FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

**Ano Letivo 2017 / 2018**

# Análise e Transformação de Dados

## Relatório – Mini Projeto

## Discentes

André Correia – 2016210208 – Andre.Correia@student.uc.pt – Engenharia Informática

André Leite – 2015250489 – uc2015250489@student.uc.pt – Engenharia Informática

# Índice

# Nota Prévia

A análise e transformação de dados tem vindo a desempenhar, no seio industrial e não só, um papel fulcral no que toca a aspeto de previsão e manutenção de qualidade e noutros aspetos mais consideráveis.

Por muitas vezes recebemos um conjunto de dados que não apresentam muita coerência, ou porque existem falhas nos dados, tais como “NAN’s”, ou dados que se dispersam da realidade traçada pela série. Nestes casos é importante proceder à manutenção deste mesmo conjunto de dados, para melhor clarificar a sua significância e para então, poder estudar e esmiuçar toda a informação possível através deste mesmo conjunto.

Neste trabalho prático foi aglomerado todo o conhecimento adquirido para efetuar esta análise do dataset que foi atribuído.

Bem como referido, era pretendida uma análise da série temporal associada ao dataset e efetuar o seu pré-processamento, ou seja, corrigir as imperfeições na série de modo a deixá-la o mais coerente possível para proceder aos estudos. Era também necessário decompor a série em componentes que salientassem os movimentos erráticos e estruturais da mesma. E finalmente, identificar o modelo que melhor se ajusta ao crescimento da série bem como testar este modelo para eventos futuros, previstos pelo estudo da mesma série, já regularizada.

# 1ª Parte (Análise da série e pré-processamento)

*<u>Função para obtenção de dados:</u>*

```
function [dados,tempo] = getData()

    disp('PROJETO - Análise e Transformação de Dados - 2018');
    fprintf('\nNumero da sua PL (pratica laboratorial)\n');
    PL = input( 'PL: ' , 's');
    fich = "dataset_ATD_PL"+PL+".csv";
    disp(fich);

    allData = load(fich); %load dos dados do dataset (nao pode ter cabeçalho nas tabelas)
    time = allData(1:end,1); %obtenção de dados na primeira coluna
    data = allData(1:end,2); %obtenção de dados na segunda coluna
    dados = data;
    tempo = time;
    treatData(dados); %passagem de dados para tratamento em função

end
```

A função **getData** é responsável pela obtenção dos dados no ficheiro atribuído a cada turma PL. A informação é adquirida pela variável **allData** que utiliza a função **load**, que faz o carregamento dos dados para a variável.

Por sua vez, como o ficheiro Excel é composto por duas colunas, é necessário carregar cada uma delas para variáveis diferentes. Desta forma, carregamos os dados referentes ao **consumo elétrico diário** em hw/h para a variável **data** e o mesmo é feito para a variável **time**, que recebe o **registo dos dias**.

Esta função faz o retorno destas variáveis para uma outra função, que irá proceder ao tratamento dos dados.

*<u>Função para tratamento de dados:</u>*

```
function treatData(x1)

    N=length(x1); %comprimento da serie temporais
    t=(0:N-1)'; %escala temporal
```

```
x1r=x1; %atribuição a variavel temporária
```

Esta será a função que vai realizar todas operações necessárias relativas a cada alínea deste projeto. Cada secção está dividida por um comentário que contem o número referente à mesma alínea.

**1.1** :

```
N=length(x1); %comprimento da serie temporais
t=(0:N-1)'; %escala temporal
x1r=x1; %atribuição a variavel temporária
figure(22)
plot(x1r);
```

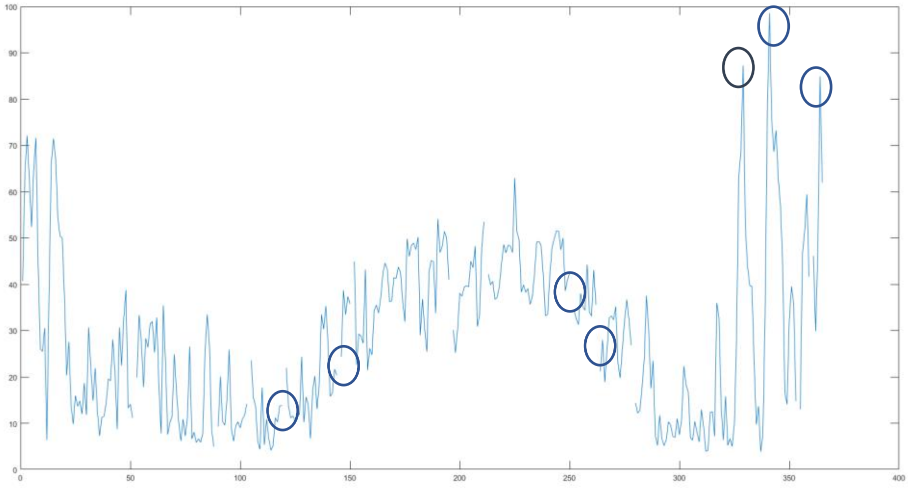



Figura 1 – Série temporal não regularizada

Nesta questão era meramente pedido para apresentar graficamente a série temporal a utilizar no projeto. É claramente visível que vão ser necessários retirar os Nan's e ajustar os outliers ao desenvolvimento da série.

**1.2, 1.3 :**

```
if any(isnan(x1)) %pesquisa de valores NAN
        ind=find(isnan(x1)); %indices de NAN em vetor
        for k=1:length(ind)
            tt=t(ind(k)-2:ind(k)-1);
            xx=x1r(ind(k)-2:ind(k)-1);
            x1r(ind(k))=interp1(tt,xx,t(ind(k)),'pchip','extrap');
%interpolação com método pchip para substituição de valores NAN
        end
```

```
end

x1n = x1r;

media = mean(x1r); %calculo da media
padrao = std(x1r); %calculo do desvio padrao
max = media + (1*padrao); %valor para secção mensal (+desvio)
min = media - (1*padrao); %valor para secção mensal (-desvio)
disp(max);

for i=1:length(x1r) %check de outliers
    if x1r(i)>max %para valores acima da média + desvio padrão
        x1r(i)= media+(2.5*padrao); %resolução de outliers (LOW)
    elseif x1r<min %para valores abaixo da média - desvio padrão
        x1r(i)= media-(2.5*padrao); %resolução de outliers (HIGH)
    end
end
```

Neste excerto de código, que agora iremos abordar e explicar, podemos verificar que são feitas as verificações para existência de valores não existentes na série e a posterior extrapolação  e de seguida o ajuste de outliers, que iremos agora explicar.

A forma como abordamos o problema da falta de valores na série é primeiro verificando realmente de temos mesmo valores não preenchido e nesse caso inserimos num vetor defenido como ind as posições nas quais há valores ausentes. Desta forma podemos efetuar a **extrapolação** , nesse indice do vetor ind, no vetor **x1r** ,que irá ser a nossa série regularizada, com o método "**pchip**". Sendo uma forma eficaz de estimar valores inexistentes , dado o histórico da série.

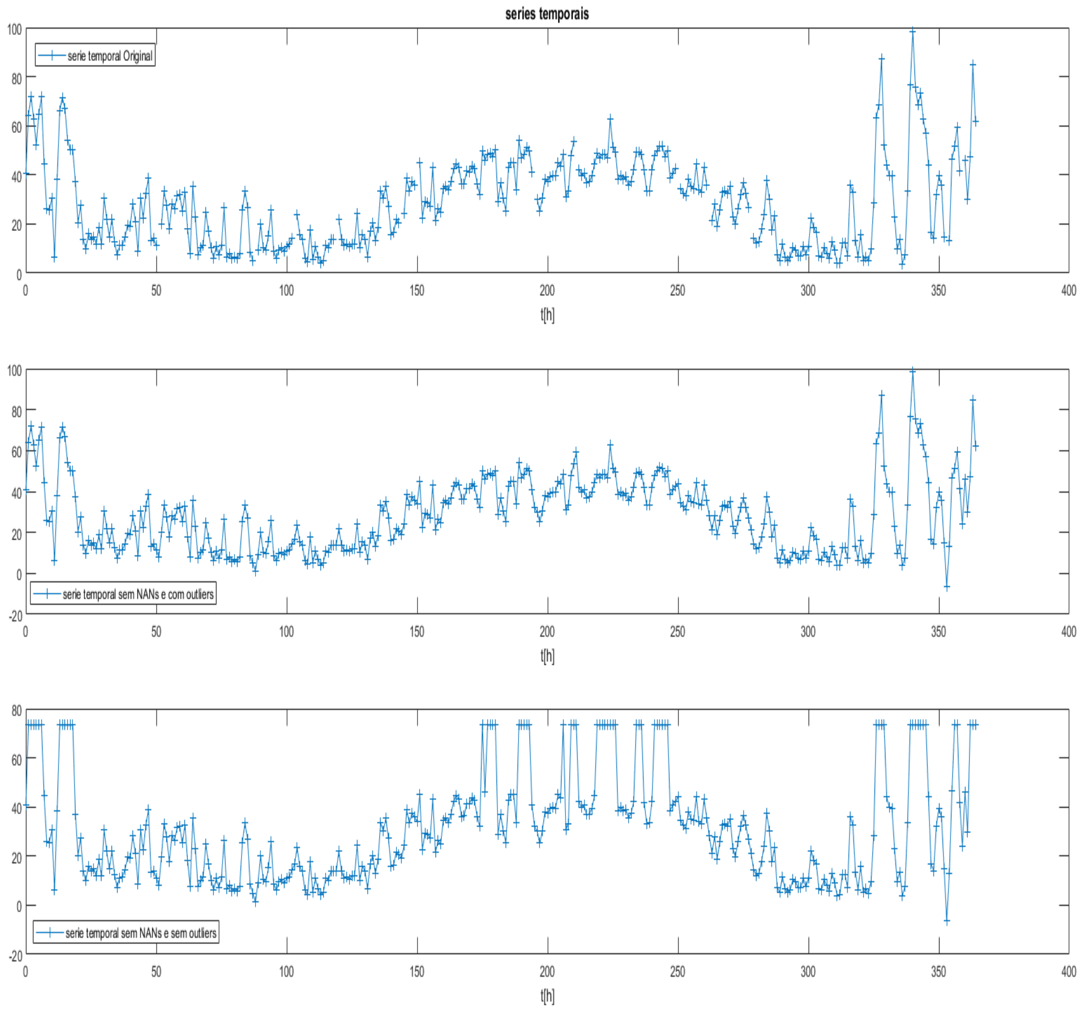
Em seguida é necessário também verificar a existência de outliers, numa primeira fase é necessário o cálculo dá média e do desvio padrão. Estes cálculos são concebidos utilizandos duas funções , **mean()**  e **std(**), calcunado a média e o desvio padrão respetivamente. Os resultados gerados por estas funções em prol , sempre, da série , já sem NAN's, são guardadas em variáveis, **media** e **padrao** que posteriormente irão ser utilizadas para calcular o valor máximo , em coerência, que os valores da série poderão atingir, e o mesmo para um valor mínimo, guardados em **max** e **min**, respetivamente.

Finalmente, iremos percorrer a série, verificando quais os valores que estão acima do **max** e abaixo do **min**, nestes casos, iremos substituir o valor pela média mais  ou menos 2.5 vez o desvio padrão, como sugerido no enunciado.

As séries resultantes em cada passo foram guardadas em variáveis, para que no fim fosse possivel encontrar um ponto critico de entre as três. Como é possivel verificar pela proxima página, temos em primeiro caso a série temporal não regularizada, com Nans e outliers, no segundo caso a série já sem NAN's mas ainda com outliers e finalmente, a série já completamente regularizada. É possivel verificar as mudanças feitas, uma vez que foram ajustados valores que ascendiam até aos 100 kwh e agora no máximo atingem os kwh, custo do ajuste feito e uma vez que a média maxima com o desvio padrão, calculada, é cerca de **46.61,** pelo que só a partir deste valor foram alterados.

Desta forma a série já se encontra regularizada de modo a poder passar à próxima etápa.

```
%Sumarização gráfica e comparação de dados adquiridos
    figure(1)
    subplot(311)
    plot(t,x1,'+-');
    legend('serie temporal Original','Location','northwest')
    xlabel('t[h]');
    title('series temporais');
    subplot(312)
    plot(t,x1n,'+-');
    legend('serie temporal sem NANs e com
outliers','Location','northwest')
    xlabel('t[h]');
    subplot(313)
    plot(t,x1r,'+-');
    legend('serie temporal sem NANs e sem
outliers','Location','northwest')
    xlabel('t[h]');
```



# 2ª Parte (Decomposição da série temporal)

**2.1, 2.2:**

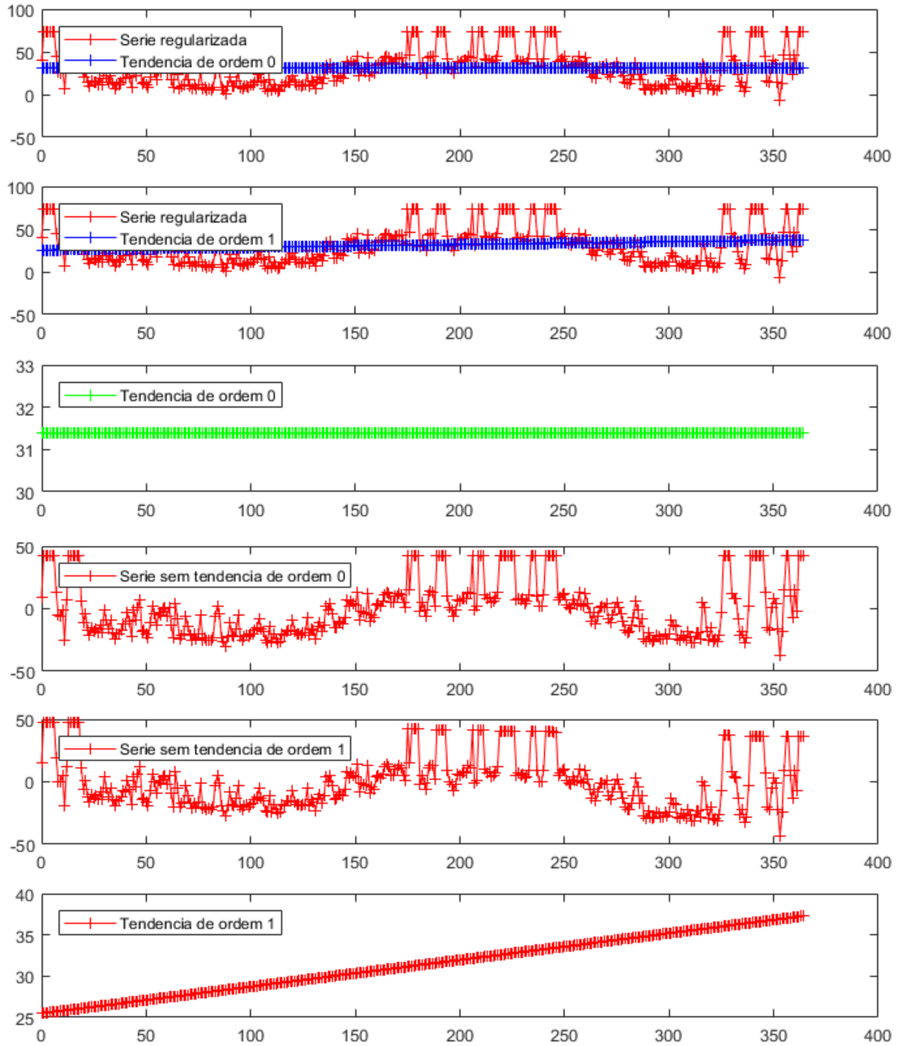
```
x1rz = detrend(x1r,'constant'); %aproximação polinomial de ordem 0
trz = x1r - x1rz; %tendencia de grau 0
x1ru = detrend(x1r,'linear'); %aproximação polinomial de ordem 1
tru = x1r - x1ru; %tendencia de grau 1
```

Nesta alínea, como sugerido, calculamos o melhor ajuste dada a série regularizada **x1r**, obtendo assim as tendências para diferentes graus. Neste caso, calculamos a aproximação polinomial de ordem 0 que guardamos em **x1rz** e posteriormente obtendo a **tendência** subtraindo **x1r** a **x1rz**. O mesmo é feito para a aproximação polinomial de **ordem 1 (linear)**, que guardamos em **x1ru**, obtendo essa **tendência** subtraindo a **x1r**, **x1ru**.

De seguida fazemos a sua representação gráfica.

```
    %Sumarização gráfica e comparação de dados
    figure(2)
    subplot(621)
    plot(t,x1r,'+-r',t,trz,'+-b');
    legend('Serie regularizada','Tendencia de ordem
0','Location','northwest');
    subplot(622)
    plot(t,x1rz,'+-r');
    legend('Serie sem tendencia de ordem 0','Location','northwest');
    subplot(623)
    plot(t,x1r,'+-r',t,tru,'+-b');
    legend('Serie regularizada','Tendencia de ordem
1','Location','northwest');
    subplot(624)
    plot(t,x1ru,'+-r');
    legend('Serie sem tendencia de ordem 1','Location','northwest');
    subplot(625)
    plot(t,trz,'+-g');
    legend('Tendencia de ordem 0','Location','northwest');
    subplot(626)
    plot(t,tru,'+-r');
    legend('Tendencia de ordem 1','Location','northwest');
```

Criamos seis gráficos diferentes para melhor visualizar as componentes. Sendo o primeiro gráfico para a série regularizada com a sua tendência de ordem 0, o segundo para a mesma série sem a tendência de ordem 0, o terceiro é a série com a tendência de ordem 1 , o quarto gráfico corresponde à série sem a componente da tendência de ordem 1 e finalmente o quinto e sexto gráficos correspondem meramente à tendência de ordem 0 e 1, respetivamente.




*Os gráficos não estão por ordem. O eixo das ordenadas corresponde a t[h] (tempo em dias)

O que podemos verificar pelos gráficos é que a tendência de grau 0 corresponde á média da série enquanto que a tendência de grau 1 corresponde ao crescimento linear da série temporal.

**2.3, 2.4, 2.5:**

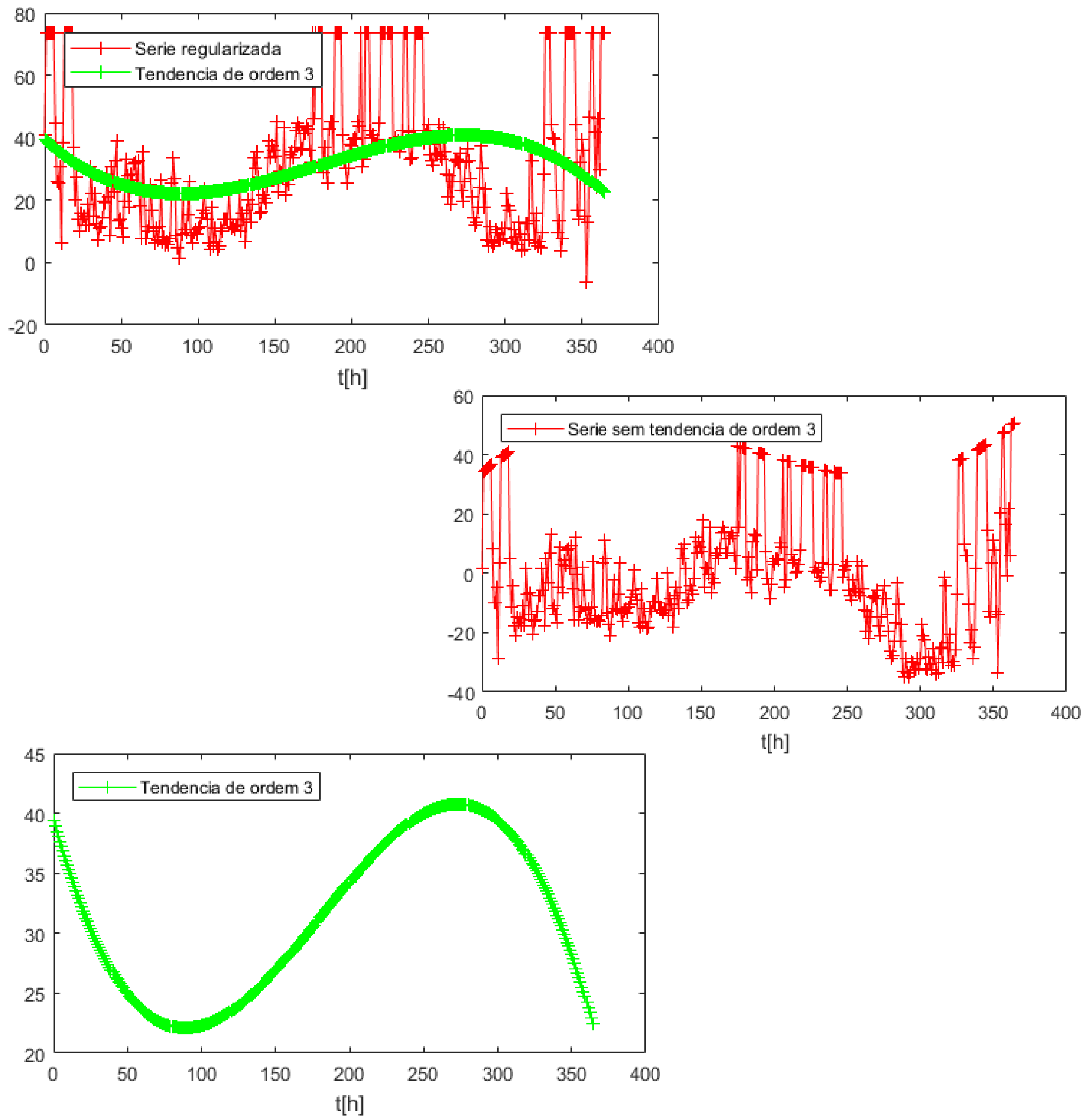
```
    pl1 = polyfit(t,x1r,3); %coeficientes dos polinomios
    trd = polyval(pl1,t); %tendencia de ordem 3
    x1rd = x1r - trd; %calculo de serie reguralizada sem tendencia

    %Sumarização Gráfica e comparação de dados
    figure(3)
    subplot(331)
    plot(t,x1r,'+-r',t,trd,'+-g');
    legend('Serie regularizada','Tendencia de ordem
3','Location','northwest');
    xlabel('t[h]');
```

```
subplot(332)
plot(t,x1rd,'+-r');
legend('Serie sem tendencia de ordem 3','Location','northwest');
xlabel('t[h]');
subplot(333)
plot(t,trd,'+-g');
legend('Tendencia de ordem 3','Location','northwest');
xlabel('t[h]');
```

Nesta secção utilizamos a função **polyfit** que recebe como parâmetros um vetor com o tamanho da série e com os dias de **1 a 365**, a **série** regularizada e o **grau do polinómio**. Neste caso estamos a usar para um polinómio de grau 3 e o que a função vai retornar para a varável **pl1** são os coeficientes do polinómio de grau 3 que melhor se adequa á série temporal regularizada. Assim, utilizando o **polyval**, que recebe como parâmetros os coeficientes e o vetor tempo, ajusta a melhor tendência dessa ordem.

Para obter a série regularizada sem tendência basta retirar à série original (**x1r**) a tendência calculada anteriormente (**trd**).

Graficamente obtemos o seguinte.



Porquê um polinómio de grau 3 e não grau 2 ou até mesmo 4?

Nesta fase é esperado que tenhamos um sentido crítico quanto ao ajuste dos dados e adequar as nossas decisões ao que melhor descreve a série. Com isto, procedemos de antemão ao teste com polinómio de grau 2 mas o seus crescimento não se adequava à série, portanto, procedemos ao grau superior e obtivemos estes resultados que não são ótimos, mas já albergam alguma proximidade com a série em si. No caso do quatro grau, se o aplicássemos, futuramente iriamos ter problemas de acordo com a sazonalidade que estabelecemos, então, mantivemos o 3º grau, mesmo que não ideal, é o mais adequado ao nosso estudo.

**2.6, 2.7, 2.8:**

```
%Estimativa Sazonal
temp = repmat([1:7]',52,1); %sazonalidade semanal
sz = dummyvar(temp); %gera matriz dummy
x1r = x1r(1:end-1,:);
x1rd = x1rd(1:end-1,:);
bs = sz\x1rd;
st = sz*bs; %componente sazonal
x1s = x1rd - st;
t = t(1:end-1,:);
```

Na estimativa da sazonalidade decidimos utilizar uma sazonalidade semanal. Esta sazonalidade como veremos mais em frente limita um pouco o estudo de dados, uma vez que estamos a lidar com vetores de dados até 7 de cada vez, correspondendo ao registo máximo numa semana. Contudo decidimos mesmo assim continuar com a sazonalidade semanal.

Neste excerto, usamos a função repmat, recebendo como parâmetros (r1,r2,…,rn), para obter uma lista de escalares, r1,..,rN, que descrever como as copias de A são organizados em cada dimensão.

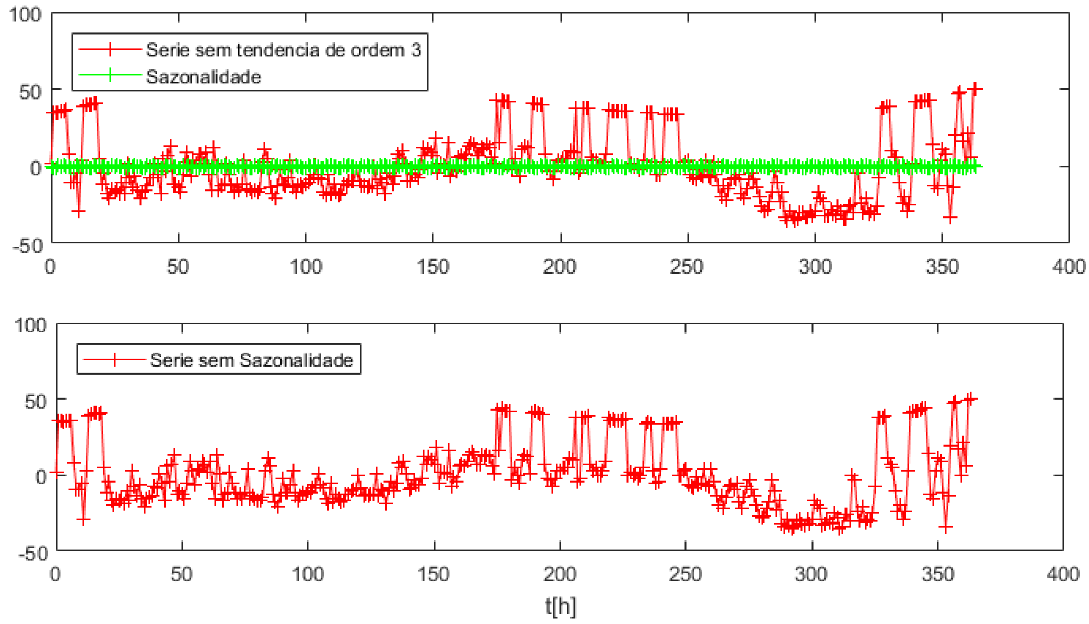
A variável **st** guarda a componente sazonal, calculada a partir da componente tendencial de ordem 3, que era a que melhor se adequava no nosso caso.

Para obter a série regularizada sem a sua componente sazonal, guardámo-la em **x1s** após retirar à série sem tendência de grau 3 (**x1rd**) a componente sazonal (**st**).

Para estas alíneas a sumarização gráfica foi realizada da seguinte forma.

```
%Sumarização Gráfica e comparação de dados
figure(4)
subplot(421)
plot(t,x1rd,'+-r',t,st,'+-g');
legend('Serie sem tendencia de ordem 
4','Sazonalidade','Location','northwest');
xlabel('t[h]');
subplot(422)
plot(t,x1s,'+-r');
legend('Serie sem Sazonalidade','Location','northwest');
xlabel('t[h]');
```

Que dá origem aos seguintes gráficos.



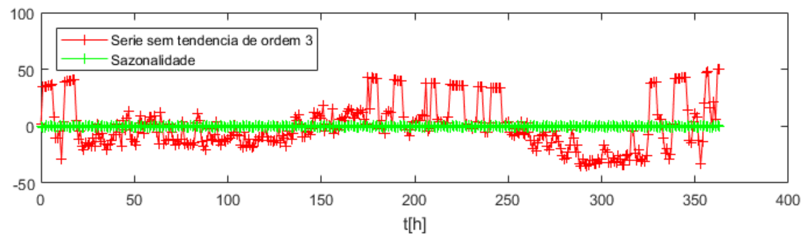

**2.9, 2.10:**

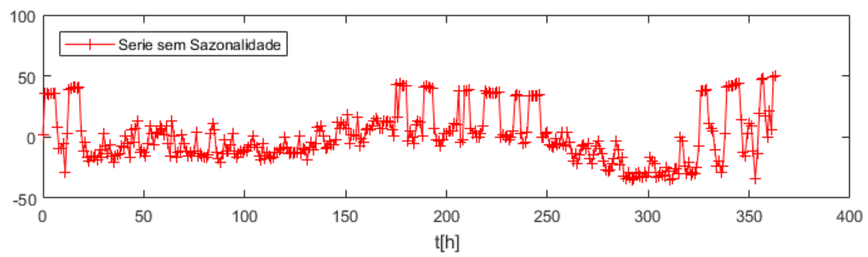
```
irreg = x1r-pl1-st; %componente irregular
snirreg = x1r-irreg; %serie temporal sem componente irregular

%Sumarização Gráfica e comparação de dados
figure(5)
subplot(521)
plot(t,snirreg,'+-r');
legend('Serie sem componente irregular','Location','northwest');
subplot(522)
plot(t,irreg,'+-r');
legend('Serie com componente irregular','Location','northwest');
```

Para o cálculo da componente irregular fizemos apenas uma subtração entre a série original regularizada (**x1r**), os coeficientes dos polinómios de grau 3 (**pl1**) e a componente sazonal (**st**), guardando-a na variável **irreg**.

Para obtenção da série temporal sem essa mesmo componente irregular, bastou-nos subtrair essa mesma componente irregular à série original regularizada (**x1r**), guardando o resultado em **snirreg**.

Dados isto passamos para expor os resultados graficamente.






# 3ªParte (Estimação de modelos)

**3.1, 3.2, 3.3:**

```
    tt=(0:2*N-1)'; %escala temporal para previsão


    est = adftest(x1r); %teste de estacionaridade da série
regularizada
    estst = adftest(st); %teste de estacionaridade da componente
sazonal

    disp(est);

    t1=(0:6)';
    y1=st(1:7); %componente sazonal
    disp(y1);

    figure(6)
    subplot(621)
    autocorr(y1); %indicador para a ordem de nc(modelo MA)
    subplot(622)
    parcorr(y1); %indicador para a ordem de na(modelo AR)
```

Nesta parte do projeto já estamos mais focados em perceber a **estacionaridade** da série, ou seja, verificar se a séria com a qual trabalhamos é estacionária ou não.

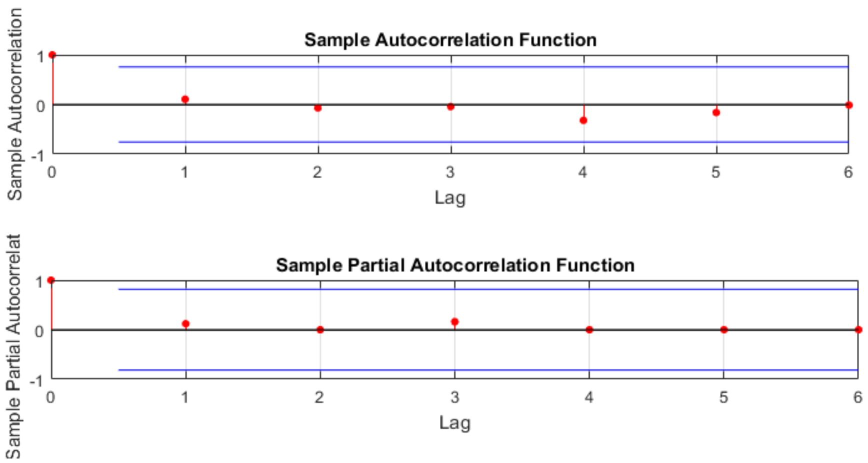
Para procedermos a este teste utilizamos a função **adftest**(x1r) que recebe como parâmetro **x1r**, que não é mais que a série regularizada que utilizamos até aqui. Esta função irá dar retorno de um valor. Este valor será guardado na variável temporária **est**. O retorno dado por esta função é um valor numeral, que se for 1, indica-nos que a nossa série é realmente estacionária. O que acontece neste nosso caso. Esta informação será útil para a estimação dos modelos mais adiante.

O mesmo é feita para a componente sazonal calculada na secção anterior, **st**.

Para proceder à identificação do modelo, representamos graficamente **a função de Auto correlação** (**autocorr**), que deduz a ordem de **nc** (Modelo MA) e utilizamos a **função de auto correlação parcial** da componente sazonal (**parcorr**(y1)), que deduz o indicador para a ordem de na (Modelo AR).

A razão pela qual indicamos vetores de 0 a 6 e 1 a 7 é pelo mero facto que a nossa componente sazonal é semanal.

Feito isto, é representado graficamente os resultados destas funções dados os seus parâmetros.




**3.4, 3.5, 3.6:**

```
id_y1 = iddata(y1, [], 1, 'TimeUnit', 'days');

opt1_AR = arOptions('Approach', 'ls');
na1_AR = 3; %histórico da variável
model1_AR = ar(id_y1,na1_AR, opt1_AR); %modelo AR
pcoef1_AR = polydata(model1_AR); %parâmetros do modelo AR

y1_AR = y1(1:na1_AR);

for k=na1_AR+1:7
    y1_AR(k)=sum(-pcoef1_AR(2:end)'.*flip(y1_AR(k-na1_AR:k-1)));
end

y1_AR2=repmat(y1_AR,52,1);

y1_ARf=forecast(model1_AR,y1(1:na1_AR),7-na1_AR);
y1_ARf2=repmat([y1(1:na1_AR); y1_ARf],52,1);
```

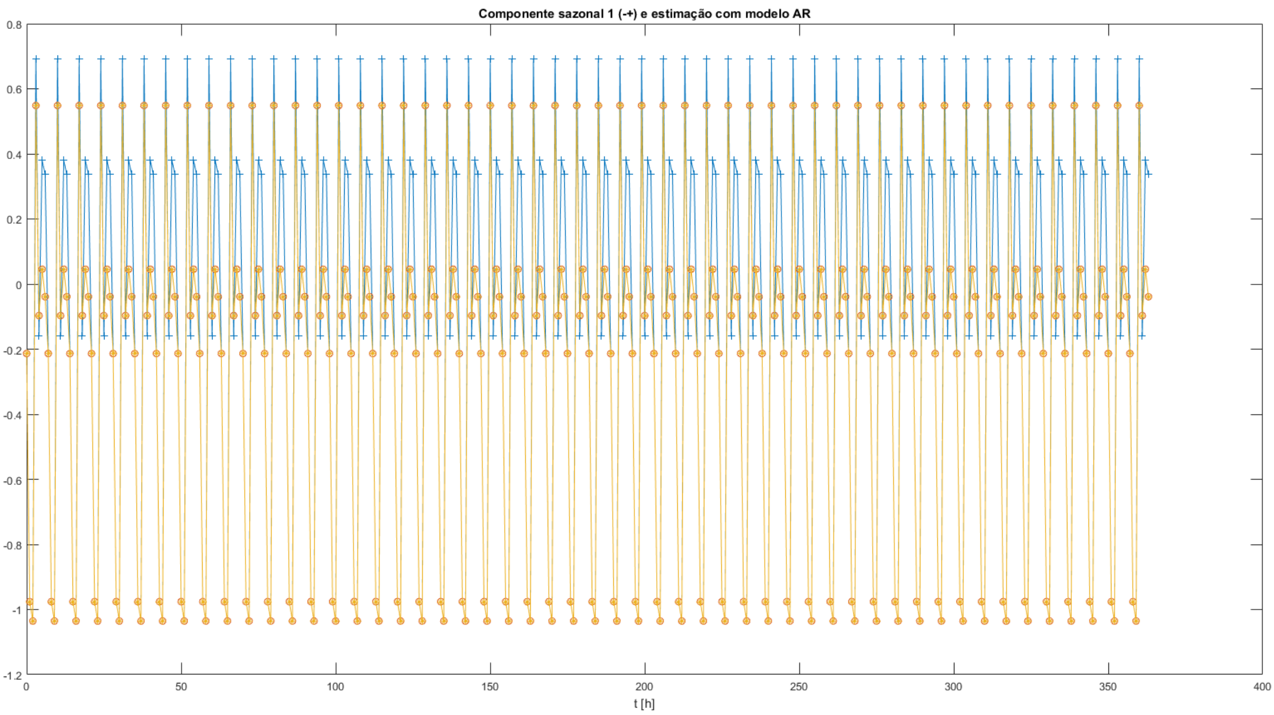
Em primeira fase criamos o objeto **id_y1** com a função **iddata**, com unidade de tempo em dias.

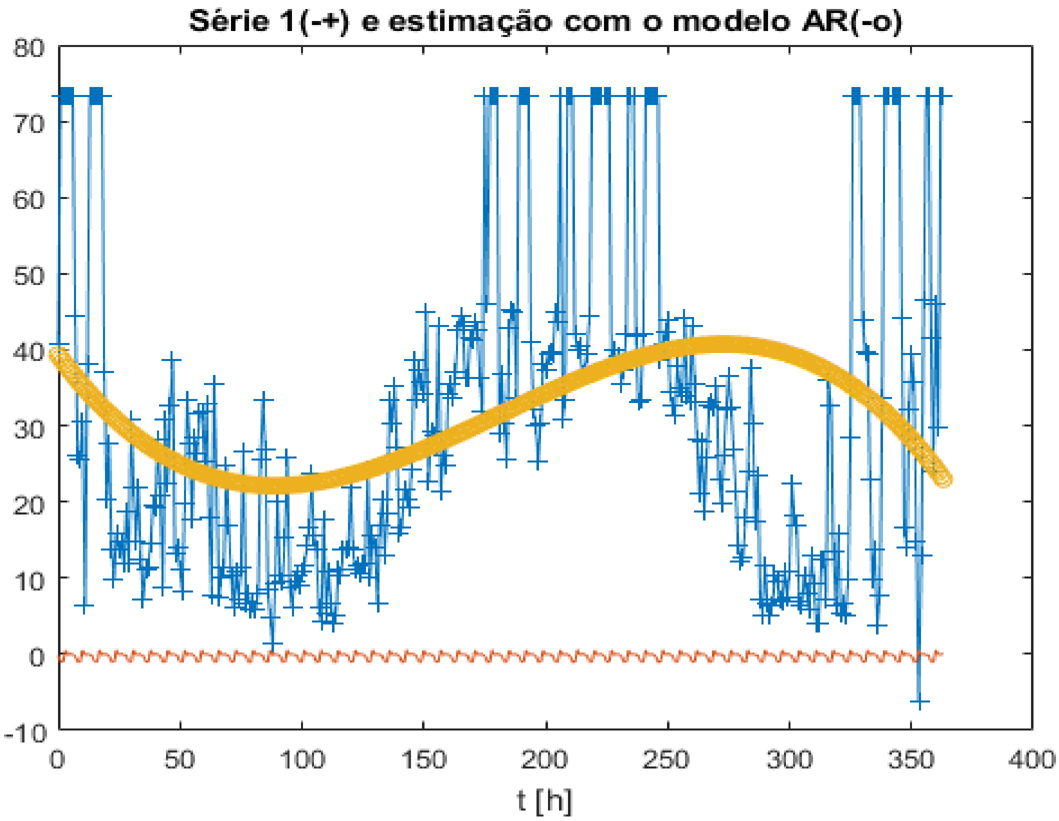
Em seguida vamos estimar o modelo AR, para a componente sazonal que determinamos e considerando o resultado que nos deu na função de auto correlação parcial, deduzir o valor de na, que é o histórico da variável a considerar. Utilizando a função **arOptions** criamos a opções padrão para criar o modelo ar. A seguir vamos criar o modelo ar com a função **ar** que tem como parâmetros **id_y1** que é o objeto **iddata** para componente sazonal, **na1_AR** que é o histórico da variável e finalmente **opt1_AR** que é são as opções criadas anteriormente.

O **na (na1_AR)** foi deduzido tendo em conta a função de auto correlação parcial e a própria componente sazonal, uma vez que se o na fosse mais elevado que 3, iriamos ter um número de informação para estudo demasiado reduzida, pelo que o modelo não iria concluir nada, tendo sido testados esses valores. Dito isto a melhor opção foi mesmo colocar o **na** a 3.

Para comparar graficamente a componente sazonal com o modelo utilizamos a função **forecast** que gera a média e os erros quadrados relativamente à série.

```
figure(7)
plot(t,st,'-+',t,y1_AR2,'-o',t,y1_ARf2,'-*');
xlabel('t [h]');
title('Componente sazonal 1 (-+) e estimação com modelo AR');
figure(8)
plot(t,x1r,'-+',t,y1_AR2, t, trd(1:end-1) ,'-o');
xlabel('t [h]');
title('Série 1(-+) e estimação com o modelo AR(-o)');
```






**3.9, 3.10, 3.11**

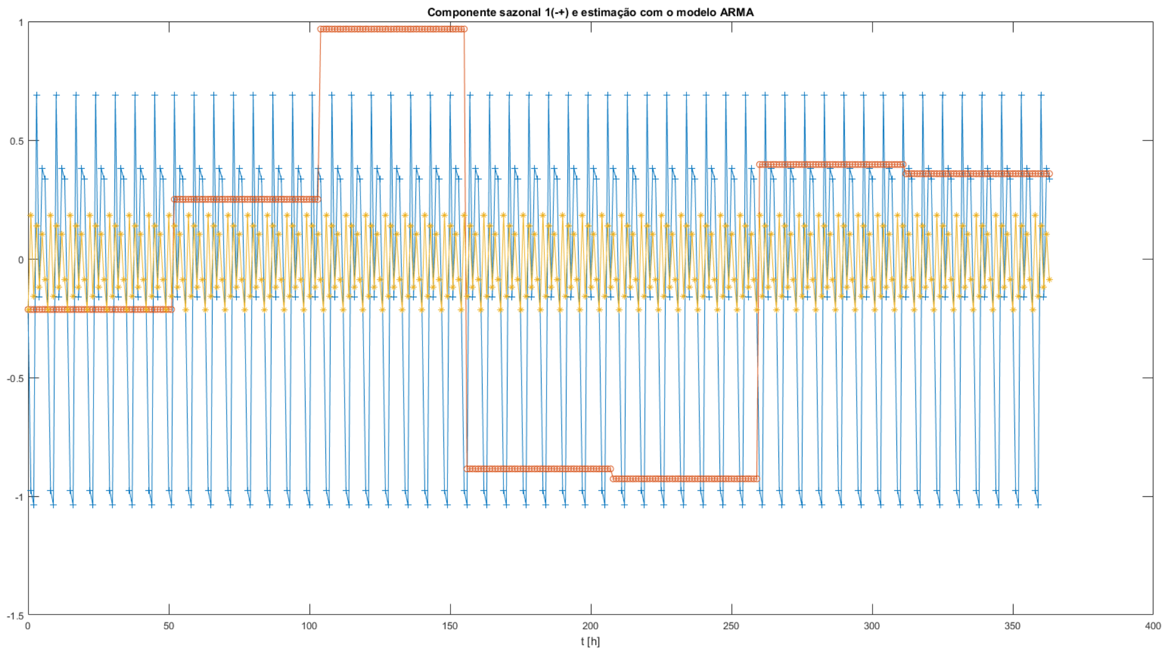
```
opt1_ARMAX = armaxOptions('SearchMethod', 'auto');
na1_ARMA=1;
nc1_ARMA=1;
model1_ARMA = armax(id_y1,[na1_ARMA nc1_ARMA], opt1_ARMAX);
[pa1_ARMA,pb1_ARMA,pc1_ARMA] = polydata(model1_ARMA);

e = randn(7,1); %ruído branco
y1_ARMA = y1(1:na1_ARMA);

for k=na1_ARMA+1:7
    y1_ARMA(k)=sum(-pa1_ARMA(2:end)'.*flip(y1_ARMA(k-na1_ARMA:k-1)))+sum(pc1_ARMA'.*flip(e(k-nc1_ARMA:k)));
end
y1_ARMA2=repmat(y1_ARMA,52,1);

%Simulação do modelo arma com forecast
y1_ARMAf=forecast(model1_ARMA, y1(1:na1_ARMA),7-na1_ARMA);
y1_ARMAf2=repmat([y1(1:na1_ARMA); y1_ARMAf],52,1);
y1_ARMA2 = y1_ARMA2(:);
figure(10) %compara a componente sazonal com a sua estimação
plot(t,st,'-+',t,y1_ARMA2,'-o',t,y1_ARMAf2,'-*');
xlabel('t [h]');
title('Componente sazonal 1(-+) e estimação com o modelo ARMA');

figure(11) %compara a série com o modelo ARMA + tendência
plot(t,x1r,'-+',t,y1_ARMA2+trd(1:end-1),'-o');
xlabel('t [h]');
title('Série 1 (-+) e estimação com o modelo ARMA(-o)')
```

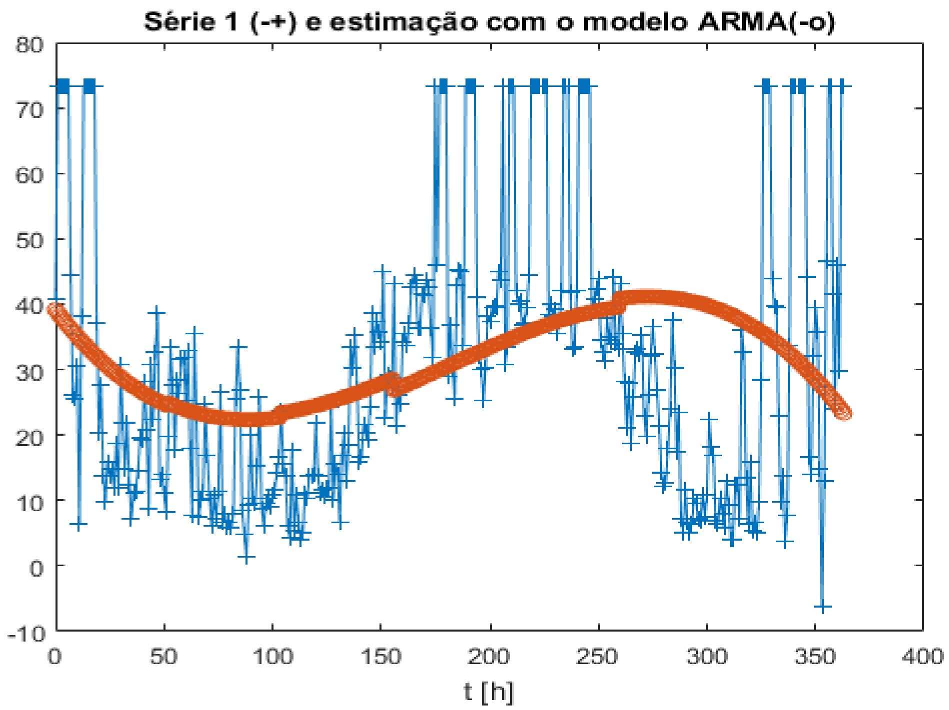
O procedimento para a estimação do modelo ARMA é muito semelhante à estimação do modelo anteriormente estimado. Contudo, neste caso, a seguir ao criar as opções padrão para o modelo, temos de deduzir valores para o **na** e para o **nc**, que é o histórico do ruído branco a considerar, tendo em contra o resultado na **função de auto correlação**.

Nesta parte, para estimar o modelo ARMA, utilizamos a função armax que leva como parâmetros o objeto **iddata** (**id_y1**), os históricos deduzidos e as opções padrão para o modelo a estimar, **opt1_ARMAX**.

A dedução nesta fase foi novamente feita com base na função de auto correlação parcial e normal, tendo sempre em conta as limitações da componente sazonal semanal, utilizada para este projeto.

Para obter o ruido branco, geramos um número aleatório de 1 a 7, guardado em **e**.



**3.13, 3.14, 3.15, 3.16, 3.17**:

```
  %Estimação de um modelo ARIMA
p1_ARIMA = 3;
q1_ARIMA = 3;
Mdl = arima(p1_ARIMA,0,q1_ARIMA);
EstMd1 = estimate(Mdl,x1r(1:N-1),"Y0",x1r(1:p1_ARIMA+1));

%Simulação do modelo ARIMA
y1_ARIMA = simulate(EstMd1,N-1);

figure(13) %compara a serie com a sua estimaçao
plot(t,x1r,'-+',t,y1_ARIMA,'-o');
xlabel('t [h]');
title('Série 1(-+) e estimação com o modelo ARIMA(-o)')

%métrica para análise
E1_ARIMA = sum((x1r-y1_ARIMA(1:N-1)).^2);

%Simulação do modelo ARIMA para 2N
y1_ARIMA2 = simulate(EstMd1,2*N);

figure(14) %faz a previsão para 2N
plot(t,x1r,'-+',tt,y1_ARIMA2,'-o');
xlabel('t [h]');
title('Série 1(-+) e estimação com o modelo ARIMA(-o)')
```

Em último ponto, é nos pedido para estimar também um modelo ARIMA, em alternativa ao modelos anteriormente estimados.

Para estimar este modelo é necessário ter em mente valores adequados para **p1_ARIMA**, que é o grau do histórico da variável a considerar, **D** , que no nosso caso é 0 uma vez que a série é estacionária e finalmente **q1_ARIMA** que é o grau do histórico de ruído branco a considerar. Estes valores novamente serão deduzidos tendo em mente todas as limitações geradas pelas decisões tomadas até então.

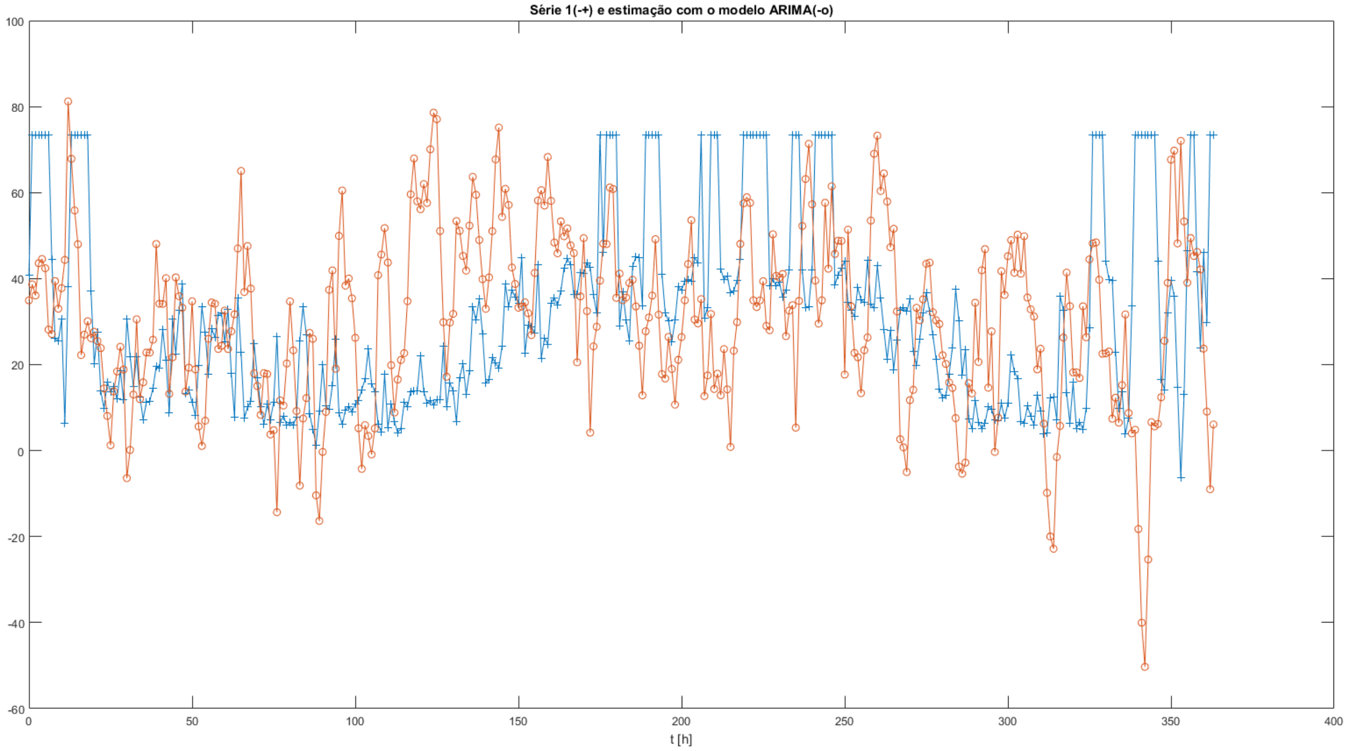
Todos estes valores serão usados na função **arima**, que vai criar a base para o modelo.

Em seguido usar a função **estimate** estimar o modelo da série.

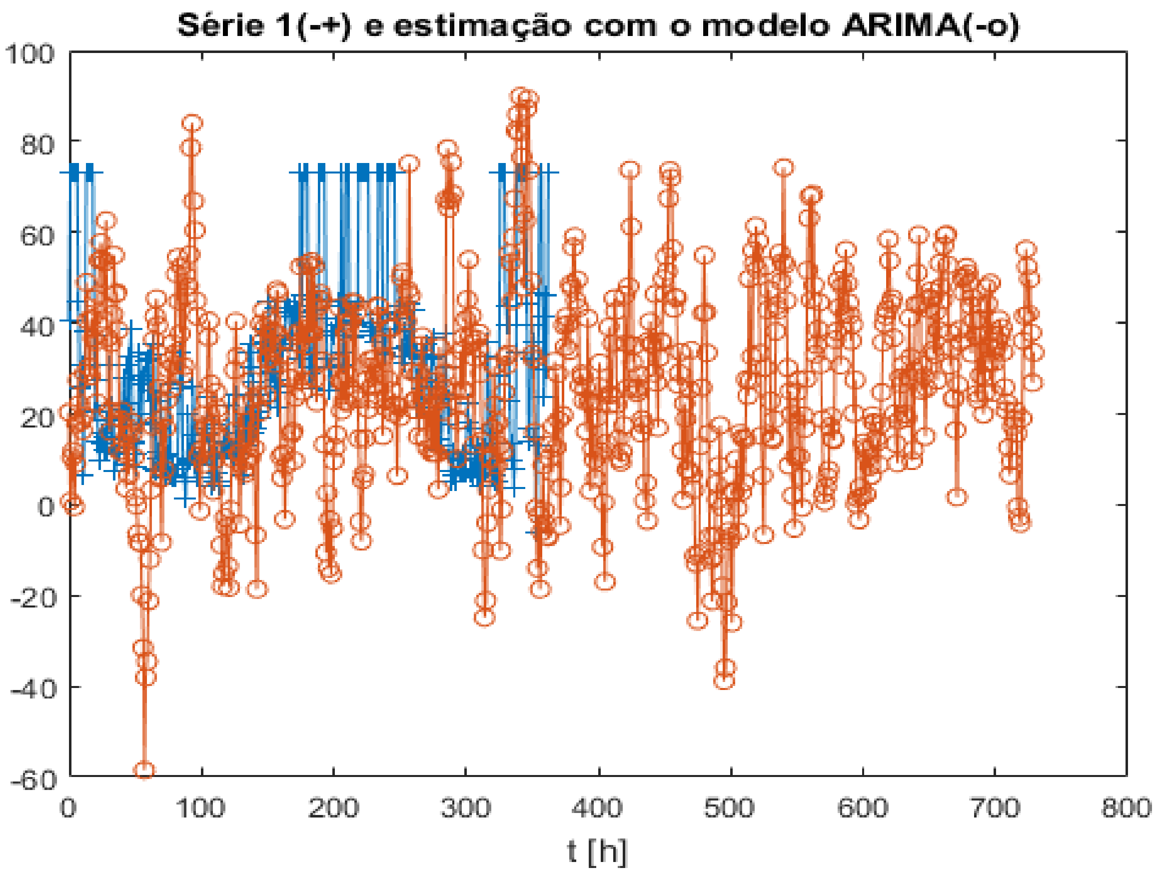
Para simular o modelo ARIMA para a série usamos **simulate**, com parâmetros **estMd1**, que seria a estimativa para o modelo ARIMA e **N-1** que é o tamanho da série.

O gráfico que compara a série com a sua estimação:

***(PRÓXIMA PÁGINA)***




Para estimar o comportamento da série seguindo este modelo para uma duração que o dobro da original, neste caso 365 dias, utilizamos uma linha temporal de 730 dias e este é o resultado obtido.



Em situações ótima, este modelo iria estar determinado com valores positivos tendo em conta a tendência somada, mas neste caso mesmo elevando e ajustando os valores o modelo não dispersava muito deste resultado, pelo que decidimos expor aquele que nos deu em primeira mão.

# Modelo mais adequado

Aceitaríamos o modelo ARMA como mais adequado a esta série de dados.

Tendo em mente que este modelo estatístico é central particularmente em análise de séries temporárias, acaba por ser uma globalização e generalização dos modelos, também estudados, AR e MA (modelo de auto-regressão e Modelo de médias móveis).

Estes modelos são todos adequados ao tipo de dados, ajustando-se a estes.

Em relação ao modelo AR (auto-regressão), presente no modelo ARMA, garante que a variável que evolui é estimada tendo em conta valores passados.

Finalmente, era aceitável considerar um modelo que acaba por ser uma generalização de dois modelos estimados, AR (modelo de auto-regressão), MA (média móvel) estruturando o modelo ARMA(modelo de auto-regressão e média móvel). O que diferencia o ARIMA do ARMA é o facto de este ser não ser integrado. Esta integração garante mais fidelidade a modelos não estacionários, que não é o caso presente neste estudo.

# Suma

Deste projeto podemos elevar alguns aspetos mais salientes e determinantes. A importância de cuidar os dados que nos são gerados num determinado tempo e com um determinado objetivo é determinante nas conclusões que tomamos em prol do avanço tecnológico ou até mesmo para melhorar as condições de pensamento da sociedade. Quanto mais fidedignos forem os resultados obtidos por um bom data set de informação bem mantida, mais precisa a conclusão.

Quanto ao teste para um tempo específico e para fazer previsões do mesmo sistema de modo a poder precaver soluções para melhorar as previsões, é determinante sabermos como abordar uma boa estratégia para obter estas previsões.

Em suma, é determinante para um dado público, a maneira como se abordam estes dados e esta informação. Os cuidados a ter e as estratégias mais bem delineadas e adequadas a um objetivo que centra nesta informação.