Devoir Maison 2

Partie Théorique

Exercice 1

Par définition, un modèle est identifiable si la fonction $\theta \mapsto \mathbb{P}_{n,\theta}$ est injective.
On va d'abord montrer que ϕ est injective. Soit $(t_1, t_2) \in \mathbb{R}^2$, tels que $\phi(t_1) = \phi(t_2)$:

$$\phi(t_2) - \phi(t_1) = \frac{1}{1+e^{-t_1}} - \frac{1}{1+e^{-t_2}} = 0$$

Donc :

$$e^{-t_2} = e^{-t_1}$$

Comme la fonction exponentielle est injective, $t_1 = t_2$ et donc ϕ l'est aussi.
On va procéder par contradiction. Soient $(\theta_1, \theta_2) \in (\mathbb{R}^p)^2$ donnés, $\theta_1 \neq \theta_2$, avec $\mathbb{P}_{n,\theta_1} = \mathbb{P}_{n,\theta_2}$.
Comme ϕ est injective, $\forall i \in \{1, \dots, n\}$ on doit avoir $\theta_1^T \boldsymbol{x}_i = \theta_2^T \boldsymbol{x}_i$
Donc :

$$\boldsymbol{X}_n \theta_1 = \boldsymbol{X}_n \theta_2$$

$$\boldsymbol{X}_n (\theta_1 - \theta_2) = 0$$

Comme on suppose $\boldsymbol{X}_n$ de rang p, il doit être inversible, ce qui nous donne $\theta_1 = \theta_2$. Le modèle est donc identifiable.

□

Exercice 2

Soit $\boldsymbol{u} \in \mathbb{R}^p$, nous avons que :

$$\boldsymbol{u}^T \boldsymbol{F}_n(\theta) \boldsymbol{u} = \boldsymbol{u}^T \left(\sum_{i=1}^{n} h(\theta^T \boldsymbol{x}_i) \boldsymbol{x}_i \boldsymbol{x}_i^T \right) \boldsymbol{u}$$

$$\boldsymbol{u}^T \boldsymbol{F}_n(\theta) \boldsymbol{u} = \sum_{i=1}^{n} h(\theta^T \boldsymbol{x}_i) \boldsymbol{u}^T \boldsymbol{x}_i \boldsymbol{x}_i^T \boldsymbol{u}$$

$$\boldsymbol{u}^T \boldsymbol{F}_n(\theta) \boldsymbol{u} = \sum_{i=1}^{n} h(\theta^T \boldsymbol{x}_i) \boldsymbol{u}^T \boldsymbol{x}_i (\boldsymbol{u}^T \boldsymbol{x}_i)^T$$

$$\boldsymbol{u}^T \boldsymbol{F}_n(\theta)\boldsymbol{u} = \sum_{i=1}^{n} h(\theta^T \boldsymbol{x}_i)||\boldsymbol{u}^T \boldsymbol{x}_i||^2$$

Comme $h : \mathbb{R} \to [0,1]$, $\boldsymbol{u}^T \boldsymbol{F}_n(\theta)\boldsymbol{u} \geq 0$.
Supposons que $\boldsymbol{u}^T \boldsymbol{F}_n(\theta)\boldsymbol{u} = 0$. Comme $\forall i \in \{1, \dots n\}$, $h(\theta^T \boldsymbol{x}_i)||\boldsymbol{u}^T \boldsymbol{x}_i||^2 \geq 0$, nous avons que $\forall i \in \{1, \dots, n\}$:

$$h(\theta^T \boldsymbol{x}_i)||\boldsymbol{u}^T \boldsymbol{x}_i||^2 = 0$$

On a que $\phi : \mathbf{R} \to (0,1)$ (seulement dans la limite à $\pm\infty$ qu'il atteint ses bornes), donc $\forall i \in \{1, \dots, n\}$:

$$h(\theta^T \boldsymbol{x}_i) = \varphi(h(\theta^T \boldsymbol{x}_i))(1 - \varphi(h(\theta^T \boldsymbol{x}_i))) > 0$$

Donc $\forall i \in \{1, \dots, n\}$:

$$||\boldsymbol{u}^T \boldsymbol{x}_i||^2 = 0$$

Donc :

$$\boldsymbol{X}_n \boldsymbol{u} = 0$$

Or, comme $\boldsymbol{X}_n$ est de rang p, donc inversible :

$$\boxed{\boldsymbol{u} = 0}$$

Ce qui nous permet de conclure que $\boldsymbol{F}_n(\theta)$ est définie positive.

Exercice 3

On a que :

$$h(t) = \varphi(t)(1 - \varphi(t)) = \varphi(t) - \varphi(t)^2$$

Donc :

$$h'(t) = \varphi'(t) - 2\varphi(t)\varphi'(t) = \varphi'(t)(1 - 2\varphi(t))$$

Comme $\varphi(t) = \frac{e^t}{1+e^t} \in (0,1)$, on a que :

$$|1 - 2\varphi(t)| \in (-1, 1)$$

Devoir Maison 2

Aussi, $\varphi'(t) = \frac{e^{-t}}{(1+e^{-t})^2}$ donc :

$$|\varphi'(t)| \in (0,1)$$

Donc :

$$|h'(t)| = |\varphi'(t)||1-2\varphi(t)| < 1$$

Donc, par le theorème des accroissements finis :

$$|h(x)-h(y)| \leq |x-y| \sup_{z\in(x,y)} h'(z) < |x-y|$$

Donc h est 1-Lipschitzienne sur $\mathbb{R}$.

Exercice 4

Nous avons que notre échantillon consiste d'une suite de variables aléatoires de Bernoulli. Nous pouvons donc écrire la vraisemblance comme :

$$\theta \mapsto L_n(\theta) = \prod_{i=1}^{n} \varphi(\theta^T \boldsymbol{x}_i)^{Y_i}(1-\varphi(\theta^T \boldsymbol{x}_i))^{1-Y_i}$$

Sa log-vraisemblance est donc :

$$\theta \mapsto l_n(\theta) = \sum_{i=1}^{n}[Y_i \log(\varphi(\theta^T \boldsymbol{x}_i)) + (1-Y_i)\log((1-\varphi(\theta^T \boldsymbol{x}_i)))]$$

Exercice 5

Prenons le gradient de la log-vraisemblance, soit $k \in \{1,\ldots,n\}$:

$$\begin{aligned}
\frac{\partial}{\partial\theta_k} l_n(\theta) &= \frac{\partial}{\partial\theta_k}\left[\sum_{i=1}^{n} Y_i \log\left(\varphi\left(\sum_{j=1}^{d}\theta_j x_i^{(j)}\right)\right) + (1-Y_i)\log\left(1-\varphi\left(\sum_{j=1}^{d}\theta_j x_i^{(j)}\right)\right)\right] \\
&= \sum_{i=1}^{n} Y_i \frac{\partial}{\partial\theta_k}\log\left(\varphi\left(\sum_{j=1}^{d}\theta_j x_i^{(j)}\right)\right) + (1-Y_i)\frac{\partial}{\partial\theta_k}\log\left(1-\varphi\left(\sum_{j=1}^{d}\theta_j x_i^{(j)}\right)\right) \\
&= \sum_{i=1}^{n} Y_i \frac{\varphi'(\theta^T x_i)x_i^{(k)}}{\varphi(\theta^T x_i)} + (1-Y_i)\frac{-\varphi'(\theta^T x_i)x_i^{(k)}}{1-\varphi(\theta^T x_i)}
\end{aligned}$$

$$= \sum_{i=1}^{n} \left(Y_i - \varphi(\theta^T x_i)\right) x_i^{(k)}$$

Alors, on a que:

$$\nabla l_n(\theta) = \sum_{i=1}^{n} x_i \left(Y_i - \varphi(\theta^T x_i)\right) = \mathbf{X}_n^T \left[\mathbf{Y}_n - \Phi_n(\theta)\right]$$

On observe que:

$$\nabla^2 l_n(\theta) = \mathrm{Jac}\left(\nabla l_n(\theta)\right) = -\mathbf{X}_n^T \mathrm{Jac}\left(\Phi_n(\theta)\right)$$

Ou Jac est la matrice jacobienne:

$$\left(\mathrm{Jac}(\Phi)\right)_{i,j} = \left(\frac{\partial \Phi_i}{\partial \theta_j}\right)_{i,j}$$

On calcule:

$$\left(\frac{\partial \Phi_i}{\partial \theta_j}\right)_{i,j} = \varphi'(\theta^T x_i) x_i^j = h(\theta^T x_i) x_i^j$$

Car $\varphi' = h$. Donc:

$$\left(\nabla^2 l_n(\theta)\right)_{i,j} = -\sum_{k=1}^{n} h(\theta^T x_k) x_j^{(k)} x_k^{(i)}$$

Par définition:

$$\nabla^2 l_n(\theta) = -\mathbf{F}_n(\theta)$$

Maintenant on s'interesse a calculer $\mathbb{E}_{n,\theta}\left[\nabla l_n(\theta) \nabla l_n(\theta)^T\right]$.

$$\nabla l_n(\theta)^T = \left[\mathbf{Y}_n^T - \Phi_n(\theta)^T\right] \mathbf{X}_n$$

$$\nabla l_n(\theta) \nabla l_n(\theta)^T = \mathbf{X}_n^T \left[\mathbf{Y}_n \mathbf{Y}_n^T + \Phi_n(\theta)\Phi_n(\theta)^T - \Phi_n(\theta)\mathbf{Y}_n^T - \mathbf{Y}_n \Phi_n(\theta)^T\right] \mathbf{X}_n$$

$$\mathbb{E}_{n,\theta}\left[\nabla l_n(\theta) \nabla l_n(\theta)^T\right] = \mathbf{X}_n^T \left(\mathbb{E}_{n,\theta}\left[\mathbf{Y}_n \mathbf{Y}_n^T\right] + \Phi_n(\theta)\Phi_n(\theta)^T - \Phi_n(\theta)\mathbb{E}_{n,\theta}\left[\mathbf{Y}_n^T\right] - \mathbb{E}_{n,\theta}\left[\mathbf{Y}_n\right] \Phi_n(\theta)^T\right) \mathbf{X}_n$$

Comme $\mathbb{E}_{n,\theta}\left[\mathbf{Y}_n\right] = \Phi_n(\theta)$, alors:

$$\mathbb{E}_{n,\theta}\left[\nabla l_n(\theta) \nabla l_n(\theta)^T\right] = \mathbf{X}_n^T \left(\mathbb{E}_{n,\theta}\left[\mathbf{Y}_n \mathbf{Y}_n^T\right] - \Phi_n(\theta)\Phi_n(\theta)^T\right) \mathbf{X}_n$$

On voit aussi que, comme Y_i sont variables de Bernoulli i.i.d., :

$$\left(\mathbb{E}_{n,\theta}\left[\mathbf{Y}_n \mathbf{Y}_n^T\right]\right)_{i,j} = \begin{cases} \varphi(\theta^t x_i)\varphi(\theta^t x_i) \text{ si } i \neq j \\ \varphi(\theta^t x_i) \text{ si } i = j \end{cases}$$

Aussi, il est evident que:

$$\left(\Phi_n(\theta)\Phi_n(\theta)^T\right)_{i,j} = \varphi(\theta^t x_i)\varphi(\theta^t x_i)$$

Alors:

$$\left(\mathbb{E}_{n,\theta}\left[\nabla l_n(\theta)\nabla l_n(\theta)^T\right]\right)_{i,j} = \begin{cases} 0 \text{ si } i \neq j \\ \varphi(\theta^t x_i)(1-\varphi(\theta^t x_j)) = h(\theta^t x_i) \text{ si } i = j \end{cases}$$

Donc:

$$\mathbb{E}_{n,\theta}\left[\nabla l_n(\theta)\nabla l_n(\theta)^T\right] = \mathbf{X}_n^T\left[\operatorname{diag}\left(h(\theta^T x_i)\right)_{i=1}^n\right]\mathbf{X}_n = \mathbf{F}_n(\theta)$$

Comme $\mathbf{F}_n(\theta)$ est positive définie, alors $\nabla^2 l_n(\theta)$ est négative définie et l_n est concave presque sûrement.

Exercice 6

Supposons $\exists \theta_\star \in \mathbb{R}^p \backslash \{0_{\mathbb{R}^p}\}$ tel que $\forall k \in \{1, \ldots, n\}$:

$$\begin{cases} \theta_\star^T x_k > 0 & \text{si } Y_k = 1 \\ \theta_\star^T x_k < 0 & \text{si } Y_k = 0 \end{cases}$$

Nous avons que :

$$l_n(\theta) = \sum_{i=1}^n [Y_i \log(\varphi(\theta^T \boldsymbol{x}_i)) + (1-Y_i)\log((1-\varphi(\theta^T \boldsymbol{x}_i)))]$$

Prenons $\lambda > 0$. Donc :

$$l_n(\lambda\theta_\star) = \sum_{i=1}^n [Y_i \log(\varphi(\lambda\theta_\star^T \boldsymbol{x}_i)) + (1-Y_i)\log((1-\varphi(\lambda\theta_\star^T \boldsymbol{x}_i)))]$$

On remarque que :

$$\varphi(-t) = \frac{1}{1+e^t} = \frac{e^{-t}}{1+e^{-t}} = 1-\varphi(t)$$

Nous pouvons donc réécrire :

$$l_n(\lambda\theta_\star) = \sum_{i=1}^n [Y_i \log(\varphi(\lambda\theta_\star^T \boldsymbol{x}_i)) + (1-Y_i)\log(\varphi(-\lambda\theta_\star^T \boldsymbol{x}_i))]$$

Prenons $i \in \{1, \ldots, n\}$, on remarque que :

- Si $Y_i = 1$: $\lambda\theta_\star^T x_i > 0$
- Si $Y_i = 0$: $\lambda\theta_\star^T x_i < 0$

Alors :

$$l_n(\lambda\theta_\star) = \sum_{i=1}^{n} \log(\varphi(\lambda|\theta_\star^T \boldsymbol{x}_i|))$$

Et donc, comme φ est une fonction continue avec $\lim_{t\to+\infty}\varphi(t) = 1$ et $x \mapsto \log(x)$ est aussi continue :

$$\lim_{\lambda\to+\infty} l_n(\lambda\theta_\star) = \lim_{\lambda\to+\infty}\sum_{i=1}^{n} \log(\varphi(\lambda|\theta_\star^T \boldsymbol{x}_i|)) = 0$$

On a que $\forall\theta \in \mathbb{R}^p$, comme $L_n(\theta)$ est un produit de probabilités et φ n'atteint ses limites qu'à l'infini, $L_n(\theta) < 1$, et donc $l_n(\theta) < 0$.

Donc $\sup_{\theta\in\mathbb{R}^p} l_n(\theta) = 0$ n'est pas atteint en $\mathbb{R}^p$ et l'estimateur de maximum de vraisemblance n'existe pas.

Exercice 7

On rappele que :

$$L_n(\theta) = \prod_{i=1}^{n} (\varphi(\theta^T x_i))^{Y_i}(1 - \phi(\theta^T x_i))^{1-Y_i}$$

Soit $\bar\theta \in \mathbb{R}^p$, et $\lambda \in \mathbb{R}$. On appelle $\theta_\lambda = \lambda\theta_\star + (\bar\theta - \theta_\star)$. On remarque que :

$$\varphi(\theta_\lambda^T x_i) = \varphi(\lambda\theta_\star^T x_i + (\bar\theta - \theta_\star)^T x_i)$$

$$\varphi(\theta_\lambda^T x_i) = \varphi((\lambda - 1)\theta_\star^T x_i + \bar\theta x_i)$$

Et aussi :

$$1 - \varphi(\theta_\lambda^T x_i) = \varphi(-\theta_\star^T x_i) = \varphi((1 - \lambda)\theta_\star^T x_i - \bar\theta x_i)$$

Regardons donc les possibilités pour les facteurs du produit de $L_n(\theta_\lambda) = \prod_{i=1}^{n} L_n^{(i)}(\theta_\lambda)$:

- Si $Y_i = 1$: $L_n^{(i)}(\theta_\lambda) = \varphi((\lambda - 1)\theta_\star^T x_i + \bar\theta x_i)$

- Si $Y_i = 1$: $L_n^{(i)}(\theta_\lambda) = \varphi((1-\lambda)\theta_\star^T x_i - \bar{\theta} x_i)$

On vérifie les possibilités sur $\theta_\star^T x_i$:

- Si $\theta_\star^T x_i > 0$: $Y_i = 1$ et donc $L_n^{(i)}(\theta_\lambda) = \varphi((\lambda - 1)|\theta_\star^T x_i| + \bar{\theta} x_i)$
- Si $\theta_\star^T x_i < 0$: $Y_i = 1$ et donc $L_n^{(i)}(\theta_\lambda) = \varphi((\lambda - 1)|\theta_\star^T x_i| - \bar{\theta} x_i)$
- Si $\theta_\star^T x_i = 0$, donc $i \in \mathcal{E}$ et $L_n^{(i)}(\theta_\lambda) = \varphi(\bar{\theta}^T x_i)^{Y_i} \varphi(-\bar{\theta}^T x_i)^{1-Y_i}$

Nous avons donc que :

$$L_n(\theta_\lambda) = \prod_i^n L_n^{(i)}(\theta_\lambda) = \underbrace{\prod_{i \in \mathcal{E}} \varphi(\bar{\theta}^T x_i)^{Y_i} \varphi(-\bar{\theta}^T x_i)^{1-Y_i}}_{\text{constant}} + \underbrace{\prod_{i \notin \mathcal{E}} (\varphi(\lambda - 1)|\theta_\star^T x_i| + (-1)^{1-Y_i} \bar{\theta}^T x_i)}_{\text{croissant en } \lambda}$$

On peut donc faire λ croître autant qu'on veut. Si on suppose qu'un certain $\tilde{\theta}$ est celui de maximum de vraisemblance, on peut toujours trouver un λ tel que $L_n(\theta_\lambda) > L_n(\tilde{\theta})$ où $\theta_\lambda = \lambda\theta_\star + \tilde{\theta} - \theta_\star$.

Exercice 8

On suit l'indication, on va d'abord montrer qu'il existe $\zeta > 0$ tel que $\forall \theta \in \mathcal{S}(0,1)$ on a $\theta^T x_{k_{1,\theta}} > \zeta$ et $\theta^T x_{k_{2,\theta}} < -\zeta$.

Par hypothèse, $\forall \theta \in \mathbb{R}^d \backslash 0_{\mathbb{R}^p}$, $\exists k_1, k_2$ tels que $\theta^T x_{k_{1,\theta}} > 0$ et $\theta^T x_{k_{2,\theta}} < 0$. Posons donc :

$$\zeta := \inf_{\theta \in \mathcal{S}(0,1)} \left(\frac{1}{2} \min\{|\theta^T x_{k_{1,\theta}}|, |\theta^T x_{k_{2,\theta}}|\} \right) \geq 0$$

Et on appelle :

$$\zeta_\theta = \frac{1}{2} \min\{|\theta^T x_{k_{1,\theta}}|, |\theta^T x_{k_{2,\theta}}|\}$$

On va montrer par absurde que $\zeta \neq 0$.

On suppose donc que $\zeta = 0$. On construit la suite $(\theta_n)_{n\in\mathbb{N}}$ telle que la suite $(\zeta_{\theta_n})_{n\in\mathbb{N}}$ converge à zéro.

Comme il y a un nombre fini d'indices $i \in \{1,2\}$, $\exists \chi : \mathbb{N} \to \mathbb{N}$ une fonction strictement croissante telle qu'on extrait une sous-suite de $(\theta_{\zeta(n)})_{n\in\mathbb{N}}$ où tous les éléments sont de la forme :

$$\zeta_{\chi(n)} = \frac{1}{2}|\theta^T_{\chi(n)} x_{k_{\sigma,\theta_{\chi(n)}}}|$$

Où $\sigma \in \{1, 2\}$ est fixe.

Comme il s'agit d'une sous-suite d'une suite qui converge, elle converge aussi à zéro.

Par le même argument, comme $k \in \{1, \ldots, n\}$ est fini, on peut extraire une sous-suite avec la fonction $\psi : \mathbb{N} \to \mathbb{N}$ telle que :

$$\zeta_{(\chi\circ\psi)(n)} = \frac{1}{2}|\theta^T_{(\chi\circ\psi)(n)} x_{\tilde{k}_{\sigma,\theta_{(\chi\circ\psi)(n)}}}|$$

Où $\tilde{k} \in \{1, \ldots, n\}$ est fixe. On a par le même argument que cette sous-suite converge à zéro.

Or, comme la transformation entre les deux sous-suites $(\theta_{(\chi\circ\psi)(n)})_{n\in\mathbb{N}}$ et $(\zeta_{(\chi\circ\psi)(n)})_{n\in\mathbb{N}}$ est maintenant continue et $(\theta_{(\chi\circ\psi)(n)})_{n\in\mathbb{N}} \in \mathcal{S}(0, 1)^{\mathbb{N}}$, où $\mathcal{S}(0, 1)$ est un compacte, on a que $\exists\theta_\star$ tel que $\theta^T_\star x_{\tilde{k}_{\sigma,\theta_\star}}$, ce qui est absurde par l'énoncé.

On va maintenant montrer que $\forall M > 0$, $\exists\lambda_M$ tel que $\forall\theta \in \mathcal{S}(0, 1)$ et $\forall\lambda > \lambda_M$, $l_n(\lambda\theta) \leq -M$. Pour faire cela on va montrer que $\lim_{\lambda\to+\infty} l_n(\lambda\theta) \to -\infty$ pour tout $\theta \in \mathcal{S}(0, 1)$.

On fixe $M > 0$ et λ_M qui sera définie à posteriori. Soit $\theta \in \mathcal{S}(0, 1)$ et $\lambda > 0$. Comme avant, on analyse les différents cas :

- Cas 1 ($i \in I_1$) : Si $Y_i = 1$ et $\lambda\theta^T x_i > 0$
- Cas 2 ($i \in I_2$) : Si $Y_i = 0$ et $\lambda\theta^T x_i < 0$
- Cas 3 ($i \in I_3$) : On n'a pas besoin d'étudier ces cas car soit $\lambda\theta^T_\star x_i = 0$ et donc la contribution est $\log(\frac{1}{2})$ ou il est classifié correctement, et donc sa contribution tend à 0 quand nous allons prendre la limite en $\lambda \to +\infty$. On appellera leur contribution à la log-vraisemblance de $\alpha(\lambda)$, où $\alpha(\lambda) \to 0$ quand $\lambda \to +\infty$.

Dans le cas 1 nous avons que, d'après ce qu'on vient de montrer :

$$\log(\varphi(\lambda\theta^T x_i)) = \log(\varphi(\lambda_{\theta,i}\theta^T x_{k_{2,\theta}})) \leq \log(\varphi(-\lambda_{\theta,i}\zeta))$$

Où $\lambda_{\theta,i} = \lambda\frac{\theta^T x_i}{\theta^T x_{k_{2,\theta}}}$.

Dans le cas 2 nous avons que, d'après ce qu'on vient de montrer :

$$\log(\varphi(1 - \lambda\theta^T x_i)) = \log(1 - \varphi(\lambda'_{\theta,i}\theta^T x_{k_{1,\theta}})) \leq 1 - \log(\varphi(-\lambda'_{\theta,i}\zeta))$$

Où $\lambda'_{\theta,i} = \lambda\frac{\theta^T x_i}{\theta^T x_{k_{1,\theta}}}$.

Or, comme $\theta \in \mathcal{S}(0, 1)$ compacte, nous avons que les deux fonctions $\theta \mapsto \lambda'_{\theta,i}$ et $\theta \mapsto \lambda'_{\theta,i}$ ont un infimum et cet infimum est non nul car les fonctions sont continues par morceau et

non nulles. En plus, comme il y a un nombre fini d'indices, nous avons avons un $\lambda^\star$ tel que $\forall i \in \{1,\dots,n\}$ dans les cas 1 et 2, $\lambda_{\theta,i} > \lambda^\star$ et $\lambda'_{\theta,i} > \lambda^\star$.
Or, comme notre $\lambda^\star$ est indépendant de θ, nous avons donc :

- Dans le cas 1 : $\log(\varphi(\lambda\theta^T x_i)) < \log(\varphi(-\lambda^\star\zeta))$
- Dans le cas 2 : $\log(\varphi(1-\lambda\theta^T x_i)) < 1 - \log(\varphi(\lambda^\star\zeta))$

Et donc :

$$l_n(\lambda\theta) = \sum_{i\in I_1} \log(\varphi(\lambda\theta^T x_i)) + \sum_{i\in I_2}(1-\log(\varphi(\lambda\theta^T x_i))) + \alpha(\lambda)$$

Donc :

$$l_n(\lambda\theta) \leq \sum_{i\in I_1} \log(\varphi(-\lambda^\star\zeta)) + \sum_{i\in I_2}\log(1-\varphi(\lambda^\star\zeta)) + \alpha(\lambda)$$

Comme $\lambda^\star$ est croissant en λ :

$$l_n(\lambda\theta) \leq \sum_{i\in I_1} \log(\varphi(-\lambda^\star\zeta)) + \sum_{i\in I_2}\log(1-\varphi(\lambda^\star\zeta)) + \alpha(\lambda) \to -\infty$$

Or, comme cela tend vers $-\infty$, $\forall M > 0$, $\exists \lambda_M$ tel que $l_n(\lambda_M\theta) < -M$.
Le fait que $l_n(\theta)$ est borné par zéro pour tout $\theta \in \mathbb{R}^p$ nous donne que $\exists M > 0$ tel que $\forall\theta \in \mathbb{R}^p$, $l_n(\theta) \leq M$. D'après ce qu'on vient de montrer, nous avons aussi qu'il existe λ_M tel que $\forall\theta \in \mathcal{S}(0,1)$ et $\forall\lambda > \lambda_M$, $l_n(\lambda\theta) \leq -M$. Or, cela nous permets de conclure que :

$$\sup_{\theta\in\mathbb{R}^p} l_n(\theta) = \sup_{\theta\in\bar{\mathbb{B}}(0,\lambda_M)} l_n(\theta)$$

Comme $\bar{\mathbb{B}}(0,\lambda_M)$ est une boule fermé compacte, et $\theta \mapsto l_n(\theta)$ est une somme et composition de fonctions continues, nous avons que le supremum est atteint à un $\theta_\star$ particulier. Il s'agit donc du maximum de vraisemblance. En plus, comme la fonction $\theta \mapsto l_n(\theta)$ est concave, ce maximum est unique.

Exercice 9

Comme h est 1-Lipschitzienne On a que $\forall\theta,\vartheta \in \mathbb{R}^p$:

$$||\mathbf{F}_n(\theta) - \mathbf{F}_n(\vartheta)|| = \left|\left|\sum_{i=1}^n \left[h(\theta^T x_i) - h(\vartheta^T x_i)\right] x_i x_i^T\right|\right|$$

$$\leq \sum_{i=1}^{n} \left| h(\theta^T x_i) - h(\vartheta^T x_i) \right| \left| \left| x_i x_i^T \right| \right|$$

$$\leq \sum_{i=1}^{n} ||\theta - \vartheta|| \, ||x_i|| \left| \left| x_i x_i^T \right| \right|$$

$$\leq n \, ||\theta - \vartheta|| \left(\frac{1}{n} \sum_{i=1}^{n} ||x_i||^3 \right)$$

$$\leq Cn \, ||\theta - \vartheta||$$

Pour un $C \in \mathbb{R}$, d'après l'hypothèse 2.

Exercice 10

Pour simplicité, appelons $g = \nabla l_n$ et df la différentielle d'une fonction f. Notons que comme $\mathbf{F}_n$ est de classe $\mathcal{C}^\infty$ alors g l'est aussi, car $dg = \nabla^2 l_n = -\mathbf{F}_n$. En faisant un développement limité au tour de $\theta \in \mathbb{R}^p$:

$$g(\hat{\theta}_n) = g(\theta) + dg(\theta)(h) + \frac{1}{2} d^2 g(\xi)(h)(h)$$

Ou $h = \hat{\theta}_n - \theta$ et $\xi \in B(\theta, ||h||)$ la boule de rayon $||h||$ au tour de θ.
Alors, comme $||dg(\theta) - dg(\vartheta)|| = ||\mathbf{F}_n(\theta) - \mathbf{F}_n(\vartheta)|| \leq Cn \, ||\theta - \vartheta||$ on sait que $\forall \theta \in \mathbb{R}^p$:

$$\left| \left| d^2 g(\theta) \right| \right| \leq nC$$

Choisissons:

$$R_n = \frac{d^2 g(\xi)(h)}{2n} = \frac{d^2 g(\xi)(\hat{\theta}_n - \theta)}{2n}$$

Alors on vérifie que:

$$\nabla l_n(\hat{\theta}_n) - \nabla l_n(\theta) = g(\hat{\theta}_n) - g(\theta) = dg(\theta)(h) + \frac{1}{2} d^2 g(\xi)(h)(h) = \left[-\mathbf{F}_n(\theta) + nR_n \right] (\hat{\theta}_n - \theta)$$

Ce qu'entraîne:

$$\frac{\nabla l_n(\hat{\theta}_n) - \nabla l_n(\theta)}{\sqrt{n}} = \left(\frac{-\mathbf{F}_n(\theta)}{n} + R_n \right) \sqrt{n} (\hat{\theta}_n - \theta)$$

Il nous reste juste a vérifier la convergence de R_n. Pour cela, on note que:

$$||R_n|| = \frac{1}{2n} \left| \left| d^2 g(\xi)(\hat{\theta}_n - \theta) \right| \right|$$

$$\leq \frac{1}{2n} \sup_{\vartheta \in \mathbb{R}^p} \left|\left|d^2 2g(\vartheta)\right|\right| \left|\left|\hat{\theta}_n - \theta\right|\right| \leq \frac{C}{2} \left|\left|\hat{\theta}_n - \theta\right|\right|$$

Alors, comme $\hat{\theta}_n \xrightarrow{\mathbb{P}_{n,\theta}-\text{proba}} \theta$ on a que:

$$R_n \xrightarrow{\mathbb{P}_{n,\theta}-\text{proba}} 0$$

Exercice 11

D'après l'exercice 5, on a que:

$$\frac{1}{\sqrt{n}} \nabla l_n(\theta) = \sum_{i=1}^{n} \frac{1}{\sqrt{n}} \left(Y_i - \varphi(\theta^T x_i)\right) x_i$$

A titre de simplicité, appelons:

$$\Gamma_{n,i} = \frac{1}{\sqrt{n}} \left(Y_i - \varphi(\theta^T x_i)\right) x_i$$

Alors $\{(\Gamma_{n,i})_{i=1}^n, n \in \mathbb{N}\}$ est un tableau triangulaire de variables aléatoires définis sur le même espace de probabilités. Nous véfifions les hypothèses du théorème de Linderberg-Feller.

Notons que $\mathbb{E}\left[\Gamma_{n,i}\right] = 0$ et que:

$$\left|\left|\Gamma_{n,i}\right|\right|^2 = \frac{1}{n} \left|\left|x_i\right|\right|^2 \left|Y_i - \varphi(\theta^T x_i)\right|^2$$

Alors:

$$\mathbb{E}\left[\left|\left|\Gamma_{n,i}\right|\right|^2\right] = \frac{1}{n} \left|\left|x_i\right|\right|^2 \mathrm{Var}(Y_i) = \frac{1}{n} \left|\left|x_i\right|\right|^2 h(\theta^T x_i) \leq \infty$$

Aussi, par l'hypothèse 1:

$$\begin{aligned} \sum_{i=1}^{n} \mathrm{Var}(\Gamma_{n,i}) &= \sum_{i=1}^{n} \frac{1}{n} x_i h(\theta^T x_i) x_i^T \\ &= \frac{1}{n} \mathbf{F}_n(\theta) \xrightarrow{n \to \infty} Q(\theta) \end{aligned}$$

Maintenant on fixe $\varepsilon > 0$ et on calcule l'évènement:

$$A_n = \{\left|\left|\Gamma_{n,i}\right|\right| > \varepsilon\} = \left\{\frac{\sqrt{n}\varepsilon}{\left|\left|x_i\right|\right|} < \left|Y_i - \varphi(\theta^T x_i)\right|\right\}$$

Comme $|Y_i - \varphi(\theta^T x_i)| < 1$, alors pour n assez grand, $\mathbb{P}_{n,\theta}(A_n) = 0$. C'est a dire que pour n assez grand $\mathbf{1}_{\{||\Gamma_{n,i}|| > \varepsilon\}} = 0$ presque partout.

Donc:

$$\lim_{n\to\infty} \sum_{i=1}^{n} \mathbb{E}\left[||\Gamma_{n,i}||^2 \mathbf{1}_{\{||\Gamma_{n,i}||>\varepsilon\}}\right] = 0$$

Finalement on applique le théorème de Linderberg-Feller pour concluir que:

$$\frac{1}{\sqrt{n}} \nabla l_n(\theta) = \sum_{i=1}^{n} \Gamma_{n,i} \xrightarrow{\mathbb{P}_{n,\theta}} \mathcal{N}\left(0, Q(\theta)\right)$$

Exercice 12

Comme $\mathbf{F}_n(\theta) \xrightarrow{n\to\infty} Q(\theta)$, $R_n \xrightarrow{\mathbb{P}_{n,\theta}-\text{proba}} 0$ (constante) et que $\frac{1}{\sqrt{n}} \nabla l_n(\theta) \xRightarrow{\mathbb{P}_{n,\theta}} \mathcal{N}\left(0, Q(\theta)\right)$, alors, par Slustky:

$$Q(\theta)\sqrt{n}(\hat{\theta}_n - \theta) \xRightarrow{\mathbb{P}_{n,\theta}} \mathcal{N}\left(0, Q(\theta)\right)$$

Alors:

$$\sqrt{n}(\hat{\theta}_n - \theta) \xRightarrow{\mathbb{P}_{n,\theta}} Q(\theta)^{-1}\mathcal{N}\left(0, Q(\theta)\right) = \mathcal{N}\left(0, Q(\theta)^{-1}\right)$$

Exercice 13

Comme $\mathbf{F}_n(\hat{\theta}_n^{MV})$ est une application continue en $\hat{\theta}_n^{MV}$, alors

$$\beta_{n,k} = \left(\left[\frac{\mathbf{F}_n(\hat{\theta}_n^{MV})}{n}\right]^{-1}\right)_{k,k}$$

l'est aussi.
Comme $\hat{\theta}_n^{MV} \xrightarrow{\mathbb{P}_{m,\theta}-\text{proba}} \theta$ une constante, alors $\beta_{n,k} \xrightarrow{\mathbb{P}_{m,\theta}-\text{proba}} \left(Q(\theta)^{-1}\right)_{k,k} = \gamma_k$. Comme $x \mapsto 1/\sqrt{x}$ est continue sur $\mathbb{R}_+$, donc $1/\sqrt{\beta_{n,k}} \xrightarrow{\mathbb{P}_{m,\theta}-\text{proba}} 1/\sqrt{\gamma_k}$.
En regardant par composante on voit que:

$$\sqrt{n}(\hat{\theta}_{n,k} - \theta_k) \xRightarrow{\mathbb{P}_{n,\theta}} \mathcal{N}\left(0, \left(Q(\theta)^{-1}\right)_{k,k}\right) = \mathcal{N}\left(0, \gamma_k\right)$$

Donc, par Skutsky:

$$\sqrt{\frac{n}{\beta_{n,k}}}(\hat{\theta}_{n,k}^{MV} - \theta_k) \xRightarrow{\mathbb{P}_{n,\theta}} \frac{1}{\sqrt{\gamma_k}}\mathcal{N}\left(0, \gamma_k\right) = \mathcal{N}(0, 1)$$

Devoir Maison 2

Exercice 14

L'intervalle asymptotique de niveau de couverture α pour θ_k est donné par:

$$\mathcal{I}_{1-\alpha} = \left[\hat{\theta}_{n,k}^{MV} - \sqrt{\frac{\beta_{n,k}}{n}} z_{1-\alpha/2}, \hat{\theta}_{n,k}^{MV} + \sqrt{\frac{\beta_{n,k}}{n}} z_{1-\alpha/2}\right]$$

Ou z_ζ est le ζ-quantile d'une loi Gaussienne centrée et réduite.

Exercice 15

On définit le test symétrique a l'intervalle de confiance:

$$\phi\colon \hat{\theta}_{n,k}^{MV} \mapsto \mathbf{1}_{\{|\hat{\theta}_{n,k}^{MV}| > z_{1-\alpha/2}\sqrt{\beta_{n,k}/n}\}}$$

Un test asymptotique de niveau α pour la hypothèse H_0.

Exercice 16

La p-valeur asymptotique de ce test $\bar{\alpha}$ satisfait:

$$\hat{\theta}_{n,k}^{MV} = z_{1-\bar{\alpha}/2}\sqrt{\beta_{n,k}/n}$$

Alors:

$$\bar{\alpha} = 2\left(1 - \Phi\left(\sqrt{\frac{n}{\beta_{n,k}}}\hat{\theta}_{n,k}^{MV}\right)\right)$$

Ou Φ est la fonction de répartition d'une gaussienne centrée et réduite.

gr5_DM2

October 19, 2022

[ ]:
```python
import pandas as pd
import statsmodels.api as sm
```

1 Premier traitement des données

[ ]:
```python
data = pd.read_csv("Titanic.csv")
```

[ ]:
```python
data.head()
```

[ ]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

[ ]:
```python
data_clean = data.drop(columns = ["PassengerId", "Ticket", "Cabin", "Embarked"])
data_clean['Age'] =  data_clean['Age'].fillna(value = data_clean['Age'].mean())
```

[ ]:
```python
print(data_clean.shape)
data_clean.head()
```

```
(891, 8)
```

```
[ ]:    Survived  Pclass                                               Name  \
     0         0       3                            Braund, Mr. Owen Harris
     1         1       1  Cumings, Mrs. John Bradley (Florence Briggs Th…
     2         1       3                             Heikkinen, Miss. Laina
     3         1       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)
     4         0       3                           Allen, Mr. William Henry

           Sex   Age  SibSp  Parch     Fare
     0    male  22.0      1      0   7.2500
     1  female  38.0      1      0  71.2833
     2  female  26.0      0      0   7.9250
     3  female  35.0      1      0  53.1000
     4    male  35.0      0      0   8.0500
```

1.1 Exercice 1

```
[ ]: alive = data_clean['Survived'].sum()
     total = data_clean['Survived'].shape[0]

     print(f"{total - alive} passagers sont décédés, ce qui répresente {(100 *␣
      ↪(total - alive) / total):.2f} % du total.")
     print()
```

```
549 passagers sont décédés, ce qui répresente 61.62 % du total.
```

1.2 Exercice 2

```
[ ]: total_man = data_clean.groupby('Sex')['Name'].count()['male']
     total_woman = data_clean.groupby('Sex')['Name'].count()['female']

     print(f"Percentage d'hommes : {(100*total_man/(total_woman+total_man)):.2f} %,␣
      ↪percentage de femmes {(100*total_woman/(total_woman+total_man)):.2f} %.")
```

```
Percentage d'hommes : 64.76 %, percentage de femmes 35.24 %.
```

```
[ ]: data_clean_dead = data_clean.loc[data_clean['Survived'] == 0]

     dead_man = data_clean_dead.groupby('Sex')['Name'].count()['male']
     dead_woman = data_clean_dead.groupby('Sex')['Name'].count()['female']

     print(f"Parmi les personnes décédés, {(dead_man/(dead_man + dead_woman)*100):.
      ↪2f} % ont été des hommes et {(dead_woman/(dead_man + dead_woman)*100):.2f} %␣
      ↪des femmes.")
```

```
Parmi les personnes décédés, 85.25 % ont été des hommes et 14.75 % des femmes.
```

```
[ ]: data_clean_alive = data_clean.loc[data_clean['Survived'] == 1]
```

```
alive_man = data_clean_alive.groupby('Sex')['Name'].count()['male']
alive_woman = data_clean_alive.groupby('Sex')['Name'].count()['female']

print(f"Parmi les personnes survivantes, {(alive_man/(alive_man +␣
↪alive_woman)*100):.2f} % ont été des hommes et {(alive_woman/(alive_man +␣
↪alive_woman)*100):.2f} % des femmes.")
```

```
Parmi les personnes survivantes, 31.87 % ont été des hommes et 68.13 % des
femmes.
```

Observation : Malgré le fait que la majorité de la population du titanic était composé des hommes, on observe que la plupart des personnes qui ont survecu l'accident ont été des femmes (68.13 % contre 31.87 %) et plus d'hommes ont décédé (85.25 % contre 14.75%).

1.3 Exercice 3

[ ]:
```
n_first = data_clean['Pclass'].value_counts()[1]
n_second = data_clean['Pclass'].value_counts()[2]
n_third = data_clean['Pclass'].value_counts()[3]

n_total = data_clean['Pclass'].count()

print(f"Il y avait {n_first} passagers en première classe ({(100*n_first/
↪n_total):.2f} %), {n_second} en deuxième ({(100*n_second/n_total):.2f} %) et␣
↪{n_third} en troisième ({(100*n_third/n_total):.2f} %).")
```

```
Il y avait 216 passagers en première classe (24.24 %), 184 en deuxième (20.65 %)
et 491 en troisième (55.11 %).
```

[ ]:
```
total_dead = dead_man + dead_woman

dead_first = data_clean_dead.groupby('Pclass')['Name'].count()[1]
dead_second = data_clean_dead.groupby('Pclass')['Name'].count()[2]
dead_third = data_clean_dead.groupby('Pclass')['Name'].count()[3]

print("Au total : \n")

print(f"Parmi les personnes décédés, {(dead_first/(total_dead)*100):.2f} % ont␣
↪été de première classe.")
print(f"Parmi les personnes décédés, {(dead_second/(total_dead)*100):.2f} % ont␣
↪été de deuxième classe.")
print(f"Parmi les personnes décédés, {(dead_third/(total_dead)*100):.2f} % ont␣
↪été de troisième classe.\n")

print("On regarde dans une même classe : \n")
print(f"Parmi les personnes de première classe, {(100*dead_first / n_first):.
↪2f} % sont décédés")
```

```
print(f"Parmi les personnes de deuxième classe, {(100*dead_second / n_second):.
 ↪2f} % sont décédés")
print(f"Parmi les personnes de troisième classe, {(100*dead_third / n_third):.
 ↪2f} % sont décédés")
```

```
Au total :

Parmi les personnes décédés, 14.57 % ont été de première classe.
Parmi les personnes décédés, 17.67 % ont été de deuxième classe.
Parmi les personnes décédés, 67.76 % ont été de troisième classe.

On regarde dans une même classe :

Parmi les personnes de première classe, 37.04 % sont décédés
Parmi les personnes de deuxième classe, 52.72 % sont décédés
Parmi les personnes de troisième classe, 75.76 % sont décédés
```

[ ]:
```
total_alive = alive_man + alive_woman

alive_first = data_clean_alive.groupby('Pclass')['Name'].count()[1]
alive_second = data_clean_alive.groupby('Pclass')['Name'].count()[2]
alive_third = data_clean_alive.groupby('Pclass')['Name'].count()[3]

print("Au total : \n")

print(f"Parmi les personnes qui ont survécu, {(alive_first/(total_alive)*100):.
 ↪2f} % ont été de première classe.")
print(f"Parmi les personnes qui ont survécu, {(alive_second/(total_alive)*100):.
 ↪2f} % ont été de deuxième classe.")
print(f"Parmi les personnes qui ont survécu, {(alive_third/(total_alive)*100):.
 ↪2f} % ont été de troisième classe.\n")

print("On regarde dans une même classe : \n")
print(f"Parmi les personnes de première classe, {(100*alive_first / n_first):.
 ↪2f} % ont survécu")
print(f"Parmi les personnes de deuxième classe, {(100*alive_second / n_second):.
 ↪2f} % ont survécu")
print(f"Parmi les personnes de troisième classe, {(100*alive_third / n_third):.
 ↪2f} % ont survécu")
```

```
Au total :

Parmi les personnes qui ont survécu, 39.77 % ont été de première classe.
Parmi les personnes qui ont survécu, 25.44 % ont été de deuxième classe.
Parmi les personnes qui ont survécu, 34.80 % ont été de troisième classe.

On regarde dans une même classe :
```

```
Parmi les personnes de première classe, 62.96 % ont survécu
Parmi les personnes de deuxième classe, 47.28 % ont survécu
Parmi les personnes de troisième classe, 24.24 % ont survécu
```

Observations : On observe que la plupart des personnes qui ont survecu ont été dans la première classe (aussi plus grad taux de survie). En plus ce sont ceux qui étaient dans la troisième classe qui sont décédés le plus (aussi plus grand taux de mort).

1.4 Exercice 4

[ ]:
```
data_clean_corr = data_clean.copy()

# Turns Sex into numerical
data_clean_corr['Sex'] = data_clean['Sex'].map(lambda x: 1 if x == 'male' else␣
↪0)

# Creates category columns for Class
data_clean_corr = data_clean_corr.join(pd.
↪get_dummies(data_clean_corr['Pclass'])).drop(columns='Pclass')
data_clean_corr = data_clean_corr.rename(columns={1: '1st Class', 2: '2nd␣
↪Class', 3: '3rd Class'})

# Covariables
covar_names = ['Sex', 'Age', 'SibSp', 'Parch', 'Fare', '1st Class', '2nd␣
↪Class', '3rd Class']

display(data_clean_corr.corr())
```

```
           Survived       Sex       Age     SibSp     Parch      Fare  \
Survived   1.000000 -0.543351 -0.069809 -0.035322  0.081629  0.257307
Sex       -0.543351  1.000000  0.084153 -0.114631 -0.245489 -0.182333
Age       -0.069809  0.084153  1.000000 -0.232625 -0.179191  0.091566
SibSp     -0.035322 -0.114631 -0.232625  1.000000  0.414838  0.159651
Parch      0.081629 -0.245489 -0.179191  0.414838  1.000000  0.216225
Fare       0.257307 -0.182333  0.091566  0.159651  0.216225  1.000000
1st Class  0.285904 -0.098013  0.319916 -0.054582 -0.017633  0.591711
2nd Class  0.093349 -0.064746  0.006589 -0.055932 -0.000734 -0.118557
3rd Class -0.322308  0.137143 -0.281004  0.092548  0.015790 -0.413333

           1st Class  2nd Class  3rd Class
Survived    0.285904   0.093349  -0.322308
Sex        -0.098013  -0.064746   0.137143
Age         0.319916   0.006589  -0.281004
SibSp      -0.054582  -0.055932   0.092548
Parch      -0.017633  -0.000734   0.015790
Fare        0.591711  -0.118557  -0.413333
1st Class   1.000000  -0.288585  -0.626738
2nd Class  -0.288585   1.000000  -0.565210
```

```
3rd Class  -0.626738  -0.565210   1.000000
```

Nous observons que certaines variables sont fortement corrélées avec le taux de survie, comme, par exemple, le sexe du passager et la classe dans laquele il voyage. Cela suggère que ces variables peuvent être utilisées pour prédire la probabilité de survie d'un passager. De plus, certaines variables sont corrélées entre elles, comme le tarif du billet et la classe (un billet de meilleure classe coûte plus).

1.5 Exercice 5

[ ]:
```
alpha=.05
Xs = data_clean_corr[covar_names]

logit_reg = sm.Logit(data_clean_corr[['Survived']], Xs).fit()
print(logit_reg.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.442576
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:               Survived   No. Observations:                  891
Model:                          Logit   Df Residuals:                      883
Method:                           MLE   Df Model:                            7
Date:                Wed, 19 Oct 2022   Pseudo R-squ.:                  0.3354
Time:                        23:13:30   Log-Likelihood:                -394.34
converged:                       True   LL-Null:                       -593.33
Covariance Type:            nonrobust   LLR p-value:                 6.452e-82
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Sex           -2.7609      0.199    -13.856      0.000      -3.151      -2.370
Age           -0.0395      0.008     -5.035      0.000      -0.055      -0.024
SibSp         -0.3501      0.110     -3.194      0.001      -0.565      -0.135
Parch         -0.1133      0.118     -0.964      0.335      -0.344       0.117
Fare           0.0030      0.002      1.223      0.221      -0.002       0.008
1st Class      3.8409      0.447      8.602      0.000       2.966       4.716
2nd Class      2.8177      0.348      8.091      0.000       2.135       3.500
3rd Class      1.6910      0.291      5.818      0.000       1.121       2.261
==============================================================================
```

[ ]:
```
logit_reg.conf_int(alpha).rename({0: '5%', 1: '95%'}, axis=1)
```

[ ]:
```
                5%       95%
Sex      -3.151458 -2.370395
Age      -0.054863 -0.024117
SibSp    -0.564915 -0.135246
Parch    -0.343866  0.117198
```

```
Fare       -0.001804  0.007786
1st Class   2.965671  4.716044
2nd Class   2.135153  3.500336
3rd Class   1.121332  2.260654
```

Les intervalles de confiance montrent que les variables ‘Parch’ et ‘Fare’ ne sont pas significatives à niveau 5% car ils contiennent zéro.

1.6 Exercice 6

[ ]:
```
print(logit_reg.pvalues)
```

```
Sex          1.165424e-43
Age          4.781978e-07
SibSp        1.403955e-03
Parch        3.352689e-01
Fare         2.214914e-01
1st Class    7.865927e-18
2nd Class    5.929875e-16
3rd Class    5.955611e-09
dtype: float64
```

Le p-valeur obtenu pour le test sur ‘Parch’ confirme l’observation faite dans l’exercice 5. La haute p-valeur indique que, pour notre modèle, cette variable ne porte pas beaucoup d’information sur le taux de survie.

1.7 Exercice 7

[ ]:
```
covars_noparch = covar_names.copy()
covars_noparch.remove('Parch')

Xs = data_clean_corr[covars_noparch]

logit_reg = sm.Logit(data_clean_corr[['Survived']], Xs).fit()
print(logit_reg.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.443106
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:               Survived   No. Observations:                  891
Model:                          Logit   Df Residuals:                      884
Method:                           MLE   Df Model:                            6
Date:                Wed, 19 Oct 2022   Pseudo R-squ.:                  0.3346
Time:                        23:13:30   Log-Likelihood:                -394.81
converged:                       True   LL-Null:                       -593.33
Covariance Type:            nonrobust   LLR p-value:                 1.210e-82
==============================================================================
```

```
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Sex           -2.7245      0.195    -13.981      0.000      -3.106      -2.343
Age           -0.0392      0.008     -5.013      0.000      -0.055      -0.024
SibSp         -0.3783      0.106     -3.561      0.000      -0.587      -0.170
Fare           0.0025      0.002      1.066      0.286      -0.002       0.007
1st Class      3.8158      0.444      8.595      0.000       2.946       4.686
2nd Class      2.7664      0.343      8.061      0.000       2.094       3.439
3rd Class      1.6340      0.284      5.758      0.000       1.078       2.190
==============================================================================
```

[ ]:
```
logit_reg.conf_int(alpha).rename({0: '5%', 1: '95%'}, axis=1)
```

[ ]:

	5%	95%
Sex	-3.106472	-2.342584
Age	-0.054544	-0.023879
SibSp	-0.586601	-0.170082
Fare	-0.002077	0.007030
1st Class	2.945675	4.685900
2nd Class	2.093773	3.439090
3rd Class	1.077795	2.190121

Les nouveaux intervalles de confiance indiquent que la variable ‘Fare’ n’est toujours pas significative à niveau 5%.

1.8 Exercice 8

[ ]:
```
print(logit_reg.pvalues)
```

```
Sex          2.034713e-44
Age          5.371748e-07
SibSp        3.699590e-04
Fare         2.864641e-01
1st Class    8.310466e-18
2nd Class    7.585026e-16
3rd Class    8.501479e-09
dtype: float64
```

La p-valeur montre, comme pour ‘Parch’, que ‘Fare’ n’est pas une variable significative pour notre modèle. Cela peut s’expliquer par la forte corrélation entre cette variable et la classe du passager, rendant son utilisation redondante quand la classe est déjà fournie au modèle.

1.9 Exercice 9

[ ]:
```
covars_nofare = covars_noparch.copy()
covars_nofare.remove('Fare')

Xs = data_clean_corr[covars_nofare]
```

```python
logit_reg = sm.Logit(data_clean_corr[['Survived']], Xs).fit()
print(logit_reg.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.443793
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:               Survived   No. Observations:                  891
Model:                          Logit   Df Residuals:                      885
Method:                           MLE   Df Model:                            5
Date:                Wed, 19 Oct 2022   Pseudo R-squ.:                  0.3336
Time:                        23:13:30   Log-Likelihood:                -395.42
converged:                       True   LL-Null:                       -593.33
Covariance Type:            nonrobust   LLR p-value:                 2.366e-83
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Sex           -2.7402      0.194    -14.110      0.000      -3.121      -2.360
Age           -0.0399      0.008     -5.111      0.000      -0.055      -0.025
SibSp         -0.3583      0.104     -3.437      0.001      -0.563      -0.154
1st Class      4.0274      0.400     10.072      0.000       3.244       4.811
2nd Class      2.8376      0.337      8.410      0.000       2.176       3.499
3rd Class      1.6796      0.281      5.976      0.000       1.129       2.230
==============================================================================
```

[ ]:
```python
logit_reg.conf_int(alpha).rename({0: '5%', 1: '95%'}, axis=1)
```

[ ]:

	5%	95%
Sex	-3.120844	-2.359592
Age	-0.055136	-0.024570
SibSp	-0.562530	-0.153989
1st Class	3.243699	4.811148
2nd Class	2.176266	3.498922
3rd Class	1.128785	2.230466

[ ]:
```python
print(logit_reg.pvalues)
```

```
Sex          3.284395e-45
Age          3.206039e-07
SibSp        5.871505e-04
1st Class    7.353226e-24
2nd Class    4.110044e-17
3rd Class    2.282220e-09
dtype: float64
```

Toutes les variables considérées dans le nouveau modèle ont des coefficients non nuls à niveau 5%. Indiquant que toutes les variables sont importantes pour la prévision du taux de survie.

1.10 Exercice 10

```
[ ]: ex_man = pd.DataFrame(data={'Sex': [1], 'Age': [22], 'SibSp': [0], '1st Class':␣
     ↪[1], '2nd Class': [0], '3rd Class': [0]})
     ex_woman = pd.DataFrame(data={'Sex': [0], 'Age': [22], 'SibSp': [0], '1st␣
     ↪Class': [1], '2nd Class': [0], '3rd Class': [0]})

     p_man = logit_reg.predict(ex_man)[0]
     p_woman = logit_reg.predict(ex_woman)[0]

     print("Le modele nous dit que:")
     print(f"La probabilité de survie de l'homme décrit par la question 10. est de␣
     ↪{p_man * 100:.2f} %")
     print(f"La probabilité de survie de la femme est de {p_woman * 100:.2f} %")
```

```
Le modele nous dit que:
La probabilité de survie de l'homme décrit par la question 10. est de 60.12 %
La probabilité de survie de la femme est de 95.89 %
```