

# Étude de quelques méthodes régularisées pour la régression linéaire

[julien.chiquet@gmail.com](mailto:julien.chiquet@gmail.com)

séminaire professionnel ENSAI - 2018

## 1 Objectifs

Ces séances visent à étudier les propriétés empiriques de quelques méthodes régularisées à partir de données simulées et sur un exemple de données réelles. Un objectif connexe est la mise en place d'une méthodologie de simulations numériques, démarche très utile dans le cadre de travaux de recherche.

- utilisation de la sélection stepwise, de la régression ridge, du Lasso et de ses variantes
- mise au point d'un protocole complet de simulations dans le cadre du modèle linéaire
- acquisition de nouvelles compétences R (**glmnet**, **parallel**, **ggplot2**)
- application à un jeu de données en génomique

*Remarques* : les étudiants peuvent travailler en binôme.

## 2 Mise au point d'un protocole de simulation

La simulation de données est (ou *devrait être*) un passage obligé de tout travail de recherche en statistique – théorique ou appliquée – lors de l'étude d'une nouvelle stratégie d'estimation. En effet, les simulations sont un moyen d'étude des propriétés des méthodes dans un cadre parfaitement *contrôlé*. Afin de préciser les *propriétés d'intérêt*, il est nécessaire de ce donner un cadre d'étude. Nous choisissons ici le problème de la sélection de variables prédictives dans le modèle linéaire.

### 2.1 Cadre d'étude: sélection de variables dans le modèle linéaire

Pour une méthode de sélection de variables, les *propriétés d'intérêt* sont

- la capacité à retrouver les variables pertinentes,
- la capacité à prédire une nouvelle observation,
- la précision de l'estimation.

La notion de *cadre contrôlé* se réfère aux hypothèses qui sont faites lors de la définition du modèle statistique. Un intérêt important des simulations numériques est de se placer dans le "bon cas" (*i.e.* celui prévu par la théorie) afin d'étudier les propriétés de la méthode dans ce cadre: on s'assure que les performances sont

conformes à celles attendues. On peut ensuite se permettre de s'éloigner du cadre bien contrôlé par la théorie afin d'étudier la robustesse de la méthode à des écarts aux hypothèses.

On s'attache ici au modèle linéaire gaussien homoscedastique:

$$y_i = x_i^\top \beta^* + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \quad i = 1, \dots, n \quad (1)$$

soit matriciellement

$$\mathbf{y} = \mathbf{X}\beta^* + \varepsilon,$$

avec  $\mathbf{y} = (y_1, \dots, y_n)^\top$  un vecteur de  $\mathbb{R}^n$ ,  $\mathbf{X}$  une matrice de  $\mathcal{M}_{n,p}(\mathbb{R})$  et  $\beta^* = (\beta_1, \dots, \beta_p)^\top$  un vecteur de  $\mathbb{R}^p$  dont  $p_0$  éléments sont non-nuls. On note  $\mathcal{A}^* = \{j : \beta_j^* \neq 0\}$  l'ensemble de ces éléments. On suppose que les prédictors sont gaussiens multivariés de matrice de variance-covariance  $\Sigma$ .

## 2.2 Contrôle de la difficulté

Lors de l'analyse d'une procédure statistique, il est important de pouvoir contrôler précisément la difficulté du problème pour déterminer le champs d'applicabilité de cette méthode. Ceci permet en particulier de mesurer sa robustesse aux écarts aux hypothèses requit par l'analyse théorique.

Dans le cadre de la sélection de variables pour le modèle linéaire gaussien, divers phénomènes sont associés à la difficulté du problème. Dans le cas des méthodes de sélection de variables comme le LASSO, ces phénomènes sont liés par l'analyse théorique aux paramètres du modèle linéaire qui permettent de déterminer plus ou moins précisément les conditions d'applicabilité de cette méthode. On pense en particulier

- aux phénomène de grande dimension ( $n < p$ ), contrôlé par le rapport  $n/p$ ,
- au niveau de “sparsité” du problème, contrôlé par le cardinal de  $\mathcal{A}^* = p_0^*$ ,
- au niveau de bruit, qu'on peut contrôler par le rapport signal sur bruit ou le coefficient de détermination,
- à la structure de dépendance entre les prédictors, i.e., la covariance  $\Sigma$  du vecteur  $X = (X_1, \dots, X_p)$ .

On propose dans la suite d'intégrer ces divers paramètres à une fonction `rlm` permettant de générer des données issues du modèle (1).

### 2.2.1 V1: modèle linéaire simple

Écrire une fonction `rlm(n,p,p0,sigma)` qui renvoie une liste contenant un vecteur `y`, une matrice `x`, un vecteur `beta` avec  $p_0$  entrées non nulles de magnitude choisie selon une loi uniforme entre 1 et 2 et de signe positif ou négatif. Les prédictors sont tirés selon des distribution gaussiennes univariés indépendantes de variance unitaire.

### 2.2.2 V2: intégration du coefficient de détermination

Plutôt que de fixer la variance du bruit  $\sigma^2$ , on préfère fixer la puissance du signal par rapport à la puissance du bruit, par exemple en contrôlant le niveau du coefficient de détermination, qui mesure la part de variabilité expliquée par le modèle:

$$R^2 = \frac{SCM}{SCT} = 1 - \frac{SCR}{SCT}.$$

Déterminer dans le cadre du modèle (1) quelle valeur de  $\sigma^2$  choisir pour obtenir des données ayant un coefficient de détermination donné. Modifier la fonction en conséquence: elle prend désormais en argument `rlm(n,p,p0,r2)` et renvoie une liste `list(y,x,beta,sigma)`.

### 2.2.3 V3: structure de dépendance des prédicteurs

Les prédicteurs ont pour l'instant été considérés comme indépendants. On propose de modéliser une forme de dépendance entre prédicteurs à l'aide d'une loi gaussienne multivariée telle que  $X \sim \mathcal{N}(\mathbf{0}, \Sigma)$ . Outre le cas indépendant ( $\Sigma = \mathbf{I}_p$ ), on considèrera 2 autres scénarios:

- une dépendance de type longitudinale:  $\Sigma_{ij} = \rho^{|i-j|}$ .
- une dépendance par bloc: soit une partition en  $K$  groupes, alors

$$\Sigma_{ij} = \begin{cases} 1 & \text{si } i = j, \\ \rho & \text{si } i \text{ et } j \text{ sont dans le même groupe,} \\ 0 & \text{sinon.} \end{cases}$$

Écrire deux fonctions `rlm.long(n, p, p0, r2, rho)` et `rlm.bloc(n, p, K0, r2, rho, K)` adaptées à ces deux scénarios. On remarquera que dans le second cas, la sparsité est défini en terme de nombre de bloc non nuls  $K_0$ . Pour la génération d'un vecteur gaussien multivarié, on utilisera le package **mvtnorm**.

### 2.2.4 V4: ensemble test, ensemble d'apprentissage

Afin d'évaluer les performances des estimateurs, il est indispensable de générer des données de test. Amender les fonctions précédentes de sorte à prendre en argument `n.test` et renvoyer en plus des variables précédentes des ensembles `train` et `test` indiquant les observations. On pourra affecter une valeur par défaut à `n.test` dépendant de `n.train`, par exemple `10*n.train`.

---

On dispose finalement de 3 fonctions `rlm.indep`, `rlm.long` et `rlm.bloc` renvoyant une liste contenant les variables `y,x,beta,Sigma,sigma,train,test` qui serviront à la génération de données pour les simulations.

## 2.3 Implémentation des méthodes régularisées

On se propose d'étudier les procédures suivantes:

- la régression “stepwise” avec critère AIC et BIC,
- la régression “ridge”,
- la régression “lasso”,
- la variante “elastic-net” (ridge + lasso),

On comparera ces procédures dites régularisées aux méthodes de référence suivantes:

- les moindres carrés ordinaires,
- les moindres carrés oracle (où l'on suppose connaître  $\mathcal{A}^*$ ).

Écrire une fonction par estimateur, du type `getStepwiseAIC`, `getStepwiseBIC`, `getLasso`, etc. qui récupère la valeur de  $\hat{\beta}$ . Toutes les méthodes seront implémentées à l'aide de la fonction `glmnet` du package du même nom sauf la régression stepwise (fonction `step` du package **MASS**, ou `regsubset` du package **leaps**).

*Quelques remarques*

- Pour les expériences en grande dimension ( $n < p$ ), l'estimateur des moindres carrés n'est pas défini de manière unique: on utilisera un estimateur avec une faible pénalité ridge pour le régulariser.
- Pour les méthodes régularisées, on choisira le paramètre  $\lambda$  par validation croisée à l'aide des fonctions `cv.glmnet`.

### 2.3.1 Lasso adaptatif

Le Lasso adaptatif est une version modifiée du lasso proposée pour palier notamment au problème de biais. L'idée est de procéder en deux temps. Dans un premier temps, on estime des paramètres  $\hat{\beta}_{\text{init}}$  à l'aide du lasso. Ce premier estimateur va être utilisé comme poids pour une deuxième étape de lasso, en fixant  $w_j = \hat{\beta}_{j,\text{init}}$  tel que:

$$\hat{\beta}_{\lambda}^{\text{ada}} = \arg \min_{\beta \in \mathbb{R}^p} \left( \frac{1}{n} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^p \frac{|\beta_j|}{|w_j|} \right).$$

L'idée est que si  $\hat{\beta}_{j,\text{init}} = 0$  alors  $\hat{\beta}_{j,\text{ada}} = 0$  de telle sorte que la première étape de lasso sert de pré-sélection. De plus, si  $\hat{\beta}_{j,\text{init}}$  est grand, le lasso adaptatif utilisera une pénalisation plus petite, donc un shrinkage plus petit pour le coefficient  $j$ , ce qui est sensé diminuer le biais pour ce coefficient.

Implémentez le lasso adaptatif à l'aide de l'option `penalty.factor` dans `glmnet` qui permet d'introduire des  $\lambda$ s différents pour chaque élément de  $\beta$ . Créer une fonction `getadalasso(X,Y,beta.init)` qui calcule l'estimateur lasso adaptatif à partir d'un premier vecteur  $\hat{\beta}_{\text{init}}$ .

### 2.3.2 Group-Lasso

Une autre modification populaire du Lasso est une version groupée du lasso – ou group-Lasso – qui suppose la connaissance d’une partition *a priori* des prédicteurs notées  $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_K\}$ . La sélection s’opère donc par groupe:

$$\hat{\beta}_{\lambda}^{\text{grp}} = \arg \min_{\beta \in \mathbb{R}^p} \left( \frac{1}{n} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{k=1}^K \|\beta_{\mathcal{G}_k}\| \right).$$

Implémentez une fonction `getgrplasso` à l’aide du package **grplasso**.

### 2.3.3 Autres méthodes

Pour les plus avancés, il existe une variété de méthodes de sélection de variables à la marge des méthodes régularisées. EN particulier

- des pénalités concaves, de moindre biais que la norme  $\ell_1$  du lasso (voir le package **picasso**)
- des méthodes d’ensemble comme le boosting (voir le package **gbm**) améliorant les capacité de prédiction

Les étudiants les plus avancés pourront intégrer ces méthodes de sélection de variables.

## 3 Comparaison des méthodes

### 3.1 Évaluation des performances

On s’intéresse aux performances des méthodes à la fois en terme de capacité prédictive et en terme de sélection de variables. On considère à cet égard les grandeurs suivantes:

- l’erreur quadratique moyenne de  $\hat{\beta}$  (**mse**)
- l’erreur moyenne de prédiction calculée sur l’ensemble test, (**err**)
- la précision du support estimé  $\hat{\mathcal{A}}$  (**acc**,  $(TN + TP)/p$ )
- la sensibilité de  $\hat{\mathcal{A}}$  (**spe**,  $TP/(FN + TP)$ )
- la spécificité de  $\hat{\mathcal{A}}$  (**sen**,  $TN/(TN + FP)$ )

On a noté TP, FP, TN, FN respectivement pour *true positive*, *false positive*, *true negative* et *false negative*.

Écrire une fonction `getPerformance` qui calcule tous ces indices pour un estimateur  $\hat{\beta}$  donné.

## 3.2 Planning de simulations

Créer un script de simulation pour chaque scénario de matrice de covariance des prédicteurs  $\Sigma$ , en commençant par exemple par le cas orthogonal.

Chaque simulation doit renvoyer un `data.frame` de la forme suivante, afin de faciliter le tracé des résultats à l'aide du package **ggplot2**.

```
##      method mse   err acc sen spe n.p  r2 simu
## 11   lasso  0.3 -0.25 0.92 0.9 0.75 0.5 0.75    1
## 12  ridge  1.65 -0.17 0.92 0.9 0.75 0.5 0.75    1
## 13 adalasso 0.51  0.07 0.92 0.9 0.75 0.5 0.75    1
```

Écrire une fonction `one.simu(i)` permettant d'effectuer la simulation numéro  $i$  pour toutes les méthodes et pour toutes les valeurs des paramètres de simulation que vous aurez choisis (commencer doucement...). Cette fonction sera ensuite facilement parallélisable (par exemple avec le package **parallel** ou **pbmccapply**).

### 3.2.1 Interprétations des résultats

- Représentez les boxplots des indicateurs de performance en fonction de  $n/p$  et de la valeur du  $R^2$ . On utilisera le package **ggplot2**.
- Quels sont les effets de la grande dimension sur les performances des estimateurs (en estimation, en sélection) ?
- Explorer les différents scénarios. Y a t-il des méthodes plus adaptées à certains scénarios ? Si oui, pourquoi ?

## 4 Analyse de jeux de données en science du vivant

Cette partie des tutoriaux est à réaliser vous même, sur la base des méthodes que vous aurez utilisées au cours des travaux dirigés encadrées. Il s'agit d'étudier un jeu de données en science du vivant selon une variante des modèles précédemment utilisés. Les étudiants sont libre d'analyser d'autres jeux de données dans un cadre de régression (linéaire, logistique, Poisson, etc), tant que la sélection de variable est en jeu.

### 4.1 Jeu de données "HIV"

#### 4.1.1 Description

Jeu de données de génotypes associés au niveau du virus du VIH dans le sang. 605 individus ont été génotypés pour 300 SNPs.

#### 4.1.2 Format

Lors du chargement des données, deux objets sont créés :

1. X - une matrice 605x300 donnant les génotypes de 605 individus pour 300 SNPs. 2. y - un vecteur de taille 605 donnant le niveau du virus du VIH dans le sang, pour chaque individu.

```
load("../data/HIVdata.rda")
ls(); str(X); str(y)

## [1] "X" "y"

## num [1:605, 1:300] 3 2 2 3 2 2 1 1 3 2 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:605] "030101" "030102" "060101" "060102" ...
## ..$ : chr [1:300] "rs1264550" "rs9261947" "rs1079541" "rs9295871" ...
## num [1:605] 4.56 5.18 5.2 6.18 6.11 4.64 2.91 6 5.9 4.74 ...
```

#### 4.1.3 objectifs :

1. Sélectionner les SNPs qui sont associés au niveau du virus dans le sang.
2. Sélectionner des ensembles de SNPs intégrant une forme de structure de corrélation dans les données.
3. Évaluer les performances prédictives du modèles

#### 4.1.4 référence

Dalmasso, C., Carpentier, W., Meyer, L., Rouzioux, C., Goujard, C., Chaix, M. L., ... & Theodorou, I. (2008). Distinct genetic loci control plasma HIV-RNA and cellular HIV-DNA levels in HIV-1 infection: the ANRS Genome Wide Association 01 study. *PloS one*, 3(12), e3907-e3907.

## 4.2 Jeu de données "colorectal"

### 4.2.1 Description

Jeu de données de niveaux d'expression de gène associés à des tissus tumoraux ou sain dans le cancer du colon. 62 tissus ont été analysés pour 2000 gènes ou assimilés.

### 4.2.2 Format

Lors du chargement des données, trois objets sont créés :

1. X - une matrice 62x2000 donnant les niveaux d'expression (log transformés) relevé dans les tissus du colons de 62 patients.
2. y - un vecteur de taille 62 indiquant le statut du tissus (-1: tumoral, 1: sain).
3. genes.info - une liste de longueur 2000 donnant des informations sur les 2000 gènes considérés.

```
load("../data/colorectal.rda")
ls()
```

```
## [1] "genes.info" "X"          "y"
```

#### 4.2.3 objectifs :

1. Sélectionner les gènes liés au cancer colo-rectal à l'aide d'un modèle gaussien
2. Sélectionner les gènes liés au cancer colo-rectal à l'aide d'un modèle logistique.
3. Prédire le statut d'un tissu.

#### 4.2.4 référence

U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays", *PNAS*, vol. 96, 1999.

### 4.3 Jeu de données 'Bardet'

#### 4.3.1 description

Jeu de données simplifié d'expression de gènes associées au syndrome de Bardet-Biedl. Les échantillons ont été biopsiés à partir de tissus d'oeil de 120 rats.

#### 4.3.2 format

Lors du chargement des données, la liste **bardet** est créé, contenant deux objets:

1. x - une matrice 120 x 100, donnant les expressions associées à 120 rats pour 100 sondes associées à 20 gènes. 5 sondes consécutives correspondent au même gène.
2. y - un vecteur de taille 120 donnant le niveau d'expression du gène TRIM32.

```
load("../data/bardet.rda")
str(bardet)
```

```
## List of 2
## $ x: num [1:120, 1:100] 0.44705 0.46684 0.01498 0 0.00281 ...
## $ y: num [1:120] 8.42 8.36 8.41 8.29 8.27 ...
```

#### 4.3.3 objectifs

1. Sélectionner les sondes les plus prédictives de l'expression de TRIM32.
2. Opérer une sélection de sonde "par groupe" associée à chaque gène, sachant que 5 prédicteurs consécutifs sont des sondes associées au même gène.



### 3. Évaluer les performances prédictives du modèle

#### 4.3.4 référence

T. Scheetz, K. Kim, R. Swiderski, A. Philp, T. Braun, K. Knudtson, A. Dorrance, G. DiBona, J. Huang, T. Casavant, V. Sheffield, E. Stone .Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences of the United States of America*, 2006.

## 4.4 Cookie dough data set

### 4.4.1 Description

This data set contains measurements from quantitative NIR spectroscopy. The example studied arises from an experiment done to test the feasibility of NIR spectroscopy to measure the composition of biscuit dough pieces (formed but unbaked biscuits). Two similar sample sets were made up, with the standard recipe varied to provide a large range for each of the four constituents under investigation: fat, sucrose, dry flour, and water. The calculated percentages of these four ingredients represent the 4 responses. There are 40 samples in the calibration or training set (with sample 23 being an outlier) and a further 32 samples in the separate prediction or validation set (with example 21 considered as an outlier).

An NIR reflectance spectrum is available for each dough piece. The spectral data consist of 700 points measured from 1100 to 2498 nanometers (nm) in steps of 2 nm.

### 4.4.2 Format

A data frame of dimension 72 x 704. The first 700 columns correspond to the NIR reflectance spectrum, the last four columns correspond to the four constituents fat, sucrose, dry flour, and water. The first 40 rows correspond to the calibration data, the last 32 rows correspond to the prediction data.

```
load("../data/cookiesDough.RData")
head(cookie[, 1:5])
```

```
##           1           2           3           4           5
## 1 0.249846 0.249745 0.249441 0.249036 0.248885
## 2 0.256096 0.256096 0.256353 0.256559 0.256868
## 3 0.274770 0.274720 0.274819 0.274770 0.274819
## 4 0.243735 0.243735 0.243782 0.243688 0.243453
## 5 0.243405 0.243261 0.243261 0.243357 0.243070
## 6 0.253504 0.253356 0.253356 0.253504 0.253701
```

### 4.4.3 References

B.G. Osborne, T. Fearn, A.R. Miller, and S. Douglas (1984): Application of Near-Infrared Reflectance Spectroscopy to Compositional Analysis of Biscuits and Biscuit Dough. *Journal of the Science of Food and Agriculture*, 35, pp. 99 - 105.

## 4.5 Nutrimouse

### 4.5.1 description

The nutrimouse dataset contains the expression measure of 120 genes potentially involved in nutritional problems and the concentrations of one hepatic fatty acids for forty mice.

### 4.5.2 format

A data frame with 121 columns and 40 rows. The first 120 numerical variables are gene expression. The last columns contains the contentration (in proportion) of fat lipid.

```
load("../data/mice.rda")
ls()
```

```
## [1] "bardet" "cookie" "mice"
```

### 4.5.3 référence

Martin, P. G. P., Guillou, H., Lasserre, F., Dejean, S., Lan, A., Pascussi, J.-M., San Cristobal, M., Legrand, P., Besse, P. and Pineau, T. (2007). Novel aspects of PPAR-mediated regulation of lipid and xenobiotic metabolism revealed through a multigenomic study. *Hepatology* 54, 767-777.

## 4.6 Ferretin data set

### 4.6.1 description

This data set has been collected at the Australian National Sport Institue, representing the concentration in Ferretin and various covariate for 102 men et 100 women.

### 4.6.2 format

A data frame with 13 columns and 202 rows.

- Sport Sport
- Sex male or female

- Ht Height in cm
- Wt Weight in kg
- LBM Lean body mass
- RCC Red cell count
- WCC White cell count
- Hc Hematocrit
- Hg Hemoglobin
- Ferr Plasma ferritin concentration
- BMI Body mass index =  $\text{weight}/\text{height}^2$
- SSF Sum of skin folds
- XBFat % body fat

```
load("../data/ferritin.RData")
head(ferritin)
```

##	Sex	Sport	RCC	WCC	Hc	Hg	Ferr	BMI	SSF	X.Bfat	LBM	Ht	Wt
## 1	female	BBall	3.96	7.5	37.5	12.3	60	20.56	109.1	19.75	63.32	195.9	78.9
## 2	female	BBall	4.41	8.3	38.2	12.7	68	20.67	102.8	21.30	58.55	189.7	74.4
## 3	female	BBall	4.14	5.0	36.4	11.6	21	21.86	104.6	19.88	55.36	177.8	69.1
## 4	female	BBall	4.11	5.3	37.3	12.6	69	21.88	126.4	23.66	57.18	185.0	74.9
## 5	female	BBall	4.45	6.8	41.5	14.0	29	18.96	80.3	17.64	53.20	184.6	64.6
## 6	female	BBall	4.10	4.4	37.4	12.5	42	21.04	75.2	15.58	53.77	174.0	63.7

## 4.7 University Rank

### 4.7.1 description

Global score for 200 Universities as a function of various predictors. A data frame with the following columns

- World\_Rank : Rang de l'université
- University\_Name: Nom de l'université
- Country: Localisation de l'université
- Teaching\_Rating: Taux de la qualité d'enseignement de l'université, entre 0-100 .
- Inter\_Outlook\_Rating: Taux de la composition internationale de l'université, entre 0-100.
- Research\_Rating: Taux de la qualité de recherche de l'université, entre 0-100.
- Citations\_Rating: Taux de citations des papiers par d'autres universités, entre 0-100.
- Industry\_Income\_Rating: Taux de l'investissement des entreprises dans la recherche de l'université, entre 0-100.
- Total\_Score: Score Final (Variable à expliquer).
- Num\_Students: Nombre total des étudiants.
- Student.Staff\_Ratio: Ratio entre le nombre des étudiants et le nombre des membres académiques.
- X..Inter\_Students: Pourcentage des étudiants étrangers.
- X..Female\_Students: Pourcentage des étudiantes.

— Year: Année académique.

```
load("../data/univ.RData")
head(univ[, 1:3])
```

```
##      World_Rank      University_Name
## 1          2    California Institute of Technology
## 2         75      University of Oxford
## 3         89      Stanford University
## 4        102    University of Cambridge
## 5        111 Massachusetts Institute of Technology
## 6        122      Harvard University
##
##      Country
## 1 United States of America
## 2      United Kingdom
## 3 United States of America
## 4      United Kingdom
## 5 United States of America
## 6 United States of America
```

## 4.8 Parkinson dataset

### 4.8.1 Description

This dataset is composed of a range of biomedical voice measurements from people with early-stage Parkinson's disease recruited to a six-month trial of a telemonitoring device for remote symptom progression monitoring. Columns in the table contain 16 biomedical voice measures. Each row corresponds to one of 5,875 voice recording from these individuals. The main aim of the data is to predict the total UPDRS scores from the 16 voice measures, a Clinical score for the disease.

### 4.8.2 Format

A vector  $y$  with the Clinician's total UPDRS score and a matrix  $x$  with the following columns

- 1-5 Jitterxx - Several measures of variation in fundamental frequency
- 6-11 Shimmerxx - Several measures of variation in amplitude
- 12 NHR - a measures of ratio of noise to tonal components in the voice
- 13 HNR - a measures of ratio of noise to tonal components in the voice
- 14 RPDE - A nonlinear dynamical complexity measure
- 15 DFA - Signal fractal scaling exponent
- 16 PPE - A nonlinear measure of fundamental frequency variation

```
load("../data/parkinson.Rdata")
head(x)[, 1:3]
```

```
##      Jitter... Jitter.Abs. Jitter.RAP
```

```
## [1,] 0.00662 3.380e-05 0.00401
## [2,] 0.00300 1.680e-05 0.00132
## [3,] 0.00481 2.462e-05 0.00205
## [4,] 0.00528 2.657e-05 0.00191
## [5,] 0.00335 2.014e-05 0.00093
## [6,] 0.00353 2.290e-05 0.00119
```

```
head(y)
```

```
## [1] 34.398 34.894 35.389 35.810 36.375 36.870
```

## 4.9 Breast cancer data

### 4.9.1 description

The goal is to predict whether epithelial cells are benign or malignant, based on 9 cytological features assessed on a scale of 1 to 10.

### 4.9.2 format

A vector `y` for the status of the cell (benign, malignant) and a matrix `bc.raw` with the following columns

- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape
- Marginal Adhesion
- Single Epithelial Cell Size
- Bare Nuclei
- Bland Chromatin
- Normal Nucleoli
- Mitoses

```
load("../data/breast.RData")
head(bc.raw)[, 1:3]
```

```
##   Clump.Thickness Uniformity.of.Cell.Size Uniformity.of.Cell.Shape
## 1                5                4                4
## 2                3                1                1
## 3                6                8                8
## 4                4                1                1
## 5                8               10               10
## 6                1                1                1
```