

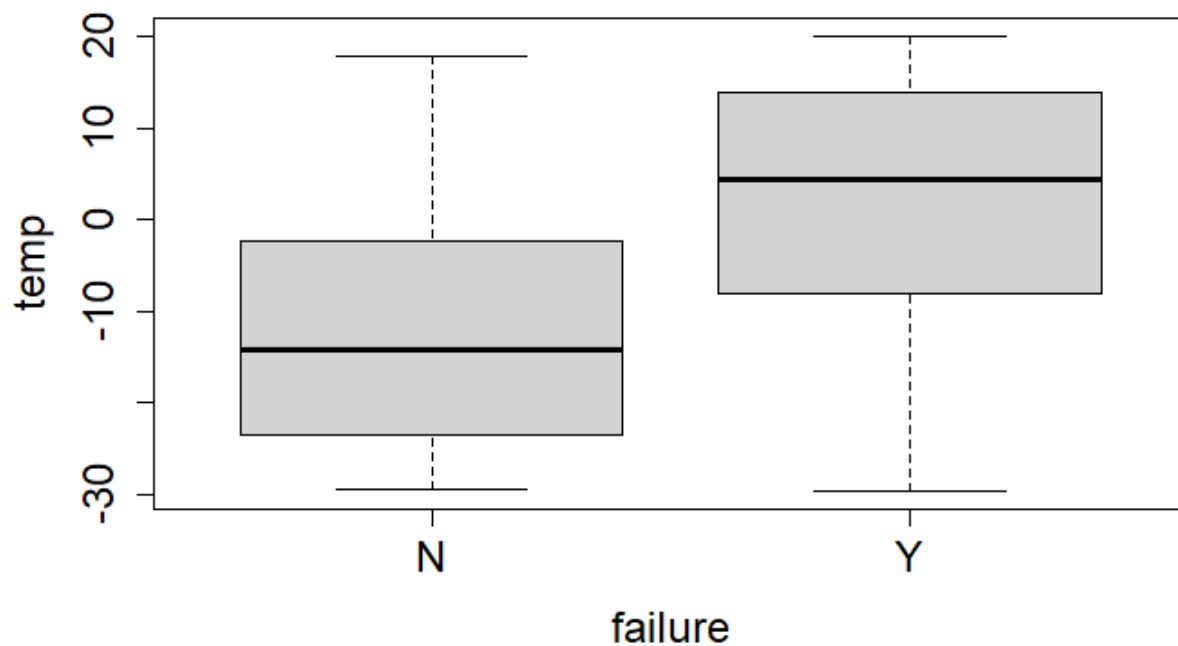
ESI 4606: Analytics I - Foundations of Data Science

Mid-term Exam - Part I

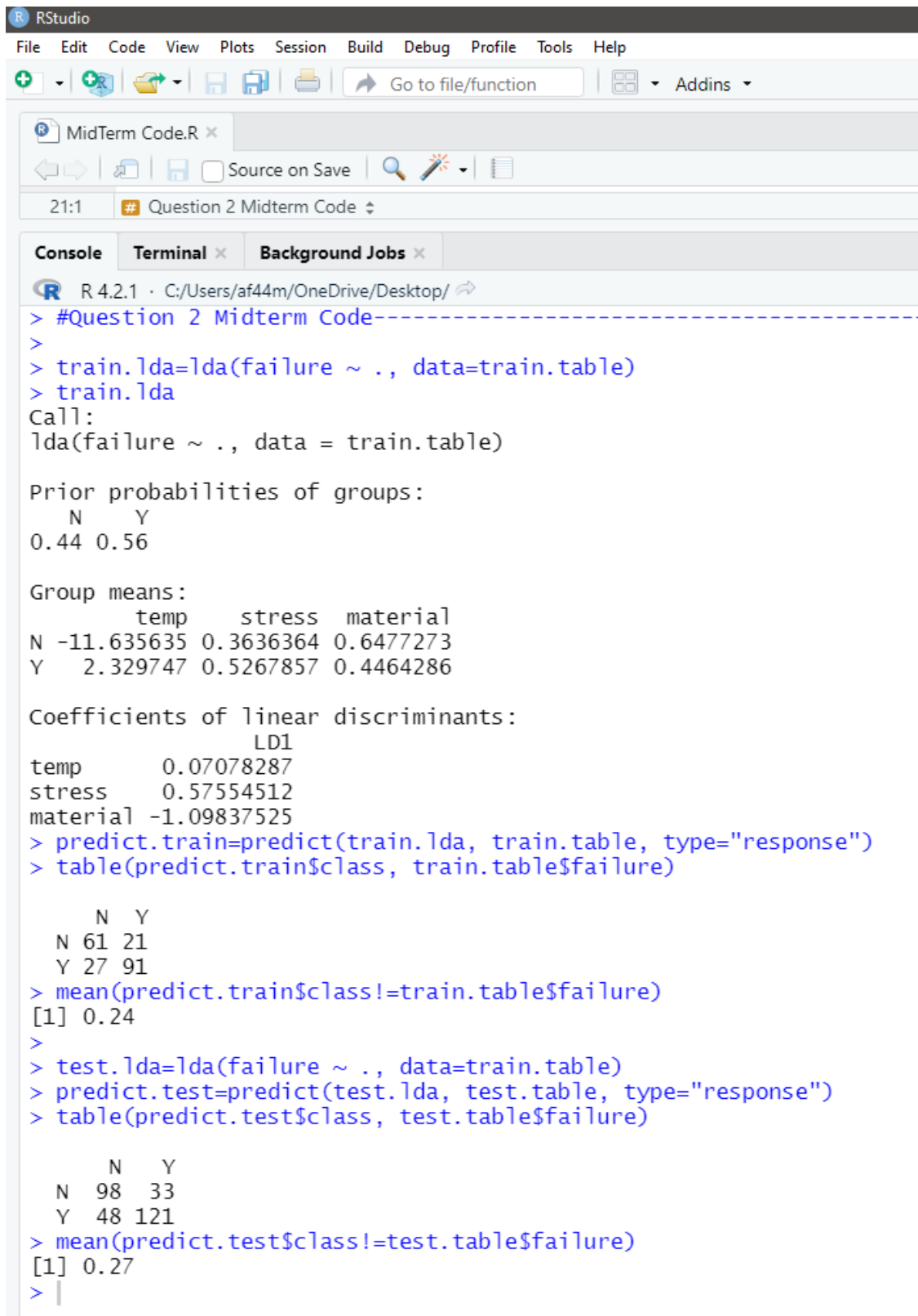
Due: October 26st (11:30PM), 2022

Morgan Harrison

1. The boxplots data visualization reveals that the median of Yes is higher than No. Based on the graphic created and the data, Yes and No have similar dispersions and no extreme recognizable outliers.



2. See Following Information.



The screenshot shows the RStudio interface with the following components:

- Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for file operations (new, open, save, print) and a search bar labeled "Go to file/function".
- Source Editor:** Displays a file named "MidTerm Code.R" with line 21:1 selected. The code is titled "# Question 2 Midterm Code".
- Console:** Shows the output of the R code execution.

The R code and its output in the console are as follows:

```
> #Question 2 Midterm Code-----
>
> train.lda=lda(failure ~ ., data=train.table)
> train.lda
Call:
lda(failure ~ ., data = train.table)

Prior probabilities of groups:
      N      Y
0.44 0.56

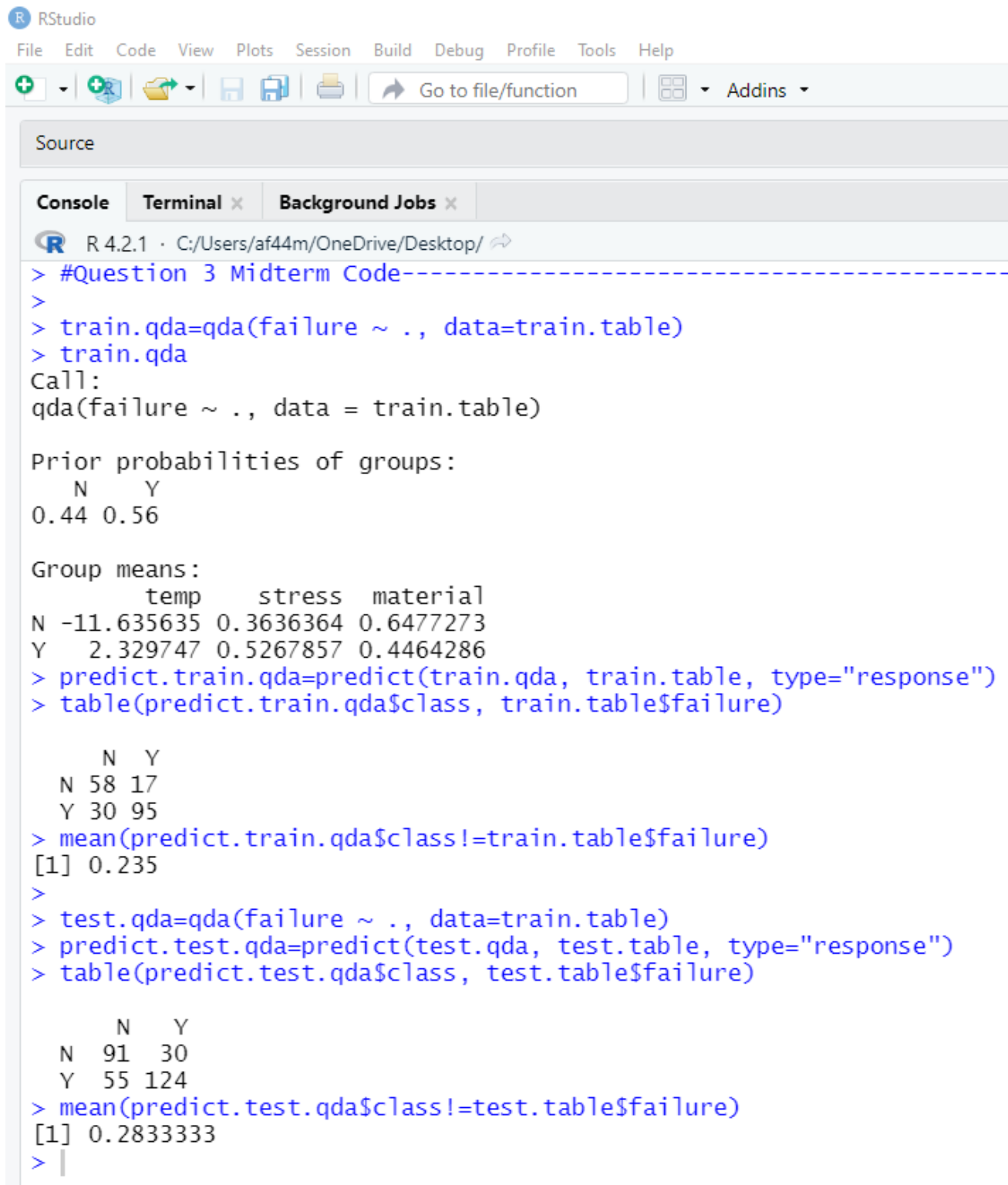
Group means:
      temp      stress      material
N -11.635635  0.3636364  0.6477273
Y   2.329747  0.5267857  0.4464286

Coefficients of linear discriminants:
              LD1
temp          0.07078287
stress         0.57554512
material     -1.09837525
> predict.train=predict(train.lda, train.table, type="response")
> table(predict.train$class, train.table$failure)

      N      Y
N  61  21
Y  27  91
> mean(predict.train$class!=train.table$failure)
[1] 0.24
>
> test.lda=lda(failure ~ ., data=train.table)
> predict.test=predict(test.lda, test.table, type="response")
> table(predict.test$class, test.table$failure)

      N      Y
N  98  33
Y  48 121
> mean(predict.test$class!=test.table$failure)
[1] 0.27
> |
```

3. See Following Information.



The screenshot shows the RStudio interface with the console window active. The console displays the execution of several R commands for a Quadratic Discriminant Analysis (QDA). The commands include training a QDA model, predicting on training data, and testing the model on a separate dataset. The output shows the prior probabilities of groups, group means for various variables, and the resulting classification counts and error rates.

```
R 4.2.1 · C:/Users/af44m/OneDrive/Desktop/
> #Question 3 Midterm Code-----
>
> train.qda=qda(failure ~ ., data=train.table)
> train.qda
Call:
qda(failure ~ ., data = train.table)

Prior probabilities of groups:
      N      Y
0.44 0.56

Group means:
      temp      stress      material
N -11.635635  0.3636364  0.6477273
Y   2.329747  0.5267857  0.4464286
> predict.train.qda=predict(train.qda, train.table, type="response")
> table(predict.train.qda$class, train.table$failure)

      N      Y
N 58 17
Y 30 95
> mean(predict.train.qda$class!=train.table$failure)
[1] 0.235
>
> test.qda=qda(failure ~ ., data=train.table)
> predict.test.qda=predict(test.qda, test.table, type="response")
> table(predict.test.qda$class, test.table$failure)

      N      Y
N 91 30
Y 55 124
> mean(predict.test.qda$class!=test.table$failure)
[1] 0.2833333
> |
```

4. See Following Information.

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ [Go to file/function] [Addins]

Source

Console Terminal x Background Jobs x

R 4.2.1 · C:/Users/af44m/OneDrive/Desktop/
glm(formula = failure ~ ., family = binomial, data = train.table)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2475 -0.8085  0.4231  0.8447  2.4392

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.08451    0.32353   3.352 0.000802 ***
temp         0.08628    0.01369   6.302 2.93e-10 ***
stress       0.72828    0.34476   2.112 0.034649 *
material     -1.45557    0.37554  -3.876 0.000106 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 274.37  on 199  degrees of freedom
Residual deviance: 206.49  on 196  degrees of freedom
AIC: 214.49

Number of Fisher Scoring iterations: 4

> predict.train.glm=predict(train.glm, train.table, type="response")
> train.reponse=rep("N", nrow(train.table))
> train.reponse[predict.train.glm>0.5]<-"Y"
> table(train.reponse,train.table$failure)

train.reponse  N   Y
               N  61  21
               Y  27  91

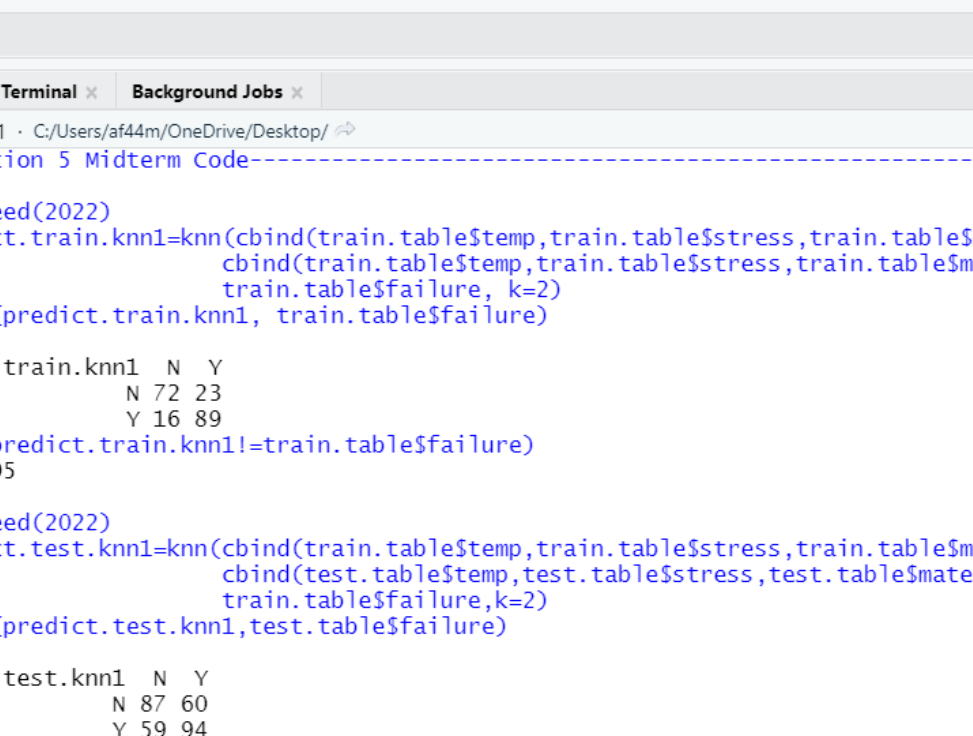
> mean(train.reponse!=train.table$failure)
[1] 0.24

>
> test.glm=glm(failure~., data=train.table, family = binomial)
> predict.test.glm=predict(test.glm, test.table, type="response")
> test.reponse=rep("N", nrow(test.table))
> test.reponse[predict.test.glm>0.5]<-"Y"
> table(test.reponse,test.table$failure)

test.reponse   N   Y
               N  98  32
               Y  48 122

> mean(test.reponse!=test.table$failure)
[1] 0.2666667
```

5. See Following Information.



The screenshot shows the RStudio environment with the following components:

- Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for file operations (new, open, save, print) and a search bar labeled "Go to file/function".
- Source Panel:** Displays the R script being edited.
- Console Panel:** Shows the execution of R commands and their output.
 - Commands:


```
> #Question 5 Midterm Code-----
>
> set.seed(2022)
> predict.train.knn1=knn(cbind(train.table$temp,train.table$stress,train.table$material),
+                         cbind(train.table$temp,train.table$stress,train.table$material),
+                         train.table$failure, k=2)
> table(predict.train.knn1, train.table$failure)

predict.train.knn1  N  Y
                  N 72 23
                  Y 16 89

> mean(predict.train.knn1!=train.table$failure)
[1] 0.195

>
> set.seed(2022)
> predict.test.knn1=knn(cbind(train.table$temp,train.table$stress,train.table$material),
+                       cbind(test.table$temp,test.table$stress,test.table$material),
+                       train.table$failure,k=2)
> table(predict.test.knn1,test.table$failure)

predict.test.knn1  N  Y
                  N 87 60
                  Y 59 94

> mean(predict.test.knn1!=test.table$failure)
[1] 0.3966667
> |
```
 - Output: The console shows the results of the knn function, including confusion matrices and accuracy calculations.

The screenshot shows the RStudio environment with the following components:

- Top Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for file operations (add, open, save, print) and a search bar labeled "Go to file/function".
- Source Panel:** Displays the R script being executed.
- Console Panel:** Shows the execution output, including the R prompt, code execution, and the resulting confusion matrices and mean accuracy values.

The R code and its output are as follows:

```
> #Question 6 Midterm Code-----
>
> set.seed(2022)
> predict.train.knn2=knn(cbind(train.table$temp,train.table$stress,train.table$material),
+                         cbind(train.table$temp,train.table$stress,train.table$material),
+                         train.table$failure, k=7)
> table(predict.train.knn2, train.table$failure)

predict.train.knn2  N  Y
                  N 60 22
                  Y 28 90

> mean(predict.train.knn2!=train.table$failure)
[1] 0.25
>
> set.seed(2022)
> predict.test.knn2=knn(cbind(train.table$temp,train.table$stress,train.table$material),
+                       cbind(test.table$temp,test.table$stress,test.table$material),
+                       train.table$failure,k=7)
> table(predict.test.knn2,test.table$failure)

predict.test.knn2  N  Y
                  N 84 33
                  Y 62 121

> mean(predict.test.knn2!=test.table$failure)
[1] 0.3166667
> |
```

7. See Information Below.

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ [Go to file/function] Addins

Source

Console Terminal Background Jobs
R 4.2.1 · C:/Users/af44m/OneDrive/Desktop/
> #Question 7 Midterm Code-----
>
> train.MLR = train.table
> test.MLR = test.table
>
> train.MLR$failure=rep("0",200)
> train.MLR$failure[train.table$failure=="Y"]="1"
> test.MLR$failure=rep("0",300)
> test.MLR$failure[test.table$failure=="Y"]="1"
> trainlm=lm(failure~., data=train.MLR)
> summary(trainlm)

Call:
lm(formula = failure ~ ., data = train.MLR)

Residuals:
    Min       1Q   Median       3Q      Max
-0.94982 -0.33663  0.06521  0.34625  1.03182

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.695845   0.053489  13.009 < 2e-16 ***
temp         0.016139   0.002003   8.057 7.50e-14 ***
stress       0.131228   0.059800   2.194  0.0294 *
material     -0.250436   0.060037  -4.171 4.55e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4204 on 196 degrees of freedom
Multiple R-squared:  0.2972,    Adjusted R-squared:  0.2865
F-statistic: 27.63 on 3 and 196 DF,  p-value: 6.03e-15

> predict.train.lm=predict(trainlm, newdata=train.MLR, type="response")
> response.train.lm=rep("N",200)
> response.train.lm[predict.train.lm>0.5]="Y"
> table(response.train.lm,train.table$failure)

response.train.lm  N  Y
                  N 61 21
                  Y 27 91
> mean(response.train.lm!=train.table$failure)
[1] 0.24
>
>
> predict.test.lm=predict(trainlm, newdata=test.MLR, type="response")
> response.test.lm=rep("N",300)
> response.test.lm[predict.test.lm>0.5]="Y"
> table(response.test.lm,test.table$failure)

response.test.lm  N  Y
                  N 98 33
                  Y 48 121
> mean(response.test.lm!=test.table$failure)
[1] 0.27
>

```

8. The Logistic Regression is the best model to use as it has the lowest misclassification error rate of 0.267. The worst model to use is the K-Nearest Neighbor = 2 classification as its misclassification error rate is 0.397. It had the highest amount of incorrect error without being able to distinguish between positive and negative predictions.
9. The Multiple Linear Regression is the best model to use for reliability improvement based on the results as it analyzes the individual variables so that statistical analysis can better identify possible areas for improvement as opposed to the other models. For instance, as indicated by P-Value being less than the alpha value, we can conclude that temperature, stress, and material are all statistically significant factors. All three of these factors affect the number of failures within the real-life problem and model.

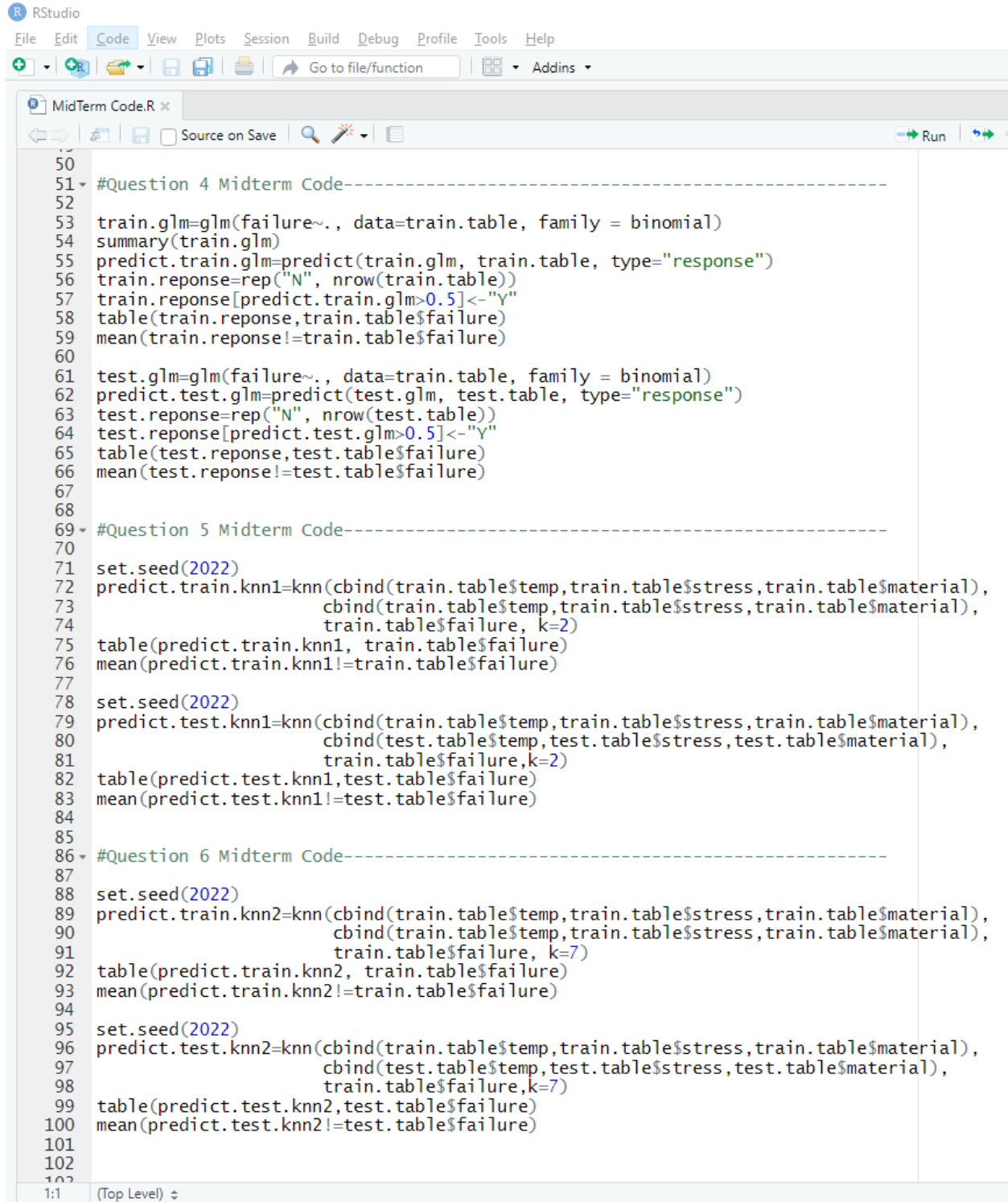
Appendix

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins

MidTerm Code.R x
Source on Save Run

1 data.train2022=read.table("C:/Users/af44m/OneDrive/Desktop/data.train2022.txt")
2 data.test2022=read.table("C:/Users/af44m/OneDrive/Desktop/data.test2022.txt")
3
4 library(MASS)
5 library(class)
6
7 train.table=data.train2022
8 train.table$failure=as.factor(train.table$failure)
9 test.table=data.test2022
10 test.table$failure=as.factor(test.table$failure)
11
12
13
14 #Question 1 Midterm Code-----
15
16 boxplot(temp ~ failure, data = train.table, cex.lab=1.5, cex.axis=1.5, cex.main=1.5,
17         cex.sub=1.5)
18
19
20
21 #Question 2 Midterm Code-----
22
23 train.lda=lda(failure ~ ., data=train.table)
24 train.lda
25 predict.train=predict(train.lda, train.table, type="response")
26 table(predict.train$class, train.table$failure)
27 mean(predict.train$class!=train.table$failure)
28
29 test.lda=lda(failure ~ ., data=train.table)
30 predict.test=predict(test.lda, test.table, type="response")
31 table(predict.test$class, test.table$failure)
32 mean(predict.test$class!=test.table$failure)
33
34
35
36 #Question 3 Midterm Code-----
37
38 train.qda=qda(failure ~ ., data=train.table)
39 train.qda
40 predict.train.qda=predict(train.qda, train.table, type="response")
41 table(predict.train.qda$class, train.table$failure)
42 mean(predict.train.qda$class!=train.table$failure)
43
44 test.qda=qda(failure ~ ., data=train.table)
45 predict.test.qda=predict(test.qda, test.table, type="response")
46 table(predict.test.qda$class, test.table$failure)
47 mean(predict.test.qda$class!=test.table$failure)
48
49
50
51 #Question 4 Midterm Code-----
52
53 train.glm=glm(failure~., data=train.table, family = binomial)
54 summary(train.glm)
1:1 (Top Level) ↕

Console
```



```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins

MidTerm Code.R x
Source on Save Run

82 table(predict.test.knn1, test.table$failure)
83 mean(predict.test.knn1!=test.table$failure)
84
85
86 #Question 6 Midterm Code-----
87
88 set.seed(2022)
89 predict.train.knn=knn(cbind(train.table$temp,train.table$stress,train.table$material),
90                       cbind(train.table$temp,train.table$stress,train.table$material),
91                       train.table$failure, k=7)
92 table(predict.train.knn2, train.table$failure)
93 mean(predict.train.knn2!=train.table$failure)
94
95 set.seed(2022)
96 predict.test.knn2=knn(cbind(train.table$temp,train.table$stress,train.table$material),
97                      cbind(test.table$temp,test.table$stress,test.table$material),
98                      train.table$failure,k=7)
99 table(predict.test.knn2,test.table$failure)
100 mean(predict.test.knn2!=test.table$failure)
101
102
103
104 #Question 7 Midterm Code-----
105
106 train.MLR = train.table
107 test.MLR = test.table
108
109 train.MLR$failure=rep("0",200)
110 train.MLR$failure[train.table$failure=="Y"]="1"
111 test.MLR$failure=rep("0",300)
112 test.MLR$failure[test.table$failure=="Y"]="1"
113 trainlm=lm(failure~., data=train.MLR)
114 summary(trainlm)
115 predict.train.lm=predict(trainlm, newdata=train.MLR, type="response")
116 response.train.lm=rep("N",200)
117 response.train.lm[predict.train.lm>0.5]="Y"
118 table(response.train.lm,train.table$failure)
119 mean(response.train.lm!=train.table$failure)
120
121
122 predict.test.lm=predict(trainlm, newdata=test.MLR, type="response")
123 response.test.lm=rep("N",300)
124 response.test.lm[predict.test.lm>0.5]="Y"
125 table(response.test.lm,test.table$failure)
126 mean(response.test.lm!=test.table$failure)
127
128
129
```