

Risk Analysis of the Space Shuttle

Access to the libraries

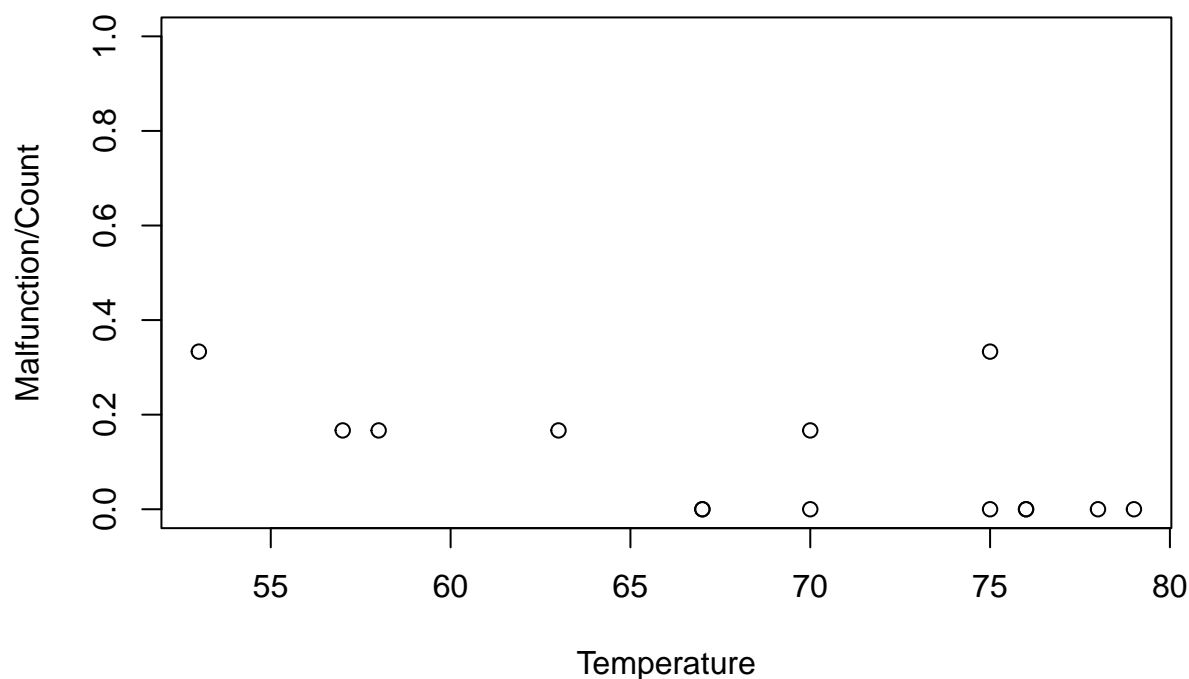
Open the file

```
data= read.csv("C:/Users/sofic/OneDrive/Escritorio/Sofi/R Projects/Challenger - SMPE/mocrr-reproducibi.  
data
```

##	Date	Count	Temperature	Pressure	Malfunction
## 1	4/12/81	6	66	50	0
## 2	11/12/81	6	70	50	1
## 3	3/22/82	6	69	50	0
## 4	11/11/82	6	68	50	0
## 5	4/04/83	6	67	50	0
## 6	6/18/82	6	72	50	0
## 7	8/30/83	6	73	100	0
## 8	11/28/83	6	70	100	0
## 9	2/03/84	6	57	200	1
## 10	4/06/84	6	63	200	1
## 11	8/30/84	6	70	200	1
## 12	10/05/84	6	78	200	0
## 13	11/08/84	6	67	200	0
## 14	1/24/85	6	53	200	2
## 15	4/12/85	6	67	200	0
## 16	4/29/85	6	75	200	0
## 17	6/17/85	6	70	200	0
## 18	7/2903/85	6	81	200	0
## 19	8/27/85	6	76	200	0
## 20	10/03/85	6	79	200	0
## 21	10/30/85	6	75	200	2
## 22	11/26/85	6	76	200	0
## 23	1/12/86	6	58	200	1

Filter data with pressure=200 and eliminate value with date: 7/2903/85 and plot it

```
data_filtered <- data[data$Pressure == 200 & data$Date != "7/2903/85",]  
plot(data=data_filtered,Malfunction/Count ~Temperature,ylim=c(0,1))
```



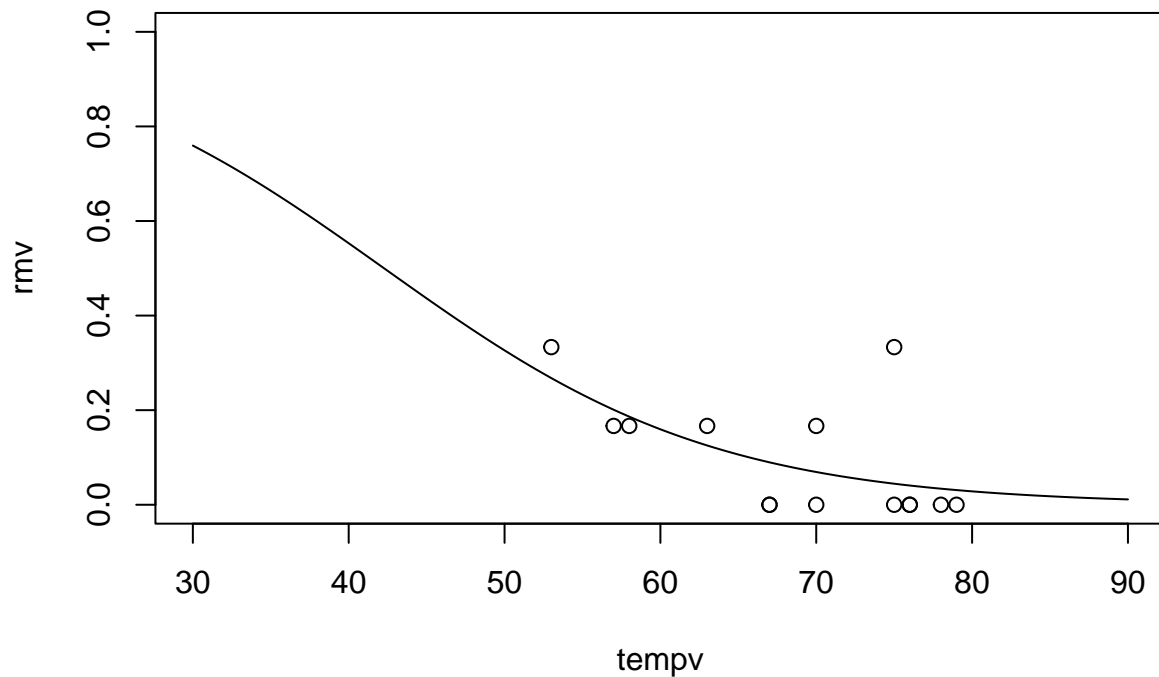
Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
logistic_reg = glm(data=data_filtered, Malfunction/Count ~ Temperature, weights=Count,
family=binomial(link='logit'))
summary(logistic_reg)
```

```
##
## Call:
## glm(formula = Malfunction/Count ~ Temperature, family = binomial(link = "logit"),
##     data = data_filtered, weights = Count)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.96165    2.97440   1.332  0.1829
## Temperature -0.09372    0.04625  -2.026  0.0427 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15.931  on 13  degrees of freedom
## Residual deviance: 11.552  on 12  degrees of freedom
## AIC: 27.289
##
## Number of Fisher Scoring iterations: 5
```

Predicting failure probability

```
tempv = seq(from=30, to=90, by = .5)
rmv <- predict(logistic_reg,list(Temperature=tempv),type="response")
plot(tempv,rmv,type="l",ylim=c(0,1))
points(data=data_filtered, Malfunction/Count ~ Temperature)
```



Confidence on the prediction

```
data_flat <- data.frame()

for(i in 1:nrow(data_filtered)){
  temperature <- data_filtered[i, "Temperature"]
  malfunction <- data_filtered[i, "Malfunction"]
  d <- data.frame(
    Temperature = temperature,
    Malfunction = rep(0, times = data_filtered[i, "Count"])
  )
  if(malfunction > 0){
    d[1:malfunction, "Malfunction"] <- 1
  }
  data_flat <- rbind(data_flat, d)
}
dim(data_flat)
```

```
## [1] 84 2
```

```
str(data_flat)
```

```
## 'data.frame': 84 obs. of 2 variables:
## $ Temperature: int 57 57 57 57 57 57 63 63 63 63 ...
## $ Malfunction: num 1 0 0 0 0 0 1 0 0 0 ...
```

Logistic regression

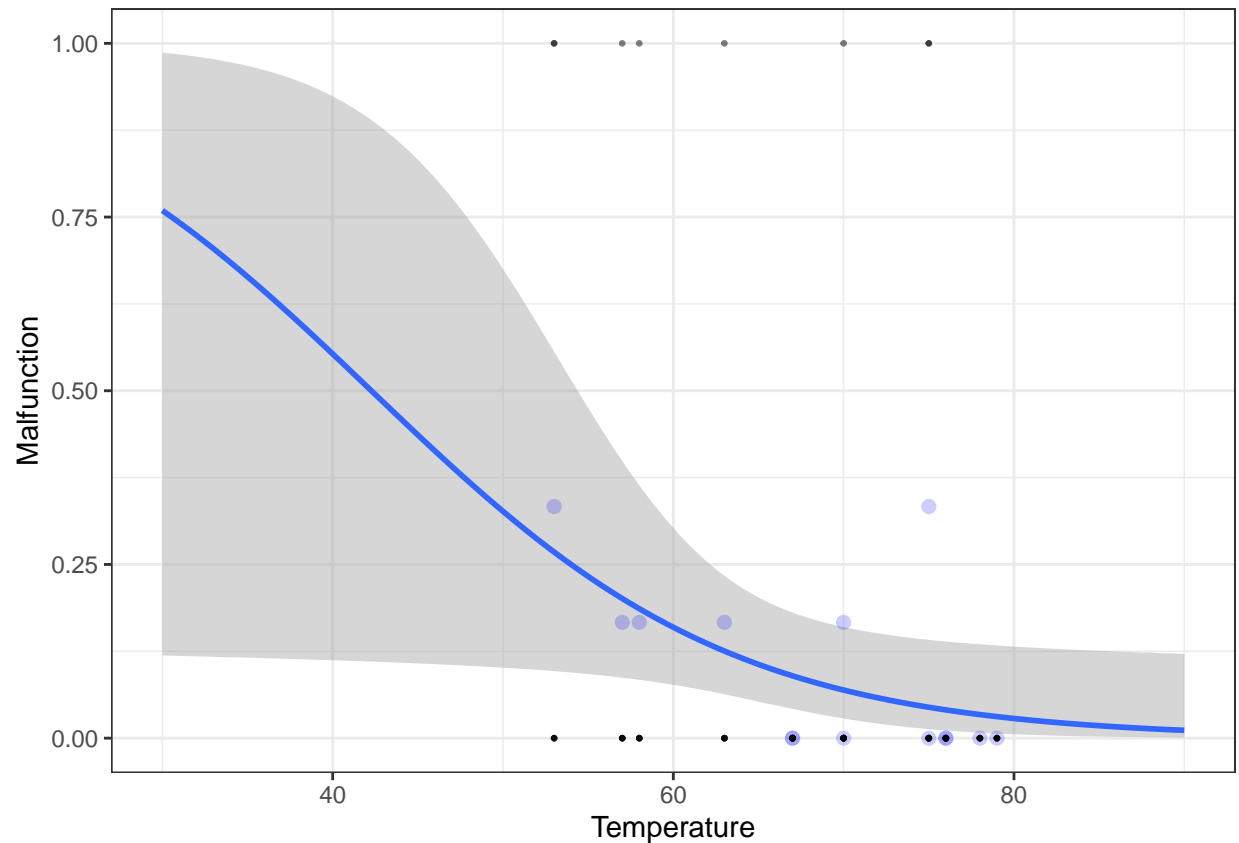
```
logistic_reg_flat=glm(data=data_flat,Malfunction~ Temperature,family=binomial(link='logit'))
summary(logistic_reg)
```

```
##
## Call:
## glm(formula = Malfunction/Count ~ Temperature, family = binomial(link = "logit"),
##      data = data_filtered, weights = Count)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.96165    2.97440   1.332  0.1829
## Temperature -0.09372    0.04625  -2.026  0.0427 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 15.931  on 13  degrees of freedom
## Residual deviance: 11.552  on 12  degrees of freedom
## AIC: 27.289
##
## Number of Fisher Scoring iterations: 5
```

Plot Confidence interval

```
ggplot(data=data_flat, aes(y=Malfunction, x=Temperature)) +
  geom_smooth(method = "glm", method.args = list(family = "binomial"), fullrange=T) +
  geom_point(data=data_filtered, aes(y=Malfunction/Count, x=Temperature),alpha=.2, size = 2, color="blue") +
  geom_point(alpha=.5, size = .5) +
  xlim(30,90) + ylim(0,1) + theme_bw()
```

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



Numbers to check logistic Regression

```
pred_link = predict(logistic_reg_flat,list(Temperature=30),type="link",se.fit = T)
pred_link
```

```
## $fit
##      1
## 1.149994
##
## $se.fit
## [1] 1.608328
##
## $residual.scale
## [1] 1
```

```
logistic_reg$family$linkinv(pred_link$fit)
```

```
##      1
## 0.7595098
```

```
critval = 1.96
logistic_reg$family$linkinv(c(pred_link$fit-critval*pred_link$se.fit,
pred_link$fit+critval*pred_link$se.fit))
```

```
##      1      1
## 0.1189587 0.9866436
```