

TP 0 : Rappels de R

Zaineb Smida

Table des matières

0.1	Importation et manipulation de données	1
0.2	Résumé statistique	3
0.2.1	Résumé statistique des variables quantitatives	3
0.2.2	Résumé des variables qualitatives	5
0.2.3	Résumé globale	6
0.3	Quelques graphiques de base	7
0.3.1	Une variable quantitative	7
0.3.2	Deux variables quantitatives	9
0.3.3	Une variable qualitative	11
0.4	Autres fonctions utiles	12

0.1 Importation et manipulation de données

- Pour importer un fichier texte sous **R**, on utilise la fonction `read.table()`. Si la première ligne du fichier contient le nom des variables, on utilise l'option `header=T`. Par ailleurs, le paramètre `sep` = indique le caractère qui sépare les colonnes.

```
regions <- read.table("data/regions.txt", header = TRUE)
```

L'objet créé est de type `data.frame` : les observations sont en lignes et les variables en colonnes. Un tel objet, contrairement à des matrices, peut contenir à la fois des variables quantitatives et qualitatives.

La fonction `str()` peut s'appliquer sur tous les types d'objets. Quand on l'applique sur un `data.frame`, elle permet de reprer les variables quantitatives (`numeric`, `integer`) et les variables qualitatives (`character` et ou `factor`):

```
str(regions)
```

```
'data.frame':  21 obs. of  9 variables:
 $ NOM      : chr  "A" "Q" "U" "N" ...
 $ REGION   : chr  "Alsace" "Aquitain" "Auvergne" "Bas-Norm" ...
 $ POPUL    : int  1624 2795 1320 1390 1600 2795 2370 1340 1090 1730 ...
 $ TACT     : num  39.1 36.6 37.5 38.6 38.3 ...
 $ SUPERF   : int  8280 41308 26013 17589 31582 27208 39151 25606 16202 12317 ...
 $ NBENTR   : int  35976 85531 40494 35888 40714 73763 56753 24060 27481 37461 ...
 $ NBBREV    : int   241 256 129 91 223 296 229 155 159 181 ...
 $ CHOM     : num   5.2 10.2 9.3 9 8.1 9.5 7.9 9.3 7.1 10.8 ...
 $ TELEPH   : int   700 1300 600 600 750 1300 1100 550 450 750 ...
```

- On peut visualiser le tableau de données avec la fonction `View()` :

```
View(regions)
```

- On obtient la taille du jeu de données avec la fonction `dim()` :

```
dim(regions)
```

```
[1] 21  9
```

- Affichage des premières lignes du fichier :

```
head(regions, 5)
```

	NOM	REGION	POPUL	TACT	SUPERF	NBENTR	NBBREV	CHOM	TELEPH
1	A	Alsace	1624	39.14	8280	35976	241	5.2	700
2	Q	Aquitain	2795	36.62	41308	85531	256	10.2	1300
3	U	Auvergne	1320	37.48	26013	40494	129	9.3	600
4	N	Bas-Norm	1390	38.63	17589	35888	91	9.0	600
5	O	Bourgogn	1600	38.26	31582	40714	223	8.1	750

équivalent à :

```
regions[1:5, ]
```

	NOM	REGION	POPUL	TACT	SUPERF	NBENTR	NBBREV	CHOM	TELEPH
1	A	Alsace	1624	39.14	8280	35976	241	5.2	700
2	Q	Aquitain	2795	36.62	41308	85531	256	10.2	1300
3	U	Auvergne	1320	37.48	26013	40494	129	9.3	600
4	N	Bas-Norm	1390	38.63	17589	35888	91	9.0	600
5	O	Bourgogn	1600	38.26	31582	40714	223	8.1	750

- Affichage du nom des colonnes :

```
names(regions)
```

```
[1] "NOM"      "REGION" "POPUL"   "TACT"    "SUPERF" "NBENTR" "NBBREV" "CHOM"
[9] "TELEPH"
```

- Sélection d'une colonne : on peut utiliser l'opérateur \$ suivi du nom de la variable à extraire. Le résultat est un vecteur :

```
regions$POPUL
```

```
[1] 1624 2795 1320 1390 1600 2795 2370 1340 1090 1730 10660 2110
[13] 720 2300 2430 3960 3060 1810 1590 4260 5350
```

- Sélection d'une ou plusieurs colonnes, soit par les indices de colonnes (attention, l'indice commence à 1 et pas 0 comme d'autres langages):

```
regions[, c(1, 2)]
regions[, c("NOM", "REGION")]
```

- On sélectionne toutes les colonnes sauf les colonnes 1 et 2

```
regions[, -c(1, 2)]
```

0.2 Résumé statistique

La plupart des fonctions de **R** s'appliquent sur des vecteurs. On utilisera des fonctions différentes selon qu'il s'agit d'une variable quantitative ou qualitative.

0.2.1 Résumé statistique des variables quantitatives

Par exemple, pour calculer la moyenne de la variable NBBREV, on fait :

```
mean(regions$NBBREV)
```

```
[1] 587.0952
```

Dans **R**, les valeurs manquantes sont codées des **NA**. Lorsqu'un vecteur contient des **NA**, le calcul de la moyenne ne sera pas possible sauf si on ajoute l'option `na.rm = T`. Cela est valable pour la plupart des fonctions ci-dessous :

- le minimum et le maximum :

```
max(regions$NBBREV)
```

```
[1] 6722
```

```
min(regions$NBBREV)
```

```
[1] 73
```

- la médiane et le quantile d'ordre α :

```
median(regions$NBBREV)
```

```
[1] 223
```

```
quantile(regions$NBBREV, 0.5)
```

```
50%
```

```
223
```

- l'écart-type et la variance :

```
var(regions$NBBREV)
```

```
[1] 2063457
```

```
sd(regions$NBBREV)
```

```
[1] 1436.474
```

Il est possible d'appliquer la même fonction sur plusieurs variables d'un jeu de données en utilisant la fonction `sapply()` de la façon suivante :

```
sapply(regions[, 3:ncol(regions)], max)
```

POPUL	TACT	SUPERF	NBENTR	NBBREV	CHOM	TELEPH
10660.00	46.04	48698.00	273604.00	6722.00	13.20	5800.00

0.2.2 Résumé des variables qualitatives

Une variable qualitative est codée sous la forme d'un vecteur de **character** ou un **factor**. Par exemple, on crée une nouvelle variable qui vaut "petit" si la superficie est inférieure à 15000, "moyenne" si elle est comprise entre 15000 et 30000 et "grande" sinon

```
regions$quali <- "moyenne"
regions$quali[regions$SUPERF < 15000] <- "petite"
regions$quali[regions$SUPERF >= 30000] <- "grande"
```

Sur une variable qualitative, on calcule en général la table de contingence, en utilisant la fonction `table()` :

```
tab <- table(regions$quali)
tab
```

grande	moyenne	petite
7	10	4

On calcule les proportions avec la fonction `prop.table()` :

```
round(prop.table(tab), digits = 3)
```

grande	moyenne	petite
0.333	0.476	0.190

On peut convertir la variable en **factor** si on est sûr du nombre de modalités possibles de la variable.

```
regions$quali <- factor(regions$quali, levels = c("petite", "moyenne", "grande"))
```

0.2.3 Résumé globale

- La fonction `summary()` permet de calculer quelques résumés statistiques sur toutes les variables (quantitatives ou qualitatives) d'un jeu de données :

```
summary(regions)
```

NOM	REGION	POPUL	TACT
Length:21	Length:21	Min. : 720	Min. :32.05
Class :character	Class :character	1st Qu.: 1590	1st Qu.:36.62
Mode :character	Mode :character	Median : 2110	Median :37.48
		Mean : 2681	Mean :37.23
		3rd Qu.: 2795	3rd Qu.:38.26
		Max. :10660	Max. :46.04

SUPERF	NBENTR	NBBREV	CHOM
Min. : 8280	Min. : 21721	Min. : 73.0	Min. : 5.200
1st Qu.:16942	1st Qu.: 36285	1st Qu.: 155.0	1st Qu.: 7.900
Median :25809	Median : 48353	Median : 223.0	Median : 9.300
Mean :25728	Mean : 69827	Mean : 587.1	Mean : 9.186
3rd Qu.:31582	3rd Qu.: 78504	3rd Qu.: 278.0	3rd Qu.:10.100
Max. :48698	Max. :273604	Max. :6722.0	Max. :13.200

TELEPH	quali
Min. : 350	petite : 4
1st Qu.: 700	moyenne:10
Median : 950	grande : 7
Mean :1262	
3rd Qu.:1300	
Max. :5800	

- La fonction `cor()` permet de calculer la matrice de corrélation sur plusieurs variables quantitatives :

```
cor(regions[, 3:9])
```

	POPUL	TACT	SUPERF	NBENTR	NBBREV	CHOM
POPUL	1.00000000	0.51376438	0.024369703	0.98101936	0.9213741	-0.07313003
TACT	0.51376438	1.00000000	-0.059255061	0.51571338	0.7084501	-0.69854149
SUPERF	0.02436970	-0.05925506	1.000000000	0.14929185	-0.1639580	0.06205849

```

NBENTR  0.98101936  0.51571338  0.149291848  1.00000000  0.8916071 -0.07804957
NBBREV  0.92137414  0.70845007 -0.163957955  0.89160714  1.0000000 -0.25657627
CHOM    -0.07313003 -0.69854149  0.062058491 -0.07804957 -0.2565763  1.00000000
TELEPH  0.99391186  0.55526402  0.004764791  0.98290899  0.9444463 -0.09833108
      TELEPH
POPUL    0.993911864
TACT     0.555264016
SUPERF   0.004764791
NBENTR   0.982908993
NBBREV   0.944446274
CHOM     -0.098331084
TELEPH   1.000000000

```

0.3 Quelques graphiques de base

0.3.1 Une variable quantitative

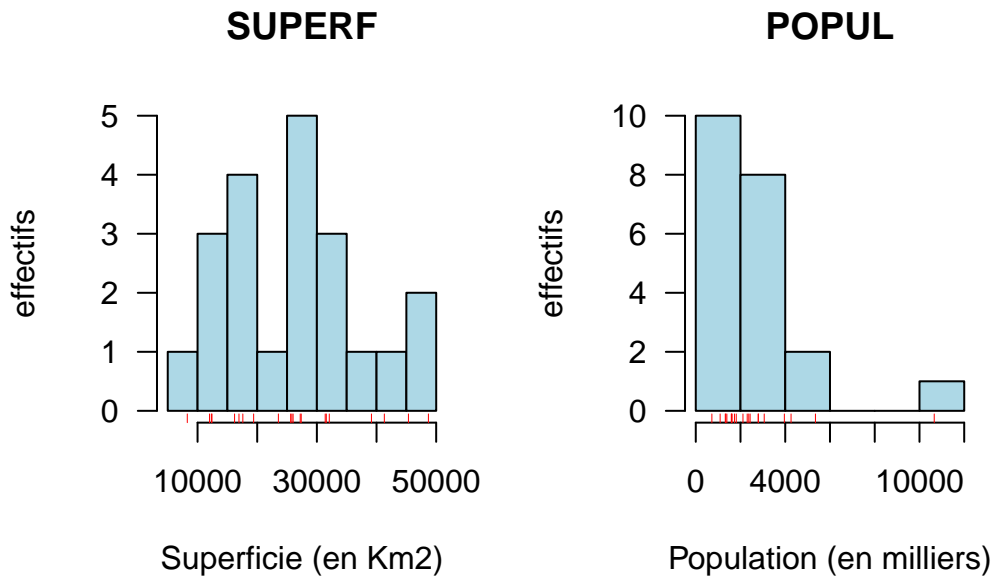
On peut utiliser l'histogramme. Il existe de nombreuses options pour customiser le graphique. Ici, on n'en présente que quelques-unes :

```

par(mfrow = c(1, 2), # permet de représenter deux histogrammes dans 1 figure
    las = 1)         # l'échelle des ordonnées est écrit horizontalement
hist(regions$SUPERF,
     col = "lightblue", # couleur de l'histogramme
     main = "SUPERF",   # titre
     xlab = "Superficie (en Km2)", # légende sur l'axe des abscisses
     ylab = "effectifs") # légende sur l'axe des ordonnées
rug(regions$SUPERF, col = "red") # représente les observations sur l'axe x

hist(regions$POPUL,
     col = "lightblue",
     main = "POPUL",
     xlab = "Population (en milliers)", ylab = "effectifs")
rug(regions$POPUL, col = "red")

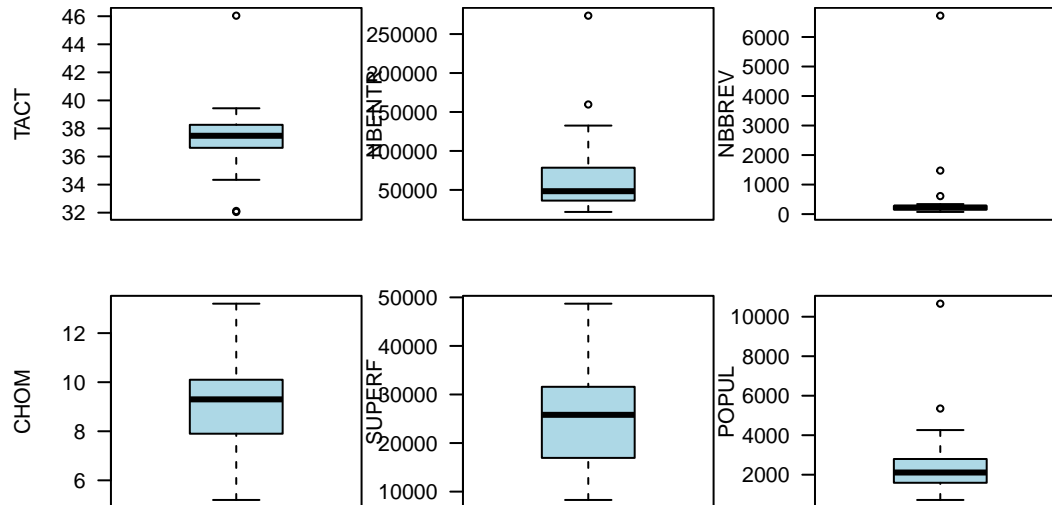
```



Les boîtes à moustaches sont des alternatives et permettent plus facilement d'identifier les points atypiques :

```
par(mfrow = c(2, 3), # Permet de représenter 6 boxplots dans 1 figure
    mar = c(3, 4, 0, 0), las = 1) # Gère les marges et l'orientation des labels

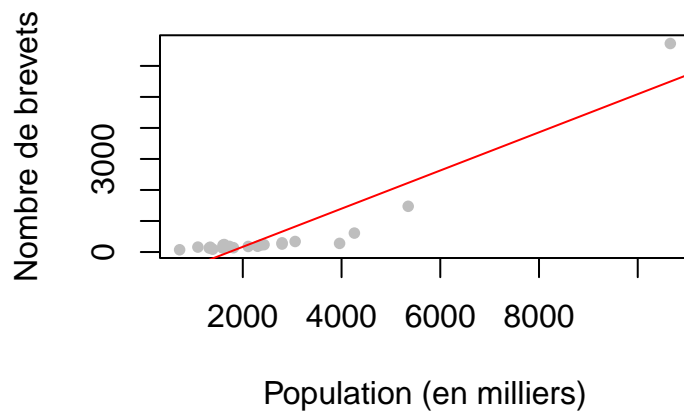
# Création des boxplots avec des couleurs différentes pour chaque variable
boxplot(regions$TACT, ylab = "TACT", col = "lightblue")
boxplot(regions$NBENTR, ylab = "NBENTR", col = "lightblue")
boxplot(regions$NBBREV, ylab = "NBBREV", col = "lightblue")
boxplot(regions$CHOM, ylab = "CHOM", col = "lightblue")
boxplot(regions$SUPERF, ylab = "SUPERF", col = "lightblue")
boxplot(regions$POPUL, ylab = "POPUL", col = "lightblue")
```

0.3.2 Deux variables quantitatives

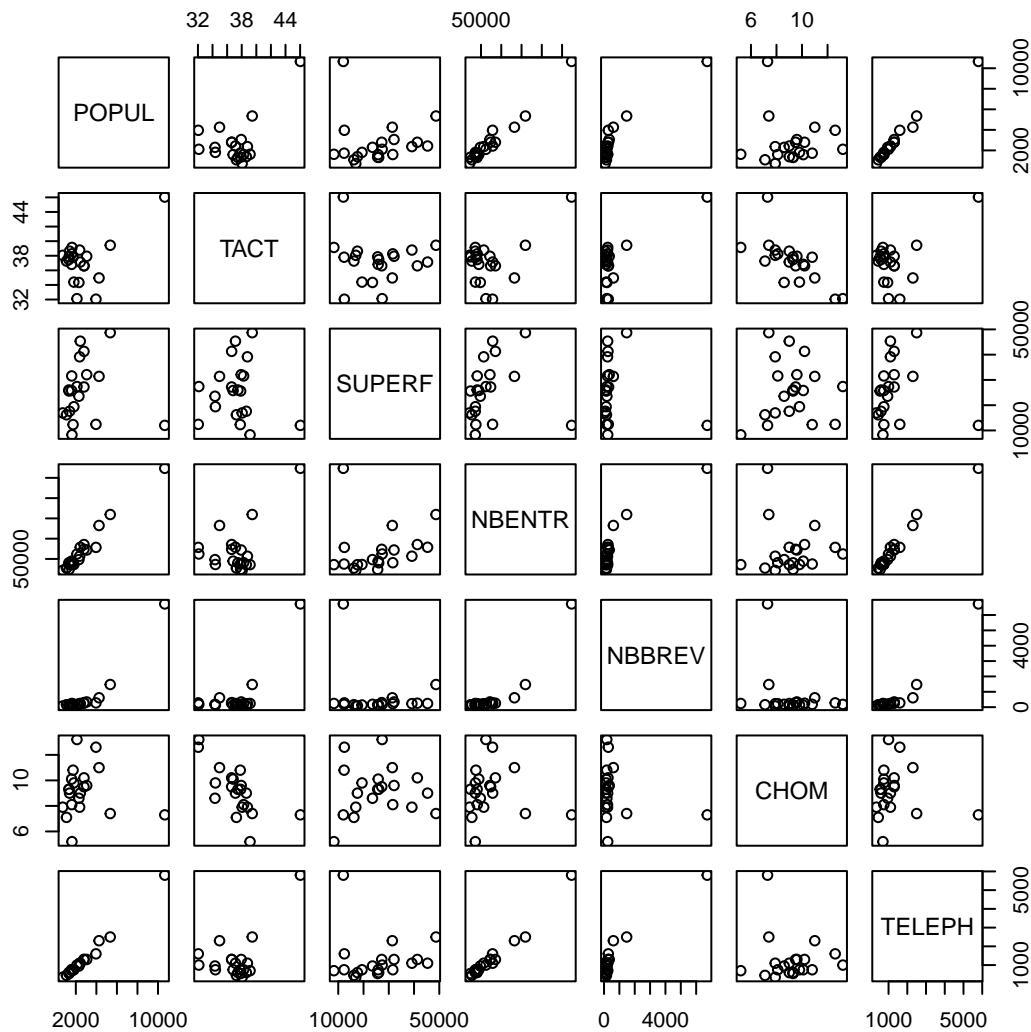
Pour étudier le lien entre deux variables quantitatives, on utilise le nuage de points :

```
plot(NBBREV ~ POPUL, # syntaxe de type formule y ~ x
     data = regions,
     pch = 16, # type de points
     cex = 0.8, # taille des points
     col = "grey", # couleur des points
     xlab = "Population (en milliers)",
     ylab = "Nombre de brevets"
)
abline(lm(NBBREV ~ POPUL, data = regions), col = "red") # droite de régression linéaire
```



La fonction `pairs()` représente toutes les paires de nuage de points possible :

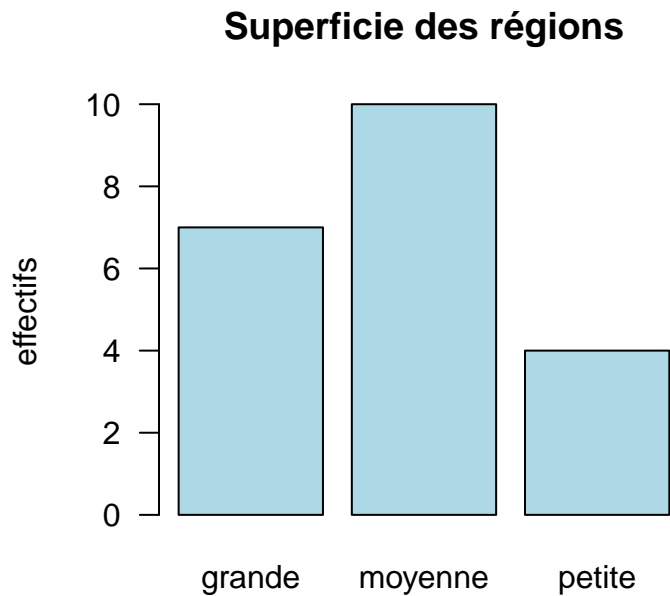
```
pairs(regions[, 3:9])
```



0.3.3 Une variable qualitative

Pour les variables qualitatives, on utilise en général un diagramme en barres, qu'on applique sur la table de contingence :

```
par(las = 1)
barplot(tab, main = "Superficie des régions",
        ylab = "effectifs", col = "lightblue")
```



Si on veut observer le lien entre deux variables qualitatives, on utilise également la fonction `barplot()`.

0.4 Autres fonctions utiles

- `setwd()` : pour changer le répertoire de travail
- `install.packages()` : installer des packages depuis le CRAN
- `library()` : charger une librairie au cours de la session

Référence utile : https://cran.r-project.org/doc/contrib/Paradis-rdebuts_fr.pdf