

Rapport PFE11

Application du filtre de Kalman aux prédictions des cours de la bourse avec les modèles ARMA et ARIMA

Par : Antoine Delporte

Djahiz Meliani

Jed Younsi

Le modèle ARMA-ARIMA

Il existe deux catégories de modèles pour rendre compte d'une série temporelle. Les premiers considèrent que les données sont une fonction du temps ($y = f(t)$). Cette catégorie de modèle peut être ajustée par la méthode des moindres carrés, ou d'autres méthodes itératives.

L'analyse des modèles par transformée de Fourier est une version sophistiquée de ce type de modèle.

Une seconde catégorie de modèles cherche à déterminer chaque valeur de la série en fonction des valeurs qui la précède ($y_t = f(y_{t-1}, y_{t-2}, \dots)$). C'est le cas des modèles ARIMA ("AutoRegressive– Integrated – Moving Average"). Cette catégorie de modèles a été popularisée et formalisée par Box et Jenkins (1976).

L'objectif essentiel des modèles ARIMA est de permettre une prédiction de l'évolution future d'un phénomène. Son développement dans le domaine de l'économétrie est basé sur ce principe. D'où son utilisation dans ce projet, pour prédire les variations futures d'un cours d'un actif financier.

Un autre intérêt, peut-être plus essentiel en ce qui concerne la recherche scientifique, est de comprendre la signification théorique de ces différents processus. Il est clair cependant que cette interprétation dépend de la nature du phénomène étudié, et des modèles dont le chercheur dispose pour en rendre compte.

Le modèle ARMA est un cas particulier du modèle beaucoup plus général ARIMA où le I désigne *Integrated* en anglais ou *Intégrée* en français. En effet, le modèle ARMA ne permet de traiter que les séries dites stationnaires (des moments du premier ordre qui sont invariants au cours du temps). Les modèles ARIMA permettent de traiter les séries non stationnaires après avoir déterminé le niveau d'intégration (le nombre de fois qu'il faut différencier la série avant de la rendre stationnaire).

En statistique, les modèles ARMA (modèles *autorégressifs et moyenne mobile*), ou aussi modèle de Box-Jenkins, sont les principaux modèles de séries temporelles.

Étant donné une série temporelle X_t , le modèle ARMA est un outil pour comprendre et prédire, éventuellement, les valeurs futures de cette série. Le modèle est composé de deux parties : une partie autorégressive (AR) et une partie moyenne-mobile (MA). Le modèle est généralement noté $\text{ARMA}(p,q)$, où p est l'ordre de la partie AR (nombre de termes autorégressifs) et q l'ordre de la partie MA (nombre de moyennes mobiles).

définition — Le modèle $\text{ARMA}(p,q)$ est un processus temporel discret $(X_t, t \in \mathbb{N})$ vérifiant :

$$X_t = \varepsilon_t + \sum_{i=1}^p \varphi_i X_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

$$X_t = \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} \quad \text{pour tout } t.$$

où les paramètres φ_i et θ_i sont constants, et les termes d'erreurs ε_i sont indépendants du processus.

Un modèle autorégressif $\text{AR}(p)$ est un $\text{ARMA}(p,0)$

Un modèle moyenne mobile $\text{MA}(q)$ est un $\text{ARMA}(0,q)$

Sous certaines conditions, ces processus sont stationnaires. Comme nous le verrons par la suite, ces processus peuvent s'écrire sous la forme :

$$\Phi(L) X_t = \Theta(L) \varepsilon_t, \quad \text{où } \Phi(L) = 1 - \phi_1 L - \dots - \phi_p L^p$$

$$\text{et } \Theta(L) = 1 + \theta_1 L + \dots + \theta_q L^q,$$

L représentant l'opérateur retard, au sens où $L X_t = X_{t-1}$, et avec la convention $L^p = L \circ L^{p-1}$, soit $L^p X_t = X_{t-p}$: la série (Y_t) telle que $Y_t = L^p X_t$ est alors la série (X_t) retardée de p périodes.

Parallèlement, on dira qu'un processus non-stationnaire est intégré d'ordre 1, si en le différenciant une fois, on obtient un processus stationnaire : (X_t) (non-stationnaire) sera dit intégré d'ordre 1 si le processus (Y_t) définit $Y_t = \Delta X_t = X_t - X_{t-1} = (1-L) X_t$ est stationnaire.

On dira, par extension, que (X_t) est intégré d'ordre d si (X_t) est non-stationnaire, ..., (Y_t) où $Y_t = (1-L)^{d-1} X_t$, est non-stationnaire, et (Z_t) où $Z_t = (1-L)^d X_t$, est stationnaire. On appellera alors processus ARIMA (p, d, q) un processus (X_t) pouvant se mettre sous la forme

$$\Pi(L) X_t = \Phi(L) (1-L)^d X_t = \Theta(L) \varepsilon_t, \quad \text{où } (\varepsilon_t) \text{ est un bruit blanc.}$$

Pour les données réelles, on notera que $d = 1, 2$ ou 3 (au maximum). Cela signifie que (Y_t) définit comme différence d'ordre d du processus (X_t) , soit $Y_t = (1-L)^d X_t$, suit un processus $\text{ARMA}(p, q)$.

On parlera d'ailleurs de présence de racine unité : 1 est alors racine du polynôme autorégressif $\Pi(z)$. Par généralisation, on peut considérer le cas où $\exp(2i\pi/s)$ est racine du polynôme autorégressif : c'est à dire que $\Pi(L) = (1 - L^s) \Phi(L)$. On dira alors que l'on est en présence d'une racine unité saisonnière, qui engendreront les modèles SARIMA. Les modèles intégrés sont très présents dans les séries économiques, par exemple les séries d'indices boursiers, d'indice de production, d'indice de prix.... Les modèles SARIMA sont également très présents dès lors que les séries sont très saisonnières (avec une forte périodicité trimestrielle, annuelle...etc).

Dans notre cas :

On note $S(n)$ la modélisation générique du cours d'un actif financier.

Notons : $Y_n = \log(S(n))$

Le modèle ARIMA (0,1,q) s'écrit pour Y_n comme ceci :

$$Y_{n+1} = Y_n + \sum_{i=0}^q \theta_i \varepsilon_{n+1-i} + \mu$$

avec $\theta_0, \dots, \theta_q, \mu$ des paramètres fixes et ε_n suit une Gaussienne centrée réduite qui sont des innovations indépendantes et une valeur initiale Y_0 donné.

Mise sous forme d'état

$$Y_{n+1} = Y_n + \theta_0 \varepsilon_{n+1} + \theta_1 \varepsilon_n + \theta_2 \varepsilon_{n-1} + \mu$$

Considérons le vecteur $X_n = \begin{pmatrix} Y_n \\ \varepsilon_n \\ \varepsilon_{n-1} \end{pmatrix}$

On a : $X_{n+1} = \begin{pmatrix} Y_{n+1} \\ \varepsilon_{n+1} \\ \varepsilon_n \end{pmatrix}$, en décomposant on peut écrire X_{n+1} sous la forme :

$$X_{n+1} = A X_n + B + U_n$$

avec : $A = \begin{pmatrix} 1 & \theta_1 & \theta_2 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$, $B = \begin{pmatrix} \mu \\ 0 \\ 0 \end{pmatrix}$, $Q = \text{Var}(U_n) = \begin{pmatrix} \theta_0^2 & \theta_0 & 0 \\ \theta_0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

Filtre de Kalman

On considère le système décrit par les deux équations d'état :

$$\begin{cases} x_{k+1} = A_k x_k + u_k + \alpha_k \\ y_k = C_k x_k + \beta_k \end{cases}$$

Si on a déjà traité les mesures $y_0 \quad y_1 \quad \dots \quad y_{k-1}$ et que l'on note le vecteur aléatoire (vecteur d'état) $x_{k \vee k-1}$ d'estimée $\hat{x}_{k \vee k-1}$ et de matrice de covariance $\Gamma_{k \vee k-1}$.

La première étape de correction nous donne l'état $x_{k \vee k}$:

$$\begin{aligned} \hat{x}_{k \vee k} &= \hat{x}_{k \vee k-1} + K_k \cdot \tilde{y}_k \quad (\text{estimée corrigée}) \\ \Gamma_{k \vee k} &= \Gamma_{k \vee k-1} - K_k \cdot C_k \Gamma_{k \vee k-1} \end{aligned}$$

La deuxième étape de prédiction :

Connaissant les mesures $y_0 \quad y_1 \quad \dots \quad y_k$, l'état est représenté par $x_{k+1 \vee k}$ et on calcule son espérance $\hat{x}_{k+1 \vee k}$ et sa covariance $\Gamma_{k+1 \vee k}$:

$$\begin{cases} \hat{x}_{k+1 \vee k} = A_k \hat{x}_{k \vee k} + u_k \\ \Gamma_{k+1 \vee k} = A_k \Gamma_{k \vee k} A_k^T + \Gamma_{\alpha_k} \end{cases}$$

Et le filtre de Kalman complet est résumé par les équations suivantes :

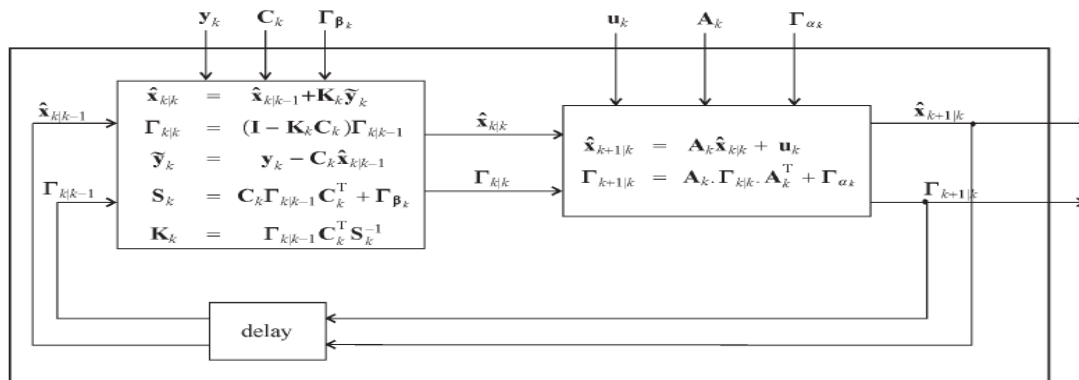
$$\begin{aligned} \hat{x}_{k+1 \vee k} &= A_k \hat{x}_{k \vee k} + u_k \quad (\text{estimée corrigée}) \\ \Gamma_{k+1 \vee k} &= A_k \Gamma_{k \vee k} A_k^T + \Gamma_{\alpha_k} \quad (\text{covariance prédite}) \\ \hat{x}_{k \vee k} &= \hat{x}_{k \vee k-1} + K_k \cdot \tilde{y}_k \quad (\text{estimée corrigée}) \\ \tilde{y}_k &= y_k - C_k \hat{x}_{k \vee k-1} \quad (\text{innovation}) \\ S_k &= C_k \Gamma_{k \vee k-1} C_k^T + \Gamma_{\beta_k} \quad (\text{covariance de l'innovation}) \\ K_k &= \Gamma_{k \vee k-1} C_k^T S_k^{-1} \quad (\text{gain de Kalman}) \\ \Gamma_{k \vee k} &= \Gamma_{k \vee k-1} - K_k \cdot C_k \Gamma_{k \vee k-1} \end{aligned}$$

D'où

le

schéma

suivant :



Plus précisément :

Le modèle structural de base des séries temporelles est décrit par l'équation

$$y_t = \mu_t + \gamma_t + \varepsilon_t, \quad t = 1, \dots, T.$$

Avec l'opérateur de retard, on a l'équation

$$(1 + L + \dots + L^{s-1})\gamma_t = S(L)\gamma_t = \omega_t, \quad t = 1, \dots, T.$$

Puis

$$\Delta_s = 1 - L^s = (1 + L + \dots + L^{s-1})(1 - L) = S(L)\Delta$$

D'où

$$\Delta_s \gamma_t = (1 - L)\omega_t$$

Pour le filtre de Kalman, on reprend les deux équations :

équation de transition

$\alpha_t = T_t \alpha_{t-1} + \eta_t$, avec η_t ($t = 1 \dots T$) distribués normalement et de manière indépendante, de moyenne 0 et variance σ_η^2 , T_t une matrice fixée de taille $m \times m$

équation de mesure

$$y_t = z_t' \alpha_t + \varepsilon_t, \quad \text{avec } t = 1, \dots, T.$$

α_t est un vecteur de taille $m \times 1$ et z_t est un vecteur fixé de taille $m \times 1$.

On note a_{t-1} l'estimateur minimum MMSE de α_{t-1} au temps $t-1$ et P_{t-1} la matrice de covariance de l'erreur d'estimation $a_{t-1} - \alpha_{t-1}$. Lorsque y_t est disponible, a_{t-1} et P_{t-1} sont mis à jours par le filtre de Kalman. Si $\alpha_0 \sim N(a_0, P_0)$ avec

α_0 et P_0 connus, le filtre de Kalman produit un ensemble de T erreurs de prédiction v_t ($t = 1, \dots, T$), avec leurs variances respectives f_t .

Pour $s=4$, l'équation de transition devient

$$\alpha_t = \begin{bmatrix} \mu_t \\ \beta_t \\ \dots \\ y_t \\ y_{t-1} \\ y_{t-2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & & & 0 \\ 0 & 1 & & & \\ & & -1 & -1 & -1 \\ 0 & 1 & 0 & 0 & \\ & 0 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} \mu_{t-1} \\ \beta_{t-1} \\ \dots \\ y_{t-1} \\ y_{t-2} \\ y_{t-3} \end{bmatrix} + \begin{bmatrix} \eta_t \\ \zeta_t \\ \omega_t \\ 0 \\ 0 \end{bmatrix}.$$

L'équation de mesure devient :

$$y_t = (1 \ 0 \ 1 \ 0 \ 0) * \alpha_t + \varepsilon_t, \quad t=1, \dots, T.$$

Pour l'implémentation du filtre de Kalman, il faut savoir estimer $h_t = \sigma_\varepsilon^2$ et $Q_t = \text{Diag}(\sigma_\eta^2, \sigma_\zeta^2, \sigma_\omega^2, 0, 0)$, et également les valeurs initiales du filtre. Par ailleurs le modèle à espace d'états est invariant avec le temps puisque les paramètres utilisés sont constants avec t. Finalement, le filtre de Kalman converge vers un état stable lorsque P_t devient une matrice définie positive invariante avec le temps P.

Estimation du maximum de vraisemblance dans le domaine temporelle

On utilise la décomposition de l'erreur de prédiction pour obtenir avec le filtre de kalman la fonction de vraisemblance (lorsque les perturbations ε_t et η_t sont normalement distribuées).

Le modèle basique structurel contient 4 inconnues $\sigma_\varepsilon^2, \sigma_\zeta^2, \sigma_\omega^2, \sigma_\eta^2$ mais l'optimisation non-linéaire n'a besoin que de 3 inconnues et la 4^{ème} peut être mise en dehors de la fonction de vraisemblance. Le filtre de Kalman peut être utilisé en tant que fonction de ces trois paramètres. Pour les trois méthodes qui suivent, on admettra que le vecteur d'état contient k éléments non stationnaires.

1^{ère} décomposition de l'erreur de prédiction : « diffusion préalable »

Lorsque α_0 a une diffusion préalable, les valeurs initiales peuvent être construites à partir des k observations (α_k et P_k). La fonction de vraisemblance est définie pour les observations y_{k+1}, \dots, y_T conditionnellement à y_1, \dots, y_k . La décomposition de l'erreur de décomposition donne la vraisemblance suivante :

$$\log L = \frac{-(T-k)}{2} \log 2\pi - \frac{1}{2} \sum_{t=k+1}^T \log f_t - \frac{1}{2} \sum_{t=k+1}^T v_t^2 / f_t$$

Où v_t est l'erreur de prédiction à un pas et f_t est sa variance. Les deux quantités sont données par le filtre de Kalman.

Pour implémenter en R ou matlab l'équation précédente, on initialise le filtre = t=0 avec $a_0=0$ et $P_0=xI$, où x est un nombre fini pris assez grand. Des approximations précises de a_k et P_k sont ensuite obtenues après k itérations du filtre de Kalman (on écrit les équations pour les k observations termes de α_k) et donc un cas particulier de la généralisation des moindres carrés.

La méthode ci-dessus peut être généralisée pour couvrir les cas où le vecteur d'état contient un sous vecteur d'éléments générés par un processus stationnaire de moyenne connue et de matrice de covariance V. On a alors

$$P_0 = \begin{bmatrix} xI & 0 \\ 0 & V \end{bmatrix}$$

Cette première décomposition correspond à la définition usuelle de la fonction de vraisemblance pour un modèle ARIMA

2^{ème} décomposition de l'erreur de prédiction : vecteur d'état initial fixé

Si le vecteur d'état initial est fixé, on a

$$y_t = x_t' \alpha_0 + w_t, \quad t=1, \dots, T \quad \text{avec} \quad x_t = T' z_t \quad \text{et} \quad w_t = z_t' \sum_{j=1}^T T^{t-j} \eta_j + \varepsilon_t, \quad t=1, \dots, T$$

Le terme de perturbation w_t est de moyenne nulle et de matrice de covariance Ω_0 . L'estimateur MV $\hat{\alpha}_0$ de α_0 peut être implémenté par la méthode générale des moindres carrés. On a ensuite

$$\Omega_0 \mathbf{v} - \frac{1}{2} (y - x_0 \hat{\alpha}_0)' \Omega_0^{-1} (y - x_0 \hat{\alpha}_0)$$

$$\log L = -\frac{T}{2} \log 2\pi - \frac{1}{2} \log$$

Avec $y = x_0' \alpha_0 + w$, $V(w) = \Omega_0$.

3^{ème} décomposition de l'erreur de prédiction : filtre de Kalman à état constant

On implémente α_k à partir des k observations comme en méthode 1, mais on initialise la matrice de covariance de l'erreur de prédiction P_k égale à la matrice de covariance de l'état constant \hat{P}_\square . On utilise ensuite la formule de la 1^{ère} décomposition pour déterminer la fonction de vraisemblance.

$\Delta_s y_t$ que l'on a calculé au début suit un processus ARMA dont les paramètres sont sujet à des contraintes non-linéaires.

Variance de l'erreur de prédiction et prédictions post-échantillonnage :

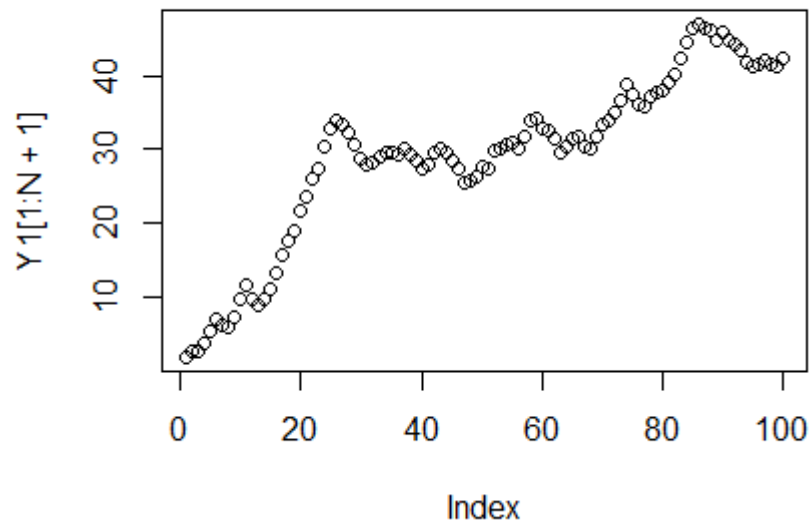
La variance de l'erreur de prédiction $\tilde{\sigma}^2$ peut être estimé par le filtre de Kalman d'après $\tilde{\sigma}^2 = f_T$. Par ailleurs σ^2 est la variance des perturbations dans le modèle ARIMA.

Partie expérimentale

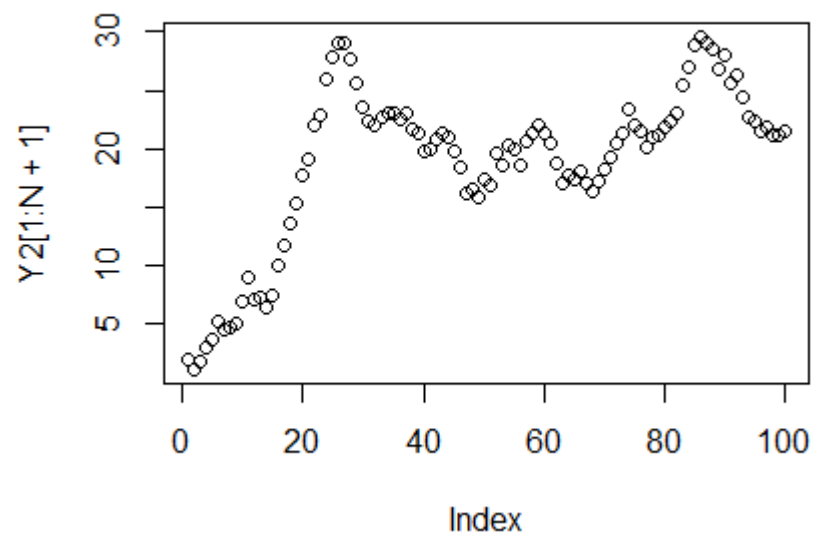
1^{ère} question

Simulons pour $N=100$, pour chaque jeu de paramètre, les réalisations du ARIMA(0,1,2) à l'aide de la fonction fournie en annexe.

Pour theta1 et mu1 :



Pour theta2 et mu2 :

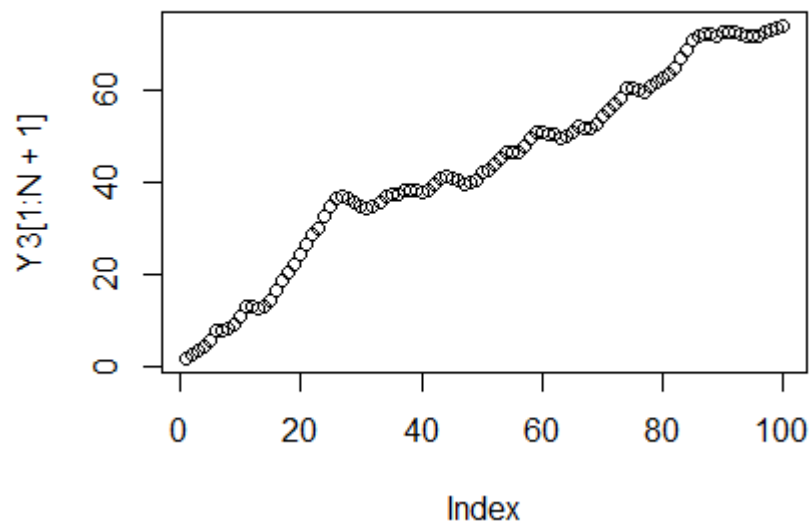


Pour

theta3

et

mu3 :



Commentaires : On a réalisé la fonction de simulation des valeurs du cours avec pour paramètres d'entrée le nombre de simulation, les paramètres theta et les moyennes. Cette fonction réalise une boucle sur le nombre N en séparant le cas $i=1$ et $i>1$. On remarque des variations plus importantes pour le deuxième cas, cela est très probablement dû à une moyenne mu plus importante que pour les autres cas. Ensuite, les variations générales des trois courbes sont croissantes (même si l'on a présence de minimas locaux). Donc ces courbes obtenues ne sont pas forcément adaptées à la modélisation d'un cours de bourse, dont les variations générales ont souvent des phases décroissantes.

Code :

```
N <- 100
mu1 <- 0.5
theta1 <- c(1,0.6,0.4)
mu2 <- 0.3
theta2 <- c(1,0.3,0.8)
mu3 <- 0.8
theta3 <- c(0.5,0.5,0.5)
Y0 <- 0
epsilon <- rnorm(N+1,mean=0,sd=1)
ysimul <- function(Y0,N,mu,theta) {
  for (i in 1:N) {
    if (i == 1){ Y <- c(Y0,Y0+mu+theta[1]*epsilon[2]+theta[2]*epsilon[1]+mu)}
  }
```

```

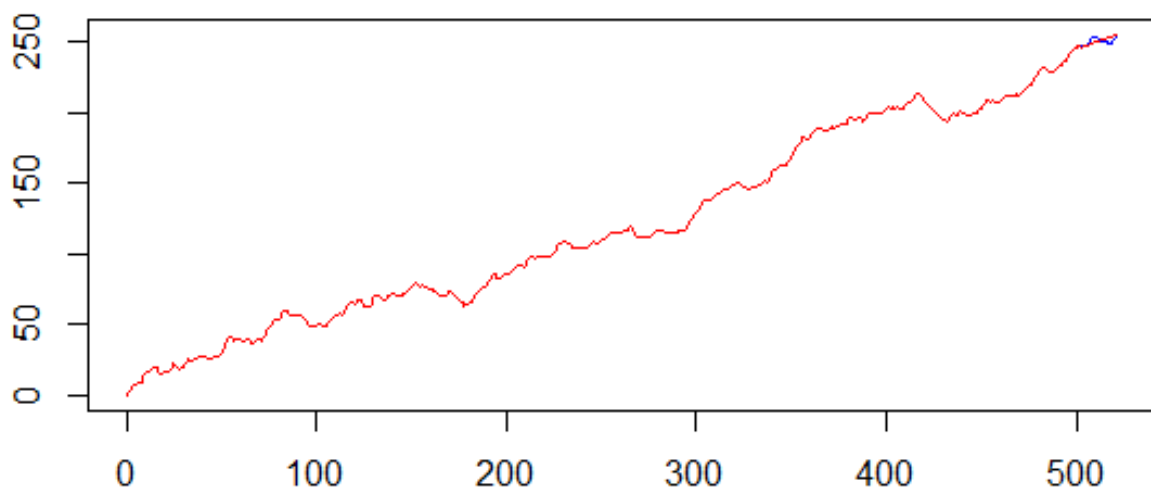
else {Y <- c(Y,Y[i-1]+theta[1]*epsilon[i]+theta[2]*epsilon[i-1]+theta[3]*epsilon[i-2]+mu)}
}
return(Y)
}
Y1 <- ysimul(Y0,N,mu1,theta1)
Y2 <- ysimul(Y0,N,mu2,theta2)
Y3 <- ysimul(Y0,N,mu3,theta3)
plot(Y1[1:N])
plot(Y2[1:N])
plot(Y3[1:N])

```

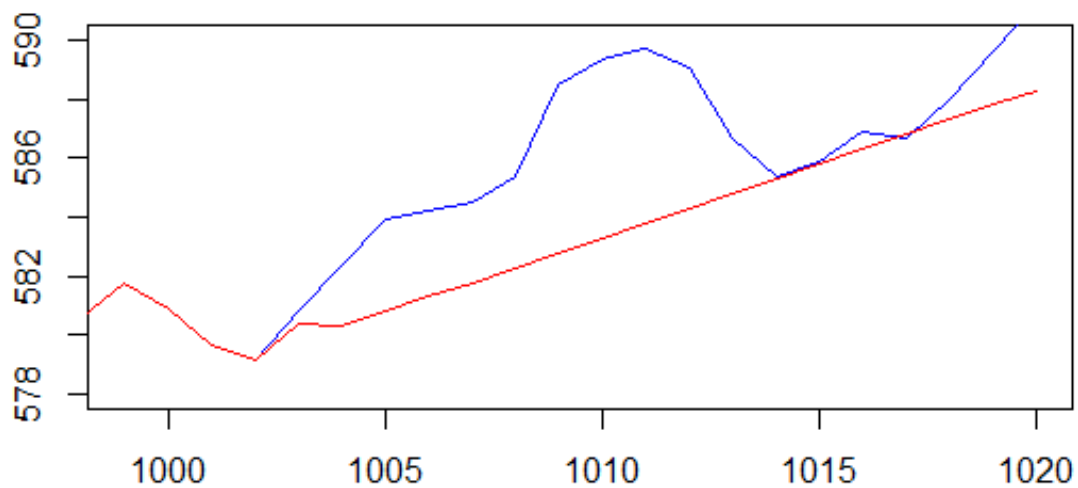
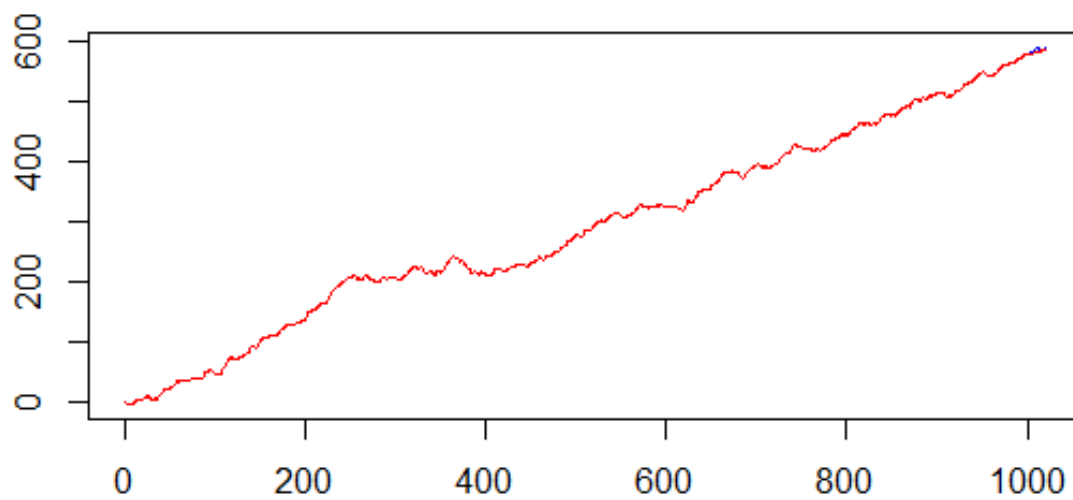
2^{ème} question

L'implémentation du filtre de Kalman se réalisa en deux étapes. Tout d'abord on initialise les paramètres nécessaires, puis on écrit dans une boucle allant de 1 à $N+t$ (t étant le nombre de valeurs de prédiction voulues) les formules de prédiction à un pas et de mise à jour, en séparant le cas suivant que l'indice de la boucle est inférieur ou supérieur à N .

Pour $N=500$ on obtient cette prévision (en rouge la valeur prédite à partir de $t=500$ jusqu'à 520, en bleu la vraie valeur du cours) :



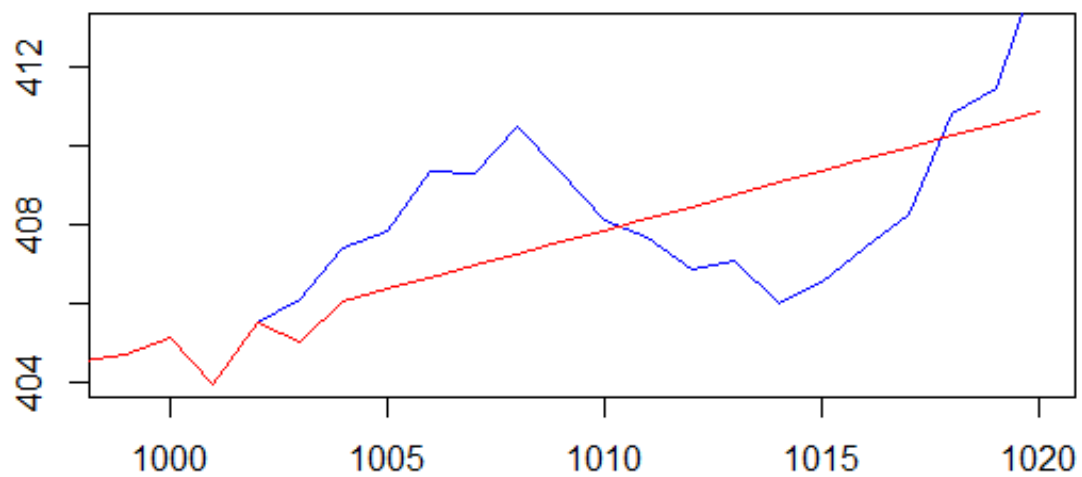
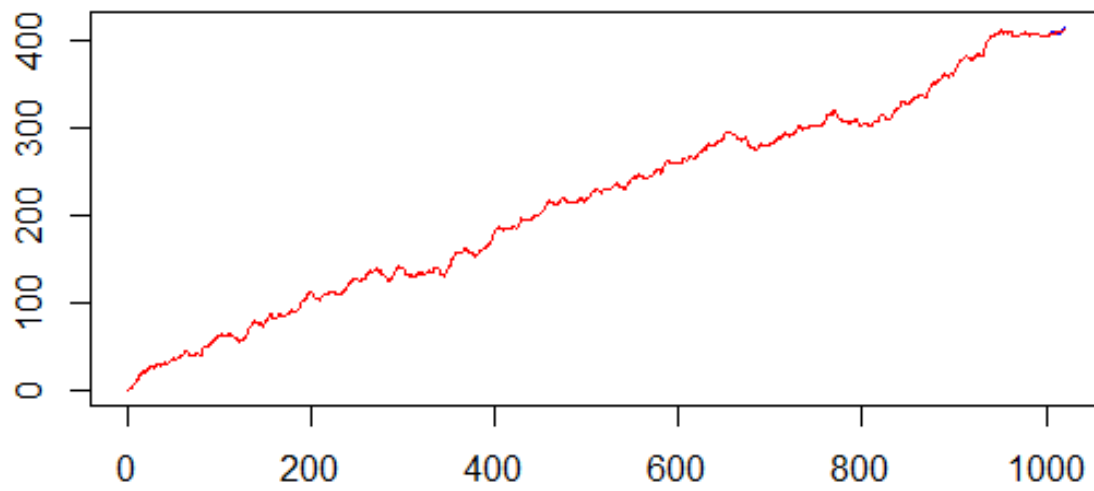
Pour $N=1000$ et pour Y1 (premier jeu de paramètres) :



On obtient ici comme erreur (erreur quadratique moyenne standard entre la courbe bleue et la courbe rouge) :

```
> erreur(1001,N)
[1] 9.097035
```

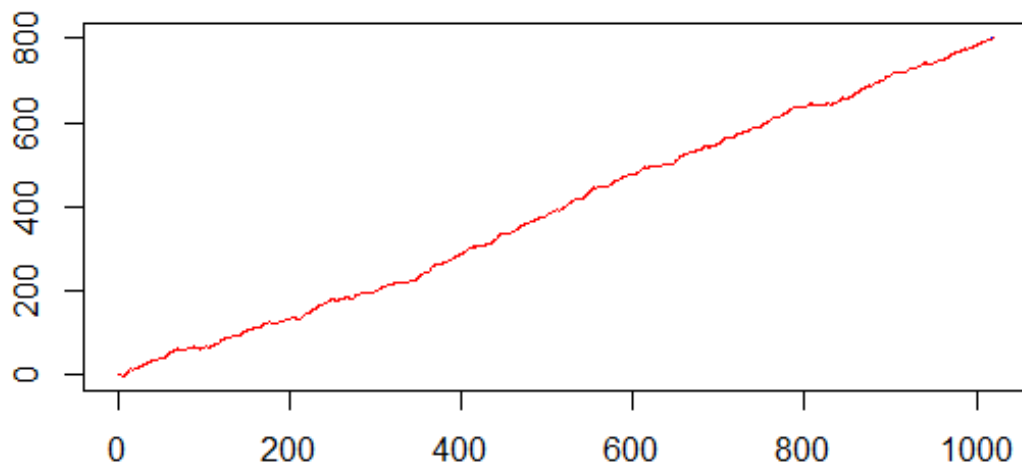
Pour $N=1000$ et pour $Y2$ (deuxième jeu de paramètres) :



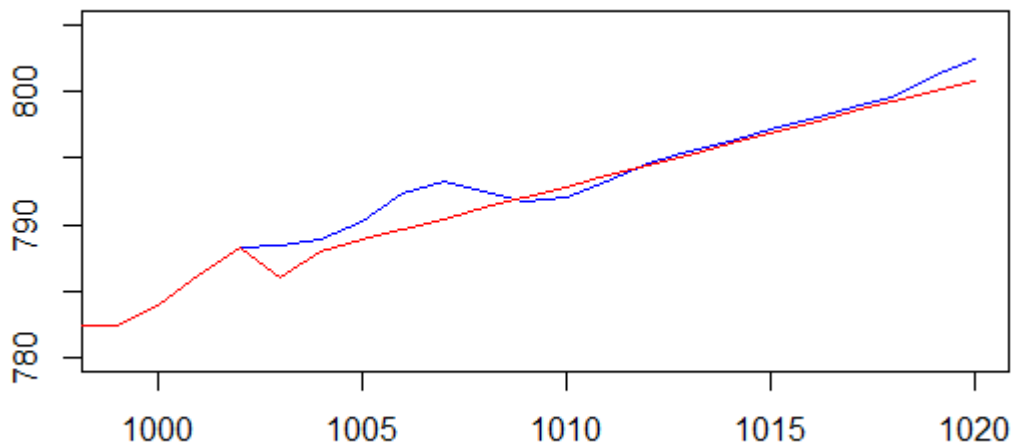
On obtient une erreur de prédiction égale à :

```
> erreur(1001,N)
[1] 3.894303
```

Pour le troisième jeu de paramètres :



Le zoom donne :



On obtient une erreur égale à :

```
> erreur(1001,N)
[1] 1.517577
```

Remarques : On obtient une erreur plus importante pour le premier jeu de paramètre par rapport aux deuxième et troisième erreurs. Ensuite, les variations générales de la prédiction respectent dans chaque cas la vraie valeur du cours jusque $N=1020$, donc le filtre de Kalman est un bon moyen de prédiction.

Code :

```
predict_kalman <- function(Y,N,t,theta,mu) {
  A <- rbind(c(1,theta[2],theta[3]),c(0,0,0),c(0,1,0))
  B <- cbind(c(mu,0,0))
```

```

Q <- rbind(c(theta[1]*theta[1],theta[1],0),c(theta[1],1,0),c(0,0,0))

resultat <- list()
x_n_n <- list()
gamma_n_n <- list()
x_nplusun_n <- list()
gamma_nplusun_n <- list()
x_n_n[[1]] <- cbind(c(Y[1],0,0))
gamma_n_n[[1]] <- rbind(c(0,0,0),c(0,1,0),c(0,0,1))
epsilon_estime <- rep(N+t)
Y_estime <- c(Y[1])
seq <- 1

for (i in 1:(N+t)) {
  if (i < N) {
    x_nplusun_n[[i]] <- A%%x_n_n[[i]]+B
    gamma_nplusun_n[[i]] <- A%%gamma_n_n[[i]]%%t(A)+Q
    delta <- cbind(c(gamma_nplusun_n[[i]][1],gamma_nplusun_n[[i]][4],gamma_nplusun_n[[i]]
[7]))
    eta <- gamma_nplusun_n[[i]][1]
    x_n_n[[i+1]] <- x_nplusun_n[[i]]+(Y[i+1]-x_nplusun_n[[i]][1])/eta*delta
    gamma_n_n[[i+1]] <- gamma_nplusun_n[[i]]-(delta%%t(delta))/eta
    seq <- c(seq,dnorm(Y[i+1], x_nplusun_n[[i]][1], sqrt(eta)))
    epsilon_estime[i+1] <- x_n_n[[i+1]][2]
    epsilon_estime[i] <- x_n_n[[i+1]][3]
    X_estime <- x_n_n[[i+1]]
  }
  if (i > N+1) {
    X_estime = A%%X_estime+B
    Y_estime <- c(Y_estime,X_estime[1])
  }
  else {

```

```

    Y_estime <- c(Y_estime,Y[i+1])
  }
}
resultat[[1]] <- Y_estime
resultat[[2]] <- seq
return (resultat)
}

```

Avec le code suivant on effectue la prédiction et on affiche les résultats

```

N<-1020
epsilon <- rnorm(N+1,mean=0,sd=1)
Y1 <- ysimul(Y0,N,mu1,theta1)
resultat <- predict_kalman(Y1,1000,20,theta1,mu1)
Yr <- resultat[[1]]
vraisemblance <- sum(log(resultat[[2]]))
x <- c(1:1020)
y1 <- Y1
y2 <- c(Y1[1:1000],Yr[1001:1020])
plot(x, y1, type = "n", ylim = range(c(y1, y2)), xlab = "", ylab = "")
lines(x, y1, col = "blue")
lines(x, y2, col = "red")

```

3^{ème} question

Dans cette question, nous simulons des séries entières suivant un modèle ARIMA (0,1,2) dans trois cas de figures différents. Ensuite avec la méthode de la grille nous tentons de retrouver les paramètres utilisés dans ces trois cas. La méthode de la grille est une méthode itérative qui nous permet de trouver les trois paramètres θ et le paramètre de moyenne en testant pour chacun de ces paramètres des valeurs différentes. Pour chacun de ces tests nous faisons une prédiction à l'aide du filtre de Kalman et nous calculons ensuite la vraisemblance de la prédiction. On garde à la fin les paramètres maximisant cette vraisemblance.

Les différentes valeurs de test sont choisies dans une grille de valeur allant dans notre cas de 0 à 1 avec un pas que nous prenons à 0.1. Plus le pas choisi est faible et plus précis sera le

calcul des différents paramètres. Cependant le calcul sera aussi beaucoup plus long car il faudra itérer l'algorithme sur une grille beaucoup plus grande.

Code :

```
grille = function(Y,N,t) {  
  vraisemblance <- -Inf  
  mu<-0  
  theta<-list()  
  for(m in c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)) {  
    for(theta1 in c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)) {  
      for(theta2 in c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)) {  
        for(theta3 in c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)) {  
          resultat <- predict_kalman(Y,N,t,c(theta1,theta2,theta3),m)  
          vraisemblance_temp <- sum(log(resultat[[2]]))  
          if (vraisemblance_temp > vraisemblance) {  
            vraisemblance <- vraisemblance_temp  
            theta <- c(theta1,theta2,theta3)  
            mu <- m  
          }  
        }  
      }  
    }  
  }  
  return (c(theta[3],theta[2],theta[1],mu))  
}
```

Résultats :

Cas	Theta réel	Théta estimé	Moyenne réelle	Moyenne estimée
Cas 1	(1,0.6,0.4)	(1,0.6,0.4)	0.5	0.4
Cas 2	(1,0.3,0.8)	(1,0.3,0.8)	0.3	0.3

Cas 3	(0.5,0.5,0.5)	(0.5,0.5,0.5)	0.8	0.7
-------	---------------	---------------	-----	-----

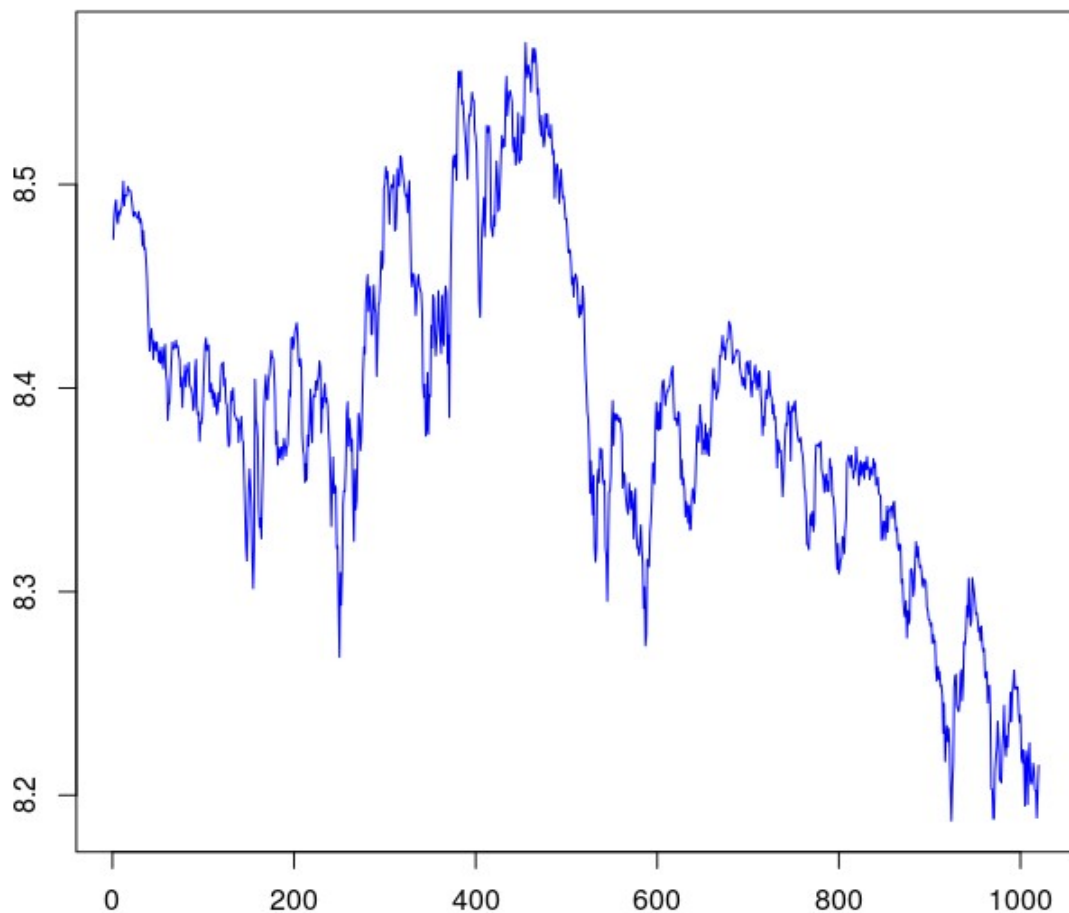
On observe que la méthode fonctionne plutôt bien. Dans les trois cas, on trouve les bonnes valeurs pour theta. Dans deux cas on trouve une valeur de moyenne différente mais assez proche de la valeur attendue.

4^{ème} question

Le but de cette question est de proposer un protocole permettant d'appliquer ce que l'on a vu précédemment à des données issues du marché.

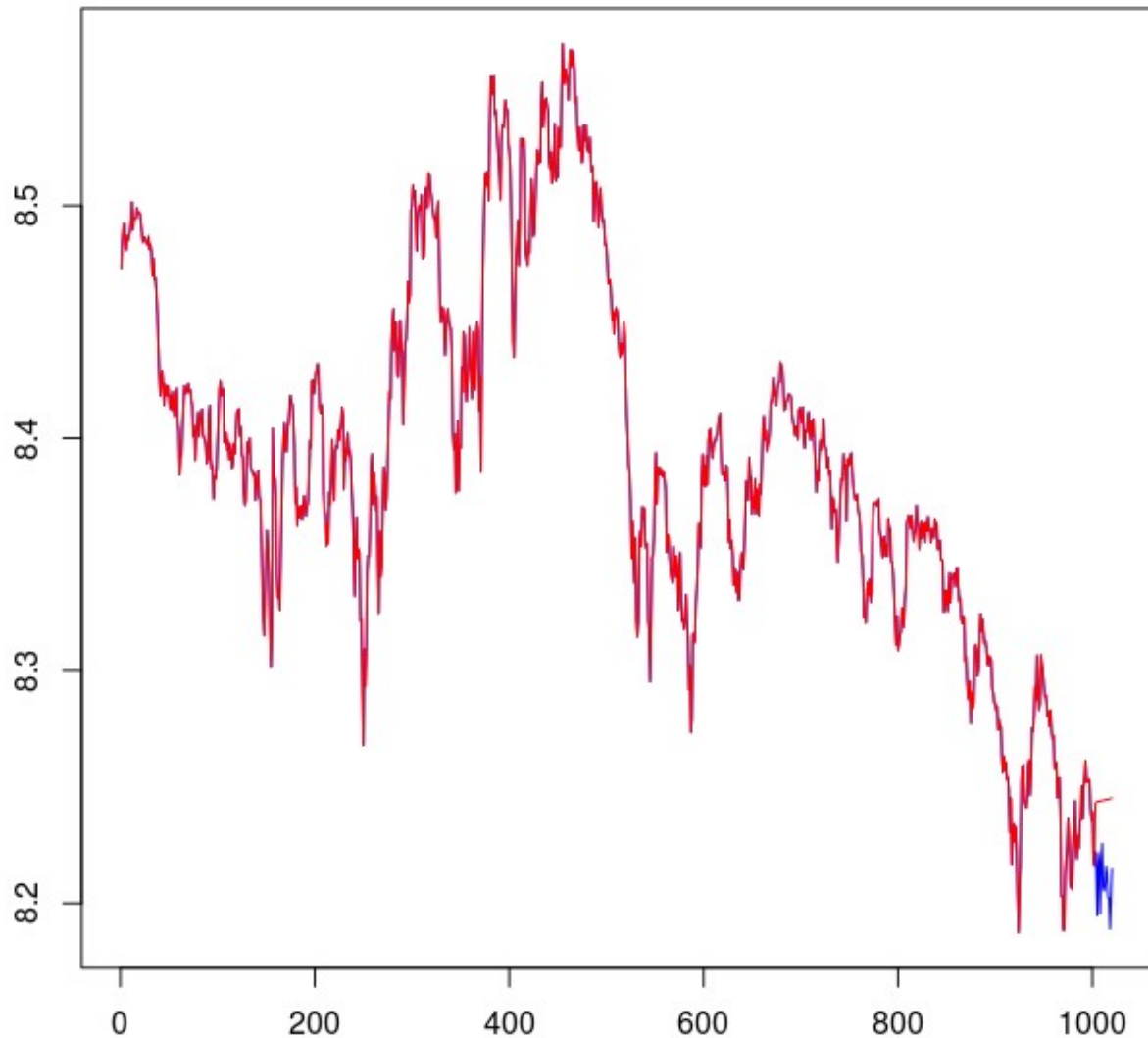
Pour cela, on commence par récupérer des données sur le prix de l'indice CAC40. On prend un peu plus de 1000 valeurs journalières allant de janvier 2013 à janvier 2017. Le protocole est le suivant : on crée à partir du log de 1020 valeurs une première série entière ; ensuite avec la méthode de la grille on détermine les paramètres theta et mu nous permettant de modéliser cette série sous la forme d'un ARIMA (0,1,2) ; puis grâce au filtre de Kalman on prédit à partir des 1000 premières valeurs les 20 suivantes.

En valeur log, la courbe pour les données récupérées est la suivante :



En appliquant la méthode de la grille, on obtient des valeurs très peu précises car pour plus de précision il faut itérer avec un pas très petit, ce qui rend le calcul beaucoup trop long.

En utilisant une fonction de R, la fonction `arima()` avec en paramètre notre série entière ainsi que l'ordre recherché, on obtient le theta suivant $(-0.0115, -0.0328, 0.0290)$. En utilisant cette valeur pour la prédiction on a le résultat suivant :



On observe que la prédiction est pas très précise. Cela peut être du au choix des paramètres theta et mu, ou bien au fait que la serie ne soit pas très stationnaire en comparaison aux séries testées dans les premières questions.