Logistic Regression

Azeez Olalekan, Baki

08/01/2022

# Regression Analysis

## What is regression?

Regression is a model or statistical analysis conducted to determine the effect of a variable(s)(predictor/independent)on another variable(s)(dependent/outcome).

Mathematically, y = ax+bx+c.

#Example of Regression

In this analysis, we are going to be performing both linear regression and logistic regression on a dataset and use the result to predict a new set of input data. So let get started!

# The Data: Binary data

Let get familiar with the data. The data name is Binary data with 4 variables (admit,gre,gpa,rank)and 400 observations. Admit: whether the student was admitted or rejected. 1:admitted, 0:rejected. gre: test scores gpa: grade rank: the rank of the school the student graduated from. 1:highest ranked school, 2:high ranked, 3: moderate ranked, 4:low ranked.

# Loading Data

Let start by importing the data.

Next, let check the structure of the data.

str(data)

## 'data.frame': 400 obs. of 4 variables:  
## $ admit: int 0 1 1 1 0 1 1 0 1 0 ...  
## $ gre : int 380 660 800 640 520 760 560 400 540 700 ...  
## $ gpa : num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...  
## $ rank : int 3 3 1 4 4 2 1 2 3 2 ...

## Exploratory Analysis

Before we delve into regression, let get familiar with the data using charts and graphs.

#Summary of the data

summary(data)

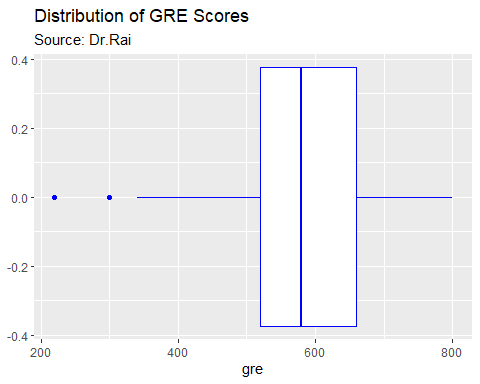
## admit gre gpa rank   
## Min. :0.0000 Min. :220.0 Min. :2.260 Min. :1.000   
## 1st Qu.:0.0000 1st Qu.:520.0 1st Qu.:3.130 1st Qu.:2.000   
## Median :0.0000 Median :580.0 Median :3.395 Median :2.000   
## Mean :0.3175 Mean :587.7 Mean :3.390 Mean :2.485   
## 3rd Qu.:1.0000 3rd Qu.:660.0 3rd Qu.:3.670 3rd Qu.:3.000   
## Max. :1.0000 Max. :800.0 Max. :4.000 Max. :4.000

# Inteepretation

From summary statistics, the minimum score in the gre was 220 while the maximum was 800, with mean of 587.7. The highest gpa was 4.00 and lowest was 2.26, mean= 3.34.

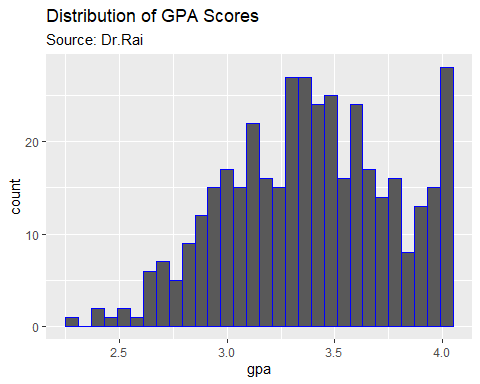
# Boxplot for GRE

library(ggplot2)  
ggplot(data=data, aes(gre))+  
 geom\_boxplot(colour="blue")+  
 ggtitle("Distribution of GRE Scores", 'Source: Dr.Rai')

 #Histogram for GPA

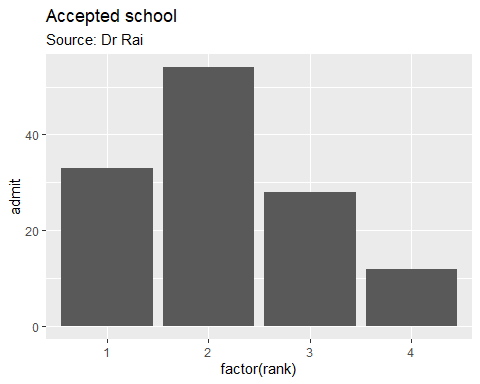
ggplot(data=data, aes(gpa))+  
 geom\_histogram(colour="blue")+  
 ggtitle("Distribution of GPA Scores", 'Source: Dr.Rai')

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



##Barchart for Rank

ggplot(data=data, aes(x= factor(rank), y = admit))+  
 geom\_col()+  
 ggtitle("Accepted school", "Source: Dr Rai")

 From the barchart, we noticed that most of the students admitted were from highest, high and moderate ranked schools.

Lastly, let check if there is any correlation among the variables.

#Correlation

cor <- cor(data[,2:3], method = c('pearson'))  
cor

## gre gpa  
## gre 1.0000000 0.3842659  
## gpa 0.3842659 1.0000000

#corrplot(cor, method = c("ellipse"), type = "upper")

The correlation plot shows that there is a correlation among the variables. Moderately positive correlation was observed between gpa and gre, i.e the higher one of the variables the higher the second one. But gpa and gre have weak negative correlation with school rank.

## Linear Regression

Let partition the data into 80% train data and 20% validate data.

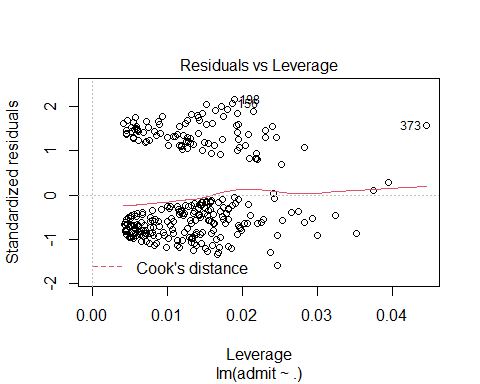
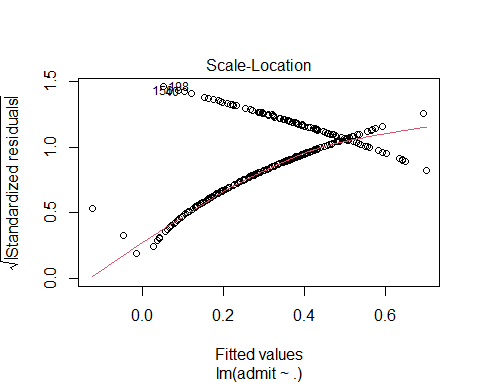
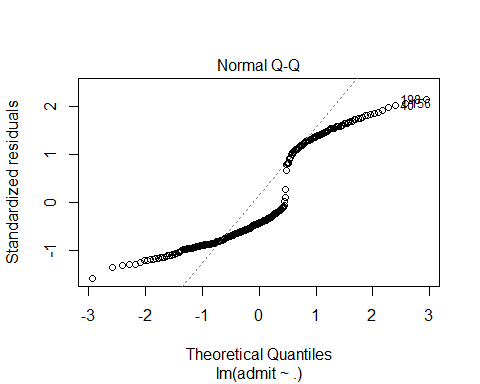
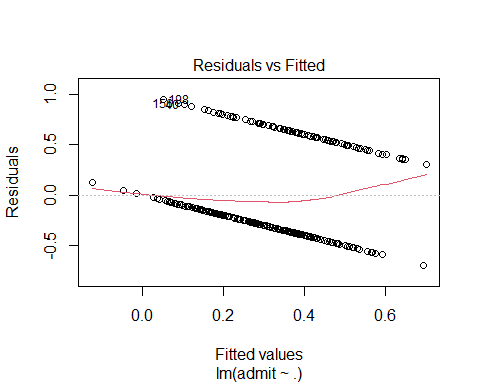
partition <- sample(2,nrow(data), replace = TRUE, prob = c(0.8, 0.2))  
  
train <- data[partition==1,]  
validate <- data[partition==2,]

Now, we have two new data of train containing 324 observation and 4 variables and validate data to check our model, it contains 76 obs and 4 variables. Let perform linear regression on the train dataset.

linear\_model <- lm(admit~.,data = train)  
summary(linear\_model)

##   
## Call:  
## lm(formula = admit ~ ., data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.6937 -0.3516 -0.1887 0.4948 0.9475   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.4686723 0.2487674 -1.884 0.06053 .   
## gre 0.0005974 0.0002425 2.464 0.01430 \*   
## gpa 0.1941005 0.0731328 2.654 0.00838 \*\*  
## rank -0.0861892 0.0279024 -3.089 0.00220 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4451 on 300 degrees of freedom  
## Multiple R-squared: 0.1003, Adjusted R-squared: 0.09129   
## F-statistic: 11.15 on 3 and 300 DF, p-value: 5.89e-07

plot(linear\_model)

 # Interpretation The result of the linear regression on the train data shows gpa and rank have significant impact on the admission of a student. GRE has no impact of coefficient of 0.0003. School rank is negative i.e the lower the school rank (high number) the less the student being admitted. Also, the Adjusted R-squared: 0.08258, shows the model is only capturing 8% of the variance in data.

# Prediction on train data

pred\_train <- predict(linear\_model, train)  
head(pred\_train)

## 1 2 3 4 5 6   
## 0.20048909 0.37941750 0.69949026 0.18811130 0.06595272 0.39530305

head(train)

## admit gre gpa rank  
## 1 0 380 3.61 3  
## 2 1 660 3.67 3  
## 3 1 800 4.00 1  
## 4 1 640 3.19 4  
## 5 0 520 2.93 4  
## 6 1 760 3.00 2

Let test the model on the validate data

pred\_test <- predict(linear\_model, validate)  
head(pred\_test)

## 7 10 11 13 27 33   
## 0.3581228 0.5380293 0.4409227 0.6755928 0.5162524 0.2911641

head(validate)

## admit gre gpa rank  
## 7 1 560 2.98 1  
## 10 0 700 3.92 2  
## 11 0 800 4.00 4  
## 13 1 760 4.00 1  
## 27 1 620 3.61 1  
## 33 0 600 3.40 3

# Interpretation

The model predict that the first candidate from the validate data should be rejected which was actually rejected from the test data.

# Accuracy of the model/Confusion Matrix

pred\_train\_1 = ifelse(pred\_train>0.5,1,0)  
accuracy <- table(prediction=pred\_train\_1, actual=train$admit)  
accuracy

## actual  
## prediction 0 1  
## 0 193 75  
## 1 14 22

From the result, it shows that 207 students were not admitted and they are predicted not admitted. 15 students were not admitted but the model predicted them to be admitted. Also. 89 students were admitted, while the model predicted not admitted. 20 were admitted and the model predicted to be admitted.

# NOTE: Please note that linear regression is not suitable for this type of dataset. Linear regression can be used if the predicted variable is not categorical. Logistic regression is best used as the result is on classification.

# Logistic Regression

Note: admit and rank are not numerical variables, they are factors but because linear regression predict numerical variable we did not convert them. But we can convert them for logistic regression to actually get the full insight.

# Converting “admit” and “rank” to factor varibales

str(data)

## 'data.frame': 400 obs. of 4 variables:  
## $ admit: int 0 1 1 1 0 1 1 0 1 0 ...  
## $ gre : int 380 660 800 640 520 760 560 400 540 700 ...  
## $ gpa : num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...  
## $ rank : int 3 3 1 4 4 2 1 2 3 2 ...

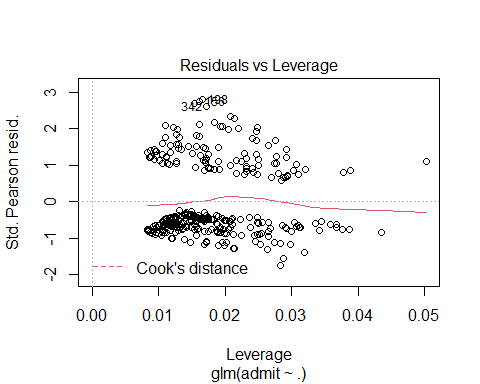
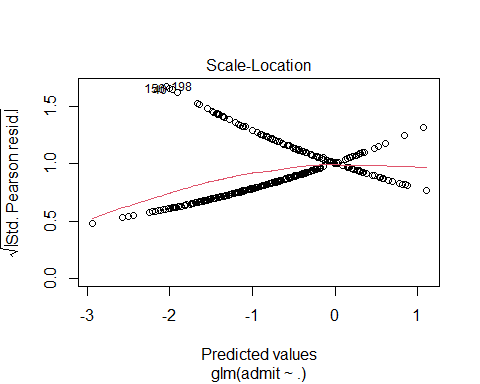
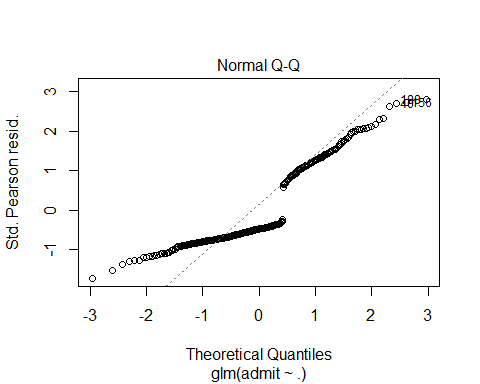
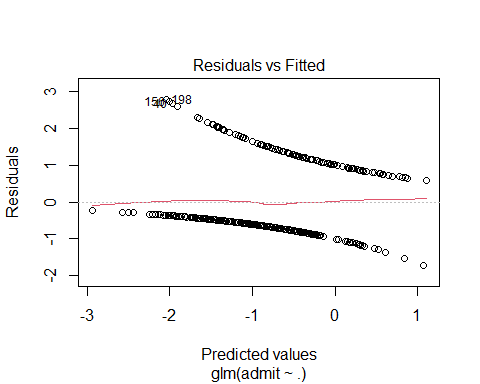
data$admit <- as.factor(data$admit)  
data$rank <- as.factor(data$rank)

partition <- sample(2,nrow(data), replace = TRUE, prob = c(0.8, 0.2))  
  
train1 <- data[partition==1,]  
validate1 <- data[partition==2,]

lgr <- glm(admit~.,data = train1, family = "binomial")  
summary(lgr)

##   
## Call:  
## glm(formula = admit ~ ., family = "binomial", data = train1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6544 -0.9034 -0.6367 1.1532 2.0803   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.576810 1.239154 -3.693 0.000221 \*\*\*  
## gre 0.002245 0.001171 1.916 0.055328 .   
## gpa 0.971198 0.362590 2.679 0.007395 \*\*   
## rank2 -0.523745 0.356803 -1.468 0.142136   
## rank3 -1.161579 0.386829 -3.003 0.002675 \*\*   
## rank4 -1.499801 0.455344 -3.294 0.000989 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 422.29 on 330 degrees of freedom  
## Residual deviance: 386.72 on 325 degrees of freedom  
## AIC: 398.72  
##   
## Number of Fisher Scoring iterations: 4

plot(lgr)

 ## Interpretation The result shows that gre has a significant effect at 98% confident interval on the admission of the students. GPA also has strong effect which is significant at 94%. Also, students from school ranked lowest are likely not being accepted as the coefficient shows negative values. Therefore, we can say that the higher the gpa from best school the higher the probability of being accepted.

#Prediction of the model with train data

prediction <- predict(lgr, train1, type = "response")  
head(prediction)

## 1 2 3 4 6 7   
## 0.2011205 0.3334848 0.7509888 0.1762794 0.3820525 0.3952009

head(train1)

## admit gre gpa rank  
## 1 0 380 3.61 3  
## 2 1 660 3.67 3  
## 3 1 800 4.00 1  
## 4 1 640 3.19 4  
## 6 1 760 3.00 2  
## 7 1 560 2.98 1

The prediction of the train data from the model shows that student 1 should not be admitted and he/she was not admitted. student 2 was also predicted to be not admitted but was actually admitted.

#Prediction on the Test data

prediction1 <- predict(lgr, validate1, type = "response")  
head(prediction1)

## 5 9 13 23 27 32   
## 0.1126795 0.2255192 0.7338213 0.1201617 0.5795791 0.3145872

head(validate1)

## admit gre gpa rank  
## 5 0 520 2.93 4  
## 9 1 540 3.39 3  
## 13 1 760 4.00 1  
## 23 0 600 2.82 4  
## 27 1 620 3.61 1  
## 32 0 760 3.35 3

## Confusion Matrix

pre <- ifelse(prediction>0.5, 1, 0)  
cm <- table(prediction= pre, actual= train1$admit)  
cm

## actual  
## prediction 0 1  
## 0 197 81  
## 1 23 30

rate <- sum(diag(cm))/sum(cm)  
rate

## [1] 0.6858006

The confusion matrix shows that 202 students were actually not admitted and the model correctly predict them not admitted. 14 were actually not admitted but the model predict them to be admitted. Also, 77 were actually admitted and were missed predicted while 19 were correctly predicted to be admitted.