Analyse statistique avec R

Arthur Tenenhaus et Vincent Guillemot

Thursday, September 10, 2015

La Régression multiple

La régression mutiple est une méthode statistique adaptée à l'étude de la liaison entre une variable quantitative Y et un ensemble de p variables explicatives $X_1, X_2, \ldots, X_p$ quantitatives ou qualitatives. L'exemple compagnon de cette séance, **prévision du prix d'une automobile**, servira à illustrer cette méthode.

Les commandes suivantes permettent de charger et mettre en forme le jeu de données AUTO.

```
library(pheatmap)
A <- read.table("AUTO.csv", header=TRUE, sep="\t")
rownames(A) = A[, 1]
A = A[, -1]
head(A)
```

```
##                   CYL PUI LON LAR POIDS VITESSE  PRIX
## ALFASUD-TI-1350  1350  79 393 161   870     165 30579
## AUDI-100-L       1588  85 468 177  1110     160 39990
## SIMCA-1307-GLS   1294  68 424 168  1050     152 29600
## CITROEN-GS-CLUB  1222  59 412 161   930     151 28250
## FIAT-132-1600GLS 1585  98 439 164  1105     165 34900
## LANCIA-BETA-1300 1297  82 429 169  1080     160 35480
```

On se propose de construire un modèle de prédiction du prix d'une automobile à partir des 6 variables caractéristiques suivantes.

```
colnames(A)[-7]
```

```
## [1] "CYL"     "PUI"     "LON"     "LAR"     "POIDS"   "VITESSE"
```

Ces variables ont été mesurées sur 18 automobiles.

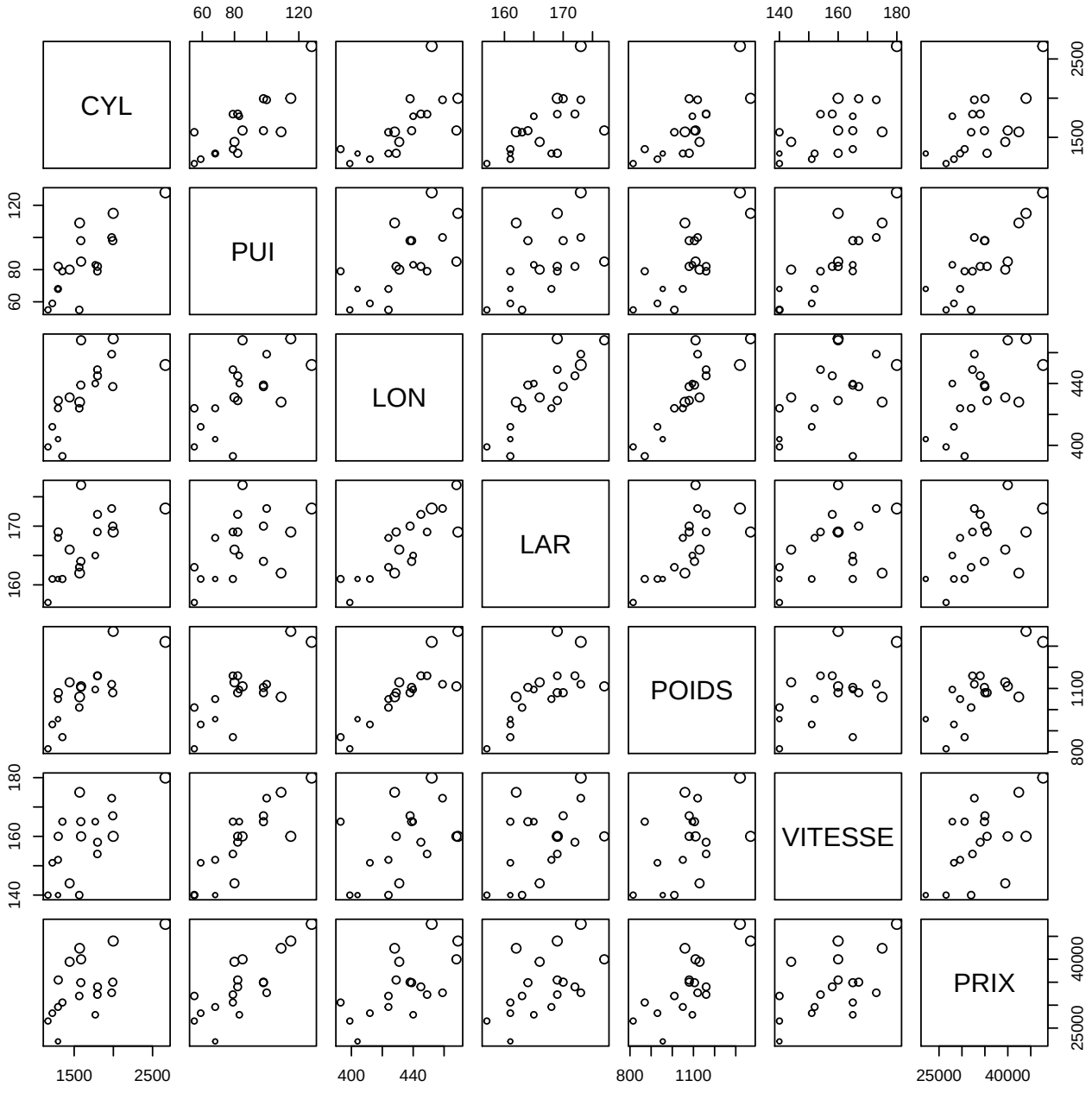
```
NROW(A)
```

```
## [1] 18
```

Analyse exploratoire des données

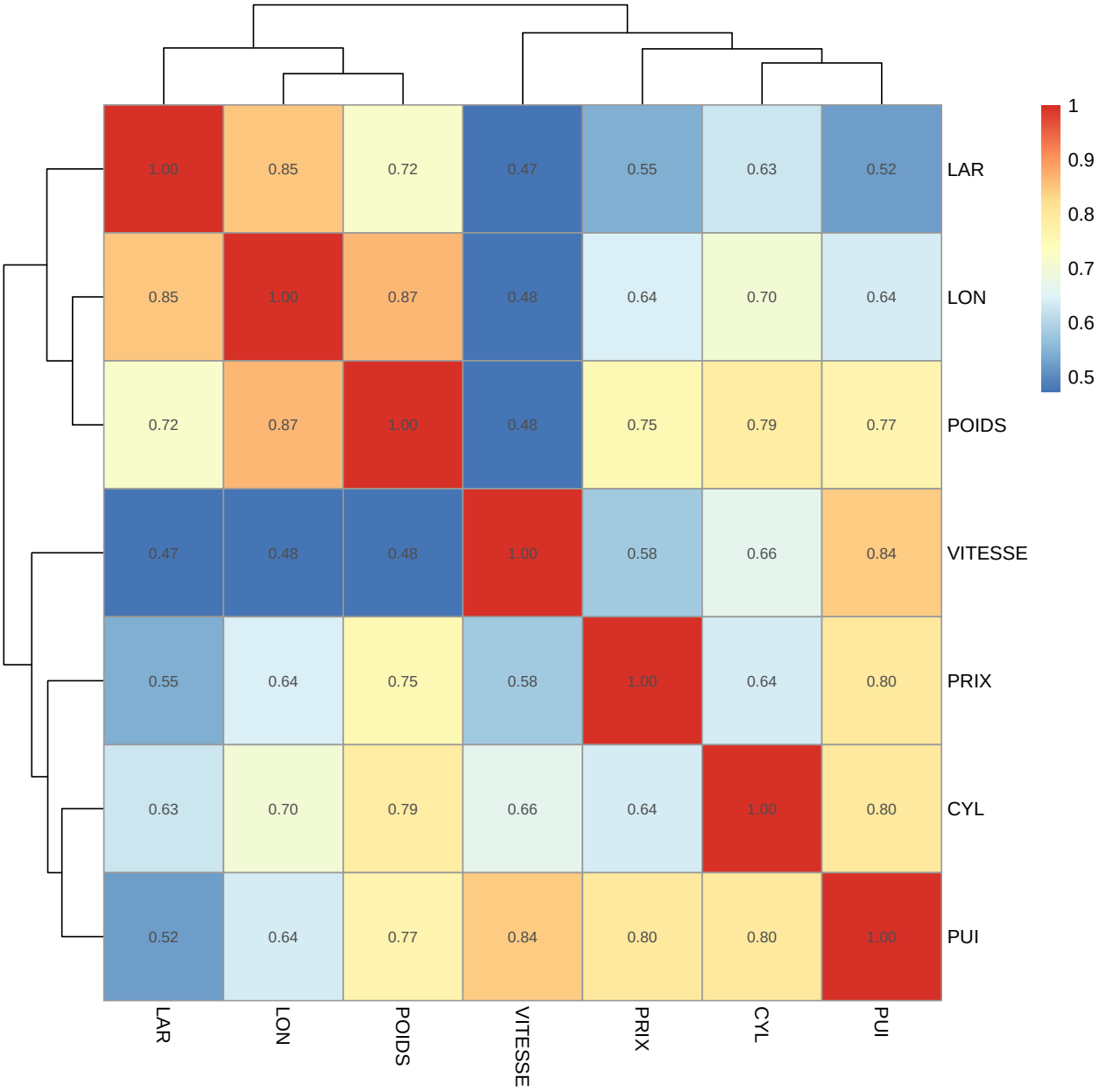
Toute bonne modélisation doit être précédée d'une étape d'analyse exploratoire des données. L'objectif de cette analyse exploratoire est de matérialiser au travers de figures de mérite et d'indicateurs le contenu des données. La figure suivante renvoie l'ensemble des graphes bivariés, la taille des points reflétant la valeur de la variable à expliquer.

```
pairs(A, cex = A$PRIX/median(A$PRIX))
```



La figure suivante permet de visualiser la structure de corrélation entre variables.

```
pheatmap(cor(A), display_numbers = TRUE)
```



On constate de fortes multicolinéarités entre variables (CYL, PUI et VITESSE) d'une part et (LAR, LON, POIDS) d'autre part. Cette structure de corrélation entre variables peut également être exhibée par l'analyse en composante principale (ACP). La commande suivante permet de réaliser une ACP

```
res.pca = princomp(A, cor = TRUE)
summary(res.pca)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4
## Standard deviation     2.2516392 0.9312795 0.67398876 0.56826510
## Proportion of Variance 0.7242684 0.1238973 0.06489441 0.04613217
## Cumulative Proportion  0.7242684 0.8481658 0.91306016 0.95919234
```

```
##                            Comp.5     Comp.6      Comp.7
## Standard deviation     0.40365770 0.28547690 0.203019773
## Proportion of Variance 0.02327708 0.01164244 0.005888147
## Cumulative Proportion  0.98246942 0.99411185 1.000000000
```

La variance de la composante s'obtient également par la commande suivante

```
apply(res.pca$scores, 2, function(x) sd(x)*sqrt((NROW(A)-1)/(NROW(A))))
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 2.2516392 0.9312795 0.6739888 0.5682651 0.4036577 0.2854769 0.2030198
```

La variance cumulée s'obtient comme suit:

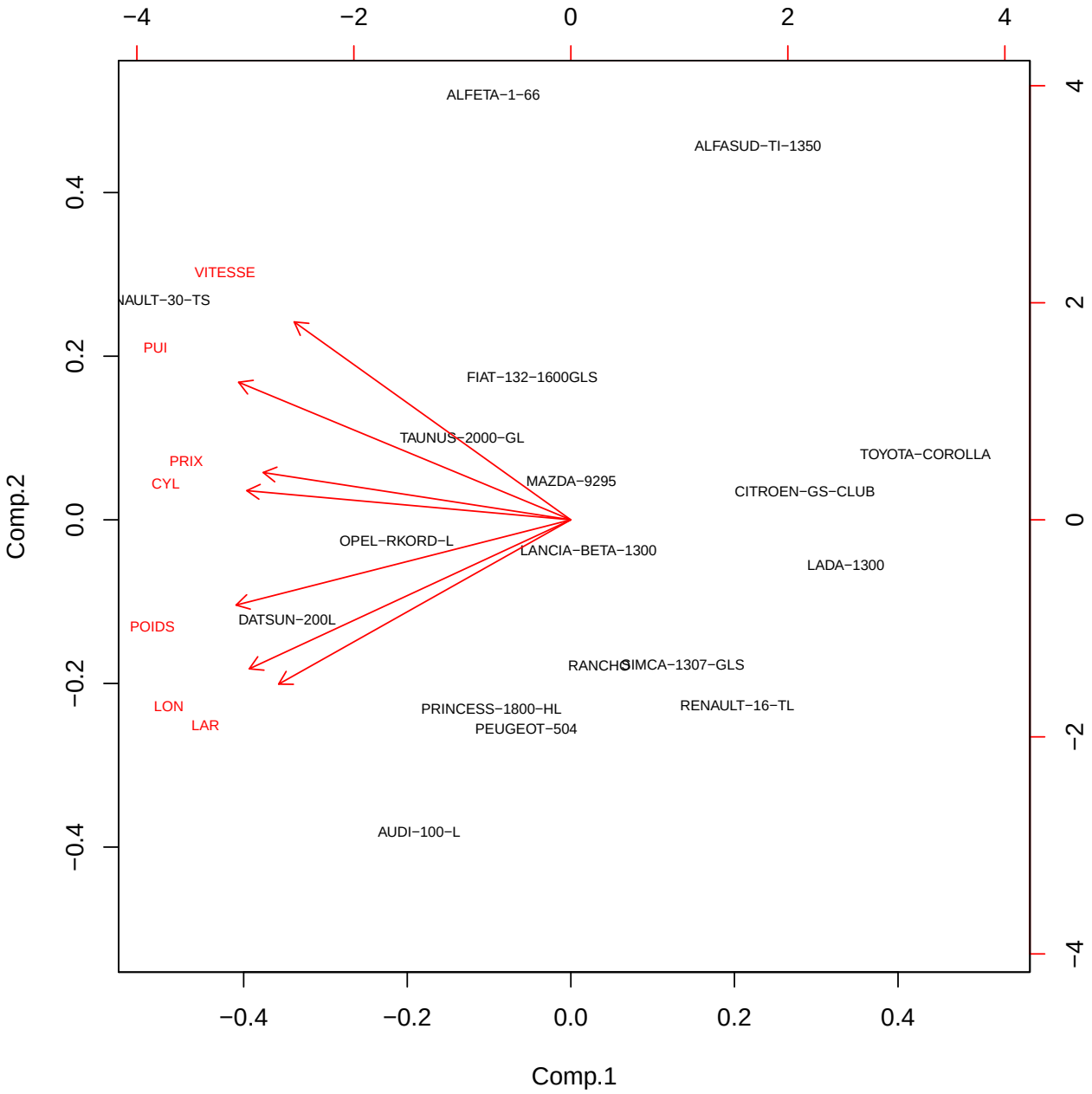
```
variance = apply(res.pca$scores, 2, function(x) var(x)*(NROW(A)-1)/(NROW(A)))
cumsum(variance)/sum(variance)
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 0.7242684 0.8481658 0.9130602 0.9591923 0.9824694 0.9941119 1.0000000
```

On constate que les deux premières composantes capturent près de 85% de l'information présente dans les données. On en déduit qu'une représentation des données sur le premier plan principal fournit une bonne approximation des relations entre individus.

Le biplot associé est représenté ci-dessous

```
biplot(res.pca, cex = .6)
```



Calcul manuel des coefficients de régression et via la fonction lm()

On cherche maintenant à construire un modèle prédictif du prix d'une automobile en fonction de ses grandeurs caractéristiques "CYL", "PUI", "LON", "LAR", "POIDS", "VITESSE". La régression multiple est la méthode de choix pour construire ce type de modèle.

```
y = A$PRIX
X = cbind(1, as.matrix(A[, -7]))
colnames(X) = c("Intercept", "CYL", "PUI", "LON", "LAR", "POIDS", "VITESSE")
beta_hat = solve(t(X)%*%X)%*%t(X)%*%y
beta_hat
```

```
##                   [,1]
## Intercept -8234.624688
## CYL          -3.505583
## PUI         282.174039
## LON         -15.126858
## LAR         208.831802
## POIDS        12.574568
## VITESSE    -111.039281
```

```
res.lm = lm(PRIX ~ ., data = A)
summary(res.lm)
```

```
##
## Call:
## lm(formula = PRIX ~ ., data = A)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8290.9 -1721.2  -167.4  2912.2  5420.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8234.625  42720.145  -0.193    0.851
## CYL            -3.506      5.551  -0.632    0.541
## PUI           282.174    174.890   1.613    0.135
## LON           -15.127    129.753  -0.117    0.909
## LAR           208.832    412.064   0.507    0.622
## POIDS          12.575     24.623   0.511    0.620
## VITESSE      -111.039    222.266  -0.500    0.627
##
## Residual standard error: 4406 on 11 degrees of freedom
## Multiple R-squared:  0.7091, Adjusted R-squared:  0.5504
## F-statistic: 4.468 on 6 and 11 DF,  p-value: 0.01562
```

On constate que les deux approches conduisent aux mêmes coefficients de régression.

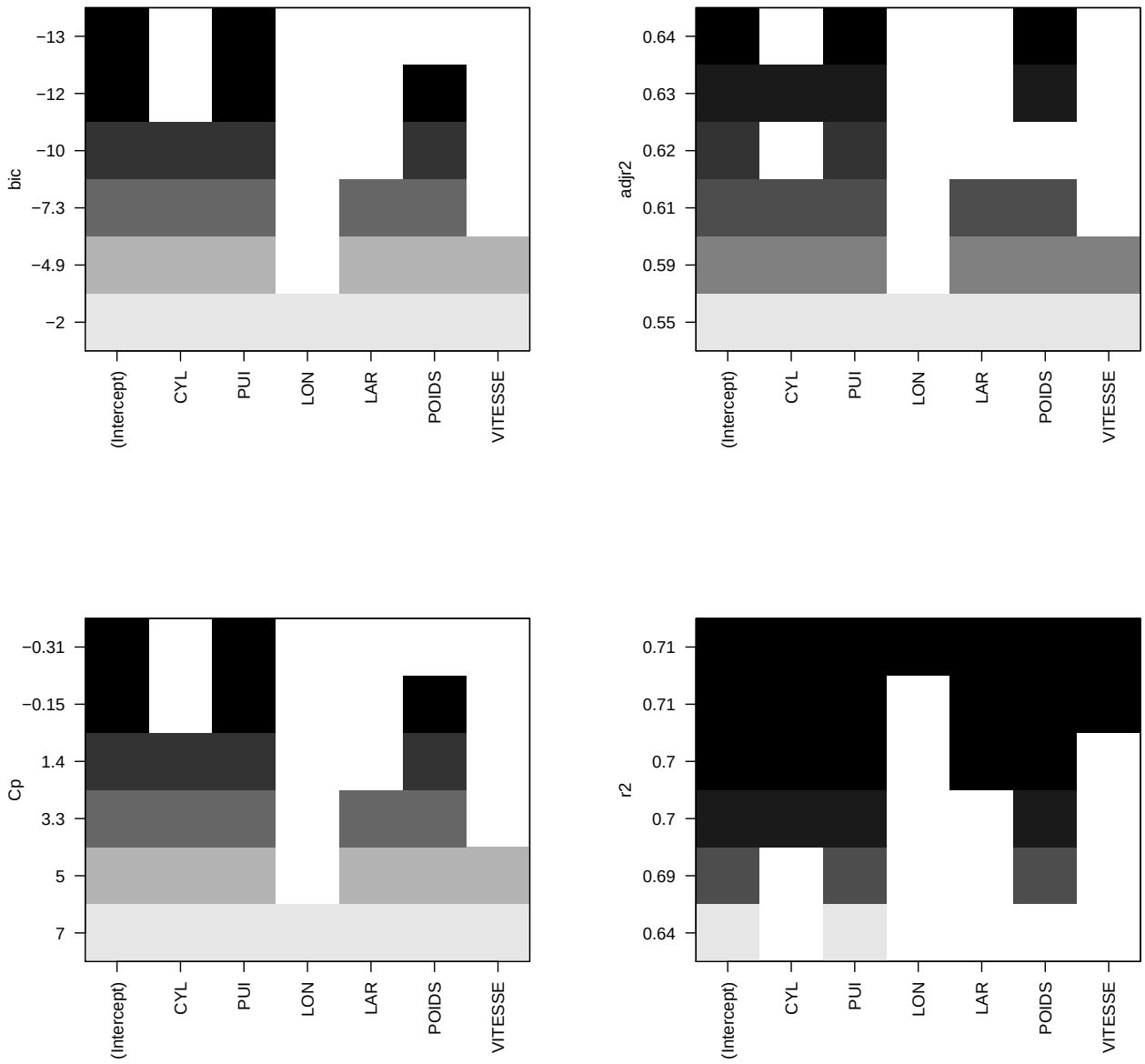
Il faut en permanence conserver un esprit critique sur les modèles générés : L'examen des coefficients de régression et leur niveau de significativité nous conduit à rejeter le modèle. En effet, contrairement à ce que nous laissait conclure l'analyse exploratoire des données, aucune des variables n'est significative. De plus, les signes des coefficients de régression associés à CYL, LON et VITESSE ne sont pas en cohérence avec l'intuition. D'où peut provenir le problème ?

Comme discuté ci-dessous, la matrice de corrélation montre une forte corrélation entre variables. Or, nous savons que $\text{var}(\hat{\boldsymbol{\beta}}) = \sigma^2 \left(\mathbf{X}^t\mathbf{X}\right)^{-1}$, et donc la variance de l'estimateur des moindres carrées peut exploser en présence de fortes multicolinéarités entre variables. Il faut sans doute, parmi les paquets de variables corrélées (CYL, PUI et VITESSE) et (LAR, LON, POIDS), sélectionner un représentant de chaque. Pour ce faire, compte tenu du faible nombre de variables nous proposons d'utiliser une approche exhaustive. La fonction `regsubsets` du package `leaps` permet ce type d'analyse.

```
library(leaps)
exh.search = regsubsets(PRIX ~ ., data = A, method = "exhaustive")
```

Pour pouvoir utiliser les résultats de cette proocédure, le graphique est l'outil le plus approprié. La fonction `regsubsets` propose 4 critères de choix : Le BIC, le Cp, le R^2_{adj} et le R^2. La Figure ci-dessous reporte les résultats associés à ces 4 critères.

```
layout(matrix(1:4, 2, 2))
plot(exh.search, scale = "bic")
plot(exh.search, scale = "Cp")
plot(exh.search, scale = "adjr2")
plot(exh.search, scale = "r2")
```



Le modèle sélectionné dépend du critère considéré. Par exemple, si on considère un critère de type R^2-ajusté, les variables retenues sont PUI et POIDS. Le modèle final associé est reporté ci-dessous.

```
res.final = lm(PRIX~PUI+POIDS, data = A)
summary(res.final)
```

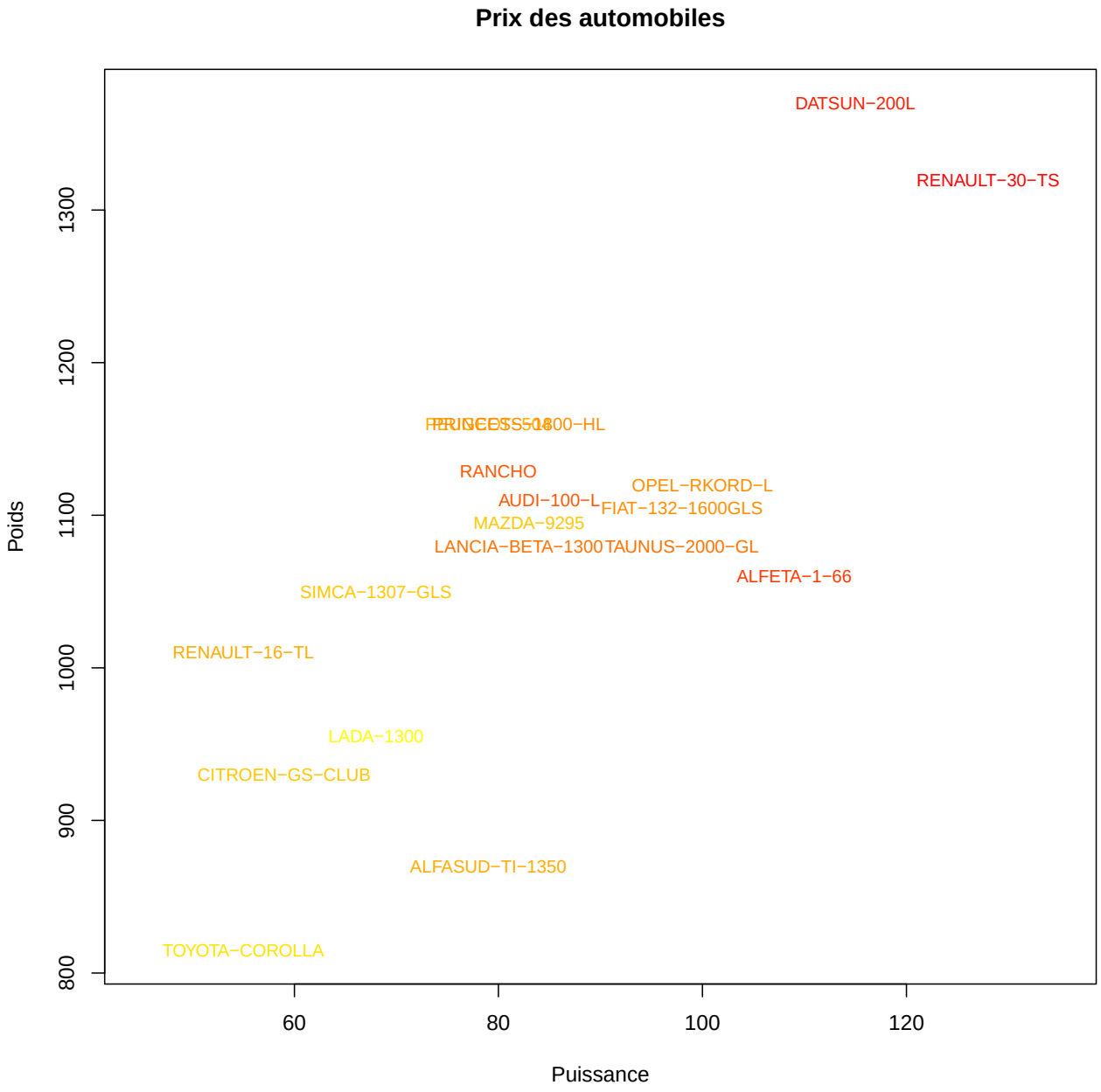
```
##
## Call:
```

```
## lm(formula = PRIX ~ PUI + POIDS, data = A)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7149.3 -1875.6   300.1  2012.2  5264.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1784.60    8031.21   0.222   0.8271
## PUI           173.02      72.42   2.389   0.0305 *
## POIDS          16.44      10.77   1.526   0.1479
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3916 on 15 degrees of freedom
## Multiple R-squared:  0.6866, Adjusted R-squared:  0.6448
## F-statistic: 16.43 on 2 and 15 DF,  p-value: 0.0001663
```

La figure ci-dessous présente le graphe bivarié de la puissance de la voiture en fonction du poids. La code couleur reflétant le prix de la voiture.

```
#Creation d'une fonction générant palette de couleur continue
yrPal <- colorRampPalette(c('yellow','red'))

# Cette commande permet de créer un vecteur de couleur qui dépend des valeurs
# prise par PRIX.
Col <- yrPal(10)[as.numeric(cut(A$PRIX,breaks = 10))]
plot(A$PUI, A$POIDS , col = "white", xlim = c(45, 135),
     main  = "Prix des automobiles",
     xlab = "Puissance", ylab = "Poids")
text(A$PUI, A$POIDS, rownames(A), col = Col, cex = .8)
```



La régression logistique

La régression logistique est une méthode statistique adaptée à l'étude de la liaison entre une variable qualitative Y et un ensemble de p variables explicatives $X_1, X_2, \ldots, X_p$ quantitatives ou qualitatives. L'exemple compagnon de cette séance, **prévision de la faillite d'entreprises**, servira à illustrer cette méthode.

Présentation des données et notations

On se propose de construire un modèle de prévision de la faillite d'une entreprise à partir de données financière récoltées par R.A. Johnson et D.W. Wichern en 1982. Ces données financières annuelles ont été recueillies sur 21 entreprises approximativement deux ans avant leur faillite, et, à peu près à la même époque, sur 25

sociétés financièrement solides. Les données réunissent, pour chaque entreprise, deux ratios financiers et leur situation deux ans plus tard :

variable	Signification
X_1	Flux de trésorerie / Dette totale
X_2	Actif à court terme / Dette à court terme
Y	Faillite (1), non faillite (0)

Chargement des données

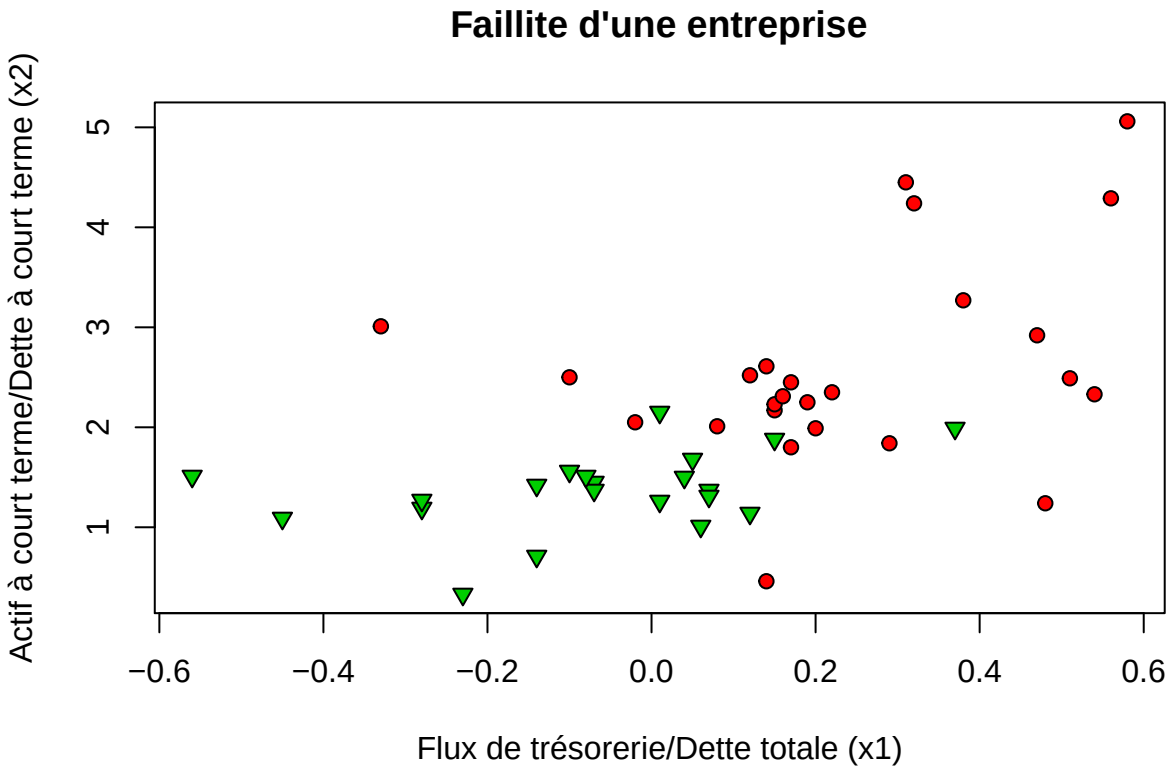
```
A = read.table("faillite.txt",
               header = TRUE)
X = as.matrix(A[, 2:3])
y = A[, 4]
head(cbind(X, y))
```

```
##         x1   x2 y
## [1,] -0.45 1.09 1
## [2,] -0.56 1.51 1
## [3,]  0.06 1.01 1
## [4,] -0.07 1.45 1
## [5,] -0.10 1.56 1
## [6,] -0.14 0.71 1
```

Visualisation de données

La figure suivante renvoie le graphe bivarié ($\mathbf{x}_1$, $\mathbf{x}_2$), la couleur/forme des points reflétant la valeur de la variable à expliquer.

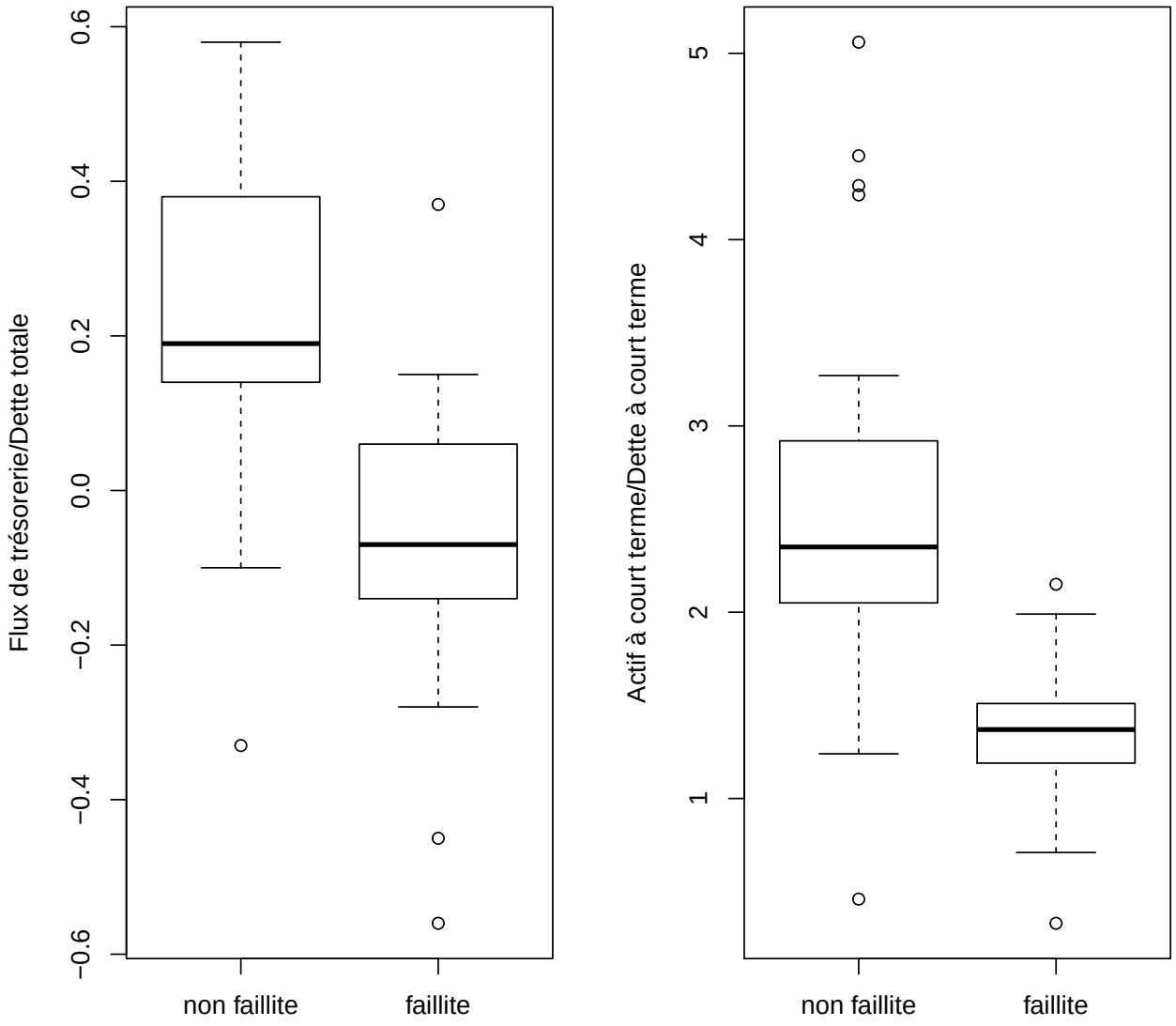
```
plot(X,
     bg = c("red", "green3")[as.factor(y)],
     pch = c(21, 25)[as.factor(y)],
     main = "Faillite d'une entreprise",
     xlab = "Flux de trésorerie/Dette totale (x1)",
     ylab = "Actif à court terme/Dette à court terme (x2)"
)
```



Il semble que ces deux variables soient porteuses d'information discriminante.

On s'intéresse maintenant aux deux variables séparément. Les boîtes à moustaches des deux ratios financiers selon le critère de faillite sont présentés sur la figure suivante. La boîte à moustache permet de visualiser de manière très compacte la dispersion des données. La boîte centrale est construite à partir du premier et du troisième quartile et est partagée par la médiane. Les "moustaches" vont du premier quartile au minimum et du troisième quartile au maximum. Par convention, les moustaches ne doivent pas dépasser une fois et demi la distance interquartile. Si les points extrêmes sont trop loin des quartiles, ils apparaîtront comme isolés (outliers) sur le graphique.

```
layout(matrix(t(1:2), 1, 2))
boxplot(X[, 1]~factor(y,
                      labels = c("non faillite", "faillite")),
                      ylab = "Flux de trésorerie/Dette totale")
boxplot(X[, 2]~factor(y,
                      labels = c("non faillite", "faillite")),
                      ylab = "Actif à court terme/Dette à court terme")
```



Test de Student

Pour chaque variable prise séparément, il est tout à fait possible de réaliser un test d'égalité des moyennes calculées sur chaque classe d'entreprises. La commande suivante permet de réaliser ces test

```
ttest_x1 = t.test(X[, 1]~y)
ttest_x1
```

```
##
##  Welch Two Sample t-test
##
## data:  X[, 1] by y
## t = 4.8206, df = 43.075, p-value = 1.812e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1769717 0.4315235
## sample estimates:
## mean in group 0 mean in group 1
```

```
##      0.23520000     -0.06904762
```

```
ttest_x2 = t.test(X[, 2]~y)
ttest_x2
```

```
##
##  Welch Two Sample t-test
##
## data:  X[, 2] by y
## t = 5.5038, df = 32.448, p-value = 4.392e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.7730984 1.6807683
## sample estimates:
## mean in group 0 mean in group 1
##        2.593600        1.366667
```

Comme on pouvait s'y attendre, les deux tests de Student conduisent à rejeter l'hypothèse d'égalité des moyennes (p.value associée aux deux tests $< .05$). Ceci nous permet de conclure que ces deux ratios financiers se comportent différemment dans des situations de faillite ou de non-faillite.

L'algorithme de Newton-Raphson pour la régression logistique

On cherche maintenant à construire un modèle prédicition de la variable qualitative faillite/non-faillite à partir de ces deux ratios financiers. Pour ce faire, nous nous proposons d'utiliser la régression logistique. Les paramètres du modèle logistique sont estimés par maximum de vraisemblance. L'algorithme utilisé pour trouver l'estimateur du maximum de vraisemblance est l'algorithme de Newton-Raphson.

Posons $\pi_i = P(Y = 1|X = \boldsymbol{x}_i)$ et $\boldsymbol{\pi}$ le vecteur de probabilités tel que le ième élément égal π_i. Notons $\mathbf{X}$ la matrice formée d'une première colonne de coordonnées constantes égales à 1 et des 2 colonnes correspondant aux variables $\mathbf{x}_1$, $\mathbf{x}_2$ observées sur les n individus. Notons enfin $\mathbf{V}$ la matrice diagonale formée des $\pi_i(1 - \pi_i)$. l'étape courante de l'algorithme de Newton-Raphson peut s'écrire comme suit :

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} + \left(\mathbf{X}^t\mathbf{V}\mathbf{X}\right)^{-1}\mathbf{X}^t(\mathbf{y} - \boldsymbol{\pi}) \tag{1}$$

Le code ci-dessous implémente l'algorithme de Newton-Raphson pour la régression logistique

```
my_lr = function(X, y, tolerance = 1e-6, max.iter=200){
  X = cbind(1, X)
  beta_s = rep(0, NCOL(X))
  pi = runif(NROW(X), 0, 1)
  V = diag(pi*(1-pi))
  iter = 1

  made.changes = TRUE

  while (made.changes & (iter < max.iter))
  {
    iter = iter + 1
    made.changes <- FALSE
    beta_s_plus_1 = beta_s + solve(t(X)%*%V%*%X)%*%t(X)%*%(y-pi)
```

```
    pi = drop(1/(1+exp(-X%*%beta_s_plus_1)))
    V = diag(pi*(1-pi))

    relative.change = drop(crossprod(beta_s_plus_1 - beta_s))/drop(crossprod(beta_s))
    made.changes = (relative.change > tolerance)

    beta_s = beta_s_plus_1

    if (iter == 200)
      warning("The Newton-Raphson algorithm did not converge after 200 iterations.")

  }
  if (iter< 200)
    print(paste("The Newton-Raphson algorithm converges after",  iter, "iterations"))

  return(list(beta = beta_s , proba = pi))
}
```

Comparison between my_lr and glm

```
res.mylr = my_lr(X, y)
```

```
## [1] "The Newton-Raphson algorithm converges after 7 iterations"
```

```
res.mylr
```

```
## $beta
##         [,1]
##     5.940161
## x1 -6.556415
## x2 -3.019117
##
## $proba
##  [1] 9.963147e-01 9.936497e-01 9.239641e-01 8.830247e-01 8.682965e-01
##  [6] 9.911152e-01 7.593700e-01 9.057591e-01 7.933145e-01 9.289660e-01
## [11] 9.984248e-01 8.214450e-01 3.505767e-01 9.849795e-01 3.275600e-01
## [16] 7.629709e-02 8.705474e-01 6.318644e-01 8.880036e-01 8.470434e-01
## [21] 9.809542e-01 7.237522e-03 3.423639e-01 1.619867e-03 1.092357e-01
## [26] 1.285648e-04 7.282346e-05 7.908939e-02 4.705589e-01 6.932041e-02
## [31] 3.523238e-01 1.687117e-01 2.784847e-01 9.742559e-01 5.428809e-02
## [36] 1.448061e-01 1.107576e-01 1.799959e-01 9.614268e-03 2.721454e-01
## [41] 2.787595e-01 2.292076e-05 2.011431e-01 2.580530e-03 7.101070e-02
## [46] 1.966421e-06
```

La fonction **glm()** disponible dans le package **stats** implémente le modèle linéaire généralisé. La régression logistique binaire comme cas particulier du modèle linéaire généralisé est donc disponible via cette fonction.

```
res.glm = glm(y~X, family = binomial)
summary(res.glm)
```

```
##
## Call:
## glm(formula = y ~ X, family = binomial)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.70538  -0.48365  -0.00942   0.47678   2.26853
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    5.940      1.985   2.992  0.00277 **
## Xx1           -6.556      2.905  -2.257  0.02402 *
## Xx2           -3.019      1.002  -3.013  0.00259 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 63.421  on 45  degrees of freedom
## Residual deviance: 28.636  on 43  degrees of freedom
## AIC: 34.636
##
## Number of Fisher Scoring iterations: 6
```

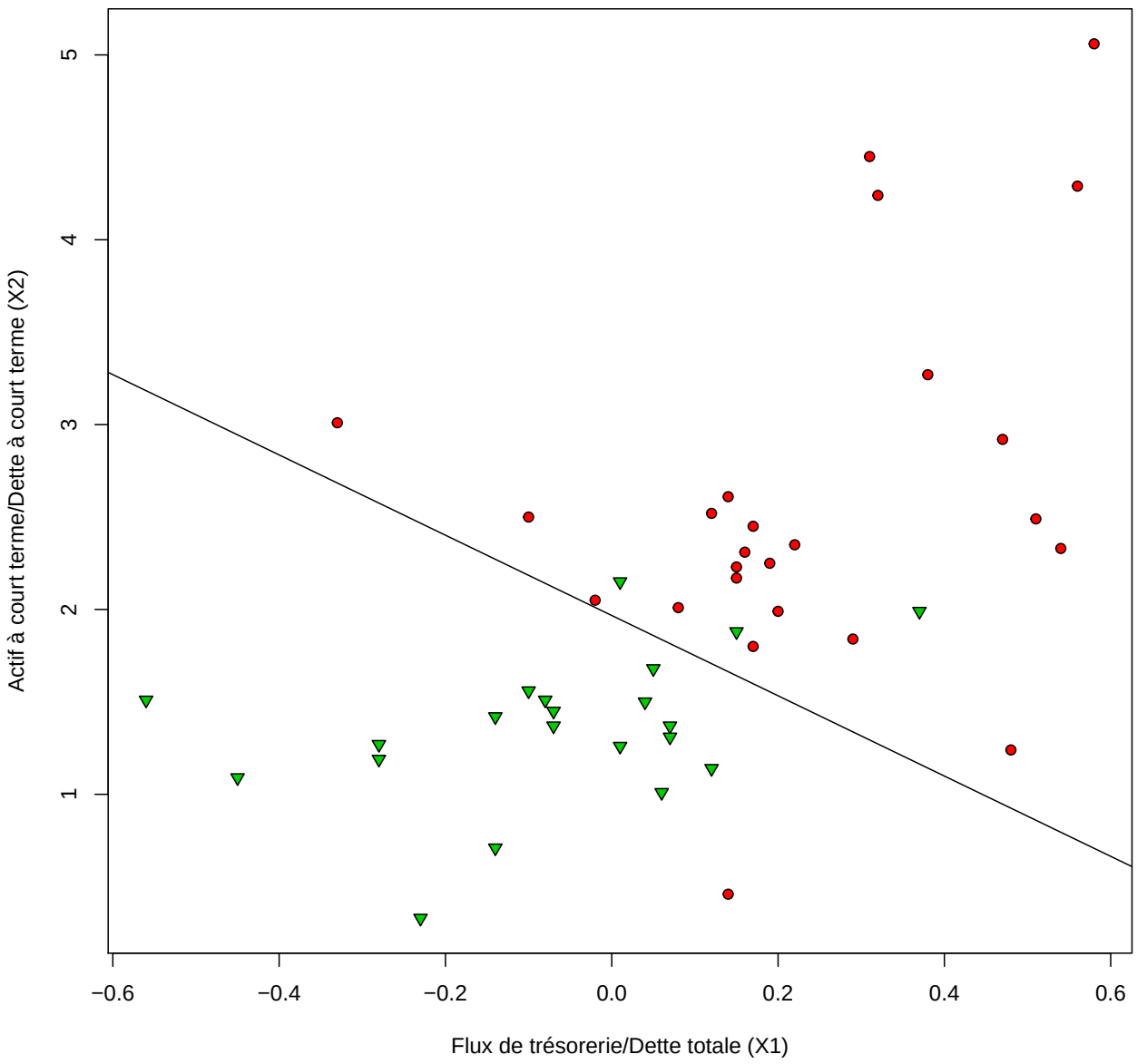
Reste alors à comparer l'estimateur du maximum de vraisemblance issu de notre implémentation à celui estimé par la fonction **glm()** ... verdict....

Visualisation de la frontière de décision

La figure suivante représente les entreprises dans le plan des variables $\mathbf{x}_1$ et $\mathbf{x}_2$ et séparer les deux classes d'entreprise par la droite d'iso-probabilité 0.5 d'équation :

$$5.94 - 6.556X_1 - 3.019X_2 = 0 \iff X_2 = -\frac{6.556}{3.019}X_1 + \frac{5.94}{3.019}$$

```
plot(X,
     bg = c("red", "green3")[as.factor(y)],
     pch = c(21, 25)[as.factor(y)],
     main = "Faillite d'une entreprise",
     xlab = "Flux de trésorerie/Dette totale (X1)",
     ylab = "Actif à court terme/Dette à court terme (X2)"
)
b = res.mylr$beta[1]
a1 = res.mylr$beta[2]
a2 = res.mylr$beta[3]
abline(-b/a2, -a1/a2, col = "black")
```


Faillite d'une entreprise
Actif à court terme/Dette à court terme (X2)
Flux de trésorerie/Dette totale (X1)
1
2
3
4
5
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6