

Treball de Fi de grau

## Valoració del risc en el mercat continu

---

Albert Pitarque Méndez

Juny 2017

Grau d'Estadística Aplicada

Tutora: Isabel Serra

Cotutora: Alejandra Cabaña



## Agraïments

Agraeixo a la Universitat Autònoma de Barcelona per haver-me donat la oportunitat d'estudiar la carrera d'Estadística durant els últims 4 anys i als professors de la carrera els quals m'han fet aprendre i anar descobrint a poc a poc els diversos àmbits de la Estadística. Agraeixo també als membres del Servei d'Estadística els ànims en fer aquest treball i motivar-me a seguir estudiant.

Tot i que estic agraït a tots els professors i professors, cal destacar-ne a dues en especial. Per una banda a l'Alejandra Cabaña que m'ha ajudat molt en el treball, sobretot en la part del test de permutacions i en els últims dies del treball. Per altre banda a la Isabel Serra que em va introduir a la teoria dels valors extrems, em va proposar enfocar aquest treball en aquesta direcció i em va orientar quan no sabia com continuar. La veritat és que aquest treball, sense la orientació i motivació que m'han donat i sobretot, la gran quantitat d'hores de reunions que hem tingut, no hagués estat possible.

Per últim vull fer un agraïment especial a la meva família. Ja desde petit han estat animant-me a estudiar i a fer-me veure quin era el millor camí. Sempre han estat amb mi, tant en les bones com en les dolentes. Van ser els que em van impulsar a estudiar estadística i els que durant la elaboració d'aquest treball més m'han animat i recolzat.

# Índex

|  |           |
|--|-----------|
| <b>1 Plantejament del problema</b>                           | <b>6</b>  |
| <b>2 Metodologia</b>   | <b>6</b>  |
| 2.1 Test de permutacions . . . . .                           | 6         |
| 2.2 Teoria dels valors extrems. . . . .                      | 7         |
| 2.3 Clauset-Estimador Estàndard . . . . .                    | 7         |
| 2.4 Mètode CV . . . . .                                      | 8         |
| <b>3 Resultat</b>  | <b>9</b>  |
| 3.1 Desenvolupament del test teòric exacte . . . . .         | 9         |
| <b>4 Potència empírica del test - Mètode de Monte Carlo</b>  | <b>11</b> |
| 4.1 Potència del test amb el mètode de Clauset . . . . .     | 12        |
| 4.2 Potència del test amb el mètode CV . . . . .             | 12        |
| <b>5 Aplicació del test a les dades d'un actiu financer</b>  | <b>13</b> |
| 5.1 Netflix . . . . .  | 13        |
| 5.2 SPS . . . . .  | 16        |
| 5.3 McDonald's . . . . .                                     | 19        |
| <b>6 Aplicació del test a una cartera d'actius financers</b> | <b>22</b> |
| 6.1 Cartera formada per dos actius financers . . . . .       | 22        |
| 6.2 Cartera formada per 3 actius financers . . . . .         | 25        |
| <b>7 Conclusions</b>   | <b>26</b> |
| <b>8 Referències</b>   | <b>27</b> |
| <b>A Annex</b>   | <b>28</b> |
| A.1 Desenvolupament del test teòric exacte . . . . .         | 28        |
| A.2 Plfit . . . . .  | 28        |
| A.3 Simulació dels resultats (Mètode de Clauset)             | 35        |
| A.4 Llibreria ercv . . . . .                                 | 36        |
| A.5 Funció thrselect . . . . .                               | 46        |
| A.6 Potència del test mètode CV . . . . .                    | 48        |
| A.7 Estudi de les accions . . . . .                          | 49        |
| A.8 Estudi de una cartera de dues accions . . . . .          | 52        |
| A.9 Estudi de la cartera de 3 accions . . . . .              | 54        |

# Valoració del risc en el mercat continu

Albert Pitarque Méndez

Isabel Serra

Alejandra Cabaña

Paraules clau: *Carteres, Cues, Rendibilitats, Actius, Permutacions, Risc*

En aquest treball s'ha fet un estudi sobre el risc en les carteres d'actius financers.

En el primer capítol fem un plantejament del problema que volem estudiar. El que volem és mirar si les cues d'una cartera d'actius financers poden ser considerades iguals. En aquesta part del treball expliquem quin és el contrast d'hipòtesis que farem i explicarem que és una cua.

En el capítol 2 comentarem la metodologia del treball. En aquest apartat s'explicarà que és un test de permutacions i com fer-lo. Es farà una petita explicació sobre la teoria dels valors extrems i s'explicaran el mètode de Clauset i del coeficient de variació residual (CV) que són dos mètodes de modelització de cues i els que hem utilitzat en la elaboració d'aquest estudi.

En el tercer capítol desenvolupem el test de forma teòrica. Primer de tot veiem d'on surt el estimador màxim versemblant. Veurem que el nostre estadístic de contrast segueix una distribució  $\beta'$  que només depèn del nombre de valors positius i negatius de les dades. Hem mirat com variant aquest valor canvien les regions d'acceptació del test teòric.

A continuació calculem la potència dels tests pels diferents mètodes generant diversos valors de distribucions t-Student diferents. Veurem que amb el mètode de Clauset el test funciona molt millor que amb el mètode CV.

En el cinquè capítol aplicarem el test a les dades d'actius financers. Aquests seran Netflix, SPS i McDonald's. Per a cada un d'ells farem estadística descriptiva i aplicarem el test. Veurem que per cap d'ells podem considerar que les cues són diferents.

Per últim crearem carteres de dos i tres actius financers i aplicarem el test a dades reals. Veurem que per al cas de dos actius, amb les dades que tenim, no és possible crear carteres que tinguin més risc de guanys inesperats que pèrdues inesperades, però en el cas de la cartera de 3 actius si que és possible.

# Valoración del riesgo en el mercado continuo

Albert Pitarque Méndez

Isabel Serra

Alejandra Cabaña

Palabras clave: *Carteras, Colas, Rendibilidades, Activos, Permutaciones, Riesgo*

En este trabajo se ha realizado un estudio sobre el riesgo en las carteras de activos financieros

En el primer capítulo hacemos un planteamiento del problema que queremos estudiar. Lo que queremos hacer es mirar si las colas de una cartera de activos financieros pueden ser consideradas iguales. En esta parte del trabajo se explica que contraste de hipótesis llevaremos a cabo i explicaremos que es una cola.

En el capítulo 2 comentaremos la metodología del trabajo. En este apartado se explicará que es un test de permutaciones y como llevarlo a cabo. Se hará una pequeña explicación sobre la teoría de los valores extremos i se explicarán el método de Clauset i el del coeficiente de variación residual (CV) que son dos métodos de modelización de colas y los que hemos usado en la elaboración de este estudio.

En el tercer capítulo desarrollamos el test de forma teórica. En primer lugar veremos de donde sale el estimador de máxima verosimilitud. Veremos que nuestro estadístico de contraste sigue una distribución  $\beta'$  que depende únicamente del numero de valores positivos y negativos de los datos. Hemos observado como variando estos valores, cambian las regiones de aceptación de nuestro test teórico.

A continuación calculamos la potencia de los tests usando los diferentes métodos generando diversos valores de distribuciones t-Student diferentes. Veremos que con el método de Clauset el test funciona mucho mejor que con el método CV.

En el quinto capítulo aplicaremos el test a los datos de activos financieros. Estos serán Netflix, SPS y McDonald's. Para cada uno de ellos haremos estadística descriptiva i aplicaremos el test. Veremos que en ninguno de ellos podemos considerar que las colas son diferentes.

Finalmente crearemos carteras de dos y tres activos financieros y aplicaremos el test a datos reales. Veremos que en el caso de dos activos, con los datos que tenemos, no es posible crear carteras que tengan más riesgo de ganancias inesperadas que pérdidas inesperadas, pero en el caso de la cartera de 3 activos si que es posible.

# Risk Assessment in Continuous Markets

Albert Pitarque Méndez

Isabel Serra

Alejandra Cabaña

Palabras clave: *Portfolio, Tails, Returns, Assets, Permutations, Risk*

In this study we take a look on the portfolio's risk.

In the first section we raise the problem that we want to study. Our main objective is determine if the tails of a distribution can be considered equal. In this part of the study we explain the statistical hypothesis testing that we made and explain what a tail is.

In the section 2 we will talk about the methods used in the study. In this section we will explain what a permutation test is and how to do them. We will make a brief explanation about the extreme value theory and we will apply the Clauset method and the coefficient of variation method (CV) that are two methods designed for modeling tail distributions and the ones that we used.

In the next section we develop a theoretical test. First of all we will see how can we obtain the maximum likelihood estimation. We will see that the distribution of our statistic follows a  $\beta'$  distribution that only depends of the number of positive and negative values of the data. We observed how changing this values affect at the region of acceptance of our theoretical test.

In the next section we calculate the statistical power of the test using the two methods and generating values of different t-Student distributions. We will see how Clauset method works better than the CV method.

In the fifth chapter we will apply the test on assets data. Those will be Netflix, SPS and McDonald's. For each of them we will get de descriptive statistics and we will apply the test. We will observe that in none of them we can considerate that the tails are different.

Finally we will create portfolios of 2 and 3 assets and we will apply the test on real data. We will note that in the two assets case is impossible to create a portfolio that has more risk of earnings than loses with the data that we have but in the case of 3 assets we can make it.

# 1 Plantejament del problema

L'objectiu d'aquest estudi és elaborar un test que ens permeti veure si les cues de la dreta i de l'esquerra d'una distribució es poden considerar igualment pesades. Les cues fan referència a allò que passa, segons la estadística habitual una de cada 20 o 100 vegades, és a dir amb molt poca freqüència. La distribució de l'estadístic del test es trobarà mitjançant un test de permutacions.

La llei dels extrems d'una col·lecció de variables aleatòries, es pot caracteritzar amb un paràmetre  $\alpha \in \mathbb{R}$ . Per a cada una de les cues s'ha estimat el valor d' $\alpha$  i a partir d'aquests estimadors utilitzarem el quocient per veure si és raonable pensar que els índex de les cues són iguals. Un cop hem trobat l'estadístic de contrast del nostre test hem de plantejar les hipòtesis.

Definim  $\alpha_+$  com el valor  $\alpha$  que caracteritza la cua de la dreta i  $\alpha_-$  com al valor  $\alpha$  que caracteritza la cua esquerra. Com hem dit abans el nostre objectiu és comprovar si les distribucions de les cues podien ser considerades iguals. Si això passes consideraríem que els valors d' $\alpha_+ i \alpha_-$  són el mateix. És per això que les hipòtesis seran:

$$\begin{cases} H_0: \frac{\alpha_+}{\alpha_-} = 1 \\ H_1: \frac{\alpha_+}{\alpha_-} \neq 1 \end{cases}$$

En el cas que es compleixi la hipòtesi nul·la voldrà dir que les cues poden ser considerades iguals. Per calcular la distribució de l'estadístic d'aquest test es farà un test de permutacions. Aquest mètode serà explicat a l'apartat 2.1.

# 2 Metodologia

## 2.1 Test de permutacions

En aquest estudi, una part important del treball d'investigació utilitza els tests de permutacions. Els tests de permutacions van ser introduïts per Fisher però no eren fàcils d'implementar en aquella època. Ara, gràcies a la potència dels ordinadors es poden obtenir bons resultats.

El procediment utilitzat per a realitzar un test de permutacions és el següent:

1. Especificar les hipòtesis nul·la i alternativa.
2. Definir l'estadístic de contrast que utilitzarem per fer el test.
3. Definir unes dades noves que s'obtindran a partir del remostreig sense reposició d'aquestes.
4. Amb aquestes noves dades calcular l'estadístic de contrast.
5. Repetir els passos 2 i 3 un nombre determinat de vegades. Repetint aquest pas 1000 vegades és considerat suficient. Veure [2].
6. Si el valor de l'estadístic de contrast de les dades originals és més que el 95% dels estadístics de les mostres considerarem que podem rebutjar la hipòtesi nul·la.

En aquest estudi utilitzarem aquest tipus de test per tal de comparar si les dues cues de la distribució de les rendibilitats de certs actius financers poden ser considerades iguals.

## 2.2 Teoria dels valors extrems.

Suposem que  $X_1, X_2, \dots$ , són variables aleatòries independents idènticament distribuïdes. Prenem com a  $M_n$  el valor màxim d'aquestes variables aleatòries que normalment són valors que s'han mesurat amb certa periodicitat. En teoria, la distribució de  $M_n$  es pot derivar donant com a resultat:

$$P\{M_n \leq z\} = P\{X_1 \leq z, \dots, X_n \leq z\} = P\{X_1 \leq z\} \cdot \dots \cdot P\{X_n \leq z\} = \{F(z)\}^n \quad (1)$$

Aquesta funció de distribució  $F$ , que és la que segueixen les variables  $X_1, X_2, \dots$ , és desconeguda. Una de les possibilitats per a fer la estimació sobre quina és aquesta distribució  $F(z)$ , seria utilitzar els valors obtinguts de les variables i fer una aproximació. El problema està en que un petit error a la funció  $F$ , pot suposar un gran error en la funció  $F^n$ .

Una altre possibilitat és acceptar que desconeixem la funció de distribució. Aleshores per fer una aproximació de la funció observarem el comportament de  $F$  quan  $n$  tendeix a infinit. Veurem, que per aquells valors de les dades en els que la funció de distribució és inferior a 1,  $F^n$  tendeix a 0 quan  $n \rightarrow 0$  però, quan la funció de distribució pren valor 1,  $F^n \rightarrow 1$  de tal manera que la funció  $M_n$  degenera en cert punt.

Per tal de corregir això, es pot rescalar  $M_n$  fent que obtinguem una nova funció de distribució  $G(z)$  que és una funció que no degenera a cap valor. Aquesta funció, depenent dels paràmetres pot aproximar-se a una funció Gumbel, Frécht o Weibull. Aquestes tres distribucions es poden escriure en funció de tres paràmetres. Un d'ells la anomenat /textitShape que ens indicarà en quin dels 3 possibles límits ens trobem. Aquestes distribucions es coneixen com la GEV de l'anglès Generalized Extreme Value

La funció  $G(z)$  té una distribució:

$$G(Z) = \exp \left\{ -[1 + \alpha \left( \frac{z - \mu}{\sigma} \right)^{-\frac{1}{\alpha}}] \right\} \quad (2)$$

on  $\mu$  i  $\sigma > 0$ . Aleshores per un valor suficientment gran de  $\mu$ , la funció de distribució de  $X - \mu$ , condicionada a que  $X > \mu$  és aproximadament:

$$H(y) = 1 - \left( 1 + \frac{\alpha y}{\psi} \right)^{-\alpha} \quad (3)$$

La família de distribucions que hem definit en la equació (3) s'anomena família de distribucions de Pareto generalitzades (GPD). Els valors dels paràmetres de la distribució GPD venen determinats únicament per els corresponents a la distribució GEV però si tinguessim una mostra molt gran, els paràmetres de la distribució GEV es veurien afectats mentre que els de la GPD no. El paràmetre  $\alpha$  també definit en la equació (3) es coneix com a index de la cua i és el valor que ens interessa estudiar ja que és el que ens permet modelitzar les cues.

## 2.3 Clauset-Estimador Estàndard

El mètode de Clauset és un dels mètodes que hem fet servir per calcular el valor  $\alpha$  del nostre estimador. Tal i com hem vist en l'apartat 2.2, podem aproximar la distribució de les cues per una  $GPD(\sigma, \alpha)$  però, el mètode de Clauset consisteix en fixar el valor de  $\sigma$  de la equació (2) de tal manera que la distribució passa a ser una  $PL_\sigma(\alpha)$  d'on fa la estimació del paràmetre. Aquest valor estimat és el que nosaltres utilitzarem per calcular l'estadístic del test.

La distribució Power-law (PL) és un tipus especial de distribució de probabilitat que té diverses formes de ser definida matemàticament.

$$p(x) = Cx^{-\alpha} \quad \text{per } x \geq x_{min} \quad \text{i } x \geq 0 \quad (4)$$

On la constant normalitzadora  $C = (\alpha - 1)x_{min}^{\alpha-1}$ . Cal destacar que és imprescindible que  $\alpha > 1$  ja que és la unica manera que la funció normalitzi. Sabent això, una altra manera de definir aquesta funció és:

$$p(x) = \frac{\alpha - 1}{x_{min}} \left( \frac{x}{x_{min}} \right)^{-\alpha} \quad \text{per } x \geq x_{min} \quad \text{i } x \geq 0 \quad (5)$$

En aquest estudi, suposem que les variàncies de la cua positiva i de la cua negativa són semblants és per això que ignorem el paràmetre escala de la GPD i ajustem la distribució de la cua a una funció PL. Tot i així, com a continuació d'aquest treball, es podria probar a ajustar la funció a una distribució GPD per veure com afecta als resultats.

## 2.4 Mètode CV

El mètode del coeficient de variació residual (CV) per modelar valors extrems funciona consisteix en el següent:

Sigui  $X$  una funció continua no negativa amb funció de distribució  $F(X)$ . Per qualsevol llindar  $t > 0$ , la funció de distribució de la funció condicional a l'excés de la variable aleatòria  $X-t$  denotada  $(X-t|X > t)$  rep el nom de distribució residual de  $X$  sobre  $t$  i ve donada per

$$1 - F_t(X) = \frac{1 - F(x+t)}{1 - F(t)} \quad (6)$$

La quantitat  $M(t) = E(X_t)$  s'anomena mitjana residual i  $V(t) = \text{var}(X_t)$  s'anomena variància residual. El coeficient de variació (CV) ve donat per

$$CV(t) = CV(X_t) = \sqrt{V(t)}/M(t) \quad (7)$$

Aquesta funció és independent del paràmetre escala. Un mètode que s'està utilitzant a l'hora d'analitzar cues és el CV-plot que consisteix en aplicar la funció CV als excessos de la mostra on  $t$  seran els estadístics d'ordre.

En aquest treball hem utilitzat aquest mètode per trobar a partir de quan es considerava que començava la cua. Un cop havíem trobat aquest valor utilitzàvem l'estimació màxim versemblant per calcular  $\alpha$ . Veure apartat 2.3

### 3 Resultat

#### 3.1 Desenvolupament del test teòric exacte

Tal i com hem comentat en l'apartat 1 l'objectiu del nostre estudi és crear un test que ens permeti saber si les lleis dels extrems d'una distribució poden ser considerades iguals. Així doncs ens plantegem el contrast d'hipòtesis:

$$\begin{cases} H_0: \frac{\alpha_+}{\alpha_-} = 1 \\ H_1: \frac{\alpha_+}{\alpha_-} \neq 1 \end{cases}$$

Aquest valor  $\alpha$  té diversos mètodes per ser calculat. En aquest estudi hem utilitzat el mètode de Clauset. Clauset parteix de que la distribució que segueixen les dades és una distribució  $\beta'(n_1, n_2)$  on  $n_1$  és el nombre de valors que considerem extrems per l'extrem positiu i  $n_2$  és el nombre de valors que considerem extrems per l'extrem negatiu.

**Teorema 1** Sigui  $\hat{\alpha}_1$  i  $\hat{\alpha}_2$  els estimadors màxim versemblants de  $X_1$  i  $X_2$  i  $x_{ij}$  el valor  $j$  de la variable  $i$  on  $j \in [1, n_i]$ .  $X_i \sim PL \rightarrow \frac{\hat{\alpha}_1}{\hat{\alpha}_2} \sim \beta'(n_1, n_2)$

Demostració

$$\hat{\alpha}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \log \left( \frac{x_{ij}}{x_{min_i}} \right)$$

d'on tenim que

$$\sum_{j=1}^{n_i} \log \left( \frac{x_{ij}}{x_{min_i}} \right) \sim Gamma(n_i, \alpha)$$

ja que

$$\log \left( \frac{x_{ij}}{x_{min_i}} \right) \sim Exp(\alpha)$$

i la suma d'exponentials dona lloc a una distribució Gamma. En aquestes definicions,  $x$  representa les dades pertanyents als valors extrems que segueixen una distribució  $PL(\alpha)$  i  $x_{min}$  representa el valor a partir del qual es considera que les dades són valors extrems.

Un cop definit que  $\alpha$  segueix una distribució Gamma, el nostre estadístic seguirà la següent distribució:

$$\frac{\alpha_+}{\alpha_-} = \frac{Gamma(n_+, \alpha_+)}{Gamma(n_-, \alpha_-)} \sim \beta'(n_+, n_-)$$

De la distribució que segueix el nostre estimador podem extreure uns valors crítics que serviran per valorar si el quocient és o no és igual a 1. Aquests valors critics dependran del nombre de valors que tinguem a cada cua. La taula i el gràfic que es presenten a continuació mostren com van variant els punts critics a mesura que va canviant la mida de les cues.

Tant en la taula 1 com en el gràfic 1 veiem com a mesura que es van fent grans les mides de les cues, el valor del límit inferior es va fent més gran i el valor del límit superior es va fent més petit. En unes dades reals, nosaltres desconeixem la distribució que segueixen les dades i és per això que quan fem el test no podem utilitzar el test exacte. La distribució que presenta Clauset és només una distribució teòrica.

| $n_- \backslash n_+$ | 10             | 20             | 40             | 60             | 80             | 100            |
|----------------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 10                   | [0.406, 2.464] | [0.437, 2.068] | [0.457, 1.884] | [0.464, 1.825] | [0.468, 1.796] | [0.47, 1.778]  |
| 20                   | [0.484, 2.287] | [0.533, 1.875] | [0.567, 1.679] | [0.58, 1.614]  | [0.587, 1.582] | [0.592, 1.562] |
| 40                   | [0.531, 2.19]  | [0.596, 1.764] | [0.643, 1.555] | [0.663, 1.483] | [0.674, 1.447] | [0.682, 1.425] |
| 60                   | [0.548, 2.156] | [0.62, 1.724]  | [0.674, 1.508] | [0.698, 1.433] | [0.712, 1.394] | [0.721, 1.37]  |
| 80                   | [0.557, 2.139] | [0.632, 1.703] | [0.691, 1.483] | [0.717, 1.405] | [0.733, 1.365] | [0.743, 1.34]  |
| 100                  | [0.562, 2.128] | [0.64, 1.691]  | [0.702, 1.468] | [0.73, 1.388]  | [0.746, 1.346] | [0.757, 1.32]  |

Taula 1: Regió d'acceptació del test

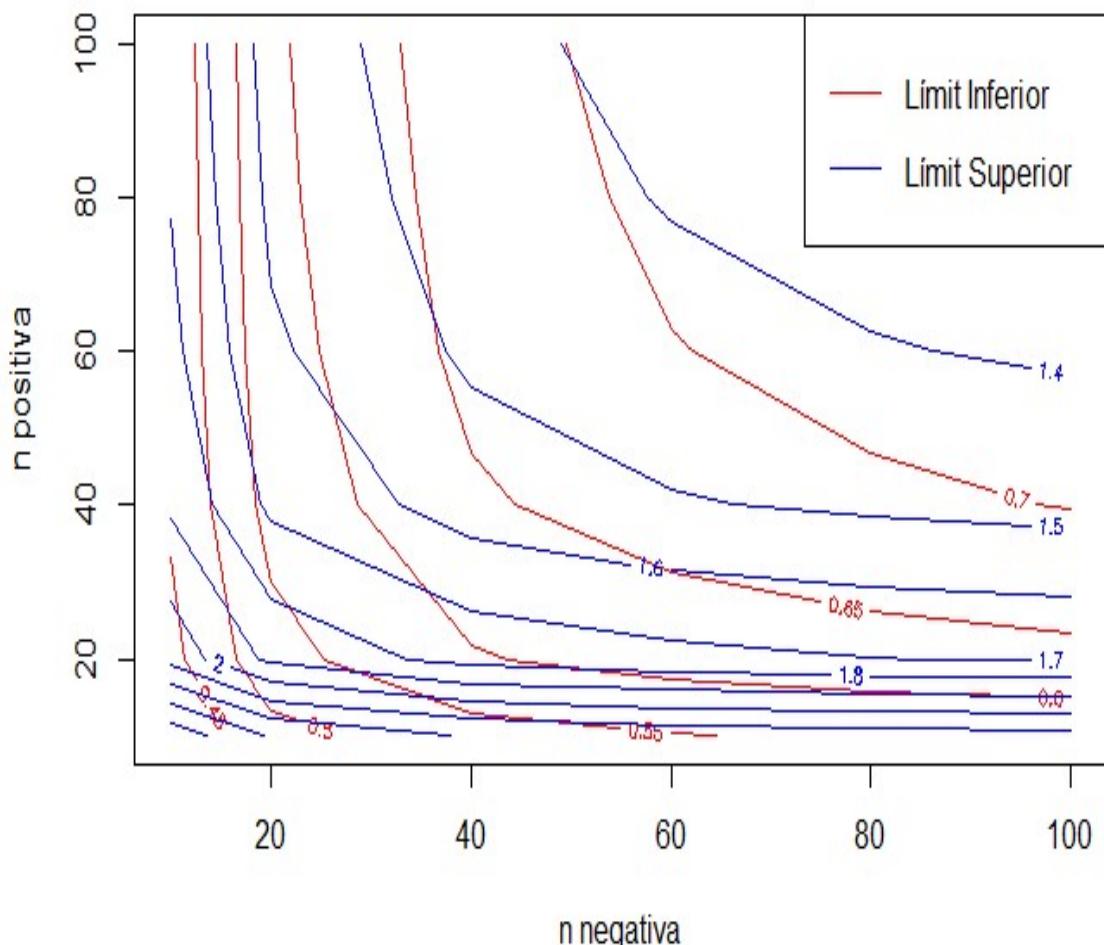


Figura 1: Valors dels límits dependent de les mides de les cues

## 4 Potència empírica del test - Mètode de Monte Carlo

El que hem fet per comprovar que el nostre estadístic de contrast funciona és generar valors positius i negatius a partir de dues distribucions t-student que podien ser o no de diferents graus de llibertat. S'han fet diverses proves canviant els paràmetres de les distribucions per veure que funcioni per a totes. De les distribucions generades, sabem quin ha de ser el resultat del nostre test així que hem repetit aquest test un gran nombre de vegades i hem calculat quin era el percentatge de vegades que s'equivocava. S'han creat mostres de 1000 valors per a cada distribució i s'ha fet el remostreig de les cues 1000 vegades.

L'algoritme creat per calcular els valors d' $\alpha$  consisteix en el següent:

1. Separem les dades en dos submostres. Una contindrà tots els valors negatius i l'altre tots els valors positius. La submostra amb els valors negatius la transformarem prenent els valors absoluts. Això ho fem perquè no es pot calcular  $\alpha$  amb una mostra de valors negatius.
2. D'aquestes submostres calculem  $\alpha$  i  $x_{min}$  amb el mètode que estiguem utilitzant (Clauset o CV). Amb els valors  $\alpha$  calculem l'estadístic que hem definit per comparar les cues. És important tindre en compte el valor del estadístic calculat en aquest pas ja que és el que ens donarà el p-valor al final del test.
3. Un cop tenim els valors  $\alpha$  inicials, mirem quin dels  $x_{min}$  és el que fa que tinguem menys valors a les cues i tornem a separar les submostres definint com a cua els valors que superin aquest  $x_{min}$  i com a cos els valors que no ho facin.
4. A continuació ajuntem les dues cues i les reordenem. Assignarem cada meitat de valors a una cua i les ajuntarem amb els cossos corresponents.
5. Amb la mostra rejuntada tornem a calcular els  $\alpha$  i l'estadístic i el guardem en un vector.
6. Repetim els passos 3, 4 i 5 1000 vegades tal i com diuen Davison i Hinkey [2].
7. Mirem quina és la distribució que segueix el vector dels estadístics i mirem on escau el estadístic inicial. El valor que prengui aquest estadístic inicial en la funció de distribució dels estadístics serà el p-valor del test.

## 4.1 Potència del test amb el mètode de Clauset

| gl  | 0.2  | 0.4  | 0.6  | 0.8  | 1    | 2    | 3    | 4    | 20   |
|-----|------|------|------|------|------|------|------|------|------|
| 0.2 | 0.08 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 0.4 |      | 0.06 | 0.02 | 0    | 0    | 0    | 0    | 0    | 0    |
| 0.6 |      |      | 0.04 | 0.18 | 0.02 | 0    | 0    | 0    | 0    |
| 0.8 |      |      |      | 0.12 | 0.36 | 0    | 0    | 0    | 0    |
| 1   |      |      |      |      | 0.04 | 0    | 0    | 0    | 0    |
| 2   |      |      |      |      |      | 0.08 | 0.52 | 0.22 | 0.1  |
| 3   |      |      |      |      |      |      | 0.08 | 0.74 | 0.02 |
| 4   |      |      |      |      |      |      |      | 0.02 | 0.48 |
| 20  |      |      |      |      |      |      |      |      | 0.08 |

**Taula 2:** Potència del test amb el mètode de Clauset i distribucions t-Student

En la diagonal de la taula 3 tenim l'error de tipus 1 que representa la probabilitat de que sent certa la hipòtesi nul·la, el test la rebutgi. En el nostre cas només es podrà calcular l'error quan les dues segueixin la mateixa distribució. Com s'observa a la taula, en tots els casos l'error de tipus 1 es troba al voltant del 5%

La resta de valors de la taula corresponen als errors de tipus 2 que representen la probabilitat de que cent falsa la hipòtesi nul·la el test la accepti. Observem que el test té problemes per diferenciar les dues de la t-student quan els graus de llibertat són grans, és a dir, si les dues dues de les distribucions són lleugeres però diferents, el test no funciona massa bé.

## 4.2 Potència del test amb el mètode CV

| gl  | 0.2  | 0.4  | 0.6  | 0.8  | 1    | 2    | 3    | 4    | 20   |
|-----|------|------|------|------|------|------|------|------|------|
| 0.2 | 0.06 | 0    | 0    | 0    | 0    | 0.32 | 0.16 | 0.08 | 0.08 |
| 0.4 |      | 0.04 | 0.64 | 0.3  | 0.1  | 0.02 | 0    | 0    | 0    |
| 0.6 |      |      | 0.06 | 0.8  | 0.52 | 0    | 0    | 0    | 0    |
| 0.8 |      |      |      | 0.02 | 0.8  | 0    | 0.36 | 0.26 | 0.24 |
| 1   |      |      |      |      | 0.02 | 0.16 | 0.52 | 0.44 | 0.34 |
| 2   |      |      |      |      |      | 1    | 0.08 | 0.02 | 0    |
| 3   |      |      |      |      |      |      | 0.92 | 0.22 | 0.02 |
| 4   |      |      |      |      |      |      |      | 0.78 | 0.06 |
| 20  |      |      |      |      |      |      |      |      | 0.94 |

**Taula 3:** Potència del test amb el mètode CV i distribucions t-Student

Igual que en la taula 3, en la diagonal de la taula 1 tenim l'error de tipus 1 però aquí tenim bastants problemes a mesura que van creixent els graus de llibertat de les distribucions. Pel que fa a la resta de valors veiem que també tenim problemes quan els graus de llibertat són grans. Aquests problemes apareixen inclòs en menys graus de llibertat que fent el test pel mètode de Clauset. Això ve donat a que no hem pogut calibrar de forma adequada el procés. És per aquesta raó que tots els tests que se'ls hi han fet a les carteres han estat fets amb el mètode de Clauset.

## 5 Aplicació del test a les dades d'un actiu financer

Un cop hem comprovat la potència d'aquest test l'aplicarem en dades de rendibilitats de diverses accions. Primer de tot farem una petita estadística descriptiva d'aquestes rendibilitats i després aplicarem el test. Les dades de les accions van des del 3 de Gener de 2011 fins al 31 de Desembre de 2016. A continuació, a la taula 4 presentem els estadístics principals i els valors d' $\alpha$  de les accions que hem escollit. Les accions que hem escollit són de Netflix, una empresa dedicada a l'entreteniment, SPS, una empresa dedicada al comerç informàtic i a McDonald's, una empresa dedicada al menjar. Les carteres que formem a partir d'aquestes accions estaran ben diversificades.

|                      | Netflix   | SPS      | McDonald's |
|----------------------|-----------|----------|------------|
| $\bar{X}$            | 0.00105   | -0.001   | 0.0004339  |
| Sd                   | 0.036     | 0.0235   | 0.00954    |
| Median               | -0.000192 | -0.00065 | 0.000789   |
| min                  | -0.4292   | -0.1483  | -0.0457    |
| max                  | 0.3522    | 0.2262   | 0.078      |
| $\alpha_+$ (Clauset) | 3.2606    | 4.26     | 3.761      |
| $\alpha_-$ (Clauset) | 3.2202    | 4.322    | 3.593      |

Taula 4: Estadístics de les accions

### 5.1 Netflix

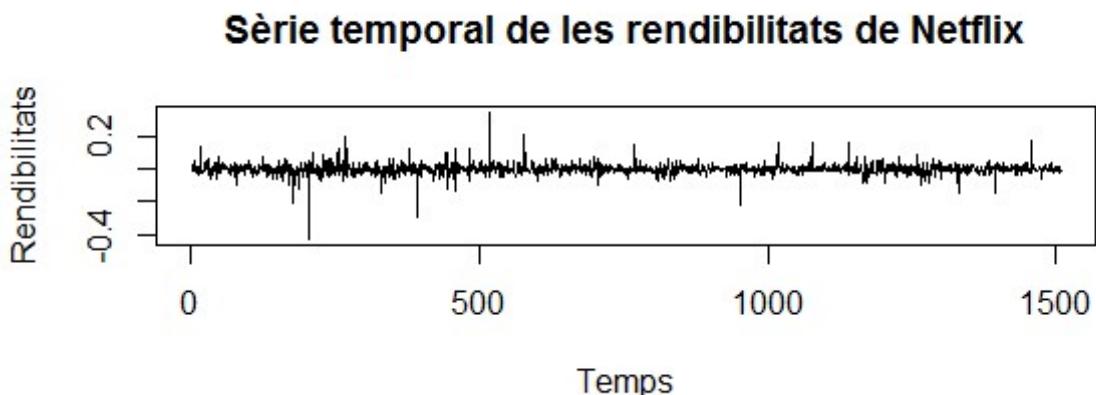
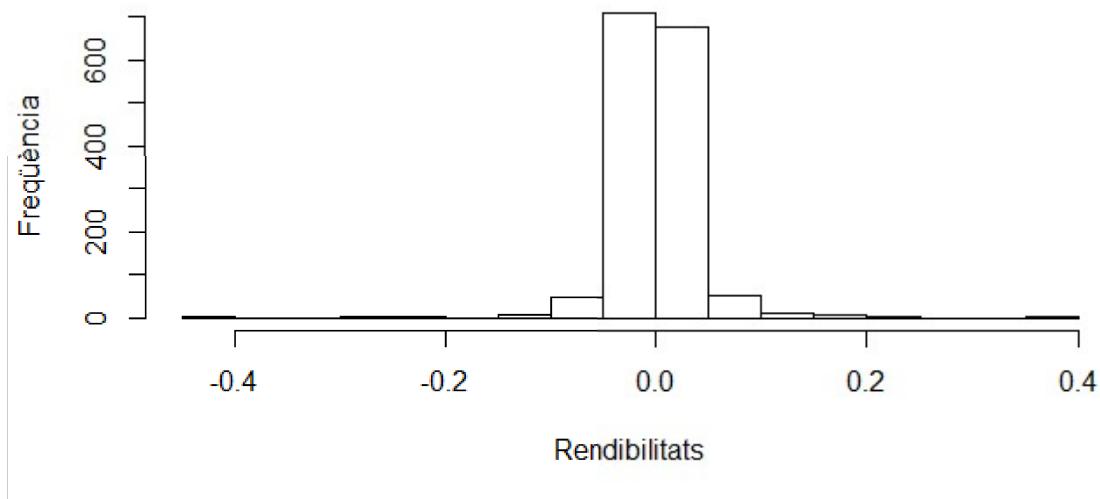


Figura 2: Sèrie temporal de les rendibilitats de les accions de Netflix

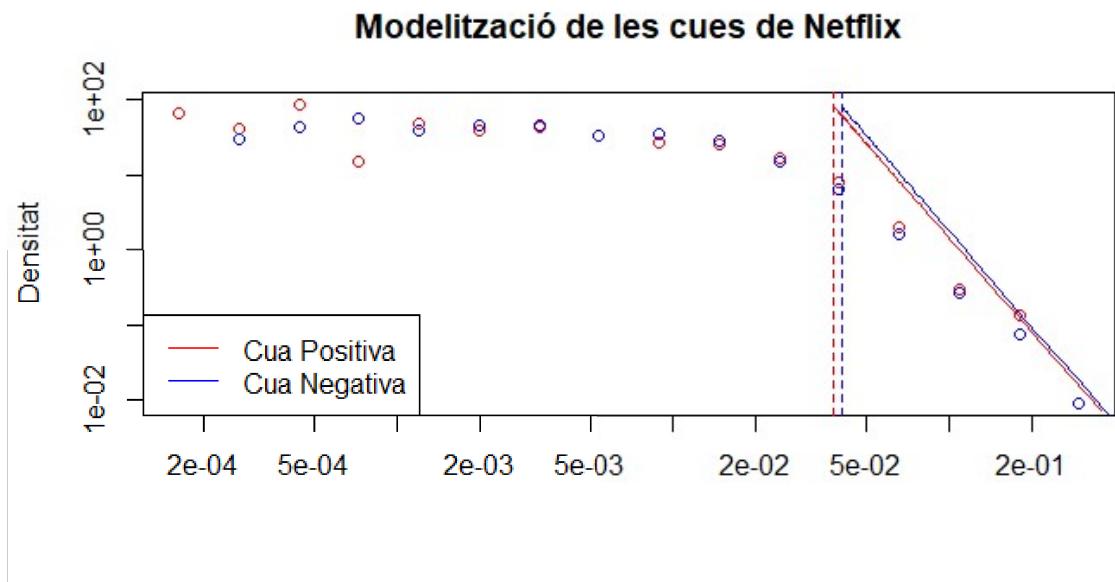
Com podem observar en el gràfic 2 hi ha dues rendibilitats que són bastant més elevades que la resta. Aquestes corresponen als dies 15 d'Octubre de 2011 i 24 de Gener del 2013. Aquestes rendibilitats no corresponen a cap split de les accions i s'han revisat les dues dates i no s'ha trobat cap fet especial que fes que els preus variessin en excés.

### Histograma de les rendibilitats de Netflix



**Figura 3:** Histograma de les rendibilitats de les accions de Netflix

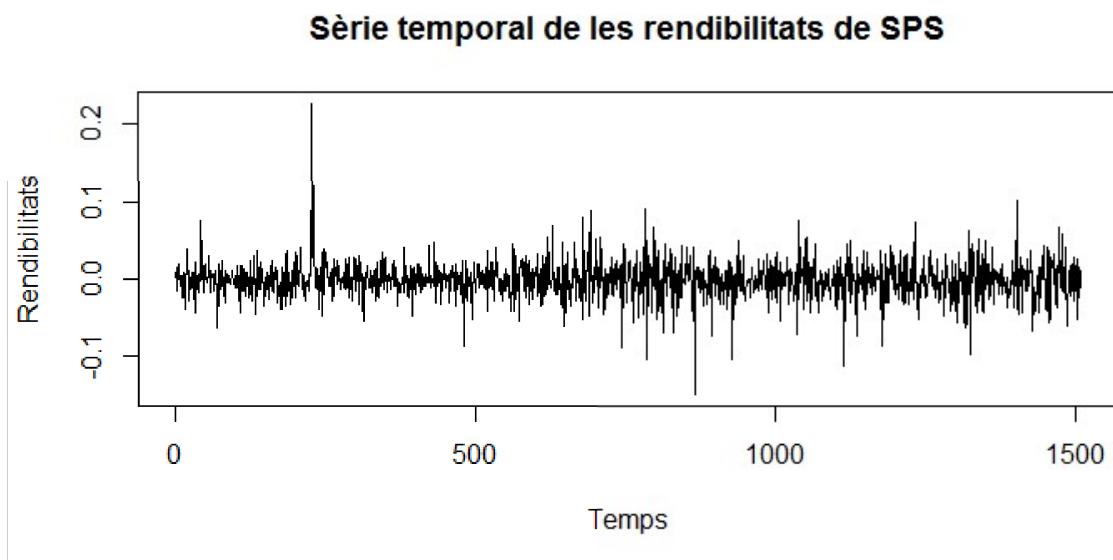
En el gràfic 3, veiem representades les rendibilitats de les accions de Netflix en forma d'histograma. Veiem que exceptuant els dos valors anòmals que hem detectat abans, la resta de valors es troben repartits de forma simètrica al voltant del 0 tal i com cap esperar de les rendibilitats. A continuació aplicarem el test que hem creat per veure si podem considerar que les cues són iguals.



**Figura 4:** Gràfic de les cues de les rendibilitats de les accions de Netflix

Observant el gràfic 4 veiem que pràcticament les cues de la distribució són iguals. Si apliquem el test en aquestes dades, reafirmem el que podíem observar en el gràfic, el p-valor és de 0.68. Això vol dir que no tenim prous evidències estadístiques per rebutjar la hipòtesi de que les cues siguin diferents.

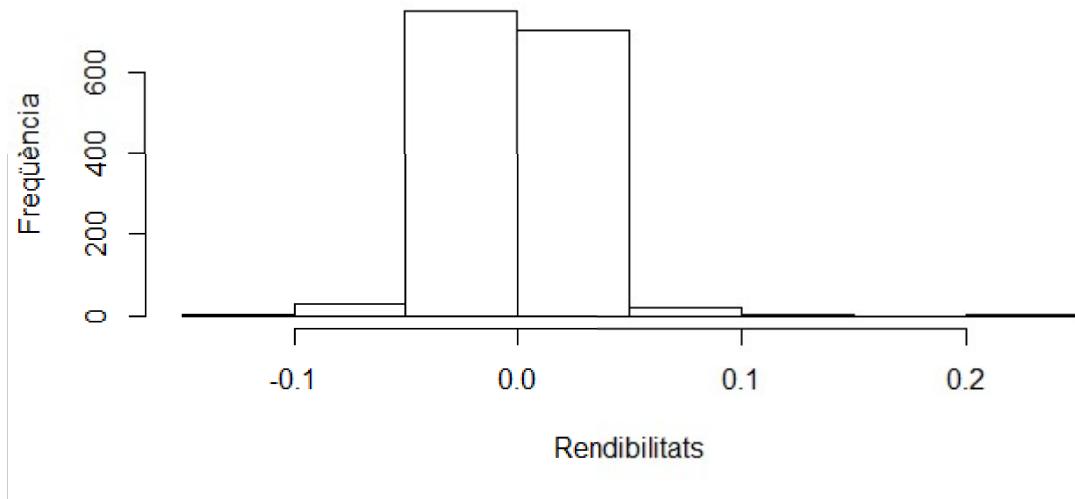
## 5.2 SPS



**Figura 5:** Sèrie temporal de les rendibilitats de les accions de SPS

Com podem observar en el gràfic 5 hi ha una rendibilitat que és una mica més elevada que la resta. Aquesta correspon al dia 29 de Novembre de 2011. Aquestes rendibilitats no corresponen a cap split de les accions i s'ha revisat la data i no s'ha trobat cap noticia que pogués fer que la rendibilitat fos més alta.

### Histograma de les rendibilitats de SPS



**Figura 6:** Histograma de les rendibilitats de les accions de SPS

En el gràfic 6, veiem representades les rendibilitats de les accions de Netflix en forma d'histograma. Veiem que pel cantó positiu tenim un valor més extrem que pel valor negatiu però es troben, propers al 0. A continuació aplicarem el test que hem creat per veure si podem considerar que les cues són iguals.

## Modelització de les cues de SPS

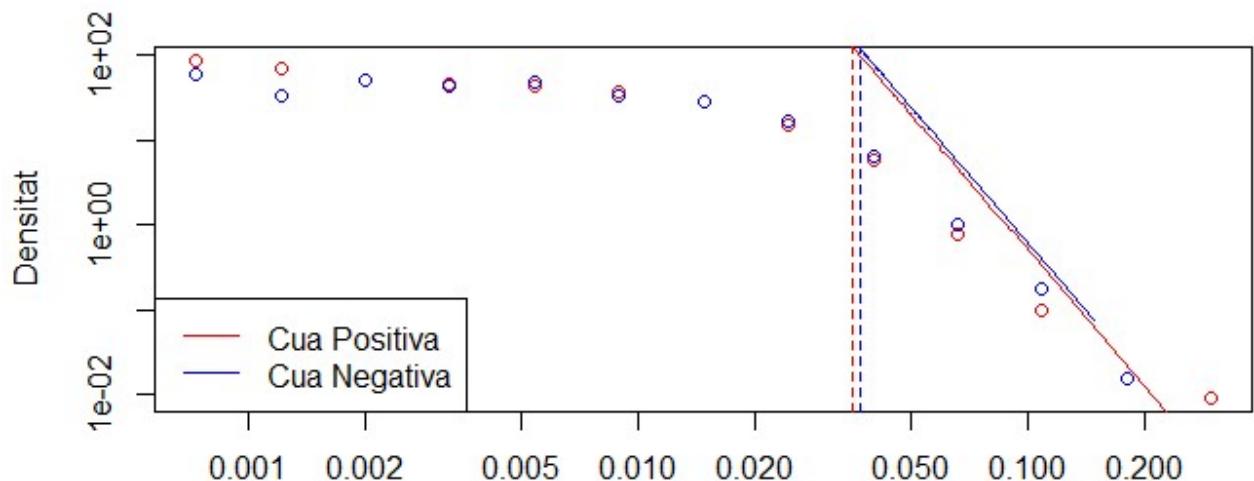
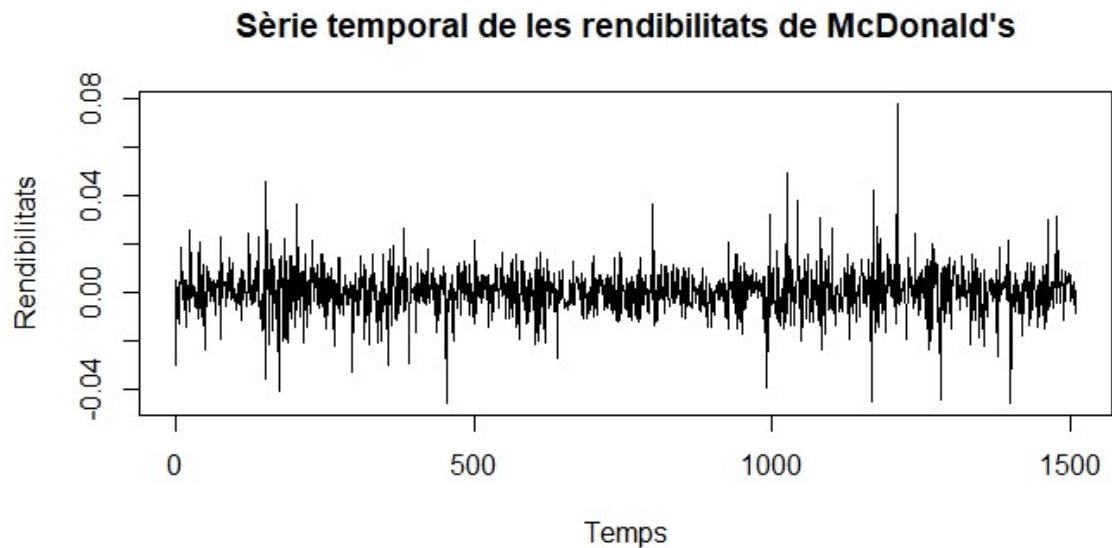


Figura 7: Gràfic de les cues de les rendibilitats de les accions de SPS

Observant el gràfic 7 veiem que la cua positiva es troba una mica per sota de la negativa i és possible que siguin diferents. Si apliquem el test en aquestes dades, tot i les diferències que poguéssim observar en el gràfic, el p-valor és de 0.34. Això vol dir que no tenim prous evidències estadístiques per rebutjar la hipòtesi de que les cues siguin diferents.

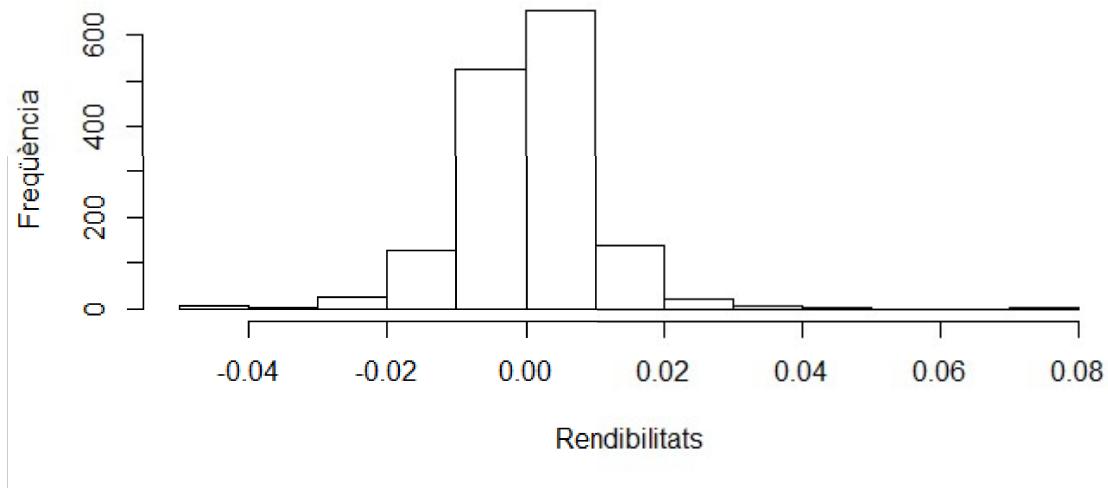
### 5.3 McDonald's



**Figura 8:** Sèrie temporal de les rendibilitats de les accions de McDonald's

Com podem observar en el gràfic 8 les rendibilitats no són massa diferents les unes de les altres ja que el valor de la rendibilitat més alta és 0.08. Tot i que les diferències són petites, si que és cert que hi ha valors més elevats per a les rendibilitats positives que per les negatives.

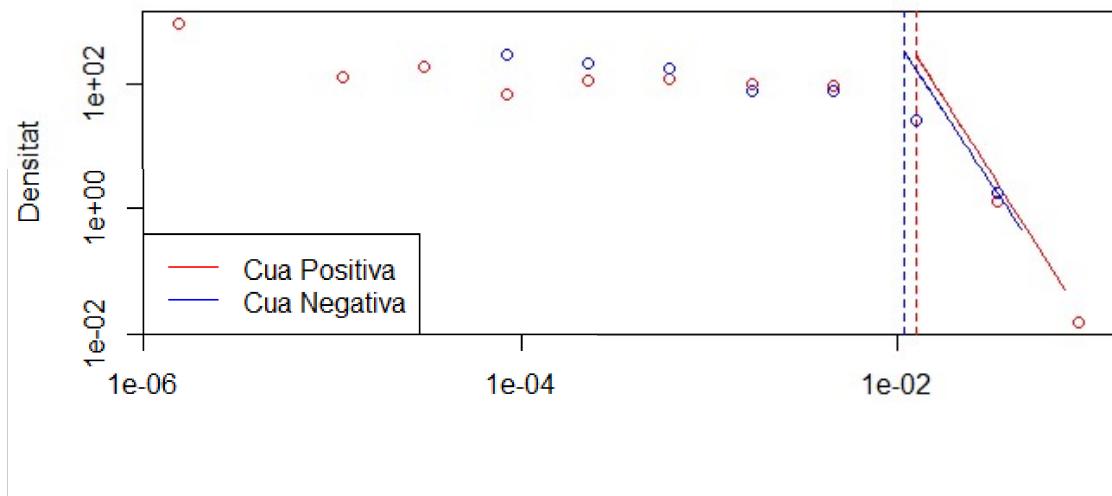
### Histograma de les rendibilitats de McDonald's



**Figura 9:** Histograma de les rendibilitats de les accions de McDonald's

En el gràfic 9, veiem representades les rendibilitats de les accions de McDonald's en forma d'histograma. Tal i com hem comentat, les rendibilitats positives semblaven més grans que les negatives i cap la possibilitat de que el nostre test ens doni un p-valor significatiu. A continuació aplicarem el test que hem creat per veure si podem considerar que les dues són iguals.

## Modelització de les cues de McDonald's



**Figura 10:** Gràfic de les cues de les rendibilitats de les accions de McDonald's

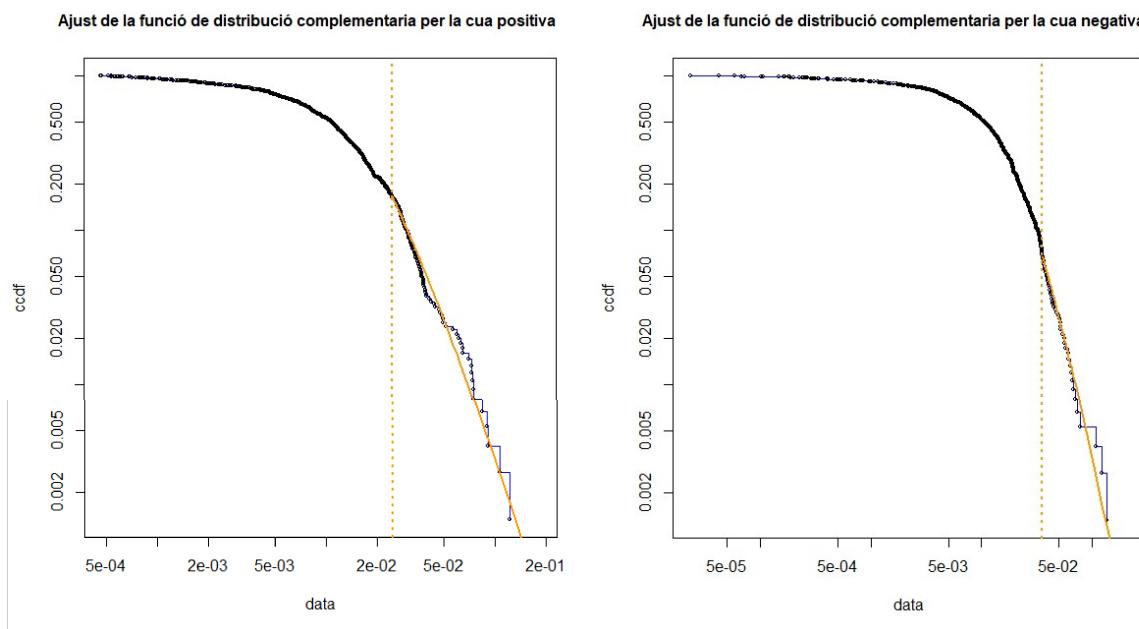
Observant el gràfic 10 veiem que les cues de la distribució semblen diferents. Tot i així, si apliquem el test en aquestes dades, veiem que el p-valor és de 0.24. Això vol dir que no tenim prou evidències estadístiques per rebutjar la hipòtesi de que les cues siguin diferents.

## 6 Aplicació del test a una cartera d'actius financers

En aquest apartat utilitzarem les dades de les rendibilitats de l'apartat 5. En aquest apartat veurem com varia el p-valor del nostre test en dos situacions. En el primer dels casos veurem com varia en una cartera formada per dues accions. En el segon cas veurem com varia en una cartera formada per tres accions.

### 6.1 Cartera formada per dos actius financers

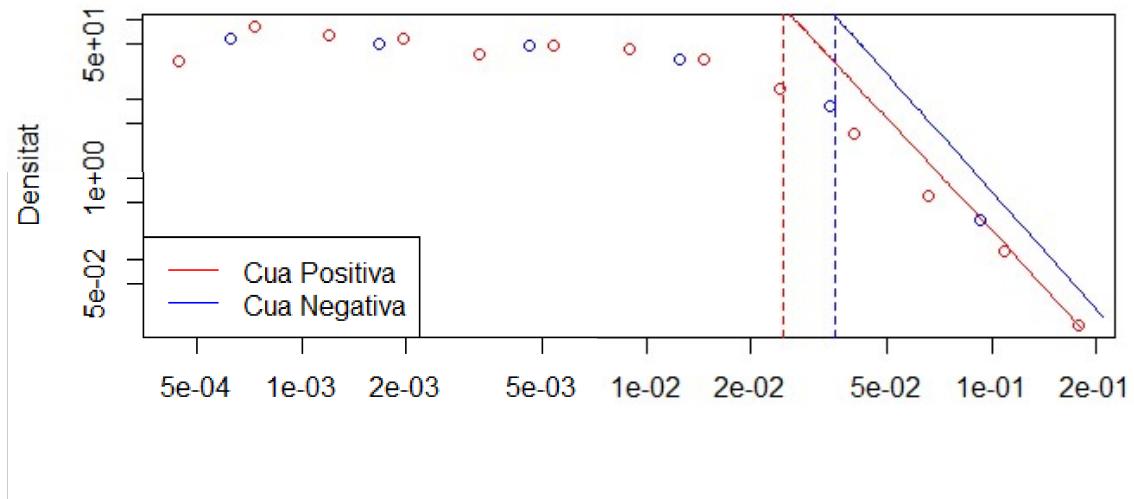
La cartera d'aquest apartat està formada per accions de Netflix i per accions de SPS. Primer de tot mode-litzarem les cues de la cartera en el que el 50% dels diners invertits estan invertits en Netflix i l'altre 50% en accions de SPS.



**Figura 11:** Gràfic de les distribucions complementaries de les cues

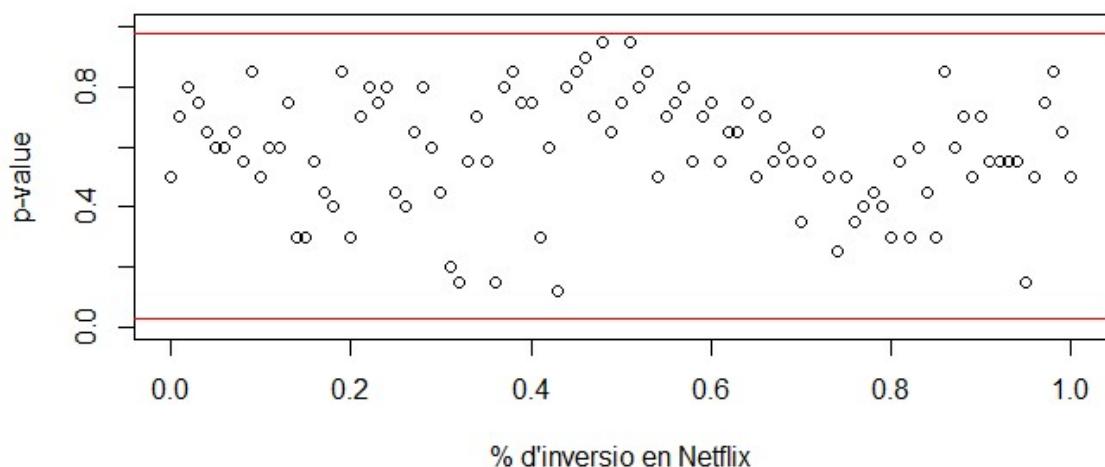
Com podem observar en el gràfic 11 la cua positiva té molts més valors que la negativa i en aquest gràfic sembla que la cua positiva decreix més lentament que la cua negativa. Per comparar millor les cues tindrem en compte el gràfic següent.

## Modelització de les cues de la cartera



**Figura 12:** Modelització de les cues de la cartera

Tot i que el valor de  $x_{min}$  de la cua positiva és molt més petit que el de la cua negativa, les línies que representen els valors d' $\alpha$  van pràcticament paral·leles. Això indica que els valors d' $\alpha$  són molt semblants. Els valors  $x_{min}$  dels gràfics 11 i 12 són diferents degut a que els mètodes per trobar aquest valor han estat diferents. Passem a veure com seran els quantils de totes les carteres possibles.



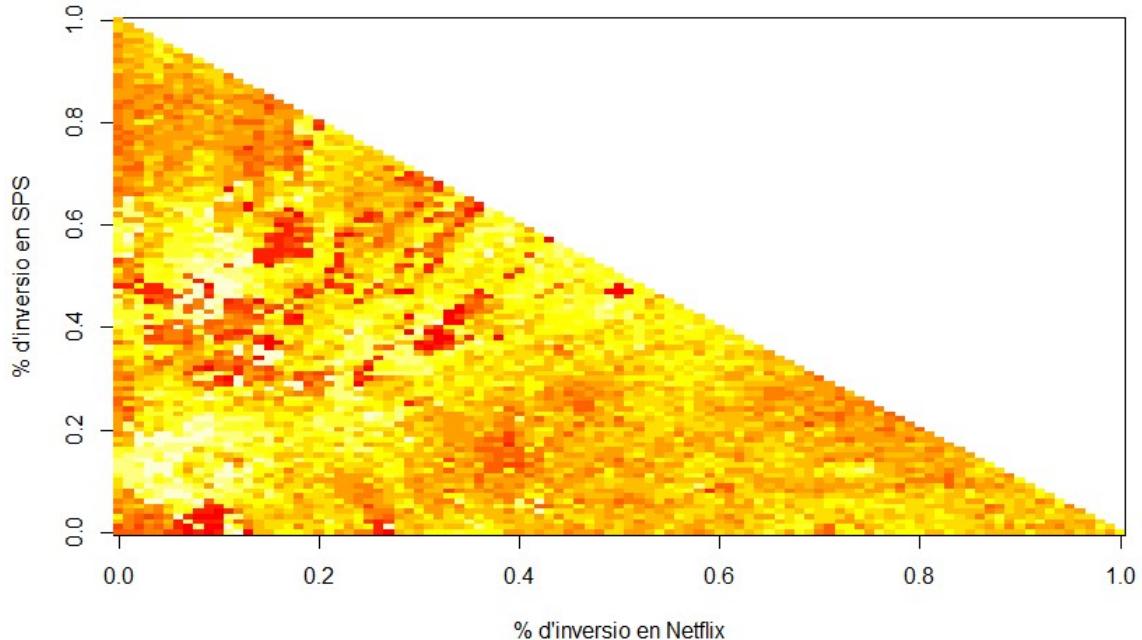
**Figura 13:** Gràfic dels p-valors cartera amb 2 accions

En el gràfic 13 veiem que tots els quantils es situen dins de la regió d'acceptació per tant combinant aquestes dues accions no hi ha cap manera de que poguem considerar que les cues siguin diferents. Tot i així veiem que els quantils quan el % d'inversió de cada una de les accions es troba al voltant del 50% els p-valors són més alts.

El que ens interessa de les carteres de dues accions és tindre les carteres que apareixen a la part d'abaix del gràfic ja que són les que ens poden reportar guanys inesperats més grans que les pèrdues. En el nostre cas, no hi ha cap combinació amb la que poguem considerar que les cues siguin diferents. Si haguéssim d'invertir en una cartera, el millor seria una que tingui entre un 70% i un 85% dels diners a Netflix, ja que és una zona en la que hi ha un grup de p-valors mitjanament baixos.

## 6.2 Cartera formada per 3 actius financers

La cartera d'aquest apartat està formada per accions de Netflix, SPS i McDonald's.



**Figura 14:** Gràfic dels p-valors cartera amb 3 accions

Per veure els quantils de les carteres formades per 3 accions hem de mirar el gràfic 14. En aquells punts del gràfic on la casella és de color blanc tenim que el quantil està entre 0 i 0.1. Aquests punts són possibilitats en que la cua negativa de la distribució de les rendibilitats de la cartera és menys pesant que la cua positiva. Aquests punts ens reportaran beneficis inesperats més grans que unes pèrdues inesperades. En canvi, per les caselles de color vermell, tenim quantils més elevats on la cua negativa seria més pesant que la cua positiva.

A nosaltres com a inversors el que ens interessa és situar-nos en les zones blanques. Cal especificar que és millor que sigui una zona on hi hagin bastants punts blancs ja que així no cal anar actualitzant la cartera cada dia. Si la nostre cartera fos creada a partir de les accions de Netflix, SPS i McDonald's, una bona inversió seria aquella en la que invertim entre un 0% i un 10% dels diners a Netflix, entre un 10% i un 20% dels diners a SPS i la resta a accions de McDonald's. Cal destacar que en aquest test suposem que el paràmetre escala de les cues és el mateix.

## 7 Conclusions

En aquest treball hem vist dos mètodes diferents per a modelar distribucions de cues i hem intentat crear un test que ens permetés comparar aquestes cues per veure quina era més lleugera. Al estudiar la potència del test hem observat que el mètode de Clauset funcionava millor per al test que volíem desenvolupar degut a que el mètode CV no està fet per a trobar el valor mínim de la cua i això afecta al test.

Hem aplicat el test creat a partir del mètode de Clauset a accions de Netflix, SPS i McDonald's per veure com funcionava i puguer fer-nos una idea de com funcionaria quan ho apliquéssim a una cartera d'accions. Hem aplicat el test a Carteres de dues accions i hem vist que amb les accions que havíem triat per fer aquestes Carteres tots els casos es trobaven no podíem rebutjar la hipòtesi nul·la.

Per al cas de la cartera de tres accions si que veiem que al aplicar el test hi havia algunes regions on no podíem considerar que les cues fossin igual de pesants. Hem determinat també en quines zones ens interessa invertir quan volem que els guanys esperats siguin més grans que les pèrdues inesperades i quines inversions cal evitar en aquesta cartera per tal d'evitar l'efecte contrari.

## 8 Referències

- [1] SINGH, K. XIE, M. *Bootstrap: A Statistical Method*
- [2] DAVISON, A.C. HINKLEY, D.V.(1997) *Bootstrap Methods and Their Application*
- [3] COLES, S.(2001) *An Introduction to Statistical Modeling of Extreme Values*
- [4] CLAUSET, A. (2011) *Inference, Models and Simulation for Complex Systems*
- [5] Clauset, A., Shazili, R. M. E. J. Newman (2009) *Power-Law Distributions in Empirical Data*
- [6] CABANÀ, A. *Tests de Permutaciones*
- [7] SERRA, I. (2013) *Statistical models for tails and applications*
- [8] CASTILLO, J. SERRA, I. (2014) *Likelihood inference for Generalized Pareto Distribution*
- [9] PADILLA, M. CASTILLO, J. (2015) *Modelling extreme values by the residual coefficient of variation*

## A Annex

### A.1 Desenvolupament del test teòric exacte

```
ns<-c(10,20,40,60,80,100)
k<-length(ns)
matlo<-matrix(rep(0,k^2),ncol=k)
for (i in 1:k)
{
  for (j in 1:k)
  {
    matlo[i,j]<-(qbeta(0.025,ns[i],ns[j])/(1-qbeta(0.025,ns[i],ns[j])))*(ns[j]/ns[i])
  }
}
image(matlo)
image(ns,ns,log(matlo))
contour(ns,ns,(matlo))

matup<-matrix(rep(0,k^2),ncol=k)
for (i in 1:k)
{
  for (j in 1:k)
  {
    matup[i,j]<-(qbeta(0.975,ns[i],ns[j])/(1-qbeta(0.975,ns[i],ns[j])))*(ns[j]/ns[i])
  }
}
contour(ns,ns,(matlo),col="blue")
contour(ns,ns,(matup),add=T,col="red")

contour(ns,ns,matlo,xlab="n negativa",ylab="n positiva",col="red")
contour(ns,ns,matup,col="blue",add=T)
legend("topright",c("Límit Inferior","Límit Superior"),col=c("Red","Blue"),lty=c(1,1))
abline(0,1)
#obs: si les mides són iguals, l'interval simètric en escala log

plot(ns,(matlo[6,]),ylim=c(0,3))
points(ns,(matup[6,]))
points(ns,(matlo[1,]))
points(ns,(matup[1,]))
```

### A.2 Plfit

```
#####
#
# library
#
library(VGAM) # zeta function
library(R.matlab) # read matlab file just for test purpose
#
```

```

#####
#
# test zone
#
#####
runtest <- function(){
  #read a matlab file
  #x generated by x = randht(10000, 'xmin', 15, 'powerlaw', 2.5) with matlab (continuous law)
  x<-readMat(file("x_2.5_15_10000.mat","rb"))$x[,1]
  #plfit matlab:alpha=2.4975,xmin=18.5752,D=0.0062
  plfit(x)
  #plfit R:alpha=2.497485,xmin=18.57524,D=0.006174977
  #x0 generated by x0 =floor(x) with matlab (big discrete approximation)
  x0<-readMat(file("x0_2.5_15_10000.mat","rb"))$x[,1]
  #plfit matlab:alpha=2.4700,xmin=15,D=0.0056
  plfit(x0)
  #plfit r:alpha=2.47,xmin=15,D=0.00564463

}

#####
#
# PLFIT fits a power-law distributional model to data.
#
# PLFIT(x) estimates x_min and alpha according to the goodness-of-fit
# based method described in Clauset, Shalizi, Newman (2007). x is a
# vector of observations of some quantity to which we wish to fit the
# power-law distribution  $p(x) \sim x^{-\alpha}$  for  $x \geq x_{\min}$ .
# PLFIT automatically detects whether x is composed of real or integer
# values, and applies the appropriate method. For discrete data, if
#  $\min(x) > 1000$ , PLFIT uses the continuous approximation, which is
# a reliable in this regime.
#
# The fitting procedure works as follows:
# 1) For each possible choice of x_min, we estimate alpha via the
#    method of maximum likelihood, and calculate the Kolmogorov-Smirnov
#    goodness-of-fit statistic D.
# 2) We then select as our estimate of x_min, the value that gives the
#    minimum value D over all values of x_min.
#
# Note that this procedure gives no estimate of the uncertainty of the
# fitted parameters, nor of the validity of the fit.
#
# Example:
#   x <- (1-runif(10000))^{(-1/(2.5-1))}
#   plfit(x)
#
# Version 1.0  (2008 February)
# Version 1.1  (2008 February)
#   - correction : division by zero if limit >= max(x) because the unique R function do no sort

```

```

#           and the matlab function do...
# Version 1.1 (minor correction 2009 August)
#   - correction : lines 230 zdiff calcul was wrong when xmin=0 (thanks to Naoki Masuda)
#   - gpl version updated to v3.0 (asked by Felipe Ortega)
# Version 1.2 (2011 August)
#   - correction for method "limit" thanks to David R. Pugh
#     xmins <- xmins[xmins<=limit] is now xmins <- xmins[xmins>=limit]
#   - "fixed" method added for xmins from David R. Pugh
#   - modifications by Alan Di Vittorio:
#     - correction : zdiff calculation was wrong when xmin==1
#     - the previous zdiff correction was incorrect
#     - correction : x has to have at least two unique values
#     - additional discrete x input test : discrete x cannot contain the value 0
#     - added option to truncate continuous xmin search when # of obs gets small
#
# Copyright (C) 2008,2011 Laurent Dubroca laurent.dubroca_at_gmail.com
# (Stazione Zoologica Anton Dohrn, Napoli, Italy)
# Distributed under GPL 3.0
# http://www.gnu.org/copyleft/gpl.html
# PLFIT comes with ABSOLUTELY NO WARRANTY
# Matlab to R translation based on the original code of Aaron Clauset (Santa Fe Institute)
# Source: http://www.santafe.edu/~aaronc/powerlaws/
#
# Notes:
#
# 1. In order to implement the integer-based methods in Matlab, the numeric
# maximization of the log-likelihood function was used. This requires
# that we specify the range of scaling parameters considered. We set
# this range to be seq(1.5,3.5,0.01) by default. This vector can be
# set by the user like so,
#
#   a <- plfit(x,"range",seq(1.001,5,0.001))
#
# 2. PLFIT can be told to limit the range of values considered as estimates
# for xmin in two ways. First, it can be instructed to sample these
# possible values like so,
#
#   a <- plfit(x,"sample",100)
#
# which uses 100 uniformly distributed values on the sorted list of
# unique values in the data set. Alternatively, it can simply omit all
# candidates below a hard limit, like so
#
#   a <- plfit(x,"limit",3.4)
#
# In the case of discrete data, it rounds the limit to the nearest
# integer.
#
# Finally, if you wish to force the threshold parameter to take a specific value
# (useful for bootstrapping), simply call plfit() like so
#

```

```

#      a <- plfit(x, "fixed", 3.5)
#
# 3. When the input sample size is small (e.g., < 100), the estimator is
# known to be slightly biased (toward larger values of alpha). To
# explicitly use an experimental finite-size correction, call PLFIT like
# so
#
#      a <- plfit(x, finite=TRUE)
#
# 4. For continuous data, PLFIT can return erroneously large estimates of
# alpha when xmin is so large that the number of obs x >= xmin is very
# small. To prevent this, we can truncate the search over xmin values
# before the finite-size bias becomes significant by calling PLFIT as
#
#      a = plfit(x, nosmall=TRUE);
#
# which skips values xmin with finite size bias > 0.1.
#
#####
plfit<-function(x=rpareto(1000,10,2.5),method="limit",value=c(),finite=FALSE,nowarn=FALSE,nosmall=FALSE)
#init method value to NULL
vec <- c() ; sampl <- c() ; limit <- c(); fixed <- c()
#####
#
# test and trap for bad input
#
switch(method,
       range = vec <- value,
       sample = sampl <- value,
       limit = limit <- value,
       fixed = fixed <- value,
       argok <- 0)

if(exists("argok")){stop("(plfit) Unrecognized method")}

if( !is.null(vec) && (!is.vector(vec) || min(vec)<=1 || length(vec)<=1) ){
  print(paste("(plfit) Error: 'range' argument must contain a vector > 1; using default."))
  vec <- c()
}
if( !is.null(sampl) && ( !(sampl==floor(sampl)) || length(sampl)>1 || sampl<2 ) ){
  print(paste("(plfit) Error: 'sample' argument must be a positive integer > 2; using default."))
  sample <- c()
}
if( !is.null(limit) && (length(limit)>1 || limit<1) ){
  print(paste("(plfit) Error: 'limit' argument must be a positive >=1; using default."))
  limit <- c()
}
if( !is.null(fixed) && (length(fixed)>1 || fixed<=0) ){
  print(paste("(plfit) Error: 'fixed' argument must be a positive >0; using default."))
  fixed <- c()
}

```

```

# select method (discrete or continuous) for fitting and test if x is a vector
fdattype<- "unknow"
if( is.vector(x,"numeric") ){ fdattype<- "real" }
if( all(x==floor(x)) && is.vector(x) ){ fdattype<- "integer" }
if( all(x==floor(x)) && min(x) > 1000 && length(x) > 100 ){ fdattype <- "real" }
if( fdattype=="unknow" ){ stop("(plfit) Error: x must contain only reals or only integers.") }

#
# end test and trap for bad input
#
#####
#####
#####
#####

#
# estimate xmin and alpha in the continuous case
#
if( fdattype=="real" ){

  xmins <- sort(unique(x))
  xmins <- xmins[-length(xmins)]

  if( !is.null(limit) ){
    xmins <- xmins[xmins>=limit]
  }
  if( !is.null(fixed) ){
    xmins <- fixed
  }
  if( !is.null(sampl) ){
    xmins <- xmins[unique(round(seq(1,length(xmins),length.out=sampl)))]
  }

  dat <- rep(0,length(xmins))
  z   <- sort(x)

  for( xm in 1:length(xmins) ){
    xmin <- xmins[xm]
    z   <- z[z>=xmin]
    n   <- length(z)
    # estimate alpha using direct MLE
    a   <- n/sum(log(z/xmin))
    # truncate search if nosmall is selected
    if( nosmall ){
      if((a-1)/sqrt(n) > 0.1){
        dat <- dat[1:(xm-1)]
        print(paste("(plfit) Warning : xmin search truncated beyond",xmins[xm-1]))
        break
      }
    }
    # compute KS statistic
    cx   <- c(0:(n-1))/n
    cf   <- 1-(xmin/z)^a
}
}

```

```

    dat[xm] <- max(abs(cf-cx))
}

D      <- min(dat)
xmin  <- xmins[min(which(dat<=D))]
z     <- x[x>=xmin]
n     <- length(z)
alpha <- 1 + n/sum(log(z/xmin))

if( finite ){
  alpha <- alpha*(n-1)/n+1/n # finite-size correction
}
if( n<50 && !finite && !nowarn){
  print("(plfit) Warning : finite-size bias may be present")
}

#
#  end continuous case
#
#####
#####
##### estimate xmin and alpha in the discrete case
#
if( fdattype=="integer" ){

  if( is.null(vec) ){ vec<-seq(1.5,3.5,.01) } # covers range of most practical scaling parameters
  zvec <- zeta(vec)

  xmins <- sort(unique(x))
  xmins <- xmins[-length(xmins)]

  if( !is.null(limit) ){
    limit <- round(limit)
    xmins <- xmins[xmins>=limit]
  }

  if( !is.null(fixed) ){
    xmins <- fixed
  }

  if( !is.null(sampl) ){
    xmins <- xmins[unique(round(seq(1,length(xmins),length.out=sampl)))]
  }

  if( is.null(xmins) || length(xmins) < 2){
    stop("(plfit) error: x must contain at least two unique values.")
  }
}

```

```

if(length(which(xmins==0) > 0)){
  stop("(plfit) error: x must not contain the value 0.")
}

xmax <- max(x)
dat <- matrix(0,nrow=length(xmins),ncol=2)
z <- x
for( xm in 1:length(xmins) ){
  xmin <- xmins[xm]
  z    <- z[z>=xmin]
  n    <- length(z)
  # estimate alpha via direct maximization of likelihood function
  # vectorized version of numerical calculation
  # matlab: zdiff = sum( repmat((1:xmin-1)',1,length(vec)).^-repmat(vec,xmin-1,1) ,1);
  if(xmin==1){
    zdiff <- rep(0,length(vec))
  }else{
    zdiff <- apply(rep(t(1:(xmin-1)),length(vec))^-t(kronecker(t(array(1,xmin-1)),vec)),2,sum)
  }
  # matlab: L = -vec.*sum(log(z)) - n.*log(zvec - zdiff);
  L <- -vec*sum(log(z)) - n*log(zvec - zdiff);
  I <- which.max(L)
  # compute KS statistic
  fit <- cumsum(((xmin:xmax)^-vec[I])) / (zvec[I] - sum((1:(xmin-1))^(-vec[I])))
  cdi <- cumsum(hist(z,c(min(z)-1,(xmin+.5):xmax,max(z)+1),plot=FALSE)$counts/n)
  dat[xm,] <- c(max(abs(fit - cdi)),vec[I])
}
D      <- min(dat[,1])
I      <- which.min(dat[,1])
xmin  <- xmins[I]
n      <- sum(x>=xmin)
alpha <- dat[I,2]

if( finite ){
  alpha <- alpha*(n-1)/n+1/n # finite-size correction
}
if( n<50 && !finite && !nowarn){
  print("(plfit) Warning : finite-size bias may be present")
}

#
# end discrete case
#
#####
#
# return xmin, alpha and D in a list
return(list(xmin=xmin,alpha=alpha,D=D))
}

```

### A.3 Simulació dels resultats (Mètode de Clauset)

```

ps<-c()
for (contt in 1:50)
{
  x1<-alfanegc1
  x2<-alfaposc1

  alpha1<-plfit(alfanegc1)$alpha
  alpha2<-plfit(alfaposc1)$alpha

  est0<-alpha1/alpha2
  est0p<-alpha1/(alpha2+alpha1)
  est0
  est0p

  min1<-plfit(x1)$xmin
  min2<-plfit(x2)$xmin

  tail1<-x1[x1>min1]
  tail2<-x2[x2>min2]

  l1<-length(tail1)
  l2<-length(tail2)

  l<-min(l1,l2)

  tail1<-rev(sort(x1))[1:l]
  tail2<-rev(sort(x2))[1:l]

  min1<-min(tail1)
  min2<-min(tail2)

  body1<-x1[x1<=min1]
  body2<-x2[x2<=min2]

  tails<-c(tail1/min1,tail2/min2)

  ests<-c()
  estsp<-c()
  for (contt in 1:50)
  {
    t<-sample(2*l,1)
    tail1<-tails[t]*min1
    tail2<-tails[-t]*min2
    sx1<-c(body1,tail1)
    sx2<-c(body2,tail2)
    alpha1<-plfit(sx1)$alpha
    alpha2<-plfit(sx2)$alpha
  }
}

```

```

estsp<-c(estsp,alpha1/(alpha2+alpha1))
}
z<-ecdf(eststs)
zp<-ecdf(estsp)
ps<-rbind(ps,c(z(1),z(est0),zp(0.5),zp(est0p)))
}
(sum(ps[,2]<0.025)+sum(ps[,2]>0.975))/50

```

#### A.4 Llibreria ercv

```

#####
#####
#####
#####
#####
#
#          cievi
#
#####
#####
#####
#####
#####
#####

cievi<-function(nextremes,evi=0,conf.level=0.90,m=10,nsim=100)
{
  if (evi>=1/2) stop("The asymptotic confidence interval needs evi<1/2")
  evis<-numeric(nsim)
  for(sim in 1:nsim)
  {
    if (evi==0) dt<-rexp(nextremes) else dt<-(1/evi)*((1-runif(nextremes))^{(-evi)-1})
    evis[sim]<-as.numeric(Tm(data=dt,m=m,nsim=0)[ "evi"])
  }
  quantile(evis,c((1-conf.level)/2,1-(1-conf.level)/2))
}

#####
#####
#####
#####
#####
#
#          cvplot
#
#####
#####
#####
#####
#####
#####

cvplot<-function(data, threshold = NA, nextremes = NA, omit=4, evi=0, main="CVplot", conf.level=0.90, ...
{
  ###### controls
  omit<-floor(omit)
  if (omit<2) warning("The parameter omit have to be bigger than 1",call.=TRUE)
  omit<-max(omit,2)
  data <- as.numeric(data)
  data<-data[!is.na(data)]
  data <- sort(data)
  if (is.na(nextremes) && is.na(threshold)) threshold<-min(data)
  if (!is.na(nextremes) && !is.na(threshold)) stop("Enter EITHER a threshold or the number of upper extremes")
  if (!is.na(nextremes)) threshold<-rev(data)[as.numeric(nextremes)]
}

```

```

#####
##### initialization of vars
#####
n0<-length(data)
data<- data[data > threshold]- threshold
data <- sort(data)
n <- length(data)
k0<-n0-n
ks<-1:(n-omit)
ifelse(n-omit>50,ksr<-1:(n-max(omit,20)),ksr<-ks)
evi<-evi[!is.na(evi)]
if (sum(evi[evi>=1/4])!=0) warning("The asymptotic confidence interval needs evi<1/4",call. = TRUE)
evi<-evi[evi<1/4]
nevi<-length(evi)
conf.level<-sort(conf.level[conf.level>0.5&conf.level<1])
nci<-length(conf.level)
#####
##### the computation of residual coefficient of variation
#####
x<-rev(data)
t<-data.frame(i=1:n,x=x,cx=c(0,cumsum(x)[1:(n-1)]),cx2=c(0,cumsum(x^2)[1:(n-1)]))
f<-function(p) (1/(p[1]-2))*(p[4]-(1/(p[1]-1))*p[3]^2)/((1/(p[1]-1))*p[3]-p[2])^2
rcv<-rev(sqrt(apply(t,1,f)))[(omit+1):n])
#####
##### the computation of confidence intervals
#####
u<-c()
l<-c()
if ((nevi*nci)!=0)
{
qs<-qnorm(1-(1-expand.grid(evi,conf.level)[,2])/2)
for (i in 1:(nevi*nci))
{
j<-i-trunc((i-1)/nevi)*nevi
cv<-1/sqrt(1-2*evi[j])
sigma<-sqrt((1-evi[j])^2*(6*evi[j]^2-evi[j]+1)/((1-2*evi[j])^2*(1-3*evi[j])*(1-4*evi[j]))) #if evi=0, th
u<-cbind(u,cv+qs[i]*sigma/sqrt(n-ks))
l<-cbind(l,cv-qs[i]*sigma/sqrt(n-ks))
}
}
#####
##### plot of residual coefficient of variation
#####
par(mar = c(5, 6, 5, 3))
ifelse(nevi!=0,setpoint<-c(rcv[!is.na(rcv)],u[ksr,],l[ksr,]),setpoint<-rcv[!is.na(rcv)])
plot(ks+k0-1,rcv,type="l",ylim=c(min(setpoint),max(setpoint)),
xlab="Excluded sample size",ylab=paste("Coefficient of variation"),col="blue",...)
title(main=main,outer=F,adj=1,line=4,cex.main=1)
mtext("Threshold",side=3,line=2.5,cex=0.66)
ti<-c(1+(k0-1),sort(axTicks(1))[c(-1,-(length(axTicks(1)))])],min(max(axTicks(1)),n+(k0-1)))
lti<-data[ti-(k0-1)]+threshold

```



```

##### inside
#####
if (!is.na(evi))
{
  if (length(evi)>1) stop("Enter only one value or NA-value for extreme value index (evi) parameter")
  if (evi>=1/2) stop("The hypothesis testing needs evi<1/2",call. = TRUE)
}
p<-round(exp(log(omit/n)/m),digits=2)
Ws<-p^(0:m)
Ps<-1-Ws
Cs<-frcv(data,Ps)
if (is.na(evi)) cx<-(1-p)*sum(Ws*Cs)/(1-p^(m+1)) else cx=1/sqrt(1-2*evi)
xi<-(cx^2-1)/(2*cx^2)
tm<-n*sum(Ws*(Cs-cx)^2)
#####
##### output
#####
Tms.pvalue(n=n,tms=tm/(m+1),m=m,evi=xi,hat=is.na(evi),omit=omit,nsim=nsim)
}

#####
#####
#####
#####
#
#          thrselect
#
#####
#####
#####
#####
#####

thrselect<-function(data, threshold = NA, nextremes = NA, omit = 16, evi = NA, m = 10, nsim = 100, conf
{
#####
##### controls and inizialization of vars
#####
omit<-floor(omit)
if (omit<2) warning("The parameter omit have to be bigger than 1",call.=TRUE)
omit<-max(omit,2)
data <- as.numeric(data)
data<-data[!is.na(data)]
data <- sort(data)
if (is.na(nextremes) && is.na(threshold)) threshold<-min(data)
if (!is.na(nextremes) && !is.na(threshold)) stop("Enter EITHER a threshold or the number of upper extremes")
if (!is.na(nextremes)) threshold<-rev(data)[as.numeric(nextremes)]
data<- data[data >= threshold]- threshold
n <- length(data)
#####
##### inside

```

```

#####
p<-round(exp(log(omit/n)/m),digits=2)
Ws<-p^(0:m)
Ps<-1-Ws
if (length(data[data>=quantile(data,Ps[m+1])])<2) stop("Reduce m or increase omit")
Ns<-round(n*Ws)
Qs<-as.vector(quantile(data,Ps))
Cs<-frcv(data,Ps)
aux<-function(r) sum(Cs[r:(m+1)]*Ws[r:(m+1)])/sum(Ws[r:(m+1)])
if (!is.na(evi)) CXs<-rep(1/sqrt(1-2*evi),m+1) else CXs<-apply(matrix(1:(m+1),ncol=1),1,FUN=aux)
XIs<-evicv(CXs)
aux<-function(r) n*sum(Ws[r:(m+1)]*(Cs[r:(m+1)]-CXs[r])^2)/(m+1-r)
TMSs<-apply(matrix(1:(m+1),ncol=1),1,FUN=aux)
aux<-function(r) Tms.pvalue(n=Ns[r],tms=TMSs[r],m=m-r+1,evi=XIs[r],hat=is.na(evi),omit,nsim)$pvalue
PVs<-apply(matrix(1:(m),ncol=1),1,FUN=aux)
PVs<-c(PVs,1)
#####
##### output #####
options<-data.frame(m=seq(m,0,-1),nextremes=Ns,threshold=Qs+threshold,rcv=Cs,cvopt=CXs,evi=XIs,tms=TMSs)
solution<-options$options$pvalue>(1-conf.level),][1,]
outlist<-list(solution=solution,options=options)
if (oprint==T) print(solution)
thrselect<-outlist
}

#####
# tdata
#
#####
# tdata<-function(data, threshold = NA, nextremes = NA,sigma=NA)
{
#####
### controls and inizialization of vars
#####
data <- as.numeric(data)
data<-data[!is.na(data)]
data <- sort(data)
if (is.na(nextremes) && is.na(threshold)) threshold<-min(data)
if (!is.na(nextremes) && !is.na(threshold)) stop("Enter EITHER a threshold or the number of upper extremes")
if (!is.na(nextremes)) threshold<-rev(data)[as.numeric(nextremes)]
data<- data[data >=threshold]- threshold

```









```

getnpv <- function(p,m,Ps,Ws,n,xi,hat,nsim)
{
tmbvector <- replicate(n = nsim, expr = gettmb(p=p,m=m,Ps=Ps,Ws=Ws,n=n,xi=xi,hat=hat))
sum(tmbvector > (tms*(m+1)))
}
nrej<-getnpv(p,m,Ps,Ws,n,xi,hat,nsim)
data.frame(cxopt=cx,evi=xi,tms=tms,pvalue=nrej/nsim,row.names="")
}

#####
##### function for fitting gpd
#####

egpd<-function(x,evi=NA,heavy=NA)
{
int<-c(-100*max(x),100*max(x))
if(!is.na(evi)) heavy<-(evi>0)
if(!is.na(heavy)) int<-int*c(heavy,1-heavy)
fk<-function(sigma) -mean(log(1-x/sigma))
fp<-function(sigma) length(x)*(-log(fk(sigma)*sigma)+fk(sigma)-1)
fk2<-function(sigma) -(evi-mean(log(1-x/sigma)))^2
if (is.na(evi)) sigma<-optimize(fp,interval=int,maximum=T)$maximum
else sigma<-optimize(fk2,interval=int,maximum=T)$maximum
list(xi=-fk(sigma),psi=fk(sigma)*sigma)
}

```

## A.5 Funció thrselect

```

#####
frcv<-function(data,Ps)
{
Qs<-as.vector(quantile(data,Ps))
Cs<-c()
for(k in 1:length(Qs))
{
if (length(data[data>=Qs[k]]-Qs[k])==1) Cs<-c(Cs,0)
else Cs<-c(Cs,sd(data[data>=Qs[k]]-Qs[k])/mean(data[data>=Qs[k]]-Qs[k]))
}
Cs
}

#####
#####
```

```

#
#           thrselect
#
#####
#####
#####
#####

thrselect<-function(data, threshold = NA, nextremes = NA, omit = 16, evi = NA, m = 10, nsim = 100, conf = 0.95)
{
  PVs <- c(NA)
  i <- 1
  omit<-floor(omit)
  if (omit<2) warning("The parameter omit have to be bigger than 1",call.=TRUE)
  omit<-max(omit,2)
  data <- as.numeric(data)
  data<-data[!is.na(data)]
  data <- sort(data)
  if (is.na(nextremes) && is.na(threshold)) threshold<-min(data)
  if (!is.na(nextremes) && !is.na(threshold)) stop("Enter EITHER a threshold or the number of upper extremes")
  if (!is.na(nextremes)) threshold<-rev(data)[as.numeric(nextremes)]
  data<- data[data >= threshold]- threshold
  n <- length(data)
  p<-round(exp(log(omit/n)/m),digits=2)
  Ws<-p^(0:m)
  Ps<-1-Ws
  if (length(data[data>=quantile(data,Ps[m+1])])<2) stop
  ("Reduce m or increase omit")
  Ns<-round(n*Ws)
  Qs<-as.vector(quantile(data,Ps))
  Cs<-frcv(data,Ps)
  aux<-function(r) sum(Cs[r:(m+1)]*Ws[r:(m+1)]) / sum(Ws[r:(m+1)])
  if (!is.na(evi)) CXs<-rep(1/sqrt(1-2*evi),m+1) else CXs<-apply(matrix(1:(m+1),ncol=1),1,FUN=aux)
  XIs<-evicv(CXs)
  aux<-function(r) n*sum(Ws[r:(m+1)]*(Cs[r:(m+1)]-CXs[r])^2)/(m+1-r)
  TMSs<-apply(matrix(1:(m+1),ncol=1),1,FUN=aux)

  aux<-function(r) Tms.pvalue(n=Ns[r],tms=TMSs[r],m=m-r+1,evi=XIs[r],hat=is.na(evi),omit,nsim)$pvalue

  while(i<m){
    PVs[i] <- aux(i)
    if(PVs[i]>=0.05){break}
    else{
      i=i+1
    }
  }

  Ns <- Ns[1:i]
}

```

```

    Qs <- Qs[1:i]
    Cs <- Cs[1:i]
    CXs <- CXs[1:i]
    XIIs <- XIIs[1:i]
    TMSs <- TMSs[1:i]

    return(Qs[i]+threshold)
}

```

## A.6 Potència del test mètode CV

```

ps<-c()
for (contt in 1:50)
{

  x1<-abs(rt(1000,0.2))
  x2<- abs(rt(1000,0.2))

  xmin1<- thrselect(x1,m=100)
  xmin2<- thrselect(x2,m=100)

  x1aux <- x1[x1>xmin1]
  x2aux <- x2[x2>xmin2]

  aux1 <- x1aux/xmin1
  aux2 <- x2aux/xmin2

  alpha1 <- mean(log(aux1))^-1
  alpha2 <- mean(log(aux2))^-1

  est0<-alpha1/alpha2
  est0p<-alpha1/(alpha2+alpha1)
  est0
  est0p

  tail1<-x1[x1>xmin1]
  tail2<-x2[x2>xmin2]

  l1<-length(tail1)
  l2<-length(tail2)

  l<-min(l1,l2)

  tail1<-rev(sort(x1))[1:l]
  tail2<-rev(sort(x2))[1:l]

  tails<-c(tail1/xmin1,tail2/xmin2)
}

```

```

estss<-c()
estsp<-c()
for (cont in 1:50)
{
  t<-sample(2*l,1)
  tail1<-tails[t]
  tail2<-tails[-t]
  alpha1 <- mean(log(tail1))^-1
  alpha2 <- mean(log(tail2))^-1
  estss<-c(estss,alpha1/alpha2)
  estsp<-c(estsp,alpha1/(alpha2+alpha1))
}
z<-ecdf(estss)
zp<-ecdf(estsp)
ps<-rbind(ps,c(est0,z(est0),zp(0.5),zp(est0p)))
}

(sum(ps[,2]<0.025)+sum(ps[,2]>0.975))/50

```

## A.7 Estudi de les accions

```

#####
# NETFLIX

df<-function(x,alpha,c) ifelse(x>=c,(alpha/c)*(x/c)^(-(alpha+1)),0)

NETFLIX <- read.csv("NETFLIX.csv",header=T,sep=",")
NETFLIX <- NETFLIX[length(NETFLIX$Date):1,]
NFLX_close=NETFLIX$Adj.Close
prices_NFLX=as.numeric(NFLX_close)

returns_NFLX =log(prices_NFLX[2:length(prices_NFLX)]/prices_NFLX[1:length(prices_NFLX)-1])

mean(returns_NFLX)
sd(returns_NFLX)
median(returns_NFLX)
min(returns_NFLX)
max(returns_NFLX)
plfit(alfanegNFLX$alpha)
plfit(alfaposNFLX$alpha)

plot(returns_NFLX,type="l",main="Sèrie temporal de les rendibilitats de Netflix",ylab="Rendibilitats",xlab="")

hist(returns_NFLX,ylab="Freqüència",xlab="Rendibilitats", main="Histograma de les rendibilitats de Netflix")

x <- returns_NFLX
alfanegNFLX <- abs(x[x<median(x)])+0.00000001
alfaposNFLX <- abs(x[x>median(x)])-median(x)
alfaneg <- as.numeric(plfit(alfanegNFLX)$alpha)
alfapos <- as.numeric(plfit(alfaposNFLX)$alpha)
xminneg <- as.numeric(plfit(alfanegNFLX)$xmin)

```

```

xminpos <- as.numeric(plfit(alfaposNFLX)$xmin)

trans_hist1 <- hist(log(alfaposNFLX),plot=F)
trans_hist1 <- hist(alfaposNFLX,breaks=exp(trans_hist1$breaks))

trans_hist2 <- hist(log(alfanegNFLX),plot=F)
trans_hist2 <- hist(alfanegNFLX,breaks=exp(trans_hist2$breaks))

plot(trans_hist1$mids,trans_hist1$density,log="xy",ylab="Densitat",xlab="",col="Red")
xs1<-sort(alfaposNFLX)
ys1<-c();for(xi in xs1) ys1<-c(ys1,df(xi,alfapos,xminpos))
lines(xs1,ys1,col="red")
abline(v=xminpos,lty=2,col="red")

points(trans_hist2$mids,trans_hist2$density,
       ylab="",xlab="",col="Blue",log="xy")
xs2<-sort(alfanegNFLX)
ys2<-c();for(xi in xs2) ys2<-c(ys2,df(xi,alfaneg,xminneg))
lines(xs2,ys2,col="blue")
abline(v=xminneg,lty=2,col="blue")

title(main="Modelització de les cues de Netflix")
legend("bottomleft",c("Cua Positiva","Cua Negativa"),col=c("Red","Blue"),lty=c(1,1))

##### SPS

SPS <- read.csv("SPSC.csv",header=T,sep=",")
SPS <- SPS[length(SPS$Date):1,]
SPS_close=SPS$Adj.Close
prices_SPS=as.numeric(SPS_close)

returns_SPS =log(prices_SPS[2:length(prices_SPS)]/prices_SPS[1:length(prices_SPS)-1])

mean(returns_SPS)
sd(returns_SPS)
median(returns_SPS)
min(returns_SPS)
max(returns_SPS)
plfit(alfanegSPS)$alpha
plfit(alfaposSPS)$alpha

plot(returns_SPS,type="l",main="Sèrie temporal de les rendibilitats de SPS",ylab="Rendibilitats",xlab="T")

hist(returns_SPS,ylab="Frequència",xlab="Rendibilitats", main="Histograma de les rendibilitats de SPS")

x<- returns_SPS
alfanegSPS <- abs(x[x<median(x)])+0.00000001
alfaposSPS <- abs(x[x>median(x)])-median(x)
alfaneg <- as.numeric(plfit(alfanegSPS)$alpha)
alfapos <- as.numeric(plfit(alfaposSPS)$alpha)
xminneg <- as.numeric(plfit(alfanegSPS)$xmin)

```

```

xminpos <- as.numeric(plfit(alfaposSPS)$xmin)

trans_hist1 <- hist(log(alfaposSPS),plot=F)
trans_hist1 <- hist(alfaposSPS,breaks=exp(trans_hist1$breaks))

trans_hist2 <- hist(log(alfanegSPS),plot=F)
trans_hist2 <- hist(alfanegSPS,breaks=exp(trans_hist2$breaks))

plot(trans_hist1$mid,trans_hist1$density,log="xy",ylab="Densitat",xlab="",col="Red")
xs1<-sort(alfaposSPS)
ys1<-c();for(xi in xs1) ys1<-c(ys1,df(xi,alfapos,xminpos))
lines(xs1,ys1,col="red")
abline(v=xminpos,lty=2,col="red")

points(trans_hist2$mid,trans_hist2$density,
       ylab="",xlab="",col="Blue",log="xy")
xs2<-sort(alfanegSPS)
ys2<-c();for(xi in xs2) ys2<-c(ys2,df(xi,alfaneg,xminneg))
lines(xs2,ys2,col="blue")
abline(v=xminneg,lty=2,col="blue")

title(main="Modelització de les cues de SPS")
legend("bottomleft",c("Cua Positiva","Cua Negativa"),col=c("Red","Blue"),lty=c(1,1))

#####
### McDonald's #####
#####

MCD <- read.csv("MCD.csv",header=T,sep=",")
MCD_close=MCD$Adj.Close
prices_MCD=as.numeric(MCD_close)

returns_MCD =log(prices_MCD[2:length(prices_MCD)]/prices_MCD[1:length(prices_MCD)-1])

mean(returns_MCD)
sd(returns_MCD)
median(returns_MCD)
min(returns_MCD)
max(returns_MCD)

plot(returns_MCD,type="l",main="Sèrie temporal de les rendibilitats de McDonald's",ylab="Rendibilitats")

hist(returns_MCD,ylab="Frequència",xlab="Rendibilitats", main="Histograma de les rendibilitats de McDonald's")

x<- returns_MCD
alfanegMCD <- abs(x[x<median(x)])+0.00000001
alfaposMCD <- abs(x[x>median(x)])-median(x)
alfaneg <- as.numeric(plfit(alfanegMCD)$alpha)
alfapos <- as.numeric(plfit(alfaposMCD)$alpha)
xminneg <- as.numeric(plfit(alfanegMCD)$xmin)
xminpos <- as.numeric(plfit(alfaposMCD)$xmin)

```

```

trans_hist1 <- hist(log(alfaposMCD),plot=F)
trans_hist1 <- hist(alfaposMCD,breaks=exp(trans_hist1$breaks))

trans_hist2 <- hist(log(alfanegMCD),plot=F)
trans_hist2 <- hist(alfanegMCD,breaks=exp(trans_hist2$breaks))

plot(trans_hist1$mids,trans_hist1$density,log="xy",ylab="Densitat",xlab="",col="Red")
xs1<-sort(alfaposMCD)
ys1<-c();for(xi in xs1) ys1<-c(ys1,df(xi,alfapos,xminpos))
lines(xs1,ys1,col="red")
abline(v=xminpos,lty=2,col="red")

points(trans_hist2$mids,trans_hist2$density,
ylab="",xlab="",col="Blue",log="xy")
xs2<-sort(alfanegMCD)
ys2<-c();for(xi in xs2) ys2<-c(ys2,df(xi,alfaneg,xminneg))
lines(xs2,ys2,col="blue")
abline(v=xminneg,lty=2,col="blue")

title(main="Modelització de les cues de McDonald's")
legend("bottomleft",c("Cua Positiva","Cua Negativa"),col=c("Red","Blue"),lty=c(1,1))

```

## A.8 Estudi de una cartera de dues accions

```

for(i in 0:100){

  j <- i/100
  ps<-c()
  cartera <- j*returns_NFLX+((1-j)*returns_SPS)
  x1 <- abs(cartera[cartera<median(cartera)])+0.000001
  x2 <- abs(cartera[cartera>median(cartera)])+0.000001

  alpha1<-plfit(x1)$alpha
  alpha2<-plfit(x2)$alpha

  est0<-alpha1/alpha2

  min1<-plfit(x1)$xmin
  min2<-plfit(x2)$xmin

  tail1<-x1[x1>min1]
  tail2<-x2[x2>min2]

  l1<-length(tail1)
  l2<-length(tail2)

  l<-min(l1,l2)
}

```

```

tail1<-rev(sort(x1))[1:1]
tail2<-rev(sort(x2))[1:1]

min1<-min(tail1)
min2<-min(tail2)

body1<-x1[x1<=min1]
body2<-x2[x2<=min2]

tails<-c(tail1/min1,tail2/min2)

estss<-c()
estsp<-c()
for (cont in 1:50)
{
  t<-sample(2*l,1)
  tail1<-tails[t]*min1
  tail2<-tails[-t]*min2
  sx1<-c(body1,tail1)
  sx2<-c(body2,tail2)
  alpha1<-plfit(sx1)$alpha
  alpha2<-plfit(sx2)$alpha
  ests<-c(estss,alpha1/alpha2)
  estsp<-c(estsp,alpha1/(alpha2+alpha1))
}
z<-ecdf(ests)
pval[i+1] <- z(est0)
}
pval[44]<-0.12
pvalors <- data.frame(seq(0,1,by=0.01),pval)
names(pvalors) <- c("% d'inversio en Netflix","p-value")
plot(pvalors,ylim=c(0,1))
abline(h=0.025,col="red")
abline(h=0.975,col="red")

carteragrafic <- 0.5*returns_NFLX+0.5*returns_SPS
alphaposC1 <- abs(carteragrafic[carteragrafic>median(carteragrafic)])+0.00000001
alphanegC1 <- abs(carteragrafic[carteragrafic<median(carteragrafic)])+0.00000001
alfaneg <- as.numeric(plfit(alphanegC1)$alpha)
alfapos <- as.numeric(plfit(alphaposC1)$alpha)
xminneg <- as.numeric(plfit(alphanegC1)$xmin)
xminpos <- as.numeric(plfit(alphaposC1)$xmin)

trans_hist1 <- hist(log(alphaposC1),plot=F)
trans_hist1 <- hist(alphaposC1,breaks=exp(trans_hist1$breaks))

trans_hist2 <- hist(log(alphanegC1),plot=F)
trans_hist2 <- hist(alphanegC1,breaks=exp(trans_hist2$breaks))

plot(trans_hist1$mid,trans_hist1$density,log="xy",ylab="Densitat",xlab="",col="Red")

```

```

xs1<-sort(alphaposC1)
ys1<-c();for(xi in xs1) ys1<-c(ys1,df(xi,alfapos,xminpos))
lines(xs1,ys1,col="red")
abline(v=xminpos,lty=2,col="red")

points(trans_hist2$mids,trans_hist2$density,
       ylab="",xlab="",col="Blue",log="xy")
xs2<-sort(alphanegC1)
ys2<-c();for(xi in xs2) ys2<-c(ys2,df(xi,alfaneg,xminneg))
lines(xs2,ys2,col="blue")
abline(v=xminneg,lty=2,col="blue")

title(main="Modelització de les cues de la cartera")
legend("bottomleft",c("Cua Positiva","Cua Negativa"),col=c("Red","Blue"),lty=c(1,1))

par(mfrow=c(1,2))
ccdfplot(alfaposMCD,fitpot(alfaposMCD,evi=1/alfapos,threshold=xminpos),log="xy",main="Distribució acumulada de la cua positiva")
ccdfplot(alfanegMCD,fitpot(alfanegMCD,evi=1/alfaneg,threshold=xminneg),log="xy",main="Distribució acumulada de la cua negativa")

```

## A.9 Estudi de la cartera de 3 accions

```

pval2 <- matrix(NA, ncol=101,nrow=101)

for(i in 0:100){
  for(j in 0:100){
    aux1 <- i/100
    aux2 <- j/100

    if(aux1+aux2>1){
      pval2[(i+1),(j+1)] <- NA
    }
    else{
      aux3 <- 1-aux1-aux2

      ps<-c()
      cartera <- aux1*returns_NFLX+aux2*returns_SPS+aux3*returns_MCD
      x1 <- abs(cartera[cartera<median(cartera)])+0.000001
      x2 <- abs(cartera[cartera>median(cartera)])+0.000001

      alpha1<-plfit(x1)$alpha
      alpha2<-plfit(x2)$alpha

      est0<-alpha1/alpha2

      min1<-plfit(x1)$xmin
      min2<-plfit(x2)$xmin
    }
  }
}
```

```

tail1<-x1[x1>min1]
tail2<-x2[x2>min2]

l1<-length(tail1)
l2<-length(tail2)

l<-min(l1,l2)

tail1<-rev(sort(x1))[1:l]
tail2<-rev(sort(x2))[1:l]

min1<-min(tail1)
min2<-min(tail2)

body1<-x1[x1<=min1]
body2<-x2[x2<=min2]

tails<-c(tail1/min1,tail2/min2)

estss<-c()
estsp<-c()
for (cont in 1:50)
{
  t<-sample(2*l,1)
  tail1<-tails[t]*min1
  tail2<-tails[-t]*min2
  sx1<-c(body1,tail1)
  sx2<-c(body2,tail2)
  alpha1<-plfit(sx1)$alpha
  alpha2<-plfit(sx2)$alpha
  ests<-c(estss,alpha1/alpha2)
  estsp<-c(estsp,alpha1/(alpha2+alpha1))
}
z<-ecdf(ests)
pval2[(i+1),(j+1)] <- z(est0)
}
}
}

image(pval2,xlab="% d'inversio en Netflix",ylab="% d'inversio en SPS")

```