

Rendimento real médio mensal de todos os trabalhos efetivos

Econometria II - Séries Temporais

Francisco Alexandre

2023-08-23

Email: [alecorreia94@alu.ufc.br]

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

Limpeza da memória do R:

```
rm(list = ls())
```

1 Introdução

A análise de séries temporais desempenha um papel crucial no campo da economia e na compreensão das tendências econômicas ao longo do tempo. Neste estudo, apresenta-se uma série temporal que representa o rendimento real médio mensal de todos os trabalhos efetivos no Brasil. Trata-se do rendimento bruto real médio efetivamente recebido no mês de referência em todos os trabalhos que as pessoas ocupadas com rendimento tinham na semana de referência.

Essa série reflete não apenas a situação econômica do país, mas também as condições de vida de seus habitantes. Os dados utilizados neste estudo foram obtidos por meio da biblioteca `ipeatadar`, disponibilizada pelo Instituto de Pesquisa Econômica Aplicada (IPEA). Desta forma, foram selecionadas informações da Pesquisa Nacional por Amostra de Domicílios Contínua (PNADC), Fonte confiável e abrangente de dados sobre o mercado de trabalho brasileiro.

A série temporal em análise abrange um período de 137 observações mensais, que se estende desde março de 2021 até julho de 2023. Essa janela temporal nos permite examinar tendências e padrões que podem ter ocorrido ao longo desse período, bem como identificar qualquer sazonalidade que possa afetar o rendimento real médio. A análise desses dados é de grande relevância, pois fornece *insights* valiosos sobre a dinâmica econômica do país. Ao longo deste estudo, exploro diversos aspectos, incluindo a distribuição dos rendimentos, estatísticas descritivas que resumem as características centrais da série, além de gráficos que ajudam a visualizar padrões e sazonalidades.

Além disso, realiza-se uma análise mais profunda por meio de correlogramas para identificar possíveis modelos ARMA (AutoRegressive Moving Average) de forma a melhorar a compreensão sobre a estrutura temporal subjacente da série. Também aplica-se a técnica de dessazonalização X11 do Census Bureau dos Estados Unidos, para isolar a tendência e os componentes de erro da série, a fim de examiná-los com maior precisão.

Este estudo visa contribuir para uma melhor compreensão da dinâmica do mercado de trabalho no Brasil, permitindo uma análise mais informada das flutuações no rendimento real médio mensal de todos os trabalhos efetivos. Na próxima seção, apresento os resultados desta análise, incluindo histograma da distribuição dos rendimentos, estatísticas descritivas que resumem as principais características da série temporal, correlogramas, estimação e testes sobre os modelos e por fim a previsão feita através do modelo selecionado pelos critérios estatísticos. Boa leitura!

Bibliotecas utilizadas no estudo:

```
library(knitr)
library(ipeadata)
library(dplyr)
library(forecast)
library(dygraphs)
library(urca)
library(rugarch)
```

```
# Obtendo todas as séries disponíveis
series_ipeadata <- available_series()
```

```
#Seleciona Rendimento real médio de todos os trabalhos efetivos mensais
#codigo: PNADC12_RRTE12
dados_RRTE <- ipeadata("PNADC12_RRTE12")
```

```
# Selecione apenas as variáveis "data" e "valor" de dados_RRTE
dados_RRTE <- select(dados_RRTE, date, value)
```

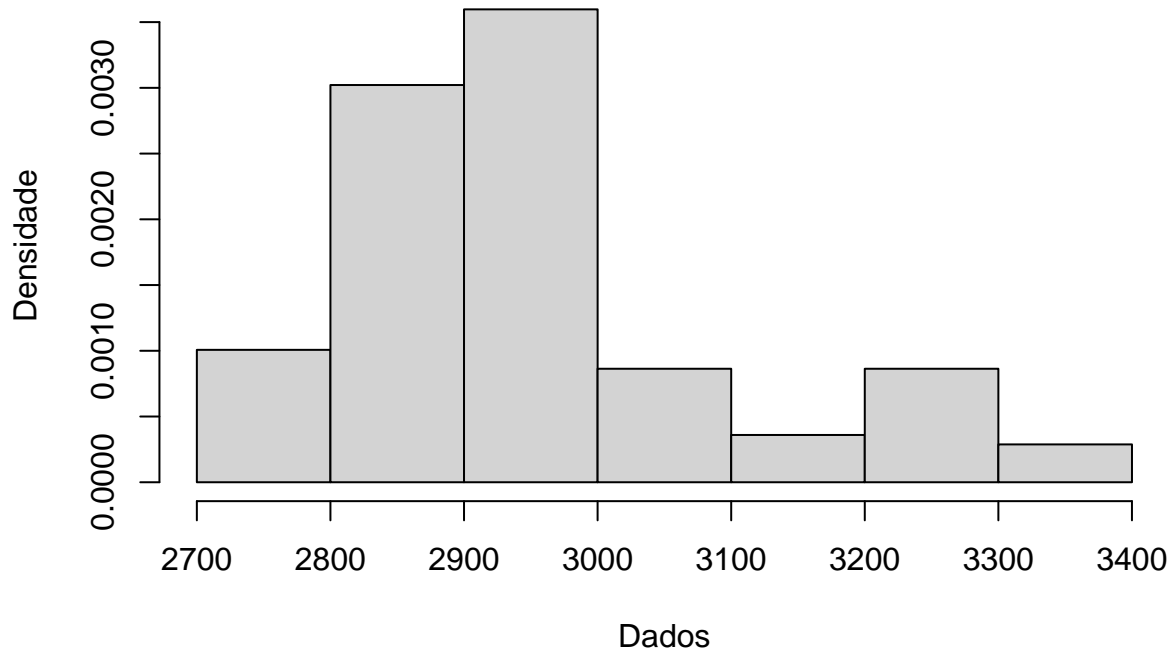
2 Resultados

2.1 Histograma da distribuição dos dados de rendimento

Logo abaixo temos o histograma para melhor visualização da distribuição dos dados.

```
#histograma
hist(dados_RRTE$value,prob=T,
      main="Rendimento real médio de todos os trabalhos efetivos mensais",
      ylab = "Densidade", xlab = "Dados")
```

Rendimento real médio de todos os trabalhos efetivos mensais



Identifica-se pelo histograma da série de dados que a maioria dos resultados se concentram no intervalo de 2.800 a 3.000 reais em média.

2.2 Estatísticas Descritivas

```
# Calcular estatísticas descritivas
estatisticas_RRTE <- dados_RRTE %>%
  summarise(
    Média = mean(value),
    Mediana = median(value),
    Mínimo = min(value),
    Máximo = max(value),
    Desvio_Padrão = sd(value),
    Variância = var(value)
  )

# Use a função kable para formatar a tabela
estatisticas_formatadas <- kable(estatisticas_RRTE,
  format = "markdown",
  col.names = c("Média", "Mediana", "Mínimo",
    "Máximo", "Desvio Padrão", "Variância"))

estatisticas_formatadas
```

Média	Mediana	Mínimo	Máximo	Desvio Padrão	Variância
2953.763	2924	2721	3356	145.2787	21105.91

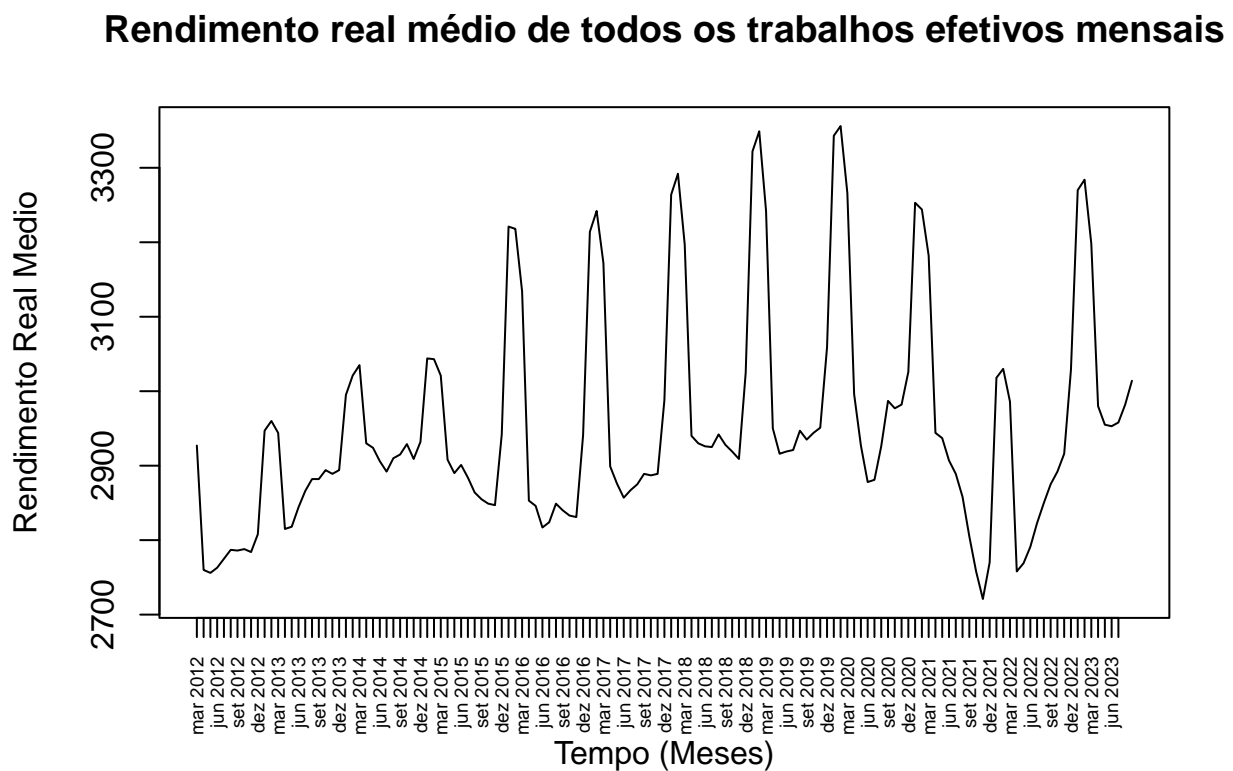
Como era de se esperar a média (2.944,35 reais) encontra-se na maioria dos dados no histograma. Nota-se também com destaque o desvio padrão, isto é, a distância dos valores em torno da média é de (145,8 reais) indicando pouca variação da renda média real no período analisado.

2.3 Gráficos dá série

```
# Cria um vetor de datas para o eixo x
datas <- seq(as.Date("2012-03-01"), as.Date("2023-07-01"), by = "1 month")

# Cria o gráfico de séries temporais
plot.ts(dados_RRTE$value,
        main = "Rendimento real médio de todos os trabalhos efetivos mensais",
        xaxt = "n", ylab = "Rendimento Real Medio", xlab = "Tempo (Meses)")

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



Logo abaixo temos o mesmo gráfico apresentado com mais detalhes para melhor interação e visualização (Obs: este gráfico abaixo deve ser visto em formato .html)

```

#Série
serie = ts(dados_RRTE$value, start = c(2012,3), frequency = 12)

#limpando possiveis outliers
tsclean.serie <- tsclean(serie)

#para html
# Gráfico dos dados com detalhes
dygraph(
  data = serie,
  main = "Rendimento Real Medio do Brasil de 2012 a 2023",
  xlab = "Meses",
  ylab = "Rendimento Real Medio"
)

```

É fácil notar que há uma sazonalidade nesta série com amplitudes de renda entre os meses de janeiro a março, geralmente começando a aumentar em dezembro. Dessa forma, esta série possui sazonalidade do tipo multiplicativa, podendo visualmente ser identificada, pois a amplitude dela aumenta com o passar do tempo. Observa-se abaixo (2.4) esses aspectos nitidamente pela função de autocorrelação. Ainda pode-se destacar uma forte queda da renda entre fevereiro e dezembro de 2021, que pode ter ocorrido pelos efeitos adversos causados pela pandemia de covid-19. Também nota-se que não há outliers nesta série.

2.4 Teste de raiz unitária

Pelo gráfico a série não parece produzir tendência, mas parece possuir drift por conta das características da própria série. Abaixo é feito um teste para identificar se a série possui raiz unitária.

```

#Teste de raiz unitária com 5 lags, drift e criterios BIC
ur <- ur.df(y = serie, lags = 5, type = "drift", selectlags = "BIC")
ur@testreg

##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -131.58  -44.84  -20.00   30.87  262.05
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1063.85359   200.68404    5.301 4.94e-07 ***
## z.lag.1      -0.35927    0.06785   -5.295 5.06e-07 ***
## z.diff.lag1    0.57782    0.08814    6.556 1.26e-09 ***
## z.diff.lag2    0.12218    0.08400    1.454 0.148282
## z.diff.lag3   -0.31127    0.08222   -3.786 0.000235 ***
## z.diff.lag4    0.27888    0.08400    3.320 0.001174 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.65 on 127 degrees of freedom
## Multiple R-squared:  0.4637, Adjusted R-squared:  0.4426
## F-statistic: 21.96 on 5 and 127 DF, p-value: 8.331e-16

```

```
ur@cval
```

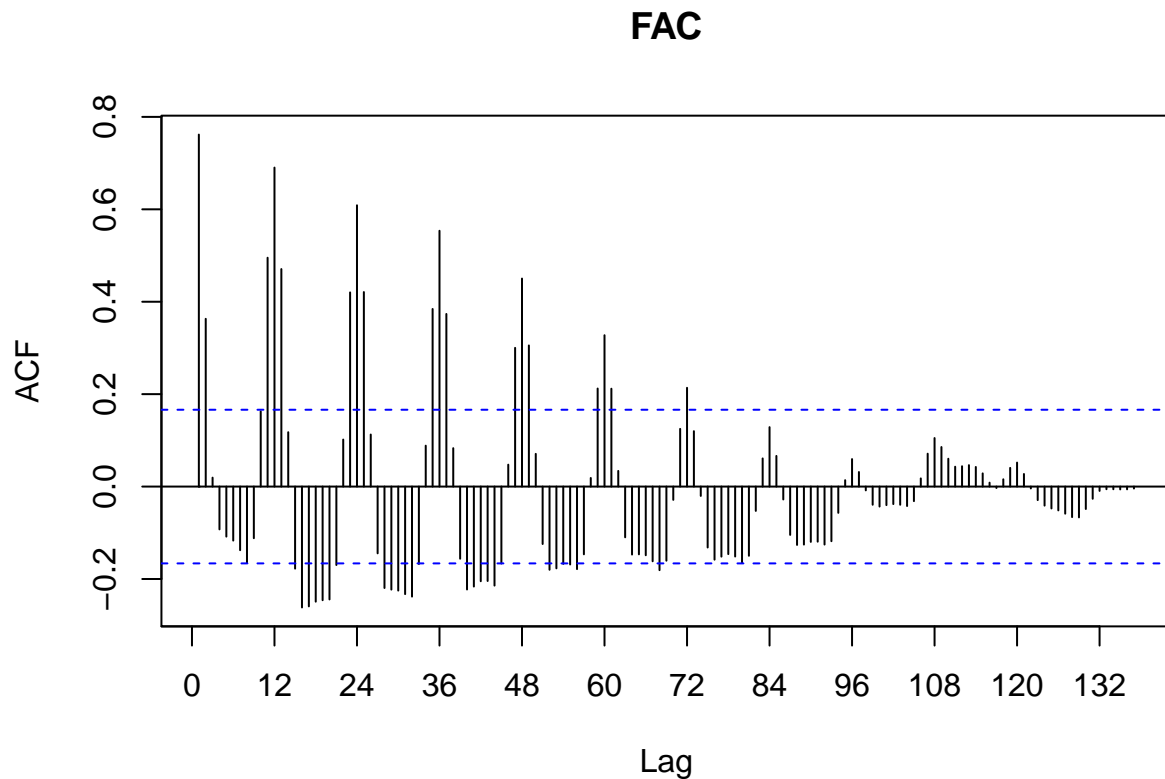
```
##          1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

O modelo foi selecionado com 4 lags, mesmo sendo fornecido 5 lags. Analisando os resultados encontra-se o coeficiente $z.lag.1$, parâmetro de interesse para o teste de raiz unitária e para avaliar a sua significância precisa-se da tabela de valores críticos que fica na variável `ur@cval` do teste.

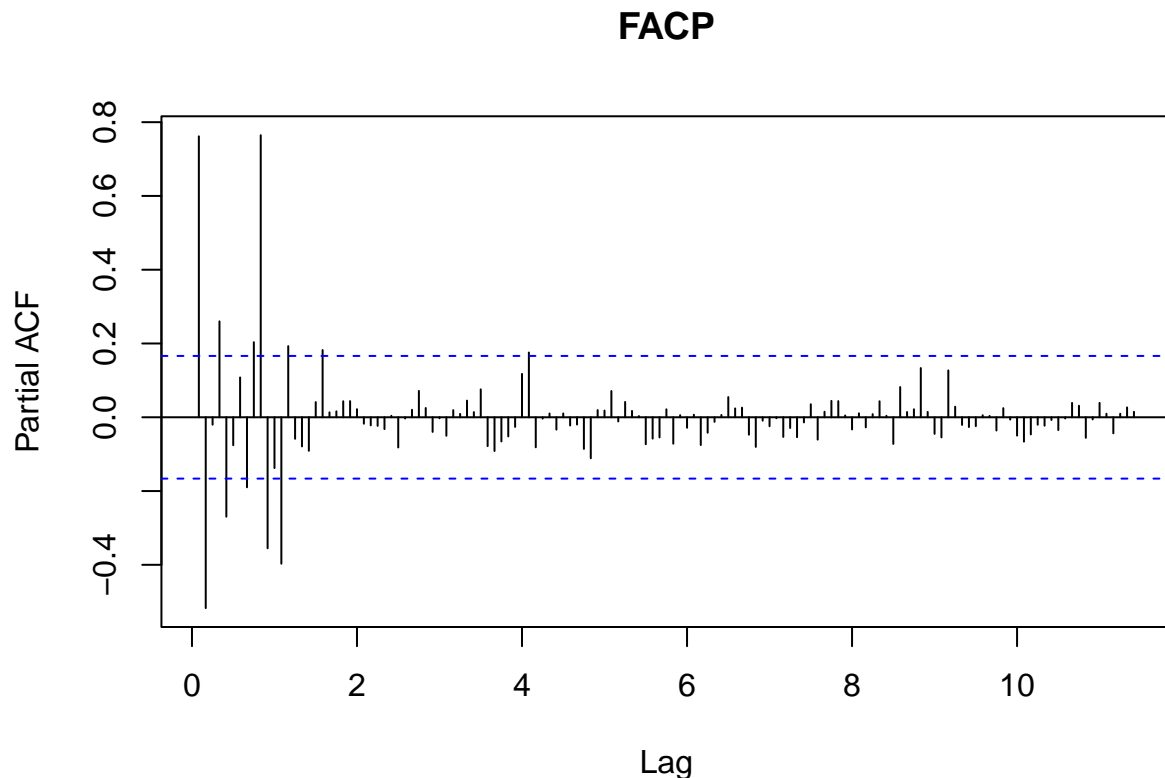
Diante disso, τ_2 é a estatística referente ao coeficiente $z.lag.1$ e estes são os dados que interessam, a informação de significância da tabela coefficients refere-se ao teste $-t$. Na mesma tabela temos que o valor da estatística para $z.lag.1$ é -5.295 e avaliando os níveis críticos de τ_2 conclui-se que é possível rejeitar a hipótese nula para $z.lag.1$ e, portanto, a série não possui raiz unitária e é estacionária. Vamos agora analisar o correlograma da serie abaixo.

2.5 Correlogramas e modelos ARMA(p,q)

```
#FAC E FACP da serie refet
Acf(serie,lag=137,main="FAC" )
```



```
pacf(serie, lag=137, main="FACP")
```



Ao observar o correlograma, percebe-se uma sazonalidade do tipo multiplicativa que incide em épocas específicas e em seus arredores. Isto é, a significância estatística não se limita apenas aos períodos, mas também abrange os arredores da série de dados. Considerando essa análise, pondera-se sobre alguns modelos do tipo ARMA que podem ser apropriados. O AR(2) surge como uma escolha adequada, uma vez que na Função de Autocorrelação (FAC), há um declínio exponencial visível, enquanto na Função de Autocorrelação Parcial (FACP), ocorre um truncamento no segundo lag. Também é possível sugerir a possibilidade de um MA(2), uma vez que existe um truncamento no segundo lag da FAC. Dessa forma, pretende-se testar um modelo do tipo ARMA(2,2), AR(2) e MA(2) para uma análise mais profunda presente nos dados é aconselhável verificar todas as combinações possíveis de um ARMA(2,2). Mas antes devemos dessazonalizar a série.

2.6 Decomposição X11

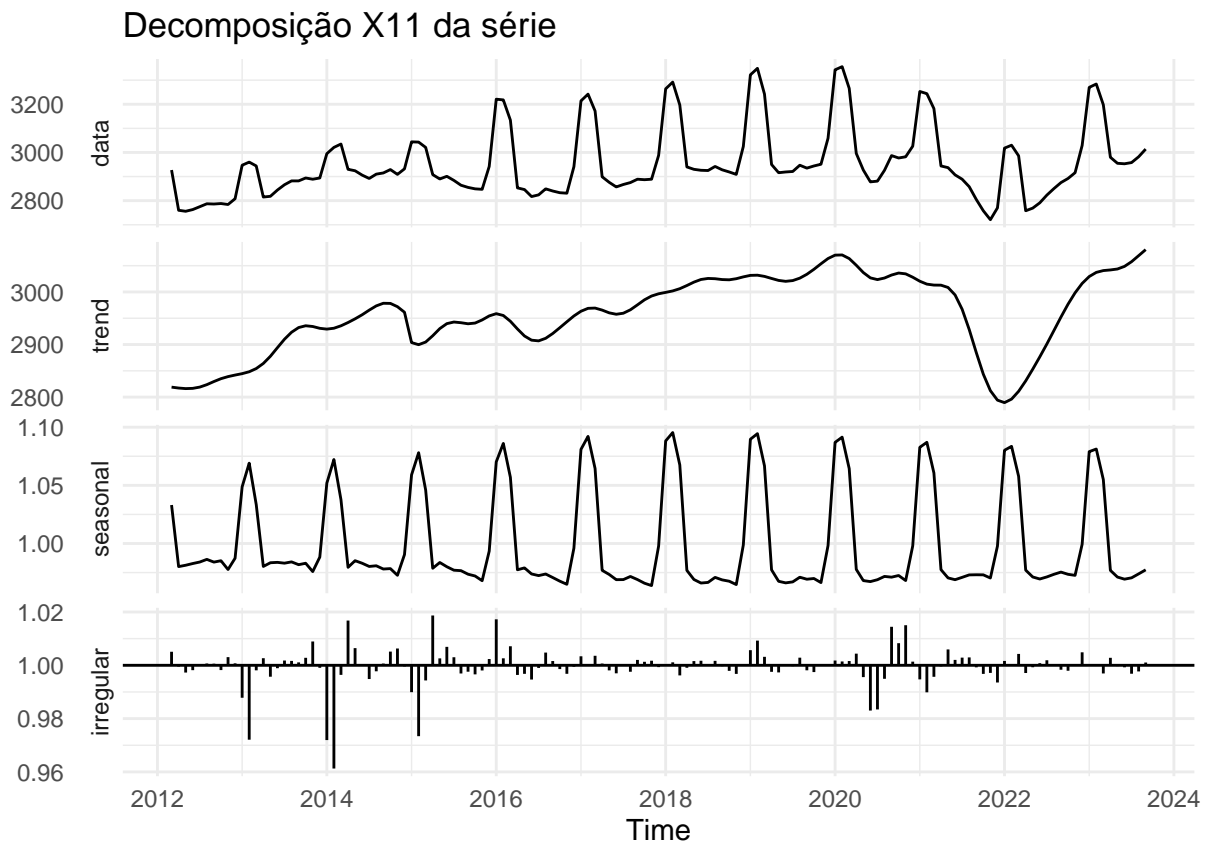
Ao perceber a sazonalidade um método comum para dessazonalizar séries temporais é utilizar a metodologia do Census Bureau dos Estados Unidos da América, conhecida como X11 (US BUREAU OF THE CENSUS, 2013). O primeiro passo é carregar o pacote 'seasonal' (SAX e EDDERBUETTEL, 2018) no ambiente de trabalho. Para visualizar os resultados, utilizei a função 'autoplot' do pacote 'fpp2' (HYNDMAN, 2018), que nos permite criar um gráfico informativo.

```
serie = ts(dados_RRTE$value, start = c(2012,3), frequency = 12)

# decomposicao pelo X11 do Census Bureau
library(seasonal)
fit <- seas(serie, x11 = "")
```

```
# uso a serie 'serie', aplico o 'seas' x11
# e gero 'fit'

library(fpp2)
autoplot(fit) + ggtitle("Decomposição X11 da série")+theme_minimal()
```

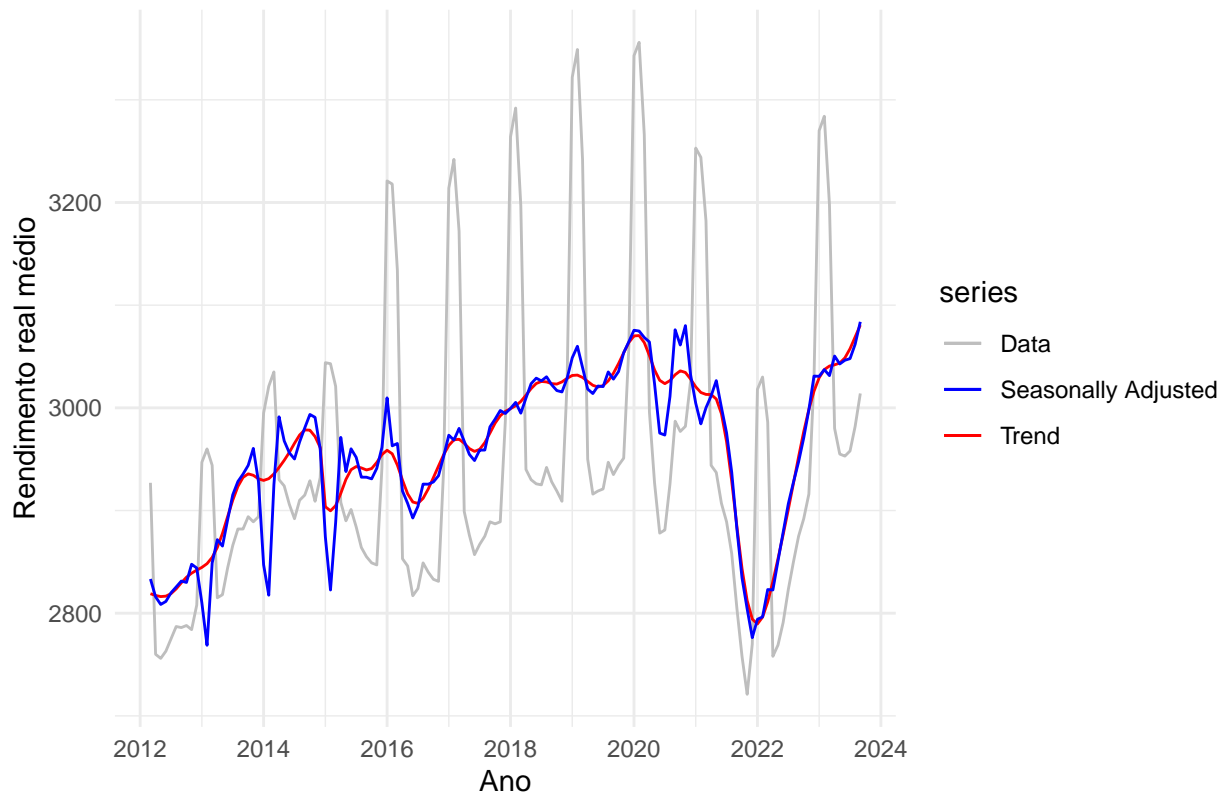


Abaixo temos um gráfico que mostra a série dessazonalizada e a sua tendência obtida através da técnica *Rolling Mean*.

2.7 Gráfico da série dessazonalizada

```
autoplot(serie, series = "Data") + autolayer(trendcycle(fit),
                                             series = "Trend") + theme_minimal()+
  autolayer(seasadj(fit),
            series = "Seasonally Adjusted") + xlab("Ano") + ylab("Rendimento real médio") +
  ggtitle("Rendimento real médio de todos os trabalhos efetivos mensais no Brasil") +
  scale_colour_manual(values = c("gray", "blue", "red"), breaks = c("Data",
                                                                    "Seasonally Adjusted",
                                                                    "Trend"))
```


Rendimento real médio de todos os trabalhos efetivos mensais no Brasil



Observa-se na série temporal a presença de componentes sazonais. Para isolar esses elementos, foi realizada a decomposição da série, resultando na extração da componente dessazonalizada. Em outras palavras, a série dessazonalizada consiste apenas na parte que reflete a tendência e os componentes de erro, eliminando completamente a influência sazonal. Portanto, agora dispomos da série temporal original, porém, sem a presença da sazonalidade, o que nos permite analisá-la de forma mais precisa e identificar padrões e correlações sem a interferência das variações sazonais.

Dessa forma, ao analisar o gráfico, observa-se um crescimento no rendimento médio ao longo dos anos, com a influência de algum choque no período entre 2021 e 2022, caracterizado por um forte aumento no final de 2022, seguido pela retomada da tendência de crescimento anterior a esse choque.

Baseado no correlograma da série estima-se abaixo os modelos ARMA adequados

2.8 Tabela dos modelos ARMA(p,q)

```
# Lista de ordens ARMA a serem testadas
ordens <- expand.grid(p = 0:2, q = 0:2)

# Função para ajustar modelo ARMA e calcular AIC e BIC
ajustar_modelo_arma <- function(p, q) {
  modelo <- Arima(seasadj(fit), order = c(p, 0, q))
  return(c(especificacao = paste0("ARMA(", p, ",", q, ")"),
    ln_verossimilhanca = round(logLik(modelo), digits = 3),
    qtd_parametros = length(coef(modelo)),
    tamanho_amostra = length(serie),
```

```

      AIC = round(AIC(modelo), digits = 2),
      BIC = round(BIC(modelo), digits = 2)))
}
# Aplicar a função para cada ordem ARMA
resultados <- apply(ordens, 1, function(row) ajustar_modelo_arma(row[1], row[2]))

# Converter a matriz de resultados em um dataframe
resultados_df <- as.data.frame(t(resultados))

# Exibir os resultados
# Usando a função kable para criar a tabela
tabela_markdown <- kable(resultados_df, format = "markdown", align = "c")
tabela_markdown

```

especificacao	ln_verossimilhanca	qtd_parametros	tamanho_amostra	AIC	BIC
ARMA(0,0)	-804.847	1	139	1613.69	1619.56
ARMA(1,0)	-659.705	2	139	1325.41	1334.21
ARMA(2,0)	-651.169	3	139	1310.34	1322.08
ARMA(0,1)	-733.721	2	139	1473.44	1482.25
ARMA(1,1)	-650.112	3	139	1308.22	1319.96
ARMA(2,1)	-649.818	4	139	1309.64	1324.31
ARMA(0,2)	-686.163	3	139	1380.33	1392.06
ARMA(1,2)	-649.427	4	139	1308.85	1323.53
ARMA(2,2)	-648.038	5	139	1308.08	1325.68

Note que pela tabela o modelo ARMA(2,2) tem a maior log verossimilhança, o que sugere que se ajusta melhor aos dados em comparação com outros modelos. Porém, pelos critérios AIC e BIC o melhor modelo a ser utilizado é um ARMA(1,2). Para uma melhor tomada de decisão analisa-se a significância dos parâmetros e o comportamento dos modelos.

```

# Estima o modelo ARMA(2,2)
modelo_arma_22 <- Arima(seasadj(fit), order = c(2, 0, 2))
summary(modelo_arma_22)

```

```

## Series: seasadj(fit)
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1      ma2      mean
##          0.3055  0.5120  1.0599  0.4074 2957.1762
## s.e.      0.1977  0.1865  0.1880  0.1083  27.5486
##
## sigma^2 = 667.9: log likelihood = -648.04
## AIC=1308.08 AICc=1308.71 BIC=1325.68
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.8315763 25.37388 18.76241 0.02073579 0.6382285 0.2669796
##              ACF1
## Training set -0.02489332

```

```

# Coletar informações do modelo ARIMA(2,0,2)
coef_arma_22 <- coef(modelo_arma_22)
desv_p_arma_22 <- sqrt(diag(vcov(modelo_arma_22)))
t_val_arma_22 <- coef_arma_22 / desv_p_arma_22

# Criar uma tabela com os valores dos coeficientes, desvios padrões e valores
# t (excluindo o intercepto)
tabela_coef_desv_t <- data.frame(Valor_Coeficiente =coef_arma_22[-5],
                                Desv_P = desv_p_arma_22[-5],
                                Valor_t = t_val_arma_22[-5])

# Usando a função kable para criar a tabela
tabela_markdown <- kable(tabela_coef_desv_t, format = "markdown", align = "c")
tabela_markdown

```

	Valor_Coeficiente	Desv_P	Valor_t
ar1	0.3054535	0.1976622	1.545331
ar2	0.5120384	0.1865204	2.745214
ma1	1.0598722	0.1880427	5.636337
ma2	0.4074186	0.1082822	3.762564

Pela tabela acima podemos ver que no AR(1) possui o único valor t que não ultrapassa 1.96, isto é, para um nível de significância de 5% mostra não ser um parâmetro com significancia estatística. Dessa Forma, Fica mais claro que nosso modelo preferível seja o ARMA(2,2). Cabe destacar que essa estimação foi feita retirando a sazonalidade dos dados, ao deixar o efeito de sazonalidade o AR(2) é que se torna não significativo. Por fim, como último teste observa-se o correlograma dos resíduos a fim de identificar indícios de normalidade.

```

# Lista de ordens ARMA(2,2)
ordens <- expand.grid(p = 0:2, q = 0:2)

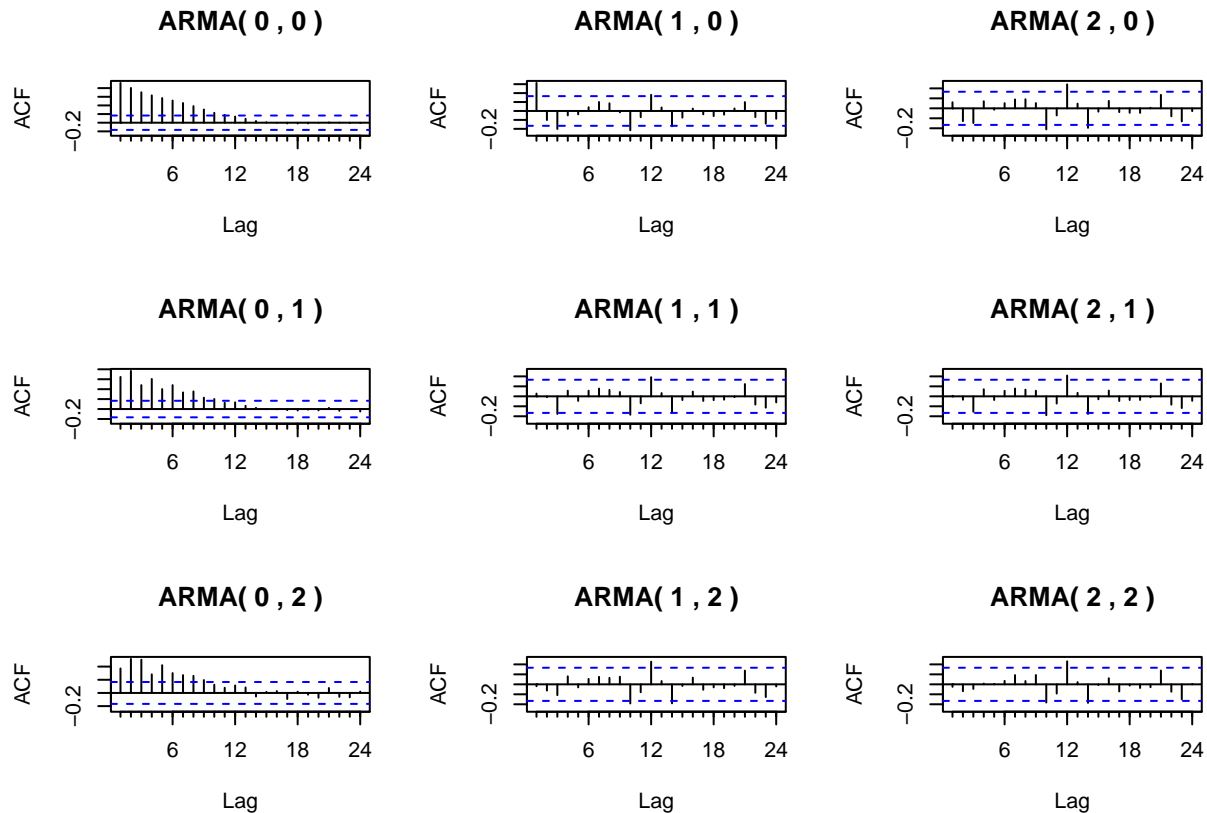
# Configurar o layout para 3x3 (ou o número desejado)
par(mfrow = c(3, 3))

# Loop para ajustar modelos e criar correlogramas
for (i in 1:nrow(ordens)) {
  p <- ordens$p[i]
  q <- ordens$q[i]

  # Ajustar modelo ARMA
  modelo <- Arima(seasadj(fit), order = c(p, 0, q))
  residuos <- residuals(modelo)

  # Criar correlograma dos resíduos
  modelo_nome <- paste("ARMA(", p, ",", q, ")")
  Acf(residuos, main = modelo_nome)
}

```



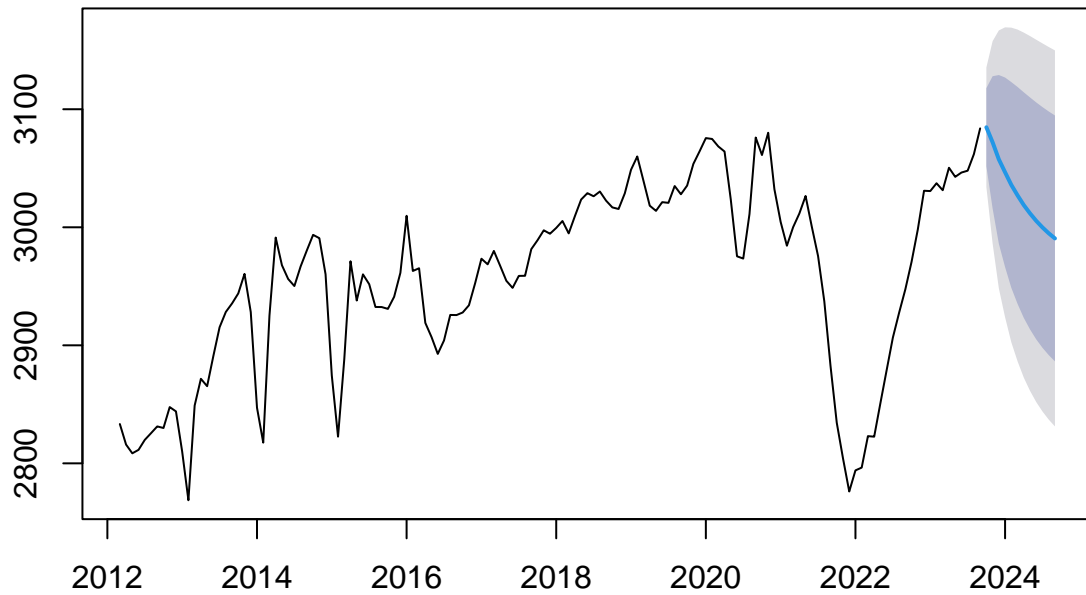
```
# Restaurar a configuração padrão
par(mfrow = c(1, 1))
```

O comportamento dos resíduos de todos os modelos mostra que o modelo ARMA(2,2) tem seus resíduos dentro das faixas de significância exceto por alguns períodos em que há algumas linhas que a ultrapassam, isso possivelmente se deve aos períodos de sazonalidade que não puderam ser completamente expurgados na dessazonalização. O mesmo acontece com o modelo ARMA(1,2). Diante disso, pondera-se que a melhor escolha seja utilizar o ARMA(2,2) para previsão, pois foi o único que se mostrou mais eficiente em todos os testes realizados.

```
# Gerar previsões
previsoes <- forecast(modelo_arma_22, h = 12) # Ajuste h conforme necessário

# Plotar as previsões
plot(previsoes, main = "Previsões ARMA(2,2)")
```

Previsões ARMA(2,2)

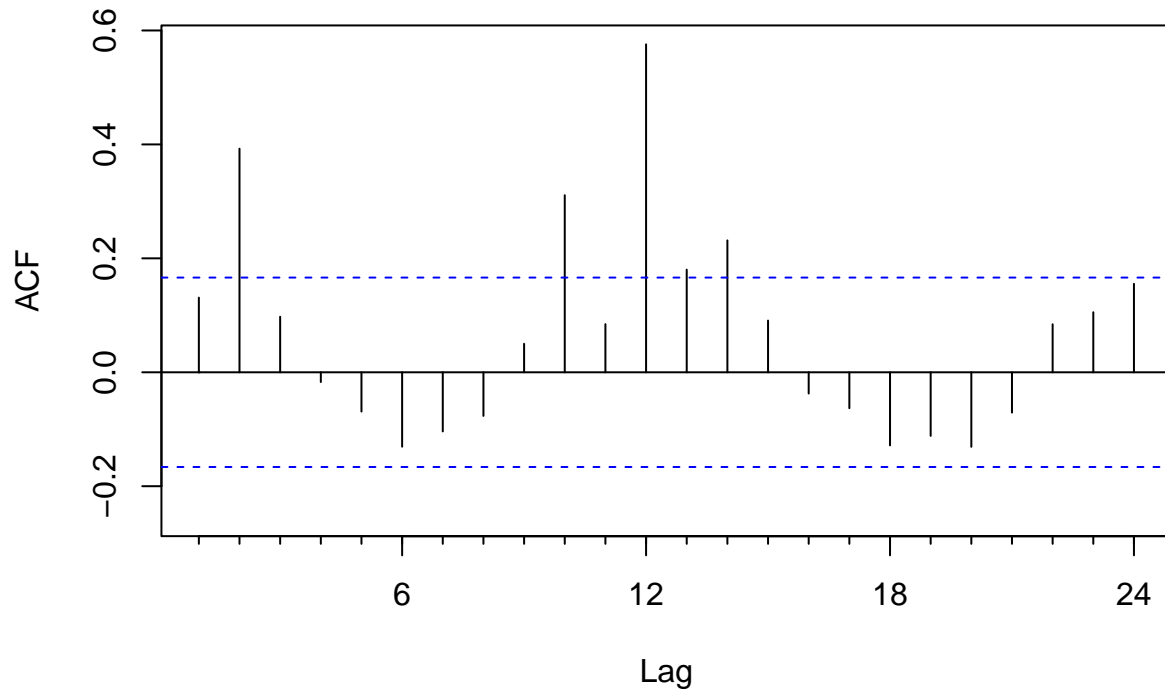


Como ultima etapa de análise deste estudo mostramos os testes em relação a heterocedasticidade condicional do modelo escolhido.

```
# Calcular os resíduos ao quadrado
residuos_quadrados <- residuos^2

# Criar correlograma dos resíduos ao quadrado
Acf(residuos_quadrados, main = "Correlograma dos Resíduos ao Quadrado")
```

Correlograma dos Resíduos ao Quadrado



```
# Teste Ljung-Box para os resíduos ao quadrado
Box.test(residuos_quadrados, lag = 20, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data:  residuos_quadrados
## X-squared = 122.02, df = 20, p-value < 2.2e-16
```

Pelo correlograma e o teste de Box-Ljung pode-se inferir presença de efeitos ARCH nos resíduos já que há picos significativos. Então, provavelmente o modelo apresenta alguma dependência temporal. Além disso, rejeita-se a hipótese nula de que todos os coeficientes de autocorrelação dos resíduos são iguais a 0. Dessa forma, agora pode-se ajustar os resíduos da estimação em um modelo GARCH(1,1).

```
# Ajustar um modelo GARCH aos resíduos
modelo_garch <- ugarchspec(variance.model = list(model = "sGARCH",
                                                  garchOrder = c(1, 1)),
                           mean.model = list(armaOrder = c(0, 0)))
ajuste_garch <- ugarchfit(modelo_garch, data = residuos_quadrados)

print(ajuste_garch)
```

```
##
## *-----*
## *          GARCH Model Fit          *
```

```

## *-----*
##
## Conditional Variance Dynamics
## -----
## GARCH Model   : sGARCH(1,1)
## Mean Model    : ARFIMA(0,0,0)
## Distribution   : norm
##
## Optimal Parameters
## -----
##           Estimate Std. Error  t value Pr(>|t|)
## mu        230.39340  3.3113e+01  6.95773  0.00000
## omega     1675.57559  2.3900e+03  0.70108  0.48325
## alpha1     0.25304  3.4846e-02  7.26183  0.00000
## beta1      0.74596  2.5342e-02 29.43529  0.00000
##
## Robust Standard Errors:
##           Estimate Std. Error  t value Pr(>|t|)
## mu        230.39340  5.2873e+01  4.35751  0.000013
## omega     1675.57559  3.2740e+03  0.51178  0.608805
## alpha1     0.25304  6.7315e-02  3.75911  0.000171
## beta1      0.74596  4.9826e-02 14.97116  0.000000
##
## LogLikelihood : -1146.303
##
## Information Criteria
## -----
##
## Akaike          16.551
## Bayes           16.636
## Shibata         16.550
## Hannan-Quinn    16.585
##
## Weighted Ljung-Box Test on Standardized Residuals
## -----
##                               statistic p-value
## Lag[1]                      4.482 0.034246
## Lag[2*(p+q)+(p+q)-1][2]      8.889 0.003761
## Lag[4*(p+q)+(p+q)-1][5]     12.306 0.002315
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## -----
##                               statistic p-value
## Lag[1]                      0.05109 0.8212
## Lag[2*(p+q)+(p+q)-1][5]      3.10319 0.3885
## Lag[4*(p+q)+(p+q)-1][9]      4.58842 0.4922
## d.o.f=2
##
## Weighted ARCH LM Tests
## -----
##           Statistic Shape Scale P-Value
## ARCH Lag[3]    0.1531 0.500 2.000 0.6956

```

```

## ARCH Lag[5]      0.6895 1.440 1.667 0.8266
## ARCH Lag[7]      1.4630 2.315 1.543 0.8288
##
## Nyblom stability test
## -----
## Joint Statistic: 2.2056
## Individual Statistics:
## mu      0.1527
## omega   0.4037
## alpha1  1.3812
## beta1   1.1483
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:      1.07 1.24 1.6
## Individual Statistic: 0.35 0.47 0.75
##
## Sign Bias Test
## -----
##              t-value  prob sig
## Sign Bias      1.5739 0.1179
## Negative Sign Bias 0.4924 0.6232
## Positive Sign Bias 0.7126 0.4773
## Joint Effect    4.2240 0.2383
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## -----
##   group statistic p-value(g-1)
## 1    20      90.78 2.408e-11
## 2    30     102.08 4.535e-10
## 3    40     119.71 3.942e-10
## 4    50     141.22 7.154e-11
##
##
## Elapsed time : 0.064183

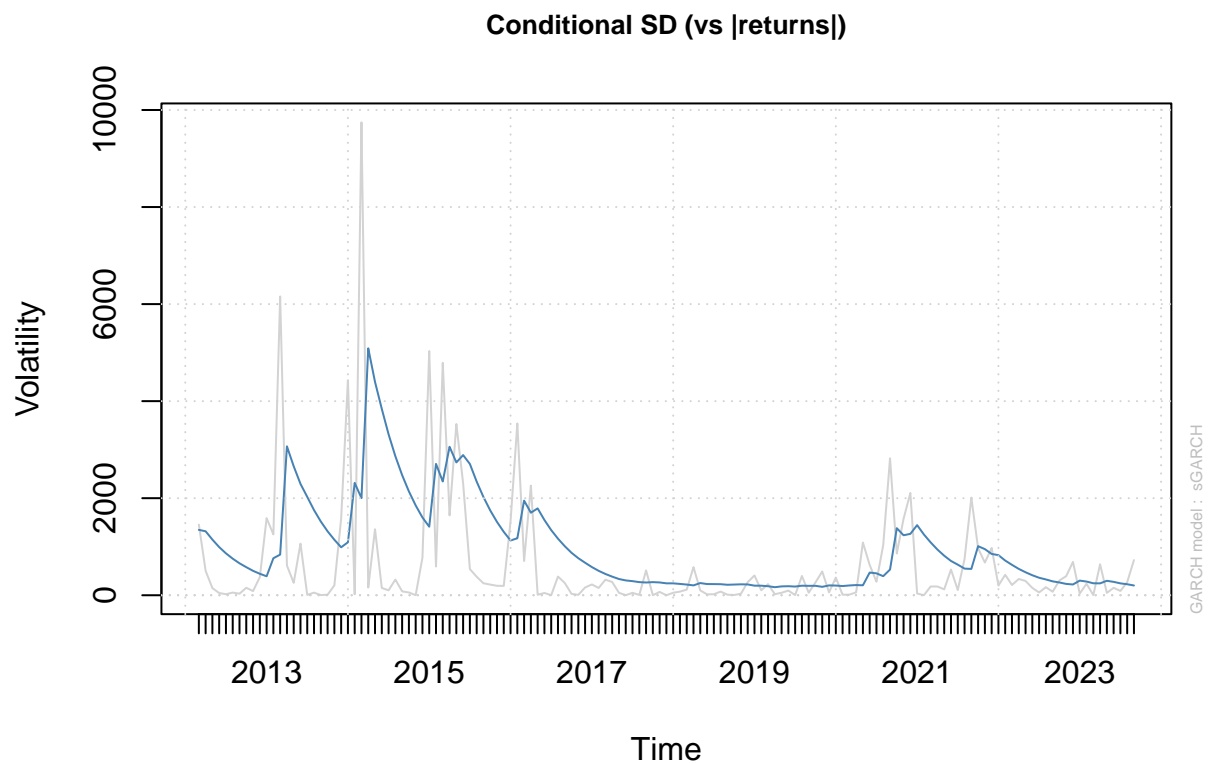
```

O modelo GARCH(1,1) parece se ajustar bem aos dados, com parâmetros significativos e resíduos que não exibem autocorrelação serial ou efeitos ARCH significativos.

```

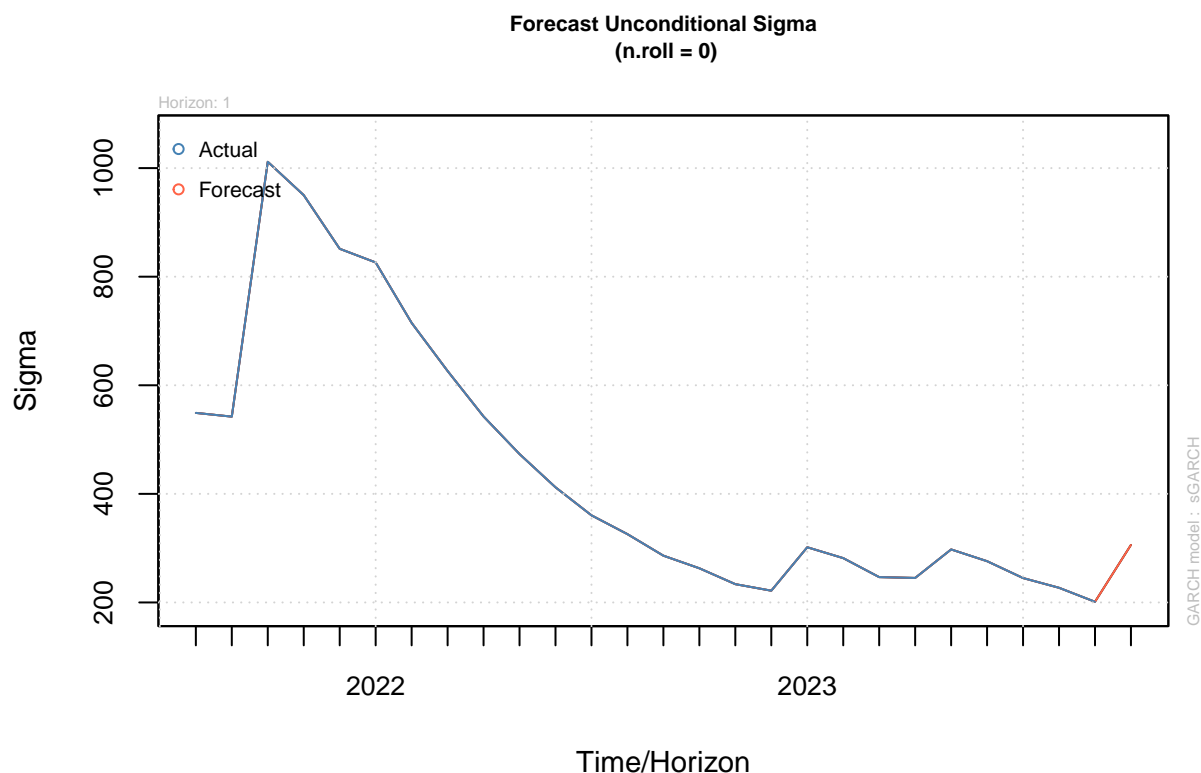
# Gráfico da variância ou desvio padrão condicional
plot(ajuste_garch, which = 3)

```

Observa-se no gráfico acima a volatilidade condicional (cinza) e os retornos absolutos (azul). Há momentos de grande volatilidade o que mostra bastante dependência temporal nos resíduos. Por fim, realiza-se a previsão para o modelo GARCH(1,1).

```
# Realiza o forecast  
forecast_garch <- ugarchforecast(ajuste_garch, n.ahead = 1)  
  
plot(forecast_garch, which = 3) # Forecast da variância
```



Para a previsão da série observa-se uma previsão de aumento da variância em um período a frente.

3 Conclusões

A série temporal do rendimento real médio mensal estudada oferece *insights* valiosos sobre a economia do país e as condições de vida da população. Ela reflete a dinâmica do mercado de trabalho brasileiro e suas flutuações ao longo do tempo.

A análise revelou a presença de sazonalidade na série temporal. Esta sazonalidade é do tipo multiplicativa e está relacionada a variações nos rendimentos ao longo do ano, com destaque para os meses de dezembro a fevereiro, nos quais as amplitudes de renda são mais evidentes.

A aplicação da metodologia X11 do Census Bureau dos Estados Unidos permitiu a dessazonalização da série, isolando a tendência e os componentes de erro. Isso nos possibilitou analisar a série de forma mais precisa, identificando padrões e correlações sem a influência das variações sazonais.

O modelo ARMA(2,2) foi identificado como o mais apropriado para modelar a série temporal, proporcionando uma base sólida para previsões futuras. A escolha foi fundamentada em critérios estatísticos robustos.

Além disso, a presença de heterocedasticidade condicional nos resíduos foi reconhecida e tratada com a aplicação de um modelo GARCH(1,1). O que pode melhorar a precisão das previsões, levando em consideração a variabilidade nas amplitudes dos resíduos ao longo do tempo.

E como pôde ser notado há uma previsão de queda do rendimento real médio nos próximos meses, isto é, nos últimos meses de 2023 e nos primeiros meses de 2024, colocando fim a um período de crescimento da renda média desde 2022.

4 Referências

- FIGUEIREDO, Adriano Marcos Rodrigues. Séries Temporais: decomposição clássica e a abordagem X11. Campo Grande-MS, Brasil: RStudio/Rpubs, 2019. Disponível em http://rpubs.com/amrofi/decompose_x11_varejoms.
- HYNDMAN, Rob. (2018). fpp2: Data for “Forecasting: Principles and Practice” (2nd Edition). R package version 2.3. Disponível em: <https://CRAN.R-project.org/package=fpp2>.
- Luiz Eduardo S. Gomes, Jessyka A. P. Goltara (2022). ipeadatar: API Wrapper for ‘Ipeadata’. R package version 0.1.6. URL: <https://CRAN.R-project.org/package=ipeadatar>.
- SAX C.; EDDERBUETTEL, D. (2018). “Seasonal Adjustment by X-13ARIMA-SEATS in R.” Journal of Statistical Software, 87(11), 1-17. doi: 10.18637/jss.v087.i11 (URL: <https://doi.org/10.18637/jss.v087.i11>).
- US BUREAU OF THE CENSUS (2013). X-13ARIMA-SEATS Reference Manual Accessible HTML Output Version. Staff Statistical Research Division, US Bureau of the Census, disponível em: <http://www.census.gov/ts/x13as/docX13ASHTML.pdf>.