Ce document contient des fonctions R à maîtriser pour les tests et l'examen. Pour plus d'informations et détails, consultez le menu aide de RStudio. Le document sera mis à jour et complété lors du quadrimestre.

Résumé statistiques des données

Soit un vecteur `x` de longueur n.

- `mean(x)`: moyenne des valeurs du vecteur.
- `var(x)`: variance des valeurs du vecteur (avec dénominateur $n-1$, ce qui correspond à $(S')^2$ dans la notation du syllabus).
- `sd(x)`: écart-type des valeurs du vecteur (avec dénominateur $n-1$, ce qui correspond à S' dans la notation du syllabus).
- `quantile(x,p)`: quantile d'ordre p de x, avec $p \in (0,1)$. Si p est omis, la sortie donne des quartiles: $p = 0\%, 25\%, 50\%, 75\%, 100\%$.

Distribution normale

Considérons une loi normale $N(\mu, \sigma^2)$ d'espérance μ et de variance σ^2.

- `pnorm(q, mean =` μ`, sd =` σ`)` : fonction de répartition en q de la loi normale.
- `qnorm(p, mean =` μ`, sd =` σ`)` : quantile d'ordre p pour la loi normale.
- `rnorm(n, mean =` μ`, sd =` σ`)` : génération de n nombres aléatoires suivant la loi normale.

Remarques :

Les fonctions `qnorm` et `pnorm` contiennent également l'argument `lower.tail`.
Si `lower.tail = TRUE`, la probabilité considérée est $P(X \leq x)$.
Si `lower.tail = FALSE` : $P(X > x)$

Quand vous ne précisez pas de valeurs pour les arguments `mean`, `sd` et `lower.tail`, les valeurs par défaut suivantes sont directement attribuées :
`mean = 0`, `sd = 1`, `lower.tail = TRUE`.

`?fonction` : Commande permettant d'ouvrir le menu aide d'une fonction R.
Exemple : `?pnorm`

Distribution χ^2

Considérons une loi chi-carré avec df degrés de liberté.

- `pchisq(q, df)` : fonction de répartition en q de la loi χ^2.
- `qchisq(p, df)` : quantile d'ordre p ($0 \leq p \leq 1$) pour la loi χ^2.
- `rchisq(n, df)` : génération de n nombres aléatoires suivant la loi χ^2.

Les fonctions `pchisq` et `qchisq` contiennent également l'argument `lower.tail`.
Par défaut, `lower.tail = TRUE`.

Quand `lower.tail = TRUE` la probabilité considérée est $P(X \leq x)$.
Dans ce cas, la fonction `qchisq` donne le quantile inférieur.

Quand `lower.tail = FALSE` : $P(X > x)$
Dans ce cas, la fonction `qchisq` donne le quantile supérieur.

Exemple : Le quantile supérieur d'ordre $\alpha = 5\%$ d'une distribution $\chi^2_{df=1}$ se calcule en R via :
`qchisq(p = 0.05, df = 1, lower.tail = FALSE)` $\rightarrow$ Cela donne 3.841

Remarque : Afin d'obtenir les mêmes quantiles que ceux donnés dans la table chi-carré disponible sur Moodle, vous devez calculer les quantiles supérieurs et donc utiliser `lower.tail = FALSE`.

Régression linéaire

La fonction `lm` permet d'ajuster des modèles linéaires dans R.

Supposons Y la variable dépendante, X la variable explicative et ε une variable aléatoire de moyenne 0 et de variance σ^2.
Pour ajuster le modèle linéaire simple $Y = \beta_0 + \beta_1 X + \varepsilon$, on utilise dans R :

```
lm(y ~ x)
```

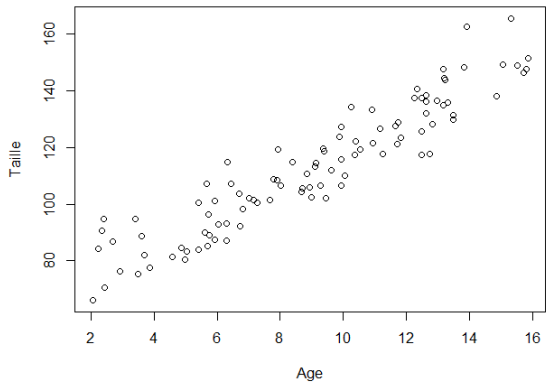
où `y` et `x` représentent des vecteurs d'observations pour les variables aléatoires dépendante et indépendante respectivement.

Pour obtenir un résumé détaillé des résultats de l'ajustement, il faut sauver les valeurs retournées par `lm` dans un objet R appelé `m`, par exemple. Ensuite on applique la fonction `summary` sur l'objet `m`.

```
m = lm(y ~ x)
summary(m)
```

Exemple et interprétation des résultats

Pour 100 garçons (`n=100`), nous souhaitons modéliser leur taille en fonction de leur âge. Les vecteurs `y` et `x` contiennent donc les tailles (en cm) et les âges de 100 garçons. Ces données sont représentées sur le graphe ci-dessous.



Voici les résultats obtenus avec R grâce aux commandes `lm` et `summary`.

```
> m = lm(y ~ x)
> summary(m)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-16.055  -6.354  -1.094   5.434  22.487

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   62.263      2.095   29.72   <2e-16 ***
x              5.610      0.215   26.09   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.944 on 98 degrees of freedom
Multiple R-squared:  0.8741,    Adjusted R-squared:  0.8729
F-statistic: 680.7 on 1 and 98 DF,  p-value: < 2.2e-16
```

Dans la partie `coefficients`, la colonne `Estimate` nous donne $\hat{\beta}_0 = 62.263$ et $\hat{\beta}_1 = 5.610$.

Les deux dernières colonnes donnent les informations concernant les tests statistiques réalisés individuellement sur les coefficients β_0 et β_1. En détail (voir slide 7-18),

▷ La colonne `t value` donne la valeur observée de la statistique de test T avec $\beta_{i0} = 0$, càd, l'hypothèse nulle du test est $H_0 : \beta_i = 0$.

▷ La colonne `Pr(>|t|)` donne les p-valeurs associées à chacun des tests. Si la probabilité est suffisamment faible, nous pouvons rejeter H_0 selon laquelle le coefficient β_i vaut 0.

La partie `residuals` donne les quartiles des résidus. Les résidus sont définis par $e = y - \hat{\beta}_0 + \hat{\beta}_1 x$. La commande `resid(m)` permet d'afficher le vecteur des 100 résidus.

`Multiple R-squared` correspond au R^2 et vaut 0.874. On peut conclure que l'ajustement du modèle aux données est de bonne qualité.

Le modèle ajusté obtenu est donc $\hat{Y} = 62.263 + 5.610\ x$ et est représenté par la droite rouge ci-dessous.