Assignment 5

Nitya Kari

2023-12-14

Load Data and Packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.2.1     v dplyr   1.1.1
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(HDclassif)
```

```
## Loading required package: MASS
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(rms)
```

```
## Loading required package: Hmisc
##
## Attaching package: 'Hmisc'
##
## The following object is masked from 'package:psych':
##
##     describe
##
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
##
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(dplyr)


data <- read.csv("/Users/nitwit/Desktop/google trends/diabetes.csv")
```

This dataset contains information about whether a patient has diabetes or not based on certain diagnostic measurements. All the patients in this dataset are Pima Indian females who are at least 21 years of age. There are many predictor variables in this dataset (pregnancy, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, age) as well as the diabetes outcome variable. There are 768 seperate values in this dataset.

This dataset was obtained from Kaggle.com (https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database) but is originally from the National Institute of Diabetes and Digestive and Kidney Diseases.

Research Question

For this dataset, my research question is "Can the aforementioned variables accurately predict whether a person has diabetes or not?"

Essentially, I would like to use the predictor variables in this dataset to see I can train an algorithm to detect whether a person of Pima Indian descent has diabetes.

Ho (null hypothesis): There is no relationship or predictive power between the variables and the presence of diabetes Ha (alternative hypothesis): There is a relationship or predictive power between the variables and the presence of diabetes

Variables of Interest

Outcome variable -> whether the person has diabetes or not Predictor variables -> pregnancy, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, age The predictor variables are all continuous numeric variables while the outcome is categorical (0: no diabetes, 1: has diabetes).

Data Wrangling

```
data_complete <- na.omit(data)

outcome <- data_complete$Outcome

# data_complete <- data_complete %>%
#   select(-Outcome)

#check for collinearity
numeric_columns <- data[sapply(data, is.numeric)]
correlation_matrix <- cor(numeric_columns)
highly_correlated <- which(abs(correlation_matrix) >= 0.70, arr.ind = TRUE)
print(highly_correlated)
```

```
##                          row col
## Pregnancies                1   1
## Glucose                    2   2
## BloodPressure              3   3
## SkinThickness              4   4
## Insulin                    5   5
## BMI                        6   6
## DiabetesPedigreeFunction   7   7
## Age                        8   8
## Outcome                    9   9
```

```
# ensuring that data is on the same scale
numeric_columns <- data[sapply(data, is.numeric)]
scaled_data <- scale(numeric_columns)
scaled_data <- as.data.frame(scaled_data)
str(scaled_data)
```

```
## ’data.frame':    768 obs. of  9 variables:
##  $ Pregnancies             : num  0.64 -0.844 1.233 -0.844 -1.141 ...
##  $ Glucose                 : num  0.848 -1.123 1.942 -0.998 0.504 ...
##  $ BloodPressure           : num  0.15 -0.16 -0.264 -0.16 -1.504 ...
##  $ SkinThickness           : num  0.907 0.531 -1.287 0.154 0.907 ...
##  $ Insulin                 : num  -0.692 -0.692 -0.692 0.123 0.765 ...
##  $ BMI                     : num  0.204 -0.684 -1.103 -0.494 1.409 ...
##  $ DiabetesPedigreeFunction: num  0.468 -0.365 0.604 -0.92 5.481 ...
##  $ Age                     : num  1.4251 -0.1905 -0.1055 -1.0409 -0.0205 ...
##  $ Outcome                 : num  1.365 -0.732 1.365 -0.732 1.365 ...
```

All the predictor variables have high collinearity (above 0.7)

Analysis and Visualization

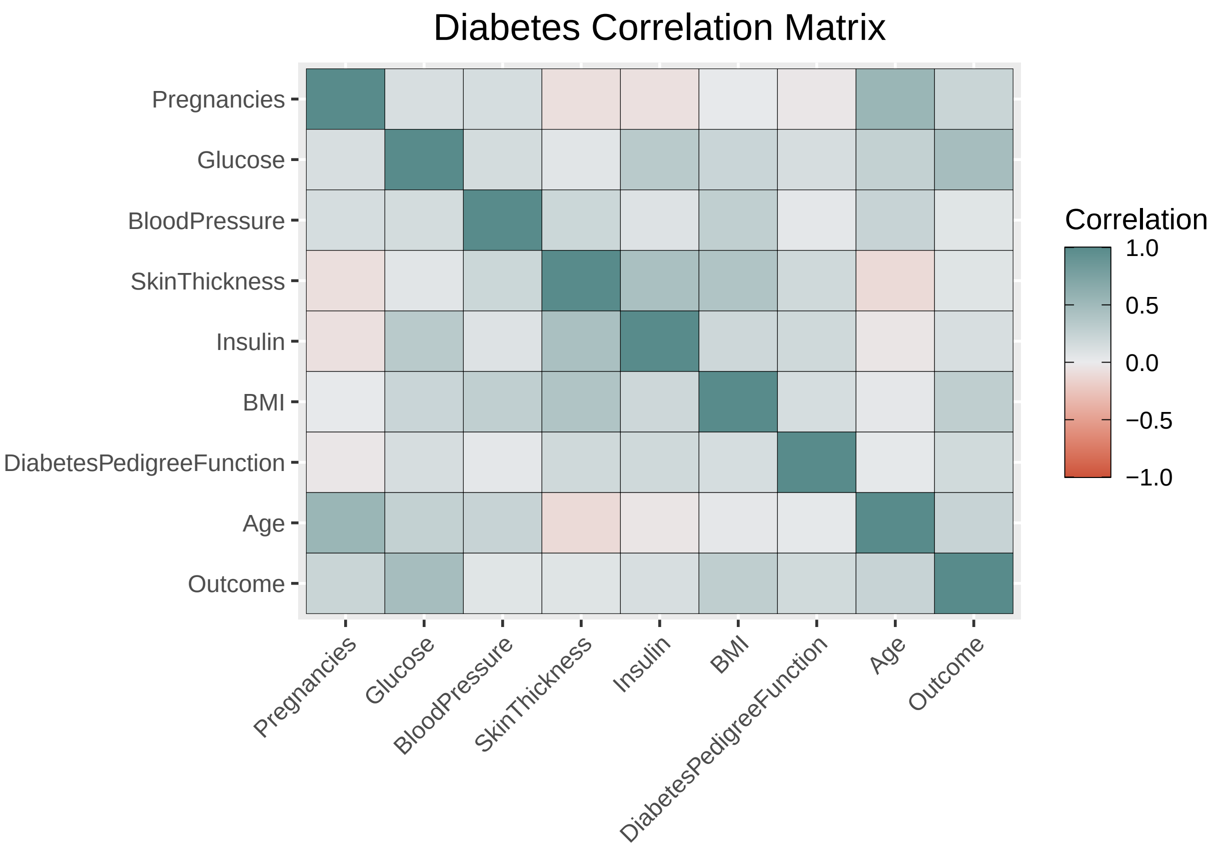
I plan to use a PCA to perform an unsupervised dimension reduction and because the variables are aggregate.

```
data_pca <- prcomp(scaled_data, center = TRUE, scale = TRUE)
summary(data_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.5338 1.3320 1.0584 0.93912 0.91903 0.85724 0.69887
## Proportion of Variance 0.2614 0.1971 0.1245 0.09799 0.09385 0.08165 0.05427
## Cumulative Proportion  0.2614 0.4585 0.5830 0.68100 0.77485 0.85650 0.91077
##                            PC8     PC9
## Standard deviation     0.64667 0.62041
## Proportion of Variance 0.04646 0.04277
## Cumulative Proportion  0.95723 1.00000
```

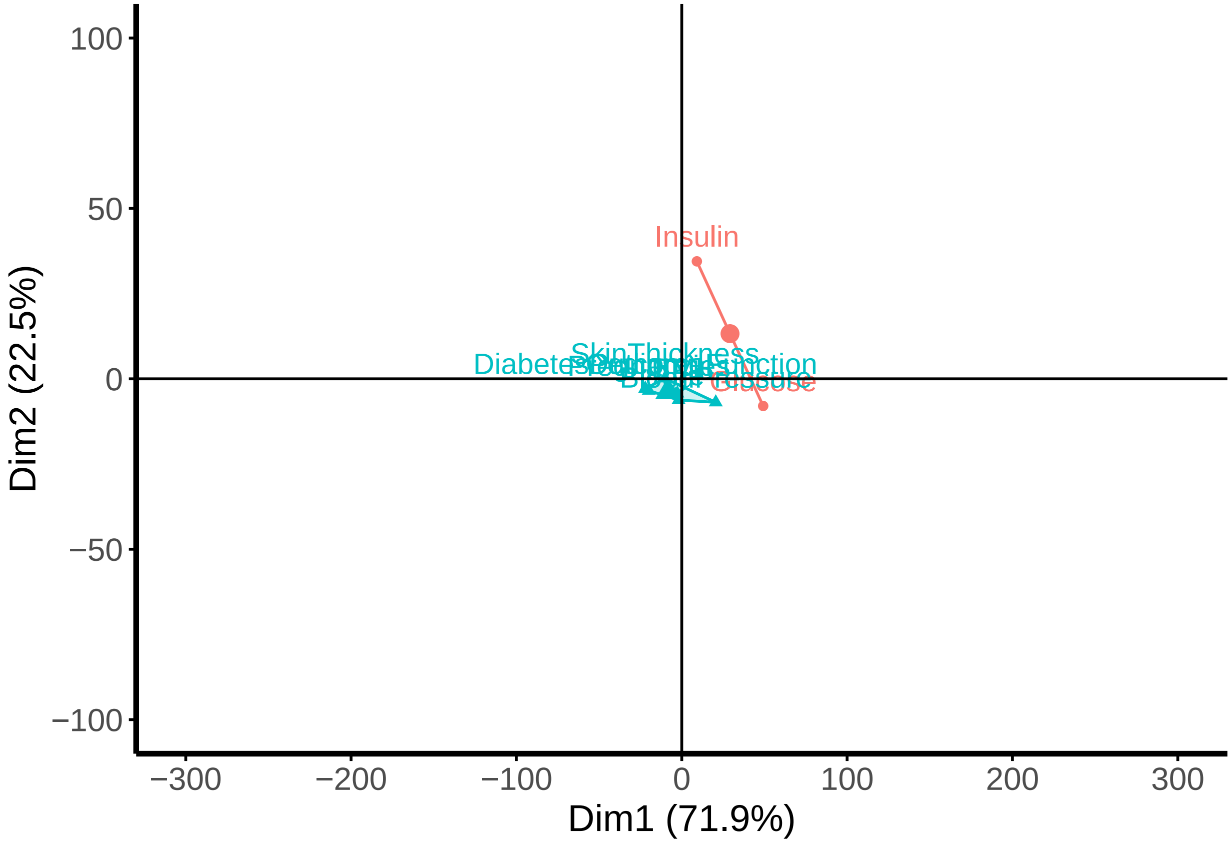
```
setup_ggplot2_heatmap <- function(
    correlation_matrix, # input for correlation matrix
    type = c("full", "lower", "upper")
    # whether matrix should depict the full, lower,
    # or upper matrix
)
{

  # Ensure correlation matrix is a `matrix` object
  corr_mat <- as.matrix(correlation_matrix)

  # Determine triangle
  if(type == "lower"){
    corr_mat[upper.tri(corr_mat)] <- NA
  }else if(type == "upper"){
    corr_mat[lower.tri(corr_mat)] <- NA
  }

  # Convert to long format
  corr_df <- data.frame(
    Var1 = rep(colnames(corr_mat), each = ncol(corr_mat)),
    Var2 = rep(colnames(corr_mat), times = ncol(corr_mat)),
    Correlation = as.vector(corr_mat)
  )

  # Set levels
  corr_df$Var1 <- factor(
    corr_df$Var1, levels = colnames(corr_mat)
  )
  corr_df$Var2 <- factor(
    corr_df$Var2, levels = rev(colnames(corr_mat))
  )
  corr_df$Correlation <- as.numeric(corr_df$Correlation)

  # Return data frame for plotting
```

```r
  return(corr_df)

}

# Obtain full matrix
data_lower <- setup_ggplot2_heatmap(
  cor(data_complete), type = "full"
)

# Plot correlation matrix
ggplot(
  data = data_lower,
  aes(x = Var1, y = Var2, fill = Correlation)
) +
  geom_tile(color = "black") +
  scale_fill_gradient2(
    low = "#CD533B", mid = "#EAEBED",
    high = "#588B8B", limits = c(-1, 1),
    guide = guide_colorbar(
      frame.colour = "black",
      ticks.colour = "black"
    )
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title = element_blank(),
    plot.title = element_text(size = 14, hjust = 0.5)
  ) +
  labs(title = "Diabetes Correlation Matrix")
```



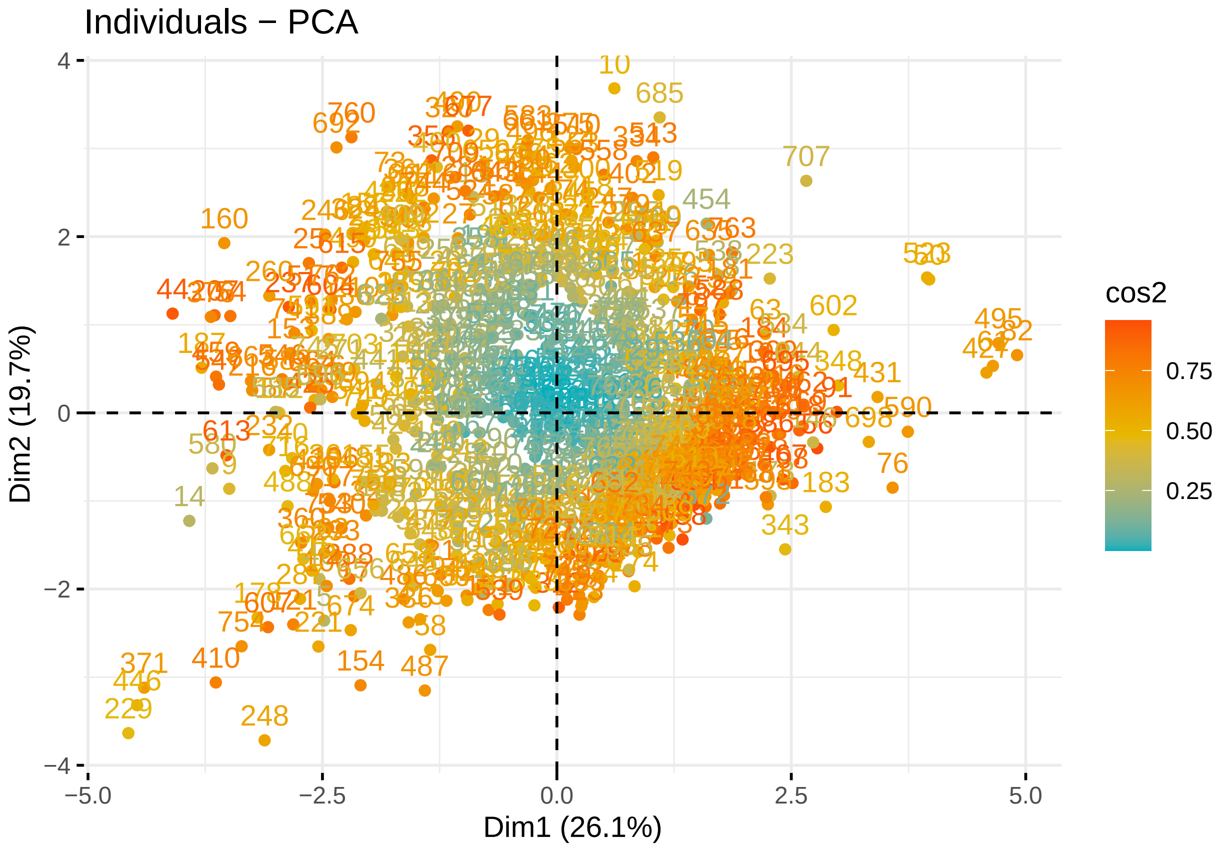
> Looking at the correlation matrix, we can see that most of the varaibles have a slight correlation to each other. It is interesting to note that not many of the predictor variables are blaringly correlated to the outcome variable (has diabetes). Glucose, as a predictor variable, appears to have the most correlation to the outcome variable. However, all of the correlations to the outcome variable appear to be positive.

```
# Obtain two principal components
components <- kmeans(
  x = t(as.matrix(data_complete)),
  centers = 2
)

# Visualize
fviz_cluster(
  components,
  data = t(as.matrix(data_complete))
) +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0) +
  scale_x_continuous(
    limits = c(-300, 300),
    breaks = seq(-300, 300, 100)
  ) +
  scale_y_continuous(
    limits = c(-100, 100),
    breaks = seq(-100, 100, 50)
  ) +
  theme(
    plot.title = element_blank(),
    panel.background = element_blank(),
    axis.line = element_line(linewidth = 1),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    legend.position = "none"
  )
```

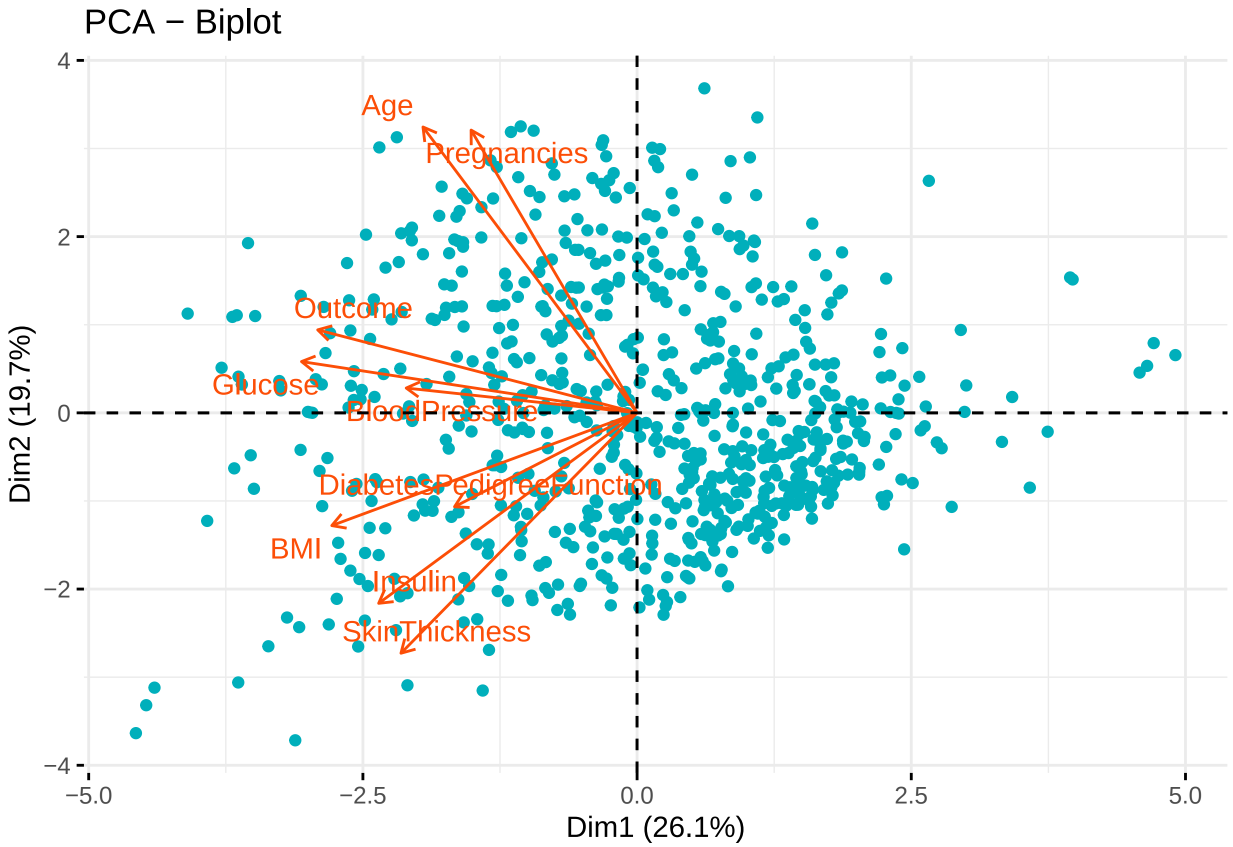


> Through this, we can see the directionality of greatest variance.

```
# Produce 2-dimensional plot
fviz_pca_ind(
  data_pca,
  c = "point", # Observations
  col.ind = "cos2", # Quality of representation
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = FALSE
)
```



```
# Biplot
fviz_pca_biplot(
  data_pca, repel = TRUE,
  col.var = "#FC4E07", # Variables color
  col.ind = "#00AFBB",  # Individuals color
  label = "var" # Variables only
)
```



> The individual 2D plot showcases that there is a lot more variability in the first, second, and third quadrants. The direction of the variability also matches up the direction we saw in the cluster visualization. This is also further emphasized by the Biplot visualization as all of the variable were in the left quadrants, albeit their varied directions.

```
# Barlett's test
cortest.bartlett(scaled_data)
```

```
## R was not square, finding R from data
```

```
## $chisq
## [1] 1223.57
##
## $p.value
## [1] 1.375839e-233
##
## $df
## [1] 36
```

```
# KMO
KMO(scaled_data)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = scaled_data)
## Overall MSA =  0.62
## MSA for each item =
##              Pregnancies                  Glucose            BloodPressure
##                     0.59                     0.61                     0.65
##            SkinThickness                  Insulin                      BMI
##                     0.58                     0.58                     0.68
## DiabetesPedigreeFunction                      Age                  Outcome
##                     0.78                     0.60                     0.65
```

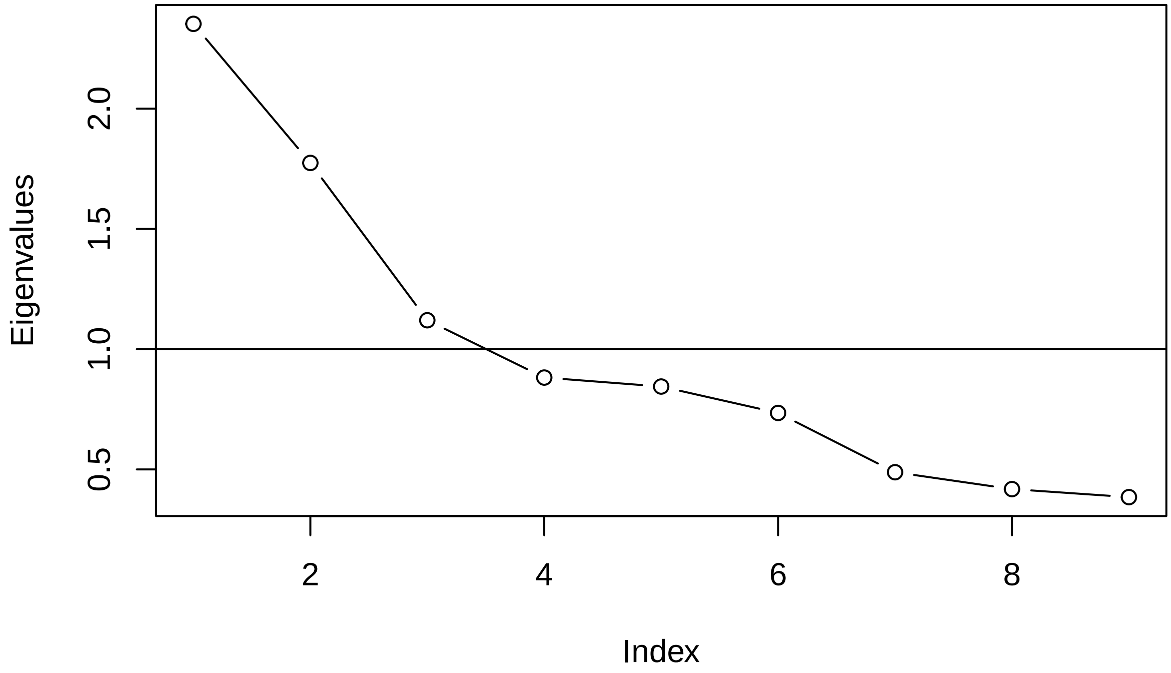
I used the Barlett's test to test for significant relationships. The large chi squared statistic showcases that there are large discrepancy in variability. Additionally, the p-value is very small which goes to show that there is strong evidence against the null hypothesis which is that there is no correlation between the outcome variable and the predictor variables. Since the p-value is less than 0.05, this dataset is appropriate for PCA.

Looking at the KMO results, the MSA value of 0.62 indicates that this dataset is moderately accurate for factor analysis. All of the variables have MSA values in similar ranges (which may not be ideal as they tend to lie between 0.5 and 0.6), with the DiabetesPedigreeFunction MSA value being the closest to 1.

```r
# PCA with {psych}
initial_pca <- principal(scaled_data, nfactors = ncol(scaled_data), rotate = "oblimin")
```

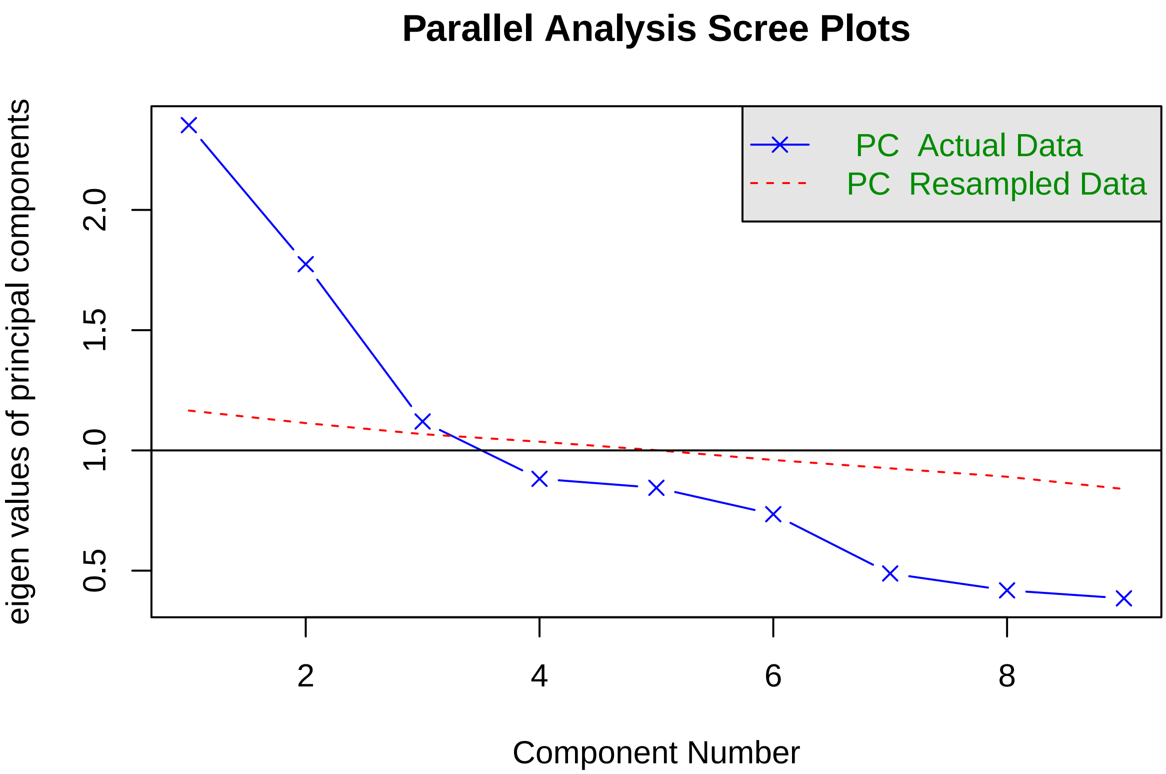
```
## Loading required namespace: GPArotation
```

```r
# Plot
plot(initial_pca$values, type = "b", ylab = "Eigenvalues"); abline(h = 1);
```



> Looking at this plot, the ideal number of components would be 3 components.

```
# PCA with {psych}
parallel_pca <- fa.parallel(
x = scaled_data, fa = "pc",
sim = FALSE # ensures resampling
)
```



```
## Parallel analysis suggests that the number of factors =  NA  and the number of components =  3
```

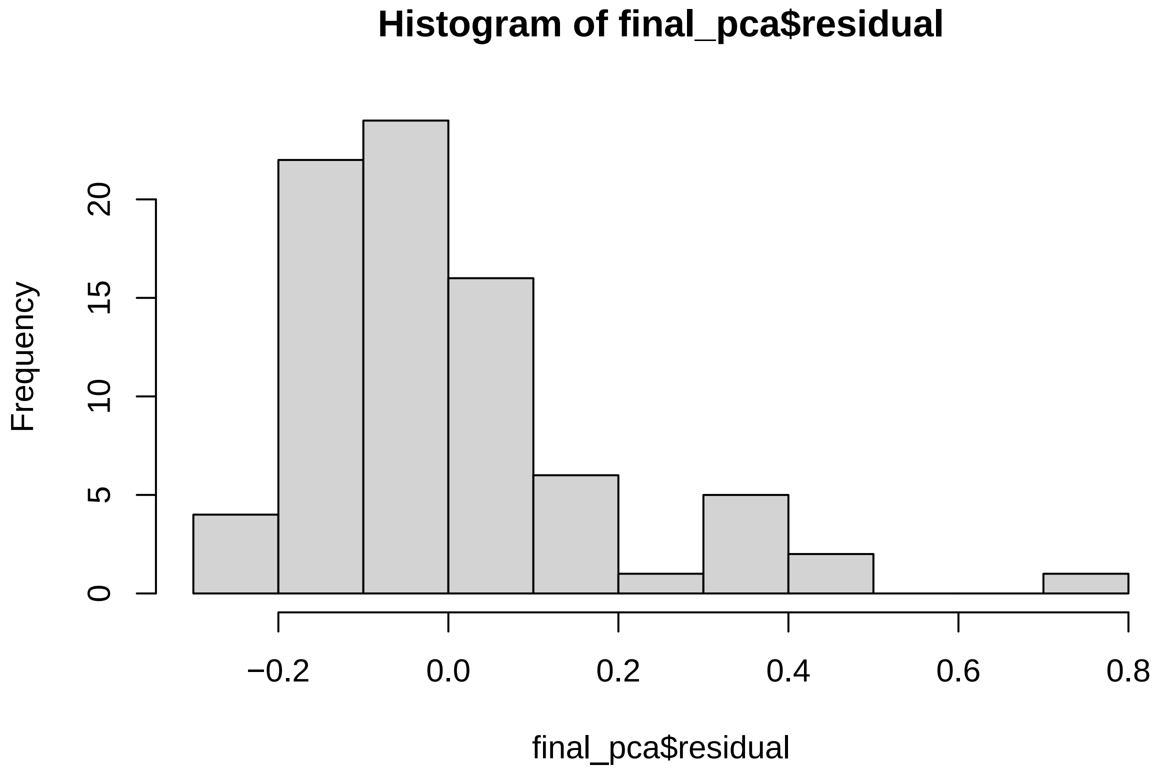
This plot also showcases that the ideal number of components is around 3. However, it is interesting to note that the resampled data line showcases a component number that is around 5.

```
# PCA with {psych}
final_pca <- principal(
r = scaled_data, nfactors = 3,
rotate = "oblimin", # Correlated dimensions
residuals = TRUE # Obtain residuals
)

# Perform Shapiro-Wilk Test
shapiro.test(final_pca$residual)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  final_pca$residual
## W = 0.85449, p-value = 2.005e-07
```

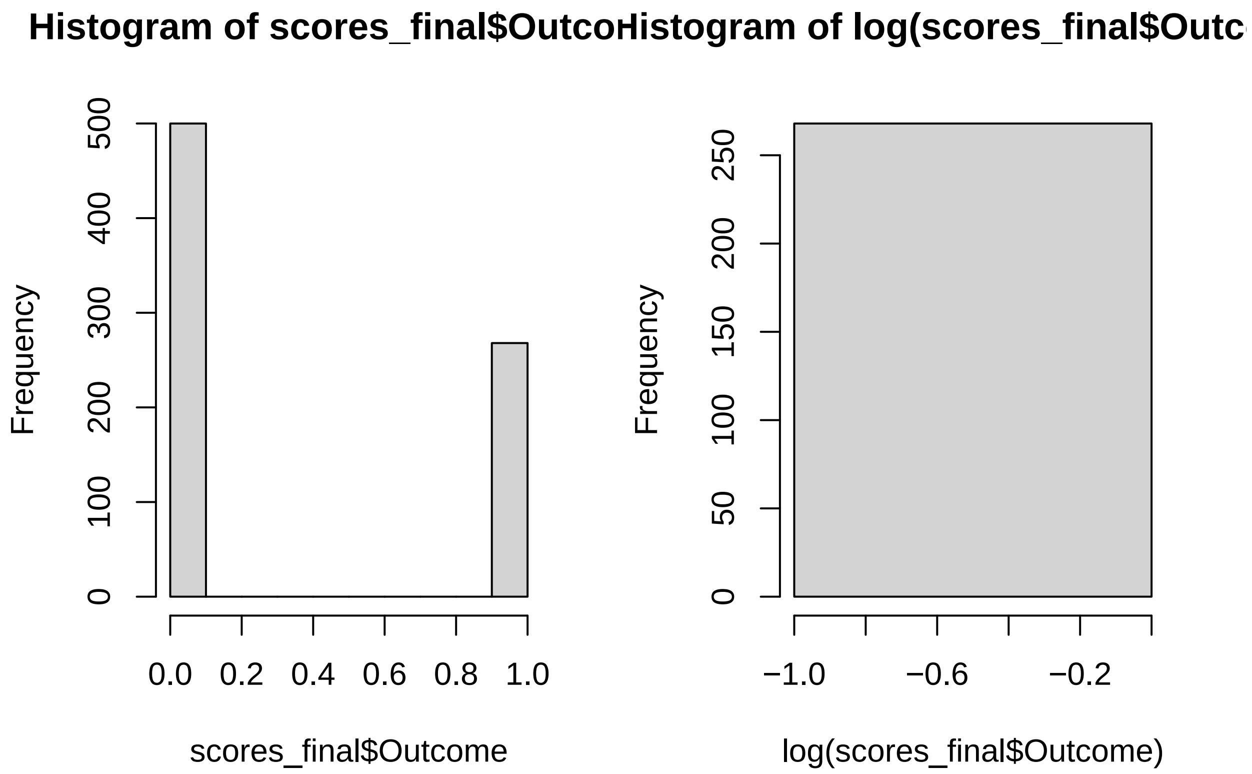
```
# Check out histogram
hist(final_pca$residual)
```



```
# Check loadings
loadings <- round(final_pca$loadings[,1:3], 3)
# For interpretation, set less than 0.30 to ""
loadings[loadings < 0.30] <- ""
# Print loadings
View(as.data.frame(loadings))
```

Looking at the Scores, we can see that the variables have been sorted into the 3 components that we requested. Pregnancy, blood pressure, and age are in TC2. Glucose, insulin, the diabetes pedigree function, and the diabetes outcome are in TC1. Finally, blood pressure, skin thickness, insulin, and BMI are in TC3.

```
pca_scores <- final_pca$scores

colnames(pca_scores) <- c(
"TC2", "TC1", "TC3"
)

pca_scores <- as.data.frame(pca_scores)
pca_scores$Outcome <- outcome
scores_final <- na.omit(pca_scores)

layout(matrix(1:2, nrow = 1)); hist(scores_final$Outcome); hist(log(scores_final$Outcome))
```



```
# Set seed
set.seed(1234)
# Set up training and testing
train_index <- sample(
  1:nrow(scores_final),
  round(nrow(scores_final) * 0.70)
)
test_index <- setdiff(
  1:nrow(scores_final),
  train_index
)
# Perform logistic regression
data_lrm <- lrm(
  Outcome ~ .,
  data = scores_final[train_index, -8]
)
```

```
data_lrm
```

```
## Logistic Regression Model
##
## lrm(formula = Outcome ~ ., data = scores_final[train_index, -8])
##
##                        Model Likelihood      Discrimination    Rank Discrim.
##                              Ratio Test             Indexes          Indexes
## Obs           538    LR chi2     443.61      R2       0.783    C       0.969
##  0            362    d.f.             3      R2(3,538)0.559    Dxy     0.939
##  1            176    Pr(> chi2) <0.0001    R2(3,355.3)0.711    gamma   0.939
## max |deriv| 4e-12                            Brier    0.061    tau-a   0.414
##
##           Coef    S.E.   Wald Z Pr(>|Z|)
## Intercept -1.5303 0.1996 -7.67  <0.0001
## TC2        0.9453 0.1776  5.32  <0.0001
## TC1        3.5833 0.3324 10.78  <0.0001
## TC3       -0.1003 0.1633 -0.61  0.5389
```

```
exp(data_lrm$coefficients)
```

```
##  Intercept        TC2        TC1        TC3
##  0.2164621  2.5734650 35.9918256  0.9045454
```

```
# Regular logistic
data_logm <- glm(
  Outcome ~ .,
  data = scores_final[train_index,-8],
  family = "binomial"
)
# Get classes
predicted <- factor(
  ifelse(predict(data_logm) > 0.50, 1, 0)
)
# Get test classes while were at it
test_predicted <- factor(
  ifelse(predict(
    data_logm,
    newdata = scores_final[test_index,]
) > 0.50, 1, 0)
)
```

```
# Confusion matrix
confusionMatrix(data = predicted, positive = "1",
reference = factor(scores_final$Outcome[train_index]))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 349  40
##          1  13 136
```

```
##
##                Accuracy : 0.9015
##                  95% CI : (0.8731, 0.9253)
##     No Information Rate : 0.6729
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.767
##
##  Mcnemar’s Test P-Value : 0.0003551
##
##             Sensitivity : 0.7727
##             Specificity : 0.9641
##          Pos Pred Value : 0.9128
##          Neg Pred Value : 0.8972
##              Prevalence : 0.3271
##          Detection Rate : 0.2528
##    Detection Prevalence : 0.2770
##       Balanced Accuracy : 0.8684
##
##        ’Positive’ Class : 1
##
```

```
# Confusion matrix
confusionMatrix(data = test_predicted, positive = "1",
reference = factor(scores_final$Outcome[test_index]))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 130  19
##          1   8  73
##
##                Accuracy : 0.8826
##                  95% CI : (0.8338, 0.9212)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.7505
##
##  Mcnemar’s Test P-Value : 0.05429
##
##             Sensitivity : 0.7935
##             Specificity : 0.9420
##          Pos Pred Value : 0.9012
##          Neg Pred Value : 0.8725
##              Prevalence : 0.4000
##          Detection Rate : 0.3174
##    Detection Prevalence : 0.3522
##       Balanced Accuracy : 0.8678
##
##        ’Positive’ Class : 1
##
```

Discussion

Looking at the data analysis and visualizations, it is important to note that the KMO MSA avlues did not have any variables with ideal values. However, as mentioned earlier, the p-value that we calculated using the Barlett's is incredibly low which means that we can reasonably conclude that the null hypothesis is rejected and we can accept the alternate hypothesis that there is a correlation between the predictor variables and the outcome variable of diabetes in the female Pima Indian population.

In this assignment, we used PCA to obtain 3 dimensions in our Female Pima Indian Diabetes data. Using these dimensions, we then classified each participant according to their diabetes outcome. Looking at the logistic regression model, the TC1 dimension had the highest predicatbility with ach standard deviation being associated with a nearly 36 times higher odds of having diabetes. In our training data, we could accurately predict the diabetes outcome with 90.15% accuracy rate with a slightly higher Kappa (0.77). In our testing data, we could accurately predict with 88.26% accuracy rate with a similar Kappa to the training data (0.75).

All in all, this goes to show that the predictor variables in this dataset can be used to predict the diabetes outcome of women in the Pima Indian population.