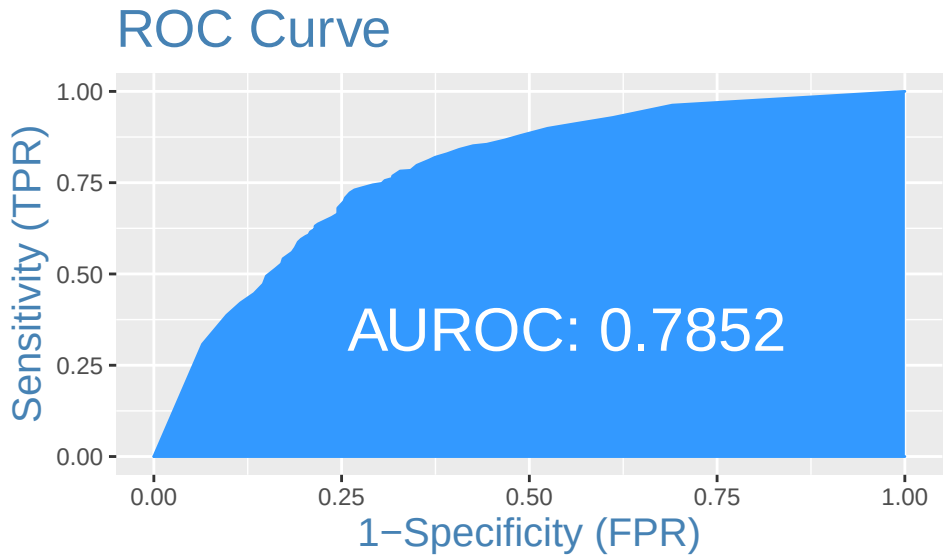
```
## [1] "The specificity of the model: 0.746964"
```

```r
sprintf("The total misclassification error rate of the model: %f", me)
```

```
## [1] "The total misclassification error rate of the model: 0.268500"
```

Code: *R*

```r
# plot the ROC Curve of the model
plotROC(test_samp1.1$Diabetes_binary,valid_pred2)
```



The Naive Bayes model performed the worst of all four models when measuring by the AU-ROC measure with a value of 0.7852 or 78.52%