Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Estatística

Monitoria PMG
Artur de Souza Plácido Teixeira
Pedro Henrique de Matos Araujo
Rafael Dias de Brito Oliveira
e Adrian Luna (orientador)

Sumário

1 Introdução à Séries Temporais **2**
1.1 Decomposição Clássica . . . 3
1.1.1 Sazonalidade . . . 4
1.1.2 Tendência . . . 4
1.2 Estacionariedade . . . 5
1.2.1 Processos Estocásticos Estacionários . . . 5
1.2.2 Processos Estocásticos Não-Estacionários . . . 6
1.3 Tipos de Séries Temporais . . . 6
1.3.1 Processos Auto-Regressivos Estacionários . . . 6
1.3.2 Processos Auto-Regressivos Não-Estacionários . . . 8
1.3.3 Processo de Médias Móveis (MA) . . . 8
1.4 Processo Auto-Regressivo e com Médias Móveis (ARMA) . . . 8

2 Utilizando o RCommander: downloads e dados **11**
2.1 Como baixar o plugin . . . 11
2.2 Como escolher e baixar o banco de dados . . . 11
2.3 Como importar o banco de dados no RCommander . . . 13

3 Utilizando o RCommander: a interface e modelagens do plugin **15**
3.1 Ts-Data . . . 15
3.1.1 Time Series Conversion . . . 15
3.1.2 Historical Monthly Stock Prices . . . 16
3.1.3 Box Cox Trans . . . 19
3.1.4 Edit TS Data . . . 19
3.1.5 Temporal Aggregation . . . 19
3.2 Ts-Models . . . 20
3.2.1 Run Multiple ARMA . . . 21
3.2.2 Run Multiple GARCH . . . 22
3.2.3 Forecasting . . . 22
3.2.4 Filtering Models . . . 23
3.2.5 ARIMA/GARCH . . . 23
3.2.6 Decomposition . . . 24

Capítulo 1

Introdução à Séries Temporais

Por definição, uma série temporal é uma coleção de observações feitas sequencialmente ao longo do tempo, portanto, quando se lida com séries temporais a ordem dos dados cumpre um papel essencial. Tal instrumental é de grande relevância para as Ciências Sociais como um todo, uma vez que grandes temas abordados por essa área do saber podem ser observados utilizando uma série temporal, como taxas de mortalidade e fecundidade, fluxos de migração, dentre outros.

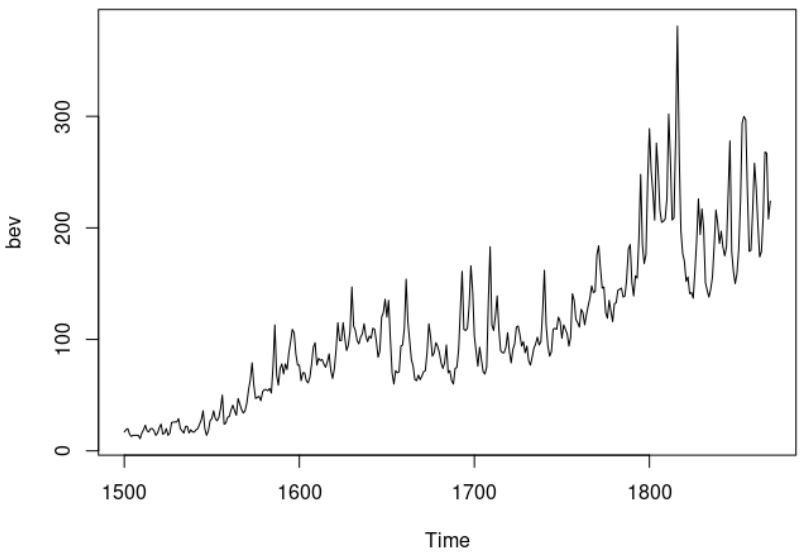
Formalmente, uma série temporal é definida como um processo estocático, ou seja, uma série de variáveis aleatórias indexadas ao tempo, tal processo possui a seguinte notação:

$$\{Y(t,\omega), t \in T, \omega \in \Omega\} \tag{1.1}$$

em que cada $t \in T, Y(t,\cdot)$ é uma variável aleatória no espaço Ω. Ou seja, para cada ponto no tempo existe um valor específico atraleado à ele, que juntos compõem o espaço Ω. Usando linguagem matemática, a série temporal observada é construída por uma sequencia de números:

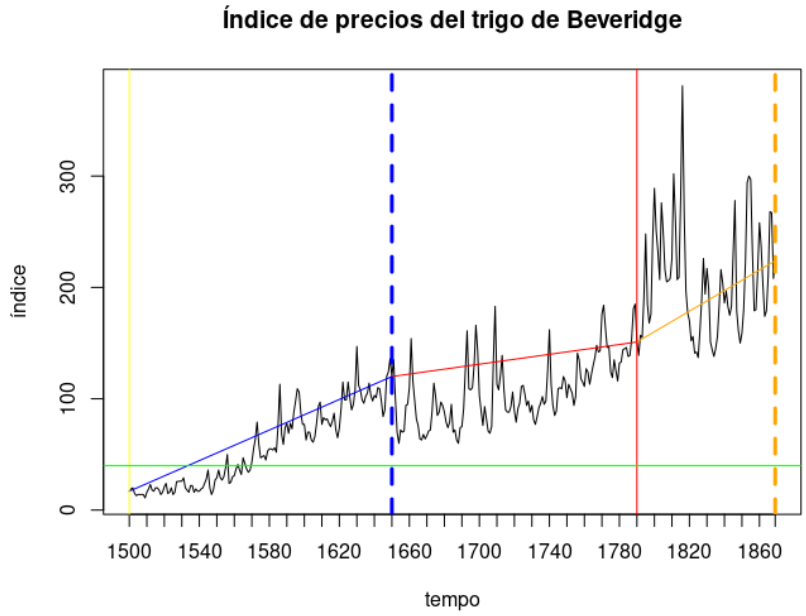
$$\{y\}_{t=1}^{T} = \{y_1, y_2, y_3, ..., y_T\} \tag{1.2}$$

Figura 1.1: Índice de Preços do Trigo Beveridge, 1500–1869



O exemplo da figura é uma serie com os preços do trigo[1], observe que a serie tem comportamentos diversos ao longo do tempo, como é possível ver na seguinte figura:

Figura 1.2: Índice de Preços do Trigo Beveridge, 1500–1869



A disposição de dados dessa forma traz diversas vantagens nos permitindo usar técnicas descritivas, modelos probabilísticos, métodos não parametricos, dentre outros para abordar esses valores. Na proxima seção vamos observar como podemos analizar cada tramo da serie de valores.

1.1 Decomposição Clássica

Quando estamos trabalhando com séries temporais, muitas de suas propriedade podem ser captadas assumindo-se a chamada decomposição clássica, que consiste:

$$Y_t = T_t + S_t + \epsilon_t \tag{1.3}$$

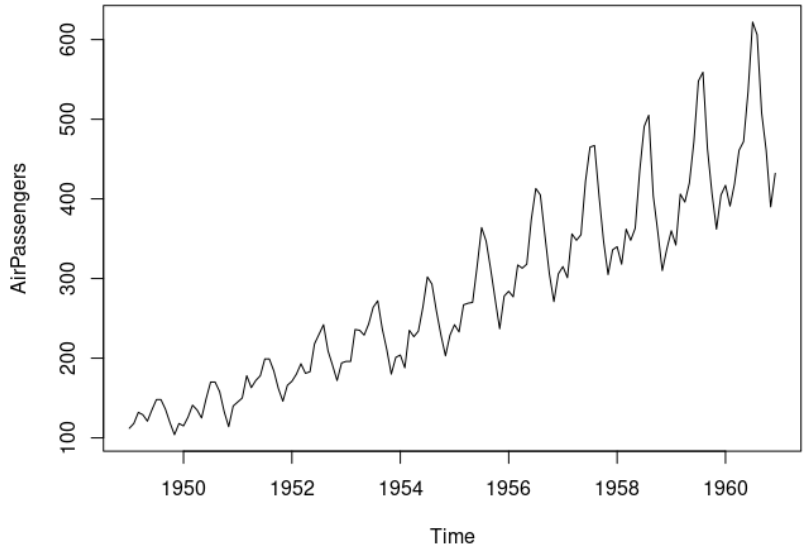
em que Y_t é uma variável estocástica, ou aleatória, T_t é o componente de tendência da série, S_t é o componente cíclico ou sazonal e ϵ_t é erro aleatório, ou seja, a parte não explicada do modelo (com média zero e variância constante).

Vamos ver com um exemplo, como decompor em cada uma das peças. O exemplo são os números mensais de passageiros de linha aérea, 1949-1960[2].

[1] A serie "bev"é parte do pacote *tseries* do R

[2] A serie "AirPassengers"é parte do pacote *tseries* do R

Figura 1.3: Números mensais de passageiros de linha aérea, 1949-1960



1.1.1 Sazonalidade

O componente cíclico em questão, se repete a cada intervalo de tempo fixo s, sendo assim:

$$... = S_{t-2s} = S_{t-s} = S_t = S_{t+s} = S_{t+2s} \tag{1.4}$$

Muitas séries apresentam um comportamento que tende a se repetir em intervalos de tempo equidistantes (a cada s período de tempo), como é o caso de funções trigonométricas.

Existem diversas formas de eliminar a sazonalidade dos dados, mas o mais simples e direto é captar os efeitos da sazonalidade por meio de *dummies*. De modo ilustrativo, suponha um conjunto de dados hipotéticos com frequência trimestrais, podemos fazer a seguinte regressão:

$$Y_t = \beta_0 + \beta_1 * D_2 + \beta_2 * D_3 + \beta_3 * D_4 + \epsilon_t \tag{1.5}$$

Isto também significa que todos os efeitos medidos por D_i serão relativos ao primeiro trimestre. Ainda podemos usar ϵ_t como a série dessazonalizada, ou seja:

$$Y_t^* = \beta_0 + \beta_1 * D_2 + \beta_2 * D_3 + \beta_3 * D_4 \tag{1.6}$$

$$\epsilon_t = Y_t - Y_t^* \tag{1.7}$$

1.1.2 Tendência

Por outro lado, a tendência é uma mudança de longo prazo no nível médio da série, ou, em termos formais:

$$Y_t = \beta_0 + \beta_1 t + \epsilon_t \tag{1.8}$$

sendo o nível médio da série no tempo t é dado por: $m = \beta_0 + \beta_1 t$. Note que o termo de tendência é uma função determinística no tempo.

De forma semelhante ao que fora feito para remoção da sazonalidade, estimamos a seguinte equação:

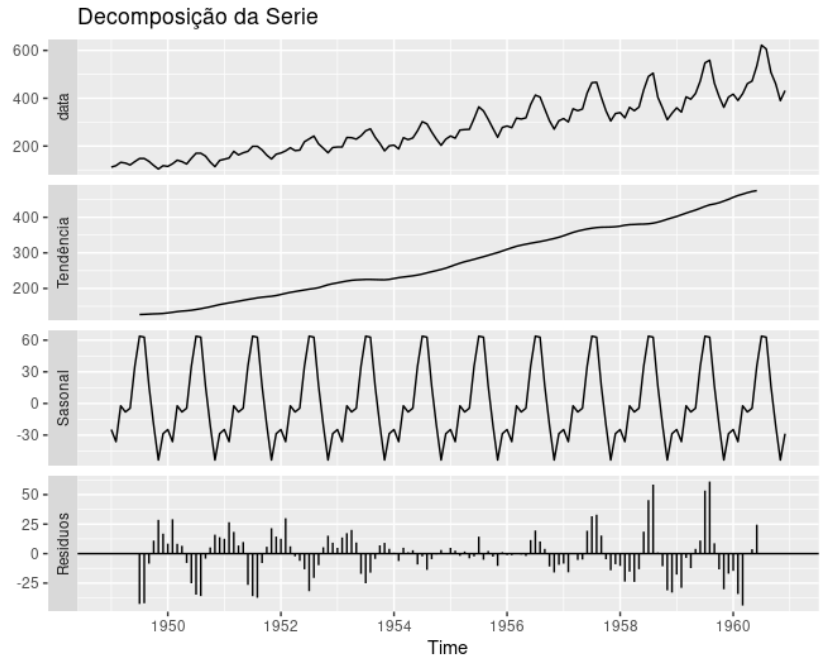
$$Y_t^* = \beta_0 + \beta_1 t \tag{1.9}$$

Chamaremos de residuos a diferença:

$$\epsilon_t = Y_t - Y_t^* \tag{1.10}$$

Assim sendo, temos a série original, menos a parte que capta os efeitos da tendência. Vejamos como fica o exemplo da linha aerea com esta decomposição.

Figura 1.4: Decomposição da série de números mensais de passageiros de linha aérea, 1949-1960



1.2 Estacionariedade

1.2.1 Processos Estocásticos Estacionários

O conceito de estacionariedade se divide em fracamente estacionário (ou apenas estacionário) e estritamente estacionário. O primeiro, perpassa por um processo estocástico se, e somente se este for:

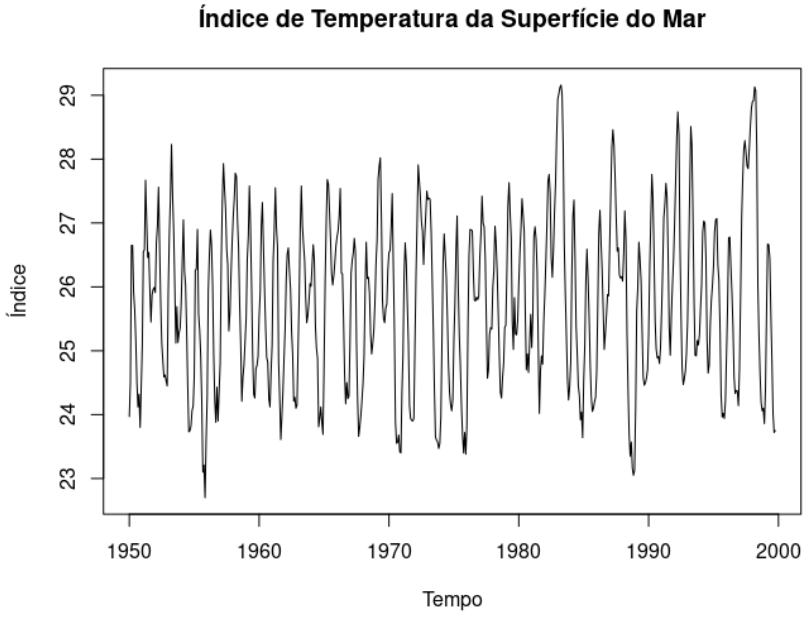
1. $EY(t) = \mu(t) = \mu, \forall t \in T$
2. $EY^2(t) < \infty, \forall t \in T$
3. $\gamma(t_1, t_2) = \text{Cov}(Y(t_1), Y(t_2))$ é uma função apenas de $|t_1 - t_2|$

Por outro lado, um processo estocástico é estritamente estacionário se todas as suas distribuições permanecem as mesmas sob translações no tempo.

Um exemplo de visualização de um processo estacionário pode ser visto na Figura 1.5 no Índice de Temperatura da Superfície do Mar[3], em que apesar haver flutuações nos valores observados, eles oscilam em volta de um valor fixo.

[3]A serie "nino3"é parte do pacote *tseries* do R

Figura 1.5: Índice de Temperatura da Superfície do Mar



1.2.2 Processos Estocásticos Não-Estacionários

Séries não estacionárias têm uma tendência que pode ter uma natureza determinística ou estocástica. A série não estacionária determinística, acrescida de um componente aleatório (erro ou ϵ extraído de uma distribuição normal, flutua em torno de uma tendência temporal.

$$y_t = c + \delta t + \epsilon_t \tag{1.11}$$

Por outro lado, a série com tendência estocástica move-se em torno de médias flutuantes.

$$y_t = c + y_{t-1} + \epsilon_t \tag{1.12}$$

Em outras palavras, a séries determinísticas, são aquelas cuja tendência de crescimento é sempre a mesma, ou seja, a taxa média de crescimento (ou decrescimento) dos valores tende a ser um valor fixo. Por outro lado, a série com tendência estocástica é aquele cuja média da taxa média de crescimento (ou decrescimento) varia, no exemplo da Equação (1.12) vemos que o valor de y_t depende do valor de y_{t-1}, não sendo, portanto, um valor fixo.

1.3 Tipos de Séries Temporais

1.3.1 Processos Auto-Regressivos Estacionários

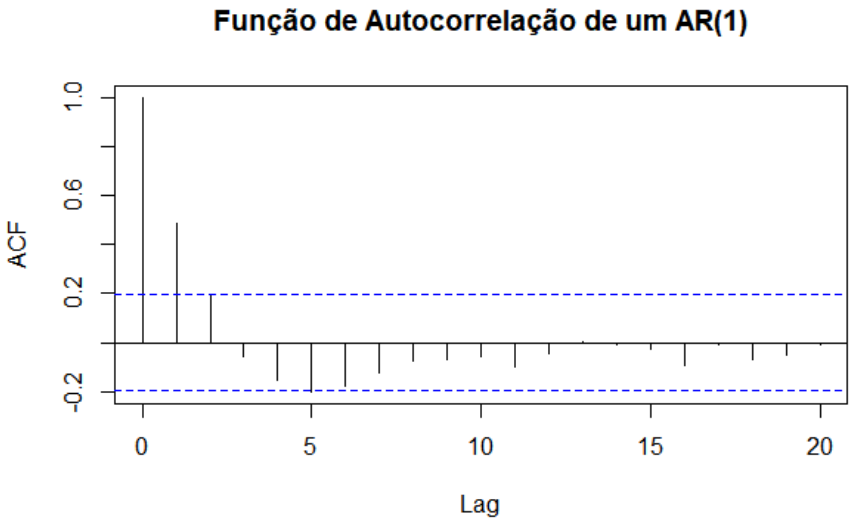
Um processo estocástico auto-regressivo pode ser modelado por um modelo auto-regressivo de ordem p. De acordo com esse modelo, uma série temporal y_t é descrita apenas por seus valores passados e por um erro normalmente distribuído, centrado em 0 com variância constante ϵ.

A versão mais simples de um modelo AR é aquela em que y_t depende somente de y_{t_1} e de ϵ_t. Diz-se, nesse caso, que o modelo é auto-regressivo de ordem 1, o que se indica abreviadamente por AR(1).

$$y_t = \phi y_{t-1} + \epsilon_t \tag{1.13}$$

Função de Autocorrelação

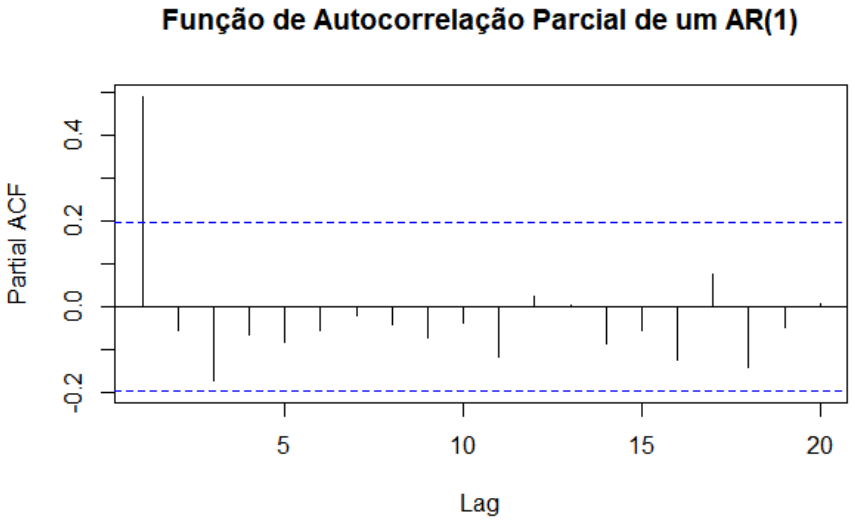
A função de autocorrelação é uma medida de correlação de duas observações distintas, separadas por s unidades de tempo (y_t, y_{t-s}). Ou seja, ela apresenta um padrão gráfico em que vemos em cada coluna a correlação entre a variável y_t e uma de suas demais observações defasadas no tempo.



Dito isso, vemos que a primeira coluna tem altura 1, ou seja a correlação de y_t e y_t é, obviamente, 1. Já a correlação entre y_t e y_{t-1} é de aproximadamente 0.5, enquanto a correlação de y_t e y_{t-2} é aproximandamente 0.2, e assim sucessivamente.

Função de Autocorrelação Parcial

Por outro lado, a função de autocorrelação parcial determina a ordem p do processo autoregressivo $(AR(p))$.



Como já enunciado, vemos que há significância (ou seja, a coluna em questão está acima do traceajado azul) na primeira *lag*, demonstrando que se trata de um processo AR(1), como já fora mencionado.

1.3.2 Processos Auto-Regressivos Não-Estacionários

Já vimos como funciona o processo auto-regressivo estacionário, dito isso, o processo não estacionário é aquele cuja média ou variância não serão constantes, ou seja, dependentes do tempo.

Dessa forma, para remover a tendência determinsítica, deve-se estimar y_t contra o tempo e armazenar os resíduos. Os resíduos armazenados constituem uma nova série que deverá ser modelada de forma separada. Por outro lado, para remover uma tendência estocástica, basta aplicar uma diferenciação na série.

Diante disso, chegamos ao conceito de processo estocástio integrado, que é aquele que se torna estacionário por meio da diferenciação. Dizemos que se um processo é integrado de ordem 1 (ou I(1)) quando esse se torna estacionário após retirarmos sua primeira diferença. De forma análoga, se um processo somente se tornar estacionário na d-ésima diferença, dizemos que é um processo integrado de ordem d, ou I(d), assim como um processo não diferenciado estacionário é de ordem 0.

Uma vez diferenciado e tornado estacionário, podemos utilizar os ferramentais apresentados na seção anterior.

1.3.3 Processo de Médias Móveis (MA)

Uma série temporal também pode ser explicada a partir de seus choques aleatórios contemporânos e passados. Um processo é dito de médias móveis $MA(q)$, em que q indica a defasagem mais elevada, podendo ser escrito como

$$y_t = \mu + \epsilon_t - \theta_1 \epsilon_{t-1} - ... - \theta_q \epsilon_{t-q} \tag{1.14}$$

1.4 Processo Auto-Regressivo e com Médias Móveis (ARMA)

Um processo autoregressivo e com médias móveis (ARMA(p,q)) é dito por uma série temporal mista, tal que:

$$y_t = c + \phi y_{t-1} + ... + \phi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - ... - \epsilon_q \epsilon_{t-q} \tag{1.15}$$

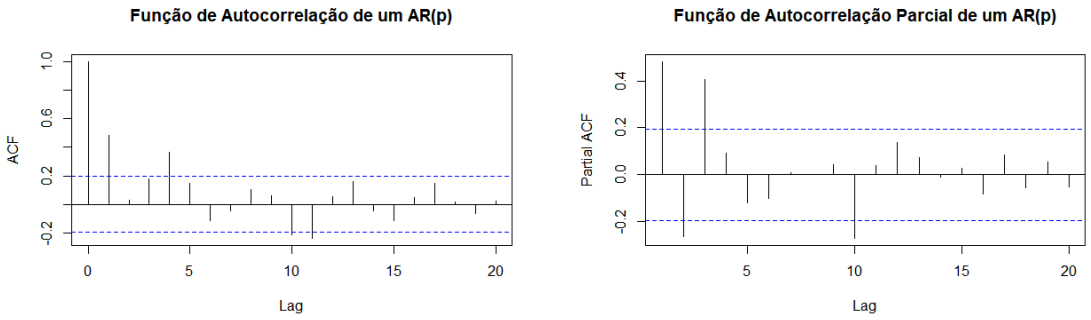
Dessa foma, como o nome já sugere, o processo ARMA possui características tanto de um processo auto-regressivo quanto de um processo de médias móveis. Dessa forma, para identificarmos esses processos dentro de uma série já dada, vamos utilizar os ferramentais gráficos da FAC e da FACP.

Identificação Usando as Funções de Autocorrelação

Como já vimos anteriormente, o processo AR(p) possui um comportamento gráfico específico nas funções de autocorrelação, sendo esse:

1. Decaimento Geométrico na Função de Autocorrelação (FAC)
2. Corte na defasagem p na Função de Autocorrelação Parcial (FACP)

Figura 1.6: FAC e FACP de um processo AR(p)

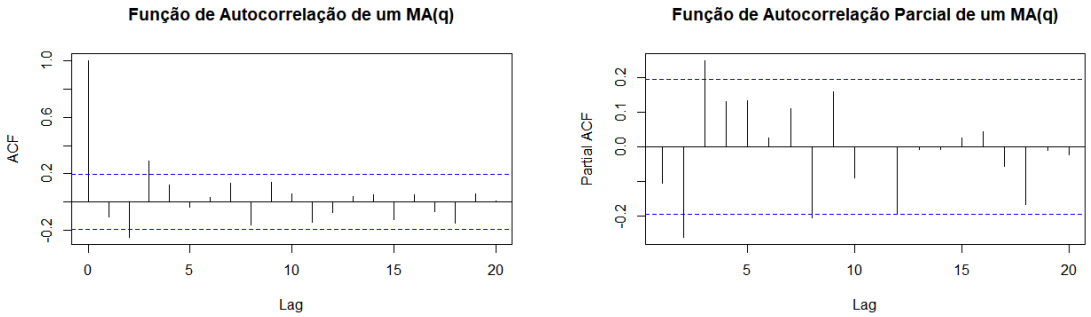


Pelo gráfico, há um decaimento geométrico na FAC, o que nos mostra os indícios de um processo autoregressivo, enquanto há um corte na *lag* 3 da FACP, o que nos mostra indícios de ser um AR(3).

Já o processo de média móvel, apresenta um comportamento oposto ao processo autoregressivo nos gráficos de FAC e FACP

1. Corte na defasagem q na Função de Autocorrelação (FAC)
2. Decaimento Geométrico na Função de Autocorrelação Parcial (FACP)

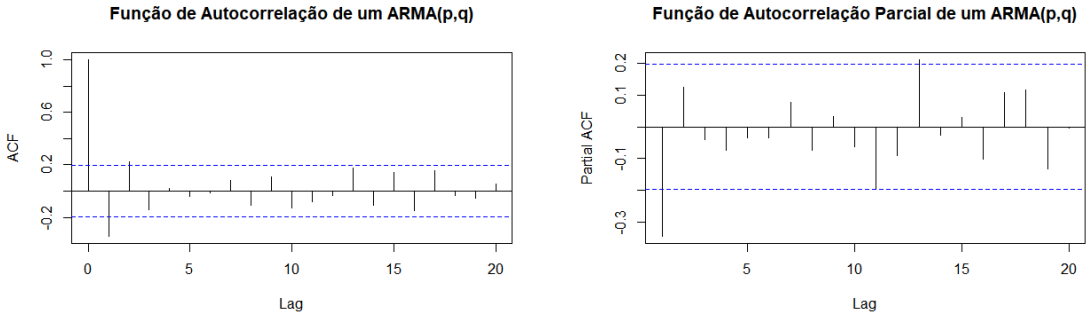
Figura 1.7: FAC e FACP de um processo MA(q)



Pelo gráfico, há um decaimento geométrico na FACP, o que nos mostra os indícios de uma média móvel, enquanto há um corte na *lag* 3 da FAC, o que nos mostra indícios de ser um MA(3). Importante ressaltar que os cortes em questão se trata de colunas abaixo do tracejado azul, ou seja, do nível de significância.

Contudo, quando estamos trabalhando com um processo ARMA(p,q), a identificação fica mais difícil, vejamos:

Figura 1.8: FAC e FACP de um processo ARMA(p,q)



Isso posto, em ambos os gráficos há decaimento geométrico e corte em *lags*, o que dificulta a idenfificação visual do processo em questão. Dessa forma, o ferramental apresentado serve apenas como um processo de identificação inicial. Além disso, utilizaremos outros ferramentais, como o Critério de Informação de Aike (AIC).

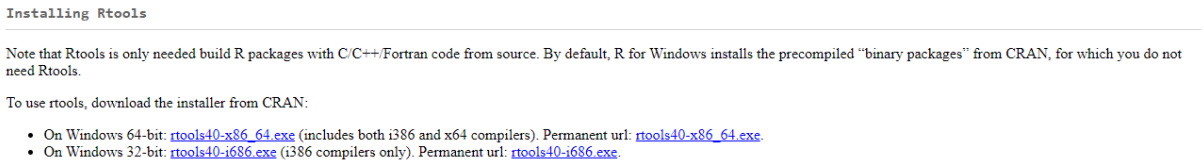
Capítulo 2

Utilizando o RCommander: downloads e dados

2.1 Como baixar o plugin

Primeiramente será necessário baixar o Rtools4 compatível com o seu dispositivo no site do Rtools no CRAN e instalá-lo no seu computador e ambiente R. Na sequência, baixe a versão mais recente do RcmdrPlugin.epack, disponível no repositório de pacotes do CRAN.

Figura 2.1: Locais para instalar o pacote



Installing Rtools

Note that Rtools is only needed build R packages with C/C++/Fortran code from source. By default, R for Windows installs the precompiled "binary packages" from CRAN, for which you do not need Rtools.

To use rtools, download the installer from CRAN:

- On Windows 64-bit: rtools40-x86_64.exe (includes both i386 and x64 compilers). Permanent url: rtools40-x86_64.exe.
- On Windows 32-bit: rtools40-i686.exe (i386 compilers only). Permanent url: rtools40-i686.exe.

No exemplo abaixo foi utilizada a versão *RcmdrPlugin.epack_1.2.5.tar.gz*. Por fim, é preciso rodar as seguintes linhas de código no RStudio:

```
options("install.lock"= FALSE)
install.packages("fracdiff")
install.packages("forecast")
```

E se o arquivo estiver no diretório:

```
install.packages("RcmdrPlugin.epack_1.2.5.tar.gz", repos=NULL, type = "source")
```

Ou, caso contrário se não estiver dando certo de primeira, substituir *RcmdrPlugin.epack_1.2.5.tar.gz* pelo caminho para o arquivo no seu dispositivo.

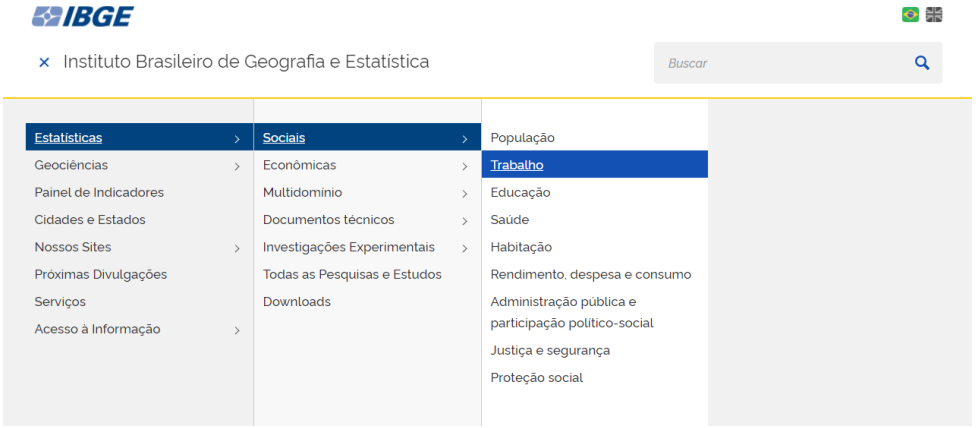
2.2 Como escolher e baixar o banco de dados

Para fazermos nossas análises, vamos utilizar de um banco de dados licenciado e gratuito, disponibilizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE). Primeiro, devemos clicar na caixinha logo abaixo e seremos direcionados para o site do IBGE.

IBGE

Depois, seguimos os "três tracinhos", selecionamos "Estatísticas" > "Sociais" > "Trabalho".

Figura 2.2: Site do IBGE. Caminho até os dados separados por tipo



Depois de seguirmos esses passos, selecionamos o tema de "PNAD Contínua - Pesquisa Nacional por Amostra de Domicílios Contínua".

Figura 2.3: Seção de PNAD Contínua

A partir daqui devemos selecionar os "Downloads"e procurar a pasta "Anual" > "Caracteristicas_Adicionais_do_Mercado_de_Trabalho_2012_2017". Agora, baixamos os dados do tipo "xls".

Figura 2.4: Pacotes de dados relacionados ao PNAD

O documento ficará salvo na sua pasta de Downloads!

2.3 Como importar o banco de dados no RCommander

Utilizaremos o comando *read_excel()* do pacote **readxl**. Para isso, é preciso saber como copiar o caminho de um arquivo:

Como copiar o caminho de um arquivo:

1. Vá ao local do arquivo em seu dispositivo.
2. Clique no arquivo desejado com o botão direito do mouse e, ao mesmo tempo, o botão "Shift".
3. Por fim, clique em *Copiar como caminho* (esta opção não aparecerá se o passo anterior não for feito corretamente).

Agora, no R, siga conforme o exemplo abaixo:

Figura 2.5: Código utilizado para carregar e formatar os dados para a modelagem no capítulo 3

```
install.package("readxl")
install.package("tidyverse")

library(readxl)
library(tidyverse)
dados <- read_excel(path = "C:Users/Eu/Downloads/nome_do_arquivo.xls",
                    sheet = 2,
                    range = "H8:M9")
dados <- gather(data = dados,
                key = Ano,
                value = Observação)
```

Usando a função, salvamos o banco de dados no objeto "dados1". Para entendermos melhor o que foi feito no exemplo, vale à pena explicar o que é esta linha de comando antes de aplicar a função *read_excel()* e quais são os 3 argumentos usados junto, bem como a função *gather()* e quais são os 3 argumentos usados junto.

library()

Para utilizar a função *read_excel()* e a função *gather()*, é necessário antes carregar os pacotes nas quais elas estão presentes, readxl e tidyverse, respectivamente. Lembre-se que ante disso é necessário ter baixado o pacote com o comando *install.packages()*.

read_excel()

Esta função que extrairá os dados específicos desejados do banco de dados baixado. No exemplo utilizamos 3 argumentos. Estes são:

path: é o caminho até o seu arquivo no seu dispositivo. O caminho deve ser escrito entre aspas e é necessário que todas as barras sejam do tipo "*forward slash*"ou "/", ao invés deste tipo "*backslash*"ou "\". Obs.: É necessário trocar o valor presente no exemplo para o valor/caminho que você copiou no seu dispositivo conforme o passo anterior.

sheet: é a ordem da folha que vai ser lida para a extração dos dados. Ou seja, se vamos extrair dados da n-ésima folha do conjunto de planilhas, colocamos "n"neste argumento da função. No nosso caso de banco de dados, vamos extrair os dados da segunda folha do arquivo, portanto informaremos o valor "2".

range: é o intevalo da planilha na qual serão retirados os nomes das variáveis e seus valores. Neste caso são 2 linhas entre as colunas H e M. Desta forma, leremos os valores dos anos e dos valores das observações de "Pessoas ocupadas ou que anteriormente já foram ocupadas (mil pessoas)".

gather()

Apesar de termos carregado o banco de dados no recorte desejado, ele não está no formato desejado (após carregar com *read_excel()*. No caso, temos um banco de dados com 6 variáveis de 1 observação cada, mas na verdade, desejamos obter um banco com 2 variáveis de 6 observações cada), sendo 1 das variáveis obtidas pelo nome de cada variável do banco de dados extraído. Para corrigir isso, utilizaremos a função *gather()*. Abaixo temos a explicação dos 3 argumentos utilizados:

data: é informado o banco de dados a ser alterado. Neste, foi informado "dados1".

key: é informado o nome da primeira variável do novo banco de dados. Seus valores serão retirados do nome de cada variável de "dados1".

value: é informado o nome da segunda variável do novo banco de dados. Seus valores serão retirados da única observação de cada variável de "dados1".

Aplicando a função *gather()*, salvamos o banco de dados transformado por cima do anterior no mesmo objeto "dados1". Na sequência faremos modelagens de séries temporais para este banco de dados e para outros bancos de dados quando necessário.

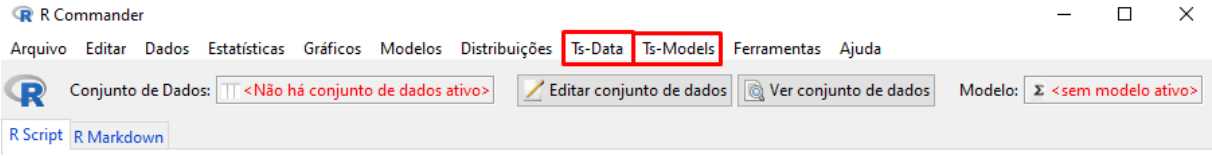
Capítulo 3

Utilizando o RCommander: a interface e modelagens do plugin

O plugin tem algumas funções desatualizadas, ou seja, nem todas as funções da interface funcionam corretamente e, portanto, estas não serão explicadas mais a fundo nos tópicos abaixo.

Existem duas partes do plugin (Ts-Data e Ts-Models) que estarão na interface do RCommander conforme a Figura 3.1 e serão explicadas separadamente a seguir.

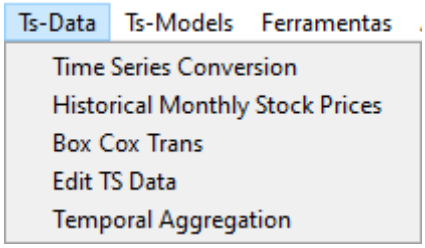
Figura 3.1: RCommander com destaque às duas opções adicionadas pelo plugin epack



3.1 Ts-Data

A sessão Ts-Data tem funções que fazem alterações no banco de dados. Clicando em Ts-Data no RCommander, ficará visível que há 5 funções diferentes (vide imagem 3.2), que serão explicadas a seguir.

Figura 3.2: Opções de Ts-Data no RCommander

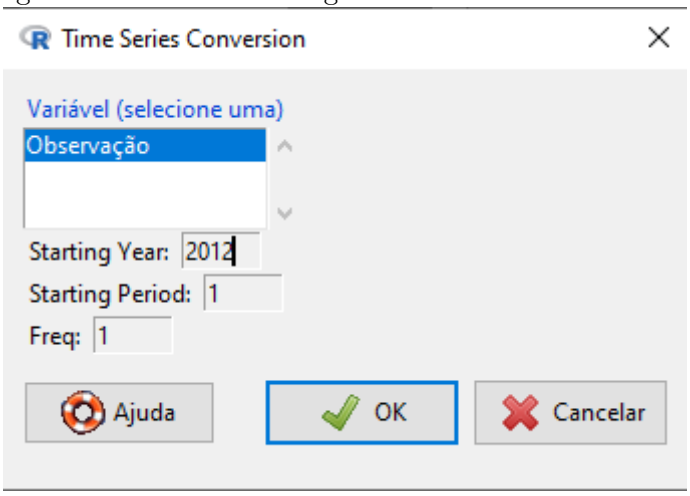


3.1.1 Time Series Conversion

Esta função tem como objetivo adequar o nosso banco de dados para ser uma série temporal. Para exemplificar, será utilizado o banco de dados que foi ensinado a baixar no capítulo 2.3:

Após carregar o banco de dados e seguir o passo a passo, é importante verificar se o banco de dados está carregado no RCommander, e depois, clicar para executar a função *Time Series Conversion*. Isso abrirá a caixa de diálogo abaixo.

Figura 3.3: Caixa de diálogo de *Time Series Conversion*



Na caixa de diálogo selecione a variável de interesse, informe o ano inicial, a periodicidade e a frequencia anual das observações, que neste caso são, respectivamente, a única variável de interesse "Observação", 2012, 1 e 1, já que há apenas uma observação por ano. Agora, pode-se confirmar a mudança do banco de dados, conforme visto abaixo.

Figura 3.4: Banco de dados após a modificação no plugin do RCommander

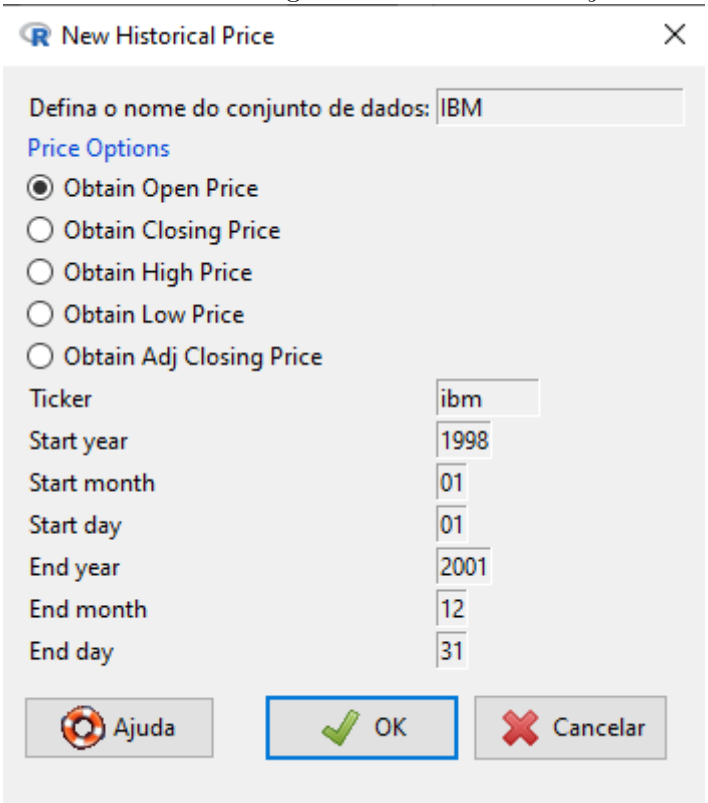
```
> dados1$Observação
Time Series:
Start = 2012
End = 2017
Frequency = 1
[1] 131458.9 133679.7 135885.7 139374.1 139074.0 142187.8
```

3.1.2 Historical Monthly Stock Prices

Esta função tem como objetivo montar um banco de dados valores **mensais**[1] de preços de ações carregando dados da bolsa de valores dos Estados Unidos. Na figura 3.5, está a caixa de diálogo gerada por essa função.

[1]Não ficou claro como funciona a escolha do valor. Ou seja, se a função apenas retira valores mensais, qual dos 30, 31 ou 28 dias que foi escolhido? Foi escolhido um valor médio? Por quê/para quê informor dia de início da série se o valor é mensal? Pode-se até ter hipóteses que respondam essas perguntas, mas não ficou claro para os redatores desta apostila as respostas disso com toda certeza.

Figura 3.5: Caixa de diálogo de *Historical Monthly Stock Prices*



Nesta caixa de diálogo, o usuário informa o nome desejado para o banco de dados em "Defina o nome do conjunto de dados:". Em "Price Options", é escolhido entre obter o preço de abertura, obter o preço de fechamento, obter o preço máximo, obter o preço mínimo e obter o preço anterior ao preço de fechamento, respectivamente referenciadas na caixa de diálogo por "Obtain Opening Price", "Obtain Closing Price", "Obtain High Price", "Obtain Low Price"e "Obtain Adj Closing Price". Abaixo desta parte pede-se para informar o símbolo de ações, ano, mês e dia de início da série e ano, mês e dia de final da série a ser obtida, respectivamente referenciadas na caixa de diálogo por "Ticker", "Start year", "Start month", "Start day", "End year", "End month"e "End day".

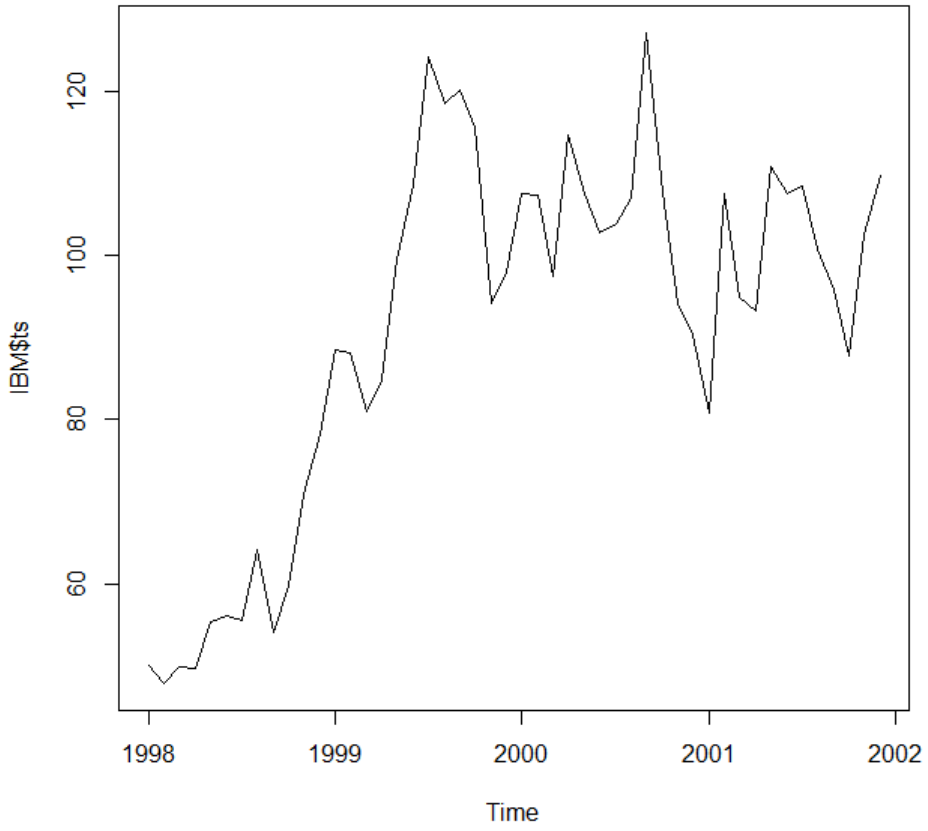
No exemplo acima, foi pedido os valores de abertura de 01/01/1998 a 31/12/2001 para a empresa de símbolo de ações "ibm"[2], gerando o banco abaixo:

[2]Não ficou claro como deve ser informado para mudar a empresa na qual será obtida os dados. Foi-se utilizado "ibm"pois era o valor *default*. Desta forma, não está ensinado aqui como modificar o valor de "Ticker".

Figura 3.6: Banco de dados gerado pela função *Historical Monthly Stock Prices*

```
> IBM
          ts
1   49.95220
2   47.80115
3   49.80282
4   49.65344
5   55.41946
6   56.10660
7   55.44933
8   64.05353
9   54.01530
10  59.78131
11  70.95483
12  78.15488
13  88.43212
14  88.19312
15  81.02294
16  84.75741
17  99.42638
18 108.62811
19 124.28298
20 118.54684
21 120.10038
22 115.67878
23  94.16826
24  98.05210
25 107.49283
26 107.43308
27  97.51434
28 114.72276
29 107.55258
30 102.83221
31 103.96750
32 107.07457
33 127.21080
34 107.67208
35  94.10850
36  90.28442
37  80.78394
38 107.45698
39  94.93308
40  93.16444
41 110.75526
42 107.55258
43 108.50861
44 100.71702
45  95.74570
46  87.68642
47 102.53346
48 109.75143
```

Figura 3.7: Gráfico com os valores de IBM gerados pela função *Historical Monthly Stock Prices*



3.1.3 Box Cox Trans

Esta função está desatualizada e portanto não há funcionalidade.

3.1.4 Edit TS Data

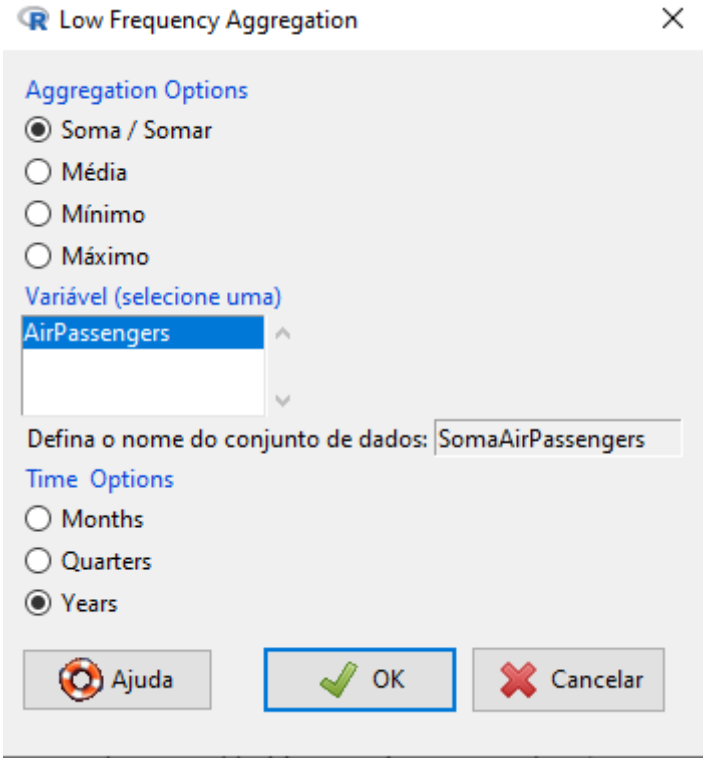
Esta função está desatualizada e portanto não há funcionalidade.

3.1.5 Temporal Aggregation

Esta função tem como objetivo compilação de informações de bancos de dados com a intenção de preparar conjuntos de dados combinados para processamento de dados. Ou seja, há como objetivo gerar um novo banco de dados com a soma, média, valor mínimo ou valor máximo para um período de tempo de mês, de trismetre ou de ano. Abaixo temos a caixa de diálogo utilizando o banco de dados *AirPassengers*[3].

[3]Por mais que o banco de dados seja carregado pela função *data()*, é necessário transformá-lo em um dataframe (por exemplo utilizando o código *AirPassengers<-data.frame(AirPassengers)*), pois o RCommander não reconhecerá o conjunto de dados caso ele não seja um dataframe.

Figura 3.8: Caixa de diálogo de *Temporal Aggregation*



O banco de dados escolhido contém informações mensais por 12 anos e no exemplo acima estamos combinando as informações por ano somando-as, ou seja, deseja-se fazer uma nova variável na qual tenha-se 12 observações da soma dos valores dos meses de cada ano. Como resultado temos o banco de dados abaixo salvo conforme informado na caixa de diálogo.

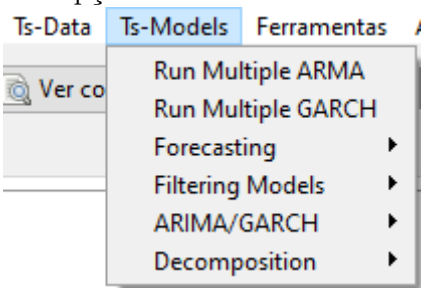
Figura 3.9: Banco de dados gerado pela função *Temporal Aggregation*

```
> SomaAirPassengers
      x
1  1520
2  1676
3  2042
4  2364
5  2700
6  2867
7  3408
8  3939
9  4421
10 4572
11 5140
12 5714
```

3.2 Ts-Models

A sessão Ts-Models tem funções que fazem modelagens estatísticas de previsão. Clicando em Ts-Models no RCommander, ficará visível que há 6 funções diferentes (vide Figura 3.10), porém apenas 3 delas serão explicadas a seguir por causa da não atualização do plugin ou da não explicação do conceito nesta apostila.

Figura 3.10: Opções de Ts-Models no RCommander



3.2.1 Run Multiple ARMA

Esta função tem como objetivo encontrar os valores corretos dos parâmetros para um modelo ARIMA. Utilizando o banco de dados que foi ensinado a baixar no capítulo 2.3, temos a caixa de diálogo abaixo em que seleciona a variável temporal de interesse gerando a saída que está na sequência.

Figura 3.11: Caixa de diálogo de *Run Multiple ARMA*

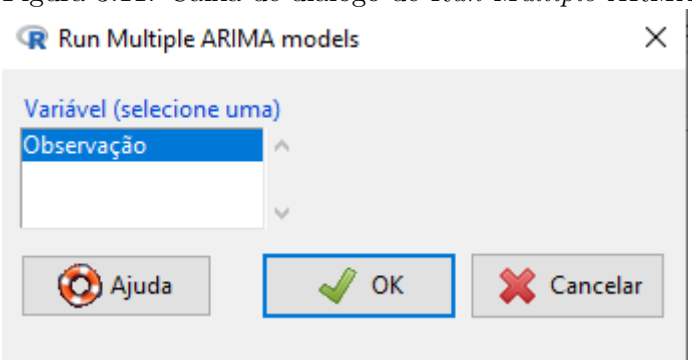


Figura 3.12: Saída da função *Run Multiple ARMA*

```
Rcmdr>  bulkfit(dados1$Observação)
$res
      ar d ma         AIC
 [1,]  0 0  0   119.44706
 [2,]  0 0  1   119.68894
 [3,]  0 0  2   119.03581
 [4,]  0 1  0    94.50970
 [5,]  0 1  1    96.17589
 [6,]  0 1  2    95.96874
 [7,]  0 2  0    76.34715
 [8,]  0 2  1    75.34541
 [9,]  0 2  2    76.04979
[10,]  1 0  0   117.96761
[11,]  1 0  1   119.68193
[12,]  1 0  2   119.65291
[13,]  1 1  0    95.67239
[14,]  1 1  1   102.86080
[15,]  1 1  2    96.17760
[16,]  0 0  0 99999.00000
[17,]  1 2  1    76.10482
[18,]  1 2  2    78.03443
[19,]  0 0  0 99999.00000
[20,]  2 0  1   126.25347
[21,]  2 0  2   118.56200
[22,]  0 0  0 99999.00000
[23,]  2 1  1    96.20620
[24,]  2 1  2    98.15957
[25,]  2 2  0    76.11776
[26,]  2 2  1    78.07423
[27,]  2 2  2    80.03143

$min
      ar        d       ma      AIC
 0.00000  2.00000  1.00000 75.34541
```

Conforme pode ser visto acima, o modelo que minimiza o valor do AIC é o ARIMA com parâmetros ar = 0; d ou i = 2; ma = 1.

3.2.2 Run Multiple GARCH

Se um modelo auto-regressivo de médias móveis (ARMA) for assumido para a variância do erro, tem-se um modelo de heteroscedasticidade condicional auto-regressiva generalizada (GARCH). Como não houve um desenvolvimento nesta apostila sobre este tipo de modelagem, não será aprofundado sobre o funcionamento desta função do plugin.

3.2.3 Forecasting

Apesar de as funções da parte *ARIMA/GARCH* estarem funcionando, o RCommander não tem conseguido reconhecer a presença dos modelos, gerando a mensagem de erro:

RcmdrMsg: [1] ERRO: Não há modelos para escolher

Desta forma não tem como utilizar esta função.

3.2.4 Filtering Models

A desatualização do plugin fica bem clara aqui, pois não é possível utilizar nenhuma das funções presentes nesta sessão pois o RCommander gera a mensagem:

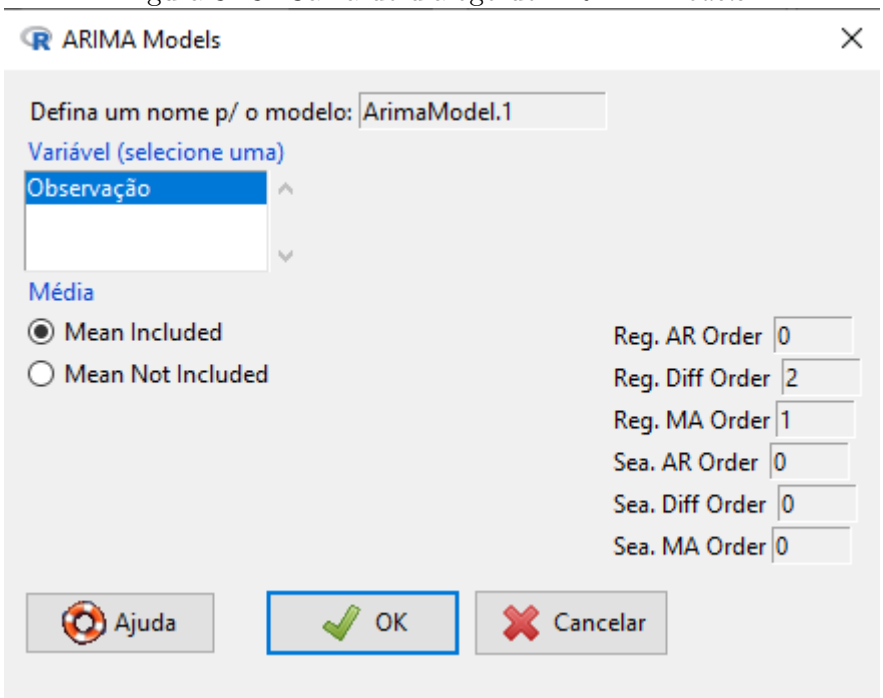
RcmdrMsg: [2] ERRO: O modelos não está mais disponível

Desta forma não tem como utilizar esta função.

3.2.5 ARIMA/GARCH

Esta função gera o modelo ARIMA/GARCH para a variável temporal de interesse com a especificação dos parâmetros pelo usuário. Abaixo temos a caixa de diálogo para gerar um modelo ARIMA. O nome do modelo é gerado automáticamente mas pode ser modificado pelo usuário. Utilizando o banco de dados da PNAD já usado em exemplos anteriores e transformado por *Time Series Conversion* em Ts-Data[4], nós selecionamos um modelo com a variável temporal "Observação", incluindo a média e sendo uma modelagem ARIMA(0,2,1), ou seja com parâmetros ar = 0; i ou d = 2; ma = 1, conforme foi calculado na função *Run Multiple ARMA* em Ts-Models. Note que, como não é um modelo SARIMA, ou seja não é um modelo sazonal, a parte "Sea."tem valores iguais a 0.

Figura 3.13: Caixa de diálogo de *ARIMA Models...*



A saída do modelo gerado está abaixo.

[4]É essencial a transformação gerada por *Time Series Conversion* em Ts-Data, pois sem uma variável *temporal* com a informação do período, a função não funciona. Alguns bancos de dados como o *AirPassengers* já estão convertidos para uma série temporal e portanto não precisariam deste ajuste.

Figura 3.14: Saída do modelo ARIMA

```
> ArimaModel.1
Series: dados1$Observação
ARIMA(0,2,1)

Coefficients:
          ma1
      -1.0000
s.e.   0.6251

sigma^2 estimated as 2921362:  log likelihood=-35.67
AIC=75.35   AICc=87.35   BIC=74.12
```

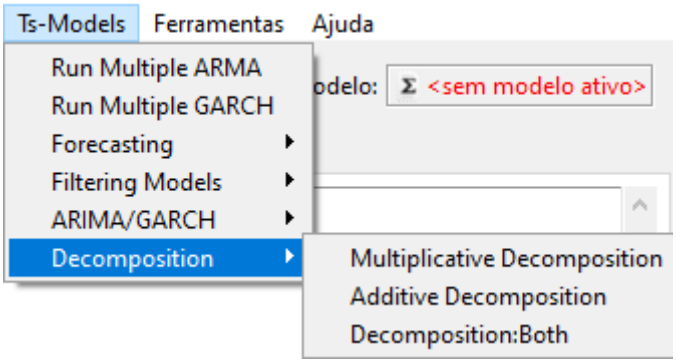
Não será explicada a modelagem de GARCH pelo mesmo motivo supracitado na subsessão *Run Multiple GARCH*.

3.2.6 Decomposition

Esta função teria como objetivo fazer a decomposição da série temporal, porém, na prática, ela faz a previsão (*forecasting*) para um modelo com decomposição. Para exemplificar esta função, será utilizado o banco de dados *AirPassengers*[5] que no Capítulo 1.1 foi demonstrado o que é uma decomposição aditiva.

Primeiramente, precisa-se clicar na opção "Additive Decomposition"que fica dentro da interface "*Decomposition*"e pode ser acessada ao passar o cursor do mouse sobre esta interface conforme a imagem abaixo.

Figura 3.15: Opções de *Decomposition* dentro de Ts-Models no RCommander



Na caixa de diálogo abaixo, foi selecionada a única variável do banco de dados e escolhido o número de passos de previsões a serem realizadas em "Number of Forecasts:"[6]. O resultado da função está na imagem 3.17.

[5]Por mais que o banco de dados seja carregado pela função *data()*, é necessário transformá-lo em um dataframe (por exemplo utilizando o código *AirPassengers<-data.frame(AirPassengers)*), pois o RCommander não reconhecerá o conjunto de dados caso ele não seja um dataframe.

[6]Há também o valor "Freq"sendo pedido que no caso os redatores desta apostila não conseguiram identificar a sua funcionalidade, visto que em diferentes testes modificando o valor de "Freq"não foi obtida mudanças significativas nos valores.

Figura 3.16: Caixa de diálogo de *Additive Decomposition*

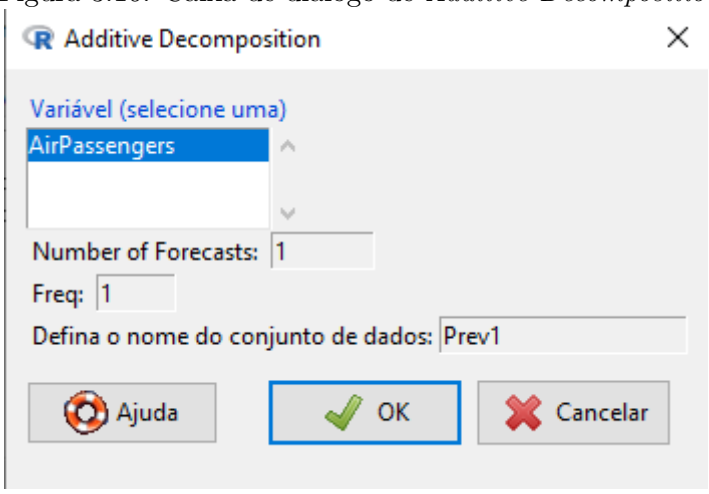


Figura 3.17: Saída de *Additive Decomposition* para o banco AirPassengers e pedindo previsões 1 passo à frente

```
Rcmdr>  Prev1<-decom6(AirPassengers,se1=1,fore1=1)
(Intercept)          t1
  87.652778    2.657184
      pred    lower    upper
1 472.9444 380.6292 565.2597
```

A saída acima informa o valor predito para 1 passo à frente em "pred"e o intervalo de confiança para esta predição em "lower"e "upper". Além disso há os valores da modelagem da reta de tendência, com o intercepto, neste caso, igual a 87.652778 e a inclinação de 2.657184.

Na próxima figura, há a saída caso fossem pedidos 20 previsões em "Number of Forecasts:".

Figura 3.18: Saída de *Additive Decomposition* para o banco AirPassengers e e pedindo previsões de 1 a 20 passos à frente

```
Rcmdr>  Prev2<-decom6(AirPassengers,se1=1,fore1=20)
(Intercept)          t1
  87.652778    2.657184
       pred    lower    upper
1  472.9444 380.6292 565.2597
2  475.6016 383.2600 567.9432
3  478.2588 385.8905 570.6271
4  480.9160 388.5207 573.3113
5  483.5732 391.1505 575.9959
6  486.2304 393.7799 578.6808
7  488.8875 396.4090 581.3661
8  491.5447 399.0377 584.0518
9  494.2019 401.6661 586.7378
10 496.8591 404.2941 589.4241
11 499.5163 406.9218 592.1108
12 502.1735 409.5491 594.7978
13 504.8307 412.1761 597.4852
14 507.4878 414.8027 600.1730
15 510.1450 417.4290 602.8611
16 512.8022 420.0549 605.5495
17 515.4594 422.6805 608.2383
18 518.1166 425.3057 610.9274
19 520.7738 427.9306 613.6169
20 523.4309 430.5551 616.3067
```