

TP Statistiques 1

Juhyun Park, Angie Pineda, Nicolas Brunel

4 février 2022

Préliminaires

Le TP suppose que vous avez suivi l'introduction au langage R (<https://www.r-project.org/>), dans l'environnement R Studio (<https://rstudio.com/>) effectué grâce à package swirl (<https://swirlstats.com>). Vous aurez suivi avec attention les indications de la page <https://swirlstats.com/students.html>.

Le cours interactif à suivre est “The R Programming Environment”, et de suivre les leçons de 1 à 10.

En complément, pour appréhender plus globalement la logique du langage nous vous suggérons la lecture du document:

<http://www3.jouy.inra.fr/miaj/public/formation/initiationRv10.pdf>

- Petites commandes utiles :
nettoyer son espace de travail: `rm(list=ls())`
nettoyer sa console: `Ctrl +L`
retrouver le répertoire de travail: `getwd()` changer d'emplacement de travail: `setwd()`
- La rédaction des compte-rendu de TP se fera à l'aide de Rmarkdown (et RStudio). Les compte-rendus seront remis au format RMarkdown, accompagnés du document pdf du même nom.

Génération et sauvegarde de données

Dans cette partie, on va apprendre à générer des échantillons (i.i.d) issus d'une loi de probabilité, appartenant au famille de lois usuelles.

R permet de simuler un grand nombre de lois via des fonctions de la forme `rfunc(n,p1,p2,...)` où *func* indique la loi de probabilité, *n* est le nombre d'observations (variables) à générer et *p1*, *p2*, ... sont les paramètres de la loi. Pour ce faire on aura besoin de utiliser `help()` pour les fonctions suivantes:

Lois	Nom sous R
Gaussienne	<code>rnorm(n,mean=0,std=1)</code>
Uniforme	<code>runif(n,min=0,max=1)</code>
Poisson	<code>rpois(n,lambda)</code>
Exponentielle	<code>rexp(n,rate=1)</code>
χ^2	<code>rchisq(n,df)</code>
Binomiale	<code>rbinom(n,size,prob)</code>
Cauchy	<code>rcauchy(n,location=0,scale=1)</code>

Retrouvez la définition de ces fonctions dans vos cours de probas ou stats (voire sur internet).

Pour chacune de ces fonctions, générez une échantillon de 40 observations i.i.d. (indépendantes et identiquement distribuées), insérez dans un vecteur inclus dans un `data.frame`, puis utilisez les fonctions `write.csv` et/ou `write.table` pour les enregistrer. Il serait pertinent de noter les paramètres utilisés (moyenne,std,...) dans le nom de votre variable/fichier enregistré.

```

rnorm(10)

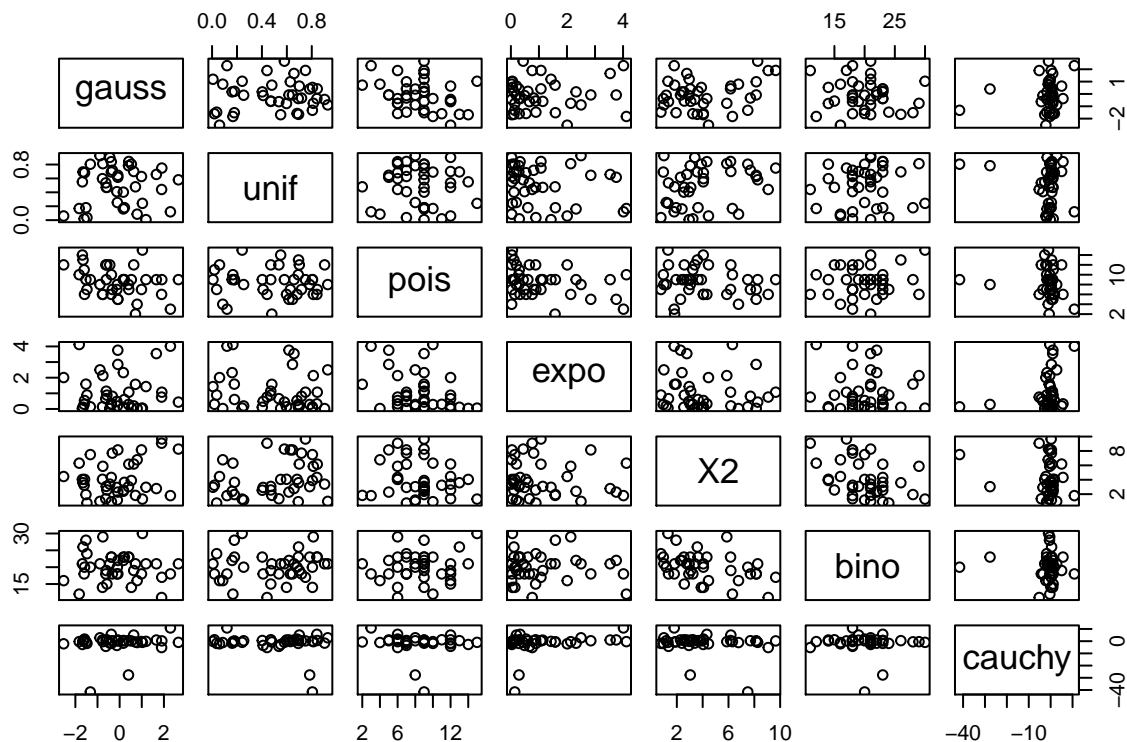
## [1]  2.3891397 -0.2707698  0.8146920 -0.5730573  1.0872111  0.3022538
## [7]  1.1165215  1.0449363 -1.0327817  0.2354385

data <- data.frame(gauss=rnorm(40,mean=0),
                   unif=runif(40,min=0,max=1),
                   pois=rpois(40,8),
                   expo=rexp(40,rate=1),
                   X2=rchisq(40,5),
                   bino=rbinom(40,100,0.2),
                   cauchy=rcauchy(40,location=0,scale=1)

)

plot(data)

```



```

write.csv(data, file = "data.csv")

```

Charger des données depuis un fichier txt (texte) et csv (comma separated variables)

Nettoyez votre espace de travail. Utilisez les fonctions `read.csv` et/ou `read.table`, pour charger la distribution Gaussienne que vous avez généré. Que remarquez-vous?

Pensez à utiliser `header=TRUE`.

```

read.csv('data.csv', sep=" ")[,1]

```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
```

Tracer les données

Générez un vecteur qui contient 10 réalisations de la loi normale $N(0,1)$. Tracez les points obtenus en utilisant 'plot', et mettant sur l'axe des x un vecteur séquentiel de la taille de votre vecteur.

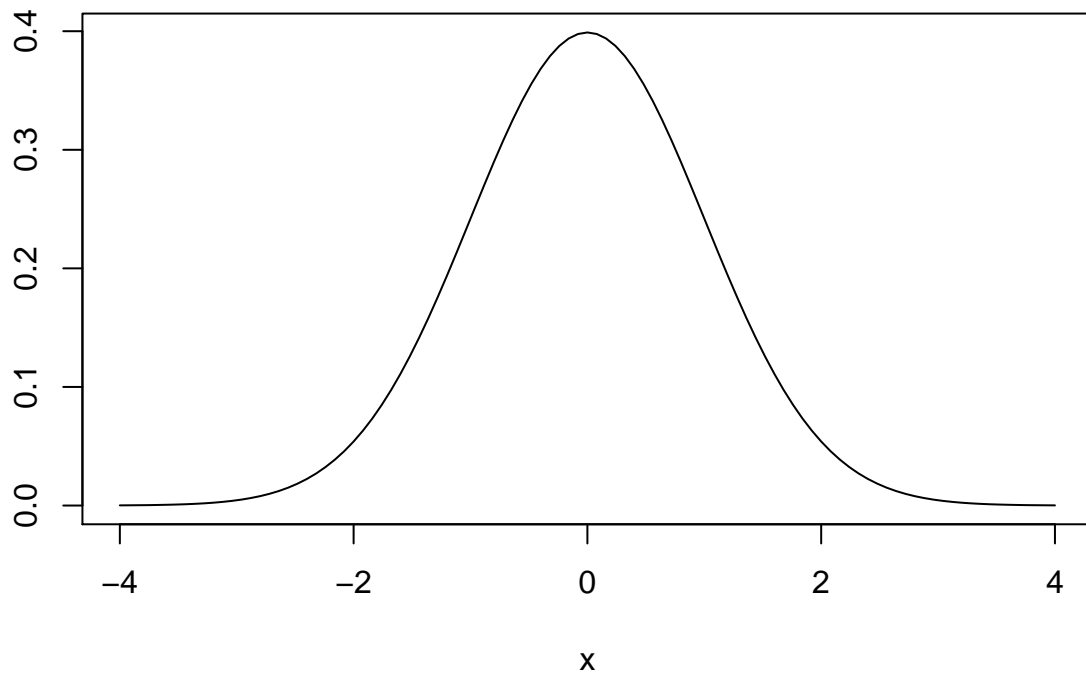
Que remarquez-vous? (Utilisez la commande 'abline(h=0)')

```
x<-seq(10)
vect_norm=rnorm(10,0,1)
moy<-0
std<-1
plot(function(x) dnorm(x,moy,std),moy-4*std,moy+4*std,xab="x",ylab="",main="")
```

```
## Warning in plot.window(...): "xab" n'est pas un paramètre graphique
## Warning in plot.xy(xy, type, ...): "xab" n'est pas un paramètre graphique
## Warning in axis(side = side, at = at, labels = labels, ...): "xab" n'est pas un
## paramètre graphique

## Warning in axis(side = side, at = at, labels = labels, ...): "xab" n'est pas un
## paramètre graphique

## Warning in box(...): "xab" n'est pas un paramètre graphique
## Warning in title(...): "xab" n'est pas un paramètre graphique
```



Tracez également les lignes horizontales 1 et -1. Que remarquez-vous? Combien de points sont en dehors de ces lignes? La même chose avec les lignes horizontales 2 et -2, 3 et -3. Que remarquez vous?

Effectuez la même chose avec des vecteurs contenant 100 et 1000 valeurs. Que remarquez vous?

Chargez le fichier 'distribution_inconnue.csv' que vous pouvez trouver sous:

Est-ce que vous pouvez conclure quelque chose sur cette distribution, à partir d'une visualisation?

Testez avec d'autres distributions. Que remarquez-vous?

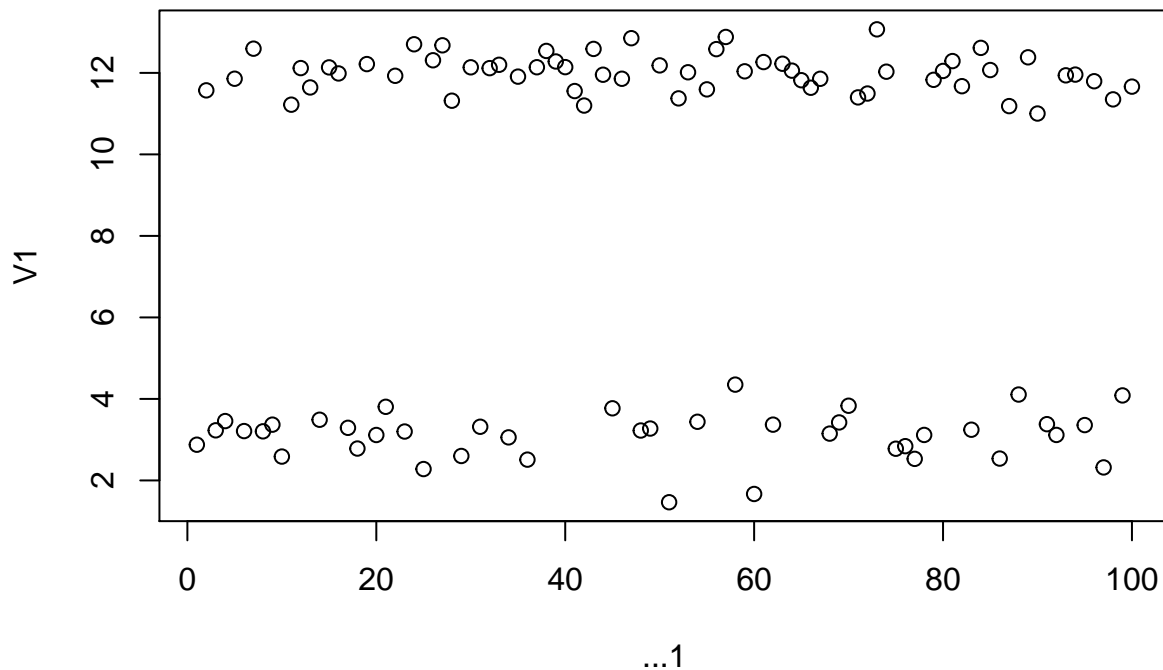
```
library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

library(readr)
distrib<-read_csv(file = "distribution_inconnue.csv")

## New names:
## * `` -> ...1
## Rows: 100 Columns: 2
## -- Column specification -----
## Delimiter: ","
## db1 (2): ...1, V1
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

plot(distrib)
```



Histogrammes

La visualisation des résultats précédents nous donnent certaines informations sur la distribution dont ils sont issus.

Les histogrammes sont une autre façon d'évaluer visuellement les données d'un échantillon. Ils représentent la densité de distribution de valeurs de réalisations de notre échantillon par segments (fréquence).

Utilisez `help()` pour la fonction `hist()`.

Appliquez la fonction pour l'échantillon de 100 réalisations que vous avez créé, et pour 'distribution_inconnue.csv'. Que remarquez vous?

Testez les différents paramétrages de la fonction: `breaks` et `freq`.

Effectuez la même chose pour des distributions de Cauchy avec des paramétrages différents.

Moments d'ordre supérieur

Les moments centrés donnent de l'information sur la forme de la distribution. Si on connaît la densité d'une loi, on peut calculer ses moments. Mais quand on a uniquement accès à un échantillon i.i.d, les moments sont alors estimés empiriquement.

- Empiriquement:
Skewness ou coefficient d'asymétrie —> Si négatif, la densité est penchée vers la gauche, si positif la densité penchée vers la droite.
Kurtosis ou coefficient d'applatissement —> Si négatif, la densité a des queues qui décroissent plus vite que celle de la loi normale; si positif, les queues sont plus épaisses que celles d'une gaussienne.

Moment	Ordre	Formule	Estimateur
Moyenne	1	$E[X] = \int_{-\infty}^{\infty} x dF(x) = m$	$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$
Variance	2	$E[(X - m)^2] = \int_{-\infty}^{\infty} (x - m)^2 dF(x) = \sigma^2$	$\sigma_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$
Skewness	3	$E\left[\left(\frac{X-m}{\sigma}\right)^3\right] = \int_{-\infty}^{\infty} \left(\frac{x-m}{\sigma}\right)^3 dF(x)$	$b_1 = \frac{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^3}{\left[\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2\right]^{3/2}}$
Kurtosis	4	$E\left[\left(\frac{X-m}{\sigma}\right)^4 - 3\right] = \int_{-\infty}^{\infty} \left(\frac{x-m}{\sigma}\right)^4 dF(x) - 3$	$g_2 = \frac{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^4}{\left[\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2\right]^2} - 3$

Sous R il existe les fonctions `skewness()` et `kurtosis()`. Calculez les moments centrés des 4 premiers ordres pour les échantillons que vous avez généré et stockez les résultats dans une matrice. Commentez les résultats obtenus et comparez les avec les valeurs théoriques de ces distributions.

Quantiles et Boxplot

Une autre façon de représenter une distribution (pour la visualiser ou l'estimer) est d'utiliser les quantiles.

Le quantile x_α d'une variable aléatoire X est défini comme: $P(X \leq x_\alpha) = \alpha$ (pour tout $\alpha \in [0, 1]$) ou de façon équivalente: $P(X > x_\alpha) = 1 - \alpha$.

Comme avant, entre connaître la distribution réelle et essayer de "faire parler les données", il y a une grande différence. On s'appuie sur notre échantillon pour essayer d'avoir plus d'informations sur la distribution sous-jacente.

*Quantiles spéciaux: Q_1 : 1er quartile. La valeur en dessous de laquelle on a le quart des valeurs de notre échantillon.

Q_2 : Médiane. La valeur en dessous de laquelle on a la moitié des valeurs.

Q_3 : 3ème quartile. La valeur en dessous de laquelle on a les trois-quarts des valeurs de notre échantillon.

Le boxplot nous permet de voir les valeurs entre Q_1 , Q_2 et Q_3 , ainsi que la moyenne, et l'intervalle interquartile $\Delta = Q_3 - Q_1$. Toute valeur en dehors des limites (moustaches) $Q_1 - 1.5\Delta$ et $Q_3 + 1.5\Delta$ est marquée avec des points individuels.

Regardez l'aide de la fonction `boxplot()` et appliquez-la sur les différents ensembles générés. Pour le tableau précédent, contenant les moments de ordre 1 à 4, ajoutez 3 colonnes qui contiennent les 3 quantiles.

Interprétation visuelle

Simulez 3 ensembles de 100 individus avec la loi de Cauchy avec des paramétrisations différentes. Effectuez toutes les démarches vues dans ce TP. Que remarquez-vous?

Analyse d'un jeu de données

Chargez des données réelles (Auto.csv) dans un data frame. Décrivez la distribution de ces données (par des indicateurs numériques et graphiques).