```
> #Measuring the performance of Logistic Regression as a classifier
> #Using the Challenger data as an example
>
> library(faraway)
> library(caret)
Loading required package: lattice

Attaching package: ‘lattice’

The following object is masked from ‘package:faraway’:

    melanoma

Loading required package: ggplot2
Warning message:
package ‘ggplot2’ was built under R version 3.6.3
> library(pROC)
Type 'citation("pROC")' for a citation.

Attaching package: ‘pROC’

The following objects are masked from ‘package:stats’:

    cov, smooth, var

Warning message:
package ‘pROC’ was built under R version 3.6.3
> library(epiDisplay)
Loading required package: foreign
Loading required package: survival

Attaching package: ‘survival’

The following object is masked from ‘package:caret’:

    cluster

The following objects are masked from ‘package:faraway’:

    rats, solder

Loading required package: MASS
Loading required package: nnet

Attaching package: ‘epiDisplay’

The following object is masked from ‘package:pROC’:

    ci
```

```
The following object is masked from ‘package:ggplot2’:

    alpha

The following object is masked from ‘package:lattice’:

    dotplot

Warning message:
package ‘epiDisplay’ was built under R version 3.6.3
>
>
> if (FALSE)
+ {"
+ O-Ring data analyzed using a logistic model in R
+ "}
>
> #read in the data which is in a csv file
> oring <- read.csv("C:/Users/jmard/Desktop/Computing and Graphics in Applied
Statistics2020/Lecture 08 14Feb2020/Challenger.csv",header = TRUE)
> oring
   Launch Temp TD
1       1   53  1
2       2   56  1
3       3   57  1
4       4   63  0
5       5   66  0
6       6   67  0
7       7   67  0
8       8   67  0
9       9   68  0
10     10   69  0
11     11   70  0
12     12   70  1
13     13   70  1
14     14   70  1
15     15   72  0
16     16   73  0
17     17   75  0
18     18   75  1
19     19   76  0
20     20   76  0
21     21   78  0
22     22   79  0
23     23   80  0
24     24   81  0
> nrow(oring)
[1] 24
> summary(oring)
     Launch           Temp             TD
```

```
 Min.   : 1.00   Min.   :53.00   Min.   :0.0000
 1st Qu.: 6.75   1st Qu.:67.00   1st Qu.:0.0000
 Median :12.50   Median :70.00   Median :0.0000
 Mean   :12.50   Mean   :69.92   Mean   :0.2917
 3rd Qu.:18.25   3rd Qu.:75.25   3rd Qu.:1.0000
 Max.   :24.00   Max.   :81.00   Max.   :1.0000
>
> logistic <- glm(TD ~ Temp,data=oring,family=binomial(link='logit'))
> summary(logistic)

Call:
glm(formula = TD ~ Temp, family = binomial(link = "logit"), data = oring)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2125  -0.8253  -0.4706   0.5907   2.0512

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 10.87535    5.70291   1.907   0.0565 .
Temp        -0.17132    0.08344  -2.053   0.0400 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 28.975  on 23  degrees of freedom
Residual deviance: 23.030  on 22  degrees of freedom
AIC: 27.03

Number of Fisher Scoring iterations: 4

>
> windows(7,7)
> #generate ROC curve
> ROCresult <- roc(oring$TD ~ logistic$fitted)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> plot(ROCresult, legacy.axes = TRUE)
> names(ROCresult)
 [1] "percent"           "sensitivities"     "specificities"
 [4] "thresholds"        "direction"         "cases"
 [7] "controls"          "fun.sesp"          "auc"
[10] "call"              "original.predictor" "original.response"
[13] "predictor"         "response"          "levels"
> ROCresult$auc
Area under the curve: 0.7227
>
> logistic.display(logistic)
```

```
Logistic regression predicting TD

                  OR(95%CI)         P(Wald's test) P(LR-test)
Temp (cont. var.) 0.84 (0.72,0.99)  0.04           0.015

Log-likelihood = -11.5152
No. of observations = 24
AIC value = 27.0305

>
> #now save the graph in a pdf file
>  pdf(file="C:/users/jmard/Desktop/Computing and Graphics in Applied
Statistics2020/Output/Logistic_as_a_ClassifierR_Figure.pdf")
>
> plot(TD~Temp,data=oring,xlab="Temperature",ylab="Thermal Distress")
> lines(oring$Temp,logistic$fitted, type="l", col="red")
> title(main="O-Ring Data with Fitted Logistic Regression Line")
>
> #Assessing the predictive ability of the model
> #would like to see how the model is doing as a classifier
> #Our decision boundary will be 0.5. If predicted probability of P(TD|Temp) >
0.5 then predicted.TD = 1 otherwise predicted.TD=0
> #Note that for some applications, thresholds different than 0.5 could be a
better option
>
> #This analysis is provided for instructional purposes only
> #we should be using test data and should be performing Cross Validation
> #we are using the training data set so overfitting is a concern
>
> predicted.TD <- predict(logistic,data=oring,type='response')  #using the
type='response' option generates P(TD|at each level of Temp)
> predicted.TD <- ifelse(predicted.TD > 0.5,1,0)  #predicted.TD is 1 if predicted
P(TD|Temp) > 0.50, 0 otherwise
>
> table(predicted.TD, oring$TD)

predicted.TD  0  1
           0 16  4
           1  1  3
>
> misClasificError <- mean(predicted.TD != oring$TD)  #  != is the symbol for not
equal
> misClasificError
[1] 0.2083333
>
> #check misClassificationError rate
> 5/24
[1] 0.2083333
>
> print(paste('Accuracy',1-misClasificError))
```

```
[1] "Accuracy 0.791666666666667"
>
> oring <- cbind(oring,predicted.TD)
> data.frame(oring)
   Launch Temp TD predicted.TD
1       1   53  1            1
2       2   56  1            1
3       3   57  1            1
4       4   63  0            1
5       5   66  0            0
6       6   67  0            0
7       7   67  0            0
8       8   67  0            0
9       9   68  0            0
10     10   69  0            0
11     11   70  0            0
12     12   70  1            0
13     13   70  1            0
14     14   70  1            0
15     15   72  0            0
16     16   73  0            0
17     17   75  0            0
18     18   75  1            0
19     19   76  0            0
20     20   76  0            0
21     21   78  0            0
22     22   79  0            0
23     23   80  0            0
24     24   81  0            0
>
> predicted.TD <- as.factor(predicted.TD)  #need to be sure this variable is a
factor
> oring$TD <- as.factor(oring$TD)  #need to be sure this variable is a factor
>
> confusionMatrix(oring$TD,predicted.TD)  #found in the caret library
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 16  1
         1  4  3

               Accuracy : 0.7917
                 95% CI : (0.5785, 0.9287)
    No Information Rate : 0.8333
    P-Value [Acc > NIR] : 0.8005

                  Kappa : 0.4231

 Mcnemar's Test P-Value : 0.3711
```

Modeling 0 as an event

=19/24

measure of agreement

test of agreement - paired observations

```
            Sensitivity : 0.8000
            Specificity : 0.7500
         Pos Pred Value : 0.9412
         Neg Pred Value : 0.4286
             Prevalence : 0.8333
         Detection Rate : 0.6667
   Detection Prevalence : 0.7083
      Balanced Accuracy : 0.7750

       'Positive' Class : 0

>
> dev.off()
null device
          1
>
```

=16/20
= 3/4
=16/17
=3/7