Devoir Maison 1

1 Théorie

1.1 Estimation par moindres carrées du vecteur $\vec{\beta}$

Exercice 1

On a que $\forall \vec{u} \in \mathbb{R}^d$:

$$J_n(\vec{u}) := ||\vec{Y} - Z\vec{u}||^2$$

Soit $\hat{u} \in \text{argmin}_{\vec{u} \in \mathbb{R}^p} J_n(\vec{u})$. Nous avons que $\hat{u}$ est solution de :

$$\nabla_{\vec{u}} J_n(\vec{u}) = -2Z^t(\vec{Y} - Z\vec{u}) = 0$$

Cela est égale à zéro si et seulement si :

$$Z^t\vec{Y} = Z^t Z\vec{Y}\vec{u} \tag{1}$$

Donc $\hat{u}$ est solution de l'équation 1.

$\square$

Exercice 2

Nous avons que :

$$Z^{\#} = (Z^t Z)^{-1} Z^t$$

Montrons d'abord l'identité :

$$Z^{\#} Z = (Z^t Z)^{-1} Z^t Z = (Z^t Z)^{-1}(Z^t Z) = \mathrm{I}_p$$

Regardons maintenant :

$$H := ZZ^{\#} = Z(Z^t Z)^{-1} Z^t$$

$$H^2 = Z(Z^t Z)^{-1} Z^t Z(Z^t Z)^{-1} Z^t$$

$$H^2 = Z(Z^t Z)^{-1}(Z^t Z)(Z^t Z)^{-1} Z^t$$

$$H^2 = Z(Z^t Z)^{-1} Z^t = H$$

Devoir Maison 1

Donc H est une projection.
Soit $\vec{u} \in \mathbb{R}^n$ un vecteur. Nous avons que:

$$H\vec{u} = Z(Z^tZ)^{-1}Z^t\vec{u} = Z\underbrace{\left[(Z^tZ)^{-1}Z^t\vec{u}\right]}_{\phi\in\mathbb{R}^p}$$

C'est a dire, $H\vec{u} \in \mathrm{Vect}(\vec{Z}^1, \dots, \vec{Z}^p)$. Donc H est le projecteur orthogonal sur l'espace vectoriel engendrée par les colonnes de la matrice Z.

□

Exercice 3

L'estimateur de moindres carrées $\hat{\beta}$ est tel que :

$$\hat{\beta} \in \mathrm{argmin}_{\vec{\beta}\in\mathbf{R}^p} J_n(\vec{\beta})$$

Comme montré dans l'exercice 1, cela est équivalent à :

$$Z^tY = Z^tZ\hat{\beta}$$

Or, comme Z^tZ est une matrice carrée de dimension $p \times p$ inversible, nous avons un système d'équations avec une solution unique pour $\hat{\beta}$.
Donc :

$$\hat{\beta} = (Z^tZ)^{-1}Z^tY = Z^{\#}Y$$

□

Exercice 4

Nous utilisons le fait que les erreurs de régression sont homoscédastiques, donc que $\forall\theta \in \Theta$:

$$\mathbb{E}_\theta[\varepsilon(\theta)] = 0$$

Nous avons que :

$$Y = Z\vec{\beta} + \sigma\varepsilon(\theta)$$

Donc :

$$\mathbb{E}_\theta[Y] = \mathbb{E}_\theta[Z\vec{\beta}] + \sigma\mathbb{E}_\theta[\varepsilon(\theta)] = \mathbb{E}_\theta[Z\vec{\beta}]$$

En utilisant le faite que l'espérance est linéaire et le résultat obtenu dans l'exercice 3 ($\hat{\beta} = Z^{\#}Y$) :

$$Z^{\#}\mathbb{E}_\theta[\vec{Y}] = \mathbb{E}_\theta[Z^{\#}\vec{Y}] = \mathbb{E}_\theta[\hat{\beta}]$$

Donc, en remplaçant dans l'expression et en utilisant la linéarité de l'espérance :

$$\mathbb{E}_\theta[\hat{\beta}] = Z^{\#}\mathbb{E}_\theta[\vec{Y}] = Z^{\#}\mathbb{E}_\theta[Z\vec{\beta}] = \mathbb{E}_\theta[Z^{\#}Z\vec{\beta}] = \mathbb{E}_\theta[\vec{\beta}] = \vec{\beta}$$

Donc $\hat{\beta}$ est un estimateur sans biais.

$\square$

Exercice 5

Comme $\hat{\beta} = Z^{\#}Y$, nous avons que :

$$\mathrm{Var}_\theta(\hat{\beta}) = Z^{\#}(\mathrm{Var}_\theta Y)Z^{\#t}$$

On remarque que :

$$Y = Z\vec{\beta} + \sigma\vec{\varepsilon}(\theta)$$

Comme $\mathrm{Var}_\theta(\vec{\beta}) = 0$, en utilisant que $\mathrm{Var}_\theta(\vec{\varepsilon}(\theta)) = \mathrm{I}_n$:

$$\mathrm{Var}_\theta(Y) = \sigma^2\mathrm{Var}(\vec{\varepsilon}(\theta)) = \sigma^2\mathrm{I}_n$$

Donc :

$$\mathrm{Var}_\theta(\hat{\beta}) = \sigma^2 Z^{\#}Z^{\#t}$$

On remarque aussi que :

$$Z^{\#}Z^{\#t} = (Z^tZ)^{-1}Z^tZ((Z^tZ)^{-1})^t = ((Z^tZ)^{-1})^t$$

Comme Z^tZ est symétrique, nous avons que :

$$Z^{\#}Z^{\#t} = (Z^tZ)^{-1}$$

Donc :

$$\mathrm{Var}_\theta(\hat{\beta}) = \sigma^2(Z^tZ)^{-1}$$

$\square$

Devoir Maison 1

Exercice 6

Démontrons par double implication.

$\boxed{\implies}$

Supposons que $\tilde{\beta}$ est sans biais.

Nous avons que :

$$Y = Z\vec{\beta} + \sigma\vec{\varepsilon}(\theta)$$

Donc :

$$\mathbb{E}_\theta[\vec{\varepsilon}] = 0 \implies \mathbb{E}_\theta[Y] = \mathbb{E}_\theta(Z\vec{\beta}) = Z\vec{\beta}$$

On multiplie par B des deux cotés :

$$\mathbb{E}_\theta(BY) = BZ\vec{\beta}$$

$$\mathbb{E}_\theta(\tilde{\beta}) = BZ\vec{\beta}$$

Comme $\tilde{\beta}$ est sans biais :

$$\vec{\beta} = BZ\vec{\beta}$$

Donc $BZ = \mathrm{I}_p$.

$\boxed{\impliedby}$

On suppose que $BZ = \mathrm{I}_p$, donc :

$$BZ\vec{\beta} = \vec{\beta} \implies B\mathbb{E}_\theta[Z\vec{\beta}] = \vec{\beta}$$

Vu que $\mathbb{E}_\theta[\varepsilon] = 0$, $\mathbb{E}_\theta[Z\vec{\beta}] = \mathbb{E}[Y]$, donc :

$$B\mathbb{E}_\theta[Z\vec{\beta}] = \vec{\beta} \implies \mathbb{E}_\theta[BY] = \vec{\beta} \implies \mathbb{E}_\theta[\tilde{\beta}] = \vec{\beta}$$

Et donc $\tilde{\beta}$ est sans biais, ce qui finit la preuve par double implication.

$\square$

Exercice 7

Nous avons que :

$$\mathbb{E}_\theta[\varepsilon] = 0 \implies \mathbb{E}_\theta[Y] = Z\vec{\beta}$$

Développons l'expression :

$$\mathbb{E}_\theta[(\tilde{\beta} - \vec{\beta})(\hat{\beta} - \vec{\beta})^t] = \mathbb{E}_\theta[\tilde{\beta}\hat{\beta}^t - \tilde{\beta}\vec{\beta}^t - \vec{\beta}\hat{\beta} + \vec{\beta}\vec{\beta}^t]$$

On utilise le fait que $\hat{\beta} = Z^{\#}Y$ et $\tilde{\beta} = BY$:

$$\mathbb{E}_\theta[(\tilde{\beta} - \vec{\beta})(\hat{\beta} - \vec{\beta})^t] = \mathbb{E}_\theta[BYY^tZ^{\#t} - BY\vec{\beta}^t - \vec{\beta}Y^tZ^{\#t}] + \vec{\beta}\vec{\beta}^t$$

$$\mathbb{E}_\theta[(\tilde{\beta} - \vec{\beta})(\hat{\beta} - \vec{\beta})^t] = B\mathbb{E}_\theta[YY^t]Z^{\#t} - B\mathbb{E}_\theta[Y]\vec{\beta}^t - \vec{\beta}\mathbb{E}_\theta[Y^t]Z^{\#t} + \vec{\beta}\vec{\beta}^t \tag{2}$$

On utilise à nouveau que $\mathbb{E}_\theta[\varepsilon] = 0$, ce qui implique que $\mathbb{E}_\theta[Y] = Z\vec{\beta}$:

$$\sigma^2\varepsilon\varepsilon^t = (Y - Z\vec{\beta})(Y^t - \vec{\beta}^tZ^t) = YY^t - Y\vec{\beta}^tZ^t - Z\vec{\beta}Y^t + Z\vec{\beta}\vec{\beta}^tZ^t$$

Comme $\text{Var}_\theta(\varepsilon) = \mathbb{E}_\theta[\varepsilon\varepsilon^t] = \text{I}_n$, nous avons que :

$$\sigma^2\mathbb{E}_\theta[\varepsilon\varepsilon^t] = \sigma^2\text{I}_n = \mathbb{E}_\theta[YY^t] - \mathbb{E}_\theta[Y]\vec{\beta}^tZ^t - Z\vec{\beta}\mathbb{E}_\theta[Y^t] + Z\vec{\beta}\vec{\beta}^tZ^t$$

Vu que $\mathbb{E}_\theta[Y] = Z\vec{\beta}$, nous avons que :

$$\sigma^2\mathbb{E}_\theta[\varepsilon\varepsilon^t] = \sigma^2\text{I}_n = \mathbb{E}_\theta[YY^t] - Z\vec{\beta}\vec{\beta}^tZ^t - Z\vec{\beta}\vec{\beta}^tZ^t + Z\vec{\beta}\vec{\beta}^tZ^t$$

Donc :

$$\mathbb{E}_\theta[YY^t] = Z\vec{\beta}\vec{\beta}^tZ^t + \sigma^2\text{I}_n \tag{3}$$

On utilise 2 et 3 avec le faite que $\mathbb{E}_\theta[Y] = Z\vec{\beta}$:

$$\mathbb{E}_\theta[(\tilde{\beta} - \vec{\beta})(\hat{\beta} - \vec{\beta})^t] = BZ\vec{\beta}\vec{\beta}^tZ^tZ^{\#t} + B\sigma^2\text{I}_nZ^{\#t} - BZ\vec{\beta}\vec{\beta}^t - \vec{\beta}\vec{\beta}^tZ^tZ^{\#t} + \vec{\beta}\vec{\beta}^t$$

Or, on rappelle que $BZ = \text{I}_n$:

$$\mathbb{E}_\theta[(\tilde{\beta} - \vec{\beta})(\hat{\beta} - \vec{\beta})^t] = \vec{\beta}\vec{\beta}^tZ^tZ^{\#t} + \sigma^2BZ^{\#t} - \vec{\beta}\vec{\beta}^t - \vec{\beta}\vec{\beta}^tZ^tZ^{\#t} + \vec{\beta}\vec{\beta}^t$$

$$\mathbb{E}_\theta[(\tilde{\beta} - \vec{\beta})(\hat{\beta} - \vec{\beta})^t] = \sigma^2BZ^{\#t}$$

$$\mathbb{E}_\theta[(\tilde{\beta} - \vec{\beta})(\hat{\beta} - \vec{\beta})^t] = \sigma^2BZ((Z^tZ)^{-1})^t$$

$$\mathbb{E}_\theta[(\tilde{\beta} - \vec{\beta})(\hat{\beta} - \vec{\beta})^t] = \sigma^2((Z^tZ)^{-1})^t$$

Comme Z^tZ est une matrice symétrique, $((Z^tZ)^{-1})^t = (Z^tZ)^{-1}$, donc :

$$\boxed{\mathbb{E}_\theta[(\tilde{\beta} - \vec{\beta})(\hat{\beta} - \vec{\beta})^t] = \sigma^2(Z^tZ)^{-1}}$$

□

Exercice 8

On remarque que comme $\tilde{\beta} = BY$ et $\mathrm{Var}_\theta[Y] = \sigma^2 \mathrm{I}_n$:

$$\mathrm{Var}_\theta(\tilde{\beta}) = \sigma^2 BB^t = \mathrm{Cov}_\theta(\tilde{\beta}, \tilde{\beta})$$

Nous avons aussi que :

$$\hat{\beta} = Z^\# Y \implies \mathrm{Cov}_\theta(\hat{\beta}, \hat{\beta}) = \mathrm{Var}_\theta \hat{\beta} = \sigma^2 Z^\# Z^{\#t} = \sigma^2 (Z^tZ)^{-1} = \mathrm{Cov}_\theta(\tilde{\beta}, \hat{\beta})$$

Nous avons donc que :

$$\mathrm{Var}_\theta(\tilde{\beta} - \hat{\beta}) = \mathrm{Cov}_\theta(\tilde{\beta} - \hat{\beta}, \tilde{\beta} - \hat{\beta}) = \mathrm{Var}_\theta(\tilde{\beta}) + \mathrm{Var}_\theta(\hat{\beta}) - 2\mathrm{Cov}_\theta(\tilde{\beta}, \hat{\beta})$$

On utilise que

$$\mathrm{Cov}_\theta(\tilde{\beta}, \hat{\beta}) = \mathrm{Var}_\theta(\hat{\beta})$$

Donc, comme la matrice de covariance est positive semi-définie :

$$\mathrm{Var}_\theta(\tilde{\beta} - \hat{\beta}) = \mathrm{Var}_\theta(\tilde{\beta}) - \mathrm{Var}_\theta(\hat{\beta}) \geq 0$$

Donc $\forall x \in \mathbb{R}^p$:

$$x^t(\mathrm{Var}_\theta(\tilde{\beta}) - \mathrm{Var}_\theta(\hat{\beta}))x \geq 0 \implies \mathrm{Var}_\theta \tilde{\beta} \succcurlyeq \mathrm{Var}_\theta \hat{\beta}$$

□

1.2 Estimation de la variance σ^2 et Coefficient de détermination

Exercice 9

Nous avons que :

$$\mathbb{E}_\theta[\hat{\sigma}^2] = \frac{1}{n-p}\mathbb{E}_\theta[\mathrm{tr}((Y - Z\hat{\beta})(Y - Z\hat{\beta})^t)] = \frac{1}{n-p}\mathrm{tr}(\mathrm{Var}(Y - Z\hat{\beta}))$$

On remarque que :

$$\mathrm{Var}((\mathrm{I}_p - H)Y) = \sigma^2(\mathrm{I}_p - H)\mathrm{I}(\mathrm{I}_p - H)^t$$

$$\mathrm{Var}((\mathrm{I}_p - H)Y) = \sigma^2(\mathrm{I}_p - H)(\mathrm{I}_p - H)^t$$

$$\mathrm{Var}((\mathrm{I}_p - H)Y) = \sigma^2(\mathrm{I}_p - H - H + H)$$

$$\mathrm{Var}((\mathrm{I}_p - H)Y) = \sigma^2(\mathrm{I}_p - H)$$

On remplace dans l'équation précédente :

$$\mathbb{E}_\theta[\hat{\sigma}^2] = \frac{1}{n-p}\mathrm{tr}(\sigma^2\mathrm{I}_p - \sigma^2 H) = \frac{\sigma^2}{n-p}(\mathrm{tr}\mathrm{I}_p - \mathrm{tr}H)$$

Or, nous avons que $\mathrm{tr}\mathrm{I}_n = n$. Aussi, comme H est une matrice $p \times p$ projection orthonormal nous avons que $\mathrm{tr}H = n$.
Nous avons donc que :

$$\mathbb{E}_\theta[\hat{\sigma}^2] = \sigma^2$$

Exercice 10

Comme H est projection ortogonal $Y = HY + (\mathrm{I}_p - H)Y$, cela implique que :

$$||Y||^2 = ||HY||^2 + ||(I - H)Y||^2$$

Or, comme $\mathrm{SSE} = ||HY||^2$ et $\mathrm{RSS} = ||(\mathrm{I}_p - H)Y||^2$:

$$||Y||^2 = \mathrm{SSE} + \mathrm{RSS}$$

1.3 Cas de la régression linéaire gaussienne

Exercice 11

Nous avons que $Y \sim \mathcal{N}(Z\vec{\beta}, \sigma^2\mathrm{I}_n)$

Écrivons maintenant la log-vraisemblance :

$$p_\theta(Y) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n e^{-\frac{||Y-Z\vec{\beta}||^2}{2\sigma^2}}$$

$$\ln(\vec{\beta}, \sigma^2) = -\frac{n}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}||Y - Z\vec{\beta}||^2$$

Nous avons donc que :

$$\nabla_{\vec{\beta}}\ln(\hat{\beta}_{\mathrm{MV}}, \hat{\sigma}^2_{\mathrm{MV}}) = 0 \implies \hat{\beta}_{\mathrm{MV}} = \mathrm{argmin}_{\phi\in\mathbb{R}^p}||Y - Z\phi||^2 \tag{4}$$

Or, cela est le même résultat que les moindres carrées :

$$\boxed{Z^{\#}Y = \hat{\beta}_{\mathrm{MV}}}$$

Nous avons aussi que :

$$\partial_{\sigma^2}\ln(\hat{\beta}_{\mathrm{MV}}, \hat{\sigma}^2_{\mathrm{MV}}) = 0 \implies -\frac{n}{\hat{\sigma}^2_{\mathrm{MV}}} + \frac{1}{(\hat{\sigma}^2_{\mathrm{MV}})^2}||Y - Z\hat{\beta}_{\mathrm{MV}}||^2 = 0$$

Ce qui implique que :

$$\boxed{\hat{\sigma}^2_{\mathrm{MV}} = \frac{1}{n}||Y - Z\hat{\beta}||^2}$$

Cela est un point de maximum car :

$$\partial^2_{\sigma^2}\ln(\hat{\beta}_{\mathrm{MV}}, \hat{\sigma}^2_{\mathrm{MV}}) = \frac{n}{(\hat{\sigma}^2_{\mathrm{MV}})^2} - \frac{2}{(\hat{\sigma}^2_{\mathrm{MV}})^3}||Y - Z\hat{\beta}||^2 = -\frac{1}{(\hat{\sigma}^2_{\mathrm{MV}})^3}||Y - Z\hat{\beta}||^2 < 0$$

Exercice 12

Sous $\mathbb{P}_\theta$, comme $Y \sim \mathcal{N}(Z\vec{\beta}, \sigma^2\mathrm{I}_n)$ vecteur gaussian, nous avons que :

$$\hat{\beta} = Z^{\#}Y \sim \mathcal{N}(Z^{\#}Z\vec{\beta}, \sigma^2 Z^{\#}Z^{\#t})$$

Or, mais comme déjà démontré, $Z^{\#}Z = \mathrm{I}_n$ et $Z^{\#}Z^{\#t} = (Z^tZ)^{-1}$:

$$\hat{\beta} = Z^{\#}Y \sim \mathcal{N}(\vec{\beta}, \sigma^2(Z^tZ)^{-1})$$

Exercice 13

Par le théorème de Cochran :

$$\frac{||(\mathrm{I}_n - H)Y||^2}{\sigma^2} \sim \chi^2(\dim(\mathrm{I}_n - H))$$

Or, comme H est un projecteur dans un espace de dimension p, $\dim(\mathrm{I}_n - H) = n - p$:

$$n\frac{\hat{\sigma}^2_{\mathrm{MV}}}{\sigma^2} \sim \chi^2(n-p)$$

Nous avons donc que :

$$\hat{\sigma}^2_{\mathrm{MV}} \sim \frac{\sigma^2}{n}\xi \quad \text{où} \quad \xi \sim \chi^2(n-p)$$

Exercice 14

D'après le théorème de Cochram, HY et $(\mathrm{I}_n - H)Y$ sont indépendants. Comme $\rho \mapsto Z\rho$ et $\rho \mapsto \frac{||\rho||^2}{n}$ sont mesurables sur Θ, alors $\hat{\beta} = Z^{\#}Y$ et $\hat{\sigma} = \frac{1}{n}||(\mathrm{I}_n - H)Y||^2$ sont indépendants sur $\mathbb{P}_\theta$.

1.4 Tests statistiques, cas régression linéaire gaussienne

Exercice 15

D'après les résultats précédentes:

$$\hat{\beta} \sim \mathcal{N}(\beta, \sigma^2(Z^tZ)^{-1})$$

Alors, $\forall x \in \mathbb{R}^p$:

$$x^t\beta \sim \mathcal{N}(x^t\beta, \sigma^2 x^t(Z^tZ)^{-1}x)$$

Ce qu'implique:

$$\eta = \frac{x^t(\hat{\beta} - \beta)}{\sigma\sqrt{x^t(Z^tZ)^{-1}x}} \sim \mathcal{N}(0,1)$$

Aussi:

$$\lambda = (n-p)\frac{\hat{\sigma}^2}{\sigma^2} \sim \chi^2(n-p)$$

Devoir Maison 1

Par définition de la Loi T de Student:

$$\frac{x^t(\hat{\beta}-\beta)}{\hat{\sigma}\sqrt{x^t(Z^tZ)^{-1}x}} = \frac{\eta}{\sqrt{\lambda/(n-p)}} \sim T(n-p)$$

Exercice 16

Notons $t_{\alpha}^{(n-p)}$ le α-quantil de la loi T de Student a $n-p$ degrees de liberté. Alors, on écrit:

$$\mathbb{P}_\theta\left(t_{\alpha/2}^{(n-p)} \leq \frac{x^t(\hat{\beta}-\beta)}{\hat{\sigma}\sqrt{x^t(Z^tZ)^{-1}x}} \leq t_{1-\alpha/2}^{(n-p)}\right) = 1-\alpha$$

Alors:

$$\mathbb{P}_\theta\left(x^t\hat{\beta} - t_{1-\alpha/2}^{(n-p)}\Delta \leq x^t\beta \leq x^t\hat{\beta} - t_{\alpha/2}^{(n-p)}\Delta\right) = 1-\alpha$$

Avec $\Delta = \hat{\sigma}\sqrt{x^t(Z^tZ)^{-1}x}$. D'où l'intervalle de confiance à niveau α:

$$\boxed{\mathcal{I}_{1-\alpha} = \left[x^t\hat{\beta} - t_{1-\alpha/2}^{(n-p)}\Delta, x^t\hat{\beta} - t_{\alpha/2}^{(n-p)}\Delta\right]}$$

Ci-dessous nous utiliserons toujours :

$$\Delta := \hat{\sigma}\sqrt{x^t(Z^tZ)^{-1}x}$$

Exercice 17

Dans le cadre des hypothèses H_0 et H_1 de l'énonce On propose le test suivante:

$$\phi_\alpha \colon z \in \mathbb{R}^p \mapsto \mathbf{1}_{\{|z^tx|>c_\alpha\}}$$

Et on calcule son niveau:

$$\bar{\alpha}(\phi_\alpha) = \mathbb{P}_{H_0}(\phi_\alpha = 1) = \mathbb{P}_{H_0}\left(|\hat{\beta}^tx| > c_\alpha\right)$$

$$= \mathbb{P}_{H_0}\left(\left|\frac{\hat{\beta}^tx}{\Delta}\right| > \frac{c_\alpha}{\Delta}\right) = F^{T(n-p)}(c_\alpha/\Delta) - F^{T(n-p)}(-c_\alpha/\Delta)$$

Ou $F^{T(n-p)}$ est la fonction de répartition d'une loi T de Student de $n-p$ dégrées de liberté. D'après la parité de la loi:

$$\bar{\alpha}(\phi_\alpha) = 2(1 - F^{T(n-p)}(c_\alpha/\Delta))$$

En fixant le niveau du test a α:

$$\bar{\alpha}(\phi_\alpha) = \alpha = 2(1 - F^{T(n-p)}(c_\alpha/\Delta))$$

Alors:

$$F^{T(n-p)}(c_\alpha/\Delta) = 1 - \alpha/2$$
$$c_\alpha = \Delta t^{n-p}_{1-\alpha/2} = -\Delta t^{n-p}_{\alpha/2}$$

Alors, le test s'écrit:

$$\phi_\alpha \colon z \mapsto \mathbf{1}_{\{|z^t x| > \Delta t^{n-p}_{1-\alpha/2}\}}$$

Exercice 18

Par définition de la p-valeur:

$$\hat{\alpha}(Z) = \text{arginf}_{\alpha \in [0,1]} \phi_\alpha(Z) = 1$$

Ce que arrive quand $|\hat{\beta}^t x| = c_\alpha$. Alors:

$$\hat{\alpha} = 2\left(1 - F^{T(n-p)}\left(\frac{|\hat{\beta}^t x|}{\Delta}\right)\right)$$

Exercice 19

On sait que

$$\hat{\beta} - \beta \sim \mathcal{N}\left(0, \sigma^2 (Z^t Z)^{-1}\right)$$

Un vecteur gaussien. Alors, comme $rang(A) = q$, on a que:

$$A(\hat{\beta} - \beta) \sim \mathcal{N}\left(0, \sigma^2 A^t (Z^t Z)^{-1} A\right)$$

On rapelle le résultat suivante: Pour une matrice de covariance $\Sigma \in \mathcal{M}_{d\times d}$:

$$X \sim \mathcal{N}(\vec{0}, \Sigma) \implies X^t \Sigma^{-1} X \sim \chi^2(d)$$

Alors:

$$\frac{1}{\sigma^2}\left(A(\hat{\beta} - \beta)\right)^t \left[A(Z^t Z)^{-1} A^t\right]^{-1} \left(A(\hat{\beta} - \beta)\right) \sim \chi^2(q)$$

On rappelle que $(n-p)\hat{\sigma}^2/\sigma^2 \sim \chi^2(n-p)$. Or, comme $\hat{\beta}$ et $\hat{\sigma}^2$ sont indépendantes par Cochran, leurs transformations le sont aussi. Alors, par la définition de la loi de Fisher F:

Devoir Maison 1

$$\frac{\frac{1}{\sigma^2}\left(A(\hat{\beta}-\beta)\right)^t [A(Z^tZ)^{-1}A^t]^{-1}\left(A(\hat{\beta}-\beta)\right)}{(n-p)\hat{\sigma}^2/\sigma^2} \times \frac{n-p}{q} \sim F(q, n-p)$$

Donc:

$$\frac{1}{q\hat{\sigma}^2}\left(A(\hat{\beta}-\beta)\right)^t \left[A(Z^tZ)^{-1}A^t\right]^{-1}\left(A(\hat{\beta}-\beta)\right) \sim F(q, n-p)$$

Exercice 20

Ici on pose:

$$A_{2\times p} = \begin{pmatrix} I_2 \\ 0 \end{pmatrix}$$

Comme Z^tZ est une matrice réelle symétrique, alors elle est diagonalisable sur une base orthonomale. Donc il existe une matrice orthonormal P et une matrice diagonale D tels que:

$$Z^tZ = P^tDP$$

D'après l'exercice 8, $(Z^tZ)^{-1}$ est une matrice de covariance. Alors elle est définie positive, ce qu'implique que les valeurs propres de Z^tZ sont positives. Naturellement $D \geq 0$.
Alors:

$$\Lambda = (A(Z^tZ)^{-1}A^t)^{-1} = (AP^t)D(AP^t)^t$$

Une matrice 2×2 diagonalisable. On note $\{v_1, v_2\}$ la famille de vecteurs propres orthonormale et $\{\lambda_1, \lambda_2\}$ les valeurs propres associés.
On note que $\forall l \in \mathbb{R}_*^+$:

$$\left(A(\hat{\beta}-\beta)\right)^t \Lambda \left(A(\hat{\beta}-\beta)\right) = l$$

$$\left((\beta_1, \beta_2) - (\hat{\beta}_1, \hat{\beta}_2)\right)^t \Lambda \left((\beta_1, \beta_2) - (\hat{\beta}_1, \hat{\beta}_2)\right) = l$$

Décrit une ellipse sur le plan (β_1, β_2), centrée en $(\hat{\beta}_1, \hat{\beta}_2)$ de axis orthogonaux v_1 et v_2
On écrit $f_\alpha^{q,n-p}$ le α-quantil d'une loi de Fisher de paramètres $q, n-p$.
Donc la région de confiance de (β_1, β_2) est une couronne elliptique:

$$\boxed{2\hat{\sigma}^2 f_{\alpha/2}^{2,n-p} \leq \left((\beta_1, \beta_2) - (\hat{\beta}_1, \hat{\beta}_2)\right)^t \Lambda \left((\beta_1, \beta_2) - (\hat{\beta}_1, \hat{\beta}_2)\right) \leq 2\hat{\sigma}^2 f_{1-\alpha/2}^{2,n-p}}$$

In [ ]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
from  matplotlib.patches import Ellipse

from scipy.stats import chi2
from scipy.stats import t as student
from scipy.stats import f as fisher
```

In [ ]:
```
df = pd.read_csv("ozone_complet.txt", sep=';').dropna()
df
```

Out[ ]:

	maxO3	T6	T9	T12	T15	T18	Ne6	Ne9	Ne12	Ne15	...	Vdir9	Vvit9	Vdir12
19950401	47.6	10.1	11.6	13.3	13.6	12.2	8.0	8.0	8.0	8.0	...	290.0	4.0	300.0
19950402	56.2	9.5	9.4	13.8	17.4	16.3	8.0	8.0	7.0	0.0	...	160.0	2.0	180.0
19950403	61.8	3.6	8.0	16.8	21.5	20.2	4.0	5.0	2.0	2.0	...	20.0	2.0	340.0
19950404	50.8	9.5	10.5	11.4	12.2	11.4	8.0	7.0	7.0	7.0	...	10.0	4.0	350.0
19950405	59.8	9.8	10.8	13.8	14.3	13.3	8.0	7.0	8.0	8.0	...	340.0	2.0	280.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
20020926	76.0	7.0	13.7	17.0	17.9	15.3	4.0	7.0	8.0	7.0	...	290.0	4.0	330.0
20020927	79.0	10.2	11.5	18.3	20.0	17.1	9.0	9.0	2.0	6.0	...	70.0	2.0	120.0
20020928	91.0	6.2	14.1	19.8	21.1	18.1	0.0	0.0	1.0	2.0	...	120.0	3.0	120.0
20020929	89.0	8.1	14.8	20.4	22.0	18.4	4.0	5.0	5.0	4.0	...	120.0	4.0	130.0
20020930	99.0	9.4	13.8	21.7	24.7	19.8	6.0	6.0	6.0	3.0	...	140.0	2.0	140.0

1366 rows × 23 columns

1)

In [ ]:
```
Y = np.matrix(df.maxO3v).transpose()
Z = np.matrix([df.T12, df.Vx, df.Ne12, np.ones((Y.size,))]).transpose().astype(float

n = Y.size
p = Z.shape[1]

Z_sharp = (Z.transpose() @ Z) ** -1 * Z.transpose()
beta_hat = Z_sharp * Y

beta_hat
```

Out[ ]:
```
matrix([[ 1.44661518],
        [ 0.88444208],
        [-1.33507028],
        [64.66754543]])
```

In [ ]:
```
epsilon = Y - Z * beta_hat
sigma_hat = np.linalg.norm(epsilon) / np.sqrt(n - p)
```

```
sigma_hat
```

Out[ ]: 20.97181272920036

2)

In [ ]:
```
def one_hot(i, n):
    a = np.zeros((n, 1))
    a[i] = 1
    return a.transpose()

def beta_interval(x_t, beta_hat, alpha):
    delta = sigma_hat * np.sqrt(x_t * ((Z.transpose() @ Z) ** -1) * x_t.transpose())
    quant = student.ppf(1 - alpha / 2, n - p)

    return x_t * beta_hat + delta * np.array([-quant, quant])

def sigma_2_interval(sigma_hat, alpha):
    quant = np.array([1 / chi2.ppf(1 - alpha / 2, n - p), 1 / chi2.ppf(alpha / 2, n
    return sigma_hat ** 2 * (n - p) * quant
```

In [ ]:
```
alpha = .05
beta_intervals = [beta_interval(one_hot(i, p), beta_hat, alpha) for i in range(p)]
beta_intervals
```

Out[ ]:
```
[matrix([[1.18142349, 1.71180687]]),
 matrix([[0.52652976, 1.24235439]]),
 matrix([[-1.93511358, -0.73502697]]),
 matrix([[57.36033557, 71.97475529]])]
```

In [ ]:
```
sigma_interval = np.sqrt(sigma_2_interval(sigma_hat, alpha))
sigma_interval
```

Out[ ]: array([20.21303734, 21.79021216])

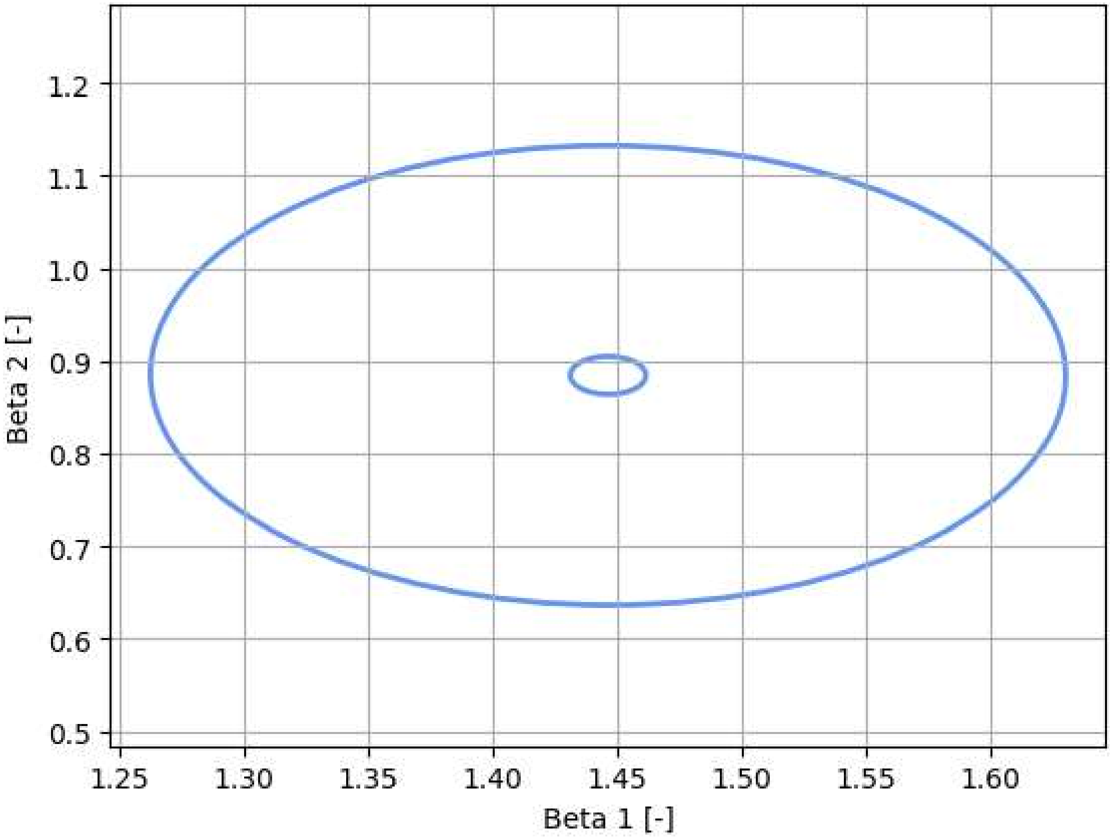
3)

In [ ]:
```
def conf_ellip(A, Z, beta_hat, sigma_hat, alpha, rem_zeros=False):
    q = A.shape[0]

    Z_proj = A @ np.linalg.inv(Z.transpose() @ Z) @ A.transpose()
    if rem_zeros:
        nzero_rows = np.flatnonzero(~np.all(Z_proj == 0, axis=0))
        nzero_cols = np.flatnonzero(~np.all(Z_proj == 0, axis=1))
        Z_proj = Z_proj[nzero_rows,:]
        Z_proj = Z_proj[:, nzero_cols]

    Lambda = np.linalg.inv(Z_proj) / (q * sigma_hat ** 2)
    l, v = np.linalg.eig(Lambda)

    f1 = fisher.ppf(alpha / 2, q, n - p)
    f2 =  fisher.ppf(1 - alpha / 2, q, n - p)

    d1 = np.sqrt(f1 / l)
    d2 = np.sqrt(f2 / l)
```

```
    gamma = np.degrees(np.arctan2(v[0, 1], v[0, 0]))
    ell1 = (d1[0], d1[1], gamma)
    ell2 = (d2[0], d2[1], gamma)

    return (ell1, ell2)
```

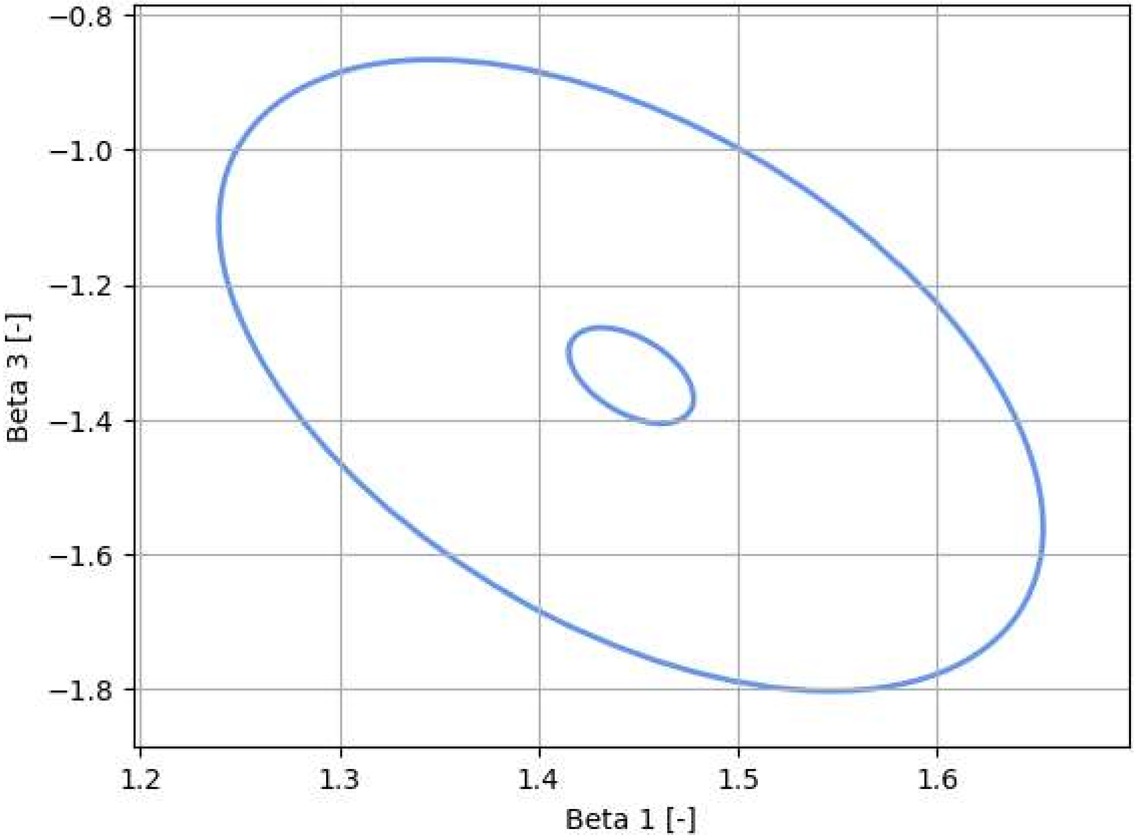
In [ ]:

```
A = np.zeros((2, p))
A[0, 0] = 1
A[1, 1] = 1

ell1, ell2 = conf_ellip(A, Z, beta_hat, sigma_hat, alpha, rem_zeros=False)

small_obj = Ellipse(beta_hat[:2], *ell1)
large_obj = Ellipse(beta_hat[:2], *ell2)

ax = plt.gca()
ax.add_artist(small_obj)
ax.add_artist(large_obj)

small_obj.set_fill(False)
small_obj.set_edgecolor('cornflowerblue')
small_obj.set_linewidth(2)

large_obj.set_fill(False)
large_obj.set_edgecolor('cornflowerblue')
large_obj.set_linewidth(2)

ax.set_xlim(-.2 + beta_hat[0], .2 + beta_hat[0])
ax.set_ylim(-.4 + beta_hat[1], .4 + beta_hat[1])

ax.set_xlabel("Beta 1 [-]")
ax.set_ylabel("Beta 2 [-]")

ax.grid()

plt.show()
```



In [ ]:

```
q = 3
A = np.zeros((q, p))
A[0, 0] = 1
A[2, 2] = 1

ell1, ell2 = conf_ellip(A, Z, beta_hat, sigma_hat, alpha, rem_zeros=True)

small_obj = Ellipse(beta_hat[[0,2]], *ell1)
large_obj = Ellipse(beta_hat[[0,2]], *ell2)

ax = plt.gca()
ax.add_artist(small_obj)
ax.add_artist(large_obj)

small_obj.set_fill(False)
small_obj.set_edgecolor('cornflowerblue')
small_obj.set_linewidth(2)

large_obj.set_fill(False)
large_obj.set_edgecolor('cornflowerblue')
large_obj.set_linewidth(2)

ax.set_xlim(-.25 + beta_hat[0], .25 + beta_hat[0])
ax.set_ylim(-.55 + beta_hat[2], .55 + beta_hat[2])

ax.set_xlabel("Beta 1 [-]")
ax.set_ylabel("Beta 3 [-]")

ax.grid()
plt.show()
```



4)

(i)

La question numéro (i) peut être formulé autrement comme le test d'hypothèse suivant :

$$H0 : \beta_2 = 0, \text{contre } H1 : \beta_2 \neq 0$$

C'est à dire que nous testons si le paramètre qui multiplie Vx est égal à zéro, ce qui indiquerait une non-dépendance de $O3$ par rapport à Vx (une dépendance non-linéaire resterait toujours possible).

(ii)

De même, nous pouvons considérer le test suivant:

$$H0 : \beta_3 = 0, \text{contre } H1 : \beta_3 \neq 0$$

(iii)

Pour vérifier l'influence par Vx ou $T12$ nous pouvons considérer le test suivant:

$$H0 : (\beta_1, \beta_3) = 0, \text{contre } H1 : (\beta_1, \beta_3) \neq 0$$

Où l'hypothèse $H0$ est l'inexistence d'une dépendance lineaire entre $O3$ et (Vx, $T12$)

5)

La procédure pour les tests (i) et (ii) découle directement de la formule de l'item 17.

Pour le test (iii) nous pouvons utiliser la dualité entre les tests d'hypothèse et les intervalles de confiance. C'est-à-dire, nous vérifions si la région de confiance de (β_1, β_3), à niveau α égal à celui du test désiré, contient l'origine.

In [ ]:
```
def test_beta(x, alpha, beta_hat, sigma_hat, Z):
    x_t = x.transpose()
    delta = sigma_hat * np.sqrt(x_t * ((Z.transpose() @ Z) ** -1) * x)
    quant = student.ppf(1 - alpha / 2, n - p)

    return np.abs(beta_hat.transpose() * x) > delta * quant
```

In [ ]:
```
def p_valeur(x, beta_hat, sigma_hat, Z):
    x_t = x.transpose()
    delta = sigma_hat * np.sqrt(x_t * ((Z.transpose() @ Z) ** -1) * x)

    return 2 * (1 - student.cdf((np.abs(beta_hat.transpose() @ x))/(delta), n - p))
```

(i)

In [ ]:
```
if test_beta(one_hot(1, p).transpose(), alpha, beta_hat, sigma_hat, Z):
    print("H0 rejetée.")
    print("p-valeur = " + str(p_valeur(one_hot(1, p).transpose(), beta_hat, sigma_ha
else:
    print("H0 acceptée.")
    print("p-valeur = " + str(p_valeur(one_hot(1, p).transpose(), beta_hat, sigma_ha
```

```
H0 rejetée.
p-valeur = 1.393017828643650e-06
```

(ii)

In [ ]:
```
if test_beta(one_hot(2, p).transpose(), alpha, beta_hat, sigma_hat, Z):
    print("H0 rejetée.")
    print("p-valeur = " + str(p_valeur(one_hot(2, p).transpose(), beta_hat, sigma_ha
else:
    print("H0 acceptée.")
    print("p-valeur = " + str(p_valeur(one_hot(2, p).transpose(), beta_hat, sigma_ha
```

```
H0 rejetée.
p-valeur = 1.3690449021552809e-05
```

(iii)

Le deuxième graphique de la question pratique 3 nous montre que l'origine n'est pas comprise dans la région de confiance de (β_1, β_3). Alors, nous rejetons $H0$.

6)

En conclusion, les tests montrent que les facteurs Vx, $T12$ et $N12$ influencent la valeur de $O3$.

Bien sur, ces résultats sont plausibles sous l'hypothèse que $O3$ suit un modèle gaussien linéaire dans les variables explicatives. En considérant des autres types de modèles plus complexes on pourrait arriver a des autres résultat.