regression_logistique

Meline Saubin

2 avril 2020

Analyse de risque de lancement d'une fusée

On souhaite retrouver les résultats de l'analyse de risque effectuée par Siddhartha et al. 1989.

Informations techniques

Nous utiliserons le language R :

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

```
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 8.1 x64 (build 9600)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=French_France.1252  LC_CTYPE=French_France.1252
## [3] LC_MONETARY=French_France.1252 LC_NUMERIC=C
## [5] LC_TIME=French_France.1252
##
## 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] Rcpp_0.12.18     bindr_0.1.1      knitr_1.20       magrittr_1.5
##  [5] tidyselect_0.2.4 munsell_0.5.0    colorspace_1.3-2 R6_2.2.2
##  [9] rlang_0.2.2      stringr_1.3.1    plyr_1.8.4       dplyr_0.7.6
## [13] tools_3.5.1      grid_3.5.1       gtable_0.2.0     withr_2.1.2
## [17] htmltools_0.3.6  assertthat_0.2.0 yaml_2.2.0       lazyeval_0.2.1
## [21] rprojroot_1.3-2  digest_0.6.17    tibble_1.4.2     crayon_1.3.4
## [25] bindrcpp_0.2.2   purrr_0.2.5      glue_1.3.0       evaluate_0.11
## [29] rmarkdown_1.10   stringi_1.1.7    compiler_3.5.1   pillar_1.3.0
## [33] scales_1.0.0     backports_1.1.2  pkgconfig_2.0.2
```

Avec les librairies disponibles :

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

```
## Session info -------------------------------------------------------------
```

```
##  setting  value
##  version  R version 3.5.1 (2018-07-02)
##  system   x86_64, mingw32
##  ui       RTerm
```

```
##  language (EN)
##  collate  French_France.1252
##  tz       Europe/Paris
##  date     2020-04-02
```

```
## Packages -----------------------------------------------------------------
```

```
##  package    * version date       source
##  assertthat   0.2.0   2017-04-11 CRAN (R 3.5.1)
##  backports    1.1.2   2017-12-13 CRAN (R 3.5.0)
##  base       * 3.5.1   2018-07-02 local
##  bindr        0.1.1   2018-03-13 CRAN (R 3.5.1)
##  bindrcpp     0.2.2   2018-03-29 CRAN (R 3.5.1)
##  colorspace   1.3-2   2016-12-14 CRAN (R 3.5.1)
##  compiler     3.5.1   2018-07-02 local
##  crayon       1.3.4   2017-09-16 CRAN (R 3.5.1)
##  datasets   * 3.5.1   2018-07-02 local
##  devtools     1.13.6  2018-06-27 CRAN (R 3.5.1)
##  digest       0.6.17  2018-09-12 CRAN (R 3.5.1)
##  dplyr        0.7.6   2018-06-29 CRAN (R 3.5.1)
##  evaluate     0.11    2018-07-17 CRAN (R 3.5.1)
##  ggplot2    * 3.0.0   2018-07-03 CRAN (R 3.5.1)
##  glue         1.3.0   2018-07-17 CRAN (R 3.5.1)
##  graphics   * 3.5.1   2018-07-02 local
##  grDevices  * 3.5.1   2018-07-02 local
##  grid         3.5.1   2018-07-02 local
##  gtable       0.2.0   2016-02-26 CRAN (R 3.5.1)
##  htmltools    0.3.6   2017-04-28 CRAN (R 3.5.1)
##  knitr        1.20    2018-02-20 CRAN (R 3.5.1)
##  lazyeval     0.2.1   2017-10-29 CRAN (R 3.5.1)
##  magrittr     1.5     2014-11-22 CRAN (R 3.5.1)
##  memoise      1.1.0   2017-04-21 CRAN (R 3.5.1)
##  methods    * 3.5.1   2018-07-02 local
##  munsell      0.5.0   2018-06-12 CRAN (R 3.5.1)
##  pillar       1.3.0   2018-07-14 CRAN (R 3.5.1)
##  pkgconfig    2.0.2   2018-08-16 CRAN (R 3.5.1)
##  plyr         1.8.4   2016-06-08 CRAN (R 3.5.1)
##  purrr        0.2.5   2018-05-29 CRAN (R 3.5.1)
##  R6           2.2.2   2017-06-17 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)
##  rmarkdown    1.10    2018-06-11 CRAN (R 3.5.1)
##  rprojroot    1.3-2   2018-01-03 CRAN (R 3.5.1)
##  scales       1.0.0   2018-08-09 CRAN (R 3.5.1)
##  stats      * 3.5.1   2018-07-02 local
##  stringi      1.1.7   2018-03-12 CRAN (R 3.5.0)
##  stringr      1.3.1   2018-05-10 CRAN (R 3.5.1)
##  tibble       1.4.2   2018-01-22 CRAN (R 3.5.1)
##  tidyselect   0.2.4   2018-02-26 CRAN (R 3.5.1)
##  tools        3.5.1   2018-07-02 local
##  utils      * 3.5.1   2018-07-02 local
##  withr        2.1.2   2018-03-15 CRAN (R 3.5.1)
##  yaml         2.2.0   2018-07-25 CRAN (R 3.5.1)
```

Jeu de données

On importe le jeu de données à l'adresse suivante : https://app-learninglab.inria.fr/moocrr/gitlab/moocrr-session3/moocrr-reproducibility-study/raw/master/data/shuttle.csv?inline=false

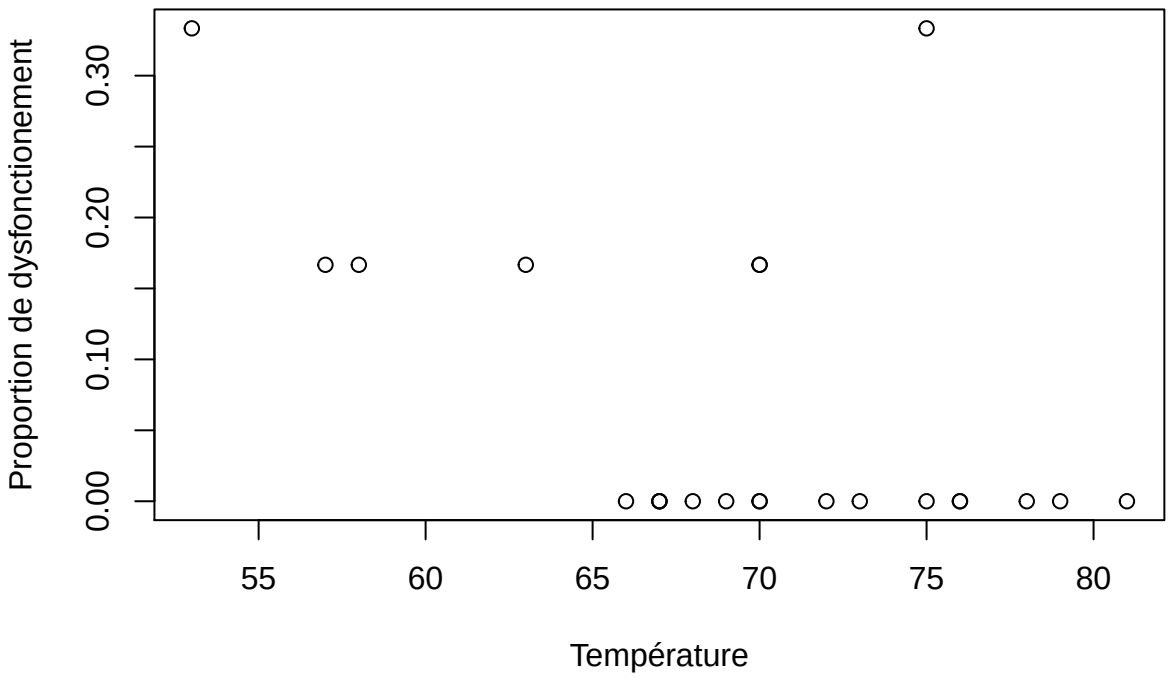
```
data = read.csv("https://app-learninglab.inria.fr/moocrr/gitlab/moocrr-session3/moocrr-reproducibility-s
data
```

```
##         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
```

Premiers graphiques et question scientifique

La question que l'on se pose est : y-a-t-il un lien entre la proportion de dysfonctionnements et la température, si oui lequel? On peut donc commencer par tracer la proportion de dysfonctionnements en fonction de la température :

```
# Ajout de la colonne Prop_malfunction pour la proportion de dysfonctionnements :
data$Prop_malfunction = data$Malfunction/data$Count

plot(data$Temperature, data$Prop_malfunction, xlab = "Température",
     ylab = "Proportion de dysfonctionement")
```



On a ici l'impression que pour de faibles températures, la proportion de dysfonctionnements est plus élevée. Cette observation n'a aucune valeur statistique, il va donc falloir construire un modèle de régression logistique pour vérifier cette hypothèse.

Régression logistique

```
reg_log = glm(Prop_malfunction ~ Temperature, weights = Count, family = binomial(link="logit"),
              data = data)

summary(reg_log)
```

```
##
## Call:
## glm(formula = Prop_malfunction ~ Temperature, family = binomial(link = "logit"),
##     data = data, weights = Count)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.95227  -0.78299  -0.54117  -0.04379   2.65152
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.08498    3.05247   1.666   0.0957 .
## Temperature -0.11560    0.04702  -2.458   0.0140 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

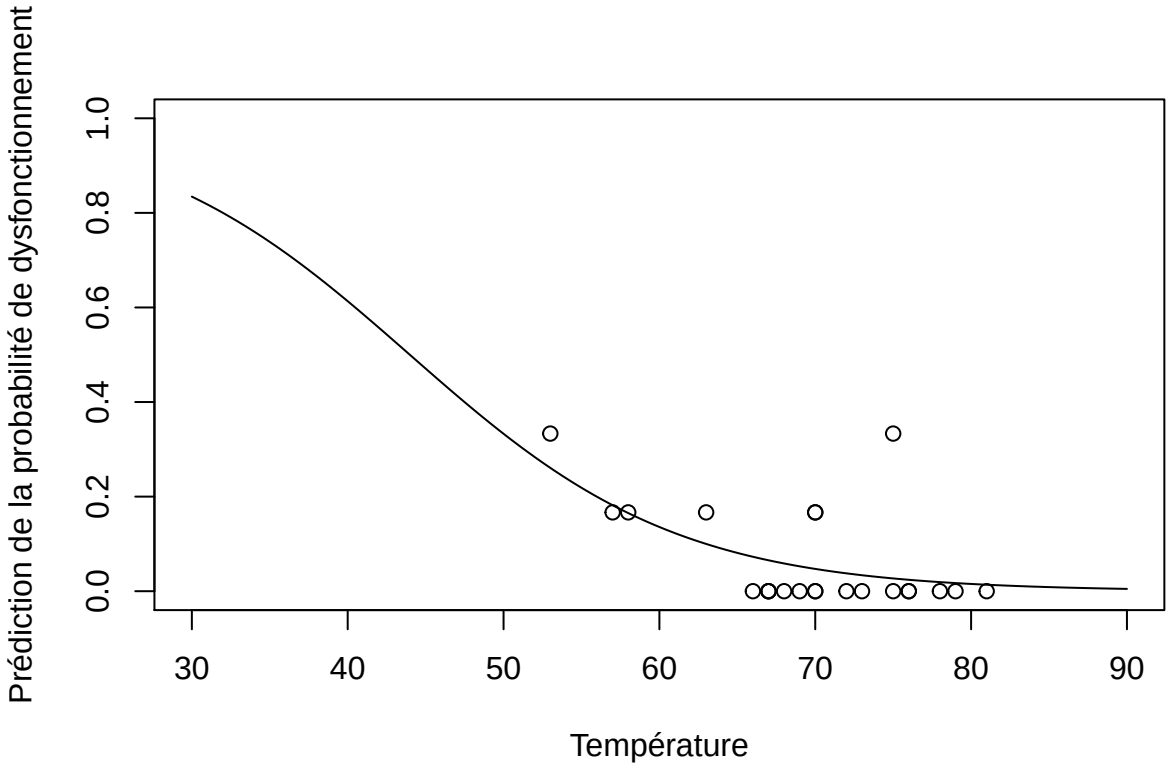
```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 24.230  on 22  degrees of freedom
## Residual deviance: 18.086  on 21  degrees of freedom
## AIC: 35.647
##
## Number of Fisher Scoring iterations: 5
```

Le nombre d'itérations de fisher est suffisamment faible, on peut donc observer que la température a une influence significative sur la proportion de dysfonctionnement, en choissiant un seuil de risque de 5%. On retrouve les informations suivantes : $\hat{\alpha} = 5.08498$ et $\hat{\beta} = -0.11560$ avec leurs erreurs standard $s_{\hat{\alpha}} = 3.05247$ et $s_{\hat{\beta}} = 0.04702$. La déviance résiduelle est $G^2 = 18.086$ avec 21 degrés de liberté.

Courbes de prédiction

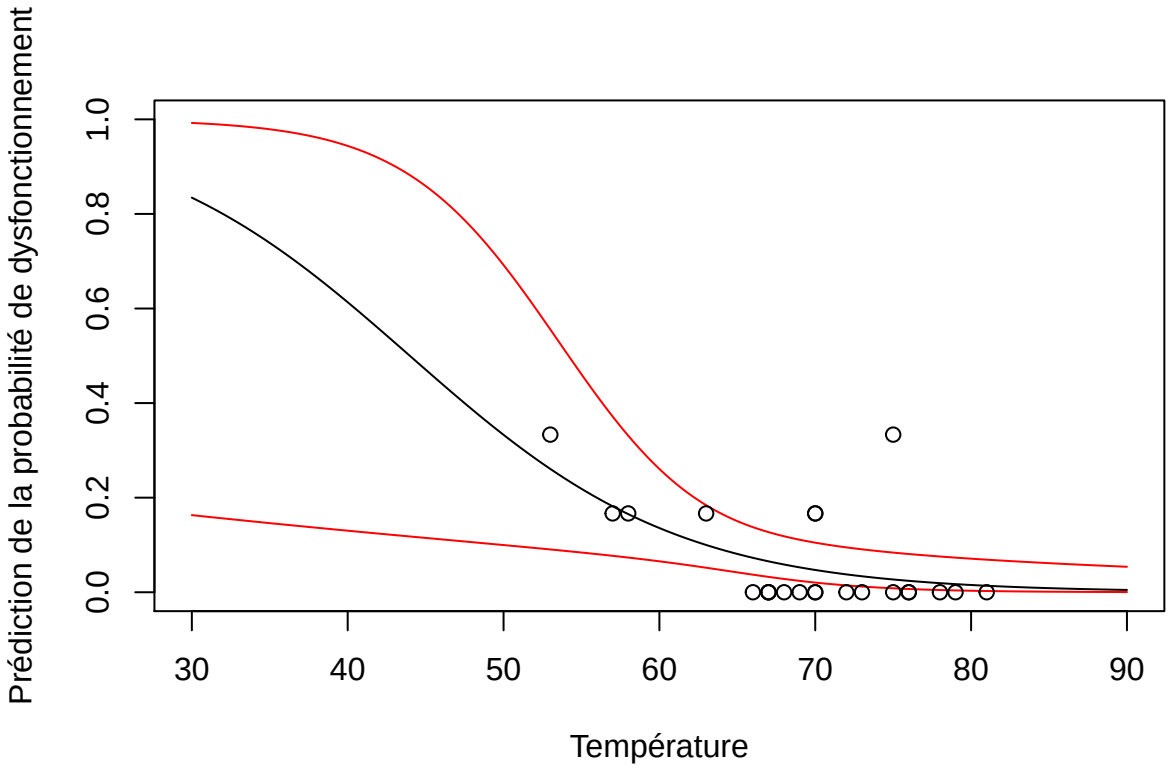
On sait que pour le lancement en question, la température était de 31°F, or ici on n'a un jeu de données que sur un intervalle de températures de 53 à 81°F, on doit donc élargir l'intervalle pour la prédiction :

```
intervalle_temp = seq(from=30, to=90, by = .1)
pred = predict(reg_log,list(Temperature=intervalle_temp),type="response")
plot(intervalle_temp,pred,type="l",ylim=c(0,1), xlab = "Température",
     ylab = "Prédiction de la probabilité de dysfonctionnement")
points(data$Prop_malfunction ~ data$Temperature)
```



On aurait donc, d'après cette prédiction, une probabilité de dysfonctionnement d'environ 0.8 à la température de 31°F. Cependant on n'a pas pris en compte l'intervalle de confiance de cette prédiction. Le code suivant s'inspire des résultats de recherches sur un forum de discussion pour construire un intervalle de confiance sur une prédiction.

```r
intervalle_temp = seq(from=30, to=90, by = .1)

# On ajoute l'argument se.fit = TRUE pour avoir les erreurs standard, avec cette fois le type "link"
pred = predict(reg_log,list(Temperature=intervalle_temp),type="link", se.fit = TRUE)

# Pour un intervalle de confiance à 95%,
# On suppose une distribution de loi normale des erreurs autour de la valeur prédite
critval <- 1.96

upr <- pred$fit + (critval * pred$se.fit)
lwr <- pred$fit - (critval * pred$se.fit)
fit <- pred$fit

# On applique ensuite la fonction de lien inversée
pred$fit <- reg_log$family$linkinv(fit)
pred$upr <- reg_log$family$linkinv(upr)
pred$lwr <- reg_log$family$linkinv(lwr)


plot(intervalle_temp,pred$fit,type="l",ylim=c(0,1), xlab = "Température",
     ylab = "Prédiction de la probabilité de dysfonctionnement")
lines(intervalle_temp,pred$lwr,type="l",ylim=c(0,1), xlab = "Température",
      ylab = "Prédiction de la probabilité de dysfonctionnement", col = 'red')
lines(intervalle_temp,pred$upr,type="l",ylim=c(0,1), xlab = "Température",
      ylab = "Prédiction de la probabilité de dysfonctionnement", col = 'red')
points(data$Prop_malfunction ~ data$Temperature)
```



On retrouve une courbe similaire à l'article, avec la proportion de dysfonctionnement, et non le nombre de dysfonctionnement. Cependant, elle n'est quand même pas identique à la courbe de l'article.