Entrega1

José Ignacio Díez Ruiz – 100487766
Carlos Roldán Piñero – 100484904
Pablo Vidal Fernández – 100483812

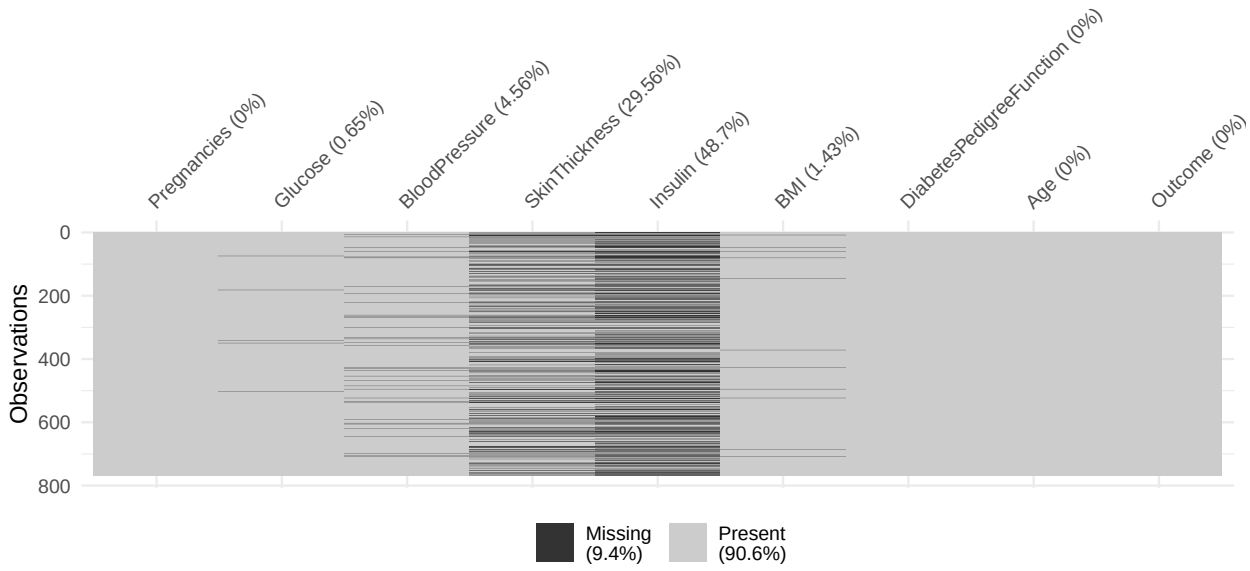
2022-12-16

Step 1

Perform a graphical analysis of the data set and try to obtain interesting conclusions from the analysis. Take into account the qualitative variable of interest to see which variables are the most informative to distinguish the groups formed by such variable.

The first thing that we do in our dataset is change the 0's in the variables that are not Pregnancies or Outcome to NA's.

```
require(tidyverse)
require(GGally)
require(visdat)

data <- read.csv("diabetes.csv")

data[data==0]<-NA

data$Pregnancies[is.na(data$Pregnancies)] <- 0
data$Outcome[is.na(data$Outcome)] <- 0
vis_miss(data)
```



The variable Insulin has nearly a 50% of NA's. We cannot impute a variable with that many NA's, so our first thought would be to drop the variable. However, if we did that, we would be left with less than 8 numerical variables, disobeying the guidelines of the project. We opt to remove all the rows with NA in that variable, and, as a consequence, we are left with very few NA's.

```
data.clean <- data %>% filter(!is.na(Insulin))
sum(is.na(data.clean))
```

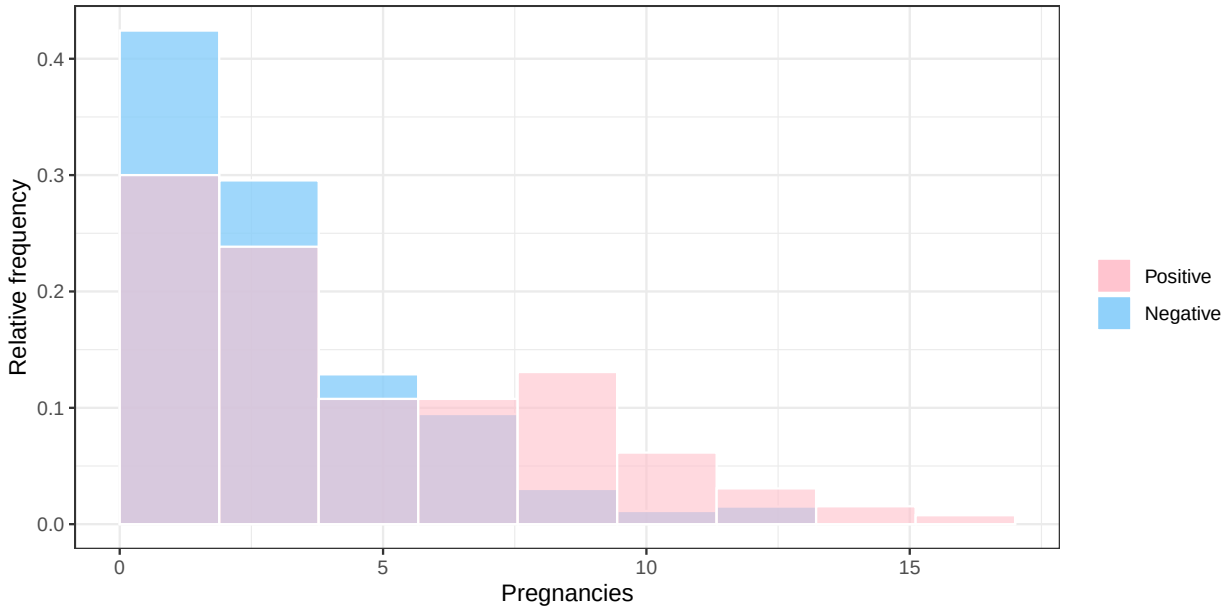
```
## [1] 2
```

We opt to remove all the rows with NA in that variable, and, as a consequence, we are left with very few NA's.

```
data0 <- data.clean[data.clean$Outcome == 0,]
data1 <- data.clean[data.clean$Outcome == 1,]
data.clean$Outcome <- factor(data.clean$Outcome, c(0,1), c("Negative", "Positive"))

histogram_by_groups <- function(data0, data1, var, label = NULL){
  if(is.null(label)){
    label <- var
  }
  ggplot(data0, aes(x = eval(parse(text = var)))) + geom_histogram(aes(
    y = after_stat(count / sum(count)), fill = "Negative"), bins = 10,
    colour = "white", alpha = 0.8, boundary = 0) +
    geom_histogram(data = data1, aes(x = eval(parse(text = var)), y = after_stat(
      count / sum(count)), fill = "Positive"), bins = 10, colour = "white",
      alpha = 0.6, boundary = 0, inherit.aes = F) +
        theme_bw() + scale_fill_manual(name = "", breaks =
                                         c("Positive", "Negative"),
                                       values =
                                         c("Positive" = "pink",
                                           "Negative" = "lightskyblue")) +
        xlab(label) + ylab("Relative frequency")
}
```

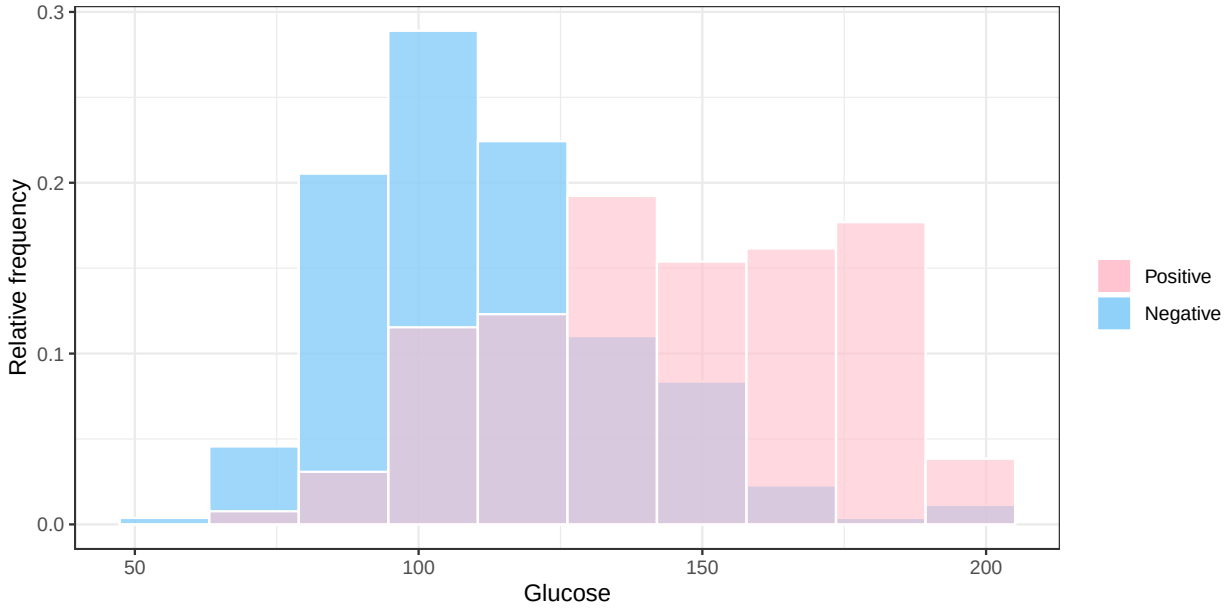
Let's inspect the relative histogram of the numerical variables, splitting by the categorical variable:

```
histogram_by_groups(data0, data1, "Pregnancies")
```



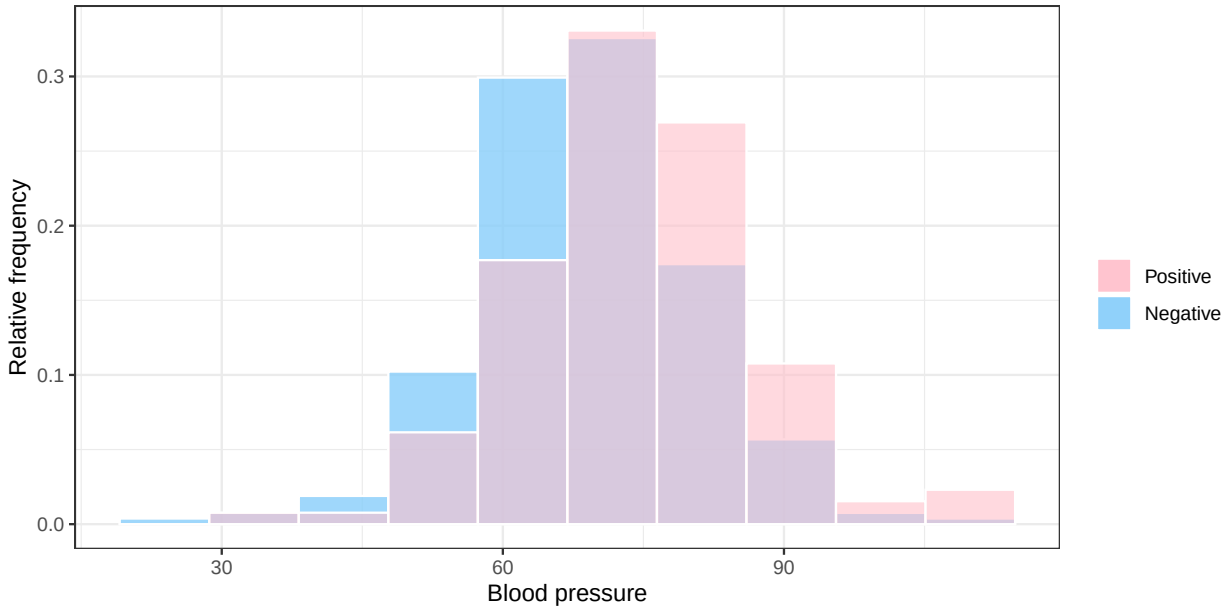
We can see that people who have diabetes have had more pregnancies than those who don't have diabetes.

```
histogram_by_groups(data0, data1, "Glucose")
```



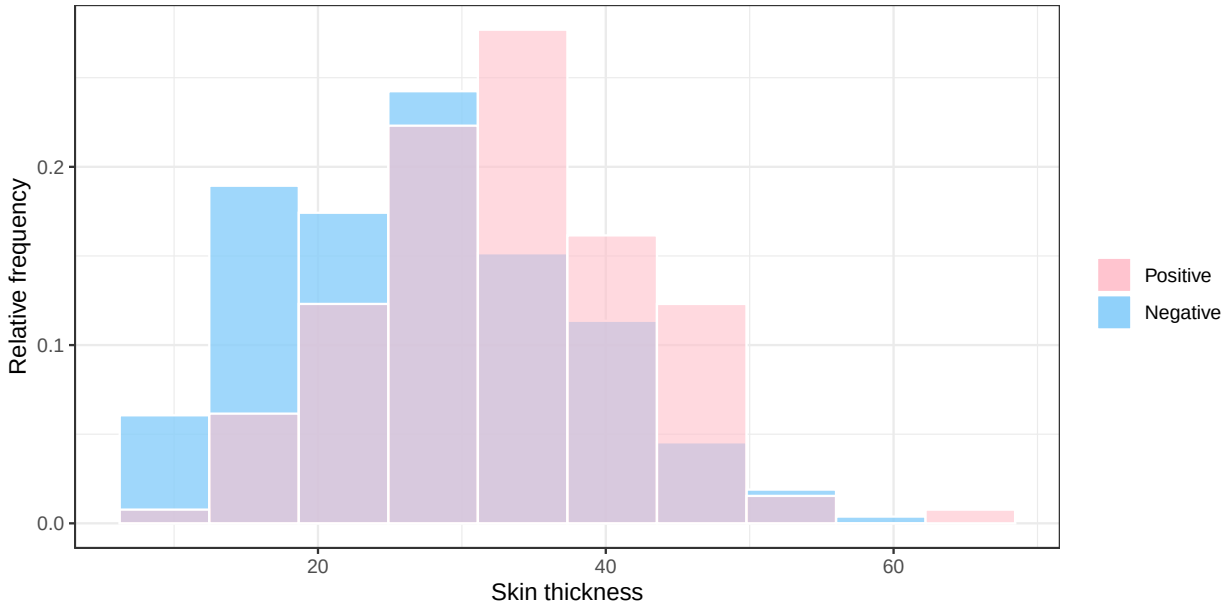
We can see that people who have diabetes have higher levels of glucose.

```
histogram_by_groups(data0, data1, "BloodPressure", "Blood pressure")
```



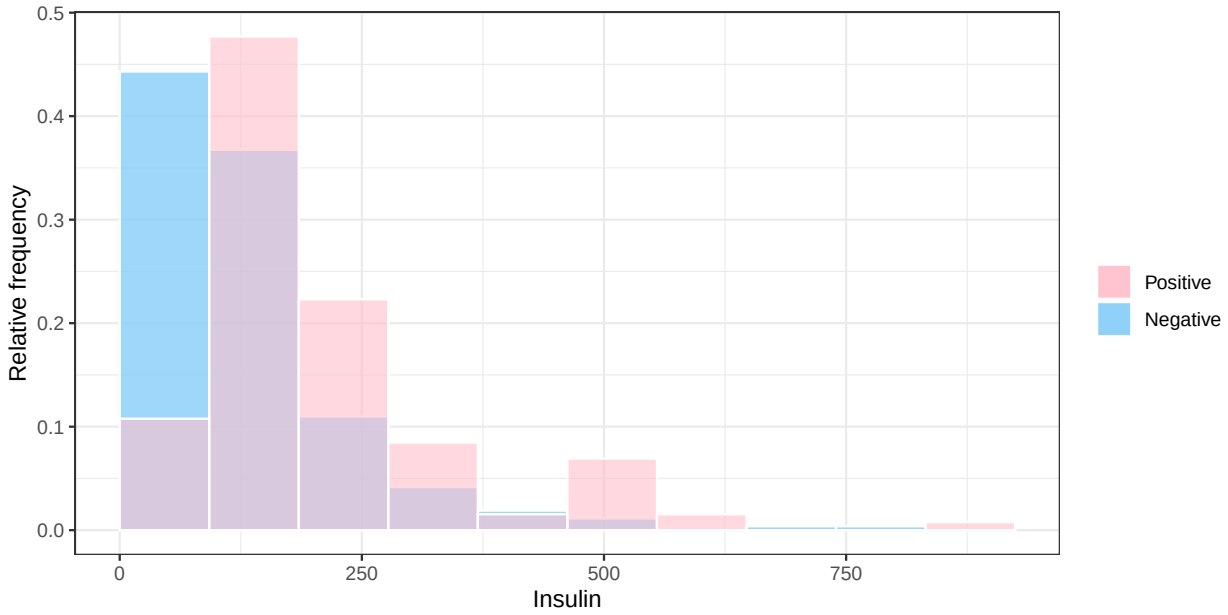
It seems that blood pressure might be a bit higher for those who had diabetes.

```
histogram_by_groups(data0, data1, "SkinThickness", "Skin thickness")
```



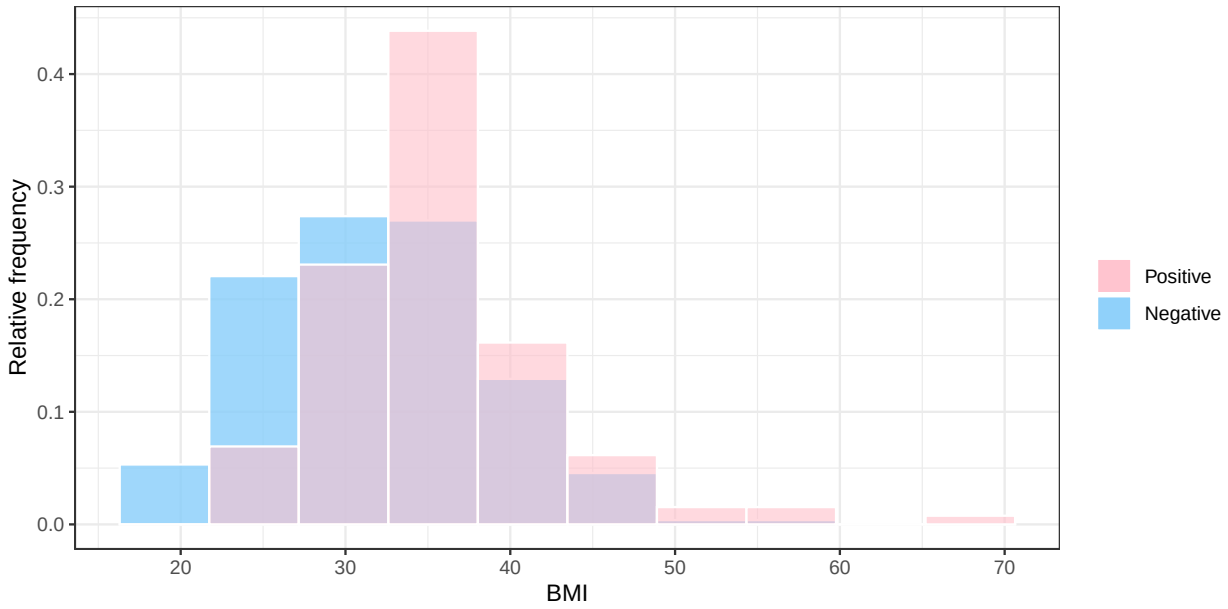
People who have diabetes then to have higher skin thickness.

```
histogram_by_groups(data0, data1, "Insulin")
```



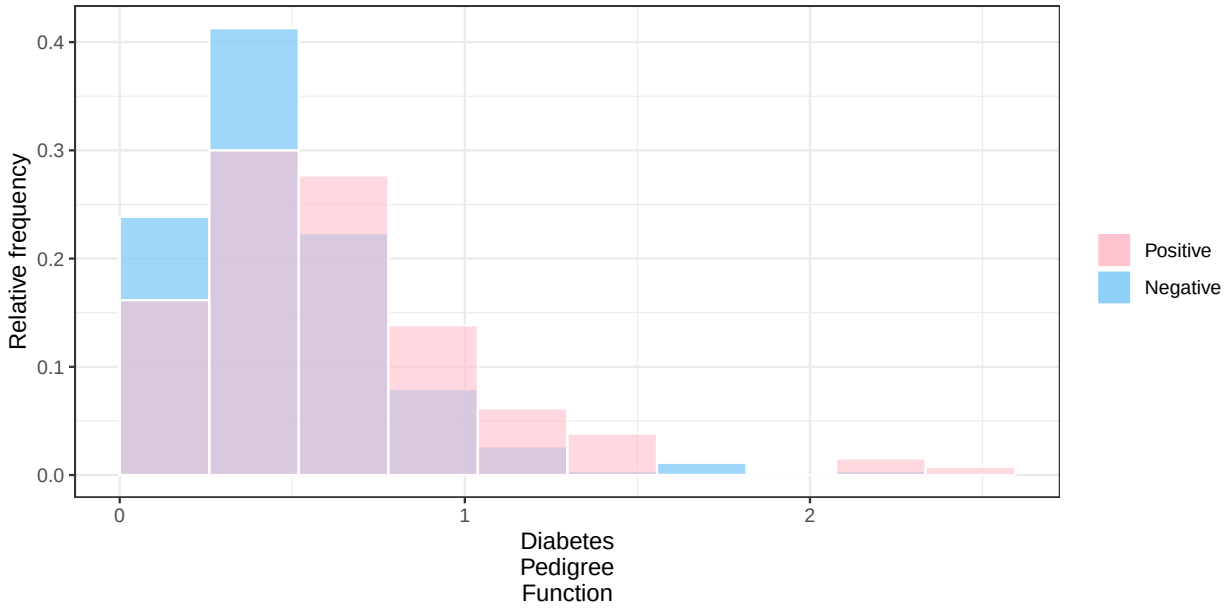
Toca arreglar lo de los 0s.

```
histogram_by_groups(data0, data1, "BMI")
```



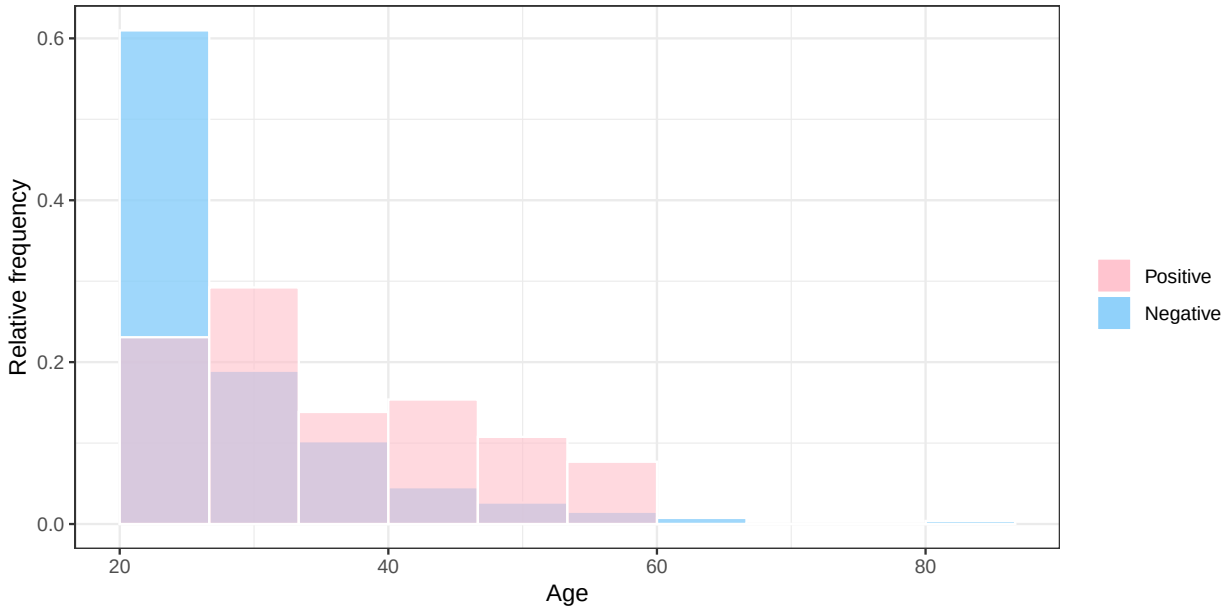
People with diabetes tend to have higher BMI.

```
histogram_by_groups(data0, data1, "DiabetesPedigreeFunction",
                    "Diabetes\nPedigree\nFunction")
```



It seems that people with diabetes might have higher diabetes pedigree function.

```
histogram_by_groups(data0, data1, "Age")
```

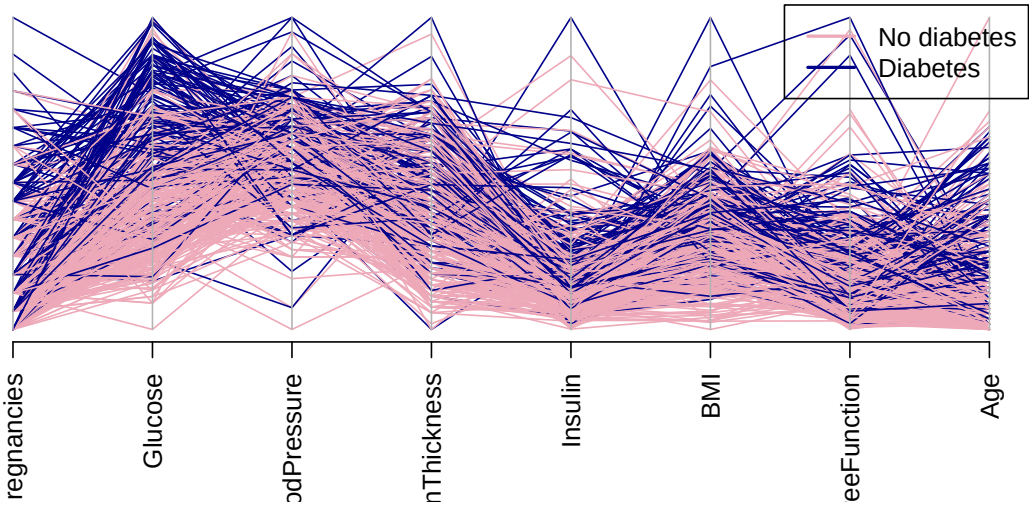


It seems that there are more young people who do not have diabetes.

Now, let's take a look at some multivariate plots. We'll begin by inspecting the Parallel Coordinate Plot:

```
require(MASS)

colors <- c("pink2", "darkblue")
col1 <- colors[1]
col2 <- colors[2]
vec_col <- as.character(data.clean$Outcome)
vec_col[vec_col=="Negative"] <- col1 # esta línea y la siguiente no van
```

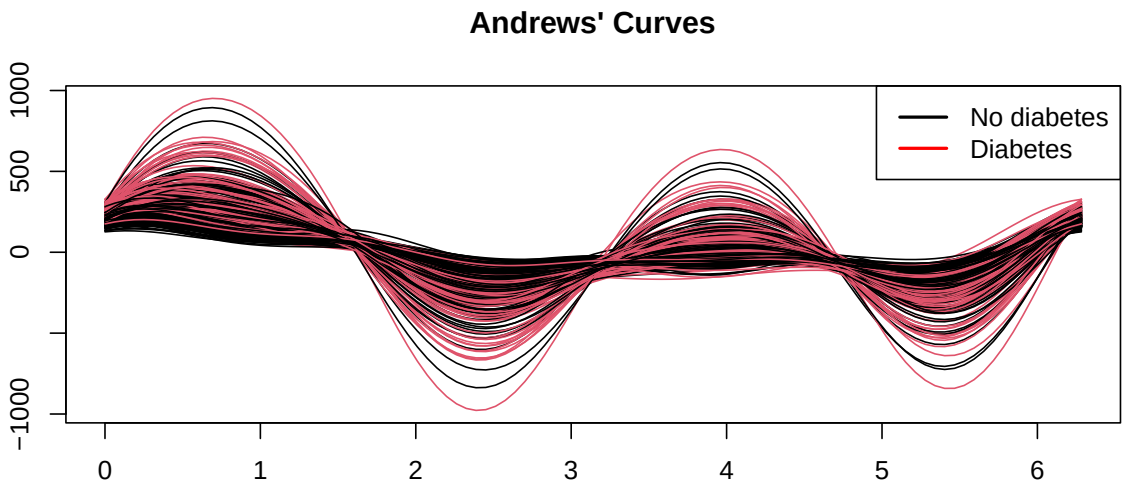
```
vec_col[vec_col=="Positive"] <- col2

par(las=2)
parcoord(data.clean[,-9], col = vec_col)

legend("topright", legend = c("No diabetes", "Diabetes"),
       col = colors, lty = 1, lwd = 2)
```



It seems that, overall, the blue lines are over the pink lines. This is most notable on the Glucose and BMI variables.

The Andrew's plot is the following:

```
require(pracma)

andrewsplot(as.matrix(na.omit(data.clean[,-9])), na.omit(data.clean)[,9],
            style = "cart")
legend("topright", legend = c("No diabetes", "Diabetes"),
       col = c("black", "red"), lty = 1, lwd = 2)
```



Again, we see that the two groups are different. The group of people who have diabetes tend to have more volatile curves.

Step 2

Estimate the main characteristics of the quantitative variables (mean vector, covariance matrix, correlation matrix) with all the observations in the data set as well as in each of the groups with the most appropriate method. Give conclusions from the analysis.

As we have n»p, we can estimate those characteristics with the sample mean, sample covariance and sample correlation matrix.

For the overall data, we have:

```
numerical_data <- data.clean[,-9]
sapply(numerical_data, mean, na.rm = T)
```

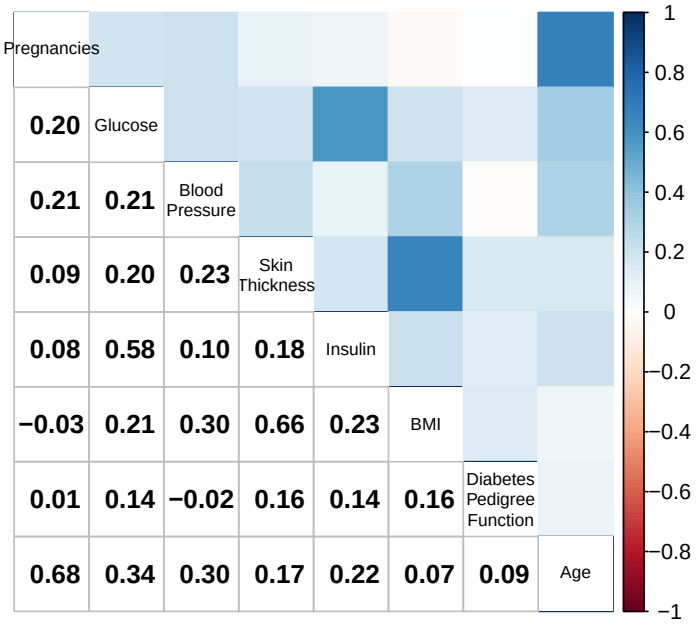
```
##              Pregnancies                  Glucose            BloodPressure
##                3.2868020              122.6157761                70.6548223
##            SkinThickness                  Insulin                      BMI
##                29.1065990              155.5482234                33.0725191
## DiabetesPedigreeFunction                      Age
##                0.5255431               30.8147208
```

```
cov(numerical_data, use = "complete.obs")
```

```
##                           Pregnancies     Glucose BloodPressure SkinThickness
## Pregnancies              10.313247038   19.652043    8.56198131     3.1479331
## Glucose                  19.652043426  952.387781   80.99446735    64.5376716
## BloodPressure             8.561981314   80.994467  156.15230440    30.5631557
## SkinThickness             3.147933086   64.537672   30.56315570   110.5951707
## Insulin                  30.144188110 2131.662900  146.29516154   227.7104885
## BMI                      -0.572057519   45.439613   26.73217809    49.0997064
```

```
## DiabetesPedigreeFunction  0.008390234    1.494605   -0.06895125     0.5831391
## Age                      22.263309672  108.179694   38.24591576    17.9966922
##                                Insulin         BMI DiabetesPedigreeFunction
## Pregnancies                 30.144188  -0.5720575              0.008390234
## Glucose                   2131.662900  45.4396132              1.494605381
## BloodPressure              146.295162  26.7321781             -0.068951250
## SkinThickness              227.710489  49.0997064              0.583139086
## Insulin                  14123.347226 189.0815935              5.580071585
## BMI                        189.081594  49.3879939              0.385491683
## DiabetesPedigreeFunction     5.580072   0.3854917              0.119361988
## Age                        263.163618   5.0047823              0.299663513
##                                   Age
## Pregnancies                22.2633097
## Glucose                   108.1796936
## BloodPressure              38.2459158
## SkinThickness              17.9966922
## Insulin                   263.1636176
## BMI                         5.0047823
## DiabetesPedigreeFunction    0.2996635
## Age                       104.0558419
```

```
require(corrplot)

correlation <- cor(numerical_data, use = "complete.obs")
colnames(correlation) <- c("Pregnancies", "Glucose",
                           "Blood\nPressure", "Skin\nThickness",
                           "Insulin", "BMI",
                           "Diabetes\nPedigree\nFunction",
                           "Age")

corrplot.mixed(correlation, lower = "number", upper = "color",
         diag = "n", tl.col = "black", tl.cex = 0.65,
         lower.col = "black")
```



There are some variables that seem to be correlated. The positive correlation between age and pregnancies isn't surprising, but there seems to be a positive correlationship between insulin levels and skin thickness. Skin thickness and BMI also seem to have a positive relationship.

Let's take a look into the group of people who have diabetes:

```
numerical_data<-data1[,-9]
sapply(numerical_data, mean, na.rm = T)
```

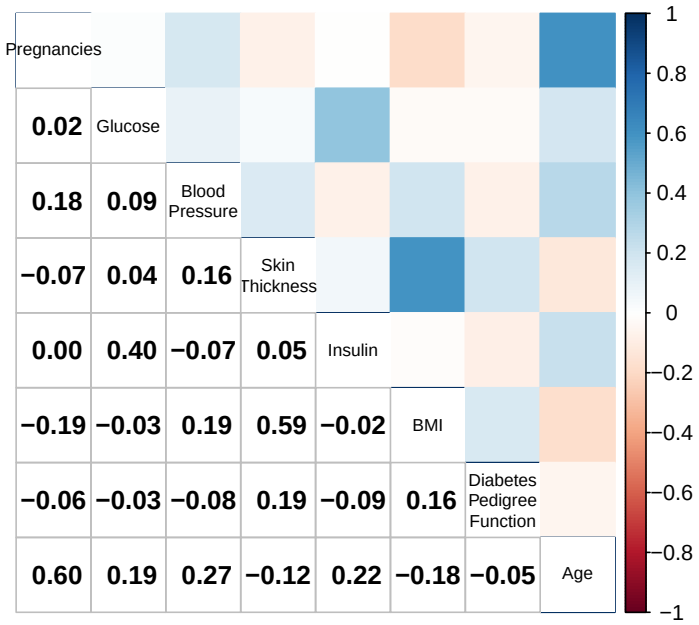
```
##               Pregnancies                   Glucose             BloodPressure
##                 4.4692308               145.1923077                74.0769231
##             SkinThickness                   Insulin                       BMI
##                32.9615385               206.8461538                35.7776923
## DiabetesPedigreeFunction                       Age
##                 0.6255846                35.9384615
```

```
cov(numerical_data, use = "complete.obs")
```

```
##                           Pregnancies      Glucose BloodPressure SkinThickness
## Pregnancies               15.33625522    2.1726297     9.0023852    -2.6717352
## Glucose                    2.17262970  890.3890877    35.9773405    10.8214073
## BloodPressure              9.00238521   35.9773405   169.5599284    19.7161598
## SkinThickness             -2.67173524   10.8214073    19.7161598    92.9830054
## Insulin                   -1.73345259 1576.1073345  -126.6082290    66.3428742
## BMI                       -4.89720334   -5.3491652    17.0203339    38.4386702
## DiabetesPedigreeFunction -0.09500513   -0.3331831    -0.4041849     0.7468444
## Age                       25.19189028   60.1127013    37.4931425   -12.5992844
##                                Insulin         BMI DiabetesPedigreeFunction
## Pregnancies                  -1.733453  -4.8972033              -0.09500513
## Glucose                    1576.107335  -5.3491652              -0.33318306
## BloodPressure              -126.608229  17.0203339              -0.40418485
## SkinThickness                66.342874  38.4386702               0.74684436
## Insulin                   17609.262970 -17.4964818              -4.79549076
## BMI                         -17.496482  45.3560101               0.44432555
## DiabetesPedigreeFunction     -4.795491   0.4443255               0.16476279
## Age                         310.664878 -12.6316160              -0.22908002
##                                Age
## Pregnancies               25.19189
## Glucose                   60.11270
## BloodPressure             37.49314
## SkinThickness            -12.59928
## Insulin                  310.66488
## BMI                      -12.63162
## DiabetesPedigreeFunction  -0.22908
## Age                      113.09696
```

```
correlation <- cor(numerical_data, use = "complete.obs")
colnames(correlation) <- c("Pregnancies", "Glucose",
                           "Blood\nPressure", "Skin\nThickness",
                           "Insulin", "BMI",
                           "Diabetes\nPedigree\nFunction",
                           "Age")
```

```
corrplot.mixed(correlation, lower = "number", upper = "color",
         diag = "n", tl.col = "black", tl.cex = 0.65,
         lower.col = "black")
```



Now, let's take a look into the group of people who don't have diabetes and compare the results:

```
numerical_data<-data0[,-9]
sapply(numerical_data, mean, na.rm = T)
```

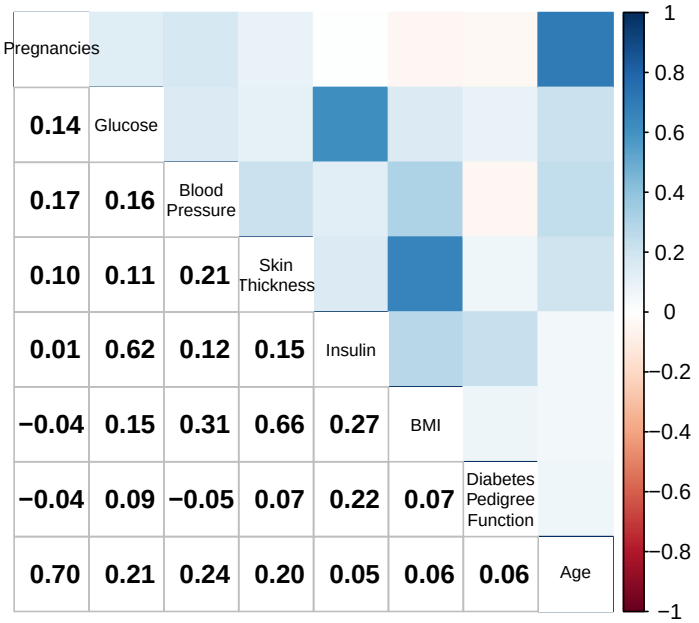
```
##              Pregnancies                  Glucose            BloodPressure
##                2.7045455              111.4562738               68.9696970
##            SkinThickness                  Insulin                      BMI
##               27.2083333              130.2878788               31.7353612
## DiabetesPedigreeFunction                      Age
##                0.4762803               28.2916667
```

```
cov(numerical_data, use = "complete.obs")
```

```
##                           Pregnancies      Glucose BloodPressure SkinThickness
## Pregnancies                 6.8531046    8.7221637     5.405253     2.7141353
## Glucose                     8.7221637  607.2347255    46.151151    27.1629669
## BloodPressure               5.4052528   46.1511509   141.439677    26.3333918
## SkinThickness               2.7141353   27.1629669    26.333392   108.8711649
## Insulin                     1.7985142 1560.3386564   152.531953   163.8987453
## BMI                        -0.7796716   25.4569492    24.787763    46.9028721
## DiabetesPedigreeFunction   -0.0297423    0.6794484    -0.164378     0.2128541
## Age                        16.4841186   47.0335322    25.857389    18.7589132
##                                Insulin         BMI DiabetesPedigreeFunction
## Pregnancies                   1.798514  -0.7796716                -0.0297423
## Glucose                    1560.338656  25.4569492                 0.6794484
## BloodPressure               152.531953  24.7877629                -0.1643780
## SkinThickness               163.898745  46.9028721                 0.2128541
```

```
## Insulin                   10532.132140 190.0357433                6.8485110
## BMI                         190.035743  46.1716278                0.1522229
## DiabetesPedigreeFunction      6.848511   0.1522229                0.0895444
## Age                          48.655933   3.5642932                0.1744434
##                                 Age
## Pregnancies               16.4841186
## Glucose                   47.0335322
## BloodPressure             25.8573894
## SkinThickness             18.7589132
## Insulin                   48.6559328
## BMI                        3.5642932
## DiabetesPedigreeFunction   0.1744434
## Age                       80.8022725
```

```
correlation <- cor(numerical_data, use = "complete.obs")
colnames(correlation) <- c("Pregnancies", "Glucose",
                           "Blood\nPressure", "Skin\nThickness",
                           "Insulin", "BMI",
                           "Diabetes\nPedigree\nFunction",
                           "Age")

corrplot.mixed(correlation, lower = "number", upper = "color",
         diag = "n", tl.col = "black", tl.cex = 0.65,
         lower.col = "black")
```

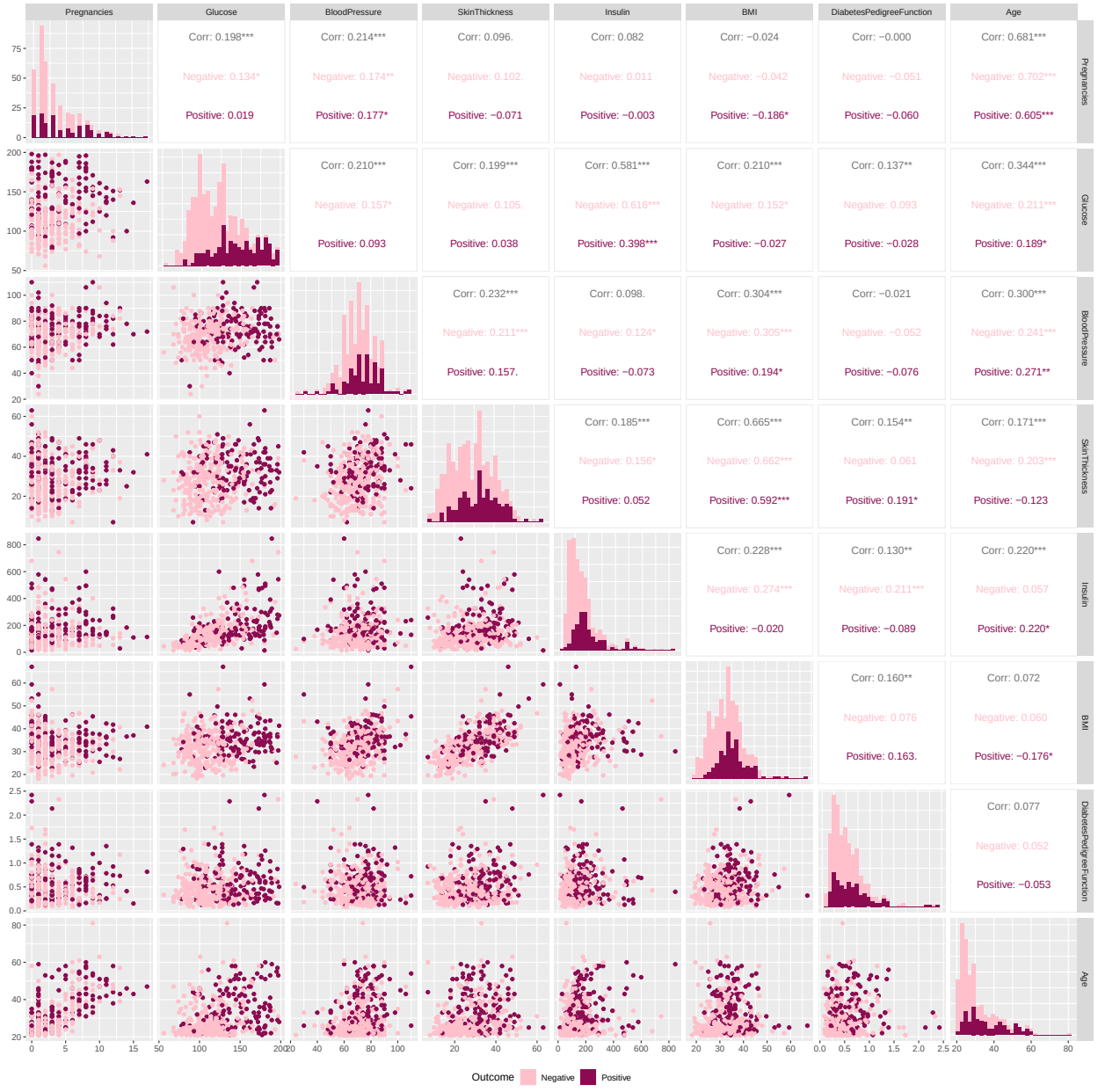


The major changes are that the correlation between skin thickness and diabetes pedigree function is lower in the group who don't have diabetes, and the correlation between BMI and age is positive (in the group of people who have diabetes, it was negative).

Taking a look at the means of the variables, we can see what the histograms already reflected: people with diabetes tend to have had more pregnancies, and glucose and insulin levels are higher.

A good summary is presented in the following plot, that gives the scatterplots and the correlations:

```
ggpairs(data.clean, aes(color = Outcome), legend = 1, columns = c(1:(length(data.clean)-1)),
        diag = list(continuous = "barDiag")  ) +
  theme(legend.position = "bottom") + scale_fill_manual(values = c("pink", "deeppink4")) +
  scale_color_manual(values = c("pink", "deeppink4")) + labs(fill = "Outcome")
```



Step 3

Try to find outliers as well as any other characteristic of interest.

Taking a look at the univariate level:

```r
findOutliers <- function(data, fields){
  outliers <- list()
  for (field in fields){
    qs  <- quantile(data[[field]], c(0.25, 0.75), na.rm = TRUE)
    iqr <- qs[2] - qs[1]
    lq  <- qs[1] - 1.5*iqr
    hq  <- qs[2] + 1.5*iqr
    outliers[[field]] <- which((data[[field]] < lq) & (data[[field]] > hq))
  }
  return (outliers)
}

outliers <- findOutliers(data.clean, names(data)[names(data) != "Outcome"])

outliers
```

```
## $Pregnancies
## integer(0)
##
## $Glucose
## integer(0)
##
## $BloodPressure
## integer(0)
##
## $SkinThickness
## integer(0)
##
## $Insulin
## integer(0)
##
## $BMI
## integer(0)
##
## $DiabetesPedigreeFunction
## integer(0)
##
## $Age
## integer(0)
```

Using the method that the boxplots use to detect outliers, there are not any outliers in the data.

Step 4

Impute missing data.

```r
sum(is.na(data.clean))
```

```
## [1] 2
```

There are only two missing values. We will impute them using the **mice** package, by predictive mean matching.

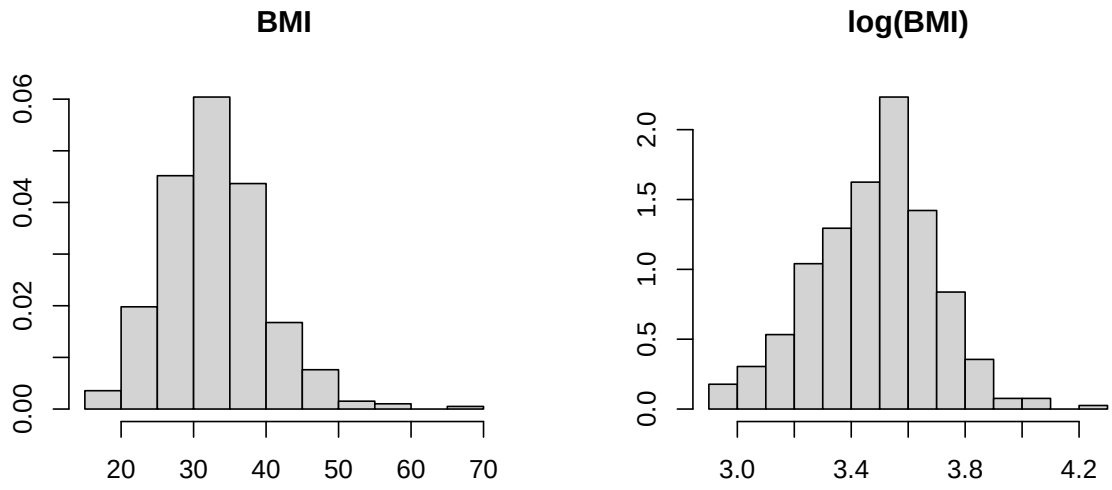
```
require(mice)
dataIm <- mice(data.clean, m = 1, method = "pmm")
data <- complete(dataIm)
```

Step 5

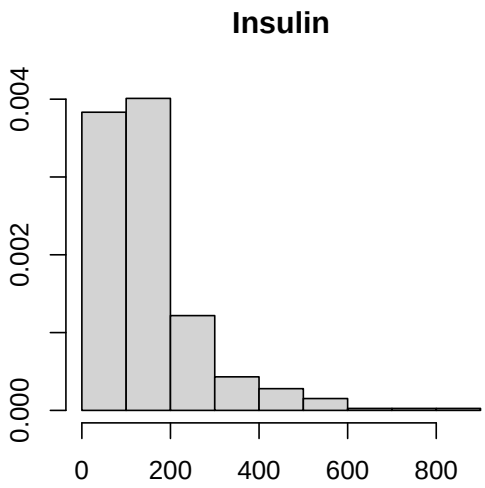
Carry out dimension reduction (principal component analysis, independent component analysis and factor analysis). Once more, obtain conclusions from the analysis.

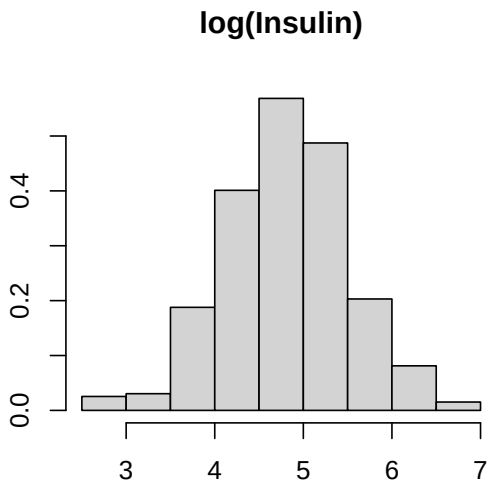
We will begin with PCA, but first, we will take the logarithm of some of the variables to make them more symmetric:

```
par(mfrow = c(1,2))
hist(data$BMI, main = "BMI", freq = F, xlab = "", ylab = "")
hist(log(data$BMI), main = "log(BMI)", freq = F, xlab = "", ylab = "")
```

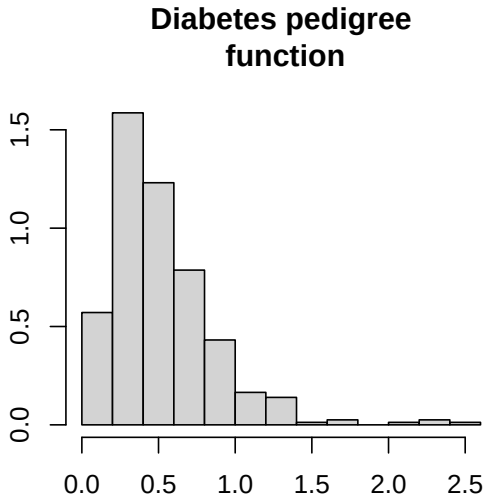


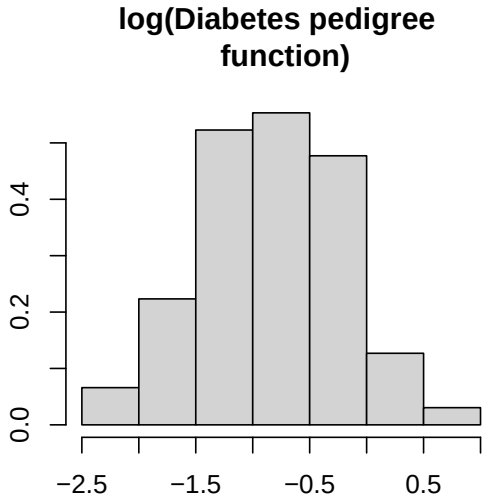

```
par(mfrow = c(1,2))
hist(data$Insulin, main = "Insulin", freq = F, xlab = "", ylab = "")
hist(log(data$Insulin), main = "log(Insulin)", freq = F, xlab = "", ylab = "")
```






```
par(mfrow = c(1,2))
hist(data$DiabetesPedigreeFunction,
     main = "Diabetes pedigree\n function", freq = F, xlab = "", ylab = "")
hist(log(data$DiabetesPedigreeFunction), main = "log(Diabetes pedigree \n function)", freq = F, xlab =
```

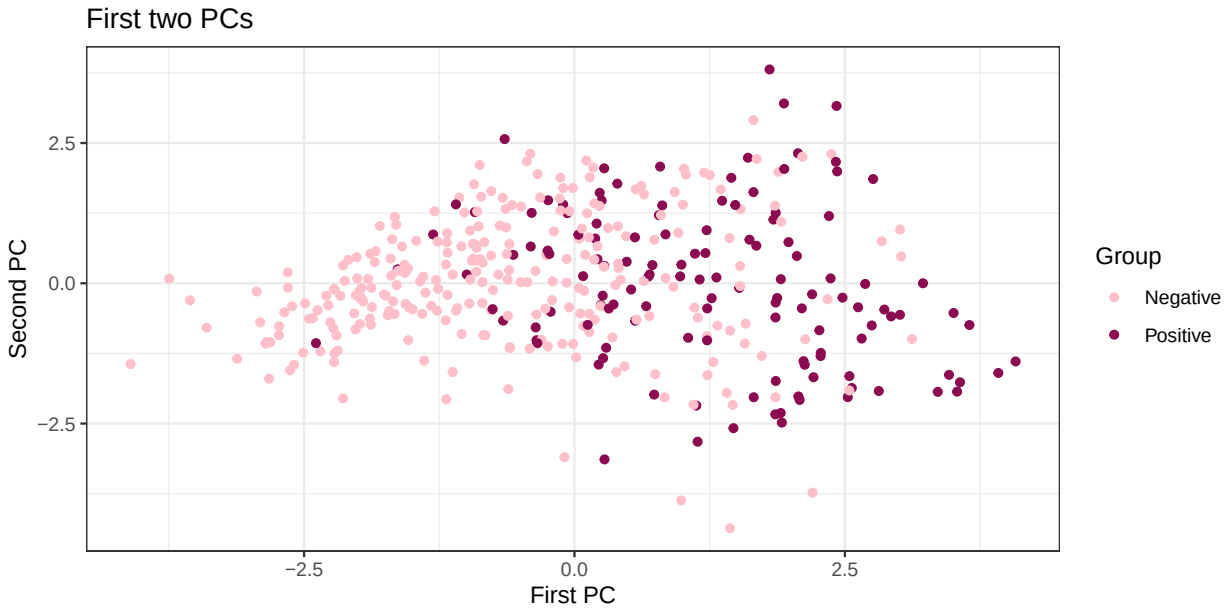





```
data$BMI <- log(data$BMI)
data$Insulin <- log(data$Insulin)
data$DiabetesPedigreeFunction <- log(data$DiabetesPedigreeFunction)

colnames(data)[5:7] <- paste0("log_", colnames(data)[5:7])
```

We must scale the data before performing the PCA.

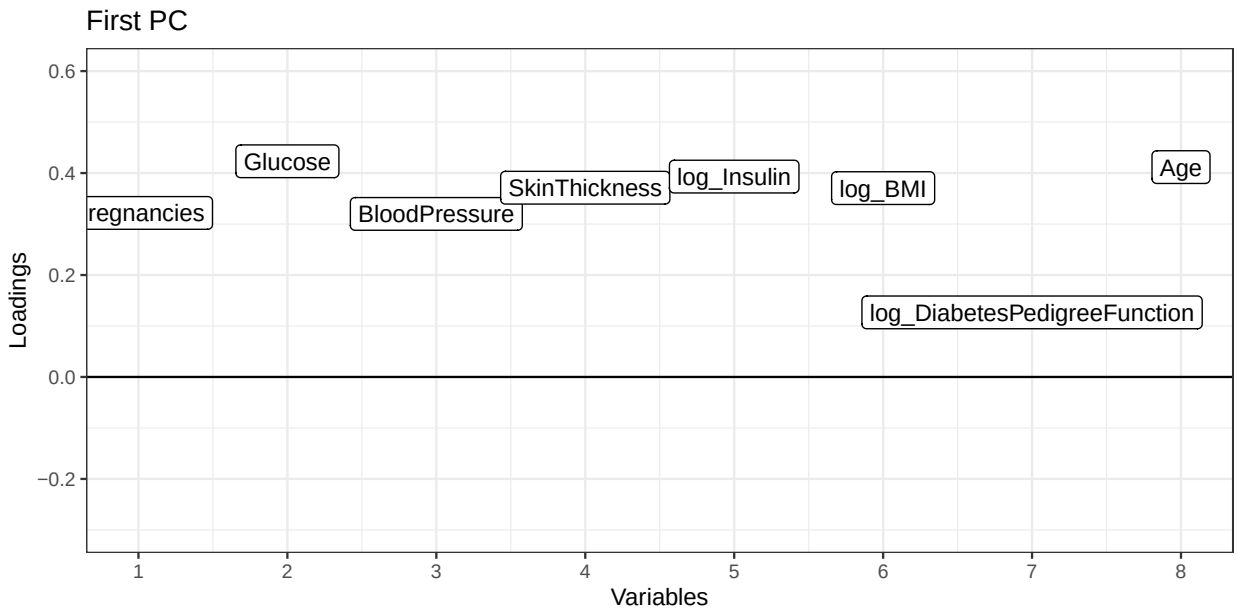
```r
data_pcs <- prcomp(data[,-9], scale = TRUE)

colours <- c("pink", "deeppink4")
vec_col <- ifelse(data$Outcome == "Negative", colours[1],
                  colours[2])

df <- as.data.frame(data_pcs$x[,1:2])
df$group <- data$Outcome
colnames(df)<-c("x", "y", "group")

ggplot(df, aes(x = x, y = y, col = group)) + geom_point() +
  xlab("First PC") + ylab("Second PC") + ggtitle("First two PCs") +
  scale_colour_manual(values = c("pink", "deeppink4")) + theme_bw() + labs(col = "Group")
```



Using the first two principal components, we can see that the first PC separates reasonably well the two groups.

We can see the loadings of the variables in each PC. For example, for the first one:

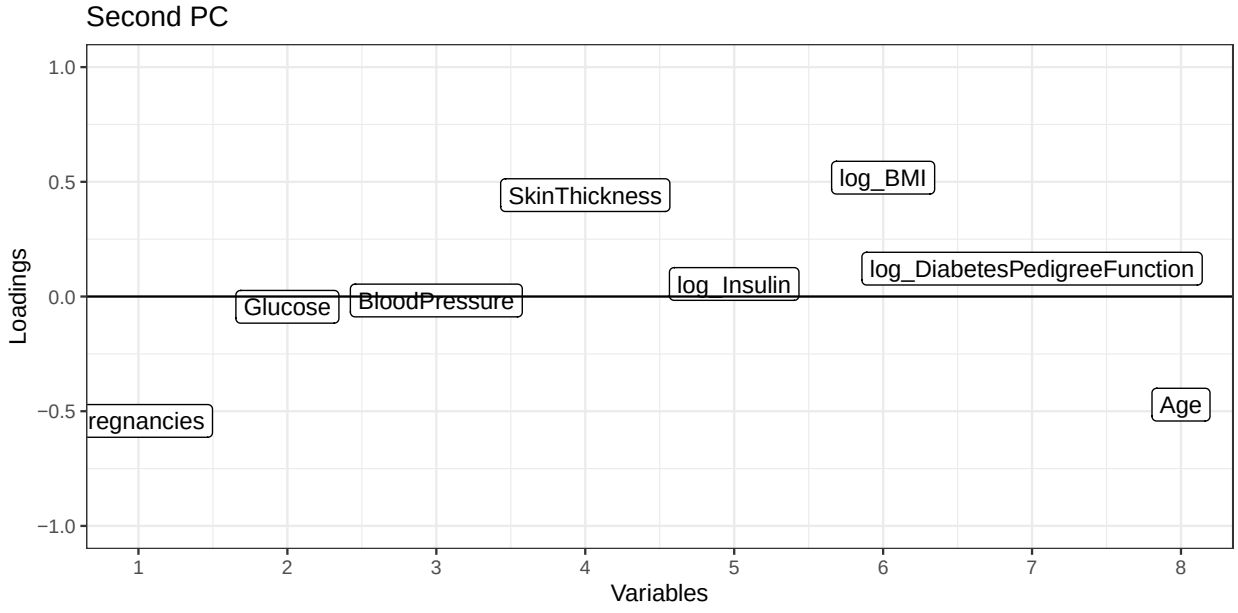
```r
p <- ncol(data[,-9])
df2 <- data.frame(x = 1:p, y = data_pcs$rotation[,1])

ggplot(df2, aes(x = x, y = y)) + geom_point() +
  geom_label(label = colnames(data[,-9]), label.size = 0.3) + xlab("Variables") +
  ylab("Loadings") +
  ggtitle("First PC") +
  theme_bw() + xlim(0, 10) +
  ylim(-0.3, 0.6) + scale_x_continuous(breaks = c(0:10)) +
  geom_hline(yintercept = 0)
```



All the variables contribute positively to the first PC.

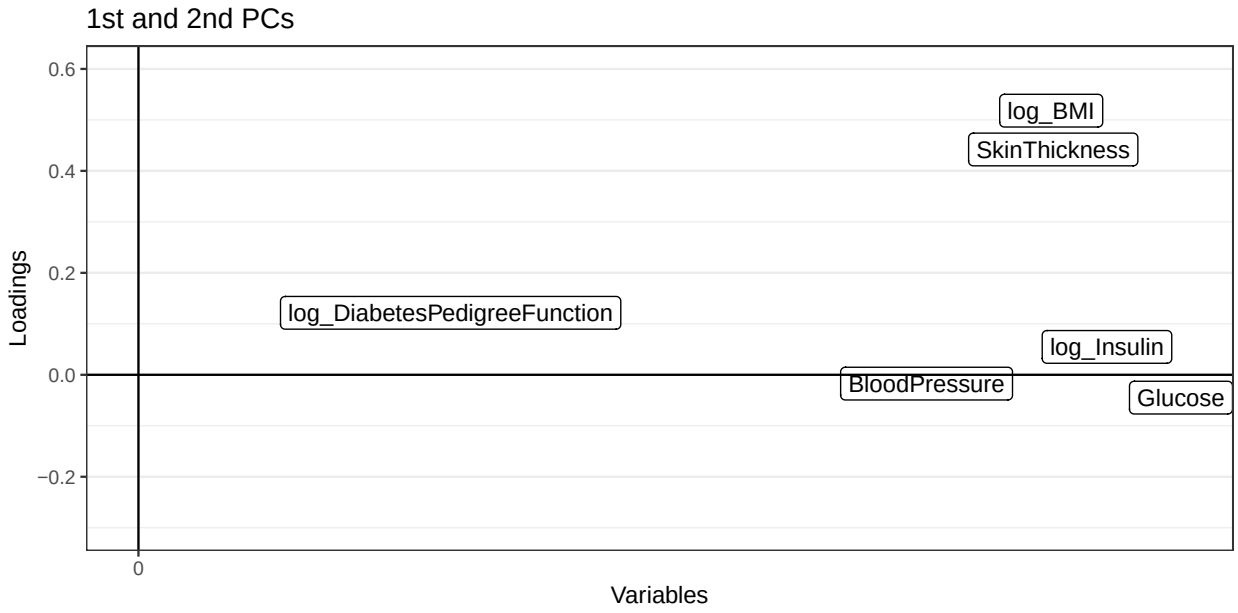
The second PC:

```
df3 <- data.frame(x = 1:p, y = data_pcs$rotation[,2])

ggplot(df3, aes(x = x, y = y)) + geom_point() +
  geom_label(label = colnames(data[,-9]), label.size = 0.3) + xlab("Variables") +
  ylab("Loadings") +
  theme_bw() + xlim(c(-1, 10)) +
  ggtitle("Second PC") +
  ylim(-1, 1) + scale_x_continuous(breaks = c((-1):10)) +
  geom_hline(yintercept = 0)
```



Pregnancies and Age have a negative loading, while log_BMI and skin thickness have a positive loading.

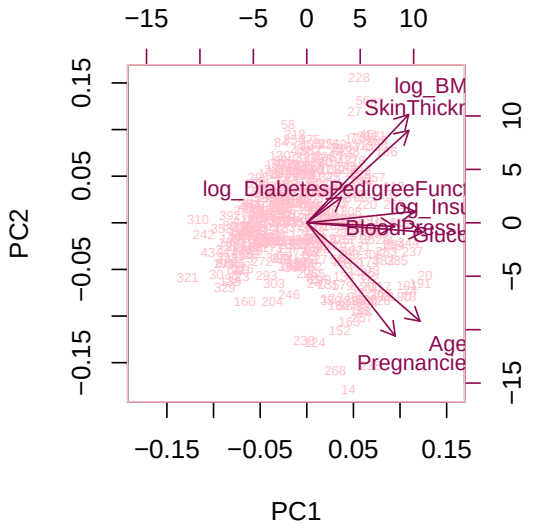
```
df4 <- data.frame(x = data_pcs$rotation[,1], y = data_pcs$rotation[,2])

ggplot(df4, aes(x = x, y = y)) + geom_point() +
  geom_label(label = colnames(data[,-9]), label.size = 0.3) + xlab("Variables") +
  ylab("Loadings") +
  theme_bw() + xlim(0, 10) +
  ggtitle("1st and 2nd PCs") +
  ylim(-0.3, 0.6) + scale_x_continuous(breaks = c(0:10)) +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0)
```



We can see in this plot the information conveyed by the two last plots.

Graphing the biplot:

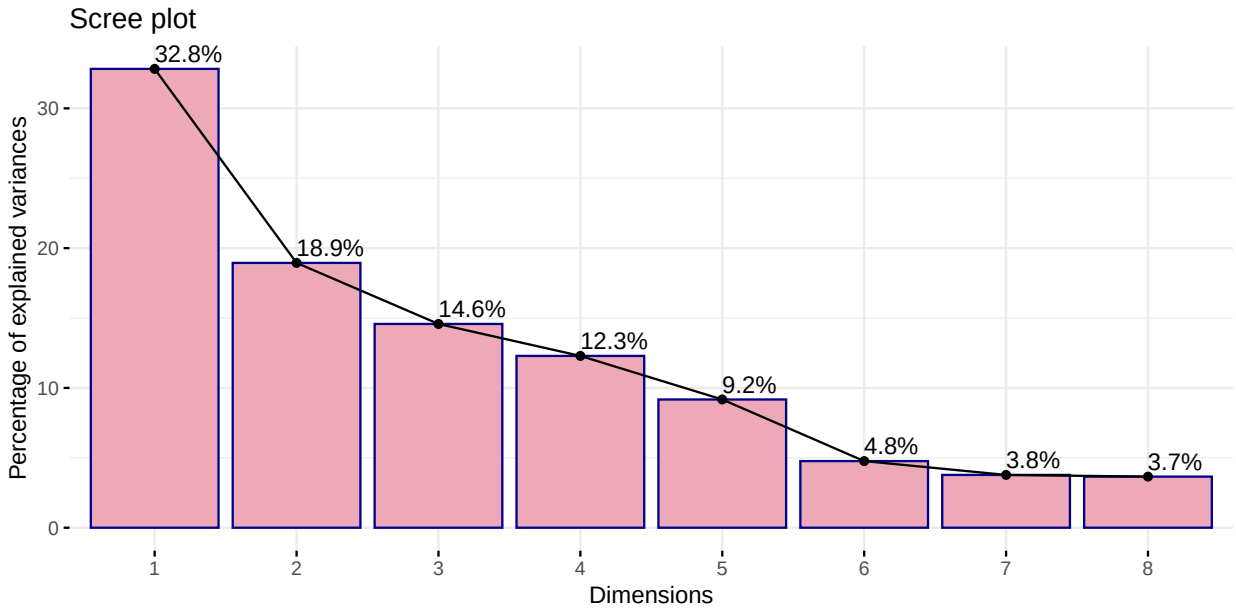
```
biplot(data_pcs, col = vec_col, cex = c(0.5,0.8))
```



log_BMI, log_DiabetesPedigreeFunction and skin thickness seem to be uncorrelated with age and pregnancies.

Looking into the variance retained by the PCs:

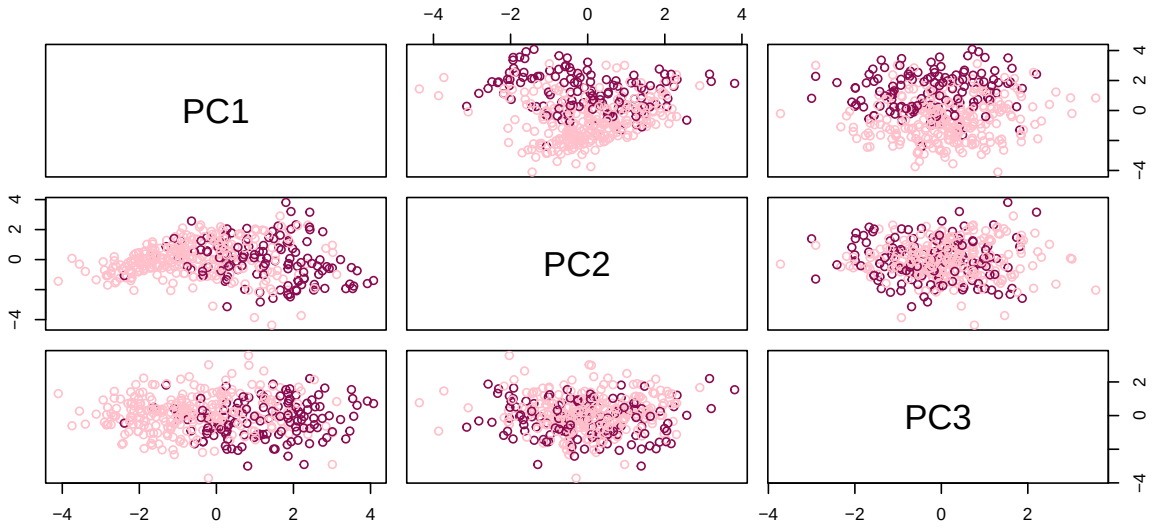
```
library(factoextra)
fviz_eig(data_pcs, ncp = 17, addlabels = T, barfill = col1, barcolor = col2)
```



We can see that the first three dimensions retain nearly 2/3 of the total variance. Using the plot, we believe that 3 PCs should be retained.

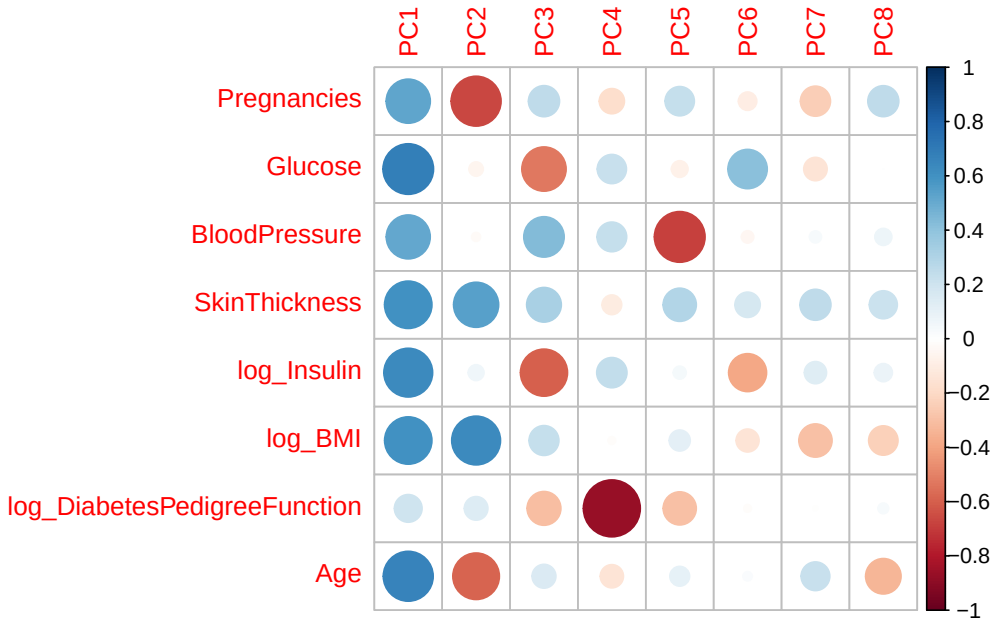
Let's see the scatterplots:

```
pairs(data_pcs$x[,1:3], col = vec_col)
```

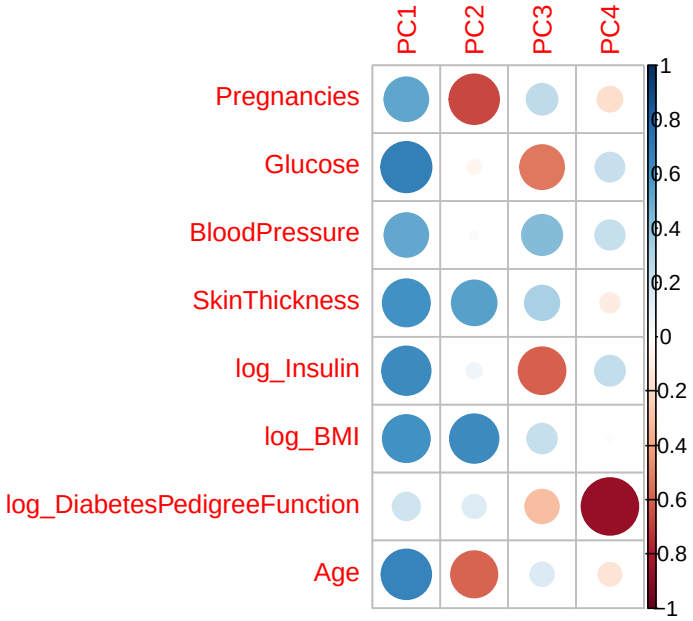


corrplot(cor(X,X_pcs$x$), $is.corr = T$)$corrplot(cor(X, X_pcs$x[,1:4]),$is.corr=T)

```
corrplot(cor(data[,-9], data_pcs$x), is.corr = T)
```




```
corrplot(cor(data[,-9], data_pcs$x[,1:4]), is.corr = T)
```



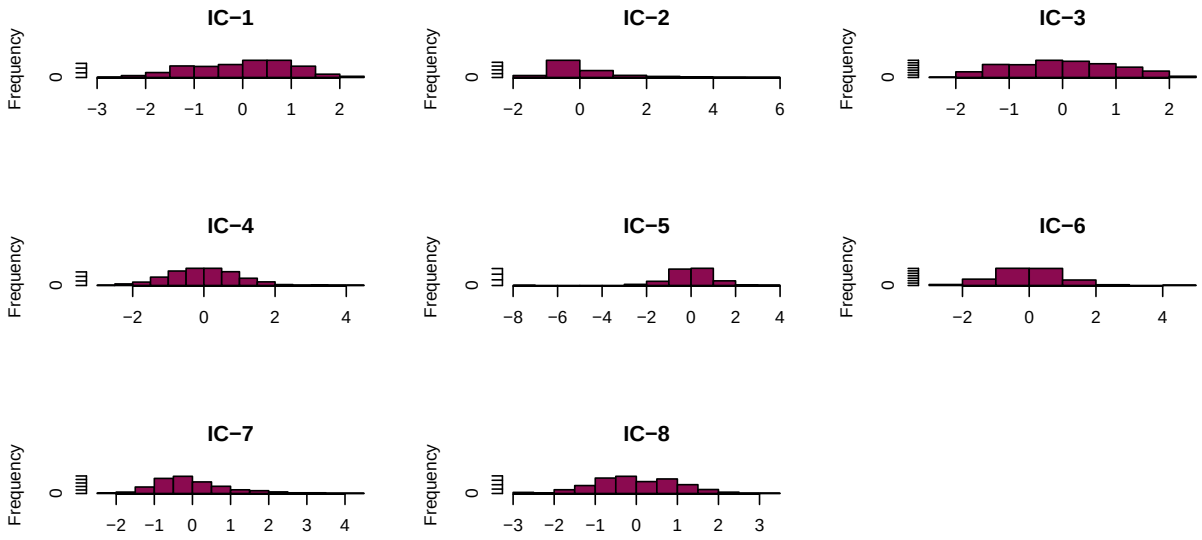
We can see again that the first PC is positively correlated with all the original variables. The second PC is positively correlated with skin thickness and log_BMI, and negatively correlated with age and pregnancies. The third PC is negatively correlated with log_insulin and glucose, and slightly correlated with blood pressure and skin thickness. The fourth PC is negatively correlated with log_DiabetesPedigreeFunction and the fifth PC with blood pressure. The remaining PCs are not correlated heavily with any variables.

Moving on to Independent Component Analysis:

```
require(ica)
data_trans_ica <- icafast(data[,-9], nc = p, alg = "par")
Z <- data_trans_ica$S
colnames(Z) <- sprintf("IC-%d", seq(1,8))
n <- nrow(data)
Z <- Z * sqrt((n-1)/n)

par(mfrow = c(3,3))
sapply(colnames(Z), function(cname){hist(as.data.frame(Z)[[cname]],
                          main = cname, col = "deeppink4", xlab = "")})
```

```
##          IC-1                          IC-2
## breaks   numeric,12                    numeric,9
## counts   integer,11                    integer,8
## density  numeric,11                    numeric,8
## mids     numeric,11                    numeric,8
## xname    "as.data.frame(Z)[[cname]]" "as.data.frame(Z)[[cname]]"
## equidist TRUE                          TRUE
##          IC-3                          IC-4
## breaks   numeric,11                    numeric,16
## counts   integer,10                    integer,15
## density  numeric,10                    numeric,15
## mids     numeric,10                    numeric,15
## xname    "as.data.frame(Z)[[cname]]" "as.data.frame(Z)[[cname]]"
## equidist TRUE                          TRUE
##          IC-5                          IC-6
```

```
## breaks   numeric,13                  numeric,9
## counts   integer,12                  integer,8
## density  numeric,12                  numeric,8
## mids     numeric,12                  numeric,8
## xname    "as.data.frame(Z)[[cname]]" "as.data.frame(Z)[[cname]]"
## equidist TRUE                        TRUE
##          IC-7                        IC-8
## breaks   numeric,15                  numeric,14
## counts   integer,14                  integer,13
## density  numeric,14                  numeric,13
## mids     numeric,14                  numeric,13
## xname    "as.data.frame(Z)[[cname]]" "as.data.frame(Z)[[cname]]"
## equidist TRUE                        TRUE
```

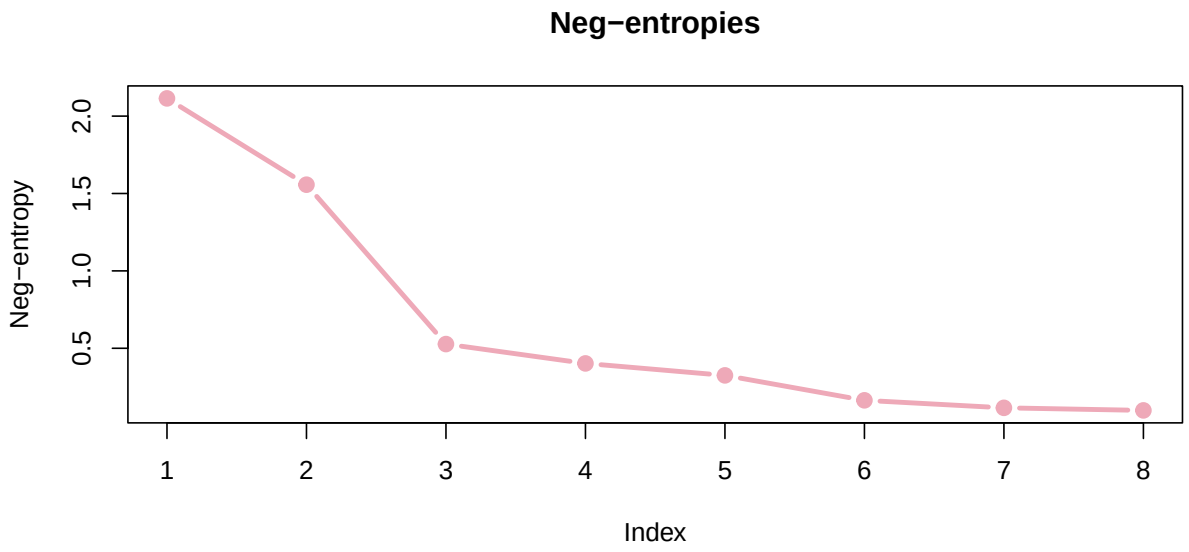


As we can see, all the ICs are centered around 0.

```
neg_entropy <- function(z){1/12 * mean(z^3)^2 + 1/48 * mean(z^4)^2}
Z_neg_entropy <- apply(Z, 2, neg_entropy)
ic_sort <- sort(Z_neg_entropy, decreasing = TRUE, index.return = TRUE)$ix
ic_sort
```

```
## [1] 5 2 7 6 4 8 1 3
```

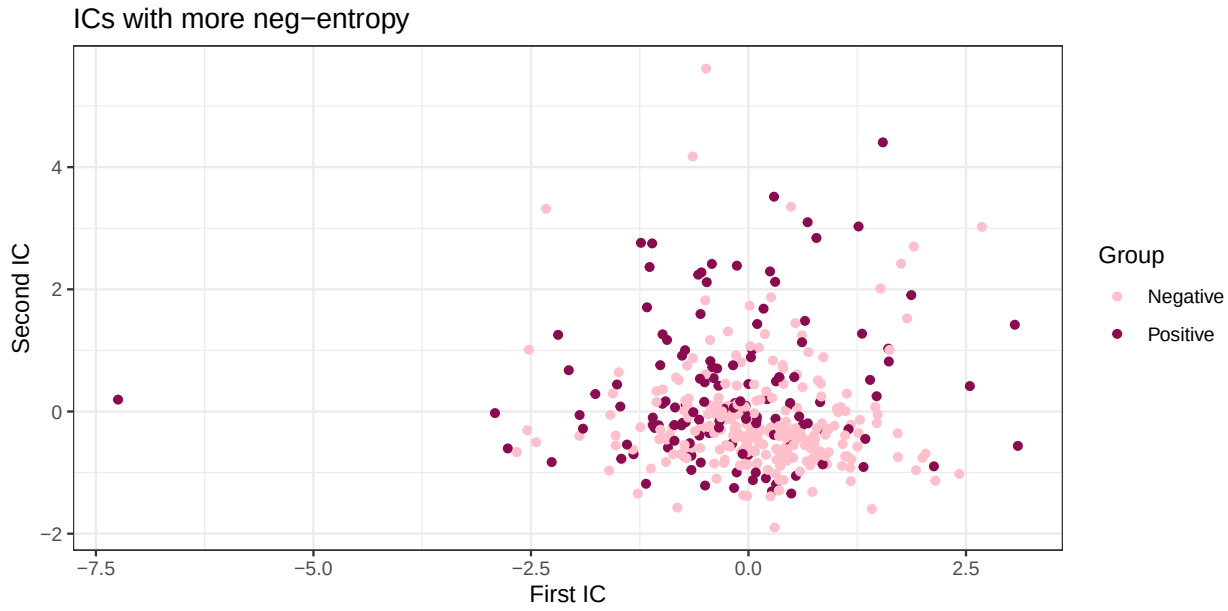
```
par(mfrow = c(1,1))
plot(Z_neg_entropy[ic_sort], type = "b", col = col1, pch = 19,
     ylab = "Neg-entropy", main = "Neg-entropies", lwd = 3)
```




```
Z_ic_imp <- Z[, ic_sort]
```

There are two ICs with negative entropy clearly greater than the other six.

```
df5 <- data.frame(x = Z_ic_imp[,1], y = Z_ic_imp[,2], group = data[,9])
ggplot(df5, aes(x = x, y = y, col = group)) + geom_point() +
  xlab("First IC") + ylab("Second IC") + ggtitle("ICs with more neg-entropy") +
  scale_colour_manual(values = c("pink", "deeppink4")) + theme_bw() + labs(col = "Group")
```



There are some points who may be outliers, and a clear outlier from the "Positive" group:

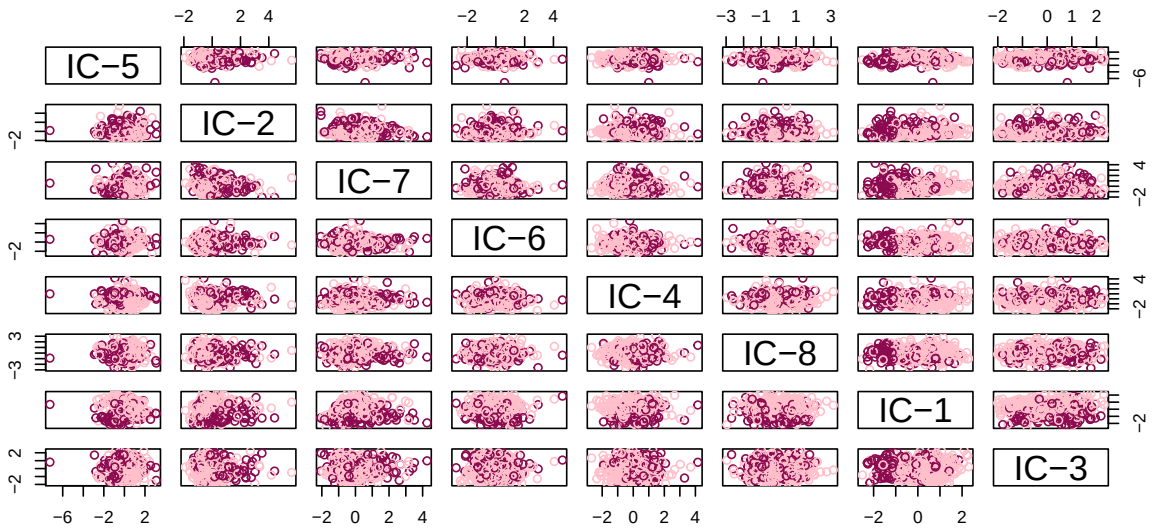
```
which(df5$x < -2.5)
```

```
## [1]  19  56 163 228 295 368
```

```
which(df5$y < -3)
```

```
## integer(0)
```

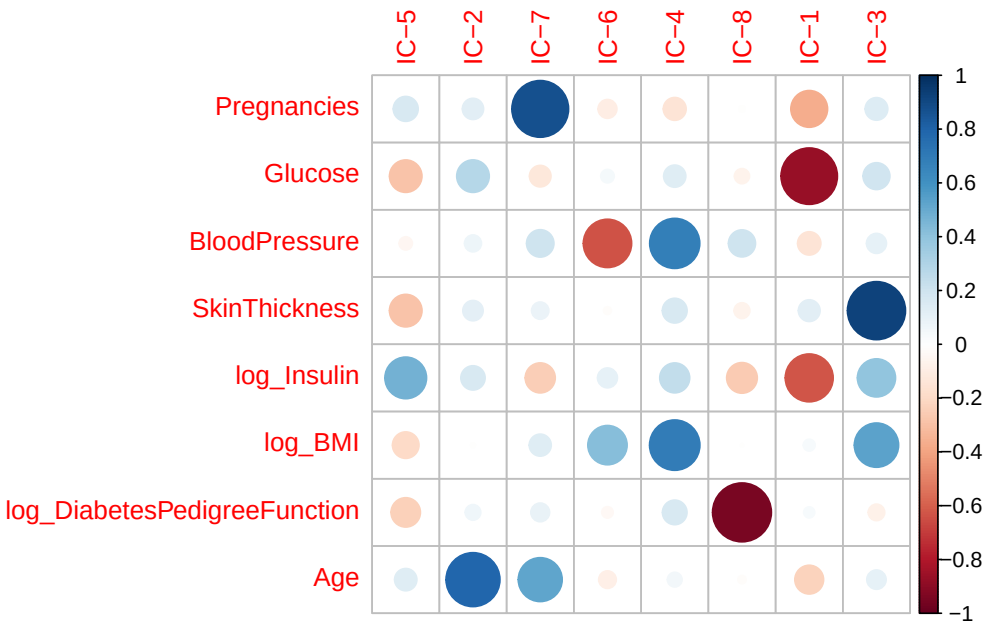
```
pairs(Z_ic_imp, col = vec_col)
```



It seems that IC1, IC2 and IC3 are able to differentiate the groups. IC1 was the IC with the lowest entropy, while IC3 is the second IC with more negative entropy.

The correlation between the original values and Z:

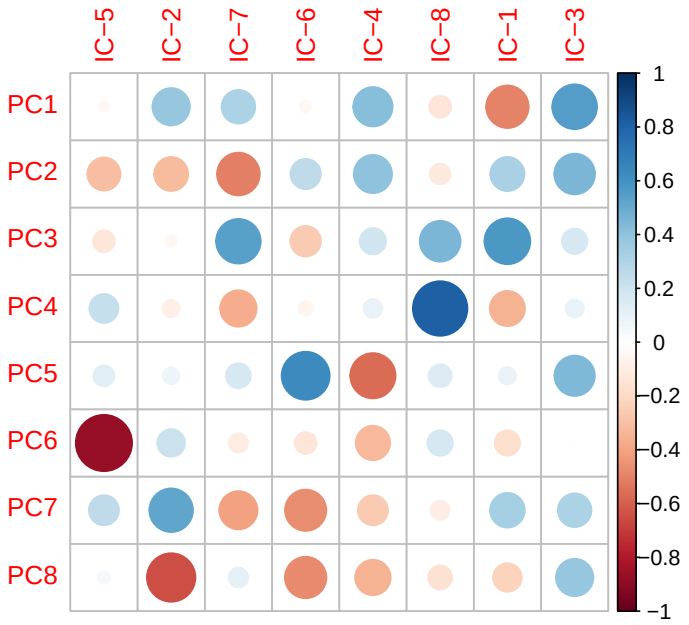
```
corrplot(cor(data[,-9], Z_ic_imp), is.corr = T)
```



Most ICs have at least one original variable that they are highly correlated with.

The correlation between the PCs and the ICs is:

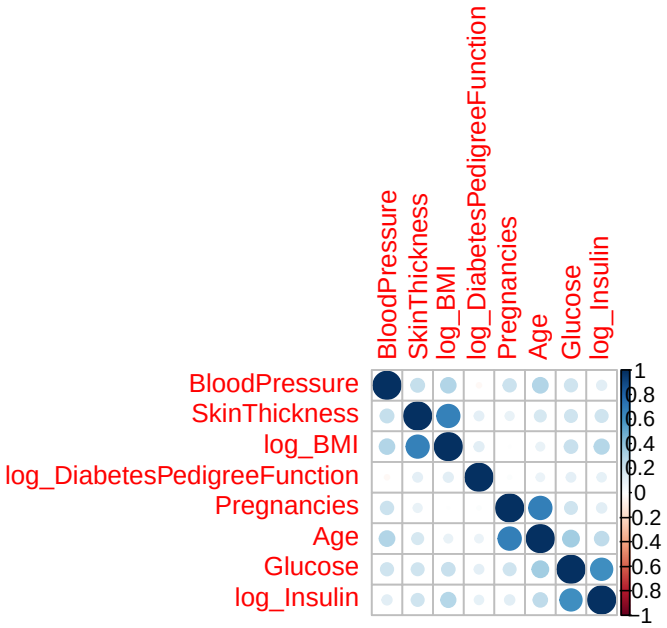
```
colnames(data_pcs$x)<-paste0("PC", 1:8)
corrplot(cor(data_pcs$x, Z_ic_imp), lower.col = "black")
```



We have the same with the PCs. All the ICs have correlations with more than one PC.

Now, we will perform factor analysis.

```
require(psych)
corrplot(cor(data[,-9]), order = "hclust")
```



There are groups of correlated variables that may suggest a factor structure.

We will focus on the first three PCs.

The initial estimates of M and Sigma_nu is:

```
r <- 3
Y <- scale(data[,-9])
Y_pcs <- prcomp(Y)
M_0 <- Y_pcs$rotation[,1:r] %*% diag(Y_pcs$sdev[1:r])
S_y <- cov(Y)
Sigma_nu_0 <- diag(diag(S_y - M_0 %*% t(M_0)))
Sigma_nu_0
```

```
##           [,1]     [,2]      [,3]     [,4]      [,5]      [,6]      [,7]
## [1,] 0.2153076 0.000000 0.0000000 0.000000 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 0.251044 0.0000000 0.000000 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.000000 0.5391157 0.000000 0.0000000 0.0000000 0.0000000
## [4,] 0.0000000 0.000000 0.0000000 0.238202 0.0000000 0.0000000 0.0000000
## [5,] 0.0000000 0.000000 0.0000000 0.000000 0.2367265 0.0000000 0.0000000
## [6,] 0.0000000 0.000000 0.0000000 0.000000 0.0000000 0.1755768 0.0000000
## [7,] 0.0000000 0.000000 0.0000000 0.000000 0.0000000 0.0000000 0.8416111
## [8,] 0.0000000 0.000000 0.0000000 0.000000 0.0000000 0.0000000 0.0000000
##           [,8]
## [1,] 0.0000000
## [2,] 0.0000000
## [3,] 0.0000000
## [4,] 0.0000000
## [5,] 0.0000000
## [6,] 0.0000000
## [7,] 0.0000000
## [8,] 0.1951026
```

The estimation of M without varimax rotation is:

```
MM <- S_y - Sigma_nu_0
MM_eig <- eigen(MM)
MM_values <- MM_eig$values
MM_vectors <- MM_eig$vectors
M_1 <- MM_eig$vectors[,1:r] %*% diag(MM_eig$values[1:r])^(1/2)
M_1
```

```
##            [,1]         [,2]       [,3]
## [1,] -0.5078641  0.612045424 -0.2971279
## [2,] -0.6532412  0.049348044  0.5003773
## [3,] -0.4285483  0.001845186 -0.2051417
## [4,] -0.5768428 -0.499218329 -0.3170991
## [5,] -0.6127446 -0.061039576  0.5670143
## [6,] -0.5889778 -0.615724495 -0.2287772
## [7,] -0.1468150 -0.063693032  0.0487939
## [8,] -0.6521083  0.542100972 -0.1916721
```

After the varimax rotation, we arrive at the following:

```
M <- varimax(M_1)
M <- loadings(M)[1:p,1:r]
M
```

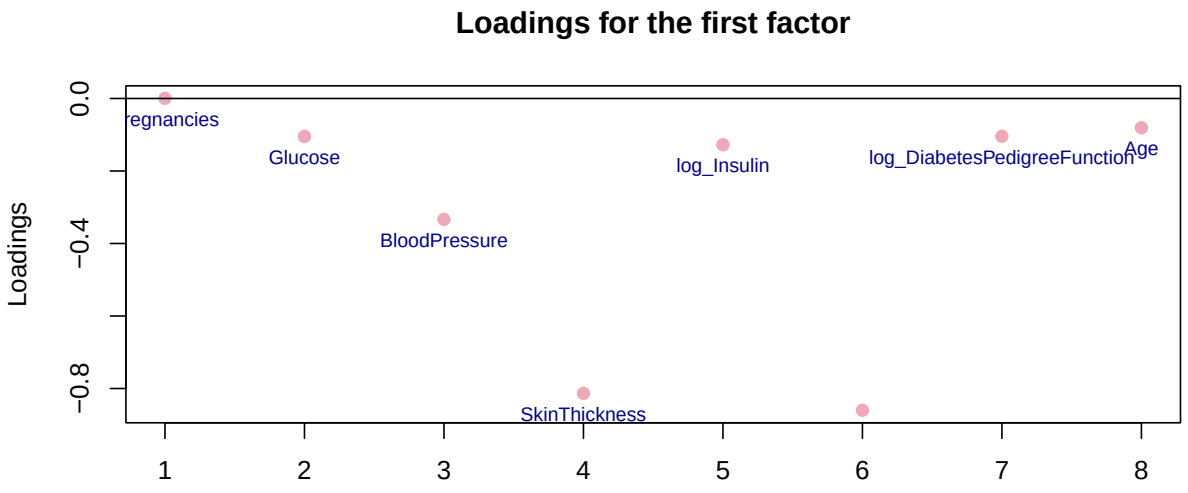
```
##               [,1]        [,2]       [,3]
## [1,]  0.0005026846  0.84766628 0.04766654
## [2,] -0.1048418644  0.20719877 0.79095740
## [3,] -0.3334523373  0.32485372 0.09497247
## [4,] -0.8133493889  0.09042161 0.11315973
## [5,] -0.1277137138  0.07805335 0.82357990
## [6,] -0.8601353388 -0.02196437 0.19502649
## [7,] -0.1041863743  0.01763691 0.12971669
## [8,] -0.0806665916  0.83691004 0.22120444
```

```
Sigma_nu <- diag(diag(S_y - M %*% t(M)))
Sigma_nu
```

```
##           [,1]      [,2]      [,3]      [,4]     [,5]      [,6]      [,7]
## [1,] 0.2791895 0.0000000 0.0000000 0.0000000 0.000000 0.0000000 0.0000000
## [2,] 0.0000000 0.3204632 0.0000000 0.0000000 0.000000 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000 0.7742598 0.0000000 0.000000 0.0000000 0.0000000
## [4,] 0.0000000 0.0000000 0.0000000 0.3174816 0.000000 0.0000000 0.0000000
## [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.299313 0.0000000 0.0000000
## [6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.2216494 0.0000000
## [7,] 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000 0.9720077
## [8,] 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000 0.0000000
##           [,8]
## [1,] 0.0000000
## [2,] 0.0000000
## [3,] 0.0000000
## [4,] 0.0000000
## [5,] 0.0000000
## [6,] 0.0000000
```

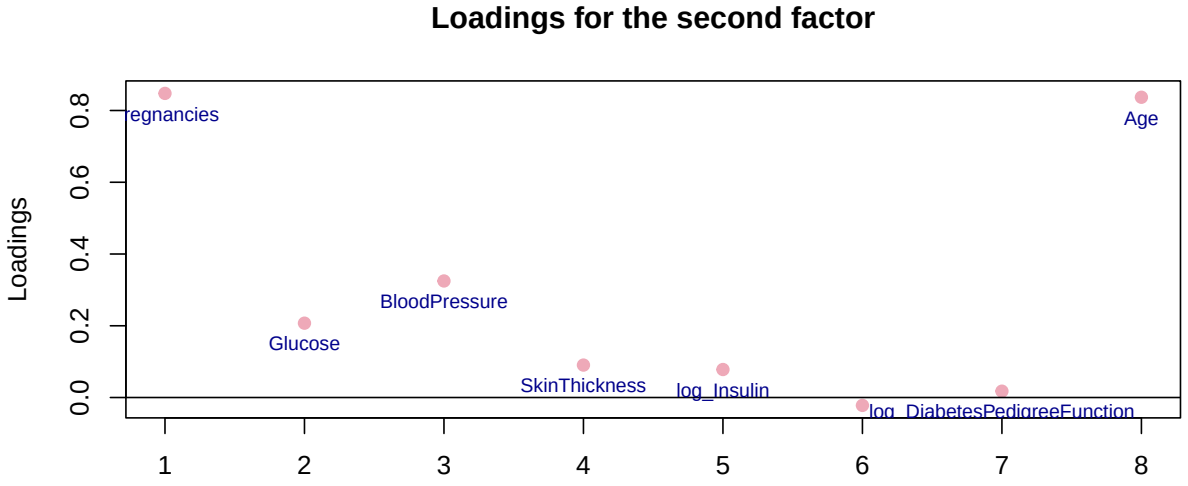
```
## [7,] 0.0000000
## [8,] 0.2441431
```

```
X<-data[,-9]
plot(1:p, M[,1], pch = 19, col = col1, xlab = "", ylab = "Loadings", main = "Loadings for the first fac
abline(h = 0)
text(1:p, M[,1], labels = colnames(X), pos = 1, col = col2, cex = 0.75)
```



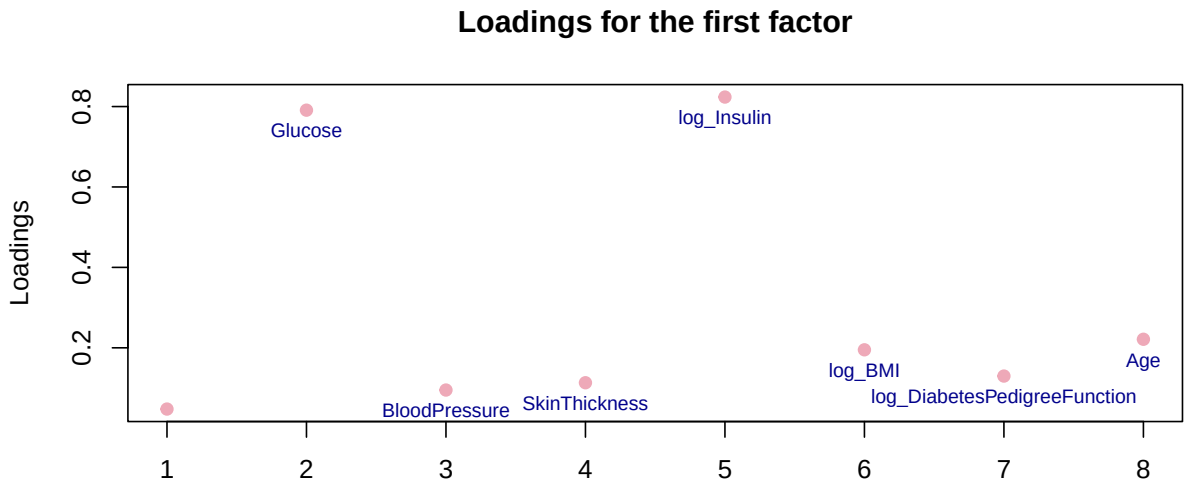
The first factor appears to be very related to skin thickness.

```
plot(1:p, M[,2], pch = 19, col = col1, xlab = "", ylab = "Loadings", main = "Loadings for the second fa
abline(h = 0)
text(1:p, M[,2], labels = colnames(X), pos = 1, col = col2, cex = 0.75)
```



The second factor appears to be very related to pregnancies and age.

```
plot(1:p, M[,3], pch = 19, col = col1, xlab = "", ylab = "Loadings", main = "Loadings for the first fac
abline(h = 0)
text(1:p, M[,3], labels = colnames(X), pos = 1, col = col2, cex = 0.75)
```



The third factor appears to be related to log_Insulin and glucose. We lack the medical expertise to judge if this makes sense or not, but it is what our data says.