# Exploring Risk Factors for Diabetes

CFAS440 - Generalised Linear Models II

Harry Baines

35315878

h.baines3@lancaster.ac.uk

**Abstract**

Diabetes is a major lifelong disease which affects millions of people worldwide today. In this paper, we seek to explore and identify the risk factors for Diabetes by implementing a generalised linear model to predict if a patient has Diabetes. A dataset provided by the National Institute of Diabetes and Digestive and Kidney Diseases will enable us to fit a logistic model to predict if an individual is diabetic based on a set of diagnostic measurements.

## 1 Introduction

### 1.1 Background and Motivation

Diabetes is a chronic disease that occurs either when the pancreas does not produce enough insulin (Type 1 Diabetes) or when the body cannot effectively use the insulin it produces (Type 2 Diabetes). Insulin is a hormone that regulates blood sugar in the body. Hyperglycaemia, or raised blood sugar, is a common effect of uncontrolled diabetes and over time leads to serious damage to many of the body's systems, especially the nerves and blood vessels [1].

The data set used in this paper, donated by Vincent Sigillito, is a collection of medical diagnostic measurements of 768 examples from a population living near Phoenix, Arizona in the USA. All patients are females at least 21 years old of Pima Indian heritage.

### 1.2 Aims

The aims of this paper are as follows:

1. Find the best logistic regression model to predict if a patient is diabetic or not
2. Identify which covariates have a significant effect on whether a person has Diabetes
3. Find the most parsimonious regression model
4. Find and interpret relationships between the explanatory variables

### 1.3 Data Description

The data are described by 8 variables as summarised in Table 1. The Outcome variable indicates if this patient was tested positive for Diabetes or not (indicated by a 1 and 0 respectively). The other 7 measurements will be our predictor variables, which are quantitative medical measurements taken from each patient:

| Variable | Description | Data Type |
|---|---|---|
| Pregnancies | Number of pregnancies | Integer |
| BloodPressure | Diastolic blood pressure (mm Hg) | Integer |
| SkinThickness | Triceps skin fold thickness (mm) | Integer |
| Insulin | 2-Hour serum insulin ($\mu$U/ml) | Integer |
| BMI | Body Mass Index | Numeric |
| DiabetesPedigreeFunction | Synthesis of the history of Diabetes in relatives | Numeric |
| Age | Age of the individual in years | Integer |
| Outcome | Yes or No, for diabetic according to WHO criteria | Integer |

Table 1: Diabetes data set variable descriptions.

### 1.4 Report Outline

In section 2 we outline the mathematical theory behind logistic regression, the technique we will use to develop our predictive model. In section 3 we conduct an Exploratory Data Analysis of the Diabetes dataset. In section 4 we implement the model, using stepwise selection to obtain a model with only significant covariates. In section 5 we carry out regression diagnostics to check our model assumptions hold and check for potential outliers. In section 6 we conclude with a summary of our key findings.

## 2 Theory and Methodology

In this section we primarily discuss the theory behind logistic regression, the technique we will use to fit a model to the Diabetes data in order to make a reasonable prediction for whether a patient is diabetic or not.

Logistic regression is a predictive analysis technique which models the relationship between a binary dependent variable and one or more explanatory variables. As opposed to standard linear regression, logistic regression utilises a logistic function (also known as sigmoid) which ensures computed values are between 0 and 1 - this is important as we are modelling probabilities.

In this paper we utilise the idea of a generalised linear model, a generalization of ordinary linear regression where the response variable $Y_i$ is assumed to follow a probability distribution from the exponential family with mean $\mu_i$. A generalised linear model consists of three components:

- A set of response variables $Y_1, Y_2, ..., Y_N$, assumed to have the same distribution from the exponential family
- A linear predictor with terms consisting of explanatory variables: $\eta_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + ...$
- A link function, linking the linear predictor to the mean of the distribution: $g(\mu_i) = \eta_i$, where $\mu_i = E(Y_i)$

Logistic regression is the default generalized linear model for the binomial family. The model estimates the probability $p$ of a binary outcome Y. This model is linked to a linear equation using the logit function, taking probability values bounded between 0 and 1.

The odds ratio is a measure of association between exposure and an outcome, computed by taking the probability of success divided by the probability of failure, with values ranging anywhere between 0 and $\infty$. We obtain the logit (left side of equation (1):) by taking the logarithm of both sides:

$$\log(\frac{\mathrm{p}}{1-\mathrm{p}}) = \beta_0 + \beta_1 \mathrm{X} + \epsilon \tag{1}$$

The logit function is responsible for converting probabilities to log-odds, where the inverse of the logit converts back to normal probabilities. The logit link function will enable the linear predictor with terms consisting of explanatory variables, factors and interactions to be linked to the mean of the distribution.

Fundamentally this means increasing X by one unit changes the logit by $\beta_0$. The unknown values for coefficients $\beta_0$ and $\beta_1$ will be estimated given our dataset using the `glm()` function in R.

The binomial distribution is used for logistic regression and we can prove it belongs to the exponential family (see Appendix B for the proof) [2].

## 3 Exploratory Analysis

In this section we conduct an Exploratory Data Analysis on the diabetes dataset to understand, inspect and clean the data without making any assumptions about what the data may contain. This will involve calculating summary statistics for the data, identifying potential outliers, dealing with missing values and understanding the distributions and relationships between the explanatory variables.

### 3.1 Response Variable

The Outcome variable indicates if a patient had Diabetes and is our response variable, so we convert this to a factor variable using R. The dataset had 500 women who didn't have Diabetes and 268 women that were diagnosed with Diabetes.

## 3.2 Summary Statistics and Missing Values

First we compute summary statistics for the data (see Appendix C for the full table). We notice minimum values of 0 for BloodPressure, SkinThickness, Insulin and BMI. We know that in reality values of 0 for these variables are highly unlikely if not impossible, so we conclude this is how the missing values are represented. In summary we found 35 missing values for BloodPressure (5% of the data), 227 missing values for SkinThickness (30% of the data), 11 missing values for BMI (1.4% of the data) and 374 missing values for Insulin (50% of the data). These zero measurements should not be included in the model we will develop in the following chapter.

Considering a large proportion of measurements for Insulin and SkinThickness are missing, and the data contains only 768 observations, it would be unreasonable to delete the observations containing missing values as we would lose information gathered for other variables. Therefore we perform imputation, a process of substituting missing values with appropriate alternate values. We utilise the technique of generalized imputation, that is, we compute the median value of the variable associated with the missing value, and substitute this value where missing values are present for that particular variable.

## 3.3 Graphical Analysis

The plot in Appendix D shows the majority of patients were aged 21-30 with a generally decreasing number of patients in the older age categories.

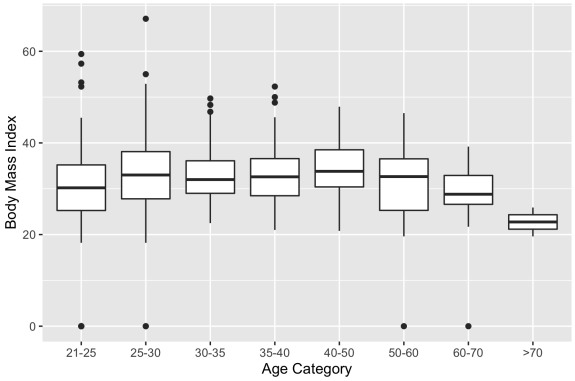


Figure 1: Box-plot of BMI against Age Category (before imputation).

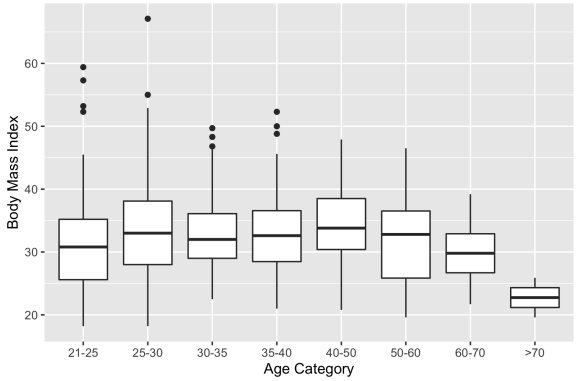


Figure 2: Box-plot of BMI against Age Category (after imputation).

From the box-plot in Figure 1 we notice our missing values when BMI is 0. We generally notice a symmetrical data distribution for BMI for each age category as the median value lies roughly in the middle of each box. After imputation, (Figure 2), the missing values have been replaced with the median value. There is no significant difference to the original data distribution after imputing the missing values.

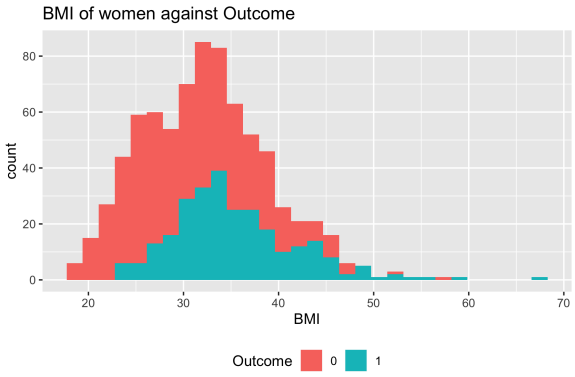


Figure 3: Box-plot of BMI by each Outcome.

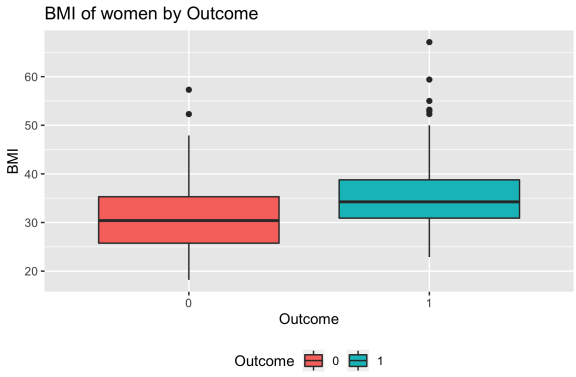


Figure 4: Box-plot of BMI against Outcome.

The plots in Figures 3 and 4 show that all women who had Diabetes generally had BMI values greater

than women who didn't have Diabetes. When we develop our model in section 4 we expect the BMI predictor variable to influence the Outcome response variable.

### 3.4 Correlations

If we observe highly correlated variables in our data, we should remove them as high correlations among predictors means you can predict a variable using another predictor variable, a problem more commonly known as multicollinearity [3]. We can visualise correlations between explanatory variables using a correlogram (see Appendix E) - we observe no strong correlations between any pair of variables, so there is no need to drop any variables before implementing the regression model.

## 4 Estimation and Model Selection

In this section we will develop the best prediction model using backwards logistic regression to predict the binary outcome variable (if someone has Diabetes or not).

### 4.1 Baseline Model

We begin by implementing a simple classification method known as ZeroR. This classifier simply predicts the majority class. It is used primarily for determining a baseline performance as a benchmark for more complex classification methods. We begin by constructing a frequency table for the Outcome variable, and we find 500 patients weren't diabetic (class 0) and 268 patients were diabetic (class 1). As class 0 is the most frequent value in the frequency table (500), and Outcome=0 is the ZeroR model for the Diabetes dataset, our baseline accuracy is $\frac{500}{768} = 65\%$. Any model with an accuracy less than this baseline accuracy won't be considered as a better model.

### 4.2 Logistic Regression Model

We divide the data into training and testing sets with an 80% (614 observations) and 20% (154 observations) split respectively. We begin by fitting a model to the training data subset with all explanatory variables using `glm()` in R using the logit link function (see Appendix F). From the summary we notice all variables apart from BloodPressure and SkinThickness are highly significant - the $p$-value is small enough to suggest there is enough evidence these variables contribute to whether or not a patient has Diabetes (i.e. the coefficient is not equal to 0).

In order to find the most parsimonious model (the one which balances minimal assumptions with the greatest explanatory power), we utilise the `step()` function in R to carry out model selection using backwards elimination. This technique repeatedly identifies the least useful term based on the Akaike information criterion (AIC), until we are left with a model where all covariates are significant. Using the AIC value avoids the problem of overfitting (see Appendix G for summary output).

#### 4.2.1 Model Coefficients

The logistic coefficients obtained give the change in log odds of the outcome for an increase of one unit in the explanatory variable. For example, for every one unit change in BMI, the log odds of having Diabetes increases by 0.0899 and the odds of having Diabetes is multiplied by $exp(0.0899) = 1.09$. This means for every increase of one unit in BMI, we expect to see an increase of $\frac{1.09}{1+1.09} = 52\%$ in the odds of having Diabetes (see Appendix H for full list of probabilities).

#### 4.2.2 Confidence Intervals

We can construct confidence intervals for the coefficient estimates using the computed standard errors (see Appendix I). For example, we are 95% confident the true coefficient estimate for BMI lies in the range 0.060 - 0.119. To convert log odds to probabilities, we exponentiate the log odds to obtain the odds ratio, and divide this value by 1 + the exponentiated log odds.

### 4.2.3 Deviance

Deviance is a measure of goodness of fit of a glm, with large numbers indicating a bad model fit. The null deviance shows how well the response variable is predicted by the fitted model including only the intercept. In our summary output, we obtained a value of 791.41 on 613 degrees of freedom. The residual deviance, a measure of how well the response is predicted by the model when the predictors are included, gives a value of 680.51 on 606 degrees of freedom. The residual deviance has reduced by 110.9 points with a loss of 7 degrees of freedom when including the explanatory variables.

### 4.2.4 AIC

The Akaike Information Criterion (AIC) gives a value to assess the quality of a fitted model compared to another fitted model. A low AIC value indicates the model is better then another model with a larger AIC. The full model obtained an AIC of 696.51 (see Appendix F) and the model after stepwise selection obtained an AIC of 692.56 (see Appendix G) - a slight decrease, indicating the latter model is better after removing non-significant covariates.

# 5 Diagnostics

## 5.1 Model Assumptions and Outliers

Fitting a linear model requires us to make as number of assumptions about our data, which may not all be true. The purpose of regression diagnostics is to verify whether these assumptions are true. In this section we discuss these assumptions, including visual checks for the presence of outliers, and we examine the ability of our model to make predictions from the testing data. We utilise cross-validation to repeatedly create new training and testing sets to evaluate our developed model.

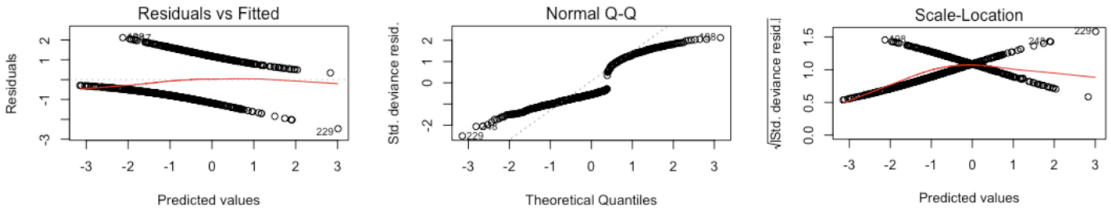


Figure 5: Diagnostic plots for fitted logistic regression model.

The left plot shows residuals against the predicted values, with the dotted line at y=0 indicating our fit line. Any points above or below the fit line have positive residuals or negative residuals respectively. The red line gives an indication of the pattern of the residuals. As our residuals demonstrate a logarithmic pattern we can conclude our model is a good fit.

An assumption we make with linear models is that the residuals are Normally distributed. The middle plot shows a Normal quantile-quantile plot used to assess whether our residuals are Normally distributed. Except for some observations at the upper and lower ends of the plot are further from the fitted diagonal line with all other points lying roughly on the dotted line we can conclude our residuals are normally distributed.

Another assumption of linear models is Homoscedasticity of variance, that is, the variance should be roughly equal for the predicted values. The right plot displays a red line which indicates the residuals have constant variance. As the residuals spread wider from each other the red line increases. We can therefore conclude the data agrees with the Homogeneity of variance assumption.

We can find outliers by looking for points with unusually large residuals. Figure 6 shows the standardised residuals against their index. We would expect around one in twenty residuals outside of the range [2, 2]. We observe indices 157, 499 and 505 as being potential outliers. If these measurements are genuine, it is likely that our model will run into more values like this when making predictions on new data, therefore we leave this small number of outliers in the data to avoid overfitting. None of the other diagnostic plots discussed in this section would result in different conclusions if any of the points were to move, so we can say these points have not affected the overall outcome of our analysis.

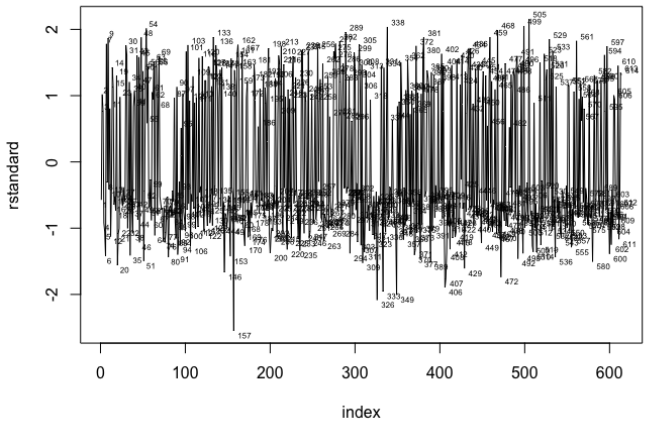


Figure 6: Standardized residuals plotted to find outliers.

### 5.2 Model Testing

We can obtain a confusion matrix which summarises the performance of our model. The accuracy is computed by adding the number of correctly predicted true positives to the number of correctly predicted true negatives, and dividing by the test set size. For our model we obtain an accuracy of 73% which is a reasonable accuracy (see Appendix J). It must be noted there were 32 false negatives (the model predicted no 32 times however the actual result was yes) and 10 false positives (the model predicted yes 10 times however the actual result was no). This ultimately means 32 patients were classified as not having Diabetes when they actually did, and 10 patients who didn't have Diabetes were classified as having Diabetes. Ideally we want to minimise the number of false positives and false negatives.

To assess how well our model predicts we utilise the method of cross validation. This method splits the data into K equal partitions (folds), using 1 fold as the test set and the others as the training set. The model is tested for accuracy - the process repeats K times, using a different fold as the test set on each iteration. We compute the average accuracy across all folds as our test accuracy. Using an 80% train set size and 20% test set size, we obtain a mean accuracy of 70.5% across all folds. We can conclude this predictive model has a reasonable accuracy when predicting on new, unseen data.

## 6 Conclusions

In this paper we successfully implemented a logistic model to predict, with a reasonable level of accuracy, whether a patient has Diabetes or not. We successfully identified the the covariates which significantly contribute to whether a patient has Diabetes through exploratory plots. In the modelling section, we used stepwise model selection on the data to achieve the most parsimonious model. We carried out regression diagnostics on our fitted model to check the assumptions made in the modelling section all hold and found no assumptions were violated. In terms of future work we envisage a more sophisticated technique for handling missing data using a prediction model, to estimate values that will substitute the missing data instead of using median values - this could potentially increase accuracy. Further, models such as Random Forests or Support Vector Machines would enable our logistic model to be compared to different models in terms of their accuracies.

## References

[1] "Diabetes," Oct 2018. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs312/en/

[2] A. J. Dobson, *An Introduction to Generalized Linear Models*, 2nd ed. Chapman & Hall, 2002.

[3] J. M. Cortina, "Interaction, nonlinearity, and multicollinearity: Implications for multiple regression," *Journal of Management*, vol. 19, no. 4, pp. 915 – 922, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/014920639390035L

# A Code

```
# Import dependencies
library(ggplot2)
library(tidyr)
library(dplyr)
library(stargazer)
library(Hmisc)
library(ggcorrplot)
library(caTools)
library(caret)
library(calibrate)

# Read in data
filename = 'diabetes.csv'
data = read.csv(filename, header=T, stringsAsFactors=F)
head(data)
attach(data)

### =========
### Exploratory analysis
### =========

# Summaries
summary(data)
stargazer(data)

# Convert outcome to factor
data$Outcome = factor(Outcome)

# Create Age Category column
data$AgeCat <- ifelse(data$Age < 21, "<21",
                 ifelse((data$Age >= 21) & (data$Age <= 25), "21-25",
                 ifelse((data$Age > 25) & (data$Age <= 30), "25-30",
                 ifelse((data$Age > 30) & (data$Age <= 35), "30-35",
                 ifelse((data$Age > 35) & (data$Age <= 40), "35-40",
                 ifelse((data$Age > 40) & (data$Age <= 50), "40-50",
                 ifelse((data$Age > 50) & (data$Age <= 60), "50-60",
                 ifelse((data$Age > 60) & (data$Age <= 70), "60-70", ">70")))))))

data$AgeCat <- factor(data$AgeCat, levels = c('<21','21-25','25-30','30-35','35-40','
    40-50','50-60','60-70','>70'))
table(data$AgeCat)

# 1. Barplots of categorical data
# Bar plot of age categories
p1 = ggplot(data, aes(x=AgeCat)) +
  geom_bar(fill='#2f4b7c') +
  labs(x="Age Category", y="Number of people")
p1
ggsave('age_cat.png', width=6, height=4, p1)

# 2. Boxplots of explanatory variables against categorical variables
# Boxplots of explanatory variables against categorical variables
bmi_box_p = ggplot(data, aes(x=AgeCat, y=BMI)) +
  geom_boxplot() +
  labs(x="Age Category", y="Body Mass Index")
bmi_box_p
ggsave('bmi_box_p.png', width=6, height=4, bmi_box_p)

insulin_box_p = ggplot(data, aes(x=AgeCat, y=Insulin)) +
  geom_boxplot() +
  labs(x="Age Category", y="Insulin Level")

ggsave('insulin_box_p.png', width=6, height=4, insulin_box_p)

# 3. Investigate the distributions of the numeric explanatory variables
BMI_outcome = ggplot(data, aes(BMI, fill = Outcome)) +
  geom_histogram() +
  theme(legend.position = "bottom") +
  ggtitle("BMI of women against Outcome")

BMI_count = ggplot(data, aes(x = Outcome, y = BMI,fill = Outcome)) +
```

```
  geom_boxplot(binwidth = 5) +
  theme(legend.position = "bottom") +
  ggtitle("BMI of women by Outcome")

ggsave('BMI_outcome_count.png', width=6, height=4, BMI_outcome)
ggsave('BMI_outcome.png', width=6, height=4, BMI_count)

# 4. Correlations between variables.
# Compute correlation matrix
cor_mat <- round(cor(data[1:7]), 2)
ggcorrplot(cor_mat, type = "lower", lab = TRUE)

# Detect missing data
cols_change = colnames(data)[!colnames(data) %in% c("Pregnancies", "Outcome")]
print(apply(data[cols_change], 2, sum))

sapply(data, function(x) sum(x==0))
(sapply(data, function(x) sum(x==0)) / nrow(data)) * 100

# Replace columns containing missing values with NA
data[, 2:5][data[, 2:5] == 0] <- NA

# Impute missing values by calculating medians
data$BloodPressure = impute(data$BloodPressure, median)
data$SkinThickness = impute(data$SkinThickness, median)
data$Insulin = impute(data$Insulin, median)
data$BMI = impute(data$BMI, median)


### =========
### Modelling
### =========
data = data[, !(names(data) %in% c("AgeCat"))]

# Split data into train and test sets
set.seed(100)
n = nrow(data)
train_inds = sample(n, trunc(0.80*n))
train = data[train, ]
test = data[-train, ]

# ZeroR
table(data$Outcome)
acc = 500/length(data$Outcome)

# Fit logistic regression model using all covariates
fit = glm(Outcome ~ ., data = train, family = binomial(link = "logit")) # probit
summary(fit)

# Choose best logistic model by using step()
fit2 = step(fit)
summary(fit2)

## Confidence intervals
confint.default(fit2)

logit2prob <- function(logit){
  odds <- exp(logit)
  prob <- odds / (1 + odds)
  return(prob)
}

logit2prob(coef(fit2))

### ============
### Diagnostics
### ============

# Apply the model to the testing sample
test_pred = predict(fit2, test, type = "response")
pred_test = as.data.frame(cbind(test$Outcome, test_pred))
colnames(pred_test) <- c("Original", "Test_pred")
pred_test$Outcome <- ifelse(pred_test$Test_pred > 0.5, 1, 0)
```

```
error <- mean(pred_test$Outcome != test$Outcome)
print(paste('Test Data Accuracy', round(1-error,2)*100,'%'))

# Residual plots
summary(residuals(fit2))
par(mfrow=c(2,2))
plot(fit2)

# Testing the Model
glm_prob = predict(fit2, newdata = test, type = "response")
glm_pred = ifelse(glm_prob > 0.5, 1, 0)
print("Confusion Matrix")
table(Predicted = glm_pred, Actual = test$Outcome)

# Cross validation
accuracies = matrix(nrow=100, ncol=1)

for (i in 1:100) {
  train_ind = sample.split(data$Pregnancies, SplitRatio = 0.8)
  test_ind = !train_ind

  cur_fit = glm(Outcome ~ ., data = data[train_ind,], family="binomial")

  prediction = predict(cur_fit, data[test_ind,], family="binomial")

  probs = exp(prediction)/ (1+exp(prediction))
  outcomes = probs > 0.5

  cm = table(Actual = data$Outcome[test_ind], Predicted = outcomes)
  accuracies[i] = sum(diag(cm))/sum(test_ind)
}

mean(accuracies)

# Confusion matrix
confusionMatrix(
  factor(glm_pred, levels = 0:1),
  factor(test$Outcome, levels = 0:1)
)

# Outliers
rstandard = rstandard(fit2)
index = seq(1, length(rstandard), 1)
plot(index, rstandard,'l')
textxy(index,rstandard, index)
```

# B Proof that the Binomial Distribution belongs to the Exponential Family

Considering a single random variable Y whose probability distribution depends on a single parameter $\theta$, the distribution belongs to the exponential family if it can be written in the following form:

$$f(y;\theta) = s(y)t(\theta)e^{a(y)b(\theta)} \tag{2}$$

where a, b, s and t are known functions. We can rewrite equation (2) as follows:

$$f(y;\theta) = exp[a(y)b(\theta) + c(\theta) + d(y)] \tag{3}$$

where $s(y) = exp\ d(y)$ and $t(\theta) = exp\ c(\theta)$.

Consider a series of trials (binary events) each with only two possible outcomes: success or failure. Let the random variable Y be the number of successes in $n$ independent trials in with probability of success $p$. Then Y is distributed binomially with probability density function:

$$f(y;p) = \binom{n}{y} p^y (1-p)^{n-y} \tag{4}$$

where y takes the values 0, 1, ... , $n$. Assuming $p$ is known, the probability function can be rewritten as:

$$f(y;p) = \exp\left[ylogp - ylog(1-p) + nlog(1-p) + \log\binom{\mathrm{n}}{\mathrm{y}}\right] \tag{5}$$

which is in the same form as equation (3) with $b(p) = \log(p)$ - $\log(1$ - $p) = \log[\frac{p}{1-p}]$.

# C Summary statistics

Table 2: Summary statistics for diabetes data.

| Statistic | N | Mean | St. Dev. | Min | Pctl(25) | Pctl(50) | Pctl(75) | Max |
|---|---|---|---|---|---|---|---|---|
| Pregnancies | 768 | 3.845 | 3.370 | 0 | 1 | 3.000 | 6 | 17 |
| BloodPressure | 768 | 69.105 | 19.356 | 0 | 62 | 72.00 | 80 | 122 |
| SkinThickness | 768 | 20.536 | 15.952 | 0 | 0 | 23.00 | 32 | 99 |
| Insulin | 768 | 79.799 | 115.244 | 0 | 0 | 30.5 | 127.2 | 846 |
| BMI | 768 | 31.993 | 7.884 | 0.000 | 27.300 | 32.00 | 36.600 | 67.100 |
| DiabetesPedigreeFunction | 768 | 0.472 | 0.331 | 0.078 | 0.244 | 0.3725 | 0.626 | 2.420 |
| Age | 768 | 33.241 | 11.760 | 21 | 24 | 29.00 | 41 | 81 |
| Outcome | 768 | 0.349 | 0.477 | 0 | 0 | 0.000 | 1 | 1 |

# D Bar plot of age categories

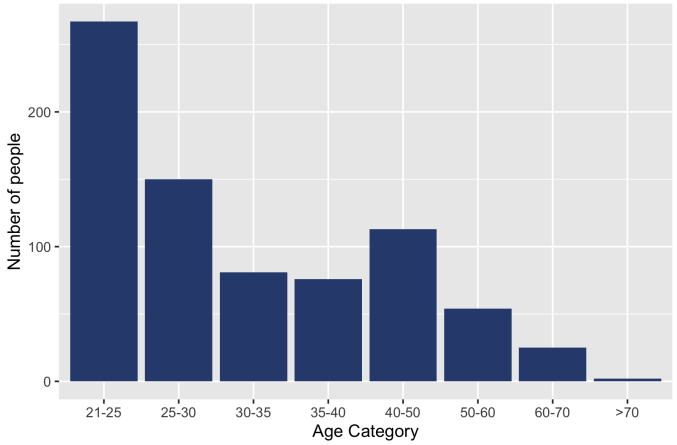


Figure 7: Bar chart of age categories.

# E Correlogram of covariates

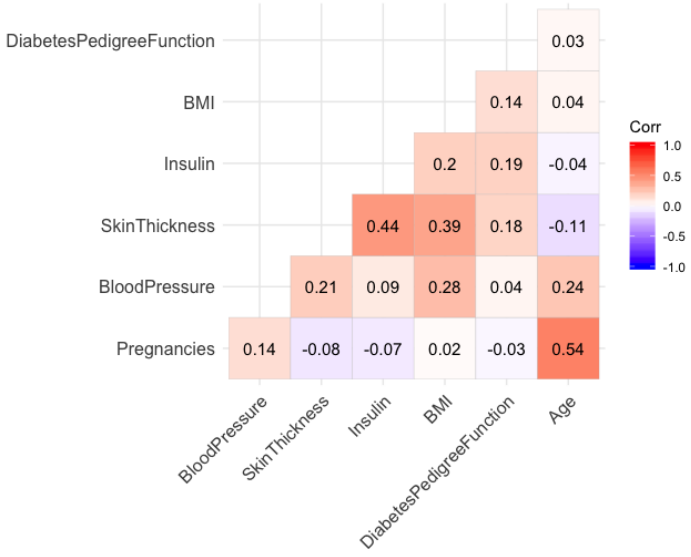


Figure 8: Correlations between explanatory variables.

# F Summary output for glm

```
Call:
glm(formula = Outcome ~ ., family = binomial(link = "logit"),
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5044  -0.8436  -0.5635   1.0344   2.1123

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)              -5.9341192  0.7426584  -7.990 1.35e-15 ***
Pregnancies               0.0934670  0.0324440   2.881  0.00397 **
BloodPressure             0.0019817  0.0089308   0.222  0.82440
SkinThickness             0.0008337  0.0135457   0.062  0.95092
Insulin                   0.0031574  0.0011158   2.830  0.00466 **
BMI                       0.0882379  0.0183520   4.808 1.52e-06 ***
DiabetesPedigreeFunction  0.8723488  0.2948632   2.958  0.00309 **
Age                       0.0277867  0.0096667   2.874  0.00405 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 791.41  on 613  degrees of freedom
Residual deviance: 680.51  on 606  degrees of freedom
AIC: 696.51

Number of Fisher Scoring iterations: 4
```

Figure 9: Output of the glm() model summary. L-R: variable name, regression coefficient, standard error, $z$-value, $p$-value

# G Summary output for glm after backwards elimination

```
Call:
glm(formula = Outcome ~ Pregnancies + Insulin + BMI + DiabetesPedigreeFunction +
    Age, family = binomial(link = "logit"), data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5001  -0.8425  -0.5656   1.0368   2.1075

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)              -5.841989   0.622847  -9.379  < 2e-16 ***
Pregnancies               0.093591   0.032419   2.887  0.00389 **
Insulin                   0.003141   0.001111   2.827  0.00470 **
BMI                       0.089914   0.015031   5.982 2.21e-09 ***
DiabetesPedigreeFunction  0.870658   0.294463   2.957  0.00311 **
Age                       0.028485   0.009177   3.104  0.00191 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 791.41  on 613  degrees of freedom
Residual deviance: 680.56  on 608  degrees of freedom
AIC: 692.56

Number of Fisher Scoring iterations: 4
```

Figure 10: Output of the glm() model summary after backwards elimination. L-R: variable name, regression coefficient, standard error, $z$-value, $p$-value

# H Log odds as probabilities

```
          (Intercept)              Pregnancies                  Insulin
            0.00289466                0.52338064                0.50078536
                  BMI DiabetesPedigreeFunction                      Age
            0.52246336                0.70488261                0.50712083
```

Figure 11: Model coefficient estimates converted from log odds scale to probability scale

# I Confidence Intervals

```
                              2.5 %       97.5 %
(Intercept)             -7.0627455399 -4.621232003
Pregnancies              0.0300509476  0.157130661
Insulin                  0.0009636639  0.005319215
BMI                      0.0604539248  0.119374027
DiabetesPedigreeFunction 0.2935219875  1.447794189
Age                      0.0104985974  0.046471896
```

Figure 12: Constructed confidence intervals for logistic model after backwards selection.

# J Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 88 32
         1 10 24

               Accuracy : 0.7273
                 95% CI : (0.6497, 0.7958)
    No Information Rate : 0.6364
    P-Value [Acc > NIR] : 0.010720

                  Kappa : 0.3565

 Mcnemar's Test P-Value : 0.001194

            Sensitivity : 0.8980
            Specificity : 0.4286
         Pos Pred Value : 0.7333
         Neg Pred Value : 0.7059
             Prevalence : 0.6364
         Detection Rate : 0.5714
   Detection Prevalence : 0.7792
      Balanced Accuracy : 0.6633

       'Positive' Class : 0
```

Figure 13: Confusion matrix and statistics for the fitted logistic regression model.