Final Project

Srinath Kasturirangan, Amber Tuttman, George Smith, Alize Marsh

13 June 2021

Various functions of the tidyverse packages (dplyr, readr, etc..) are used in this assignment.

```
healthcaredata <- read.csv(
  "C:/Users/abc/Desktop/SYRACUSE/IST_687/Final_Project/healthcare-dataset-stroke-data.csv")

colnames(healthcaredata) <- c("id", "gender", "age", "hypertenstion", "heartdisease", "married",
                    "work_type", "residence_type", "avg_glucose_level", "bmi",
                    "smoking_status", "stroke")

## Explore the data
str(healthcaredata)
```

```
## 'data.frame':    5110 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
##  $ gender           : chr  "Male" "Female" "Male" "Female" ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertenstion    : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heartdisease     : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ married          : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
##  $ residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

Background on Data: According to the World Health Organization (WHO) strokes are the 2nd leading cause of death globally and account for approximately 11% of total deaths. This dataset is used to predict whether a patient is likely to have a stroke based on different variables such as gender, age, various disease and smoking status. Each row in the data provides relevant information about a single patient.

Source of Data: Kaggle (https://www.kaggle.com/fedesoriano/stroke-prediction-dataset)

Load the data. This Heathcare data set comes from Kaggle and the CSV was downloaded and saved locally.

Raw data set includes 5110 observations and 12 variables

Mix of integers, character strings and numbers

Data cleaning will be necessary to make the data set more user friendly

```
summary(healthcaredata)
```

```
##        id           gender               age        hypertenstion
##  Min.   :   67   Length:5110        Min.   : 0.08   Min.   :0.00000
##  1st Qu.:17741   Class :character   1st Qu.:25.00   1st Qu.:0.00000
##  Median :36932   Mode  :character   Median :45.00   Median :0.00000
```

```
##  Mean   :36518                     Mean   :43.23   Mean   :0.09746
##  3rd Qu.:54682                     3rd Qu.:61.00   3rd Qu.:0.00000
##  Max.   :72940                     Max.   :82.00   Max.   :1.00000
##   heartdisease       married           work_type         residence_type
##  Min.   :0.00000   Length:5110        Length:5110        Length:5110
##  1st Qu.:0.00000   Class :character   Class :character   Class :character
##  Median :0.00000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :0.05401
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  avg_glucose_level     bmi            smoking_status         stroke
##  Min.   : 55.12    Length:5110        Length:5110        Min.   :0.00000
##  1st Qu.: 77.25    Class :character   Class :character   1st Qu.:0.00000
##  Median : 91.89    Mode  :character   Mode  :character   Median :0.00000
##  Mean   :106.15                                          Mean   :0.04873
##  3rd Qu.:114.09                                          3rd Qu.:0.00000
##  Max.   :271.74                                          Max.   :1.00000
```

ID variable not useful in current state and can be eliminated

Data set includes babies/children as young as 0.08 years of age all the way through adults 82 years of age

Age is expected to have a direct influence on some of the other variables found within this dataset (i.e.; Marital Status, and Work Type, etc.) and as such could impact results if not properly factored in

Eliminated the ID column

Convert "Yes" and "No" data to 1 and 0

Convert BMI to a numeric

Transformed data to fit the needs of the models

```
healthcaredata <- healthcaredata %>%
  select(-id) %>%
  mutate(bmi = as.numeric(bmi)) %>%
  na.omit()
```

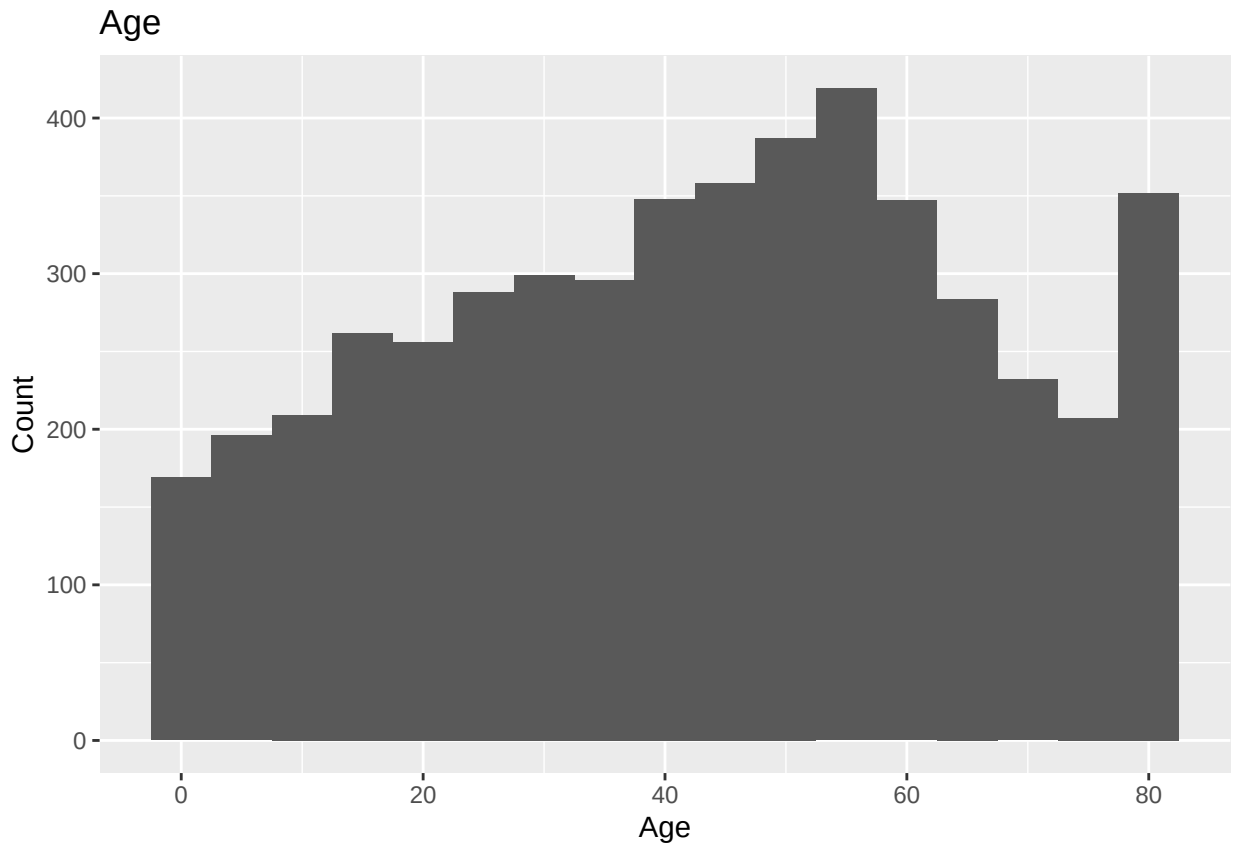
```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
head(healthcaredata)
```

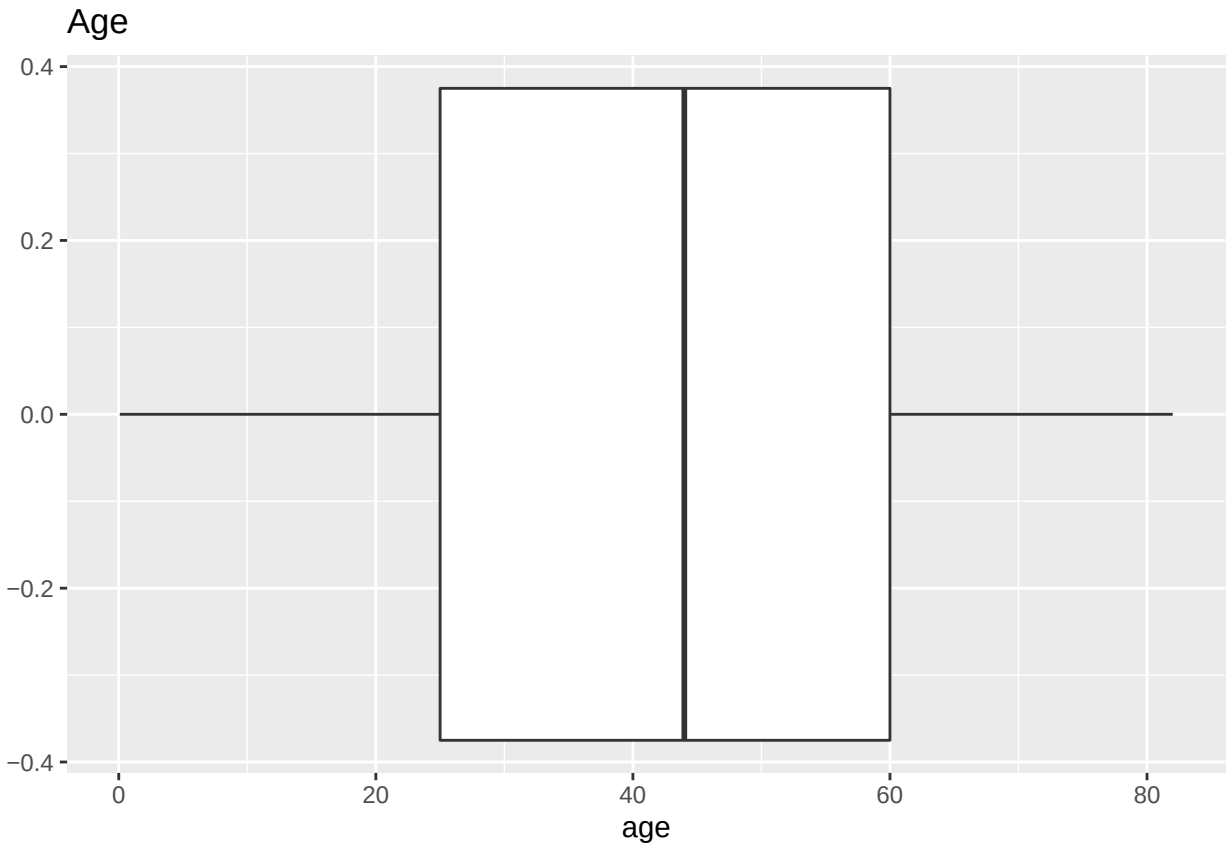
```
##   gender age hypertenstion heartdisease married     work_type residence_type
## 1   Male  67             0            1     Yes       Private          Urban
## 3   Male  80             0            1     Yes       Private          Rural
## 4 Female  49             0            0     Yes       Private          Urban
## 5 Female  79             1            0     Yes Self-employed          Rural
## 6   Male  81             0            0     Yes       Private          Urban
## 7   Male  74             1            1     Yes       Private          Rural
##   avg_glucose_level  bmi  smoking_status stroke
## 1            228.69 36.6 formerly smoked      1
## 3            105.92 32.5    never smoked      1
## 4            171.23 34.4          smokes      1
## 5            174.12 24.0    never smoked      1
## 6            186.21 29.0 formerly smoked      1
## 7             70.09 27.4    never smoked      1
```

```
par(mfrow=c(2, 2))
```

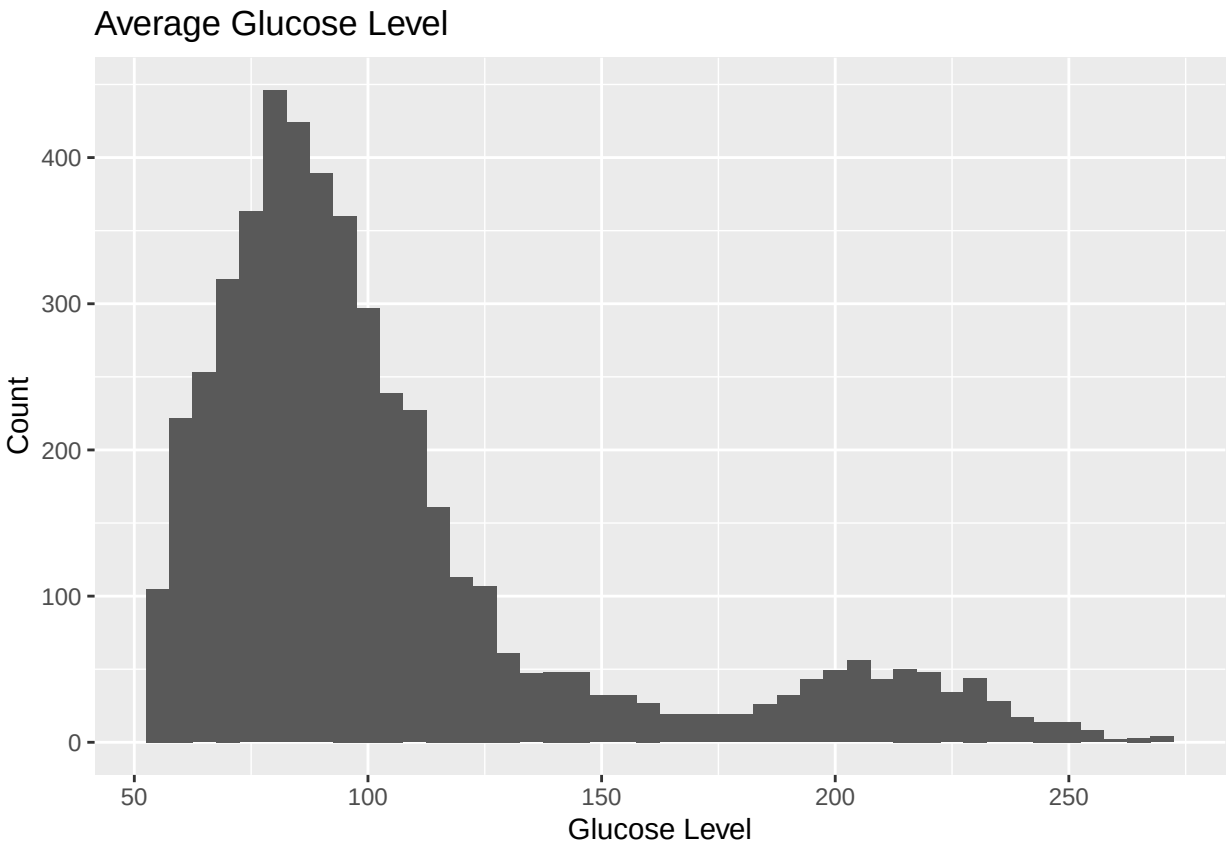
```
ggplot(healthcaredata, aes(x=age)) +
  geom_histogram(binwidth = 5) +
  ggtitle ("Age") +
  labs(x="Age", y="Count")
```



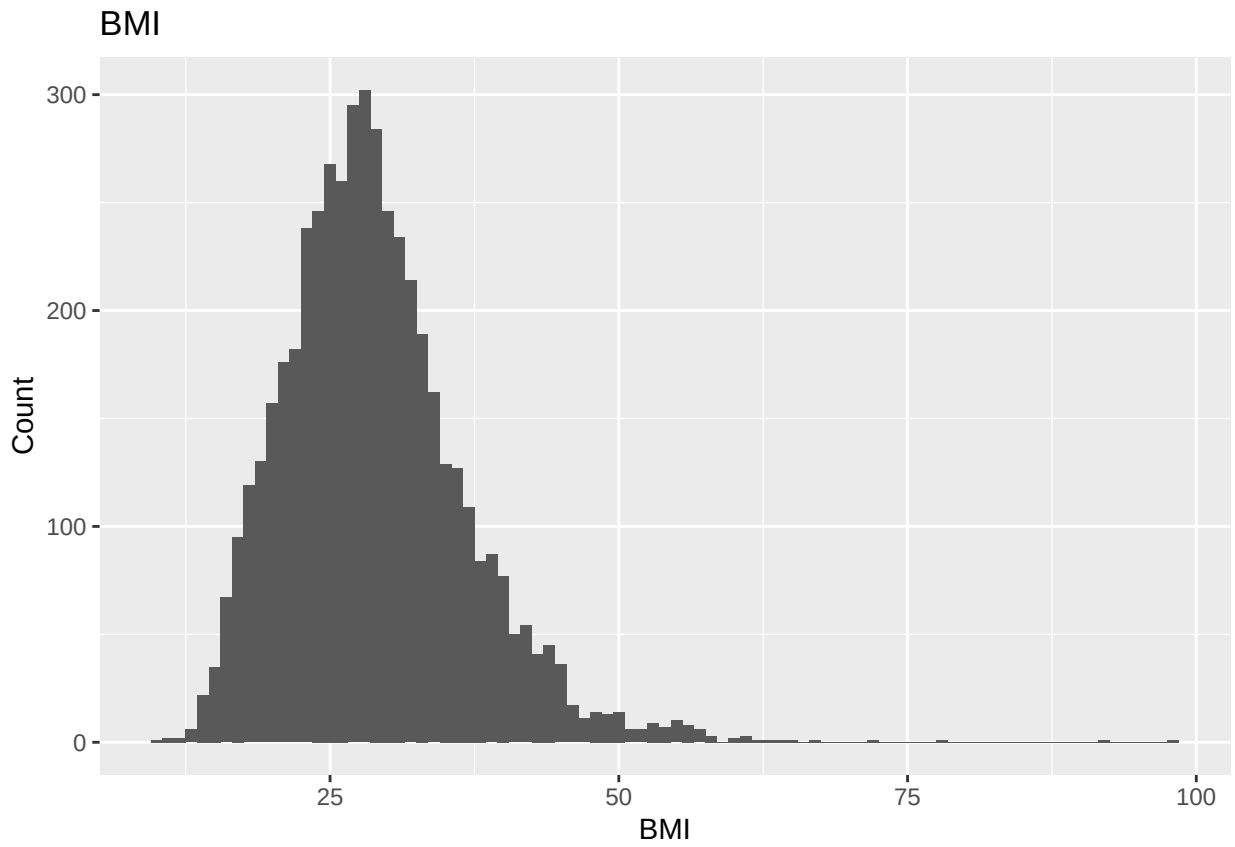

```
ggplot(healthcaredata, aes(x=age)) +
  geom_boxplot() +
  ggtitle ("Age")
```



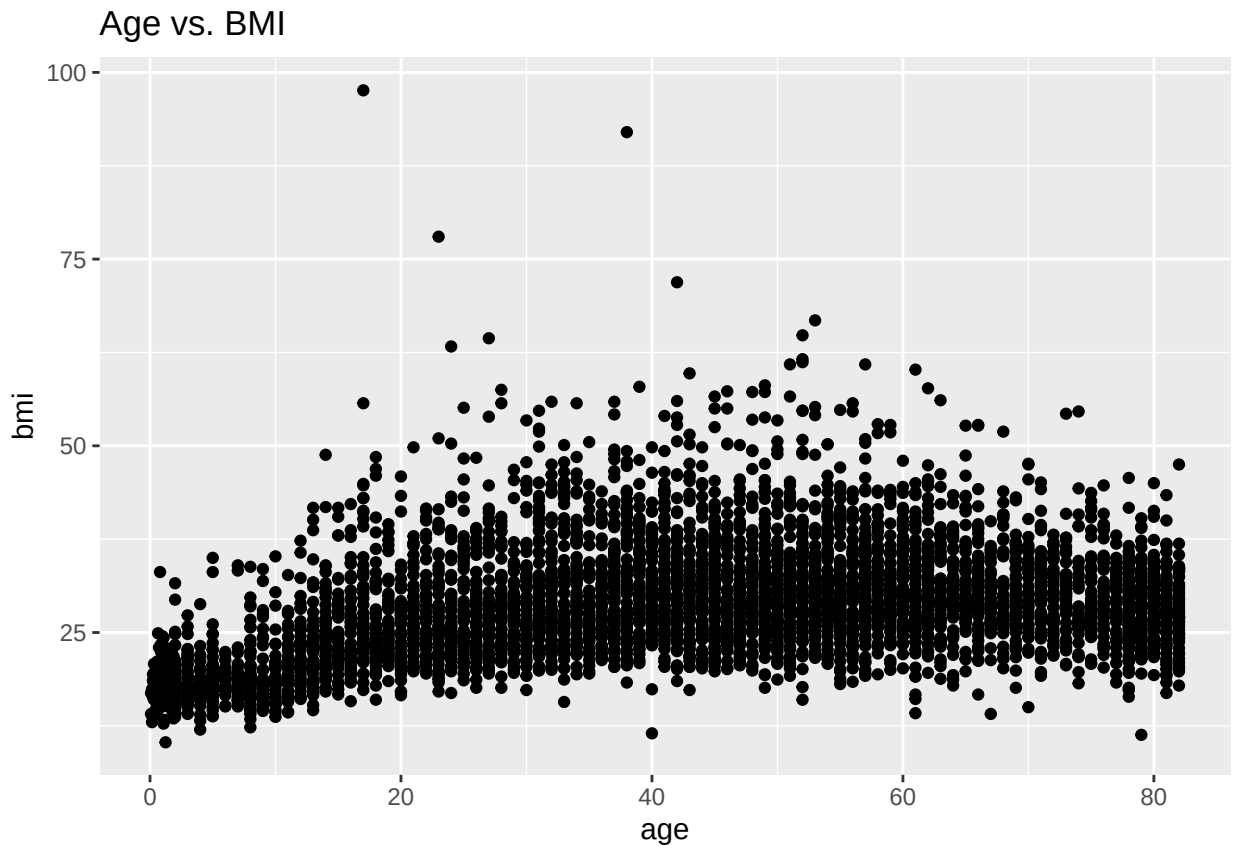

```
ggplot(healthcaredata, aes(x=avg_glucose_level)) +
  geom_histogram(binwidth = 5) +
  ggtitle ("Average Glucose Level") +
  labs(x="Glucose Level",y="Count")
```



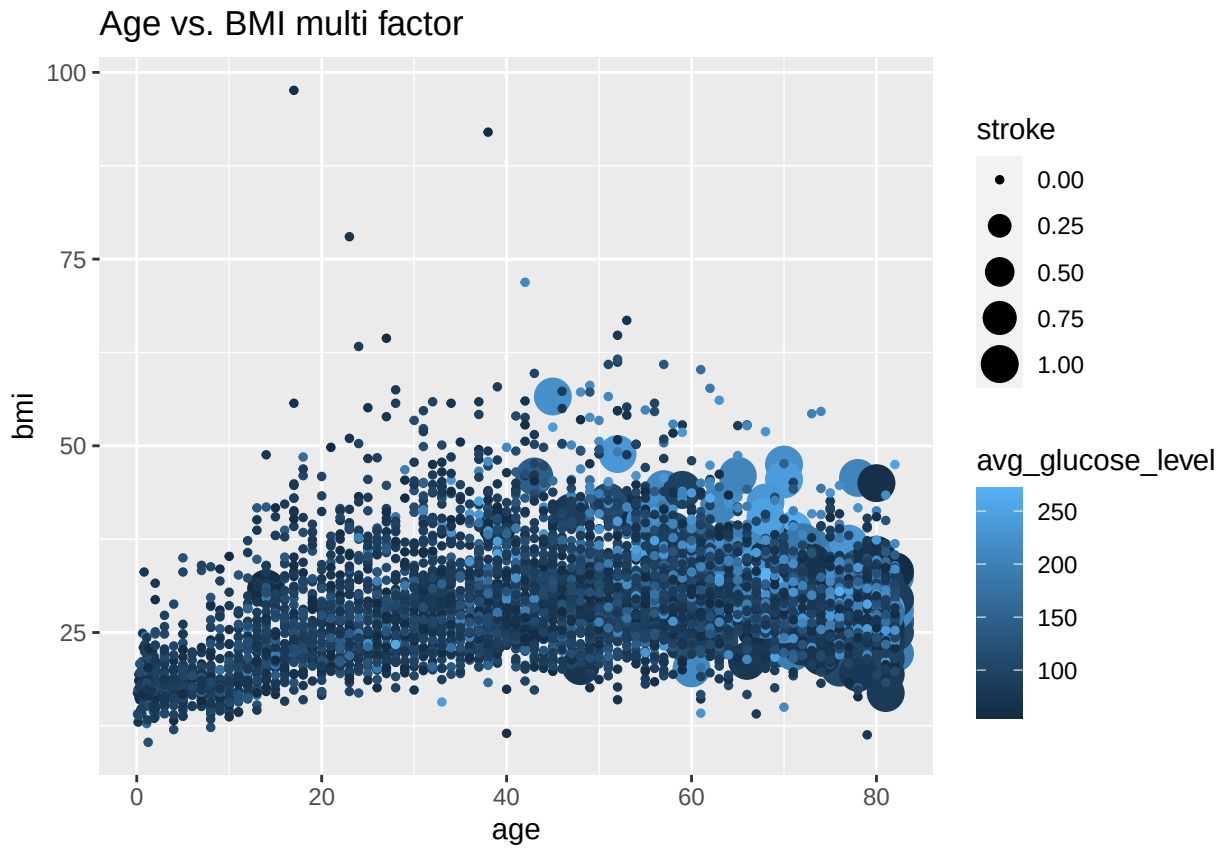

```
ggplot(healthcaredata, aes(x=bmi)) +
  geom_histogram(binwidth = 1) +
  ggtitle ("BMI") +
  labs(x="BMI",y="Count")
```



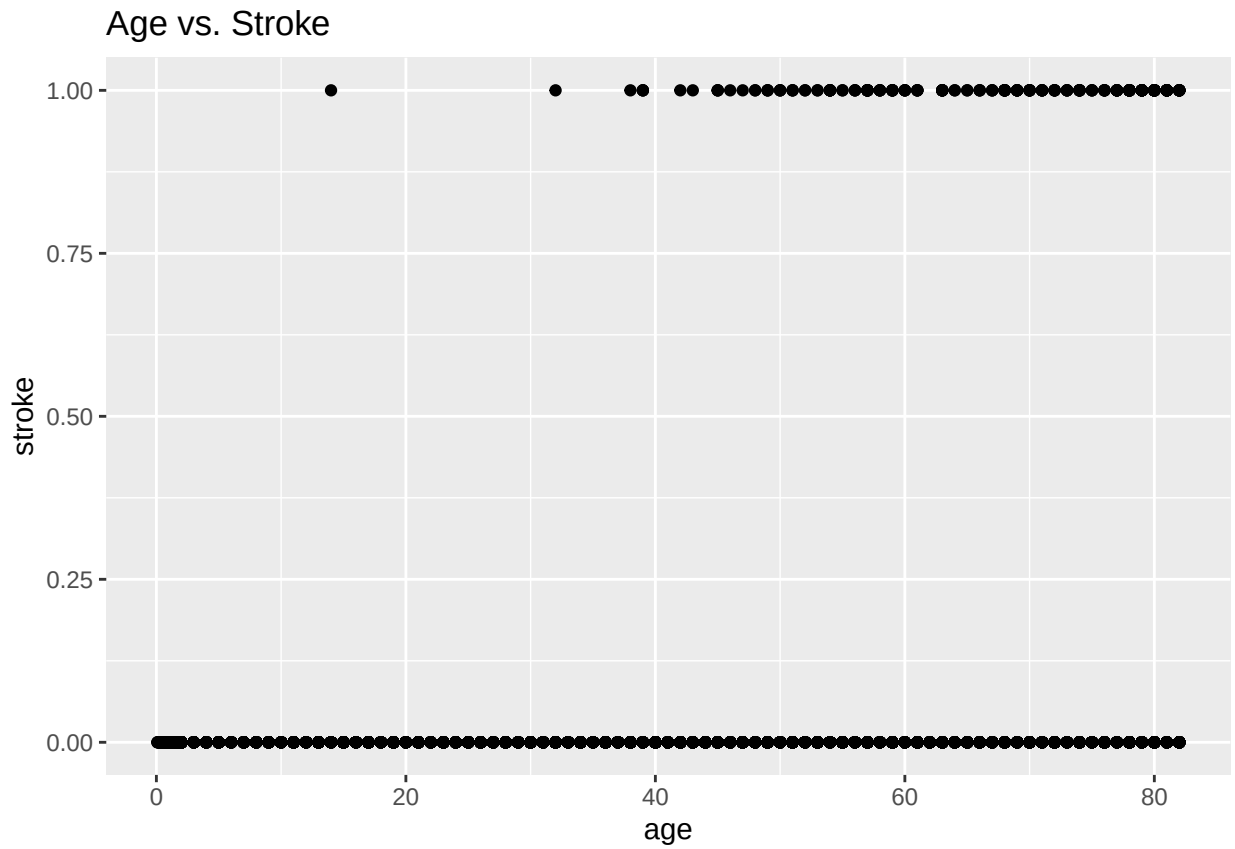

```
ggplot(healthcaredata, aes(x=age, y=bmi)) +
  geom_point() +
  ggtitle ("Age vs. BMI")
```



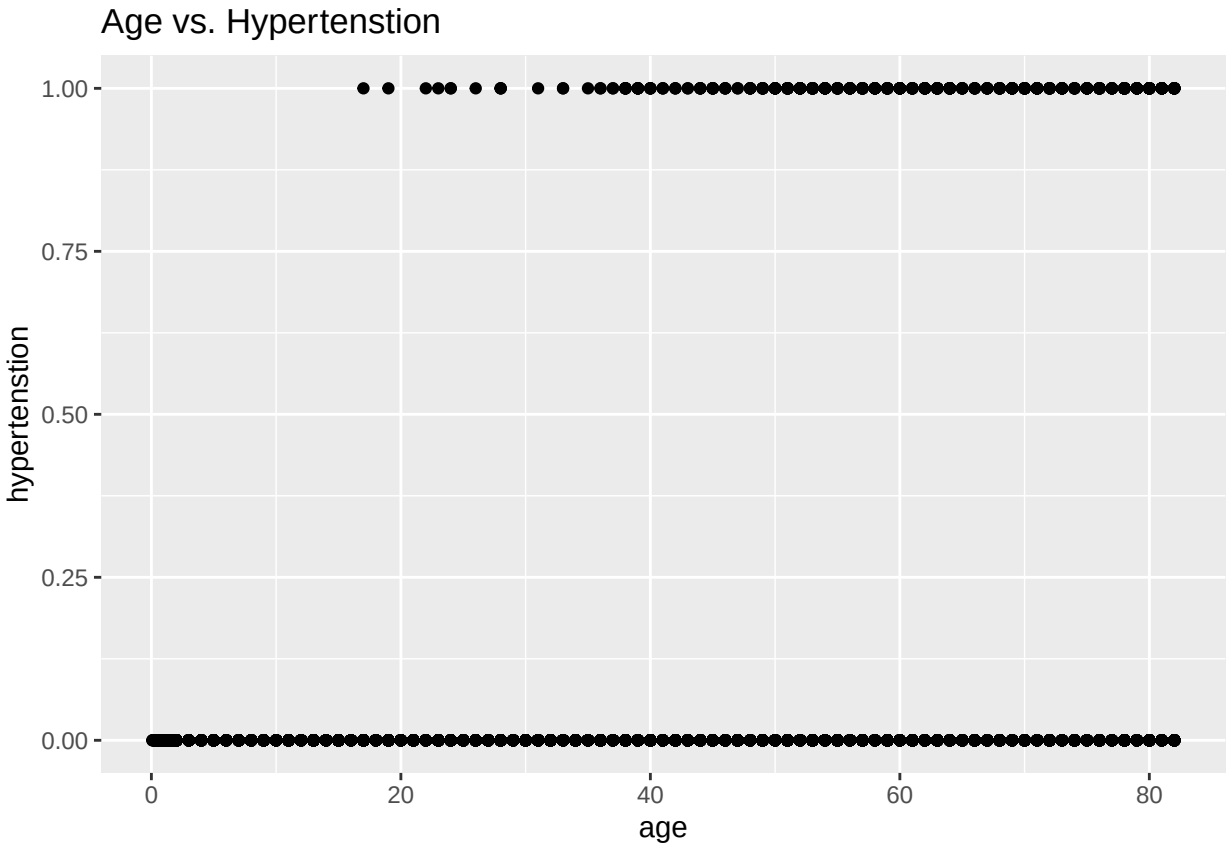

```
ggplot(healthcaredata, aes(x=age, y=bmi)) +
  geom_point(aes(size=stroke, color=avg_glucose_level)) +
  ggtitle ("Age vs. BMI multi factor")
```




```
ggplot(healthcaredata, aes(x=age, y=stroke)) +
  geom_point() +
  ggtitle ("Age vs. Stroke")
```




```
ggplot(healthcaredata, aes(x=age, y=hypertenstion)) +
  geom_point() +
  ggtitle("Age vs. Hypertenstion")
```



Data Observations from the above plots:

1) Distribution of ages in data set shown through a histogram. Median ages falls around 45 years of age. Ages appear to follow a normal distribution with slight skewness. There is also a large group of individuals in this data set who are above 80 years of age.

2) Glucose levels in our data distribution have two peaks. The larger of the two peaks falls in a lower glucose level range (~80) while the second is in a much higher glucose level range (~210).

3) BMI is evenly distributed and follows a normal distribution. Average BMI looks to fall around 28.

4) Stroke value of 0 indicates that the patient did not have a stroke. Stroke value of 1 indicates that the patient had a stroke. Through this simple chart, we can see that most patients that had strokes were above age 40. Few isolated stroke cases were found in individuals below 40 years of age.

5) Hypertension of 0 indicates that the patient does not have hypertension. Hypertension of 1 indicates that the patient has hypertension. Through this simple chart, we can see a few cases of patients having hypertension in their twenties to early thirties. We can also see that the number of patients with hypertension picks up from their mid thirties onward, as evidenced by the density of the plotted points.

6) Age vs. BMI displays a scatterplot that has been encoding with additional data attributes that allows one to visually see patterns that could be forming within the data. Age is on the X axis, while BMI is on the Y. The size of the data point tells us information on whether the patient had a stroke or not with the larger dot indicating a stroke. Additionally, the color of the data point gives us insight into the glucose level of the patient, with the lighter blue indicating higher average glucose levels. Seen in the scatterplot, there is an increase in the number of large points plotted age 40 and upward, indicating that strokes occurred more frequently in patients age 40 and up. We can also visually see that on the righthand side of the chart that there are more light blue dots than there are on the left-hand side of the chart. This indicates to us that age might also play a factor in one having higher average glucose

levels. We can also see that a lot of the lighter blue dots on the right side of the plot are also large dots, which indicate that someone who was 40+ and had higher glucose levels were also the patients that had a stroke. From this visualization, BMI does not seem to have much of an impact.

```
#layout(matrix(c(1, 2, 3, 4), 2, 2)) # optional 4 graphs/page
```

Logistic Regression models

Model 1 - Logistic regression of stroke as response variable Vs All other variables

Model Overview:

1) Coefficient with p-values less than 0.05 are considered significant
2) A small p-value indicates that is unlikely we will observe a relationship between the predictor variables and the response variables due to chance

Model One:

Model one was run including all the variables in our data set Age, hypertension, and average glucose level proved to be significant during our first model run

Age and average glucose level are highly significant with p-value's between 0 and 0.001

Hypertension is moderately significant with a p-value between between 0.001 and 0.01 Gender, heart disease, marital status, work type, residence type, BMI and smoking status variables are not significant and therefore we can not say anything about them

```
set.seed(10)

model_1 <- glm(stroke ~ ., data = healthcaredata, family = binomial)

# summary of the model
summary(model_1)
```

```
##
## Call:
## glm(formula = stroke ~ ., family = binomial, data = healthcaredata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1823  -0.2947  -0.1524  -0.0744   3.5251
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -7.360e+00  1.067e+00  -6.895 5.37e-12 ***
## genderMale                 -1.463e-02  1.544e-01  -0.095 0.924525
## genderOther                -1.135e+01  2.400e+03  -0.005 0.996225
## age                         7.348e-02  6.347e-03  11.578  < 2e-16 ***
## hypertenstion               5.249e-01  1.750e-01   2.999 0.002711 **
## heartdisease                3.488e-01  2.072e-01   1.683 0.092381 .
## marriedYes                 -1.152e-01  2.473e-01  -0.466 0.641394
## work_typeGovt_job          -6.817e-01  1.114e+00  -0.612 0.540660
## work_typeNever_worked      -1.082e+01  5.090e+02  -0.021 0.983036
## work_typePrivate           -5.208e-01  1.100e+00  -0.473 0.635943
## work_typeSelf-employed     -9.459e-01  1.119e+00  -0.845 0.397906
## residence_typeUrban         4.514e-03  1.500e-01   0.030 0.975990
## avg_glucose_level           4.652e-03  1.294e-03   3.595 0.000324 ***
```

```
## bmi                          4.062e-03  1.188e-02   0.342 0.732387
## smoking_statusnever smoked -6.722e-02  1.886e-01  -0.356 0.721556
## smoking_statussmokes         3.139e-01  2.295e-01   1.368 0.171310
## smoking_statusUnknown       -2.753e-01  2.471e-01  -1.114 0.265193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1728.4  on 4908  degrees of freedom
## Residual deviance: 1363.2  on 4892  degrees of freedom
## AIC: 1397.2
##
## Number of Fisher Scoring iterations: 15
```

```
# Make predictions
probabilities <- model_1 %>% predict(healthcaredata, type = "response")

predicted.classes <- ifelse(probabilities > 0.5, 1, 0)

# Model baseline accuracy
mean(predicted.classes == healthcaredata$stroke)
```

```
## [1] 0.9574251
```

Model 2 - Logistic regression of stroke as response variable Vs age + hypertenstion + heartdisease + married + work_type + avg_glucose_level

Model Two:

1) Model two was run using only the variables that proved to be statistically significant during our first run, all other variables that proved to not be significant were dropped from our model for the second run
2) Model two produced similar results as the first model in that age and average glucose levels were highly significant while hypertension was moderately significant
3) Based on visualizations of our data sets created earlier, it is no surprise our model indicates that age, hypertension and average glucose levels play a key factor in one's ability to predict a stroke Knowing which variables are significant helps us know what variables to focus on as we proceed in our analysis

```
set.seed(10)

model_2 <- glm(stroke ~ age + hypertenstion + heartdisease +
                 avg_glucose_level, data = healthcaredata, family = binomial)

# Summarize the model
summary(model_2)
```

```
##
## Call:
## glm(formula = stroke ~ age + hypertenstion + heartdisease + avg_glucose_level,
##     family = binomial, data = healthcaredata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0995  -0.2940  -0.1599  -0.0778   3.5885
##
```

```
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.660740   0.387152 -19.787  < 2e-16 ***
## age                0.067547   0.005571  12.124  < 2e-16 ***
## hypertenstion      0.539613   0.173055   3.118 0.001820 **
## heartdisease       0.404298   0.203447   1.987 0.046895 *
## avg_glucose_level  0.004802   0.001255   3.828 0.000129 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1728.4  on 4908  degrees of freedom
## Residual deviance: 1374.6  on 4904  degrees of freedom
## AIC: 1384.6
##
## Number of Fisher Scoring iterations: 7
```

```
# Make predictions
probabilities <- model_2 %>% predict(healthcaredata, type = "response")

predicted.classes <- ifelse(probabilities > 0.5, 1, 0)

# Model baseline accuracy
mean(predicted.classes == healthcaredata$stroke)
```

```
## [1] 0.9574251
```

```
anova(model_1, model_2)
```

```
## Analysis of Deviance Table
##
## Model 1: stroke ~ gender + age + hypertenstion + heartdisease + married +
##     work_type + residence_type + avg_glucose_level + bmi + smoking_status
## Model 2: stroke ~ age + hypertenstion + heartdisease + avg_glucose_level
##   Resid. Df Resid. Dev  Df Deviance
## 1      4892     1363.2
## 2      4904     1374.7 -12   -11.42
```

Further transformation of healthcaredata by bucketing “age” into groups, add “children” to “Never_worked” in work_type column

```
set.seed(10)

transformed_data <- healthcaredata %>%
  mutate(age = case_when(age < 10 ~ "le_than_10",
                         age < 30 ~ "le_than_30",
                         age < 50 ~ "le_than_50",
                         age < 70 ~ "le_than_70",
                         TRUE ~ "ge_than_70"),
         married = if_else(married == "Yes", 1, 0),
         work_type = case_when(work_type == "children" ~ "Never_worked",
                               TRUE ~ as.character(work_type)),
         bmi = as.numeric(bmi)
  ) %>%
  na.omit() %>%
```

```r
  mutate_at(c("gender", "hypertenstion", "heartdisease",
              "smoking_status", "age"), as.factor)

model_3 <- glm(stroke ~ age + hypertenstion + heartdisease + avg_glucose_level,
               data = transformed_data, family = binomial)

# Summarize the model
summary(model_3)
```

```
##
## Call:
## glm(formula = stroke ~ age + hypertenstion + heartdisease + avg_glucose_level,
##     family = binomial, data = transformed_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0093  -0.3068  -0.1547  -0.0436   3.7753
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -2.561769   0.204164 -12.548  < 2e-16 ***
## agele_than_10     -16.461146 304.110044  -0.054 0.956832
## agele_than_30      -4.843843   1.009143  -4.800 1.59e-06 ***
## agele_than_50      -2.326287   0.274292  -8.481  < 2e-16 ***
## agele_than_70      -0.964497   0.158960  -6.068 1.30e-09 ***
## hypertenstion1      0.542247   0.172130   3.150 0.001631 **
## heartdisease1       0.494925   0.201622   2.455 0.014100 *
## avg_glucose_level   0.004835   0.001254   3.855 0.000116 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1728.4  on 4908  degrees of freedom
## Residual deviance: 1387.5  on 4901  degrees of freedom
## AIC: 1403.5
##
## Number of Fisher Scoring iterations: 17
```

```r
# Make predictions
probabilities <- model_3 %>% predict(transformed_data, type = "response")

predicted.classes <- ifelse(probabilities > 0.5, 1, 0)

# Model accuracy
mean(predicted.classes == transformed_data$stroke)
```

```
## [1] 0.9574251
```

```r
anova(model_3)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
```

```
## Response: stroke
##
## Terms added sequentially (first to last)
##
##
##                   Df Deviance Resid. Df Resid. Dev
## NULL                                4908     1728.4
## age                4  304.924      4904     1423.5
## hypertenstion      1   14.137      4903     1409.3
## heartdisease       1    7.562      4902     1401.8
## avg_glucose_level  1   14.283      4901     1387.5
```

Test/Train model accuracy

```
set.seed(10)

newhealthcaredata <- transformed_data %>%
  mutate(row_num = row_number())

newhealthcaredata %>%
  group_by(stroke) %>%
  summarize(cnt = n())
```

```
## # A tibble: 2 x 2
##   stroke   cnt
##    <int> <int>
## 1      0  4700
## 2      1   209
```

```
training_data <- newhealthcaredata %>%
  group_by(stroke) %>%
  do(sample_frac(., .70))

testing_data <- newhealthcaredata %>%
  filter(!row_num %in% training_data$row_num)


model <- glm(stroke ~ age + hypertenstion + heartdisease + avg_glucose_level,
             data = training_data, family = binomial)
summary(model)$coef
```

```
##                       Estimate   Std. Error    z value     Pr(>|z|)
## (Intercept)        -2.418465449 2.431308e-01 -9.9471789 2.594335e-23
## agele_than_10     -16.529945814 3.713852e+02 -0.0445089 9.644988e-01
## agele_than_30      -4.532195705 1.012964e+00 -4.4741917 7.670089e-06
## agele_than_50      -2.453156248 3.378727e-01 -7.2605933 3.853961e-13
## agele_than_70      -1.071530568 1.927001e-01 -5.5606114 2.688312e-08
## hypertenstion1      0.414958426 2.106832e-01  1.9695848 4.888598e-02
## heartdisease1       0.538570139 2.342233e-01  2.2993878 2.148293e-02
## avg_glucose_level   0.004069516 1.522077e-03  2.6736600 7.502849e-03
```

```
probabilities <- model %>% predict(testing_data, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, 1, 0)
# predicted.classes

# Model accuracy
```

```
mean(predicted.classes == testing_data$stroke)
```

```
## [1] 0.9572301
```

Neural Network Model

Created training and testing data frames:

Randomized the data to create training and testing data frames that would include instances of patients who have / have not had strokes Included 70% of our data in the training dataframe

Ran the machine learning models using the training data:

Included 4 hidden layers

Ran the machine learning models multiple times to compare results

```
set.seed(10)

transformed_data_nn <- healthcaredata %>%
  mutate(age = case_when(age < 10 ~ "le_than_10",
                         age < 30 ~ "le_than_30",
                         age < 50 ~ "le_than_50",
                         age < 70 ~ "le_than_70",
                         TRUE ~ "ge_than_70"),
         married = if_else(married == "Yes", 1, 0),
         work_type = case_when(work_type == "children" ~ "Never_worked",
                               TRUE ~ as.character(work_type)),
         bmi = as.numeric(bmi)
  ) %>%
  na.omit() %>%
  mutate_at(c("gender", "hypertenstion", "heartdisease",
              "smoking_status", "age"), as.factor) %>%
  mutate_at(c("gender", "hypertenstion", "heartdisease",
              "smoking_status", "age"), as.numeric) %>%
  select(-work_type, -residence_type) %>%
  mutate(row_num = row_number())


str(transformed_data_nn)
```

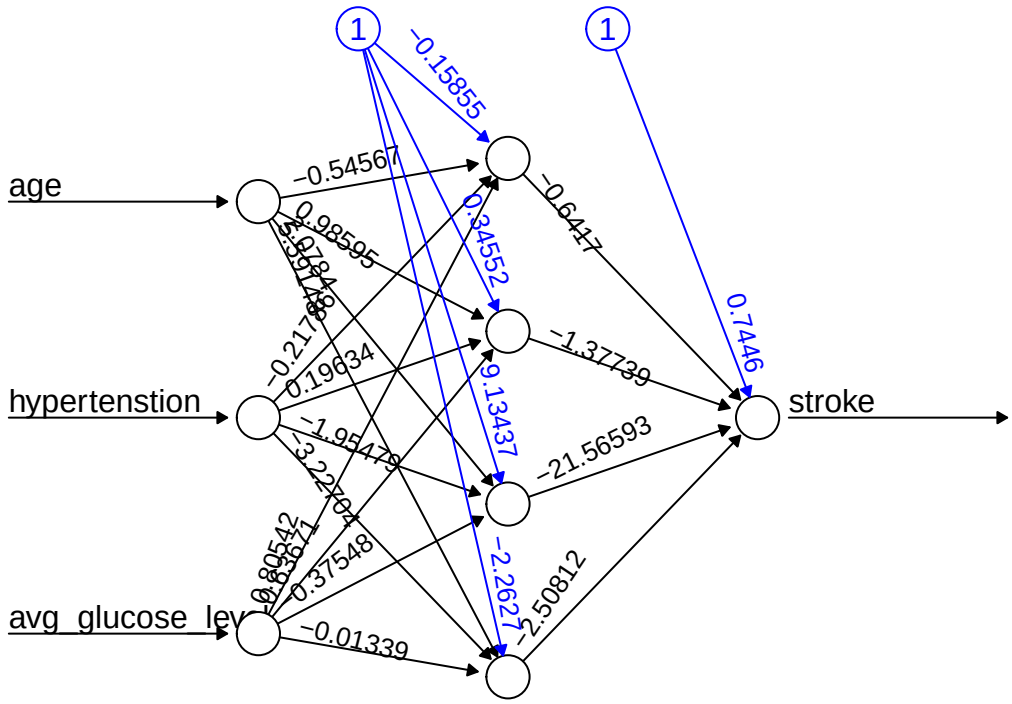
```
## 'data.frame':    4909 obs. of  10 variables:
##  $ gender           : num  2 2 1 1 2 2 1 1 1 1 ...
##  $ age              : num  5 1 4 1 1 1 5 1 1 5 ...
##  $ hypertenstion    : num  1 1 1 2 1 2 1 1 2 1 ...
##  $ heartdisease     : num  2 2 1 1 1 2 1 1 1 2 ...
##  $ married          : num  1 1 1 1 1 1 0 1 1 1 ...
##  $ avg_glucose_level: num  229 106 171 174 186 ...
##  $ bmi              : num  36.6 32.5 34.4 24 29 27.4 22.8 24.2 29.7 36.8 ...
##  $ smoking_status   : num  1 2 3 2 1 2 2 4 2 3 ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ row_num          : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
training_data_nn <- transformed_data_nn %>%
  group_by(stroke) %>%
  do(sample_frac(., .70))
```

```
testing_data_nn <- transformed_data_nn %>%
  filter(!row_num %in% training_data_nn$row_num)

#NN
healthcarenet <- neuralnet(stroke ~ age + hypertenstion + avg_glucose_level,
                           training_data_nn, hidden=4, lifesign="minimal",
                           linear.output = FALSE, threshold=.01)
```

```
## hidden: 4    thresh: 0.01    rep: 1/1    steps:     280  error: 65.35499 time: 0.52 secs
```

```
# NN plot
plot(healthcarenet, rep="best")
```




```
healthcarenet.results <- compute(healthcarenet, testing_data_nn)

results <- data.frame(actual_stroke=testing_data_nn$stroke, prediction=healthcarenet.results$net.result)

results$prediction <- round(results$prediction, 2)

mean(results$prediction)
```

```
## [1] 0.03857434
```

```
## still no predictions for actual strokes

head(results)
```

```
##   actual_stroke prediction
## 1             1       0.18
## 2             1       0.02
## 3             1       0.11
## 4             1       0.21
## 5             1       0.02
## 6             1       0.02
```

```
results <- results %>%
  mutate(predicted_stroke = if_else(prediction > 0.2, 1, 0))

results %>%
  group_by(actual_stroke, predicted_stroke) %>%
  summarize(cnt = n())
```

```
## `summarise()` has grouped output by 'actual_stroke'. You can override using the `.groups` argument.
```

```
## # A tibble: 4 x 3
## # Groups:   actual_stroke [2]
##   actual_stroke predicted_stroke   cnt
##           <int>            <dbl> <int>
## 1             0                0  1375
## 2             0                1    35
## 3             1                0    53
## 4             1                1    10
```

```
# Model accuracy
mean(results$actual_stroke == results$predicted_stroke)
```

```
## [1] 0.940258
```

Random Forest Model

Created training and testing data frames:

1) Randomized the data to create training and testing data frames that would include instances of patients who have / have not had strokes Included 70% of our data in the training dataframe

Ran the machine learning models using the training data:

1) Included 500 Trees
2) Enable Proximity/Importance of variables to plot further
3) Print the importance of variables
4) Variable Importance Plots displaying the MeanDecreaseAccuracy and MeanDecreaseGini factors
5) Create Random Forest Model # 2 based on Steps 1 - 4
6) Predict stroke on Testing data
7) Print the Model accuracy and prediction matrix

```
set.seed(10)

# randomforest data transformation
rf_data <- healthcaredata %>%
  mutate(stroke = as.character(stroke)) %>%
  mutate(stroke = as.factor(stroke))
```

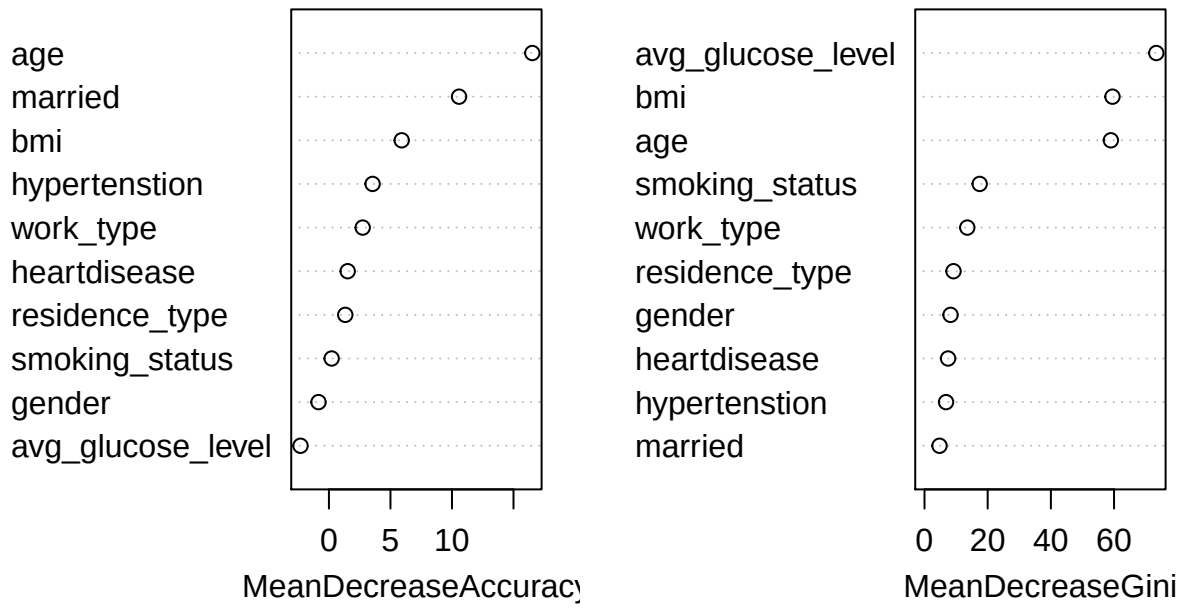
```r
# split index for training/testing model
splitIndex <- createDataPartition(rf_data[, "stroke"],
                                  p=.70, list=FALSE, times=1)

# create training/testing dataframes
trainDF <- rf_data[splitIndex, ]
testDF  <- rf_data[-splitIndex, ]

# create randomForest model # 1 (baseline) on training data
rf1 <- randomForest(
  formula = stroke ~ .,
  data = trainDF,
  ntree = 500,
  importance = TRUE,
  proximity = TRUE)

# print model details
print(rf1)
```

```
##
## Call:
##  randomForest(formula = stroke ~ ., data = trainDF, ntree = 500,      importance = TRUE, proximity =
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 4.34%
## Confusion matrix:
##      0 1  class.error
## 0 3287 3 0.0009118541
## 1  146 1 0.9931972789
```

```r
# print importance of variables
round(importance(rf1), 2)
```

```
##                       0     1 MeanDecreaseAccuracy MeanDecreaseGini
## gender             0.63 -4.70                -0.85             8.24
## age               11.47 23.98                16.53            58.99
## hypertenstion      4.23 -2.55                 3.55             6.82
## heartdisease       0.25  4.03                 1.52             7.46
## married           11.96 -6.82                10.58             4.75
## work_type          3.75 -3.09                 2.74            13.55
## residence_type     2.23 -2.40                 1.33             9.18
## avg_glucose_level -3.96  4.11                -2.31            73.38
## bmi                7.10 -4.00                 5.92            59.50
## smoking_status     1.36 -3.70                 0.22            17.46
```

```r
# Variable Importance Plot
varImpPlot(rf1)
```




```
# create randomForest model # 2 (based on importance) on training data
rf2 <- randomForest(
  formula = stroke ~ age + married + bmi + avg_glucose_level,
  data = trainDF,
  ntree = 500,
  importance = TRUE,
  proximity = TRUE)

# predict stroke on testing data
pred <- predict(rf2, testDF %>% select(-stroke))

# Model accuracy
mean(testDF$stroke == pred)
```

```
## [1] 0.9544837
```

```
# prediction matrix
table(testDF$stroke, pred)
```

```
##    pred
##        0    1
##   0 1405    5
##   1   62    0
```

Naive Bayes Model

Created training and testing data frames:

1) Randomized the data to create training and testing data frames that would include instances of patients who have / have not had strokes
2) Convert stroke to a character variable (required to convert to a factor later)
3) select only relevant variables to run the Naive Bayes model
4) Included 70% of our data in the training dataframe (30% for testing the model)

Ran the Naive Bayes model with all default options using the training data:

1) Predict stroke on Testing data
2) Print the Model accuracy and prediction matrix

```
set.seed(10)

# Naive Bayes data transformation
nb_data <- healthcaredata %>%
  mutate(stroke = as.character(stroke)) %>%
  mutate(stroke = as.factor(stroke)) %>%
  select(stroke, age, married, hypertenstion, heartdisease,
         avg_glucose_level, bmi)

# split index for training/testing model
splitIndex <- createDataPartition(nb_data[, "stroke"],
                                  p=.70, list=FALSE, times=1)

# create training/testing dataframes
trainDF <- nb_data[splitIndex, ]
testDF  <- nb_data[-splitIndex, ]

# create Naive Bayes model
nb1 <- naiveBayes(stroke ~ age + married + hypertenstion +
                    heartdisease + avg_glucose_level + bmi,
                  data = healthcaredata)

# predict stroke on testing data
pred <- predict(nb1, testDF %>% select(-stroke))

# Model accuracy
mean(testDF$stroke == pred)
```

```
## [1] 0.8872283
```

```
# prediction matrix
table(testDF$stroke, pred)
```

```
##    pred
##        0    1
##   0 1283  127
##   1   39   23
```