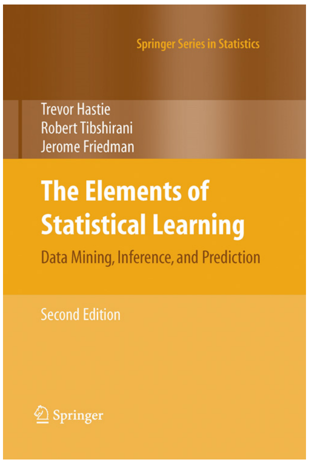
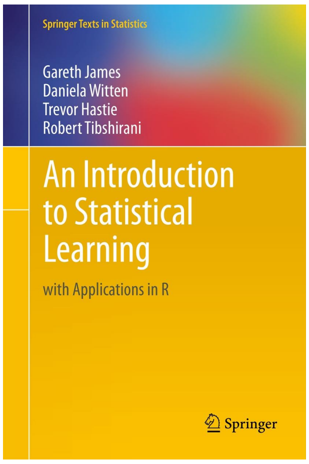
Première partie

Introduction à l'apprentissage statistique

Nous avons ici les deux principales références dont je me suis servi pour rédiger ces notes. Fait rare, ces deux livres sont disponibles gratuitement en pdf sur les sites des auteurs.

Références





1 Introduction générale

L'apprentissage statistique[1] est l'art de programmer les ordinateurs de sorte qu'ils puissent apprendre à partir des données. Voici une définition générale

> L'apprentiſſage automatique eſt la diſcipline donnant aux ordinateurſ la capacité d'apprendre ſanſ qu'ilſ ſoient explicitement programméſ.
> Arthur Samuel, 1959

Et une autre plus technique

> Étant donné une tâche T et une meſure de performance P, on dit qu'un programme informatique apprend à partir d'une expérience E ſi leſ réſultatſ obtenuſ ſur T, meſuréſ par P, ſ'améliorent avec l'expérience E.
> Tom Mitchell, 1997

1.1 Problèmes d'apprentissage statistique

La première application de l'apprentissage statistique ayant véritablement touché un large public, il s'agit du filtre anti-spam. Le filtre anti-spam de nos boites e-mails qui peut apprendre à identifier les e-mails frauduleux à partir d'exemples de pourriels ou « spam » (par exemple, ceux repérés par les utilisateurs) et de messages normaux parfois appelés « ham ». Les exemples de messages utilisés par une méthode d'apprentissage constituent le jeu de données dit **d'entraînement** (*training set* en anglais). Nous donnons ici deux problèmes typiques d'apprentissage

Détection des mails frauduleux :

— Sur 4601 mails, on a pu identifier 1813 spams.

— On a également mesuré sur chacun de ces mails la présence ou absence de 57 mots.

— *Peut-on construire à partir de ces données une méthode de détection automatique de spam ?*

Reconnaissance de chiffres manuscrits : L'apprentissage statistique est présent depuis des décennies dans certaines applications spécialisées telles que la *reconnaissance optique de caractères* ou OCR.

1. Ou apprentissage automatique ou *machine learning* en anglais

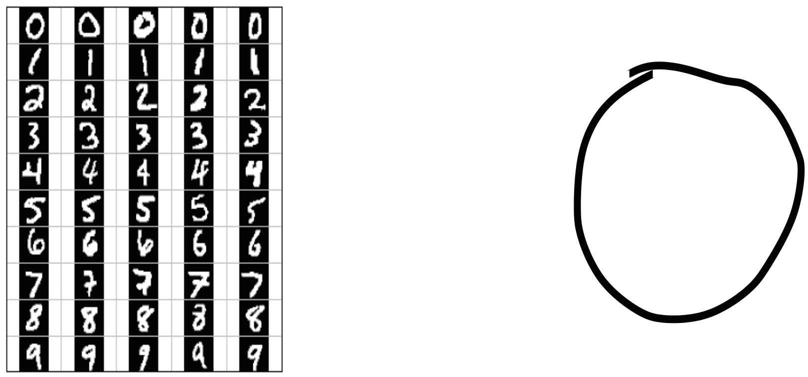


FIGURE 1 – Peut-on mettre en place une méthode automatique de reconnaissance des chiffres manuscrits.

Nous avons ici une petite liste de problèmes d'apprentissage statistique.

1. Identifier les facteurs de risque du cancer de la prostate
2. Prédire si une personne est sujette au crise cardiaque, à partir de mesure clinique, son régime et des données démographiques
3. Classification d'échantillons de tissus dans différents types de cancer, en fonction de données d'expression de gènes
4. Classer des images de tumeurs

1.2 Représentation du problème et vocabulaire

La plupart de ces problèmes peuvent être appréhendés dans un contexte de **régression ou classification supervisée** : on cherche à expliquer une variable Y par d'autres variables dites explicatives $X_1, \ldots, X_p$ souvent représentées par un vecteur une matrice X.

Y	X
Chiffre (OCR)	Image sous forme d'une matrice de pixels
Mot (reconnaissance vocale)	courbe
Spam ou ham	présence/absence d'un ensemble de mots
Type de leucémie	expressions de gênes

— Lorsque la variable à expliquer est quantitative, on parle de **régression**.
— Si elle est qualitative, on parle de **discrimination** ou **classification supervisée**[2].

1.3 Régression

Pour formuler le problème de régression, nous avons besoin des notations suivantes
— Un échantillon i.i.d $(X_1, Y_1), \ldots, (X_n, Y_n)$ à valeurs dans $\mathbb{R}^p \times \mathbb{R}$.

2. Quand il s'agit de clustering, on parle de classification non-supervisée où on cherche à détecter des groupes homogènes d'individus. Le clustering ne fait pas appel à une variable réponse qui est considérée comme le superviseur en classification supervisée.

— Objectif : Prédire ou expliquer la variable Y à partir d'une nouvelle observation X.
— Méthode : construire **une méthode de régression**

$$m : \mathbb{R}^p \mapsto \mathbb{R}.$$

— Critère de performance pour m : l'**erreur de prévision ou erreur quadratique moyenne**

$$\mathbb{E}\left[\left(Y - m(X)\right)^2\right]. \tag{1}$$

Si l'écriture $\mathbb{E}\left[\left(Y - m(X)\right)^2\right]$ semble un peu technique, on peut se contenter de l'interprétation : *la moyenne du carré de l'écart de la variable à expliquer Y et la fonction de régression m dans la population*. On fera souvent appel à sa version empirique dite **erreur d'entrainement** calculée sur le jeu de données d'**entrainement** ou d'**apprentissage** $(x_1, y_1), \dots, (x_n, y_n)$ définie par

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - m(x_i)\right)^2.$$

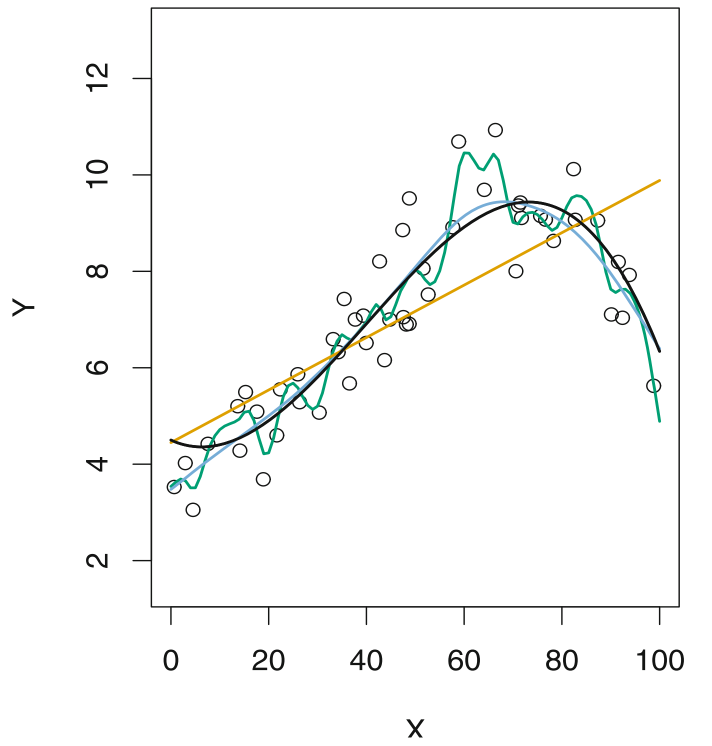


FIGURE 2 – Un exemple jouet en régression : X est en dimension 1 et Y est une variable quantitative. Le nuage de point provient de la courbe en noir (la vérité). Nous avons testé 3 méthodes de régression. Droite orange : une droite de régression (une fonction de la forme $y \approx \beta_0 + \beta_1 x$). En bleu : une fonction de régression en forme de polynôme $y \approx \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_p x^p$. En vert : une fonction obtenue par la méthode dite *spline*.

La légende de la figure 2 évoque des notions qui ne sont pas encore étudiées comme le concept de *spline*[3] et nous n'avons aucune information sur les mécanismes de calcul des différents paramètres qui interviennent dans les formules comme les coefficients β_0 et β_1 dans la droite orange. La figure illustre en partie[4] la richesse de la classe de modèles de régression m qu'on rencontre dans un problème de régression. Les trois modèles *droite de régression, polynôme et spline* illustrés sur la figure 2 sont de complexités croissantes[5]. La question du choix de la meilleure fonction de régression se pose naturellement et cela se fait en minimisant l'erreur quadratique moyenne 1.

Un champion Avec un peu de manipulation technique du concept d'espérance conditionnelle, on peut montrer que la meilleure fonction qui minimise le critère 1 est donnée par

$$m^*(x) = \mathbb{E}\big[Y|X = x\big] \tag{2}$$

appelée fonction de régression. Pour toute autre fonction m, nous avons

$$\mathbb{E}\left[\big(Y - m^*(X)\big)^2\right] \leq \mathbb{E}\left[\big(Y - m(X)\big)^2\right].$$

Problème m^*[6] est inconnu en pratique. Il faut construire une méthode de régression $\widehat{m}_n$ à partir des données $(X_1, Y_1), \ldots, (X_n, Y_n)$, tel que

$$\widehat{m}_n(x) \approx m^*(x).$$

$\widehat{m}_n$ peut être choisie parmi les méthodes de régression illustrées sur la figure 2.

Une solution Soit une valeur x_0[7] pour laquelle on souhaite prédire la valeur de y_0 à l'aide $m^*(x_0)$.

— Nous n'avons aucune observation (x_i, y_i) avec $x_i = x_0$.

— On ne sait pas calculer $\mathbb{E}\big[Y|X = x_0\big]$.

— Intuitivement, on peut faire appel à l'approximation

$$\widehat{m}(x_0) = \text{Moyenne}\Big(y_i | x_i \in \mathcal{N}(x_0)\Big)$$

où $\mathcal{N}(x_0)$ représente un certain voisinage de x_0. On peut s'intéresser aux k couples (x_i, y_i) correspondants aux x_i les plus proches de x_0 dans le jeu de données. Il s'agit de l'estimateur des k plus proches voisins[8].

3. Penser à une fonction définie par morceaux par des polynômes.
4. On citer un grand nombre d'autres modèles ...
5. On peut mesurer la complexité d'un modèle par le nombre de paramètre qu'il implique.
6. Si X est la taille d'un individu et Y son poids. $m^*(x)$ correspond à la moyenne du poids des individus de taille fixée à x dans la population.
7. Prenons une valeur sur l'axe des x de la figure 2.
8. On notera k-nn pour *k-nearest neighbors* en anglais.

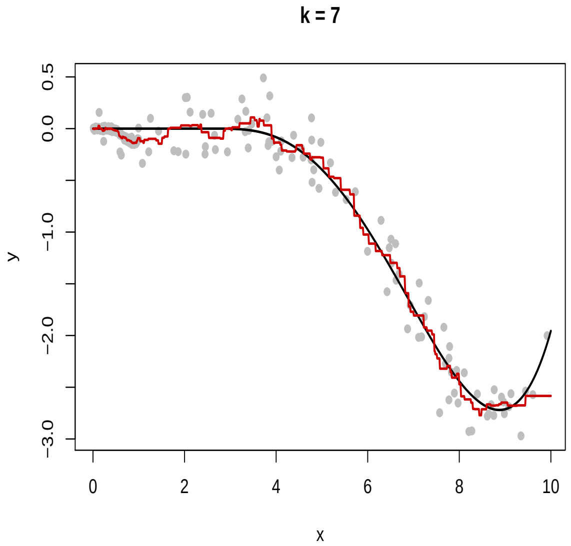


FIGURE 3 – Exemple de régression par k-nn. En noir : la vraie courbe qui correspond au modèle de simulation. En rouge, l'approximation $\widehat{m}_n$ obtenue par la méthode des 7-nn.

1.4 Classification

Commençons par l'introduction des notations, l'objectif de prédiction et le critère de performance en classification.

- Un échantillon i.i.d $(X_1, Y_1), \ldots, (X_n, Y_n)$ à valeurs dans $\mathbb{R}^p \times \{0, 1\}$ [9]. En classification, la variable réponse Y est appelée **label** ou **classe**.
- Objectif : Prédire ou expliquer la variable Y à partir d'une nouvelle observation X.
- Méthode : construire une **règle de classification**

$$g : \mathbb{R}^p \mapsto \{0, 1\}.$$

- Critère de performance pour g : **probabilité d'erreur**

$$\mathbb{P}\big(g(X) \neq Y\big).$$

  Intuitivement, cette mesure d'erreur correspond à la proportion théorique d'individus mal classés par la règle de classification g. La version empirique de l'erreur de classification calculée sur le jeu de données d'entrainement, est définie par

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\Big[y_i \neq g(x_i)\Big].$$

9. Nous allons nous restreindre à la classification binaire ou la variable réponse est à valeurs dans $\{0, 1\}$. Penser à n'importe quelle variable catégorielle à deux modalités.

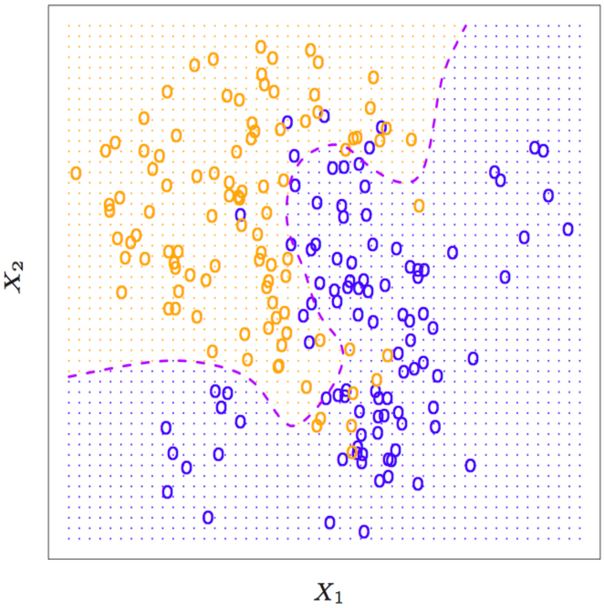


FIGURE 4 – Cette figure illustre un problème de classification simple avec une variable explicative à valeurs dans $\mathbb{R}^2$. Le label y_i de chaque individu correspond à sa couleur. La règle de classification établie est représentée par la courbe appelée frontière de classement. On peut identifier les erreurs de classement commises par le modèle proposé.

Un champion Le règle de classification donnée par :

$$g^*(x) = \begin{cases} 0 & \text{si} \quad \mathbb{P}\big(Y = 0 | X = x\big) \geq \mathbb{P}\big(Y = 1 | X = x\big) \\ 1 & \text{sinon,} \end{cases}$$

appelée **règle de Bayes** [10]. Quelque soit la règle de décision g, nous avons

$$\mathbb{P}\big(g^*(X) \neq Y\big) \leq \mathbb{P}\big(g(X) \neq Y\big).$$

Problème la règle de classification g^* est inconnue en pratique. Il faut construire une règle $\widehat{g}_n$ à partir des données d'entrainement $(x_1, y_1), \ldots, (x_n, y_n)$, telle que

$$\widehat{g}_n(x) \approx g^*(x).$$

Un candidat pour approcher $\widehat{g}_n$ est la régression logistique, couramment utilisée comme modèle de classification binaire.

Solution Soit un point x_0 sur la figure 4 pour lequel nous n'avons pas observé de label y.

— On ne sait pas calculer $\mathbb{P}\big[Y | X = x_0\big]$.

— Intuitivement, on peut faire appel à l'approximation

$$\widehat{g}_n(x_0) = \begin{cases} 1 & \text{si} \quad \frac{1}{|\mathcal{N}(x_0)|} \sum_{i \in \mathcal{N}(x_0)} \mathbb{1}\big(y_i = 1\big) > \frac{1}{|\mathcal{N}(x_0)|} \sum_{i \in \mathcal{N}(x_0)} \mathbb{1}\big(y_i = 0\big) \\ 0 & \text{sinon.} \end{cases}$$

10. C'est l'équivalent de la fonction de régression dans le monde de la classification.

où $\mathcal{N}(x_0)$ représente un certain voisinage de x_0 et $|\mathcal{N}(x_0)|$ est le nombre d'individus dans ce voisinage [11]. On peut s'intéresser aux k individus correspondants aux x_i les plus proches de x_0 dans le jeu de données. Il s'agit de l'estimateur des k plus proches voisins [12] pour la classification. Sur l'exemple illustré sur la figure 4, la règle de classification des k-nn correspond au vote majoritaire. Si les points bleu sont majoritaires parmi les k plus proches voisins alors, on affecte un label bleu à x_0 sinon on lui affecte un label orange.

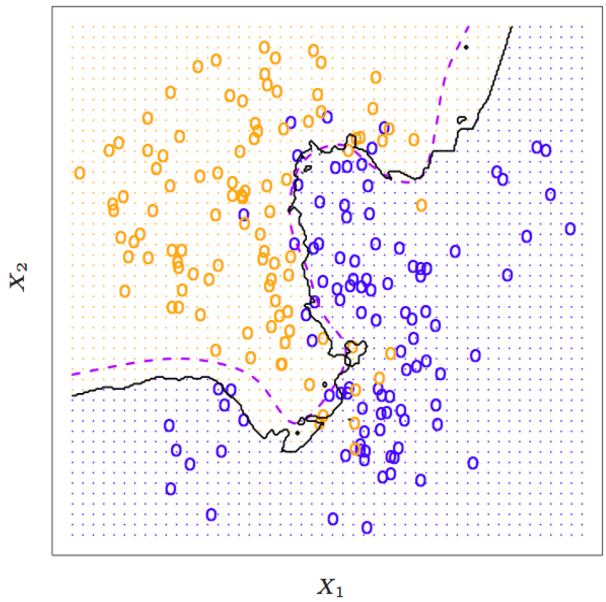


FIGURE 5 – Frontière de classification des 10 plus proches voisins.

2 Complexité de modèle et phénomène de sur-ajustement

La notion de complexité d'un modèle est liée à la famille de modèle à laquelle on s'intéresse pour résoudre un problème d'apprentissage. Sur l'exemple de régression de la figure 2, un modèle d'apprentissage correspond à une des trois courbes représentées. La complexité d'une correspond au nombre de paramètres de la courbe. Dans le cas d'un modèle obtenu par les k plus proches voisins, cette complexité est un peu plus compliquée à conceptualiser. Dans le cadre de cours on admettra que la complexité d'un modèle est inversement proportionnelle aux nombre de voisins k [13].

2.1 Complexité d'un k-nn pour la classification

Commençons par la lecture d'un jeu de données simulé issu du package ElemStatLearn qui permet de retracer une grande partie les jeux de données et les figures disponibles dans le livre [1]. Nous

11. $|\mathcal{N}(x_0)|$ est le cardinal de l'ensemble $\mathcal{N}(x_0)$.
12. On notera k-nn pour *k-nearest neighbors* en anglais.
13. Dans le cas d'une classification, on s'intéressera à la complexité de la frontière de classement. Nous allons illustré ce phénomène sur les exemples simulés que nous allons traiter.

avons besoin d'appeler le package class qui permet d'implémenter un modèle k-nn. Nous allons faire appel au jeu de données simulé mixture.example pour illustrer notre problème.

```
require(ElemStatLearn)
require(class)
data("mixture.example")
names(mixture.example)
x = mixture.example$x
y = mixture.example$y
px1 = mixture.example$px1
px2 = mixture.example$px2
xnew = mixture.example$xnew
plot(x, col=ifelse(y==1,"red", "green"), xlab="x1", ylab="x2")
?knn
```

Le bloc de code suivant permet de mettre en place un modèle de **classification par les** 15 **plus proches voisins**.

```
mod15 = knn(x, xnew, y, k=15, prob=TRUE)
summary(mod15)
prob = attr(mod15, "prob")
# fraction de votants pour la classe majoritaire!
prob = ifelse(mod15=="1", prob, 1-prob)
prob15 = matrix(prob, length(px1), length(px2))
contour(px1, px2, prob15, levels=0.5, labels="", xlab="x1", ylab="x2",
        main="15-nearest neighbour")
points(x, col=ifelse(y==1, "red", "green"))
```

- Expliquer l'objectif de chaque ligne de code du bloc précédent
- Calculer l'erreur de classement des 200 points contenus dans x par le modèle des 15-nn.
- Reprendre le bloc de code précédent pour implémenter le modèle des 1-nn. Tracer la frontière de classement. Calculer l'erreur de classement des 200 points contenus dans x par le modèle des 1-nn. Conclure.

Le bloc de code suivant permet de générer 10000 points du même modèle qui a servi pour générer le jeu de données précédent.

```
require(MASS)
means  = mixture.example$means
set.seed(123)
centers = c(sample(1:10, 5000, replace=TRUE),
            sample(11:20, 5000, replace=TRUE))
means = means[centers, ]
xtest = mvrnorm(10000, c(0,0), 0.2*diag(2)) + means
ytest = c(rep(0, 5000), rep(1, 5000))
K <- c(1,3,5,7,9,11,15,17,23,25,35,45,55,83,101,151)
```

L'objet xtest contient les 10000 points générés. Les 5000 premiers points appartiennent à un groupe et les 5000 restant appartiennent à un autre groupe.

— Pour chaque valeur k dans l'objet `K`, implémenter un modèle k-nn sur le jeu de données de départ constitué des 200 points de départ.
  — Calculer l'erreur de classement du jeu de données contenu dans `x`.
  — Calculer l'erreur de classement du jeu de données contenu dans `xtest`.
— Tracer sur la même figure, les deux erreurs de classement en fonction de la valeur de k.
— Conclure.

Deuxième partie
Régression linéaire

La partie I introduisait les problématiques de régression et de classification ainsi que les notions de complexité de modèle et le phénomène de sur-ajustement. En régression, le modèle linéaire est enseigné dans les parcours de statistique en premier cycle universitaire. La visualisation et la compréhension relativement simple du critère des moindres carrés le rend plus commode pour l'enseignement. L'inférence statistique est simplifié dans le cas d'une régression linéaire. Les sections 3 et 4 expliquent brièvement le modèle de régression linéaire en présence d'une seule variable explicative pour la régression linéaire simple ou plusieurs variables explicatives pour la régression linéaire multiple. On introduit un minimum de notations et de formalisme nécessaire pour décrire la méthode des moindres carrés. Le principe de minimisation des moindres carrés permet d'estimer les paramètres de régression dans les deux cas simple et multiple. L'exemple de régression linéaire simple permet de visualiser et d'assimiler intuitivement le principe des moindres carrés (voir la figure 7). Nous détaillons le calcul de dérivés qui permet de déduire les formules des deux coefficients de régression. Nous illusterons ce même principe via une figure en dimension 3 (voir figure 9). L'écriture matricielle du problème de régression linéaire multiple permet de faciliter le calcul du vecteur des coefficients qui minimise le principe des moindres carrés dans le cas multiple. Comme on peut le constater, l'inférence statistique liée à l'estimation des paramètres de régression ainsi que les diagnostiques de validité d'un modèle de régression linéaire sont volontairement occultés. Pour plus de détails concernant les intervalles de confiance ainsi que les tests statistique dans un modèle de régression linéaire, il est fortement recommandé de lire attentivement les sections 3.1 et 3.2 du livre [2].

3 Régression linéaire simple

Pour introduire la problématique de régression linéaire, nous allons nous servir d'un exemple de jeu de données de cancer de la prostate

3.1 Motivation : données du cancer de la prostate

La tableau 1 résume les variables qui composent le jeu de données du cancer de la prostate [14].

Question : En général, on fait appel à un modèle linéaire pour répondre à la question suivante *Existe-t-il une relation entre **lcavol** et le **lpsa** ?* Notre premier objectif est de déterminer si les données témoignent d'une association entre **lcavol** et le **lpsa**.

14. Stamey, T.A., Kabalin, J.N., McNeal, J.E., Johnstone, I.M., Freiha, F., Redwine,E.A. and Yang, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate : II. radical prostatectomy treated patients, Journal of Urology 141(5), 1076-1083.

lcavol	log(cancer volume)
lweight	log(prostate weight)
age	age
lbph	log(benign prostatic hyperplasia amount)
svi	seminal vesicle invasion
lcp	log(capsular penetration)
gleason	Gleason score
pgg45	percentage Gleason scores 4 or 5
lpsa	log(prostate specific antigen)

TABLE 1 – These data 1 come from a study that examined the correlation between the level of prostate specific antigen and a number of clinical measures in men who were about to receive a radical prostatectomy. It is data frame with 97 rows and 9 columns.

Le modèle linéaire simple permet de répondre à l'ensemble des questions suivantes

- Quelle est la relation entre **lcavol** et le **lpsa** ?
- En supposant qu'il existe une relation entre **lcavol** et le **lpsa**, nous aimerions connaître la force de cette relation. Autrement dit, compte tenu d'un certain **lcavol**, pouvons-nous prédire le **lpsa** ?
- Cette relation est-elle forte ou une prédiction du **lpsa** uniquement sur le **lcavol** serait légèrement mieux qu'une hypothèse aléatoire (relation faible) ?
- Quelles mesures expliquent le **lpsa** ? et comment ?
- La relation est-elle linéaire ? S'il y a approximativement une relation linéaire entre les différentes variables et le **lpsa**, alors la régression linéaire est appropriée. Sinon, il est encore possible de transformer les variables pour que la régression linéaire puisse être utilisée.
- A-t-on des interactions entre les différentes mesures ?

De quoi s'agit-il ?

Une régression (*regression analysis* en anglais) est une méthode pour étudier la relation fonctionnelle entre deux variables. Dans la partie régression linéaire multiple, nous allons étudier la modélisation de la relation entre plusieurs variables (> 2). La régression linéaire simple modélise la relation entre deux variables par une ligne droite, c'est à dire, une variable Y est modélisée comme une fonction linéaire d'une autre variable X.

La figure 6 illustre des modèles de régression simple autour du jeu de données cancer de la prostate.

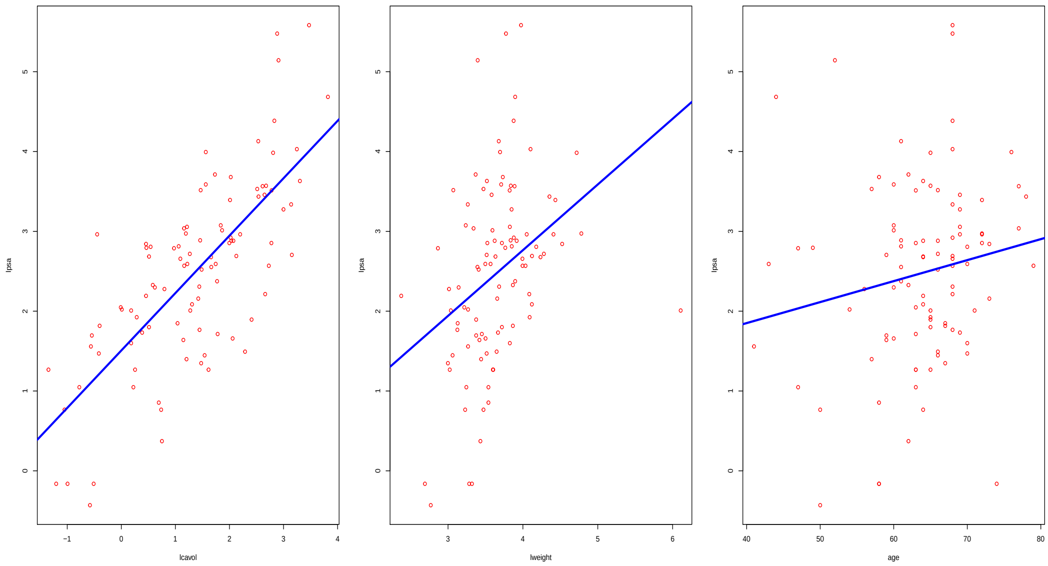


FIGURE 6 – Le nuage de points peut parfois nous donner une idée du type de relation entre les variables (linéaire, quadratique, exponentielle, ...)

3.2 Notations et un peu de formalisme

Rappelons que jeu de données est *souvent* noté

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

— x_i est la $i^{\text{ème}}$ valeur de la variable X dite ***explicative*** ou ***covariable***.
— y_i est la $i^{\text{ème}}$ valeur de la variable Y dite ***réponse*** ou ***variable à expliquer***.

Deux situations pour la collecte des valeurs $x_1, x_2, \ldots, x_n$ associées variable X
— Elles sont observées comme dans le cas des données de cancer de la prostate.
— Quand il s'agît d'une étude, elles peuvent être fixées par cette étude.

La régression linéaire est typiquement utilisée pour modéliser la relation entre deux variables Y et X telle que pour une certaine valeur spécifique de $X = x$, on peut prédire la valeur de Y. De manière formelle, la régression d'une variable aléatoire Y sur une variables aléatoire X est

$$\mathbb{E}\big(Y \mid X = x\big),$$

la valeur moyenne de Y lorsque $X = x$.
Par exemple,
— X le **lcavol**
— Y le **lpsa**.

La régression de Y sur X représente le **lpsa** moyen (l'espérance du **lpsa**) pour un **lcavol**.
La régression de Y sur X est linéaire si

$$\mathbb{E}\big(Y \mid X = x\big) = \beta_0 + \beta_1 x$$

où β_0 et β_1 désignent l'***intercept*** et la ***pente*** de la droite de régression.

Supposons que $Y_1, Y_2, \ldots, Y_n$ sont des réalisations indépendantes de la variable aléatoire Y observées aux points $x_1, x_2, \ldots, x_n$ de la variables aléatoires X. Si la régression de Y sur X est linéaire, alors pour $i = 1, 2, \ldots, n$

$$Y_i = \mathbb{E}\big(Y \mid X = x_i\big) + e_i = \beta_0 + \beta_1 x_i + e_i$$

où les e_i sont des erreurs aléatoires associées aux Y_i telles que $\mathbb{E}\big(e_i \mid X\big) = 0$.

Le terme d'erreur :

— L'ajout du terme d'erreur e_i est du à la nature aléatoire de Y. Il y a certainement une certaine variation dans Y qui ne peut être prédite ou expliquée.

— En d'autres termes, toute variation inexpliquée est appelée erreur aléatoire. Ainsi, le terme d'erreur aléatoire e_i ne dépend pas de x et ne contient aucune information sur Y.

On suppose que,

$$\mathrm{Var}\big(Y \mid X = x\big) = \sigma^2.$$

3.3 Estimation de la pente et de l'intercept dans la population

— Supposons que X est la taille, Y est le poids d'un individu choisi aléatoirement dans une population.

— Dans un modèle de régression linéaire simple, le poids moyen d'un individu pour une taille donnée est une fonction linéaire de la taille.

— En pratique, nous n'avons qu'un échantillon de données (un certain nombre de couples (x_i, y_i)) au lieu de la population entière.

— Les valeurs de β_0 et β_1 sont évidemment inconnues.

— On souhaite utiliser les données pour estimer l'intercept (β_0)et la pente (β_1).

— Cela peut être obtenu via une droite qui ajuste au mieux nos données, c'est à dire des valeurs $\hat{\beta}_0$ et $\hat{\beta}_1$ [15] telles que $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ soit aussi proche que possible de y_i.

— La notation $\hat{y}_i$ est utilisée pour noter la valeur donnée par la droite ajustée pour la distinguer de la valeur observée y_i. On dit que $\hat{y}_i$ est la valeur ***prédite*** ou ***ajustée*** de y_i.

3.4 Les résidus

En pratique, on veut minimiser la différence entre les valeurs de y_i et les valeurs prédites $\hat{y}_i$. Cette différence est appelée le résidu, notée $\hat{e}_i$,

$$\hat{e}_i = y_i - \hat{y}_i.$$

La figure 7 illustre les résidus d'une droite de régression.

15. On note $\hat{\beta}$ au lieu de β pour distinguer la estimée du paramètre de sa valeur théorique inconnue.

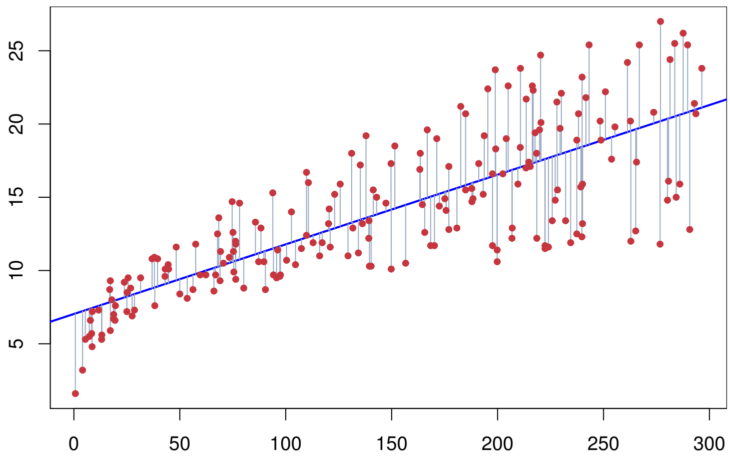


FIGURE 7 – Illustration des résidus d'une droite de régression avec un nuage de points.

3.5 Meilleure approximation au sens des moindres carrés

La méthode couramment utilisée pour choisir les valeurs β_0 et β_1 s'appelle ***méthode des moindres carrés*** . Comme son nom l'indique, β_0 et β_1 sont choisis pour minimiser la somme des résidus notée RSS (*residual sum of squares*),

$$\mathrm{RSS} = \sum_{i=1}^{n} \hat{e}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2.$$

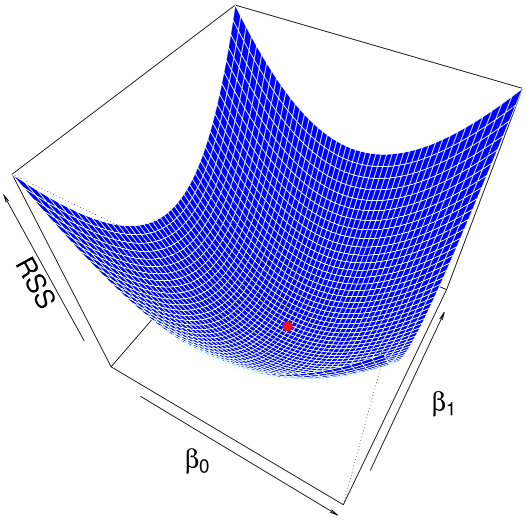
Pour que la somme des moindres carrés RSS soit minimale, il faut que β_0 et β_1 vérifient les équations nous avons les équations

$$\frac{\partial \mathrm{RSS}}{\partial \beta_0} = -2 \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) = 0$$

et

$$\frac{\partial \mathrm{RSS}}{\partial \beta_1} = -2 \sum_{i=1}^{n} x_i (y_i - \beta_0 - \beta_1 x_i) = 0$$

La figure suivante illustre la fonction RSS pour différentes valeurs de β_0 et β_1.



En réarrangeant les équations précédentes, on obtient

$$\sum_{i=1}^{n} y_i = \beta_0 n + \beta_1 \sum_{i=1}^{n} x_i$$

et

$$\sum_{i=1}^{n} x_i y_i = \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2$$

dites ***équations normales***.
La solution du système composé des deux équations précédentes en β_0 et β_1 nous donne les **estimateurs par les moindres carrés** de l'intercept

$$\hat{\beta}_0 = \bar{y}_n - \hat{\beta}_1 \bar{x}_n$$

et la pente

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\text{SXY}}{\text{SXX}}.$$

3.6 Estimation de la variance du terme d'erreur aléatoire

Revenons au modèle de régression linéaire à variance constante décrit précédemment,

$$Y_i = \beta_0 + \beta_1 x_i + e_i \quad (i = 1, 2, \ldots, n)$$

où le l'erreur aléatoire e_i est de moyenne 0 et de variance σ^2. On veut estimer $\sigma^2 = \text{Var}(e)$. Notons que

$$e_i = Y_i - (\beta_0 + \beta_1 x_i) = Y_i - \text{ la droite de régression en } x_i \quad \text{inconnue}$$

Puisque β_0 et β_1 sont inconnues, le mieux qu'on puisse faire, c'est de remplacer ces paramètres inconnus par $\hat\beta_0$ et $\hat\beta_1$ pour obtenir

$$\hat e_i = Y_i - (\hat\beta_0 + \hat\beta_1 x_i) = Y_i - \text{ la droite de régression estimée au point } x_i.$$

Les résidus peuvent être utilisés pour estimer σ^2. En effet, on peut montrer que

$$S^2 = \frac{\text{RSS}}{n-2} = \frac{1}{n-2}\sum_{i=1}^{n} \hat e_i^2$$

est un estimateur sans biais de σ^2. Deux remarques à noter

— $\bar{\hat e} = 0$ (puisque $\sum_{i=1}^n \hat e_i = 0$ car la droite des moindres carrés minimise $\text{RSS} = \sum_{i=1}^n \hat e_i^2$)

— Dénominateur dans S^2 est $n-2$ (nous avons estimé deux paramètres β_0 et β_1)

Le bloc de code suivant illustre l'ajustement d'un modèle de régression linéaire simple sur le jeu de données du cancer de la prostate.

```
## lecture et découverte du jeu de données
 # appel au package qui permet d'accéder au jeu de données
require(lasso2)
 # lecture du jeu de données
data("Prostate")
names(Prostate)
head(Prostate)
dim(Prostate)

## modèle linéaire lpsa en fonction de lcavol
 # ajustement du modèle
lm.fit <- lm(lpsa~lcavol, data=Prostate)
 # résumé du modèle de sortie
summary(lm.fit)
 # les valeurs des coefficients de régression obtenus par moindres carrés
coef(lm.fit)
## visualisation des sorties
attach(Prostate)
 # le nuage de point
plot(lcavol, lpsa, col="red")
 # ajout de la droite de régression au nuage de point
abline(lm.fit, lwd=3, col="blue")
```

4 Régression linéaire multiple

Commençons par le cas particulier d'une régression multiple, connue sous le nom de la régression polynomiale. Dans ce cas, les covariables proviennent de l'observation d'une unique covariable x et ses puissances $(x^2, x^3, \ldots)$. En régression polynomiale, on peut tracer le résultat sur un graphique. La figure 8 illustre un modèle de régression polynomiale de degré 2.

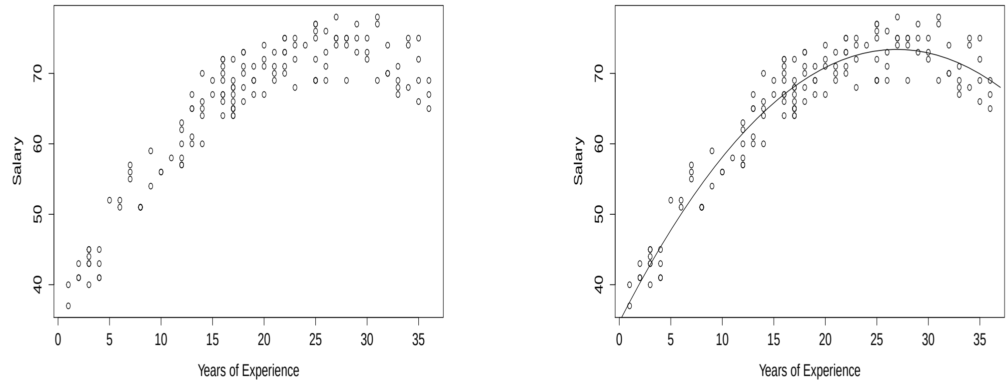


FIGURE 8 – À gauche : Le nuage de points qui suggère une tendance sous forme d'un polynôme. À droite : le modèle de régression quadratique ajusté sur le nuage de points.

La tendance dans le nuage de point suggère un modèle de régression polynomiale

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + e$$

4.1 Modèle linéaire multiple

L'estimation des paramètres du modèle de régression revient à choisir les valeurs des paramètres β_0, β_1 et β_2 qui ont permis d'obtenir le polynôme de degré 2 sur la figure 8. Dans cette partie, on introduit un minimum de formalisme nécessaire à l'explication de la méthode des moindres carrés qui permet d'estimer les paramètres du modèle. Dans le modèle de régression multiple, nous avons l'analogue des notations de la régression linéaire simple

$$\mathbb{E}\big(Y \mid X_1 = x_1, X_2 = x_2, \ldots, X_p = x_p\big) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p.$$

Ainsi,

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + e_i,$$

où e_i est l'erreur aléatoire dans Y_i telle que $\mathbb{E}\big(e_i \mid X\big) = 0$. Ici, la variable réponse Y (à expliquer) est expliquée par p variables explicatives $X_1, X_2, \ldots, X_p$ et la relation entre Y et $X_1, X_2, \ldots, X_p$ est linéaire en $\beta_0, \beta_1, \ldots, \beta_p$. Dans l'exemple précédent, Y est le salaire, $x_1 = x$ est le nombre d'années d'expérience et $x_2 = x^2$. On cherche l'équation du plan 9

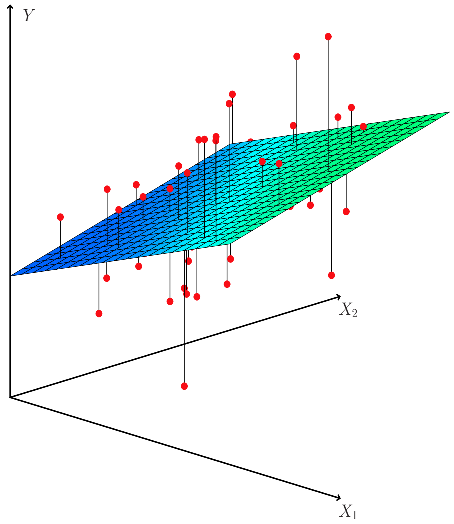


FIGURE 9 – Illustration des moindres carrés en dimension 2.

Les estimateurs des moindres carrés de $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ sont les valeurs de $b_0, b_1, b_2, \ldots, b_p$ qui minimisent la somme des carrés des résidus

$$\mathrm{RSS} = \sum_{i=1}^{n} \hat{e}_i^2 = \sum_{i=1}^{n} \big(y_i - \hat{y}_i\big)^2 = \sum_{i=1}^{n} \big(y_i - b_0 - b_1 x_{i1} - b_2 x_{i2} - \cdots - b_p x_{ip}\big)^2.$$

Pour que le RSS soit minimale en $b_0, b_1, b_2, \ldots, b_p$,

$$\begin{aligned}
\frac{\partial \mathrm{RSS}}{\partial b_0} &= -\,2 \sum_{i=1}^{n} \big(y_i - b_0 - b_1 x_{i1} - b_2 x_{i2} - \cdots - b_p x_{ip}\big) = 0 \\
\frac{\partial \mathrm{RSS}}{\partial b_1} &= -\,2 \sum_{i=1}^{n} x_{i1} \big(y_i - b_0 - b_1 x_{i1} - b_2 x_{i2} - \cdots - b_p x_{ip}\big) = 0 \\
&\;\vdots \\
\frac{\partial \mathrm{RSS}}{\partial b_p} &= -\,2 \sum_{i=1}^{n} x_{ip} \big(y_i - b_0 - b_1 x_{i1} - b_2 x_{i2} - \cdots - b_p x_{ip}\big) = 0
\end{aligned}$$

Pour pouvoir étudier les propriétés des estimateurs des moindres carrés qu'on notera $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$, nous adoptons des notations vectorielles et matricielles. On définit

— Le vecteur $\mathbf{Y}$ de dimension $(n \times 1)$.
— La matrice $\mathbf{X}$ de taille $n \times (p+1)$.
— Le vecteur de paramètres inconnus (coefficients de régression) $\boldsymbol{\beta}$ de dimension $(p+1) \times 1$.
— Le vecteur des erreurs aléatoires $\mathbf{e}$ de dimension $(n \times 1)$.

Tels que

$$\mathbf{Y}=\begin{pmatrix}y_1\\y_2\\\vdots\\y_n\end{pmatrix},\mathbf{X}=\begin{pmatrix}1 & x_{11} & \cdots & x_{1p}\\1 & x_{21} & \cdots & x_{2p}\\\vdots & \vdots & \ddots & \vdots\\1 & x_{n1} & \cdots & x_{np}\end{pmatrix},\boldsymbol{\beta}=\begin{pmatrix}\beta_0\\\beta_1\\\vdots\\\beta_p\end{pmatrix}\text{ et }\mathbf{e}=\begin{pmatrix}e_1\\e_2\\\vdots\\e_n\end{pmatrix}$$

On peut écrire le modèle de régression multiple sous forme d'équation matricielle

$$\mathbf{Y}=\mathbf{X}\boldsymbol{\beta}+\mathbf{e}.$$

De plus, si on note $\mathbf{x}_i'$ la $i^{\text{ème}}$ ligne de la matrice $\mathbf{X}$. Alors

$$\mathbf{x}_i'=\left(1,x_{i1},x_{i2},\dots,x_{ip}\right)$$

est une vecteur ligne $1\times(p+1)$ et on peut écrire

$$\mathbb{E}\left(Y\mid X=x_i\right)=\beta_0+\beta_1x_{i1}+\cdots+\beta_px_{ip}=\mathbf{x}_i'\boldsymbol{\beta}.$$

La somme des carrés des résidus est une fonction de $\boldsymbol{\beta}$ qui s'écrit sous la forme

$$\mathrm{RSS}(\boldsymbol{\beta})=\left(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\right).$$

Rappelons que $\left(\mathbf{AB}\right)'=\mathbf{B}'\mathbf{A}'$, nous avons donc

$$\begin{aligned}\mathrm{RSS}(\boldsymbol{\beta})&=\mathbf{Y}'\mathbf{Y}+\left(\mathbf{X}\boldsymbol{\beta}\right)'\mathbf{X}\boldsymbol{\beta}-\mathbf{Y}'\mathbf{X}\boldsymbol{\beta}-\left(\mathbf{X}\boldsymbol{\beta}\right)'\mathbf{Y}\\&=\mathbf{Y}'\mathbf{Y}+\boldsymbol{\beta}'\left(\mathbf{X}'\mathbf{X}\right)\boldsymbol{\beta}-2\mathbf{Y}'\mathbf{X}\boldsymbol{\beta}.\end{aligned}$$

On sait que pour $v\in\mathbb{R}^d$, $a\in\mathbb{R}^d$ et une matrice $M\in\mathbb{R}^{d\times d}$, nous avons

$$\frac{\partial(v'a)}{\partial v}=\frac{\partial(a'v)}{\partial v}=a,\text{ et }\frac{\partial(v'Mv)}{\partial v}=(M+M')v.$$

Sachant que $\mathbf{X}'\mathbf{X}$ est symétrique, à l'aide des deux formules de dérivation vectorielle précédentes, on sait montrer que la solution des moindres carrés qui minimise $\mathrm{RSS}(\boldsymbol{\beta})$ est

$$\widehat{\boldsymbol{\beta}}=\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}.$$

Le bloc de code suivant illustre les résultat d'une régression linéaire multiple sur le jeu de données du cancer de la prostate. Le modèle ajusté explique la variable lpsa en fonction de toutes les autres variables du jeu de données.

```
## lecture et découverte du jeu de données
 # appel au package qui permet d'accéder au jeu de données
require(lasso2)
 # lecture du jeu de données
data("Prostate")
## modèle linéaire lpsa en fonction des autres variables
 # ajustement du modèle
lm.fit <- lm(lpsa~., data=Prostate)
 # résumé du modèle de sortie
summary(lm.fit)
 # les valeurs des coefficients de régression obtenus par moindres carrés
coef(lm.fit)
```

4.2 Complexité d'une régression par polynômes

Nous avons illustré le phénomène de surajustement (*overfitting* en anglais) dans le contexte d'une régression par les k-nn. Dans cette partie, nous allons étudier le phénomène dit de *overfitting* dans le contexte de sélection de modèle [16]. Nous avons un jeu de données qui semble issu d'une fonction polynomiale et nous allons choisir le degré du polynôme pour la régression.

```r
# Exemple de données d'apprentissage
# 150 gaussiennes standard
ntr <- 150
x = rnorm(ntr)
# Quadratic Y's
y = 2.5*x^2 - 0.5*x + rnorm(ntr)

# Tracer le nuage de point avec la vraie courbe
plot(x,y,  pch=16, col=8)
curve(2.5*x^2-0.5*x,add=TRUE)

# Ajustons deux modèles basiques
# La constante égale à la moyenne des yi
plot(x,y,  pch=16, col=8)
curve(2.5*x^2-0.5*x,add=TRUE)
y.0 = lm(y ~ 1)
abline(h=y.0$coefficients[1])
# La droite de régression linéaire
d = seq(min(x),max(x),length.out=200) # nous avons besoin d'une grille de x
degree = 1
fm = lm(y ~ poly(x,degree))
assign(paste("y",degree,sep="."), fm)
lines(d, predict(fm,data.frame(x=d)),lty=(degree+1))
```

- Ajuster et ajouter les polynômes allant de 1 à 9
- Calculer et tracer l'erreur d'apprentissage en fonction de la complexté du modèle (ici degré du polynôme)
- Simuler un jeu de données test de taille 150 et l'ajouter à la figure précédente (avec une autre couleur)
- Calculer l'erreur de prédiction sur le jeu de données test en fonction du degré du polynôme et le comparer à l'erreur d'apprentissage

4.3 Pour aller plus loin : validation croisée à 5 folds

- Répliquer 5 fois : simuler un échantillon test et tracer les erreurs de prédition en fonction du degré du polyôme.
- Simuler un jeu de données de taille 500, à l'aide d'une validation croisée à 5 folds, déterminer le degré optimal du polynôme au sens de l'erreur de prédiction.

16. La complexité d'un modèle de régression par polynôme est donnée par le degré du polynôme.

Références

[1] Trevor HASTIE, Robert TIBSHIRANI et Jerome FRIEDMAN. *The elements of statistical learning : data mining, inference and prediction.* 2e éd. Springer, 2009. URL : `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`.

[2] Gareth JAMES et al. *An Introduction to Statistical Learning – with Applications in R.* T. 103. Springer Texts in Statistics. New York : Springer, 2013.