Assignment 4

SRK Yarra

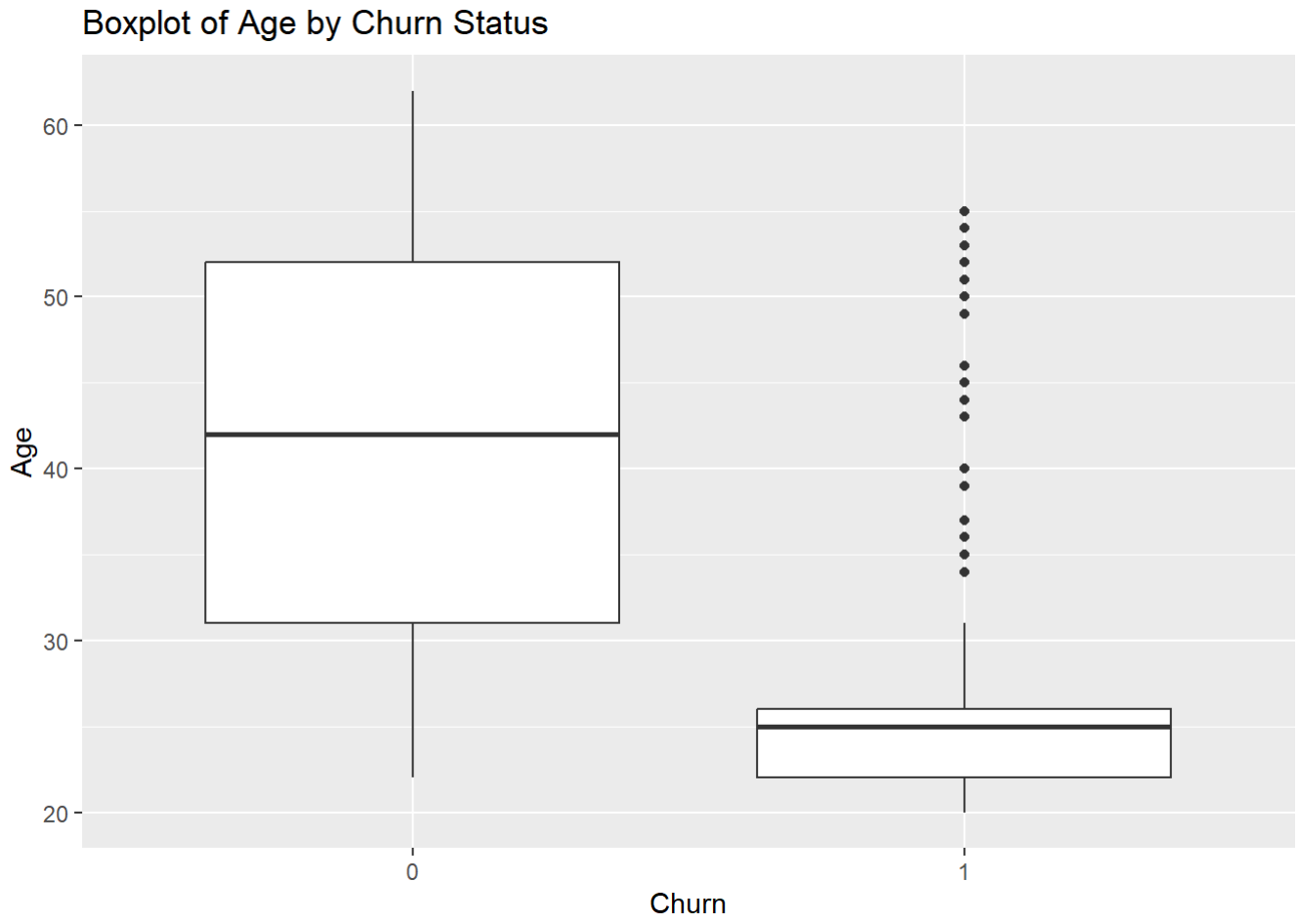
2023-11-10

Problem 1

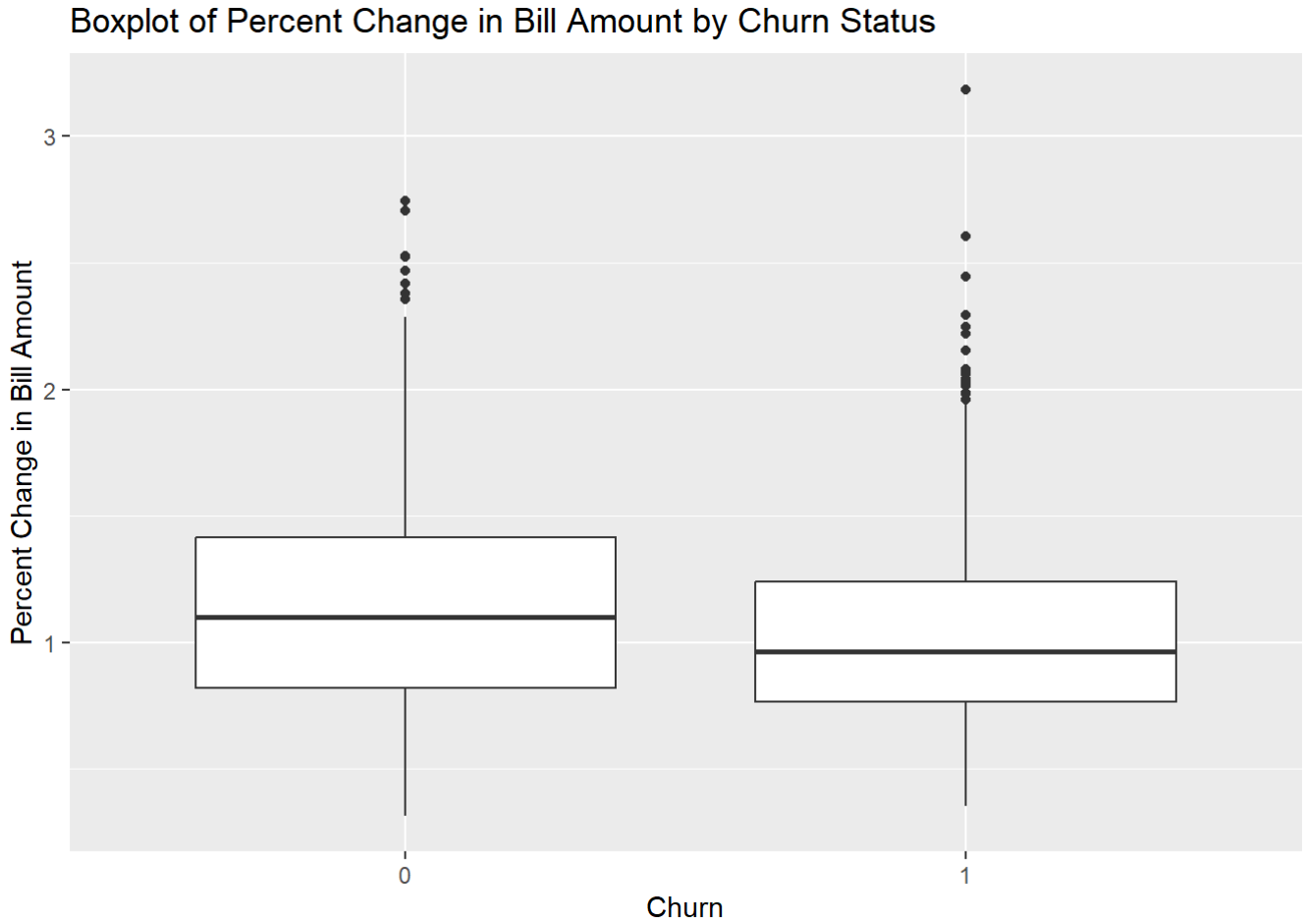
```
# Load the data
churn_train <- read.csv('churn_train.csv')
#Problem1(a)
# Boxplot for AGE by CHURN value
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
ggplot(churn_train, aes(x = as.factor(CHURN), y = AGE)) +
  geom_boxplot() +
  xlab("Churn") +
  ylab("Age") +
  ggtitle("Boxplot of Age by Churn Status")
```



```
# Boxplot for PCT_CHNG_BILL_AMT by CHURN value
ggplot(churn_train, aes(x = as.factor(CHURN), y = PCT_CHNG_BILL_AMT)) +
  geom_boxplot() +
  xlab("Churn") +
  ylab("Percent Change in Bill Amount") +
  ggtitle("Boxplot of Percent Change in Bill Amount by Churn Status")
```



1)a)Age: The boxplot of age for customers who did not churn (CHURN = 0) shows a slightly higher median age compared to those who did churn (CHURN = 1).Younger customers seem more likely to churn than older customers. This could be due to younger individuals being more open to change or more sensitive to competitive offers.

Percent Change in Bill Amount: A higher percentage increase in the bill amount is associated with a greater probability of churn. This suggests that customers are sensitive to price changes and might consider switching to another provider if their bills increase by a certain percentage. These insights could be indicative of churn behavior: while younger customers and those experiencing higher bill changes are more prone to churn, interventions targeting these factors might be effective in retaining customers

#Problem1(b)

```
# Fit the logistic regression model (assuming churn_train data is clean and all variables are
correctly formatted)
model <- glm(CHURN ~ ., data = churn_train,family=binomial())

# Summary of the model to check significant variables
summary(model)
```

```
##
## Call:
## glm(formula = CHURN ~ ., family = binomial(), data = churn_train)
##
## 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
```

```
# Remove non-significant variables and create the fit model
fit <- glm(CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT + COMPLAINT,data=churn_train,
family=binomial())

# Summary of the refined model
summary(fit)
```

```
##
## Call:
## glm(formula = CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT +
##     COMPLAINT, family = binomial(), data = churn_train)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          6.72099    0.48041  13.990  < 2e-16 ***
## TOT_ACTV_SRV_CNT    -0.54745    0.06249  -8.760  < 2e-16 ***
## AGE                 -0.17921    0.01275 -14.051  < 2e-16 ***
## PCT_CHNG_IB_SMS_CNT -0.41796    0.14377  -2.907  0.00365 **
## COMPLAINT            0.50512    0.22278   2.267  0.02337 *
## ---
## 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:  724.17  on 978  degrees of freedom
## AIC: 734.17
##
## Number of Fisher Scoring iterations: 6
```

The model expression is cHURN=6.72099-0.54745*TOT_ACTV_SRV_CNT -0.17921*AGE -0.41796*PCT_CHNG_SMS_CNT +0.50512*COMPLAINT

#Problem1(c)

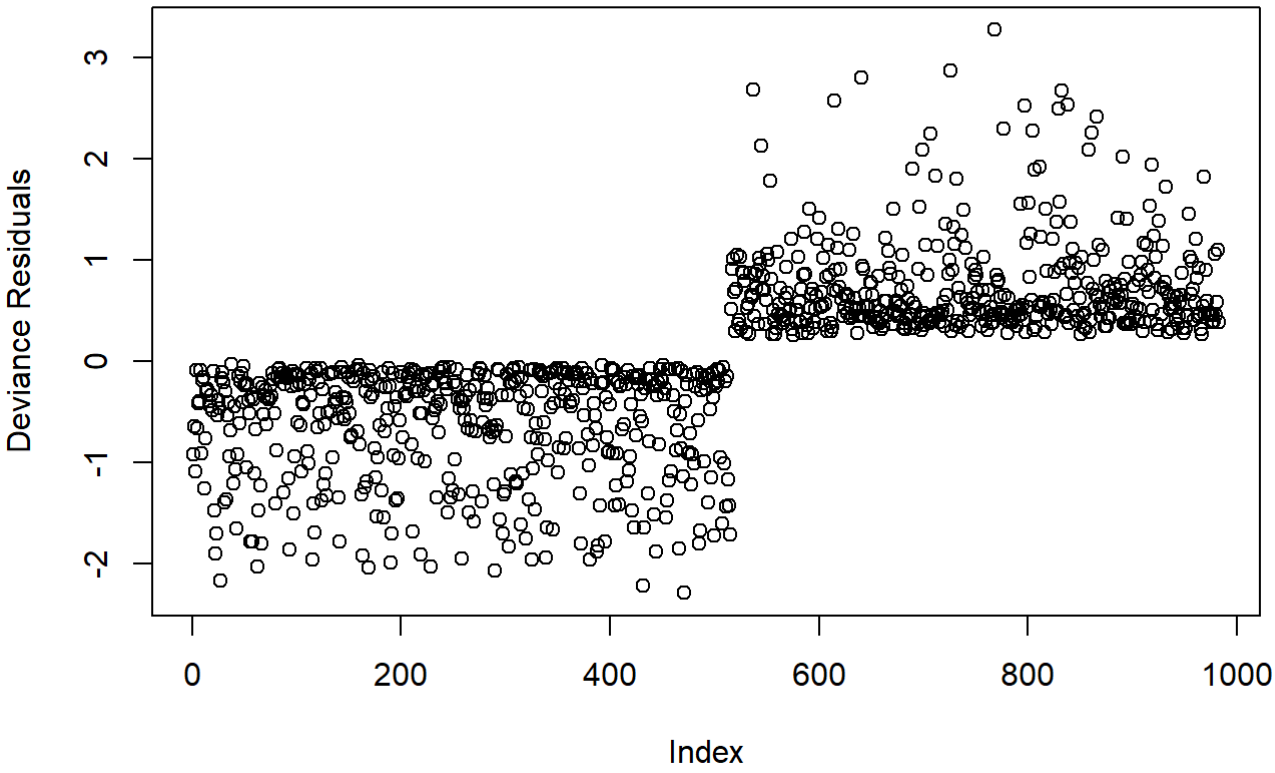
```
# Odds ratios
exp(coef(summary(fit)))
```

```
##                        Estimate Std. Error      z value Pr(>|z|)
## (Intercept)         829.6386175   1.616740 1.190704e+06 1.000000
## TOT_ACTV_SRV_CNT      0.5784238   1.064485 1.568214e-04 1.000000
## AGE                   0.8359293   1.012836 7.898980e-07 1.000000
## PCT_CHNG_IB_SMS_CNT   0.6583866   1.154620 5.463183e-02 1.003654
## COMPLAINT             1.6571826   1.249551 9.653272e+00 1.023647
```

```
coef(summary(fit))
```

```
##                       Estimate Std. Error    z value     Pr(>|z|)
## (Intercept)          6.7209902 0.48041197  13.990056 1.792718e-44
## TOT_ACTV_SRV_CNT    -0.5474484 0.06249123  -8.760403 1.945533e-18
## AGE                 -0.1792112 0.01275401 -14.051362 7.556662e-45
## PCT_CHNG_IB_SMS_CNT -0.4179630 0.14377127  -2.907139 3.647515e-03
## COMPLAINT            0.5051189 0.22278463   2.267297 2.337209e-02
```

```
# Residual plot
plot(residuals(fit, type = "deviance"), ylab = "Deviance Residuals")
```



In

terms of odd terms ratio:

TOT_ACTV_SRV_CNT (Total Number of Active Services): The odds ratio for TOT_ACTV_SRV_CNT is less than 1 (0.5784). This means that for each additional active service, the odds of churn decrease by approximately 42.2% (1 - 0.5784). In other words, customers with more active services are less likely to switch providers.

AGE (Customer Age): The odds ratio for AGE is also less than 1 (0.8359), indicating that as the age increases, the odds of churn decrease. Specifically, for each additional year of age, the odds of churn decrease by about 16.4% (1 - 0.8359).

PCT_CHNG_IB_SMS_CNT (Percent Change of Latest 2 Months Incoming SMS): The odds ratio for PCT_CHNG_IB_SMS_CNT is 0.6584, indicating that higher percentage changes in the count of incoming SMS messages are associated with a lower likelihood of churn. For every unit increase in the percent change of incoming SMS, the odds of churn decrease by roughly 34.2% (1 - 0.6584).

COMPLAINT (At Least One Complaint in the Last Two Months): The odds ratio for COMPLAINT is above 1 (1.6572), which means that having at least one complaint is associated with an increase in the odds of churn.

In more active services and older age are protective against churn, while an increase in SMS activity might indicate higher engagement and thus lower churn probability. Conversely, customer complaints are a strong indicator of potential churn.

#Problem1(d)

```
# Predicted churn probability for a new customer
new_customer <- data.frame(GENDER = "M", EDUCATION = "NA", 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, CO
MPLAINT = 1)
predict_probability <- predict(fit, new_customer, type = "response")

# Prediction interval
predict_interval <- predict(fit, new_customer, type = "response", se.fit = TRUE)
pi <- predict_interval
pi$fit <- predict_probability # predicted probability
pi$upr <- predict_probability + 1.96 * pi$se.fit # upper limit
pi$lwr <- predict_probability - 1.96 * pi$se.fit # lower limit
pi
```

```
## $fit
##         1
## 0.0429241
##
## $se.fit
##          1
## 0.01031498
##
## $residual.scale
## [1] 1
##
## $upr
##          1
## 0.06314146
##
## $lwr
##          1
## 0.02270675
```

#Problem1(E)

```
# Load the test data
churn_test <- read.csv("churn_test.csv")
source("Classify_functions.R")
# Predict churn probabilities using your logistic regression model
# Replace 'model' with the actual logistic regression model variable
predicted_probabilities <- predict(model, newdata = churn_test, type = "response")

library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
# Calculate ROC curve and get range of thresholds
roc_result <- roc(churn_test$CHURN, predicted_probabilities)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
thresholds <- roc_result$thresholds

# Select the optimal threshold by maximizing the Youden's index
optimal_idx <- which.max(roc_result$sensitivities + roc_result$specificities - 1)
optimal_threshold <- roc_result$thresholds[optimal_idx]
optimal_threshold
```

```
## [1] 0.6327031
```

```
# Now, use this optimal threshold to classify the test data
optimal_classified <- classify(predicted_probabilities, optimal_threshold)

# Create the confusion matrix using the optimal threshold
optimal_cm <- compare(optimal_classified, churn_test$CHURN)

# Output the optimal threshold and the confusion matrix
print(paste("Optimal threshold:", optimal_threshold))
```

```
## [1] "Optimal threshold: 0.632703131414302"
```

```
print(optimal_cm)
```

```
##          Predict 1 Predict 0
## Actual 1        33         6
## Actual 0         9        50
```

```
# Classify outcomes based on optimal threshold
predicted_outcomes <- ifelse(predicted_probabilities >= optimal_threshold, 'likely churn', 'u
nlikely churn')
predicted_outcomes
```

```
##               1               2               3               4
## "unlikely churn" "unlikely churn"   "likely churn" "unlikely churn"
##               5               6               7               8
##   "likely churn" "unlikely churn" "unlikely churn" "unlikely churn"
##               9              10              11              12
## "unlikely churn" "unlikely churn" "unlikely churn" "unlikely churn"
##              13              14              15              16
## "unlikely churn" "unlikely churn" "unlikely churn" "unlikely churn"
##              17              18              19              20
## "unlikely churn" "unlikely churn" "unlikely churn" "unlikely churn"
##              21              22              23              24
## "unlikely churn" "unlikely churn" "unlikely churn"   "likely churn"
##              25              26              27              28
## "unlikely churn"   "likely churn"   "likely churn" "unlikely churn"
##              29              30              31              32
## "unlikely churn" "unlikely churn" "unlikely churn" "unlikely churn"
##              33              34              35              36
## "unlikely churn" "unlikely churn" "unlikely churn" "unlikely churn"
##              37              38              39              40
## "unlikely churn" "unlikely churn" "unlikely churn" "unlikely churn"
##              41              42              43              44
## "unlikely churn" "unlikely churn"   "likely churn"   "likely churn"
##              45              46              47              48
## "unlikely churn" "unlikely churn" "unlikely churn" "unlikely churn"
##              49              50              51              52
## "unlikely churn"   "likely churn" "unlikely churn"   "likely churn"
##              53              54              55              56
## "unlikely churn" "unlikely churn" "unlikely churn" "unlikely churn"
##              57              58              59              60
## "unlikely churn" "unlikely churn" "unlikely churn" "unlikely churn"
##              61              62              63              64
##   "likely churn" "unlikely churn"   "likely churn"   "likely churn"
##              65              66              67              68
##   "likely churn"   "likely churn" "unlikely churn"   "likely churn"
##              69              70              71              72
##   "likely churn"   "likely churn"   "likely churn"   "likely churn"
##              73              74              75              76
##   "likely churn"   "likely churn"   "likely churn"   "likely churn"
##              77              78              79              80
##   "likely churn"   "likely churn"   "likely churn"   "likely churn"
##              81              82              83              84
##   "likely churn"   "likely churn"   "likely churn" "unlikely churn"
##              85              86              87              88
##   "likely churn"   "likely churn"   "likely churn"   "likely churn"
##              89              90              91              92
## "unlikely churn"   "likely churn"   "likely churn" "unlikely churn"
##              93              94              95              96
##   "likely churn"   "likely churn"   "likely churn"   "likely churn"
##              97              98
##   "likely churn"   "likely churn"
```

```
# Compute confusion matrix and associated statistics
#conf_matrix <- confusionMatrix(as.factor(predicted_outcomes), as.factor(churn_test$CHURN))
#print(conf_matrix)
fit1 <- glm(CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT + COMPLAINT,data=churn_trai
n,family=binomial())
summary(fit1)
```

```
##
## Call:
## glm(formula = CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT +
##     COMPLAINT, family = binomial(), data = churn_train)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          6.72099    0.48041  13.990  < 2e-16 ***
## TOT_ACTV_SRV_CNT    -0.54745    0.06249  -8.760  < 2e-16 ***
## AGE                 -0.17921    0.01275 -14.051  < 2e-16 ***
## PCT_CHNG_IB_SMS_CNT -0.41796    0.14377  -2.907  0.00365 **
## COMPLAINT            0.50512    0.22278   2.267  0.02337 *
## ---
## 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:  724.17  on 978  degrees of freedom
## AIC: 734.17
##
## Number of Fisher Scoring iterations: 6
```

```
# create variable y.train = observed values of Y in training set
y.train=churn_train$CHURN

#compute predicted outcome based on probability threshold equal to 0.5
yc = classify(fitted(fit1), 0.6327031)

# compares predicted outcomes with actual values in training set
m=compare(classify(fitted(fit), 0.6327031), y.train)
m
```

```
##          Predict 1 Predict 0
## Actual 1       370        98
## Actual 0        63       452
```

```
#classification metrics
sensitivity(m)
```

```
## [1] 0.8545035
```

```
accuracy(m)
```

```
## [1] 0.8362157
```

```
precision(m)
```

```
## [1] 0.7905983
```

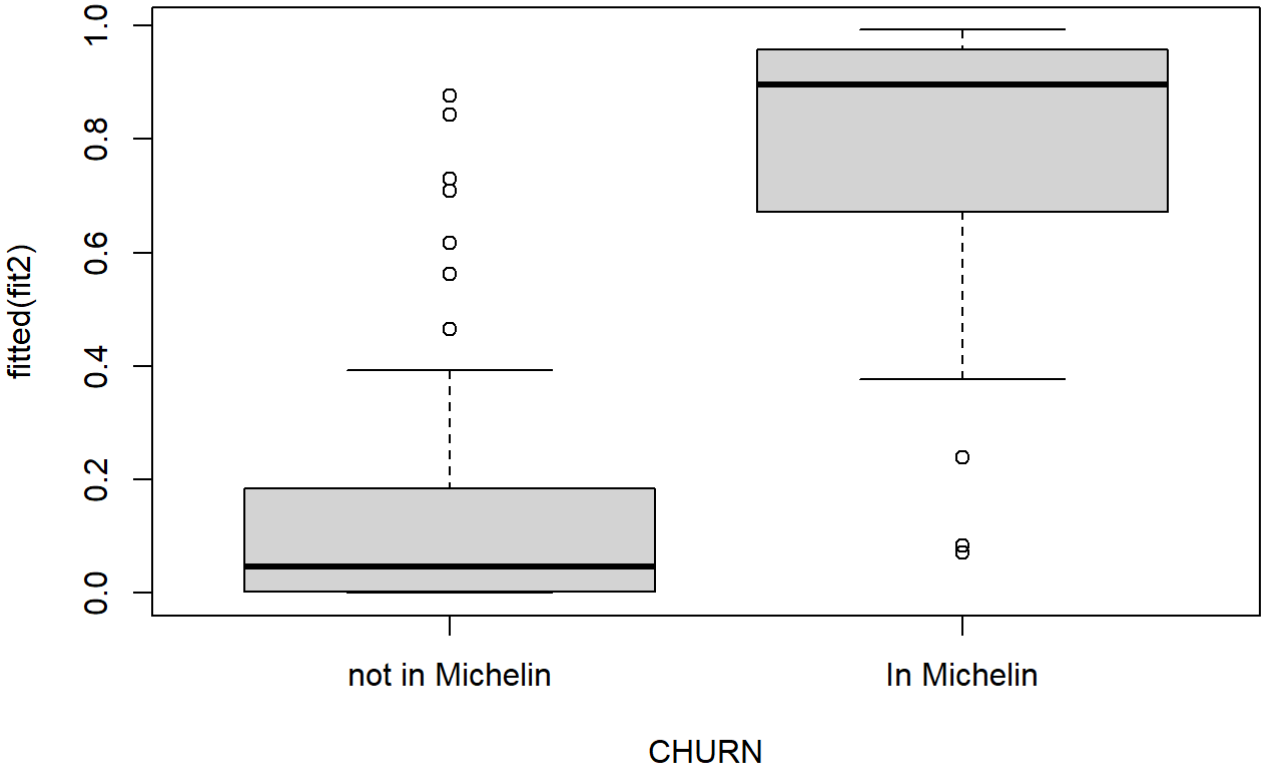
```
recall(m)
```

```
## [1] 0.8545035
```

```
#confusion matrix for test
test.myd<-read.csv("churn_test.csv")

# logistic regression model fitted using glm() function with family=binomial
#fit selected model on training set
fit2 <- glm(CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT + COMPLAINT,data=test.myd,fa
mily=binomial())
summary(fit2) # display results
```

```
##
## Call:
## glm(formula = CHURN ~ TOT_ACTV_SRV_CNT + AGE + PCT_CHNG_IB_SMS_CNT +
##     COMPLAINT, family = binomial(), data = test.myd)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         10.35313    2.39403   4.325 1.53e-05 ***
## TOT_ACTV_SRV_CNT    -0.86057    0.26590  -3.236 0.001211 **
## AGE                 -0.21059    0.05453  -3.862 0.000113 ***
## PCT_CHNG_IB_SMS_CNT -2.87018    0.92380  -3.107 0.001890 **
## COMPLAINT            0.47037    0.92260   0.510 0.610167
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 131.746  on 97  degrees of freedom
## Residual deviance:  55.386  on 93  degrees of freedom
## AIC: 65.386
##
## Number of Fisher Scoring iterations: 7
```

```
#boxplot of predicted probabilities by Yvariable
# useful visualization for classification purposes
boxplot(fitted(fit2)~CHURN, data=test.myd,
        names=c("not in Michelin", "In Michelin"))
```




```
# using functions in Classify_functions.R file to compute classification metrics
# file must be in same work directory.
source("Classify_functions.R")

# create variable  = observed values of Y
y.test=test.myd$CHURN

#compute predicted outcome based on probability threshold equal to 0.5
yc = classify(fitted(fit2), 0.5)

# compares predicted outcomes with actual values in training set
m=compare(classify(fitted(fit2), 0.50), y.train)
m
```

```
##          Predict 1 Predict 0
## Actual 1         0         0
## Actual 0        39        59
```

```
#classification metrics
sensitivity(m)
```

```
## [1] 0
```

```
accuracy(m)
```

```
## [1] 0.6020408
```

```
precision(m)
```

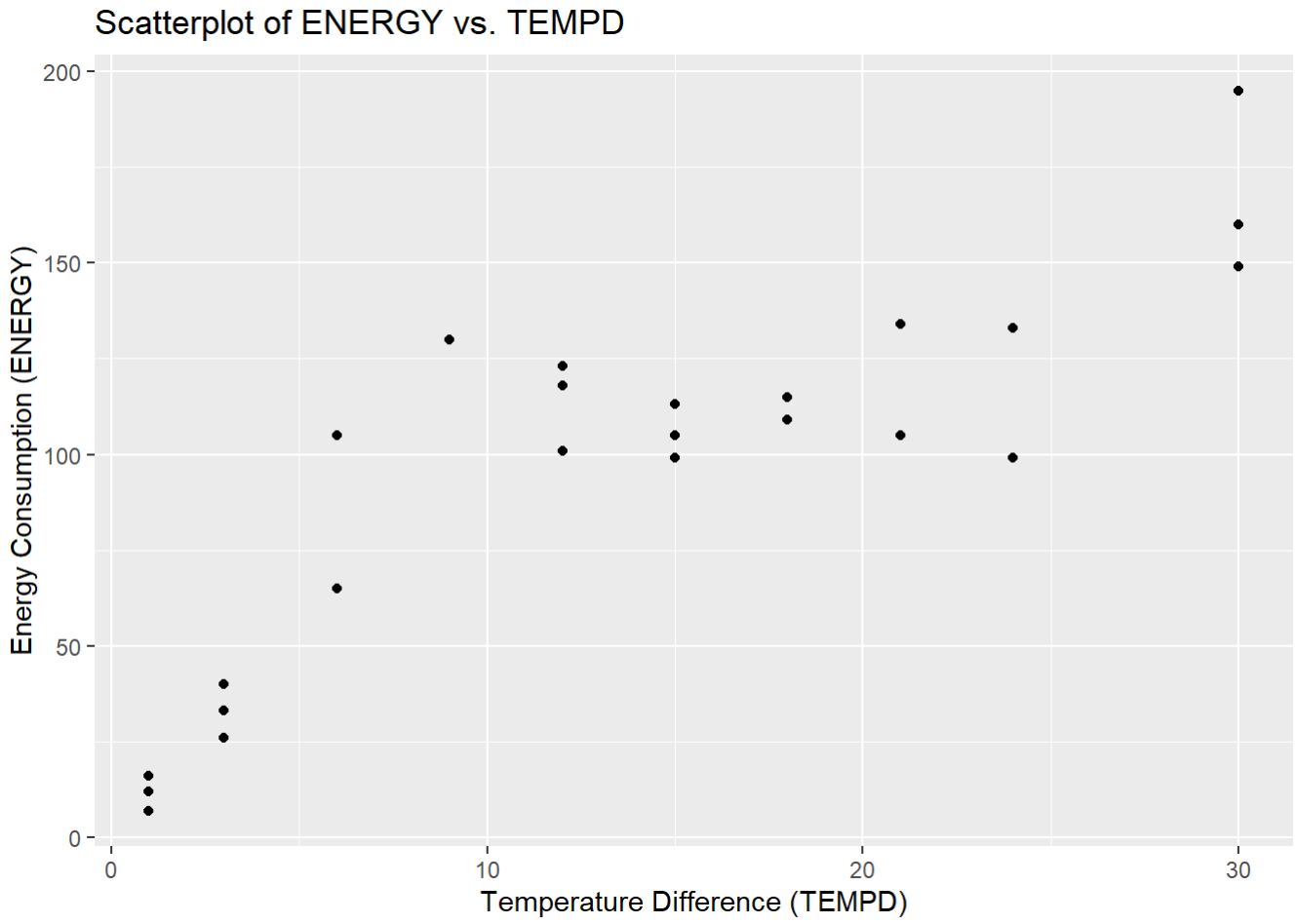
```
## [1] NaN
```

```
recall(m)
```

```
## [1] 0
```

Problem 2

```
# Install and load required packages
library(ggplot2)

# Read the data from the energytemp.txt file
data <- read.table("energytemp.txt", header = TRUE)

#problem 2(A)
# Create a scatterplot
ggplot(data, aes(x = temp, y = energy)) +
  geom_point() +
  labs(title = "Scatterplot of ENERGY vs. TEMPD",
       x = "Temperature Difference (TEMPD)",
       y = "Energy Consumption (ENERGY)")
```



```
correlation_coefficient <- cor(data$temp, data$energy)
print(paste("Correlation Coefficient:", correlation_coefficient))
```

```
## [1] "Correlation Coefficient: 0.868277043378796"
```

2)a) There seems to be a positive relationship between TEMPD and ENERGY, as the points suggest that higher temperature differences are associated with higher energy consumption.

The distribution of points does not appear to follow a simple linear trend, which might suggest the relationship could be better modeled by a nonlinear function. This is indicated by the varying spread of energy values at different temperature differences, which could imply a polynomial relationship. There are no clear outliers that deviate significantly from the overall pattern, suggesting a consistent relationship across the observed range of temperature differences.

#problem 2(B)

```
# Assuming 'data' is your dataset with columns 'TEMPD' and 'ENERGY'
data$TEMPD2 <- data$temp^2
data$TEMPD3 <- data$temp^3

# Fit a cubic regression model
cubic_model <- lm(energy ~ temp + TEMPD2 + TEMPD3, data = data)

# Display the summary of the cubic model
summary(cubic_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
```

#problem 2(c)

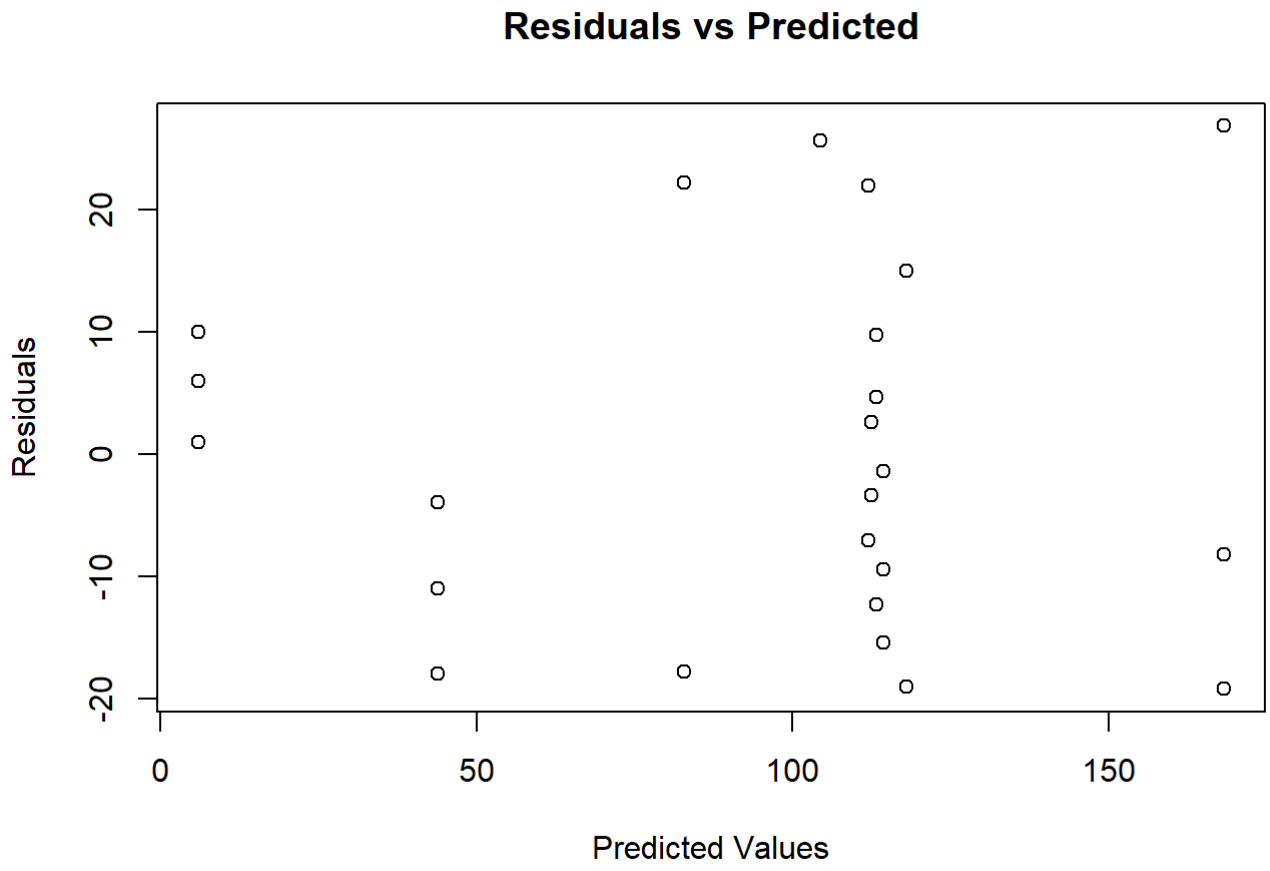
```
summary(cubic_model)
```

```
##
## 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
```

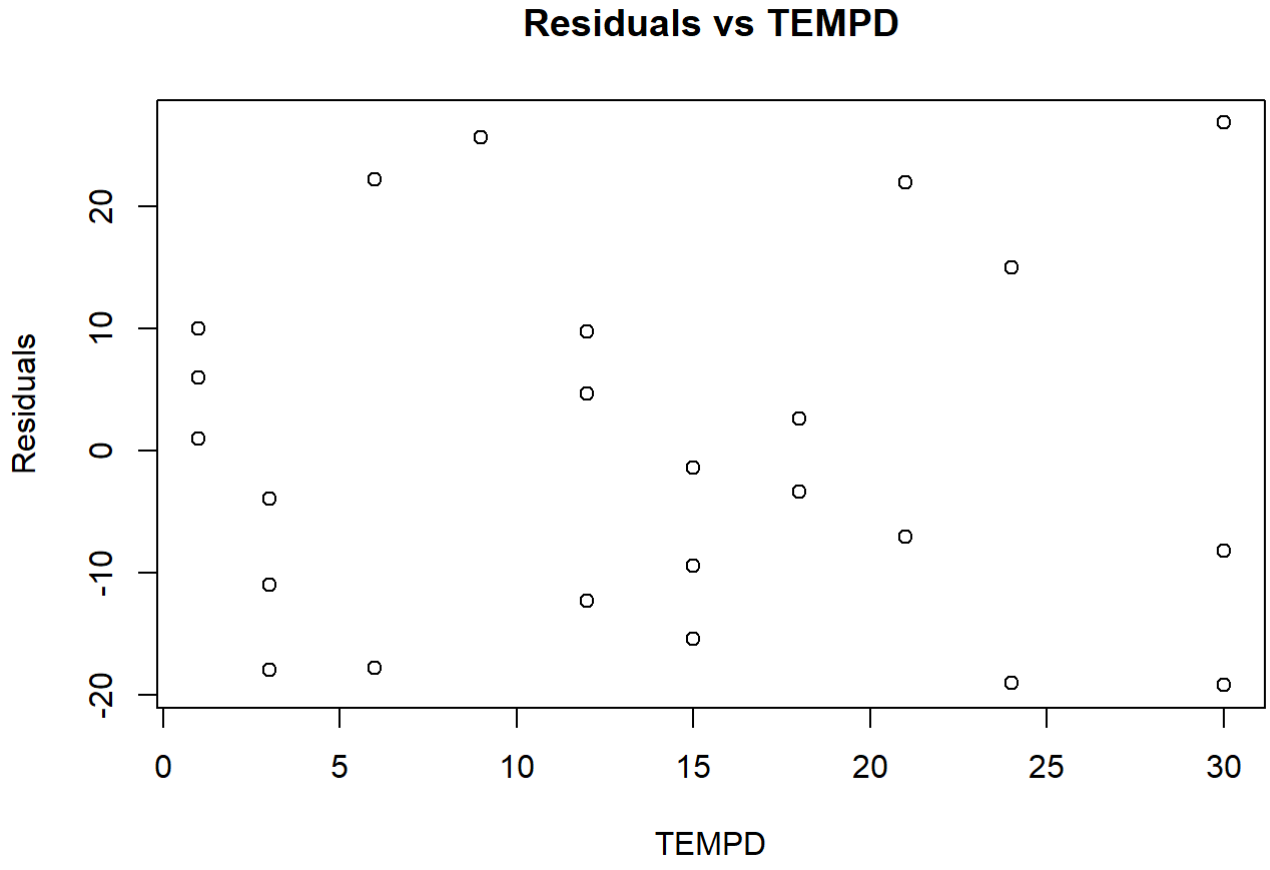
In the above summary, by the observation of probability values, all the varibles are significant as they are less than the p value which is 0.05

#problem 2(D)

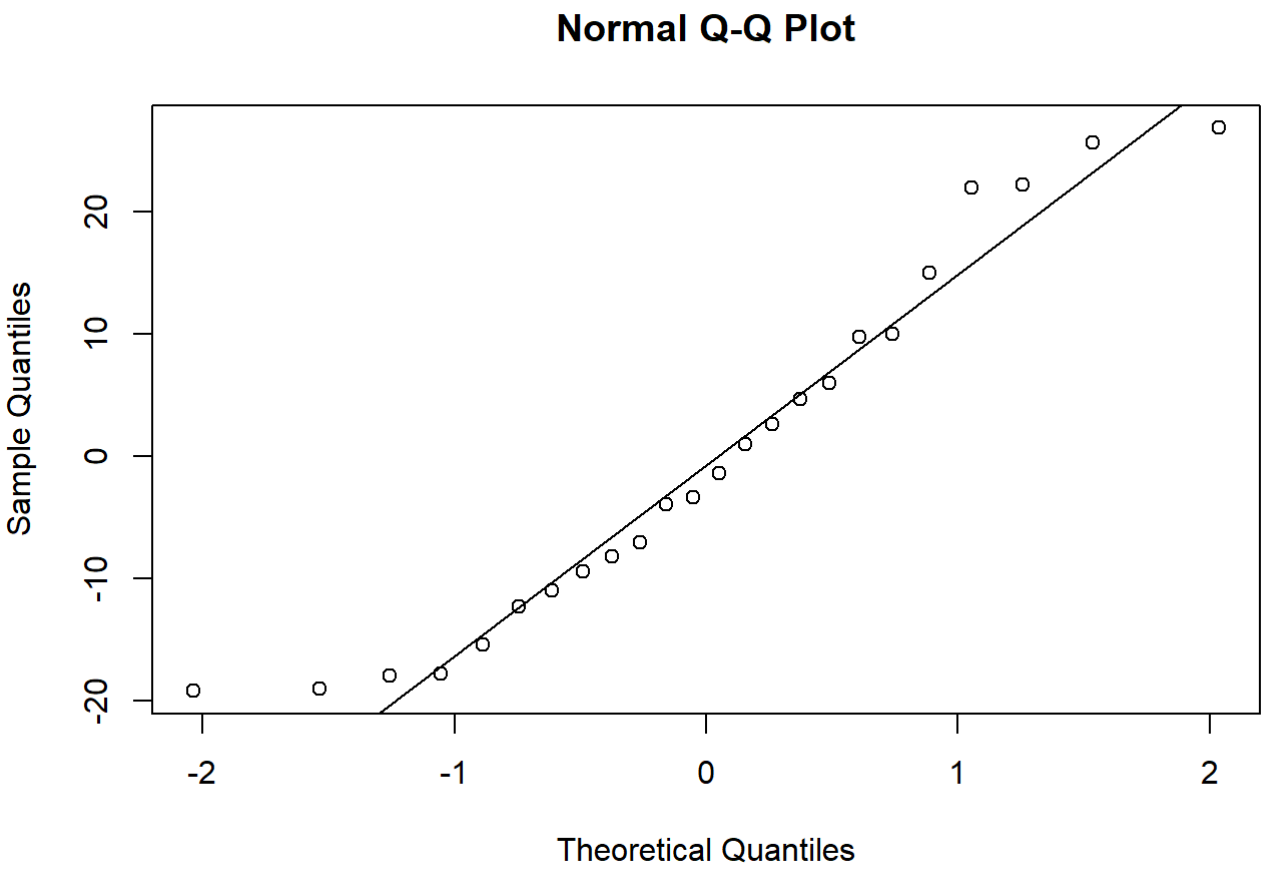
```
# Obtain the residuals from the cubic model
residuals <- residuals(cubic_model)


# Residuals vs Predicted
plot(predict(cubic_model), residuals, main = "Residuals vs Predicted", xlab = "Predicted Valu
es", ylab = "Residuals")
```



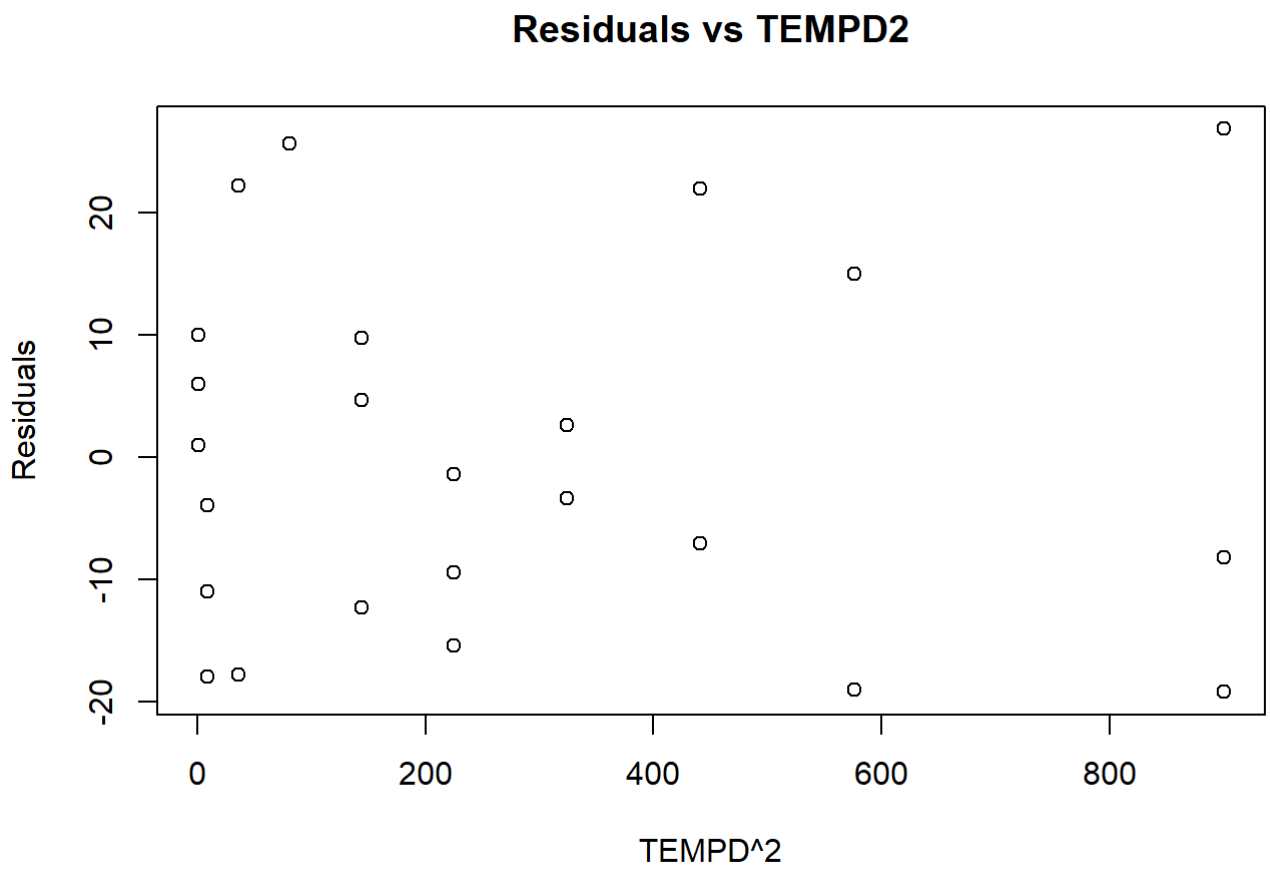

```
# Residuals vs TEMPD
plot(data$temp, residuals, main = "Residuals vs TEMPD", xlab = "TEMPD", ylab = "Residuals")
```



```
# Normal Q-Q Plot
qqnorm(residuals, main = "Normal Q-Q Plot")
qqline(residuals)
```




```
# Residuals vs TEMPD2
plot(data$TEMPD2, residuals, main = "Residuals vs TEMPD2", xlab = "TEMPD^2", ylab = "Residual
s")
```




```
# Reset plotting parameters to default
par(mfrow = c(1, 1))
```

2)d) The Residuals vs Fitted and Residuals vs TEMPD plots do not show any clear patterns, which indicates that the model's assumptions of linearity and homoscedasticity are reasonable for this data.

The Normal Q-Q Plot shows slight deviations from normality at the tails, but overall, it suggests that the normality assumption is not severely violated.

Overall, these residual plots suggest that the cubic model is a reasonable fit for the data. The assumptions of linearity, homoscedasticity, and normality of residuals are not strongly contradicted by the plots. However, slight deviations from normality should be considered, and if this were a concern, transformations or other models might be explored

#problem 2(E)

```
# Assuming 'cubic_model' is your fitted cubic regression model
model_summary <- summary(cubic_model)
model_summary
```

```
##
## 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
```

```
# Extract coefficients
coefficients <- coef(cubic_model)

# Print the coefficients
print(coefficients)
```

```
##  (Intercept)         temp       TEMPD2       TEMPD3
## -17.03623156  24.52399926  -1.49002930   0.02927784
```

```
# Create the expression of the cubic model
expression <- paste("ENERGY =", round(coefficients[1], 4), "+",
                    round(coefficients[2], 4), "*TEMPD +",
                    round(coefficients[3], 4), "*TEMPD^2 +",
                    round(coefficients[4], 4), "*TEMPD^3")

# Print the expression
print(expression)
```

```
## [1] "ENERGY = -17.0362 + 24.524 *TEMPD + -1.49 *TEMPD^2 + 0.0293 *TEMPD^3"
```

The fitted regression model expression is: ENERGY = -17.0362 + 24.524 *TEMPD + -1.49* TEMPD^2 + 0.0293 *TEMPD^3 #problem 2(F)

```r
# Create a new data frame with the specified values
new_data <- data.frame(temp = 10, TEMPD2 = 10^2, TEMPD3 = 10^3)


# Predict average energy consumption using the cubic model
predicted_energy <- predict(cubic_model, newdata = new_data)

# Print the predicted energy consumption
cat("Predicted Energy Consumption:", predicted_energy, "\n")
```

```
## Predicted Energy Consumption: 108.4787
```

2)F)The predicted average energy consumption for an average temperature difference (TEMPD) of 10°F using the fitted cubic regression model is approximately 108.48 units.