

RAPPORT DU PROJET INNOVATION : ESTIMATIONS BAYÉSIENNES

Encadré par : Pr. CHALLET Damien

HACHICHA Med Amine

MIRI Elyes

DHAHA Firas

CentraleSupélec Cours Centralien

Promotion 2020

8 Janvier 2019

Table des matières

1	Introduction :	2
1.1	Les Faits Stylisés :	2
1.2	Le modèle d'agents :	2
1.3	La language Stan et l'estimation bayésienne :	3
1.3.1	Exemple d'un program en Stan :	3
2	Modèle de Vikram et Sinha :	4
2.1	Hypothèses :	4
3	Modèle des fourmis de Alan Kirman :	6
3.1	Hypothèses :	6
3.2	Essais :	7
4	Minority Games :	8
4.1	Formulation Mathématiques :	9
4.2	Modèle Simplifié :	9
4.3	Simulation et inférence des paramètres du modèle des Minority Games :	10
4.3.1	Explication intuitive du cas $\Gamma = 30$:	16
5	Bibliographie :	16

1 Introduction :

1.1 Les Faits Stylisés :

L'étude des marchés financiers montrent qu'ils sont fortement influencés par les événements politiques et économiques de leurs environnements. Ainsi, deux marchés qui règnent dans deux environnements différents doivent être indépendants dans leurs comportements i.e. ils doivent exhiber des propriétés statistiques différentes. Cependant, une étude plus approfondie des marchés montre l'existence d'un ensemble de facteurs communs entre plusieurs les marchés : c'est ce qu'on appelle **Les Faits Stylisés/Stylized facts**.

Voilà quelques faits stylisés :

- **Absence d'auto-corrélation** : l'auto-corrélation entre les rendements des prix est souvent insignifiantes, sauf sur des intervalles de temps assez petits.
- **Asymétrie entre les gains et les pertes** : On observe des descentes importantes dans les rendements des prix mais beaucoup plus intenses que les hausses.
- **Intermittence** : Les rendements des prix montrent, sur tout échelle temporel, un haut degré de variabilité. Ceci est quantifié par la présence de éclats irréguliers dans les séries chronologiques de plusieurs estimateurs de volatilité

Ces faits, étant néanmoins qualitatifs, représentent des contraintes pour la modélisation des marchés financiers. Actuellement, il n'existe pas de modèle de marché capable de satisfaire à toutes ces contraintes. Ainsi l'une des enjeux du moment est de trouver des modèles qui satisfont le maximum de ces contraintes.

1.2 Le modèle d'agents :

Le modèle d'agents est un moyen pour modéliser les marchés financiers. Il s'agit d'une représentation sous-forme de système multi-agents, c'est à dire, un ensemble d'agents qui interagissent entre eux et avec leur milieu selon certaines relations. Ce modèle est constitué de 5 composants majeurs :

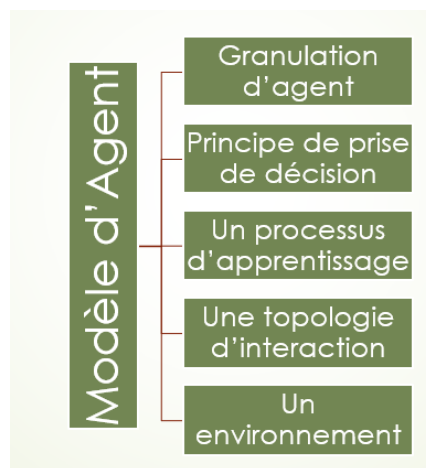


FIGURE 1 – Structure d'un modèle d'agents

- **Granulation d'agents** : l'ensemble des agents partiellement autonome en interaction.
- **Principe de prise de décision** : La condition de passage d'un état t à un état $t+1$ en effectuant une décision.
- **Un processus d'apprentissage** : Quoi que la décision prise contient de l'aléatoire, elle doit aussi dépendre des ensembles des décisions prise précédemment. Ainsi, l'agent apprend de l'information déjà acquise.
- **Une topologie d'interaction** : les décisions ne doivent pas être complètement autonomes mais dépendantes de l'ensemble des décisions prises par les autres agents.
- **Un environnement** : l'interaction ne se limite pas aux agents mais l'environnement du processus est aussi un facteur dans la prise de décision.

1.3 La language Stan et l'estimation bayésienne :

Stan est un langage de programmation probabiliste codé en C++, il est utilisé pour effectuer des inférences statistiques et surtout bayésiennes. Stan permet d'inférer une série de paramètres non-observables $(\theta_1, \theta_2, \dots, \theta_N)$ sachant une distribution des données observées (x_1, x_2, \dots, x_K) supposant un Prior $p(x, \theta) = p(x|\theta)p(\theta)$ selon la formule de Bayes :

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$$

avec

$$p(x) = \int_{\theta} p(x|\theta)p(\theta) d\theta$$

1.3.1 Exemple d'un program en Stan :

Voici un exemple d'un program écrit avec la syntaxe de Stan (figure 2), la structure est constituée de plusieurs blocs chacun ayant une fonction :

- **Functions** : Ce bloc permet de déclarer des fonctions qui seront appelées ultérieurement dans le programme, c'est souvent les distributions particulières qui sont déclarées ici.
- **Data** : Ce bloc permet de déclarer les données observées qui vont nous permettre d'effectuer l'inférence. Il faut préciser les bornes ainsi que les types de données utilisées.
- **Parameters** : Ce bloc permet de déclarer les paramètres qu'on cherche à inférer. Il faut préciser les bornes ainsi que les types des paramètres souhaités.
- **Transformed parameters/data** : les blocs "transformed" permettent de modifier les paramètres ou les données (par exemple pour des raisons de simplifications).
- **Model** : Ce bloc est centrale pour le programme car il contient la log-densité du Prior supposée $\log(p(x, \theta))$, ceci permet d'effectuer le processus d'inférence selon le modèle bayésien. L'intérêt de mettre une log densité provient du fait que Stan opère par la méthode du *Hamiltonien Monte-Carlo* qui fait intervenir la log densité.

```

4 Kir_code = """
5 functions {
6     real dist_lpdf(real y, real Fj, real N, real eps, real delta) {
7         real P1;
8         real P2;
9         P1=(1-Fj/N)*(eps+(1-delta)*Fj/(N-1));
10        P2=(Fj/N)*(eps+(1-delta)*(N-Fj)/(N-1));
11        if (y !=Fj && (y !=Fj-1) && (y!=Fj+1))
12            reject("dist_lpdf: illegal value for y: ", y);
13        if (y == Fj-1) return log(P2);
14        else if (y == Fj+1) return log(P1);
15        else return log(1-(P1+P2));
16    }
17 }
18 data {
19     int < lower =2> N;
20     int < lower =0> T;
21     vector [T] F;
22 }
23 parameters {
24     real <lower =0, upper=1> eps;
25     real <lower =0, upper =1> delta ;
26 }
27 transformed parameters {
28
29 }
30 model {
31     eps ~ beta(1,4) ;
32     delta ~ beta(2,2) ;
33     for (j in 2:T)
34         F[j]~dist(F[j-1],N,eps,delta);
35 }
36 """

```

FIGURE 2 – Syntaxe de Stan

2 Modèle de Vikram et Sinha :

Pour un premier essai de codage sur PyStan (interface de Stan sur Python), on s'est basé sur le code du modèle de Vikram et Sinha fourni dans l'article 'Reality-check for Econophysics : Likelihood-based fitting of physics-inspired market models to empirical data' de Bertschinger, Mozzhorin et Sinha [1].

2.1 Hypothèses :

Le modèle du marché dans le modèle VS comporte N traders ; chaque trader décide d'acheter ($S_i(t) = 1$) vendre ($S_i(t) = -1$) ou rester inactif ($S_i(t) = 0$). La demande normalisée de tous les traders est donnée par :

$$M_t = \frac{1}{N} \sum_{i=1}^N S_i(t)$$

et le prix suit la formule

$$p_{t+1} = \frac{1 + M_t}{1 - M_t}$$

L'évolution du système suit les lois suivantes :

$$\langle p_t \rangle_\tau = (1 - \tau)p_t + \tau \langle p_{t-1} \rangle_\tau$$

$$P(|S_i(t)| = 1) = e^{-\mu |\log(\frac{p_t}{\langle p_t \rangle_\tau})|}$$

$$r_{t+1} = \log \frac{p_{t+1}}{p_t} \approx 2M_t \sim N(0, 4\sigma_{max}^2 P(|S_i(t)| = 1))$$

On voit donc que le système est régi par le choix de 3 paramètres $(\tau, \mu, \sigma_{max})$:

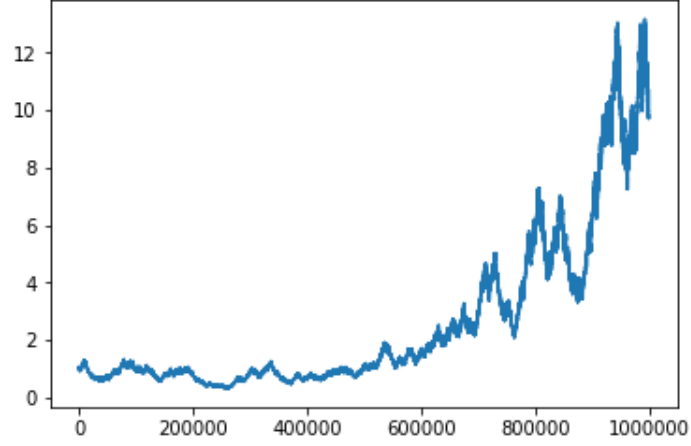


FIGURE 3 – Simulation de l'évolution du prix pour : $\tau = 0.999$; $\mu = 100$; $\sigma_{max} = 0.01$; $p_0 = 1$

Remarque : μ est grand pour empêcher la stagnation du prix (fonction exponentielle).

Un code Stan prêt à être fourni à la fin du document Reality-check for Econophysics et nous renvoie la courbe suivante pour le paramètre μ . La courbe est décalée de la valeur espérée, ceci montre que le fitting d'un modèle n'est pas toujours parfait, et aussi que pas n'importe quel modèle, à priori, pourrait être 'fitté'.

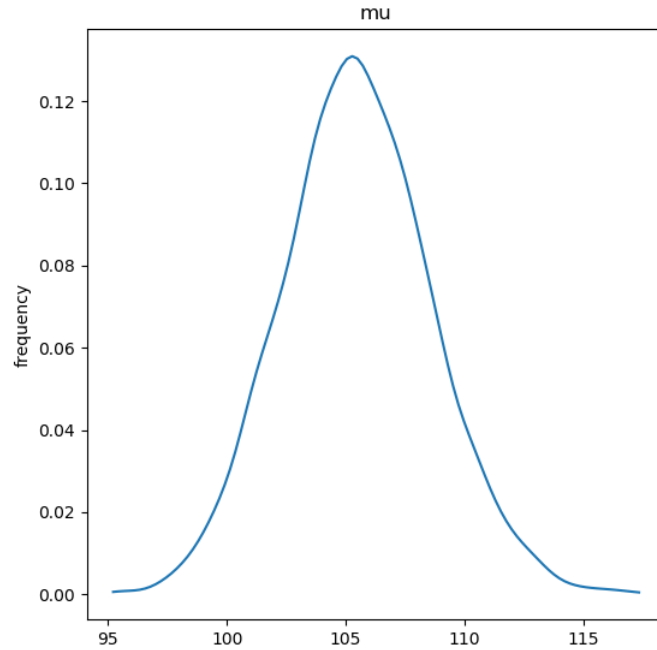


FIGURE 4 – Distribution Posterior de μ

3 Modèle des fourmis de Alan Kirman :

Des entomologistes ont remarqué que lorsque deux sources de nourriture identiques ('noire' et 'blanche') sont présentées à des fourmis, celles-ci exploitent l'une d'elles plus que l'autre. De plus, elles revenaient vers la source négligée de temps en temps. Ils ont fini par conclure que cela ne pourrait s'expliquer seulement par l'analyse du comportement individuel de chaque fourmi, mais aussi par l'interaction entre différents individus. L'article 'Ants, rationality and recruitment' [3] de Kirman reprend cette situation puisque des phénomènes similaires sont observés dans les marchés, d'où son importance pour les économistes.

3.1 Hypothèses :

- On considère qu'il y a un nombre total N de fourmis.
- Modèle de tandem-running : Le tandem-running est un comportement de recrutement. Il n'est pas basé sur l'utilisation de phéromones de piste, mais sur le contact direct entre une fourmi guide $F1$ et une fourmi suiveuse $F2$ où il y a une probabilité non nulle que l'une d'elles suit l'autre.

$$P(\text{« } F1 \text{ convertie en la couleur de } F2 \text{ »}) = 1 - \delta$$

$$P(\text{« } F1 \text{ change de couleur spontanément »}) = \epsilon$$

Cette deuxième probabilité est introduite pour que les états 0 et N fourmis noires ne soient pas absorbants. Le nombre de fourmis noires (indexé par les paliers de temps $t_0 \Rightarrow$ discret) est une chaîne de Markov discrète dans l'espace des états $[0, N]$. Le théorème des matrices stochastiques nous assure l'existence d'une distribution d'équilibre vers laquelle convergera le système. Ainsi d'un nombre de fourmis noires k à l'instant t on peut passer à l'instant $t+1$ à soit $k+1$ soit $k-1$ fourmis noires (rester avec k fourmis noires est aussi possible). On donne ci-dessous les probabilités de passage :

$$\begin{aligned}
 & \begin{array}{l} \nearrow k+1 \text{ with probability } p_1 = P(k, k+1) \\ k \\ \searrow k-1 \text{ with probability } p_2 = P(k, k-1) \end{array} \\
 & \qquad \qquad \qquad = \left(1 - \frac{k}{N}\right) \left(\epsilon + (1 - \delta) \frac{k}{N-1}\right) \\
 & \qquad \qquad \qquad = \frac{k}{N} \left(\epsilon + (1 - \delta) \frac{N-k}{N-1}\right).
 \end{aligned}$$

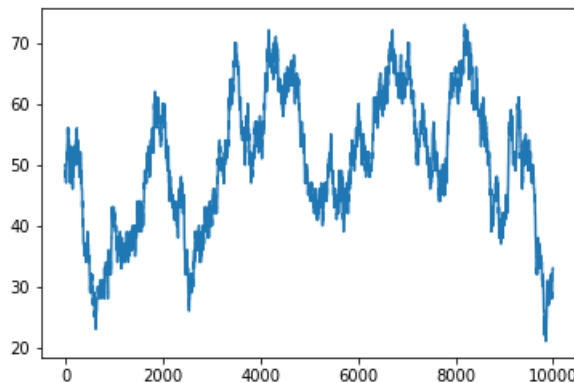


FIGURE 5 – Evolution du nombre de fourmis noirs pour $N=100$, $\epsilon=0.05$, $\delta=0.4$

On a ensuite codé le modèle sur Stan. Ci-dessous est le code utilisé.

```

4 Kir_code = ""
5 functions {
6     real dist_lpdf(real y, real Fj, real N, real eps, real delta) {
7         real P1;
8         real P2;
9         P1=(1-Fj/N)*(eps+(1-delta)*Fj/(N-1));
10        P2=(Fj/N)*(eps+(1-delta)*(N-Fj)/(N-1));
11        if (y !=Fj && (y !=Fj-1) && (y!=Fj+1))
12            reject("dist_lpdf: illegal value for y: ", y);
13        if (y == Fj-1) return log(P2);
14        else if (y == Fj+1) return log(P1);
15        else return log(1-(P1+P2));
16    }
17 }
18 data {
19     int < lower =2> N;
20     int < lower =0> T;
21     vector [T] F;
22 }
23 parameters {
24     real <lower =0, upper=1> eps;
25     real <lower =0, upper =1> delta ;
26 }
27 transformed parameters {
28
29 }
30 model {
31     eps ~ beta(1,4) ;
32     delta ~ beta(2,2) ;
33     for (j in 2:T)
34         F[j]~dist(F[j-1],N,eps,delta);
35 }
36 ""
--

```

A l'aide de tutoriels trouvés sur internet et des discussions sur les forums, on a trouvé comment ajouter le bloc functions (en comparaison avec les blocs de base évoqués au début) pour pouvoir poser nos propres distributions de probabilité et les utiliser (On peut noter qu'il faut utiliser les log-probabilité pour pouvoir les implémenter en langage Stan).

3.2 Essais :

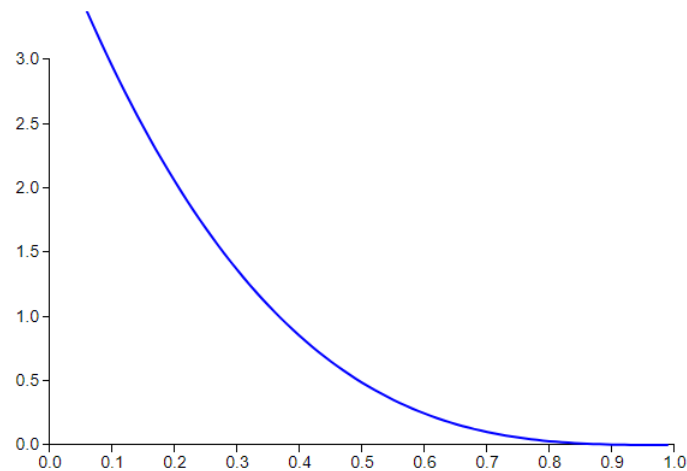


FIGURE 6 – Prior beta(1,4) pour ϵ (Valeur réelle fixée = 0.05)

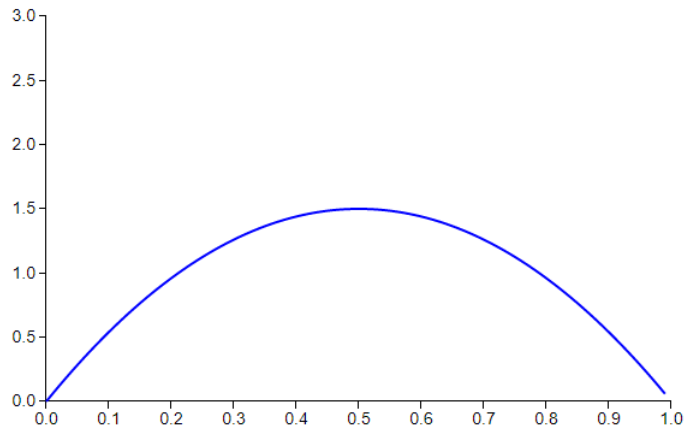
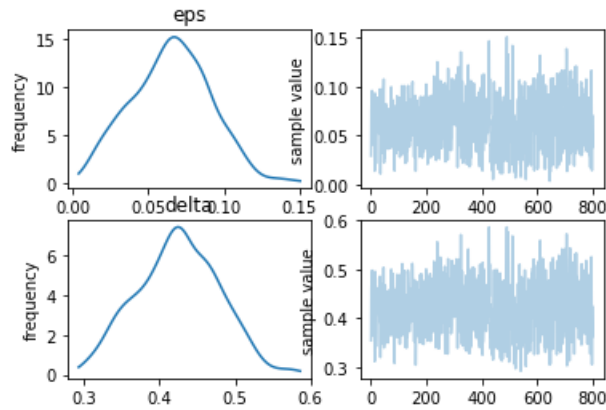


FIGURE 7 – Prior Beta(2,2) pour δ (Valeur réelle fixée = 0.4)

La simulation donne les distributions Posterior suivantes pour δ et ϵ (valeurs réelles égales à 0.05 et 0.4 respectivement) :



Les valeurs prédites par Stan sont très proches des valeurs réelles mais pas parfaitement exactes. Ceci peut avoir différentes explications :

- Faible nombre d'itérations (mais à partir d'un certain nombre, on n'augmente plus la précision)
- Le modèle en lui-même peut présenter les mêmes résultats pour différentes valeurs de paramètres ce qui le rend très difficile à fitter)

4 Minority Games :

L'hétérogénéité est progressivement reconnue comme l'un des ingrédients les plus importants de la modélisation des performances des marchés financiers.

Le jeu de la minorité est défini ainsi : à chaque pas de temps, tous les joueurs doivent choisir parmi deux alternatives ; ceux qui se trouvent dans la minorité gagnent.

Chaque joueur ou agent essaie donc de définir sa stratégie en considérant le comportement des autres : ceci se fait à l'aide d'une fonction qui représente le score de chaque joueur et que chacun essaie de maximiser.

4.1 Formulation Mathématiques :

- L'environnement est constitué par N agents.
- A chaque instant t , chaque agent i prend une décision $a_i(t) \in -1,1$
- • A l'issue de chaque tour, chaque agent i « gagne »

$$-a_i(t) \sum_{j=1}^N a_j(t) = a_i(t) \cdot A(t)$$

On voit bien à travers la dernière formule que les agents qui ont pris la décision minoritaire gagnent à l'instant t .

Comment chaque agent prend sa décision ?

- A chaque instant t les agents disposent d'une information publique $\mu(t)$
- A cette information correspond un panel de S stratégies possibles.
- Chaque agent choisit la stratégie $S_i(t)$ qui maximise le score $y_{i,s}(t)$ telle que :

$$y_{i,s}(t+1) = y_{i,s}(t) - a_{i,s}^{\mu(t)} \cdot A(t)$$

Les joueurs peuvent être de deux types : soit rapides c'est-à-dire il réagissent rapidement à l'évolution du marché et adaptent leur stratégie en permanences, soit lents : c'est-à-dire qu'ils ne changent leurs stratégies que l'après l'écoulement d'une période T .

Le jeu des minorités est parfaitement solvable, ce qui en fait un modèle idéal pour étudier et comprendre divers aspects de la dynamique de la concurrence.

4.2 Modèle Simplifié :

Vu la complexité du modèle général des Minority Games, on a considéré pour notre code Stan un modèle avec des hypothèses simplificatrices par rapport au modèle général.

- On suppose qu'il n'y a pas d'information publique disponible : $\mu(t)=1$ pour tout t .
- On postule que le score de chaque agent évolue de la manière suivante :

$$\Delta_i(t+1) = \Delta_i(t) - \frac{A(t)}{N}$$

avec $A(t) = \sum_{j=1}^N a_j(t)$

- A chaque instant t , pour l'agent i , on donne la loi de la variable aléatoire $a_i(t)$ comme suit (variable de Bernoulli) :

$$P(a_i(t+1) = 1) = \frac{1 + \tanh(\Gamma_i \Delta_i(t))}{2}$$

$$P(a_i(t+1) = -1) = \frac{1 - \tanh(\Gamma_i \Delta_i(t))}{2}$$

Où Γ_i est un paramètre propre à l'agent i qui permet à cet agent d'évaluer un prix de référence et nous renseigne sur sa tendance à se dévier ou pas par rapport à ce prix.

- Le prix évolue :

$$\Delta_i(t) = -p(t) + \Delta_i(0)$$

- Et finalement (et surtout), on suppose que : $\Delta_i(0) = 0$ et $\Gamma_i = \Gamma, \forall i \in [1, N]$

4.3 Simulation et inférence des paramètres du modèle des Minority Games :

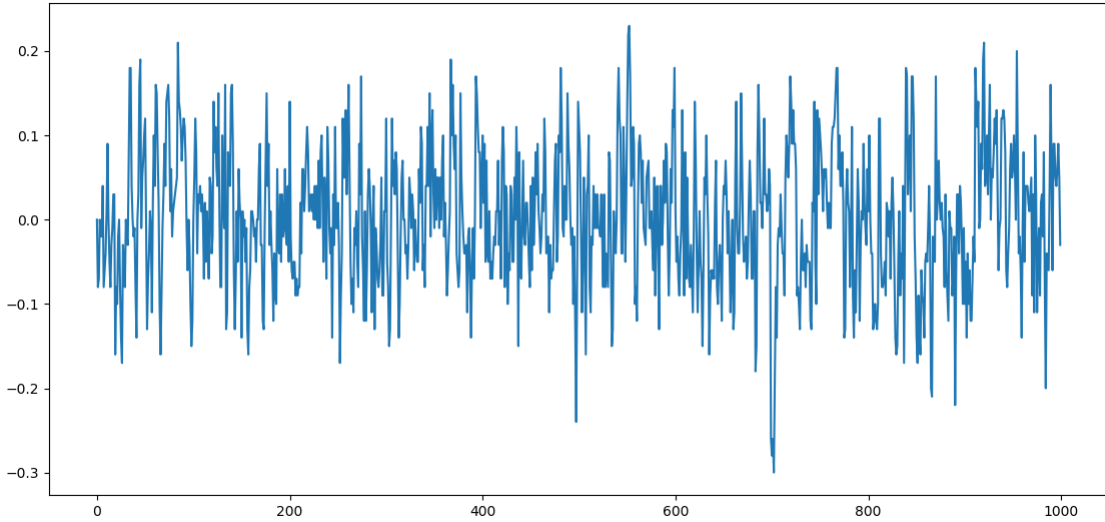
D'après la relation de récurrence précédente pour $\Delta(t)$ on a immédiatement, $\forall t > 0$:

$$\Delta(t) = \sum_{t'=1}^N -\frac{A(t')}{N}$$

A chaque instant $t > 0$, $a_i(t)$ suit une loi de Bernoulli de paramètre : $P(t) = \frac{1 + \tanh(\Gamma_i \Delta_i(t))}{2}$
d'où, selon le théorème Centrale Limite on obtient :

$$A(t) = \sum_{j=1}^N a_j(t) \xrightarrow[N \rightarrow \infty]{\text{en loi}} \mathcal{N}(N \tanh(\Gamma \Delta(t)), \frac{N}{\cosh^2(\Gamma \Delta(t))})$$

On a effectué une simulation de l'évolution du modèle pour $\Gamma=0.5$, $N=200$ et on itère le processus pour $T=1000$ fois. On obtient une évolution du prix comme celui-ci :



On voit donc bien la volatilité du prix entre deux instants relativement proches.

Rappelons que le paramètre qu'on cherche à inférer est Γ , qui est identique pour tous les agents. On a codé le modèle en Stan (fichier MGmodel.py) en utilisant pour $A(t)$ la loi normale donnée par le théorème Central Limite. On trouve alors une distribution posterior de Γ comme suit :

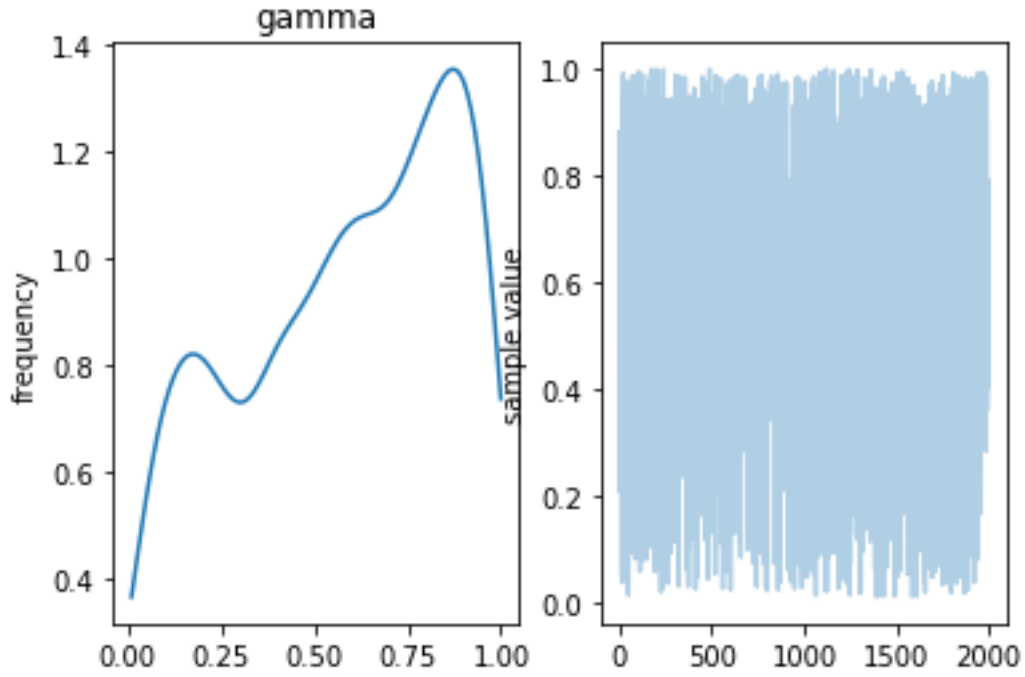


FIGURE 8 – Distribution posterior pour les paramètres fixés : $N=2000$, $T=1000$, $\Gamma=0.3$, Prior = Uniform(0.01,1)

⇒ L'algorithme de Stan ne converge pas !

Lorsqu'on a regardé la variance de la loi normale de $A(t)$ on a vu qu'elle est proportionnelle à N . Donc d'une part, on doit prendre des valeurs de N « très grandes » pour que $A(t)$ s'approche suffisamment de la loi normale et d'autre part les valeurs de N très grandes engendreront une variance « très grande » et donc la distribution de la loi normale s'étale ce qui rend impossible de la simuler correctement.

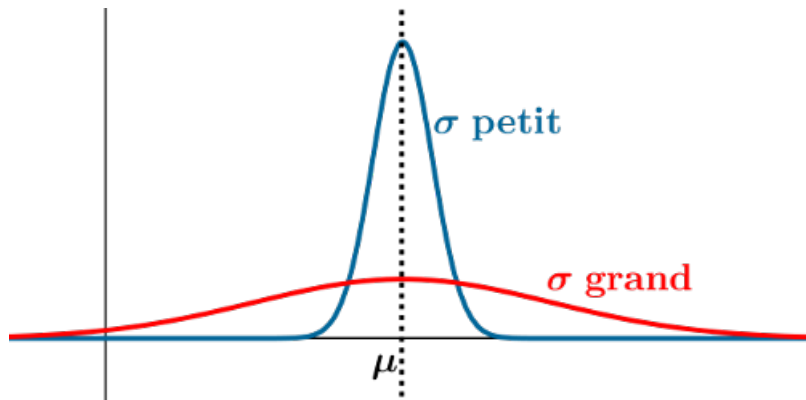


FIGURE 9 – Effet de la variance sur la densité de la loi normale.

On a décidé alors de normaliser $A(t)$ en la divisant par la racine carrée de N et maintenant au lieu de simuler $A(t)$ on simule $\frac{A(t)}{\sqrt{N}}$ avec :

$$A_1 = \frac{A(t)}{\sqrt{N}} \xrightarrow[N \rightarrow \infty]{\text{en loi}} \mathcal{N}(\sqrt{N} \tanh(\Gamma \Delta(t)), \frac{1}{\cosh^2(\Gamma \Delta(t))})$$

Le code Stan est le suivant :

```

5 MG_code = ""
6
7 data {
8   int < lower =50> N;
9   int < lower =500> T;
10  vector [T] A1;
11  vector [T] D;
12 }
13 transformed data {
14 }
15 parameters {
16   real <lower=15,upper=50> gamma;
17 }
18
19 model {
20   gamma~uniform(15,50);
21   for (j in 2:T) {
22     A1[j]~normal(sqrt(N)*tanh(gamma*D[j-1]),1/(cosh(gamma*D[j-1]))^2);
23   }
24 }
25
26 ""
27
28 model = pystan.StanModel(model_code=MG_code)
29 MG_data = {'N': 2000,
30            'T': 1000,
31            'A1': A1,
32            'D': D}
33
34 fit = model.sampling(data=MG_data, iter=1000)

```

On obtient la distribution Prior suivante pour Gamma :

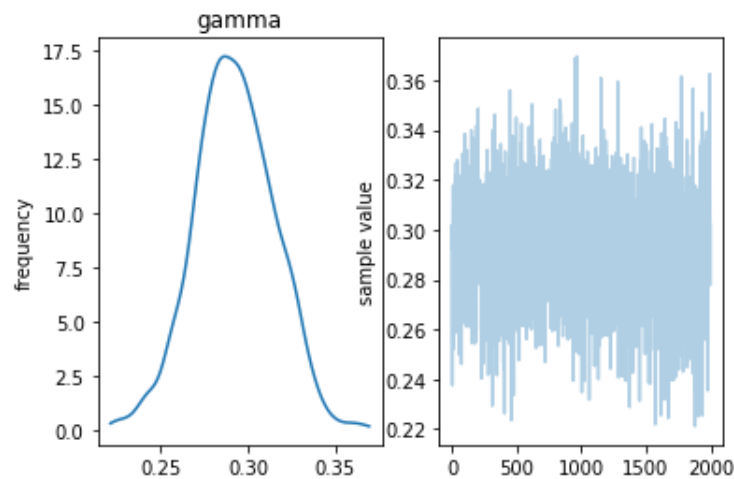


FIGURE 10 – Distribution Posterior de Gamma pour $N=2000$, $T=1000$, $\text{Gamma}=0.3$ Prior= $U(0.01,1)$

L'algorithme converge donc maintenant vers (presque) la bonne valeur de Gamma. La normalisation de $A(t)$ a donc efficacement résolu le problème de la non-convergence.

$\Gamma = 2$, Seuil Critique L'article Minority Games with heterogeneous timescales postule que la valeur $\Gamma=2$ constitue un seuil critique à partir duquel le comportement du modèle change.

On a voulu vérifier cela en faisant des simulations de prix pour des valeurs de Gamma proches de 2 mais de part et d'autre de ce seuil.

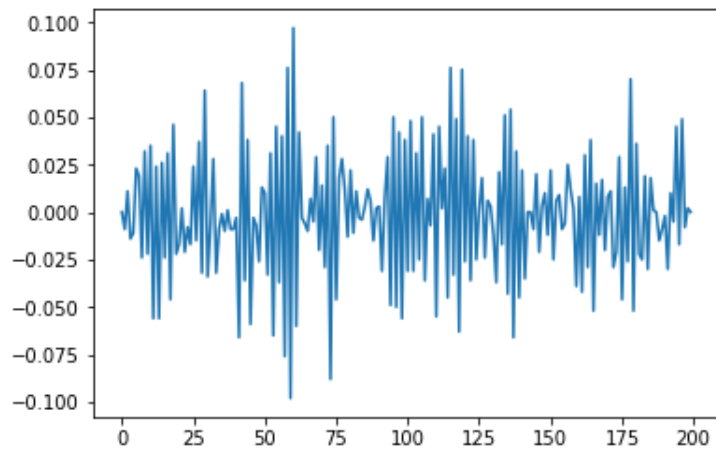


FIGURE 11 – Evolution du prix avec Gamma=1,9

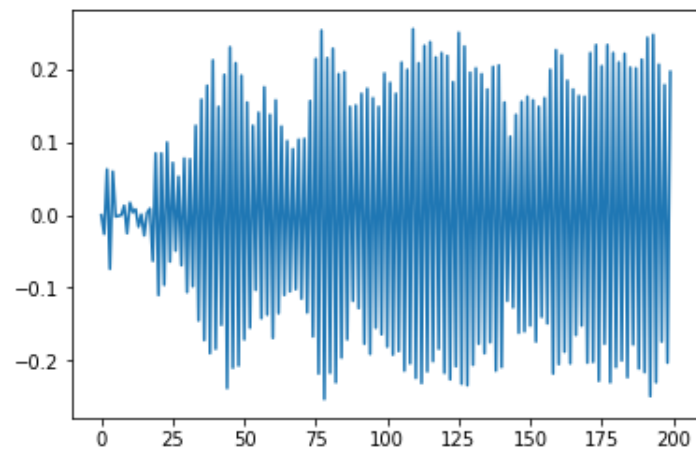


FIGURE 12 – Evolution du prix avec Gamma=2,1

On voit alors bien que pour des valeurs proches de Gamma, on a deux comportements qui sont très différents : avec $\Gamma=2,1$ les oscillations du prix sont plus fortes en amplitude et les changements de signe sont plus fréquents (presque à chaque itération, le signe change).

On s'est demandé par la suite si Stan arriverait à inférer correctement les valeurs de Γ supérieures à 2.

$$\Gamma = 2$$

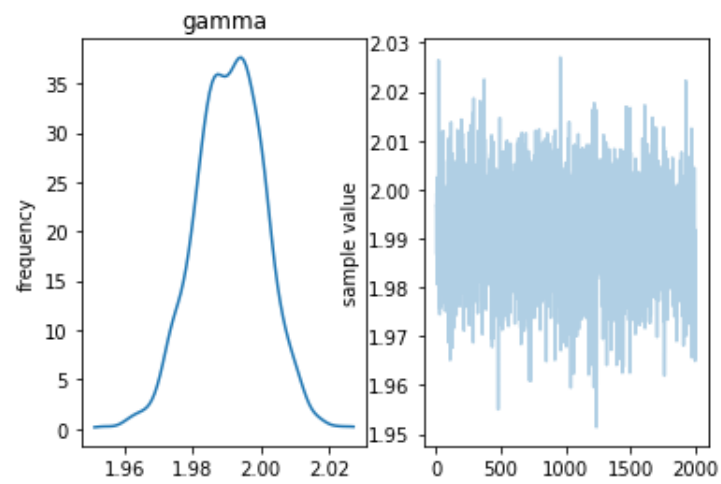


FIGURE 13 – Distribution Posterior de Γ (Prior = Uniform(0.01,4))

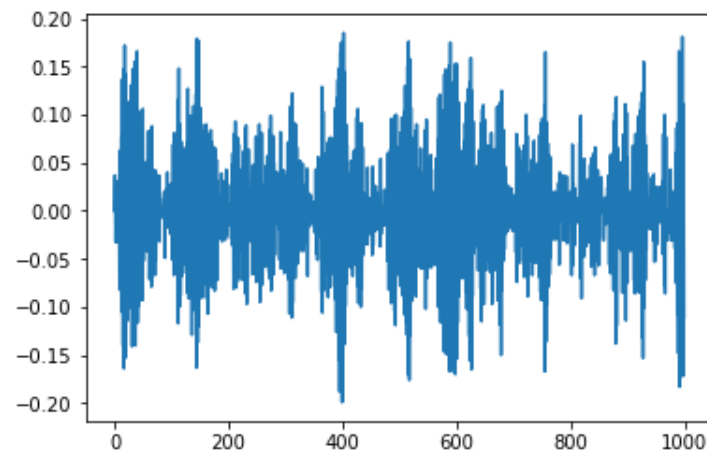


FIGURE 14 – Evolution du prix

$$\Gamma = 7$$

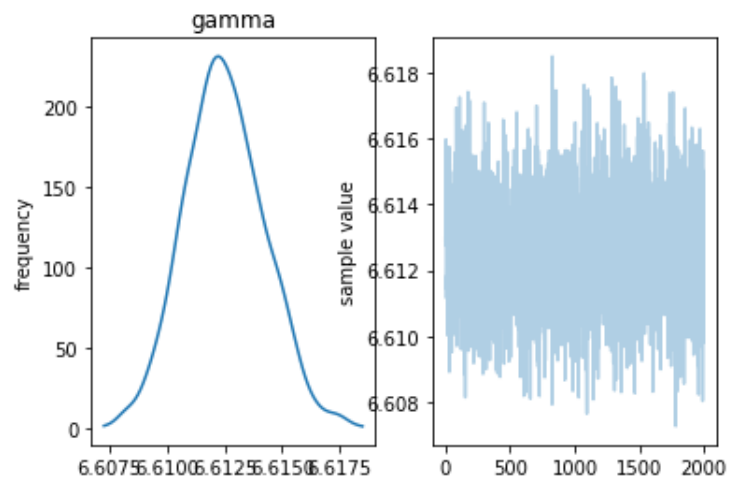


FIGURE 15 – Evolution du prix

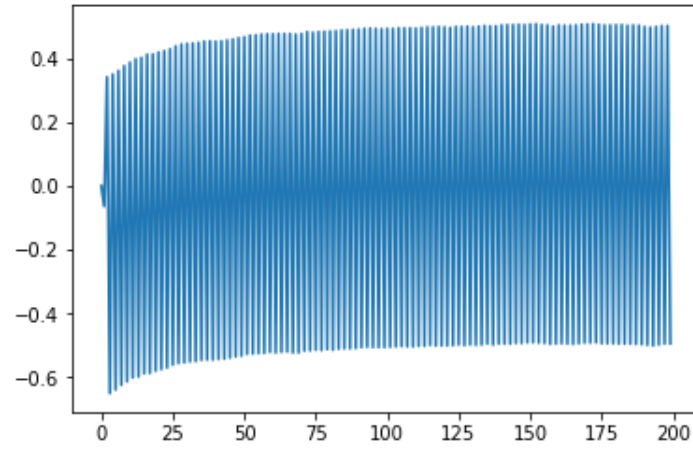


FIGURE 16 – Distribution Posterior de Γ (Prior = Uniform(3,15))

$\Gamma = 30$

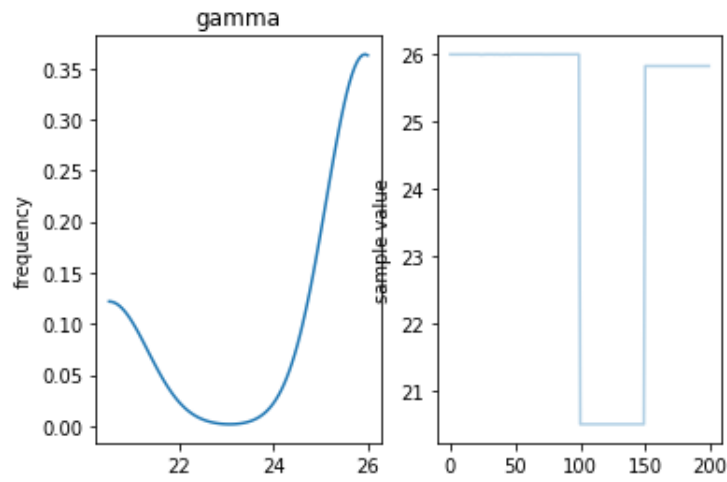


FIGURE 17 – Evolution du prix

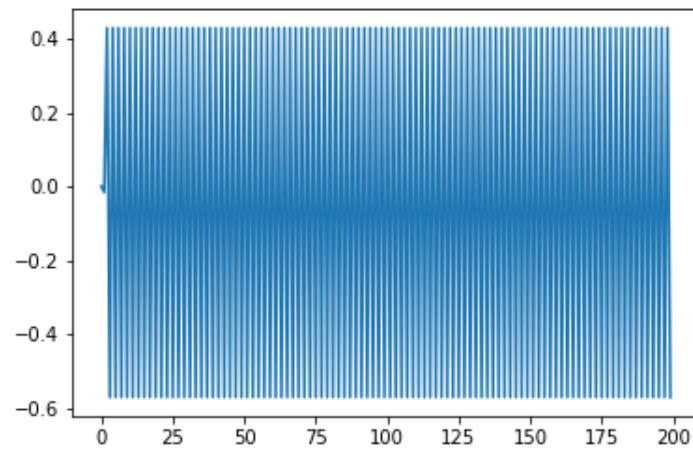


FIGURE 18 – Distribution Posterior de Γ (Prior = Uniform(15,50))

- Pour Gamma=2, Stan arrive à inférer sans problème la bonne valeur de Gamma.

- Pour $\Gamma=7$, la distribution posterior est centrée sur 6,62 qui est une valeur approximative assez proche de la valeur réelle.
- Pour $\Gamma=30$, Stan n'arrive pas à inférer la valeur de Γ .

4.3.1 Explication intuitive du cas $\Gamma = 30$:

- On rappelle que à chaque instant $t > 0$, $a_i(t)$ suit une loi de Bernoulli de paramètre $P(t) = \frac{1+\tanh(\Gamma\Delta(t))}{2}$. Or la fonction tangente hyperbolique tend exponentiellement vers -1 et +1 en $-\infty$ et $+\infty$ respectivement. Donc pour une valeur de Γ « très grande », et suivant le signe de $\Delta(t)$, $P(t)$ est très proche soit de 0 soit de 1 et donc les $a_i(t)$ sont tous presque déterministes et valent soit -1 (pour $P(t)$ proche de 0) soit +1 (pour $P(t)$ proche de 1).
- En regardant la relation de récurrence qui régit l'évolution de $\Delta(t)$ on trouve donc que cette fonction (et donc conséquemment le prix) saute à partir d'un certain instant entre deux valeurs a et $a-1$ avec a entre 0 et 1.
- Ce résultat est identique pour toutes les valeurs de Γ qui sont très grandes. Et donc pour cela, Stan ne possède pas assez de données pour faire la distinction entre les valeurs possibles.

5 Bibliographie :

- [1] Bertschinger, Mozzhorin et Sinha, March 13 2018, Reality-check for Econophysics : Likelihood-based fitting of physics-inspired market models to empirical data.
- [2] Cars Hommes, 2005, Heterogeneous Agent Models in Economics and Finance
- [3] Alan Kirman, 1993, Ants, Rationality, and Recruitment
- [4] Rama Cont, 2000 , Empirical properties of asset returns : stylized facts and statistical issues
- [5] Giancarlo Mosetti, Damien Challet, Yi-Cheng Zhang, October 11 2018 , Minority Games with heterogeneous timescales