Individual Assignment-Diabetes Data

Dataset used: Diabetes dataset from Kaggle

As, the dataset was huge, selected only 500 data values to make predictions.

Data Dictionary:

1. Diabetes_binary: Indicates whether patient is Diabetic or not after considering all factors(0- Not Diabetic; 1-Diabetic)
2. HighBP- 1: High Blood Pressure; 0- Low Blood pressure
3. BMI- Body Mass Index varying from 15-69
4. Smoker- 1: Person smokes; 0-Person doesn't smokes
5. Stroke- 1: Person had a stroke; 0-Person didn't have a stroke till now
6. HeartDisease- 1: Person had a heart disease; 0-Person didn't have a heart disease
7. PhysActivity- 1: Person does some physical workout; 0- Person doesn't perform any physical activity.
8. HvyAlcohol- 1: Person drinks alcohol on regular basis; 0- Person doesn't drink alcohol
9. AnyHealthCare- 1: Person has some kind of health care; 0- Person doesn't have any healthcare
10. GenHlth- Person has General Health ranging from 1-5. Assuming 1 being the highest and 5 being the lowest.
11. MentHlth- Person's Mental Health ranging from 0-30.
12. PhysHlth- Person's Physical Health ranging from 0-30.
13. Sex- 1- Male; 0- Female
14. Age- Age ranging from 1-13
15. Education- Education levels varying from 1-6
16. Income- Income levels varying from 1-8

Q.1) Predicted whether a person has diabetes or not using different measures

Q.2) Important/Major factors that contributed to a person having diabetes

iment.html

ser | Find

```
cumvar_Diabetes <- cumsum(propvar)
cumvar_Diabetes
```

```
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 0.2076092 0.3124541 0.3952830 0.4723930 0.5416012 0.6095218 0.6737612 0.7300345
##       PC9      PC10      PC11      PC12      PC13      PC14      PC15
## 0.7841636 0.8307054 0.8741407 0.9121975 0.9467952 0.9761875 1.0000000
```

```
matlambdas <- rbind(eigen_Diabetes,propvar,cumvar_Diabetes)
rownames(matlambdas) <- c("Eigenvalues","Prop. variance","Cum. prop. variance")
round(matlambdas,4)
```

```
##                       PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8
## Eigenvalues         3.1141 1.5727 1.2424 1.1567 1.0381 1.0188 0.9636 0.8441
## Prop. variance      0.2076 0.1048 0.0828 0.0771 0.0692 0.0679 0.0642 0.0563
## Cum. prop. variance 0.2076 0.3125 0.3953 0.4724 0.5416 0.6095 0.6738 0.7300
##                       PC9   PC10   PC11   PC12   PC13   PC14   PC15
## Eigenvalues         0.8119 0.6981 0.6515 0.5709 0.5190 0.4409 0.3572
## Prop. variance      0.0541 0.0465 0.0434 0.0381 0.0346 0.0294 0.0238
## Cum. prop. variance 0.7842 0.8307 0.8741 0.9122 0.9468 0.9762 1.0000
```

```
summary(Diabetes_pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.7647 1.2541 1.11464 1.07548 1.01888 1.00936 0.98163
## Proportion of Variance 0.2076 0.1048 0.08283 0.07711 0.06921 0.06792 0.06424
## Cumulative Proportion  0.2076 0.3125 0.39528 0.47239 0.54160 0.60952 0.67376
##                           PC8     PC9    PC10    PC11    PC12   PC13    PC14
## Standard deviation     0.91875 0.90107 0.83554 0.80717 0.75555 0.7204 0.66399
## Proportion of Variance 0.05627 0.05413 0.04654 0.04344 0.03806 0.0346 0.02939
## Cumulative Proportion  0.73003 0.78416 0.83071 0.87414 0.91220 0.9468 0.97619
##                           PC15
## Standard deviation     0.59765
## Proportion of Variance 0.02381
## Cumulative Proportion  1.00000
```

```
Diabetes_pca$rotation
```

gnment.html

owser | Find

```
# Standard deviations of scores for all the PC's classified by Survival status
tabsdsPC <- aggregate(Diabetestyp_pca[,2:16],by=list(Diabetes_binary=Diabetes$Diabetes_binary),sd)
tabfsds <- t(tabsdsPC[,-1])
colnames(tabfsds) <- t(as.vector(tabsdsPC[1]$Diabetes_binary))
tabfsds
```

```
##              0         1
## PC1  1.5651494 1.5068047
## PC2  1.2485808 1.2457169
## PC3  1.0585086 1.1186557
## PC4  1.0379513 1.0806399
## PC5  0.9326077 1.0935561
## PC6  1.0769807 0.9345409
## PC7  1.0861529 0.8654565
## PC8  0.8374170 0.9940173
## PC9  0.8472220 0.9527779
## PC10 0.7868442 0.8794344
## PC11 0.7517034 0.8597312
## PC12 0.7093162 0.8001453
## PC13 0.7601934 0.6799042
## PC14 0.6894095 0.6390050
## PC15 0.5822395 0.6003592
```

```
t.test(PC1~Diabetes$Diabetes_binary,data=Diabetestyp_pca)
```

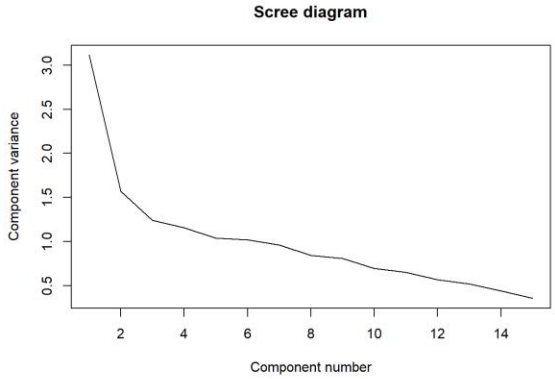
```
##
##  Welch Two Sample t-test
##
## data:  PC1 by Diabetes$Diabetes_binary
## t = -12.655, df = 496.13, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -2.010899 -1.470395
## sample estimates:
## mean in group 0 mean in group 1
##      -0.8720678       0.8685795
```

```
t.test(PC2~Diabetes$Diabetes_binary,data=Diabetestyp_pca)
```

C:/Users/vidhi/Downloads/Final_Assignment.html

Final_Assignment.html | Open in Browser | Find

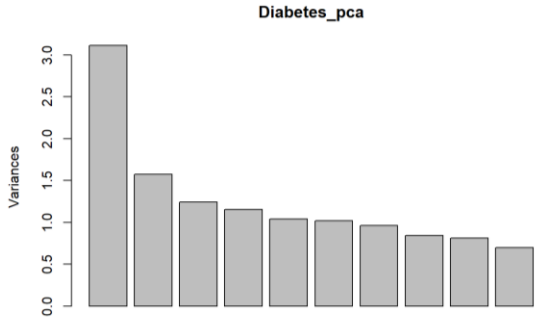

```
plot(eigen_Diabetes, xlab = "Component number", ylab = "Component variance", type = "l", main = "Scree diagram")
```




```
plot(log(eigen_Diabetes), xlab = "Component number",ylab = "log(Component variance)", type="l",main = "Log(eigenv
alue) diagram")
```



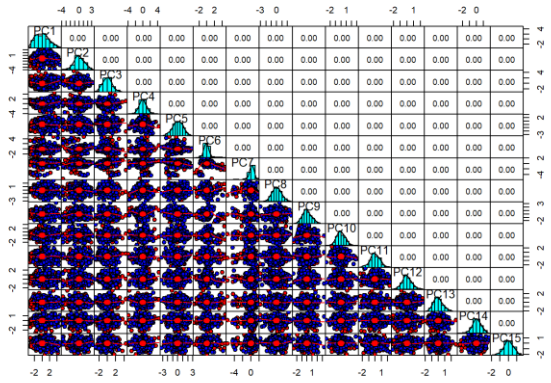
```
plot(Diabetes_pca)
```



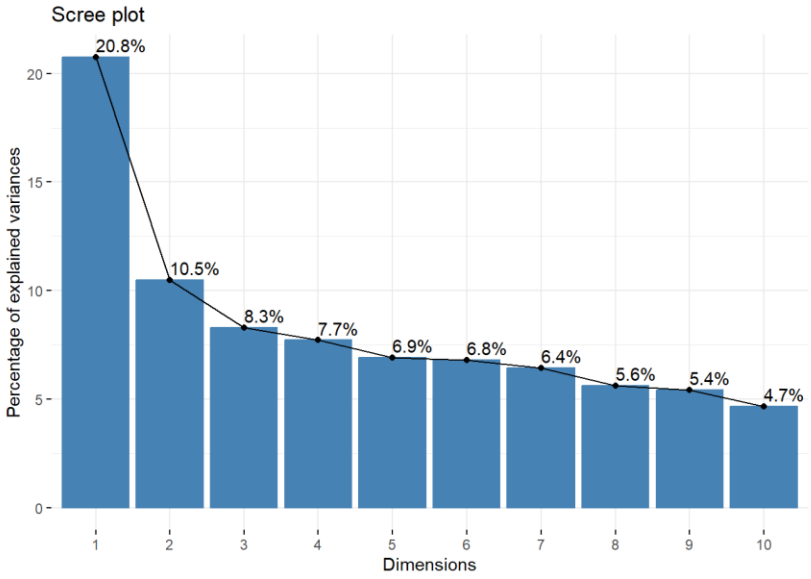
C:/Users/vidhi/Downloads/Final_Assignment.html

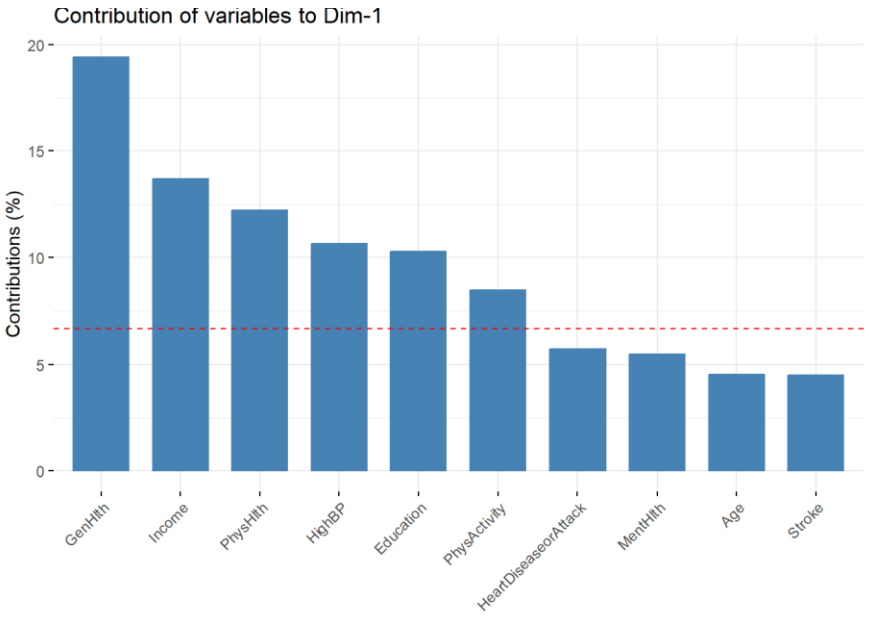
Final_Assignment.html | Open in Browser | Find

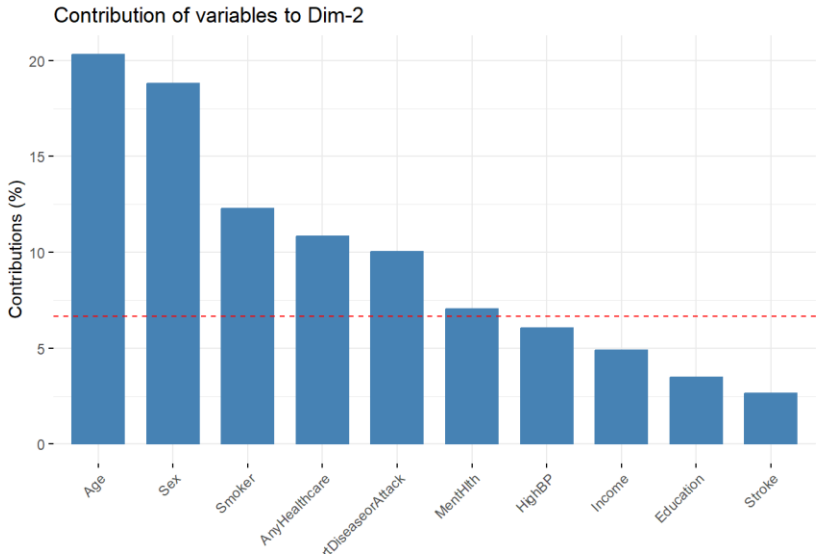
```
pairs.panels(Diabetes_pca$x,
             gap=0,
             bg = c("red", "blue")[Diabetes$Diabetes_binary],
             pch=21)
```



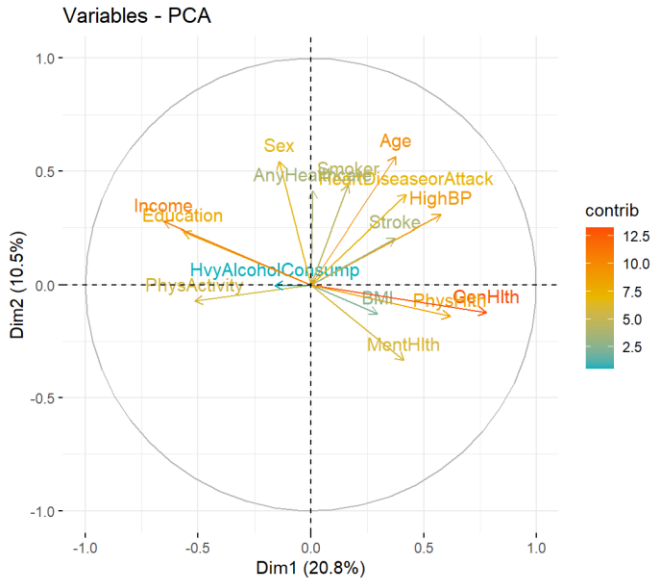

```
fviz_eig(Diabetes_pca, addlabels = TRUE)
```




Contribution of variables to Dim-1
Contributions (%)
20
15
10
5
0
GenHlth
Income
PhysHlth
HighBP
Education
PhysActivity
HeartDiseaseorAttack
MentHlth
Age
Stroke


Contribution of variables to Dim-2
Contributions (%)
20
15
10
5
0
Age
Sex
Smoker
AnyHealthcare
MentHlth
HighBP
Income
Education
Stroke

```
fviz_pca_var(res.pca, col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07")
             )
```



Inferences regarding PCA

* In this particular case, the first PC explains 20.8 % of the variance, the second PC explains 10.5 %, and so on. Together, the first seven PCs account for 67 % of the total variability in the data. The res explain less than 30% of the total variability in the data and those PC's can be removed.

* Factors contributing to 1st dimensionality: General Health,Income, Physical Health, High BP, Education and Physical Activity.

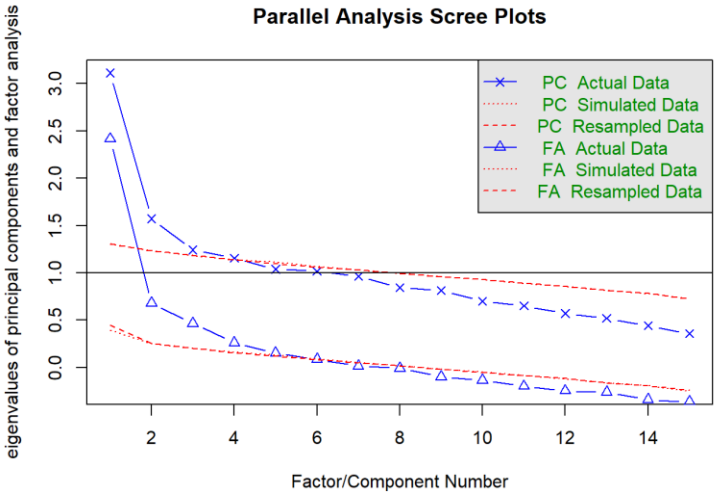
* Factors contributing to 2nd dimensionality: Age and Sex.

* From **scree plot** : it is visible that 1st and 2nd dimensions are only important i.e the factors: General Health,Income, Physical Health, High BP, Education, Physical Activity, Age and Sex are the only important factors in determing whether a person is diabetic or not.

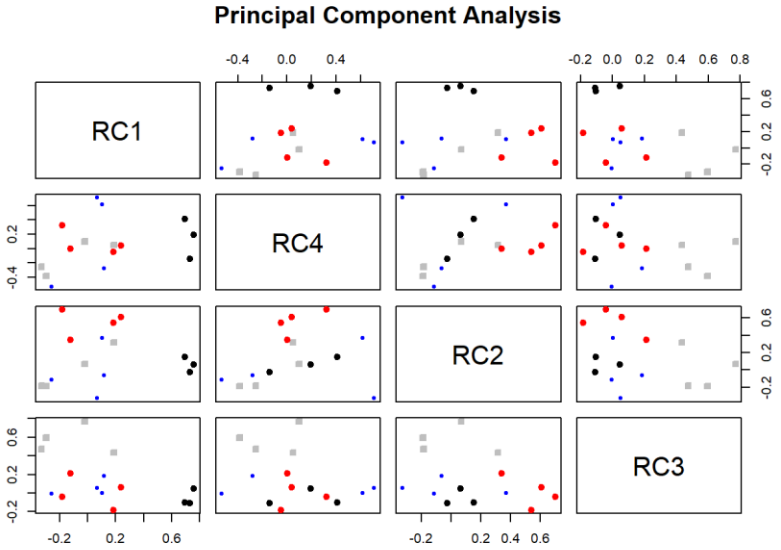
* Also, from the last plot it is visible that the highest contributors in a person being Diabetic(i.e-1) is General Health, Income after that the variables affecting diabetes are Age and High BP.

EFA:

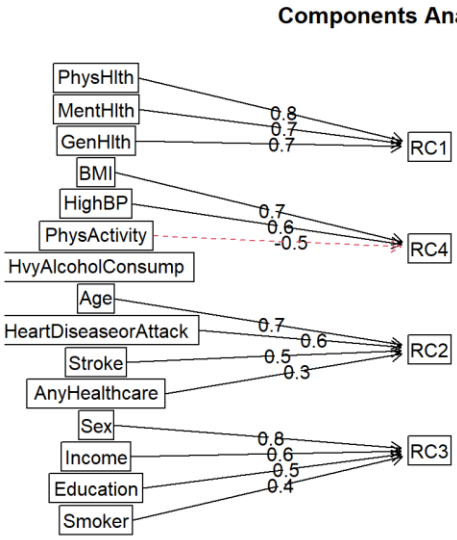
```
# Play with FA utilities

fa.parallel(Diabetes[-1]) # See factor recommendation
```



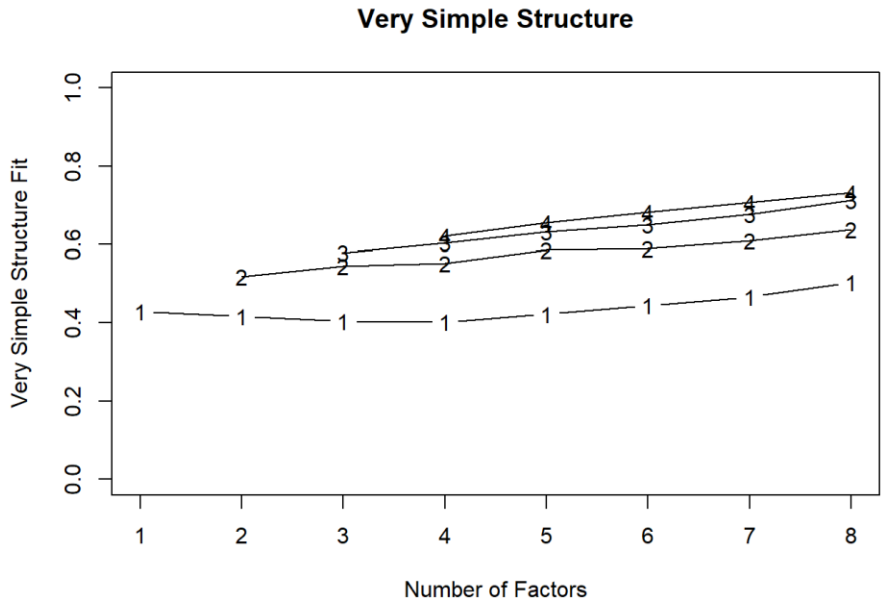

```
fa.plot(fit.pc) # See Correlations within Factors
```



```
fa.diagram(fit.pc) # Visualize the relationship
```




```
vss(Diabetes[-1]) # See Factor recommendations for a simple structure
```



Logistic Regression:

```
logistic_simple <- glm(Diabetes_binary ~ Age, data=data, family="binomial")
summary(logistic_simple)
```

```
##
## Call:
## glm(formula = Diabetes_binary ~ Age, family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7346  -1.0487   0.7088   1.0469   2.0085
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.81250    0.38872  -7.235 4.64e-13 ***
## Age          0.31275    0.04126   7.580 3.46e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 691.76  on 498  degrees of freedom
## Residual deviance: 620.66  on 497  degrees of freedom
## AIC: 624.66
##
## Number of Fisher Scoring iterations: 4
```

```
#  generating full multiple logistic regression
full_model <- glm(Diabetes_binary ~ Age + Sex + BMI, family = binomial(link = logit))
summary(full_model)
```

```
##
## Call:
## glm(formula = Diabetes_binary ~ Age + Sex + BMI, family = binomial(link = logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8327  -0.9498   0.2288   0.9602   1.9884
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.30130    0.75344  -8.363  < 2e-16 ***
## Age          0.37306    0.04626   8.065 7.34e-16 ***
## Sex         -0.57479    0.20906  -2.749  0.00597 **
## BMI          0.10595    0.01730   6.124 9.12e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 691.76  on 498  degrees of freedom
## Residual deviance: 568.06  on 495  degrees of freedom
## AIC: 576.06
##
## Number of Fisher Scoring iterations: 4
```

```
# exponentiate the confidence intervals around the log odds for each predictor variable to obtain the odds
exp(confint(full_model))
```

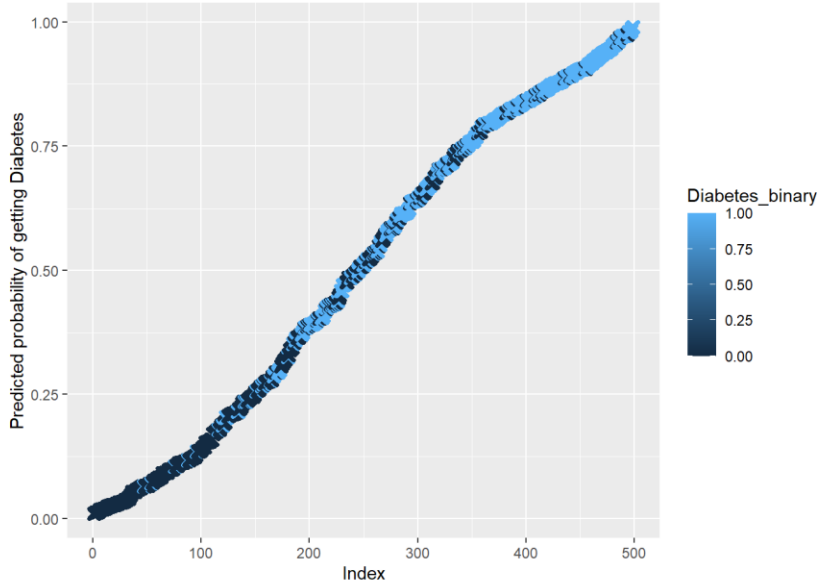
```
## Waiting for profiling to be done...
```

```
##                    2.5 %      97.5 %
## (Intercept) 0.0003935826 0.007581183
## Age         1.3303551559 1.595323514
## Sex         0.3722926660 0.845726242
## BMI         1.0758775641 1.151485599
```

```r
logistic <- glm(Diabetes_binary ~ ., data=data, family="binomial")
summary(logistic)
```

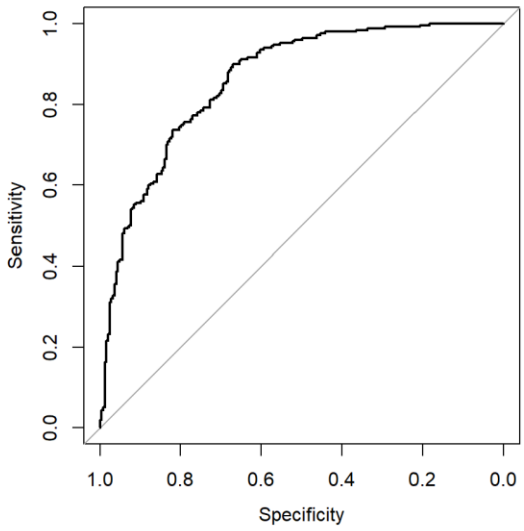
```
##
## Call:
## glm(formula = Diabetes_binary ~ ., family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7023  -0.7008   0.1464   0.7107   2.3385
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -5.934978   1.298256  -4.571 4.84e-06 ***
## HighBP                  1.004430   0.258644   3.883 0.000103 ***
## BMI                     0.071747   0.020104   3.569 0.000359 ***
## Smoker                 -0.360526   0.244289  -1.476 0.139994
## Stroke                  0.434285   0.451241   0.962 0.335837
## HeartDiseaseorAttack    0.009647   0.314612   0.031 0.975539
## PhysActivity           -0.510892   0.263966  -1.935 0.052936 .
## HvyAlcoholConsump      -0.049633   0.577585  -0.086 0.931521
## AnyHealthcare           0.305963   0.545361   0.561 0.574778
## GenHlth                 0.772766   0.146000   5.293 1.20e-07 ***
## MentHlth               -0.026902   0.015747  -1.708 0.087573 .
## PhysHlth               -0.041897   0.014747  -2.841 0.004496 **
## Sex                    -0.261409   0.256194  -1.020 0.307558
## Age                     0.282123   0.055244   5.107 3.27e-07 ***
## Education              -0.045911   0.130553  -0.352 0.725088
## Income                 -0.141062   0.064429  -2.189 0.028567 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 691.76  on 498  degrees of freedom
## Residual deviance: 463.72  on 483  degrees of freedom
## AIC: 495.72
##
## Number of Fisher Scoring iterations: 5
```

```
predicted.data <- data.frame(probability.of.Diabetes_binary=logistic$fitted.values,Diabetes_binary=data$Diabetes_
binary)
predicted.data <- predicted.data[order(predicted.data$probability.of.Diabetes_binary, decreasing=FALSE),]
predicted.data$rank <- 1:nrow(predicted.data)
## Lastly, we can plot the predicted probabilities for each sample having
## heart disease and color by whether or not they actually had heart disease
ggplot(data=predicted.data, aes(x=rank, y=probability.of.Diabetes_binary)) +
geom_point(aes(color=Diabetes_binary), alpha=1, shape=4, stroke=2) +
xlab("Index") +
ylab("Predicted probability of getting Diabetes")
```



```
par(pty = "s")
roc(data$Diabetes_binary,logistic$fitted.values,plot=TRUE)
```

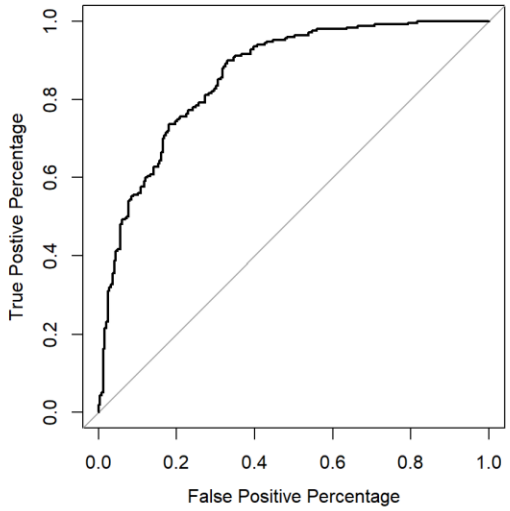
```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```




```
##
## Call:
## roc.default(response = data$Diabetes_binary, predictor = logistic$fitted.values,     plot = TRUE)
##
## Data: logistic$fitted.values in 249 controls (data$Diabetes_binary 0) < 250 cases (data$Diabetes_binary 1).
## Area under the curve: 0.8606
```

```
roc(data$Diabetes_binary,logistic$fitted.values,plot=TRUE, legacy.axes=TRUE, xlab="False Positive Percentage", yl
ab="True Postive Percentage")
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

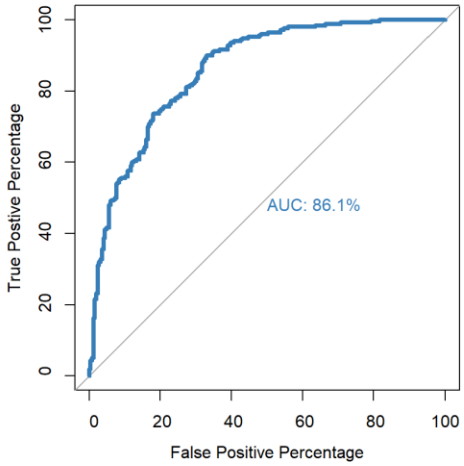



```
##
## Call:
## roc.default(response = data$Diabetes_binary, predictor = logistic$fitted.values,     plot = TRUE, legacy.axes
= TRUE, xlab = "False Positive Percentage",     ylab = "True Postive Percentage")
##
## Data: logistic$fitted.values in 249 controls (data$Diabetes_binary 0) < 250 cases (data$Diabetes_binary 1).
## Area under the curve: 0.8606
```

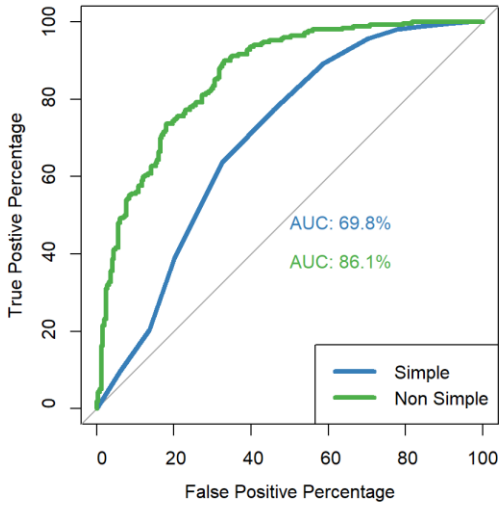
```
##
## Call:
## roc.default(response = data$Diabetes_binary, predictor = logistic$fitted.values,     percent = TRUE, plot = TR
UE, legacy.axes = TRUE, xlab = "False Positive Percentage",     ylab = "True Postive Percentage", col = "#377eb
8", lwd = 4)
##
## Data: logistic$fitted.values in 249 controls (data$Diabetes_binary 0) < 250 cases (data$Diabetes_binary 1).
## Area under the curve: 86.06%
```

```
roc(data$Diabetes_binary,logistic$fitted.values,plot=TRUE, legacy.axes=TRUE, xlab="False Positive Percentage", yl
ab="True Postive Percentage", col="#377eb8", lwd=4, percent=TRUE, print.auc=TRUE)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```




```
legend("bottomright", legend=c("Simple", "Non Simple"), col=c("#377eb8", "#4daf4a"), lwd=4)
```



Inferences regarding Logistic Regression

* Factors that have p<0.05 are good indicators of whether a person will have diabetes or not:

High BP, BMI,General health,Physical health and Age.

* Intercept: The log-odds of Survival when Age=0 is -2.812. For every unit increase in age the log odds of having diabetes will increase by 0.313.

* As p-value is <0.5 for Age component we will reject null hypothesis. It means that age factor affects a person getting diabetes.

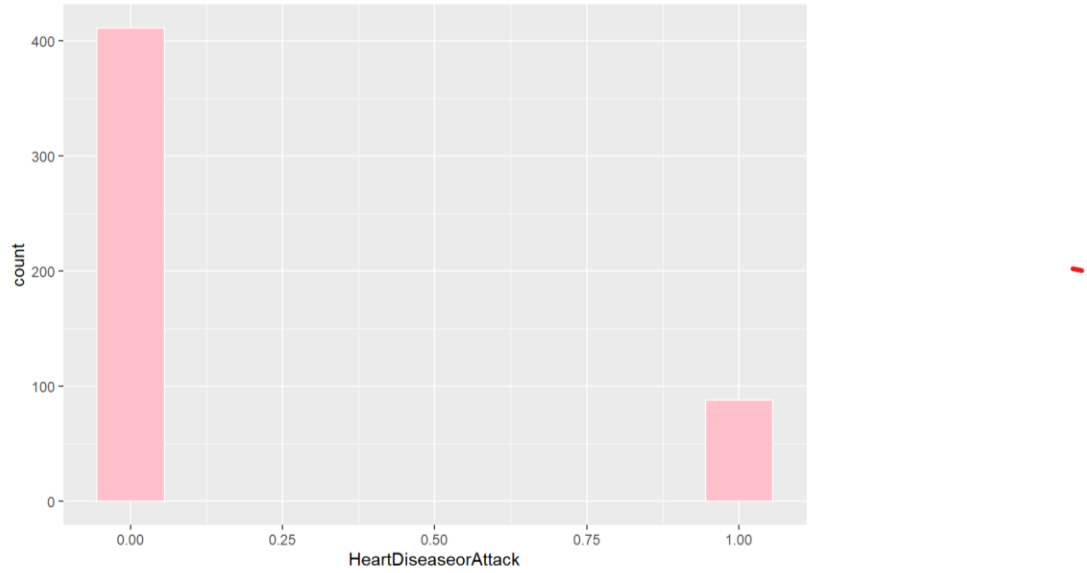
* For every increase in unit age the odds of having diabetes are 0.968 times the odds of those with one age unit less.

* The count of predicted cases increases as the probability of diabetes and age group increases, which is not surprising as diabetes is more common in older age groups.

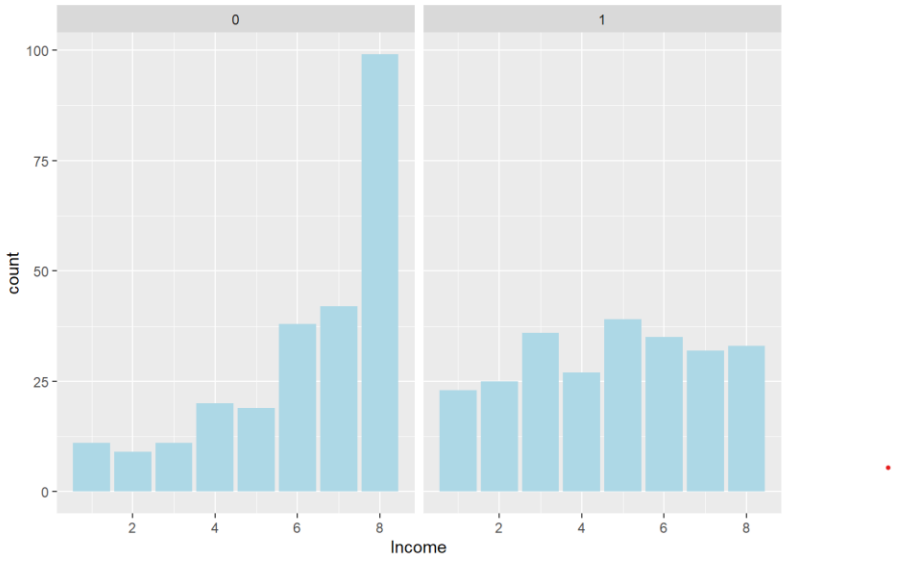
* An AUC with value of 0.86(leaning more towards 1) says that the model is able to differentiate clearly whether person will be diabetic or non-diabetic.

Scatter Plots and their inferences:

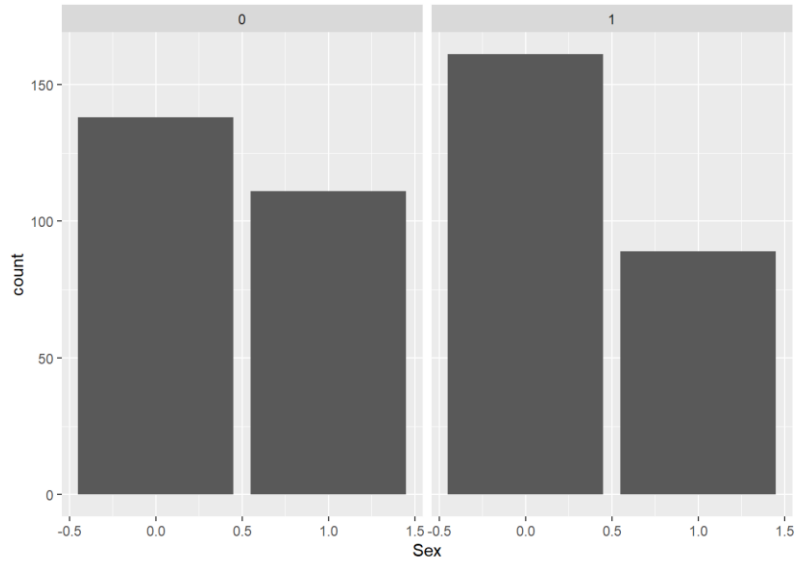
```
# Histogram

ggplot(Diabetes, aes(HeartDiseaseorAttack))+geom_histogram(fill='pink', color='white',bins=10)
```



```
ggplot(Diabetes, aes(Income)) + facet_grid(.~Diabetes_binary) + geom_bar(position="dodge",fill='lightblue')
```




```
ggplot(Diabetes, aes(Sex)) + facet_grid(.~Diabetes_binary) + geom_bar(position="dodge")+scale_fill_manual(values=
c("blue", "pink"), name="Sex", labels=c("Male", "Female"))
```



Inferences regarding Scatter Plot:

* From the bar chart we can infer probablity of Males having diabetes are higher as compared to females.

* People with higher income have less probablity of suffering from Diabetes as compared to people with more income.

* People having heartattack have less chances of having Diabetes.

Conclusion:

- General Health,Income, Physical Health, High BP, Education, Physical Activity, Age are the only important factors in determing whether a person is diabetic or not.
- Sex is not a good factor in determining whether the person has diabetes or not(as p value<0.05).
- People having heartattack have less chances of having Diabetes.
- People with higher income have less probablity of suffering from Diabetes as compared to people with more income.