Assignment04

Erik Pak

August 06, 2024

Problem 1 Churn analysis

Given the large number of competitors, cell phone carriers are very interested in analyzing and predicting customer retention and churn.

The dataset churn_train.csv contains information about a random sample of customers of a cell phone company. For each customer, company recorded the following variables:

- CHURN: 1 if customer switched provider, 0 if customer did not switch
- GENDER: M, F
- EDUCATION (categorical): code 1 to 6 depending on education levels LAST_PRICE_PLAN_CHNG_DAY_CNT: No. of days since last price plan change
- TOT_ACTV_SRV_CNT: Total no. of active services
- AGE: customer age
- PCT_CHNG_IB_SMS_CNT: Percent change of latest 2 months incoming SMS wrt previous 4 months incoming SMS
- PCT_CHNG_BILL_AMT: Percent change of latest 2 months bill amount wrt previous 4 months bill amount
- COMPLAINT: 1 if there was at least a customer's complaint in the two months, 0 no complaints

The company is interested in a churn predictive model that identifies the most important predictors affecting probability of switching to a different mobile phone company (churn = 1). Answer the following questions:

```
library(psych)        # used for describe
library(ggplot2)      # used for ggplot
library(ggpubr)       # combine scatter plots
library(QuantPsyc)    # normalize coefficients
library(car)          # VIF for a model
library(corrplot)     # correlation plot
library(dplyr)        # using filter
library(lmtest)       # likehood ratio from zoo package
```

Import train & test csv file

```
# set working directory
setwd("~/Downloads/Data")

# header in the churn_train.cvs
train.churn <- read.csv(file = 'churn_train.csv', header = TRUE, na.strings = ".")

# header in the churn_test.cvs
test.churn <- read.csv(file = 'churn_test.csv', header = TRUE, na.strings = ".")

# display train.churn
str(train.churn)
```

```
## 'data.frame':    983 obs. of  9 variables:
##  $ GENDER                     : chr  "M" "M" "F" "M" ...
##  $ EDUCATION                  : int  2 NA 1 1 1 2 4 NA 1 2 ...
##  $ LAST_PRICE_PLAN_CHNG_DAY_CNT: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ TOT_ACTV_SRV_CNT           : int  1 4 1 3 3 3 2 3 4 1 ...
##  $ AGE                        : int  36 33 37 58 38 42 42 57 30 55 ...
##  $ PCT_CHNG_IB_SMS_CNT        : num  0.842 1.397 0.644 1.825 0.451 ...
##  $ PCT_CHNG_BILL_AMT          : num  0.571 1.196 0.907 1.177 1.089 ...
##  $ CHURN                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ COMPLAINT                  : int  0 1 1 1 1 1 0 0 1 0 ...
```

Descriptive Statistics

```
# descriptive statistics for churn train
describe(train.churn)
```

```
##                              vars   n  mean    sd median trimmed  mad   min
## GENDER*                         1 983  2.68  0.47   3.00    2.72 0.00  1.00
## EDUCATION                       2 778  1.63  0.75   2.00    1.54 1.48  1.00
## LAST_PRICE_PLAN_CHNG_DAY_CNT    3 983  0.03  0.16   0.00    0.00 0.00  0.00
## TOT_ACTV_SRV_CNT                4 983  1.96  1.65   2.00    1.83 1.48  0.00
## AGE                             5 983 34.07 12.31  28.00   32.73 8.90 20.00
## PCT_CHNG_IB_SMS_CNT             6 983  1.20  0.62   1.09    1.14 0.51  0.05
## PCT_CHNG_BILL_AMT               7 983  1.10  0.43   1.03    1.06 0.40  0.32
## CHURN                           8 983  0.48  0.50   0.00    0.47 0.00  0.00
## COMPLAINT                       9 983  0.76  0.43   1.00    0.83 0.00  0.00
##                                max range  skew kurtosis   se
## GENDER*                       3.00  2.00 -0.81    -1.19 0.02
## EDUCATION                     6.00  5.00  2.03     7.93 0.03
## LAST_PRICE_PLAN_CHNG_DAY_CNT  1.00  1.00  6.02    34.27 0.01
## TOT_ACTV_SRV_CNT              7.00  7.00  0.48    -0.64 0.05
## AGE                          62.00 42.00  0.76    -0.83 0.39
## PCT_CHNG_IB_SMS_CNT           6.17  6.12  1.62     6.35 0.02
## PCT_CHNG_BILL_AMT             3.19  2.87  0.92     1.05 0.01
## CHURN                         1.00  1.00  0.10    -1.99 0.02
## COMPLAINT                     1.00  1.00 -1.23    -0.48 0.01
```

```
# summary for train churn data
summary(train.churn)
```

```
##     GENDER            EDUCATION     LAST_PRICE_PLAN_CHNG_DAY_CNT
##  Length:983         Min.   :1.000   Min.   :0.00000
##  Class :character   1st Qu.:1.000   1st Qu.:0.00000
##  Mode  :character   Median :2.000   Median :0.00000
##                     Mean   :1.627   Mean   :0.02543
##                     3rd Qu.:2.000   3rd Qu.:0.00000
##                     Max.   :6.000   Max.   :1.00000
##                     NA's   :205
##  TOT_ACTV_SRV_CNT      AGE        PCT_CHNG_IB_SMS_CNT PCT_CHNG_BILL_AMT
##  Min.   :0.000    Min.   :20.00   Min.   :0.04878     Min.   :0.3169
##  1st Qu.:0.000    1st Qu.:24.00   1st Qu.:0.79057     1st Qu.:0.7850
##  Median :2.000    Median :28.00   Median :1.08602     Median :1.0342
##  Mean   :1.959    Mean   :34.07   Mean   :1.19733     Mean   :1.0996
##  3rd Qu.:3.000    3rd Qu.:44.00   3rd Qu.:1.50965     3rd Qu.:1.3417
##  Max.   :7.000    Max.   :62.00   Max.   :6.16667     Max.   :3.1850
##
##      CHURN          COMPLAINT
##  Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:1.000
##  Median :0.0000   Median :1.000
##  Mean   :0.4761   Mean   :0.763
##  3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.000
##
```

Data Wrangling

```
# count na values
sum(is.na(train.churn))
```

```
## [1] 205
```

```
sum(is.na(test.churn))
```

```
## [1] 27
```

```
# remove all the na from train & test
train.churn <- na.omit(train.churn)
test.churn <- na.omit(test.churn)

# checking for proportion of churn
prop.table(table(train.churn$CHURN))
```

```
##
##         0         1
## 0.5141388 0.4858612
```

```
# change Male to 0 and female to 1: Male is the baseline case
# for train & test
train.churn$GENDER_F  <- ifelse(train.churn$GENDER == "F", 1,0)
test.churn$GENDER_F   <- ifelse(test.churn$GENDER == "F", 1,0)

# checking for unique values & counts
table(train.churn$EDUCATION)
```

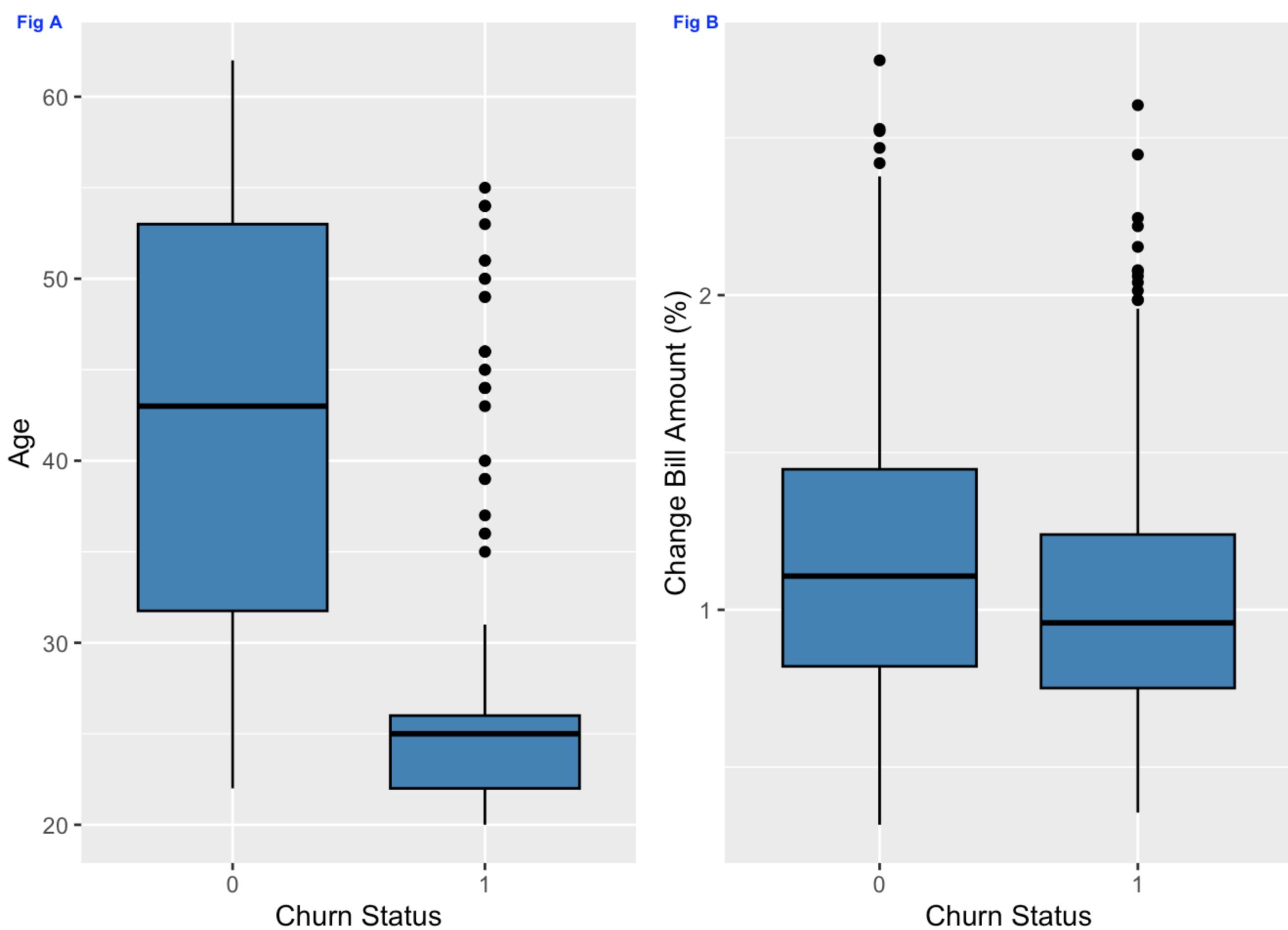
```
##
##   1   2   3   4   5   6
## 367 365  27  12   2   5
```

```
# education: value of 1 is the baseline case for train & test
train.churn$EDUCATION_2   <- ifelse(train.churn$EDUCATION == 2, 1,0)
train.churn$EDUCATION_3   <- ifelse(train.churn$EDUCATION == 3, 1,0)
train.churn$EDUCATION_4   <- ifelse(train.churn$EDUCATION == 4, 1,0)
train.churn$EDUCATION_5   <- ifelse(train.churn$EDUCATION == 5, 1,0)
train.churn$EDUCATION_6   <- ifelse(train.churn$EDUCATION == 6, 1,0)

test.churn$EDUCATION_2   <- ifelse(test.churn$EDUCATION == 2, 1,0)
test.churn$EDUCATION_3   <- ifelse(test.churn$EDUCATION == 3, 1,0)
test.churn$EDUCATION_4   <- ifelse(test.churn$EDUCATION == 4, 1,0)
test.churn$EDUCATION_5   <- ifelse(test.churn$EDUCATION == 5, 1,0)
test.churn$EDUCATION_6   <- ifelse(test.churn$EDUCATION == 6, 1,0)

# display unique value counts
table(train.churn$TOT_ACTV_SRV_CNT)
```

```
##
##   0   1   2   3   4   5   6   7
## 200 141 153 132  87  49  14   2
```

Box Plots

```
# Box plot CHURN & AGE
p_box_age <- ggplot(train.churn, aes(x=as.factor(CHURN), y=AGE, fill=CHURN)) +
                 geom_boxplot(color="black", fill="steelblue") +
                 labs(x="Churn Status", y = "Age") +
                 theme(legend.position="none")

# Box plot CHURN & PCT_CHNG_BILL_AMT
p_box_bill <- ggplot(train.churn, aes(x=as.factor(CHURN), y=PCT_CHNG_BILL_AMT,
              fill=CHURN)) + geom_boxplot(color="black", fill="steelblue") +
              labs(x="Churn Status", y = "Change Bill Amount (%)") +
              theme(legend.position="none")

# combine box plots
box_com_plot <- ggarrange(p_box_age, p_box_bill,
                 labels = c("Fig A", "Fig B"),
                 font.label = list(size = 7, color = "blue"))

# plot all
box_com_plot
```



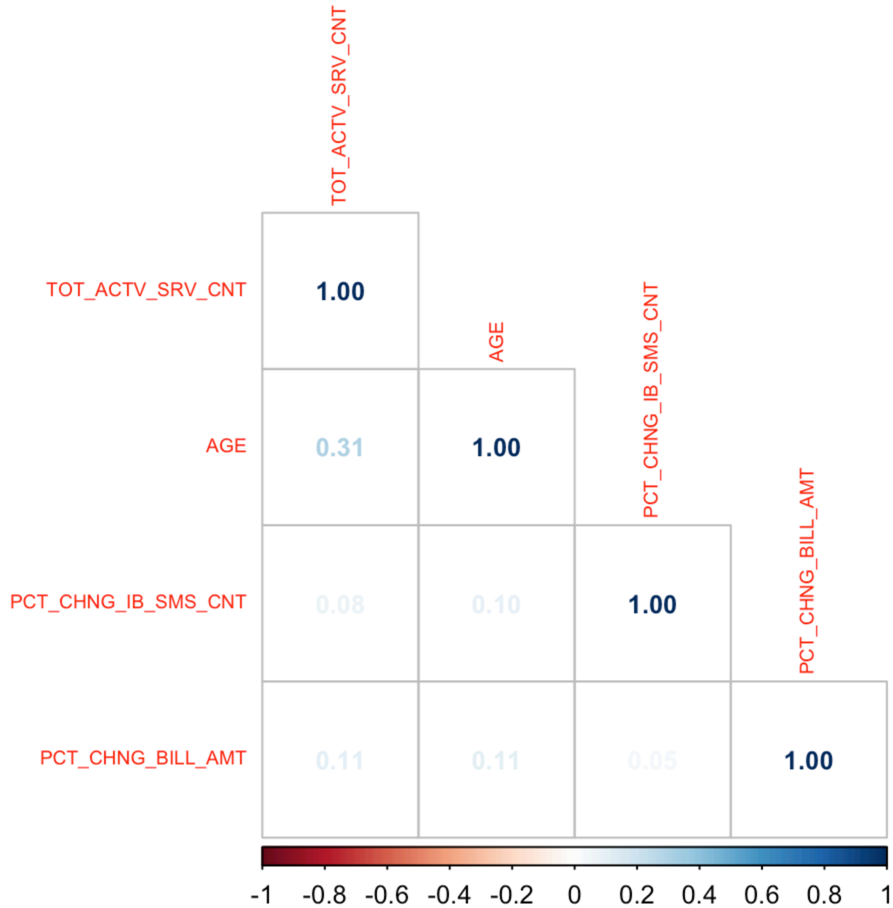
a. Create two boxplots to analyze the observed values of age and PCT_CHNG_BILL_AMT by churn value. Analyze the boxplots and discuss how customer age and changes in bill amount affect churn probabilities.

In the box plot, age is essential in the churn probabilities, and it’s clear younger customers are churning due to some reasons that require investigation to determine the causes of Churn. The median age is around 25 and is right-skewed, with a bit over a dozen outliers on the right side of the distribution. Since the right side has roughly two or three times, the whisker length, including the outliers, tells us the probability curve is much longer on the right side. Lastly, the probability curve probability is much narrower and taller than the Churn Status of “0” due to the IQR (50% of the data) being much closer. The Churn Status “0” seems slightly skewed to the left due to the median leaning more towards the right, and the whisker’s length on the left is somewhat longer.

There is a slight difference in the median for the change in bill amount, but it doesn’t impact Churn due to the change in bill amount. Both show outliers, but Churn Status “1” had roughly twice as many outliers and a slightly shorter data dispersion due to the IQR length. Also, both distributions are somewhat right skewed, and Churn Status ’1” distribution curve is likely narrower and longer than the Churn Status “0”. According to the box plots, the evidence seemed negligible on the percentage of changes in the bill reflecting Churn. However, more investigation is required because customers may be adding additional services to their accounts or for other reasons for the difference in the bill amount.

Correlation Plot

```
# select all the columns
churn.include <- train.churn[, names(train.churn) %in% c("AGE", "PCT_CHNG_IB_SMS_CNT",
"PCT_CHNG_BILL_AMT", "TOT_ACTV_SRV_CNT")]

# correlation values
corr.churn <- cor(churn.include)

# plot correlation
corr_plot <- corrplot(corr.churn, method = 'number', addCoef.col = 'green',
                      type = 'lower', number.cex = 0.8, tl.cex = 0.6,)
```




```
corr.churn
```

```
##                     TOT_ACTV_SRV_CNT        AGE PCT_CHNG_IB_SMS_CNT
## TOT_ACTV_SRV_CNT          1.00000000 0.30562708          0.07828164
## AGE                       0.30562708 1.00000000          0.09549797
## PCT_CHNG_IB_SMS_CNT       0.07828164 0.09549797          1.00000000
## PCT_CHNG_BILL_AMT         0.10692819 0.11034239          0.04705794
##                     PCT_CHNG_BILL_AMT
## TOT_ACTV_SRV_CNT           0.10692819
## AGE                        0.11034239
## PCT_CHNG_IB_SMS_CNT        0.04705794
## PCT_CHNG_BILL_AMT          1.00000000
```

Logistic Regression Model

```
# logistic initial full regression model
full_initial_Model = glm(CHURN ~ LAST_PRICE_PLAN_CHNG_DAY_CNT +
                    TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT +
                    PCT_CHNG_BILL_AMT + COMPLAINT + GENDER_F +
                    EDUCATION_2 + EDUCATION_3 + EDUCATION_4 +
                    EDUCATION_5 + EDUCATION_6,
                    data = train.churn, family=binomial())

# summary of the initial full model
summary(full_initial_Model)
```

```
##
## Call:
## glm(formula = CHURN ~ LAST_PRICE_PLAN_CHNG_DAY_CNT + TOT_ACTV_SRV_CNT +
##     AGE + PCT_CHNG_IB_SMS_CNT + PCT_CHNG_BILL_AMT + COMPLAINT +
##     GENDER_F + EDUCATION_2 + EDUCATION_3 + EDUCATION_4 + EDUCATION_5 +
##     EDUCATION_6, family = binomial(), data = train.churn)
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   7.25549    0.62457  11.617  < 2e-16 ***
## LAST_PRICE_PLAN_CHNG_DAY_CNT  0.88577    0.68234   1.298  0.19424
## TOT_ACTV_SRV_CNT             -0.60907    0.07419  -8.210  < 2e-16 ***
## AGE                          -0.16932    0.01361 -12.445  < 2e-16 ***
## PCT_CHNG_IB_SMS_CNT          -0.45750    0.17060  -2.682  0.00733 **
## PCT_CHNG_BILL_AMT            -0.40106    0.24909  -1.610  0.10739
## COMPLAINT                     0.40082    0.25888   1.548  0.12156
## GENDER_F                     -0.05457    0.23752  -0.230  0.81830
## EDUCATION_2                  -0.15246    0.22420  -0.680  0.49647
## EDUCATION_3                   0.40136    0.60866   0.659  0.50964
## EDUCATION_4                   0.61620    0.96307   0.640  0.52229
## EDUCATION_5                  12.39996  623.45024   0.020  0.98413
## EDUCATION_6                   0.70357    1.77283   0.397  0.69147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1077.9  on 777  degrees of freedom
## Residual deviance:  551.5  on 765  degrees of freedom
## AIC: 577.5
##
## Number of Fisher Scoring iterations: 13
```

```r
# Model iteration 1 logistic regression model
# removing Gender & Education using z-value create first model
Model_1 = glm(CHURN ~ LAST_PRICE_PLAN_CHNG_DAY_CNT + TOT_ACTV_SRV_CNT +
                AGE + PCT_CHNG_IB_SMS_CNT + PCT_CHNG_BILL_AMT + COMPLAINT,
                data = train.churn, family=binomial())

# summary of the Model iteration 1
summary(Model_1)
```

```
##
## Call:
## glm(formula = CHURN ~ LAST_PRICE_PLAN_CHNG_DAY_CNT + TOT_ACTV_SRV_CNT +
##     AGE + PCT_CHNG_IB_SMS_CNT + PCT_CHNG_BILL_AMT + COMPLAINT,
##     family = binomial(), data = train.churn)
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   7.21989    0.59955  12.042  < 2e-16 ***
## LAST_PRICE_PLAN_CHNG_DAY_CNT  0.85792    0.68078   1.260  0.20760
## TOT_ACTV_SRV_CNT             -0.60197    0.07327  -8.216  < 2e-16 ***
## AGE                          -0.17041    0.01362 -12.510  < 2e-16 ***
## PCT_CHNG_IB_SMS_CNT          -0.46459    0.16998  -2.733  0.00627 **
## PCT_CHNG_BILL_AMT            -0.40528    0.24776  -1.636  0.10188
## COMPLAINT                     0.41141    0.25414   1.619  0.10548
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1077.91  on 777  degrees of freedom
## Residual deviance:  553.82  on 771  degrees of freedom
## AIC: 567.82
##
## Number of Fisher Scoring iterations: 6
```

```r
# Model iteration 2 logistic regression model
# removing LAST_PRICE_PLAN_CHNG_DAY_CNT due to z-value
Model_2 = glm(CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT +
                PCT_CHNG_BILL_AMT + COMPLAINT,
                data = train.churn, family=binomial())

# summary of the initial full model
summary(Model_2)
```

```
##
## Call:
## glm(formula = CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT +
##     PCT_CHNG_BILL_AMT + COMPLAINT, family = binomial(), data = train.churn)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          7.19321    0.59692  12.050  < 2e-16 ***
## TOT_ACTV_SRV_CNT    -0.59961    0.07291  -8.224  < 2e-16 ***
## AGE                 -0.16981    0.01356 -12.522  < 2e-16 ***
## PCT_CHNG_IB_SMS_CNT -0.45999    0.16911  -2.720  0.00653 **
## PCT_CHNG_BILL_AMT   -0.38577    0.24727  -1.560  0.11873
## COMPLAINT            0.41197    0.25309   1.628  0.10358
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1077.91  on 777  degrees of freedom
## Residual deviance:  555.46  on 772  degrees of freedom
## AIC: 567.46
##
## Number of Fisher Scoring iterations: 6
```

```
# Model iteration 3 logistic regression model
# removing PCT_CHNG_BILL_AMT due to z-value
Model_3 = glm(CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT +
                COMPLAINT, data = train.churn, family=binomial())

# model summary iteration 3 logistic regression
summary(Model_3)
```

```
##
## Call:
## glm(formula = CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT +
##     COMPLAINT, family = binomial(), data = train.churn)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          6.82139    0.53790  12.682  < 2e-16 ***
## TOT_ACTV_SRV_CNT    -0.59987    0.07263  -8.260  < 2e-16 ***
## AGE                 -0.17111    0.01363 -12.556  < 2e-16 ***
## PCT_CHNG_IB_SMS_CNT -0.46744    0.16964  -2.756  0.00586 **
## COMPLAINT            0.40532    0.25341   1.599  0.10972
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1077.91  on 777  degrees of freedom
## Residual deviance:  557.91  on 773  degrees of freedom
## AIC: 567.91
##
## Number of Fisher Scoring iterations: 6
```

```
# Model iteration 4 logistic regression model
# removing COMPLAINT due to z-value
Model_4 = glm(CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT,
              data = train.churn, family=binomial())

# model summary iteration 4 logistic regression
summary(Model_4)
```

```
##
## Call:
## glm(formula = CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT,
##     family = binomial(), data = train.churn)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          7.14596    0.50574  14.130  < 2e-16 ***
## TOT_ACTV_SRV_CNT    -0.60454    0.07243  -8.347  < 2e-16 ***
## AGE                 -0.17186    0.01362 -12.614  < 2e-16 ***
## PCT_CHNG_IB_SMS_CNT -0.45208    0.16825  -2.687  0.00721 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1077.91  on 777  degrees of freedom
## Residual deviance:  560.46  on 774  degrees of freedom
## AIC: 568.46
##
## Number of Fisher Scoring iterations: 6
```

b. Fit a logistic regression model to predict the churn probability using the data in the dataset (Churn is the response variable and the remaining variables are the independent x-variables). Remove x-variables that are not significant using alpha=0.05. Write down the expression of the fitted model.

Non-significant x-variables:

- GENDER
- EDUCATION
- LAST_PRICE_PLAN_CHNG_DAY_CNT
- PCT_CHNG_BILL_AMT
- COMPLAINT

Final Model Equation:

log[P(CHURN=1) / 1 - P(CHURN=1)] = 7.14596 – 0.60454(TOT_ACTV_SRV_CNT) – 0.17186(AGE) – 0.45208(PCT_CHNG_IB_SMS_CNT)

Check the final model

```
# summary final model
summary(Model_4)
```

```
##
## Call:
## glm(formula = CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT,
##     family = binomial(), data = train.churn)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          7.14596    0.50574  14.130  < 2e-16 ***
## TOT_ACTV_SRV_CNT    -0.60454    0.07243  -8.347  < 2e-16 ***
## AGE                 -0.17186    0.01362 -12.614  < 2e-16 ***
## PCT_CHNG_IB_SMS_CNT -0.45208    0.16825  -2.687  0.00721 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1077.91  on 777  degrees of freedom
## Residual deviance:  560.46  on 774  degrees of freedom
## AIC: 568.46
##
## Number of Fisher Scoring iterations: 6
```

Analysis of multicollinearity

```
# Variance Inflation Factor (VIF)
lr = lm(CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT,
              data = train.churn)

# dispaly VIF
vif(lr)
```

```
##    TOT_ACTV_SRV_CNT                 AGE PCT_CHNG_IB_SMS_CNT
##            1.105999            1.109339            1.011919
```

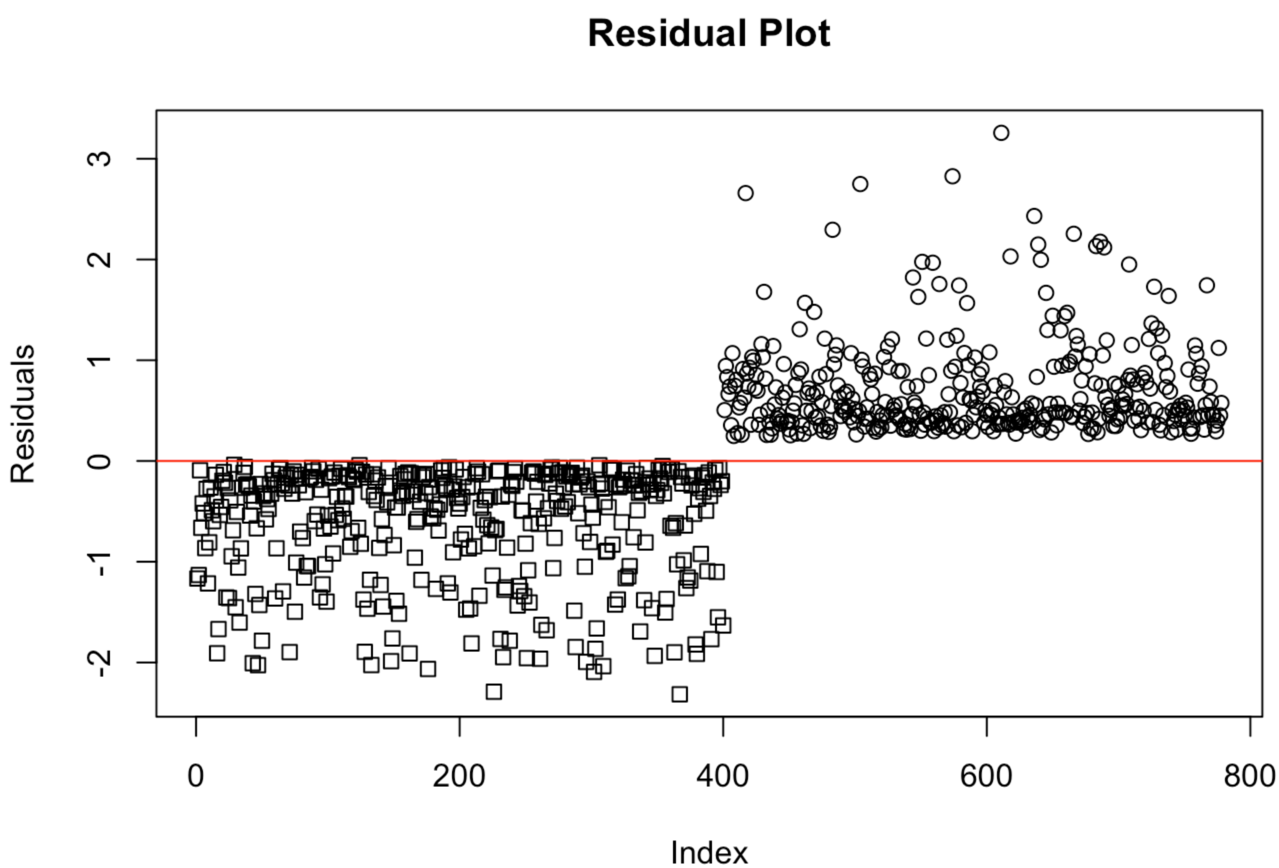
Check goodness of fit: likelihood ratio

```
# goodness of fit test
lrtest(Model_4)
```

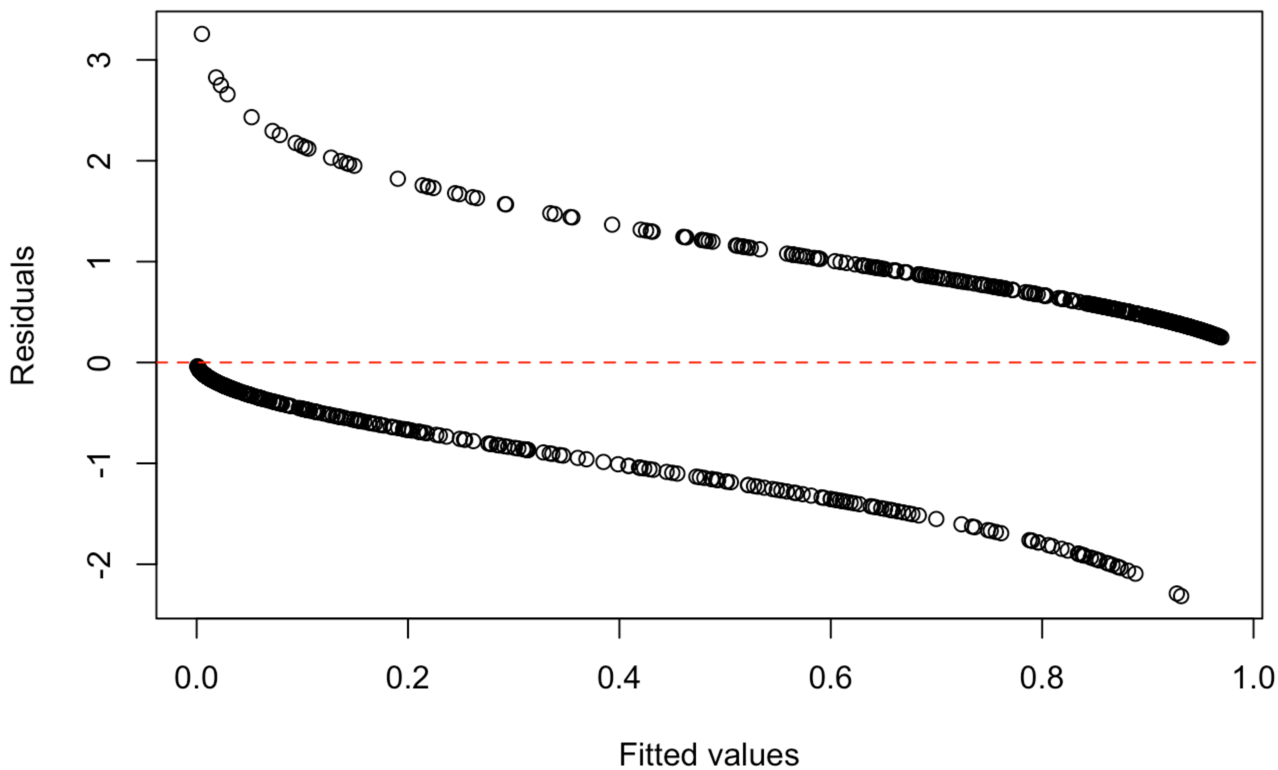
```
## Likelihood ratio test
##
## Model 1: CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT
## Model 2: CHURN ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   4 -280.23
## 2   1 -538.96 -3 517.45  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual plot of LR model

```
# scatter plots
# plot deviance (residuals) vs response variable
res <- residuals(Model_4, type="deviance")
plot(res, pch=train.churn[,8], main = "Residual Plot", ylab = "Residuals")
abline(a=0,b=0, col="red")
```




```
# plot response (predicted values) vs deviance (residuals)
# Plot of deviance residuals from logistic regression model fitted to
# the Model_4
plot(predict(Model_4, type = "response"), res, xlab="Fitted values", ylab = "Residuals")
abline(h = 0, lty = 2, col = "red")
```




```
# 95% CI for the coefficients - change in odds
exp(confint(Model_4))
```

```
## Waiting for profiling to be done...
```

```
##                           2.5 %       97.5 %
## (Intercept)         490.8151134 3576.5217996
## TOT_ACTV_SRV_CNT      0.4720748    0.6273678
## AGE                   0.8187355    0.8637519
## PCT_CHNG_IB_SMS_CNT   0.4537548    0.8808732
```

```
# compute exp(coefficients) to analyze
# change in odds for change in in X
exp(coef(Model_4))
```

```
##         (Intercept)    TOT_ACTV_SRV_CNT                 AGE PCT_CHNG_IB_SMS_CNT
##        1268.9662777           0.5463255           0.8420937           0.6363026
```

```
# odds ratios for changes in X
exp(coef(Model_4)) - 1
```

```
##         (Intercept)    TOT_ACTV_SRV_CNT                 AGE PCT_CHNG_IB_SMS_CNT
##        1267.9662777          -0.4536745          -0.1579063          -0.3636974
```

c. Analyze the final logistic regression model and discuss the effect of each variable on the churn probability. Discuss results in terms of odds ratios.

- TOT_ACTV_SRV_CNT = -0.4536745
- AGE = -0.1579063
- PCT_CHNG_IB_SMS_CNT = -0.3636974

Success decreases for all the beta odds since all the beta odds are negative values.

- TOT_ACTV_SRV_CNT the odds of CHURN will decrease 45.37% for an additional unit of TOT_ACTV_SRV_CNT increase.
- AGE the odds of CHURN will decrease by 15.79% for an additional unit of age increase.
- PCT_CHNG_IB_SMS_CNT the odds of CHURN will decrease 36.37% for an additional unit of PCT_CHNG_IB_SMS_CNT increase.

```
# 95% CI for the coefficients - change in odds
confint(Model_4)
```

```
## Waiting for profiling to be done...
```

```
##                          2.5 %     97.5 %
## (Intercept)          6.1960675  8.1821460
## TOT_ACTV_SRV_CNT    -0.7506179 -0.4662223
## AGE                 -0.1999943 -0.1464697
## PCT_CHNG_IB_SMS_CNT -0.7901983 -0.1268416
```

```
# create data frame
pred_data <- data.frame(AGE = c(43), LAST_PRICE_PLAN_CHNG_DAY_CNT = c(0),
                        TOT_ACTV_SRV_CNT = c(4), PCT_CHNG_IB_SMS_CNT = c(1.04),
                        PCT_CHNG_BILL_AMT = c(1.19), COMPLAINT = c(1))

# type = "response" for probabilities
prediction <- predict(Model_4, newdata = pred_data, type = "response",
                      se.fit = TRUE)

# compute the interval
upper <- (prediction$fit) + 1.96 *(prediction$se.fit)
lower <- (prediction$fit) - 1.96 *(prediction$se.fit)

# print prediction upper interval
upper
```

```
##          1
## 0.06292982
```

```
# print prediction lower interval
lower
```

```
##          1
## 0.02064143
```

```
# corresponding 95% confidence limits for the odds ratio are
exp(lower) - 1
```

```
##          1
## 0.02085594
```

```
exp(upper) - 1
```

```
##         1
## 0.0649521
```

d. Compute the predicted churn probability and the prediction interval for a male customer who is 43 years old, and has the following information:

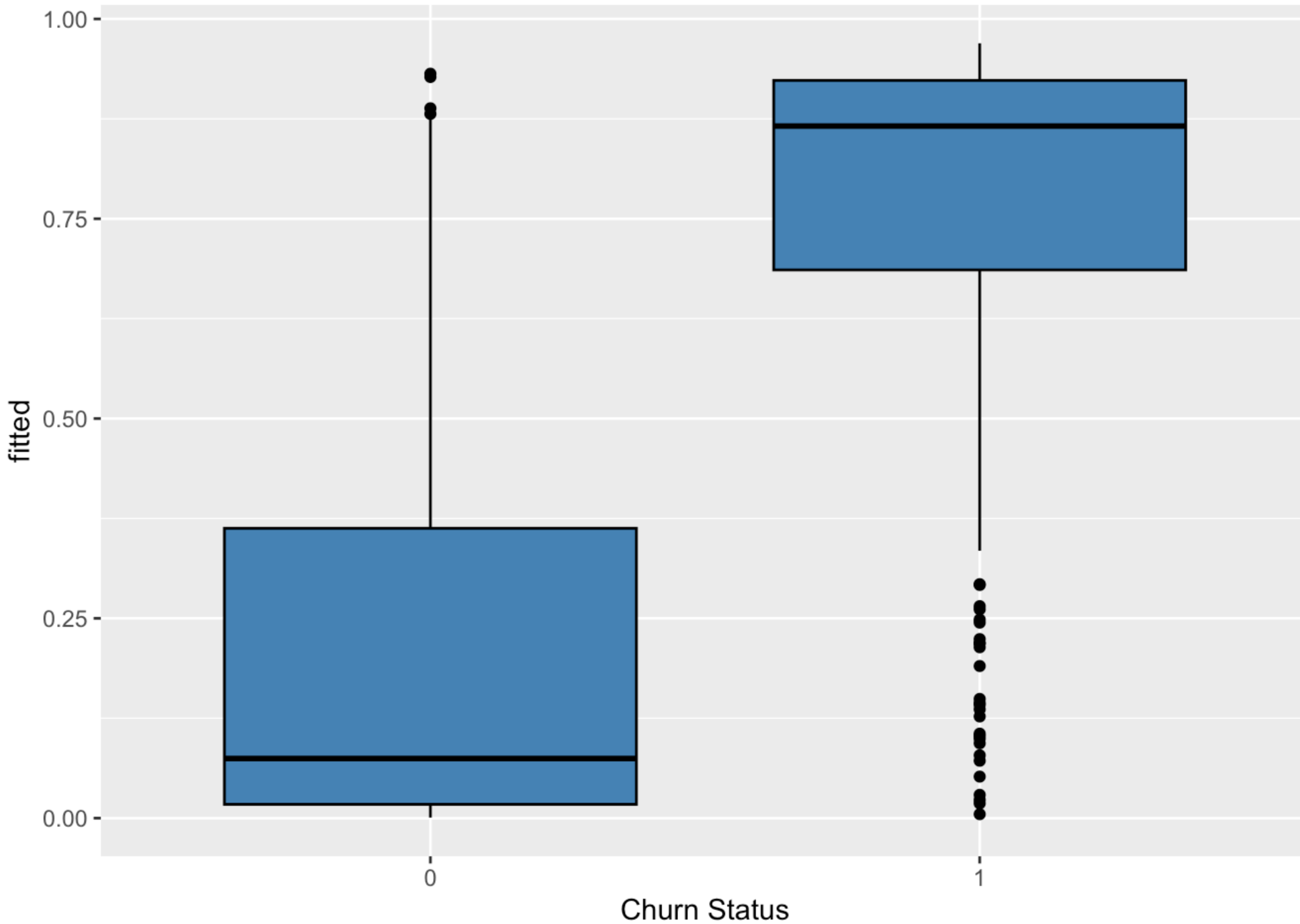
LAST_PRICE_PLAN_CHNG_DAY_CNT=0, TOT_ACTV_SRV_CN=4, PCT_CHNG_IB_SMS_CNT= 1.04, PCT_CHNG_BILL_AMT= 1.19, and COMPLAINT =1

corresponding 95% confidence limits for the odds ratio are: (2.0856, 6.495)

Give prediction of churn with 95% confidance interval is between 2.0856% to 6.495%.

Compute Classification Matrix: Training

```
# Box plot for classification
ggplot(train.churn, aes(x=as.factor(CHURN), y=fitted(Model_4), fill=CHURN)) +
                 geom_boxplot(color="black", fill="steelblue") +
                 labs(x="Churn Status", y = "fitted")
```




```
# boxplot(fitted(Model_4)~CHURN, data = train.churn,
#         names = c("No Chrun", "Churn"))

# using classify functions file to compute classification metrics
source("Classify_functions.R")

# create variable y train with observed values of yin training set
y.train <- train.churn$CHURN

# compute the predicted outcome based on probability threshold equal to 0.5
yc <- classify(fitted(Model_4), 0.5)

# compares predicted outcomes with actual values in training set
cm = compare(classify(fitted(Model_4), 0.50), y.train)

# Accuracy is a metric that summarizes the performance of a
# classification model as the number of correct predictions
# divided by the total number of predictions.
accuracy(cm)
```

```
## [1] 0.840617
```

```
# Precision is looking at the ratio of true positives to the
# predicted positives. This metric is most often used when
# there is a high cost for having false positives.
precision(cm)
```

```
## [1] 0.8082524
```

```
# Specificity is the metric that evaluates a model's ability
# to predict true negatives of each available category.
specificity(cm)
```

```
## [1] 0.8025
```

```
# Sensitivity is the metric that evaluates a model's ability
# to predict true positives of each available category.
sensitivity(cm)
```

```
## [1] 0.8809524
```

```
# Recall, also known as the true positive rate (TPR), is the
# percentage of data samples that a machine learning model
# correctly identifies as belonging to a class of interest—the “positive class”—out of t
he total samples for that class.
recall(cm)
```

```
## [1] 0.8025
```

```
# display confusion matrix
cm
```

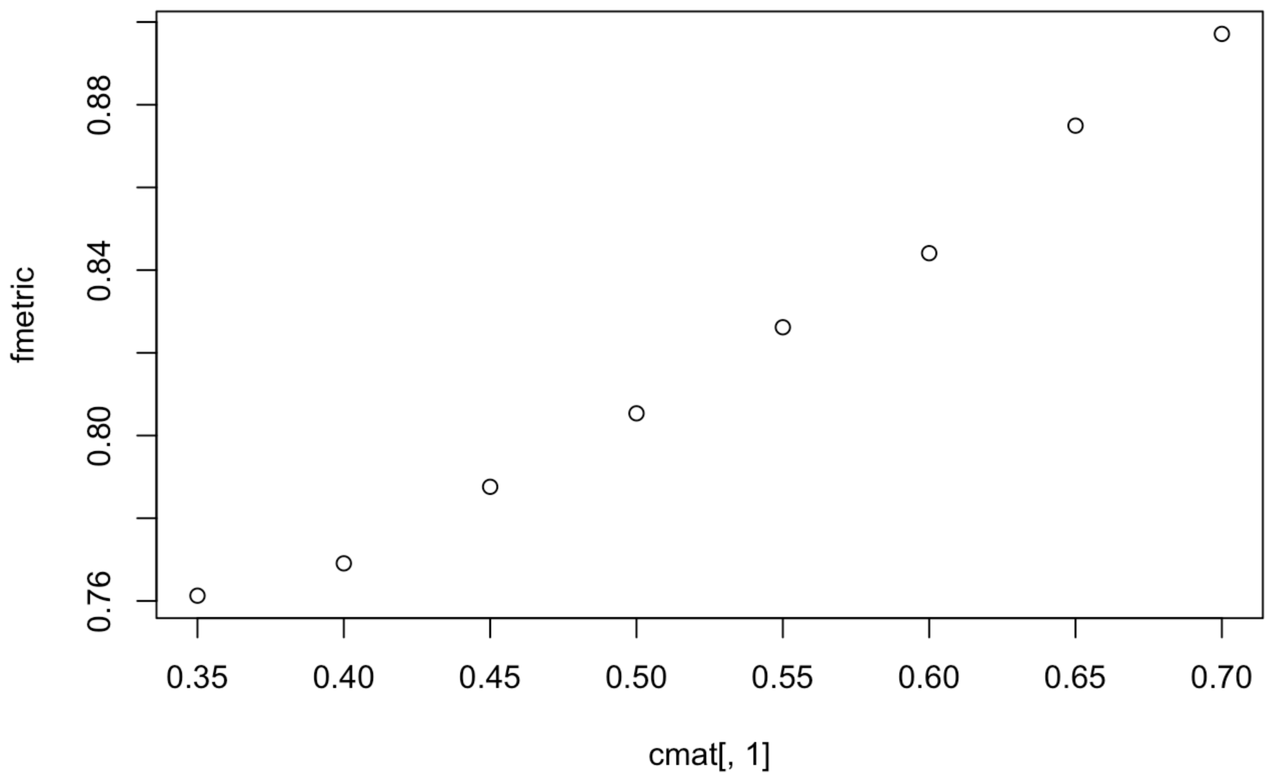
```
##          Predict 1 Predict 0
## Actual 1       333        45
## Actual 0        79       321
```

Predicted Probabilty Threshold

```
# create a list of thresholds and computes classification
# metrics for each threshold
probs=seq(0.35, 0.70, by= 0.05)

# list of predicted Y for each threshold in probs
predlist=lapply(probs, classify, plist=fitted(Model_4))

# list of classification matrices
listmat=lapply(predlist, compare, yvar=y.train)

#list of classification measures
msensitivity=as.vector(lapply(listmat, sensitivity), mode="numeric")
mprecision=as.vector(lapply(listmat, precision), mode="numeric")
mrecall=as.vector(lapply(listmat, recall), mode="numeric")
maccuracy=as.vector(lapply(listmat, accuracy), mode="numeric")
fmetric=2*mrecall*mprecision/(mrecall+mprecision)
cmat=cbind(probs,msensitivity,mprecision, mrecall, fmetric, maccuracy)
colnames(cmat)=c("probs", "sensitivity", "precision",
                 "recall","f-metric","accuracy")

# summary of classification metrics by threshold values
cmat
```

```
##      probs sensitivity precision recall  f-metric  accuracy
## [1,]  0.35   0.9232804 0.7755556 0.7475 0.7612694 0.8329049
## [2,]  0.40   0.9153439 0.7810384 0.7575 0.7690891 0.8341902
## [3,]  0.45   0.9047619 0.7953488 0.7800 0.7875996 0.8406170
## [4,]  0.50   0.8809524 0.8082524 0.8025 0.8053659 0.8406170
## [5,]  0.55   0.8597884 0.8248731 0.8275 0.8261845 0.8431877
## [6,]  0.60   0.8227513 0.8382749 0.8500 0.8440968 0.8367609
## [7,]  0.65   0.7804233 0.8651026 0.8850 0.8749382 0.8341902
## [8,]  0.70   0.7301587 0.8846154 0.9100 0.8971282 0.8226221
```

```
# plot fmetric vs probability values
plot(cmat[,1], fmetric)
```



Test Prediction using Probabilty Threshold

```
# analysis suggests threshold equal to 0.55 using
# accuracy as my predictor
# predicted outcomes in testing set
preds <-  as.vector(predict(Model_4,test.churn, type="response"))

# compute predicted outcome based on probability threshold equal to 0.55
y.pred <- classify(preds, 0.55)

# define y.test= observed values of Y in test set
y.test <- test.churn$CHURN

# compares predicted oucomes with actual values in test set
m <- compare(y.pred, y.test)

# classification matrix
m
```

```
##          Predict 1 Predict 0
## Actual 1        29         2
## Actual 0        10        30
```

e. The dataset churn_test.csv contains a new set of customers, and can be used to test the validity of the churn predictive model. Apply the methods discussed in week 8 lecture to identify a threshold T for the predicted churn probability in order to define a classification rule for customers, so that - predicted probability p(churn) > = T, then customer is a “likely churn”, and

- predicted probability p(churn) < T, then customer is a “unlikely churn”. Compute the optimal T value, and create the classification matrix summarizing classification results. Hint: You can use the Classify_functions.R in your solution.

Note: all the answers to the question are above. ;)

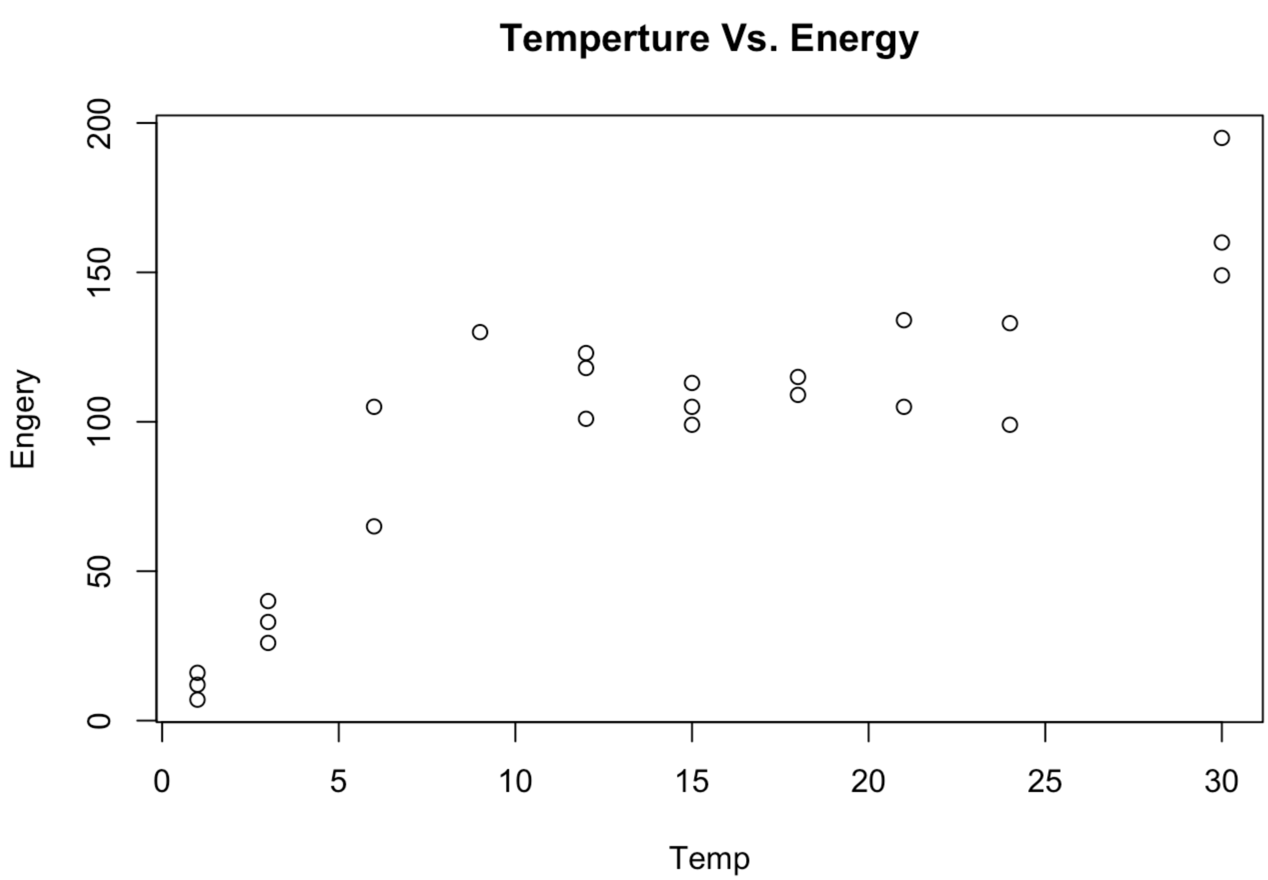
Problem 2: Extra Credit

A researcher is interested in evaluating the relationship between energy consumption by the homeowner and the difference between the internal and external temperatures. A sample of 30 homes was used in the study. During an extended period of time, the average temperature difference (in oF) (TEMPD) inside and outside the homes was recorded. The average energy consumption (ENERGY) was also recorded for each home. The data are stored in the energytemp.txt data file.

```
# header in the energytemp.txt file
mydata <- read.table(file = 'energytemp.txt', header = TRUE)
```

Scatter Plot

```
# plot of temp vs Energy
plot(mydata$temp, mydata$energy, main = "Temperture Vs. Energy",
     xlab = "Temp", ylab = "Engery")
```



a. Create a scatterplot of ENERGY (y) versus TEMPD (x) to visualize the association between the two variables. Analyze the association displayed by the scatterplot.

The scatter plot doesn't seem linear, possibly a polynomial function of degree 3, because it looks like an s-curve.

```
# create two addtional columns using exiting values
mydata$tempd2  <- mydata$temp^2
mydata$tempd3  <- mydata$temp^3

# initial model with quadratic and cubic
energy_Model_1 <- lm(energy ~ temp + tempd2 + tempd3, data =  mydata)

# summary of initial model
summary(energy_Model_1)
```

```
##
## Call:
## lm(formula = energy ~ temp + tempd2 + tempd3, data = mydata)
##
## 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
```

```
# summary of initial model
anova(energy_Model_1)
```

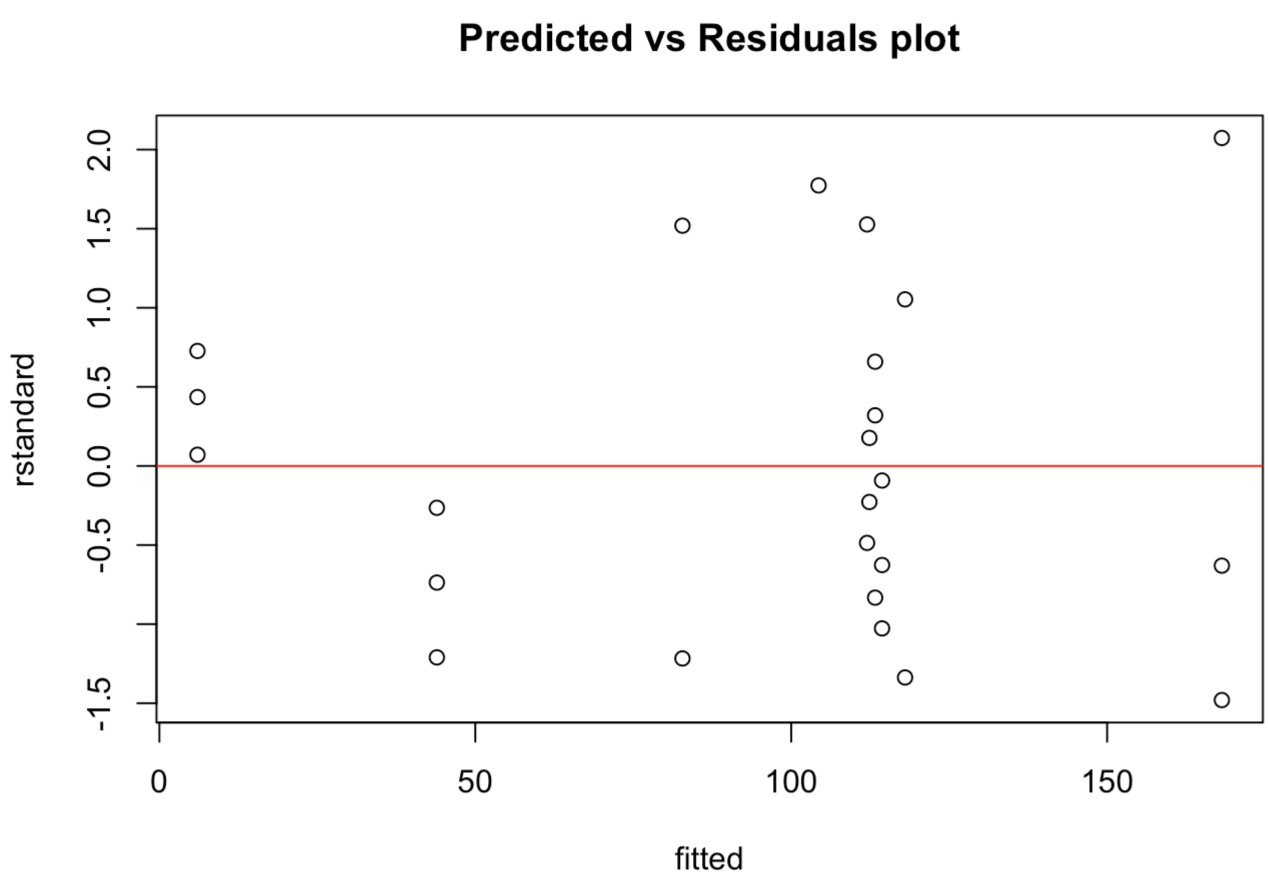
```
## Analysis of Variance Table
##
## Response: energy
##           Df Sum Sq Mean Sq F value    Pr(>F)
## temp       1  43221   43221 174.790 2.409e-11 ***
## tempd2     1   2507    2507  10.138  0.004663 **
## tempd3     1   6656    6656  26.919 4.465e-05 ***
## Residuals 20   4946     247
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

c. Are all variables in the model significant?

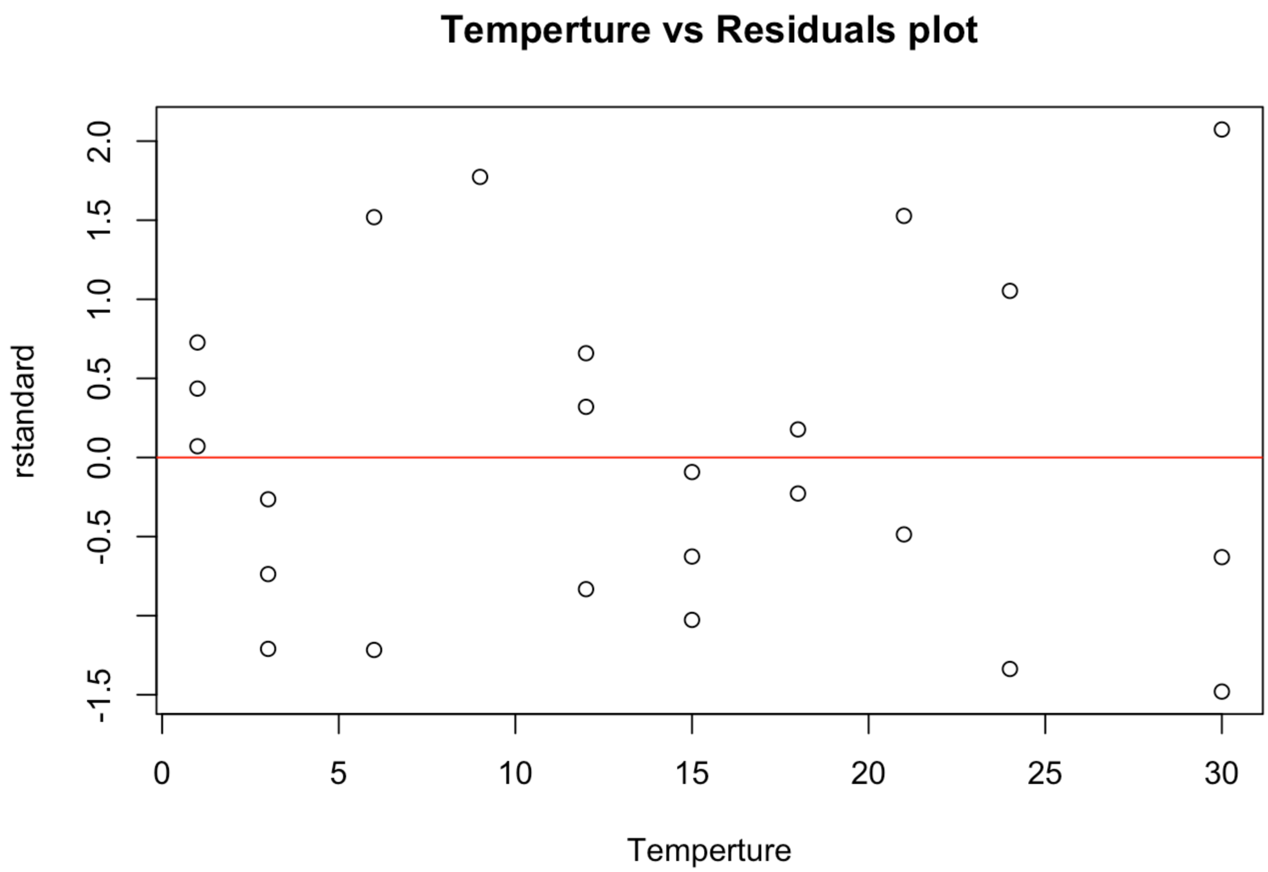
Yes, checking the summary report, all the t-vales are statistically significant, including F-statistic on the model. Also, the F-values for the variances are statistically substantial as well.

Residual Plots

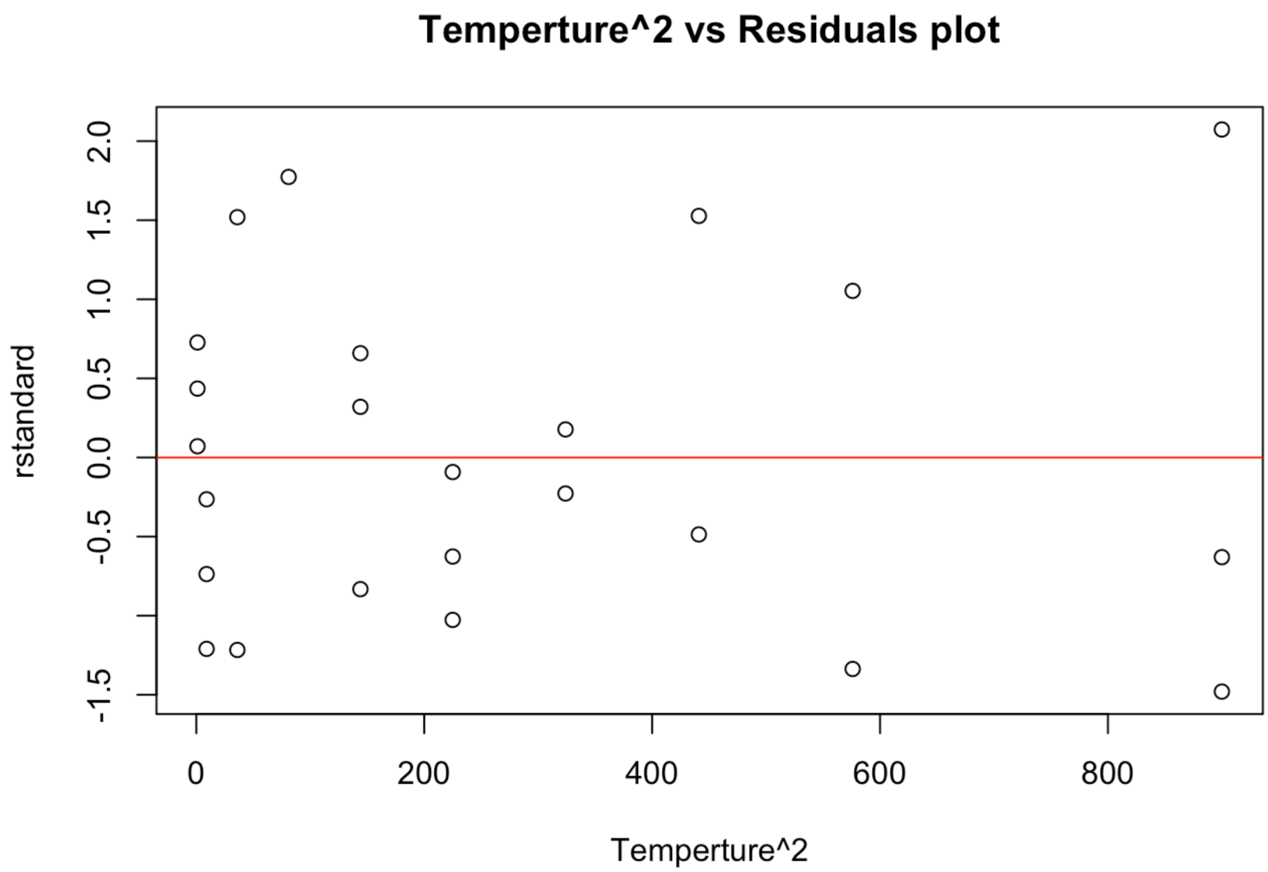
```
# residuals vs fitted values plot
plot(fitted(energy_Model_1), rstandard(energy_Model_1),
     main="Predicted vs Residuals plot", ylab = "rstandard", xlab = "fitted")
abline(a=0, b=0, col='red')
```



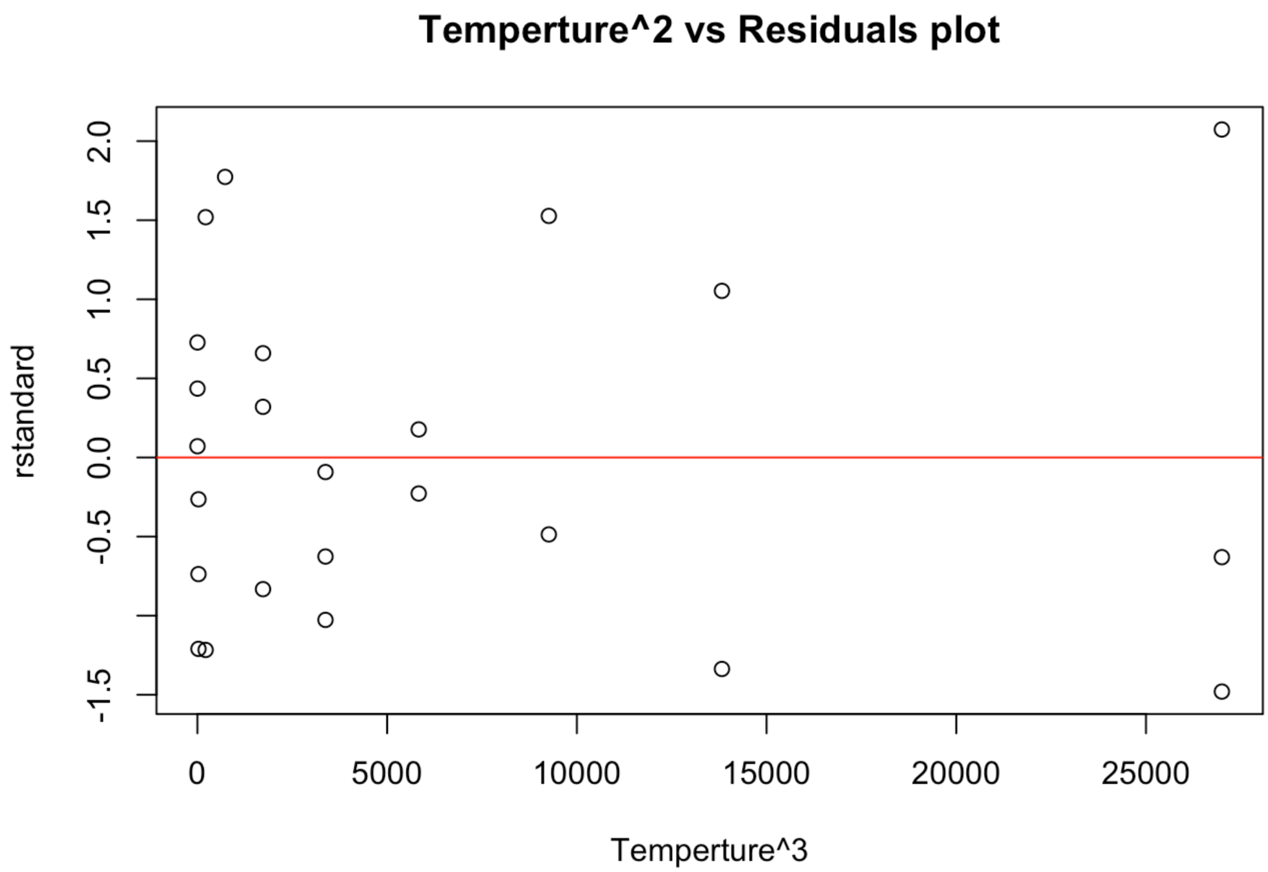

```
# Tempture vs residuals plot
plot(mydata$temp, rstandard(energy_Model_1),
     main="Temperture vs Residuals plot", ylab = "rstandard", xlab = "Temperture")
abline(a=0, b=0, col='red')
```



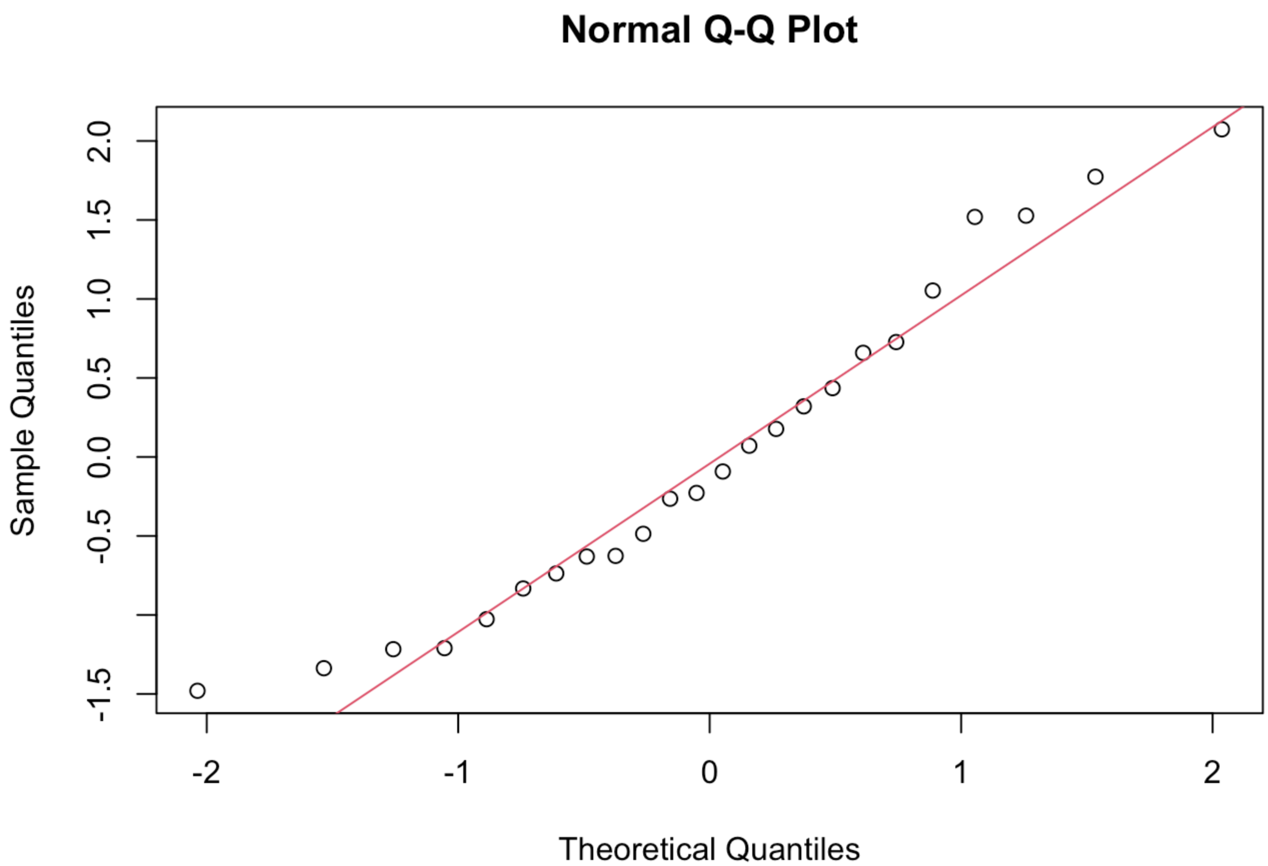

```
# Tempture^2 vs residuals plot
plot(mydata$tempd2, rstandard(energy_Model_1),
     main="Temperture^2 vs Residuals plot", ylab = "rstandard", xlab = "Temperture^2")
abline(a=0, b=0, col='red')
```




```
# Tempture^3 vs residuals plot
plot(mydata$tempd3, rstandard(energy_Model_1),
     main="Temperture^2 vs Residuals plot", ylab = "rstandard", xlab = "Temperture^3")
abline(a=0, b=0, col='red')
```




```
# normal probability plot of residuals
qqnorm(rstandard(energy_Model_1))
qqline(rstandard(energy_Model_1), col = 2)
```



d)

Create the residual plots (residuals vs predicted; residuals vs x variable; and normal plot of residuals). Analyze residual plots to evaluate the normality and constant variance assumptions.

The predicted vs. Residuals plot show the randomness of data points. The evidence of linearity Temp vs. Residuals plot show randomness of data points. Also, evidence of linearity Tempd2 vs. Residuals plot show randomness of data points, including linearity Tempd3 vs. Residuals plot show randomness of data points. Finally, the Normal Q-Q plot shows the points follow very near the straight line, and a possible sign of an outlier is present at the lower left corner of the Q-Q plot. Therefore the regression model is useable.

e. If you are satisfied with the fitted regression model, write down its expression.

y(energy) = -17.036232 + 24.523999(temp) - 1.490029(tempd2) + 0.029278(tempd3)

```
# create data frame for x values of predictions/estimations
new <- data.frame(temp=c(10), tempd2=c(100), tempd3=c(1000))

# compute predictions using the predict() function
predict(energy_Model_1, new, interval="prediction", level=0.95)
```

```
##        fit      lwr     upr
## 1 108.4787 73.37131 143.586
```

f. Use the fitted regression model to predict the average energy consumption for an average difference in temperature equal to TEMP=10.

The prediction for the Temp = 10, tempd2 = 100, and tempd3 = 1000 is generating energy commsumption is 108.4787 with 95% CI is (73.37131, 143.586).