Finanças Quantitativas

Lista de exercícios 6

Lucas Emanuel Resck Domingues

Junho de 2020

Problema 6.6 [1]

(i)

1. No lado esquerdo da equação:

$$\begin{aligned} X_t - X_{t-1} &= X_t - BX_t \\ &= (1 - B)X_t \end{aligned}$$

   Então concluímos $\phi(z) = 1 - z$. Do lado direito:

$$\begin{aligned} W_t - 1,5W_{t-1} &= W_t - 1,5BW_t \\ &= (1 - 1,5B)W_t \end{aligned}$$

   Logo, $\theta(z) = 1 - 1,5z$. Portanto, o modelo pode ser escrito:

$$\phi(B)X_t = \theta(B)W_t$$

2. O modelo é estacionário desde que a parte AR seja. Se assumimos $\phi(z) = 0$, temos $z = 1$. Como $|z| \leq 1$, o modelo não é estacionário (era necessário $|z| > 1$).

   Da mesma forma o modelo é invertível se a parte MA o é. Porém, $\theta(z) = 0$ implica em $z < 1$, de forma que o modelo não é invertível.

(ii)

1. De forma análoga ao exercício anterior, temos

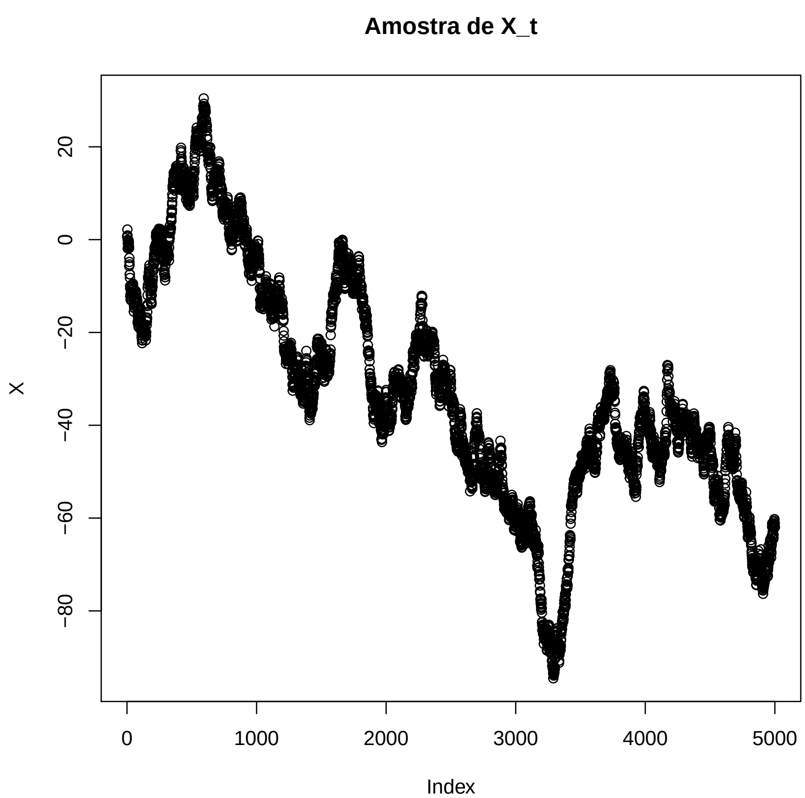
$$\phi(B)X_t = \theta(B)W_t$$

   com $\phi(z) = 1 - 0,8z$ e $\theta(z) = 1 - 0,5z$

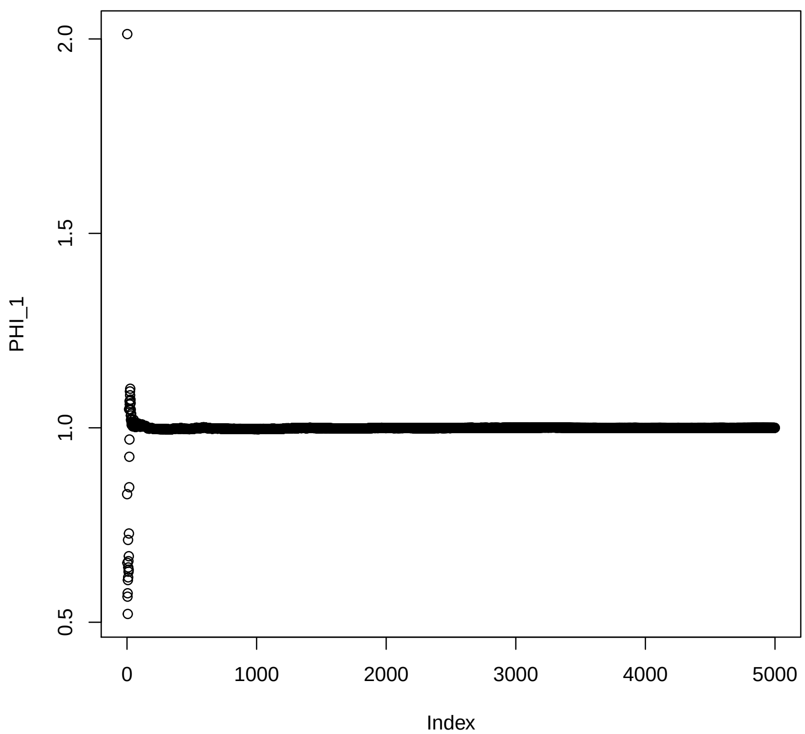
2. Neste caso, para ambos os polinômios, $|z| > 1$, de modo que o modelo é estacionário e invertível.

Problema 6.11 [1]

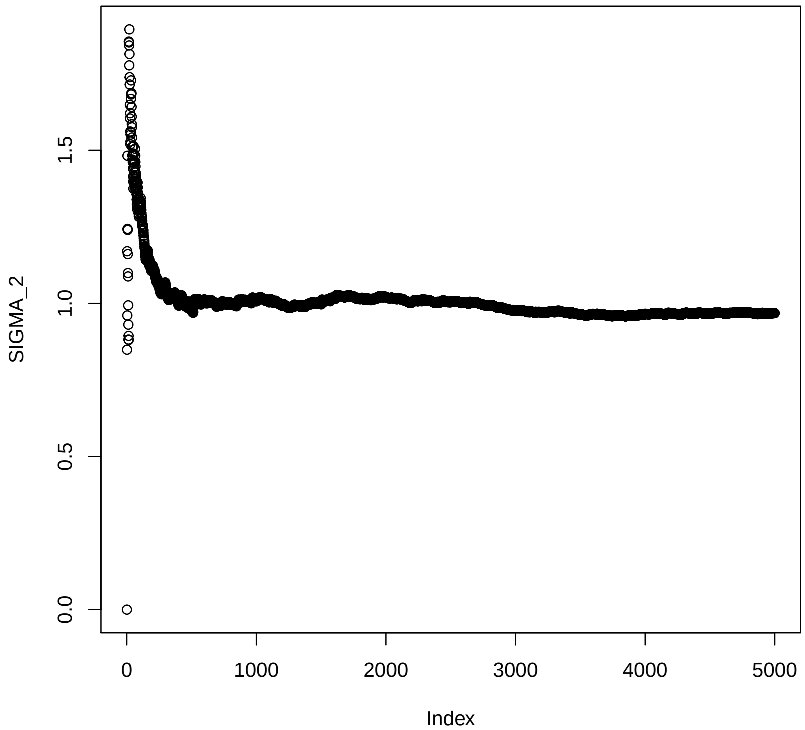
1.
```
N = 5000
fX = function(N) {
  WN = rnorm(N)
  X = c(WN[1])
  for (t in 2:N) {
    X_t = X[length(X)] + WN[t]
    X = c(X, X_t)
  }
  X
}
X = fX(N)
plot(X, main = "Amostra de X_t")
```



2.
```
fphi_1 = function(X) {
  n = length(X)
  sum_1 = 0
  sum_2 = 0
  for (i in 2:n) {
    sum_1 = sum_1 + X[i]*X[i-1]
    sum_2 = sum_2 + X[i-1]^2
  }
  sum_1/sum_2
}

fsigma_2 = function(X, phi_1) {
  n = length(X)
  sum = 0
  for (i in 2:n) {
    sum = sum + (X[i] - phi_1*X[i-1])^2
  }
  sum/(n-1)
}

fdf = function(phi_1, sigma_2) {
  (phi_1 - 1)/sqrt(sigma_2)
}

PHI_1 = c()
SIGMA_2 = c()
DF = c()
for (n in 1:N) {
  phi_1 = fphi_1(X[1:n])
  PHI_1 = c(PHI_1, phi_1)
  sigma_2 = fsigma_2(X[1:n], phi_1)
  SIGMA_2 = c(SIGMA_2, sigma_2)
  df = fdf(phi_1, sigma_2)
  DF = c(DF, df)
}

plot(PHI_1)
```




```r
plot(SIGMA_2)
```

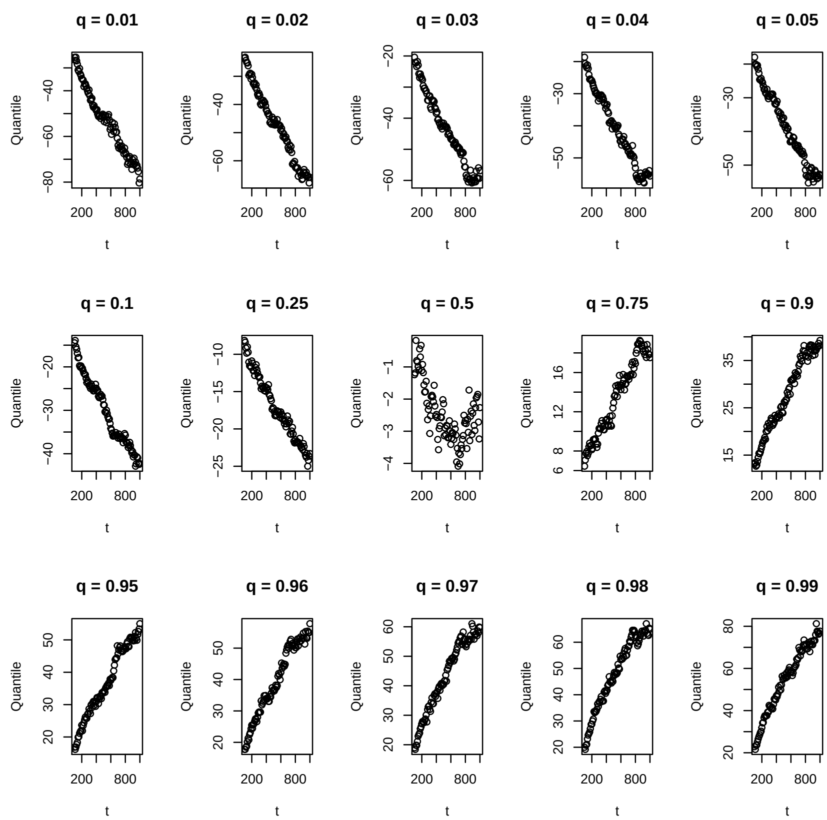


Ambas as estimativas convergem para seus valores verdadeiros (ou seja, 1).

3.
```
N = 1000
NS = 500
XX = c()
QQ = c()
t = seq(100, N, 10)
quantiles = c(0.01, 0.02, 0.03, 0.04,
              0.05, 0.1, 0.25, 0.5,
              0.75, 0.9, 0.95, 0.96,
              0.97, 0.98, 0.99)

for (i in 1:NS) {
  X = fX(N)
  XX = cbind(XX, X)
}
```

```
for (n in t) {
  q = quantile(XX[n,], quantiles, type=1)
  QQ = cbind(QQ, q)
}

QQ = t(QQ)

dim(QQ)

## [1] 91 15
```

4.
```
par(mfrow = c(3, 5))
for (i in 1:15) {
  plot(t, QQ[,i], xlab = "t", ylab = "Quantile",
       main = paste0("q = ", quantiles[i]))
}
```



5.
```r
proxy = QQ[dim(QQ)[1],]

suppressMessages(library(quantmod))
suppressWarnings(suppressMessages(getSymbols("CPN")))

## [1] "CPN"

head(CPN)

##            CPN.Open CPN.High CPN.Low CPN.Close CPN.Volume CPN.Adjusted
## 2012-06-18    16.14    16.34   16.11     16.33    3684281        16.33
## 2012-06-19    16.33    16.47   16.29     16.41    2611352        16.41
## 2012-06-20    16.44    16.44   16.21     16.33    2419812        16.33
## 2012-06-21    16.31    16.34   16.01     16.04    3306956        16.04
## 2012-06-22    16.15    16.17   16.02     16.05    3557785        16.05
## 2012-06-25    16.01    16.06   15.90     15.97    2397340        15.97

dim(CPN)

## [1] 2005    6

CPN = CPN$CPN.Close
CPN = CPN[!is.na(CPN)]
CPNLRet = diff(log(CPN))
CPNLRet = CPNLRet[2:length(CPNLRet)]
```

Para CPN:

```r
X = as.vector(CPN)
phi_1 = fphi_1(X)
sigma_2 = fsigma_2(X, phi_1)
df = fdf(phi_1, sigma_2)
proxy

##         1%         2%         3%         4%         5%        10%        25%
## -78.639800 -65.859243 -56.881600 -55.659585 -53.075828 -42.279639 -23.301486
##        50%        75%        90%        95%        96%        97%        98%
##  -2.266289  17.513441  39.239656  55.014411  57.795559  59.710879  64.945140
##        99%
##  77.689146

df

## [1] 0.0006092523
```

Portanto, não rejeitamos o passeio aleatório com nível 1%. Para CPNLRet:

```
X = as.vector(CPNLRet)
phi_1 = fphi_1(X)
sigma_2 = fsigma_2(X, phi_1)
df = fdf(phi_1, sigma_2)
proxy

##         1%         2%         3%         4%         5%        10%        25%
## -78.639800 -65.859243 -56.881600 -55.659585 -53.075828 -42.279639 -23.301486
##        50%        75%        90%        95%        96%        97%        98%
##  -2.266289  17.513441  39.239656  55.014411  57.795559  59.710879  64.945140
##        99%
##  77.689146

df

## [1] -41.12455
```

Também não rejeitamos para esse nível. Agora verificamos o *augmented DF*:

```
library(tseries)
adf.test(CPN)

##
##  Augmented Dickey-Fuller Test
##
## data:  CPN
## Dickey-Fuller = -0.7708, Lag order = 11, p-value = 0.9642
## alternative hypothesis: stationary

adf.test(CPNLRet)

## Warning in adf.test(CPNLRet):  p-value smaller than printed
p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  CPNLRet
## Dickey-Fuller = -9.0467, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

Neste caso, rejeitamos a hipótese de passeio aleatório com nível 1% apenas para o CPNLRet. Ou seja, a série de log retorno parece ser estacionária.

Problema 6.13 [1]

1.

$$\begin{aligned} X_t &= \mu + X_{t-1} + W_t \\ &= \mu + (\mu + X_{t-2} + W_{t-1}) + W_t \\ &= t\mu + x_0 + \sum_{i=1}^{t} W_i \end{aligned}$$

2.

$$\begin{aligned} \mathbb{E}\{X_t\} &= \mathbb{E}\left\{t\mu + x_0 + \sum_{i=1}^{t} W_i\right\} \\ &= t\mu + x_0 \end{aligned}$$

$$\begin{aligned} \mathrm{var}\{X_t\} &= \mathrm{var}\left\{t\mu + x_0 + \sum_{i=1}^{t} W_i\right\} \\ &= \mathrm{var}\left\{\sum_{i=1}^{t} W_i\right\} \\ &= \sum_{i=1}^{t} \mathrm{var}\{W_i\} \quad (W_i \text{ são i.i.d.}) \\ &= t\sigma^2 \end{aligned}$$

$$\begin{aligned} \gamma(s,t) &= \mathrm{cov}\{X_s, X_t\} \\ &= \mathrm{cov}\left\{s\mu + x_0 + \sum_{i=1}^{s} W_i, t\mu + x_0 + \sum_{i=1}^{t} W_i\right\} \\ &= \mathrm{cov}\left\{\sum_{i=1}^{s} W_i, \sum_{i=1}^{t} W_i\right\} \\ &= \sum_{i=1}^{s}\sum_{i=1}^{t} \mathrm{cov}\{W_i, W_j\} \\ &= \sum_{i=1}^{\min(s,t)} \mathrm{cov}\{W_i, W_i\} \\ &= \min(s,t)\sigma^2 \end{aligned}$$

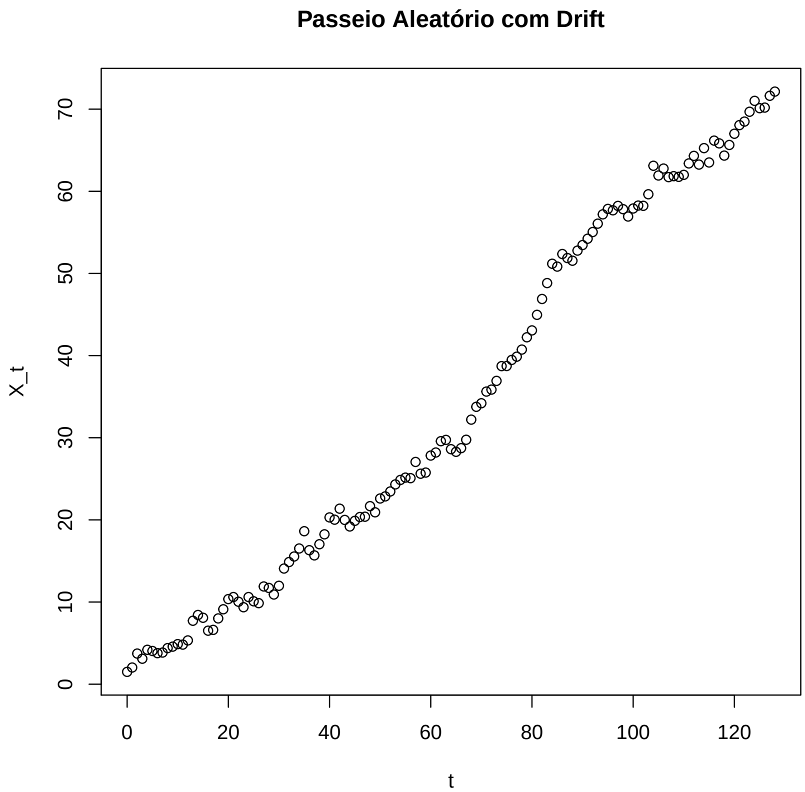
3. $\{X_t\}_t$ não é estacionário, pois as estatísticas, particularmente média e variância, mudam com o tempo.

4.
$$
\begin{aligned}
\nabla X_t &= X_t - X_{t-1} \\
&= t\mu + x_0 + \sum_{i=1}^{t} W_i - \left[(t-1)\mu + x_0 + \sum_{i=1}^{t-1} W_i\right] \\
&= \mu + W_t
\end{aligned}
$$

$$
\begin{aligned}
\gamma(s,t) &= \operatorname{cov}\{\nabla X_s, \nabla X_t\} \\
&= \operatorname{cov}\{\mu + W_s, \mu + W_t\} \\
&= \operatorname{cov}\{W_s, W_t\} \\
&= \begin{cases} \sigma^2, & s = t \\ 0, & s \neq t \end{cases}
\end{aligned}
$$

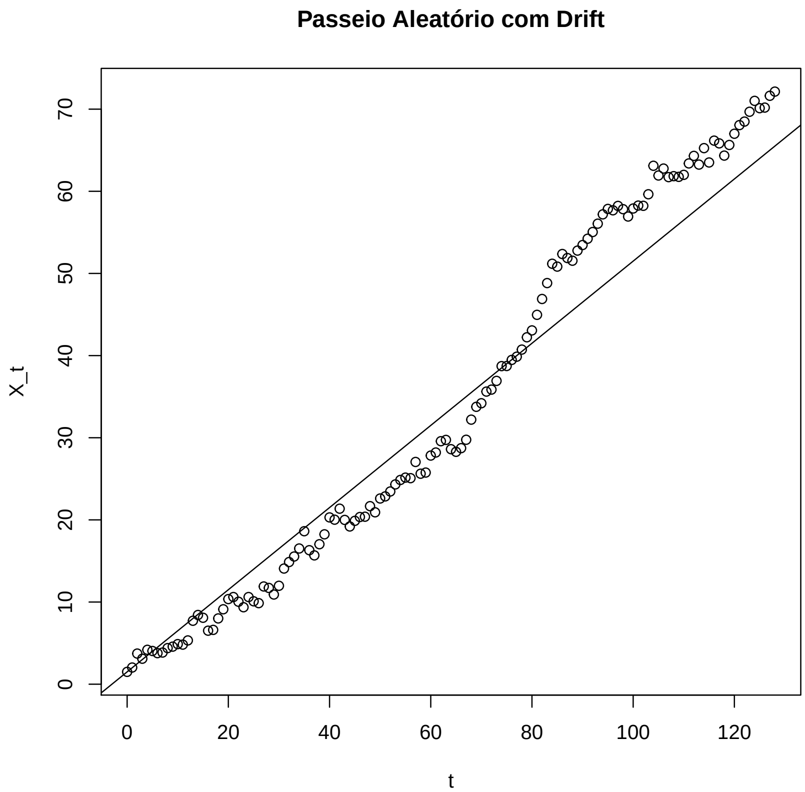
5. Como W_i são i.i.d, $W_i + \mu$ também o são, logo o processo $\{\nabla X_t\}$ é estacionário.

6.
```r
x_0 = 1.5
mu = 0.5
n = 128
WN = rnorm(n)
X = c(x_0)
par(mfrow = c(1, 1))
for (i in 1:n) {
  X = c(X, mu + X[length(X)] + WN[i])
}
plot(seq(0, n), X, main = "Passeio Aleatório com Drift",
     xlab = "t", ylab = "X_t")
```

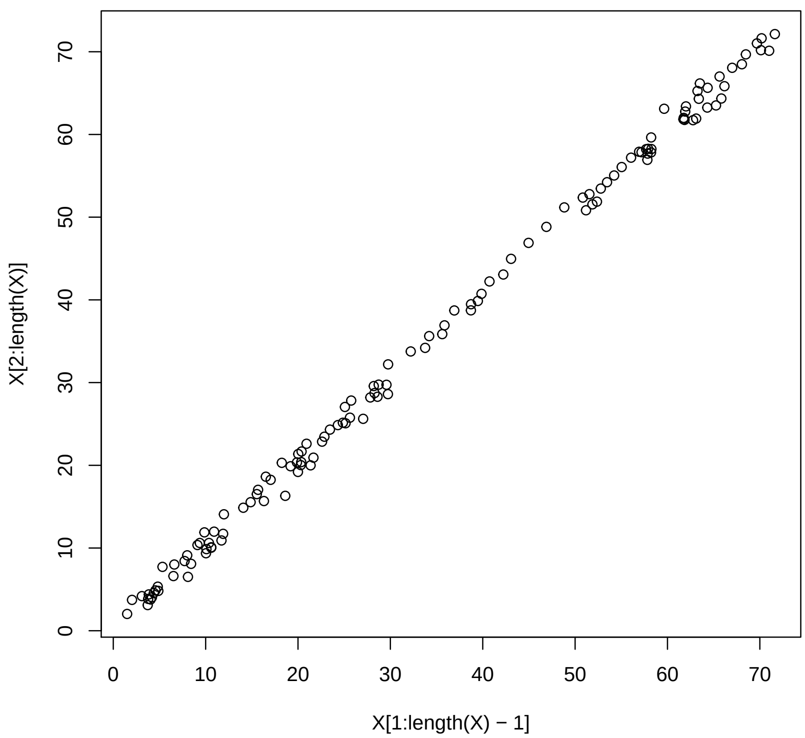


Observe que x_0 representa o ponto inicial do passeio aleatório. No valor esperado, $\mathbb{E}\{X_t\} = t\mu + x_0$, logo x_0 e μ podem ser interpretados como os parâmetros de uma função linear que aproxima esses dados.

```
plot(seq(0, n), X, main = "Passeio Aleatório com Drift",
     xlab = "t", ylab = "X_t")
abline(x_0, mu)
```




```
plot(X[1:length(X)-1], X[2:length(X)])
```



Um *lag plot* de *lag* 1 nos mostra dados muito próximos de uma reta identidade. Isso mostra que, de certa forma, X_{t-1} é muito importante para o cálculo de X_t, porém isso é óbvio pois é a definição do modelo. Na verdade, estamos plotando X_{t-1} vs. $\mu + X_{t-1} + W_t$ e verificamos que o resultado é muito próximo de uma identidade deslocada por μ com ruído de média zero.

Referências

[1] Carmona René. *Statistical analysis of financial data in R.* Springer, 2014.