TREBALL DE FI DE GRAU

Grau en Estadística

Títol: Introducció als mètodes robustos

Autor: Arnau Mercader Luque

Director: Josep Mª Oller Sala

Departament: Estadística

Convocatòria: Setembre 2016

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Introducció als mètodes robustos

Arnau Mercader Luque

Grau en Estadística
Setembre 2016

Resum

Tots els procediments estadístics es basen en assumir hipòtesis sobre les dades observades. Aquestes hipòtesis ajuden a construir models més simples i manejables des d'un punt de vista teòric i alhora computacional. Una de les simplificacions més utilitzades és suposar que les dades observades segueixen una distribució exactament normal. Aquest enfocament, que s'anomena normalment clàssic, ha dominat l'estadística en els últims dos segles. No obstant, en la pràctica la majoria de dades observades segueixen el model normal assumit, però algunes observacions segueixen un patró diferent o directament no en segueixen cap. Intuïtivament, es podria pensar que si la hipòtesi de normalitat és certa, tot i que sigui aproximadament, els resultats de la teoria clàssica també seran certs aproximadament. Lamentablement, aquest no és el cas, la gran majoria de mètodes clàssics perden completament les seves propietats òptimes davant de petites desviacions del model assumit. L'objectiu de l'estadística robusta és aconseguir estimacions que siguin gairebé tan bones com la dels mètodes clàssics quan la distribució de les dades observades sigui exactament l'assumida i que en la presència d'outliers aquestes estimacions siguin semblants a les que s'aconseguirien mitjançant mètodes clàssics sense dades atípiques.

Paraules clau: mitjana truncada, mediana, simulació Monte Carlo, funcional, funció d'influència.
MSC(2010): Primari 62F35. Secundari 65C05, 58C20.

Introduction to robust methods

Arnau Mercader Luque

Degree of Statistics
September 2016

Abstract

All the statistical procedures are based on assuming hypotheses about the observed data. These hypotheses help us to build more simple and manageable models from a theoretical and computational standpoint. One of the most widely used simplifications is to assume that the observed data follow a purely normal distribution. This approach, which is typically known as the classical one, has ruled statistics for the last two centuries. However, in practice most of the observed data follow the assumed normal model, but some observations either follow a different pattern or they do not show any trend. Intuitively, it could be thought that if the normality hypothesis is true, at least approximately, the results derived from classical theory could also be true to some extent. Regretfully, this is not the case as most of the classical methods are prone to lose completely their optimal properties in the event of having small deviations in the assumed model. The aim of robust statistics is to achieve estimates which are almost as good as those obtained from classical methods when the observed data distribution corresponds exactly with the assumed one and that in the presence of outliers these estimates are similar those presumably produced via classical methods without non-typical data.

Key words: trimmed mean, median, Monte Carlo simulation, functional, influence function.
MSC(2010): Primary 62F35. Secondary 65C05, 58C20.

Índex

Llista de gràfics

Llista de Taules

1 Introducció

Tots els procediments estadístics es basen en assumir hipòtesis sobre les dades observades. Sota aquests supòsits es poden deduir procediments òptims. Així com en el model de regressió on el procediment òptim és el de mínims quadrats. En general, per a models paramètrics, els procediments òptims clàssics són els basats en màxima versemblança. El principal desavantatge d'aquests mètodes és que són sensibles a modificacions de les hipòtesis. Per exemple, petites desviacions de les dades manifestades per algunes observacions atípiques poden invalidar la metodologia en la qual es basen aquests mètodes. En regressió, un diagnòstic dels residus pot ajudar a veure on es produeix la ruptura de les hipòtesis, però pot arribar a ser difícil i alhora requerir molt de temps. Els mètodes robustos són resistents a petites desviacions, i, a la vegada, són eficients quan els models clàssics són vàlids. Tot i això, aquests mètodes no han de substituir els altres, seran una eina més per a trobar el model final.

Primer de tot es farà una introducció a la robustesa i un plantejament inicial. Es definiran una sèrie de conceptes per a comparar estimadors clàssics amb estimadors robustos de localització i dispersió mitjançant un estudi de simulació. En una segona part, amb èmfasi teòric, es definirà la funció d'influència, concepte molt important per a l'estadística robusta, i s'il·lustraran alguns exemples. També es mencionaran altres mesures de robustesa, algunes que deriven de la mateixa funció d'influència i el punt de ruptura.

2 Robustesa

2.1 Inicis de l'estadística robusta

L'ús estadístic de la paraula *robust* apareix per primer cop en l'article "NonNormality and Tests on Variance" publicat per George Box el 1953 a *Biometrika* (Stigler [2010]). Allà, Box va indicar la mancança de robustesa en proves de comparació de variàncies. Es va dirigir principalment al test de Bartlett, que alguns havien suggerit com a pas preliminar, per a validar el supòsit d'igualtat de variàncies abans de realitzar un test ANOVA de les mitjanes. Tukey també va argumentar que una petita contaminació en la distribució de les dades podia ocasionar greus problemes, argument sorprenent per molts estadístics. Només Fisher va preveure que una petita contaminació d'aquest tipus podria tenir un efecte tan gran.
Va ser uns anys més tard quan Peter Huber, amb una beca d'investigació a Berkeley, va publicar l'any 1964 a *Annals* l'article "Robust Estimation of a Location Paremeter", on demostrava que hi havia una resposta millor al problema d'estimar un paràmetre de localització. L'estiu de 1968, Frank Hampel va completar la tesi de Huber, formalitzant el concepte de robustesa a través d'un derivat funcional de l'estimador a través de la "Influence Curve"(Funció d'influència). Altres persones que han aportat molt a la estadística robusta són Victor J. Yohai (MM-estimador), Ricardo A. Maronna i Elvezio Ronchetti entre d'altres. S'ha anat consolidant una metodologia que intenta ser més realista que els mètodes clàssics.

2.2 Propòsits

El que es vol aconseguir amb l'ús d'estimadors robustos és:

- Resistència a observacions inusuals. Observacions que no segueixen el patró de la majoria de les dades *(Robustness of validity)*.
- Resistència a violacions del supòsit de la distribució subjacent a les dades *(Robustness of efficiency)*.

Quan es fa inferència estadística a vegades només es tenen en compte les observacions, quan els supòsits que hi han al darrere són igual d'importants. Molts dels procediments estadístics més emprats es basen en el supòsit de normalitat, supòsit molt sensible a petites desviacions del model assumit. Els procediments robustos són una reacció a això.

Huber i Ronchetti descriuen tres propietats que un "bon" procediment robust ha de satisfer:

- Quan el model assumit és correcte, el procediment ha de donar resultats amb una variància mostral petita.
- Petites desviacions del model assumit han de tenir un efecte mínim en el rendiment del procediment.
- Grans desviacions del model assumit no han de causar una catàstrofe.

2.3 Plantejament

Un supòsit molt comú en estadística és suposar que les dades han estat generades per un mecanisme aleatori i que alhora aquest pot ser representat per un membre F_θ de la família paramètrica de funcions de distribució

$$\mathcal{F} = \{F_\theta : \theta \in \Theta\} \quad (1)$$

La característica més rellevant d'aquests models matemàtics és suposar que el mecanisme estocàstic que genera les dades és totalment conegut a excepció del paràmetre desconegut θ. Un dels principals problemes consisteix en estudiar θ a partir d'un estimador mostral $\hat{\theta}_n$ amb bones propietats estadístiques desitjables: biaix nul o gairebé nul amb variància petita. Un mètode que, en general, satisfà aquests requeriments és el mètode de màxima versemblança (*maximum likelihood estimation, MLE*), popularitzat per Ronald Fisher a inicis del segle XX. Molts dels mètodes emprats a la pràctica van ser derivats a partir de models paramètrics, la majoria models normals o també anomenats gaussians. A més, les propietats d'aquests mètodes s'han estudiat, com ja hem dit, suposant que es coneix la distribució de les dades.

Desafortunadament, a la pràctica aquests mecanismes aleatoris rarament obeiran amb exactitud un model paramètric. Tot i això, en molts casos, el model paramètric simplifica i proporciona una aproximació raonable del mecanisme aleatori $F \subset \mathcal{F}$ que controla la generació de les observacions. En resum, l'ús del model (1) i d'estimadors de màxima versemblança $\hat{\theta_n}$ es sol justificar pels següents arguments:

a) El model (1) es compleix aproximadament.

b) Les bones propietats del mètode utilitzat per estimar θ (màxima versemblança) són contínues, de manera que, si el model és aproximadament vàlid, l'estimador $\hat{\theta_n}$ és aproximadament òptim.

Amb freqüència, la propietat (a) és certa, mentre que la (b) no ho és en moltes situacions. En el model normal, mitificat per molts mètodes i estadístics es pot demostrar que l'eficiència de la mitjana aritmètica $\bar{X_n}$ (estimador òptim sota el model normal) pot ser pròxima a zero per distribucions que són arbritàriament pròximes a la normal. Per a mostrar un exemple, considerem 20 observacions independents $x_i = \mu + \epsilon_i$, dinou de les quals són normals amb mitjana 1 i desviació típica 0.03, i 1 observació és una dada atípica (outlier) x. La $\bar{X} = \frac{19}{20}(1) + \frac{1}{20}(x) = 0.95(1) + 0.05(x) \to \infty$, sí $x \to \infty$. Amb una sola observació, la mitjana pot tendir a infinit, per tant, pot pendre qualsevol valor.

Aquest exemple, intenta mostrar que $\bar{X_n}$, l'estimador de màxima versemblança i alhora no esbiaixat i de mínima variància (*uniformly minimum-variance unbiased, UMVU*) sota el model normal pot produir estimacions molt inestables si les dades pertanyen a una distribució que és pròxima, però no exactament igual a la normal.

Una manera d'incorporar explícitament la qüestió de que el model és només una simplificació de la distribució F_θ és suposar que la distribució F pertany a la família $\mathcal{F}_t$, definida com

$$\mathcal{F}_t = \{(1-t)F_\theta + t(G) : \theta \in \Theta\} \quad (2)$$

amb $0 < t < 0.5$ fixat, G arbritària i desconeguda

Degut a que ara la distribució $F \subset \mathcal{F}_t$, els estimadors en general presentaran biaix i per tant , el biaix serà un aspecte important a considerar. Un altre aspecte important és que ara F no està del tot especificada, donat que G és desconeguda. Una manera d'entendre millor el propòsit de considerar el model (2) i alhora il·lustrar diferents alternatives als estimadors clàssics és realitzar una comparació d'estimadors donades diferents situacions via simulació. Primer de tot, cal definir el mètode de simulació que es farà servir, els estimadors a comparar i les mesures a contrastar.

2.4 Comparació d'estimadors

Mètode de Montercarlo

Per a simular utilitzarem el mètode de Montecarlo, que el podem definir com l'art d'aproximar un valor esperat a partir de la mitjana mostral d'una funció de variables aleatòries simulades.

Sigui una variable aleatòria X, amb funció de densitat $f_X(x) > 0$ per a un conjunt de valors $x \in \mathcal{X}$, el valor esperat d'una funció g en X és

$$E(g(X)) = \sum_{x \in \mathcal{X}} g(x)\, f_X(x) \quad \text{sí X és discreta}$$

$$E(g(X)) = \int_{x \in \mathcal{X}} g(x)\, f_X(x)\, dx \quad \text{sí X és contínua}$$

Agafem S mostres dels valors $x \in \mathcal{X}$,$(x_1, ..., x_S)$, i calculem la mitjana de la funció g(x) en les S mostres, aconseguint l'estimador Monte Carlo com a

$$\tilde{g_s}(X) = \frac{1}{S} \sum_{i=1}^{S} g(x_i)$$

Si $E(g(X))$, existeix, la llei feble dels grans nombres ens diu que $\forall\, \epsilon$ arbritari i petit

$$\lim_{n \to \infty} P(|\tilde{g_s}(X) - E(g(X)) \geq \epsilon|) = 0$$

Això ens diu que sí S és gran, les diferències entre l'estimador Montecarlo $\tilde{g_s}(X)$ i l'estimador a estudiar seran molt ínfimes. També podem veure que $\tilde{g_s}(X)$ és un estimador no esbiaixat per $E(g(X))$:

$$E(\tilde{g_s}(X)) = E(\frac{1}{S} \sum_{i=1}^{S} g(x_i)) = \frac{1}{S} \sum_{i=1}^{S} E(g(x_i)) = E(g(X))$$

Procediment amb R

S'utilitzarà R per a fer la comparació d'estimadors via simulació (*veure codis enÀnnex R*). Cal generar S mostres independents de mida n sota les condicions d'interès, en aquest cas, diferents models de la família exponencial amb una perturbació t associada a una distribució G, és a dir, el plantejament del model (2). S'analitzaran diferents escenaris modificant les distribucions, la mida mostral n, on n podrà ser = $\{10, 20, 50\}$ i els percentatges de perturbació, on t podrà ser = $\{0.01, 0.05, 0.10, 0.25, 0.4\}$.

El procediment a seguir és el següent:

- Pas previ a la simulació de cada mostra $\{s_i : i = 1, ..., S\}$:

  1) Fixar el percentatge t.

  2) Simular una distribució Bernoulli de mida n amb probabilitat $P(1-t)$ d'èxit i $P(t)$ altrament. Fixada la perturbació t, la mostra no sempre estarà perturbada en t. D'aquesta manera s'intenta plasmar un escenari més realista.

- Calcular el valor de l'estimador T^k per a mostra $s_i \Rightarrow T^k_{s_1}, T^k_{s_2}, ..., T^k_{s_S}$.

  On $k = 1, ..., K$ són els estimadors a comparar i $T(\cdot) = g(\cdot)$.

- Calcular la mitjana de totes les S mostres independents ($\tilde{g}_s(X)$) i utilitzar el valor obtingut com a estimador de $E(g(X))$.

- Comparar els estimadors amb dues mesures: el biaix i l'error quadràtic mig (EQM) (*mean squared error, MSE*).

Per a obtenir aquestes mesures cal calcular el següent:

$$(1)\ \widehat{X_{MC}} = S^{-1} \textstyle\sum_{s=1}^{S} T^k_s = \bar{T}^{\,k} = \tilde{g}_s(X)$$

$$(2)\ \widehat{B_{MC}} = \bar{T}^{\,k} - \theta$$

$$(3)\ \widehat{SD_{MC}} = \sqrt{\frac{S-1}{S}} \sqrt{\frac{1}{S-1} \textstyle\sum_{s=1}^{S} (T^k_s - \bar{T}^{\,k})}$$

$$(4)\ \widehat{EQM_{MC}} = \widehat{SD_{MC}}^{\,2} + \widehat{B_{MC}}^{\,2}$$

- On $\widehat{X_{MC}}$ és la mitjana.

- On $\widehat{B_{MC}}$ és el biaix i θ és el paràmetre poblacional que volem estimar amb els diferents estimadors.

- On $\widehat{SD_{MC}}$ és la desviació estàndard sense corregir (la funció *sd* de R utilitza la versió corregida ($(S-1)^{-1}$).

- On $\widehat{EQM_{MC}}$ és l'error quadràtic mig, que es pot expressar com la suma de la variància no corregida i el biaix al quadrat (es pot veure seguidament la demostració).

- On el subíndex MC expressa Montecarlo.

L'error quadràtic mig (EQM) per a un paràmetre θ es calcula com

$$
\begin{gathered}
EQM(\hat{\theta}) = E((\hat{\theta} - \theta)^2) \\
\text{(Sumem i restem } E(\hat{\theta})) \\
= E\left[(\hat{\theta} - E(\hat{\theta}) + E(\hat{\theta}) - \theta)^2\right] \\
\text{(Apliquem producte notable} \\
\text{i la propietat esperança d'una suma, és igual a suma d'esperances)} \\
= E\left[(\hat{\theta} - E(\hat{\theta}))^2\right] + 2(E(\hat{\theta}) - \theta)\underbrace{E(\hat{\theta} - E(\hat{\theta}))}_{E(\hat{\theta})-E(\hat{\theta})=0} + E[(E(\hat{\theta} - \theta))^2] \\
= E\left[(\hat{\theta} - E(\hat{\theta}))^2\right] + E[(E(\hat{\theta} - \theta))^2] \\
= Var(\hat{\theta}) + Biaix(\hat{\theta}, \theta)^2
\end{gathered}
$$

Estimadors robustos

Per a estudiar estimadors de localització, compararem estimadors clàssics amb els següents estimadors robustos: **la mediana i la mitjana α truncada.** Per a comparar estimadors de dispersió, compararem l'estimador clàssic amb l'estimador **robust desviació de la mediana absoluta (MAD).**

Mediana

Donada una mostra $\{xi : i = 1, ..., n\}$ i obtenint les observacions ordenades de x, $x_{(i:n)} = (x_{(1)}, x_{(2)}, ..., x_{(n)})$. La mediana M es defineix segons 2 condicions:

- Sí n és impar $\Rightarrow$ La M ocupa la posició $\frac{n+1}{2}$

  $\Rightarrow M_x = x_{(\frac{n+1}{2})}$

- Sí n és par $\Rightarrow$ La M és la mitjana dels 2 valors centrals. És a dir, les observacions que ocupen les posicions $(\frac{n}{2}) = a$ i $(\frac{n}{2} + 1) = b$

  $\Rightarrow M_x = x_{((a+b)/2)}$

Mitjana α truncada

Podem calcular la mitjana α truncada *tm* (α *trimmed mean*), com

$$tm = \frac{1}{n(1-2\alpha)} \sum_{\lceil i=\alpha n \rceil}^{\lfloor (1-\alpha)n \rfloor} x_{(i:n)}$$

On $x_{(i:n)}$ són les observacions ordenades de x i els intervals del sumatori són la part entera corresponent. Per a calcular-ho amb R podem utilitzar la funció *mean* amb l'opció *trim* que representa la fracció $\alpha \in (0, 0.5)$ d'observacions $n\alpha$ que s'extreuen de cada extrem del conjunt de dades. Fins al valor d'un enter ≥ 1 no es treurà cap observació. Això passarà quan la perturbació $t = (0.01, 0.05)$ en alguns casos, es podrà observar que el resultat de *tm* no presenta cap diferència amb l'estimador clàssic.

Desviació de la mediana absoluta (MAD)

Coneguda com MAD (*Median Absolute Deviation*), és un estimador robust de dispersió que calcula la mediana M de les desviacions en valor absolut entre les observacions i la M. Podem calcular-la com

$$MAD = K * M_x(|x_i - M_x|)$$

on K és una constant d'escala. R té com a referència la constant=1.4826, que aproximadament és $\Phi^{-1}(3/4)$ (amb R: qnorm $(0.75)^{-1}$). Amb aquesta constant i mida mostral gran, $E[(MAD)] = \sigma$ sí la distribució segueix una llei normal teòrica.

2.5 Casos a estudiar

Mixtura de distribucions normals

En aquest primer cas, estudiarem el model(2) en una mixtura de distribucions normals. El model $F_t \subset \mathcal{F}_t$ a considerar és el següent:

$$F_t = (1-t)F_\theta + (t)G = (1-t)\Phi + (t)N(2,1) \quad \text{On } \Phi = N(0,1).$$

La distribució F_θ segueix un model normal estàndard i G segueix un model normal amb un increment en el paràmetre de localització μ d'una unitat. El paràmetre σ es tractarà en un altre exemple, es considera conegut i pren valor 1. Cal trobar les mesures estadístiques definides comparant així els estimadors robustos definits anteriorment amb l'estimador màxim versemblant $\hat{\theta}_n$ del model normal, que es troba com a solució de:

Sigui $X_1, ..., X_n$ una mostra aleatòria que segueix una distribució Normal amb paràmetres desconeguts $\theta = (\mu, \sigma^2)$. Cal trobar l'estimador màxim versemblant $\hat{\theta_n}$ per a θ. Primer de tot, cal calcular la funció de versemblança

$$L(\theta|x_1,...,x_n) = \prod_{i=1}^{n} f_i(x|\theta = \{\mu, \sigma^2\}) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \, exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\right]$$

Prenem el seu logaritme, i obtenim

$$log(L(\theta|x_1,...,x_n)) = -\frac{n}{2}log(2\pi) - \frac{n}{2}log(\sigma^2)\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\right]$$

Cal optimitzar la funció de versemblança respecte a μ i σ^2, $on -\infty < \mu < \infty \; i \; \sigma^2 > 0$.

Cal derivar la funció log-versemblança respecte al paràmetre θ i igualar a zero l'equació. D'aquesta manera obtenim un sistema de dos equacions amb dues incògnites:

$$\begin{cases} \dfrac{\partial \, log(L(\theta|x_1,...,x_n))}{\partial\mu} = \dfrac{1}{\sigma^2}\displaystyle\sum_{i=1}^{n}(x_i-\mu) = 0 \\ \dfrac{\partial \, log(L(\theta|x_1,...,x_n))}{\partial\sigma^2} = -\dfrac{n}{2}\dfrac{1}{\sigma^2} + \dfrac{1}{2(\sigma^2)^2}\displaystyle\sum_{i=1}^{n}(x_i-\mu)^2 = 0 \end{cases}$$

Podem utilizar l'estimador màxim versemblant $\hat{\mu}_n = \bar{X}_n$ que es troba al resoldre la primera equació per a trobar el resultat de l'altre paràmetre. És a dir, substituir μ per $\bar{X}_n$. L'estimador $\hat{\theta_n}$ per al model normal és:

$$\begin{cases} \text{Per a } \mu, \; \hat{\mu_n} = \dfrac{1}{n}\sum_{i=1}^{n} X_i = \bar{X}_n \; : \text{(La mitjana mostral)} \\ \\ \text{Per a } \sigma^2, \; \hat{\sigma_n^2} = \dfrac{1}{n}\sum_{i=1}^{n}(X_i-\bar{X}_n)^2 \; : \text{(La variància mostral)} \end{cases}$$

A continuació es mostra un gràfic que compara el biaix obtingut en l'estimador màxim versemblant, $\hat{\mu_n}$, la mediana, M_x, i la mitjana α truncada, tm_α, respecte als diferents valors de perturbació t. També es mostren taules comparatives. El biaix de tm $(\alpha = 0.4)$ serà molt semblant al de M_x, degut a que la mediana seria el resultat de buscar una mitjana truncada en $\alpha = 0.5$.

Per a n=10, observem un comportament del biaix molt semblant en tm_α $(\alpha = 0.4)$ i M_x. El biaix en aquests dos estimadors és inferior respecte a $\hat{\mu_n}$. Tot i això, l'EQM de l'estimador clàssic és més baix, per tant, $\hat{\mu_n}$ és preferible en n=10.

Si augmentem la mida mostral a n=20, veiem com ara tm_α $(\alpha = 0.05)$ presenta un biaix inferior a $\hat{\mu_n}$ i en el gràfic la línia blava que representa el biaix de tm_α $(\alpha = 0.25)$ sembla que s'apropa més al biaix de tm_α $(\alpha = 0.4)$ i M_x (línies rosa i gris respectivament), que són encara els estimadors que presenten menys biaix. L'EQM dels estimadors robustos és ara inferior al de l'estimador clàssic amb preferència per la tm_α abans que per la M_x.

Per a n=50, amb qualsevol estimador diferent a $\hat{\mu_n}$ es guanya eficiència. És preferible tm_α $(\alpha = 0.4)$ respecte a M_x.

Mida mostral n=10

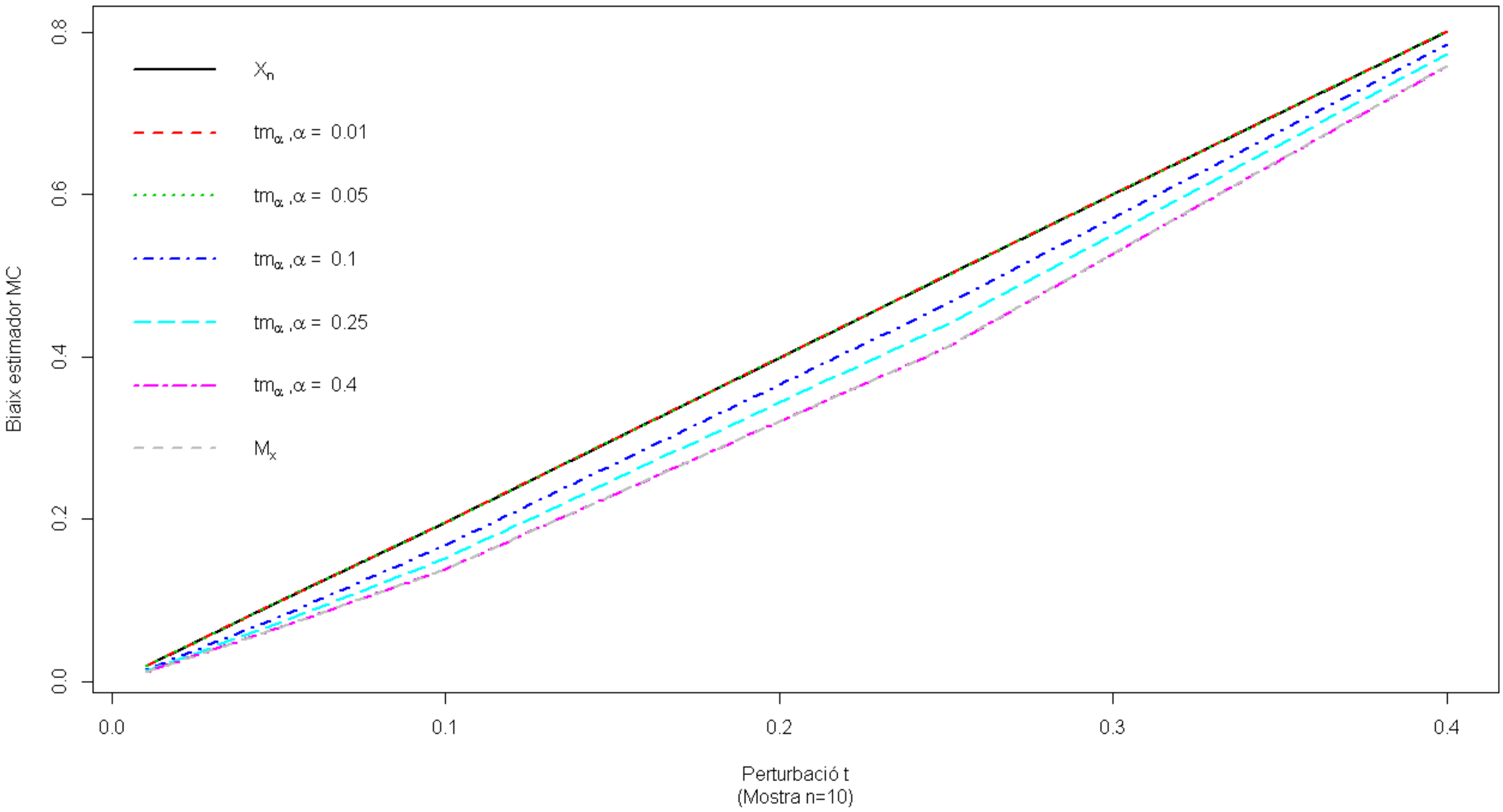

Figura 1: Comparació biaix estimadors MC, (n=10) $F_t = (1-t)\Phi + (t)N(2,1)$.

Perturbació t	X_n		M_x		$tm_{\alpha=0.01}$		$tm_{\alpha=0.05}$		$tm_{\alpha=0.1}$		$tm_{\alpha=0.25}$		$tm_{\alpha=0.4}$	
	Biaix	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM
t=0.01	0.018	0.104	0.011	0.143	0.018	0.104	0.018	0.104	0.014	0.108	0.013	0.117	0.011	0.143
t=0.05	0.098	0.129	0.065	0.160	0.098	0.129	0.098	0.129	0.080	0.128	0.072	0.133	0.065	0.160
t=0.1	0.196	0.175	0.138	0.193	0.196	0.175	0.196	0.175	0.167	0.166	0.152	0.168	0.138	0.193
t=0.25	0.500	0.428	0.412	0.421	0.500	0.428	0.500	0.428	0.465	0.409	0.440	0.402	0.412	0.421
t=0.4	0.801	0.838	0.758	0.898	0.801	0.838	0.801	0.838	0.786	0.839	0.773	0.847	0.758	0.898

Taula 1: Valors de les mesures (n=10). Model $F_t = (1-t)\Phi + (t)N(2,1)$

Mida mostral n=20

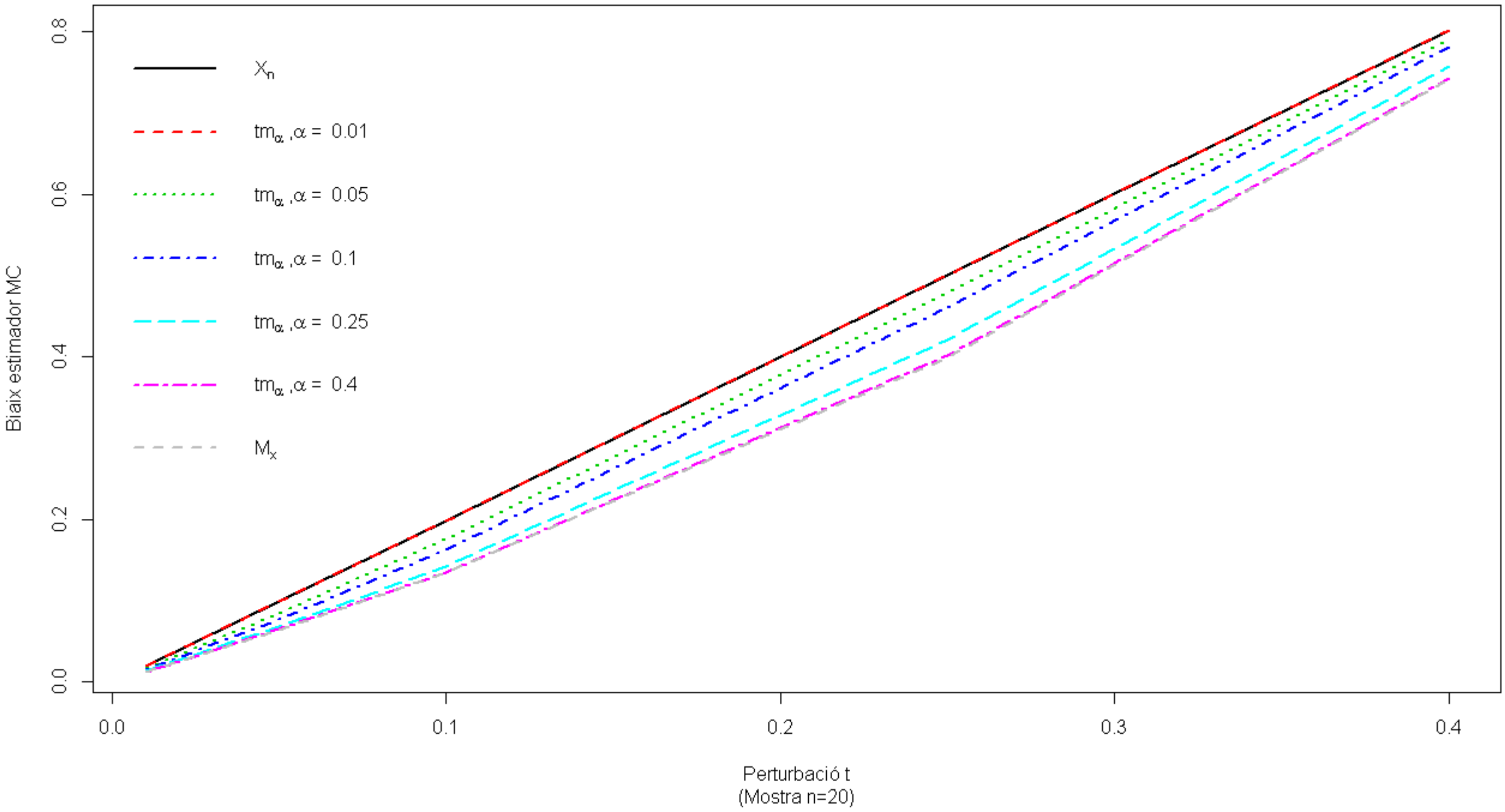

Figura 2: Comparació biaix estimadors MC, (n=20) $F_t = (1-t)\Phi + (t)N(2,1)$.

Perturbació t	$\bar{X}_n$		M_x		$tm_{\alpha=0.01}$		$tm_{\alpha=0.05}$		$tm_{\alpha=0.1}$		$tm_{\alpha=0.25}$		$tm_{\alpha=0.4}$	
	Biaix	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM
t=0.01	0.019	0.051	0.012	0.072	0.019	0.051	0.015	0.051	0.014	0.052	0.013	0.058	0.012	0.068
t=0.05	0.098	0.067	0.064	0.083	0.098	0.067	0.084	0.064	0.077	0.064	0.068	0.069	0.065	0.078
t=0.1	0.197	0.105	0.134	0.106	0.197	0.105	0.176	0.097	0.163	0.093	0.142	0.093	0.135	0.100
t=0.25	0.501	0.337	0.399	0.286	0.501	0.337	0.479	0.319	0.460	0.305	0.421	0.282	0.401	0.280
t=0.4	0.801	0.740	0.741	0.724	0.801	0.740	0.791	0.730	0.782	0.722	0.758	0.709	0.743	0.716

Taula 2: Valors de les mesures (n=20). Model $F_t = (1-t)\Phi + (t)N(2,1)$

Mida mostral n=50

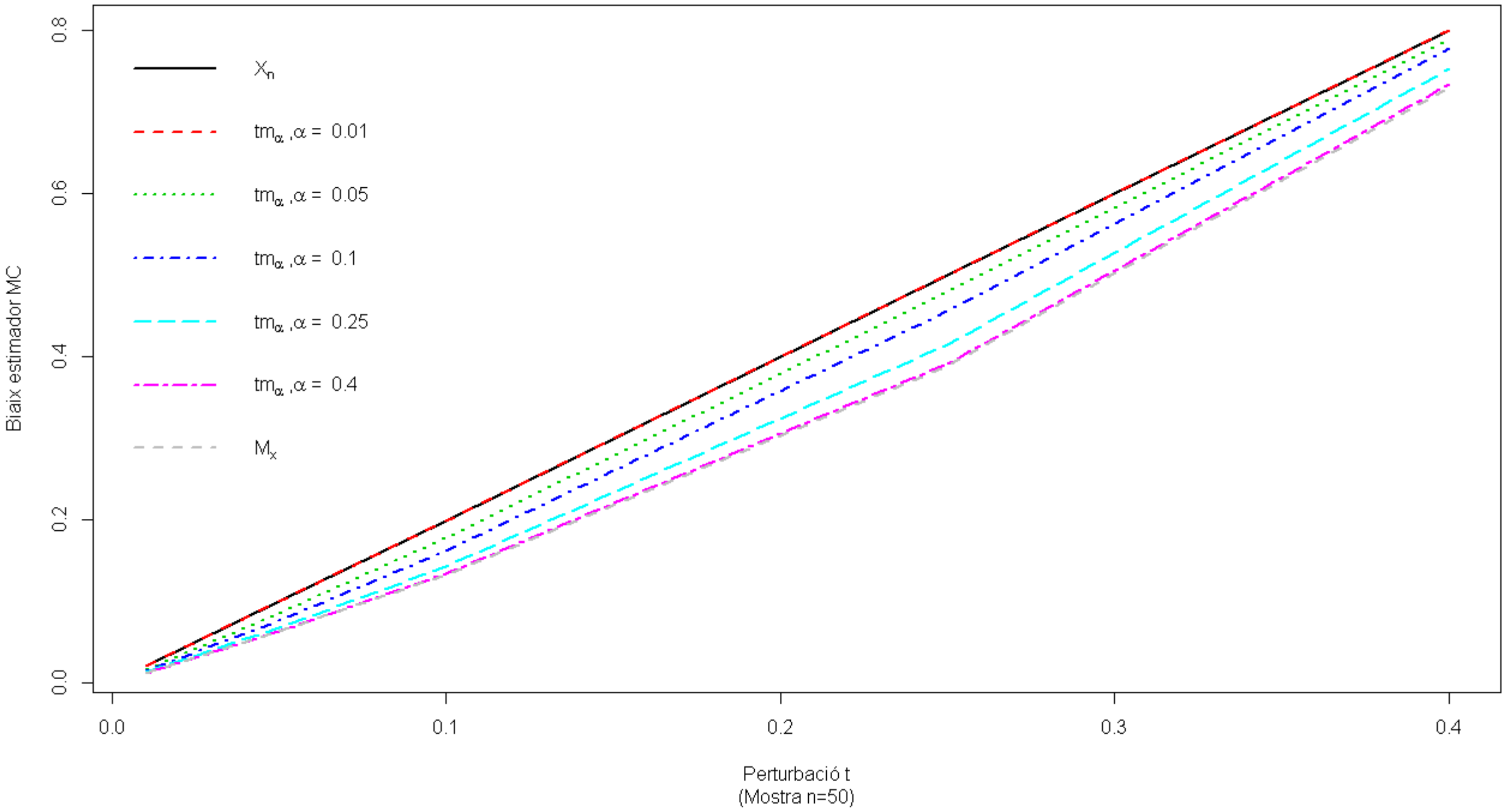

Figura 3: Comparació biaix estimadors MC, (n=50) $F_t = (1-t)\Phi + (t)N(2,1)$.

Perturbació t	$\bar{X}_n$		M_x		$tm_{\alpha=0.01}$		$tm_{\alpha=0.05}$		$tm_{\alpha=0.1}$		$tm_{\alpha=0.25}$		$tm_{\alpha=0.4}$	
	Biaix	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM
t=0.01	0.020	0.021	0.012	0.031	0.020	0.021	0.016	0.021	0.015	0.022	0.013	0.025	0.012	0.028
t=0.05	0.099	0.034	0.063	0.037	0.099	0.034	0.086	0.031	0.077	0.030	0.068	0.031	0.064	0.034
t=0.1	0.199	0.067	0.132	0.054	0.199	0.067	0.179	0.059	0.161	0.053	0.142	0.050	0.133	0.051
t=0.25	0.500	0.285	0.387	0.203	0.500	0.285	0.480	0.266	0.455	0.245	0.415	0.214	0.392	0.201
t=0.4	0.799	0.678	0.730	0.606	0.799	0.678	0.790	0.665	0.778	0.650	0.752	0.620	0.733	0.604

Taula 3: Valors de les mesures (n=50). Model $F_t = (1-t)\Phi + (t)N(2,1)$

En aquest segon cas, tornarem a estudiar el model(2) aplicat a una mixtura de distribucions normals. El model $F_t \subset \mathcal{F}_t$ a considerar és el següent:

$$F_t = (1-t)F_\theta + (t)G = (1-t)\Phi + (t)N(0,2) \quad \text{On } \Phi = N(0,1).$$

Ara, el model F_θ segueix un model normal estàndard i G segueix un model normal amb un increment en el paràmetre de dispersió σ d'una unitat. El paràmetre μ es considera conegut i pren valor 0. Cal comparar l'arrel de la variància mostral amb l'estimador robust MAD. Per a les simulacions es farà servir la constant $K = \frac{1}{\Phi(3/4)} \approx 1.482$ definida anteriorment. A continuació es mostren tres gràfics on es compara el biaix dels dos estimadors en cada mida mostral n. Podem observar que en tots els casos l'estimador robust presenta menys biaix respecte a l'estimador de màxima versemblança. Aquesta diferència s'accentua més a mesura que augmenta la perturbació i la mida mostral.

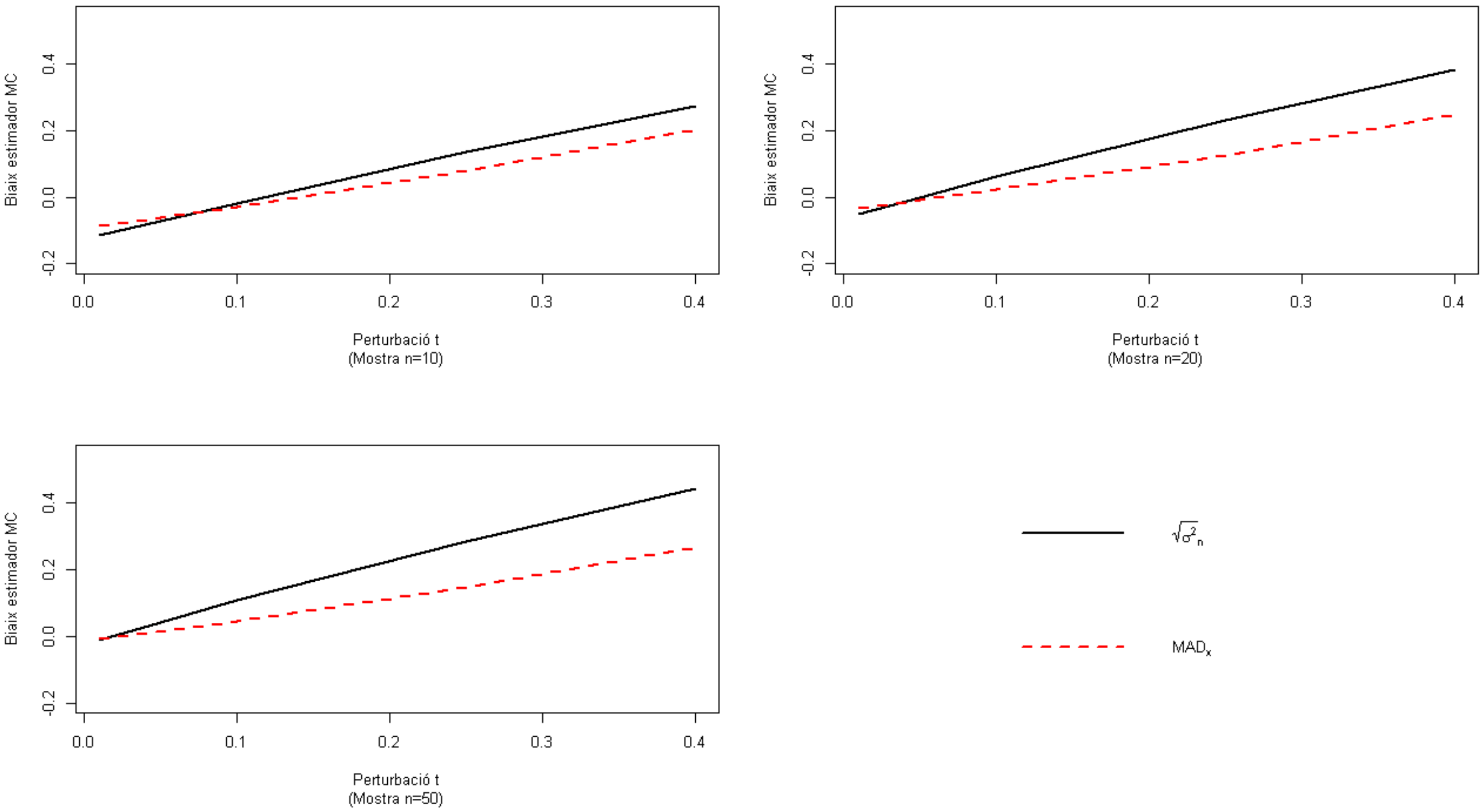

Figura 4: Comparació biaix estimadors MC, (n=10,20,50) $F_t = (1-t)\Phi + (t)N(0,2)$.

Per una banda, l'estimador MAD s'aproxima millor al valor de σ tot i que cal observar l'EQM per tenir en compte la variància. En excepció a la mida mostral n=10, on l'EQM és més baix en l'estimador clàssic, és preferible triar l'estimador robust en les altres mides mostrals, les diferències més rellevants es poden veure quan la perturbació t=(0.25,0.4). A la següent pàgina es poden veure els valors numèrics de l'EQM i el biaix.

Perturbació t	$\sqrt{\hat{\sigma_n^2}}$		MAD_x	
	Biaix	EQM	*Biaix*	EQM
t=0.01	-0.115	0.062	-0.086	0.122
t=0.05	-0.071	0.068	-0.061	0.128
t=0.1	-0.018	0.081	-0.030	0.135
t=0.25	0.136	0.138	0.078	0.182
t=0.4	0.274	0.220	0.203	0.271

Taula 4: Valors de les mesures (n=10). Model $F_t = (1-t)\Phi + (t)N(0,2)$

Perturbació t	$\sqrt{\hat{\sigma_n^2}}$		MAD_x	
	Biaix	EQM	*Biaix*	EQM
t=0.01	-0.051	0.028	-0.033	0.063
t=0.05	0	0.035	-0.010	0.066
t=0.1	0.061	0.050	0.021	0.072
t=0.25	0.230	0.121	0.125	0.106
t=0.4	0.381	0.226	0.249	0.177

Taula 5: Valors de les mesures (n=20). Model $F_t = (1-t)\Phi + (t)N(0,2)$

Perturbació t	$\sqrt{\hat{\sigma_n^2}}$		MAD_x	
	Biaix	EQM	*Biaix*	EQM
t=0.01	-0.011	0.011	-0.009	0.027
t=0.05	0.042	0.018	0.014	0.029
t=0.1	0.108	0.032	0.045	0.032
t=0.25	0.286	0.112	0.147	0.059
t=0.4	0.443	0.232	0.267	0.119

Taula 6: Valors de les mesures (n=50). Model $F_t = (1-t)\Phi + (t)N(0,2)$

Cas normal amb perturbació uniforme

En aquest cas, estudiarem el model(2) aplicat també a una distribució F_θ amb llei normal perturbada per una distribució G que segueix una llei uniforme, $U(3,4)$. El model $F_t \subset \mathcal{F}_t$ a considerar és el següent:

$$F_t = (1-t)F_\theta + (t)G = (1-t)\Phi + (t)U(3,4) \quad \text{On } \Phi = N(0,1).$$

La distribució F_θ segueix un model normal estàndard i G segueix una distribució uniforme en l'interval $(3,4)$. El paràmetre σ es considera conegut i pren valor 1. Cal trobar les mesures estadístiques definides comparant així els estimadors robustos definits anteriorment amb l'estimador màxim versemblant $\hat{\theta}_n$ sota el model normal, que és la mitjana mostral.
Un tret característic d'aquest escenari recau en comparar l'EQM. Per a mostres de mida $n = (10, 20)$ la mesura és favorable per a l'estimador clàssic quan la perturbació t és elevada (25 i 40 %). Tot i això, la M_x i la tm_α (amb $\alpha = 0.4$) són més eficients en mostres de mida n=50, $\forall$ valor de perturbació t. Si comparem els dos estimadors robustos, la mediana és preferible respecte a tm_α quan la distribució F_θ es troba molt perturbada, en $t = (0.25, 0.4)$ i alhora quan la mida mostral és gran (n=50). No es presenta el mateix comportament en mides mostrals n més petites.

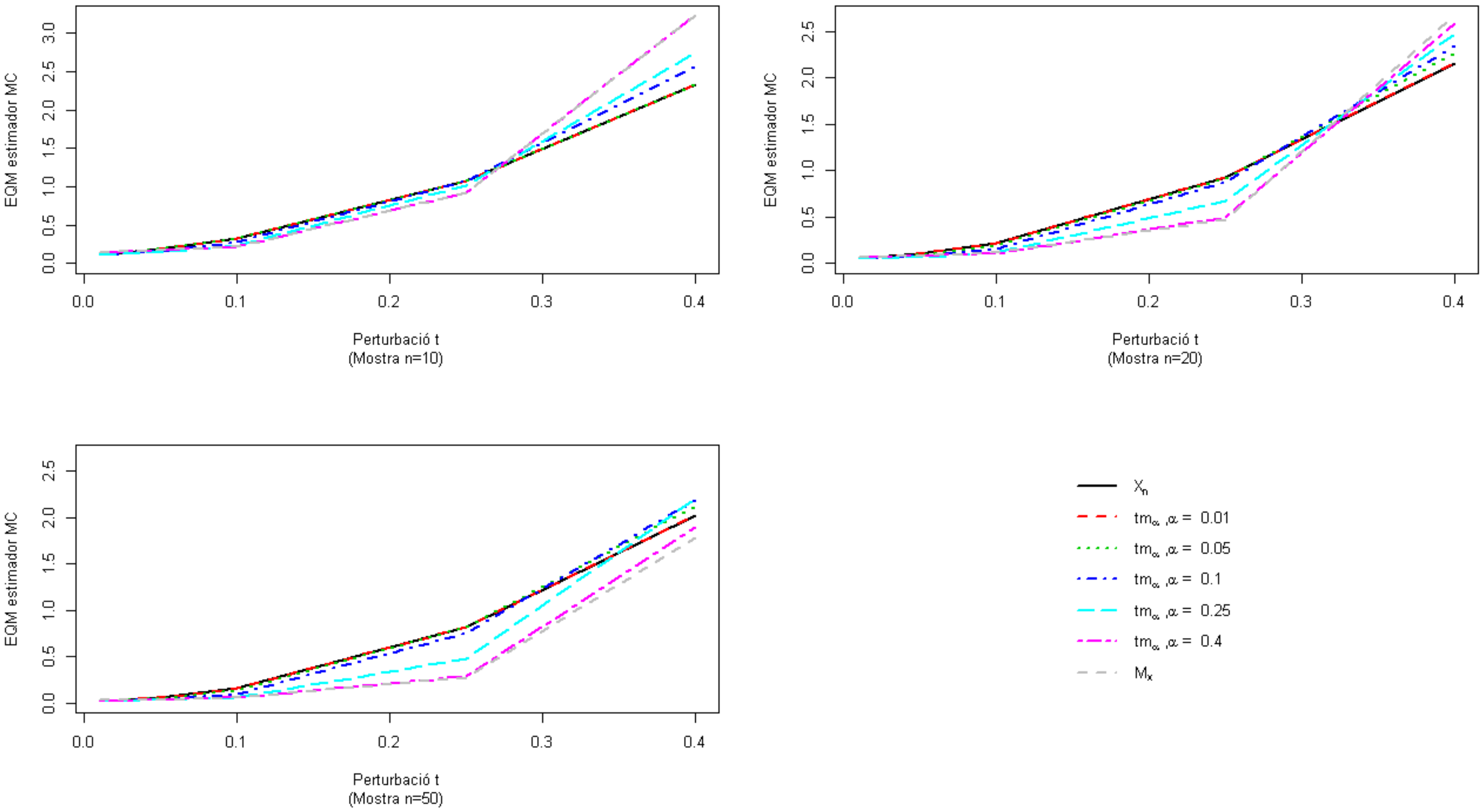

Figura 5: Comparació EQM estimadors MC, (n=10,20,50) $F_t = (1-t)\Phi + (t)U(3,4)$.

Perturbació t	$\hat{X}_n$		M_x		$tm_{\alpha=0.01}$		$tm_{\alpha=0.05}$		$tm_{\alpha=0.1}$		$tm_{\alpha=0.25}$		$tm_{\alpha=0.4}$	
	Biaix	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM
t=0.01	0.033	0.113	0.012	0.144	0.033	0.113	0.033	0.113	0.018	0.111	0.015	0.118	0.012	0.144
t=0.05	0.173	0.185	0.072	0.166	0.173	0.185	0.173	0.185	0.113	0.158	0.087	0.147	0.072	0.166
t=0.1	0.346	0.321	0.157	0.218	0.346	0.321	0.346	0.321	0.261	0.274	0.201	0.232	0.157	0.218
t=0.25	0.878	1.082	0.586	0.924	0.878	1.082	0.878	1.082	0.822	1.077	0.733	1.017	0.586	0.924
t=0.4	1.405	2.332	1.392	3.231	1.405	2.332	1.405	2.332	1.440	2.561	1.438	2.752	1.392	3.231

Taula 7: Valors de les mesures (n=10). Model $F_t = (1-t)\Phi + (t)U(3,4)$

Perturbació t	$\hat{X}_n$		M_x		$tm_{\alpha=0.01}$		$tm_{\alpha=0.05}$		$tm_{\alpha=0.1}$		$tm_{\alpha=0.25}$		$tm_{\alpha=0.4}$	
	Biaix	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM
t=0.01	0.034	0.055	0.013	0.072	0.034	0.055	0.021	0.053	0.017	0.053	0.014	0.059	0.013	0.068
t=0.05	0.174	0.105	0.068	0.085	0.174	0.105	0.131	0.090	0.102	0.079	0.075	0.073	0.069	0.080
t=0.1	0.347	0.220	0.146	0.115	0.347	0.220	0.296	0.192	0.245	0.161	0.166	0.111	0.148	0.110
t=0.25	0.878	0.923	0.492	0.474	0.878	0.923	0.860	0.917	0.821	0.877	0.642	0.668	0.510	0.491
t=0.4	1.403	2.148	1.304	2.669	1.403	2.148	1.434	2.265	1.447	2.341	1.422	2.472	1.328	2.582

Taula 8: Valors de les mesures (n=20). Model $F_t = (1-t)\Phi + (t)U(3,4)$

Perturbació t	$\hat{X}_n$		M_x		$tm_{\alpha=0.01}$		$tm_{\alpha=0.05}$		$tm_{\alpha=0.1}$		$tm_{\alpha=0.25}$		$tm_{\alpha=0.4}$	
	Biaix	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM	*Biaix*	EQM
t=0.01	0.034	0.024	0.012	0.031	0.034	0.024	0.021	0.022	0.017	0.022	0.014	0.025	0.013	0.029
t=0.05	0.174	0.061	0.067	0.038	0.174	0.061	0.132	0.048	0.094	0.036	0.074	0.033	0.068	0.036
t=0.1	0.348	0.161	0.142	0.059	0.348	0.161	0.305	0.136	0.231	0.094	0.160	0.058	0.144	0.056
t=0.25	0.874	0.825	0.447	0.271	0.874	0.825	0.865	0.818	0.817	0.751	0.613	0.482	0.462	0.284
t=0.4	1.398	2.026	1.144	1.782	1.398	2.026	1.428	2.121	1.446	2.190	1.420	2.198	1.232	1.900

Taula 9: Valors de les mesures (n=50). Model $F_t = (1-t)\Phi + (t)U(3,4)$

Cas uniforme amb perturbació exponencial

En aquest cas, estudiarem el model(2) aplicat a una distribució F_θ amb llei uniforme i una distribució G que segueix una llei exponencial, $Exp(1)$. El model $F_t \subset \mathcal{F}_t$ a considerar és el següent:

$$F_t = (1-t)F_\theta + (t)G = (1-t)U(0,1) + (t)Exp(1).$$

La distribució F_θ segueix una distribució contínua uniforme en l'interval (0,1) i G segueix una distribució exponencial amb paràmetre $\lambda = 1$ (escriure en R: *?rexp* per veure parametrització que utilitza R). Primer de tot, cal calcular l'estimador màxim versemblant de la distribució F_θ.
Sigui $X_1, ..., X_n$ una mostra aleatòria que segueix una distribució contínua uniforme en l'interval $[0, \theta]$, cal trobar l'estimador $\hat{\theta}_n$. Primer de tot, cal calcular la funció de versemblança per a θ, que és el productori de totes les densitats marginals f_i que provenen de les dades observades en θ:

$$L(\theta|x_1, ..., x_n) = \begin{cases} \dfrac{1}{\theta^n} & 0 \leq x_i \leq \theta \;\; (i = 1, ..., n) \\ 0 & altrament \end{cases}$$

Prenem el logaritme de $L(\theta|x_1, ..., x_n)$

$$log(L(\theta|x_1, ..., x_n)) = \begin{cases} -n\, log(\theta) & 0 \leq x_i \leq \theta \;\; (i = 1, ..., n) \\ 0 & altrament \end{cases}$$

De l'equació anterior, es pot veure que l'estimador $\hat{\theta}_n$ per θ ha de ser un valor θ que compleixi $0 \leq x_i \leq \theta \; \forall \; i = 1, ..., n$. La funció anterior és monòtona decreixent en θ, necessitem obtenir el valor més petit possible de θ tal que $\theta \geq x_i \; \forall \; i = 1, ..., n$ en l'ordre de maximitzar $log(L(\theta|x_1, ..., x_n))$. Aquest valor és θ=màxim$(x_1, ..., x_n) = X_{(n)}$.

$$\rightarrow \hat{\theta_n} = \text{màxim } (x_1, ..., x_n) = X_{(n)}.$$

El que volem és comparar un paràmetre de localització clàssic respecte a M_x i tm_α. L'esperança d'una distribució uniforme entre dos valors a i b és la mitjana d'aquests dos valors. Per tant, tal i com hem definit la nostra distribució F_θ l'esperança és:

$$E(X) = \frac{a+b}{2} = \frac{0+\theta}{2} = \frac{0+1}{2} = \frac{1}{2}$$

Podem substituir el valor θ per l'estimador $\hat{\theta_n}$. Com a estimador clàssic utilitzarem l'estimador $\hat{\theta_n}$ dividit 2, és a dir: $\dfrac{X_{(n)}}{2}$.

Si observem el primer gràfic, podem veure que M_x i tm_α presenten valors més baixos a qualsevol mida mostral i perturbació t respecte a l'estimador clàssic definit. Si comparem l'EQM entre els dos estimadors robustos (on tm ($\alpha = 0.01$) és la mitjana mostral en $n = (10, 20, 50)$) veiem que a mesura que augmenta t l'estimador més eficient és tm ($\alpha = 0.25$) (línia blava). En aquest cas, la mediana no és preferible en cap situació.

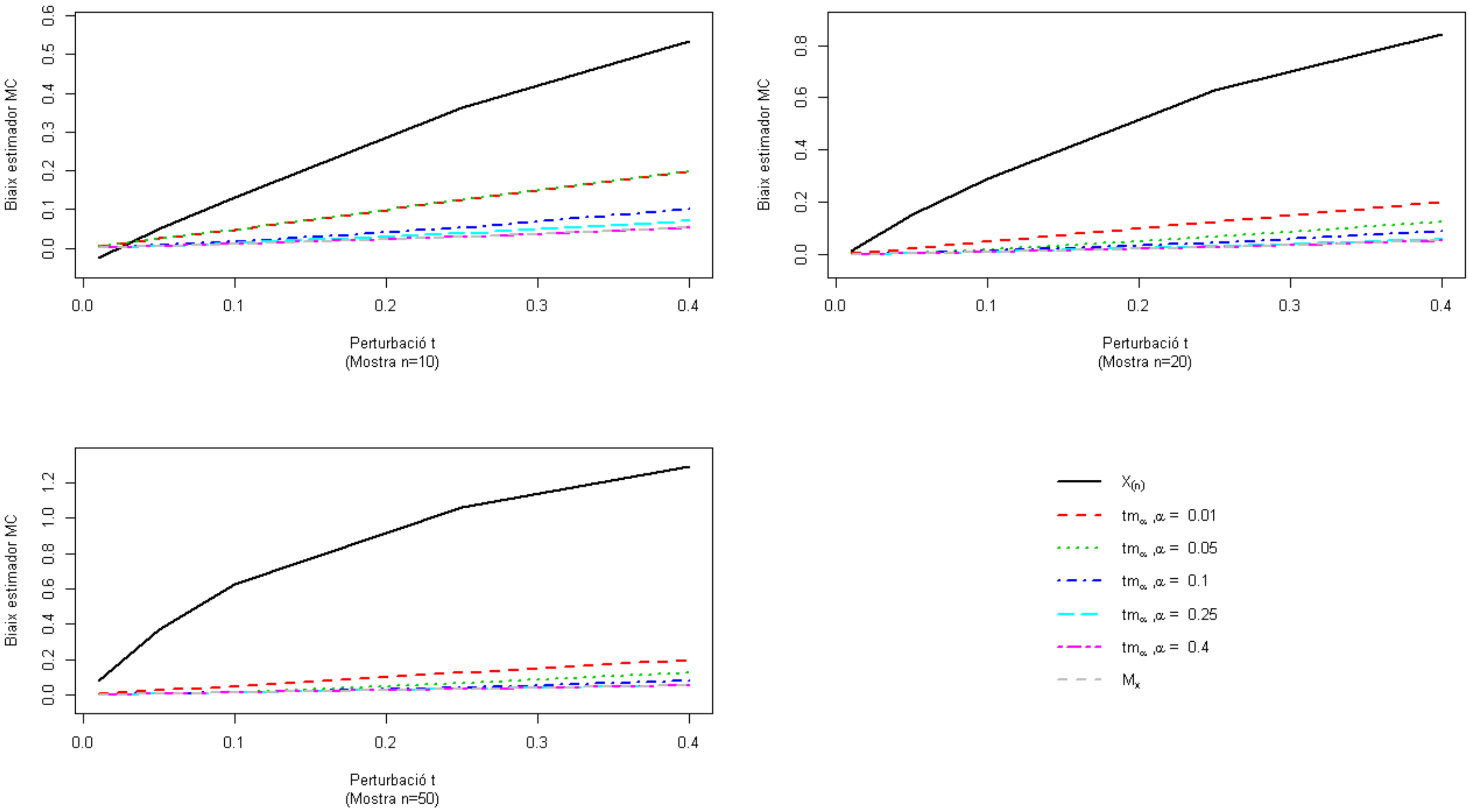

Figura 6: Comparació Biaix estimadors MC, (n=10,20,50) $F_t = (1-t)U(0,1) + (t)Exp(1)$.

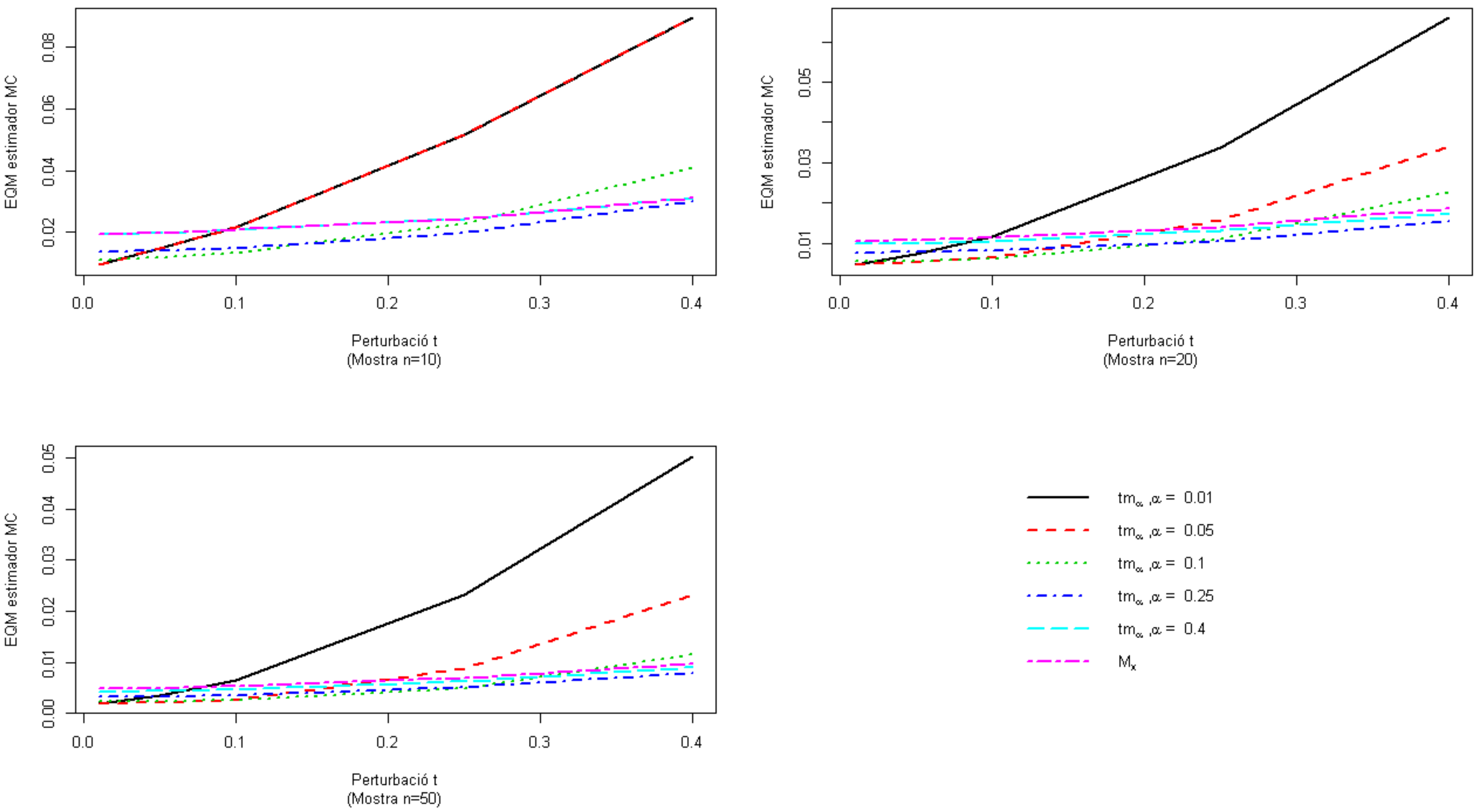

Figura 7: Comparació EQM estimadors MC (M_x i tm_α), (n=10,20,50) $F_t = (1-t)U(0,1) + (t)Exp(1)$.

3 Mesures de robustesa

En la teoria inferencial clàssica, un s'adhereix estrictament a models paramètrics. En la part de simulació, hem pogut veure com el model normal és sensible a petites desviacions que fan que els estimadors $\hat{\theta_n}$ deixin de tenir bones propietats. En la teoria robusta s'entén que un model paramètric, $\{F_\theta;\ \theta \in \Theta\}$ és una abstracció matemàtica que només és una aproximació idealitzada de la realitat (Hampel et al. [1986]). Cal construir procediments estadístics que es comportin bastant bé sota desviacions d'aquest model assumit. Per tant, no només cal considerar la distribució dels estimadors sota el model paramètric, sinó també sota altres distribucions de probabilitat. Aquesta idea es pot plasmar amb el plantejament del model (2).

Un teorema molt conegut en l'estadística, és el teorema de Glivenko-Cantelli, que ens diu el següent:

Sigui $X_n; n \geq 1$ una successió de variables aleatòries independents i idènticament distribuïdes definides en l'espai de probabilitat $(\Omega, \mathcal{A}, \mathcal{P})$ amb funció de distribució comuna F. Podem definir F_n com la funció de distribució empírica obtinguda de les n primeres variables aleatòries $X_1, ..., X_n$. Llavors,

$$\sup_{x\in\mathbb{R}} |F_n(x) - F(x)| \to 0 \;\; \text{casi segur.}$$

El càlcul d'algunes estimacions no depèn de l'ordre en el qual es comptabilitzen les observacions, és a dir, les observacions no canvien el seu valor si apliquem qualsevol permutació arbritària a les dades. Per tant, podem escriure l'estimador T_n com a

$$T_n = T(x_1, ..., x_n) = T(x_{(1)}, ..., x_{(n)})$$

On $x_{(1)} \leq x_{(2)} \leq ... \leq x_{(n)}$ (estadístics ordenats.)

Amb la idea que hi ha al darrere del teorema de Glivenko-Cantelli podem suposar ara que l'estimador només depèn de les dades a través de la distribució empírica, F_n, que podem definir com

$$F_n(t) = n^{-1} \sum I(x_i \leq t)$$

On

$$I(x_i \leq t) = \begin{cases} 1 & \text{per a} \quad x_i \leq t \\ 1 & \text{per a} \quad x_i > t \end{cases}$$

Podem entendre, per tant, l'estimador T_n com el funcional de la distribució empírica, escrivint

$$T_n = T(F_n)$$

S'utilitza la paraula funcional i no funció per indicar que el domini de T no és més gran que un subconjunt d'$\mathbb{R}^n$ però sí és un subconjunt de funcions (com ara la distribució empírica F_n en aquest cas). Es suposa que el funcional $T(F_n)$ té una extensió natural en $T(F)$, on F és un conjunt de distribucions que contenen a la família F_t. Per una banda, el funcional $T(F)$ és consistent en el sentit de Fisher:

$$T(F_\theta) = \theta \quad, \forall \theta \in \Theta$$

i $T(F_n)$ és consistent (aplicar el teorema Glivenko-Cantelli al funcional).

3.1 Aproximació lineal d'un funcional

Si suposem que una funció real $f(x)$ és diferenciable en x_0. Llavors

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x - x_0) + o(x - x_0) \\ &= f(x_0) + L(x) + o(x - x_0) \end{aligned}$$

$$\text{On } \lim_{x \to x_o} \frac{o(x - x_0)}{x - x_0} = 0$$

o representa un error i $L(x)$ és una 'funció lineal aproximada' que satisfà $L(x_0) = 0$. Podem definir ara el concepte d'aproximació lineal per a un funcional. Cal introduir el concepte de funcional lineal i el de distància entre dues funcions de distribució.

El funcional T(F) és lineal sí

$$T[(1 - \alpha)F_1 + \alpha F_2] = (1 - \alpha)T(F_1) + \alpha T(F_2). \ \forall 0 \leq \alpha \leq 1.$$

El concepte de distància $d(F_1, F_2)$ entre dues funcions de distribució F_1 i F_2 es pot definir de diferents maneres. Per exemple, es pot utilitzar la distància de variació total o de Kolmogorov

$$d(F_1, F_2) = \sup_x |F_1(x) - F_2(x)|$$

Un altra manera és buscar la distància de Lévy

$$d(F_1, F_2) = \sup_g \left| \int_{\mathbb{R}} g(x) d[F_1 - F_2](x) \right|$$

On el suprem és el conjunt de totes les funcions contínues amb suport finit i $\sup_x |g(x)| \leq 1$. Cal introduir també el concepte de derivada d'un funcional. El funcional T(F) és 'diferenciable' en F_0 sí existeix una funció lineal contínua $L(F)$ tal que

$$T(F) = T(F_0) + L(F) + o[d(F, F_0)]$$

amb $\dfrac{o[d(F, F_0)]}{d(F, F_0)} \to 0 \quad \text{com} \quad d(F, F_0) \to 0.$

L'aproximació del funcional a partir d'una funció lineal contínua es pot expressar com

$$L(F) = \int a(x)dF(x).$$

Això no deixa de ser una representació integral d'un funcional lineal (*per més informació, teorema de la representació de Riesz*). Alguns exemples de funcionals amb aquesta representació són

- La mitjana: $T(F) = \int x \, dF(x)$
- El moment r-èssim: sigui r un enter, $T(F) = \int x^r \, dF(x)$
- La variància: $T(F) = \int x^2 \, dF(x) - (\int x \, dF(x))^2$

La funció $a(\cdot)$ normalment s'anomena nucli (*kernel*) del funcional lineal L(F), $\forall 0 \leq \alpha \leq 1$.

$$L[(1-\alpha)F_1 + \alpha F_2] = \int a(x)d[(1-\alpha)F_1 + \alpha F_2](x)$$

$$= (1-\alpha)\int a(x)dF_1(x) + \alpha \int a(x)d_F 2(x)$$

$$= (1-\alpha)L(F_1) + \alpha L(F_2)$$

$L(F) = T(F) - T(F_0) - o[d(F, F_0)]$, i $o[d(F_0, F_0)] = 0$. El promig que la funció nucli $a(\cdot)$ segueixi F_0 és

$$L(F_0) = \int a(x)dF_0(x) = 0$$

3.1.1 Derivada Gateaux

És equivalent a 'derivades direccionals'. Per a fixar idees, suposem que la distància $d(F, F_0)$ és la distància de variació total

$$d(F, F_0) = \sup_x |F(x) - F_0(x)|$$

i $F_{t,G} = (1-t)F_0 + tG$.

En aquest cas, la distància és

$$d(F_{t,G}, F_0) = d[((1-t)F_0 + tG), F_0]$$

$$= \sup_x |(1-t)F_0(x) + tG(x) - F_0(x)|$$

$$= \sup_x |t[G(y - F_0(y))]|$$

$$= |t| \sup_x |G(x) - F_0(x)|$$

El funcional $T(F)$ s'anomena diferenciable Gateaux en F_0 en la direcció de G sí

$$T(F_{t,G}) = T(F_0) + \int a(x)dF_{t,G}(x) + o(t).$$

Notar que,

$$\int a(x)dF_{t,G}(x) = (1-t)\int a(x)dF_0(x) + t\int a(x)dG(x) = t\int a(x)dG(x).$$

Que es pot entredre com una aproximació de la perturbació en l'estimació.

3.2 La funció d'influència (IF)

Sí T(F) és diferenciable Gateaux en F_0 en la direcció de G llavors,

$$T(F_{t,G}) = T(F_0) + t\int a(x)dG(x) + o(t).$$

i llavors

$$\lim_{t\to 0} \frac{T(F_{t,G}) - T(F_0)}{t} = \int a(x)dG(x) + \lim_{t\to 0}\frac{o(t)}{t} = \int a(x)dG(x)$$

Agafant $G = \Delta_x$, una distribució amb probabilitat 1 en el punt x i 0 altrament, tenim

$$\int a(x)d\Delta_x(x) = a(x).$$

El que ens interessa trobar és la funció nucli en el punt x, que es pot trobar calculant el límit

$$\lim_{t\to 0} \frac{T((1-t)F_0 + t\Delta_x) - T(F_0)}{t} = a(x).$$

Sí el límit anterior existeix, s'anomena funció d'influència de T en x i F_0, normalment escrita com $IF(x, T, F_0)$ (veure més detalls a Hampel et al. [1986]).

Podem definir la funció d'influència com

$$IF(x, F, T) = \lim_{t \to 0} \frac{T((1-t)F + t\Delta_x) - T(F)}{t}.$$

La importància de la funció d'influència recau en la seva interpretació: descriu l'efecte d'una contaminació molt petita (d'aquí el mot infinitesimal) en el funcional $T(F_0)$. Una característica que es considera en els estimadors robustos és que IF sigui acotada. Els estimadors amb aquesta característica se'ls anomena B-robust. Tot i això, la influència limitada no és una condició necessària ni suficient de robustesa: existeixen estimadors robustos amb influència ilimitada i estimadors no robustos amb influència limitada. Els estimadors MM, proposats per Yohai, són un exemple dels primers.

Per a derivar la funció d'influència d'una estimació T_n en una distribució F en el punt x, es poden seguir els següents pasos:

1) Derivar la representació funcional $T_n = T(F_n)$.

2) Derivar el funcional T(F), canviant F_n per F(consistència)

3) Mesurar $g(t) = T((1-t)F + tG) = T((1-t)F + t\Delta_x)$

4) Calcular $\frac{\mathrm{d}}{\mathrm{d}t} g(t) = \frac{\mathrm{d}}{\mathrm{d}t} T((1-t)F + t\Delta_x) = h(t)$

5) Aplicar en $IF(x, F, T) = h(0) = \left.\frac{\mathrm{d}}{\mathrm{d}t} T((1-t)F + t\Delta_x)\right|_{t=0} = \left.\frac{\mathrm{d}}{\mathrm{d}t} T(F_t)\right|_{t=0}$

6) Amb $F_t = (1-t)F + tG = (1-t)F + t\Delta_x$.

En totes les següents integrals, $\int = \int_{\mathbb{R}}$. Tots els càlculs s'integraran en el conjunt $\mathbb{R}$ tot i que es podria integrar en altres espais, com ara un espai de funcions. Els aspectes tècnics mencionats en aquest apartat es poden veure en tema 2 de Huber [1981].

3.2.1 Exemples IF

Per a mostrar un primer exemple, considerem el funcional de la mitjana, $T(F) = \int x\, dF(x) = \mu$. On F és una funció de distribució contínua arbritària. Primer de tot calculem l'esperança de $T((1-t)F + t\Delta_x)$.

$$E[T((1-t)F + t\Delta_x)] = \int x\; d\left\{(1-t)F + t\Delta_x)\right\}$$

$$= (1-t)\int x\; dF(x) + t\int x\; d\Delta_x(x) =$$

$$= (1-t)\mu + tx$$

Substituïm en IF(x;μ,F):

$$IF(x;\mu,F) = \lim_{t\to 0}\frac{(1-t)\mu + tx - T(F)}{t}$$

$$= \lim_{t\to 0}\frac{(1-t)\mu + tx - \mu}{t} = \lim_{t\to 0}\frac{\mu - t\mu + tx - \mu}{t} = x - \mu$$

Per a μ, la funció d'influència no està acotada. Una sola observació pot arribar a influir molt en l'estimació de la mitjana. No es considera un estimador B-robust.

Considerem ara el funcional de la variància, que és pot expressar com

$$(E(X^2) - E(X)^2 = E[(X - E(X)]^2)$$

.

El funcional és defineix com

$$T(F) = \int x^2 \, dF(x) - (\int x \, dF(x))^2 = \int (x - \mu(F))^2 dF(x)$$

Considerem $F_t = (1-t)F + tG$, cal calcular

$$\frac{\mathrm{d}}{\mathrm{d}t} T(F_t)\Big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathbb{R}} (x - \mu(F_t))^2 \, dF_t(x)\Big|_{t=0}$$

$$= \left\{ \int 2(x - \mu(F_t))(-1) \, \frac{\mathrm{d}}{\mathrm{d}t} \underbrace{\mu(F_t)}_{\mu(F_t) = E_G(x) - E_F(x)} dF_t + \int (x - \mu(F_t))^2 \, d(G - F) \Big|_{t=0} \right\}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \mu(F_t) = \frac{\mathrm{d}}{\mathrm{d}t} \int x d \underbrace{((1-t)F + tG)}_{F_t} = \int_x d(G - F) = \int x dG - \underbrace{\int x dF}_{\mu(F)} = \underbrace{\int (x - \mu(F)) dG}_{\int \mu(F) dG = \mu(F)}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} T(F_t)\Big|_{t=0} = \left\{ -2 \int (x - \mu(F_t)) \left\{ \int (x - \underbrace{\mu(F)}_{=\mu}) \, dG \right\} dF_t + \int (x - \mu(F_t))^2 d(G - F) \Big|_{t=0} \right\}$$

$$= -2 \left\{ \int (x - \mu) \underbrace{\int [(x - \mu) \, dG] \, dF}_{\int (x - \mu) dF(x) = 0} \right\} + \int (x - \mu)^2 d(G - F)$$

$$= 0 + \int (x - \mu)^2 dG - \sigma^2$$

La IF per a la variància és $IC(x; T, F) = (x - \mu(F))^2 - \sigma_F^2$. La variància és un estimador amb IF no acotada, per tant no desitjable en el sentit de buscar estimadors amb la propietat B-robust.

Considerem ara el funcional de la mediana, $T(F) = F^{-1}(1/2)$, i suposem que F té densitat f positiva en $F^{-1}(1/2)$ (pàg 56. Huber [1981]).

Amb $F_t = (1-t)F + tG$, cal calcular

$$\frac{\mathrm{d}}{\mathrm{d}t}T(F_t)\bigg|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t}(F_t)^{-1}(1/2)\bigg|_{t=0}$$

Cal veure que $F_t((F_t)^{-1}(1/2)) = 1/2$ que és el valor que volem trobar. Per tant

$$0 = \frac{\mathrm{d}}{\mathrm{d}t}(F_t(F_t)^{-1}(1/2))\bigg|_{t=0}$$

$$= \frac{\mathrm{d}}{\mathrm{d}t}\left\{F_t(F_t^{-1}(1/2)) + t(G-F)(F_t^{-1}(1/2))\right\}\bigg|_{t=0}$$

$$= f(F^{-1}(1/2))\dot{T}(F; G-F) + (G-F)(F^{-1}(1/2)) + 0.$$

Diem a $F_t(F_t^{-1}(1/2)) = F_t(a)$, es fa la següent aproximació:

$$F_t(a) = F_0(a) + \frac{\mathrm{d}}{\mathrm{d}t}F_t(a)\bigg|_{t=0} t + o(t^2).$$

$$\text{On } F_t(a) = (1-t)F(a) + tG(a) \text{ i } \frac{\mathrm{d}}{\mathrm{d}t}F_t(a) = -F(a) + G(a).$$

On $\dot{T}(F; G-F)$ juga el paper de la funció lineal aproximada $L(\cdot)$ en t=0 i és igual a

$$\dot{T}(F; G-F) = -\frac{(G-F)(F^{-1}(1/2))}{f(F^{-1}(1/2))}$$

$$= -\frac{\int (I_{(-\infty, F^{-1}(1/2))}(x) - (1/2))dG(x)}{f(F^{-1}(1/2))} = \frac{\int_{-\infty}^{F^{-1}(1/2)=M_x} dG(x) - 1/2}{f(F^{-1}(1/2))}$$

La funció d'influència de la mediana es troba acotada, per tant, té la propietat B-robust.

$$\text{La seva } IF(x, T, F) = \frac{1}{f(F^{-1}(1/2))}\left\{I_{(-\infty, F^{-1}(1/2)(x)} - 1/2\right\}$$

Si volem aconseguir l'estimació d'un p-quantil, $T(F) = F^{-1}(p)$, el càlcul és semblant al de la mediana, cal canviar el valor $p = (1/2)$ per un altre p. Per tant, el resultat de $T(\cdot)$ serà

$$\dot{T}(F; G-F) = -\frac{(G-F)(F^{-1}(p))}{f(F^{-1}(p))}$$

Considerem ara el funcional de la mitjana α truncada.

$$T(F) = \frac{1}{1-2\alpha} \int_{\alpha}^{1-\alpha} F^{-1}(x)dx$$

Calculem la IF derivant respecte a t el funcional $T(F_t)$ en el punt t=0.

$$\frac{\mathrm{d}}{\mathrm{d}t} T(F_t)\bigg|_{t=0} = \frac{1}{1-2\alpha} \int_{\alpha}^{1-\alpha} \frac{\mathrm{d}}{\mathrm{d}t} F_t^{-1}(x)\bigg|_{t=0} dx$$

$$\text{On } \frac{\mathrm{d}}{\mathrm{d}t} F_t^{-1}(x)\bigg|_{t=0} = -\frac{(G-F)(F^{-1}(x))}{f((F^{-1}(x))}$$

$$= \frac{\mathrm{d}}{\mathrm{d}t} T(F_t)\bigg|_{t=0} = \frac{1}{1-2\alpha} \int_{\alpha}^{1-\alpha} \left(-\frac{(G-F)(F^{-1}(x))}{f((F^{-1}(x))} \right) dx$$

Realitzem un canvi de variable i també una assignació als intervals del funcional

$$\begin{cases} y = F^{-1}(x) & F(y) = x \\ \\ dx = F'(y) = f(y)dy & \\ \hline \\ 1-\alpha = F(b) & b = F^{-1}(1-\alpha) \\ \alpha = F(a) & a = F^{-1}(\alpha) \end{cases}$$

Ara tenim la següent expressió:

$$-\frac{1}{1-2\alpha} \int_{a=F^{-1}(\alpha)}^{b=F^{-1}(1-\alpha)} \frac{(G-F)\ (y)}{f(y)} f(y)dy$$

Simplifiquem f(y)

$$= -\frac{1}{1-2\alpha} \int_a^b (G-F)(y)dy = -\frac{1}{1-2\alpha} \left\{ \int_a^b G(y)dy - \int_a^b F(y)dy \right\}$$

$$= -\frac{1}{1-2\alpha} \int_a^b F(y)dy - \int_a^b G(y)dy$$

Ara:

$$= \int_a^b F(y)dy = \left\{ [yF(y)]_a^b - \int_a^b yf(y)dy \right\}$$

Integracio per parts: Integrem 1, derivem F. Si dividim per $\dfrac{1}{1-2\alpha}$ tenim:

$$= \frac{1}{1-2\alpha}\int_a^b F(y)dy = \frac{1}{1-2\alpha}(b\underbrace{F(b)}_{1-\alpha} - a\underbrace{F(a)}_{\alpha} - tm$$

$$= \frac{1}{1-2\alpha}(b(1-\alpha) - a\alpha) - tm$$

Per tant:

$$\frac{1}{1-2\alpha}(b(1-\alpha) - a\alpha) - tm - \frac{1}{1-2\alpha}\int_a^b G(y)dy$$

$$\text{Ara } G(y) = \begin{cases} 0 & y < s \\ 1 & y \geq s \end{cases}$$

Definida la funció indicadora en s (Δ_x), llavors:

$$\int_a^b G(y) = \begin{cases} 0 & s < a \\ s-a & s \in [a,b] \\ b-a & s > b \end{cases}$$

El resultat de la funció d'influència, varia segons s i és igual a:

$$\frac{1}{1-2\alpha}(b(1-\alpha) - a\alpha) - tm - \frac{1}{1-2\alpha}\begin{cases} 0 \ \ s < a \\ s-a \ \ s \in [a,b] \\ b-a \ \ s > b \end{cases}$$

Per a $s \in [a, b]$

$$\frac{1}{1-2\alpha}(b - b\alpha - a\alpha + a - s) - tm$$

$$= \frac{-\alpha(b+a)}{1-2\alpha} + \frac{b+a}{1-2\alpha} - \frac{s}{1-2\alpha} - tm$$

Per a $s < a$

$$\frac{-\alpha(b+a)}{1-2\alpha} + \frac{b}{1-2\alpha} - tm$$

Per a $s > b$

$$\frac{-\alpha(b+a)}{1-2\alpha} + \frac{a}{1-2\alpha} - tm$$

La IF de la mitjana truncada depèn del valor s però es troba acotada degut a que trunquem la mitjana clàssica en una proporció α, per tant té la propietat B-robust.

3.3 Mesures que deriven de la funció d'influència

3.3.1 Gross-error sensitivity

A partir de la informació que ens dóna la funció d'influència, podem obtenir altres mesures de robustesa. El primer i més important és el suprem del valor absolut d'IF. Com hem vist, IF descriu l'efecte d'una contaminació infinitesimal en el punt x en l'estimador. Per tant, podem definir *gross-error sensitivity*(pàg 87 Hampel et al. [1986]) de T an F com

$$\gamma^* = \sup_x |IF(x; T, F)|$$

El *gross-error sensitivity* mesura aproximadament la pitjor influència que una petita quantitat de contaminació fixada té en el valor de l'estimador. És desitjable que γ^* sigui finit, en aquest cas es pot dir que el funcional T és *B-robust* en F.

3.3.2 Local-shift sensitivity

Una altra mesura té a veure amb petites fluctuacions en les observacions. Quan alguns valors canvien lleugerament, com succeeix en l'arrodoniment i petites imprecisions en les dades. Intuïtivament, l'efecte de desplaçar lleugerament una observació des del punt x a un punt proper y es pot mesurar com a $IF(y; T, F) - IF(x; T, F)$, ja que una observació és afegida en y i una altra és eliminada en x. Si normalitzem aquesta diferència podem definir el pitjor efecte de *'wiggling'(bellugar,desplaçar)* com

$$\lambda^* = \sup_{x \neq y} \frac{|IF(y; T, F) - IF(x; T, F)|}{|y - x|}$$

3.3.3 Rejection point

Aquesta mesura reflecteix la idea de rebutjar els valors extrems en la seva totalitat. En relació amb la funció d'influència, es pot interpretar com que la IF s'esvaeix directament d'una àrea determinada. En el cas que la funció de distribució F sigui simètrica (i posant com a centre de la seva simetria el valor 0), això permet definir el punt de rebuig com

$$\rho^* = \inf_{r>0} \{IF(x; T, F) = 0 \; \forall \, |x| > r\}$$

3.3.4 Change-of-variance function (CVF)

Introduïda per Hampel(1986) amb una analogia directa amb la IF. On $V(T, F)$ és la variància asimptòtica. Podem definir CVF com

$$CVF(x; T, F) = \lim_{t \to 0} \frac{V(T, (1-t)F + t\Delta_x) - V(T, F)}{t}$$

3.3.5 Change-of-variance sensitivity (CVS)

L'estimador $T_n = T(F_n)$ del funcional $T(F)$ és anomenat *V-robust* si $k^*(T,F) < \infty$. Amb analogia al *gross-error sensitivity* podem definir CVS com

$$k^*(T,F) = \sup_x \frac{CVF(x;T,F)}{V(T,F)}$$

3.4 Punt de ruptura

El punt de ruptura (*Breakdown point, BP*) és una mesura per determinar la resistència d'un estimador. Podem definir el BP d'un estimador com la fracció més petita de contaminació que causa a l'estimador una "ruptura" i no representa el patró de la majoria de dades. És una mesura global de robustesa i no en una sola observació, com és el cas de la funció d'influència.

Podem definir BP com:

$$BP(T_n, X_n) = min\left\{\frac{m}{n}; \sup_{\tilde{X}_n} \left\| T_n(\tilde{X}_n) - T_n(X_n) \right\| = \infty\right\}$$

- $\tilde{X}_n$ és qualsevol conjunt de dades contaminades obtingudes canviant **m** observacions originals per a valors arbritaris.
- El valor més alt BP que es pot esperar és 50%, si més de la meitat de les dades estan contaminades, no es pot diferenciar entre dades "bones" i "dolentes".
- S'aconsella utilitzar un estimador amb BP més alt que la fracció esperada de valors atípics.

4 IF en el model normal

Considerem ara el model normal de localització

$$F(x) = F_\theta(x) = \Phi(x-\theta) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}}\, e^{-\frac{1}{2}(t-\theta)^2}\, dt$$

i que $\theta = 0$ (pàg 88 Hampel et al. [1986]).

4.1 Mitjana

Per a la mitjana

$$T(F) = \int x\, dF(x) = E_F(x)$$

.

- $IF(x, T, \Phi) = x$
- Eficiència= 100%
- BP=0% (El valor x pot tendir a ∞. Una sola observació pot canviar molt l'estimació.)

$$\text{On } IF(x; T, \Phi) = \lim_{t\to 0} \frac{\int ud[(1-t)\Phi + t\Delta_x](u) - \int ud\Phi(u)}{t}$$

$$= \lim_{t\to 0} \frac{(1-t)\int ud\Phi(u) + t\int ud\Delta_x(u) - \int ud\Phi(u)}{t}$$

$$= \lim_{t\to 0} \frac{tx}{t} = x$$

Ja que $\int ud\Phi(u) = 0$.

4.2 Mediana

Per a la mediana

$$T(F) = F^{-1}(1/2)$$

.

- $IF(x, T, \Phi) = \dfrac{sign(x)}{2\phi(0)}$
- Eficiència= $\dfrac{2}{\pi} \approx 64\%$ (veure en Hampel et al. [1986])
- BP=50% (molt a respecte a α de mitjana truncada i 0 en mitjana).

Agafant l'exemple trobat en Kahn [2015], que primerament calcula la funció d'influència de la mediana, M_x, en el cas on F_θ segueix una llei uniforme(0,1) i després considera el cas general utilitzant el teorema de la transformació inversa. Calculant la variància asimptòtica $V(T, \Phi) = \int IF(x, T, \Phi)^2 d\Phi(x)$, el resultat és el següent:

$$IF(x, M_x, \Phi)^2 = \begin{cases} \dfrac{0.5^2}{f_x^2(M_x)} = \dfrac{0.25}{f_x^2(0)} = \dfrac{\frac{1}{4}}{\frac{1}{(\sqrt{2\pi})^2}} = \dfrac{\pi}{2} \;\; x > 0 \\ \\ 0 \;\; x = 0 \\ \\ \dfrac{0.5^2}{f_x^2(M_x)} = \dfrac{0.25}{f_x^2(0)} = \dfrac{\frac{1}{4}}{\frac{1}{(\sqrt{2\pi})^2}} = \dfrac{\pi}{2} \;\; x > 0 \\ \\ \rightarrow \text{On } f_x(0) = \dfrac{1}{\sqrt{2\pi}} e^0 = \dfrac{1}{\sqrt{2\pi}}. \end{cases}$$

L'eficiència en la mitjana és 1. Cal calcular el ràtio entre les dues variàncies per a saber l'eficiència relativa de M_x. El valor és igual a $\dfrac{1}{\frac{\pi}{2}} \approx 0.64$.

4.3 Mitjana α truncada

Agafant els resultats de la IF en el cas general, ara tenim:

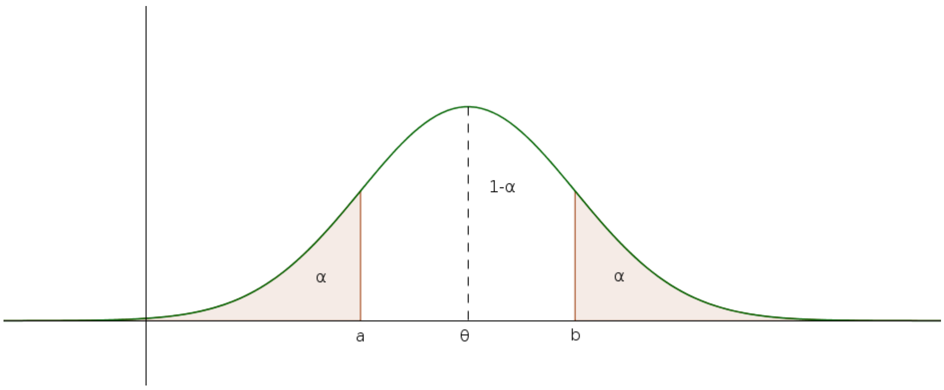

Figura 8: Representació de la Mitjana α truncada en Φ.

Podem veure que $\theta - a = b - \theta \rightarrow 2\theta - b = a$ per la simetria de la distribució normal i que volem trobar l'estimador $tm = \theta$.

Considerant:

- $\Phi(b-\theta) = 1-\alpha$
- $\Phi(a-\theta) = \theta$
- $b-\theta = \Phi^{-1}(1-\alpha)$
- $b = \theta + \Phi^{-1}(1-\alpha)$

La IC resulta:

sí $s \in [2\theta - b, b]$

$$= \frac{-\alpha}{1-2\alpha}(b+2\theta-b) + \frac{b+2\theta-b}{1-2\alpha} - \frac{s}{1-2\alpha} - \theta$$
$$= \frac{-2\theta\alpha + 2\theta - s - \theta + 2\alpha\theta}{1-2\alpha} = \frac{\theta - s}{1-2\alpha}$$

sí $s < a$

$$= -\frac{\alpha}{1-2\alpha}2\theta + \frac{b}{1-2\alpha} - \theta$$
$$= \frac{-2\alpha\theta + b - \theta + 2\alpha\theta}{1-2\alpha} = \frac{b-\theta}{1-2\alpha} = \frac{\Phi^{-1}(1-\alpha)}{1-2\alpha}$$

sí $s > b$

$$= -\frac{\alpha}{1-2\alpha}2\theta + \frac{2\theta - b}{1-2\alpha} - \theta$$
$$= \frac{-2\alpha\theta + 2\theta - b - \theta + 2\alpha\theta}{1-2\alpha} = \frac{\theta - b}{1-2\alpha} = -\frac{\Phi^{-1}(1-\alpha)}{1-2\alpha}$$

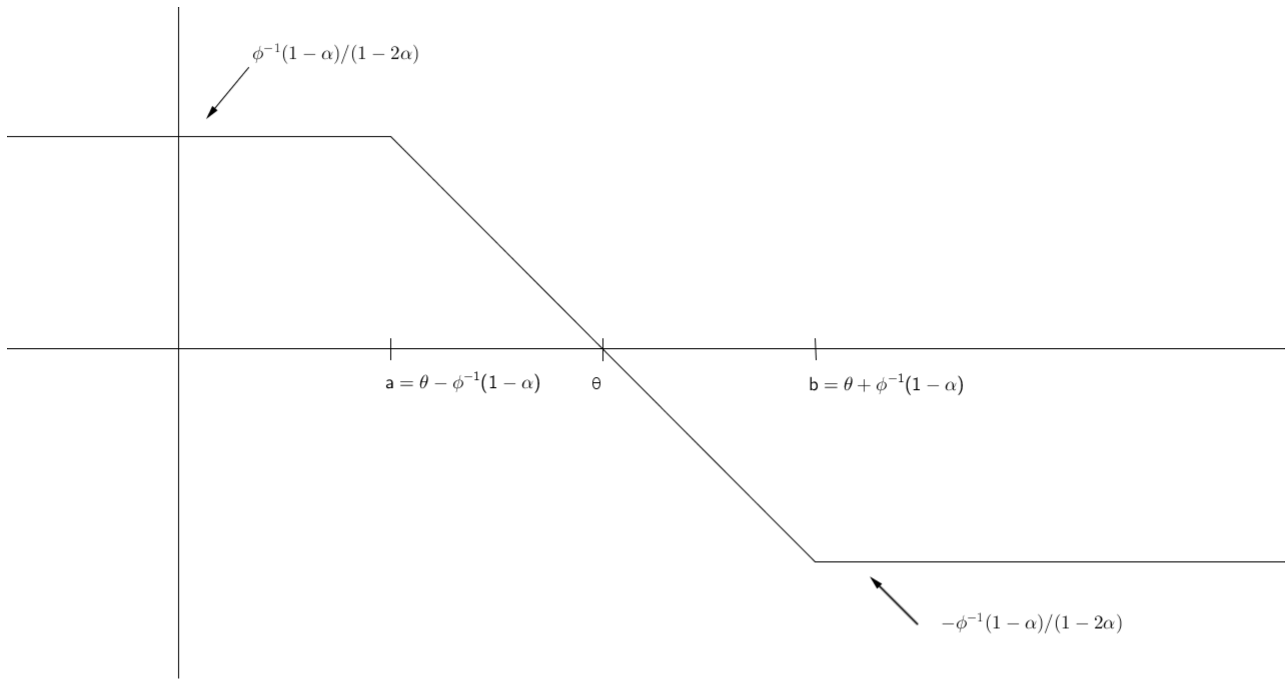

Figura 9: Gràfic de $IF(s, tm_\alpha, \Phi)$

5 Comentaris

La idea de que els models paramètrics són una aproximació de la realitat justifica la recerca d'estimadors que siguin més estables davant de petites desviacions del model suposat. Tot i que no existeix un únic criteri de robustesa, es tracta de resoldre un compromís entre estabilitat i eficiència. Aquest compromís introdueix una notable complicació als mètodes a emprar que dificulten la seva aplicació pràctica. Apart d'això, el desenvolupament teòric ha contribuït a popularitzar la noció de tractar l'estimador com a restricció d'un funcional (definit sobre l'espai de funcions de la distribució) al conjunt de les distribucions empíriques. Aquesta idea, ja es trobava implícita en la noció de consistència de Fisher i les primeres implicacions estadístiques es van dur a terme amb detall per Von Mises (1947). El que es proposa és estudiar la distribució asintòtica d'aquests funcionals.

De totes les eines i mesures de robustesa en podem distingir dues. Per una banda, tenim el punt de ruptura que intenta quantificar la robustesa en termes generals. El punt de ruptura (PR), per tant, es pot entendre com la quantitat màxima de contaminació en una distribució que pot tolerar un estimador de manera que encara proporcioni alguna informació sobre el paràmetre d'interès. Com hem definit abans, el punt de ruptura màxim és 50%. Estimadors amb PR alt són l'estimador de medianes repetides o els anomenats LMS(*least median of squares*). Molts d'ells, es construeixen a partir de dónar "robustesa" a la mesura d'error del criteri mínim quadràtic mitjançant una mesura robusta de la dispersió dels residus. Un estimador molt conegut amb un punt de ruptura alt i alhora eficient és el MM-estimador, proposat per Yohai (1987).

D' altra banda tenim la funció d'influència (IF), proposada per Hampel et al. [1986] que és sens dubte un dels conceptes més populars dintre de la teoria robusta. Des d'un punt de vista més matemàtic, la funció d'influència és per a un funcional estadístic el que el vector gradient és per a una funció real de n variables. El que proporciona aquesta funció és el terme lineal o de primer ordre del desenvolupament de Taylor que permet demostrar normalitat asintòtica per aquests estimadors que es defineixen com un funcional diferenciable. A conseqüència d'aquesta demostració asintòtica, la IF també ens permet trobar la variància asintòtica del funcional, calculant el quadrat d'aquest terme de primer ordre, és a dir, de la IF.

Tot i els esforços de Huber i Hampel per dónar forma matemàtica a les idees de robustesa, especialment en un context de localització univariant, queden molts aspectes a tractar i estudiar en l'estadística robusta. En aquest treball s'ha donat molt d'èmfasi a la funció d'influència, el que Hampel va anomenar enfocament infinitesimal, ja que s'estudia una perturbació per a un valor $t = 0$. Aquesta teoria està relacionada amb la teoria minimax que es pot trobar en Huber(1981), a partir de la derivada de la corba de biaix màxim i el suprem de la funció d'influència.

Referències

R Andersen. *Modern Methods For Robust Regression*. Thousand Oaks: SAGE Publications, 2008.

Patrick Breheny. Statistical functions and influence functions. URL http://web.as.uky.edu/statistics/users/pbreheny/621/F12/notes/8-28.pdf.

Brenton R. Clarke. A review of differentiability in relation to robustness with an application to seismic data analysis. *INSA (Indian National Science Academy)*, 66 (5):467–482, 2000. URL http://insa.nic.in/writereaddata/UpLoadedFiles/PINSA/Vol66A_2000_5_Art03.pdf.

Frank R. Hampel, Peter J. Rousseeuw, and A. Stahel Werner. *Robust Statistics: The Approach Based on Influence Functions*. 1986.

P.J. Huber. *Robust Statistics*. John Wiley & Sons, Inc., 1981.

Jay Kahn. Influence functions for fun and profit. 2015. URL http://j-kahn.com/files/influencefunctions.pdf.

R. A. Maronna, R. D. Martin, and V.J Yohai. *Robust Statistics: Theory and Methods*. Wiley, New York, 2006.

P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc., 1987. ISBN 0-471-85233-3. doi: 10.1002/0471725382.

Peter J. Rousseeuw and Elvezio Ronchetti. Influence curves of general statistics. *Journal of Computational and Applied Mathematics*, 7(3):161 – 166, 1981. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/0771-050X(81)90013-9. URL http://www.sciencedirect.com/science/article/pii/0771050X81900139.

Stephen M. Stigler. The changing history of robustness. *The American Statistician*, 64(4):277–281, 2010. doi: 10.1198/tast.2010.10159.

Dimitar Vandev. Robust methods in industrial statistics. URL http://www.fmi.uni-sofia.bg/fmi/statist/personal/vandev/papers/plenary_dv.pdf.

Ruber H. Zamar. Estimación robusta. 36(137):327–387, 1994.

```r
## Script - estudi simulació en el plantejament del model(2).
##


# Vector de Mides mostrals n a tractar:
n<-c(10,20,50)

# Vector Perturbació t a tractar:
t<-c(0.01,0.05,0.10,0.25,0.4)

# Mida simulació
S=10000

# Random per a tornar a generar.
# 10000(=S) valors ~ Uniforme (min=1,max=150000)

set.seed(16275711)
random<-round(runif(S,1,150000),0)



# Comparació estimadors localització:
# Mitjana truncada i Mediana amb màxim versemblant de la distribució

##################################

## Distribució Normal N(0,1) + Distribució Normal N(2,1)
# Màxim versemblant (Mitjana mostral, Var. mostral no corregida)


# Paràmetres sota condicions de model paramètric
mu=0  ; sigma=1 ;


# MIDA MOSTRAL n=10  n[1]
# Creem un data.frame per a cada estimador
# Mitjana mostral (màxim versemblant)

mean_mc<-data.frame(
  mean_mc11 = numeric(S),
  mean_mc12 = numeric(S),
  mean_mc13 = numeric(S),
  mean_mc14 = numeric(S),
  mean_mc15 = numeric(S)
)
```

```r
# Mediana
med_mc<-data.frame(
  med_mc11 = numeric(S),
  med_mc12 = numeric(S),
  med_mc13 = numeric(S),
  med_mc14 = numeric(S),
  med_mc15 = numeric(S)
)


# Trimmed mean alpha = t[1]

trim_mc_t1<-data.frame(
  trim_mc11_t1 = numeric(S),
  trim_mc12_t1 = numeric(S),
  trim_mc13_t1 = numeric(S),
  trim_mc14_t1 = numeric(S),
  trim_mc15_t1 = numeric(S)
)
# Trimmed mean alpha = t[2]

trim_mc_t2<-data.frame(
  trim_mc11_t2 = numeric(S),
  trim_mc12_t2 = numeric(S),
  trim_mc13_t2 = numeric(S),
  trim_mc14_t2 = numeric(S),
  trim_mc15_t2 = numeric(S)
)
# Trimmed mean alpha = t[3]

trim_mc_t3<-data.frame(
  trim_mc11_t3 = numeric(S),
  trim_mc12_t3 = numeric(S),
  trim_mc13_t3 = numeric(S),
  trim_mc14_t3 = numeric(S),
  trim_mc15_t3 = numeric(S)
)
# Trimmed mean alpha = t[4]

trim_mc_t4<-data.frame(
  trim_mc11_t4 = numeric(S),
  trim_mc12_t4 = numeric(S),
  trim_mc13_t4 = numeric(S),
  trim_mc14_t4 = numeric(S),
  trim_mc15_t4 = numeric(S)
)
# Trimmed mean alpha = t[5]
```

```r
trim_mc_t5<-data.frame(
  trim_mc11_t5 = numeric(S),
  trim_mc12_t5 = numeric(S),
  trim_mc13_t5 = numeric(S),
  trim_mc14_t5 = numeric(S),
  trim_mc15_t5 = numeric(S)
)

# Assignem a un objecte la posició del vector n[1]=10
n1<-n[1]

for (j in 1:length(t))

{
  for(i in 1:S)

  {


    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
    af<-sum(a=="F")
    ag<-sum(a=="G")


    #Simulem mostra s_i
    mostra_af<-rnorm(af,mu,sigma)
    mostra_ag<-rnorm(ag,mu+2,sigma)
    mostra<-c(mostra_af,mostra_ag)



    #Càlcul en el data.frame

    # Màxim versemblant
    mean_mc[i,j] = mean(mostra)

    # Mediana
    med_mc[i,j] = median(mostra)

    # Mitjana truncada en alpha=perturbació t[1] = 0.01
    trim_mc_t1[i,j]= mean(mostra,trim=t[1])
    # Mitjana truncada en alpha=perturbació t[2] = 0.05
    trim_mc_t2[i,j]= mean(mostra,trim=t[2])
    # Mitjana truncada en alpha=perturbació t[3] = 0.1
```

```
    trim_mc_t3[i,j]= mean(mostra,trim=t[3])
    # Mitjana truncada en alpha=perturbació t[4] = 0.25
    trim_mc_t4[i,j]= mean(mostra,trim=t[4])
    # Mitjana truncada en alpha=perturbació t[5] = 0.4
    trim_mc_t5[i,j]= mean(mostra,trim=t[5])

  }

}


# Càlcul mesures

# Mitjana de les S mostres

# Mitjana
mean_MC=apply(mean_mc,2,mean)
# Mediana
med_MC=apply(med_mc,2,mean)

# Mitjana truncada en alpha=perturbació t[1] = 0.01
trim_t1_MC=apply(trim_mc_t1,2,mean)
# Mitjana truncada en alpha=perturbació t[2] = 0.05
trim_t2_MC=apply(trim_mc_t2,2,mean)
# Mitjana truncada en alpha=perturbació t[3] = 0.1
trim_t3_MC=apply(trim_mc_t3,2,mean)
# Mitjana truncada en alpha=perturbació t[4] = 0.25
trim_t4_MC=apply(trim_mc_t4,2,mean)
# Mitjana truncada en alpha=perturbació t[5] = 0.4
trim_t5_MC=apply(trim_mc_t5,2,mean)




# Desviació no corregida de les S mostres


sd_mean_MC=apply(mean_mc,2,sd)
sd_mean_MC=sd_mean_MC*sqrt((S-1)/S)

sd_med_MC=apply(med_mc,2,sd)
sd_med_MC=sd_med_MC*sqrt((S-1)/S)


sd_trim_t1_MC=apply(trim_mc_t1,2,sd)
sd_trim_t1_MC=sd_trim_t1_MC*sqrt((S-1)/S)
```

```
sd_trim_t2_MC=apply(trim_mc_t2,2,sd)
sd_trim_t2_MC=sd_trim_t2_MC*sqrt((S-1)/S)

sd_trim_t3_MC=apply(trim_mc_t3,2,sd)
sd_trim_t3_MC=sd_trim_t3_MC*sqrt((S-1)/S)

sd_trim_t4_MC=apply(trim_mc_t4,2,sd)
sd_trim_t4_MC=sd_trim_t4_MC*sqrt((S-1)/S)

sd_trim_t5_MC=apply(trim_mc_t5,2,sd)
sd_trim_t5_MC=sd_trim_t5_MC*sqrt((S-1)/S)


# Biaix

b_mean_MC=mean_MC-rep(mu,5)

b_med_MC=med_MC-rep(mu,5)


b_trim_t1_MC=trim_t1_MC-rep(mu,5)
b_trim_t2_MC=trim_t2_MC-rep(mu,5)
b_trim_t3_MC=trim_t3_MC-rep(mu,5)
b_trim_t4_MC=trim_t4_MC-rep(mu,5)
b_trim_t5_MC=trim_t5_MC-rep(mu,5)

# EQM

eqm_mean_MC=b_mean_MC^2+sd_mean_MC^2

eqm_med_MC=b_med_MC^2+sd_med_MC^2


eqm_trim_t1_MC=b_trim_t1_MC^2+sd_trim_t1_MC^2
eqm_trim_t2_MC=b_trim_t2_MC^2+sd_trim_t2_MC^2
eqm_trim_t3_MC=b_trim_t3_MC^2+sd_trim_t3_MC^2
eqm_trim_t4_MC=b_trim_t4_MC^2+sd_trim_t4_MC^2
eqm_trim_t5_MC=b_trim_t5_MC^2+sd_trim_t5_MC^2




# bty="n" sense caixa en llegenda


plot(t,b_mean_MC,type="l",col=1,lty=1,lwd=2,
     main="",
```

```
     sub="(Mostra n=10)",
     xlab="Perturbació t",ylab="Biaix estimador MC")

lines(t,b_trim_t1_MC,type="l",col=2,lty=2,lwd=2)
lines(t,b_trim_t2_MC,type="l",col=3,lty=3,lwd=2)
lines(t,b_trim_t3_MC,type="l",col=4,lty=4,lwd=2)
lines(t,b_trim_t4_MC,type="l",col=5,lty=5,lwd=2)
lines(t,b_trim_t5_MC,type="l",col=6,lty=6,lwd=2)
lines(t,b_med_MC,type="l",    col=8,lty=8,lwd=2)


legend("topleft",
       legend=c(expression(X [n]),

                expression(paste(tm[alpha]," ,",alpha," =  0.01")),
                expression(paste(tm[alpha]," ,",alpha," =  0.05")),
                expression(paste(tm[alpha]," ,",alpha," =  0.1")),
                expression(paste(tm[alpha]," ,",alpha," =  0.25")),
                expression(paste(tm[alpha]," ,",alpha," =  0.4")),
                expression(M [x])),
       col=c(1:6,8),lty=c(1:6,8),lwd=2,bty="n")


save.image("n01_n21_n1.Rdata")

#taula 1
# Mitjana n1
round(b_mean_MC,3)
round(eqm_mean_MC,3)
#Mediana n1
round(b_med_MC,3)
round(eqm_med_MC,3)
#tm alpha=0.01
round(b_trim_t1_MC,3)
round(eqm_trim_t1_MC,3)
#tm alpha=0.05
round(b_trim_t2_MC,3)
round(eqm_trim_t2_MC,3)
#tm alpha=0.1
round(b_trim_t3_MC,3)
round(eqm_trim_t3_MC,3)
#tm alpha=0.25
round(b_trim_t4_MC,3)
round(eqm_trim_t4_MC,3)
#tm alpha=0.4
round(b_trim_t5_MC,3)
round(eqm_trim_t5_MC,3)
```

```r
#############################################

# MIDA MOSTRAL n=20


# Mitjana mostral (màxim versemblant)

mean_mc_n2<-data.frame(
  mean_mc11_n2 = numeric(S),
  mean_mc12_n2 = numeric(S),
  mean_mc13_n2 = numeric(S),
  mean_mc14_n2 = numeric(S),
  mean_mc15_n2 = numeric(S)
)


# Mediana
med_mc_n2<-data.frame(
  med_mc11_n2 = numeric(S),
  med_mc12_n2 = numeric(S),
  med_mc13_n2 = numeric(S),
  med_mc14_n2 = numeric(S),
  med_mc15_n2 = numeric(S)
)


# Trimmed mean alpha = t[1]

trim_mc_t1_n2<-data.frame(
  trim_mc11_t1_n2 = numeric(S),
  trim_mc12_t1_n2 = numeric(S),
  trim_mc13_t1_n2 = numeric(S),
  trim_mc14_t1_n2 = numeric(S),
  trim_mc15_t1_n2 = numeric(S)
)
# Trimmed mean alpha = t[2]

trim_mc_t2_n2<-data.frame(
  trim_mc11_t2_n2 = numeric(S),
  trim_mc12_t2_n2 = numeric(S),
  trim_mc13_t2_n2 = numeric(S),
  trim_mc14_t2_n2 = numeric(S),
  trim_mc15_t2_n2 = numeric(S)
)
# Trimmed mean alpha = t[3]

trim_mc_t3_n2<-data.frame(
```

```r
  trim_mc11_t3_n2 = numeric(S),
  trim_mc12_t3_n2 = numeric(S),
  trim_mc13_t3_n2 = numeric(S),
  trim_mc14_t3_n2 = numeric(S),
  trim_mc15_t3_n2 = numeric(S)
)
# Trimmed mean alpha = t[4]

trim_mc_t4_n2<-data.frame(
  trim_mc11_t4_n2 = numeric(S),
  trim_mc12_t4_n2 = numeric(S),
  trim_mc13_t4_n2 = numeric(S),
  trim_mc14_t4_n2 = numeric(S),
  trim_mc15_t4_n2 = numeric(S)
)
# Trimmed mean alpha = t[5]

trim_mc_t5_n2<-data.frame(
  trim_mc11_t5_n2 = numeric(S),
  trim_mc12_t5_n2 = numeric(S),
  trim_mc13_t5_n2 = numeric(S),
  trim_mc14_t5_n2 = numeric(S),
  trim_mc15_t5_n2 = numeric(S)
)

# Assignem a un objecte la posició del vector n[1]=10
n1<-n[2]

for (j in 1:length(t))

{
  for(i in 1:S)

  {


    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
    af<-sum(a=="F")
    ag<-sum(a=="G")


    #Simulem mostra s_i
    mostra_af<-rnorm(af,mu,sigma)
    mostra_ag<-rnorm(ag,mu+2,sigma)
```

```
    mostra<-c(mostra_af,mostra_ag)



    #Càlcul en el data.frame

    # Màxim versemblant
    mean_mc_n2[i,j] = mean(mostra)

    # Mediana
    med_mc_n2[i,j] = median(mostra)

    # Mitjana truncada en alpha=perturbació t[1] = 0.01
    trim_mc_t1_n2[i,j]= mean(mostra,trim=t[1])
    # Mitjana truncada en alpha=perturbació t[2] = 0.05
    trim_mc_t2_n2[i,j]= mean(mostra,trim=t[2])
    # Mitjana truncada en alpha=perturbació t[3] = 0.1
    trim_mc_t3_n2[i,j]= mean(mostra,trim=t[3])
    # Mitjana truncada en alpha=perturbació t[4] = 0.25
    trim_mc_t4_n2[i,j]= mean(mostra,trim=t[4])
    # Mitjana truncada en alpha=perturbació t[5] = 0.4
    trim_mc_t5_n2[i,j]= mean(mostra,trim=t[5])

  }

}

# Càlcul mesures

# Mitjana de les S mostres

# Mitjana
mean_MC_n2=apply(mean_mc_n2,2,mean)
# Mediana
med_MC_n2=apply(med_mc_n2,2,mean)

# Mitjana truncada en alpha=perturbació t[1] = 0.01
trim_t1_MC_n2=apply(trim_mc_t1_n2,2,mean)
# Mitjana truncada en alpha=perturbació t[2] = 0.05
trim_t2_MC_n2=apply(trim_mc_t2_n2,2,mean)
# Mitjana truncada en alpha=perturbació t[3] = 0.1
trim_t3_MC_n2=apply(trim_mc_t3_n2,2,mean)
# Mitjana truncada en alpha=perturbació t[4] = 0.25
trim_t4_MC_n2=apply(trim_mc_t4_n2,2,mean)
# Mitjana truncada en alpha=perturbació t[5] = 0.4
trim_t5_MC_n2=apply(trim_mc_t5_n2,2,mean)
```

```r
# Desviació no corregida de les S mostres

sd_mean_MC_n2=apply(mean_mc_n2,2,sd)
sd_mean_MC_n2=sd_mean_MC_n2*sqrt((S-1)/S)

sd_med_MC_n2=apply(med_mc_n2,2,sd)
sd_med_MC_n2=sd_med_MC_n2*sqrt((S-1)/S)


sd_trim_t1_MC_n2=apply(trim_mc_t1_n2,2,sd)
sd_trim_t1_MC_n2=sd_trim_t1_MC_n2*sqrt((S-1)/S)

sd_trim_t2_MC_n2=apply(trim_mc_t2_n2,2,sd)
sd_trim_t2_MC_n2=sd_trim_t2_MC_n2*sqrt((S-1)/S)

sd_trim_t3_MC_n2=apply(trim_mc_t3_n2,2,sd)
sd_trim_t3_MC_n2=sd_trim_t3_MC_n2*sqrt((S-1)/S)

sd_trim_t4_MC_n2=apply(trim_mc_t4_n2,2,sd)
sd_trim_t4_MC_n2=sd_trim_t4_MC_n2*sqrt((S-1)/S)

sd_trim_t5_MC_n2=apply(trim_mc_t5_n2,2,sd)
sd_trim_t5_MC_n2=sd_trim_t5_MC_n2*sqrt((S-1)/S)


# Biaix

b_mean_MC_n2=mean_MC_n2-rep(mu,5)

b_med_MC_n2=med_MC_n2-rep(mu,5)


b_trim_t1_MC_n2=trim_t1_MC_n2-rep(mu,5)
b_trim_t2_MC_n2=trim_t2_MC_n2-rep(mu,5)
b_trim_t3_MC_n2=trim_t3_MC_n2-rep(mu,5)
b_trim_t4_MC_n2=trim_t4_MC_n2-rep(mu,5)
b_trim_t5_MC_n2=trim_t5_MC_n2-rep(mu,5)

# EQM

eqm_mean_MC_n2=b_mean_MC_n2^2+sd_mean_MC_n2^2

eqm_med_MC_n2=b_med_MC_n2^2+sd_med_MC_n2^2
```

```
eqm_trim_t1_MC_n2=b_trim_t1_MC_n2^2+sd_trim_t1_MC_n2^2
eqm_trim_t2_MC_n2=b_trim_t2_MC_n2^2+sd_trim_t2_MC_n2^2
eqm_trim_t3_MC_n2=b_trim_t3_MC_n2^2+sd_trim_t3_MC_n2^2
eqm_trim_t4_MC_n2=b_trim_t4_MC_n2^2+sd_trim_t4_MC_n2^2
eqm_trim_t5_MC_n2=b_trim_t5_MC_n2^2+sd_trim_t5_MC_n2^2



plot(t,b_mean_MC_n2,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=20)",
     xlab="Perturbació t",ylab="Biaix estimador MC")

lines(t,b_trim_t1_MC_n2,type="l",col=2,lty=2,lwd=2)
lines(t,b_trim_t2_MC_n2,type="l",col=3,lty=3,lwd=2)
lines(t,b_trim_t3_MC_n2,type="l",col=4,lty=4,lwd=2)
lines(t,b_trim_t4_MC_n2,type="l",col=5,lty=5,lwd=2)
lines(t,b_trim_t5_MC_n2,type="l",col=6,lty=6,lwd=2)
lines(t,b_med_MC_n2,type="l",    col=8,lty=8,lwd=2)


legend("topleft",
       legend=c(expression(X [n]),
                expression(paste(tm[alpha]," ,",alpha," =  0.01")),
                expression(paste(tm[alpha]," ,",alpha," =  0.05")),
                expression(paste(tm[alpha]," ,",alpha," =  0.1")),
                expression(paste(tm[alpha]," ,",alpha," =  0.25")),
                expression(paste(tm[alpha]," ,",alpha," =  0.4")),
                expression(M [x])),
       col=c(1:6,8),lty=c(1:6,8),lwd=2,bty="n")


save.image("n01_n21_n2.Rdata")

#taula 1
# Mitjana n1
round(b_mean_MC_n2,3)
round(eqm_mean_MC_n2,3)
#Mediana n1
round(b_med_MC_n2,3)
round(eqm_med_MC_n2,3)
#tm alpha=0.01
round(b_trim_t1_MC_n2,3)
round(eqm_trim_t1_MC_n2,3)
#tm alpha=0.05
round(b_trim_t2_MC_n2,3)
round(eqm_trim_t2_MC_n2,3)
```

```r
#tm alpha=0.1
round(b_trim_t3_MC_n2,3)
round(eqm_trim_t3_MC_n2,3)
#tm alpha=0.25
round(b_trim_t4_MC_n2,3)
round(eqm_trim_t4_MC_n2,3)
#tm alpha=0.4
round(b_trim_t5_MC_n2,3)
round(eqm_trim_t5_MC_n2,3)



##############

# MIDA MOSTRAL n=50



# Mitjana mostral (màxim versemblant)

mean_mc_n3<-data.frame(
  mean_mc11_n3 = numeric(S),
  mean_mc12_n3 = numeric(S),
  mean_mc13_n3 = numeric(S),
  mean_mc14_n3 = numeric(S),
  mean_mc15_n3 = numeric(S)
)


# Mediana
med_mc_n3<-data.frame(
  med_mc11_n3 = numeric(S),
  med_mc12_n3 = numeric(S),
  med_mc13_n3 = numeric(S),
  med_mc14_n3 = numeric(S),
  med_mc15_n3 = numeric(S)
)


# Trimmed mean alpha = t[1]

trim_mc_t1_n3<-data.frame(
  trim_mc11_t1_n3 = numeric(S),
  trim_mc12_t1_n3 = numeric(S),
  trim_mc13_t1_n3 = numeric(S),
  trim_mc14_t1_n3 = numeric(S),
  trim_mc15_t1_n3 = numeric(S)
```

```
)
# Trimmed mean alpha = t[2]

trim_mc_t2_n3<-data.frame(
  trim_mc11_t2_n3 = numeric(S),
  trim_mc12_t2_n3 = numeric(S),
  trim_mc13_t2_n3 = numeric(S),
  trim_mc14_t2_n3 = numeric(S),
  trim_mc15_t2_n3 = numeric(S)
)
# Trimmed mean alpha = t[3]

trim_mc_t3_n3<-data.frame(
  trim_mc11_t3_n3 = numeric(S),
  trim_mc12_t3_n3 = numeric(S),
  trim_mc13_t3_n3 = numeric(S),
  trim_mc14_t3_n3 = numeric(S),
  trim_mc15_t3_n3 = numeric(S)
)
# Trimmed mean alpha = t[4]

trim_mc_t4_n3<-data.frame(
  trim_mc11_t4_n3 = numeric(S),
  trim_mc12_t4_n3 = numeric(S),
  trim_mc13_t4_n3 = numeric(S),
  trim_mc14_t4_n3 = numeric(S),
  trim_mc15_t4_n3 = numeric(S)
)
# Trimmed mean alpha = t[5]

trim_mc_t5_n3<-data.frame(
  trim_mc11_t5_n3 = numeric(S),
  trim_mc12_t5_n3 = numeric(S),
  trim_mc13_t5_n3 = numeric(S),
  trim_mc14_t5_n3 = numeric(S),
  trim_mc15_t5_n3 = numeric(S)
)

# Assignem a un objecte la posició del vector n[1]=10
n1<-n[3]

for (j in 1:length(t))

{
  for(i in 1:S)

  {
```

```r
    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
    af<-sum(a=="F")
    ag<-sum(a=="G")


    #Simulem mostra s_i
    mostra_af<-rnorm(af,mu,sigma)
    mostra_ag<-rnorm(ag,mu+2,sigma)
    mostra<-c(mostra_af,mostra_ag)



    #Càlcul en el data.frame

    # Màxim versemblant
    mean_mc_n3[i,j] = mean(mostra)

    # Mediana
    med_mc_n3[i,j] = median(mostra)

    # Mitjana truncada en alpha=perturbació t[1] = 0.01
    trim_mc_t1_n3[i,j]= mean(mostra,trim=t[1])
    # Mitjana truncada en alpha=perturbació t[2] = 0.05
    trim_mc_t2_n3[i,j]= mean(mostra,trim=t[2])
    # Mitjana truncada en alpha=perturbació t[3] = 0.1
    trim_mc_t3_n3[i,j]= mean(mostra,trim=t[3])
    # Mitjana truncada en alpha=perturbació t[4] = 0.25
    trim_mc_t4_n3[i,j]= mean(mostra,trim=t[4])
    # Mitjana truncada en alpha=perturbació t[5] = 0.4
    trim_mc_t5_n3[i,j]= mean(mostra,trim=t[5])

  }

}


# Càlcul mesures

# Mitjana de les S mostres

# Mitjana
mean_MC_n3=apply(mean_mc_n3,2,mean)
# Mediana
```

```r
med_MC_n3=apply(med_mc_n3,2,mean)

# Mitjana truncada en alpha=perturbació t[1] = 0.01
trim_t1_MC_n3=apply(trim_mc_t1_n3,2,mean)
# Mitjana truncada en alpha=perturbació t[2] = 0.05
trim_t2_MC_n3=apply(trim_mc_t2_n3,2,mean)
# Mitjana truncada en alpha=perturbació t[3] = 0.1
trim_t3_MC_n3=apply(trim_mc_t3_n3,2,mean)
# Mitjana truncada en alpha=perturbació t[4] = 0.25
trim_t4_MC_n3=apply(trim_mc_t4_n3,2,mean)
# Mitjana truncada en alpha=perturbació t[5] = 0.4
trim_t5_MC_n3=apply(trim_mc_t5_n3,2,mean)





# Desviació no corregida de les S mostres


sd_mean_MC_n3=apply(mean_mc_n3,2,sd)
sd_mean_MC_n3=sd_mean_MC_n3*sqrt((S-1)/S)

sd_med_MC_n3=apply(med_mc_n3,2,sd)
sd_med_MC_n3=sd_med_MC_n3*sqrt((S-1)/S)


sd_trim_t1_MC_n3=apply(trim_mc_t1_n3,2,sd)
sd_trim_t1_MC_n3=sd_trim_t1_MC_n3*sqrt((S-1)/S)

sd_trim_t2_MC_n3=apply(trim_mc_t2_n3,2,sd)
sd_trim_t2_MC_n3=sd_trim_t2_MC_n3*sqrt((S-1)/S)

sd_trim_t3_MC_n3=apply(trim_mc_t3_n3,2,sd)
sd_trim_t3_MC_n3=sd_trim_t3_MC_n3*sqrt((S-1)/S)

sd_trim_t4_MC_n3=apply(trim_mc_t4_n3,2,sd)
sd_trim_t4_MC_n3=sd_trim_t4_MC_n3*sqrt((S-1)/S)

sd_trim_t5_MC_n3=apply(trim_mc_t5_n3,2,sd)
sd_trim_t5_MC_n3=sd_trim_t5_MC_n3*sqrt((S-1)/S)


# Biaix

b_mean_MC_n3=mean_MC_n3-rep(mu,5)

b_med_MC_n3=med_MC_n3-rep(mu,5)
```

```r
b_trim_t1_MC_n3=trim_t1_MC_n3-rep(mu,5)
b_trim_t2_MC_n3=trim_t2_MC_n3-rep(mu,5)
b_trim_t3_MC_n3=trim_t3_MC_n3-rep(mu,5)
b_trim_t4_MC_n3=trim_t4_MC_n3-rep(mu,5)
b_trim_t5_MC_n3=trim_t5_MC_n3-rep(mu,5)

# EQM

eqm_mean_MC_n3=b_mean_MC_n3^2+sd_mean_MC_n3^2

eqm_med_MC_n3=b_med_MC_n3^2+sd_med_MC_n3^2


eqm_trim_t1_MC_n3=b_trim_t1_MC_n3^2+sd_trim_t1_MC_n3^2
eqm_trim_t2_MC_n3=b_trim_t2_MC_n3^2+sd_trim_t2_MC_n3^2
eqm_trim_t3_MC_n3=b_trim_t3_MC_n3^2+sd_trim_t3_MC_n3^2
eqm_trim_t4_MC_n3=b_trim_t4_MC_n3^2+sd_trim_t4_MC_n3^2
eqm_trim_t5_MC_n3=b_trim_t5_MC_n3^2+sd_trim_t5_MC_n3^2


save.image("n01_n21_n3.RData")


plot(t,b_mean_MC_n3,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=50)",
     xlab="Perturbació t",ylab="Biaix estimador MC")

lines(t,b_trim_t1_MC_n3,type="l",col=2,lty=2,lwd=2)
lines(t,b_trim_t2_MC_n3,type="l",col=3,lty=3,lwd=2)
lines(t,b_trim_t3_MC_n3,type="l",col=4,lty=4,lwd=2)
lines(t,b_trim_t4_MC_n3,type="l",col=5,lty=5,lwd=2)
lines(t,b_trim_t5_MC_n3,type="l",col=6,lty=6,lwd=2)
lines(t,b_med_MC_n3,type="l",    col=8,lty=8,lwd=2)


legend("topleft",
       legend=c(expression(X [n]),

                 expression(paste(tm[alpha]," ,",alpha," =  0.01")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.05")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.1")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.25")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.4")),
                 expression(M [x])),
       col=c(1:6,8),lty=c(1:6,8),lwd=2,bty="n")
```

```
#taula 1
# Mitjana n1
round(b_mean_MC_n3,3)
round(eqm_mean_MC_n3,3)
#Mediana n1
round(b_med_MC_n3,3)
round(eqm_med_MC_n3,3)
#tm alpha=0.01
round(b_trim_t1_MC_n3,3)
round(eqm_trim_t1_MC_n3,3)
#tm alpha=0.05
round(b_trim_t2_MC_n3,3)
round(eqm_trim_t2_MC_n3,3)
#tm alpha=0.1
round(b_trim_t3_MC_n3,3)
round(eqm_trim_t3_MC_n3,3)
#tm alpha=0.25
round(b_trim_t4_MC_n3,3)
round(eqm_trim_t4_MC_n3,3)
#tm alpha=0.4
round(b_trim_t5_MC_n3,3)
round(eqm_trim_t5_MC_n3,3)

##############################
# CAS DISPERSIÓ SD vs. MAD####
##############################
# CAS n=10
# desv. mostral (màxim versemblant)

sd_mc_disp_n1<-data.frame(
  sd_mc11_disp_n1 = numeric(S),
  sd_mc12_disp_n1 = numeric(S),
  sd_mc13_disp_n1 = numeric(S),
  sd_mc14_disp_n1 = numeric(S),
  sd_mc15_disp_n1 = numeric(S)
)


# Mediana
mad_mc_disp_n1<-data.frame(
  mad_mc11_disp_n1 = numeric(S),
  mad_mc12_disp_n1 = numeric(S),
  mad_mc13_disp_n1 = numeric(S),
  mad_mc14_disp_n1 = numeric(S),
  mad_mc15_disp_n1 = numeric(S)
```

```
)



# Assignem a un objecte la posició del vector n[1]=10
n1<-n[1]

for (j in 1:length(t))

{
  for(i in 1:S)

  {


    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
    af<-sum(a=="F")
    ag<-sum(a=="G")


    #Simulem mostra s_i
    mostra_af<-rnorm(af,mu,sigma)
    mostra_ag<-rnorm(ag,mu,sigma+1)
    mostra<-c(mostra_af,mostra_ag)



    #Càlcul en el data.frame

    # Màxim versemblant
    sd_mc_disp_n1[i,j] = sd(mostra)*((n1-1)/n1)

    # MAD
    mad_mc_disp_n1[i,j] = mad(mostra)


  }

}


# Càlcul mesures

# Mitjana de les S mostres
```

```
# Var no corregida
sd_MC_disp_n1=apply(sd_mc_disp_n1,2,mean)

# MAD
mad_MC_disp_n1=apply(mad_mc_disp_n1,2,mean)



# Desviació no corregida de les S mostres


sd_sd_MC_disp_n1=apply(sd_mc_disp_n1,2,sd)
sd_sd_MC_disp_n1=sd_sd_MC_disp_n1*sqrt((S-1)/S)

sd_mad_MC_disp_n1=apply(mad_mc_disp_n1,2,sd)
sd_mad_MC_disp_n1=sd_mad_MC_disp_n1*sqrt((S-1)/S)


# Biaix

b_sd_MC_disp_n1=sd_MC_disp_n1-rep(1,5)

b_mad_MC_disp_n1=mad_MC_disp_n1-rep(1,5)



# EQM

eqm_sd_MC_disp_n1=b_sd_MC_disp_n1^2+sd_sd_MC_disp_n1^2

eqm_mad_MC_disp_n1=b_mad_MC_disp_n1^2+sd_mad_MC_disp_n1^2


# Càlcul taula

round(b_sd_MC_disp_n1,3)
round(b_mad_MC_disp_n1,3)

round(eqm_sd_MC_disp_n1,3)
round(eqm_mad_MC_disp_n1,3)

# Cas n=20
# desv. mostral (màxim versemblant)

sd_mc_disp_n2<-data.frame(
  sd_mc11_disp_n2 = numeric(S),
  sd_mc12_disp_n2 = numeric(S),
```

```
  sd_mc13_disp_n2 = numeric(S),
  sd_mc14_disp_n2 = numeric(S),
  sd_mc15_disp_n2 = numeric(S)
)


# MAD
mad_mc_disp_n2<-data.frame(
  mad_mc11_disp_n2 = numeric(S),
  mad_mc12_disp_n2 = numeric(S),
  mad_mc13_disp_n2 = numeric(S),
  mad_mc14_disp_n2 = numeric(S),
  mad_mc15_disp_n2 = numeric(S)
)


# Assignem n=20
n1<-n[2]

for (j in 1:length(t))

{
  for(i in 1:S)

  {


    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
    af<-sum(a=="F")
    ag<-sum(a=="G")


    #Simulem mostra s_i
    mostra_af<-rnorm(af,mu,sigma)
    mostra_ag<-rnorm(ag,mu,sigma+1)
    mostra<-c(mostra_af,mostra_ag)



    #Càlcul en el data.frame

    # Màxim versemblant
    sd_mc_disp_n2[i,j] = sd(mostra)*((n1-1)/n1)
```

```r
    # MAD
    mad_mc_disp_n2[i,j] = mad(mostra)


  }

}



# Càlcul mesures

# Mitjana de les S mostres

# Var no corregida
sd_MC_disp_n2=apply(sd_mc_disp_n2,2,mean)

# MAD
mad_MC_disp_n2=apply(mad_mc_disp_n2,2,mean)



# Desviació no corregida de les S mostres


sd_sd_MC_disp_n2=apply(sd_mc_disp_n2,2,sd)
sd_sd_MC_disp_n2=sd_sd_MC_disp_n2*sqrt((S-1)/S)

sd_mad_MC_disp_n2=apply(mad_mc_disp_n2,2,sd)
sd_mad_MC_disp_n2=sd_mad_MC_disp_n2*sqrt((S-1)/S)


# Biaix

b_sd_MC_disp_n2=sd_MC_disp_n2-rep(1,5)

b_mad_MC_disp_n2=mad_MC_disp_n2-rep(1,5)



# EQM

eqm_sd_MC_disp_n2=b_sd_MC_disp_n2^2+sd_sd_MC_disp_n2^2

eqm_mad_MC_disp_n2=b_mad_MC_disp_n2^2+sd_mad_MC_disp_n2^2

# Mesures
```

```
round(b_sd_MC_disp_n2,3)
round(b_mad_MC_disp_n2,3)
round(eqm_sd_MC_disp_n2,3)
round(eqm_mad_MC_disp_n2,3)






# CAS DISPERSIÓ SD vs. MAD
# Cas n=50
# desv. mostral (màxim versemblant)

sd_mc_disp_n3<-data.frame(
  sd_mc11_disp_n3 = numeric(S),
  sd_mc12_disp_n3 = numeric(S),
  sd_mc13_disp_n3 = numeric(S),
  sd_mc14_disp_n3 = numeric(S),
  sd_mc15_disp_n3 = numeric(S)
)


# MAD
mad_mc_disp_n3<-data.frame(
  mad_mc11_disp_n3 = numeric(S),
  mad_mc12_disp_n3 = numeric(S),
  mad_mc13_disp_n3 = numeric(S),
  mad_mc14_disp_n3 = numeric(S),
  mad_mc15_disp_n3 = numeric(S)
)


# Assignem n=50
n1<-n[3]

for (j in 1:length(t))

{
  for(i in 1:S)

  {


    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
```

```
    af<-sum(a=="F")
    ag<-sum(a=="G")


    #Simulem mostra s_i
    mostra_af<-rnorm(af,mu,sigma)
    mostra_ag<-rnorm(ag,mu,sigma+1)
    mostra<-c(mostra_af,mostra_ag)



    #Càlcul en el data.frame

    # Màxim versemblant
    sd_mc_disp_n3[i,j] = sd(mostra)*((n1-1)/n1)

    # MAD
    mad_mc_disp_n3[i,j] = mad(mostra)


  }

}



# Càlcul mesures

# Mitjana de les S mostres

# Var no corregida
sd_MC_disp_n3=apply(sd_mc_disp_n3,2,mean)

# MAD
mad_MC_disp_n3=apply(mad_mc_disp_n3,2,mean)



# Desviació no corregida de les S mostres


sd_sd_MC_disp_n3=apply(sd_mc_disp_n3,2,sd)
sd_sd_MC_disp_n3=sd_sd_MC_disp_n3*sqrt((S-1)/S)

sd_mad_MC_disp_n3=apply(mad_mc_disp_n3,2,sd)
sd_mad_MC_disp_n3=sd_mad_MC_disp_n3*sqrt((S-1)/S)
```

```r
# Biaix

b_sd_MC_disp_n3=sd_MC_disp_n3-rep(1,5)

b_mad_MC_disp_n3=mad_MC_disp_n3-rep(1,5)



# EQM

eqm_sd_MC_disp_n3=b_sd_MC_disp_n3^2+sd_sd_MC_disp_n3^2
eqm_mad_MC_disp_n3=b_mad_MC_disp_n3^2+sd_mad_MC_disp_n3^2

# Mesures
round(b_sd_MC_disp_n3,3)
round(b_mad_MC_disp_n3,3)
round(eqm_sd_MC_disp_n3,3)
round(eqm_mad_MC_disp_n3,3)






par(mfrow=c(2,2))
plot(t,b_sd_MC_disp_n1,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=10)",
     xlab="Perturbació t",ylab="Biaix estimador MC",
     ylim=c(-0.2,max(b_sd_MC_disp_n3)+0.1))
lines(t,b_mad_MC_disp_n1,type="l",col=2,lty=2,lwd=2)


plot(t,b_sd_MC_disp_n2,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=20)",
     xlab="Perturbació t",ylab="Biaix estimador MC",
     ylim=c(-0.2,max(b_sd_MC_disp_n3)+0.1))
lines(t,b_mad_MC_disp_n2,type="l",col=2,lty=2,lwd=2)



plot(t,b_sd_MC_disp_n3,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=50)",
     xlab="Perturbació t",ylab="Biaix estimador MC",
     ylim=c(-0.2,max(b_sd_MC_disp_n3)+0.1))
```

```
lines(t,b_mad_MC_disp_n3,type="l",col=2,lty=2,lwd=2)



plot(0,xaxt='n',yaxt='n',bty='n',pch='',ylab='',xlab='')

legend("center",
       legend=c(expression(sqrt(sigma^2)[n]),
                expression(MAD [x])),
       col=c(1:8),lty=c(1:8),lwd=2,bty="n")


##############################################
#  F=Distribució Normal (0,1)                #
#  G=Unif(3,4)                               #
#                                            #
#     Cas 2. (1-t)N(0,1)+(t)Unif(3,4)        #
##############################################

# MIDA MOSTRAL n=10  n[1]


# Mitjana mostral (màxim versemblant)

mean_mc_n1_unif<-data.frame(
  mean_mc11_n1_unif = numeric(S),
  mean_mc12_n1_unif = numeric(S),
  mean_mc13_n1_unif = numeric(S),
  mean_mc14_n1_unif = numeric(S),
  mean_mc15_n1_unif = numeric(S)
)


# Mediana
med_mc_n1_unif<-data.frame(
  med_mc11_n1_unif = numeric(S),
  med_mc12_n1_unif = numeric(S),
  med_mc13_n1_unif = numeric(S),
  med_mc14_n1_unif = numeric(S),
  med_mc15_n1_unif = numeric(S)
)


# Trimmed mean alpha = t[1]

trim_mc_t1_n1_unif<-data.frame(
  trim_mc11_t1_n1_unif = numeric(S),
```

```
  trim_mc12_t1_n1_unif = numeric(S),
  trim_mc13_t1_n1_unif = numeric(S),
  trim_mc14_t1_n1_unif = numeric(S),
  trim_mc15_t1_n1_unif = numeric(S)
)
# Trimmed mean alpha = t[2]

trim_mc_t2_n1_unif<-data.frame(
  trim_mc11_t2_n1_unif = numeric(S),
  trim_mc12_t2_n1_unif = numeric(S),
  trim_mc13_t2_n1_unif = numeric(S),
  trim_mc14_t2_n1_unif = numeric(S),
  trim_mc15_t2_n1_unif = numeric(S)
)
# Trimmed mean alpha = t[3]

trim_mc_t3_n1_unif<-data.frame(
  trim_mc11_t3_n1_unif = numeric(S),
  trim_mc12_t3_n1_unif = numeric(S),
  trim_mc13_t3_n1_unif = numeric(S),
  trim_mc14_t3_n1_unif = numeric(S),
  trim_mc15_t3_n1_unif = numeric(S)
)
# Trimmed mean alpha = t[4]

trim_mc_t4_n1_unif<-data.frame(
  trim_mc11_t4_n1_unif = numeric(S),
  trim_mc12_t4_n1_unif = numeric(S),
  trim_mc13_t4_n1_unif = numeric(S),
  trim_mc14_t4_n1_unif = numeric(S),
  trim_mc15_t4_n1_unif = numeric(S)
)
# Trimmed mean alpha = t[5]

trim_mc_t5_n1_unif<-data.frame(
  trim_mc11_t5_n1_unif = numeric(S),
  trim_mc12_t5_n1_unif = numeric(S),
  trim_mc13_t5_n1_unif = numeric(S),
  trim_mc14_t5_n1_unif = numeric(S),
  trim_mc15_t5_n1_unif = numeric(S)
)

# Assignem a un objecte la posició del vector n[1]=10
n1<-n[1]

for (j in 1:length(t))
```

```
{
  for(i in 1:S)

  {


    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
    af<-sum(a=="F")
    ag<-sum(a=="G")


    # Simulem mostra s_i
    mostra_af<-rnorm(af,mu,sigma)
    mostra_ag<-runif(ag,3,4)
    mostra<-c(mostra_af,mostra_ag)



    #Càlcul en el data.frame

    # Màxim versemblant
    mean_mc_n1_unif[i,j] = mean(mostra)

    # Mediana
    med_mc_n1_unif[i,j] = median(mostra)

    # Mitjana truncada en alpha=perturbació t[1] = 0.01
    trim_mc_t1_n1_unif[i,j]= mean(mostra,trim=t[1])
    # Mitjana truncada en alpha=perturbació t[2] = 0.05
    trim_mc_t2_n1_unif[i,j]= mean(mostra,trim=t[2])
    # Mitjana truncada en alpha=perturbació t[3] = 0.1
    trim_mc_t3_n1_unif[i,j]= mean(mostra,trim=t[3])
    # Mitjana truncada en alpha=perturbació t[4] = 0.25
    trim_mc_t4_n1_unif[i,j]= mean(mostra,trim=t[4])
    # Mitjana truncada en alpha=perturbació t[5] = 0.4
    trim_mc_t5_n1_unif[i,j]= mean(mostra,trim=t[5])

  }

}


# Càlcul mesures
```

```
# Mitjana de les S mostres

# Mitjana
mean_MC_n1_unif=apply(mean_mc_n1_unif,2,mean)
# Mediana
med_MC_n1_unif=apply(med_mc_n1_unif,2,mean)

# Mitjana truncada en alpha=perturbació t[1] = 0.01
trim_t1_MC_n1_unif=apply(trim_mc_t1_n1_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[2] = 0.05
trim_t2_MC_n1_unif=apply(trim_mc_t2_n1_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[3] = 0.1
trim_t3_MC_n1_unif=apply(trim_mc_t3_n1_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[4] = 0.25
trim_t4_MC_n1_unif=apply(trim_mc_t4_n1_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[5] = 0.4
trim_t5_MC_n1_unif=apply(trim_mc_t5_n1_unif,2,mean)




# Desviació no corregida de les S mostres


sd_mean_MC_n1_unif=apply(mean_mc_n1_unif,2,sd)
sd_mean_MC_n1_unif=sd_mean_MC_n1_unif*sqrt((S-1)/S)

sd_med_MC_n1_unif=apply(med_mc_n1_unif,2,sd)
sd_med_MC_n1_unif=sd_med_MC_n1_unif*sqrt((S-1)/S)


sd_trim_t1_MC_n1_unif=apply(trim_mc_t1_n1_unif,2,sd)
sd_trim_t1_MC_n1_unif=sd_trim_t1_MC_n1_unif*sqrt((S-1)/S)

sd_trim_t2_MC_n1_unif=apply(trim_mc_t2_n1_unif,2,sd)
sd_trim_t2_MC_n1_unif=sd_trim_t2_MC_n1_unif*sqrt((S-1)/S)

sd_trim_t3_MC_n1_unif=apply(trim_mc_t3_n1_unif,2,sd)
sd_trim_t3_MC_n1_unif=sd_trim_t3_MC_n1_unif*sqrt((S-1)/S)

sd_trim_t4_MC_n1_unif=apply(trim_mc_t4_n1_unif,2,sd)
sd_trim_t4_MC_n1_unif=sd_trim_t4_MC_n1_unif*sqrt((S-1)/S)

sd_trim_t5_MC_n1_unif=apply(trim_mc_t5_n1_unif,2,sd)
sd_trim_t5_MC_n1_unif=sd_trim_t5_MC_n1_unif*sqrt((S-1)/S)
```

```r
# Biaix

b_mean_MC_n1_unif=mean_MC_n1_unif-rep(mu,5)

b_med_MC_n1_unif=med_MC_n1_unif-rep(mu,5)

b_trim_t1_MC_n1_unif=trim_t1_MC_n1_unif-rep(mu,5)
b_trim_t2_MC_n1_unif=trim_t2_MC_n1_unif-rep(mu,5)
b_trim_t3_MC_n1_unif=trim_t3_MC_n1_unif-rep(mu,5)
b_trim_t4_MC_n1_unif=trim_t4_MC_n1_unif-rep(mu,5)
b_trim_t5_MC_n1_unif=trim_t5_MC_n1_unif-rep(mu,5)

# EQM

eqm_mean_MC_n1_unif=b_mean_MC_n1_unif^2+sd_mean_MC_n1_unif^2

eqm_med_MC_n1_unif=b_med_MC_n1_unif^2+sd_med_MC_n1_unif^2

eqm_trim_t1_MC_n1_unif=b_trim_t1_MC_n1_unif^2+sd_trim_t1_MC_n1_unif^2
eqm_trim_t2_MC_n1_unif=b_trim_t2_MC_n1_unif^2+sd_trim_t2_MC_n1_unif^2
eqm_trim_t3_MC_n1_unif=b_trim_t3_MC_n1_unif^2+sd_trim_t3_MC_n1_unif^2
eqm_trim_t4_MC_n1_unif=b_trim_t4_MC_n1_unif^2+sd_trim_t4_MC_n1_unif^2
eqm_trim_t5_MC_n1_unif=b_trim_t5_MC_n1_unif^2+sd_trim_t5_MC_n1_unif^2


##############
##############
# Cas n=20


# Mitjana mostral (màxim versemblant)

mean_mc_n2_unif<-data.frame(
  mean_mc11_n2_unif = numeric(S),
  mean_mc12_n2_unif = numeric(S),
  mean_mc13_n2_unif = numeric(S),
  mean_mc14_n2_unif = numeric(S),
  mean_mc15_n2_unif = numeric(S)
)


# Mediana
med_mc_n2_unif<-data.frame(
  med_mc11_n2_unif = numeric(S),
  med_mc12_n2_unif = numeric(S),
  med_mc13_n2_unif = numeric(S),
  med_mc14_n2_unif = numeric(S),
```

```r
  med_mc15_n2_unif = numeric(S)
)


# Trimmed mean alpha = t[1]

trim_mc_t1_n2_unif<-data.frame(
  trim_mc11_t1_n2_unif = numeric(S),
  trim_mc12_t1_n2_unif = numeric(S),
  trim_mc13_t1_n2_unif = numeric(S),
  trim_mc14_t1_n2_unif = numeric(S),
  trim_mc15_t1_n2_unif = numeric(S)
)
# Trimmed mean alpha = t[2]

trim_mc_t2_n2_unif<-data.frame(
  trim_mc11_t2_n2_unif = numeric(S),
  trim_mc12_t2_n2_unif = numeric(S),
  trim_mc13_t2_n2_unif = numeric(S),
  trim_mc14_t2_n2_unif = numeric(S),
  trim_mc15_t2_n2_unif = numeric(S)
)
# Trimmed mean alpha = t[3]

trim_mc_t3_n2_unif<-data.frame(
  trim_mc11_t3_n2_unif = numeric(S),
  trim_mc12_t3_n2_unif = numeric(S),
  trim_mc13_t3_n2_unif = numeric(S),
  trim_mc14_t3_n2_unif = numeric(S),
  trim_mc15_t3_n2_unif = numeric(S)
)
# Trimmed mean alpha = t[4]

trim_mc_t4_n2_unif<-data.frame(
  trim_mc11_t4_n2_unif = numeric(S),
  trim_mc12_t4_n2_unif = numeric(S),
  trim_mc13_t4_n2_unif = numeric(S),
  trim_mc14_t4_n2_unif = numeric(S),
  trim_mc15_t4_n2_unif = numeric(S)
)
# Trimmed mean alpha = t[5]

trim_mc_t5_n2_unif<-data.frame(
  trim_mc11_t5_n2_unif = numeric(S),
  trim_mc12_t5_n2_unif = numeric(S),
  trim_mc13_t5_n2_unif = numeric(S),
  trim_mc14_t5_n2_unif = numeric(S),
```

```r
  trim_mc15_t5_n2_unif = numeric(S)
)

# Assignem a un objecte la posició del vector n[2]=20
n1<-n[2]

for (j in 1:length(t))

{
  for(i in 1:S)

  {


    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
    af<-sum(a=="F")
    ag<-sum(a=="G")


    # Simulem mostra s_i
    mostra_af<-rnorm(af,mu,sigma)
    mostra_ag<-runif(ag,3,4)
    mostra<-c(mostra_af,mostra_ag)



    #Càlcul en el data.frame

    # Màxim versemblant
    mean_mc_n2_unif[i,j] = mean(mostra)

    # Mediana
    med_mc_n2_unif[i,j] = median(mostra)

    # Mitjana truncada en alpha=perturbació t[1] = 0.01
    trim_mc_t1_n2_unif[i,j]= mean(mostra,trim=t[1])
    # Mitjana truncada en alpha=perturbació t[2] = 0.05
    trim_mc_t2_n2_unif[i,j]= mean(mostra,trim=t[2])
    # Mitjana truncada en alpha=perturbació t[3] = 0.1
    trim_mc_t3_n2_unif[i,j]= mean(mostra,trim=t[3])
    # Mitjana truncada en alpha=perturbació t[4] = 0.25
    trim_mc_t4_n2_unif[i,j]= mean(mostra,trim=t[4])
    # Mitjana truncada en alpha=perturbació t[5] = 0.4
    trim_mc_t5_n2_unif[i,j]= mean(mostra,trim=t[5])
```

```
  }

}


# Càlcul mesures

# Mitjana de les S mostres

# Mitjana
mean_MC_n2_unif=apply(mean_mc_n2_unif,2,mean)
# Mediana
med_MC_n2_unif=apply(med_mc_n2_unif,2,mean)

# Mitjana truncada en alpha=perturbació t[1] = 0.01
trim_t1_MC_n2_unif=apply(trim_mc_t1_n2_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[2] = 0.05
trim_t2_MC_n2_unif=apply(trim_mc_t2_n2_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[3] = 0.1
trim_t3_MC_n2_unif=apply(trim_mc_t3_n2_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[4] = 0.25
trim_t4_MC_n2_unif=apply(trim_mc_t4_n2_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[5] = 0.4
trim_t5_MC_n2_unif=apply(trim_mc_t5_n2_unif,2,mean)




# Desviació no corregida de les S mostres


sd_mean_MC_n2_unif=apply(mean_mc_n2_unif,2,sd)
sd_mean_MC_n2_unif=sd_mean_MC_n2_unif*sqrt((S-1)/S)

sd_med_MC_n2_unif=apply(med_mc_n2_unif,2,sd)
sd_med_MC_n2_unif=sd_med_MC_n2_unif*sqrt((S-1)/S)


sd_trim_t1_MC_n2_unif=apply(trim_mc_t1_n2_unif,2,sd)
sd_trim_t1_MC_n2_unif=sd_trim_t1_MC_n2_unif*sqrt((S-1)/S)

sd_trim_t2_MC_n2_unif=apply(trim_mc_t2_n2_unif,2,sd)
sd_trim_t2_MC_n2_unif=sd_trim_t2_MC_n2_unif*sqrt((S-1)/S)

sd_trim_t3_MC_n2_unif=apply(trim_mc_t3_n2_unif,2,sd)
sd_trim_t3_MC_n2_unif=sd_trim_t3_MC_n2_unif*sqrt((S-1)/S)
```

```
sd_trim_t4_MC_n2_unif=apply(trim_mc_t4_n2_unif,2,sd)
sd_trim_t4_MC_n2_unif=sd_trim_t4_MC_n2_unif*sqrt((S-1)/S)

sd_trim_t5_MC_n2_unif=apply(trim_mc_t5_n2_unif,2,sd)
sd_trim_t5_MC_n2_unif=sd_trim_t5_MC_n2_unif*sqrt((S-1)/S)


# Biaix

b_mean_MC_n2_unif=mean_MC_n2_unif-rep(mu,5)

b_med_MC_n2_unif=med_MC_n2_unif-rep(mu,5)

b_trim_t1_MC_n2_unif=trim_t1_MC_n2_unif-rep(mu,5)
b_trim_t2_MC_n2_unif=trim_t2_MC_n2_unif-rep(mu,5)
b_trim_t3_MC_n2_unif=trim_t3_MC_n2_unif-rep(mu,5)
b_trim_t4_MC_n2_unif=trim_t4_MC_n2_unif-rep(mu,5)
b_trim_t5_MC_n2_unif=trim_t5_MC_n2_unif-rep(mu,5)

# EQM

eqm_mean_MC_n2_unif=b_mean_MC_n2_unif^2+sd_mean_MC_n2_unif^2

eqm_med_MC_n2_unif=b_med_MC_n2_unif^2+sd_med_MC_n2_unif^2

eqm_trim_t1_MC_n2_unif=b_trim_t1_MC_n2_unif^2+sd_trim_t1_MC_n2_unif^2
eqm_trim_t2_MC_n2_unif=b_trim_t2_MC_n2_unif^2+sd_trim_t2_MC_n2_unif^2
eqm_trim_t3_MC_n2_unif=b_trim_t3_MC_n2_unif^2+sd_trim_t3_MC_n2_unif^2
eqm_trim_t4_MC_n2_unif=b_trim_t4_MC_n2_unif^2+sd_trim_t4_MC_n2_unif^2
eqm_trim_t5_MC_n2_unif=b_trim_t5_MC_n2_unif^2+sd_trim_t5_MC_n2_unif^2


##############
##############
# Cas n=50


# Mitjana mostral (màxim versemblant)

mean_mc_n3_unif<-data.frame(
  mean_mc11_n3_unif = numeric(S),
  mean_mc12_n3_unif = numeric(S),
  mean_mc13_n3_unif = numeric(S),
  mean_mc14_n3_unif = numeric(S),
  mean_mc15_n3_unif = numeric(S)
)
```

```r
# Mediana
med_mc_n3_unif<-data.frame(
  med_mc11_n3_unif = numeric(S),
  med_mc12_n3_unif = numeric(S),
  med_mc13_n3_unif = numeric(S),
  med_mc14_n3_unif = numeric(S),
  med_mc15_n3_unif = numeric(S)
)


# Trimmed mean alpha = t[1]

trim_mc_t1_n3_unif<-data.frame(
  trim_mc11_t1_n3_unif = numeric(S),
  trim_mc12_t1_n3_unif = numeric(S),
  trim_mc13_t1_n3_unif = numeric(S),
  trim_mc14_t1_n3_unif = numeric(S),
  trim_mc15_t1_n3_unif = numeric(S)
)
# Trimmed mean alpha = t[2]

trim_mc_t2_n3_unif<-data.frame(
  trim_mc11_t2_n3_unif = numeric(S),
  trim_mc12_t2_n3_unif = numeric(S),
  trim_mc13_t2_n3_unif = numeric(S),
  trim_mc14_t2_n3_unif = numeric(S),
  trim_mc15_t2_n3_unif = numeric(S)
)
# Trimmed mean alpha = t[3]

trim_mc_t3_n3_unif<-data.frame(
  trim_mc11_t3_n3_unif = numeric(S),
  trim_mc12_t3_n3_unif = numeric(S),
  trim_mc13_t3_n3_unif = numeric(S),
  trim_mc14_t3_n3_unif = numeric(S),
  trim_mc15_t3_n3_unif = numeric(S)
)
# Trimmed mean alpha = t[4]

trim_mc_t4_n3_unif<-data.frame(
  trim_mc11_t4_n3_unif = numeric(S),
  trim_mc12_t4_n3_unif = numeric(S),
  trim_mc13_t4_n3_unif = numeric(S),
  trim_mc14_t4_n3_unif = numeric(S),
  trim_mc15_t4_n3_unif = numeric(S)
)
# Trimmed mean alpha = t[5]
```

```r
trim_mc_t5_n3_unif<-data.frame(
  trim_mc11_t5_n3_unif = numeric(S),
  trim_mc12_t5_n3_unif = numeric(S),
  trim_mc13_t5_n3_unif = numeric(S),
  trim_mc14_t5_n3_unif = numeric(S),
  trim_mc15_t5_n3_unif = numeric(S)
)

# Assignem a un objecte la posició del vector n[3]=50
n1<-n[3]

for (j in 1:length(t))

{
  for(i in 1:S)

  {


    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
    af<-sum(a=="F")
    ag<-sum(a=="G")


    # Simulem mostra s_i
    mostra_af<-rnorm(af,mu,sigma)
    mostra_ag<-runif(ag,3,4)
    mostra<-c(mostra_af,mostra_ag)



    #Càlcul en el data.frame

    # Màxim versemblant
    mean_mc_n3_unif[i,j] = mean(mostra)

    # Mediana
    med_mc_n3_unif[i,j] = median(mostra)

    # Mitjana truncada en alpha=perturbació t[1] = 0.01
    trim_mc_t1_n3_unif[i,j]= mean(mostra,trim=t[1])
    # Mitjana truncada en alpha=perturbació t[2] = 0.05
    trim_mc_t2_n3_unif[i,j]= mean(mostra,trim=t[2])
    # Mitjana truncada en alpha=perturbació t[3] = 0.1
```

```
    trim_mc_t3_n3_unif[i,j]= mean(mostra,trim=t[3])
    # Mitjana truncada en alpha=perturbació t[4] = 0.25
    trim_mc_t4_n3_unif[i,j]= mean(mostra,trim=t[4])
    # Mitjana truncada en alpha=perturbació t[5] = 0.4
    trim_mc_t5_n3_unif[i,j]= mean(mostra,trim=t[5])

  }

}


# Càlcul mesures

# Mitjana de les S mostres

# Mitjana
mean_MC_n3_unif=apply(mean_mc_n3_unif,2,mean)
# Mediana
med_MC_n3_unif=apply(med_mc_n3_unif,2,mean)

# Mitjana truncada en alpha=perturbació t[1] = 0.01
trim_t1_MC_n3_unif=apply(trim_mc_t1_n3_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[2] = 0.05
trim_t2_MC_n3_unif=apply(trim_mc_t2_n3_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[3] = 0.1
trim_t3_MC_n3_unif=apply(trim_mc_t3_n3_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[4] = 0.25
trim_t4_MC_n3_unif=apply(trim_mc_t4_n3_unif,2,mean)
# Mitjana truncada en alpha=perturbació t[5] = 0.4
trim_t5_MC_n3_unif=apply(trim_mc_t5_n3_unif,2,mean)




# Desviació no corregida de les S mostres


sd_mean_MC_n3_unif=apply(mean_mc_n3_unif,2,sd)
sd_mean_MC_n3_unif=sd_mean_MC_n3_unif*sqrt((S-1)/S)

sd_med_MC_n3_unif=apply(med_mc_n3_unif,2,sd)
sd_med_MC_n3_unif=sd_med_MC_n3_unif*sqrt((S-1)/S)


sd_trim_t1_MC_n3_unif=apply(trim_mc_t1_n3_unif,2,sd)
sd_trim_t1_MC_n3_unif=sd_trim_t1_MC_n3_unif*sqrt((S-1)/S)
```

```r
sd_trim_t2_MC_n3_unif=apply(trim_mc_t2_n3_unif,2,sd)
sd_trim_t2_MC_n3_unif=sd_trim_t2_MC_n3_unif*sqrt((S-1)/S)

sd_trim_t3_MC_n3_unif=apply(trim_mc_t3_n3_unif,2,sd)
sd_trim_t3_MC_n3_unif=sd_trim_t3_MC_n3_unif*sqrt((S-1)/S)

sd_trim_t4_MC_n3_unif=apply(trim_mc_t4_n3_unif,2,sd)
sd_trim_t4_MC_n3_unif=sd_trim_t4_MC_n3_unif*sqrt((S-1)/S)

sd_trim_t5_MC_n3_unif=apply(trim_mc_t5_n3_unif,2,sd)
sd_trim_t5_MC_n3_unif=sd_trim_t5_MC_n3_unif*sqrt((S-1)/S)


# Biaix

b_mean_MC_n3_unif=mean_MC_n3_unif-rep(mu,5)

b_med_MC_n3_unif=med_MC_n3_unif-rep(mu,5)

b_trim_t1_MC_n3_unif=trim_t1_MC_n3_unif-rep(mu,5)
b_trim_t2_MC_n3_unif=trim_t2_MC_n3_unif-rep(mu,5)
b_trim_t3_MC_n3_unif=trim_t3_MC_n3_unif-rep(mu,5)
b_trim_t4_MC_n3_unif=trim_t4_MC_n3_unif-rep(mu,5)
b_trim_t5_MC_n3_unif=trim_t5_MC_n3_unif-rep(mu,5)

# EQM

eqm_mean_MC_n3_unif=b_mean_MC_n3_unif^2+sd_mean_MC_n3_unif^2

eqm_med_MC_n3_unif=b_med_MC_n3_unif^2+sd_med_MC_n3_unif^2

eqm_trim_t1_MC_n3_unif=b_trim_t1_MC_n3_unif^2+sd_trim_t1_MC_n3_unif^2
eqm_trim_t2_MC_n3_unif=b_trim_t2_MC_n3_unif^2+sd_trim_t2_MC_n3_unif^2
eqm_trim_t3_MC_n3_unif=b_trim_t3_MC_n3_unif^2+sd_trim_t3_MC_n3_unif^2
eqm_trim_t4_MC_n3_unif=b_trim_t4_MC_n3_unif^2+sd_trim_t4_MC_n3_unif^2
eqm_trim_t5_MC_n3_unif=b_trim_t5_MC_n3_unif^2+sd_trim_t5_MC_n3_unif^2



# Mesures per a n=(10,20,50)


# EQM junts

par(mfrow=c(2,2))
plot(t,eqm_mean_MC_n1_unif,type="l",col=1,lty=1,lwd=2,
     main="",
```

```r
     sub="(Mostra n=10)",
     xlab="Perturbació t",ylab="EQM estimador MC",
     ylim=c(0,max(eqm_med_MC_n1_unif)))

lines(t,eqm_trim_t1_MC_n1_unif,type="l",col=2,lty=2,lwd=2)
lines(t,eqm_trim_t2_MC_n1_unif,type="l",col=3,lty=3,lwd=2)
lines(t,eqm_trim_t3_MC_n1_unif,type="l",col=4,lty=4,lwd=2)
lines(t,eqm_trim_t4_MC_n1_unif,type="l",col=5,lty=5,lwd=2)
lines(t,eqm_trim_t5_MC_n1_unif,type="l",col=6,lty=6,lwd=2)
lines(t,eqm_med_MC_n1_unif,type="l",    col=8,lty=8,lwd=2)

#####
plot(t,eqm_mean_MC_n2_unif,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=20)",
     xlab="Perturbació t",ylab="EQM estimador MC",
     ylim=c(0,max(eqm_med_MC_n2_unif)))
lines(t,eqm_trim_t1_MC_n2_unif,type="l",col=2,lty=2,lwd=2)
lines(t,eqm_trim_t2_MC_n2_unif,type="l",col=3,lty=3,lwd=2)
lines(t,eqm_trim_t3_MC_n2_unif,type="l",col=4,lty=4,lwd=2)
lines(t,eqm_trim_t4_MC_n2_unif,type="l",col=5,lty=5,lwd=2)
lines(t,eqm_trim_t5_MC_n2_unif,type="l",col=6,lty=6,lwd=2)
lines(t,eqm_med_MC_n2_unif,type="l",    col=8,lty=8,lwd=2)
#####

plot(t,eqm_mean_MC_n3_unif,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=50)",
     xlab="Perturbació t",ylab="EQM estimador MC",
     ylim=c(0,max(eqm_med_MC_n2_unif)))

lines(t,eqm_trim_t1_MC_n3_unif,type="l",col=2,lty=2,lwd=2)
lines(t,eqm_trim_t2_MC_n3_unif,type="l",col=3,lty=3,lwd=2)
lines(t,eqm_trim_t3_MC_n3_unif,type="l",col=4,lty=4,lwd=2)
lines(t,eqm_trim_t4_MC_n3_unif,type="l",col=5,lty=5,lwd=2)
lines(t,eqm_trim_t5_MC_n3_unif,type="l",col=6,lty=6,lwd=2)
lines(t,eqm_med_MC_n3_unif,type="l",    col=8,lty=8,lwd=2)



plot(0,xaxt='n',yaxt='n',bty='n',pch='',ylab='',xlab='')
legend("center",
      legend=c(expression(X [n]),

               expression(paste(tm[alpha]," ,",alpha," =  0.01")),
               expression(paste(tm[alpha]," ,",alpha," =  0.05")),
               expression(paste(tm[alpha]," ,",alpha," =  0.1")),
```

```r
                expression(paste(tm[alpha]," ,",alpha," =  0.25")),
                expression(paste(tm[alpha]," ,",alpha," =  0.4")),
                expression(M [x])),
       col=c(1:6,8),lty=c(1:6,8),lwd=2,bty="n")

### TAULES


# Mitjana
round(b_mean_MC_n1_unif,3)
round(eqm_mean_MC_n1_unif,3)
#Mediana
round(b_med_MC_n1_unif,3)
round(eqm_med_MC_n1_unif,3)
#tm alpha=0.01
round(b_trim_t1_MC_n1_unif,3)
round(eqm_trim_t1_MC_n1_unif,3)
#tm alpha=0.05
round(b_trim_t2_MC_n1_unif,3)
round(eqm_trim_t2_MC_n1_unif,3)
#tm alpha=0.1
round(b_trim_t3_MC_n1_unif,3)
round(eqm_trim_t3_MC_n1_unif,3)
#tm alpha=0.25
round(b_trim_t4_MC_n1_unif,3)
round(eqm_trim_t4_MC_n1_unif,3)
#tm alpha=0.4
round(b_trim_t5_MC_n1_unif,3)
round(eqm_trim_t5_MC_n1_unif,3)
##########################

##########################
# Mitjana
round(b_mean_MC_n2_unif,3)
round(eqm_mean_MC_n2_unif,3)
#Mediana
round(b_med_MC_n2_unif,3)
round(eqm_med_MC_n2_unif,3)
#tm alpha=0.01
round(b_trim_t1_MC_n2_unif,3)
round(eqm_trim_t1_MC_n2_unif,3)
#tm alpha=0.05
round(b_trim_t2_MC_n2_unif,3)
round(eqm_trim_t2_MC_n2_unif,3)
#tm alpha=0.1
round(b_trim_t3_MC_n2_unif,3)
round(eqm_trim_t3_MC_n2_unif,3)
```

```
#tm alpha=0.25
round(b_trim_t4_MC_n2_unif,3)
round(eqm_trim_t4_MC_n2_unif,3)
#tm alpha=0.4
round(b_trim_t5_MC_n2_unif,3)
round(eqm_trim_t5_MC_n2_unif,3)




# Mitjana
round(b_mean_MC_n3_unif,3)
round(eqm_mean_MC_n3_unif,3)
#Mediana
round(b_med_MC_n3_unif,3)
round(eqm_med_MC_n3_unif,3)

#tm alpha=0.01
round(b_trim_t1_MC_n3_unif,3)
round(eqm_trim_t1_MC_n3_unif,3)
#tm alpha=0.05
round(b_trim_t2_MC_n3_unif,3)
round(eqm_trim_t2_MC_n3_unif,3)
#tm alpha=0.1
round(b_trim_t3_MC_n3_unif,3)
round(eqm_trim_t3_MC_n3_unif,3)
#tm alpha=0.25
round(b_trim_t4_MC_n3_unif,3)
round(eqm_trim_t4_MC_n3_unif,3)
#tm alpha=0.4
round(b_trim_t5_MC_n3_unif,3)
round(eqm_trim_t5_MC_n3_unif,3)

###
###################################

## Distribució Uniforme (0,1) + Exponencial(1)
# Màxim versemblant (Mitjana mostral, Var. mostral no corregida)




# MIDA MOSTRAL n=10  n[1]




# Maxim mostral (màxim versemblant)
```

```r
mean_mc_n1_unif_exp<-data.frame(
  mean_mc11_n1_unif_exp = numeric(S),
  mean_mc12_n1_unif_exp = numeric(S),
  mean_mc13_n1_unif_exp = numeric(S),
  mean_mc14_n1_unif_exp = numeric(S),
  mean_mc15_n1_unif_exp = numeric(S)
)


# Mediana
med_mc_n1_unif_exp<-data.frame(
  med_mc11_n1_unif_exp = numeric(S),
  med_mc12_n1_unif_exp = numeric(S),
  med_mc13_n1_unif_exp = numeric(S),
  med_mc14_n1_unif_exp = numeric(S),
  med_mc15_n1_unif_exp = numeric(S)
)



# Trimmed mean alpha = t[1]

trim_mc_t1_n1_unif_exp<-data.frame(
  trim_mc11_t1_n1_unif_exp = numeric(S),
  trim_mc12_t1_n1_unif_exp = numeric(S),
  trim_mc13_t1_n1_unif_exp = numeric(S),
  trim_mc14_t1_n1_unif_exp = numeric(S),
  trim_mc15_t1_n1_unif_exp = numeric(S)
)
# Trimmed mean alpha = t[2]

trim_mc_t2_n1_unif_exp<-data.frame(
  trim_mc11_t2_n1_unif_exp = numeric(S),
  trim_mc12_t2_n1_unif_exp = numeric(S),
  trim_mc13_t2_n1_unif_exp = numeric(S),
  trim_mc14_t2_n1_unif_exp = numeric(S),
  trim_mc15_t2_n1_unif_exp = numeric(S)
)
# Trimmed mean alpha = t[3]

trim_mc_t3_n1_unif_exp<-data.frame(
  trim_mc11_t3_n1_unif_exp = numeric(S),
  trim_mc12_t3_n1_unif_exp = numeric(S),
  trim_mc13_t3_n1_unif_exp = numeric(S),
  trim_mc14_t3_n1_unif_exp = numeric(S),
  trim_mc15_t3_n1_unif_exp = numeric(S)
)
```

```r
# Trimmed mean alpha = t[4]

trim_mc_t4_n1_unif_exp<-data.frame(
  trim_mc11_t4_n1_unif_exp = numeric(S),
  trim_mc12_t4_n1_unif_exp = numeric(S),
  trim_mc13_t4_n1_unif_exp = numeric(S),
  trim_mc14_t4_n1_unif_exp = numeric(S),
  trim_mc15_t4_n1_unif_exp = numeric(S)
)
# Trimmed mean alpha = t[5]

trim_mc_t5_n1_unif_exp<-data.frame(
  trim_mc11_t5_n1_unif_exp = numeric(S),
  trim_mc12_t5_n1_unif_exp = numeric(S),
  trim_mc13_t5_n1_unif_exp = numeric(S),
  trim_mc14_t5_n1_unif_exp = numeric(S),
  trim_mc15_t5_n1_unif_exp = numeric(S)
)

# Assignem a un objecte la posició del vector n[1]=10
n1<-n[1]

for (j in 1:length(t))

{
  for(i in 1:S)

  {


    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
    af<-sum(a=="F")
    ag<-sum(a=="G")


    # Simulem mostra s_i
    mostra_af<-runif(af,0,1)
    mostra_ag<-rexp(ag,1)
    mostra<-c(mostra_af,mostra_ag)



    #Càlcul en el data.frame
```

```
    # Màxim versemblant (maxim)
    mean_mc_n1_unif_exp[i,j] = max(mostra)/2

    # Mediana
    med_mc_n1_unif_exp[i,j] = median(mostra)


    # Mitjana truncada en alpha=perturbació t[1] = 0.01
    trim_mc_t1_n1_unif_exp[i,j]= mean(mostra,trim=t[1])
    # Mitjana truncada en alpha=perturbació t[2] = 0.05
    trim_mc_t2_n1_unif_exp[i,j]= mean(mostra,trim=t[2])
    # Mitjana truncada en alpha=perturbació t[3] = 0.1
    trim_mc_t3_n1_unif_exp[i,j]= mean(mostra,trim=t[3])
    # Mitjana truncada en alpha=perturbació t[4] = 0.25
    trim_mc_t4_n1_unif_exp[i,j]= mean(mostra,trim=t[4])
    # Mitjana truncada en alpha=perturbació t[5] = 0.4
    trim_mc_t5_n1_unif_exp[i,j]= mean(mostra,trim=t[5])

  }

}

# Càlcul mesures # Mitjana de les S mostres

# Max
mean_MC_n1_unif_exp=apply(mean_mc_n1_unif_exp,2,mean)
# Mediana
med_MC_n1_unif_exp=apply(med_mc_n1_unif_exp,2,mean)

# Mitjana truncada en alpha=perturbació t[1] = 0.01
trim_t1_MC_n1_unif_exp=apply(trim_mc_t1_n1_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[2] = 0.05
trim_t2_MC_n1_unif_exp=apply(trim_mc_t2_n1_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[3] = 0.1
trim_t3_MC_n1_unif_exp=apply(trim_mc_t3_n1_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[4] = 0.25
trim_t4_MC_n1_unif_exp=apply(trim_mc_t4_n1_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[5] = 0.4
trim_t5_MC_n1_unif_exp=apply(trim_mc_t5_n1_unif_exp,2,mean)




# Desviació no corregida de les S mostres


sd_mean_MC_n1_unif_exp=apply(mean_mc_n1_unif_exp,2,sd)
```

```
sd_mean_MC_n1_unif_exp=sd_mean_MC_n1_unif_exp*sqrt((S-1)/S)

sd_med_MC_n1_unif_exp=apply(med_mc_n1_unif_exp,2,sd)
sd_med_MC_n1_unif_exp=sd_med_MC_n1_unif_exp*sqrt((S-1)/S)

sd_trim_t1_MC_n1_unif_exp=apply(trim_mc_t1_n1_unif_exp,2,sd)
sd_trim_t1_MC_n1_unif_exp=sd_trim_t1_MC_n1_unif_exp*sqrt((S-1)/S)

sd_trim_t2_MC_n1_unif_exp=apply(trim_mc_t2_n1_unif_exp,2,sd)
sd_trim_t2_MC_n1_unif_exp=sd_trim_t2_MC_n1_unif_exp*sqrt((S-1)/S)

sd_trim_t3_MC_n1_unif_exp=apply(trim_mc_t3_n1_unif_exp,2,sd)
sd_trim_t3_MC_n1_unif_exp=sd_trim_t3_MC_n1_unif_exp*sqrt((S-1)/S)

sd_trim_t4_MC_n1_unif_exp=apply(trim_mc_t4_n1_unif_exp,2,sd)
sd_trim_t4_MC_n1_unif_exp=sd_trim_t4_MC_n1_unif_exp*sqrt((S-1)/S)

sd_trim_t5_MC_n1_unif_exp=apply(trim_mc_t5_n1_unif_exp,2,sd)
sd_trim_t5_MC_n1_unif_exp=sd_trim_t5_MC_n1_unif_exp*sqrt((S-1)/S)


# Biaix

b_mean_MC_n1_unif_exp=mean_MC_n1_unif_exp-rep(0.5,5)
b_med_MC_n1_unif_exp=med_MC_n1_unif_exp-rep(0.5,5)

b_trim_t1_MC_n1_unif_exp=trim_t1_MC_n1_unif_exp-rep(0.5,5)
b_trim_t2_MC_n1_unif_exp=trim_t2_MC_n1_unif_exp-rep(0.5,5)
b_trim_t3_MC_n1_unif_exp=trim_t3_MC_n1_unif_exp-rep(0.5,5)
b_trim_t4_MC_n1_unif_exp=trim_t4_MC_n1_unif_exp-rep(0.5,5)
b_trim_t5_MC_n1_unif_exp=trim_t5_MC_n1_unif_exp-rep(0.5,5)

# EQM

eqm_mean_MC_n1_unif_exp=b_mean_MC_n1_unif_exp^2+sd_mean_MC_n1_unif_exp^2

eqm_med_MC_n1_unif_exp=b_med_MC_n1_unif_exp^2+sd_med_MC_n1_unif_exp^2


eqm_trim_t1_MC_n1_unif_exp=b_trim_t1_MC_n1_unif_exp^2+sd_trim_t1_MC_n1_unif_exp^2
eqm_trim_t2_MC_n1_unif_exp=b_trim_t2_MC_n1_unif_exp^2+sd_trim_t2_MC_n1_unif_exp^2
eqm_trim_t3_MC_n1_unif_exp=b_trim_t3_MC_n1_unif_exp^2+sd_trim_t3_MC_n1_unif_exp^2
eqm_trim_t4_MC_n1_unif_exp=b_trim_t4_MC_n1_unif_exp^2+sd_trim_t4_MC_n1_unif_exp^2
eqm_trim_t5_MC_n1_unif_exp=b_trim_t5_MC_n1_unif_exp^2+sd_trim_t5_MC_n1_unif_exp^2


###
```

```
# MIDA MOSTRAL n=20  n[2]


# Maxim mostral (màxim versemblant)

mean_mc_n2_unif_exp<-data.frame(
  mean_mc11_n2_unif_exp = numeric(S),
  mean_mc12_n2_unif_exp = numeric(S),
  mean_mc13_n2_unif_exp = numeric(S),
  mean_mc14_n2_unif_exp = numeric(S),
  mean_mc15_n2_unif_exp = numeric(S)
)


# Mediana
med_mc_n2_unif_exp<-data.frame(
  med_mc11_n2_unif_exp = numeric(S),
  med_mc12_n2_unif_exp = numeric(S),
  med_mc13_n2_unif_exp = numeric(S),
  med_mc14_n2_unif_exp = numeric(S),
  med_mc15_n2_unif_exp = numeric(S)
)



# Trimmed mean alpha = t[1]

trim_mc_t1_n2_unif_exp<-data.frame(
  trim_mc11_t1_n2_unif_exp = numeric(S),
  trim_mc12_t1_n2_unif_exp = numeric(S),
  trim_mc13_t1_n2_unif_exp = numeric(S),
  trim_mc14_t1_n2_unif_exp = numeric(S),
  trim_mc15_t1_n2_unif_exp = numeric(S)
)
# Trimmed mean alpha = t[2]

trim_mc_t2_n2_unif_exp<-data.frame(
  trim_mc11_t2_n2_unif_exp = numeric(S),
  trim_mc12_t2_n2_unif_exp = numeric(S),
  trim_mc13_t2_n2_unif_exp = numeric(S),
  trim_mc14_t2_n2_unif_exp = numeric(S),
  trim_mc15_t2_n2_unif_exp = numeric(S)
)
# Trimmed mean alpha = t[3]

trim_mc_t3_n2_unif_exp<-data.frame(
  trim_mc11_t3_n2_unif_exp = numeric(S),
```

```r
  trim_mc12_t3_n2_unif_exp = numeric(S),
  trim_mc13_t3_n2_unif_exp = numeric(S),
  trim_mc14_t3_n2_unif_exp = numeric(S),
  trim_mc15_t3_n2_unif_exp = numeric(S)
)
# Trimmed mean alpha = t[4]

trim_mc_t4_n2_unif_exp<-data.frame(
  trim_mc11_t4_n2_unif_exp = numeric(S),
  trim_mc12_t4_n2_unif_exp = numeric(S),
  trim_mc13_t4_n2_unif_exp = numeric(S),
  trim_mc14_t4_n2_unif_exp = numeric(S),
  trim_mc15_t4_n2_unif_exp = numeric(S)
)
# Trimmed mean alpha = t[5]

trim_mc_t5_n2_unif_exp<-data.frame(
  trim_mc11_t5_n2_unif_exp = numeric(S),
  trim_mc12_t5_n2_unif_exp = numeric(S),
  trim_mc13_t5_n2_unif_exp = numeric(S),
  trim_mc14_t5_n2_unif_exp = numeric(S),
  trim_mc15_t5_n2_unif_exp = numeric(S)
)

# Assignem a un objecte la posició del vector n[2]=20
n1<-n[2]

for (j in 1:length(t))

{
  for(i in 1:S)

  {


    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
    af<-sum(a=="F")
    ag<-sum(a=="G")


    # Simulem mostra s_i
    mostra_af<-runif(af,0,1)
    mostra_ag<-rexp(ag,1)
    mostra<-c(mostra_af,mostra_ag)
```

```
    #Càlcul en el data.frame

    # Màxim versemblant (maxim)
    mean_mc_n2_unif_exp[i,j] = max(mostra)/2

    # Mediana
    med_mc_n2_unif_exp[i,j] = median(mostra)


    # Mitjana truncada en alpha=perturbació t[1] = 0.01
    trim_mc_t1_n2_unif_exp[i,j]= mean(mostra,trim=t[1])
    # Mitjana truncada en alpha=perturbació t[2] = 0.05
    trim_mc_t2_n2_unif_exp[i,j]= mean(mostra,trim=t[2])
    # Mitjana truncada en alpha=perturbació t[3] = 0.1
    trim_mc_t3_n2_unif_exp[i,j]= mean(mostra,trim=t[3])
    # Mitjana truncada en alpha=perturbació t[4] = 0.25
    trim_mc_t4_n2_unif_exp[i,j]= mean(mostra,trim=t[4])
    # Mitjana truncada en alpha=perturbació t[5] = 0.4
    trim_mc_t5_n2_unif_exp[i,j]= mean(mostra,trim=t[5])

  }

}

# Càlcul mesures # Mitjana de les S mostres

# Max
mean_MC_n2_unif_exp=apply(mean_mc_n2_unif_exp,2,mean)
# Mediana
med_MC_n2_unif_exp=apply(med_mc_n2_unif_exp,2,mean)

# Mitjana truncada en alpha=perturbació t[1] = 0.01
trim_t1_MC_n2_unif_exp=apply(trim_mc_t1_n2_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[2] = 0.05
trim_t2_MC_n2_unif_exp=apply(trim_mc_t2_n2_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[3] = 0.1
trim_t3_MC_n2_unif_exp=apply(trim_mc_t3_n2_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[4] = 0.25
trim_t4_MC_n2_unif_exp=apply(trim_mc_t4_n2_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[5] = 0.4
trim_t5_MC_n2_unif_exp=apply(trim_mc_t5_n2_unif_exp,2,mean)
```

```
# Desviació no corregida de les S mostres


sd_mean_MC_n2_unif_exp=apply(mean_mc_n2_unif_exp,2,sd)
sd_mean_MC_n2_unif_exp=sd_mean_MC_n2_unif_exp*sqrt((S-1)/S)

sd_med_MC_n2_unif_exp=apply(med_mc_n2_unif_exp,2,sd)
sd_med_MC_n2_unif_exp=sd_med_MC_n2_unif_exp*sqrt((S-1)/S)


sd_trim_t1_MC_n2_unif_exp=apply(trim_mc_t1_n2_unif_exp,2,sd)
sd_trim_t1_MC_n2_unif_exp=sd_trim_t1_MC_n2_unif_exp*sqrt((S-1)/S)

sd_trim_t2_MC_n2_unif_exp=apply(trim_mc_t2_n2_unif_exp,2,sd)
sd_trim_t2_MC_n2_unif_exp=sd_trim_t2_MC_n2_unif_exp*sqrt((S-1)/S)

sd_trim_t3_MC_n2_unif_exp=apply(trim_mc_t3_n2_unif_exp,2,sd)
sd_trim_t3_MC_n2_unif_exp=sd_trim_t3_MC_n2_unif_exp*sqrt((S-1)/S)

sd_trim_t4_MC_n2_unif_exp=apply(trim_mc_t4_n2_unif_exp,2,sd)
sd_trim_t4_MC_n2_unif_exp=sd_trim_t4_MC_n2_unif_exp*sqrt((S-1)/S)

sd_trim_t5_MC_n2_unif_exp=apply(trim_mc_t5_n2_unif_exp,2,sd)
sd_trim_t5_MC_n2_unif_exp=sd_trim_t5_MC_n2_unif_exp*sqrt((S-1)/S)


# Biaix

b_mean_MC_n2_unif_exp=mean_MC_n2_unif_exp-rep(0.5,5)
b_med_MC_n2_unif_exp=med_MC_n2_unif_exp-rep(0.5,5)

b_trim_t1_MC_n2_unif_exp=trim_t1_MC_n2_unif_exp-rep(0.5,5)
b_trim_t2_MC_n2_unif_exp=trim_t2_MC_n2_unif_exp-rep(0.5,5)
b_trim_t3_MC_n2_unif_exp=trim_t3_MC_n2_unif_exp-rep(0.5,5)
b_trim_t4_MC_n2_unif_exp=trim_t4_MC_n2_unif_exp-rep(0.5,5)
b_trim_t5_MC_n2_unif_exp=trim_t5_MC_n2_unif_exp-rep(0.5,5)

# EQM

eqm_mean_MC_n2_unif_exp=b_mean_MC_n2_unif_exp^2+sd_mean_MC_n2_unif_exp^2

eqm_med_MC_n2_unif_exp=b_med_MC_n2_unif_exp^2+sd_med_MC_n2_unif_exp^2


eqm_trim_t1_MC_n2_unif_exp=b_trim_t1_MC_n2_unif_exp^2+sd_trim_t1_MC_n2_unif_exp^2
eqm_trim_t2_MC_n2_unif_exp=b_trim_t2_MC_n2_unif_exp^2+sd_trim_t2_MC_n2_unif_exp^2
eqm_trim_t3_MC_n2_unif_exp=b_trim_t3_MC_n2_unif_exp^2+sd_trim_t3_MC_n2_unif_exp^2
```

```
eqm_trim_t4_MC_n2_unif_exp=b_trim_t4_MC_n2_unif_exp^2+sd_trim_t4_MC_n2_unif_exp^2
eqm_trim_t5_MC_n2_unif_exp=b_trim_t5_MC_n2_unif_exp^2+sd_trim_t5_MC_n2_unif_exp^2



### CAS N=50

# Maxim mostral (màxim versemblant)

mean_mc_n3_unif_exp<-data.frame(
  mean_mc11_n3_unif_exp = numeric(S),
  mean_mc12_n3_unif_exp = numeric(S),
  mean_mc13_n3_unif_exp = numeric(S),
  mean_mc14_n3_unif_exp = numeric(S),
  mean_mc15_n3_unif_exp = numeric(S)
)


# Mediana
med_mc_n3_unif_exp<-data.frame(
  med_mc11_n3_unif_exp = numeric(S),
  med_mc12_n3_unif_exp = numeric(S),
  med_mc13_n3_unif_exp = numeric(S),
  med_mc14_n3_unif_exp = numeric(S),
  med_mc15_n3_unif_exp = numeric(S)
)



# Trimmed mean alpha = t[1]

trim_mc_t1_n3_unif_exp<-data.frame(
  trim_mc11_t1_n3_unif_exp = numeric(S),
  trim_mc12_t1_n3_unif_exp = numeric(S),
  trim_mc13_t1_n3_unif_exp = numeric(S),
  trim_mc14_t1_n3_unif_exp = numeric(S),
  trim_mc15_t1_n3_unif_exp = numeric(S)
)
# Trimmed mean alpha = t[2]

trim_mc_t2_n3_unif_exp<-data.frame(
  trim_mc11_t2_n3_unif_exp = numeric(S),
  trim_mc12_t2_n3_unif_exp = numeric(S),
  trim_mc13_t2_n3_unif_exp = numeric(S),
  trim_mc14_t2_n3_unif_exp = numeric(S),
  trim_mc15_t2_n3_unif_exp = numeric(S)
)
```

```r
# Trimmed mean alpha = t[3]

trim_mc_t3_n3_unif_exp<-data.frame(
  trim_mc11_t3_n3_unif_exp = numeric(S),
  trim_mc12_t3_n3_unif_exp = numeric(S),
  trim_mc13_t3_n3_unif_exp = numeric(S),
  trim_mc14_t3_n3_unif_exp = numeric(S),
  trim_mc15_t3_n3_unif_exp = numeric(S)
)
# Trimmed mean alpha = t[4]

trim_mc_t4_n3_unif_exp<-data.frame(
  trim_mc11_t4_n3_unif_exp = numeric(S),
  trim_mc12_t4_n3_unif_exp = numeric(S),
  trim_mc13_t4_n3_unif_exp = numeric(S),
  trim_mc14_t4_n3_unif_exp = numeric(S),
  trim_mc15_t4_n3_unif_exp = numeric(S)
)
# Trimmed mean alpha = t[5]

trim_mc_t5_n3_unif_exp<-data.frame(
  trim_mc11_t5_n3_unif_exp = numeric(S),
  trim_mc12_t5_n3_unif_exp = numeric(S),
  trim_mc13_t5_n3_unif_exp = numeric(S),
  trim_mc14_t5_n3_unif_exp = numeric(S),
  trim_mc15_t5_n3_unif_exp = numeric(S)
)

# Assignem a un objecte la posició del vector n[3]=50
n1<-n[3]

for (j in 1:length(t))

{
  for(i in 1:S)

  {


    # Pas Previ a simulació de cada s_i

    set.seed(random[i])
    a<-sample(c("F","G"),n1,prob=c((1-t[j]),(t[j])),replace=T)
    af<-sum(a=="F")
    ag<-sum(a=="G")
```

```r
    # Simulem mostra s_i
    mostra_af<-runif(af,0,1)
    mostra_ag<-rexp(ag,1)
    mostra<-c(mostra_af,mostra_ag)



    #Càlcul en el data.frame

    # Màxim versemblant (maxim)
    mean_mc_n3_unif_exp[i,j] = max(mostra)/2

    # Mediana
    med_mc_n3_unif_exp[i,j] = median(mostra)


    # Mitjana truncada en alpha=perturbació t[1] = 0.01
    trim_mc_t1_n3_unif_exp[i,j]= mean(mostra,trim=t[1])
    # Mitjana truncada en alpha=perturbació t[2] = 0.05
    trim_mc_t2_n3_unif_exp[i,j]= mean(mostra,trim=t[2])
    # Mitjana truncada en alpha=perturbació t[3] = 0.1
    trim_mc_t3_n3_unif_exp[i,j]= mean(mostra,trim=t[3])
    # Mitjana truncada en alpha=perturbació t[4] = 0.25
    trim_mc_t4_n3_unif_exp[i,j]= mean(mostra,trim=t[4])
    # Mitjana truncada en alpha=perturbació t[5] = 0.4
    trim_mc_t5_n3_unif_exp[i,j]= mean(mostra,trim=t[5])

  }

}

# Càlcul mesures # Mitjana de les S mostres

# Max
mean_MC_n3_unif_exp=apply(mean_mc_n3_unif_exp,2,mean)
# Mediana
med_MC_n3_unif_exp=apply(med_mc_n3_unif_exp,2,mean)

# Mitjana truncada en alpha=perturbació t[1] = 0.01
trim_t1_MC_n3_unif_exp=apply(trim_mc_t1_n3_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[2] = 0.05
trim_t2_MC_n3_unif_exp=apply(trim_mc_t2_n3_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[3] = 0.1
trim_t3_MC_n3_unif_exp=apply(trim_mc_t3_n3_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[4] = 0.25
trim_t4_MC_n3_unif_exp=apply(trim_mc_t4_n3_unif_exp,2,mean)
# Mitjana truncada en alpha=perturbació t[5] = 0.4
```

```
trim_t5_MC_n3_unif_exp=apply(trim_mc_t5_n3_unif_exp,2,mean)


# Desviació no corregida de les S mostres


sd_mean_MC_n3_unif_exp=apply(mean_mc_n3_unif_exp,2,sd)
sd_mean_MC_n3_unif_exp=sd_mean_MC_n3_unif_exp*sqrt((S-1)/S)

sd_med_MC_n3_unif_exp=apply(med_mc_n3_unif_exp,2,sd)
sd_med_MC_n3_unif_exp=sd_med_MC_n3_unif_exp*sqrt((S-1)/S)


sd_trim_t1_MC_n3_unif_exp=apply(trim_mc_t1_n3_unif_exp,2,sd)
sd_trim_t1_MC_n3_unif_exp=sd_trim_t1_MC_n3_unif_exp*sqrt((S-1)/S)

sd_trim_t2_MC_n3_unif_exp=apply(trim_mc_t2_n3_unif_exp,2,sd)
sd_trim_t2_MC_n3_unif_exp=sd_trim_t2_MC_n3_unif_exp*sqrt((S-1)/S)

sd_trim_t3_MC_n3_unif_exp=apply(trim_mc_t3_n3_unif_exp,2,sd)
sd_trim_t3_MC_n3_unif_exp=sd_trim_t3_MC_n3_unif_exp*sqrt((S-1)/S)

sd_trim_t4_MC_n3_unif_exp=apply(trim_mc_t4_n3_unif_exp,2,sd)
sd_trim_t4_MC_n3_unif_exp=sd_trim_t4_MC_n3_unif_exp*sqrt((S-1)/S)

sd_trim_t5_MC_n3_unif_exp=apply(trim_mc_t5_n3_unif_exp,2,sd)
sd_trim_t5_MC_n3_unif_exp=sd_trim_t5_MC_n3_unif_exp*sqrt((S-1)/S)


# Biaix

b_mean_MC_n3_unif_exp=mean_MC_n3_unif_exp-rep(0.5,5)
b_med_MC_n3_unif_exp=med_MC_n3_unif_exp-rep(0.5,5)

b_trim_t1_MC_n3_unif_exp=trim_t1_MC_n3_unif_exp-rep(0.5,5)
b_trim_t2_MC_n3_unif_exp=trim_t2_MC_n3_unif_exp-rep(0.5,5)
b_trim_t3_MC_n3_unif_exp=trim_t3_MC_n3_unif_exp-rep(0.5,5)
b_trim_t4_MC_n3_unif_exp=trim_t4_MC_n3_unif_exp-rep(0.5,5)
b_trim_t5_MC_n3_unif_exp=trim_t5_MC_n3_unif_exp-rep(0.5,5)

# EQM

eqm_mean_MC_n3_unif_exp=b_mean_MC_n3_unif_exp^2+sd_mean_MC_n3_unif_exp^2

eqm_med_MC_n3_unif_exp=b_med_MC_n3_unif_exp^2+sd_med_MC_n3_unif_exp^2

eqm_trim_t1_MC_n3_unif_exp=b_trim_t1_MC_n3_unif_exp^2+sd_trim_t1_MC_n3_unif_exp^2
```

```
eqm_trim_t2_MC_n3_unif_exp=b_trim_t2_MC_n3_unif_exp^2+sd_trim_t2_MC_n3_unif_exp^2
eqm_trim_t3_MC_n3_unif_exp=b_trim_t3_MC_n3_unif_exp^2+sd_trim_t3_MC_n3_unif_exp^2
eqm_trim_t4_MC_n3_unif_exp=b_trim_t4_MC_n3_unif_exp^2+sd_trim_t4_MC_n3_unif_exp^2
eqm_trim_t5_MC_n3_unif_exp=b_trim_t5_MC_n3_unif_exp^2+sd_trim_t5_MC_n3_unif_exp^2


#save.image("unif_exp_bona.Rdata")
#load("unif_exp_bona.Rdata")


## Comparació EQM
#  mediana i trimmed (mitjana ordinària en alpha=0.01)


par(mfrow=c(2,2))
plot(t,eqm_trim_t1_MC_n1_unif_exp,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=10)",
     xlab="Perturbació t",ylab="EQM estimador MC")


lines(t,eqm_trim_t2_MC_n1_unif_exp,type="l",col=2,lty=2,lwd=2)
lines(t,eqm_trim_t3_MC_n1_unif_exp,type="l",col=3,lty=3,lwd=2)
lines(t,eqm_trim_t4_MC_n1_unif_exp,type="l",col=4,lty=4,lwd=2)
lines(t,eqm_trim_t5_MC_n1_unif_exp,type="l",col=5,lty=5,lwd=2)
lines(t,eqm_med_MC_n1_unif_exp,type="l",    col=6,lty=6,lwd=2)

#####
plot(t,eqm_trim_t1_MC_n2_unif_exp,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=20)",
     xlab="Perturbació t",ylab="EQM estimador MC")




lines(t,eqm_trim_t2_MC_n2_unif_exp,type="l",col=2,lty=2,lwd=2)
lines(t,eqm_trim_t3_MC_n2_unif_exp,type="l",col=3,lty=3,lwd=2)
lines(t,eqm_trim_t4_MC_n2_unif_exp,type="l",col=4,lty=4,lwd=2)
lines(t,eqm_trim_t5_MC_n2_unif_exp,type="l",col=5,lty=5,lwd=2)
lines(t,eqm_med_MC_n2_unif_exp,type="l",    col=6,lty=6,lwd=2)
#####

plot(t,eqm_trim_t1_MC_n3_unif_exp,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=50)",
     xlab="Perturbació t",ylab="EQM estimador MC")
```

```r
lines(t,eqm_trim_t2_MC_n3_unif_exp,type="l",col=2,lty=2,lwd=2)
lines(t,eqm_trim_t3_MC_n3_unif_exp,type="l",col=3,lty=3,lwd=2)
lines(t,eqm_trim_t4_MC_n3_unif_exp,type="l",col=4,lty=4,lwd=2)
lines(t,eqm_trim_t5_MC_n3_unif_exp,type="l",col=5,lty=5,lwd=2)
lines(t,eqm_med_MC_n3_unif_exp,type="l",    col=6,lty=6,lwd=2)



plot(0,xaxt='n',yaxt='n',bty='n',pch='',ylab='',xlab='')
legend("center",
       legend=

    c(expression(paste(tm[alpha]," ,",alpha," =  0.01")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.05")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.1")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.25")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.4")),
                 expression(M [x])),
       col=c(1:6),lty=c(1:6),lwd=2,bty="n")



# Comparació estimadors amb Biaix

par(mfrow=c(2,2))
plot(t,b_mean_MC_n1_unif_exp,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=10)",
     xlab="Perturbació t",ylab="Biaix estimador MC",
     ylim=c(-0.05,max(b_mean_MC_n1_unif_exp)+0.05))


lines(t,b_trim_t1_MC_n1_unif_exp,type="l",col=2,lty=2,lwd=2)
lines(t,b_trim_t2_MC_n1_unif_exp,type="l",col=3,lty=3,lwd=2)
lines(t,b_trim_t3_MC_n1_unif_exp,type="l",col=4,lty=4,lwd=2)
lines(t,b_trim_t4_MC_n1_unif_exp,type="l",col=5,lty=5,lwd=2)
lines(t,b_trim_t5_MC_n1_unif_exp,type="l",col=6,lty=6,lwd=2)
lines(t,b_med_MC_n1_unif_exp,type="l",    col=8,lty=8,lwd=2)

#####
plot(t,b_mean_MC_n2_unif_exp,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=20)",
     xlab="Perturbació t",ylab="Biaix estimador MC",
     ylim=c(-0.05,max(b_mean_MC_n2_unif_exp)+0.05))
```

```
lines(t,b_trim_t1_MC_n2_unif_exp,type="l",col=2,lty=2,lwd=2)
lines(t,b_trim_t2_MC_n2_unif_exp,type="l",col=3,lty=3,lwd=2)
lines(t,b_trim_t3_MC_n2_unif_exp,type="l",col=4,lty=4,lwd=2)
lines(t,b_trim_t4_MC_n2_unif_exp,type="l",col=5,lty=5,lwd=2)
lines(t,b_trim_t5_MC_n2_unif_exp,type="l",col=6,lty=6,lwd=2)
lines(t,b_med_MC_n2_unif_exp,type="l",    col=8,lty=8,lwd=2)
#####

plot(t,b_mean_MC_n3_unif_exp,type="l",col=1,lty=1,lwd=2,
     main="",
     sub="(Mostra n=50)",
     xlab="Perturbació t",ylab="Biaix estimador MC",
     ylim=c(-0.05,max(b_mean_MC_n3_unif_exp)+0.05))

lines(t,b_trim_t1_MC_n3_unif_exp,type="l",col=2,lty=2,lwd=2)
lines(t,b_trim_t2_MC_n3_unif_exp,type="l",col=3,lty=3,lwd=2)
lines(t,b_trim_t3_MC_n3_unif_exp,type="l",col=4,lty=4,lwd=2)
lines(t,b_trim_t4_MC_n3_unif_exp,type="l",col=5,lty=5,lwd=2)
lines(t,b_trim_t5_MC_n3_unif_exp,type="l",col=6,lty=6,lwd=2)
lines(t,b_med_MC_n3_unif_exp,type="l",    col=8,lty=8,lwd=2)



plot(0,xaxt='n',yaxt='n',bty='n',pch='',ylab='',xlab='')
legend("center",
       legend=c(expression(X [(n)]),

                 expression(paste(tm[alpha]," ,",alpha," =  0.01")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.05")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.1")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.25")),
                 expression(paste(tm[alpha]," ,",alpha," =  0.4")),
                 expression(M [x])),
       col=c(1:6,8),lty=c(1:6,8),lwd=2,bty="n")
```