DAR\_Assignment\_4

Sanket Praveen Patil

2023-11-11

# Problem 1:

# Reading the data

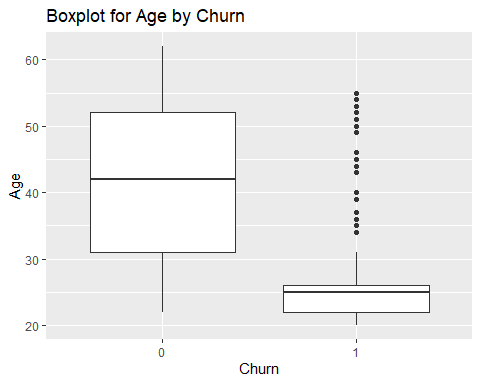
df = read.csv("churn\_train.csv", header = T)  
dim(df)

## [1] 983 9

head(df)

## GENDER EDUCATION LAST\_PRICE\_PLAN\_CHNG\_DAY\_CNT TOT\_ACTV\_SRV\_CNT AGE  
## 1 M 2 0 1 36  
## 2 M . 0 4 33  
## 3 F 1 0 1 37  
## 4 M 1 0 3 58  
## 5 F 1 0 3 38  
## 6 M 2 0 3 42  
## PCT\_CHNG\_IB\_SMS\_CNT PCT\_CHNG\_BILL\_AMT CHURN COMPLAINT  
## 1 0.8421053 0.5709716 0 0  
## 2 1.3969849 1.1955521 0 1  
## 3 0.6440678 0.9074558 0 1  
## 4 1.8245614 1.1771058 0 1  
## 5 0.4507042 1.0892169 0 1  
## 6 1.0860215 0.7079038 0 1

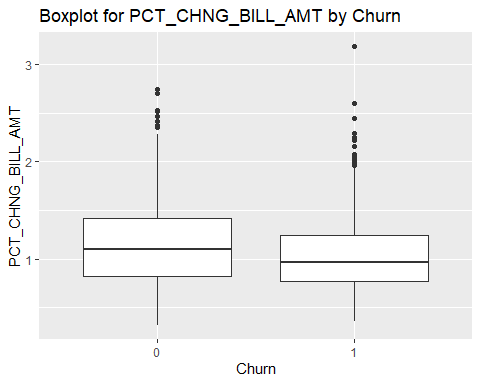
library(ggplot2)  
  
# Boxplot for Age  
ggplot(df, aes(x = as.factor(CHURN), y = AGE)) +  
 geom\_boxplot() +  
 labs(title = "Boxplot for Age by Churn", x = "Churn", y = "Age")



* By looking at the boxplot, we can see if the age value is in between 31 to 52, loan will not be approved.
* Apposite to that, if, age is in between 22 to 27, there is probability of loan getting approved.
* There are some outliers as well for age group 32 to 52 as their loan approved.

# Boxplot for PCT\_CHNG\_BILL\_AMT

ggplot(df, aes(x = as.factor(CHURN), y = PCT\_CHNG\_BILL\_AMT)) +  
 geom\_boxplot() +  
 labs(title = "Boxplot for PCT\_CHNG\_BILL\_AMT by Churn", x = "Churn", y = "PCT\_CHNG\_BILL\_AMT")



* Changes in bill will not affect the Churn value as we cannot see a separation between the two boxplots shown above.

# Question 2:

# Fitting logistic regression model

model <- glm(CHURN ~ ., data = df, family = "binomial")  
summary(model)

##   
## Call:  
## glm(formula = CHURN ~ ., family = "binomial", data = df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 6.43494 1.97935 3.251 0.00115 \*\*   
## GENDERF 0.37170 1.90624 0.195 0.84540   
## GENDERM 0.28369 1.90439 0.149 0.88158   
## EDUCATION1 0.47772 0.25127 1.901 0.05728 .   
## EDUCATION2 0.36377 0.25260 1.440 0.14983   
## EDUCATION3 0.80136 0.62368 1.285 0.19883   
## EDUCATION4 1.16679 0.99002 1.179 0.23858   
## EDUCATION5 12.87920 623.78082 0.021 0.98353   
## EDUCATION6 1.09081 1.75631 0.621 0.53455   
## LAST\_PRICE\_PLAN\_CHNG\_DAY\_CNT 0.21474 0.56433 0.381 0.70356   
## TOT\_ACTV\_SRV\_CNT -0.55387 0.06368 -8.698 < 2e-16 \*\*\*  
## AGE -0.17767 0.01272 -13.970 < 2e-16 \*\*\*  
## PCT\_CHNG\_IB\_SMS\_CNT -0.39073 0.14425 -2.709 0.00676 \*\*   
## PCT\_CHNG\_BILL\_AMT -0.41377 0.22336 -1.852 0.06396 .   
## COMPLAINT 0.52141 0.22683 2.299 0.02152 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1360.48 on 982 degrees of freedom  
## Residual deviance: 714.66 on 968 degrees of freedom  
## AIC: 744.66  
##   
## Number of Fisher Scoring iterations: 13

# Pulling out non significant variables

coefficients\_table <- summary(model)$coefficients

# Filter variables with p-values greater than 0.05

non\_significant\_vars <- coefficients\_table[coefficients\_table[, 4] > 0.05, ]

# Arrange in descending order based on p-values

non\_significant\_vars <- non\_significant\_vars[order(-non\_significant\_vars[, 4]), ]

# Display the result

non\_significant\_vars

## Estimate Std. Error z value Pr(>|z|)  
## EDUCATION5 12.8792019 623.7808189 0.0206470 0.98352725  
## GENDERM 0.2836888 1.9043909 0.1489656 0.88158075  
## GENDERF 0.3716970 1.9062439 0.1949892 0.84540141  
## LAST\_PRICE\_PLAN\_CHNG\_DAY\_CNT 0.2147376 0.5643301 0.3805177 0.70356118  
## EDUCATION6 1.0908100 1.7563088 0.6210810 0.53454634  
## EDUCATION4 1.1667870 0.9900225 1.1785460 0.23857902  
## EDUCATION3 0.8013614 0.6236766 1.2848989 0.19882760  
## EDUCATION2 0.3637661 0.2525950 1.4401158 0.14983463  
## PCT\_CHNG\_BILL\_AMT -0.4137659 0.2233597 -1.8524644 0.06395917  
## EDUCATION1 0.4777199 0.2512733 1.9011963 0.05727631

# As there are many insignificant variables, we will use backward elimination method to select significant variables.

# Apply backward selection

backward\_model <- step(model, direction = "backward", trace = 0)  
summary(backward\_model)

##   
## Call:  
## glm(formula = CHURN ~ TOT\_ACTV\_SRV\_CNT + AGE + PCT\_CHNG\_IB\_SMS\_CNT +   
## PCT\_CHNG\_BILL\_AMT + COMPLAINT, family = "binomial", data = df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 7.11401 0.53506 13.296 <2e-16 \*\*\*  
## TOT\_ACTV\_SRV\_CNT -0.54892 0.06282 -8.738 <2e-16 \*\*\*  
## AGE -0.17781 0.01270 -14.002 <2e-16 \*\*\*  
## PCT\_CHNG\_IB\_SMS\_CNT -0.41230 0.14403 -2.863 0.0042 \*\*   
## PCT\_CHNG\_BILL\_AMT -0.39914 0.22057 -1.810 0.0704 .   
## COMPLAINT 0.50489 0.22283 2.266 0.0235 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1360.48 on 982 degrees of freedom  
## Residual deviance: 720.87 on 977 degrees of freedom  
## AIC: 732.87  
##   
## Number of Fisher Scoring iterations: 6

final\_model <- glm(formula = CHURN ~ TOT\_ACTV\_SRV\_CNT + AGE + PCT\_CHNG\_IB\_SMS\_CNT +   
 PCT\_CHNG\_BILL\_AMT + COMPLAINT, family = "binomial", data = df)  
summary(final\_model)

##   
## Call:  
## glm(formula = CHURN ~ TOT\_ACTV\_SRV\_CNT + AGE + PCT\_CHNG\_IB\_SMS\_CNT +   
## PCT\_CHNG\_BILL\_AMT + COMPLAINT, family = "binomial", data = df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 7.11401 0.53506 13.296 <2e-16 \*\*\*  
## TOT\_ACTV\_SRV\_CNT -0.54892 0.06282 -8.738 <2e-16 \*\*\*  
## AGE -0.17781 0.01270 -14.002 <2e-16 \*\*\*  
## PCT\_CHNG\_IB\_SMS\_CNT -0.41230 0.14403 -2.863 0.0042 \*\*   
## PCT\_CHNG\_BILL\_AMT -0.39914 0.22057 -1.810 0.0704 .   
## COMPLAINT 0.50489 0.22283 2.266 0.0235 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1360.48 on 982 degrees of freedom  
## Residual deviance: 720.87 on 977 degrees of freedom  
## AIC: 732.87  
##   
## Number of Fisher Scoring iterations: 6

# Expression for the fitted model is as follows

#logit(p)=7.11401−0.54892×TOT\_ACTV\_SRV\_CNT−0.17781×AGE−0.41230×PCT\_CHNG\_IB\_SMS\_CNT−0.39914×PCT\_CHNG\_BILL\_AMT+0.50489×COMPLAINT

# Question 3:

Intercept = 7.11401

Odds Ratio = e^7.11401 = 1230.83

# The odds of churn when all other variables are zero are approximately 1231 times higher than the odds of not churning.

# TOT\_ACTV\_SRV\_CNT:

Coefficient: -0.54892

Odds Ratio = e^-0.54892 = 0.578

# For each unit increase in the total number of active services, the odds of churn decrease by approximately 42.2%.

# AGE:

Coefficient: -0.17781

Odds Ratio = e^-0.17781 = 0.837

# For each one-year increase in age, the odds of churn decrease by approximately 16.3%.

# PCT\_CHNG\_IB\_SMS\_CNT:

Coefficient: -0.41230

Odds Ratio = e^−0.41230 = 0.662

# A one-unit increase in the percent change of incoming SMS over the previous months is associated with a 33.8% decrease in the odds of churn.

**PCT\_CHNG\_BILL\_AMT:**

Coefficient: -0.39914

Odds Ratio = e^−0.39914 = 0.670

# A one-unit increase in the percent change of bill amount over the previous months is associated with a 33.0% decrease in the odds of churn.

**COMPLAINT:**

Coefficient: 0.50489

Odds Ratio = 0.50489 = 1.656

# Customers with a complaint in the last two months have approximately 1.656 times higher odds of churning compared to those without a complaint.

# Question 4 :

# Creating a new data frame with provided values

new\_customer <- data.frame(GENDER = "M", EDUCATION = 3, LAST\_PRICE\_PLAN\_CHNG\_DAY\_CNT = 0, TOT\_ACTV\_SRV\_CNT = 4,   
 AGE = 43, PCT\_CHNG\_IB\_SMS\_CNT = 1.04, PCT\_CHNG\_BILL\_AMT = 1.19, COMPLAINT = 1)

# Predicted Probability

predicted\_prob <- predict(final\_model, newdata = new\_customer, type = "response")

# Prediction Interval

predict\_interval <- predict(final\_model, newdata = new\_customer, interval = "prediction")

# Display the results

print(predicted\_prob)

## 1   
## 0.04202857

print(predict\_interval)

## 1   
## -3.126468

# Question 5:

df\_test = read.csv("churn\_test.csv", header = T)  
dim(df\_test)

## [1] 98 9

head(df\_test)

## GENDER EDUCATION LAST\_PRICE\_PLAN\_CHNG\_DAY\_CNT TOT\_ACTV\_SRV\_CNT AGE  
## 1 M . 0 4 33  
## 2 M . 0 3 57  
## 3 M 2 0 2 24  
## 4 M 1 0 3 50  
## 5 M 2 0 3 26  
## 6 M 2 1 2 35  
## PCT\_CHNG\_IB\_SMS\_CNT PCT\_CHNG\_BILL\_AMT CHURN COMPLAINT  
## 1 1.396985 1.1955521 0 1  
## 2 1.189189 1.1436224 0 0  
## 3 1.563025 0.6654836 0 1  
## 4 1.062500 1.3382473 0 1  
## 5 1.012766 0.6653504 0 1  
## 6 1.093458 1.4547685 0 1

# Get predicted probabilities for the test data

predicted\_probs\_test <- predict(final\_model, newdata = df\_test, type = "response")  
  
classify <- function(predicted\_probs, threshold) {  
 # Classify as 1 if predicted probability is greater than or equal to the threshold, else 0  
 predictions <- ifelse(predicted\_probs >= threshold, 1, 0)  
 return(predictions)  
}

# Threshold = 0.5

threshold <- 0.5  
class\_predictions <- classify(predicted\_probs\_test, threshold)

# Create the classification matrix

classification\_matrix <- table(df\_test$CHURN, class\_predictions)

# Confusion Matrix

conf\_matrix <- matrix(c(46, 13, 3, 36), nrow = 2, byrow = TRUE)  
colnames(conf\_matrix) <- c("Predicted 0", "Predicted 1")  
rownames(conf\_matrix) <- c("Actual 0", "Actual 1")

# Function to calculate metrics

calculate\_metrics <- function(conf\_matrix) {  
 TP <- conf\_matrix[1, 1]  
 TN <- conf\_matrix[2, 2]  
 FP <- conf\_matrix[1, 2]  
 FN <- conf\_matrix[2, 1]  
   
 # Accuracy  
 accuracy <- (TP + TN) / sum(conf\_matrix)  
   
 # Sensitivity (Recall)  
 sensitivity <- TP / (TP + FN)  
   
 # Specificity  
 specificity <- TN / (TN + FP)  
   
 # Precision  
 precision <- TP / (TP + FP)  
   
 return(c(Accuracy = accuracy, Sensitivity = sensitivity, Specificity = specificity, Precision = precision))  
}

# Calculate metrics

metrics\_0.5 <- data.frame(calculate\_metrics(conf\_matrix))  
metrics\_0.5

## calculate\_metrics.conf\_matrix.  
## Accuracy 0.8367347  
## Sensitivity 0.9387755  
## Specificity 0.7346939  
## Precision 0.7796610

# Threshold = 0.6

threshold <- 0.6  
class\_predictions <- classify(predicted\_probs\_test, threshold)

# Create the classification matrix

classification\_matrix <- table(df\_test$CHURN, class\_predictions)

# Confusion Matrix

conf\_matrix <- matrix(c(46, 13, 5, 34), nrow = 2, byrow = TRUE)  
colnames(conf\_matrix) <- c("Predicted 0", "Predicted 1")  
rownames(conf\_matrix) <- c("Actual 0", "Actual 1")  
  
metrics\_0.6 <- data.frame(calculate\_metrics(conf\_matrix))  
metrics\_0.6

## calculate\_metrics.conf\_matrix.  
## Accuracy 0.8163265  
## Sensitivity 0.9019608  
## Specificity 0.7234043  
## Precision 0.7796610

# Threshold = 0.7

threshold <- 0.7  
class\_predictions <- classify(predicted\_probs\_test, threshold)

# Create the classification matrix

classification\_matrix <- table(df\_test$CHURN, class\_predictions)

# Confusion Matrix

conf\_matrix <- matrix(c(53, 6, 8, 31), nrow = 2, byrow = TRUE)  
colnames(conf\_matrix) <- c("Predicted 0", "Predicted 1")  
rownames(conf\_matrix) <- c("Actual 0", "Actual 1")  
  
metrics\_0.7 <- data.frame(calculate\_metrics(conf\_matrix))  
metrics\_0.7

## calculate\_metrics.conf\_matrix.  
## Accuracy 0.8571429  
## Sensitivity 0.8688525  
## Specificity 0.8378378  
## Precision 0.8983051

# Threshold = 4

threshold <- 0.4  
class\_predictions <- classify(predicted\_probs\_test, threshold)

# Create the classification matrix

classification\_matrix <- table(df\_test$CHURN, class\_predictions)

# Confusion Matrix

conf\_matrix <- matrix(c(42, 17, 3, 36), nrow = 2, byrow = TRUE)  
colnames(conf\_matrix) <- c("Predicted 0", "Predicted 1")  
rownames(conf\_matrix) <- c("Actual 0", "Actual 1")  
  
metrics\_0.4 <- data.frame(calculate\_metrics(conf\_matrix))  
metrics\_0.4

## calculate\_metrics.conf\_matrix.  
## Accuracy 0.7959184  
## Sensitivity 0.9333333  
## Specificity 0.6792453  
## Precision 0.7118644

results <- metrics\_0.4  
results <- cbind(metrics\_0.4,metrics\_0.5,metrics\_0.6,metrics\_0.7)  
colnames(results) <- c("metrics\_0.4", "metrics\_0.5", "metrics\_0.6", "metrics\_0.7")  
  
print(results)

## metrics\_0.4 metrics\_0.5 metrics\_0.6 metrics\_0.7  
## Accuracy 0.7959184 0.8367347 0.8163265 0.8571429  
## Sensitivity 0.9333333 0.9387755 0.9019608 0.8688525  
## Specificity 0.6792453 0.7346939 0.7234043 0.8378378  
## Precision 0.7118644 0.7796610 0.7796610 0.8983051

# By looking at the above table, we will choose threshold as 0.7 as it shown a constant score for all 4 modules.

# Problem 2:

# Importing data into R

data <- read.table("energytemp.txt", header = TRUE)  
dim(data)

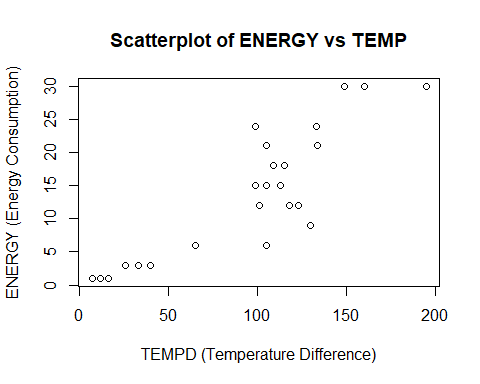
## [1] 24 2

head(data)

## energy temp  
## 1 16 1  
## 2 12 1  
## 3 7 1  
## 4 40 3  
## 5 26 3  
## 6 33 3

# Question 1:

# Scatterplot  
plot(data$energy , data$temp, main = "Scatterplot of ENERGY vs TEMP",   
 xlab = "TEMPD (Temperature Difference)", ylab = "ENERGY (Energy Consumption)")



* We can see positive linear relationship between Tempd and Energy explaining that increase in temperature difference will cause increase in the Energy consumption.

# Question 2:

# Create new variables

data$tempd2 <- data$temp^2  
data$tempd3 <- data$temp^3

# Fit a cubic model

model <- lm(energy ~ temp + tempd2 + tempd3, data = data)  
summary(model)

##   
## Call:  
## lm(formula = energy ~ temp + tempd2 + tempd3, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -19.159 -11.257 -2.377 9.784 26.841   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -17.036232 10.115284 -1.684 0.108   
## temp 24.523999 3.371636 7.274 4.91e-07 \*\*\*  
## tempd2 -1.490029 0.266166 -5.598 1.77e-05 \*\*\*  
## tempd3 0.029278 0.005643 5.188 4.47e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.73 on 20 degrees of freedom  
## Multiple R-squared: 0.9137, Adjusted R-squared: 0.9008   
## F-statistic: 70.62 on 3 and 20 DF, p-value: 8.105e-11

# Question 3:

# By looking at the summary of the model, we can say that all the 3 variables are significant as P values for respective variables are < 0.05.

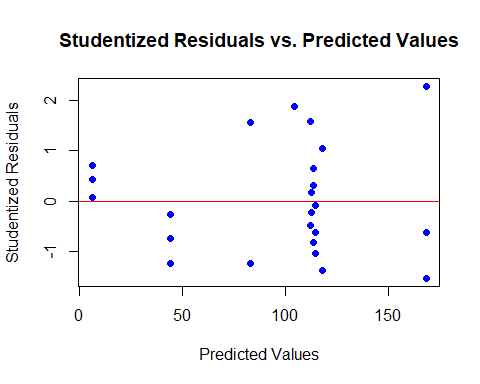
# Question 4:

# Residuals vs Predicted

predicted\_values <- predict(model)  
residuals <- rstudent(model)

# Scatter plot residuals vs predicted\_values

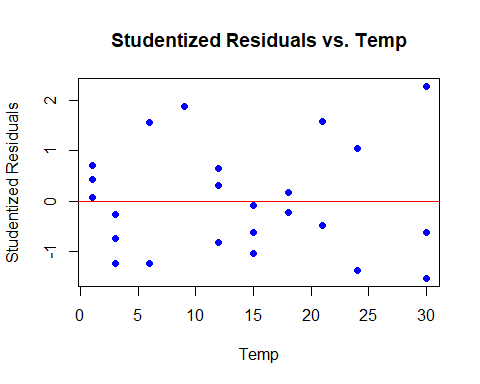
plot(predicted\_values, residuals, main="Studentized Residuals vs. Predicted Values",  
 xlab="Predicted Values", ylab="Studentized Residuals", col="blue", pch=16)  
abline(h=0, col="red")



* By seeing at the plot, we can see that residuals are randomly distributed and no pattern detected. However we cannot conclude anyting with this as we have very less amount of data. There are no outliers present.

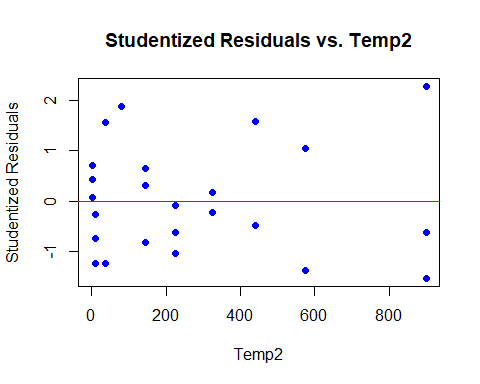
# Residuals vs Temp

plot(data$temp, residuals, main="Studentized Residuals vs. Temp",  
 xlab="Temp", ylab="Studentized Residuals", col="blue", pch=16)  
abline(h=0, col="red")



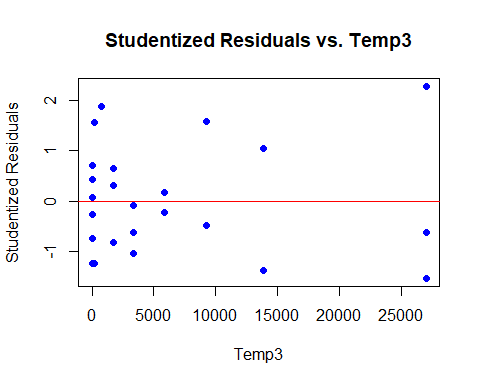
**# Residuals vs Temp2**

plot(data$tempd2, residuals, main="Studentized Residuals vs. Temp2",  
 xlab="Temp2", ylab="Studentized Residuals", col="blue", pch=16)  
abline(h=0, col="red")



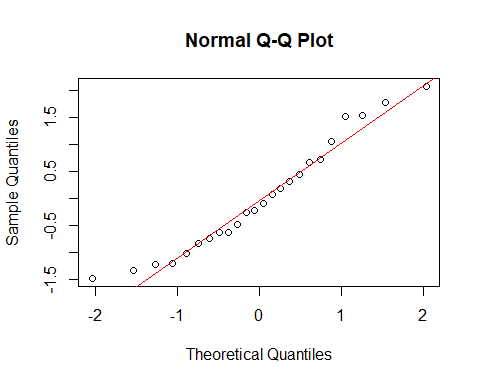
**# Residuals vs Temp3**

plot(data$tempd3, residuals, main="Studentized Residuals vs. Temp3",  
 xlab="Temp3", ylab="Studentized Residuals", col="blue", pch=16)  
abline(h=0, col="red")



**# Create a normal probability plot**

qqnorm(rstandard(model), main="Normal Q-Q Plot")  
qqline(rstandard(model), col="red")



* As we see in the QQ plot, residulas follow the line hence there should not be skewness in the distribution of the outliers. We can say that it is a good model.

# Question 5:

# y=−17.036232+24.523999⋅temp−1.490029⋅tempd2+0.029278⋅tempd3

# Question 6:

# Create a new dataframe with the values for prediction

new\_data <- data.frame(temp = 10, tempd2 = 10^2, tempd3 = 10^3)

# Predict using the fitted model

predicted\_energy <- predict(model, newdata = new\_data)  
print(predicted\_energy)

## 1   
## 108.4787