Nathalie Villa-Vialaneix

Livret de TP de

Statistique Descriptive I (M1102)

Année scolaire 2013/2014

Université de Perpignan Via Domitia, IUT
STatistique et Informatique Décisionnelle (STID)

Table des matières

1 Introduction à

 est un logiciel libre que vous pouvez installer gratuitement sur votre ordinateur personnel si vous en possédez un. Le logiciel est téléchargeable à `http://cran.univ-paris1.fr/`. Le logiciel est fourni avec des packages que vous pouvez installer par la commande : `install.packages("Rcmdr")` pour le package `Rcmdr` par exemple. Un package, une fois installé, se charge à chaque nouvelle session R, avec : `library("Rcmdr")`.

Avant propos : Récupérez sur mon site web `http://www.nathalievilla.org` (Enseignements / IUT STID Carcassonne / Statistique descriptive) le fichier nommé `desbois.csv` qui correspond aux données de l'interrogation 1. Créez sur le bureau un dossier nommé "TP-R-*MONNOM*" (où *MONNOM* est à remplacer par votre nom) et, à l'intérieur de celui-ci, un fichier nommé "TP1". Collez le fichier `desbois.csv` dans ce dernier dossier. Ouvrez l'application "Terminal" et tapez R puis, une fois le programme lancé, `library(Rcmdr)`. À la fin du TP, **sauvegardez vos données** sur une clé USB ou en vous les envoyant par courriel. En cas de doute ou de problème, demandez-moi une sauvegarde.

L'interface "Commander" de  se présente comme sur la Figure 1.1.

La fenêtre de script affiche les commandes que vous exécutez au moyen du menu dans le langage de programmation de R : elles peuvent être modifiées à la main et exécutées à nouveau pour en personnaliser certains aspects. L'intégralité des commandes exécutées (le script) peut être enregistré sous la forme d'un fichier texte.

La fenêtre de sortie permet de visualiser les résultats des commandes demandées. Les sorties peuvent aussi être enregistrées sous la forme d'un fichier texte.

La fenêtre de messages gère l'affichage des messages du programme : elle est notamment utile en cas d'erreur car elle donne des indications sur la cause de celle-ci.

Le bouton "Données" précise quel est le jeu de données courant. Pour visualiser l'ensemble des jeux de données disponibles dans l'espace de travail et pour en changer, il suffit d'appuyer sur ce bouton. L'enregistrement de l'espace de travail enregistre tous les jeux de données et toutes les variables créées durant la session.

Sous Windows, le retour à la fenêtre principale de R (cf Figure 1.2) est nécessaire pour visualiser les figures produites par le programme. Sous Linux, les figures s'affichent dans une fenêtre indépendante.

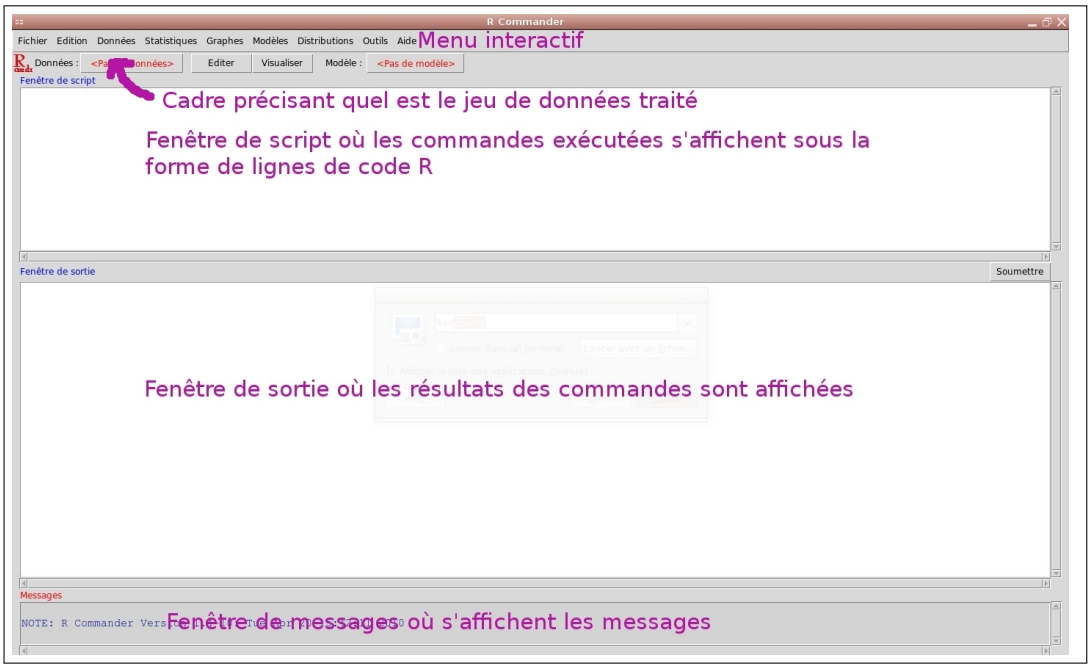

FIGURE 1.1: Interface "Commander" de R

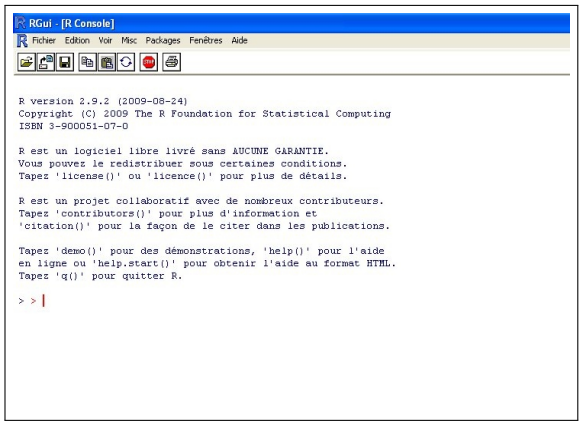

FIGURE 1.2: Interface de R sous Windows

1.1 Importer/sauvegarder des données avec R

1. **Changer le répertoire de travail**
   Pour que tous vos enregistrements soient effectués dans le répertoire "TP1" que vous venez de créer, il faut que ce répertoire soit défini comme répertoire par défaut. Pour cela, utilisez le menu "Fichier → Changer le répertoire de travail...". La commande R correspondante est `setwd("LECHEMIN")` où "LECHEMIN" est le chemin correspondant au répertoire de travail que l'on veut définir par défaut.
2. **Importer un fichier texte**
   Ouvrir le fichier `desbois.csv` avec un éditeur de texte (Notepad, par exemple). Comment se présente-t-il ? En particulier, comment sont séparés les variables ? Comment sont séparées les observations ? Quelle est la marque des décimales (un point ou une virgule) ? À quoi correspond la première ligne ?
   À l'aide du menu "Données → Importer des données → Depuis un fichier texte, le presse papier ou un lien URL..." rempli correctement, importez le fichier `desbois.csv` sous le nom "donnees" (Attention ! Évitez les accents et interdisez les espaces dans les noms de variables !). Si le fichier est importé correctement, la fenêtre de messages affiche "NOTE : Le jeu de données donnees a 1260 lignes et 31 colonnes." et le bouton de données contient désormais "donnees".
   La commande R correspondante est

   ```
   donnees<-read.table("desbois.csv",header=T,sep=",",na.strings="NA",dec=".")
   ```

   où la fonction `read.table` lit le fichier texte `desbois.csv` du répertoire par défaut avec les options `header=T` (les noms des variables sont sur la première ligne contrairement à header=F), `sep=","` (les variables sont séparées par des virgules), `na.strings="NA"` (les valeurs manquantes seront codées NA) et `dec="."` (les décimales sont indiquées par un point).
3. **Visualiser et modifier les données importées**
   Les données importées peuvent être visualisées au moyen du bouton "Visualiser". Les données importées peuvent être modifiées à la main au moyen du bouton "Éditer". Par exemple, modifier le nom de la variable CNTY en DPT (département).
4. **Sauvegarder un jeu de données au format R**
   La commande "Fichier → Sauver l'environnement R..." permet de sauvegarder l'intégralité des données importées ou créées. Sauvegarder votre environnement de travail sous le nom `TP1.Rdata`.
5. **Sauvegarder le script au format R**
   La commande "Fichier → Sauver le script ..." permet de sauvegarder le script au format texte. Sauvegarder le script sous le nom `TP1.R`.

Les fonctions R à retenir

Fonction	Usage	Options
`read.table`	Lire un fichier texte	`header` (T/F) si les noms de variables sont dans le fichier `sep` séparateur de variables `na.strings` marque de valeurs manquantes `dec` signe des décimales
`setw`	Changer le répertoire de travail	

Pour aller plus loin

1. **Ouvrir un fichier de données au format R**
   La commande "Données / Charger un jeu de données" permet de charger dans l'environnement de travail un jeu de données au format R.
   La commande R correspondante est `load("TP1.RData")` où `TP1.RData` est le fichier à charger à partir du répertoire par défaut (pour des fichiers situés ailleurs, il faut préciser le chemin complet).
2. **Importer d'autres types de fichiers avec R**
   Le menu "Données → Importer des données" permet d'importer des données qui sont dans d'autres format comme des tableurs Excel, des données au format SPSS, etc.
3. **Sauvegarder au format R**
   Pour sauvegarder uniquement les données courantes (et non l'intégralité de l'environnement de travail), il faut utiliser le menu "Données → Jeu de données actif → Sauver le jeu de données actif ...".
   La commande R correspondante est `save("donnees",file="Desbois.RData")` qui enregistre les données "donnees" dans le fichier `Desbois.RData` (dans le répertoire par défaut). On peut enregistrer plusieurs données dans le même fichier en utilisant la fonction `save` et en séparant les noms des données à sauvegarder par une virgule.
4. **Exporter des données au format texte**
   Le menu "Données → Jeu de données actif → Exporter le jeu de données actif ..." permet d'exporter le jeu de données actif au format texte. La commande correspondante pour créer un fichier texte identique à `desbois.csv` à partir des manipulations précédentes est

   ```
   write.table(donnees,"desbois2.csv",sep=",",col.names=T,row.names=F,na="")
   ```

   où la fonction `write.table` exporte un fichier de données (ici "donnees") dans un fichier texte (ici `desbois2.csv`) avec les options `sep=","` (les variables sont séparées par une virgule), `col.names=T` (les noms des variables sont inclus dans le fichier texte), `row.names=F` (les noms des observations ne sont pas inclus dans le fichier texte) et `na=""` (les valeurs manquantes seront indiqués par un vide.
5. **Ouvrir et exécuter un script**
   Le menu "Fichier → Ouvrir un script" permet d'ouvrir un script enregistré

préalablement dans la fenêtre de script. Pour l'exécuter, il suffit de sélectionner les lignes à exécuter et d'appuyer sur le bouton "Soumettre" (à droite).

6. **Connaître le contenu de l'environnement de travail**
   Pour connaître l'ensemble des fichiers de données et des variables présentes dans l'environnement de travail, il suffit de taper, dans la fenêtre de script, `ls()` et de soumettre. Les noms de toutes les variables s'affichent alors dans la fenêtre de sortie.
   Par exemple, `save(list=ls(),file="TP1.RData")` sauve tout l'environnement de travail dans le fichier `TP1.RData` comme la commande "Fichier → Sauver l'environnement R...".

1.2 Manipulations élémentaires des données avec R

1. **Définir des noms de ligne à partir d'une variable**
   À quoi correspond la première colonne de données ?
   À partir du menu "Données → Jeu de données actif → Nom des cas", utilisez cette première colonne comme nom des lignes. Visualiser le résultat.
   Les commandes R générées par ce menu sont : `row.names(donnees) <- as.character(donnees$X)` où la fonction `row.names` renvoie les noms des lignes et la fonction `as.character` transforme une variable numérique en une variable texte (les noms de lignes sont nécessairement de type texte). `donnees$X` est utilisée pour désigner la variable "X" du tableau de données "donnees". Une seconde commande R est générée : `donnees$X <- NULL` qui supprime la variable "X" du jeu de données "donnees".

2. **Recoder une variable numérique en un facteur**
   La variable "DIFF" indique si l'entreprise agricole a connu un incident de paiement (valeur 2) ou non (valeur 1). Nous allons recoder celle-ci avec des noms plus explicites : "incident" et "sain". La commande "Données → Gérer les variables dans le jeu de données actif → Convertir des variables numériques en facteur..." donne accès à une boîte de dialogue où l'on peut choisir la variable à recoder : les niveaux (modalités) seront codés par des noms et l'on garde le même nom de variable qu'initialement.
   La commande R générée est `donnees$DIFF <- factor(donnees$DIFF,labels=c('sain','incident'))` qui se comprend ainsi : la fonction `factor` convertit une variable en une variable de type "facteur" (qualitative) dont les modalités sont définies par l'option `labels` par un vecteur de valeurs successives (dans l'ordre des nombres à recoder) que l'on rentre grâce à la fonction `c` (concaténer).
   Convertir la variable DPT en une variable de type "facteur" mais en conservant les numéros de département comme code. Faites une modification similaire pour toutes les autres variables qualitatives codées numériquement.

Les fonctions R à retenir

Fonction	Usage	Options
`as.character`	Transforme une valeur en texte (variables)	
`c`	Concatène des valeurs en un vecteur	
`col.names`	Renvoie les noms des colonnes d'un jeu de données (variables)	
`factor`	Transforme une variable en une variable de type "facteur"	`labels=c(...` noms des modalités
`row.names`	Renvoie les noms des lignes d'un jeu de données (observations)	

Par ailleurs, retenir que `DDD$VVV` désigne la variable "VVV" du fichier de données "DDD".

Pour aller plus loin

1. **Recoder une variable**
   Le menu `Données` → `Gérer les variables dans le jeu de données actif` → `Recoder des variables...` permet de recoder une variable de type "facteur". Le recodage doit être rentré dans la boîte de dialogue prévu à cet effet sous la forme :
   ```
   1 = "sain"
   2 = "incident"
   ```
   (qui effectue le même recodage que celui effectué précédemment pour la variable "DIFF"). La commande R générée est alors `donnees$DIFF <- recode(donnees$DIFF,'1="sain";2="incident";',as.factor.result=T)` où la fonction `recode` effectue le recodage selon les modalités précisées au-dessus et l'option `as.factor.result=T` assure que la variable résultant de ce recodage sera de type "facteur".

2 Représentations graphiques

Avant propos : Copiez à partir de `K:` les fichiers nommés `TP1.Rdata` qui correspond aux données issues du TP1. Sur `P:Travail/TP-R`, créez un dossier nommé "TP2" et collez-y le fichier `TP1.Rdata`.
Lancez R et tapez `library(Rcmdr)`. Changer le répertoire de travail pour `P:Travail/TP-R/TP2` et, à l'aide de la commande "Données → Charger un jeu de données ...", ouvrir le fichier `TP1.Rdata`. Vérifiez que le fichier de données "donnees" est bien sélectionné dans le bouton "Données" (en haut à gauche).
Enfin, sauvez l'environnement de travail sous le nom `TP2.Rdata` et le script sous le nom `TP2.R`.

2.1 Résumés numériques

1. **Obtenir un résumé numérique du jeu de données**
   Le menu "Statistiques → Résumés → Jeu de données actif" donne un résumé numérique simple de toutes les variables du jeu de données. Quelle forme prend ce résumé selon le type de variable ?
   La commande R correspondante est `summary(donnees)`.
2. **Obtenir un tableau d'effectifs ou de fréquences pour une variable qualitative**
   Quelle est la principale variable d'intérêt dans ce jeu de données ? Pour cette variable, utilisez le menu "Statistiques → Résumés → Distributions de fréquences" pour effectuer le tableau d'effectifs. Que peut-on dire de cette variable ? En particulier, que peut-on en déduire sur la probabilité d'un agriculteur de rembourser son prêt sans incident ?
   Les commandes R générées sont :
   - `.Table <- table(donnees$DIFF)` : la fonction `table` permet d'obtenir un tableau d'effectifs (ici, de la variable "DIFF" du jeu de données "donnees". Le résultat de cette opération est stocké dans une variable ".Table" ;
   - `.Table` permet d'afficher la variable ".Table" qui contient le tableau d'effectifs ;
   - `100*.Table/sum(.Table)` : la fonction `sum` fait la somme des valeurs contenues dans le vecteur ".Table" ; elle calcule donc l'effectif total et la commande permet donc de calculer le tableau de fréquences (en pourcentages) de la variable ;
   - `remove(.Table)` : la fonction `remove` supprime une variable (ici ".Table") de l'environnement de travail.
3. **Obtenir un tableau d'effectifs ou de fréquences cumulés pour une variable qualitative ordinale**

La variable "ToF" est une variable qualitative ordinale [1]. Stockez dans une variable nommée "EffectifsToF" le tableau d'effectifs de cette variable. On obtient :
- Le tableau d'effectifs cumulés de "ToF" en exécutant la commande R `cumsum(EffectifsToF)` ;
- Le tableau de fréquences cumulées de "ToF" en éxécutant la commande R `cumsum(EffectifsToF)/sum(EffectifsToF)*100`.

Les fonctions R à retenir

Fonction	Usage	Options
`cumsum`	Calcule la somme cumulée d'un vecteur	
`remove`	Supprime une variable de l'environnement de travail	
`sum`	Calcule la somme d'un vecteur	
`table`	Calcule un tableau d'effectifs pour un vecteur	

Pour aller plus loin

1. **Dénombrer et repérer les valeurs manquantes**
   Le menu "Statistiques → Résumés → Dénombrer les observations manquantes" permet de dénombrer les observations manquantes pour chaque variable.
   Il génère la commande R `sapply(donnees,function(x)(sum(is.na(x))))` où la fonction `sapply` applique une même fonction (ici `function(x)(sum(is.na(x)))`) à toutes les variables d'un jeu de données (ici "donnees"). La commande `function` permet de définir une fonction et la fonction `is.na` retourne pour un vecteur donné, un vecteur de même taille valant T (ou 1) chaque fois que la valeur est manquante. Ainsi, `function(x)(sum(is.na(x)))` définit une fonction qui calcule le nombre total de valeurs manquantes dans un vecteur "x" donné.
   La fonction `is.na` peut être utilisée pour sélectionner les observations non manquantes d'une variable : par exemple, `DDD$VVV[!is.na(DDD$VVV)]` sélectionne les valeurs non manquantes de la variable "VVV" du jeu de données "DDD" (`[.]` est utilisé pour sélectionner des observations d'un vecteur selon la condition exprimée entre les crochets et `!` désigne la négation d'une condition).
2. **Définir un ordre pour une variable qualitative ordinale**
   Lorsqu'une variable qualitative ordinale est codée par des noms, l'ordre par défaut sous R est l'ordre alphabétique. Pour définir un ordre différent, on peut utiliser le menu "Données → Gérer les variables dans le jeu de données actif → Réordonner une variable facteur...".

2.2 Diagrammes

1. **Effectuer un diagramme en tuyaux d'orgue**
   Le menu "Graphes → Graphe en barres" permet d'effectuer un

1. C'est en fait une supposition : rien ne permet de le dire réellement.

diagramme en tuyaux d'orgue. Effectuez le diagramme en tuyaux d'orgue de la variable "DIFF". La commande R générée est `barplot(table(donnees$DIFF),xlab="DIFF",ylab="Frequency")` où la fonction `barplot` effectue un diagramme en tuyaux d'orgue (ou en barres parfois) à partir d'un tableau d'effectifs avec les options `xlab` (nom de l'axe des abscisses) et `ylab` (nom de l'axe des ordonnées).
À l'aide du menu "Graphes → Sauver le graphe ... → comme bitmap ...", sauvez le graphique effectué au format jpeg sous le nom `TuyauxOrgue-DIFF.jpeg`. La commande R générée est `dev.print(jpeg,filename="TuyauxOrgue-DIFF.jpg",width=500,height=500)` où `dev.print` permet d'exporter le graphique courant avec les options `device` qui précise le type d'export (ici "jpeg"), `filename` qui précise le nom du fichier exporté (ici "TuyauxOrgue-DIFF.jpg") et `width` et `height` qui précisent la largeur et la hauteur du fichier exporté.
Critiquez le graphique obtenu : que pourrait-on modifier ? Copiez et collez la commande ayant généré le diagramme en tuyaux d'orgue et modifiez-la selon : `barplot(table(donnees$DIFF),xlab="Incident de paiement",ylab="Effectifs",main="Répartition de l'incident de paiement\n dans l'échantillon observé",col="darkblue")`. À quoi servent les options `main` et `col` ? À quoi sert le symbole `\n` ? Enregistrez ce nouveau graphique sous `TuyauxOrgue-DIFF-2.jpeg`. Les deux graphiques obtenus sont reproduits sur la Figure 2.1.

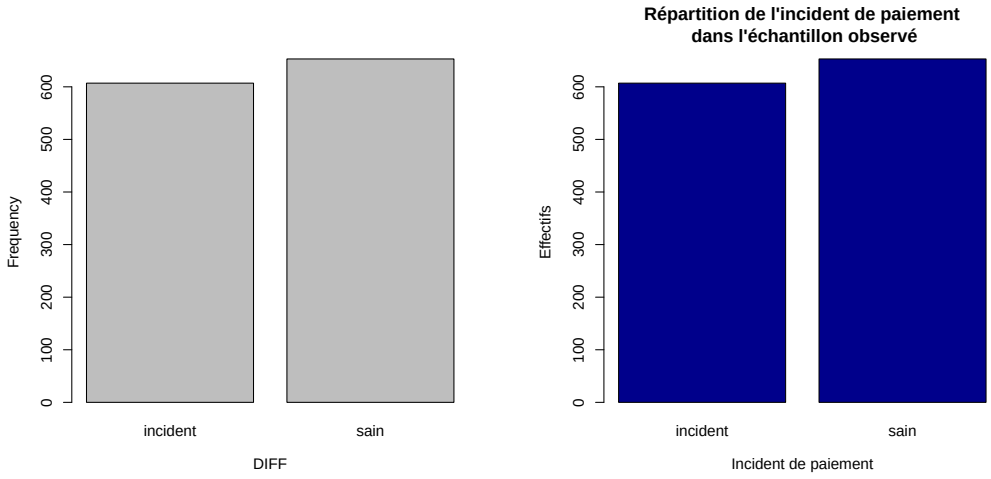

FIGURE 2.1: Diagrammes en tuyaux d'orgue obtenus pour la variable "DIFF" : commande par défaut (gauche) et commande personnalisée (droite)

2. **Effectuer un diagramme circulaire**
Le menu "Graphes → Graphe en camembert" permet d'effectuer un diagramme circulaire. Effectuez le diagramme circulaire de la variable "DPT" puis sauver ce

graphique sous `Camembert-DPT.jpeg`.
Le premier menu génère la commande R

```
pie(table(donnees$DPT),labels=levels(donnees$DPT),main="DPT",
col=rainbow(length(levels(donnees$DPT))))
```

où la fonction `pie` est utilisée pour effectuer un diagramme circulaire à partir d'un tableau d'effectifs ou de fréquences. Les options de cette fonction sont :

- `labels` qui donne les noms des modalités dans l'ordre du tableau d'effectifs donnés en premier argument. Ici, les noms des modalités sont les niveaux (fonction `levels`) de la variable "DPT" du fichier de données "donnees" ;
- `main` est le titre principal du graphique ;
- `col` sont les couleurs utilisés pour les diverses modalités ; elles sont ici générées automatiquement par la fonction `rainbow` qui génère un nombre donné de couleurs régulièrement réparties sur la palette de l'arc en ciel. Le nombre de couleurs générées dans l'exemple est égal au nombre de modalités de la variable "DPT" du fichier de données "donnees" (la fonction `length` donne la longueur d'un vecteur donné).

Personnaliser le graphique obtenu en exécutant la commande

```
pie(table(donnees$DPT),labels=c("Eure","Nord","Orne","Seine
Maritime"), main="Département d'origine des\n exploitations
agricoles", col=c("red","orange","darkgreen","blue"))
```

et enregistrez-le sous `Camembert-DPT-2.jpg`. Les deux graphiques obtenus sont reproduits sur la Figure 2.2.

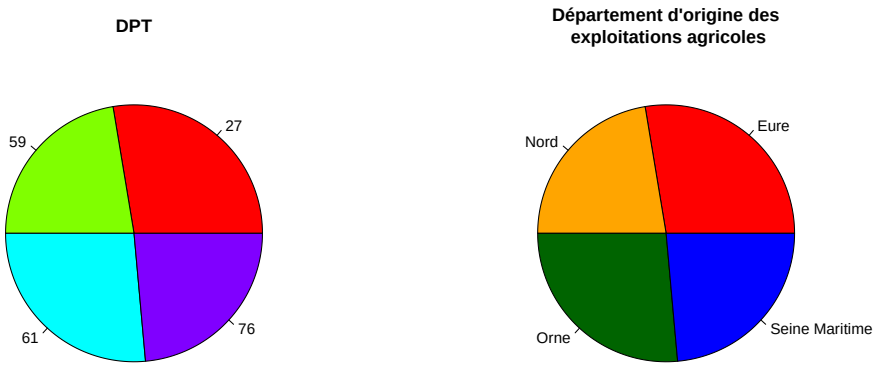

FIGURE 2.2: Diagrammes circulaires obtenus pour la variable "DPT" : commande par défaut (gauche) et commande personnalisée (droite)

Pour une variable qualitative ordinale, il faut privilégier l'utilisation de niveaux d'une même couleur plutôt que de couleurs différentes. La fonction `heat.colors`, comme la fonction `rainbow`, permet de générer un nombre donné de couleurs mais

à la différente de la précédente, les couleurs sont des niveaux de jaune/rouge. À l'aide de cette fonction, écrivez la commande R permettant d'obtenir le diagramme circulaire reproduit dans la Figure 2.3 de la variable "ToF".

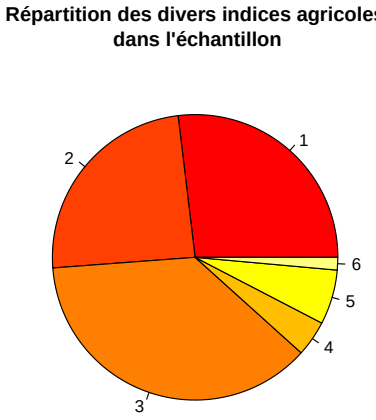

FIGURE 2.3: Diagramme circulaire personnalisé avec des niveaux de couleurs de la variable "ToF"

Les fonctions R à retenir

Fonction	Usage	Options
`barplot`	Diagramme en tuyaux d'orgue d'un tableau d'effectifs	`xlab` nom des abscisses `ylab` nom des ordonnées `main` titre principal `col` couleur des tuyaux
`dev.print`	Enregistrement d'une figure dans un fichier	`device` type de fichier (ex : jpeg) `filename` nom du fichier `width` largeur du fichier `height` hauteur du fichier
`heat.colors`	Génère un nombre donné de couleurs sur des niveaux de jaune et rouge	
`length`	Longueur d'un vecteur	
`levels`	Niveau d'un vecteur de type "facteur"	
`pie`	Diagramme circulaire d'un tableau d'effectifs ou de fréquences	`labels` noms des secteurs `main` titre principal `col` couleurs des tuyaux
`rainbow`	Génère un nombre donné de couleurs sur l'arc en ciel	

Pour aller plus loin

1. **Sauvegarder une figure au format vectoriel**
   R gère aussi l'exportation des figures au format vectoriel (postscript ou PDF). Pour cela, il faut utiliser le menu "Graphes → Sauver le graphe ... → comme PDF/Postscript/EPS". Cela se traduit dans la fonction `dev.print` par l'utilisation de l'option `device=pdf` ou bien `device=postscript` ou bien encore, pour générer un fichier EPS, par l'utilisation de la fonction `dev.copy2eps`.

2. **Effectuer un diagramme en barre**
   Il est possible, avec la fonction `barplot`, d'effectuer aussi un diagramme en barre à partir d'un tableau d'effectifs ou de fréquences. Le tableau d'effectifs doit alors être fourni à la fonction sous la forme d'une matrice à une seule colonne (et non d'un vecteur comme c'est le cas en utilisant la fonction `table`). La fonction `matrix` permet de transformer un vecteur en matrice : l'option `ncol` précise le nombre de colonnes (ici 1) et l'option `data` les données à utiliser pour remplir la matrice (ici le tableau d'effectifs). Ainsi, la commande R
   ```
   barplot(matrix(ncol=1,data=table(donnees$DIFF)/sum(table(donnees$DIFF))),
   xlab="", ylab="Fréquences",main="Répartition des incidents de
   paiement\n dans l'échantillon",col=c("red","green"),xlim=c(0,1),width=0.5)
   ```
   permettra de générer le diagramme en barre la variable "DIFF" du jeu de données "donnees". Les options supplémentaires suivantes sont utilisées :
   – `col` les couleurs des diverses parties de la barre ; contrairement au diagramme

en tuyaux d'orgue, il y en a autant que de modalités ;
- `xlim` précise les bornes de l'axe des abscisses ; combinée avec `width` qui précise la largeur de la barre, elle permet d'obtenir une barre dont la largeur n'est pas la totalité du graphique.

Cette dernière astuce permet d'insérer une légende par l'utilisation a postériori de la fonction `legend` :

```
legend("topright",legend=c("Incident","Sain"),col=c("red","green"),pch=15)
```

avec les options :
- `"topright"` qui insère la légende en haut à droite ;
- `legend` qui précise les textes de la légende ;
- `col` qui précise les couleurs de la légende ;
- `pch` qui précise la forme des symboles de la légende (ici, la valeur 15 désigne un carré plein).

3 Caractéristiques numériques

Avant propos : À faire en démarrant :
- Récupérer sur mon site web le fichier de données nommé `TP1.RData` (sur http://www.nathalievilla.org rubrique : Enseignements → IUT STID Carcassonne → Statistique descriptive ; cliquez bouton droit sur `TP1.RData` et faire Enregistrer la cible du lien sous. L'enregistrer dans `/home/stid/Documents`, par exemple)
- Créer un répertoire nommé `TP3` dans `/home/stid/Documents` ; y placer `TP1.RData`.
- Lancer R et sa librarie Commander (dans Application → Utilitaires → Terminal, écrire `R` puis, une fois le logiciel démarré, `library(Rcmdr)`)
- Définir le répertoire `TP3` comme répertoire par défaut (dans Fichier → Changer le répertoire de travail, parcourir jusqu'à sélectionner TP3)
- Ouvrir le fichier `TP1.RData` (dans Donnees → Charger un jeu de données, sélectionner TP3.RData en sélectionnant le type de fichier à "tout type de fichier")
- Enregistrer l'environnement R sous `TP3.RData` et le script R sous `TP3.R` à l'intérieur du répertoire `TP3` (dans Fichier → Enregistrer l'environnement R et dans Fichier → Enregistrer le script)

3.1 Résumés numériques

1. **Moyenne, écart type, quartiles**
   Le menu "Statistiques → Résumés → Statistiques descriptives" permet d'obtenir les caractéristiques numériques principales d'une variable quantitative. On étudie ici la variable "HECTARE" afin de voir, dans un second temps, si il existe des différences dans la capacité d'honorer un crédit selon les valeurs de cette variable. Quelles sont, pour cette variable, les valeurs de la moyenne, l'écart type, le minimum, le maximum, les trois quartiles ?
   La commande R générée est : `numSummary(donnees[,"HECTARE"],statistics=c("mean","sd","quantiles"),quantiles=c(0,.25,.5,.75,1))` où la fonction `numSummary` permet d'obtenir des statistiques sur des variables quantitatives ; les statistiques demandées sont celles listées dans `statistics=` (ici la moyenne `"mean"`, l'écart type `"sd"` et des quantiles `"quantiles"` ; les quantiles souhaités sont les quantiles dont les ordres sont listés dans `quantile=`).
   De même, déterminer les déciles de la variable "HECTARE".
2. **Moyenne, écart type, quartile en lignes de commande**
   R possède des fonctions permettant d'obtenir directement chacune des caractéristiques numériques trouvées précédemment. Tapez dans la fenêtre de script :
   ```
   mean(donnees$HECTARE)
   sd(donnees$HECTARE)
   var(donnees$HECTARE)
   ```

`quantile(donnees$HECTARE,probs=c(0,1/3,2/3,1))`
puis "Soumettre". Que donnent ces commandes ?
Quelles lignes de commande permettent de trouver :
– le coefficient de variation ?
– l'écart inter-quartile ?

Les fonctions R à retenir

Fonction	Usage	Options
`mean`	Moyenne d'une variable quantitative	
`var`	Variance d'une variable quantitative	
`sd`	Écart type d'une variable quantitative	
`quantile`	Quantile(s) d'une variable quantitative	`probs` liste des ordres des quantiles désirés

Pour aller plus loin

1. **Statistiques sur une sélection d'observations** En utilisant le menu "Données → Jeu de données actif → Sous Ensemble...", on peut sélectionner une partie des données selon une condition (par exemple, les données du département "27" : `donnees$DPT==27`) et les stocker dans un nouveau jeu de données (nommé "donnees27", par exemple). Quelles sont les statistiques de la variable "HECTARE" pour les observations du département "27" ?
Comment aurait-on pu les obtenir directement (sans utiliser le menu "Données → Jeu de données actif → Sous Ensemble...") en une ligne de commande ?

3.2 Découpage en classes

1. **Classes de même amplitude**
Le menu "Données → Gérer les variables dans le jeu de données actif → Découper une variable numérique en classes..." permet d'obtenir un regroupement automatique d'une variable quantitative en classes. Utiliser ce menu pour découper la variable "HECTARE" en 5 classes de même amplitude. Choisissez "Etendues" comme noms des niveaux et nommez la nouvelle variable ainsi créée "C1.HECTARE". Pour obtenir ceci, le menu interactif se remplit comme dans la Figure 3.1. La commande R générée est `donnees$C1.HECTARE<-bin.var(donnees$HECTARE,bins=5,method='intervals', labels=NULL)` qui se lit comme suit : la variable "C1.HECTARE" du jeu de données "donnees" reçoit le découpage en classes (fonction `bin.var`) de la variable "HECTARE" du jeu de données "donnees". Le nombre de classes, `bins`, est 5, la méthode utilisée, `method='intervals'`, est un découpage en classes de même amplitude et le nom des classes, `labels` n'est pas précisé (valeur `NULL` qui retourne la valeur par défaut décrivant chaque classe).

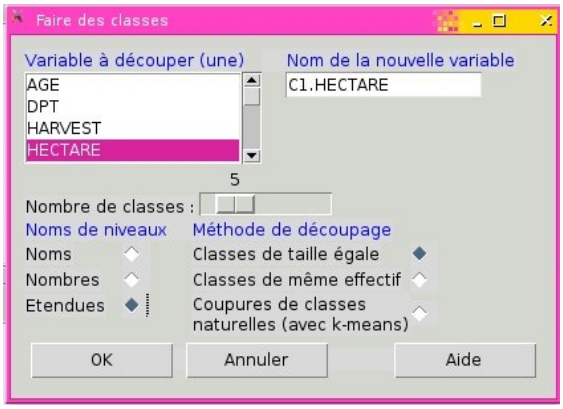

FIGURE 3.1: Découpage en 5 classes de même amplitude de la variable "HECTARE"

Visualiser la nouvelle variable en appuyant sur le bouton "Visualiser". Comment sont construits les noms des classes ? Qu'observe-t-on d'étrange dans ces noms de classes ?
Déterminer le tableau d'effectifs des classes ainsi construites. Que peut-on en dire (les classes semblent-elles pertinentes) ?

2. **Classes de même effectif**
Utiliser le même menu pour découper la variable "HECTARE" en 5 classes de même effectif et stocker le résultat dans une nouvelle variable nommée "C2.HECTARE". Faire le tableau d'effectifs de ce découpage en classes et commenter sa pertinence.

3. **Classes personnalisées**
À partir des résultats de la commande précédente, on décide de découper la variable "HECTARE" en 5 classes plus "naturelles" : $[0; 50[$, $[50; 65[$, $[65; 80[$, $[80; 100[$, $[100; 250[$. Ces classes sont construites en lignes de commande et sont stockées dans une variable appelées "C3.HECTARE". La première ligne de commande à utiliser est : `donnees$C3.HECTARE <- cut(0,50,65,80,100,250)`
Il sera, ensuite, peut-être nécessaire de remettre à jour la description des données grâce au menu "Données → Jeu de données actif → Rafraîchir le jeu de données actif".
Effectuer le tableau d'effectifs de ces classes et commenter la pertinence de ce découpage en classes.

Pour aller plus loin

1. **Donner un nom personnalisé aux classes**
Dans le menu "Données → Gérer les variables dans le jeu de données actif → Découper une variable numérique en classes...", l'option "Noms de niveaux" permet de donner un nom personnalisé aux classes. Refaites un découpage en classes de même effectif en mettant cette option à "Nombres" (dans la nouvelle variable

"C4.HECTARE") puis en mettant cette option à "Noms" et en stipulant les noms : "Très petite", "Petite", "Moyenne", "Grande", "Très grande" (dans la nouvelle variable "C5.HECTARE"). Comment ces deux options affectent-elles la ligne de commande générée ?

Les fonctions R à retenir

Fonction	Usage	Options
`cut`	Découpage en classes d'une variable quantitative	`breaks` nombre de classes ou liste des bornes des classes `labels` : noms des classes (par défaut les valeurs des bornes ; pour `labels=FALSE` des nombres)

3.3 Diagrammes

1. **Effectuer un histogramme (classes de mêmes amplitudes)**
   À l'aide du menu "Graphes → Histogramme...", effectuer l'histogramme de la variable "HECTARE" découpée en 10 classes. Attention au choix de la graduation de l'axe vertical : quelle option (entre "Fréquences", "Pourcentages" ou "Densité") faut-il choisir et pourquoi ? Le menu est à remplir comme dans la Figure 3.2 La commande générée est

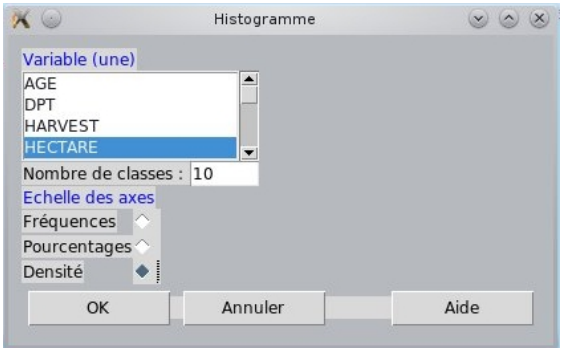

FIGURE 3.2: Histogramme de la variable "HECTARE" selon un découpage en 10 classes

```
Hist(donnees$HECTARE,scale="density",breaks=10,col="darkgray").
```

   Copier cette commande et la personnaliser en ajoutant un titre, des noms corrects aux axes et en changeant la couleur jusqu'à obtenir un graphique similaire à celle de la Figure 3.3 (à gauche). Sauvegarder ce graphique sous le nom "Histogramme-HECTARE.jpg".

2. **Effectuer un histogramme (classes personnalisées)**
   En utilisant la fonction `hist`, on peut obtenir des classes personnalisées par l'utilisation d'une liste de bornes pour l'option `breaks` à la place d'un nombre

unique indiquant le nombre de classes et avec l'option `freq=FALSE` pour que l'histogramme soit construit à partir des densités et non des fréquences. Utiliser la commande `hist` et modifier l'option `breaks` en la remplaçant par `breaks=c(0,30,50,60,70,80,90,100,150,250)`. Modifier la commande de manière à obtenir l'histogramme de la Figure 3.3 (partie droite). Sauvegarder ce graphique sous le nom "Histogramme-HECTARE-2.jpg".

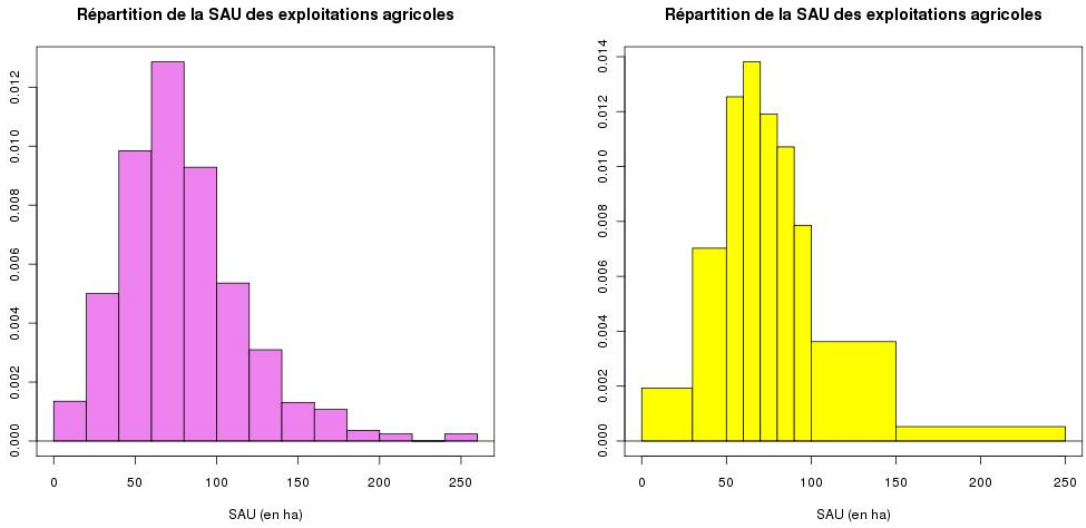

FIGURE 3.3: Répartition de la SAU dans les exploitations agricoles (histogramme personnalisé). À gauche : classes de même amplitude ; à droite : classes personnalisées

3. **Effectuer une boîte à moustaches**
   Le menu "Graphes → Boite de dispersion" permet d'effectuer une boîte à moustaches : faire celle de la variable "HECTARE" en cochant l'option "Identifier les extrêmes à la souris". La boîte à moustaches présentent 4 valeurs atypiques dans les fortes valeurs de la SAU. Cliquer sur ces points (bouton gauche de la souris) : les numéros des observations apparaissent sur la figure ; cliquer bouton droit de la souris pour terminer.
   Les commandes générées sont :

```
boxplot(incidents$HECTARE,ylab="HECTARE")
identify(rep(1,length(incidents$HECTARE)),incidents$HECTARE,rownames(incidents))
```

   la première étant utilisée pour créer la boîte à moustaches et la seconde pour identifier des points à la souris : on obtient le graphique de la Figure 3.4 (gauche).

   Recopier ces deux lignes et modifier la première de manière à obtenir le graphique de droite de la Figure 3.4. Sauvegarder ce graphique sous le nom "Boxplot-HECTARE-2.jpg".

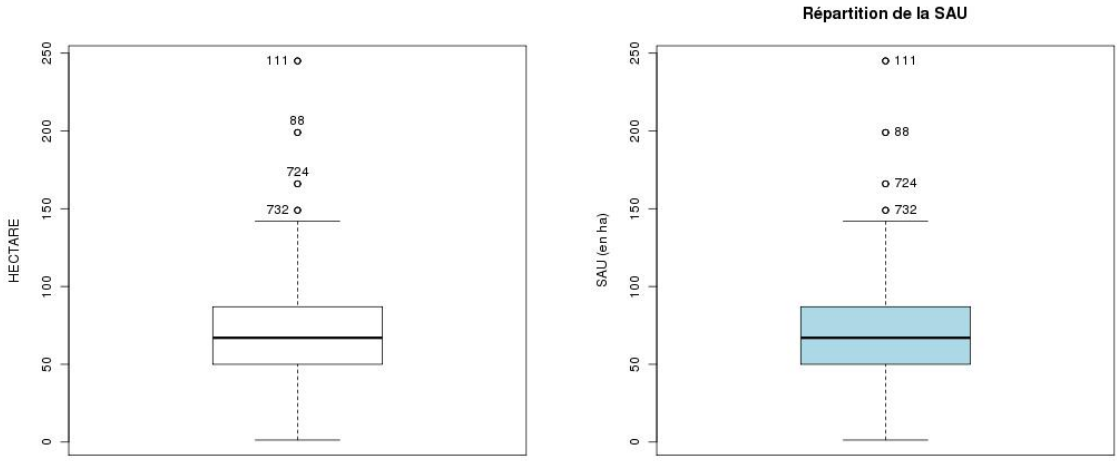

FIGURE 3.4: Répartition de la SAU dans les exploitations agricoles (boîtes à moustaches). À gauche : par défaut ; à droite : personnalisée

Les fonctions R à retenir

Fonction	Usage	Options
`hist`	Histogramme d'une variable quantitative	`freq` type de graduation sur l'axe ve (à fixer à `FALSE`) `breaks` nombre de classes ou liste de des classes
`boxplot`	Boîte à moustaches d'une variable quantitative	

Pour aller plus loin

1. **Effectuer un polygone cumulatif** Un polygone cumulatif s'effectue en ligne de commande. Pour cela, il faut procéder comme suit :
   - Stocker dans une variable les bornes des classes considérées ; par exemple, pour le découpage en classes effectué dans C3.HECTARE, cela donne :
     `BornesX<-c(0,20,65,80,100,250)`
   - Stocker dans une autre variable les effectifs cumulés des classes (le premier effectif est toujours "0") ; pour le même exemple, on a :
     `BornesY<-c(0,cumsum(table(donnees$C3.HECTARE)))`
   - Effectuer le graphique de la seconde variable en fonction de la première ; ceci est effectué par la fonction `plot` avec l'option `type="l"` pour préciser que les points doivent être reliés :
     `plot(BornesX,BornesY,type="l")`
     On peut personnaliser cette commande par :

```
plot(BornesX,BornesY,type="l",col="orange",lwd=2,lty=2,main="Polygone
cumulatif de la SAU",xlab="SAU (en ha)",ylab="Effectifs cumulés")
```
qui permet d'obtenir le graphique de la Figure 3.5 (l'option `lwd` permet d'obtenir un trait plus épais et l'option `lty` un trait en pointillé).

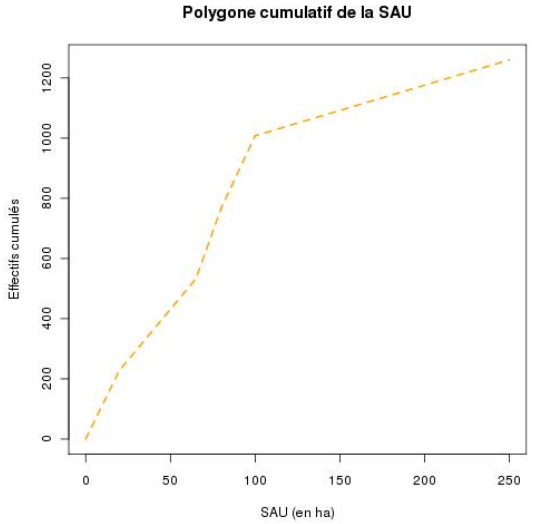

FIGURE 3.5: Polygone cumulatif de la SAU dans les exploitations agricoles

3.4 Variables centrées réduites

1. **Centrer/Réduire une variable quantitative**
   Utiliser le menu "Données $Gérer les variables dans le jeu de données actif $Standardiser les variables..." pour centrer et réduire la variable "HECTARE". Les commandes générées sont :
   ```
   .Z<-scale(donnees[,c("HECTARE")])
   donnees$Z.HECTARE<-.Z[,1]
   remove(.Z)
   ```
   La fonction `scale` retourne les valeurs centrées réduites de la variable "HECTARE" et les enregistre dans une variable "Z.HECTARE" ajoutée au jeu de données "donnees". Quel est l'utilité de chacune des lignes de commande ci-dessus ?
   Déterminer la moyenne et l'écart type de la variable "Z.HECTARE". Quel résultat est attendu ?

Les fonctions R à retenir

Fonction	Usage	Options
`scale`	Centre et réduit une variable quantitative	