université de BORDEAUX

Bordeaux INP AQUITAINE

Mémento R sur les Statistisques et la MSP

Table des matières

Florent ARNAL
florent.arnal@u-bordeaux.fr

I Gestion de données

I.1 Effacer la mémoire

```
rm( list=ls() )
```

I.2 Importation des données

Il faut, dans un premier temps, créer un fichier sur Excel et l'enregistrer au format csv.
En outre, le répertoire peut être changé en passant par :
Session / Set Working Directory / Choose Directory ...
On peut également utiliser **Import Dataset**. Les commandes R pour importer et visualiser les données sont alors :

```
donnees = read.csv2("nom.csv", sep=";", dec="...", header=TRUE, stringsAsFactors = TRUE)
```

I.3 Création d'un vecteur ou d'une matrice

```
x = c( ... )
M = matrix(data=... , nrow=... , ncol=... , byrow=... )
```

I.4 Création d'un vecteur associé à une séquence

```
y=seq(from= ... , to=... , by=... )
```

I.5 Classement des éléments d'un vecteur par ordre croissant (resp. décroissant)

```
sort(vecteur)
sort(vecteur, decreasing = TRUE)
```

I.6 Nature d'un objet (Structure)

```
str( ... )
```

I.7 Définition de facteurs et variables

```
Facteur = as.factor(Facteur)
Variable = as.numeric(Variable)
ou
donnees = transform(donnees, Facteur = as.factor(Facteur))
```

I.8 Sélection partielle de données

```
subset(Objet étudié, Modalité == " Modalité à considérer ")
subset(Objet étudié, Modalité1 == ... & Modalité2 == ...)
```

II Statistiques descriptives

II.1 Calculs de moyenne, quantiles, variance et écart-type (estimés)

```
mean(...)
quantile(x=..., p=0.75)    # renvoie le quantile d'ordre 0,75 (3ème quartile)
var(...)
sd(...)
```

II.2 Représentations graphiques

1. Partage de la fenêtre graphique
   ```
   par( mfrow=c(..., ...) )
   ```
   Pour revenir à un affichage "classique", on peut utiliser
   ```
   dev.off()
   ```

2. Choix "automatique" de couleurs :
   ```
   terrain.colors(n=...) ou rainbow(n=...)
   ```

3. Création d'un boxplot :
   ```
   boxplot(x, y, main = "...", names=c("...", "..."), col=c("...","..."))
   ```

4. Création d'un histogramme :
   ```
   hist(x, col="...", main="...")
   ```

5. Création d'un diagramme en barres :
   ```
   barplot(x, col="...", main="...")
   ```

6. Création d'un diagramme de Pareto :
   Il est nécessaire de charger le package "qcc".
   ```
   effectifs <- c(26, 13, 35, 24, 20, 52, 5, 2)
   names(effectifs) <- c("Machine1", ...)
   pareto.chart(effectifs, ylab = "ECC")
   ```

II.3 Régression linéaire

1. Recherche des coefficients de corrélation et de l'ajustement affine du type $y = ax + b$
   ```
   # Coefficient de corélation
   cor(x, y) # Modele linéaire (lm = linear model)
   modele = lm(y ~ x)
   modele
   # Récupération des coefficients a et b
   b = modele$coefficients[1] # intercept = ordonnée à l'origine
   a = modele$coefficients[2]
   ```

2. Nuage de points et droite d'ajustement d'une série (x, y)
   ```
   plot(y ~ x)
   abline(modele)
   ```

3. Résidus et valeurs prédites (pour $x = 7$)

```
resid(lm( y ~ x ))
predict(lm( y ~ x ), newdata=data.frame(x=c(7)))
```

III Calculs de probabilité

Pour chaque loi, on associe :

• La fonction de densité de probabilité pour les lois continues ou la probabilité d'un évènement élémentaire du type $\mathbb{P}(X = k)$ pour une loi discrète : préfixe d.

• La fonction de répartition : préfixe p.
On rappelle que la fonction de répartition F_X associée à une variable X est définie par :

$$F_X(x) = \mathbb{P}(X \leq x)$$

• La fonction quantile qui est la fonction "réciproque" de la fonction de répartition : préfixe q.
Ainsi, dans le cas d'une loi continue, si $F_X(x) = p$ avec F_X strictement croissante sur $\mathbb{R}$ alors

$$x = {F_X}^{-1}(p)$$

• La génération de nombres pseudo-aléatoires : préfixe r.

III.1 Exemples de lois continues

1. Lois **normales** : [dpqr]norm(... , mean = . . . , sd = ...)
   Supposons que X suit la loi $\mathcal{N}(100; 5)$.
   La probabilité $\mathbb{P}(X \leq 105)$ est donnée par :

   ```
   > pnorm(105, mean=100, sd=5)
   ```

   La valeur x telle que $\mathbb{P}(X \leq x) = 0,975$ est donnée par :

   ```
   > qnorm(0.975, mean=100, sd=5)
   ```

2. Lois de **Student** : [dpqr]t(... , df)
   df correspond aux degrés de liberté (degrees of freedom) de la loi de Student considérée.

3. Lois de **Fisher** : [dpqr]f(... , df1 , df2)
   df1 et df2 correspondent aux degrés de liberté de la loi de Fisher considérée.

4. Lois uniformes : [dpqr]unif(... , min = . . . , max = ...)

5. Lois du Khi-deux : [dpqr]chisq(... , df = ...)

III.2 Exemples de lois discrètes

1. Lois **binomiales** : [dpqr]binom(... , size = ... , prob=...)
   Soit X une variable aléatoire distribuée selon la loi binomiale $\mathcal{B}(10; 0.5)$.
   $\mathbb{P}(X = 2)$ et $\mathbb{P}(X \leq 2)$ s'obtiennent en utilisant :

   ```
   > dbinom(2, size=10, prob=0.5)
   > pbinom(2, size=10, prob=0.5)
   ```

2. Lois de **Poisson** : [dpqr]pois(... , lambda=...)

IV Tests statistiques et intervalles de confiance

IV.1 Introduction

Dans tout ce qui suit, il est à noter que lors des différents tests statistiques, R renvoie toujours les informations suivantes :
* La réalisation de la variable de décision (associée à la loi)
* La probabilité de rejeter l'hypothèse $\mathcal{H}_0$ à tort (notée p-value).

En outre, pour la plupart des tests, il est possible de choisir la nature du test.
Ainsi, on a le choix entre différentes alternatives :
* alternative="two.sided" pour un test bilatéral
* alternative="less" ou "greater" pour un test unilatéral.

Enfin, les commandes de la plupart des tests paramétriques permettent d'obtenir un intervalle de confiance associé au paramètre "testé".

Enfin, il est possible d'extraire pour un test que nous nommerons TEST un certain nombre d'informations relatives à un test telles que :

```
* TEST$statistic : la statistique du test (réalisation de la variable de décision)

* TEST$parameter : le nombre de degrés de liberté

* TEST$p.value : la probabilité du test (p-value)

* TEST$conf.int : l'intervalle de confiance associé (à 95 % par défaut)

* TEST$null.value : une information sur l'hypothèse nulle

* TEST$alternative : une chaîne de caractères spécifiant l'hypothèse alternative

* TEST$method : une chaîne de caractères indiquant le type de test utilisé

* TEST$data.name : une chaîne de caractères précisant les données utilisées.
```

IV.2 Tests du Khi-deux et Fisher exact

Pour un test d'indépendance ou d'homogénéité, il faut créer un tableau sous R (nrow correspond au nombre de lignes, ncol à celui de colonnes et byrow=TRUE indique que les données ont été introduites "en ligne") avant de réaliser le test.

chisq.test( )
fisher.test( )

```
> tableau=matrix(c(...), nrow=..., ncol=..., byrow=TRUE)
> chisq.test(x=tableau)
> fisher.test(x=tableau)
```

Pour un test d'adéquation à une loi théorique, il faut définir deux vecteurs associés aux effectifs (observés) et probabilités associées puis utiliser :

```
> chisq.test(x=..., p=...)
```

IV.3 Tests de conformité et intervalles de confiance

1. Autour d'une proportion

   binom.test(x=..., n=...., p=..., alternative="...")
   prop.test(x=..., n=..., p=..., alternative="...")

2. Autour d'une moyenne

   t.test(x=..., mu = ..., alternative="...")

3. Autour d'une variance

   library(EnvStats)
   varTest(x=..., sigma.squared = ..., alternative="...")

IV.4 Tests paramétriques de comparaisons

1. Tests de comparaison de deux proportions (cas des grands échantillons)

   prop.test(c(...,...),c(...,...),alternative="....")

   Les deux premiers arguments à compléter correspondent respectivement aux effectifs observés (pour chaque échantillon) et aux tailles d'échantillons (sous la forme de vecteurs).

2. Tests de comparaison de deux variances

   var.test(x=..., y=..., alternative="...")

3. Tests de comparaison de deux moyennes

   (a) Echantillons indépendants et variances égales

   t.test(x=..., y=..., alternative="...", var.equal=TRUE)

(b) Echantillons indépendants et variances différentes

```
t.test(x=..., y=..., alternative="...", var.equal=FALSE)
```

R propose de mettre en place un test de Welch à l'aide de la commande t.test en précisant "var.equal=FALSE".
Dans le cas d'un test de comparaison, la commande t.test appelle par défaut ce test.

(c) Echantillons appariés

```
t.test(x=..., y=..., alternative="...", paired=TRUE)
```

IV.5 Autour de la normalité d'une distribution

Pour s'assurer qu'une distribution est normale, il peut être utile de recourir à un graphique Quantite Quantile. On peut également utiliser un test de Shapiro.

```
qqnorm(...)
shapiro.test(...)
```

IV.6 ANOVA à 1 facteur

Dans ce qui suit, on note la variable "var" et le facteur "Facteur".

1. Représentation par modalité à l'aide d'un boxplot

   ```
   plot(var ~ Facteur)
   ```

2. Résumé numérique

   ```
   tapply(var, Facteur, summary)
   ```

   Les moyennes seules pouvant être obtenues avec :

   ```
   tapply(var, Facteur, mean)
   ```

3. Anova (Modèle linéaire)

   ```
   aov(var ~ Facteur)
   ```

4. Graphiques diagnostiques

   ```
   par(mfrow=c(2,2))
   anova=aov(var ~ Facteur)
   plot(anova)
   ```

5. Calculs des résidus

   ```
   residus=resid(anova)
   ```

6. Résultats de l'ANOVA

   ```
   summary(anova)
   ```

7. Détermination des effets de l'ANOVA

   ```
   model.tables(anova)
   ```

IV.7 Test de comparaisons multiples

1. Comparaisons multiples de variances

```
bartlett.test(var ~ Facteur)

library('lawstat')
levene.test(var, Facteur)
```

2. Comparaisons multiples de moyennes

```
pairwise.t.test(x=var , g=Facteur , p.adjust.method="bonferroni")

library('agricolae')
nom = SNK.test( anova, facteur,group = T)
nom
```

IV.8 Tests basés sur les rangs

1. Test de Wilcoxon

```
wilcox.test(x=... , y=..., alternative= "...", paired= ...)
```

2. Test de Kruskal-Wallis

```
kruskal.test(x=var, g=Facteur)
```

3. Test de comparaisons multiples de moyennes

```
pairwise.wilcox.test(x=var, g=Facteur, p.adjust.method=... )
```

IV.9 ANOVA à 2 facteurs

Dans ce qui suit, on note la variable "var" et les facteurs "A" et "B".

1. Cas d'une Anova sans répétition

```
anova = aov(var ~ A+B)
```

2. Cas d'une Anova avec répétitions

```
anova = aov(var ~ A*B)
```

Les graphiques d'interactions s'obtiennent à l'aide de :

```
interaction.plot(A, B, var)
```

IV.10 Puissance d'un test

1. Test de Student (de conformité ou de comparaison de 2 échantillons de même effectif)

```
power.t.test(n = ..., delta = ..., sd = ..., power = NULL,
type = c("two.sample","one.sample", "paired"), alternative = c("two.sided", "one.sided") )
# NULL définit le paramètre dont la valeur sera déterminée par R.
# par défaut, alpha=0.05 (sig.level=0.05)
```

2. Anova monofactorielle

```
power.anova.test(groups =..., n =... , between.var =... , within.var =... , power =... )
# between.var est associé à la variabilité inter ou factorielle (”variance des effets”).
# within.var est associé à la variabilité intra (”variance résiduelle”).
```

V Cartes de contrôle de la moyenne

Il est nécessaire de charger le package ”qcc”.

1. Saisie des valeurs dans un vecteur (nommé valeurs ici) et des numéros d'échantillons

```
> # Création d'un vecteur de 20 échantillons de 5 répétitions
> valeurs = c(...)
> echantillon = rep((1:20), each = 5)
```

On peut également importer un fichier csv.

2. Création de la carte de contrôle avec un cible

```
> masses <- qcc.groups(valeurs, echantillon)
> cible = 125
> carte = qcc(masses, center=cible,type="xbar")
```

3. Création de la carte de contrôle avec les limites de surveillance

```
> LS <- limits.xbar(center=cible, std.dev=carte$std.dev, sizes=carte$sizes, conf=2)
> plot(carte, restore.par = FALSE)
> abline(h = LS, lty = 5, col = "red")
```