Pima Diabetes Dataset: Missingness Exploration and Imputation

2026-01-17

Analysis context

This dataset was originally obtained from the National Institute of Diabetes and Digestive and Kidney Diseases [1]. The dataset contains the medical information of females who, at the time of inclusion in the study, were at least 21 years old and of Pima Indian heritage. The dataset's initial objective was to predict, based on certain diagnostic measurements included in the dataset (number of pregnancies the patient, BMI, insulin level, age etc.), whether or not a patient had diabetes.

As the dataset contains many values equal to zero (which, in some cases, is biologically impossible), the initial assumption of the analysis was that these values should be assigned as missing. Therefore, the aim of this analysis was to investigate this possibility, examine the missingness patterns, select an imputation method and perform predictions on the data.

[1] *Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261–265). IEEE Computer Society Press.*

Required libraries

```
library(randomForest)
library(naniar)
library(mice)
library(ggplot2)
library(ggpubr)
library(rstatix)
library(Hmisc)
library(tidyverse)
library(patchwork)
library(visdat)
library(pROC)
```

Reading of the dataset and initial inspection

```
diabetes <- read.csv("../PRIMA/diabetes_data.csv",header=T)

head(diabetes)
```

```
##   Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           6     148            72            35       0 33.6
## 2           1      85            66            29       0 26.6
```

```
## 3           8     183            64             0       0 23.3
## 4           1      89            66            23      94 28.1
## 5           0     137            40            35     168 43.1
## 6           5     116            74             0       0 25.6
##   DiabetesPedigreeFunction Age Outcome
## 1                    0.627  50       1
## 2                    0.351  31       0
## 3                    0.672  32       1
## 4                    0.167  21       0
## 5                    2.288  33       1
## 6                    0.201  30       0
```

```
str(diabetes)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
##  $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
##  $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

The dataset contains 768 observations and nine variables, all of which are integers or numeric. The *Outcome* variable should be a factor, so a transformation will be performed. Next, the dataset will be checked for any missing values.

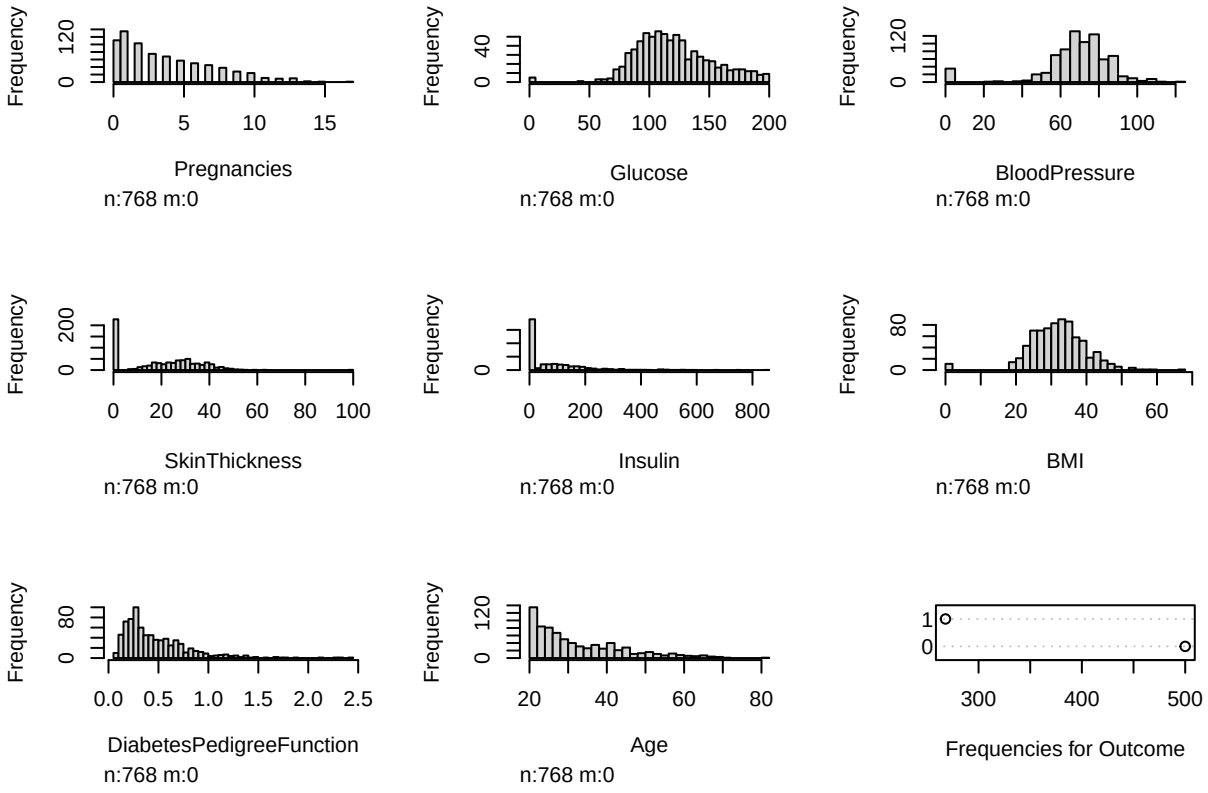
```
diabetes$Outcome<- as.factor(diabetes$Outcome)

summary(diabetes)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725           Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719           Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200           Max.   :81.00
##  Outcome
##  0:500
##  1:268
##
##
##
##
```

```r
na_counts <- colSums(is.na(diabetes))
print(na_counts)
```

```
##              Pregnancies                  Glucose            BloodPressure
##                        0                        0                        0
##            SkinThickness                  Insulin                      BMI
##                        0                        0                        0
## DiabetesPedigreeFunction                      Age                  Outcome
##                        0                        0                        0
```

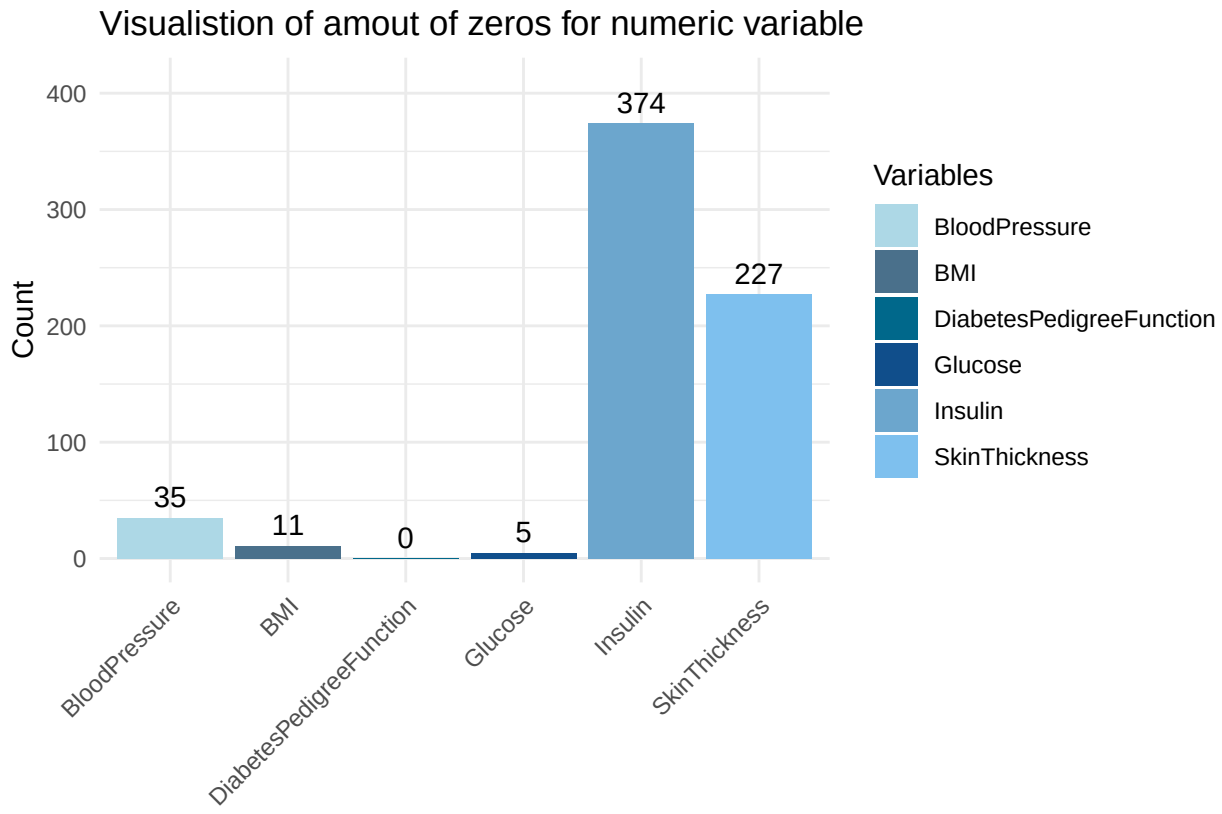
As expected, there are no missing values in the dataset. However, the dataset summary shows that some variables have zeros as minimum values. In some cases, such as BloodPressure or Insuline, this is biologically impossible. To investigate this issue further, the distribution of the data will be checked.



Some of the variables (*Glucose*, *BloodPressure*, *SkinThickness*, *DiabetesPedigreeFunction*, *Insulin*, and *BMI*) contain many values equal to 0, which is biologically implausible and likely indicates that these measurements were not collected. In the first step, a deeper analysis of the impact of these zero values on the data structure will be performed. From this analysis *Outcome* and *Pregnecies* variables were excluded as zero values in those columns are expected.

Data exploration

```
diabetes_exploration<-diabetes
diabetes_exploration_total_n <- as.data.frame(colSums(diabetes_exploration[c(2:7)]==0))
colnames(diabetes_exploration_total_n) <-"Count"
```



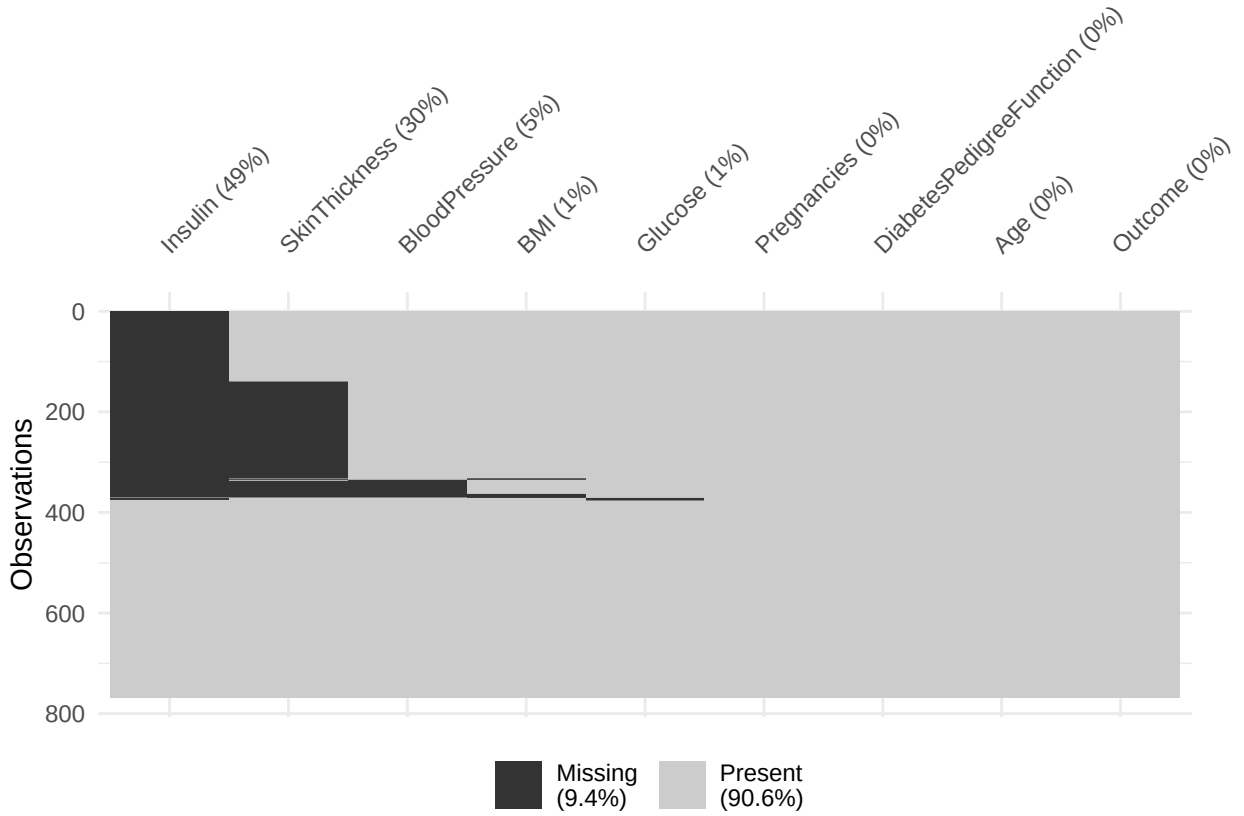
The highest number of zeros is present in *Insulin* and *SkinTickness* columns.

As mentioned, these values should probably be recorded as missing. To further analyse these values, zeros will be replaced with NAs, which allows the use of visualization techniques to explore missingness patterns. This replacement will be performed for all columns except *Pregnancies* and *Outcome*.

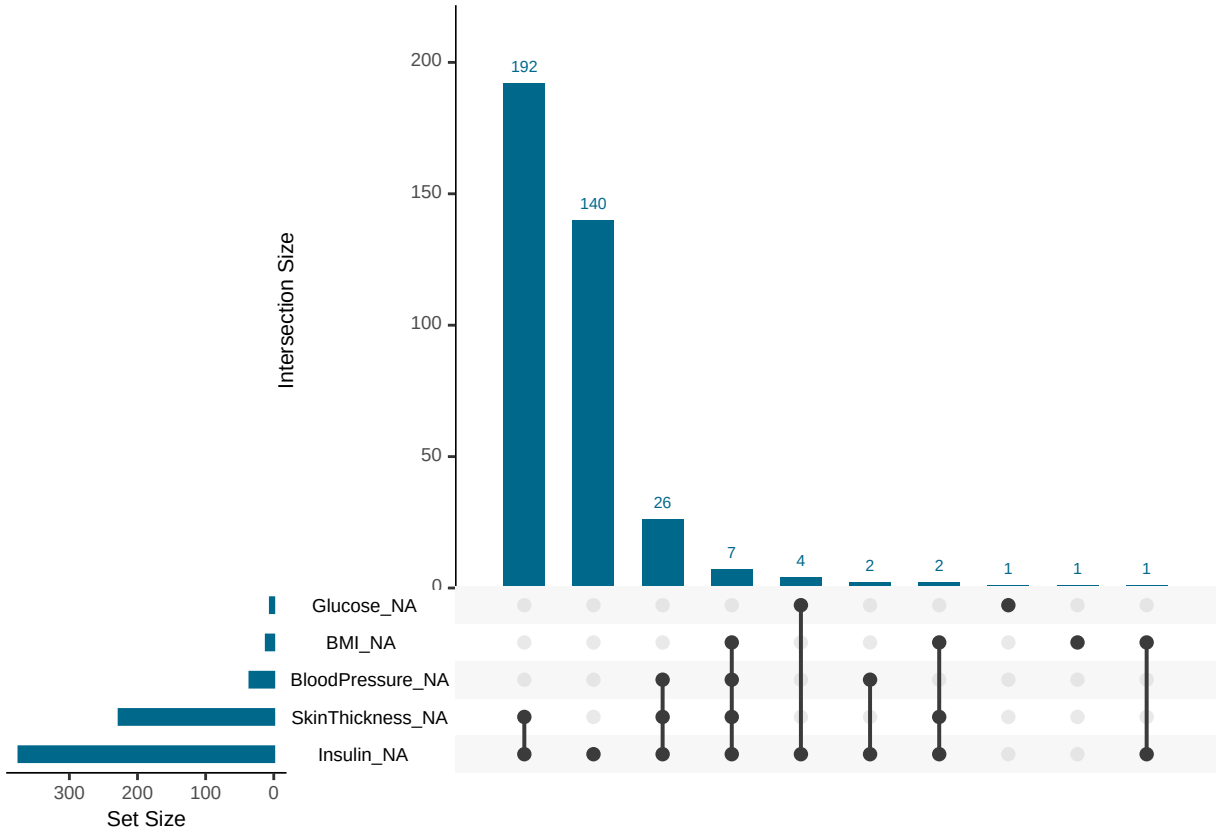
Missingness evaluation

```
diabetes_imp <- diabetes %>% mutate(across(2:8, ~ na_if(.x, 0)))
```

```
vis_miss_p<-vis_miss(diabetes_imp, cluster = T, sort_miss = T, show_perc = T, show_perc_col = T)
vis_miss_p +  # move axis below panel
  theme(axis.text.x = element_text(angle = 45, hjust = 0,vjust=0.1),
        plot.margin = margin(t = 1, r = 20, b = 10, l = 5))
```



It is visible that some missing values co-occur across variables, suggesting that the data may not be missing at random. To further explore missingness patterns, the gg_miss_upset function will be used.



The plot shows that some values tend to co-occur. The most frequent pair (192 observations) is SkinTickness and Insulin NAs, which are in the same row. This implies that the missing values are not missing completely at random. This has implications for the selection of the imputation method. The themcar_test function will be used to confirm that the missingness is not random.

```
mcar_test(diabetes_imp[2:8])
```

```
## # A tibble: 1 x 4
##   statistic    df  p.value missing.patterns
##       <dbl> <dbl>    <dbl>            <int>
## 1      146.    49 1.50e-11               11
```

The very low p-value of this test indicates that the MCAR (missing completely at random) assumption has been rejected; therefore, missingness may be MAR (missing at random) or MNAR (missing not at random). Together with the high proportion of missing values, this suggests that simple imputation methods, such as mean or median imputation, are not appropriate.

The next step will assess the plausibility of a MAR mechanism using logistic regression. Missingness indicators for each variable will be modelled sequentially as a function of the observed predictors.

```
vars <- c("Insulin", "Glucose","BMI","SkinThickness","BloodPressure")

pvals <- sapply(vars, function(v) {
  preds <- setdiff(c("Insulin","Glucose","BMI","SkinThickness","BloodPressure"), v)
  f <-as.formula(paste0("is.na(", v, ") ~", paste(preds, collapse = " + ")))
  fit <- glm(f, data = diabetes_imp, family = binomial)
  coef(summary(fit))[-1, 4] })
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: algorithm did not converge
```

```
print(pvals)
```

```
##                    Insulin   Glucose       BMI SkinThickness BloodPressure
## Glucose       0.0156630760 0.5438107 0.5184914             1             1
## BMI           0.0418066471 0.6661565 0.5837327             1             1
## SkinThickness 0.2066036585 0.4685898 0.6736850             1             1
## BloodPressure 0.0005982621 0.5780214 0.6369339             1             1
```

glm.fit: algorithm did not converge warning often occurs when the model attempts to fit a logistic regression and experiences perfect separation, whereby a predictor variable is able to perfectly categorise the response variable as 0 or 1. This is quasi-complete separation and makes it impossible to reliably estimate Wald p-values. In this case, we iterate the tested variables, assigning them as the response variable in the model and using the remaining variables as predictors. For these results, this means that based on the predictor, we can perfectly predict whether or not there will be a missing value in the dependent variable.

This indicates a very strong co-occurrence of missingness, suggesting a missing-at-random (MAR) mechanism, where missingness depends on other observed variables but not on the missing value itself, rather than missing completely at random (MCAR), where missingness is purely random and unrelated to any data, or missing not at random (MNAR), where missingness depends on the unobserved value. Given the plausibility of MAR, multiple imputation using MICE will be performed.

As the next step involves predictive modeling, the data will first be split into training and testing sets.

Data imputation

```
set.seed(123)
n <- nrow(diabetes_imp)
train_idx <- sample(seq_len(n),size = 0.7 * n)

training_dataset <- diabetes_imp[train_idx, ]
dim(training_dataset)
```

```
## [1] 537   9
```

```
testing_dataset <- diabetes_imp[-train_idx, ]
dim(testing_dataset)
```

```
## [1] 231   9
```

The datasets will be imputed separately using the MICE algorithm, and the imputation results will be visualized using bwplot and densityplot.

```
bwplot(imp_train)
```

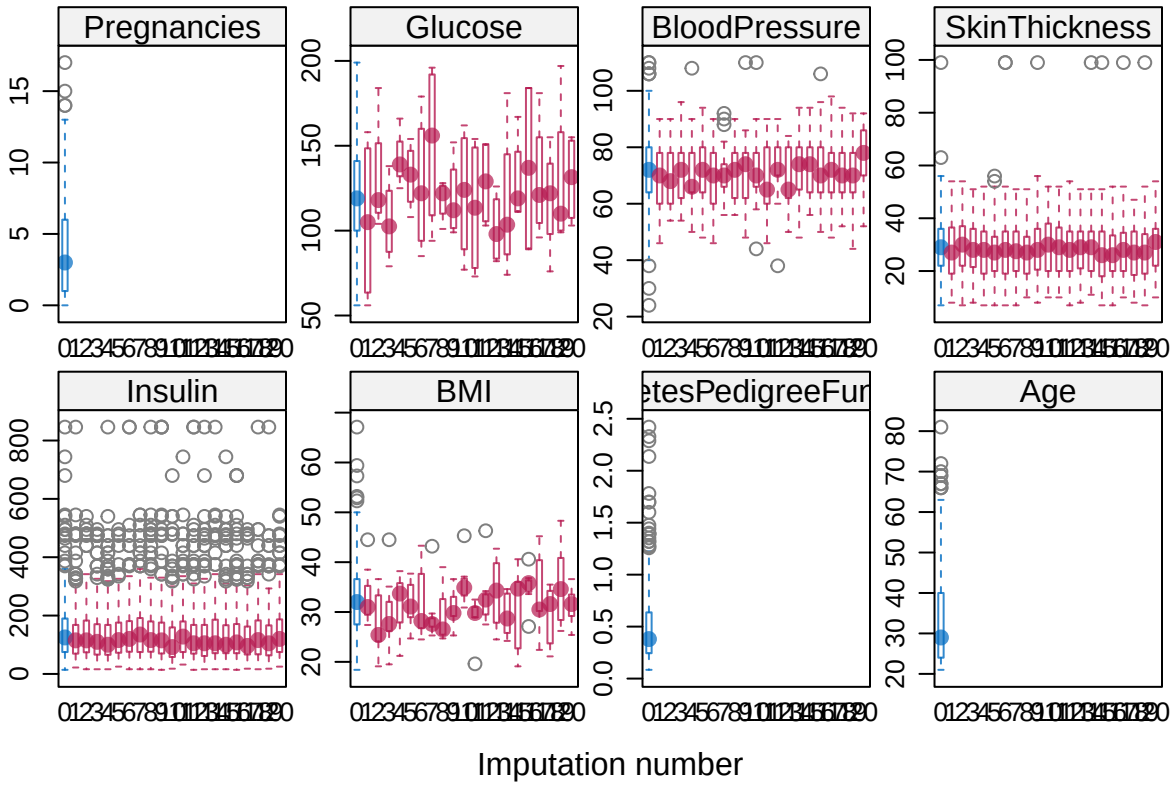
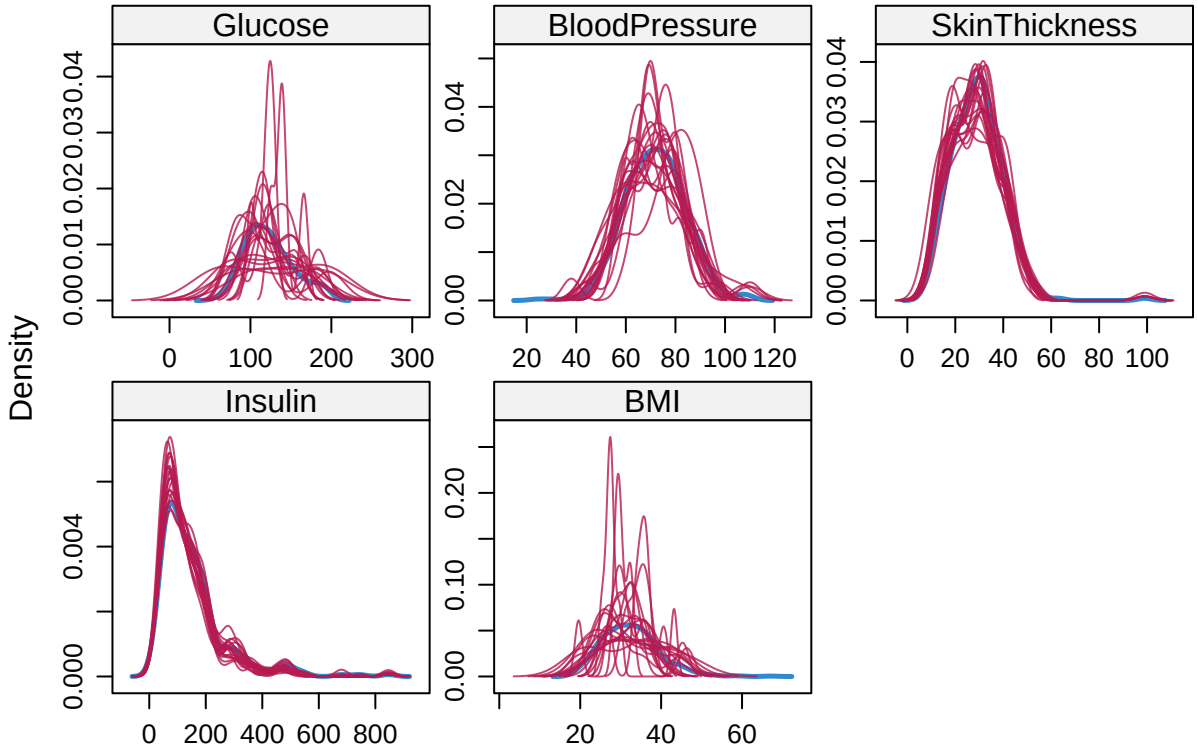


Figure 1: Diagnostics plot for train dataset imputation

```
densityplot(imp_train)
```




```
bwplot(imp_test)
```

```
densityplot(imp_test)
```

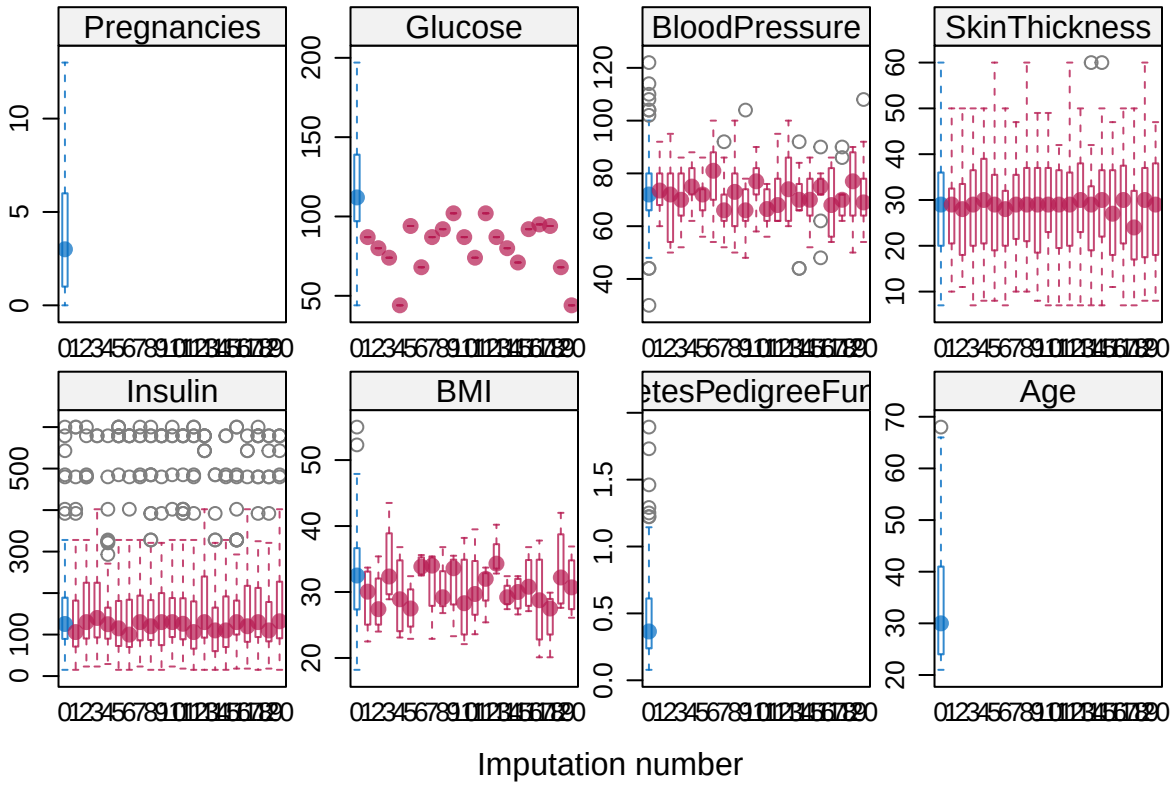
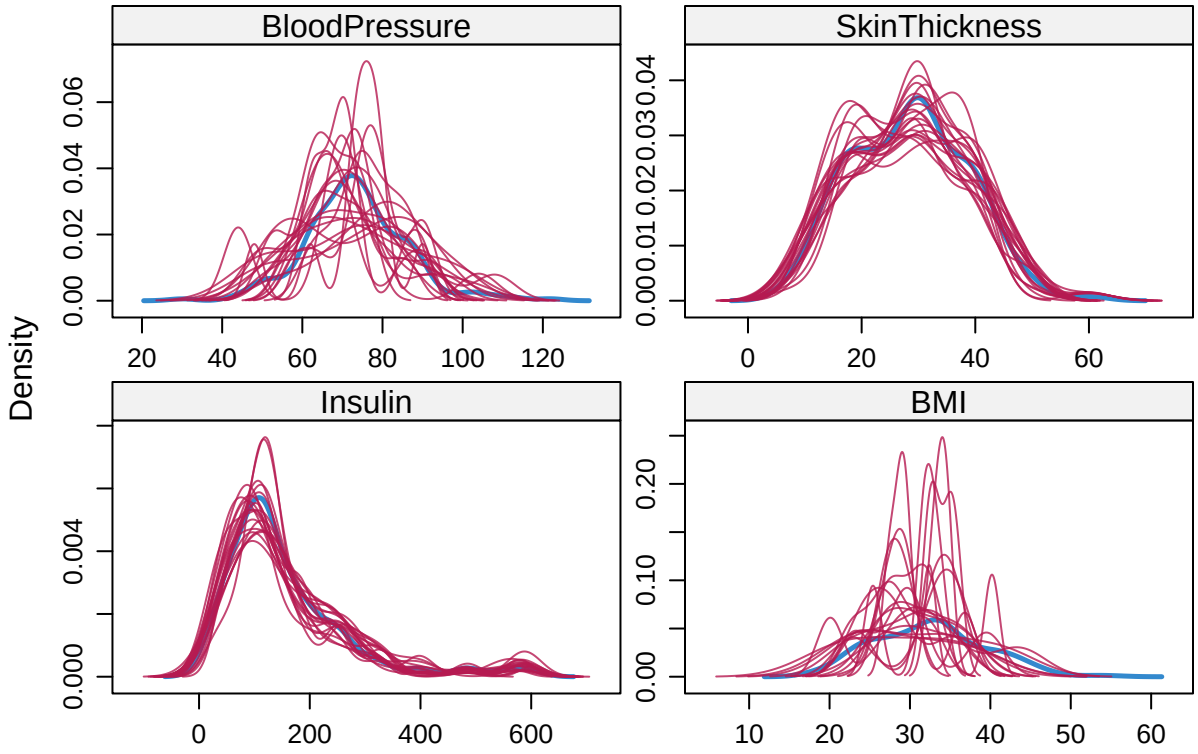


Figure 2: Diagnostics plots for test dataset imputation



The diagnostic plots of the imputed data for both the training and testing datasets showed that the distributions of the imputed data matched the original ones well, with the exception of *Glucose* and *BMI* (visible on the box-whisker plots). This is due to the low number of missing data (five for *Glucose* and eleven for *BMI*) and the high variability of the imputed data related to this number. As only one glucose value was imputed in the testing dataset, a density plot for this variable was not generated.

Models generation

Next, two predictive models will be evaluated: a logistic regression model and a random forest model. For each method, the models will be fitted separately within each imputed dataset. Predictions will then be generated using the imputed test datasets and the predicted probabilities will be averaged across the imputations to produce a pooled estimate.

```
fit_glm <- with(imp_train,
glm(Outcome~Pregnancies+Glucose+BloodPressure+SkinThickness+Insulin+BMI+DiabetesPedigreeFunction+Age,fa

completed_train  <- lapply(1:imp_train$m, function(i) complete(imp_train, i))

glm_preds <-sapply(1:imp_train$m, function(i){
  predict(fit_glm$analyses[[i]],newdata = complete(imp_test, i),type = "response" )})

final_pred_glm <- rowMeans(glm_preds)
```

```
rf_preds <- sapply(1:imp_train$m, function(i) {
  train_complete <-complete(imp_train, i)
```

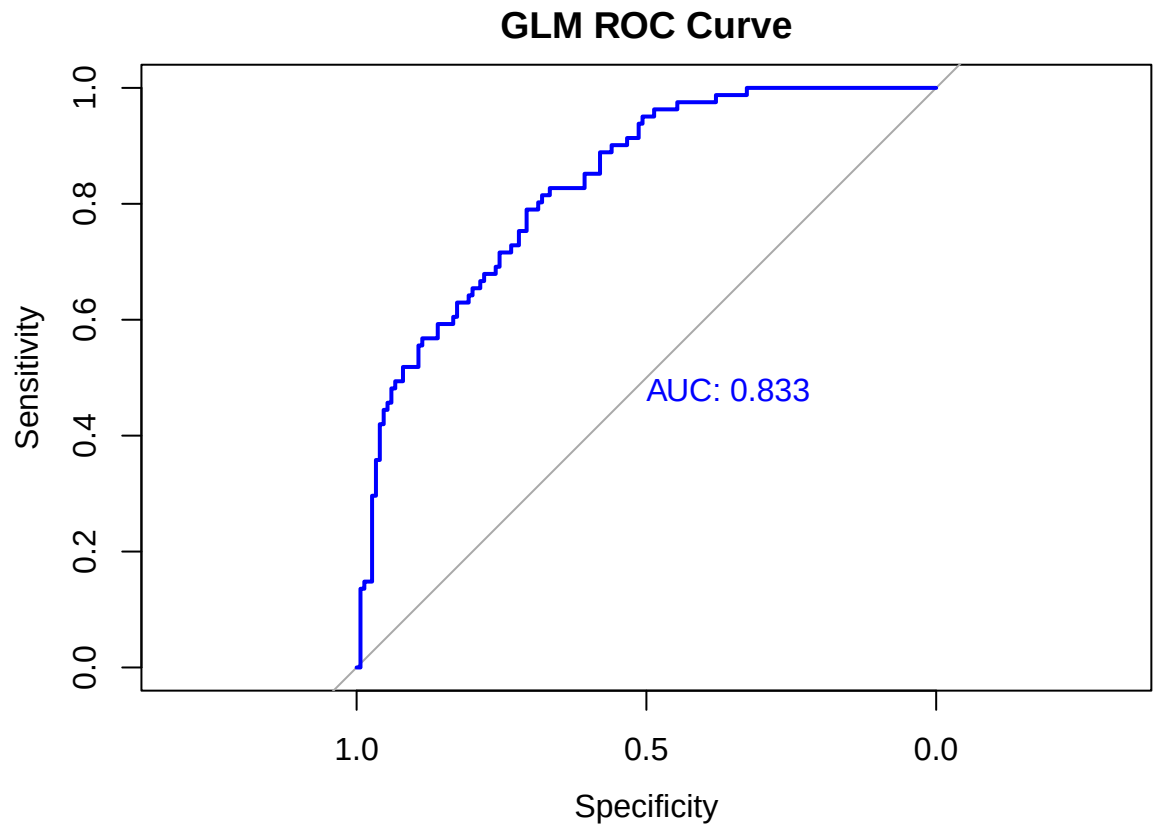
```
  test_complete <- complete(imp_test, i)
  rf <-randomForest(Outcome ~
Pregnancies+Glucose+BloodPressure+SkinThickness+Insulin+BMI+DiabetesPedigreeFunction+Age,
data = train_complete, ntree = 500)

predict(rf, newdata = test_complete, type = "prob")[,2] })

final_pred_rf <- rowMeans(rf_preds)
```

To evaluate predictive performance, the AUC will be calculated and ROC curves will be generated.

```
roc_obj <- roc(testing_dataset$Outcome, final_pred_glm)
auc(roc_obj)
```

```
## Area under the curve: 0.8333
```

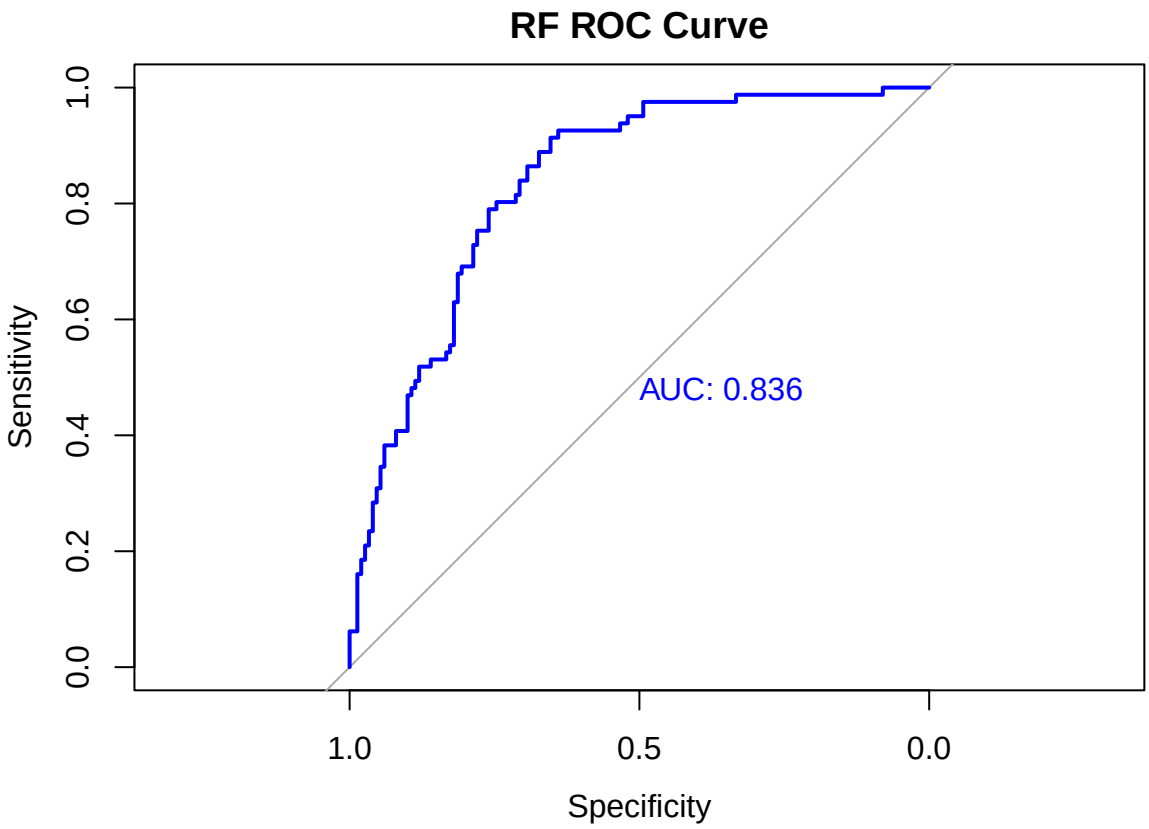
```
plot(roc_obj, col = "blue", main = "GLM ROC Curve", print.auc = TRUE)
```




```
roc_rf <- roc(testing_dataset$Outcome, final_pred_rf)
auc(roc_rf)
```

```
## Area under the curve: 0.8361
```

```r
plot(roc_rf, col = "blue", main = "RF ROC Curve", print.auc = TRUE)
```



As shown both modles were able to accuretelly predict the *Outcome* data based on the imputed data, which is indicated by high values of AUC for both models.

Summary

This analysis uses multiple imputation to handle missing data, reducing bias from complete-case analysis and enabling fair comparison of predictive models.