

Rendimento médio real no Ceará: Uma investigação utilizando VAR/VEC

Francisco Alexandre

2023-12-13

Email: [alecorreia94@alu.ufc.br]

GitHub: [https://github.com/alecorreia94/time-series-analysis-PNAD]

```
# Bibliotecas utilizadas no estudo:
library(dplyr)
library(urca)
library(forecast)
library(ipeadata)
library(vars)
library(readxl)
```

Introdução

As variáveis de ocupação e rendimento médio e inpc (Índice nacional de preço ao consumidor) que mede a variação de preços de uma cesta de produtos e serviços a partir da perspectiva das famílias de baixa renda, que têm rendimento médio de 1 a 5 salários mínimos. Foram obtidas na base de dados do IPECE (Instituto de Pesquisa Econômica do Ceará), ipecedata, enquanto que a variável salário mínimo real foi encontrada no ipeadata.

O primeiro passo da coleta foi inserir essas variáveis de mesmo tamanho e mesma periodicidade em uma planilha do excel. Após a coleta e tabulação desses dados a base foi carregada no software R.

```
# Use a função read_excel para importar os dados da planilha "dados"
caminho <- "G:/Meu Drive/DOUTORADO/ECONOMETRIA II/TRABALHO FINAL/Dados/dados.xlsx"
dados<-readxl::read_excel(caminho, sheet = "dados")
```

```
#deflacionamento do rendimento
rendimento_real <- dados$rendimento_medio / (1 + dados$inpc)
dados$rendimento_medio_real <- rendimento_real
```

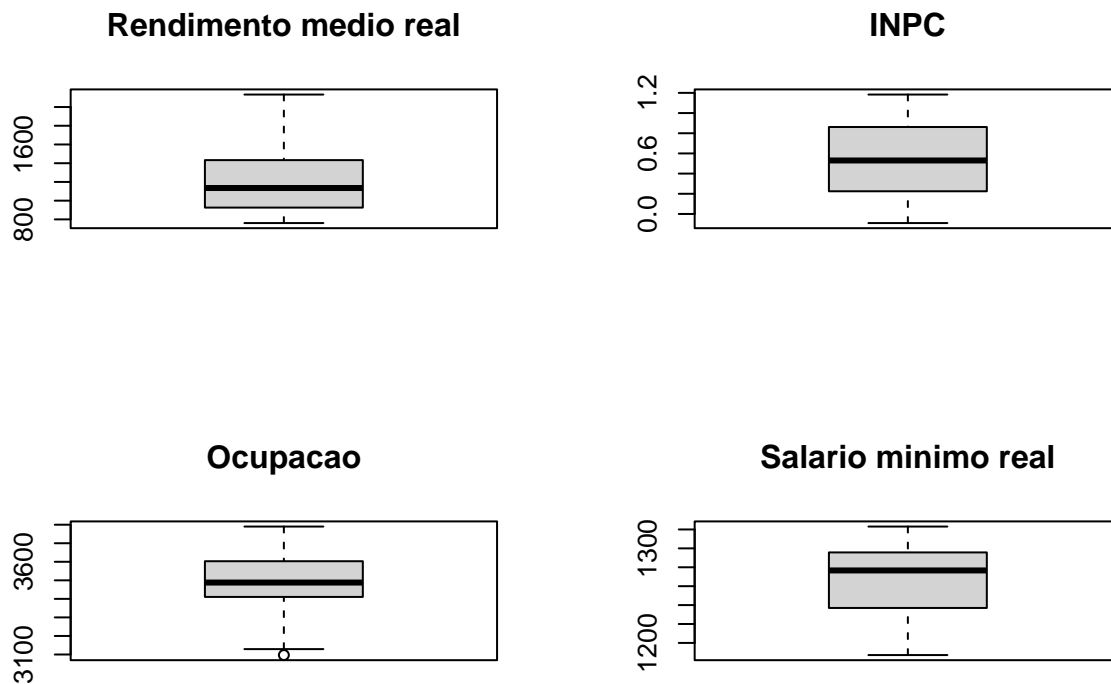
Depois de deflacionada a variável de rendimento médio buscou-se analisar os dados, dessa forma, optou-se por gerar box plots das variáveis a fim de identificar possíveis outliers.

```
rendimento_real_value <- dados$rendimento_medio_real
inpc_value <- dados$inpc
ocup_value <- dados$ocupacao
```

```

sm_value <- dados$salario_minimo_real
par(mfrow=c(2, 2))
boxplot(rendimento_real_value)
title("Rendimento medio real")
boxplot(inpc_value)
title("INPC")
boxplot(ocup_value)
title("Ocupacao")
boxplot(sm_value)
title("Salario minimo real")

```



Apenas a variavel de ocupação possui uma unica observação abaixo dos valores, no entanto, optou-se por não excluí-la e nem modificá-la a fim de na trazer mais prejuizos a análise, e também por seu valor nao ser tao abaixo do menor valor definido no grafico de box plot.

```

#testando outliers
boxplot.stats (rendimento_real_value, coef = 1.5, do.conf = TRUE, do.out = TRUE)

```

```

## $stats
## [1] 760.7634 925.7874 1135.0191 1432.8640 2133.3333
##
## $n
## [1] 32
##
## $conf

```

```
## [1] 993.389 1276.649
##
## $out
## numeric(0)
```

```
boxplot.stats (inpc_value, coef = 1.5, do.conf = TRUE, do.out = TRUE)
```

```
## $stats
## [1] -0.0900000 0.2250000 0.5300000 0.8616667 1.1833333
##
## $n
## [1] 32
##
## $conf
## [1] 0.3521744 0.7078256
##
## $out
## numeric(0)
```

```
boxplot.stats (ocup_value, coef = 1.5, do.conf = TRUE, do.out = TRUE)
```

```
## $stats
## [1] 3129.0 3410.5 3488.0 3603.0 3790.0
##
## $n
## [1] 32
##
## $conf
## [1] 3434.233 3541.767
##
## $out
## [1] 3097
```

```
boxplot.stats (sm_value, coef = 1.5, do.conf = TRUE, do.out = TRUE)
```

```
## $stats
## [1] 1187.163 1236.948 1276.633 1295.703 1323.060
##
## $n
## [1] 32
##
## $conf
## [1] 1260.223 1293.044
##
## $out
## numeric(0)
```

Séries temporais analizadas

```

#séries temporais

serie_rend = ts(dados$rendimento_medio_real, start = c(2014,01), frequency = 4)
serie_inpc = ts(dados$inpc, start = c(2014,01), frequency = 4)
serie_ocup = ts(dados$ocupacao, start = c(2014,01), frequency = 4)
serie_sm = ts(dados$salario_minimo_real, start = c(2014,01), frequency = 4)

par(mfrow=c(2, 2))
# Cria um vetor de datas para o eixo x
datas <- seq(as.Date("2014-01-01"), as.Date("2021-12-01"), by = "quarter")

# Cria o gráfico de séries temporais
plot.ts(dados$rendimento_medio_real, main = "Rendimento médio real",
        xaxt = "n", ylab = "Rendimento Real Medio", xlab = "Tempo (Trimestres)")

# Personalize o eixo x
axis(1, at = 1:length(datas), labels = format(datas, "%Y"), tick = TRUE, las = 2,cex.axis = 0.55)

# Cria o gráfico de séries temporais
plot.ts(dados$inpc, main = "INPC",
        xaxt = "n", ylab = "INPC", xlab = "Tempo (Trimestres)")

# Personalize o eixo x
axis(1, at = 1:length(datas), labels = format(datas, "%Y"), tick = TRUE, las = 2,cex.axis = 0.55)

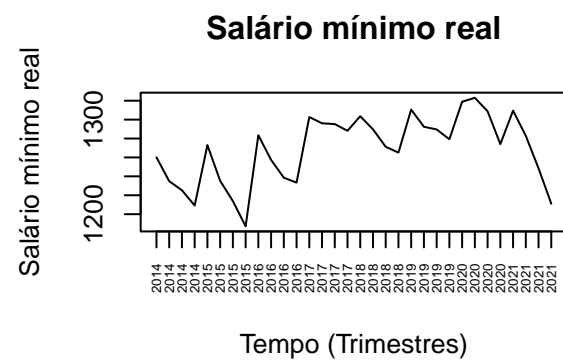
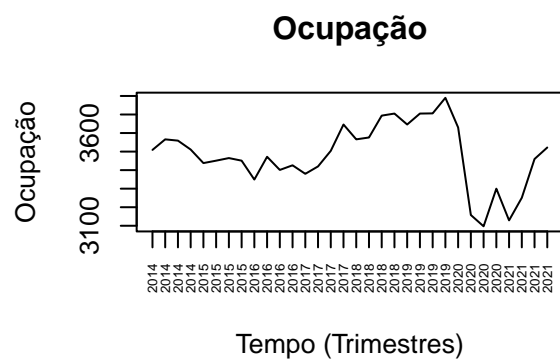
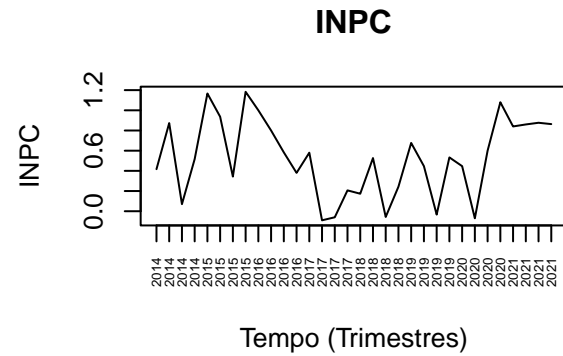
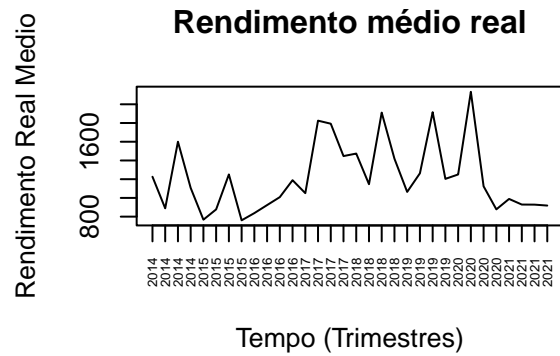
# Cria o gráfico de séries temporais
plot.ts(dados$ocupacao, main = "Ocupação",
        xaxt = "n", ylab = "Ocupação", xlab = "Tempo (Trimestres)")

# Personalize o eixo x
axis(1, at = 1:length(datas), labels = format(datas, "%Y"), tick = TRUE, las = 2,cex.axis = 0.55)

# Cria o gráfico de séries temporais
plot.ts(dados$salario_minimo_real, main = "Salário mínimo real",
        xaxt = "n", ylab = "Salário mínimo real", xlab = "Tempo (Trimestres)")

# Personalize o eixo x
axis(1, at = 1:length(datas), labels = format(datas, "%Y"), tick = TRUE, las = 2,cex.axis = 0.55)

```



Pela análise visual das séries não observa-se flutuações condizentes com sazonalidade ou tendência. Mas cabe uma análise mais profunda dos dados para verificarmos isso. Abaixo temos o correlograma das séries.

Correlogramas

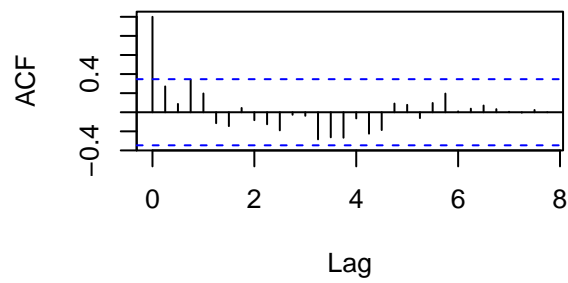
```
par(mfrow=c(2, 2))

#correlogramas

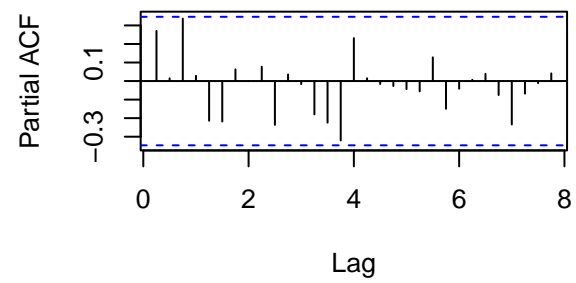
rend_FAC <- acf(serie_rend, lag=32, main="Rendimento médio real" )
rend_FACP <- pacf(serie_rend, lag=32, main="Rendimento médio real")

inpc_FAC <- acf(serie_inpc, lag=32, main="INPC")
inpc_FACP <- pacf(serie_inpc, lag=32, main="INPC")
```

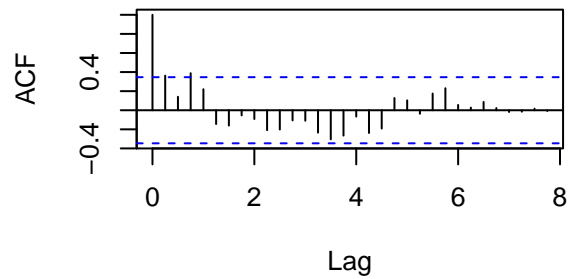
Rendimento médio real



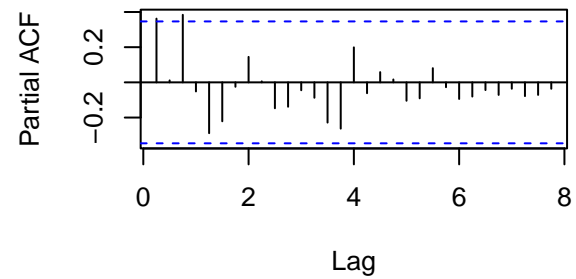
Rendimento médio real



INPC

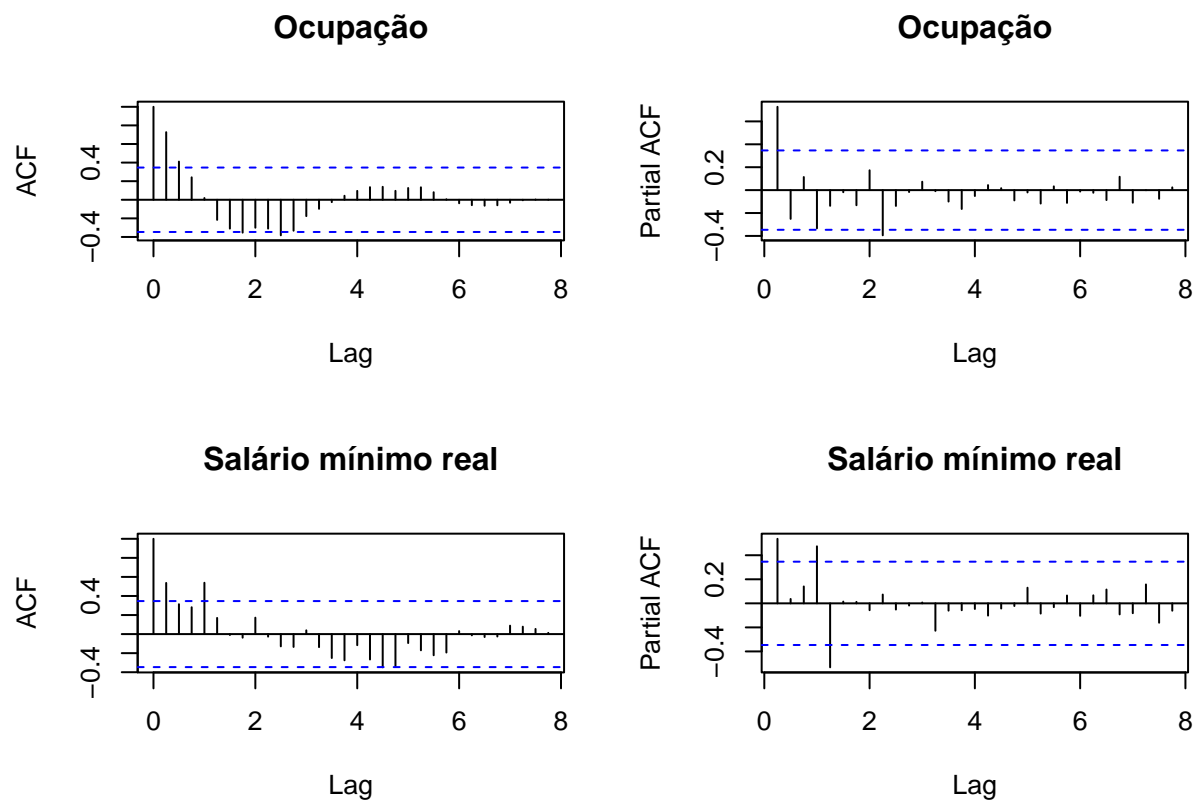


INPC



```
ocup_FAC <- acf(serie_ocup, lag=32, main="Ocupação")
ocup_FACP <- pacf(serie_ocup, lag=32, main="Ocupação")

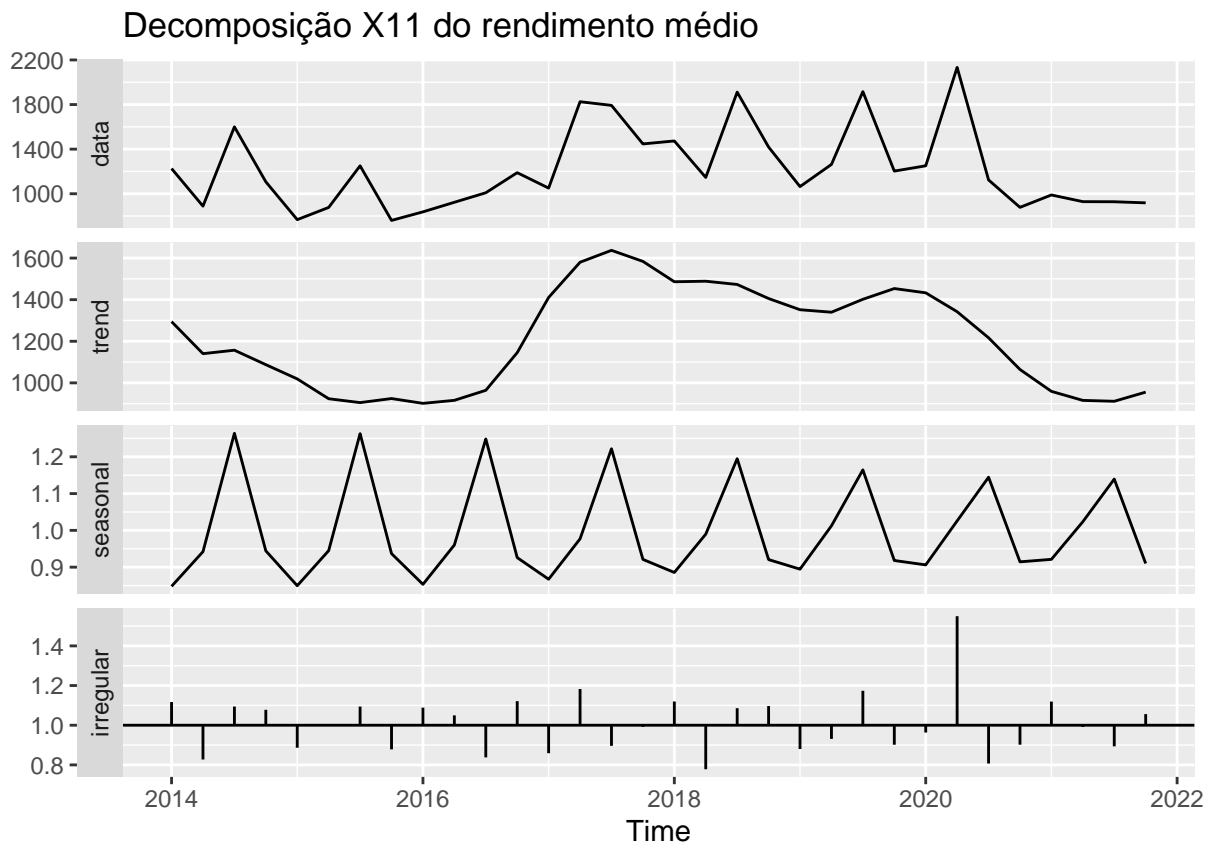
sm_FAC <- acf(serie_sm, lag=32, main="Salário mínimo real")
sm_FACP <- pacf(serie_sm, lag=32, main="Salário mínimo real")
```



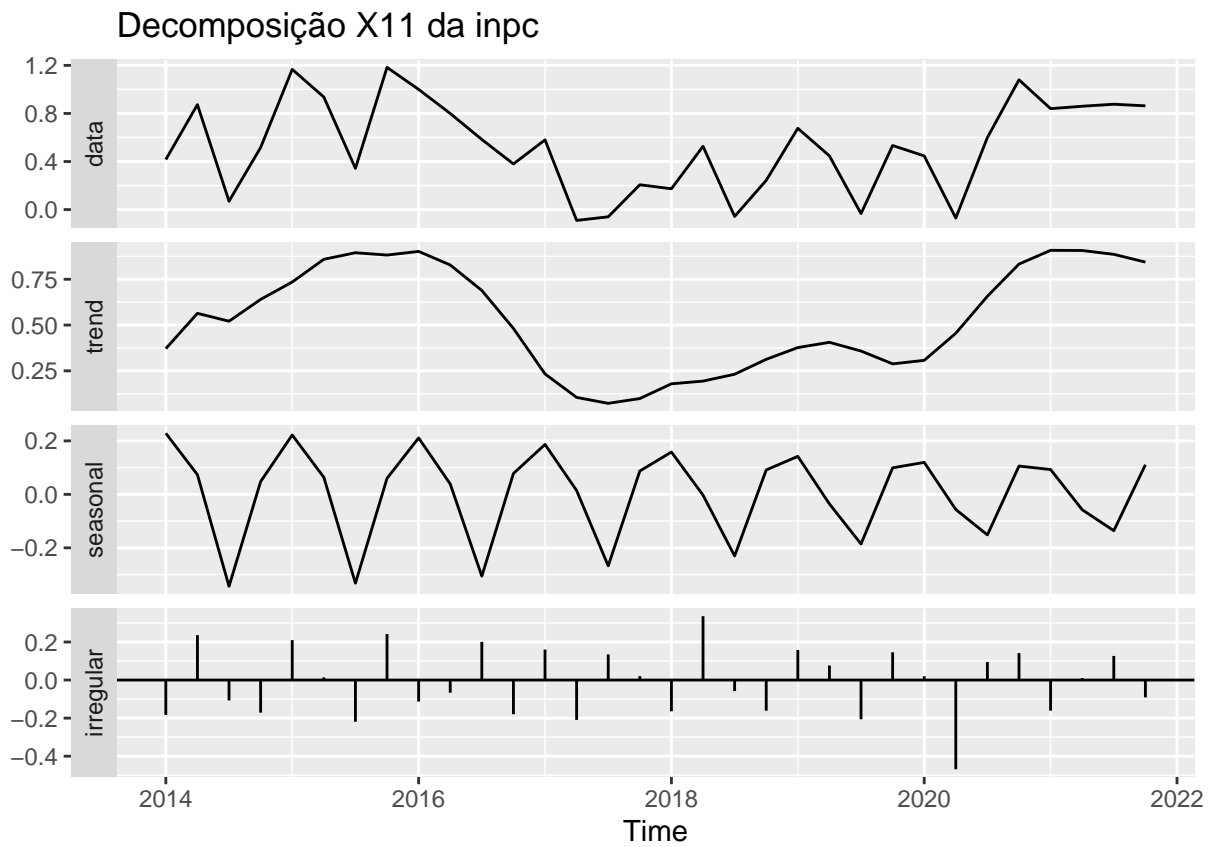
Nota-se que não há picos periódicos significativos nos correlogramas e portanto não parece ter nenhum tipo de sazonalidade identificada. No entanto, ao fazer a decomposição da série percebe-se sazonalidade em todas as séries e tendência na série de salário mínimo.

Decomposição das Séries

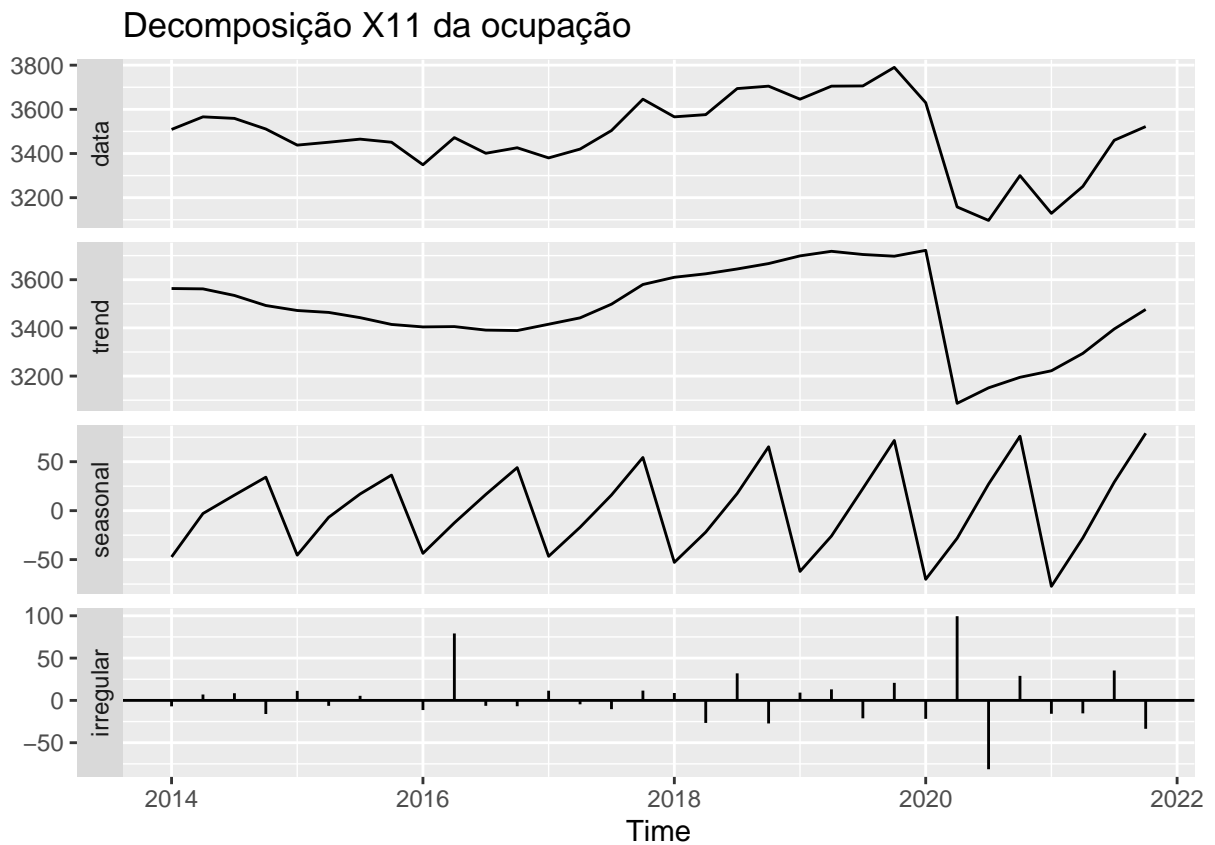
```
# decomposição pelo X11 do Census Bureau
library(seasonal)
fit <- seas(serie_rend, x11 = "")
library(fpp2)
autoplot(fit) + ggtitle("Decomposição X11 do rendimento médio")
```



```
# decomposição pelo X11 do Census Bureau  
library(seasonal)  
fit <- seas(serie_inpc, x11 = "")  
library(fpp2)  
autoplot(fit) + ggtitle("Decomposição X11 da inpc")
```

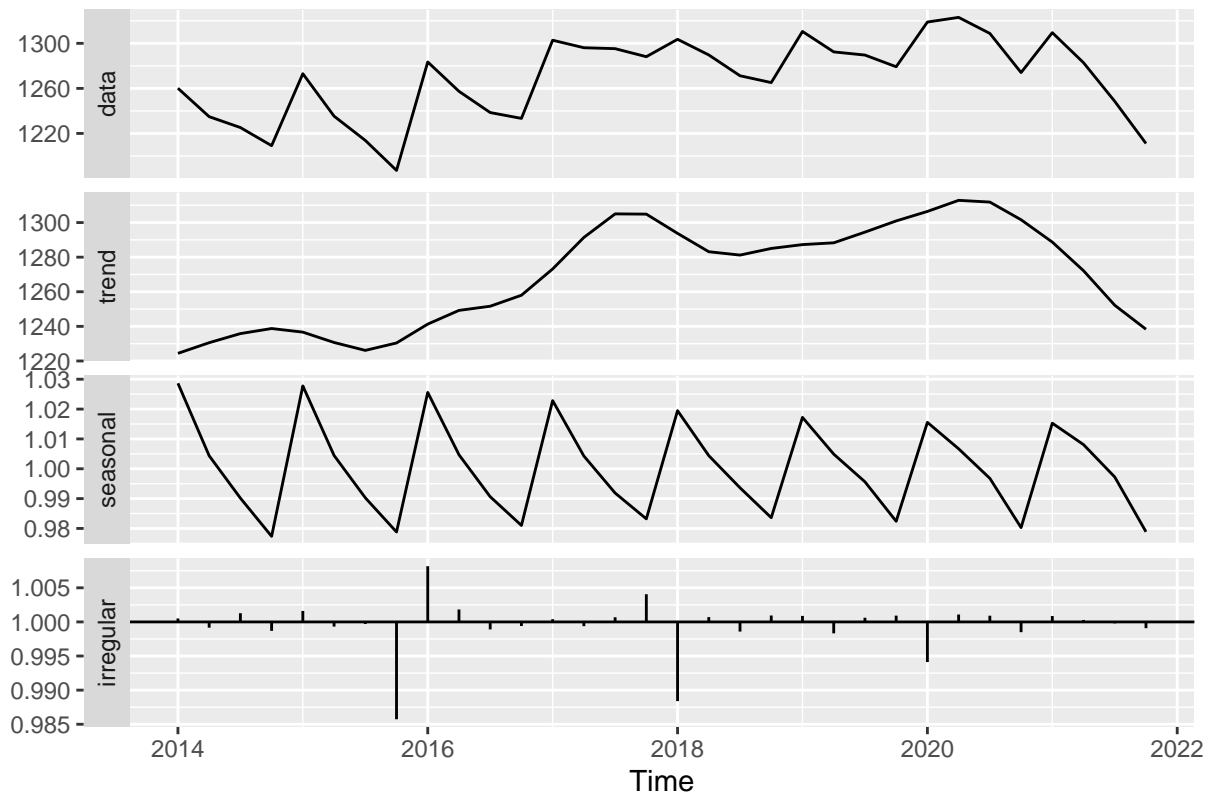



```
# decomposição pelo X11 do Census Bureau  
library(seasonal)  
fit <- seas(serie_ocup, x11 = "")  
library(fpp2)  
autoplot(fit) + ggtitle("Decomposição X11 da ocupação")
```



```
# decomposição pelo X11 do Census Bureau
library(seasonal)
fit <- seas(serie_sm, x11 = "")
library(fpp2)
autoplot(fit) + ggtitle("Decomposição X11 do salário mínimo")
```

Decomposição X11 do salário mínimo



Agora para sanar todas as dúvidas precisa-se testar se há alguma tendência nessas séries Para isso, usa-se o teste estatístico abaixo.

Teste de Tendência

```
library(trend)
library(Kendall)
mk.test(serie_rend)
```

```
##
## Mann-Kendall trend test
##
## data:  serie_rend
## z = 0.50271, n = 32, p-value = 0.6152
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##          S          varS          tau
## 3.200000e+01 3.802667e+03 6.451613e-02
```

```
mk.test(serie_inpc)
```

```
##
## Mann-Kendall trend test
```

```
##
## data:  serie_inpc
## z = 0.21084, n = 32, p-value = 0.833
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##          S          varS          tau
## 14.0000000 3801.6666667  0.0282543
```

```
mk.test(serie_ocup)
```

```
##
## Mann-Kendall trend test
##
## data:  serie_ocup
## z = 0.048675, n = 32, p-value = 0.9612
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##          S          varS          tau
## 4.000000e+00 3.798667e+03 8.097232e-03
```

```
mk.test(serie_sm)
```

```
##
## Mann-Kendall trend test
##
## data:  serie_sm
## z = 2.5784, n = 32, p-value = 0.009925
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##          S          varS          tau
## 160.0000000 3802.6666667  0.3225806
```

Apenas a variável de salário mínimo real apresenta tendência definida através do teste de Mann_Kendall, é o unico que rejeita a hipotese nula de não tendência.

Desse modo, agora vamos aos testes para decidir entre estimar um VAR ou um VEC. Para isso, primeiro verifica-se a hipótese de raiz unitaria das séries.

Teste de Raiz Unitária

```
ur_rendimento <- ur.df(y = serie_rend, lags = 3, type = "none", selectlags = "AIC")
ur_inpc <- ur.df(y = serie_inpc, lags = 3, type = "none", selectlags = "AIC")
ur_ocup <- ur.df(y = serie_ocup, lags = 3, type = "none", selectlags = "AIC")
ur_sm <- ur.df(y = serie_sm, lags = 3, type = "trend", selectlags = "AIC")
summary(ur_rendimento)
```

```
##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
```

```
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -492.2 -215.1  -55.3   216.2   828.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1      -0.03000    0.05384  -0.557  0.58238
## z.diff.lag1  -0.49384    0.16798  -2.940  0.00697 **
## z.diff.lag2  -0.50351    0.15983  -3.150  0.00420 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 361.7 on 25 degrees of freedom
## Multiple R-squared:  0.3726, Adjusted R-squared:  0.2973
## F-statistic: 4.949 on 3 and 25 DF,  p-value: 0.007824
##
##
## Value of test-statistic is: -0.5572
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.62 -1.95 -1.61
```

```
summary(ur_inpc)
```

```
##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68686 -0.16973  0.05961  0.27280  0.49931
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1      -0.02843    0.11026  -0.258  0.7986
## z.diff.lag1  -0.38158    0.17805  -2.143  0.0420 *
## z.diff.lag2  -0.53935    0.15722  -3.431  0.0021 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.3363 on 25 degrees of freedom
## Multiple R-squared:  0.3876, Adjusted R-squared:  0.3141
## F-statistic: 5.275 on 3 and 25 DF,  p-value: 0.005877
##
##
## Value of test-statistic is: -0.2579
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.62 -1.95 -1.61
```

```
summary(ur_ocup)
```

```
##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -427.21  -64.78    8.82   78.38  201.39
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1      -0.001270   0.007284  -0.174   0.863
## z.diff.lag1   0.100602   0.192362   0.523   0.606
## z.diff.lag2  -0.286742   0.202077  -1.419   0.168
##
## Residual standard error: 133.9 on 25 degrees of freedom
## Multiple R-squared:  0.08129,    Adjusted R-squared:  -0.02896
## F-statistic: 0.7374 on 3 and 25 DF,  p-value: 0.5397
##
##
## Value of test-statistic is: -0.1744
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.62 -1.95 -1.61
```

```
summary(ur_sm)
```

```
##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
##
```

```
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -35.40 -21.87   4.91  20.43  29.93
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -412.1585   450.8479  -0.914  0.370531
## z.lag.1       0.3594     0.3695   0.972  0.341374
## tt          -2.4104     1.2606  -1.912  0.068977 .
## z.diff.lag1  -0.9137     0.3347  -2.730  0.012236 *
## z.diff.lag2  -0.8264     0.2633  -3.138  0.004775 **
## z.diff.lag3  -0.7707     0.1956  -3.940  0.000698 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.23 on 22 degrees of freedom
## Multiple R-squared:  0.573, Adjusted R-squared:  0.476
## F-statistic: 5.905 on 5 and 22 DF, p-value: 0.001318
##
##
## Value of test-statistic is: 0.9725 2.0706 2.9452
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

A hipótese nula de não estacionariedade (a série temporal de rendimento médio real contém pelo menos uma raiz unitária) não pode ser rejeitada ao nível de significância de 5%, pois a estatística de teste de -0.5572 não é mais negativa do que o valor crítico de -1.95. O mesmo se aplica as outras variáveis. Desse modo, agora pode ser identificado o número de lags.

Identificação

```
library(vars)
base <- cbind(serie_rend,serie_inpc,serie_ocup,serie_sm)
VARselect(base, lag.max = 4, type = "both", season = 4)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      4      4      1      3
##
## $criteria
##              1              2              3              4
```

```
## AIC(n) 2.287647e+01 2.263342e+01 2.137174e+01 2.065303e+01
## HQ(n) 2.340010e+01 2.338978e+01 2.236082e+01 2.187484e+01
## SC(n) 2.458930e+01 2.510752e+01 2.460710e+01 2.464964e+01
## FPE(n) 9.465849e+09 9.188572e+09 4.163693e+09 5.547826e+09
```

Percebe-se que todos os critérios identificam 4 lags das variáveis em nível o que pode indicar um VAR(4). Mas antes de estimá-lo faz-se agora um teste de cointegração para entendermos se não seria melhor a utilização de VEC.

Teste de cointegração

```
library("urca")
jotest=ca.jo(base, type="trace", K=4, ecdet="none", spec="longrun", season = 4)
summary(jotest)
```

```
##
## #####
## # Johansen-Procedure #
## #####
##
## Test type: trace statistic , with linear trend
##
## Eigenvalues (lambda):
## [1] 0.7090136 0.5578248 0.4133185 0.1161615
##
## Values of teststatistic and critical values of test:
##
##          test 10pct  5pct  1pct
## r <= 3 |   3.46   6.50   8.18 11.65
## r <= 2 |  18.39  15.66  17.95 23.52
## r <= 1 |  41.24  28.71  31.52 37.22
## r = 0  |  75.80  45.23  48.28 55.43
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##          serie_rend.l4 serie_inpc.l4 serie_ocup.l4 serie_sm.l4
## serie_rend.l4      1.0000000      1.0000000      1.0000000      1.0000000
## serie_inpc.l4    684.3108251    1073.5727598    -280.020370    751.4759408
## serie_ocup.l4      0.0279652     -0.3325244     -1.324653      0.3099386
## serie_sm.l4     -2.2571181      0.7114762     -6.644354     6.5634666
##
## Weights W:
## (This is the loading matrix)
##
##          serie_rend.l4 serie_inpc.l4 serie_ocup.l4  serie_sm.l4
## serie_rend.d  -4.337283199 -1.4433856292 -0.5784209337 -4.782194e-02
## serie_inpc.d   0.005607409  0.0005076505  0.0006335701  7.992624e-05
## serie_ocup.d  -0.657629282  0.3739701757  0.3027724177 -4.094168e-02
## serie_sm.d     0.064273067 -0.0690422305 -0.0244376759 -1.379827e-02
```


O teste do traço rejeita a hipótese nula de que há zero vetores de cointegração e dessa forma nos mostra que em até $r \leq 1$, o valor do teste é maior que o valor crítico com um nível de confiança de 1%, indicando que existe 2 vetores de cointegração entre essas variáveis, e dessa forma, o melhor modelo a ser utilizado seria o VEC.

Para determinar o número de lags foi utilizado a função VARselect que indica o número de defasagens que minimiza os critérios de identificação Akaike, Schwarz, Hannan-Quin. O qual observa-se acima como resultado 4 defasagens.

Modelo VEC

```
library(tsDyn)
vecm <- VECM(base, lag = 4, r = 2, estim = "ML")
summary(vecm)
```

```
## #####
## ###Model VECM
## #####
## Full sample size: 32      End sample size: 27
## Number of variables: 4   Number of estimated slope parameters 76
## AIC 537.6769      BIC 641.3438      SSR 1255456
## Cointegrating vector (estimated by ML):
##      serie_rend serie_inpc      serie_ocup      serie_sm
## r1           1           0 -0.5205991769 -5.229949094
## r2           0           1  0.0009563336  0.003977484
##
##
##
##      ECT1      ECT2
## Equation serie_rend -2.4943(3.8565)      -1193.6383(2383.9348)
## Equation serie_inpc 0.0029(0.0027)      1.1974(1.6567)
## Equation serie_ocup -0.0301(1.0715)      -320.4205(662.3869)
## Equation serie_sm  0.4791(0.1613)*      336.7285(99.7288)**
##      Intercept      serie_rend -1
## Equation serie_rend -7425.9059(9982.6194)      0.6360(3.8714)
## Equation serie_inpc 10.3410(6.9372)      -0.0020(0.0027)
## Equation serie_ocup 2631.3211(2773.7152)      0.1461(1.0757)
## Equation serie_sm  469.7223(417.6099)      -0.4467(0.1620)*
##      serie_inpc -1      serie_ocup -1
## Equation serie_rend 615.5836(2740.1199)      -0.7400(1.3198)
## Equation serie_inpc -1.6574(1.9042)      0.0009(0.0009)
## Equation serie_ocup 411.5221(761.3545)      0.9170(0.3667)*
## Equation serie_sm  -351.8056(114.6294)*      -0.0115(0.0552)
##      serie_sm -1      serie_rend -2
## Equation serie_rend -2.3737(10.4943)      0.0558(3.6619)
## Equation serie_inpc 0.0032(0.0073)      -0.0012(0.0025)
## Equation serie_ocup 1.7645(2.9159)      -0.0804(1.0175)
## Equation serie_sm  0.2741(0.4390)      -0.4380(0.1532)*
##      serie_inpc -2      serie_ocup -2
## Equation serie_rend 215.4820(3084.8873)      1.7720(1.5130)
## Equation serie_inpc -1.1620(2.1438)      -0.0020(0.0011)
## Equation serie_ocup 93.2334(857.1497)      -0.3953(0.4204)
## Equation serie_sm  -438.6745(129.0523)**      0.0244(0.0633)
```

```
##               serie_sm -2               serie_rend -3
## Equation serie_rend 6.5239(9.4637)         -0.6816(2.8948)
## Equation serie_inpc -0.0076(0.0066)         0.0002(0.0020)
## Equation serie_ocup 0.0781(2.6295)         -0.5308(0.8043)
## Equation serie_sm   0.4757(0.3959)         -0.3365(0.1211)*
##               serie_inpc -3               serie_ocup -3
## Equation serie_rend -653.8622(2720.8412)    -0.4384(1.2186)
## Equation serie_inpc 0.1834(1.8908)           0.0006(0.0008)
## Equation serie_ocup -328.9914(755.9978)      0.4449(0.3386)
## Equation serie_sm   -372.0962(113.8229)*    -0.0758(0.0510)
##               serie_sm -3               serie_rend -4
## Equation serie_rend 2.5461(5.6220)         -0.7799(1.6258)
## Equation serie_inpc -0.0033(0.0039)         0.0006(0.0011)
## Equation serie_ocup 1.6714(1.5621)         -0.1362(0.4517)
## Equation serie_sm   -0.1146(0.2352)        -0.1630(0.0680)*
##               serie_inpc -4               serie_ocup -4
## Equation serie_rend -618.0893(1633.4374)    1.2877(1.5074)
## Equation serie_inpc 0.4323(1.1351)          -0.0012(0.0010)
## Equation serie_ocup -63.5893(453.8578)      -0.4551(0.4188)
## Equation serie_sm   -191.8875(68.3327)*     -0.1418(0.0631).
##               serie_sm -4
## Equation serie_rend 5.3816(7.8943)
## Equation serie_inpc -0.0073(0.0055)
## Equation serie_ocup -2.0108(2.1935)
## Equation serie_sm   -0.1513(0.3302)
```

Apenas a variável de salário mínimo parece se ajustar para uma condição de equilíbrio de forma positiva, fato verificado nos coeficientes ECT1 e ECT2. Essas variáveis parecem ter cointegrações fracas, apenas a variável de salário mínimo possui impactos negativos e significantes, com a renda e inpc em pelo menos 5%. exceto em ocupação que é afetada pela sua própria defasagem positivamente.

Deste modo, apesar do teste de cointegração, vou rodar também um VAR(4).

Modelo VAR(4)

```
modelovar <- VAR(base, p = 4, type = "both", season = 4)
summary(modelovar)
```

```
##
## VAR Estimation Results:
## =====
## Endogenous variables: serie_rend, serie_inpc, serie_ocup, serie_sm
## Deterministic variables: both
## Sample size: 28
## Log Likelihood: -364.064
## Roots of the characteristic polynomial:
## 1.046 1.046 0.9985 0.9985 0.9529 0.9529 0.9076 0.9076 0.8981 0.8981 0.8162 0.8162 0.8108 0.8108 0.37
## Call:
## VAR(y = base, p = 4, type = "both", season = 4L)
##
##
## Estimation results for equation serie_rend:
```

```
## =====
## serie_rend = serie_rend.l1 + serie_inpc.l1 + serie_ocup.l1 + serie_sm.l1 + serie_rend.l2 + serie_inpc.l2 + serie_ocup.l2 + serie_sm.l2 + serie_rend.l3 + serie_inpc.l3 + serie_ocup.l3 + serie_sm.l3 + serie_rend.l4 + serie_inpc.l4 + serie_ocup.l4 + serie_sm.l4 + const + trend + sd1 + sd2 + sd3
##
##          Estimate Std. Error t value Pr(>|t|)
## serie_rend.l1 -1.744e+00  8.546e-01 -2.040  0.0807 .
## serie_inpc.l1 -1.459e+03  7.414e+02 -1.967  0.0898 .
## serie_ocup.l1 -2.989e-01  1.388e+00 -0.215  0.8356
## serie_sm.l1    9.109e+00  7.407e+00  1.230  0.2585
## serie_rend.l2 -5.073e-01  1.164e+00 -0.436  0.6760
## serie_inpc.l2 -3.522e+02  9.101e+02 -0.387  0.7103
## serie_ocup.l2  2.775e+00  2.017e+00  1.376  0.2112
## serie_sm.l2    2.822e+00  7.521e+00  0.375  0.7186
## serie_rend.l3 -7.255e-01  1.113e+00 -0.652  0.5354
## serie_inpc.l3 -5.061e+02  1.134e+03 -0.446  0.6687
## serie_ocup.l3 -3.424e+00  1.716e+00 -1.996  0.0862 .
## serie_sm.l3    1.578e+00  8.677e+00  0.182  0.8609
## serie_rend.l4 -5.834e-01  1.275e+00 -0.457  0.6612
## serie_inpc.l4  2.772e+02  1.414e+03  0.196  0.8501
## serie_ocup.l4  2.622e+00  1.375e+00  1.906  0.0983 .
## serie_sm.l4    1.533e+01  1.002e+01  1.530  0.1699
## const        -3.482e+04  2.301e+04 -1.513  0.1740
## trend         -5.363e+01  4.636e+01 -1.157  0.2853
## sd1           -6.149e+02  6.112e+02 -1.006  0.3479
## sd2           -4.206e+02  1.047e+03 -0.402  0.6999
## sd3            6.541e+02  8.467e+02  0.773  0.4651
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 264.9 on 7 degrees of freedom
## Multiple R-Squared:  0.876,    Adjusted R-squared:  0.5216
## F-statistic: 2.472 on 20 and 7 DF,  p-value: 0.1117
##
##
## Estimation results for equation serie_inpc:
## =====
## serie_inpc = serie_rend.l1 + serie_inpc.l1 + serie_ocup.l1 + serie_sm.l1 + serie_rend.l2 + serie_inpc.l2 + serie_ocup.l2 + serie_sm.l2 + serie_rend.l3 + serie_inpc.l3 + serie_ocup.l3 + serie_sm.l3 + serie_rend.l4 + serie_inpc.l4 + serie_ocup.l4 + serie_sm.l4 + const + trend + sd1 + sd2 + sd3
##
##          Estimate Std. Error t value Pr(>|t|)
## serie_rend.l1  0.0016510  0.0006780   2.435  0.0451 *
## serie_inpc.l1  1.0201152  0.5881666   1.734  0.1264
## serie_ocup.l1  0.0005693  0.0011008   0.517  0.6210
## serie_sm.l1   -0.0135573  0.0058764  -2.307  0.0544 .
## serie_rend.l2  0.0010582  0.0009232   1.146  0.2894
## serie_inpc.l2  0.5566470  0.7219999   0.771  0.4659
## serie_ocup.l2 -0.0034293  0.0015998  -2.144  0.0693 .
## serie_sm.l2   -0.0096462  0.0059665  -1.617  0.1500
## serie_rend.l3  0.0012099  0.0008832   1.370  0.2130
## serie_inpc.l3  0.9650373  0.8993392   1.073  0.3188
## serie_ocup.l3  0.0030626  0.0013611   2.250  0.0592 .
## serie_sm.l3    0.0020123  0.0068840   0.292  0.7785
## serie_rend.l4  0.0007322  0.0010116   0.724  0.4926
## serie_inpc.l4 -0.0495864  1.1215113  -0.044  0.9660
## serie_ocup.l4 -0.0016933  0.0010909  -1.552  0.1646
```

```

## serie_sm.l4    -0.0142907  0.0079484  -1.798   0.1152
## const         42.6475632 18.2583728   2.336   0.0522 .
## trend         0.0632210  0.0367826   1.719   0.1293
## sd1           0.5551614  0.4848740   1.145   0.2899
## sd2           0.8522475  0.8307456   1.026   0.3391
## sd3           0.0035881  0.6716924   0.005   0.9959
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.2102 on 7 degrees of freedom
## Multiple R-Squared: 0.9264, Adjusted R-squared: 0.716
## F-statistic: 4.404 on 20 and 7 DF, p-value: 0.026
##
##
## Estimation results for equation serie_ocup:
## =====
## serie_ocup = serie_rend.l1 + serie_inpc.l1 + serie_ocup.l1 + serie_sm.l1 + serie_rend.l2 + serie_inpc.l2
##
##               Estimate Std. Error t value Pr(>|t|)
## serie_rend.l1    0.16358    0.35629   0.459   0.660
## serie_inpc.l1   279.77992   309.07435   0.905   0.395
## serie_ocup.l1    1.24702    0.57847   2.156   0.068 .
## serie_sm.l1      0.95041    3.08797   0.308   0.767
## serie_rend.l2   -0.26977    0.48512  -0.556   0.595
## serie_inpc.l2  -317.58599   379.40209  -0.837   0.430
## serie_ocup.l2   -1.02334    0.84067  -1.217   0.263
## serie_sm.l2     -2.67262    3.13532  -0.852   0.422
## serie_rend.l3   -0.47566    0.46409  -1.025   0.340
## serie_inpc.l3  -520.92889   472.59172  -1.102   0.307
## serie_ocup.l3    1.06610    0.71525   1.491   0.180
## serie_sm.l3      0.92044    3.61745   0.254   0.806
## serie_rend.l4    0.05413    0.53160   0.102   0.922
## serie_inpc.l4  -249.57651   589.34044  -0.423   0.685
## serie_ocup.l4   -1.00061    0.57327  -1.745   0.124
## serie_sm.l4     -4.25940    4.17677  -1.020   0.342
## const          9701.39420  9594.55100   1.011   0.346
## trend           14.68938   19.32880   0.760   0.472
## sd1            -29.12314   254.79535  -0.114   0.912
## sd2           -150.84295   436.54663  -0.346   0.740
## sd3           -164.87168   352.96613  -0.467   0.655
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 110.4 on 7 degrees of freedom
## Multiple R-Squared: 0.9044, Adjusted R-squared: 0.6312
## F-statistic: 3.31 on 20 and 7 DF, p-value: 0.05535
##
##
## Estimation results for equation serie_sm:
## =====
## serie_sm = serie_rend.l1 + serie_inpc.l1 + serie_ocup.l1 + serie_sm.l1 + serie_rend.l2 + serie_inpc.l2
##

```

```

##               Estimate Std. Error t value Pr(>|t|)
## serie_rend.l1  1.052e-02  6.628e-02   0.159   0.878
## serie_inpc.l1 -1.317e+01  5.750e+01  -0.229   0.825
## serie_ocup.l1  1.114e-02  1.076e-01   0.104   0.920
## serie_sm.l1    5.062e-01  5.744e-01   0.881   0.407
## serie_rend.l2 -5.374e-02  9.024e-02  -0.596   0.570
## serie_inpc.l2 -5.410e+01  7.058e+01  -0.766   0.468
## serie_ocup.l2 -1.671e-02  1.564e-01  -0.107   0.918
## serie_sm.l2    3.124e-01  5.832e-01   0.536   0.609
## serie_rend.l3 -3.691e-02  8.633e-02  -0.428   0.682
## serie_inpc.l3 -1.330e+01  8.791e+01  -0.151   0.884
## serie_ocup.l3  2.813e-04  1.331e-01   0.002   0.998
## serie_sm.l3    8.359e-03  6.729e-01   0.012   0.990
## serie_rend.l4  2.872e-02  9.889e-02   0.290   0.780
## serie_inpc.l4  3.620e+01  1.096e+02   0.330   0.751
## serie_ocup.l4  5.557e-02  1.066e-01   0.521   0.618
## serie_sm.l4   -2.459e-02  7.770e-01  -0.032   0.976
## const         1.587e+02  1.785e+03   0.089   0.932
## trend         2.439e-01  3.596e+00   0.068   0.948
## sd1           8.033e+01  4.740e+01   1.695   0.134
## sd2           5.080e+01  8.121e+01   0.626   0.551
## sd3           1.648e+01  6.566e+01   0.251   0.809
##
##
## Residual standard error: 20.55 on 7 degrees of freedom
## Multiple R-Squared:  0.9095, Adjusted R-squared:  0.6508
## F-statistic: 3.516 on 20 and 7 DF, p-value: 0.04742
##
##
##
## Covariance matrix of residuals:
##               serie_rend serie_inpc serie_ocup serie_sm
## serie_rend    70182.81  -47.01752 -18189.955 4137.184
## serie_inpc     -47.02    0.04417    9.372   -3.505
## serie_ocup   -18189.96    9.37239  12197.728 -393.299
## serie_sm       4137.18   -3.50478   -393.299  422.102
##
## Correlation matrix of residuals:
##               serie_rend serie_inpc serie_ocup serie_sm
## serie_rend      1.0000   -0.8444   -0.6217   0.7601
## serie_inpc     -0.8444    1.0000    0.4038  -0.8117
## serie_ocup     -0.6217    0.4038    1.0000  -0.1733
## serie_sm        0.7601   -0.8117   -0.1733    1.0000

```

Através do VAR(4) podemos perceber que há impacto negativo na série de rendimento médio real através da própria defasagem dela e pelo Índice de inflação. Além disso, a variável de inflação é afetada positivamente pelo rendimento médio real. Já a variável ocupação e salário mínimo real não parece ser afetada por nenhuma das demais variáveis.

No geral, muito poucos coeficientes são estatisticamente significativos em níveis convencionais ($p < 0,05$). Isso sugere evidências limitadas de relações fortes entre as variáveis o modelo.

Teste de autocorrelação serial

```
vecvar <- vec2var(jotest, r = 1)
serial.test(vecvar, lags.pt = 15)
```

```
##
## Portmanteau Test (asymptotic)
##
## data: Residuals of VAR object vecvar
## Chi-squared = 150.47, df = 180, p-value = 0.9468
```

O teste de autocorrelação serial não rejeitou a hipótese nula de ausência de autocorrelação nos resíduos do modelo VECM. O valor-p de 0.9468 é maior que o nível de significância de 0,05, indicando que não há evidências de autocorrelação serial nos resíduos. A ausência de autocorrelação serial nos resíduos indica que o modelo está bem especificado e que as estimativas dos coeficientes são válidas. É possível realizar previsões e inferências a partir do modelo com confiança.

Decomposição da variância dos erros

```
# Realize a decomposição da variância
decomposicao_var <- fevd(vecvar, n.ahead = 4)
```

```
# Exiba os resultados
print(decomposicao_var)
```

```
## $serie_rend
##      serie_rend serie_inpc serie_ocup  serie_sm
## [1,] 1.0000000 0.0000000 0.00000000 0.00000000
## [2,] 0.7164525 0.2054298 0.00196055 0.07615710
## [3,] 0.5750676 0.2439255 0.08992188 0.09108499
## [4,] 0.4504131 0.3100461 0.10139966 0.13814113
##
## $serie_inpc
##      serie_rend serie_inpc serie_ocup  serie_sm
## [1,] 0.7282227 0.2717773 0.00000000 0.00000000
## [2,] 0.3969708 0.4368643 0.01022069 0.1559442
## [3,] 0.2715523 0.3920673 0.18569810 0.1506823
## [4,] 0.1954752 0.4587522 0.17817458 0.1675980
##
## $serie_ocup
##      serie_rend serie_inpc serie_ocup  serie_sm
## [1,] 0.5913196 0.001417612 0.4072628 0.000000000
## [2,] 0.5270440 0.019150850 0.4526674 0.001137817
## [3,] 0.5084561 0.046114241 0.4425831 0.002846597
## [4,] 0.5995493 0.029603776 0.3689564 0.001890460
##
## $serie_sm
##      serie_rend serie_inpc serie_ocup  serie_sm
## [1,] 0.5975456 0.01175516 0.11798160 0.2727176
## [2,] 0.6985846 0.01147025 0.09162086 0.1983243
```

```
## [3,] 0.6042378 0.10618476 0.07543141 0.2141461
## [4,] 0.5209134 0.15322669 0.07534727 0.2505126
```

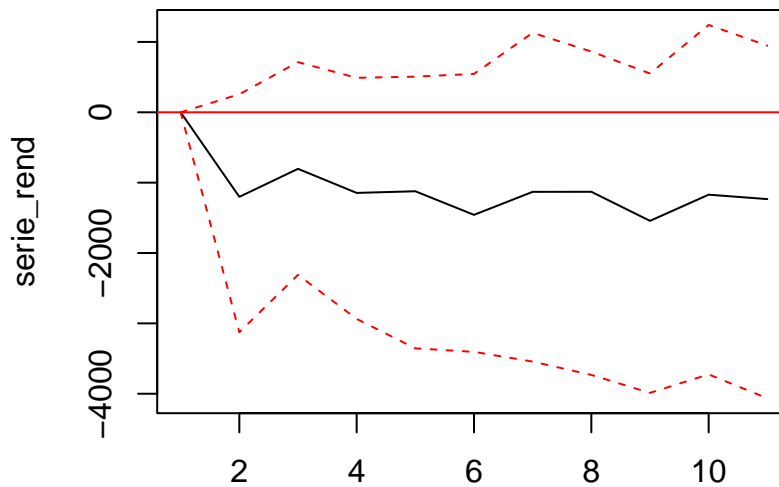
Os resultados da decomposição da variância dos erros sugerem que o rendimento médio tem total da sua variação explicada por ela mesma, ou seja, por choques internos. A variável do índice de preços ao consumidor e do número de pessoas ocupadas em postos de trabalho no Ceará indica que a parcela da variância destas variáveis explicada por si mesmas é relativamente pequena. Isso sugere que o índice de preços ao consumidor é mais volátil e que é mais influenciado por choques externos. As séries de ocupação e salário mínimo parecem ser muito influenciadas pelas variações na série de rendimento real.

Função de impulso resposta

```
# Calcule as respostas ao impulso
impulse_inpc <- irf(vecvar, impulse = "serie_inpc", response = "serie_rend", n.ahead = 10, ortho = FALSE, r
impulse_sm <- irf(vecvar, impulse = "serie_sm", response = "serie_rend", n.ahead = 10, ortho = FALSE, r
impulse_ocup <- irf(vecvar, impulse = "serie_ocup", response = "serie_rend", n.ahead = 10, ortho = FALSE, r

# Plote as respostas ao impulso
plot(impulse_inpc, main = "Resposta ao impulso da inflação no rendimento médio real")
```

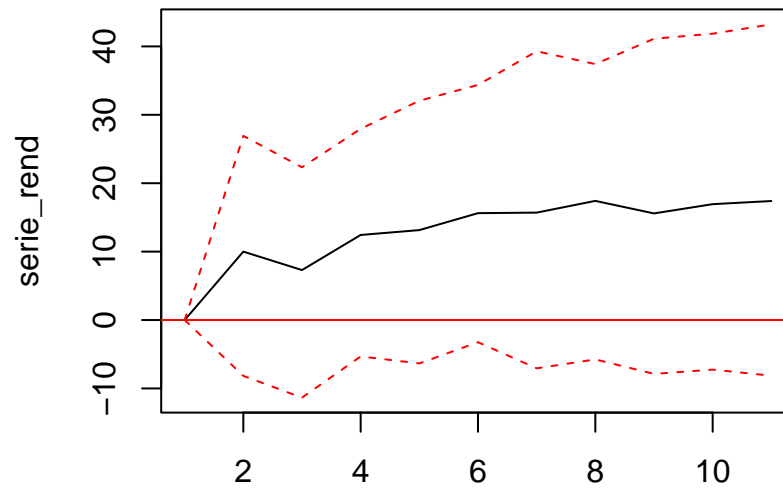
Resposta ao impulso da inflação no rendimento médio real



95 % Bootstrap CI, 1000 runs

```
plot(impulse_sm, main = "Resposta ao impulso do salário mínimo real no rendimento médio real")
```

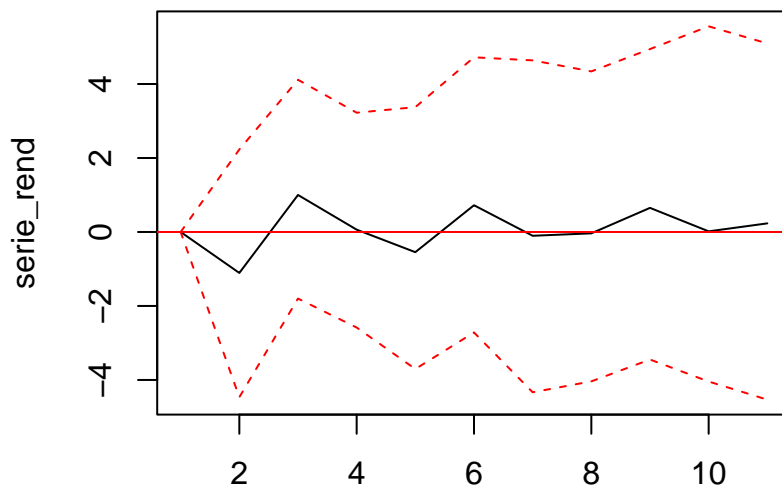
Resposta ao impulso do salário mínimo real no rendimento médio real



95 % Bootstrap CI, 1000 runs

```
plot(impulse_ocup, main = "Resposta ao impulso do número de ocupados no rendimento médio real")
```


Resposta ao impulso do número de ocupados no rendimento médio real



95 % Bootstrap CI, 1000 runs

Os resultados da função de impulso resposta sugerem que choques de inflação influenciam negativamente o rendimento real dos trabalhadores cearenses enquanto que choque no salário mínimo real influenciam positivamente, isto é, uma inflação maior contribui para decréscimo no rendimento médio real dos trabalhadores, enquanto aumento de salário mínimo real, isto é, acréscimos acima da inflação contribuem para melhora do rendimento real dos trabalhadores, e se usarmos rendimento médio real como proxy para distribuição de renda podemos inferir melhora na distribuição de renda. Já em relação ao número de pessoas ocupadas em postos de trabalho no ceará a resposta foi uma oscilação em torno de zero na variável de rendimento médio o que indica nenhuma influência do número de pessoas ocupadas no rendimento médio real do cearense.