# TP 1 Statistiques

*Nawal Belaid, Nicolas Brunel, Salim Amoukou*

*14 fevrier 2020*

## Préliminaires

Le TP suppose que vous avez suivi l'introduction au langage R (https://www.r-project.org/), dans l'environnment 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 r*func*(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 | rnorm(n,mean=0,std=1) |
| Uniforme | runif(n,min=0,max=1) |
| Poisson | rpois(n,lambda) |
| Exponentielle | rexp(n,rate=1) |
| $\chi^2$ | rchisq(n,df) |
| Binomiale | rbinom(n,size,prob) |
| Cauchy | rcauchy(n,location=0,scale=1) |

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.

### 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.

### 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 sequentiel de la taille de votre vecteur.

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

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_inconue_1_100_realisations.csv' que vous pouvez trouver sous https://pydio.pedago.ensiie.fr

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

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

### 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 echantillion. 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_inconue_1_100_realisations.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 connait 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 | $E[(X-m)^2] = \int_{-\infty}^{\infty}(x-m)^2 dF(x) = \sigma^2$ | $\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n^2}(x_i - \bar{x})^2$ |
| Skewness | 3 | $E[\left(\frac{X-m}{\sigma}\right)^3] = \int_{-\infty}^{\infty}\left(\frac{x-m}{\sigma}\right)^3 dF(x)$ | $b_1 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{[\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2]^{3/2}}$ |

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

Sous R il existe les fonctions `skewness()` et `kurtosis()`. Calculez les moments centrés des 4 premiers ordres pours les échantillons que vous avez généré et stockez les résultats dans une matrice. Commentez les resultats 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 $x_\alpha$ d'un 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 difference. On s'appuit 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 la quelle on a le quart des valeurs de notre échantillon.
$Q_2$: Médiane. La valeur en dessous de la quelle on a la moitié des valeurs.
$Q_3$: 3ème quartile. La valeur en dessous de la quelle 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 interqartile $\Delta = Q_3 - Q_1$ . Toute valeur en dehors des limites (moustaches) $Q_1 - 1.5\Delta$ et $Q_3 + 1.5\Delta$ est marqué avec des points individuels.

Regardez l'aide de la fonction `boxplot()` et appliquez la sur les différents ensembles générés Pour le tableau precedent, 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 parametrisations differentes. Effectuez toutes les demarches vues dans ce TP. Que remarquez-vous?

**Analyse d'un jeu de données**

Le fichier *Data.txt* contient les mesures sur des pieds de maïs des variables suivantes au moment de la récolte:

Hauteur : hauteur en cm

Mass : masse totale en grammes

Masse.grains : masse de grains en grammes

Couleur : couleur du grain (jaune, jaune/rouge, rouge)

Parcelle : situation (Nord, Sud, Est, Ouest)

Hauteur.J7 : hauteur en cm 7 jours après la récolte.

- Chargez les données réelles *Data* dans un data frame.
- Choisissez une ou deux variables et décrivez la distribution de chacune par des indicateurs numériques et graphiques.