TD TP 1 : Régression logistique (correction)

Exercice 1

Charger les données "prostate"

On vérifie le type des variables.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
glimpse(prostate)
```

```
## Observations: 53
## Variables: 6
## $ age     <int> 66, 68, 66, 56, 58, 60, 65, 60, 50, 49, 61, 58, 51, 67...
## $ acid    <dbl> 0.48, 0.56, 0.50, 0.52, 0.50, 0.49, 0.46, 0.62, 0.56, ...
## $ radio   <int> 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, ...
## $ taille  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ gravite <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, ...
## $ lymph   <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, ...
```

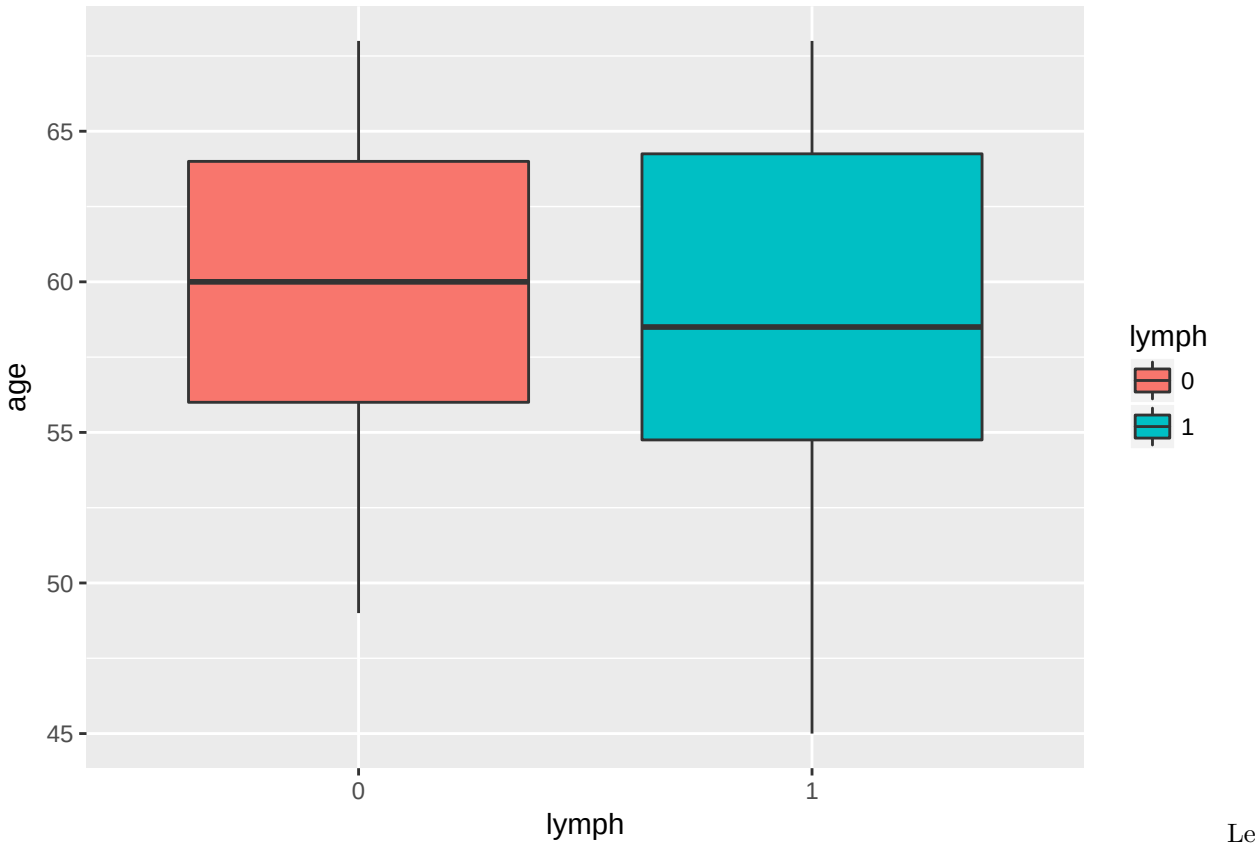
```
prostate = mutate(prostate, radio = as.factor(radio))
prostate = mutate(prostate, taille = as.factor(taille))
prostate = mutate(prostate, gravite = as.factor(gravite))
prostate = mutate(prostate, lymph = as.factor(lymph))
glimpse(prostate)
```

```
## Observations: 53
## Variables: 6
## $ age     <int> 66, 68, 66, 56, 58, 60, 65, 60, 50, 49, 61, 58, 51, 67...
## $ acid    <dbl> 0.48, 0.56, 0.50, 0.52, 0.50, 0.49, 0.46, 0.62, 0.56, ...
## $ radio   <fct> 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, ...
## $ taille  <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ gravite <fct> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, ...
## $ lymph   <fct> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, ...
```

Graphiques pour variables continues

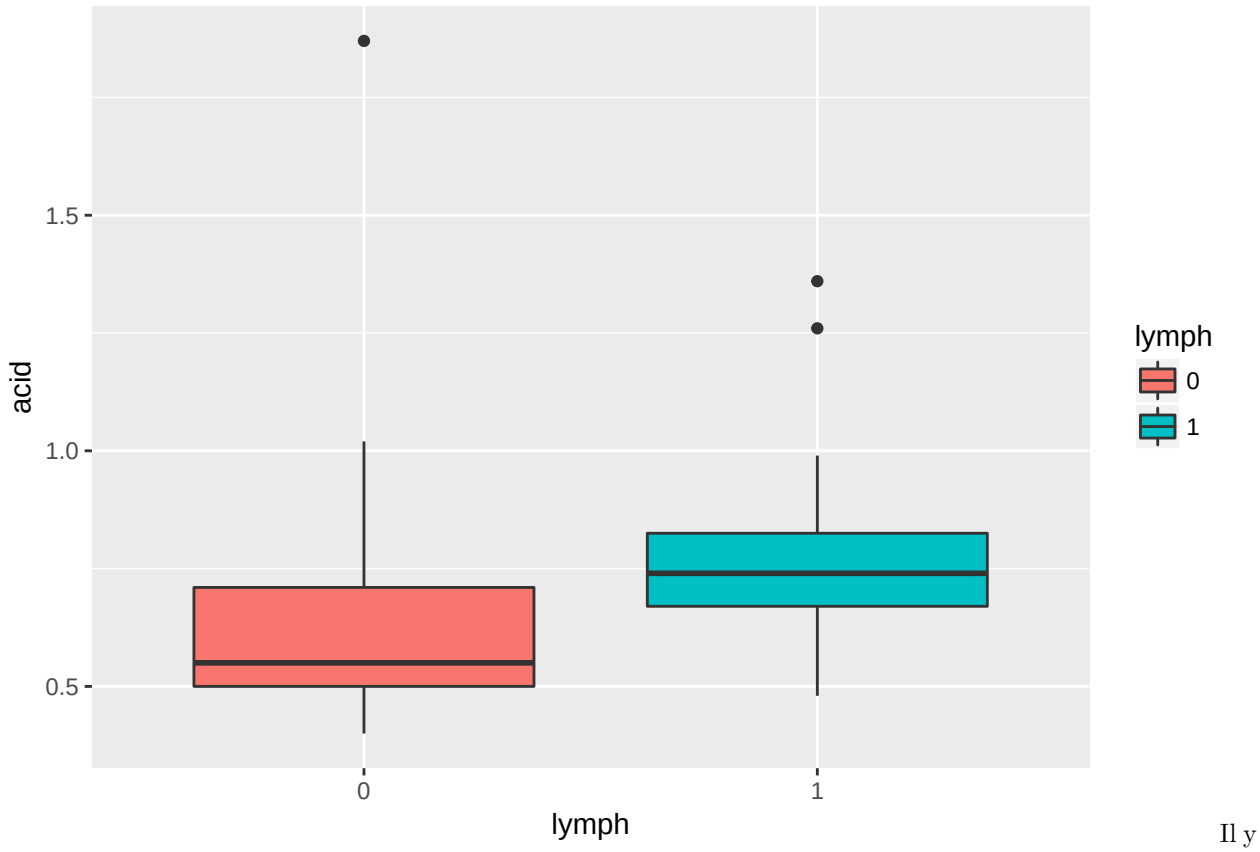
On étudie le lien entre la variable cible `lymph` et les variables continues avec des boxplots.

```
library(ggplot2)
ggplot(prostate, aes(lymph,age)) +
  geom_boxplot(aes(fill = lymph))
```



Le lien entre l'âge et la variable `lymph` est faible.

```
ggplot(prostate, aes(lymph,acid)) +
  geom_boxplot(aes(fill = lymph))
```

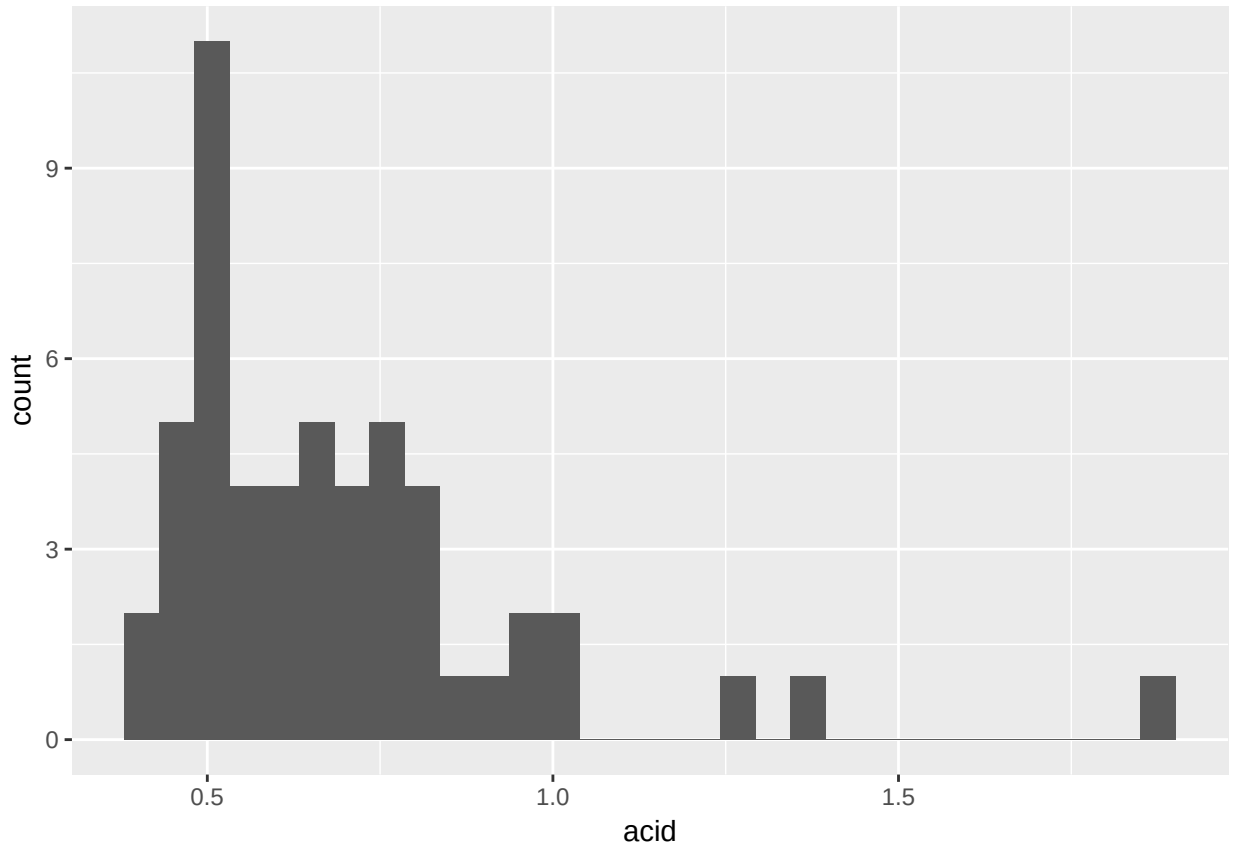


Il y a un fort lien entre les variables `acid` et `lymph`.

On vérifie maintenant la loi de la variable `acid`.

```
ggplot(prostate, aes(acid)) + geom_histogram()
```

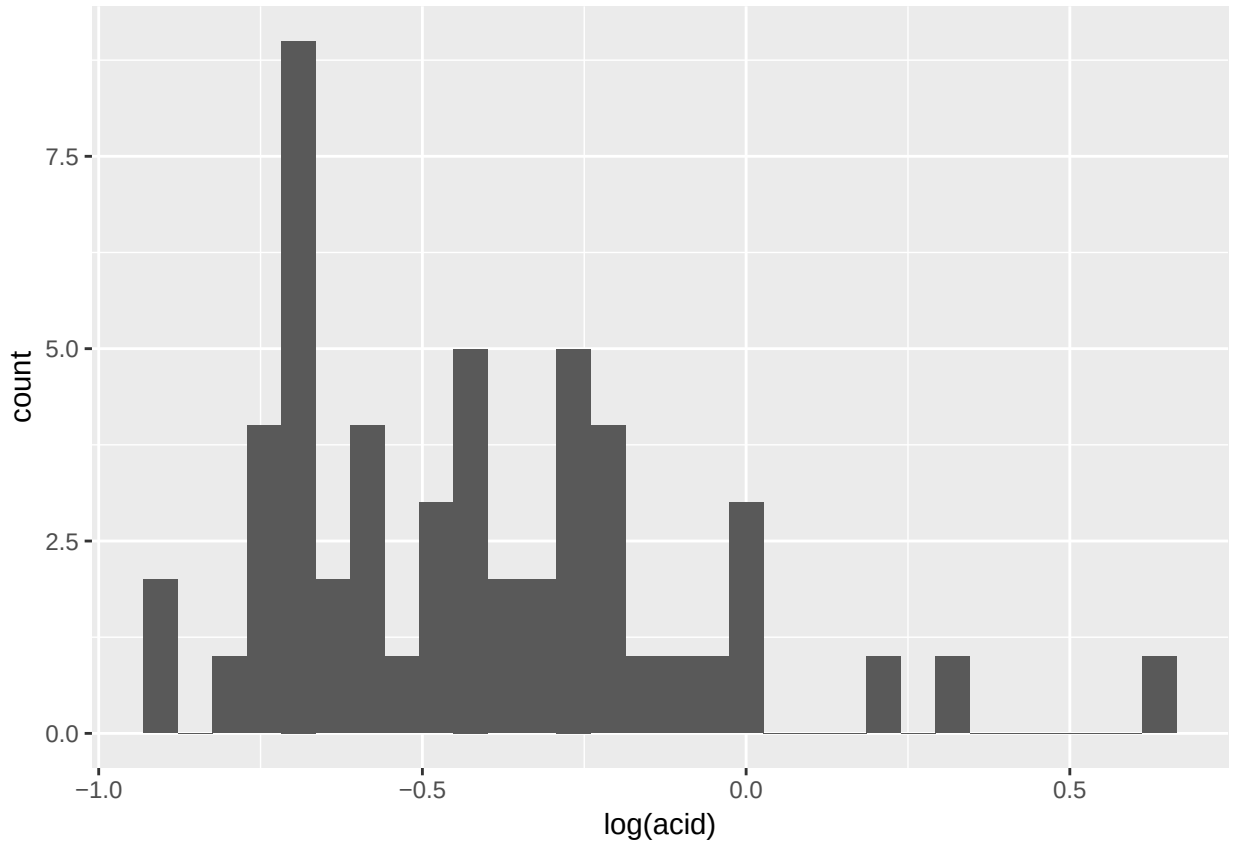
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



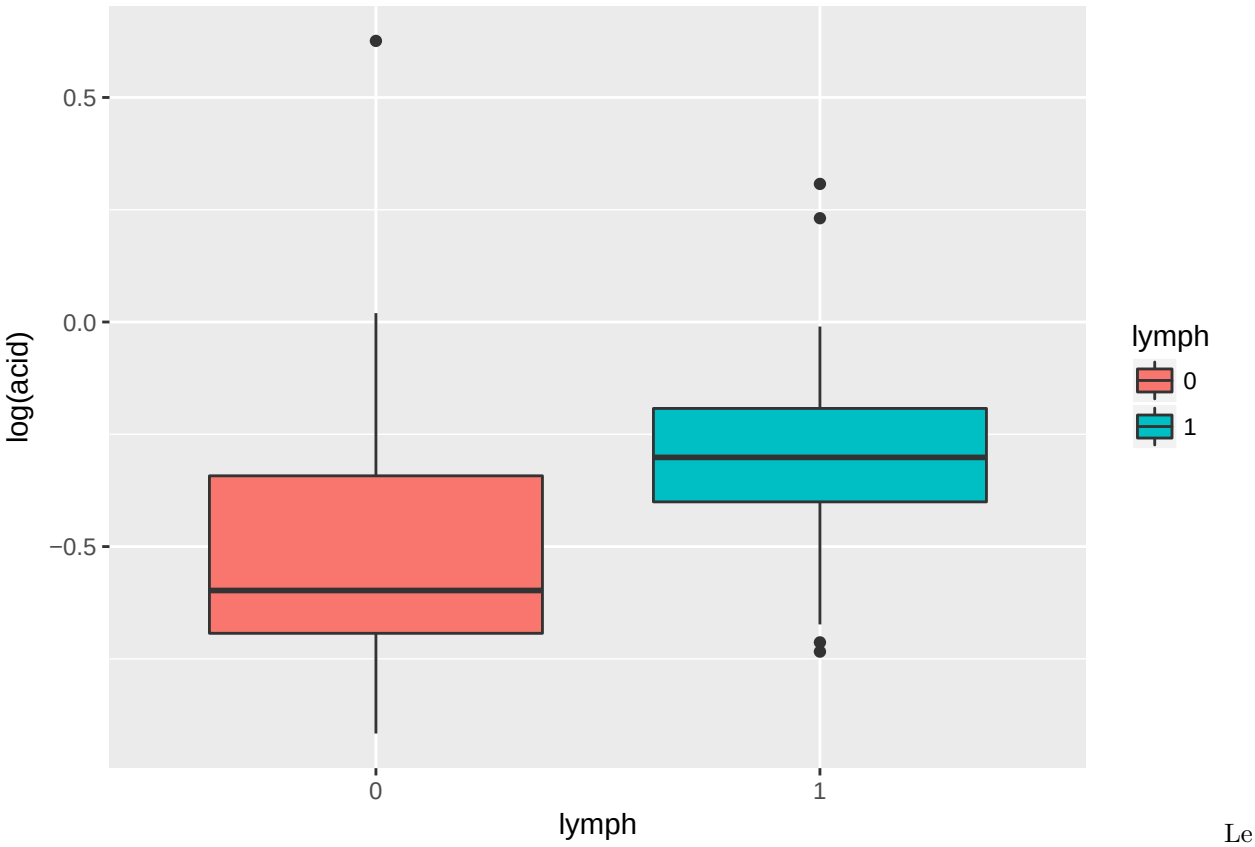
La loi est très dissymétrique, on préférera la transformer (log)

```
ggplot(prostate, aes(log(acid))) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(prostate, aes(lymph,log(acid))) +
  geom_boxplot(aes(fill = lymph))
```



Le lien entre `log(acid)` et `lymph` est toujours fort.

Variables explicatives discrètes

Pour les variables explicatives discrètes, on va étudier leur lien avec `lymph` via des tests du χ^2 d'indépendance : une p-value faible indique un lien fort.

```
table(prostate$lymph,prostate$radio)
```

```
##
##      0  1
##   0 29  4
##   1  9 11
```

```
chisq.test(prostate$lymph,prostate$radio)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  prostate$lymph and prostate$radio
## X-squared = 9.269, df = 1, p-value = 0.002331
```

```
table(prostate$lymph,prostate$taille)
```

```
##
##      0  1
##   0 21 12
##   1  5 15
```

```r
chisq.test(prostate$lymph,prostate$taille)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  prostate$lymph and prostate$taille
## X-squared = 5.9727, df = 1, p-value = 0.01453
```

```r
table(prostate$lymph,prostate$gravite)
```

```
##
##      0  1
##   0 24  9
##   1  9 11
```

```r
chisq.test(prostate$lymph,prostate$gravite)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  prostate$lymph and prostate$gravite
## X-squared = 2.98, df = 1, p-value = 0.0843
```

Les variables `taille` et `radio` semblent très liées à `lymph`. Ce n'est pas le cas pour la variable `gravite`.

Exercice 2 :

Premier modèle logistique

Voici le résultat de l'estimation dans un premier modèle logistique avec toutes les variables explicatives disponibles.

```r
fit_logistic = glm(lymph ~ age + log(acid) + radio + gravite + taille, family = "binomial" ,data = pros
summary(fit_logistic)
```

```
##
## Call:
## glm(formula = lymph ~ age + log(acid) + radio + gravite + taille,
##     family = "binomial", data = prostate)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0371  -0.6794  -0.3320   0.5845   2.0499
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.39107    3.53460   0.676   0.4987
## age         -0.06266    0.05903  -1.061   0.2885
## log(acid)    2.58494    1.19679   2.160   0.0308 *
## radio1       2.04541    0.82969   2.465   0.0137 *
## gravite1     0.84018    0.78902   1.065   0.2869
## taille1      1.55508    0.78099   1.991   0.0465 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 70.252  on 52  degrees of freedom
## Residual deviance: 46.611  on 47  degrees of freedom
## AIC: 58.611
##
## Number of Fisher Scoring iterations: 5
```

On retrouve sur les p-values des tests de Wald univariés ($\mathcal{H}_0 : \beta_j^\star = 0$) les liens ou les absences de lien observés précédemment.

Prédictions, matrice de confusion

Prédictions

On calcule les prédictions construites à partir de ce 1er modèle. `predictions` donne la valeurs des $\hat{\pi}$ pour tous les individus, `predictions_01` vaut 1 si `predictions` est $> 1/2$.

```
?predict.glm
predictions = predict(fit_logistic,type = "response")
predictions_01 = predictions > 1/2
```

```
prostate$lymph
```

```
##  [1] 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1
## [36] 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
## Levels: 0 1
```

Matrice de confusion

On peut alors calculé la matrice de confusion

```
table(as.numeric(predictions_01),prostate$lymph)
```

```
##
##      0  1
##   0 30  6
##   1  3 14
```

Il y a 3/33 faux positifs et 6/20 faux négatifs.

Courbe ROC et AUC

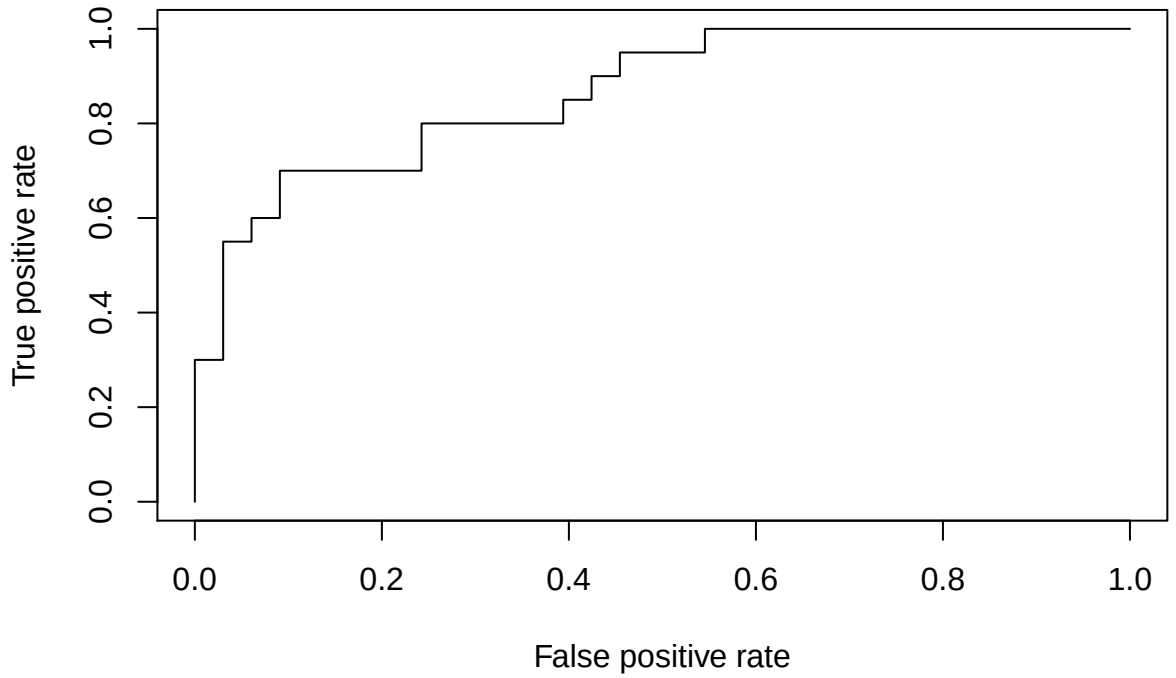
```
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
pred = prediction( predictions , prostate$lymph )
perf = performance( pred, "tpr" ,"fpr" )
plot( perf )
```




```
ROC_auc = performance( pred,"auc")
AUC = ROC_auc@y.values[[1]]
print(AUC)
```

```
## [1] 0.8651515
```

L'AUC est de 0.86.

Test de nullité simultanée des coefficients

```
fit_null = glm(lymph ~ 1, family = "binomial", data=prostate)
summary(fit_null)
```

```
##
## Call:
## glm(formula = lymph ~ 1, family = "binomial", data = prostate)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9734  -0.9734  -0.9734   1.3961   1.3961
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.5008     0.2834  -1.767   0.0772 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 70.252  on 52  degrees of freedom
```

```
## Residual deviance: 70.252  on 52  degrees of freedom
## AIC: 72.252
##
## Number of Fisher Scoring iterations: 4
```

```
anova(fit_null,fit_logistic,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: lymph ~ 1
## Model 2: lymph ~ age + log(acid) + radio + gravite + taille
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        52     70.252
## 2        47     46.611  5   23.641 0.0002545 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ou

```
fit_logistic$null.deviance - fit_logistic$deviance
```

```
## [1] 23.64097
```

```
qchisq(1-0.05,5)
```

```
## [1] 11.0705
```

```
1 - pchisq(fit_logistic$null.deviance - fit_logistic$deviance , fit_logistic$rank-1)
```

```
## [1] 0.0002544584
```

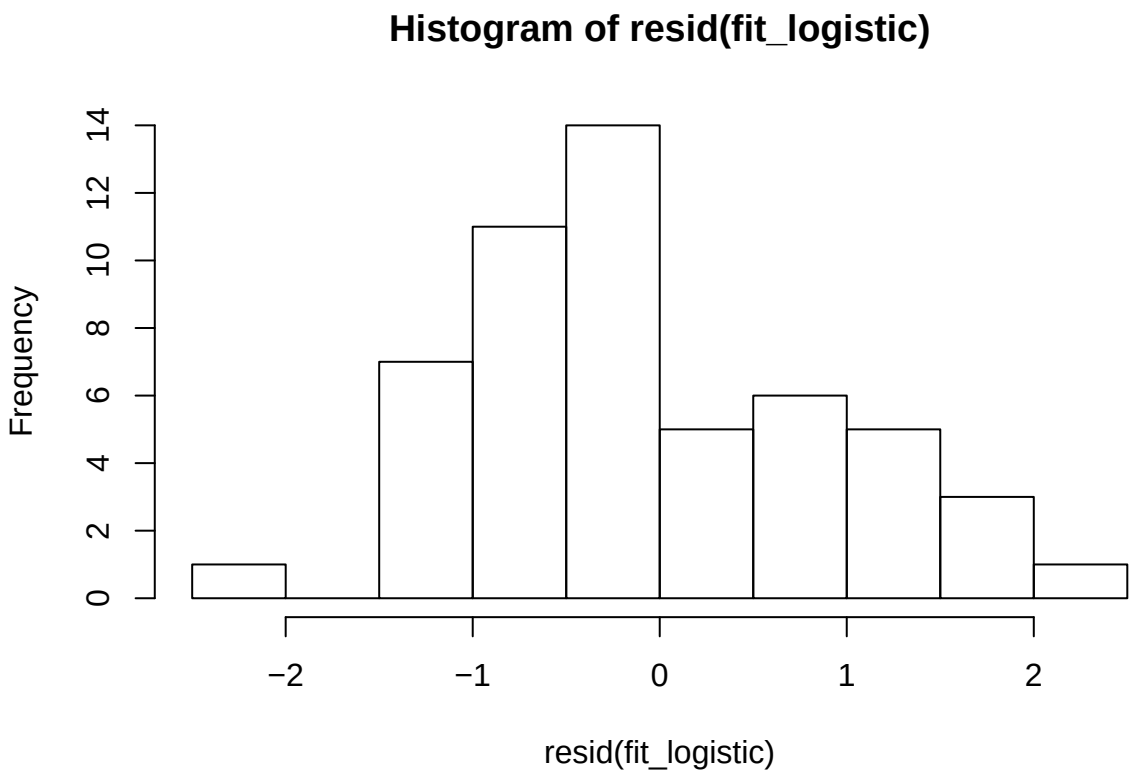
```
# pchisq est la fdr de la loi chisq
```

On rejette l'hypothèse de nullité simultanée des coefficients des variables explicatives (au niveau de 5%).

Diagnostics

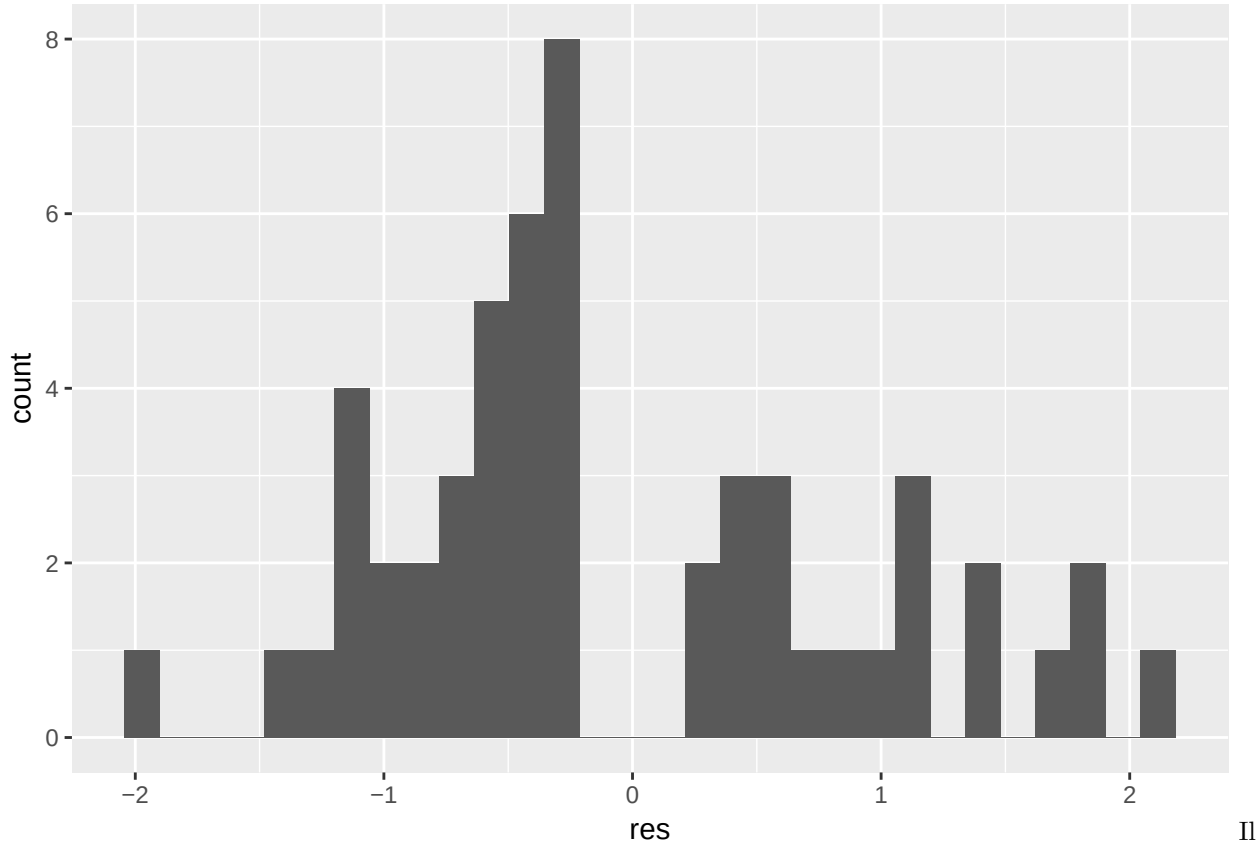
Observations aberrantes

```
hist(resid(fit_logistic))
```




```
ggplot(data.frame(res=resid(fit_logistic)),aes(res))+geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Il

n'y pas de grand résidus, trop pas d'observation aberrante. On garde les observations 35 et 37 car leurs résidus sont proches de 2 (en valeur absolue).

```
residu = resid(fit_logistic , type="deviance")

residu[which(abs(residu)>2)]
```

```
##        35        37
##  2.049846 -2.037109
```

```
sort( residu )
```

```
##         37         41         38         24         31         39
## -2.0371089 -1.3475413 -1.2527216 -1.1770339 -1.1701418 -1.1419520
##         10         40          8         32         29         20
## -1.1023963 -1.0191332 -0.9511801 -0.8932444 -0.7863201 -0.7353056
##         36         19         27          7         28         13
## -0.7309083 -0.6793834 -0.6086334 -0.5945384 -0.5879821 -0.5235905
##         18         17         12         22         30         16
## -0.5013552 -0.4876984 -0.4742849 -0.4371643 -0.4228251 -0.3698025
##         11          4         21         15          5          6
## -0.3665479 -0.3422731 -0.3320065 -0.3243858 -0.3064912 -0.2809870
##          2          3          1         53         43         46
## -0.2602600 -0.2396449 -0.2274908  0.2870655  0.2890839  0.3640485
##         44         42         51         52         34         25
##  0.4241846  0.4353853  0.5165271  0.5845411  0.6290111  0.6611841
##         45         33         50         47         14         23
##  0.7763993  0.9892048  1.1020953  1.1177321  1.1589854  1.3919361
##         49         48          9         26         35
##  1.4020724  1.7497429  1.8003787  1.9023427  2.0498456
```

Diagnostic sur X

```
X = model.matrix(fit_logistic)

eigen(cor(X[,-1]))
```

```
## $values
## [1] 1.5321277 1.1177192 0.9999733 0.7389985 0.6111813
##
## $vectors
##               [,1]       [,2]        [,3]        [,4]        [,5]
## [1,] -0.0008232896  0.1298447  0.98891861 -0.02073825  0.06891728
## [2,] -0.1079319045  0.8172021 -0.13282966 -0.50776754  0.21227318
## [3,] -0.4947398950  0.4297146 -0.03016913  0.73745685 -0.16070151
## [4,] -0.6064775813 -0.3124825 -0.01160631 -0.06662323  0.72798782
## [5,] -0.6129986997 -0.1817172  0.05789120 -0.43984178 -0.62801215
```

Il n'y pas de problème de colinéarité entre les covariables. #### Leviers d'observations

```
influences = influence(fit_logistic)
hat = influences$hat
hat[hat > 2*5/nrow(prostate)]
```

```
##        10        14        23        24        33        45        47
## 0.2738598 0.2920499 0.2179688 0.3842499 0.2229037 0.2094289 0.2385361
```

Il y a quelques observations influentes. On les garde pour l'instant, on pourra les enlever ensuite.

Procédure de sélection backward via le test de Wald

```
fit_logistic = glm(lymph ~ age + log(acid) + radio + gravite + taille, family = "binomial" ,data = pros
summary(fit_logistic)
```

```
##
## Call:
## glm(formula = lymph ~ age + log(acid) + radio + gravite + taille,
##     family = "binomial", data = prostate)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0371  -0.6794  -0.3320   0.5845   2.0499
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.39107    3.53460   0.676   0.4987
## age         -0.06266    0.05903  -1.061   0.2885
## log(acid)    2.58494    1.19679   2.160   0.0308 *
## radio1       2.04541    0.82969   2.465   0.0137 *
## gravite1     0.84018    0.78902   1.065   0.2869
## taille1      1.55508    0.78099   1.991   0.0465 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 70.252  on 52  degrees of freedom
## Residual deviance: 46.611  on 47  degrees of freedom
## AIC: 58.611
##
## Number of Fisher Scoring iterations: 5
```

Step 2 : on enlève age

```
fit_logistic2 = glm(lymph ~  log(acid) + radio + gravite + taille, family = "binomial" ,data = prostate
summary(fit_logistic2)
```

```
##
## Call:
## glm(formula = lymph ~ log(acid) + radio + gravite + taille, family = "binomial",
##     data = prostate)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0364  -0.7114  -0.3197   0.6412   2.0244
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.3055     0.7270  -1.796   0.0725 .
## log(acid)     2.5116     1.1730   2.141   0.0323 *
## radio1        2.0107     0.8212   2.448   0.0143 *
```

```
## gravite1      0.8507     0.7752   1.097   0.2725
## taille1       1.5435     0.7800   1.979   0.0478 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 70.252  on 52  degrees of freedom
## Residual deviance: 47.776  on 48  degrees of freedom
## AIC: 57.776
##
## Number of Fisher Scoring iterations: 5
```

Step 3 : on enlève gravite

```
fit_logistic3 = glm(lymph ~  log(acid) + radio + taille, family = "binomial" ,data = prostate)
summary(fit_logistic3)
```

```
##
## Call:
## glm(formula = lymph ~ log(acid) + radio + taille, family = "binomial",
##     data = prostate)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8714  -0.7521  -0.3456   0.5363   2.2826
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.1994     0.7162  -1.675  0.09400 .
## log(acid)     2.2922     1.1387   2.013  0.04412 *
## radio1        2.0550     0.7976   2.576  0.00998 **
## taille1       1.7638     0.7483   2.357  0.01842 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 70.252  on 52  degrees of freedom
## Residual deviance: 48.986  on 49  degrees of freedom
## AIC: 56.986
##
## Number of Fisher Scoring iterations: 5
```

On compare avec une sélection par AIC stepwise (par défaut dans `step`)

```
step(fit_logistic)
```

```
## Start:  AIC=58.61
## lymph ~ age + log(acid) + radio + gravite + taille
##
##             Df Deviance    AIC
## - gravite    1   47.747 57.747
## - age        1   47.776 57.776
## <none>           46.611 58.611
## - taille     1   50.893 60.893
```

```
## - log(acid)  1   51.676 61.676
## - radio      1   53.453 63.453
##
## Step:  AIC=57.75
## lymph ~ age + log(acid) + radio + taille
##
##             Df Deviance    AIC
## - age        1   48.986 56.986
## <none>           47.747 57.747
## - log(acid)  1   52.201 60.201
## - taille     1   53.949 61.949
## - radio      1   55.323 63.323
##
## Step:  AIC=56.99
## lymph ~ log(acid) + radio + taille
##
##             Df Deviance    AIC
## <none>           48.986 56.986
## - log(acid)  1   53.353 59.353
## - taille     1   55.272 61.272
## - radio      1   56.484 62.484
##
## Call:  glm(formula = lymph ~ log(acid) + radio + taille, family = "binomial",
##     data = prostate)
##
## Coefficients:
## (Intercept)    log(acid)       radio1      taille1
##      -1.199        2.292        2.055        1.764
##
## Degrees of Freedom: 52 Total (i.e. Null);  49 Residual
## Null Deviance:       70.25
## Residual Deviance: 48.99     AIC: 56.99
```

On obtient le même modèle final.

Interactions

On essaie de compliquer le modèle en ajoutant les interactions.

```
fit_logistic_interactions = glm(lymph ~ radio * taille + log(acid) * radio +  log(acid) * taille, family
summary(fit_logistic_interactions)
```

```
##
## Call:
## glm(formula = lymph ~ radio * taille + log(acid) * radio + log(acid) *
##     taille, family = "binomial", data = prostate)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0733  -0.7220  -0.4232   0.3070   2.1352
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.1667     0.8016  -1.455    0.146
```

```
## radio1              3.1699     2.4773   1.280    0.201
## taille1             1.1983     1.2782   0.937    0.349
## log(acid)           1.7332     1.5975   1.085    0.278
## radio1:taille1      1.3012     1.8869   0.690    0.490
## radio1:log(acid)    4.0052     4.3522   0.920    0.357
## taille1:log(acid)  -0.3767     2.4368  -0.155    0.877
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 70.252  on 52  degrees of freedom
## Residual deviance: 47.257  on 46  degrees of freedom
## AIC: 61.257
##
## Number of Fisher Scoring iterations: 6
```

```
#glm(lymph ~ (radio + taille + log(acid))^2, family = "binomial" ,data = prostate )
```

Les variables d'interactions changent la valeurs des coefficients estimés mais les p-values des tests de Wald associées sont grandes, on suspecte donc qu'elles n'apportent peu au modèle. On refait donc une procédure de sélection (via AIC stepwise)

```
fit_final = step(fit_logistic_interactions,trace = 0)
summary(fit_final)
```

```
##
## Call:
## glm(formula = lymph ~ radio + taille + log(acid), family = "binomial",
##     data = prostate)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8714  -0.7521  -0.3456   0.5363   2.2826
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.1994     0.7162  -1.675  0.09400 .
## radio1        2.0550     0.7976   2.576  0.00998 **
## taille1       1.7638     0.7483   2.357  0.01842 *
## log(acid)     2.2922     1.1387   2.013  0.04412 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 70.252  on 52  degrees of freedom
## Residual deviance: 48.986  on 49  degrees of freedom
## AIC: 56.986
##
## Number of Fisher Scoring iterations: 5
```

On retrouve le même modèle à 3 variables `radio`, `taille` et `log(acid)`. Les 3 variables ont des coefficients estimés positifs, leur augmentation entraîne donc une augmentation de la probabilité que `lymph` soit égal à 1.