TSINTEGER: UM PACOTE R PARA ANÁLISE DE SÉRIES TEMPORAIS CONSTITUÍDAS DE DADOS DE CONTAGEM

Bruno Patriota Soares[1]
[1]*Universidade Federal de Campina Grande, brunopatriotabps@gmail.com*

Marcelo Bourguignon Pereira[2]
[2]*Universidade Federal do Rio Grande do Norte, m.p.bourguignon@gmail.com*

Manoel Ferreira Santos Neto[3]
[3]*Universidade Federal de Campina Grande, mn.neco@gmail.com*

RESUMO: Séries temporais constituídas de dados de contagem têm chamado atenção pela sua grande aplicabilidade. Usualmente os processos estocásticos assumem que as marginais são contínuas e geralmente não são adequados para analisar séries de contagem. Neste trabalho propomos a criação de um pacote R, para análise de séries temporais constituídas de dados de contagem, que foi chamado de tsinteger. Neste pacote o usuário pode escolher diferentes tipos de operadores, a distribuição do erro e a ordem do modelo. O pacote tem funções que retornarão as estimativas dos parâmetros e medidas de qualidade de ajuste. No pacote tsinteger, o usuário tem, também, a opção de gerar valores de uma série temporal, bastando escolher as especificações acima e os valores dos parâmetros. Por fim, demonstramos a utilização do pacote com exemplos simulados.

Palavras Chaves: Séries temporais, Valores inteiros, pacote R, ajuste.

INTRODUÇÃO

A análise de dados de contagem é de grande importância na estatística uma vez que diferentes fenômenos ocorridos na natureza podem ser expressos como tais. Situações como: número de pacientes internados num hospital, número de acidentes rodoviários e número de pessoas inadimplentes são situações que motivam uma adequada análise estatística.

Podemos definir uma série temporal como sendo um conjunto de observações y_t, onde cada uma é coletada em um específico tempo t. Objetivando caracterizar a natureza dessas observações, é comum supor que cada valor da série temporal y_t é uma realização de uma variável aleatória Y_t, ou seja, y_t é uma realização da família de variáveis aleatórias ou processos estocástico Y_t. Dizemos que uma série temporal é dita contínua quando as observações são feitas continuamente no tempo. Já numa série temporal dita discreta o conjunto dos tempos em que as observações são feitas é discreto. Este trabalho considera séries temporais discretas. Vale ressaltar que esses termos não fazem referência à variável

observada y_t, essa pode assumir valores discretos ou contínuos. Se o processo é constituído de variáveis aleatórias com distribuição contínua, dizemos que o processo é de marginal contínua como, por exemplo, os processos autorregressivos e de médias móveis (ARMA) (ver Box et al., 1994). Mas se o processo é constituído por variáveis aleatórias que possuem uma distribuição discreta, então o processo é dito de marginal discreta e são chamados de *processos de valores inteiros* ou *processos de contagem*.

Nos últimos anos vem crescendo o desenvolvimento de classes de modelos para estudo de processos de contagem. De um modo geral, as pesquisas se concentram na formulação e estudo de propriedades dos modelos (Al-Osh & Alzaid, 1987), estimação (Jung et al., 2005), testes de hipóteses e distribuições assintóticas dos estimadores dos modelos para diferentes distribuições marginais discretas (Freeland & McCabe , 2005). A análise estatística baseada em séries temporais tem nos modelos autorregressivos integrados de médias móveis (ARIMA) desenvolvido por Box et al. (1994) sua técnica mais difundida. Entretanto, esses modelos assumem que as marginais são contínuas e, em geral, com distribuição normal. No entanto, quando consideramos dados de contagem e estes possuem magnitude pequena podemos encontrar dificuldades para utilização desses modelos.

O R é um software livre para análise estatística, simulações e geração de gráficos (ver R Core Team, 2015). O R deriva da linguagem de programação S (ver Becker et. al., 1988). Ele está disponível em uma ampla variedade de plataformas como UNIX, Windows e MacOS. Em R Core Team (2015) temos um manual completo do programa e em Ligges (2003) encontramos uma pequena introdução ao R. A criação de pacotes R é uma maneira prática de manter e divulgar coleções de funções construídas na linguagem R. Antes de criamos um pacote R é necessário conhecer alguns conceitos:

- *Package:* Uma extensão do sistema básico R com códigos, conjunto de dados e uma documentação com formato padronizado;
- *Library:* Um diretório contendo os pacotes instalados;
- *Repository:* Um site que disponibiliza os pacotes para instalação;
- *Source:* Versão original de um pacote com texto humanamente legíveis e códigos;
- *Binary:* Uma versão compilada do pacote com texto legível em computador e códigos, pode trabalhar em apenas uma plataforma específica;
- *Base packages:* Parte da árvore de códigos R, mantido por R Core Team;
- *Recommended Packages:* Parte de cada instalação R, mas não necessariamente mantido por R Core Team;

- *Contributed Packages:* Todo o resto. Isto não quer dizer que estes pacotes são necessariamente de qualidade inferior aos acima, por exemplo, muito *Contributed Packages* em CRAN (A rede de arquivos globais R) são mantidos por membros do R Core Team.

Neste trabalho criamos um pacote R, para análise de séries temporais constituídas de dados de contagem, que chamamos de tsinteger. Neste pacote o usuário pode escolher diferentes tipos de operadores, a distribuição do erro e a ordem do modelo. O pacote tem funções que fornecem as estimativas dos parâmetros e medidas de qualidade de ajuste.

O objetivo deste trabalho é estudar os procedimentos necessários para a criação de um pacote R, assim como as funções que compõem o mesmo; criar um pacote para séries temporais constituídas de dados de contagem com o nome tsinteger; e demonstrar a utilização do pacote tsinteger através de exemplos simulados.

METODOLOGIA

A metodologia utilizada neste trabalho constituiu em uma revisão bibliográfica sobre análise de séries temporais constituídas de dados de contagem. Essa revisão foi realizada por meio da leitura de artigos disponibilizados pelo professor orientador Dr. Manoel Santos-Neto e pelo professor Dr. Marcelo Bourguignon (DE-UFRN) sobre o conteúdo. Além disso, foram feitas inúmeras pesquisas na internet sobre o software R e como desenvolver um pacote no mesmo. Sendo este o objetivo principal deste trabalho.

Também visando uma aprendizagem mais aprofundada na teoria das séries temporais constituídas de dados de contagem, realizamos uma visita científica ao Departamento de Estatística da Universidade Federal do Rio Grande do Norte para trabalhar com o professor Dr. Marcelo Bourguignon. Como especialista da área o professor Dr. Marcelo Bourguignon vem nos auxiliando na parte teórica deste trabalho.

RESULTADOS E DISCUSSÕES

Uma das principais vantagens da criação de um pacote R é a economia de tempo em tarefas futuras. Além disso, é uma maneira bastante eficiente de difundir conhecimento com a comunidade Estatística no Brasil e no mundo. Inicialmente o pacote desenvolvido neste trabalho não estará disponível no CRAN, mas no GitHub. Antes de explicarmos o que é o GitHub é necessário falar um pouco sobre o que é o Git. O Git é um sistema de controle de versão de arquivos onde podemos desenvolver projetos e contribuir com este simultaneamente

com diversas pessoas ao mesmo tempo. Já o GitHub é um serviço web que disponibiliza várias funcionalidades extras ao Git.

Criando uma conta no GitHub

Para criar uma conta no GitHub, é necessário apenas acessar o endereço https://github.com/ e fornecer algumas informações básicas como o nome de acesso, e-mail e senha. Logo após a criação da sua conta, você irá clicar no botão +New Repostitory onde será possível criar um repositório para seu projeto. Nesta etapa você irá fornecer o nome do repositório, informar se o mesmo será público ou privado e escolher o tipo de licença.

Para o nosso projeto criamos um repositório com o nome tsinteger, sendo este público e sob a licença GNU GLP v2. O repositório do projeto pode ser acessado através do seguinte endereço https://github.com/projecttsinteger/tsintegerpackage .

Criando o pacote tsinteger

Para criação de um pacote no R é necessário seguir alguns passos, dos quais estão listados abaixo, onde é possível ver a estrutura mínima para sua criação, sendo possível observar funções criadas no decorrer da pesquisa e alguns detalhes, como documentação e instalação.

Passo 1: Instalar o R e o RStudio e instalar os pacotes necessários.

Para baixar os executáveis dos programas R e RStudio basta acessar os respectivos endereços https://cran.r-project.org/bin/windows/base/ e https://www.rstudio.com/products/rstudio/download/ .

Para a criação do pacote tsinteger no RStudio instalamos os pacotes devtools, roxygen2, testthat, knitr. No Windows precisamos instalar o Rtools que está disponível em https://cran.r-project.org/bin/windows/Rtools/. Por exemplo, para instalar o pacote devtools podemos usar o comando install.packages("devtools").

Passo 2: Clonar o diretório para o GitHub Desktop

Primeiro deve-se instalar o GitHub Desktop na sua máquina. Para realizar a instalação basta visitar o endereço https://desktop.github.com/. Com o GitHub Desktop podemos clonar o repositório a partir do GitHub. Após clonarmos o repositório tsinteger na máquina começamos a estrutura-lo e para tal abrimos o Rstudio e usamos o comando:

```
devtools::setup("caminho na máquina para o repositório clonado")
```

Desta forma, cria-se a estrutura básica necessária para construir um pacote, descrita abaixo:

DESCRIPTION: informações do pacote

NAMESPACE: trata das iterações do seu pacote com outros pacotes

R/: pasta onde fica os códigos em R

Man/: pasta onde fica a documentação

Rproj: seu projeto

Passo 3: Criar as funções

Para criação das funções é necessário um conhecimento prévio de programação. Os arquivos com as funções devem ser salvos com a extensão .R e cada arquivo pode conter mais de uma função. Abaixo uma função criada para o pacote tsinteger.

```
nginar <- function(x,series=NULL)
{
if (is.null(series))
series <- deparse(substitute(x))
xfreq <- frequency(x)
n <- length(x)
order.max = 1
r0 <- acf(x, plot = FALSE)$acf[1]
r <- acf(x, plot = FALSE)$acf[2:(order.max+1)]
R <- diag(order.max)
for(i in 1:order.max){
for(j in 1:order.max){
if(i!=j){
R[i,j] <- r[abs(i-j)]
}
}
}
residuals <- function(x,coef,mu)
```

```
{
x<- x
p<- 1
e <- NULL
for(t in (p+1):length(x) )
{
e[t] <- x[t] - ( sum(coef*x[(t-1):(t-p)]) + mu)
}
return(e)
}
coef <- round(solve(R, r), 4)
xbar <- mean(x)
mu.e <- xbar*(1-sum(coef)) #mean of the error
var.error <- r0 - sum(coef*r) #variance of the erro
mu.x <- mu.e/(1-sum(coef))
sum.var <- sum( coef*(1-coef))
Vp <- var.error + mu.x*sum.var
fitted <- nginar.sim(length(x), alpha = coef, mu = mu.e, n.start=200)
resid <- residuals(x,coef,mu.e)
rms <- sqrt(mean(resid^2,na.rm = TRUE))
AICc. <- n*log(Vp) + n*((1+order.max/n)/(1-(order.max+2)/n))
AIC. <- n*log(Vp) + 2*order.max
BIC. <- n*log(Vp) + (order.max/n)*log(n)
res <- list(order = order.max, coef = coef,  mean.e = mu.e,  var = var.error,  rms = rms,
fitted.values = fitted, bic= BIC., aicc=AICc., aic = AIC., n.used = n, order.max = order.max,
resid = resid, method = "yule-walker",  series = series,  frequency = xfreq,  call = match.call())
class(res) <- "inar"
return(res)
}
```

Passo 4: Documentar as funções

Através do pacote roxygen2 é possível escrever a documentação dentro da própria função .R. A documentação usando o pacote roxygen2 segue a seguinte estrutura:

- Todas as linhas da documentação iniciam com #';
- Os elementos da documentação são criados a partir de *tags* que iniciam com @, por exemplo @title;
- Toda documentação deve ficar antes do início da função.

```
#' Function poinar.sim
#' Simulate from an Inar model
#' @param n the length of outputs series. A strictly positive integer.
#' @param order.max the integer component p is the INAR order.
#' @param alpha a vector of INAR coefficients.
#' @param lambda the mean of the poisson distribution.
#' @param n.start the length of 'burn-in' period. If na, the default, a reasonable valve is computed.
#'@return A time-series object of class "ts".
#'@seealso \code{\link{poinar}}
#'@references
#' Du, J.G. and Li,Y.(1991): The integer-valued autorregressive (INAR(p)) model.
#' \emph{Journal of time series analysis} \bold{12}, 129--142.
#'@examples
#'# A Poisson INAR simulation
#'ts.sim <- poinar.sim(n = 100, order.max = 2, alpha = c(0.1,0.4),lambda = 2, n.start=200)
#'ts.plot(ts.sim)
#' @export
poinar.sim <- function(n, order.max, alpha,lambda, n.start=NA){
length. <- n + n.start
x <- rep(NA, times = length.)
error <- rpois(length., lambda)
for (i in 1:order.max) {
x[i] <- error[i]
}
for (t in (order.max + 1):length.) {
x[t] <- 0
for (j in 1:order.max) {
x[t] <- x[t] + rbinom(1, x[t - j], alpha[j])
```

```
}
x[t] <- x[t] + error[t]
}
ts(x[(n.start+1):length.],frequency = 1,start=1)
}
```

Usando o pacote tsinteger

Para usar o pacote tsinteger no R basta instalar e carregar como qualquer outro pacote R. A seguir ensinamos como instalar e carregar o pacote tsinteger.

```
>library(devtools) #carregando o pacote devtools

>devtools::install_github("projecttsinteger/tsintegerpackage") #instalando o pacote tsinteger

>library(tsinteger) # carregando o pacote tsinteger
```

A seguir mostrarmos como usar a função poinar.sim().

```
# Gerando um processo Poisson INAR
> ts.sim <- poinar.sim(n = 100, order.max = 2, alpha = c(0.1,0.4),lambda = 2, n.start=200)

> ts.plot(ts.sim)
```

Na Figura 1 mostramos o gráfico do processo Poisson INAR simulado pela função poinar.sim().

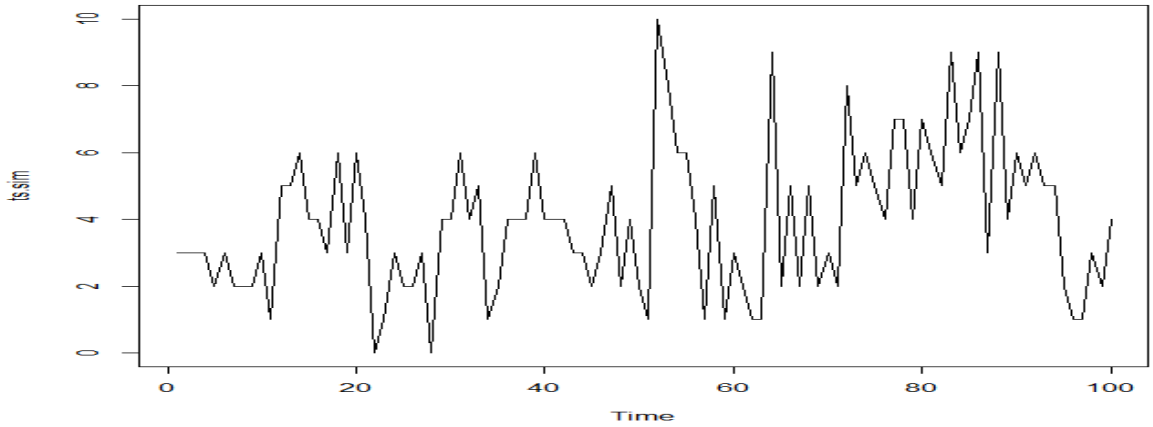

Figura 1. Processo Poisson INAR

CONCLUSÕES

Neste trabalho desenvolvemos um pacote na linguagem R chamado de tsinteger e apresentamos os procedimentos básicos necessários para criação do mesmo. No pacote tsinteger o usuário pode escolher diferentes tipos de operadores, a distribuição do erro e a ordem do modelo. Além disso, o pacote contém funções que fornecem as estimativas dos parâmetros e medidas de qualidade de ajuste.

Referências Bibliográficas

AL-OSH, M. A.; ALZAID, A. A. First-order integer valued autoregressive (INAR(1)) process. **Journal of Time Series Analysis**, 8, 1987. 261-275.

BECKER, R. A.; CHAMBERS, J. M.; WILKS, A. R. **The new S language**. [S.l.]: Chapman and Hall/CRC, 1988.

BOX, G. E. P.; JENKINS, G. M.; REINSEL, G. C. **Time series analysis:** forecating and control. New Jersey: Prentice-Hall, 1994.

CHAMBERS, J. M.; WILKS, A. R.; BECKER, R, A. **The New S Language:** A Programming Enviroment for Data Analysis and Graphics. London: Wadsworth & Brookscole, 1988.

FREELAND, K.; MCCABE, B. P. M. Asymptotic properties of CLS estimators in the Poisson AR(1) model. **Statistics & Probability Letters**, 73, 15 jun. 2005. 147-153.

JUNG, R. C.; RONNING , G.; TREMAYNE, A. R. Estimation in conditional first order autoregression with discrete support. **Statistical Papers**, 46, abr. 2005. 195-224.

LIGGERS, U. R help-desk: package management. **R News**, 3, dez. 2003. 37-39. Disponivel em: <https://cran.r-project.org/doc/Rnews/Rnews_2003-3.pdf>.