Generalized Linear Models (GLM) and Iteratively Reweighted Least Squares (IRLS)

John Xiaoyu Zhang

2023-04-29

Contents

Part I: Introduction

In this project, I will explore the application of Generalized Linear Models (GLM) and leverage the Iteratively Reweighted Least Squares (IRLS) algorithm to fit a logistic regression model.

1. Linear Models (LM): Linear Models are statistical models that assume a linear relationship between the response variable and the predictors. The goal of LM is to minimize the sum of the squared residuals, thereby providing the 'best fit' to the data. These models, however, rely on several assumptions including homoscedasticity (constant variance of errors) and normality of errors.
2. Generalized Linear Models (GLM): Generalized Linear Models extend the linear model framework by allowing for response variables that have error distribution models other than a normal distribution. GLMs can accommodate a host of different types of data including binary response variables, count response variables, non-negative response variables, and more. This enables them to model non-linear relationships between the response and predictor variables, handle non-constant variances of the response variable, and cater to non-normal distributions.

Why use GLM instead of LM? The primary reason lies in the flexibility that GLMs offer:

- Response Variable Distribution: While Linear Models (LMs) assume that the response variable follows a normal (Gaussian) distribution, this assumption might not always hold, especially for non-continuous or non-negative data. GLMs, on the other hand, can handle a variety of distributions for the response variable, such as binomial, Poisson, and gamma distributions. This added flexibility makes GLMs more versatile and applicable to a broader range of problems.

Iteratively Reweighted Least Squares(IRLS): The IRLS algorithm is applied to update the weight of linear regression every iteration based on the residual from the previous iteration. This method is necessary because the relationship between predictor variables and response variables in GLMs is defined by a link function, which may be non-linear, making it difficult to optimize using traditional methods like ordinary least squares. The core concept behind Iteratively Reweighted Least Squares (IRLS) is the utilization of the Newton-Raphson update rule for iterative refinement of the coefficients. This is achieved by employing the first and second derivatives of the log-likelihood function, which respectively represent the direction and curvature of the function. The goal is to find the minimum of the function, thus optimizing the model parameters. The iterative process continues until the changes in the coefficients between iterations fall below a predefined threshold, indicating that the model parameters have converged to their optimal values.

Part II: Algorithm and Derivation of IRLS and GLM

Algorithm

1. Initialize the model parameters (beta) to zero.
2. Calculate the predicted probabilities (p) using the current estimates of the model parameters.
3. Calculate the correction term (z) using the predicted probabilities (p).
4. Calculate the weights (W) using the predicted probabilities (p).
5. Update the model parameters (beta) using the correction term (z) and weights (W).
6. Repeat steps 2-5 until convergence.

Derivation

1. Define the likelihood function for logistic regression:

$$L(\beta) = \prod_i [p_i^{y_i} \cdot (1 - p_i)^{(1-y_i)}]$$

where

$$p_i = P(y_i = 1|X_i) = \frac{e^{X_i\beta}}{1 + e^{X_i\beta}}$$

is the predicted probability of the ith observation belonging to class 1.

2. Take the logarithm of the likelihood function to obtain the log-likelihood:

$$l(\beta) = \sum_i [y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)]$$

3. Take the first derivative (gradient) of the log-likelihood function with respect to beta:

$$\frac{dl(\beta)}{d\beta} = X^T \cdot (y - p)$$

where X is the matrix of predictor variables, y is the vector of actual responses, and p is the vector of predicted probabilities.

4. Take the second derivative (Hessian) of the log-likelihood function with respect to beta:

$$\frac{d^2l(\beta)}{d\beta^2} = -X^T \cdot W \cdot X$$

where W is a diagonal matrix with the variances

$$V_i = p_i \cdot (1 - p_i)$$

on the diagonal.

5. Use the Newton-Raphson update rule to iteratively update the coefficients:

$$\beta_{new} = \beta_{old} - (\text{Hessian})^{-1} \cdot \text{Gradient}$$

Substitute the expressions for the Hessian and Gradient:

$$\beta_{new} = \beta_{old} + (X^T \cdot W \cdot X)^{-1} \cdot X^T \cdot (y - p)$$

Now, we can relate this update rule to the weighted linear regression problem:

Define

$$z = X \cdot \beta_{old} + W^{-1} \cdot (y - p)$$

Then, the update rule can be written as:

$$\beta_{new} = (X^T \cdot W \cdot X)^{-1} \cdot X^T \cdot W \cdot z$$

This equation corresponds to the normal equation for a weighted linear regression problem with weights W, predictors X, and target variable z.

Finally, we can express z as a function of the correction term:

$$z_i = X_i \cdot \beta_{old} + \frac{y_i - p_i}{V_i}$$

So, the term

$$(y_i - p_i)/V_i$$

is the correction term that adjusts the expected response based on the difference between the actual response (y_i) and the predicted probability (p_i). The variance (V_i) is used to weight this correction term, giving more importance to observations with higher certainty (smaller variance).

Part III: Advantages and Limitations of GLM and IRLS

Advantages

1. Applicability to GLMs: IRLS is specifically designed for fitting GLMs, making it a natural choice for logistic regression, Poisson regression, and other members of the GLM family. The algorithm can handle various link functions and distributions within the GLM framework.
2. Adaptive Weights: One of the key features of IRLS is its adaptive weighting scheme, which updates the weights at each iteration based on the current estimates of the model parameters. This allows the algorithm to focus on fitting the data points that are more challenging or have higher residuals.
3. Robustness: IRLS has been shown to be robust in many situations, producing stable parameter estimates even when the underlying assumptions of the GLM are not perfectly met.
4. Efficiency: The IRLS algorithm is generally computationally efficient, as it leverages matrix operations and can take advantage of optimized linear algebra libraries.

Limitation

1. Convergence: IRLS does not always guarantee convergence to the global minimum, especially for poorly conditioned or ill-posed problems. If the initial estimates of the parameters are far from the true values, IRLS may not converge or may converge to a local minimum.
2. Outliers and Influential Points: IRLS can be sensitive to outliers or influential points, which can lead to biased estimates of the model parameters.
3. Multicollinearity: Like other linear regression techniques, IRLS can be affected by multicollinearity, which occurs when predictor variables are highly correlated. This can lead to unstable estimates and difficulties in interpreting the results.
4. Assumptions: Although GLMs are more flexible than traditional linear models, they still rely on certain assumptions, such as the appropriate choice of link function and error distribution. If these assumptions are not met, the model may produce inaccurate or biased results.

Part IV: Dataset Interpretation

Data Explaination

For the implementation of the IRLS algorithm and GLM in R, I have used the "PimaIndiansDiabetes2.csv" dataset. This dataset, containing 768 observations and 9 variables, characterizes the relationship between a person's health condition and the occurrence of diabetes. By scrutinizing these health-related variables, we may be able to assess the risk of diabetes in individuals. Therefore, creating a model that can accurately predict based on people's health conditions is of significant importance.

The variables are:

1. Pregnancies: Number of times pregnant
2. Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. BloodPressure: Diastolic blood pressure (mm Hg)
4. SkinThickness: Triceps skin fold thickness (mm)

5. Insulin: 2-Hour serum insulin (mu U/ml)
6. BMI: Body mass index (weight in kg/(height in m)^2)
7. DiabetesPedigreeFunction: Diabetes pedigree function
8. Age: Age (years)
9. Outcome: Class variable (0 or 1) 268 of 768 are 1, the others are 0

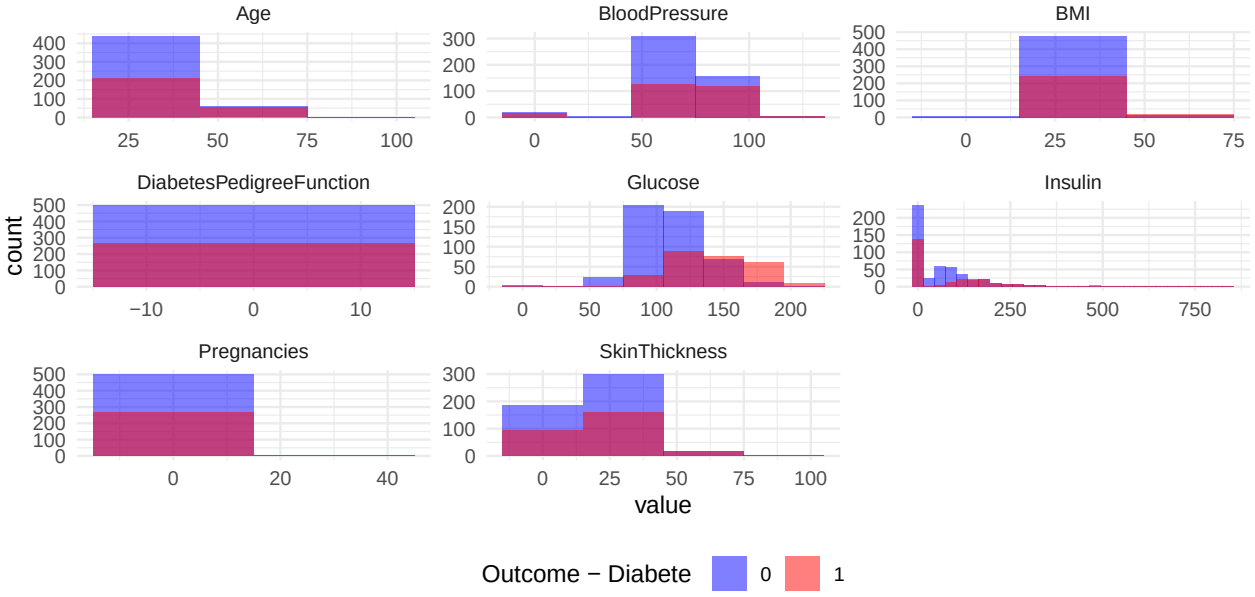
The data suggest a possible relationship between the health-related variables and the diabetes outcome.

Data Visualization

Histogram

I have started data exploration with histograms to visualize the frequency and distribution of each variable concerning the diabetes outcome. From these histograms:

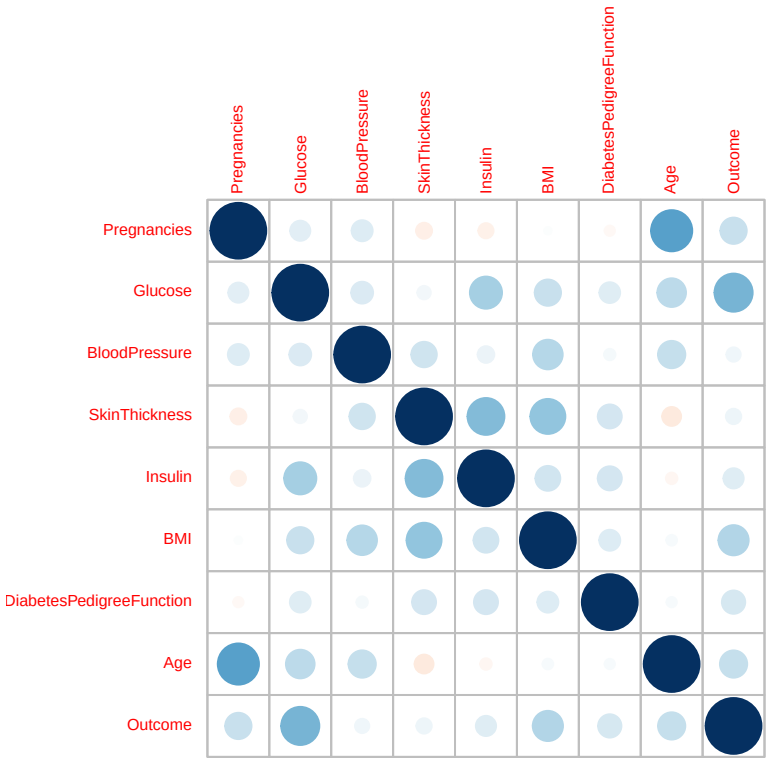
1. It's apparent that the proportion of diabetes cases significantly increases in individuals over the age of 45.
2. High blood pressure doesn't seem to have a clear correlation with diabetes, as a significant proportion of individuals with high blood pressure do not have diabetes.
3. Glucose level appears to be a strong indicator of diabetes, with the proportion of diabetes cases increasing with glucose levels.
4. The relationship between the remaining variables and diabetes isn't immediately apparent.



Correlation Plot

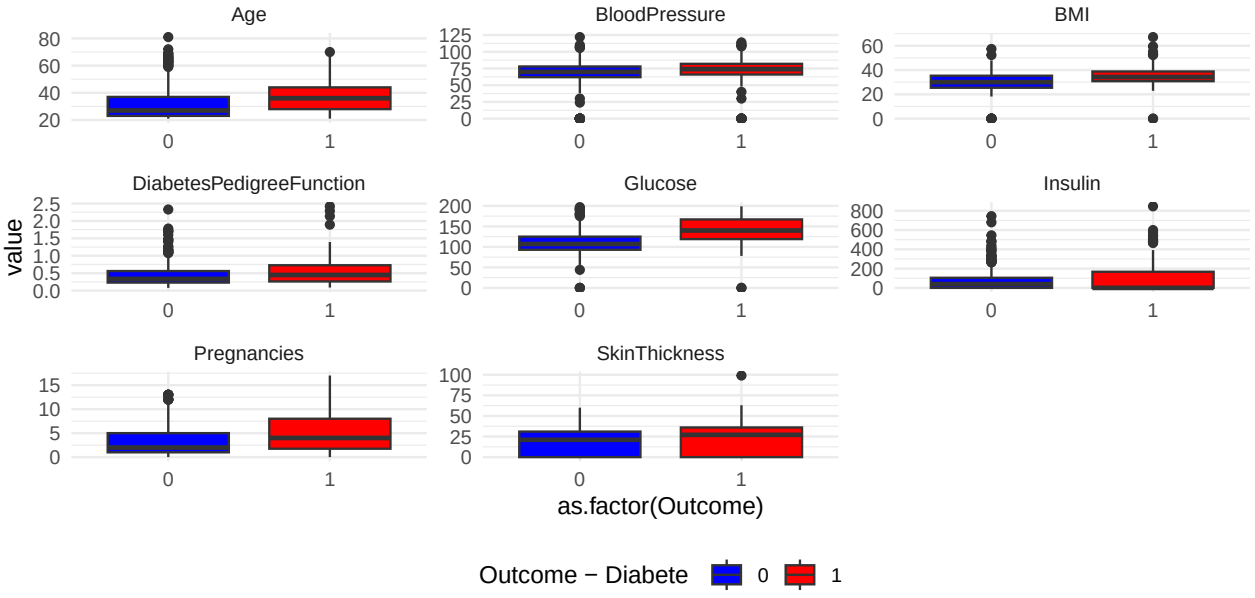
The correlation plot was used to examine the correlation between each pair of variables.

1. The interpretation aligns with the findings from the histograms. Age and glucose level emerge as the two most important variables with a strong correlation to diabetes.
2. However, it's also noteworthy that insulin and skin thickness, as well as BMI and skin thickness, exhibit strong correlations. These correlations will be an important consideration during variable selection.



Box Plot

Box plots provide a clear visualization of how different variables may influence the diabetes outcome. For instance, it's evident from the box plots that age and glucose level have different medians for the diabetes and non-diabetes groups, suggesting these factors may be important predictors in our model



Summary of Data Visualization

From the above visualizations, it's evident that certain variables demonstrate strong relationships with the diabetes outcome. The outcome variable is binary, which complicates interpretation with a linear model.

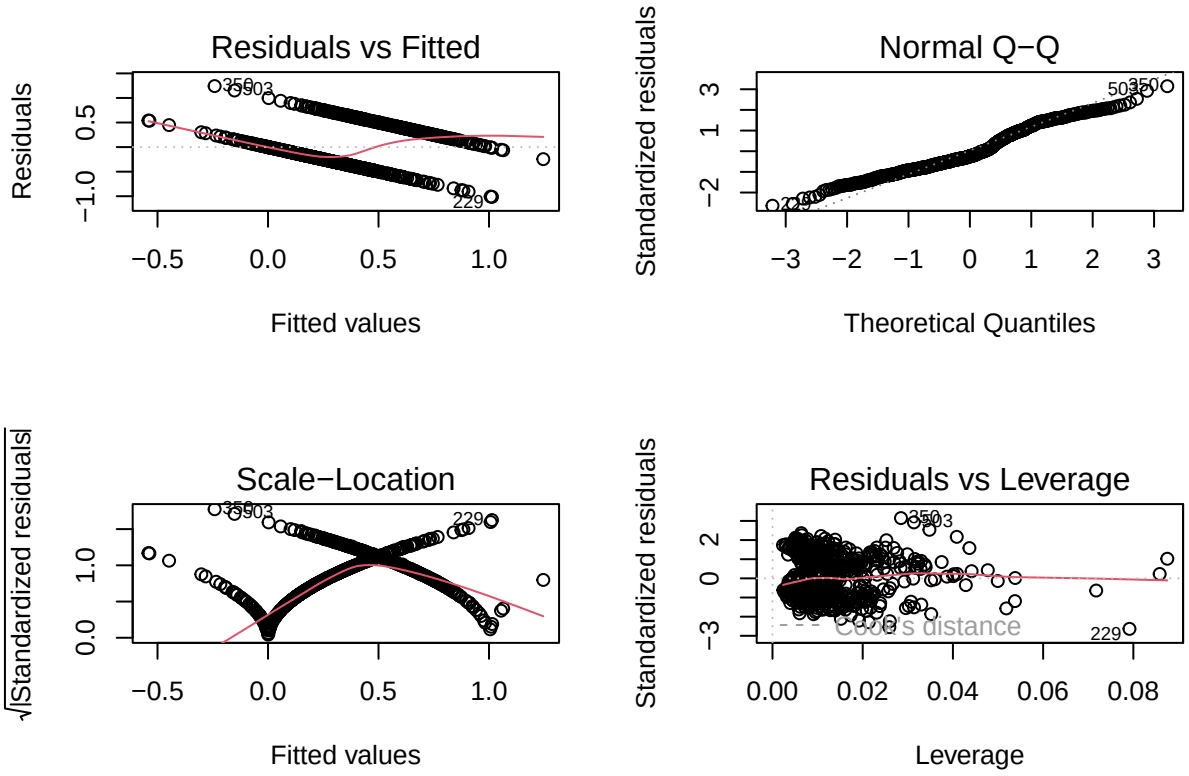
Given the limited number of variables, I will initially explore the full model and refine it based on performance

---

Part V: Implementation of GLM and IRLS Algorithm

Fitting a Linear Model (LM)

Before fitting our chosen model, it's instructive to attempt a standard linear model fit to highlight the need for a generalized linear model.



The result "Multiple R-squared: 0.3033, Adjusted R-squared: 0.2959" shows that only 30% of the variance can be explained by the model, which menas the model behave poorly on the data. So, GLM might be a better choice to fit the model.

Insights from the diagnostic plots (plot(lm_fit)) include:

1. The 'Residuals vs Fitted' plot indicate the presence of heteroscedasticity, non-linear pattern, and a non-constant variance.
2. The 'Normal Q-Q' plot corroborates the non-normal distribution of residuals.
3. The 'Scale-Location' plot illustrates a trend while the residuals should be randomly distributed around the horizontal line, with no apparent patterns or trends.
4. The 'Residuals vs Leverage' plot seems fine, without too many outliers.

In conclusion, the linear model is inadequate for our data, prompting the need for a weighted least square method, such as that used in a generalized linear model (GLM).

Fitting Weighted LM Model

The weighted least square method is used to fit the model. By considering the variance, the function is defined as follows:

```r
fit_wlm <- function(X,y,obs_var){
  #obs_var is a vector of variances for each observation
  obs_var <- as.vector(obs_var)
  X <- cbind(1,X)       #Add intercept to the X matrix
  D <- diag(1/obs_var)  #create the D matrix
  lhs <- t(X)%*%D%*%X
  rhs <- t(X)%*%D%*%y
  return(solve(lhs,rhs))
}
```

IRLS Function

1.The process is initial guess β_0 and β_1 and then compute the probability for sigmoid function. 2.Compute the variances for each observation and then use the variances to compute the new β_0 and β_1. 3.Compute Expected Response and then compute the new β_0 and β_1. 4.Repeat step 2 and 3 until the β converge.

```r
#Design the IRLS function with iteration times and break the loop when the beta converge
IRLS <- function(x, y, max_iter = 100, conv_eps = 1e-6) {
# make sure x is a matrix
  x <- matrix(x, ncol = ncol(x))
  y <- matrix(y, ncol = 1) # Ensure y is a column matrix
  ones <- matrix(1, nrow = nrow(x), ncol = 1)
  x <- cbind(ones, x) # add the ones to the x matrix
  n <- ncol(x)
  beta <- matrix(0, nrow = n, ncol = 1)
  for (i in 1:max_iter) {
    # Compute predicted probabilities (p) and variances (V)
    p <- matrix( exp(x %*% beta) / (1 + exp(x %*% beta)), ncol = 1)
    V <- p * (1 - p)
    # Compute the expected response (z)
    z <- x %*% beta + (y - p) / V
    # Update beta using weighted linear regression
    beta_new <- fit_wlm(x[,2:ncol(x)], z, 1/V)
    # Check for convergence
    if (sum((beta_new - beta)^2) <= conv_eps) {
      break
    }
    # Update beta for the next iteration
    beta <- beta_new
  }
  return(beta)
}
```

Model Fitting and Comparison with GLM

The process of fitting the model is as follows: 1. Data Splitting: To ensure a fair evaluation of the model, I split the data into training and testing datasets. 2. Model Fitting with IRLS: I proceed to fit our model using the Iteratively Reweighted Least Squares (IRLS) algorithm. This step involves the inclusion of all variables into the model. 3. Model Fitting with Built-in glm Function: For the sake of comparison, I also fit a model using R's built-in glm function. This is performed on the same set of variables as the IRLS model. 4. Model Comparison: The performance of the two models is compared using relative bias as a metric. This allows us to assess the accuracy of the IRLS function.

Upon fitting the model using 80% of the data as training data, the results reveal that the IRLS model yields comparable results to the glm function model. The beta coefficients for the two models are displayed below:

```
##                                 IRLS           GLM Relative Bias Check
## (Intercept)              -8.2129302882 -8.2130000699 -0.0008496500  Good
## Pregnancies               0.1183919936  0.1183928971 -0.0007631288  Good
## Glucose                   0.0352635124  0.0352637537 -0.0006840851  Good
## BloodPressure            -0.0130874503 -0.0130875394 -0.0006810306  Good
## SkinThickness            -0.0009585248 -0.0009585095  0.0015999531  Good
## Insulin                  -0.0009090975 -0.0009091094 -0.0013059158  Good
## BMI                       0.0860663621  0.0860672790 -0.0010653563  Good
## DiabetesPedigreeFunction  0.7810691877  0.7810790475 -0.0012623303  Good
## Age                       0.0152286591  0.0152287876 -0.0008435408  Good
```

A relative bias less than 50% for all coefficients signifies that the performance of the model is acceptable, and that the IRLS function yields results similar to the glm function. Therefore, after verifying the accuracy of our custom-built model, I decide to continue using the IRLS function for subsequent model fitting, predictions, and interpretations.

Part VI: Model Performance

Predict the Result Using IRLS Function

1. First, I predict the result using IRLS function.
2. And then convert probabilities from model IRLS and GLM function to 0 and 1.
3. Present yhe result using a dataframe.
4. And then compute and plot ROC and confusion matrix for the model to check for model performance.

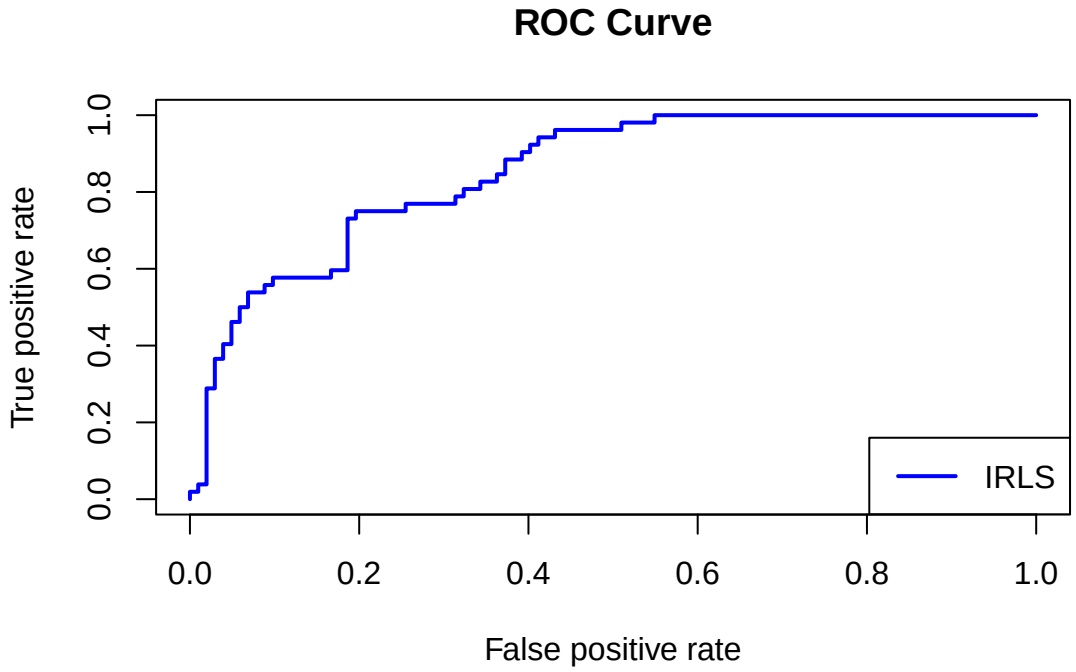
Present the Predicted Result and Result Summary

The table below displays a portion of the data frame containing the predicted results from the IRLS function. In addition to the predicted outcomes, I have added a 'real_value' column which contains the actual outcomes from the dataset. The 'predicted_result' column indicates whether the model's prediction was accurate ('good') or not ('bad').

```
##        y_pred real_value y_pred_result Check_result
## 1  0.70756143          1             1         Good
## 3  0.79695546          1             1         Good
## 9  0.74524723          1             1         Good
## 17 0.36330968          1             0          Bad
## 22 0.31803532          0             0         Good
## 27 0.74106007          1             1         Good
## 28 0.04892753          0             0         Good
## 32 0.56777723          1             1         Good
## 42 0.68427046          0             1          Bad
## 43 0.11927438          0             0         Good
```

```
## [1] "The accuracy of the previous model is:  79.2207792207792 %"
```

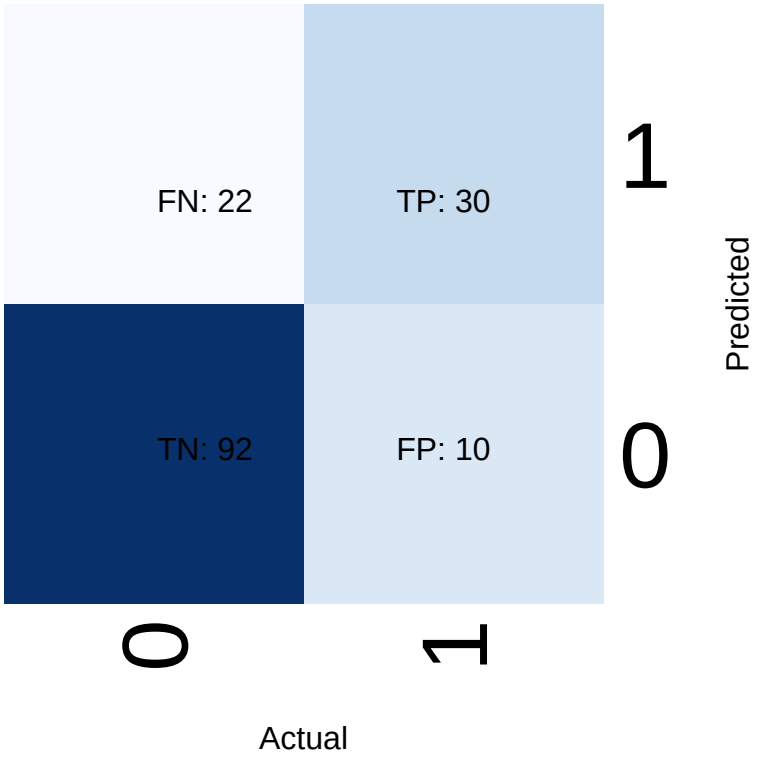
ROC Curve




```
## [1] "AUC for the IRLS model:  0.851"
```

The ROC curve for the predicted result is close to the edge and with AUC score close to 1, which means the model performance is good.

Confusion Matrix - IRLS model



The confusion matrix displayed above provides a visual overview of the model's performance. The model

correctly identified 30 instances as true positives and 92 instances as true negatives. These results imply that our IRLS model demonstrates a satisfactory level of accuracy in predicting diabetes outcomes.

Part VII: Model Improvement

The initial model performs well, but improvements can still be made. For example, correlated variables such as skin thickness and BMI could be managed more effectively. Lasso regression, a method for variable selection, is ideal for this.

Both Lasso and Ridge regression are regularization methods that prevent overfitting and manage collinearity and high dimensionality. They work by adding a penalty term to the ordinary least squares (OLS) loss function, which shrinks regression coefficients towards zero. The difference between the two is that Ridge regression shrinks coefficients of correlated variables towards each other, while Lasso pushes them towards zero.

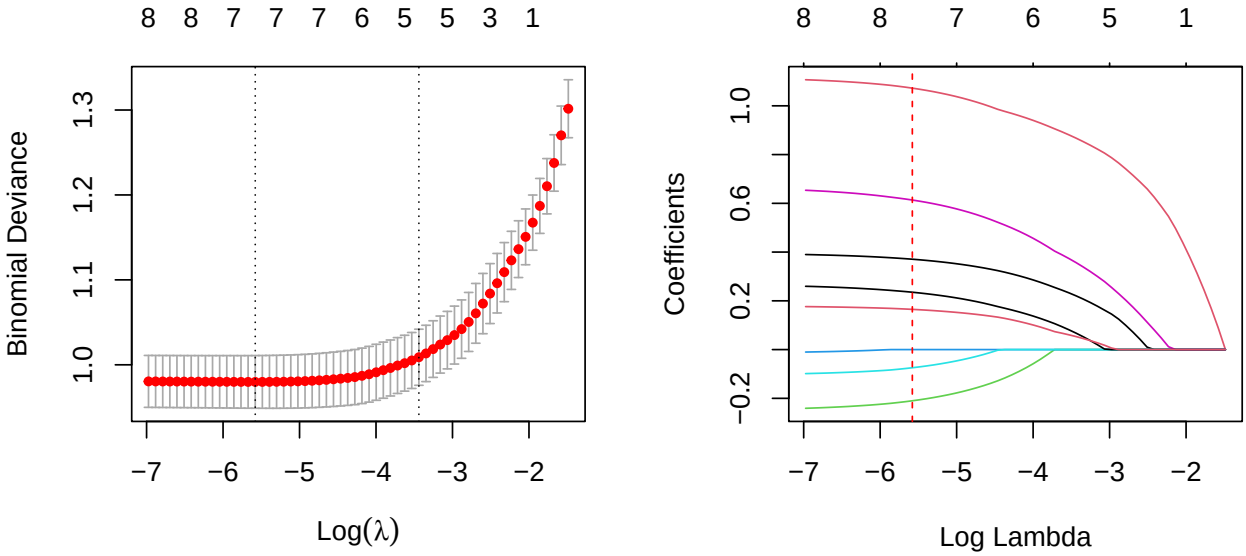
The penalty term is controlled by a parameter, lambda, which impacts the amount of shrinkage. When lambda is 0, no penalty is applied, and the result is the same as a plain OLS regression. As lambda increases, coefficients shrink towards zero. Selecting the optimal lambda is key in Lasso regression.

Lasso Regression

Apply Lasso for variable selection and compare the result with the previous model:

1. Use glmnet to select the best lambda
2. Use cross-validation to find the optimal value of the regularization parameter (lambda)
3. Refit the Lasso model with the optimal lambda value and extract the coefficients
4. Compare the result with the previous model

```
## [1] "Optimal lambda:  0.0037751095721621"
```




```
## [1] "Pregnancies"              "Glucose"
## [3] "BloodPressure"            "Insulin"
## [5] "BMI"                      "DiabetesPedigreeFunction"
## [7] "Age"
```

The Lasso regression has selected a subset of variables for the model, notably excluding the “SkinThickness” variable, which aligns with our previous data analysis suggesting its lesser importance.

Model Performance

Now refit the model using these selected variables and contrast the outcome with previous model's performance.

```
## [1] "The accuracy of the new model is:  78.0130293159609 %"
```

```
## [1] "The accuracy of the previous model is:  79.2207792207792 %"
```

The result shows that the accuracy of the new model is has slight drop. Though the accuracy of this refitted model has slightly decreased on our training data, it's crucial to note that this doesn't necessarily imply poorer performance. In fact, the model's ability to generalize, that is its performance on unseen data, could potentially improve. This is a known trade-off when using regularization techniques like Lasso, where model simplicity is favored to prevent overfitting and improve model robustness on new data.

Part VIII: Conclusion

The project began with data exploration, providing valuable insights into the structure and relationships within the dataset. Interesting patterns and correlations were uncovered, notably the correlation between "SkinThickness" and "BMI", and the distribution of "Outcome".

The next stage involved delving into the derivation of the Iteratively Reweighted Least Squares (IRLS) method, a fundamental procedure for logistic regression. Understanding the underlying mathematics brought an appreciation for the logic behind model fitting.

A logistic regression model was then fitted using both a custom IRLS function and R's in-built glm function. Comparing these results validated the performance of the custom function. The Receiver Operating Characteristic (ROC) curve and confusion matrix were used to evaluate the model's performance. The model exhibited good predictive power, with the ROC curve close to the edge and an AUC score close to 1.

The final stage was model improvement. Lasso regression, a regularization technique, was applied to select a more relevant set of variables and reduce potential overfitting. While the accuracy on the training data dropped slightly, this simplification might potentially lead to better generalization when applying the model to unseen data.

Throughout this project, the power of thoughtful data analysis and model selection became evident. The complexity of the models was balanced against their interpretability and generalizability, showing that sometimes, a simpler model can yield better results. This journey underscored the iterative and multifaceted nature of data science, where exploration, mathematical understanding, model building, and continual refinement all play essential roles.