Risk Analysis of the Space Shuttle: Pre-Challenger Prediction of Failure - copied and edited version from Arnaud Legrand

JFM

November 18, 2018

Contents

In this document I reperform some of the analysis provided in *Risk Analysis of the Space Shuttle: Pre-Challenger Prediction of Failure* by *Siddharta R. Dalal, Edward B. Fowlkes, Bruce Hoadley* published in *Journal of the American Statistical Association*, Vol. 84, No. 408 (Dec., 1989), pp. 945-957 and available at `http://www.jstor.org/stable/2290069`. I use a version of this document proposed by Arnaud Legrand in the MOOC on reproducible research 2018. I want to see if I can replicate the analysis on my computer.

On the fourth page of this article, they indicate that the maximum likelihood estimates of the logistic regression using only temperature are: =5.085 and =-0.1156 and their asymptotic standard errors are $s_{\hat{\alpha}}$=3.052 and $s_{\hat{\beta}}$=0.047. The Goodness of fit indicated for this model was G^2 =18.086 with 21 degrees of freedom. Our goal is to reproduce the computation behind these values and the Figure 4 of this article, possibly in a nicer looking way.

1 Technical information on the computer on which the analysis is run

We will be using the R language using the `ggplot2` library.

```
library(ggplot2)
sessionInfo()
```

```
R version 3.5.1 (2018-07-02)
Platform: x86_64-suse-linux-gnu (64-bit)
Running under: openSUSE Leap 15.0

Matrix products: default
BLAS: /usr/lib64/R/lib/libRblas.so
LAPACK: /usr/lib64/R/lib/libRlapack.so

locale:
 [1] LC_CTYPE=de_DE.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=de_DE.UTF-8        LC_COLLATE=de_DE.UTF-8
 [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=de_DE.UTF-8
 [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] ggplot2_3.0.0

loaded via a namespace (and not attached):
 [1] colorspace_1.3-2 scales_1.0.0     compiler_3.5.1   plyr_1.8.4
 [5] lazyeval_0.2.1   withr_2.1.2      pillar_1.3.0     gtable_0.2.0
 [9] tibble_1.4.2     crayon_1.3.4     Rcpp_0.12.18     grid_3.5.1
[13] rlang_0.2.2      munsell_0.5.0
```

Here are the available libraries

```
devtools::session_info()
```

```
Session info-------------------------------------------------------------------
 setting  value
 version  R version 3.5.1 (2018-07-02)
 system   x86_64, linux-gnu
 ui       X11
 language (EN)
 collate  de_DE.UTF-8
 tz       Europe/Berlin

Packages-----------------------------------------------------------------------
 package    * version date       source
 colorspace   1.3.2   2016-12-14 CRAN (R 3.5.1)
 crayon       1.3.4   2017-09-16 CRAN (R 3.5.1)
 devtools     1.6.1   2014-10-07 CRAN (R 3.5.1)
 ggplot2    * 3.0.0   2018-07-03 CRAN (R 3.5.1)
 gtable       0.2.0   2016-02-26 CRAN (R 3.5.1)
 lazyeval     0.2.1   2017-10-29 CRAN (R 3.5.1)
 munsell      0.5.0   2018-06-12 CRAN (R 3.5.1)
 pillar       1.3.0   2018-07-14 CRAN (R 3.5.1)
 plyr         1.8.4   2016-06-08 CRAN (R 3.5.1)
 Rcpp         0.12.18 2018-07-23 CRAN (R 3.5.1)
 rlang        0.2.2   2018-08-16 CRAN (R 3.5.1)
 rstudioapi   0.7     2017-09-07 CRAN (R 3.5.1)
 scales       1.0.0   2018-08-09 CRAN (R 3.5.1)
 tibble       1.4.2   2018-01-22 CRAN (R 3.5.1)
 withr        2.1.2   2018-03-15 CRAN (R 3.5.1)
```

2 Loading and inspecting data

Let's start by reading data:

```
data = read.csv("https://app-learninglab.inria.fr/gitlab/moocrr-session1/moocrr-reprodu
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

We know from our previous experience on this data set that filtering data is a really bad idea. We ill therefore process it as such.

Let's visually inspect how temperature affects malfunction:

```
plot(data=data, Malfunction/Count - Temperature, ylim = c(0,1))
```

Objekt 'Malfunction' nicht gefunden

3 Logistic regression

Let's assume O-rings independently fail with the same probability which solely depends on temperature. A logistic regression should allow us to estimate the influence of temperature.

```
summary(logistic_reg)
```

```
Fehler in stats::model.frame(formula = Malfunction/Count - Temperature,  :
  Objekt 'Malfunction' nicht gefunden


Fehler in summary(logistic_reg) : Objekt 'logistic_reg' nicht gefunden
```

The maximum likelyhood estimatator of the intercept and of Temperature are thus $=?$ and $=?$ and their standard errors are $s_{\hat{\alpha}}=?$ and $s_{\hat{\beta}}=?$. The Residual deviance corresponds to the Goodness of fit $G^2 = ?$ with ? degrees of freedom. Since some function does not operate as in the example given by Arnaud Legrand: **I have therefore <u>not yet</u> managed to replicate the results of the Dalal *et.al.* article**.

4 Predicting failure probability

The temperature when launching the shuttle was 31°F. Let's try to estimate the failure probability for such temperature using our model:

```
# shuttle=shuttle[shuttle$r!=0,]
tempv = seq(from=30, to=90, by = .5)
rmv <- predict(logistic_reg,list(Temperature=tempv),type="response")
plot(tempv,rmv,type="1",ylim=c(0,1))
points(data=data, Malfunction/Count - Temperature)
```

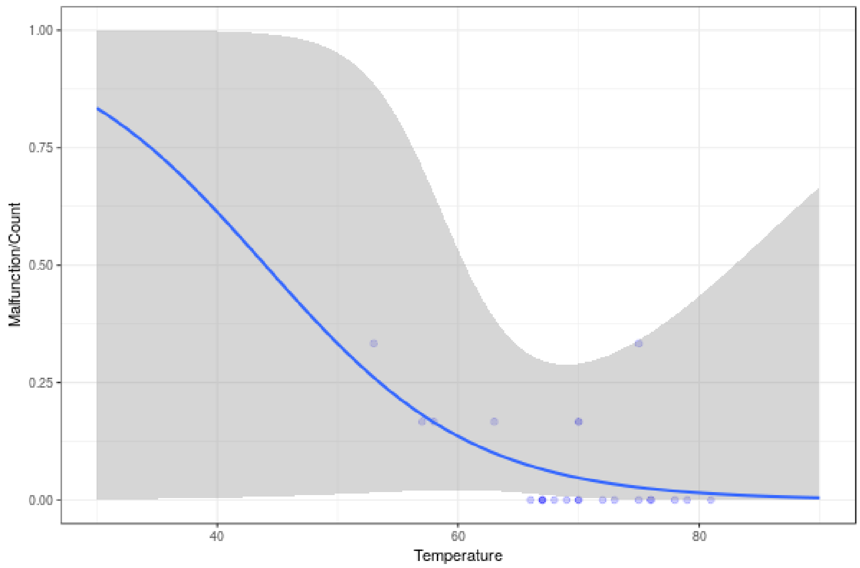
Objekt 'logistic_reg' nicht gefunden

For the error mentioned above I have not been able to plot this. This figure is <u>not yet</u> very similar to the Figure 4 of Dalal et. al. **I have <u>not yet</u> managed to replicate the Figure 4 of the Dalal *et. al.* article.**

5 Confidence on the prediction

Let's try to plot confidence intervals with `ggplot2`.

```
ggplot(data, aes(y=Malfunction/Count, x=Temperature)) + geom_point(alpha=.2, size = 2,
```



Apparently I don't have the warning Arnaud Legrand mentions from `ggplot2` indicating *"non-integer #successes in a binomial glm!"*. This seems fishy for him, but not for me. But: yes, this confidence region seems huge… It seems strange to me that the uncertainty grows so large for higher temperatures. And compared to my previous call to `glm`, I haven't indicated the weight which accounts for the fact that each ration Malfunction/Count corresponds to `Count` observations (if someone knows how to do this …) There must be something wrong.

So let's provide the "raw" data to `ggplot2`.

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

```
[1] 138   2
```

```
str(data_flat)
```

```
'data.frame': 138 obs. of  2 variables:
 $ Temperature: int  66 66 66 66 66 66 70 70 70 70 ...
 $ Malfunction: num  0 0 0 0 0 0 1 0 0 0 ...
```

Let's check whether I obtain the same regression or not:

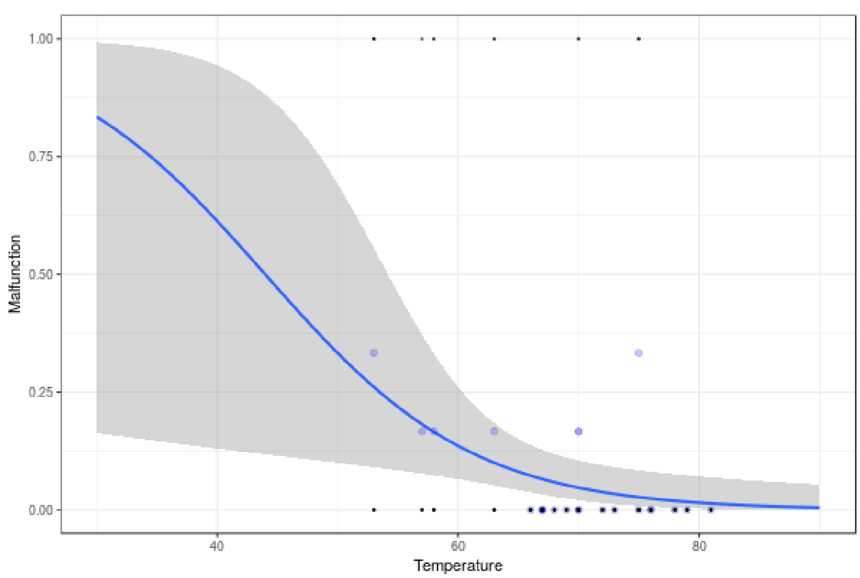
```
logistic_reg_flat = glm(data=data_flat, Malfunction - Temperature, family=binomial(link
summary(logistic_reg)
```

```
Fehler in stats::model.frame(formula = Malfunction - Temperature, data = data_flat,  :
  Objekt 'Malfunction' nicht gefunden
```

```
Fehler in summary(logistic_reg) : Objekt 'logistic_reg' nicht gefunden
```

So, again these objects are not here (same error as above, probably). So, for Arnaud Legrand this is perfect because he sees a result. For me it is not. The estimates and the standard errors for him are the same although the Residual deviance is difference since the distance is now measured with respect to each 0/1 measurement and not to rations. Let's use plot the regression for ***data_flat*** along with the ratios (***data***).

```
ggplot(data=data_flat, aes(y=Malfunction, x=Temperature)) + geom_smooth(method = "glm"
```



This confidence interval seems much more reasonable (in accordance with the data) than the previous one. Let's check whether it corresponds to the prediction obtained when calling directly predict. Obtaining the prediction can be done directly or through the link function.

Here is the "direct" (response) version I used in my very first plot:

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

```
Fehler in predict(logistic_reg_flat, list(Temperature = 30), type = "response",  :
  Objekt 'logistic_reg_flat' nicht gefunden

Fehler: Objekt 'pred' nicht gefunden
```

Again, in my version of this document I cannot find the above defined object anymore. So, I cannot replicate what Arnaud Legrand wrote: The estimated Failure probability for 30° is thus ??. However the *se.fit* value seems pretty hard to use as I can obviously not simply add ± 2 *se.fit* to *fit* to compute a confidence interval.

Here is the "link" version:

```
pred_link = predict(logistic_reg_flat,list(Temperature=39), type="link",se.fit = T)
pred.link
```

```
Fehler in predict(logistic_reg_flat, list(Temperature = 39), type = "link",  :
  Objekt 'logistic_reg_flat' nicht gefunden

Fehler: Objekt 'pred.link' nicht gefunden
```

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

```
Fehler: Objekt 'logistic_reg' nicht gefunden
```

I recover ?? for the Estimated Failure probability at 30°. But now, going through the *linkinv* function, we can use *se.fit*:

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

```
Fehler: Objekt 'logistic_reg' nicht gefunden
```

The 95% confidence interval for our estimation is thus [??,??]. This is what `ggplot2` just plotted me. This seems coherent.

I am now <u>not yet</u> rather confident that I have managed to correctly compute and plot uncertainty of my prediction. Let's be honnest, it took me a wile. My first attempts were plainly wrong (I didn't know how to do this so I trusted `ggplot2`, which I was misusing) and did not use the correct statistical method. I also feel confident now becuase this has been somehow validated by other colleagues but it will be interesting that you collect other kind of plot values that you obtained ,that differ and that you would probably have kept if you didn't have a reference to compare to . Please, provide us with as many versions as you can.

So, I'm disappointed because some error in R or in my config leads to the fact that some objects are forgotten between blocks. I will try to export the whole document in pdf to see if that changes. Unfortunately, it doesn't. Right now I do not have the time to figure out by myself how to change this. So I will only upload this document and hope it still contributes to the database in some way.