Challenger Notebook

The dataset consists of 144 observations of 5 variables consisting of:
Flight: name of flight; Date (date of flight);
Field (1 if an Oring fails and 0 otherwise);
Temp (Temperature in degrees Fahrenheit);
Pres (Leak check pressure in psi).
Each flight had 6 orings.

```
orings <- read.csv("Orings.csv")
str(orings)
```

```
## 'data.frame':    144 obs. of  5 variables:
##  $ Flight: chr  "1" "1" "1" "1" ...
##  $ Date  : chr  "4/12/1981" "4/12/1981" "4/12/1981" "4/12/1981" ...
##  $ Field : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Temp  : int  66 66 66 66 66 66 70 70 70 70 ...
##  $ Pres  : int  50 50 50 50 50 50 50 50 50 50 ...
```

```
summary(orings)
```

```
##     Flight              Date               Field              Temp      
##  Length:144         Length:144         Min.   :0.00000   Min.   :31.00  
##  Class :character   Class :character   1st Qu.:0.00000   1st Qu.:66.75  
##  Mode  :character   Mode  :character   Median :0.00000   Median :70.00  
##                                        Mean   :0.07246   Mean   :67.96  
##                                        3rd Qu.:0.00000   3rd Qu.:75.00  
##                                        Max.   :1.00000   Max.   :81.00  
##                                        NA's   :6                        
##       Pres      
##  Min.   : 50.0  
##  1st Qu.: 87.5  
##  Median :200.0  
##  Mean   :154.2  
##  3rd Qu.:200.0  
##  Max.   :200.0  
##
```

Let us check the number of orings that have failed out of 6 in each of the flights launched.

```
tapply(orings$Field, orings$Flight, sum)
```

```
##    1    2    3 41-B 41-C 41-D 41-G    5 51-A 51-B 51-C 51-D 51-F 51-G 51-I 51-J
##    0    1    0    1    1    1    0    0    0    0    3    0    0    0    0    0
##    6 61-A 61-B 61-C 61-I    7    8    9
##    0    2    0    1   NA    0    0    0
```

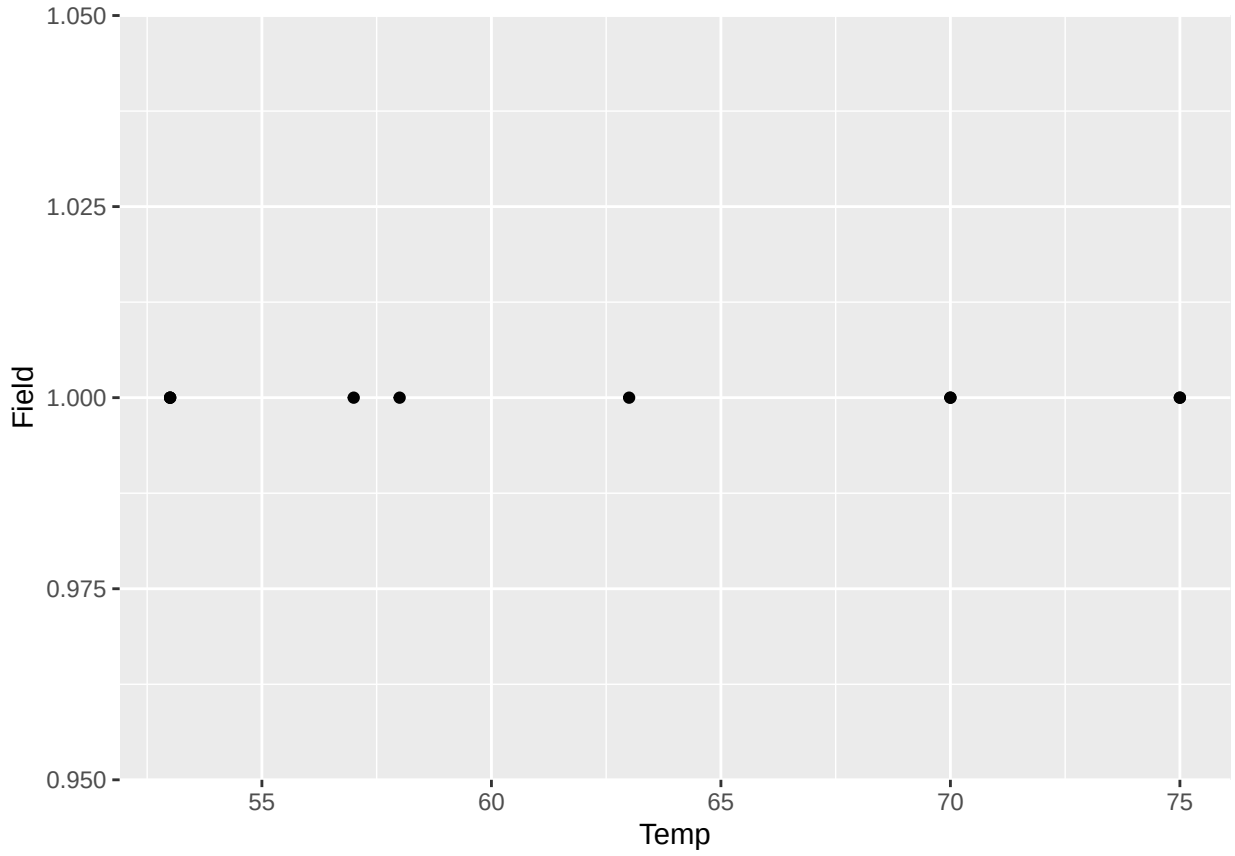
```r
table(tapply(orings$Field, orings$Flight, sum))
```

```
##
##  0  1  2  3
## 16  5  1  1
```

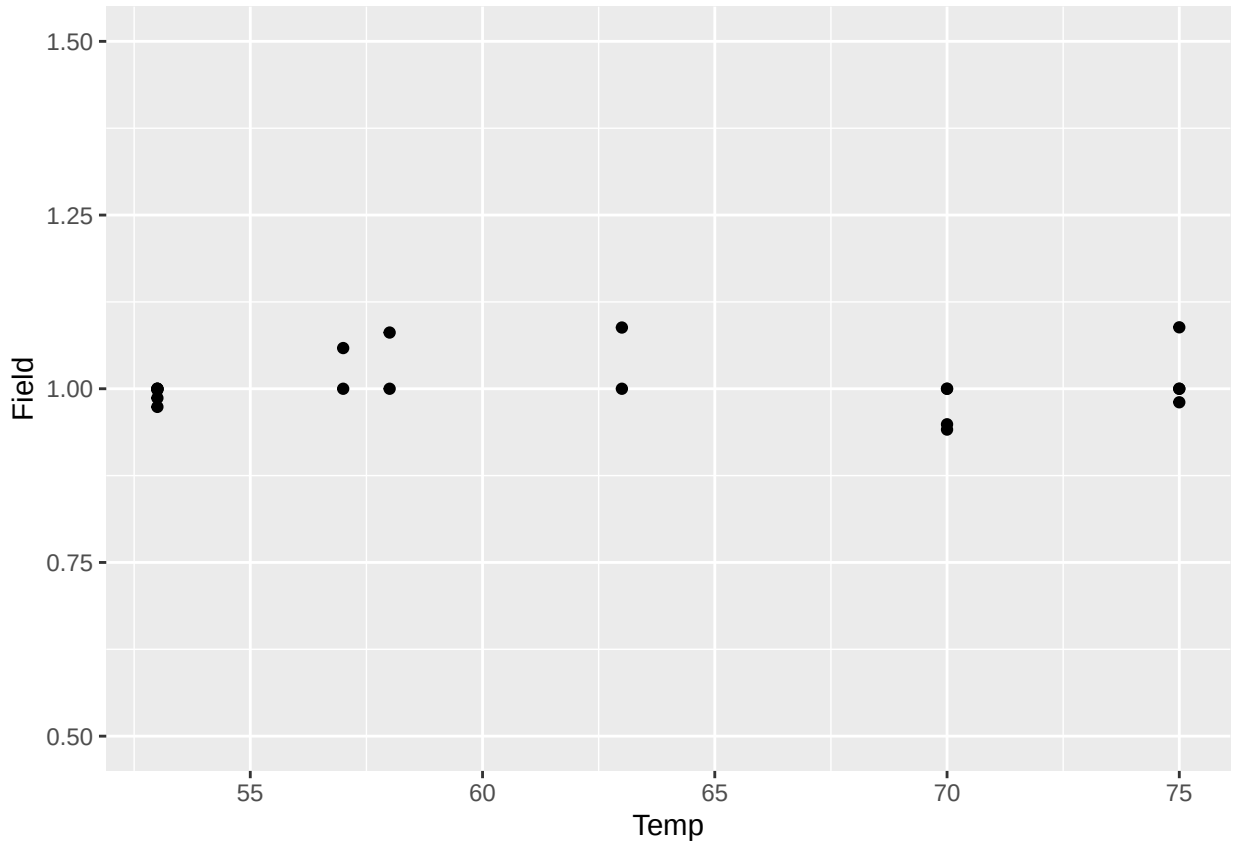
```r
# row: field
# column: flight
# in 16 of the flights, none of the orings failed
# in 5 of the flights, 1 oring failed
```

We plot the failures and temperatures.
We use jitter plot to randomly perturb points by a small amount to see the points that lie on top of each other better.

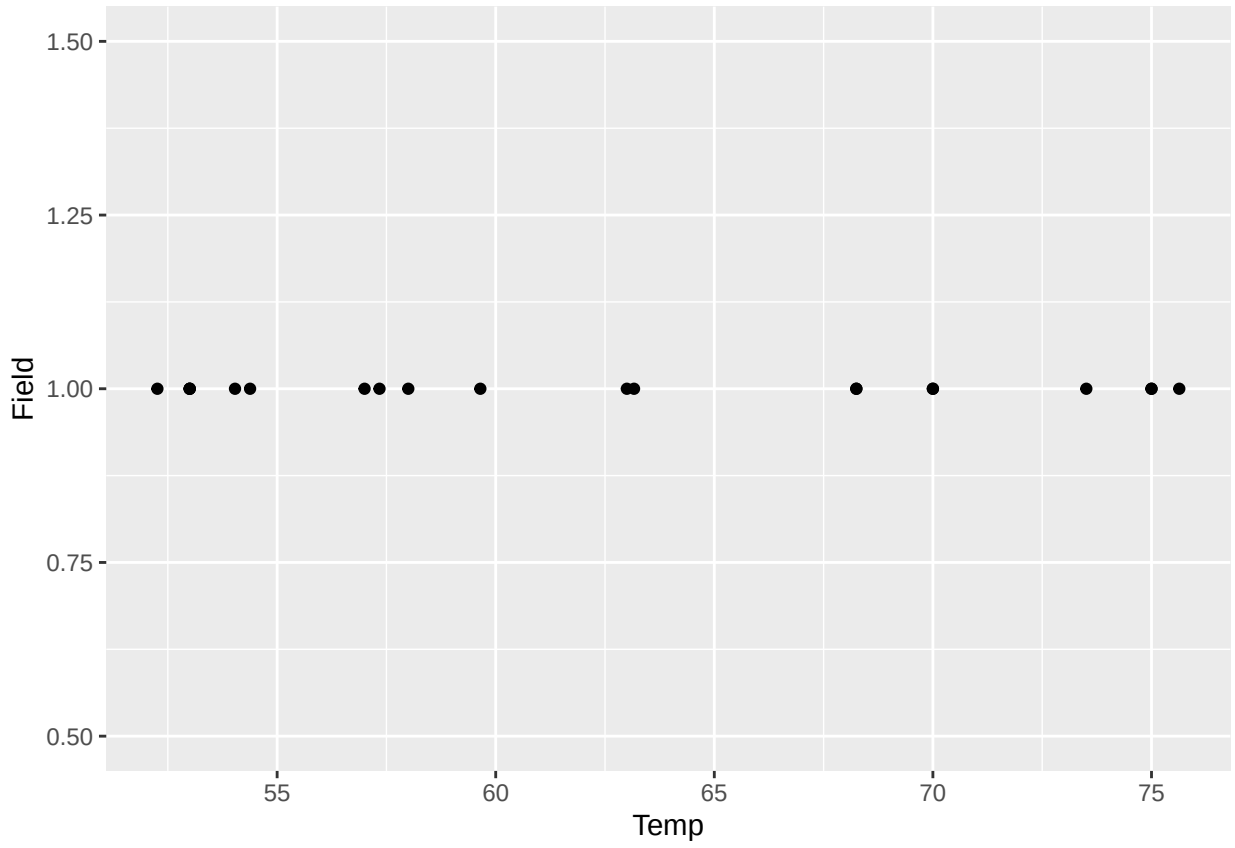
```r
library(ggplot2)
# plot graphs for orings that has failed
ggplot(orings[orings$Field > 0, ], aes(x = Temp, y = Field)) +  geom_point(na.rm = TRUE)
```



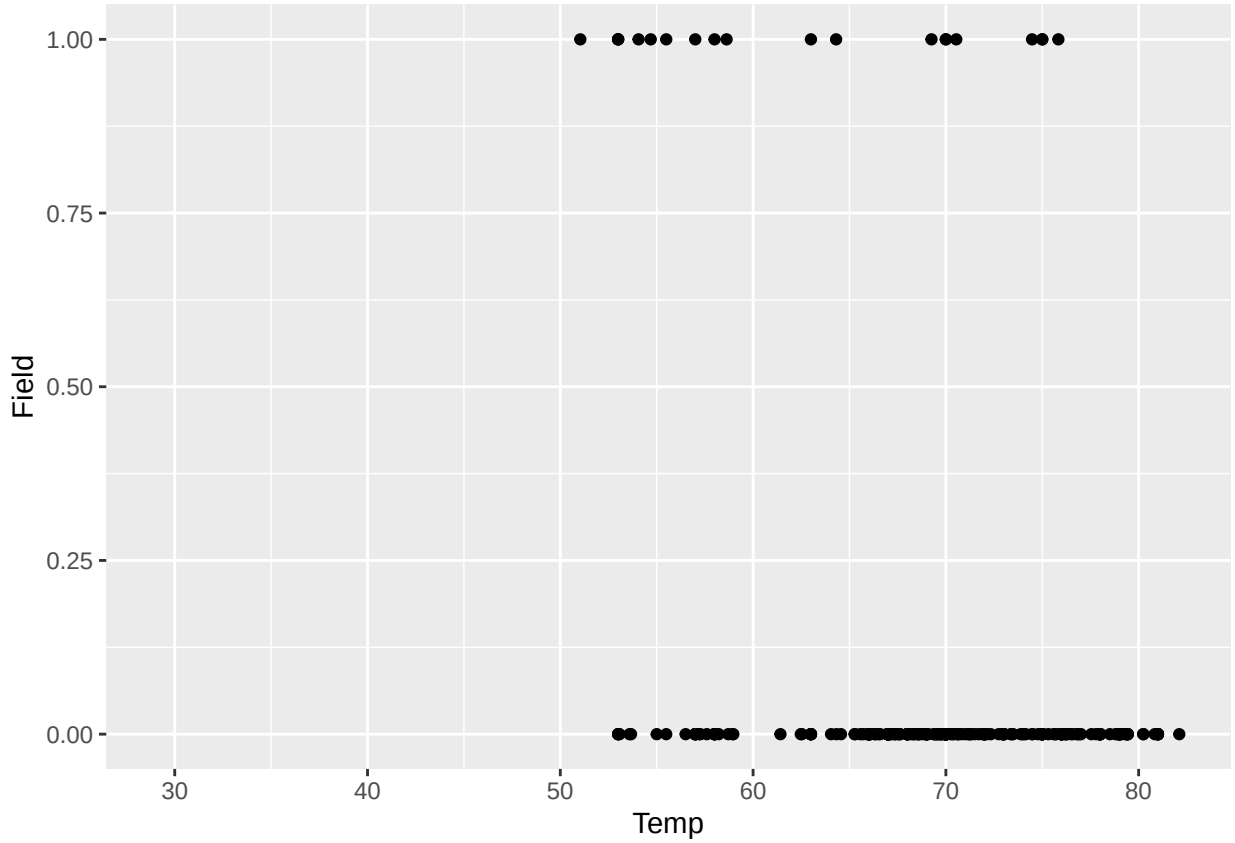

```r
ggplot(data = orings[orings$Field > 0, ], aes(x = Temp, y = Field)) + geom_point(na.rm = TRUE) + geom_j
```




```
ggplot(data = orings[orings$Field > 0, ], aes(x = Temp, y = Field)) + geom_point(na.rm = TRUE) + geom_j
```




```
# plot graphs of all orings
ggplot(data = orings, aes(x = Temp, y = Field)) + geom_point(na.rm = TRUE) + geom_jitter(na.rm = TRUE, w
```



The plots of temperature with failures only and the plot of temperatures with both failures and non-failures provides different insights. In the former, there are failures across the range with some more at the extremes. In the second case, it is much clearer that there are lesser failures at higher temperatures. It is believed that analysis of plots such as the first one led the managers to conclude that there was not a significant influence of low temperatures.

Fitting a linear regression model

- y: probability of failure/success, 0 <= probability <= 1

```
model1 <- lm(formula = Field ~ Temp + Pres, data = orings)
summary(model1)
```

```
##
## Call:
## lm(formula = Field ~ Temp + Pres, data = orings)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27680 -0.09548 -0.03457 -0.00874  0.99126
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.7264305  0.2181384   3.330  0.00112 **
## Temp        -0.0106657  0.0030704  -3.474  0.00069 ***
## Pres         0.0005783  0.0003176   1.821  0.07088 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2487 on 135 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.09941,    Adjusted R-squared:  0.08607
## F-statistic: 7.451 on 2 and 135 DF,  p-value: 0.0008524
```

```
# multiple R squared value is quite high
# linear regression perhaps does not explain the data so well
```

Now we use only Temperature

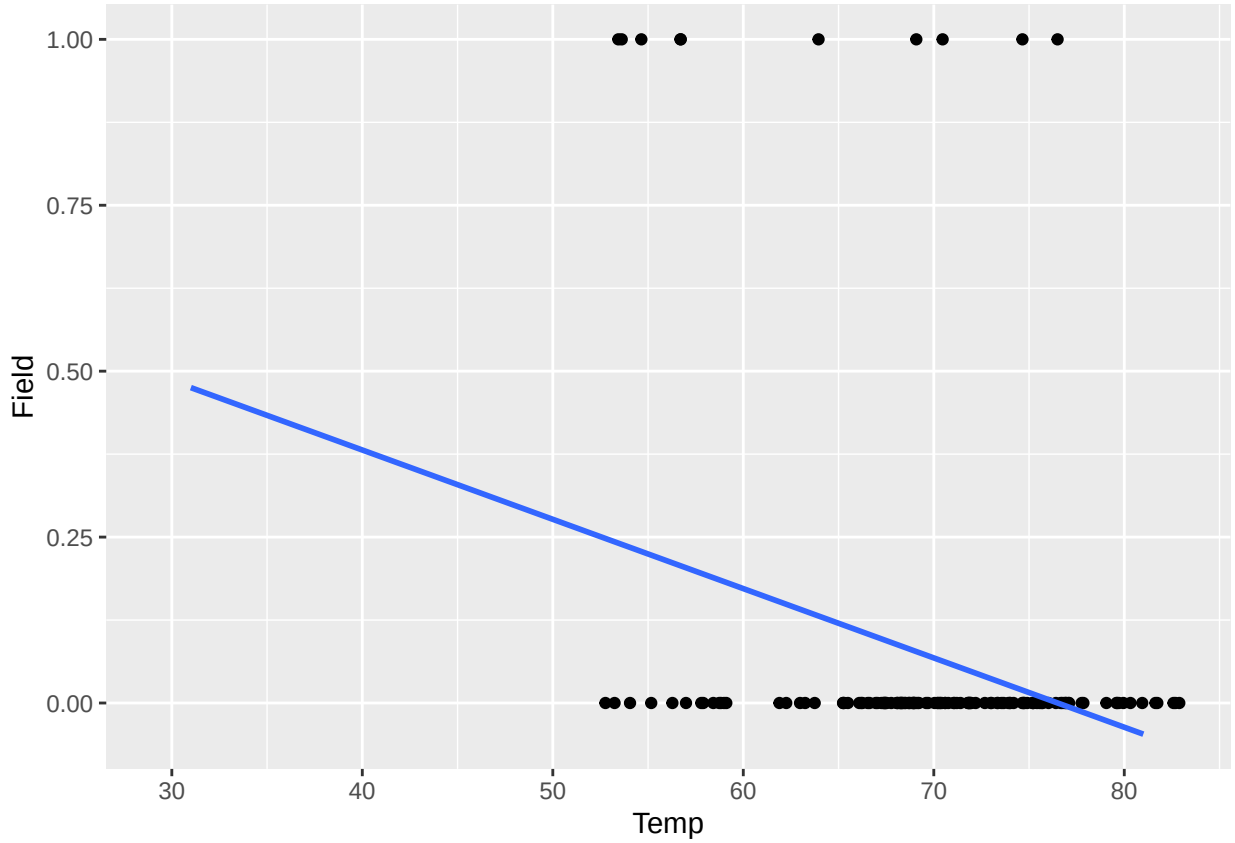
```
model2 <- lm(formula = Field ~ Temp, data = orings)
summary(model2)
```

```
##
## Call:
## lm(formula = Field ~ Temp, data = orings)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24546 -0.09925 -0.06792 -0.00526  0.98429
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.798942   0.216289   3.694 0.000319 ***
## Temp        -0.010443   0.003094  -3.375 0.000961 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2509 on 136 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.0773, Adjusted R-squared:  0.07051
## F-statistic: 11.39 on 1 and 136 DF,  p-value: 0.0009612
```

```
# r squared value dropped drastically, so this is not a good model as well

ggplot(orings, aes(Temp, Field)) + geom_jitter(na.rm = T, height = 0, width = 2) + geom_smooth(method =
```

```
## `geom_smooth()` using formula 'y ~ x'
```



The result indicates that the linear fit is not particularly convincing with a small value of R-squared, though it does identify that temperature has a significant effect and it is a negative impact.

Fitting a logistic regression model: glm() is a generalized linear model that can be used to fit a logistic regression model by choosing family=binomial.

```
model3 <- glm(formula = Field ~ Temp + Pres, data = orings, family = "binomial")
summary(model3)
```

```
##
## Call:
## glm(formula = Field ~ Temp + Pres, family = "binomial", data = orings)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9243  -0.3680  -0.2432  -0.2059   2.8217
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.958071   3.497847   1.132  0.25781
## Temp        -0.119355   0.044945  -2.656  0.00792 **
## Pres         0.008692   0.007689   1.130  0.25829
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 71.751  on 137  degrees of freedom
## Residual deviance: 60.470  on 135  degrees of freedom
##   (6 observations deleted due to missingness)
## AIC: 66.47
##
## Number of Fisher Scoring iterations: 6
```

Now we use only temperature

```
model4 <- glm(formula = Field ~ Temp, data = orings, family = binomial)
summary(model4)
```

```
##
## Call:
## glm(formula = Field ~ Temp, family = binomial, data = orings)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9155  -0.3770  -0.3075  -0.2036   2.7387
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.75183    2.97991   2.266  0.02346 *
## Temp        -0.13971    0.04647  -3.007  0.00264 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 71.751  on 137  degrees of freedom
## Residual deviance: 62.083  on 136  degrees of freedom
##   (6 observations deleted due to missingness)
## AIC: 66.083
##
## Number of Fisher Scoring iterations: 6
```

```
model3$coefficients
```

```
##  (Intercept)         Temp         Pres
##  3.958070598 -0.119355166  0.008691799
```

```
model3$aic
```

```
## [1] 66.4705
```

Model 3 describes
$Prob(Fail = 1) = exp(3.95 - 0.119Temp + 0.008Pres)/(1 + exp(3.95 - 0.119Temp + 0.008Pres))$.
Model 3 result indicates that Temp is significant at the 5% level.

```
model4$coefficients
```

```
## (Intercept)        Temp
##   6.7518341  -0.1397096
```

```
model4$aic
```

```
## [1] 66.08252
```

Model 4 has a fit given by
$P(Fail = 1) = exp(6.75 - 0.1397Temp)/(1 + exp(6.75 - 0.1397Temp))$.
In terms of AIC, Model 4 is preferred to Model 3, because Model 4 has lower AIC value. We drop the pressure variable in this example. Hence, we use model 4 to predict. In this case, failure is only dependent on temperature.

Predictions:

```
# predict value on log(odds) scale
# ie. the exponential function in the numerator (beta.x)

predict(model4, newdata = tail(orings, 1))
```

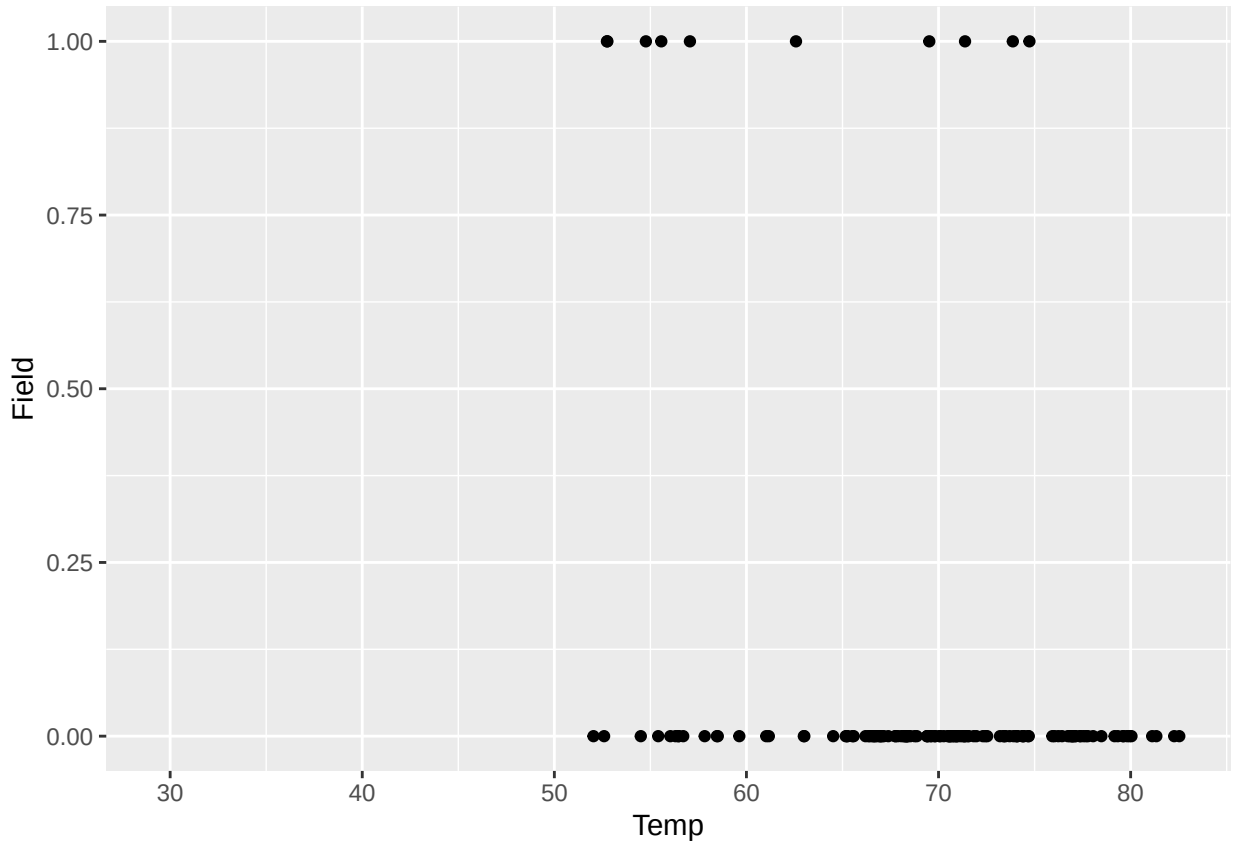
```
##      144
## 2.420837
```

```
# predict value on odds scale
# ie. probability of y = 1
predict(model4, newdata = tail(orings, 1), type = "response")
```

```
##       144
## 0.9184025
```

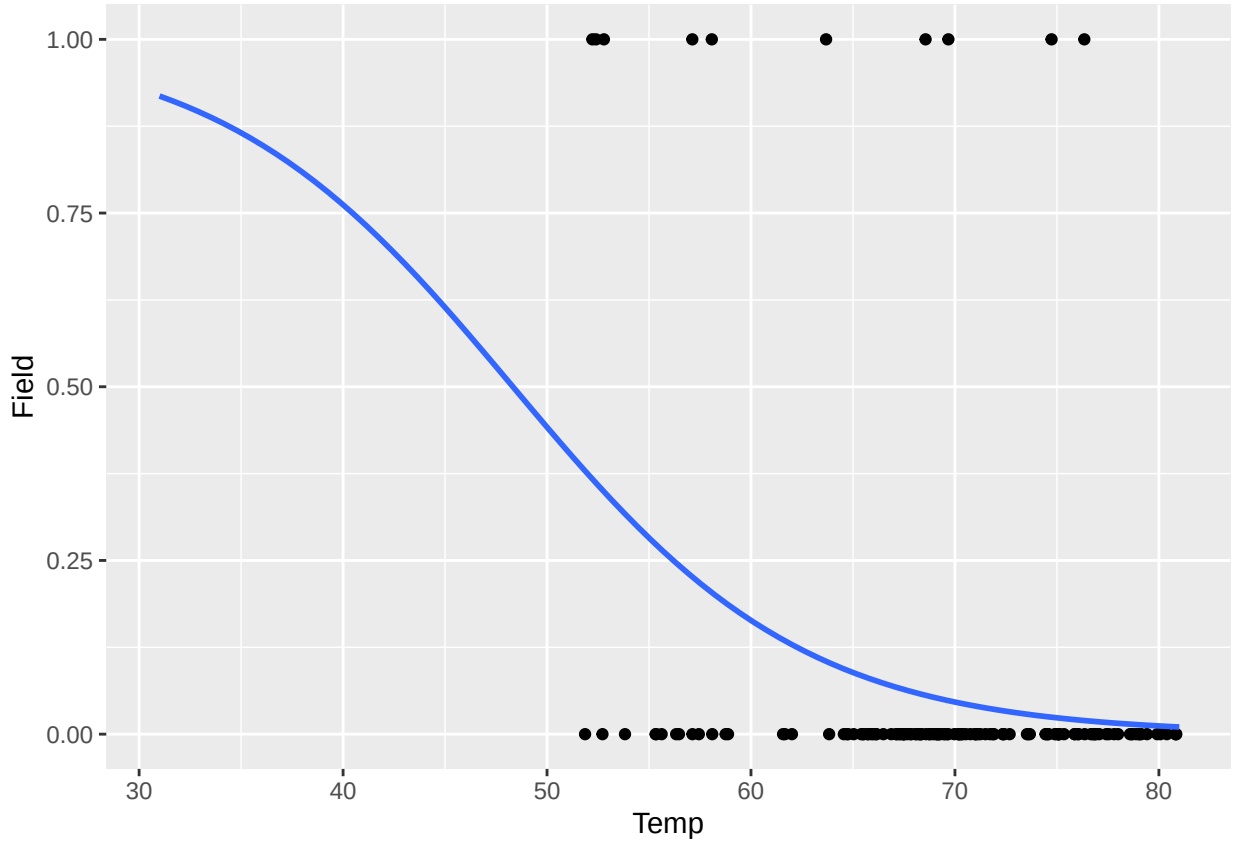
Plots

```
ggplot(data = orings, aes(x = Temp, y = Field)) + geom_jitter(na.rm = T, height = 0, width = 2)
```




```
# Logistic Regression Model Fit
ggplot(data = orings, aes(x = Temp, y = Field)) + geom_jitter(na.rm = T, height = 0, width = 2) + geom_s
```

```
## `geom_smooth()` using formula 'y ~ x'
```



The predicted probability of failure under the model (for a temperature of 31 degree Fahrenheit) is 0.918.

From the logistic regression graph, we can infer that the probability of oring failure is higher at low temperatures relative to high temperatures. - higher probabilities at low temperatures

Developing a predictive rule (classifier) and tabulating confusion matrices.
Here we are still evaluatingthis with the training data as we have a small dataset. Typically we will use a test data to check on the results.

```
# type = "response": probability values are returned
Pred <- predict(model4, newdata = orings, type = "response")

# create a table of confusion matrix
# row: prediction label 0 or 1
# column: true label 0 or 1
table(Pred[1:138] > 0.5, orings$Field[1:138])
```

```
##
##           0   1
##   FALSE 128  10
```

```
# Pred > 0.5 means that failure will happen if the threshold is > 0.5, which will predict value of 1
# values show that there will never be a failure if the threshold is set at 0.5
# there were 10 failures that happened but it was predicted as no failure will happen
# hence, model is not appropriate!
# 128 cases were predicted correctly, 10 cases were predicted incorrectly.
```

```
table(Pred[1:138] > 0.3, orings$Field[1:138])
```

```
##
##           0   1
##   FALSE 125   7
##   TRUE    3   3
```

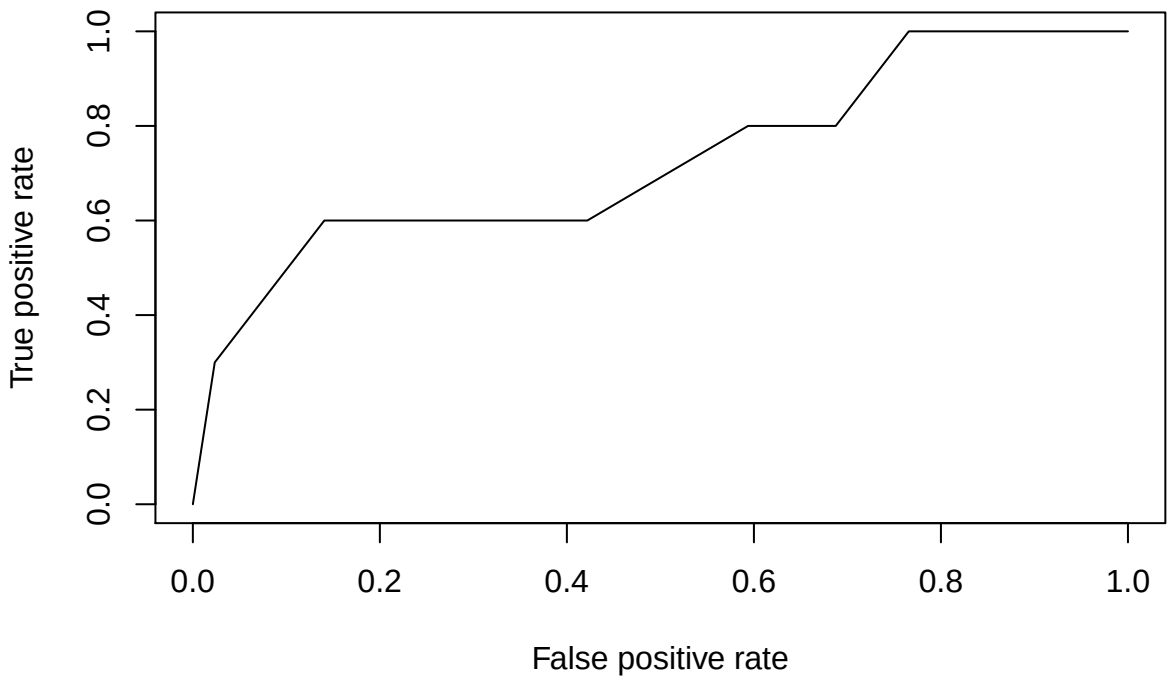
```
# threshold of > 0.3 to be a failure
# now the results show that there are lesser wrong predictions on actual failures (ie. predicted as not
```

The ROCR package is useful for visualizing the performance of classifier. The prediction function transforms the data to a standardized format and the performance function is used to do all kinds of prediction evaluations.

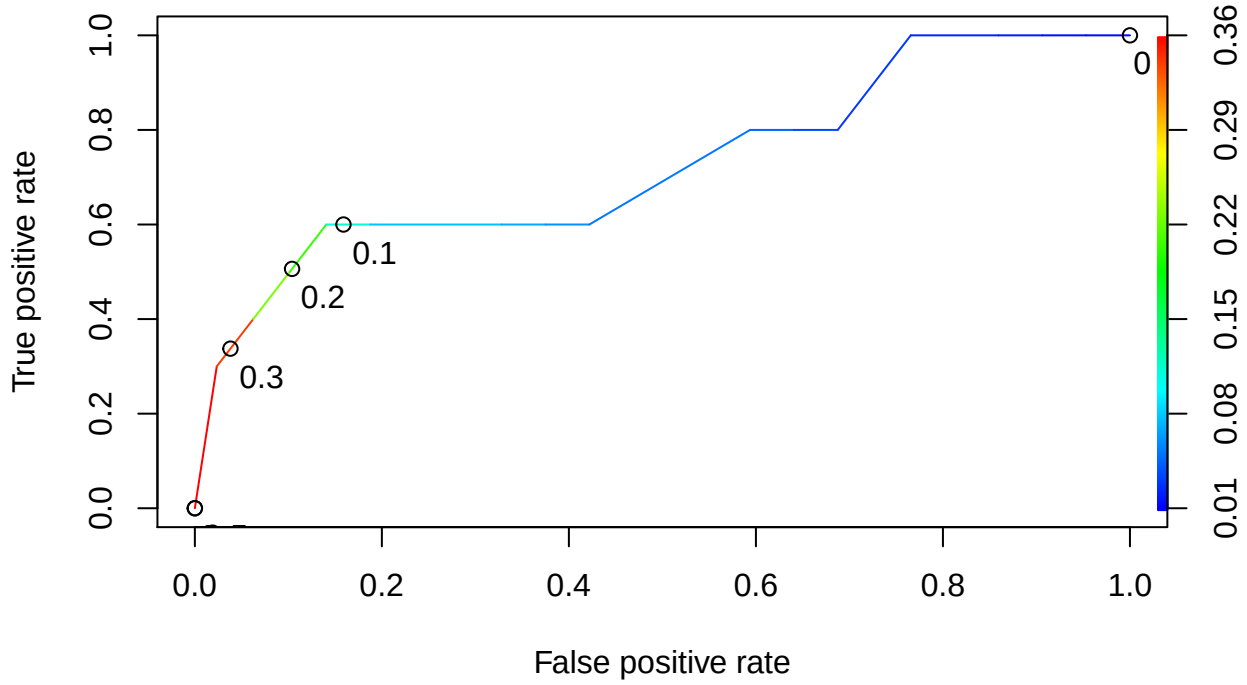
```
library(ROCR)
# creates a prediction instance in the required ROCR standardised format
# input values are prediction labels and true labels
ROCRpred <- prediction(predictions = Pred[1:138], labels = orings$Field[1:138])

?performance
# creates the ROC curve
ROCRperf <- performance(prediction.obj = ROCRpred, x.measure = "fpr", measure = "tpr")

plot(ROCRperf)
```



```
# indicate the t values across the ROC curve
# cutoffs refer to threshold t values
plot(ROCRperf, colorize = T, print.cutoffs.at = c(0, 0.1, 0.2, 0.3, 0.5, 1), text.adj = c(-0.2, 1.7))
```




```
# calculate AUC
as.numeric(performance(prediction.obj = ROCRpred, measure = "auc")@y.values)
```

```
## [1] 0.725
```

The AUC for this example is 0.725