

## Exploring Predictive Markers for Diabetes Using Principal Component Analysis and Logistic Regression

```
diabetes <- read.csv("diabetes.csv", header=TRUE)
head(diabetes)

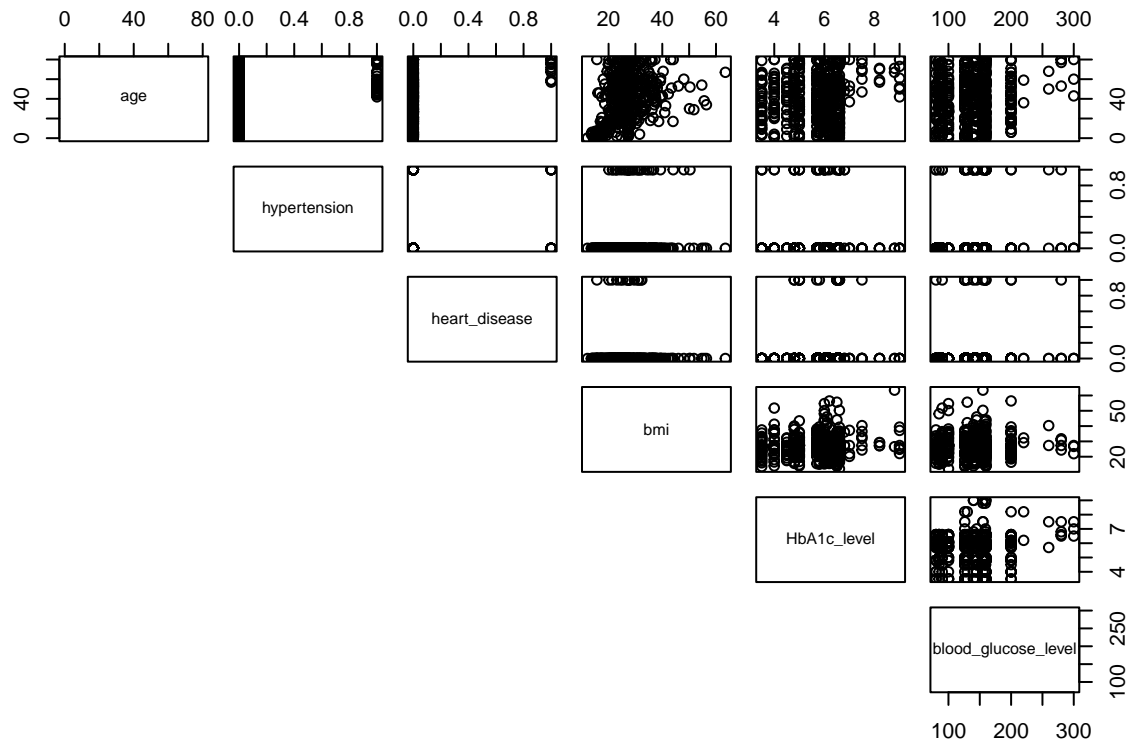
##   diabetes gender age hypertension heart_disease smoking_history  bmi
## 1         0 Female  80             0             1         never 25.19
## 2         0 Female  54             0             0         No Info 27.32
## 3         0 Male   28             0             0         never 27.32
## 4         0 Female  36             0             0         current 23.45
## 5         0 Male   76             1             1         current 20.14
## 6         0 Female  20             0             0         never 27.32
##   HbA1c_level blood_glucose_level
## 1          6.6              140
## 2          6.6              80
## 3          5.7             158
## 4          5.0             155
## 5          4.8             155
## 6          6.6              85

sapply(diabetes[c("age", "bmi", "HbA1c_level", "blood_glucose_level")], summary)

##           age           bmi HbA1c_level blood_glucose_level
## Min.      0.08000 12.15000      3.5000      80.000
## 1st Qu.  23.75000 23.10250      4.8000     100.000
## Median   42.00000 27.32000      5.8000     145.000
## Mean     41.65328 26.89078      5.6224     139.652
## 3rd Qu.  59.00000 28.93250      6.2000     159.000
## Max.     80.00000 63.48000      9.0000     300.000

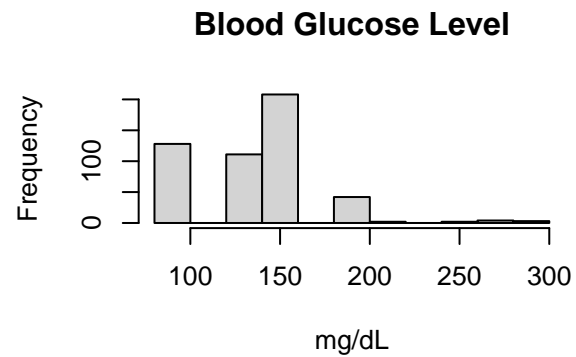
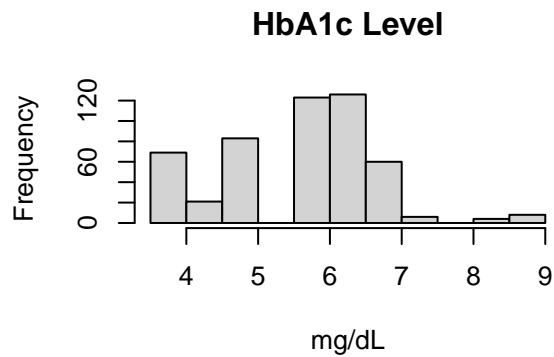
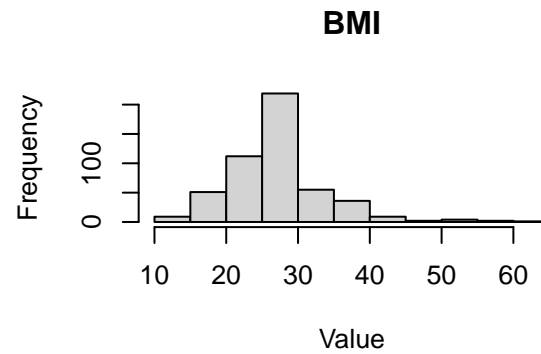
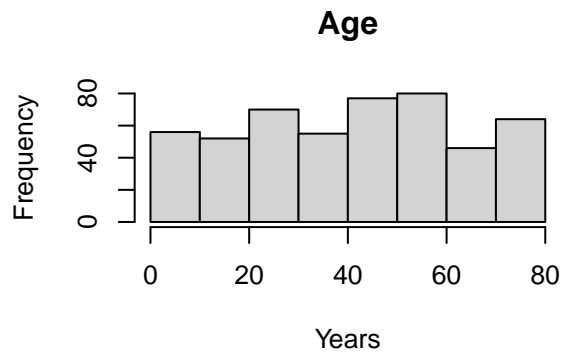
diabetes_numeric <- diabetes %>%
  select(c("age", "hypertension", "heart_disease", "bmi", "HbA1c_level", "blood_glucose_level"))

#Create a scatterplot matrix
pairs(diabetes_numeric, lower.panel = NULL)
```

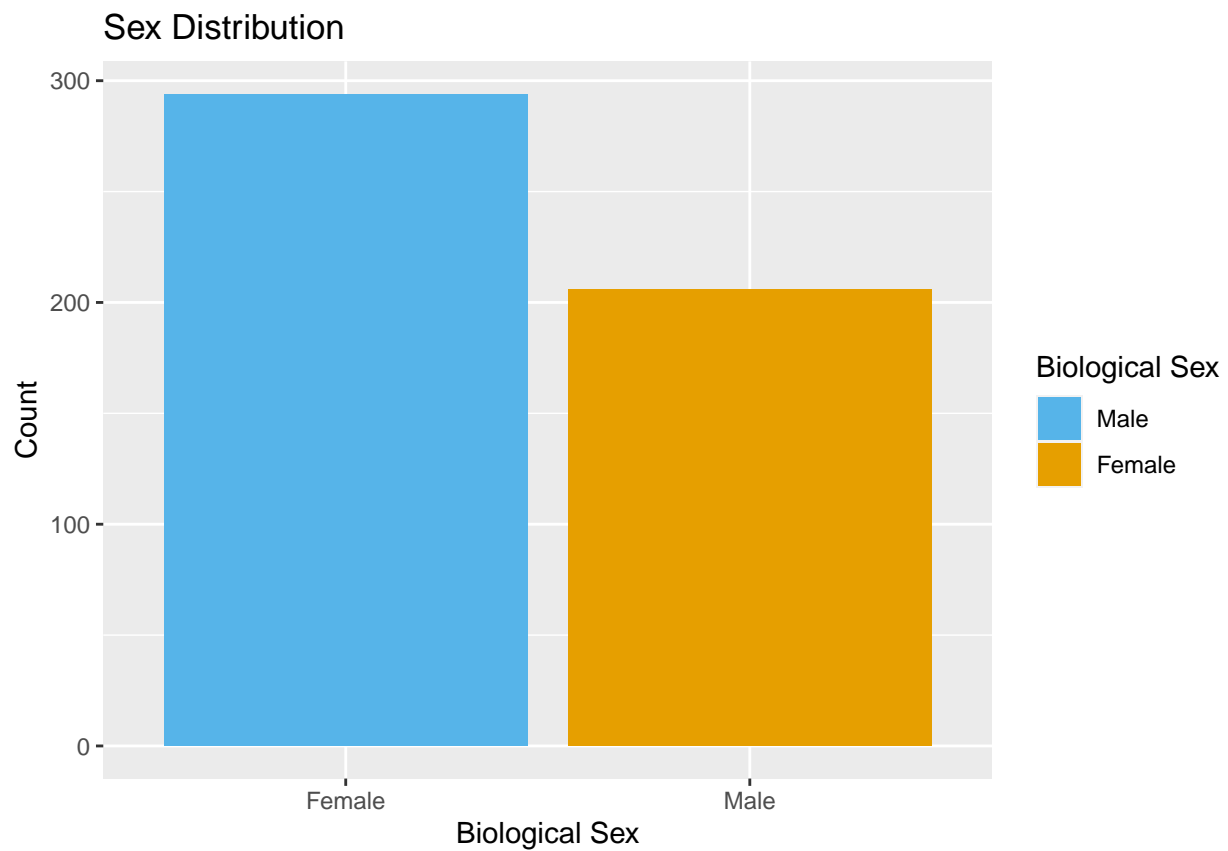


```
#Create histograms for all features
par(mfrow=c(2,2)) # Set up 2x2 grid of plots

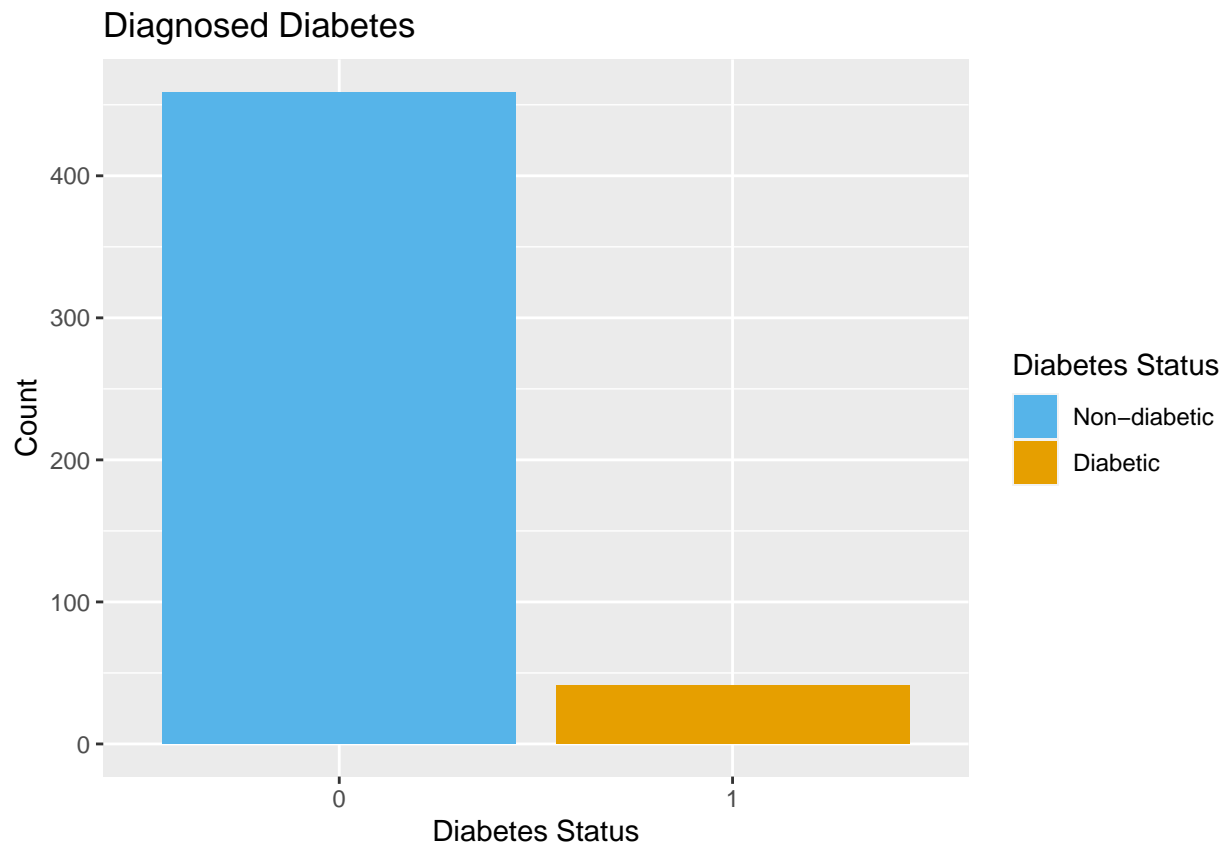
hist(diabetes$age, main="Age", xlab="Years")
hist(diabetes$bmi, main="BMI", xlab="Value")
hist(diabetes$HbA1c_level, main="HbA1c Level", xlab="mg/dL")
hist(diabetes$blood_glucose_level, main="Blood Glucose Level", xlab="mg/dL")
```



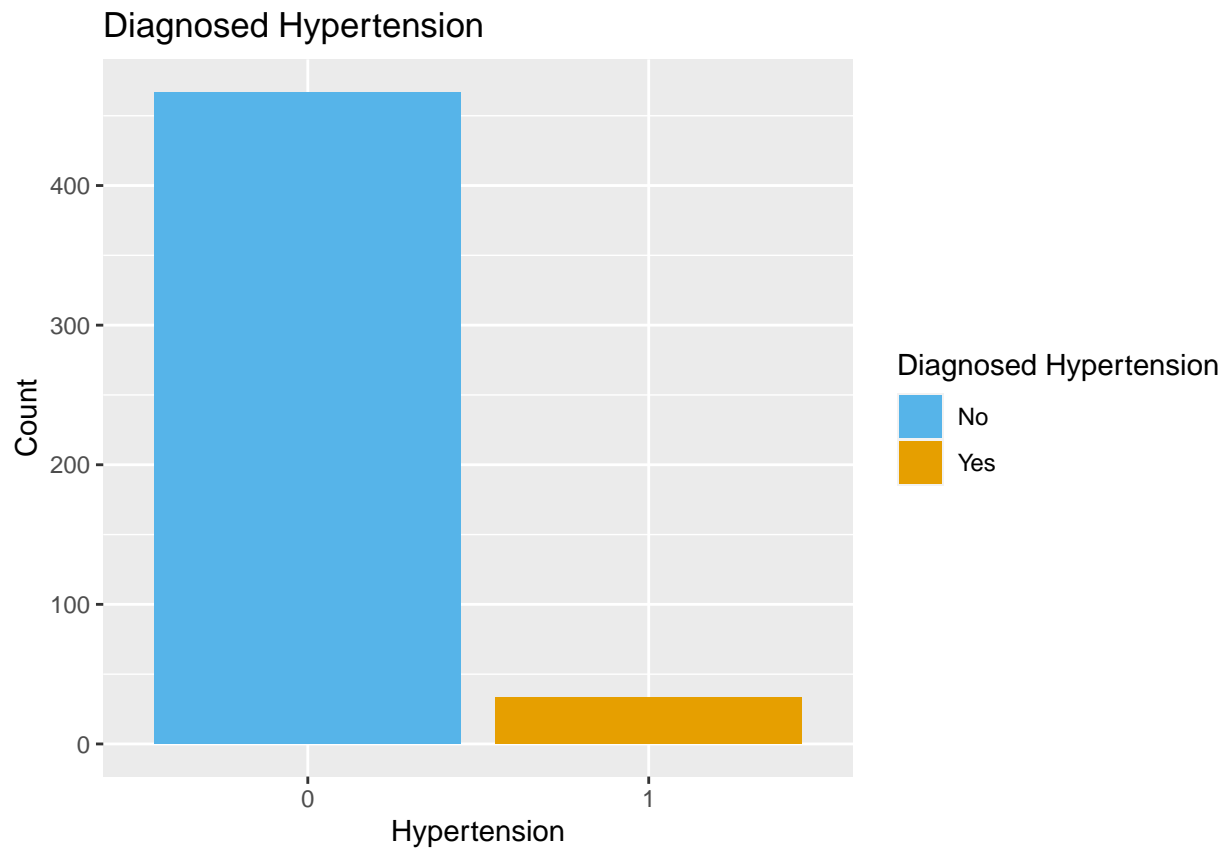
```
#Create bar graphs of discrete and categorical features
ggplot(diabetes, aes(x = factor(gender), fill = factor(gender))) +
  geom_bar() +
  labs(title="Sex Distribution", x="Biological Sex", y="Count") +
  scale_fill_manual(name = "Biological Sex",
                    values = c("#56B4E9", "#E69F00", "#999999"),
                    labels = c("Male", "Female", "Unknown"))
```



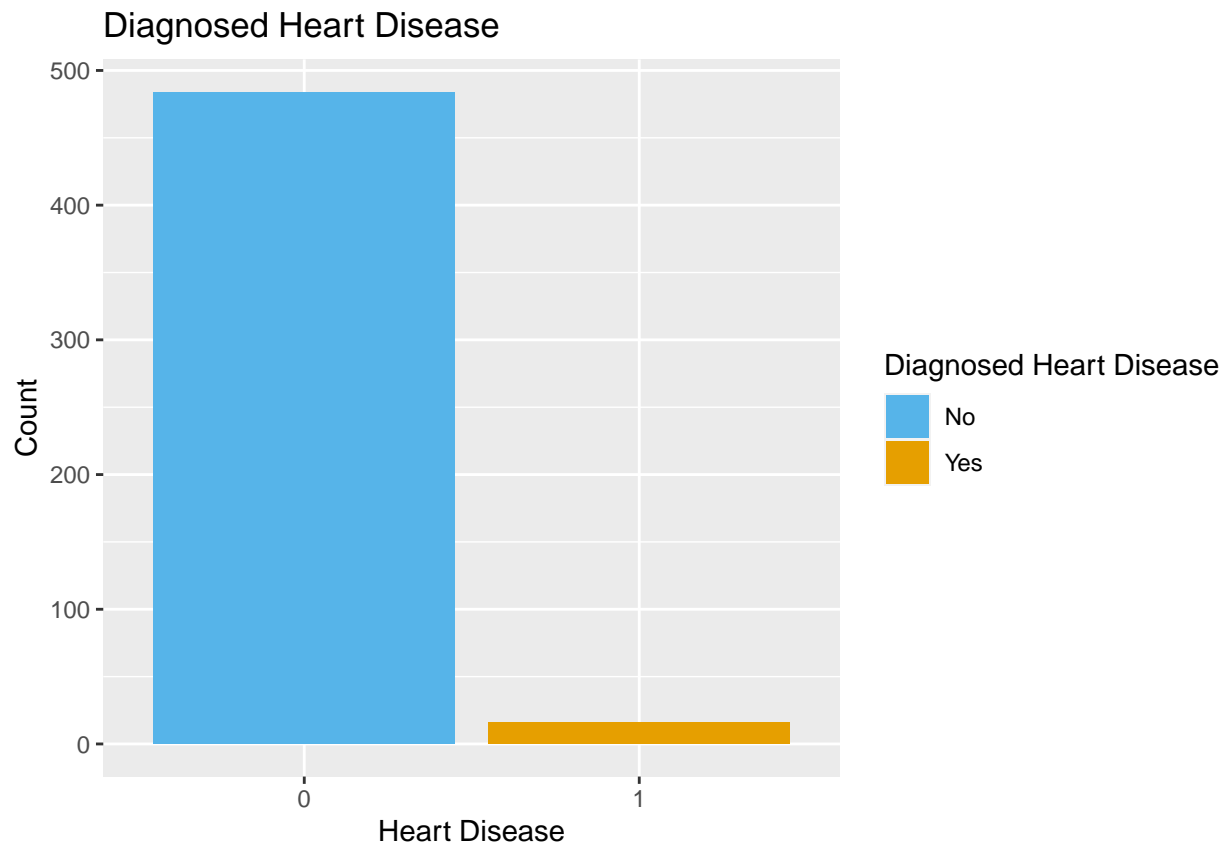
```
ggplot(diabetes, aes(x = factor(diabetes), fill = factor(diabetes))) +  
  geom_bar() +  
  labs(title = "Diagnosed Diabetes", x = "Diabetes Status", y = "Count") +  
  scale_fill_manual(name = "Diabetes Status",  
                    values = c("#56B4E9", "#E69F00"),  
                    labels = c("Non-diabetic", "Diabetic"))
```



```
ggplot(diabetes, aes(x = factor(hypertension), fill = factor(hypertension))) +  
  geom_bar() +  
  labs(title = "Diagnosed Hypertension", x = "Hypertension", y = "Count") +  
  scale_fill_manual(name = "Diagnosed Hypertension",  
                    values = c("#56B4E9", "#E69F00"),  
                    labels = c("No", "Yes"))
```



```
ggplot(diabetes, aes(x = factor(heart_disease), fill = factor(heart_disease))) +  
  geom_bar() +  
  labs(title = "Diagnosed Heart Disease", x = "Heart Disease", y = "Count") +  
  scale_fill_manual(name = "Diagnosed Heart Disease",  
                    values = c("#56B4E9", "#E69F00"),  
                    labels = c("No", "Yes"))
```



```
#Test the skewness in diabetes data
age_skew <- skewness(diabetes$age)
cat("Skewness of age:", age_skew, "\n")
```

```
## Skewness of age: -0.01478161
```

```
cat("Absolute skewness of age:", abs(age_skew), "\n")
```

```
## Absolute skewness of age: 0.01478161
```

```
bmi_skew <- skewness(diabetes$bmi)
cat("Skewness of bmi:", bmi_skew, "\n")
```

```
## Skewness of bmi: 1.208401
```

```
cat("Absolute skewness of bmi:", abs(bmi_skew), "\n")
```

```
## Absolute skewness of bmi: 1.208401
```

```
hba1c_skew <- skewness(diabetes$HbA1c_level)
cat("Skewness of HbA1c level:", hba1c_skew, "\n")
```

```
## Skewness of HbA1c level: -0.08341524
```

```
cat("Absolute skewness of HbA1c level:", abs(hba1c_skew), "\n")
```

```
## Absolute skewness of HbA1c level: 0.08341524
```

```
glucose_skew <- skewness(diabetes$blood_glucose_level)
cat("Skewness of blood glucose level:", glucose_skew, "\n")
```

```
## Skewness of blood glucose level: 0.7202705
```

```

cat("Absolute skewness of blood glucose level:", abs(glucose_skew), "\n")

## Absolute skewness of blood glucose level: 0.7202705
data_select <- diabetes %>% select(age, bmi, HbA1c_level, blood_glucose_level)

data_scale <- scale(data_select)

#Create PCA model
pca_model <- prcomp(data_scale, scale. = TRUE)
pca_model

## Standard deviations (1, .., p=4):
## [1] 1.1860099 1.0077506 0.9517091 0.8197981
##
## Rotation (n x k) = (4 x 4):
##
##          PC1          PC2          PC3          PC4
## age      -0.6465562  0.2382414 -0.07648468  0.72066375
## bmi      -0.6076987  0.4054814 -0.02835388 -0.68226330
## HbA1c_level -0.3491517 -0.5545748  0.75368930 -0.04992315
## blood_glucose_level -0.3012649 -0.6864930 -0.65214921 -0.11255318

#PC1 is made up mostly of age, bmi. PC2 is made up mostly of A1c, blood glucose. PC3 is made up mostly
summary(pca_model)

## Importance of components:
##          PC1          PC2          PC3          PC4
## Standard deviation      1.1860 1.0078 0.9517 0.8198
## Proportion of Variance 0.3517 0.2539 0.2264 0.1680
## Cumulative Proportion 0.3517 0.6056 0.8320 1.0000

pca_var <- get_pca_var(pca_model)
#Contribution % of features to their respective PC
pca_var$contrib[,1]

##          age          bmi          HbA1c_level blood_glucose_level
##          41.803487          36.929771          12.190691          9.076051

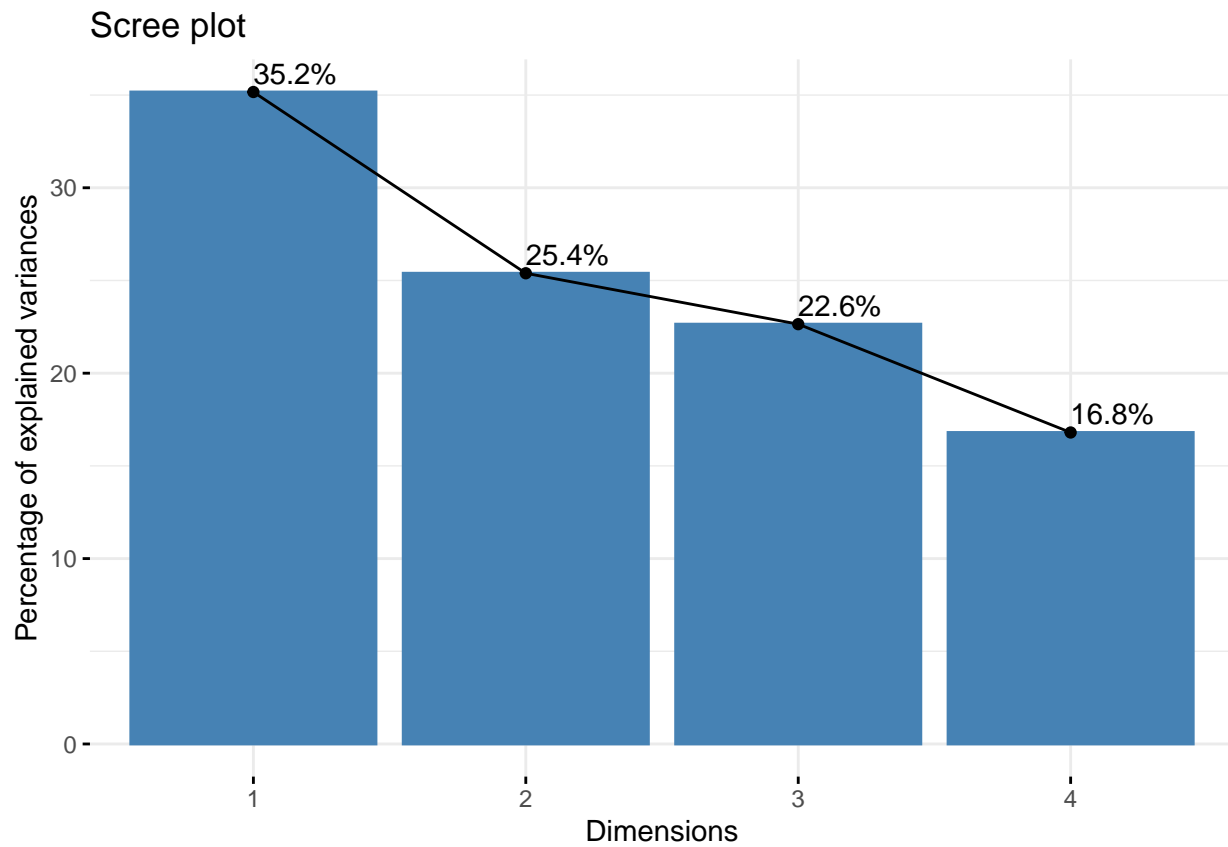
pca_var$contrib[,2]

##          age          bmi          HbA1c_level blood_glucose_level
##          5.675898          16.441513          30.755322          47.127268

fviz_eig(pca_model, addlabels = TRUE)

```



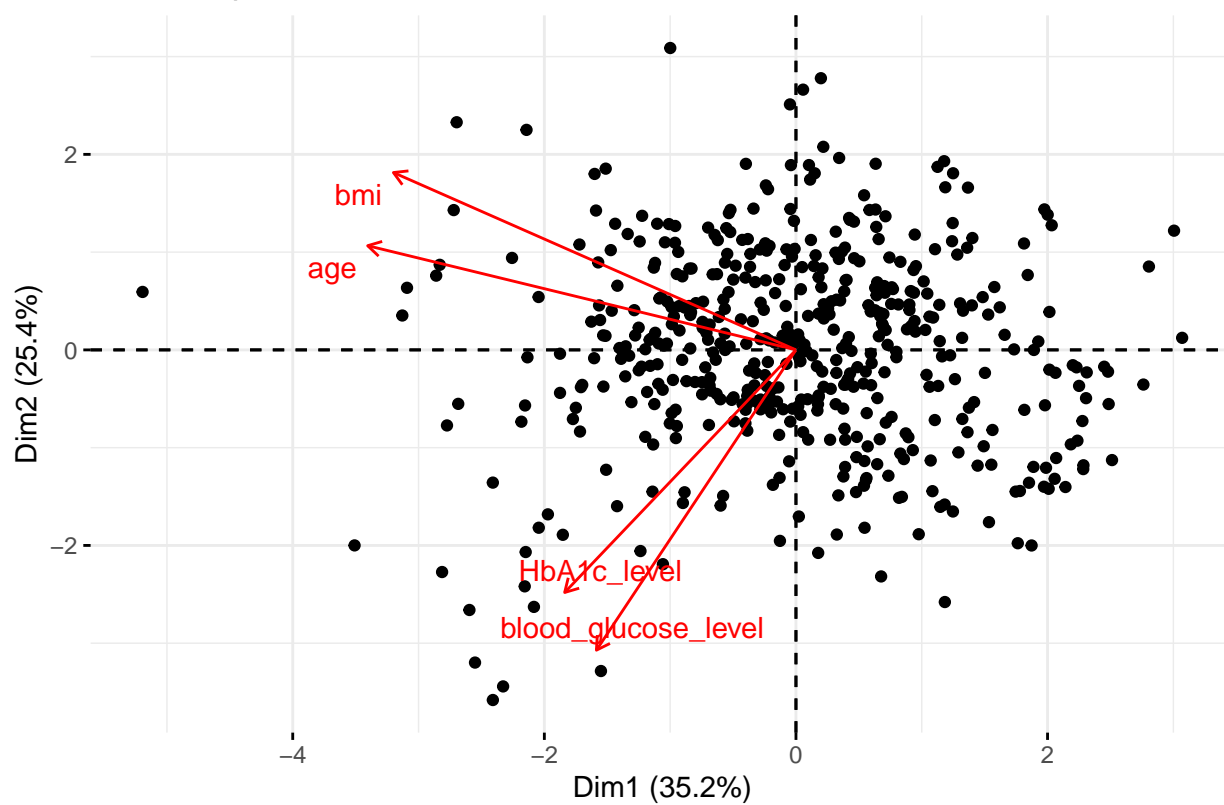


```
pca_scores <- as.data.frame(pca_model$x)

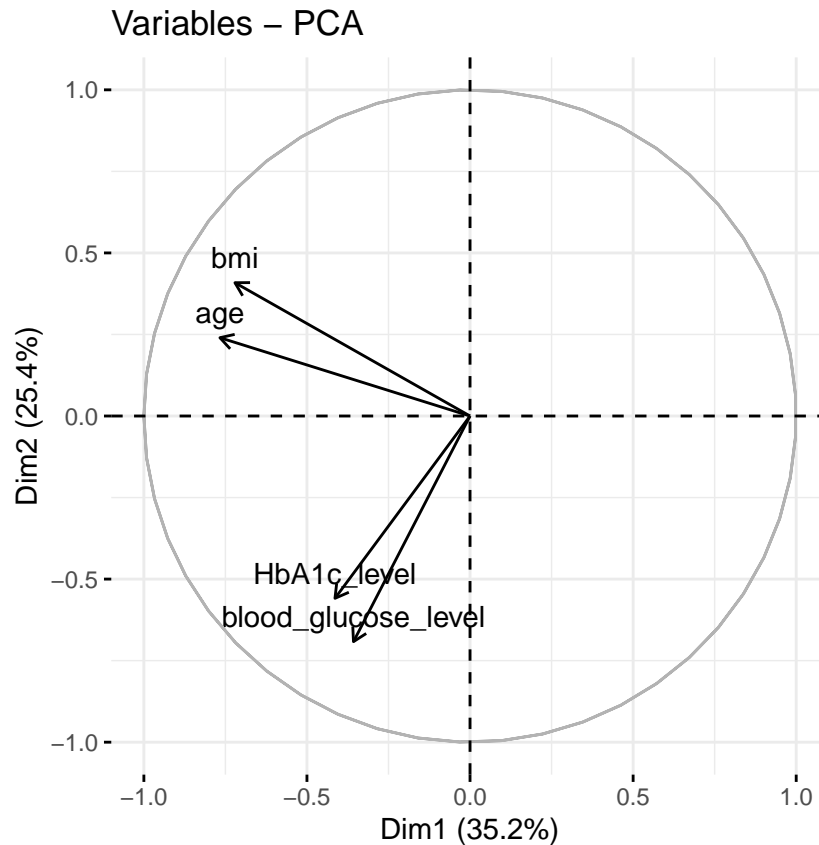
pca_scores$outcome <- diabetes$outcome

#PCA biplot
fviz_pca_biplot(pca_model, geom = c("point", "text"), label = "var", col.var = "red", repel = TRUE)
```

PCA – Biplot



```
fviz_pca_var(pca_model)
```

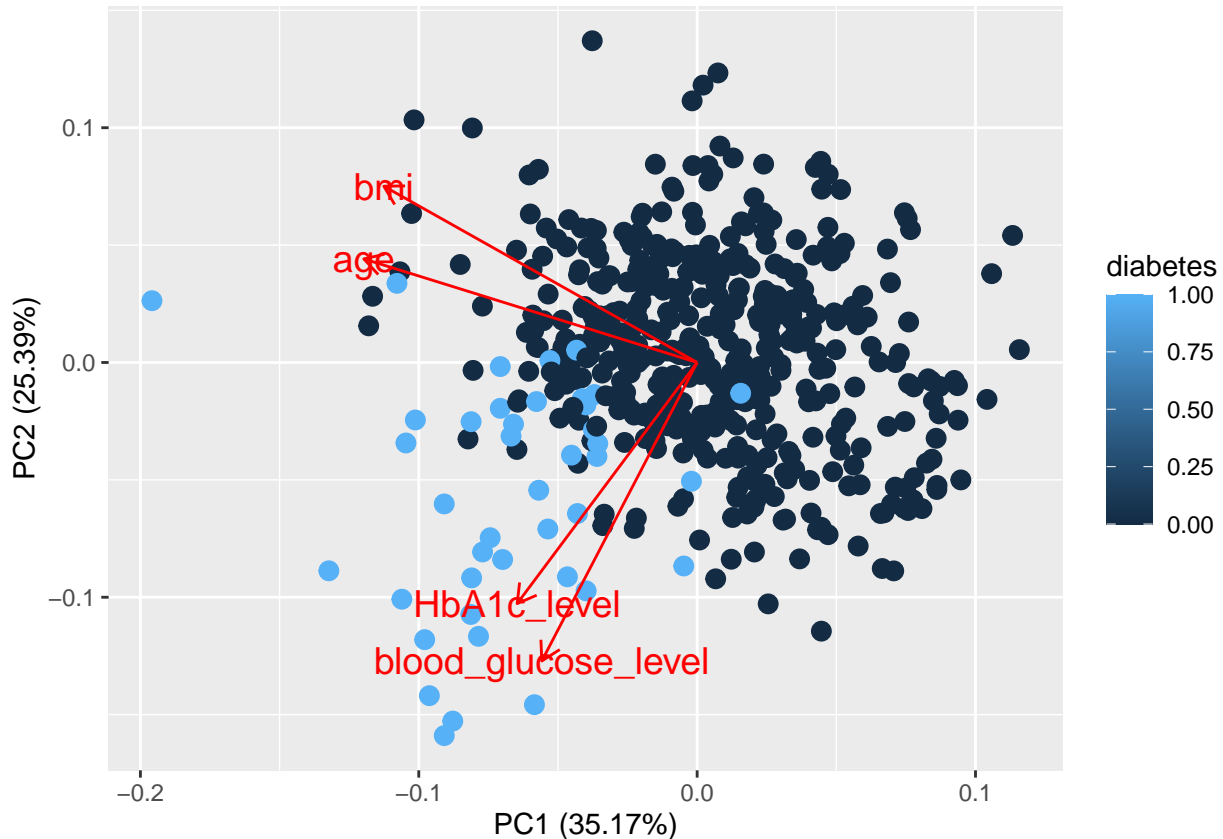


```
#Create logistic regression model with outcome as response and first 2 principal components as predictors
model <- glm(diabetes~pca_scores$PC1 + pca_scores$PC2 + pca_scores$PC3 + pca_scores$PC4, data = diabetes)

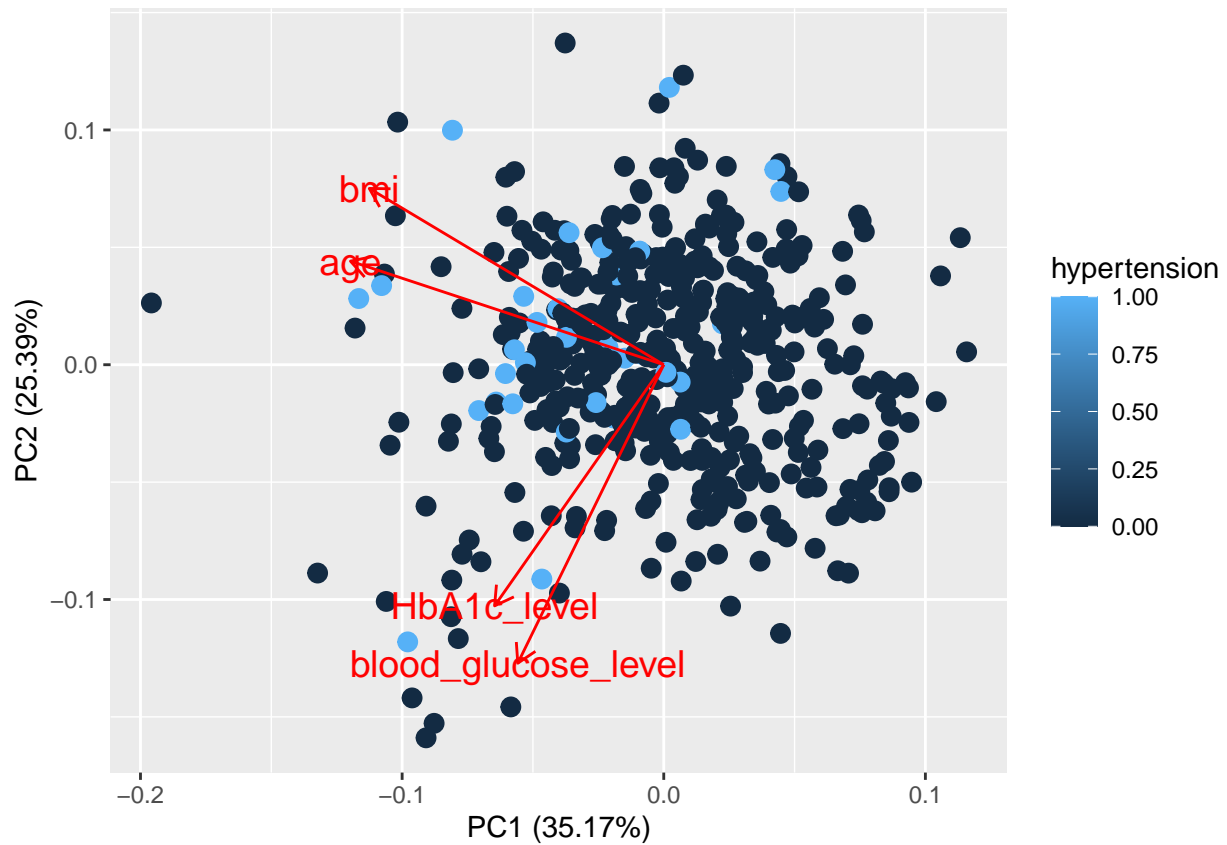
summary(model)
```

```
##
## Call:
## glm(formula = diabetes ~ pca_scores$PC1 + pca_scores$PC2 + pca_scores$PC3 +
##      pca_scores$PC4, family = binomial, data = diabetes)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.75904    0.99410  -6.799 1.05e-11 ***
## pca_scores$PC1 -2.93566    0.51983  -5.647 1.63e-08 ***
## pca_scores$PC2 -3.43703    0.68916  -4.987 6.12e-07 ***
## pca_scores$PC3  2.23192    0.69052   3.232 0.00123 **
## pca_scores$PC4  0.02427    0.30519   0.080 0.93661
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 283.627  on 499  degrees of freedom
## Residual deviance:  85.997  on 495  degrees of freedom
## AIC: 95.997
##
## Number of Fisher Scoring iterations: 9
```

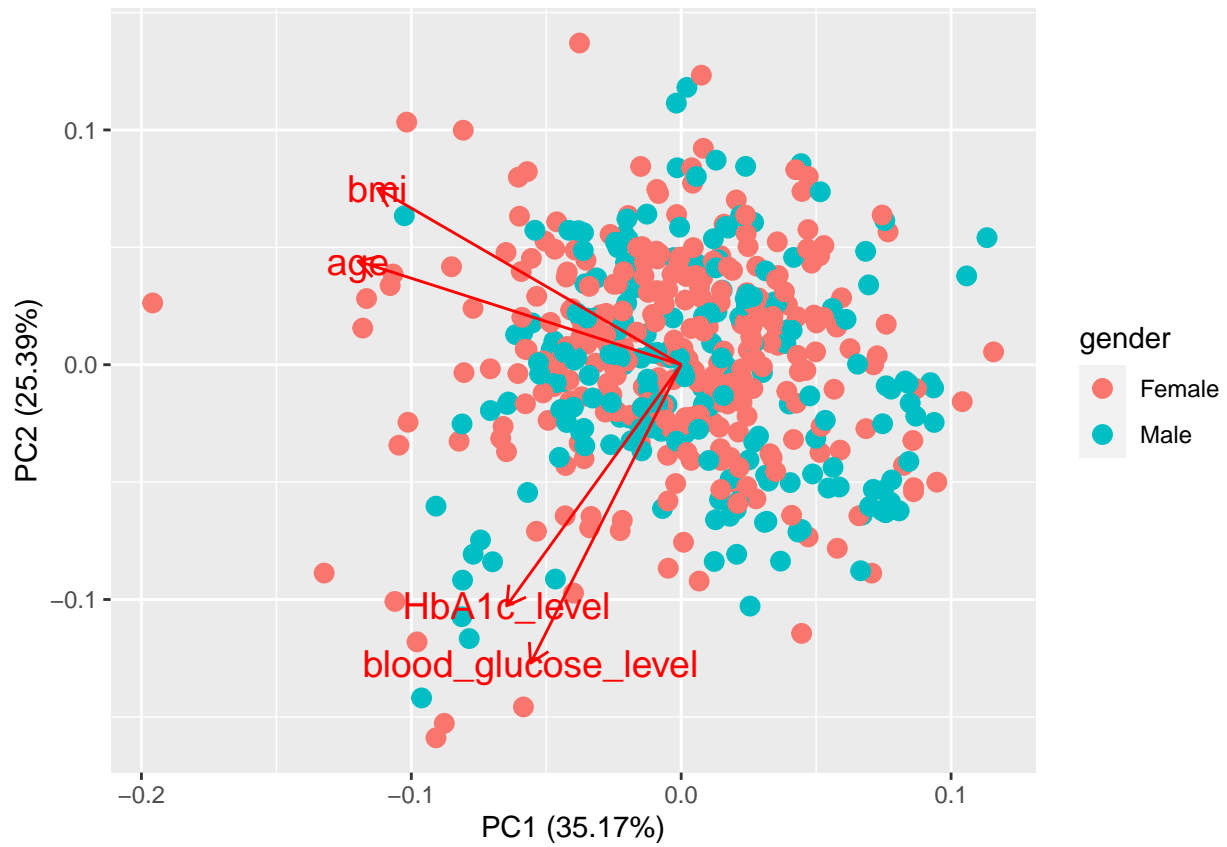
```
#Create scatterplots similar to biplot with relationship to discrete/categorical features
autoplot(
  pca_model, #The PCA model object
  data = diabetes, #The dataset being plotted
  colour = 'diabetes', #The variable being used to color the points
  loadings=TRUE, #Indicates that the plot should also show the loadings
  size = 3, #The size of the points
  loadings.label = TRUE, #Indicates that the plot should label the loadings
  loadings.label.size=5 #The size of the loading labels
)
```



```
autoplot(
  pca_model,
  data = diabetes,
  colour = 'hypertension',
  loadings=TRUE,
  size = 3,
  loadings.label = TRUE,
  loadings.label.size=5
)
```



```
autoplot(
  pca_model,
  data = diabetes,
  colour = 'gender',
  loadings=TRUE,
  size = 3,
  loadings.label = TRUE,
  loadings.label.size=5
)
```



```
autoplot(  
  pca_model,  
  data = diabetes,  
  colour = 'heart_disease',  
  loadings=TRUE,  
  size = 3,  
  loadings.label = TRUE,  
  loadings.label.size=5  
)
```

