Assignment 2

Brian Choi

2023-05-11

Question 1

Part A

$$g(w) = \frac{1}{2}\Sigma_{i=1}^{m}(y_i - w^T x_i)^2$$
$$g'(w) = \Sigma_{i=1}^{m}\{(y_i - w^T x_i)(-x_i T w^{T-1})\}$$

$$g''(w) = \Sigma_{i=1}^{m}\{(y_i - w^T x_i)(-x_i T(T-1)w^{T-2} + (-x_i T w^{T-1})(-x_i T w^{T-1})\}$$
$$= \Sigma_{i=1}^{m}\{(y_i - w^T x_i)(-x_i T(T-1)w^{T-2} + x_i^2 (T w^{T-1})^2\}$$

Since $g''(w) > 0$ the function $g(w)$ is convex.

Part B

$$h(w) = \frac{1}{2}\gamma||w||^2$$
$$h'(w) = \frac{1}{2}\gamma||2w||$$
$$h''(w) = \gamma > 0$$

Since $h''(w) > 0$, $h(w)$ is convex.

Part C

$$f(w) = g(w) + h(w)$$

Since the second derivative of $g(w)$ and $h(w)$ are both positive,

$$f''(w) = g''(w) + h''(w) > 0$$

Part D

To find the minima of $f(w)$,

$$f'(w) = \gamma||w|| + \Sigma_{i=1}^{m}\{(y_i - w^T x_i)(-x_i T w^{T-1})\} = 0$$

$$\gamma||w|| + \Sigma_{i=1}^{m}\{(x_i w^{2T-1} - y_i w^{T-1})\} = 0$$

Question 2

Part 1

$$log(\frac{1}{1-p}) = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$$

where vector x is composed of variables radius_mean, texture_mean, etc. and Y is the outcome variable diagnosis

Part 2

```
load("Breast.Rdata")
```

There are 30 attributes to predict the outcome. There are 456 samples and 113 samples in the train and test data sets respectively. The distribution of the diagnosis variable is a binomial distribution.

Part 3

```
Breast.fit <- glm(formula = factor(diagnosis) ~ ., family = "binomial", data = Breast$train)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(Breast.fit)
```

```
##
## Call:
## glm(formula = factor(diagnosis) ~ ., family = "binomial", data = Breast$train)
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -1.947e+03  1.834e+06  -0.001    0.999
## radius_mean             -6.292e+01  5.539e+05   0.000    1.000
## texture_mean            -4.053e+00  7.211e+03  -0.001    1.000
## perimeter_mean           1.410e+01  8.570e+04   0.000    1.000
## area_mean               -4.277e-01  1.794e+03   0.000    1.000
## smoothness_mean          2.603e+03  2.667e+06   0.001    0.999
## compactness_mean        -1.881e+03  3.620e+06  -0.001    1.000
## concavity_mean           2.672e+03  1.039e+06   0.003    0.998
## concave.points_mean     -1.770e+03  2.392e+06  -0.001    0.999
## symmetry_mean           -8.507e+01  1.584e+06   0.000    1.000
## fractal_dimension_mean  -7.652e+01  9.290e+06   0.000    1.000
## radius_se               -4.560e+02  6.021e+05  -0.001    0.999
## texture_se              -1.183e+02  1.263e+05  -0.001    0.999
## perimeter_se             5.475e+01  6.250e+04   0.001    0.999
## area_se                  3.338e+00  8.692e+03   0.000    1.000
## smoothness_se            8.261e+03  2.730e+07   0.000    1.000
```

```
## compactness_se           9.646e+03  6.792e+06   0.001    0.999
## concavity_se            -5.003e+03  4.997e+06  -0.001    0.999
## concave.points_se        1.445e+04  9.061e+06   0.002    0.999
## symmetry_se             -4.809e+03  5.275e+06  -0.001    0.999
## fractal_dimension_se    -7.965e+04  6.161e+07  -0.001    0.999
## radius_worst             1.402e+02  1.219e+05   0.001    0.999
## texture_worst            1.492e+01  1.078e+04   0.001    0.999
## perimeter_worst         -9.594e+00  2.223e+04   0.000    1.000
## area_worst              -4.863e-01  9.480e+02  -0.001    1.000
## smoothness_worst        -1.703e+02  2.727e+06   0.000    1.000
## compactness_worst       -9.309e+02  1.276e+06  -0.001    0.999
## concavity_worst          1.289e+02  7.715e+05   0.000    1.000
## concave.points_worst     4.287e+01  1.767e+06   0.000    1.000
## symmetry_worst           6.023e+02  3.107e+05   0.002    0.998
## fractal_dimension_worst  7.467e+03  5.395e+06   0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6.0231e+02  on 455  degrees of freedom
## Residual deviance: 1.6445e-07  on 425  degrees of freedom
## AIC: 62
##
## Number of Fisher Scoring iterations: 25
```

Part 4

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
Breast.prob = predict(Breast.fit, Breast$train, type="response")
Breast.pred = rep("B", dim(Breast$train)[1])
Breast.pred[Breast.prob > 0.5] = "M"
caret::confusionMatrix(data = factor(Breast.pred), reference = factor(Breast$train$diagnosis))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   B   M
##          B 286   0
##          M   0 170
##
##                Accuracy : 1
##                  95% CI : (0.9919, 1)
##     No Information Rate : 0.6272
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
```

```
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6272
##          Detection Rate : 0.6272
##    Detection Prevalence : 0.6272
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : B
##
```

Part 5

```
Breast.prob <- predict(Breast.fit, Breast$test, type="response")
Breast.pred <- rep("B", dim(Breast$train)[1])
Breast.pred[Breast.prob > 0.5] <- "M"
caret::confusionMatrix(data = factor(Breast.pred), reference <- factor(Breast$train$diagnosis))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   B   M
##          B 194  98
##          M  92  72
##
##                Accuracy : 0.5833
##                  95% CI : (0.5366, 0.629)
##     No Information Rate : 0.6272
##     P-Value [Acc > NIR] : 0.9758
##
##                   Kappa : 0.1026
##
##  Mcnemar's Test P-Value : 0.7168
##
##             Sensitivity : 0.6783
##             Specificity : 0.4235
##          Pos Pred Value : 0.6644
##          Neg Pred Value : 0.4390
##              Prevalence : 0.6272
##          Detection Rate : 0.4254
##    Detection Prevalence : 0.6404
##       Balanced Accuracy : 0.5509
##
##        'Positive' Class : B
##
```

Part 6

The test set accuracy is 58.33% whereas the train set accuracy is 100%. The accuracy for the train set is the accuracy of the model on the data it was constructed on whereas the test set accuracy is the accuracy of the model on data it has not yet seen. Since the test set accuracy is accuracy based off unseen data, the test set accuracy should be used for the performance of the classifier.

Part 7

Ridge regression includes all of the features in the model whereas Lasso regression also performs feature selection. However, ridge regression is works well with highly correlated features and is good for over fitted models. Lasso regression is used for models with very high number of features and can set the estimates of coefficients to exactly zero. Since a parsimonious model is desired, the lasso regression should be considered as it allows variable selection and therefore an advantage in interpretation.

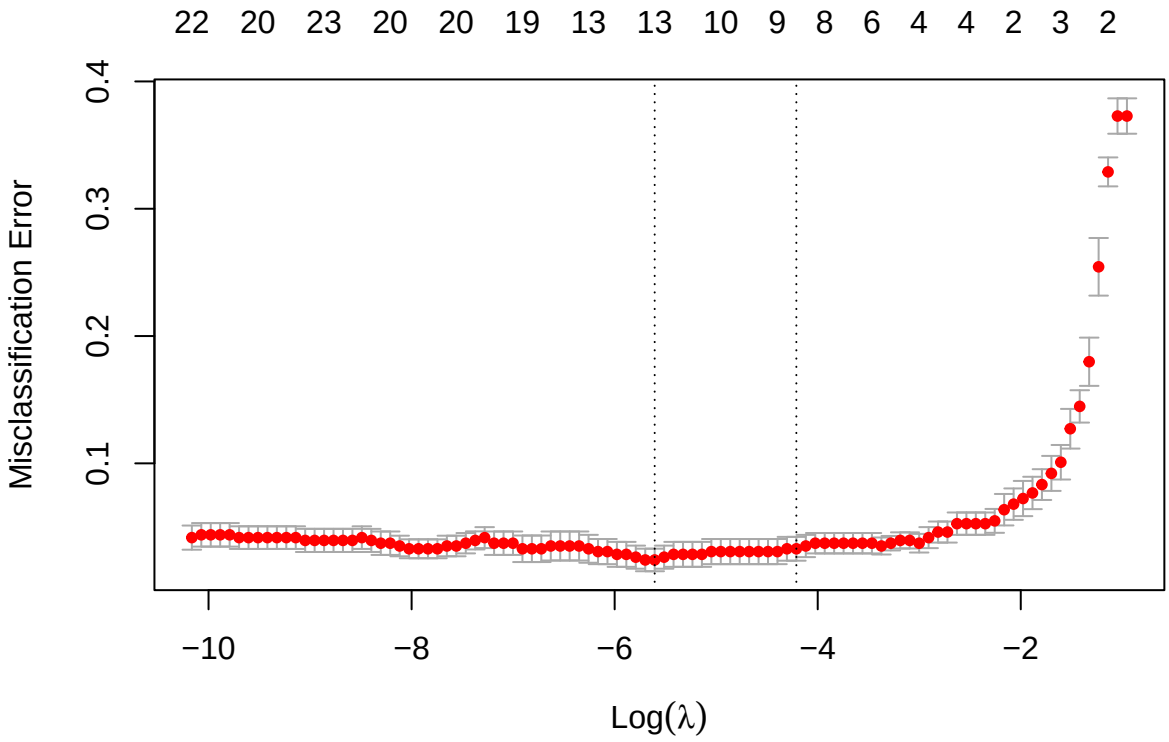
Part 8

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-7
```

```
Breast.lasso <- cv.glmnet(x=as.matrix(Breast$train[2:31]), y=Breast$train$diagnosis, alpha=1, family =
plot(Breast.lasso)
```



Part 9

```
lasso <- glmnet(x=as.matrix(Breast$train[2:31]), y=Breast$train$diagnosis, lambda=Breast.lasso$lambda.1s
colnames(as.matrix(Breast$train[2:31]))[lasso$beta[,1]!=0]
```

```
## [1] "concave.points_mean"  "radius_se"            "radius_worst"
## [4] "texture_worst"        "smoothness_worst"     "concavity_worst"
## [7] "concave.points_worst" "symmetry_worst"
```

There are 9 attributes selected in the model.

Part 10

```
lasso.prob <- predict(lasso, newx=as.matrix(Breast$test[2:31]), type="response")
lasso.pred <- rep("B", dim(Breast$train)[1])
lasso.pred[lasso.prob > 0.5] <- "M"
caret::confusionMatrix(data = factor(lasso.pred), reference = factor(Breast$train$diagnosis))
```

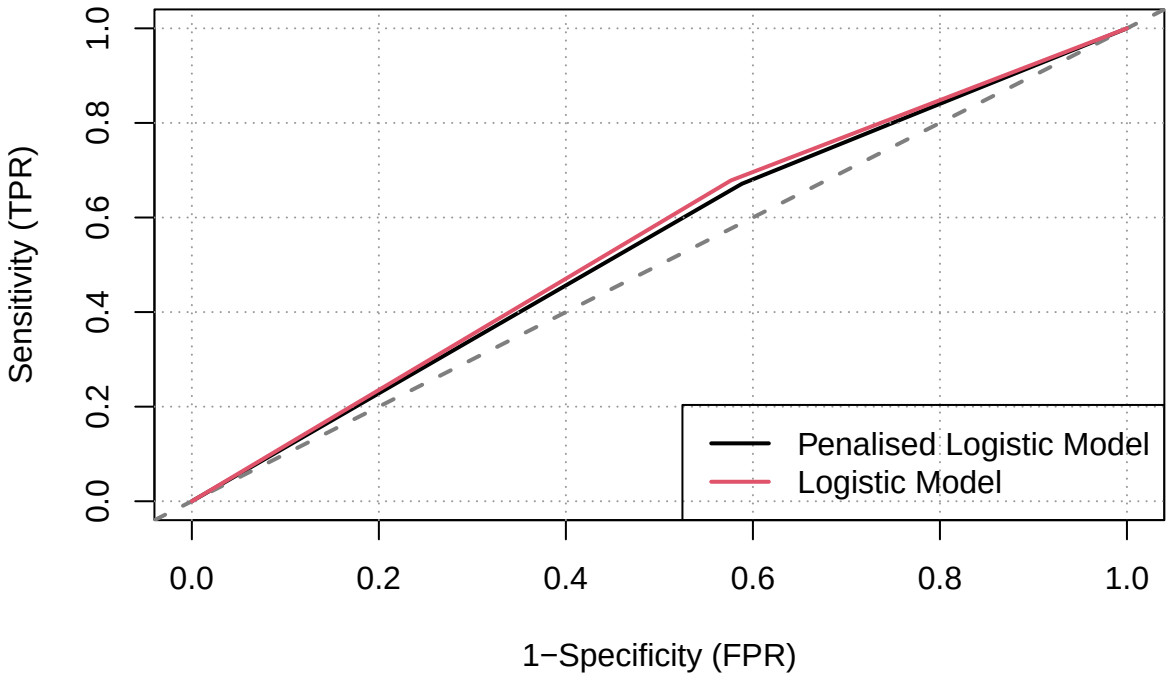
```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction   B   M
##          B 192 100
##          M  94  70
##
##                Accuracy : 0.5746
##                  95% CI : (0.5277, 0.6204)
##     No Information Rate : 0.6272
##     P-Value [Acc > NIR] : 0.9908
##
##                   Kappa : 0.0837
##
##  Mcnemar's Test P-Value : 0.7196
##
##             Sensitivity : 0.6713
##             Specificity : 0.4118
##          Pos Pred Value : 0.6575
##          Neg Pred Value : 0.4268
##              Prevalence : 0.6272
##          Detection Rate : 0.4211
##    Detection Prevalence : 0.6404
##       Balanced Accuracy : 0.5415
##
##        'Positive' Class : B
##
```

Part 11

```
library(ROCit)
roc.model1 <- rocit(score=as.numeric(lasso.pred == "B"), class=as.numeric(Breast$train$diagnosis=="B"))
roc.model2 <- rocit(score=as.numeric(Breast.pred == "B"), class=as.numeric(Breast$train$diagnosis=="B"))
```

```
plot(roc.model1, col = c(1,"gray50"), legend = FALSE, YIndex = FALSE)
lines(roc.model2$TPR~roc.model2$FPR, col = 2, lwd = 2)
legend("bottomright", col = c(1,2),
c("Penalised Logistic Model", "Logistic Model"), lwd = 2)
```



Part 12

The AUC for model penalised logistic model and logistic model respectively is:

```
roc.model1$AUC
```

```
## [1] 0.5415467
```

```
roc.model2$AUC
```

```
## [1] 0.5509255
```

Since the logistic model has a slightly higher AUC, it should be chosen.

Question 3

Part 1

```
library(boot)
```

```
##
## Attaching package: 'boot'

## The following object is masked from 'package:ROCit':
##
##     logit

## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
library(simcausal)
library(moments)
n=500
set.seed(42)
x=rbern(n, 0.45)
```

Part 2

```
m=1000
bootmom <- c()
for (i in 1:m)
{
    obs <- sample(1:n, replace=TRUE)
    bootmom[i] <- mean(x[obs])
}

bootmle <- c()
for (i in 1:m)
{
    obs <- sample(1:n, replace=TRUE)
    bootmle[i] <- min(mean(x[obs]),0.5)
}
```

Part 3

```
sd.mle =  sd(bootmle)
sd.mom = sd(bootmom)
sd.mle
```

```
## [1] 0.0226012
```

```
sd.mom
```

```
## [1] 0.02188229
```

```
biased.mle=mean(bootmle)-min(mean(x),0.5)
biased.mom=mean(bootmom)-mean(x)
biased.mle
```

```
## [1] -0.000746
```

```
biased.mom
```

```
## [1] -0.000566
```

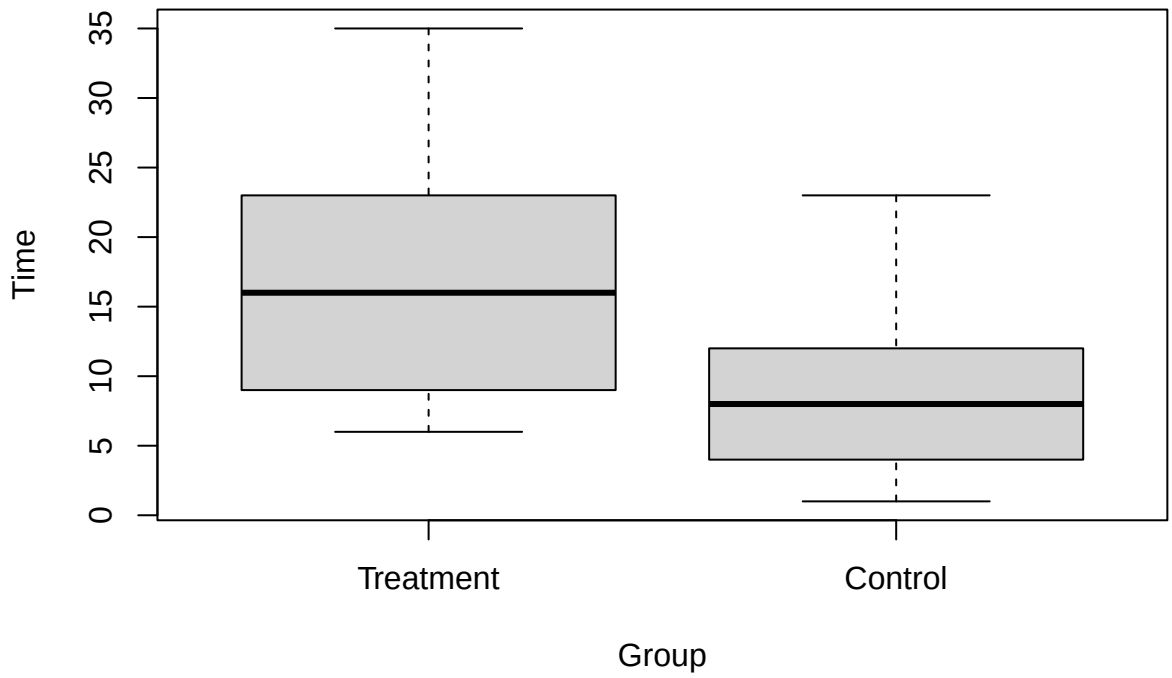
The performance for the mom estimate is better since the estimated bias is smaller relative to the estimate of the standard error.

Part 4

Question 4

Part 1

```
data = read.csv("remiss.csv")
boxplot(data$time~data$group, names=c("Treatment","Control"),xlab = "Group", ylab="Time")
```



The remission times are higher for the treatment group compared to the control group.

Part 2

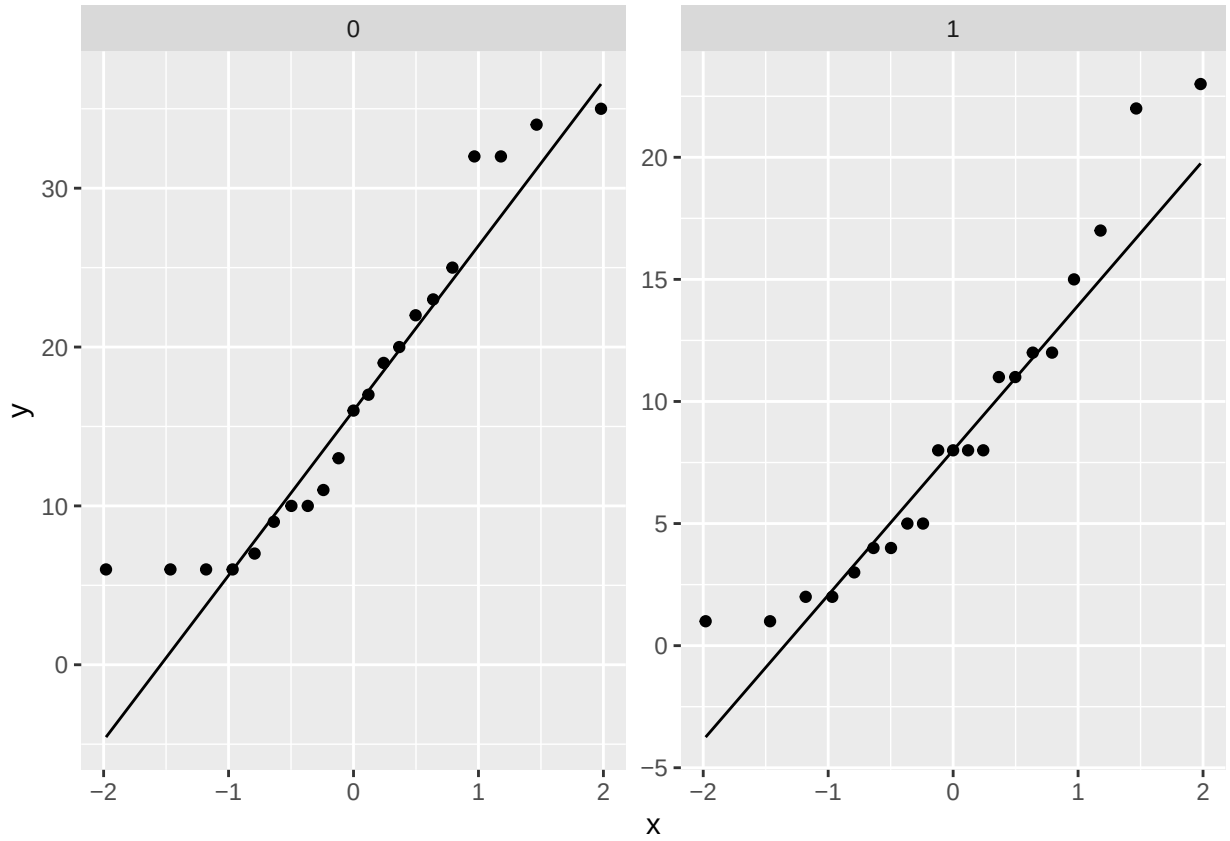
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
data %>%
  ggplot(aes(sample = time)) +
  geom_qq() + geom_qq_line() +
  facet_wrap(~group, scales = "free_y")
```



The first probability plot is the treatment plot and the second plot is the control plot. Both plots indicate a normal distribution.

Part 3

$H_0 : \mu_0 = \mu_1$ vs. $H_1 : \mu_0 \neq \mu_1$

Part 4

```
obs <- mean(data$time[data$group==0])-mean(data$time[data$group==1])
n <- 1000
sim <- numeric(n)
for(i in 1:n){
  shuffled <- sample(data$group)
  sim[i] <- mean(data$time[shuffled==0])-mean(data$time[shuffled==1])
}
alpha <- 0.05
lower_crit <- quantile(sim, alpha/2)
upper_crit <- quantile(sim, 1 - alpha/2)

p_value <- mean(abs(sim) >= abs(obs))
```

```
lower_crit
```

```
##      2.5%
## -5.285714
```

```
upper_crit
```

```
##    97.5%
## 5.666667
```

```
p_value
```

```
## [1] 0.002
```

The lower critical value is -5.38, the upper critical value is 5.76 and the p value is 0.006. Since the p-value is less than the significance level, the null hypothesis is rejected and conclude that there is enough evidence to support the alternative hypothesis.