# Análise de série temporal usando modelos paramétricos e rede neural

Walter Humberto Subiza Pina[[1]](#footnote-2)

Pedro Guilherme Costa Ferreira[[2]](#footnote-3)

José Francisco Pessanha[[3]](#footnote-4)

**Resumo**

Apresenta-se a análise de uma série temporal de economia univariada, por meio dos modelos SARIMA, HOLT-WINTERS e com modelo de Rede Neural de aprendizado de máquina. A série com valores mensais e comprimento de 8 anos, pertence ao IBGE e contém o volume de vendas ao comércio varejista ampliado (VCVA). Foi feita uma previsão de volume 12 meses à frente com cada metodologia e os resultados comparados com o volume real registrado.

**Palavras-chave:** serie temporal, sarima, Holt-Winters, rede neural, IBGE.

**Abstract**

An analysis of a uni-variate time series was made, using SARIMA and HOLT-WINTERS models as well as Neural Network. The time series from the IBGE, contain monthly values over an 8 year period of the volume of increased retail sales (VCVA). A volume forecast was made 12 months ahead with each methodology and the results compared to the actual volume recorded.

**Keywords:** Time series, sarima, Holt-Winters, neural network.

## 1- Introdução

Séries temporais, importância, uso

## 2- Objetivo

È feita uma comparação de três metodologias diferentes de análise para uma série temporal de variável econômica, apresentando brevemente a fundamentação teórica de cada uma.

## 3- Material e Método

3.1 Apresentação

A série temporal a ser analisada pertence ao IBGE e tem dados sobre o volume de vendas do comércio varejista ampliado (VVCVA) entre Janeiro de 2003 e Dezembro de 2011 com Índice de base fixa 2003 = 100. A série inclui além do comércio varejista, os setores de “Veículos, motocicletas, partes e peças” e “Material de construção” sendo extraída em Janeiro de 2018 (Figura 1).

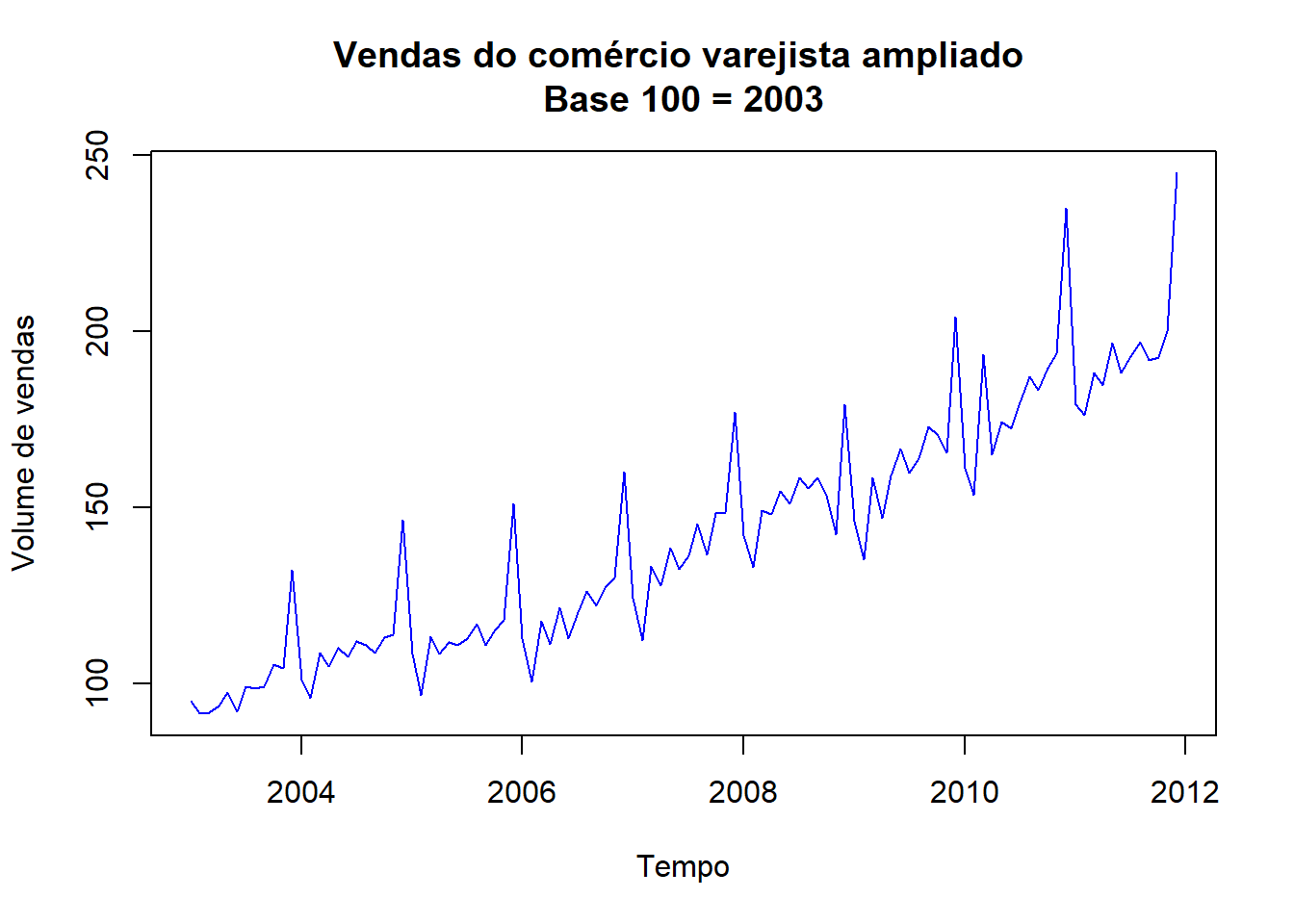


Figura 1 – Volume Vendas Comércio Varejista Ampliado (VVCVA) – 2003-2011

A série apresenta uma *tendência* crescente ao longo dos anos, coerente coma teoria econômica, assim como um pequeno aumento da *variância* em alguns anos.

Nota-se uma variação *sazonal* no fim e começo de cada ano muito semelhante até o ano de 2008, o que parece coincidir com a crise financeira de 2008, após o qual a sazonalidade continua evidente. A Figura 2 gerada com a função *stats::monthplot*, revela com mais clareza a variação sazonal, com valores extremos nos meses de fevereiro e dezembro assim como a tendência de crescimento.

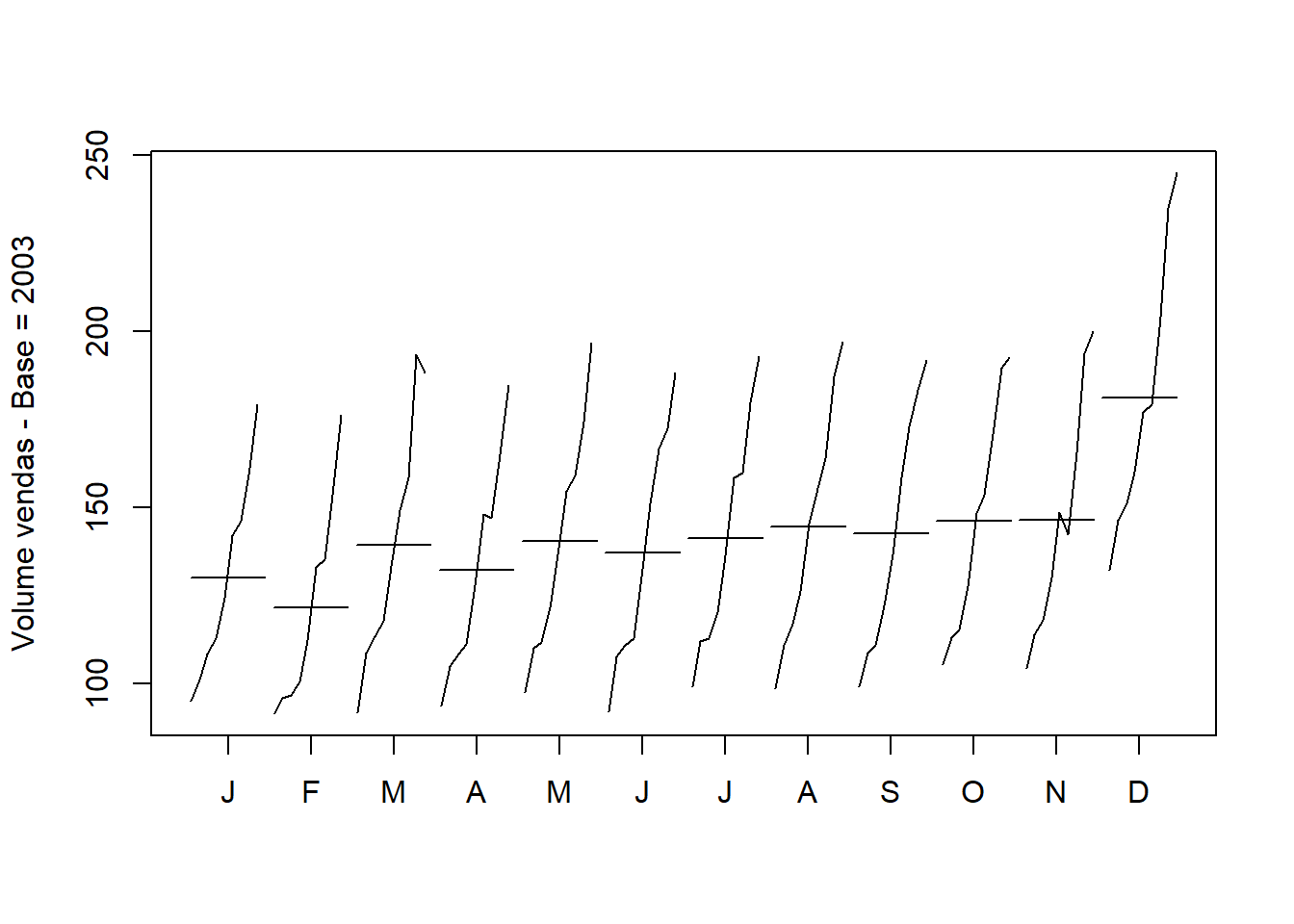


Figura 2 – Volume e média de Vendas por mês

A identificação das características mencionadas, se ajusta a um modelo de decomposição aditivo em que a série é a soma de uma componente de tendência, mais uma componente sazonal mais um residual que, idealmente, deveria ser ruido branco (não tendo contaminação das componentes anteriores. A função *stats::decompose* realiza a decomposição mencionada (Figura 3).

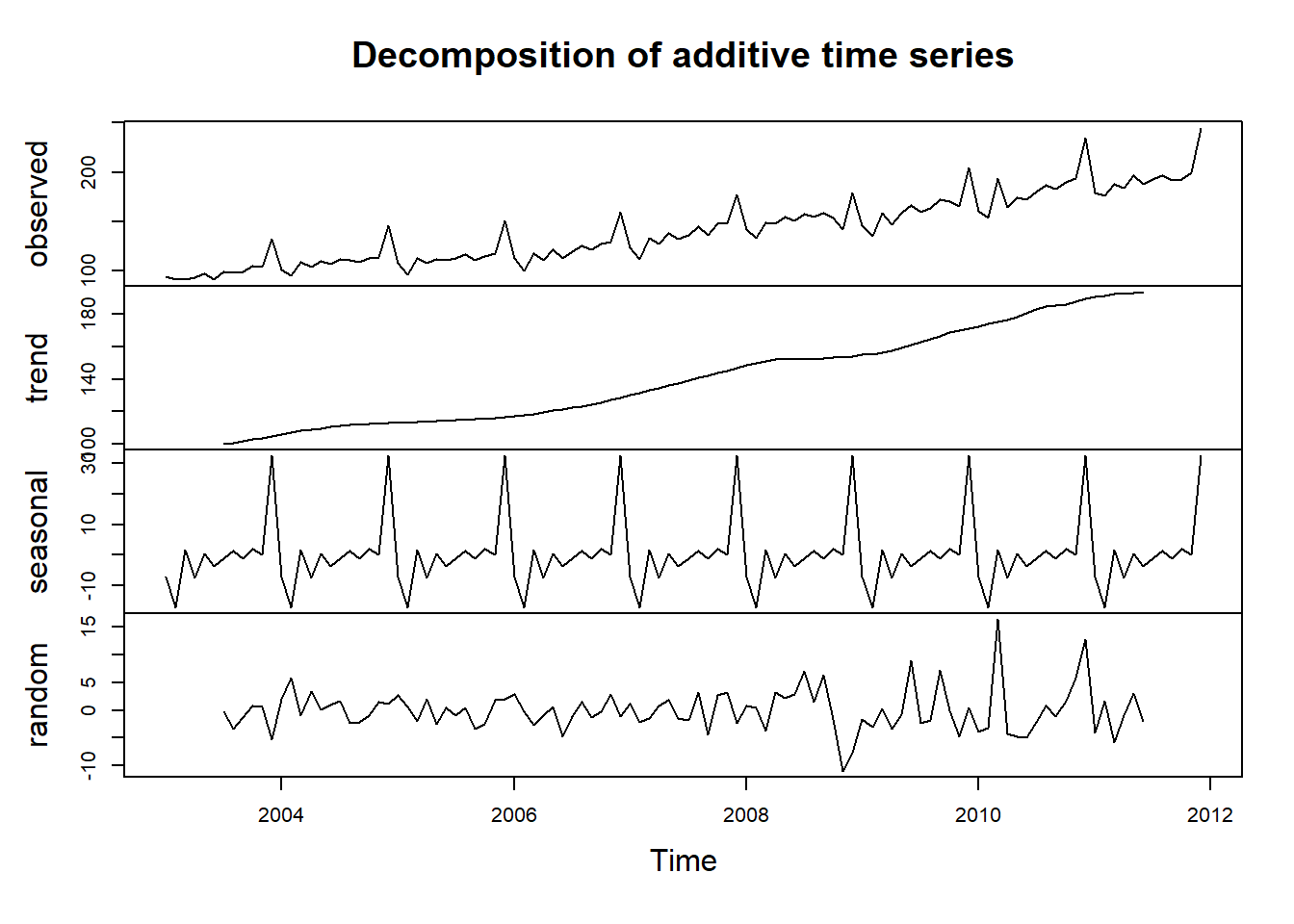


Figura 3 – Volume e média de Vendas por mês

3.2 Modelo SARIMA

Consiste numa modelagem sofisticada que envolve processos de autoregressão (AR) e médias móveis (MA) e que costumam dar bons resultados em séries longas e bem-comportadas. Acorde ao proposto por Box e Jenkins para a modelagem de séries temporais estacionárias, o processo segue um ciclo interativo de 4 partes: **especificação, estimação, diagnóstico e modelo final**. Como condição necessária para o tratamento da série com esta metodologia, devemos garantir que a média e a variância sejam estacionárias. Com o modelo final podemos fazer a predição de valores futuros. Na análise da série, além de funções base do R, foram usadas funções específicas do pacote BETS (Brazilian Economics Time Series).

A estacionalidade da série é testada através da função *BETS.corrgram*. No caso da série não ser estacionária, o número de diferenciações necessárias é determinado por um teste de raiz unitária (RU). Um dos mais usados e o teste de Dickey-Fuller Aumentado, conhecido na literatura como teste ADF(Augmented Dickey-Fuller) e implementado na função *urca::ur.df.* A hipótese H0 é que a série possui uma RU e assim sendo é estacionária.

Uma vez determinado o número de diferenciações necessária, a série é estacionada e o incremento da variância pode ser retirado com a função *log().*

No caso da sazonalidade a série é diferenciada com período de 12 e o teste de RU refeito para comprovar que a série resultante é estacionária.

Na **especificação**, é feita uma análise para determinar a classe de estrutura SARIMA que melhor se adapta, usando funções de autocorrelação e autocorrelação parcial. Assim os valores de *p, q, P* e *Q* são definidos. Pode-se determinar mais de um modelo.

Quando o período de uma série é *s=12*, o modelo denominado SARIMA de ordem (*p, d, q*) × (*P, D, Q*)*12*, é dado por (MORETTIN & TOLOI, 2004):

em que é o operador autorregressivo (**AR**) de ordem *p*,é o operador médias móveis (**MA**) de ordem *q*,é o operador **AR***sazonal* de ordem *P*,é o operador **MA***sazonal* de ordem *Q*,é o operador diferença,é o operador diferença sazonal eé ruído branco.

Na etapa de **estimação**, os parâmetros definidos são calculados e testados estatisticamente para validar sua significância. A função *forecast::auto.arima,* calcula automáticamente o melhor modelo acorde critérios AIC e BIC. Também pode ser usada a função f*orecast::Arima.* Para rápida verificação da significância, a função *BETS::t\_test* proporciona valores para cada coeficiente e analisa os critérios de rejeição das hipóteses.

No **diagnóstico** é feito uma análise dos resíduos e testes de verificação como o de Ljung-Box para conferir se o modelo é adequado. Novamente critérios AIC e BIC são usados na verificação para comprovar que os resíduos têm ausência de autocorrelação linear (*stats::Box.test*) e heterocedasticidade (ARCH ML ou teste Multiplicador de Lagrange para heterocedasticidade condicional regressiva, *FinTS::ArchTest*), assim como seguem uma distribuição do tipo normal (*normtest::jb.norm.test)*. No caso de séries de curto cumprimento, o teste de normalidade pode não ser conclusivo, pelo que um teste de Lillierfors é mais adequado. A função *nortest::lillie.test*, permite testar a normalidade para séries temporais pequenas.

O **modelo final** é escolhido, se temos mais de um os valores de RMSE e MAPE ajudam a determinar o mais adequado para a modelagem da série. A previsão de novos valores é feita com *forecast::forecast*, especificando o horizonte de previsão e o nível de confiança desejado.

3.3 Modelo Holt-Winters

O método Holt-Winters, usa médias móveis com ponderação exponencial para efetuar uma estimação das médias ajustadas por sazonalidade (ou nível), inclinação da série e sazonalidade, sendo muito usado pelo seu baixo custo computacional.

A variação sazonal pode ser modelada de duas formas: tipo multiplicativo (se a sazonalidade é alterada pelo nível da série) ou aditivo (o padrão sazonal não se altera).

A função de predição de Holt-Winters, para séries temporais de período *p* em *h* períodos a frente após a , observação é dada da seguinte forma:

onde , e são os parâmetros de nível, tendência e sazonalidade da série, expressos por:

,e

A implementação do método no R é a través da função *stats::HoltWinters,* que calcula os coeficientes *a, b,* e *s* e gera um objeto S3 da clase “HoltWinters*.* No caso de série anual, como o que vai ser analisado, o total de coeficientes é 14, correspondendo 12 deles a coeficientes sazonais.

O gráfico do objeto pode ser feito com a simples função base plot() , a qual apresenta em 4 janelas separadas os valores ajustados (“xhat”), o nível (“level”), a tendência (“trend”), e a componente sazonal (“season”). Feita a decomposição e calculados os coeficientes, a função genérica *stats::predict* permite estimar valores à frente, sendo que além do intervalo de tempo podemos solicitar um intervalo de confiança.

3.4 Rede Neural

Redes neurais estão baseadas na arquitetura de aprendizagem de organismos inteligentes, que aprendem através da experiência. A partir dos trabalhos de Santiago Ramón y Cajal sobre a estrutura microscópica do cérebro e o funcionamento dos neurônios a começos do século XX, McCulloch e Pitts fizeram o primeiro modelo computacional simulando o comportamento de um neurônio biológico.

A estrutura da rede neural em séries temporais é composta de uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Os valores da série temporal passam por uma função de normalização na camada de entrada e é dado um peso, inicialmente aleatório, para serem enviados às camadas ocultas.

A aprendizagem se dá pelos erros cometidos pelas camadas de processamento, quando comparados com o valor de saída, que é conhecido. O erro serve para redistribuir os pesos entre as camadas de processamento, e o processo se repete até o valor de saída se aproximar do valor desejado ou conhecido.

Existem diversas metodologias para a aprendizagem, em séries de comportamento regular uma das mas efetivas é a de retro propagação, que será usado nesse trabalho. A modificação dos pesos em cada repetição é dada através de uma taxa de aprendizado e um termo adicional de *momentum,* que introduze um efeito de inércia e ajuda a amortecer as oscilações em torno da melhor solução*.*

O pacote *R neuralnet* implementa a metodologia mencionada anteriormente, possibilitando o cálculo de intervalo de confiança para os pesos e gerar gráficos diversos.

## 4- Resultados e Discussão

4.1 - Modelo SARIMA

Como a série aparenta ser não estacionária foi feito um teste de RU usando o teste de Dickey-Fuller Aumentado, conhecido na literatura como teste ADF(Augmented Dickey-Fuller), que confirmou a série ser não estacionária. Foi realizada uma diferenciação, primeiro com *lag = 1* e depois com *lag = 12*  usando a função log para tirar ao mesmo tempo o incremento da variância (Figura 5).

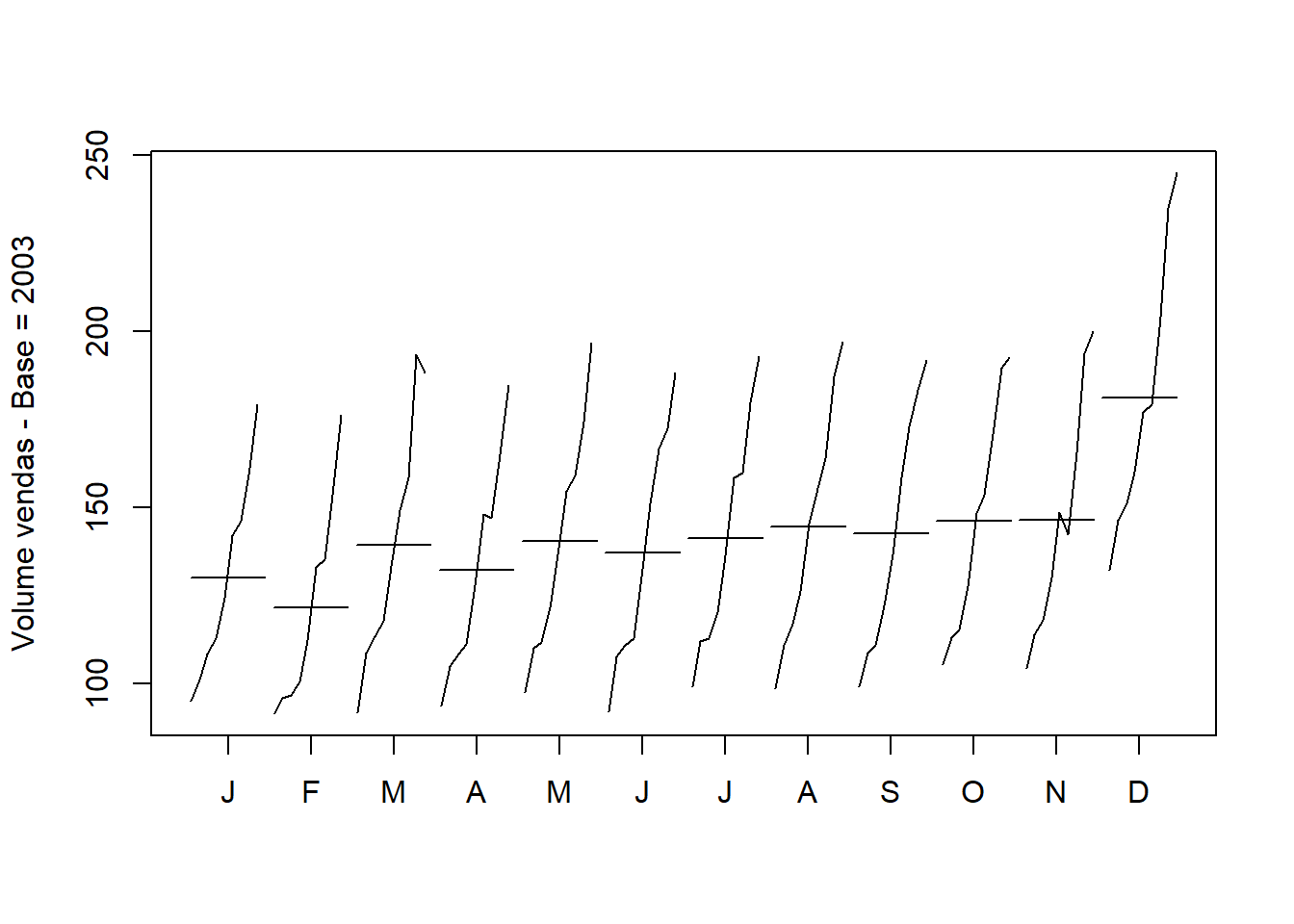


Figura 4 – Volume e média de Vendas por mês

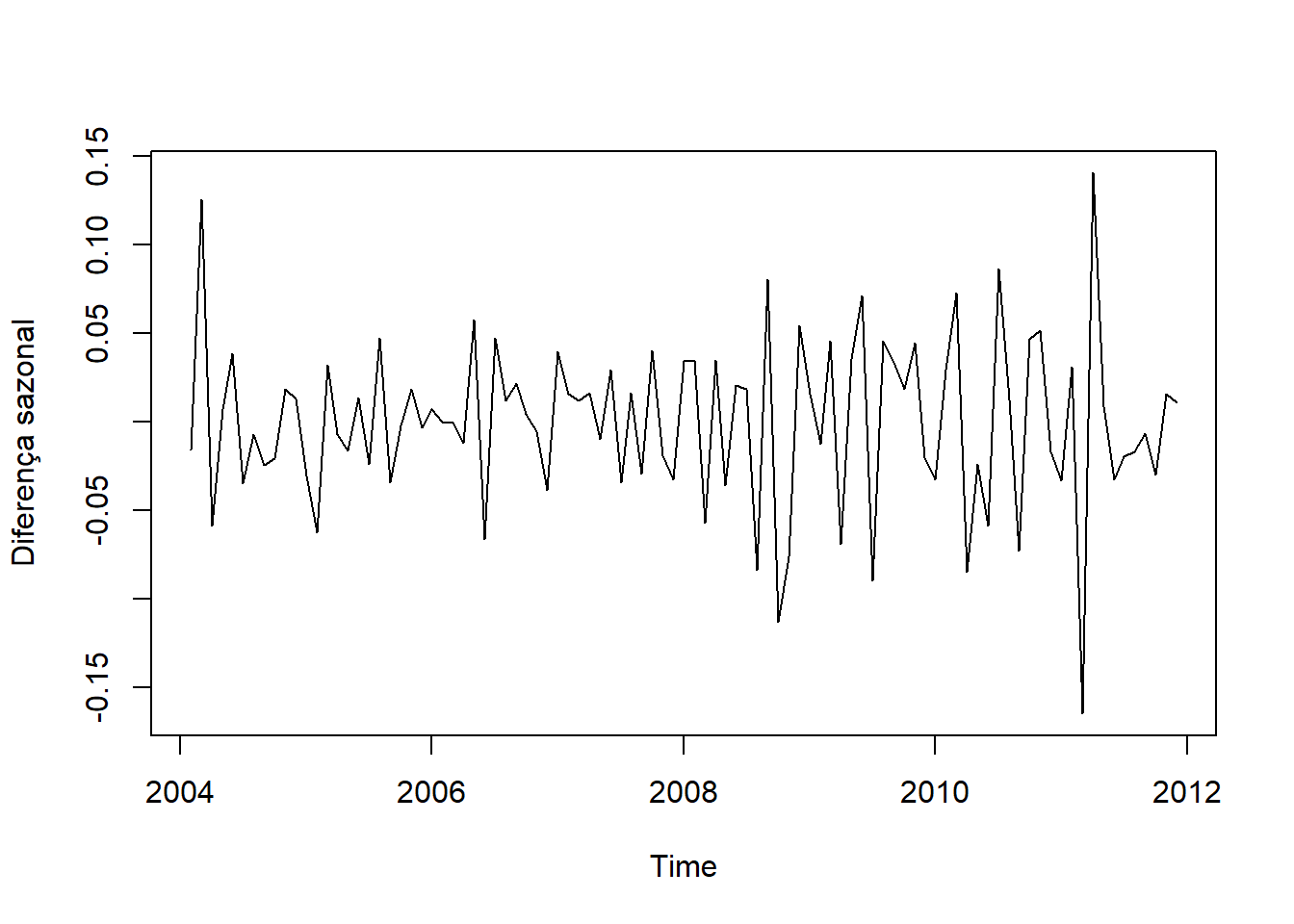


Figura 5 – Diferenciação com período de 12 meses

Feito novamente o teste de RU o valor de inferior a valor crítico a 95% de probabilidade confirmou que a série resultante é estacionária.

**Especificação e Estimação** - Foram feitos dois modelos, um com a sugestão da função *auto.arima (0,1,1)(0,1,2)[12]* e outro com uma variação (0,1,1)(0,1,1)[12]. O valor AIC do segundo foi melhor e foi o adotado. A função *BETS.t\_test* apresentou os seguintes resultados para os coeficientes:

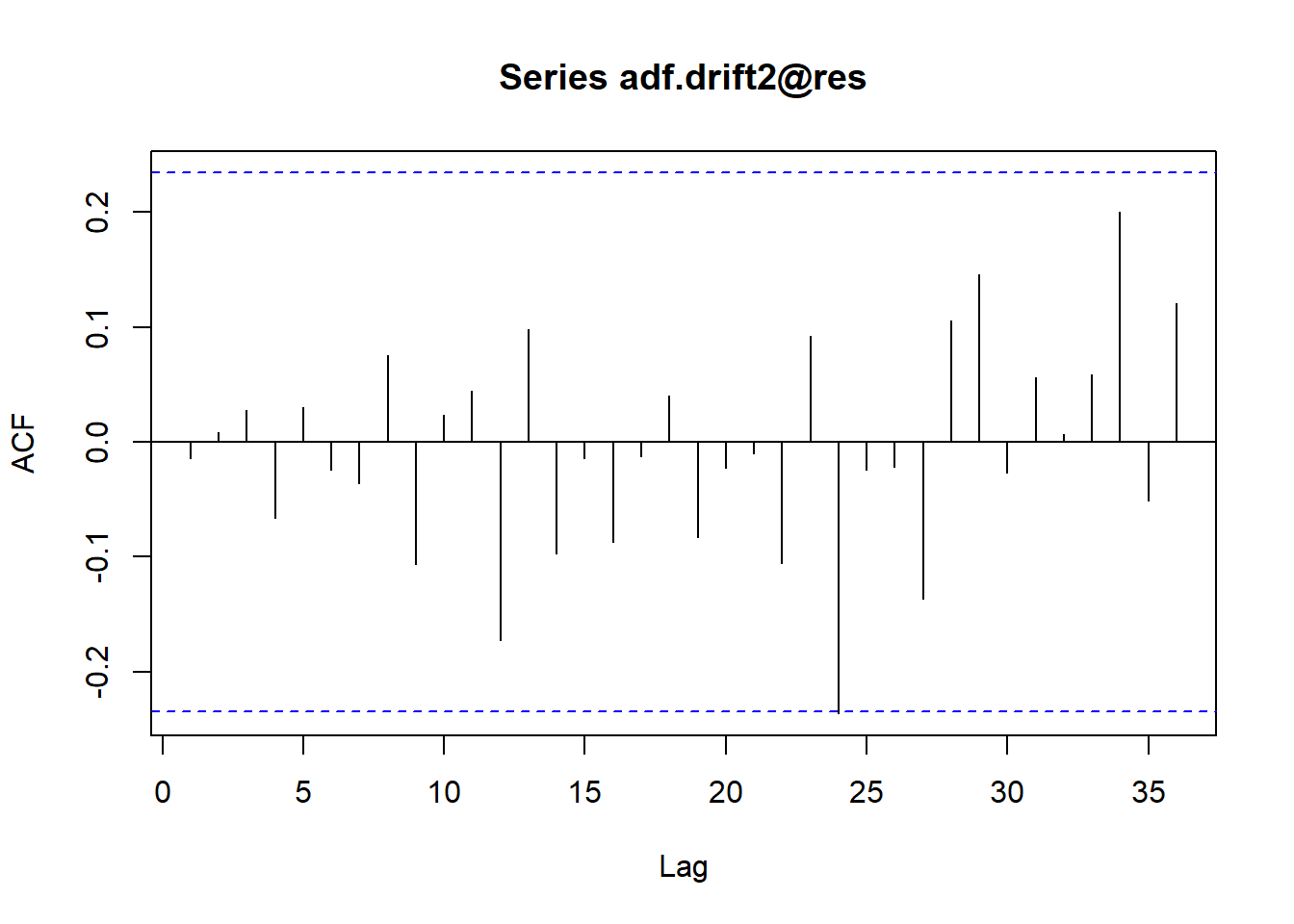


Figura 6 – Volume e média de Vendas por mês

**Diagnóstico**

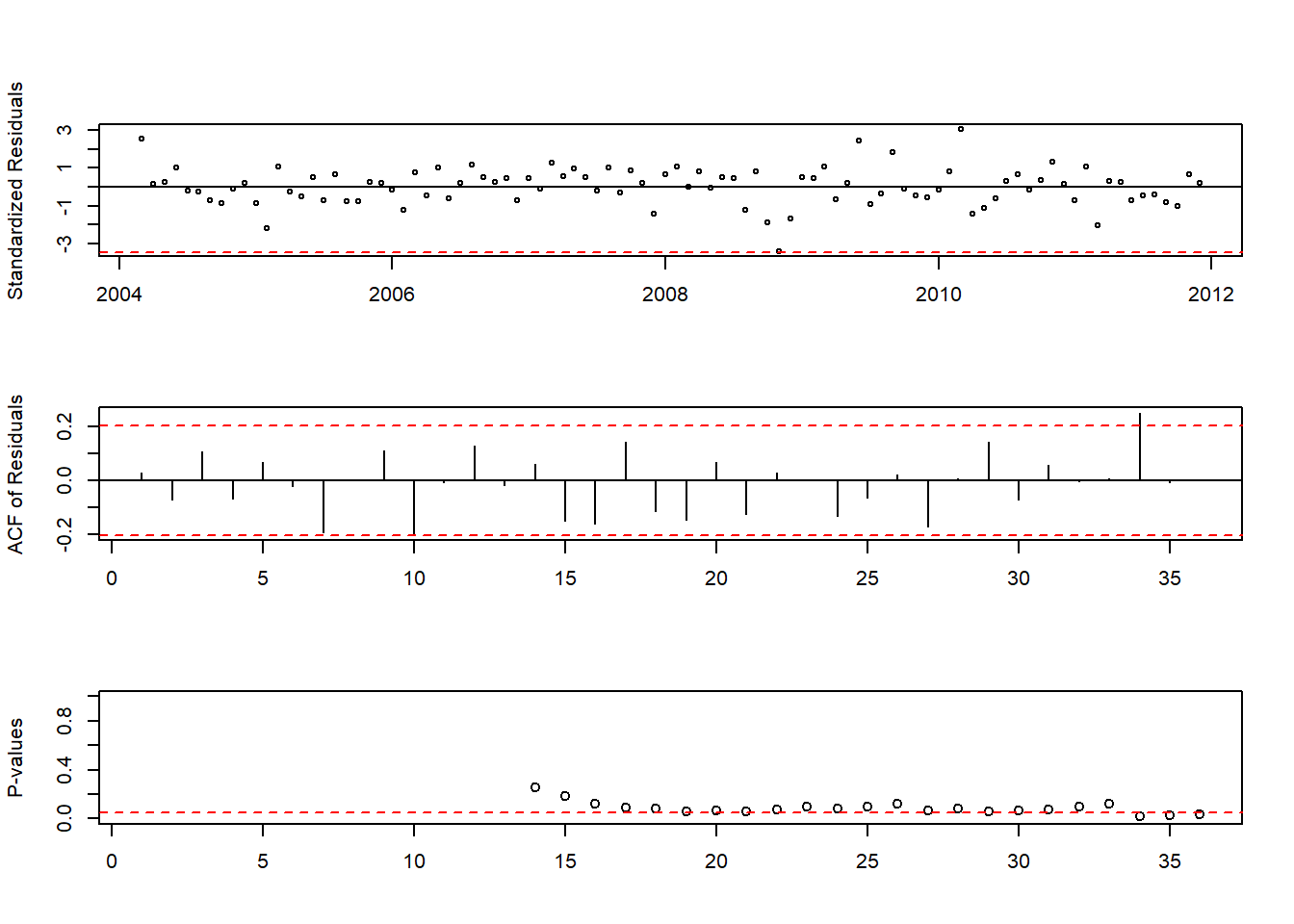
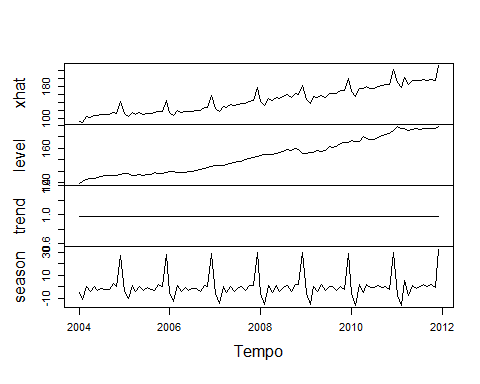
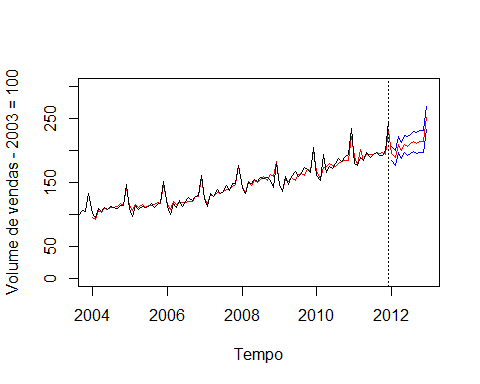


Figura 7 – Volume e média de Vendas por mês

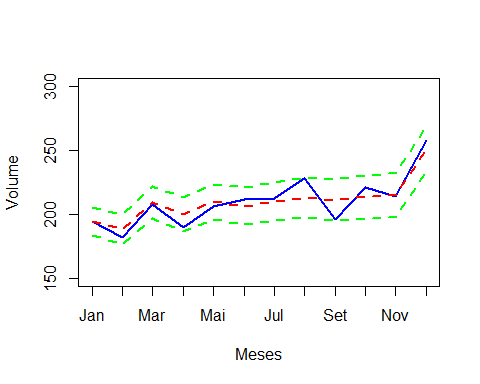
4.2 - Modelo Holt-Winters



**Figura** 8 – Decomposição da série com a função *stats::HoltWinters*



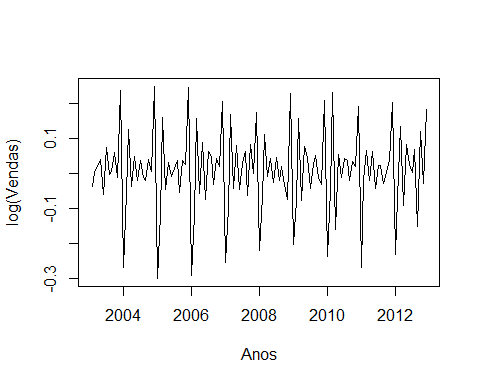
**Figura** 9 – Serie original, ajustada e previsão 2012 com intervalo de confiança



**Figura** 10 – Previsão de 2012 com valores reais em azul

4.3 - Rede Neural

Como preparação da série foi aplicada uma função *log* para estabilizar a variação e uma diferenciação de primeira ordem com passo de 1 para retirar a tendência. A figura XX apresenta o resultado dessa preparação

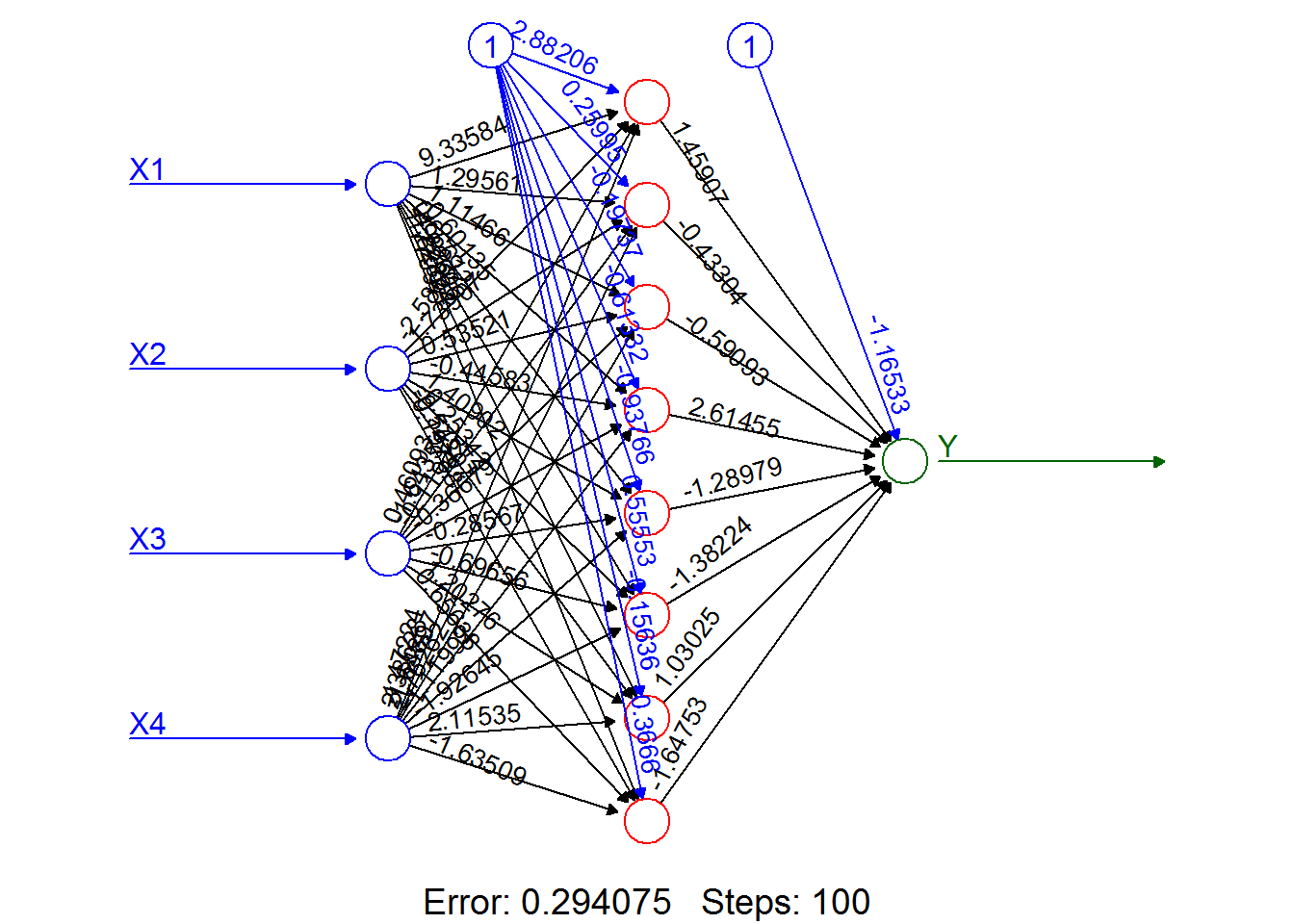


**Figura** 11 – Serie temporal após preparação dos dados

Na construção da rede neural, definimos os valores da camada de entrada, a quantidade de nós na camada oculta, assim como os valores na camada de saída. Nesse caso foram escolhidos 4 valores de entrada, correspondendo ao valor do mês anterior, o segundo mês anterior, o quarto e o décimo segundo, caracterizando valores anuais, trimestrais e de dois meses prévios. Os nós da camada oculta devem ser suficientes para modelar o fenômeno, nem muitos que ocasionem um *overfitting,* nem poucos que percam peculiaridades da série. A seguinte tabela apresenta o padrão de escolha de valores de entrada e o correspondente de saída.



A aprendizagem na rede é feita a partir de uma seleção de amostras, chamadas de treino e teste. Para esse estudo temos um total de 107 padrões dos quais vamos selecionar 95 para o treinamento 12 de teste. Serão usados 8 nós ou neurônios em uma camada oculta. Esse valor foi o melhor encontrado depois de testar de 3 a 10 nós. Os argumentos principais da função *neuralnet* são a fórmula usada (Y ~ X1 + X2 + X3 + X4), os dados de entrada e os nós da camada oculta (8). A rede formada e o treinamento com os valores e pesos finais é apresentada na seguinte figura.



**Figura** 12 – Rede neural com valores e pesos finais

Após o cálculo da rede, são feitas duas previsões de valores chamadas *in-sample* e *out-sample.* A primeira usando os valores do treinamento e a segunda sobre os valores de teste, previamente separados. As previsões servem para saber a precisão que a rede atinge. Finalmente é feita uma previsão um passo à frente sobre valores fora do intervalo da amostra. Nesse caso prevemos 12 valores para o ano de 2012 que são apresentados como uma linha vermelha, enquanto os valores reais da varável são representados por círculos cheios. Foram incluídos os valores de precisão MAD e MAPE.

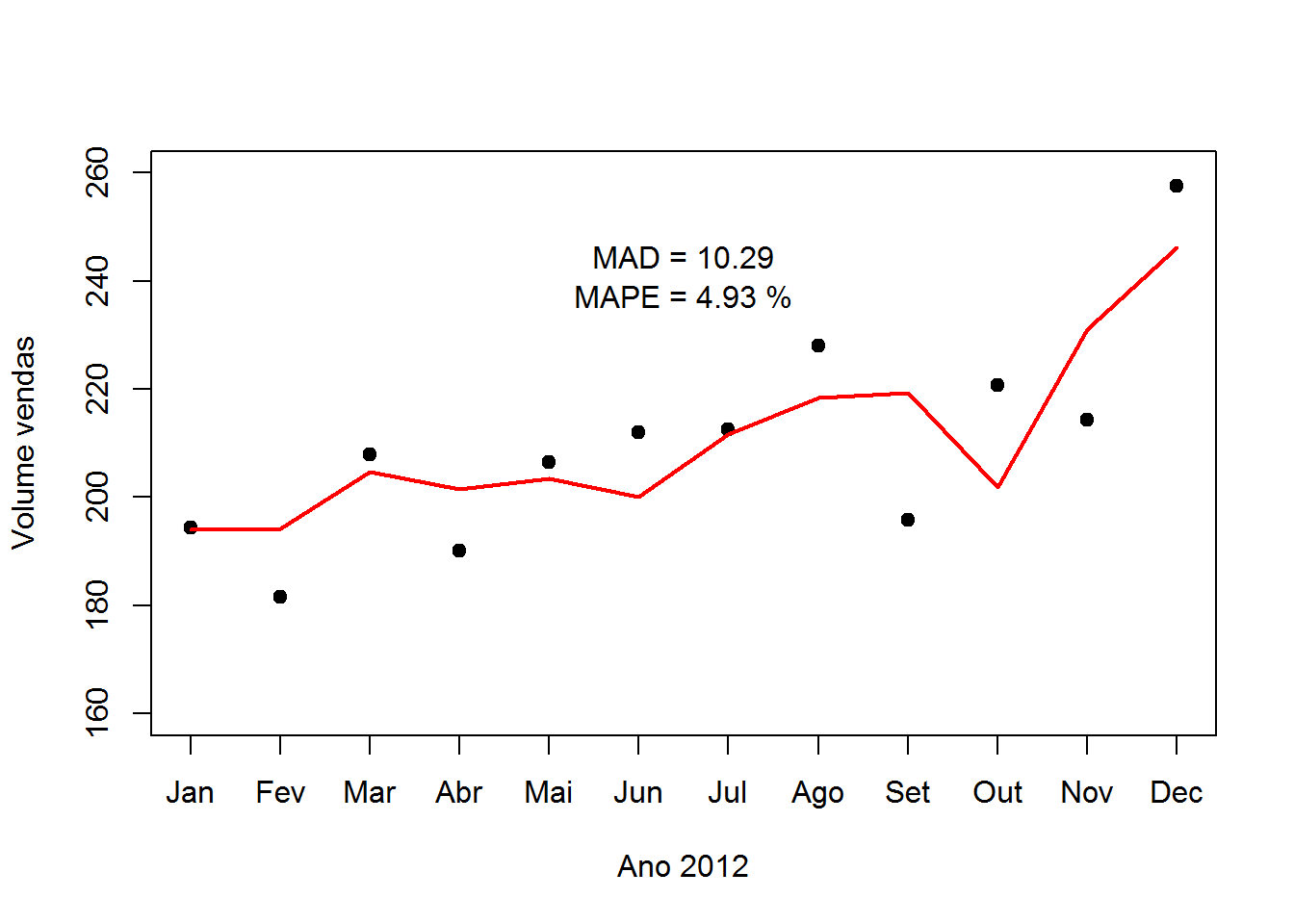
****

Figura 13 – Previsão para 2012 (linha vermelha), valores reais (pontos) e parâmetros e precisão

## 5- Conclusão

Nesta seção são apresentadas as conclusões do trabalho.

## 6- Referências

C. Chatfield (1978) The Holt-Winters Forecasting Procedure Journal of the Royal Statistical Society. Series C (Applied Statistics) Vol. 27, No. 3, pp. 264-279

C. C. Holt (1957) Forecasting seasonals and trends by exponentially weighted moving averages, ONR Research Memorandum, Carnegie Institute of Technology 52. (reprint at <http://dx.doi.org/10.1016/j.ijforecast.2003.09.015>).

P. R. Winters (1960) Forecasting sales by exponentially weighted moving averages, Management Science 6, 324–342.

## Anexo

Os scripts completos de cada análise podem ser consultados em GitHub

Resumo das principais linhas de script

**1- Preparação dos dados**

vendas <- read.csv(".../SER\_III/ST/data/IBGE\_VVCVA\_RN.csv",header=FALSE,

stringsAsFactors = F) #transposiçãoo de linha a coluna em dataframe

vendas2 <- as.data.frame(t(vendas)) # transformaçãoo em st\_Vendas temporal

st\_Vendas <- ts(vendas2, start = c(2003,1), frequency = 12)

**2- SARIMA**

**3- Holt-Winters**

HW <- HoltWinters(st\_Vendas) # função filtrado sobre a série temporal

coef <- as.data.frame(HW$coefficients) # coeficientes calculados

p <- predict(HW, 12, prediction.interval = TRUE, level = 0.95) # previsão 12 meses

# O horizonte de previsão calculado é uma matriz de três colunas contendo o valor calculado assim como os limites superior e inferior de confiança ao 95%. Incorporando os valores realmente observados em 2012, e fazendo a diferença entre real e observado temos:

hw <- as.data.frame((matrix(round(p,2), 12)), dimnames(seq(1,12,1)))

hw\_res <- cbind(hw,vol\_2012)

names(hw\_res) <- c("prev", "sup","inf","real")

hw\_res <- hw\_res %>% mutate(dif = real-prev)

**4- Rede Neural**

logst\_Vendas <- log(st\_Vendas) # estabilizar a variância

delta <- diff(logst\_Vendas,lag = 1, differences = 1) # retirar a tendência da série

normalizado <- (delta-minimo)/(maximo-minimo) # normalização da série

lags <- c(1, 2, 4, 12) # determinação de lags

inic <- max(lags) + 1 # inicar em valor número 13

nobs <- length(normalizado) # número de observações: 107

modelo\_nn <- neuralnet(formula = Y ~ X1+X2+X3+X4, data= data.train[1: ninsample, ], hidden = 8, linear.output = F) # treinamento da rede

1. Instituto Brasileiro de Geografia e Estatística (IBGE), [walter.pina@ibge.gov.br](mailto:walter.pina@ibge.gov.br) [↑](#footnote-ref-2)
2. Fundação Getúlio Vargas (FGV), [pedro.guilherme@fgv.br](mailto:pedro.guilherme@fgv.br) [↑](#footnote-ref-3)
3. Universidade do Estado do Rio de Janeiro (UERJ), [professorjmpf@hotmail.com](mailto:professorjmpf@hotmail.com) [↑](#footnote-ref-4)