OS13 : Tests d'estimation et loi extrêmes des lois de Cauchy et géométrique

Adrien WARTELLE
TRAN Quoc Nhat Han

4 octobre 2018

Sommaire

Résumé

Ce rapport montre comment estimer des paramètres pour la loi géométrique et la loi de Cauchy. On y étudie aussi les lois extrêmes correspondantes.

1 Loi géométrique

1.1 Rappel

La loi géomètrique est une loi discrète utilisée dans le cadre d'un enchainement (non fini) d'épreuves de Bernouilli. Soit X une variable aléatoire suivant cette loi $G(p)$, on a :

$$P(X = k) = (1 - p)^{k-1} p \ k \in \mathbb{N}_+$$

La variable X correspond au numéro de la première épreuve où l'on obtient un succès, celle-ci ayant une probabilité p. Ainsi, p est le seul paramètre de la loi. La fonction de répartition $F(k) = P(X \leq k) \ k \in \mathbb{N}_+$ est :

$$F(k) = 1 - (1 - p)^k$$

Démonstration. On peut remarquer que :

$$P(X = k) = (1 - p)^{k-1}(1 - (1 - p)) = (1 - p)^{k-1} - (1 - p)^k$$

Donc :

$$F(k) = \sum_{l=1}^{k} P(X = l) = \sum_{l=1}^{k} ((1 - p)^{l-1} - (1 - p)^l)$$

$$F(k) = \sum_{l=0}^{k-1} (1 - p)^l - \sum_{l=1}^{k} (1 - p)^l$$

Soit $q = 1 - p$, on a :

$$F(k) = \sum_{l=0}^{k-1} q^l - \sum_{l=1}^{k} q^l$$

$$F(k) = \frac{1 - q^k}{1 - q} - (\frac{1 - q^{k+1}}{1 - q} - 1)$$

$$F(k) = \frac{1 - q^k - 1 + q^{k+1} + 1 - q}{1 - q}$$

$$F(k) = \frac{1 - q + q^{k+1} - q^k}{1 - q}$$

$$F(k) = \frac{1 - q - q^k(1 - q)}{1 - q}$$

$$F(k) = 1 - q^k$$

On retrouve bien :

$$F(k) = 1 - (1 - p)^k$$

□

1.2 Génération de n variables aléatoires

Afin de voir à quoi ressemble la distribution et de pouvoir tester l'estimateur que nous allons calculer par la suite, on génère des échantillons de 50, 500 et 5000 variables. On utilise ainsi le code Matlab (Octave) ci-dessous :

```
p=0.1; %probability to win
pop1 = geomGen(50,p);
pop2 = geomGen(500,p);
pop3 = geomGen(5000,p);
k=1:max(pop3)+1;
distrib = (1-p).^k.*p;
subplot(221);
hist(pop1,k,1);
xlabel('k');
ylabel('Freq(k)');
title('Histogramme de 50 variables');
subplot(222);
hist(pop2,k,1);
xlabel('k');
ylabel('Freq(k)');
title('Histogramme de 500 variables');
subplot(223);
hist(pop3,k,1);
xlabel('k');
ylabel('Freq(k)');
title('Histogramme de 5000 variables');
subplot(224);
bar(k,distrib);
xlabel('k');
ylabel('P(X=k)');
title('Distribution théorique géomètrique');
```

On obtient ainsi la figure 1.

On observe bien que les histogrammes se rapprochent de la distribution théorique quand la taille de l'échantillon est grande (plusieurs milliers). On peut aussi noter que la distribution se rapproche d'une loi exponentielle (qui a pour distribution $f(t) = \lambda e^{-\lambda t}, \forall x \in \mathbb{R}_+$). En effet la loi géométrique est aussi une loi sans mémoire, c'est-à-dire que :

$$\forall k, l \in \mathbb{N}_+, P(X = k + l | X > k) = P(X = l) = (1 - p)^{l-1} p$$

Cette propriété caractérise la loi exponentielle dans le cas continue, on voit alors que la loi exponentielle correspond à la limite continue de la loi géométrique si l'on fait correspondre les valeurs k obtenues par $X \sim G(p)$ avec des intervalles de $\mathbb{R}_+$ et si l'on fait correspondre p avec la probabilité $P(t_k < T < t_k + h | t_k) = e^{-\lambda h}$, T suivant une loi exponentielle $\epsilon(\lambda)$ et $[t_k, t_k + h]$ l'intervalle correspondant à la valeur k. Pour améliorer cette "similarité", on prendra h et (donc p) le plus petit possible.

Pour générer les populations , on utilise la fonction "geomGen" qui simule n expériences où, pour chacune d'entre elle, on effectue un enchainement d'épreuve de Bernouilli en générant une variable de loi uniforme sur $[0; 1]$.

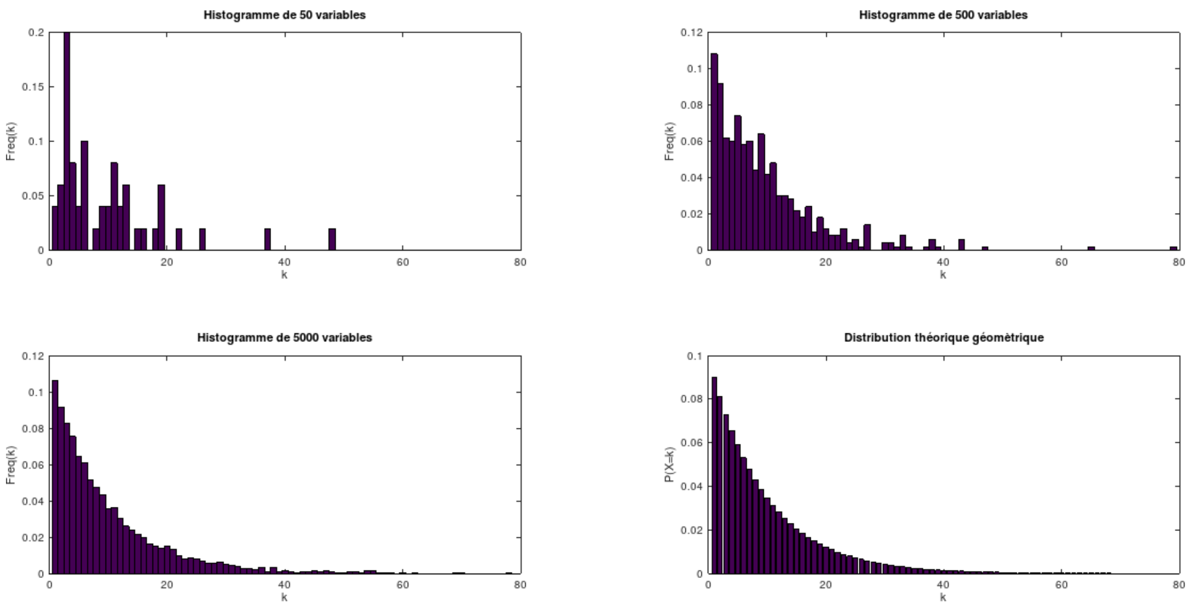

FIGURE 1 – Histogrammes et distribution de loi géométrique

Si la valeur obtenu est supérieure à p, ce qui a une probabilité p d'arriver, on arrête la boucle et la valeur générée est égale au nombre d'épreuves, sinon on continue jusqu'à obtenir cette condition. Le code Matlab (Octave) utilisé est :

```
function[X]=geomGen(n,p)
X=ones(1,n);
for (i=1:n)
  trial=rand(1,1);
  k=1;
  while (trial > p)
    trial=rand(1,1);
    k++;
  endwhile
  X(i)=k;
endfor
endfunction
```

Cette méthode revient à générer de manière répéter une variable de Bernouilli.

1.3 Estimateurs du maximum de vraisemblance

L'estimateur $\hat{p}$ du maximum de vraisemblance est la valeur qui maximise la loi de vraisemblance L(p) : $\hat{p} = arg\ \underset{p}{max}(L(p))$. On calcule L(p) dans un premier temps en assumant que les variables sont indépendantes [1] :

$$L(p) = L(X_1(p), X_2(p), ..., X_n(p)) = \prod_{i=1}^{n} (1-p)^{X_i - 1} p$$

avec $X_i,\ i \in \{1, .., n\}$ les variables d'échantillon.

$$L(p) = p^n (1-p)^{(\sum\limits_{i=1}^{n} X_i) - n}$$

On peut effectuer un passage au logarithme pour trouver un maximum car il s'agit d'une fonction strictement croissante (et défini sur $]0;1]$). L'argument du maximum du logarithme de $L(p)$ et du maximum de $L(p)$ sont les mêmes.

$$\log L(p) = n \log p + ((\sum_{i=1}^{n} X_i) - n) \log (1-p)$$

$$\frac{d \log L(p)}{dp} = \frac{n}{p} - \frac{(\sum\limits_{i=1}^{n} X_i) - n}{1-p}$$

$$\text{On a } \frac{d \log L(\hat{p})}{dp} = 0$$

$$\text{Donc } \frac{n - n\hat{p} - ((\sum\limits_{i=1}^{n} X_i) - n)\hat{p}}{\hat{p}(1-\hat{p})} = 0$$

$$\text{Ce qui implique (dans le cas différent de 0 et 1) } n - (\sum_{i=1}^{n} X_i)\hat{p} = 0$$

Finalement :

$$\hat{p} = \frac{n}{\sum\limits_{i=1}^{n} X_i}$$

De plus on peut vérifier qu'il s'agit bien d'un maximum (unique et donc global) en regardant si la fonction $L(p)$ est concave :

$$\frac{d^2 \log L(p)}{dp^2} = -\frac{n}{p^2} - \frac{\sum\limits_{i=1}^{n} X_i - n}{1-p}$$

1. on le supposera par la suite pour les échantillons générés

Sachant que $\sum\limits_{i=1}^{n} X_i > n$ et que $p \in [0;1]$, on a :

$$\frac{d^2 \log L(p)}{dp^2} < 0 \ \forall p \in [0;1]$$

La fonction $L(p)$ est donc concave et l'estimateur trouvé $\hat{p} = \frac{n}{\sum\limits_{i=1}^{n} X_i}$ maximise bien la loi de vraisemblance.

Si la résolution analytique pour trouver l'estimateur avait été impossible, l'utilisation d'un algorithme d'optimisation (comme une descente de gradient) ou encore l'utilisation d'une fonction d'intégration aurait été possible.

1.4 Test de l'estimateur

Afin de tester l'estimateur du maximum de vraisemblance pour différent taille d'échantillon, on utilise le code suivant :

```
close all;
p=1/10;
n=10.^(1:0.02:4);
delta_est=zeros(1,length(n));
for i=1:length(n)
  Y=geomGen(n(i),p);
  p_est = n(i)/sum(Y);
  delta_est(i) = (p_est-p)/p;
  i++;
endfor
plot(log10(n),delta_est);
title('Evolution écart de p_{est}');
xlabel('log10(n)');
ylabel('(p_{est}-p)/p');
```

On estime la valeur théorique $p = 0.1$ en utilisant estimateur de vraisemblance $\hat{p} = \frac{n}{\sum\limits_{i=1}^{n} X_i}$ pour des tailles d'échantillons n allant de 10 à 10000. On mesure pour chaque n l'écart relatif $\frac{\hat{p}-p}{p}$ et on obtient la figure 2.

On voit que que l'estimateur commence à être relativement précis, c'est à dire avec un biais relatif en dessous de 10%, autour de 1000 (10^3 valeurs) variables générées. On peut noter aussi que, pour des valeurs petites de n, les écarts sont beaucoup plus souvent positifs ($\hat{p} > p$) que négatifs ($\hat{p} > p$) et que ces écarts positifs vont jusqu'à plus de 75% alors que les négatifs sont en dessous de 40%. On peut alors émettre l'hypothèse que notre estimateur est biaisé.

Pour confirmer cette hypothèse, il faudrait calculer le biais de l'estimateur qui correspond à l'écart moyen (d'espérance) entre l'estimateur et le paramètre réel. En effet l'écart entre p et $\hat{p}$ est une variable aléatoire qui va

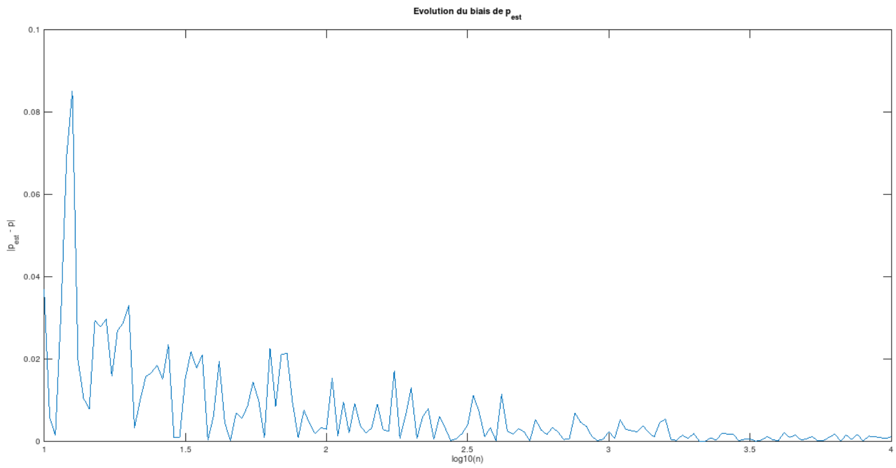

FIGURE 2 – Evolution du biais de l'estimateur

dépendre de l'échantillon généré[2]. Si on essaye d'obtenir une valeur théorique :

$$B = \mathbb{E}(\hat{p} - p)$$

$$B = \mathbb{E}(\hat{p}) - p$$

$$B = n\mathbb{E}\left(\frac{1}{\sum_{1}^{n} X_i}\right) - p$$

On voit que ce calcul n'est pas simple puisqu'il faudrait connaître la loi de l'inverse de la somme de variables géométriques[3].

Nous allons donc estimer l'espérance du biais, pour différentes valeurs de n, en utilisant la moyenne des biais obtenus sur plusieurs échantillons (pour une même taille n) :

```
close all;
p=1/10;
n=10.^(1:0.2:3);%more is too much
%n=10.^(1:0.01:2);
delta_moy_est=zeros(1,length(n));
for i=1:length(n)
  moy_p_est=0;
  for k=1:500
    Y=geomGen(n(i),p);
    moy_p_est += n(i)/sum(Y);
```

2. Il faut donc plusieurs échantillons pour le tester (pour un n donné)

3. on pourrait considérer que la somme suit une loi gaussienne en utilisant le théorème centrale-limite et d'utiliser la gaussienne inverse

```
    endfor
    moy_p_est/=k;
    delta_moy_est(i) = (moy_p_est-p)/p;
    i++;
endfor
plot(log10(n),delta_moy_est);
title('Evolution écart moyen de p_{est}');
xlabel('log10(n)');
ylabel('Moy((p_{est} - p)/p)');
```

Pour des valeurs de n allant de 10 à 1000, on calcule l'écart moyen de 500 estimations provenant de 500 échantillons différents et obtient la figure3.

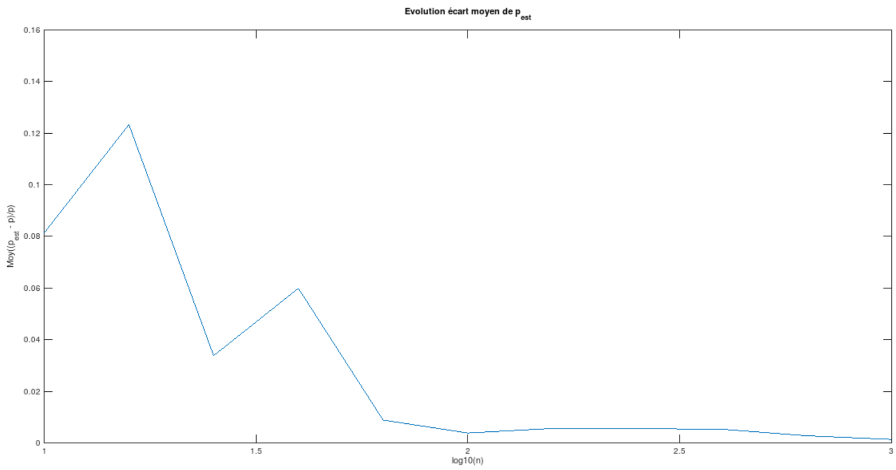

FIGURE 3 – Evolution de l'écart moyen de l'estimateur

On voit très clairement que l'estimateur a un biais positif (en considérant que 500 valeurs permettent de correctement estimer celui-ci, ce qui semble raisonnable) pour des valeurs de n inférieur à 100. On observe 2 pics, le premier à environ +12% d'erreur pour une taille de 15 et une autre de +6% pour une taille de 50.

Afin de mieux voir ce biais, on prend un pas de taille de 0.01 pour $\log_{10}(n)$, n allant de 10 à 100 et on obtient la figure 4. Le biais estimé va de +20% en n=10 et redescend proche 0 en n=100 avec des variations allant jusqu'à 5% entre chaque estimation. On peut donc raisonnablement affirmer que l'estimateur $\hat{p}$ est biaisé pour des échantillons de taille réduite (en dessous de 100) mais que le biais est suffisamment petit pour des tailles grandes (n>100).

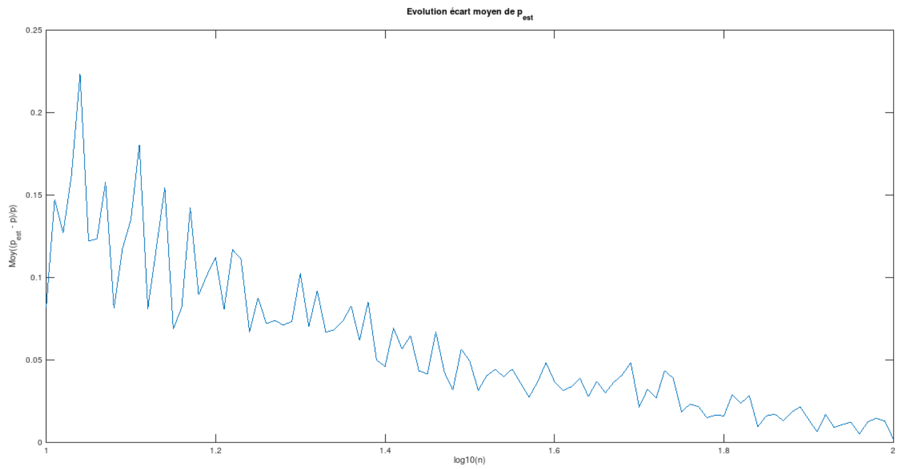

FIGURE 4 – Evolution de l'écart moyen de l'estimateur affiné

1.5 Loi extrême

On veut voir à que ressemble la loi extrême correspondant à la loi géométrique. Pour cela on génère m (10, 100, 1000 et 10000) échantillons de $n = 100$ variables et on prend la valeur maximale (1%) de l'échantillon à chaque fois pour générer un échantillon des maximums des échantillons de loi géométrique. On utilise le code suivant :

```
close all;
p=1/10;
n=100;
i=1;
for M=[10 100 1000 10000]
  Z=zeros(1,M);
  for m=1:M
    Y=geomGen(n,p);
    %Y=sort(Y);
    %Z((m-1)*(n/100)+1:m*(n/100))=Y((99*n)/100+1:n);
    %Z(m)=Y(100); %take the last value
    Z(m)=max(Y);
  endfor
  subplot(2,2,i);
  i++;
  hist(Z,100*i);
  xlabel('k');
  ylabel('Freq(k)');
  title(["Histogramme des valeurs extrêmes M=" num2str(M)]);
endfor
```

On obtient alors la figure 5.

La distribution extrême correspondante ne peut pas être une loi de gumbel puisque nous avons des valeurs uniquement positive, cela pourrait être une

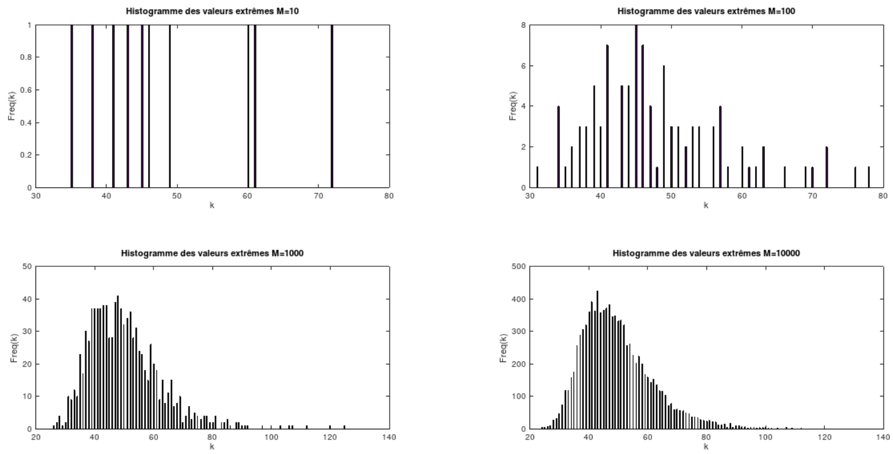

FIGURE 5 – Distributions extrêmes de la loi géométrique

loi de Weibull ....

2 Loi de Cauchy

2.1 Rappel

Soit f_X la fonction de densité de Cauchy de deux paramètres x_0 et a ($a > 0$), définie par :

$$f_X(x) = \frac{1}{\pi a \left(1 + \left(\frac{x-x_0}{a}\right)^2\right)} = \frac{1}{\pi} \frac{a}{(x-x_0)^2 + a} \tag{2.1}$$

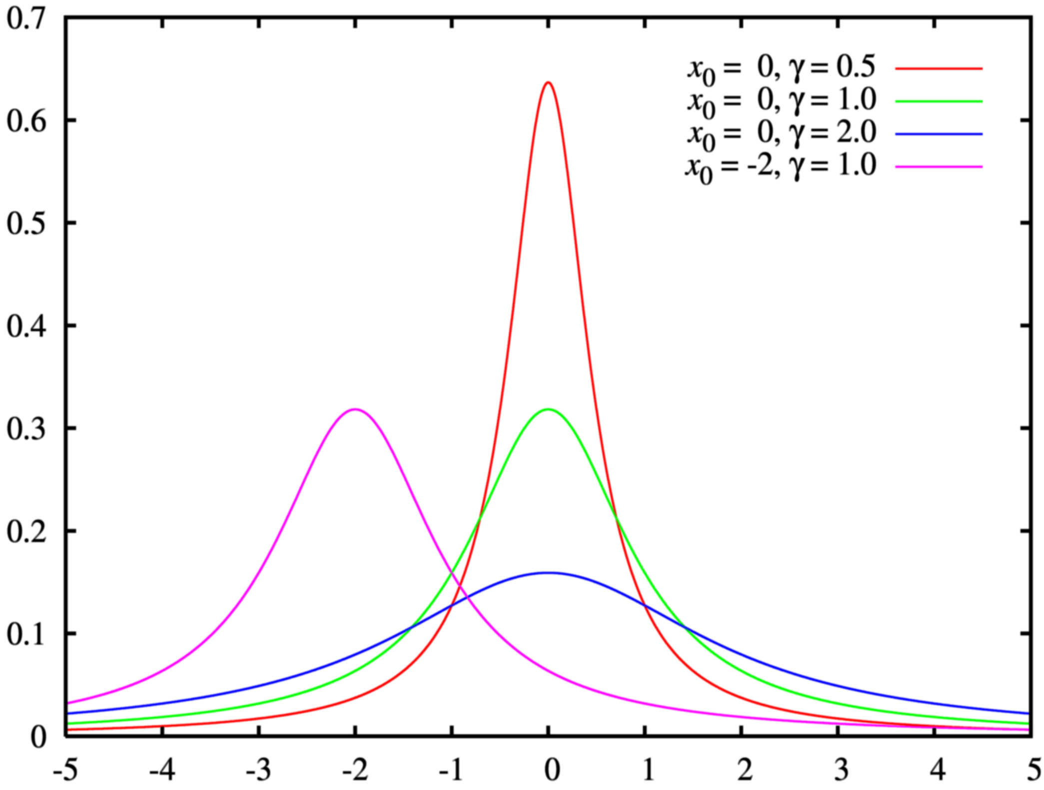

FIGURE 6 – Distribution théorique de la loi Cauchy. Source : [1]

a est dite l'échelle de la fonction, et x_0 est sa médiane.
La loi de Cauchy n'admet ni espérance ni écart type.
La fonction de répartition :

$$F_X(x) = \frac{1}{\pi} \arctan\left(\frac{x-x_0}{a}\right) + \frac{1}{2} \tag{2.2}$$

2.2 Estimer les paramètres inconnus

Calcul théorique

Soient n réalisations $x_1, x_2, ..., x_n$. Assumons que ces données suivent la loi de Cauchy (2.1).

Nous allons utiliser la méthode de maximum de rapport de vraisemblance.

En l'absence d'information de la distribution, nous assumons que ces mesures sont indépendants. Posons une variable aléatoire X_i correspondante à chaque réalisation x_i pour $i = \overline{1, n}$.

On écrit la loi conjointe de ces n variables :

$$\begin{aligned} L(X) = L\left(X_1, X_2, ..., X_n\right) &= \prod_{i=1}^{n} f_X\left(X_i\right) \\ &= \prod_{i=1}^{n} \frac{1}{\pi} \frac{a}{\left(x_i - x_0\right)^2 + a} \\ &= \pi^{-n} \prod_{i=1}^{n} \frac{a}{\left(x_i - x_0\right)^2 + a} \end{aligned}$$

On cherche l'optimum maximale.

$$\begin{aligned} \frac{\partial L}{\partial x_0}(X) &= \pi^{-n} \sum_{j=1}^{n} \left(-\frac{a\left(2x_j - 2x_0\right)}{\left[\left(x_0 - x_j\right)^2 + a\right]^2} \prod_{i=1;i\neq j}^{n} \frac{a}{\left(x_i - x_0\right)^2 + a} \right) \\ &= \pi^{-n} \left(\prod_{i=1}^{n} \frac{a}{\left(x_i - x_0\right)^2 + a} \right) \left(-\sum_{j=1}^{n} \frac{2x_0 - 2x_j}{\left(x_0 - x_j\right)^2 + a} \right) \\ \frac{\partial L}{\partial x_0}(X) = 0 &\Leftrightarrow \sum_{j=1}^{n} \frac{x_0 - x_j}{\left(x_0 - x_j\right)^2 + a} = 0 : \text{insolvable par la main} \end{aligned}$$

$$\begin{aligned} \frac{\partial L}{\partial a}(X) &= \pi^{-n} \sum_{j=1}^{n} \left(\frac{\left(x_j - x_0\right)^2 + a - a}{\left(x_j - x_0\right)^2 + a} \prod_{i=1;i\neq j}^{n} \frac{a}{\left(x_i - x_0\right)^2 + a} \right) \\ &= \pi^{-n} a^{n-1} \frac{\sum_{j=1}^{n} \left(x_0 - x_j\right)^2}{\prod_{i=1}^{n} \left(\left(x_i - x_0\right)^2 + a\right)} > 0,\ \forall a > 0 \end{aligned}$$

On ne peut donc pas estimer a par la méthode du max

Résolution par R

On génère un échantillon avec a et x_0 au choix. Puis, on utilise la fonction *mle* (Maximum Likelihood Estimator) de librairie *stats4* avec 2 valeurs initiales $\hat{a}$ et $\hat{x_0}$ pour estimer a et x_0.

Nous essayons d'estimer x_0 et a directment. L'algorithme d'approximation implémenté dans R a besoin d'un bon point de départ, sinon le résultat obtenu variera grossièrement.

— Etant donné que la médiane est théoriquement aussi x_0, choisissons x_0 comme la médiane de l'échantillon.

— Nous avons $f_X\left(x_0\right) = \frac{1}{\pi a}$. En plus, $f_X(x_0)$ est la valeur maximale d de densité. Prenons alors $\hat{a} = \frac{1}{\pi d}$.

Testons avec $x_0 = 13$ et $a = 0.5$.

PROBLEME UTF8 Cauchy.R

```
   # fixer le gemme et paramètres réels
set.seed(2018)
real_x0 = 13
real_a = 0.5
nbreaks = 40

# générér échantillons
N = 1000
x = rcauchy(n = N,
            location = real_x0,
            scale = real_a)

# définir fonction log négative pour MLE
ll = function(location, scale) {
    dist = suppressWarnings(dcauchy(x,
                                    location,
                                    scale,
                                    log = TRUE))
    -sum(dist)
}

# MLE (Maximum Likelihood Estimator)
library(stats4)
hat_x0 = median(x)
d = hist(x,
         breaks = nbreaks,
         plot = FALSE) # histogram
hat_a = 1 / (pi * max(d[["density"]]))
result = mle(ll,
             start = list(location = hat_x0, scale = hat_a))

# illustrer
hist(x,
     freq = F,
     breaks = nbreaks,
     col = "green",
     xlab = "x",
     ylab = "Densité",
     ylim = c(0, 0.2),
     main = "Loi de Cauchy - réalité vs théorique") # histogram
curve(dcauchy(x, real_x0, real_a),
```

```
       add = TRUE,
       col = "black") # theoretic
curve(dcauchy(x, result@coef[1], result@coef[2]),
       add = TRUE,
       col = "red") # calculated
legend("topright",
        legend = c("histogramme", "théorique", "réalité"),
        fill = c("green", "black", "red"))
```

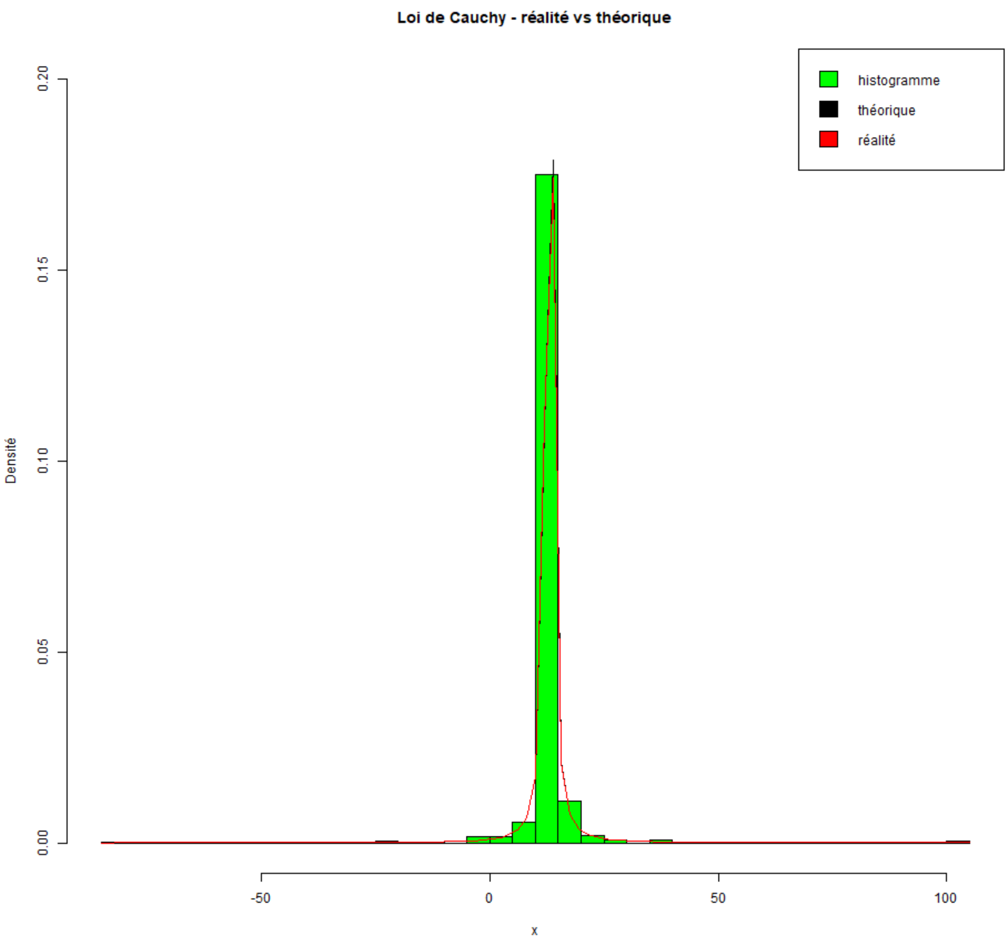

FIGURE 7 – Représentation de l'histogramme et de la courbe de densité

Pour $x_0 = 13$ et $a = 0.5$, les valeurs trouvées par R sont $\hat{x_0} = 12.9877139$ et $\hat{a} = 0.4951073$.

2.3 Test de paramètres

Nous testons ici si les valeurs trouvées par R sont vraiment approchées à celles théoriques. Pour l'instant, nous n'avons pas d'outils pour vérifier 2 variables en couple.

2.4 Test d'adéquation

R nous donne la fonction *ks.test* pour valider la compatibilité d'une échantillon avec une loi selon le test de Kolmogorov-Smirnov.

```
ks.test(x, "pcauchy", real_x0, real_a)
```

On a trouvé $D = 0.02696$, $p - value = 0.4614$, signifiant l'écart maximale est 0.02696 et le niveau d'acceptance est 0.4614. Vu que nous fixons $\alpha = 5\% < p - value$, l'échantillon dépasse largement le test. Autrement dit, il se distribue selon la loi de Cauchy.

2.5 Etude de biais d'estimateur

Maintenant on s'intéresse au biais d'estimateur. On relance l'algorithme en-dessus avec de différents observations ($N = 100, 125, 150, ..., 1100$). $x_0 = 13$ et $a = 0.5$.

```
library(stats4)
# définir fonction log négative pour MLE
ll = function(location, scale) {
    dist = suppressWarnings(dcauchy(x,
                                    location,
                                    scale,
                                    log = TRUE))
    - sum(dist)
}

# fixer les paramètres réels
set.seed(2018)
real_x0 = 13
real_a = 0.5
nbreaks = 40
minN = 100
numN = 40
delta = 25
maxN = minN + (numN - 1) * delta

Ns = seq.int(minN, maxN, delta) # tous 40 valeurs de n de 100 à 1000
results = array(NA, c(2, numN)) # à contenir tous les résultats

for (i in 1:numN) {
    N = minN + (i - 1) * delta

    # générér échantillon
    x = rcauchy(n = N,
                location = real_x0,
```

```
                 scale = real_a)

    # MLE (Maximum Likelihood Estimator)
    hat_x0 = median(x)
    d = hist(x,
             breaks = nbreaks,
             plot = FALSE) # histogram
    hat_a = 1 / (pi * max(d[["density"]]))
    result = mle(ll,
             start = list(location = hat_x0, scale = hat_a))

    # récupérer le résultat
    results[1, i] = result@coef[1] # x_0
    results[2, i] = result@coef[2] # a
}

# illustrer
plot(x = results[1, ],
     y = results[2, ],
     col = rgb(0, 100, 0, 70, maxColorValue = 255),
     pch = 16,
     cex = 1.5,
     main = "Convergence d'estimateurs",
     xlab = "Location x_0",
     ylab = "Echelle a"
     )
points(x = real_x0,
     y = real_a,
     col = "red",
     pch = 16,
     cex = 2
     )

legend("topleft",
       legend = c("valeur théorique", "valeur estimée"),
       fill = c("red", rgb(0, 100, 0, 70, maxColorValue = 255)))
```

Nous observons que les valeurs estimées se rassemblent assez proches de la valeur théorique.

2.6 Etude de loi d'extremum

Générons donc des échantillons de n valeurs ($n = 10, 100, 1000, etc.$). Parmi chacun, $\frac{n}{10}$ valeurs les plus grandes seront gardées pour étudier la loi d'extremum.

Comme les réalisations sont indépendantes et identiquement distribuées,

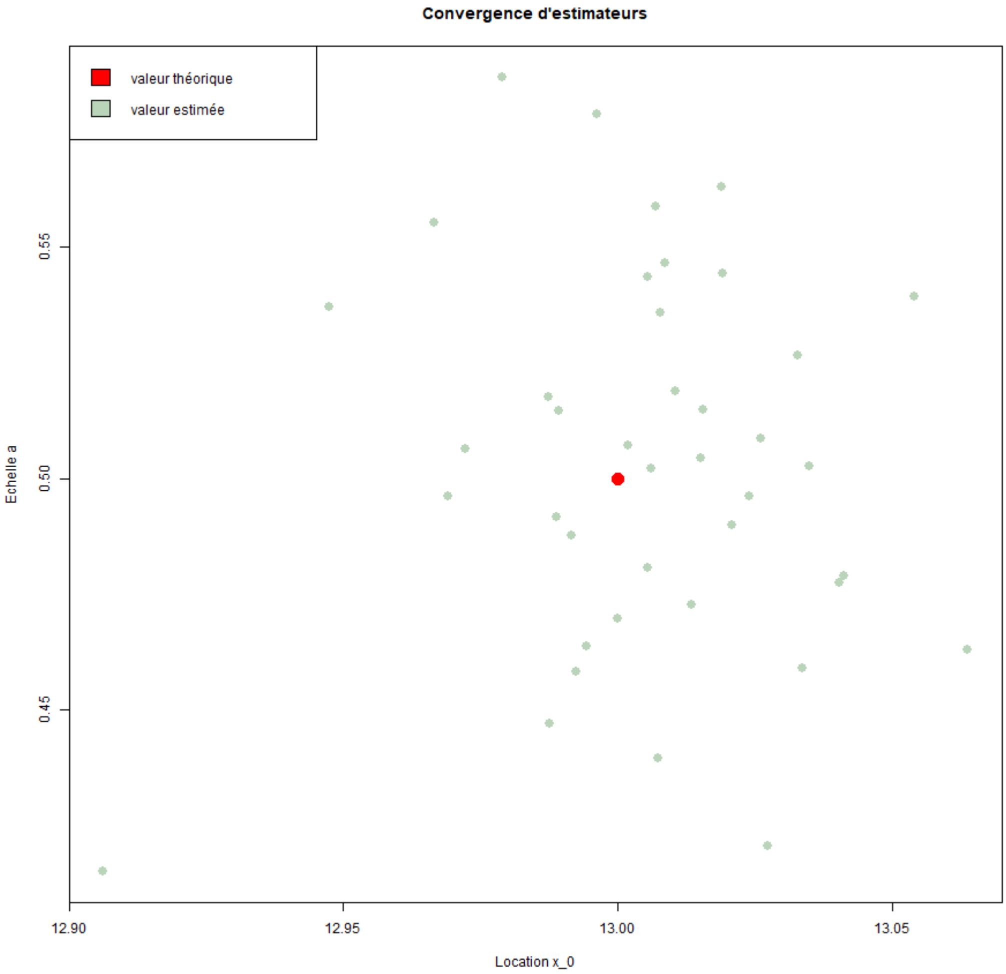

Figure 8 – Convergences de $\hat{x_0}$ et $\hat{a}$ vers valeurs théoriques

théoriquement :

$$\begin{aligned}
F_{x_{\max}}(x) &= P\left(X_{\max} \leq x\right) \\
&= P\left(X_1 \leq x, ..., X_n \leq x\right) \\
&= \prod_{i=1}^{n} F_X(x) \\
&= F_X(x)^n
\end{aligned}$$

Avec $F_X(x) = \frac{1}{\pi}\arctan\frac{x-x_0}{a} + \frac{1}{2}$, la fonction de répartition de la loi de

Cauchy. Alors,

$$F_{x_{\max}}(x) = \left(\frac{1}{\pi}\arctan\frac{x-x_0}{a} + \frac{1}{2}\right)^n$$

Or,

$$\left|\arctan\frac{x-x_0}{a}\right| < \frac{\pi}{2} \Rightarrow \frac{-\pi}{2} < \arctan\frac{x-x_0}{a} < \frac{\pi}{2} \Rightarrow 0 < \arctan\frac{x-x_0}{a} + \frac{\pi}{2} < \pi$$

Par conséquent,

$$\begin{aligned}
&\tan\left(\arctan\tfrac{x-x_0}{a} + \tfrac{\pi}{2}\right) = -\cot\left(\arctan\tfrac{x-x_0}{a}\right) = -\tfrac{a}{x-x_0} \\
&\Rightarrow \arctan\tfrac{x-x_0}{a} + \tfrac{\pi}{2} = \arctan\left(-\tfrac{a}{x-x_0}\right) = \arctan\tfrac{a}{x_0-x} \\
&\Rightarrow F_{x_{\max}}(x) = \pi^{-n}\left(\arctan\tfrac{x-x_0}{a} + \tfrac{\pi}{2}\right)^n = \pi^{-n}\arctan^n\tfrac{a}{x_0-x}
\end{aligned}$$

Nous prenons 3 valeurs les plus grandes pour chaque itération, et établissons leur histogramme.

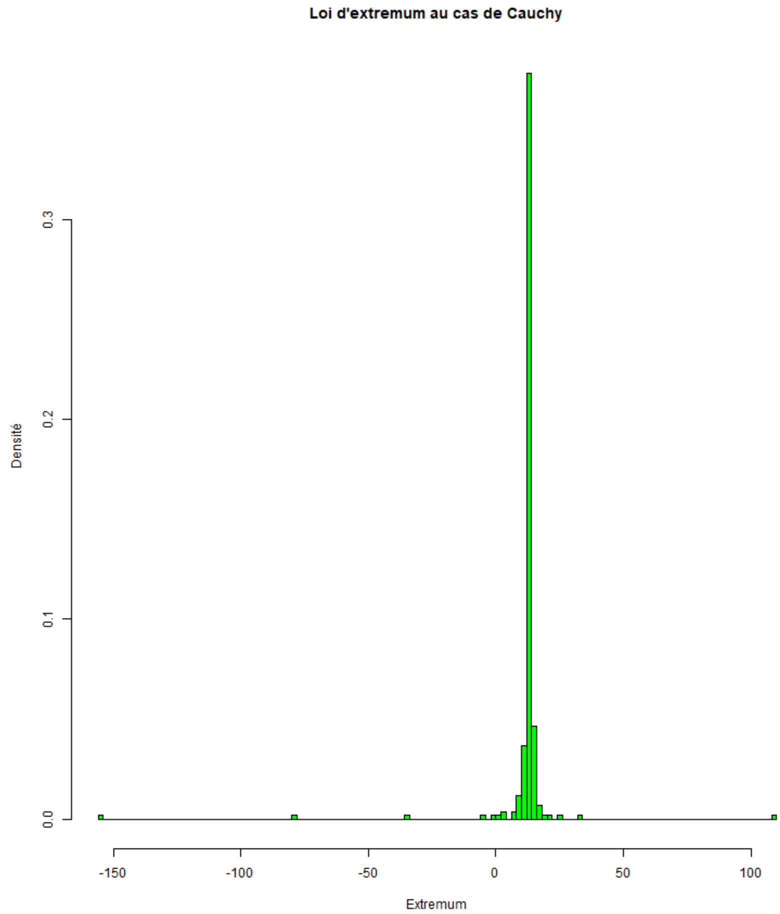

FIGURE 9 – Histogramme de valeurs extremum de la distribution de Cauchy

Afin d'approximer la loi d'extremum, nous ne regardons que les réalisations positives. En utilisant le package *poweRlaw*, nous calculons la compatibilité de la loi d'extremum.

```
results = na.omit(results) # filtrer les NA
library(poweRlaw)
positiveResults = results[which(results > 0)] # filtrer les négatifs
cauchy_pl = conpl$new(positiveResults) # objet de calculer la loi d'extremum
est = estimate_xmin(cauchy_pl) # estimer la borne inférieure
cauchy_pl$setXmin(est) # mettre à jour l'objet de distribution
plot(cauchy_pl, col = "green")
lines(cauchy_pl, col = "red")
bs_p = bootstrap_p(cauchy_pl) # tester l'estimation
bs_p$p
```

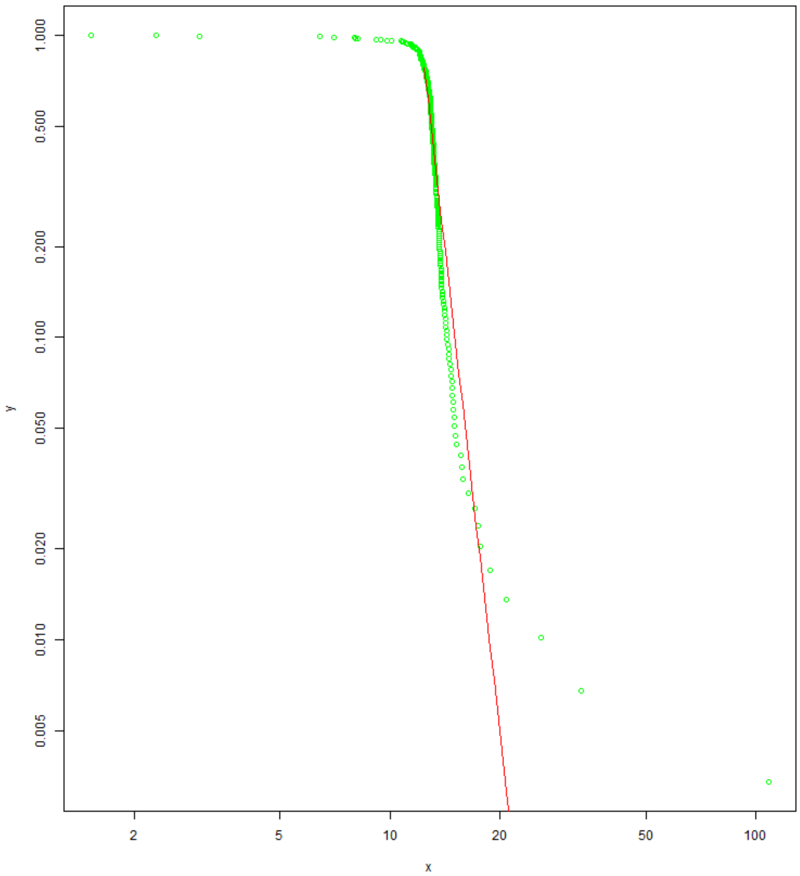

FIGURE 10 – Approcher la loi d'extremum avec celle des maximales de Cauchy. Les données (verte) ne s'approchent pas totalement de l'approximation (rouge).

Calcul de $p - value$ nous retourne 0, qui signifie l'incompatibilité de ces deux lois.

Références

[1] Wikipedia,
https://en.wikipedia.org/wiki/Cauchy_distribution https://en.wikipedia.org/wiki/Generalized_extreme_value_distribution