

Modelação e Previsão de Séries Temporais

Tarefas

2023/2024

Nota: Algumas justificações e códigos podem parecer confusos ou desorganizados, pois cada tarefa foi realizada em momentos diferentes. Além disso, cada tarefa foi aplicada em simultâneo a cinco países diferentes, e não a um único país, ao contrário do que se verifica neste documento.

Dados fornecidos

Recolhidos do site do Eurostat, referentes ao consumo mensal de eletricidade nos países da UE.

```
head(eu_countries, 5)
```

```
## # A tibble: 5 x 3
##   code name    label
##   <chr> <chr>   <chr>
## 1 AT    Austria Austria
## 2 BE    Belgium Belgium
## 3 BG    Bulgaria Bulgaria
## 4 CY    Cyprus   Cyprus
## 5 CZ    Czechia  Czechia
```

```
head(data.tsb, 5)
```

```
## # A tibble: 5 x 3
##   geo    time    values
##   <chr> <date>   <dbl>
## 1 AT    2008-01-01  6187
## 2 AT    2008-02-01  5725
## 3 AT    2008-03-01  5878
## 4 AT    2008-04-01  5438
## 5 AT    2008-05-01  5218
```

Seleção e tratamento dos dados

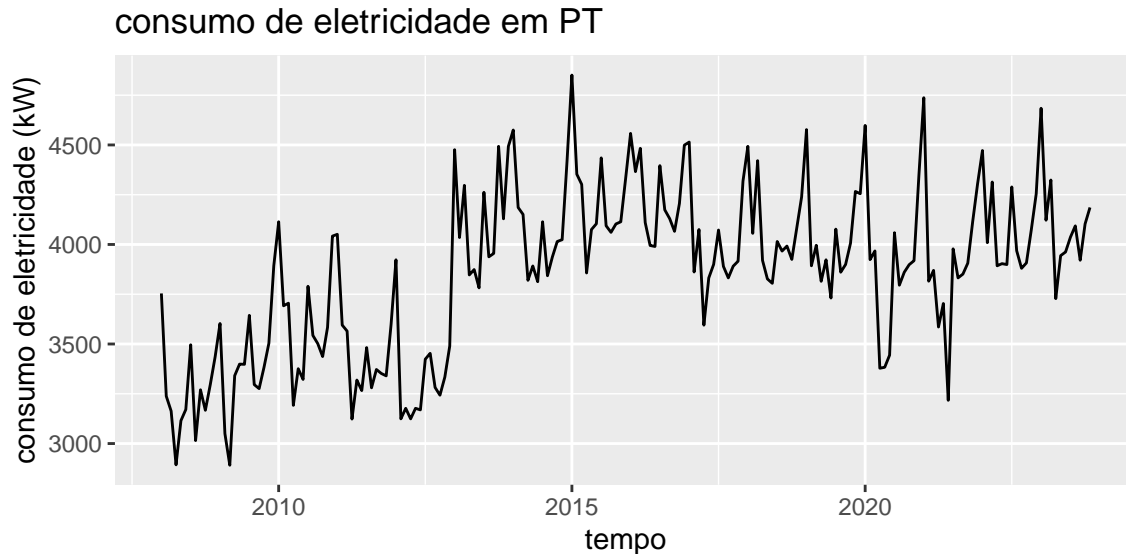
```
PT.data <- data.tsb |>
  filter(geo == "PT") |>
  select(-geo)
```

```
PT.ts <- ts(PT.data$values, start=c(2008,1), frequency=12)
```

Tarefa 1 de 4

Q1: Representação gráfica da série temporal

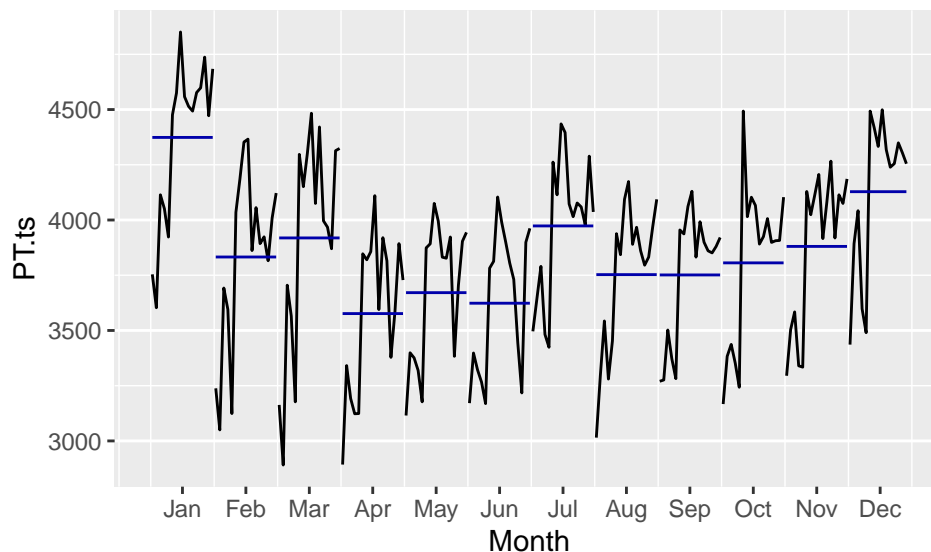
```
autoplot(PT.ts) + xlab("tempo") +  
  ylab("consumo de eletricidade (kW)") +  
  ggtitle("consumo de eletricidade em PT")
```



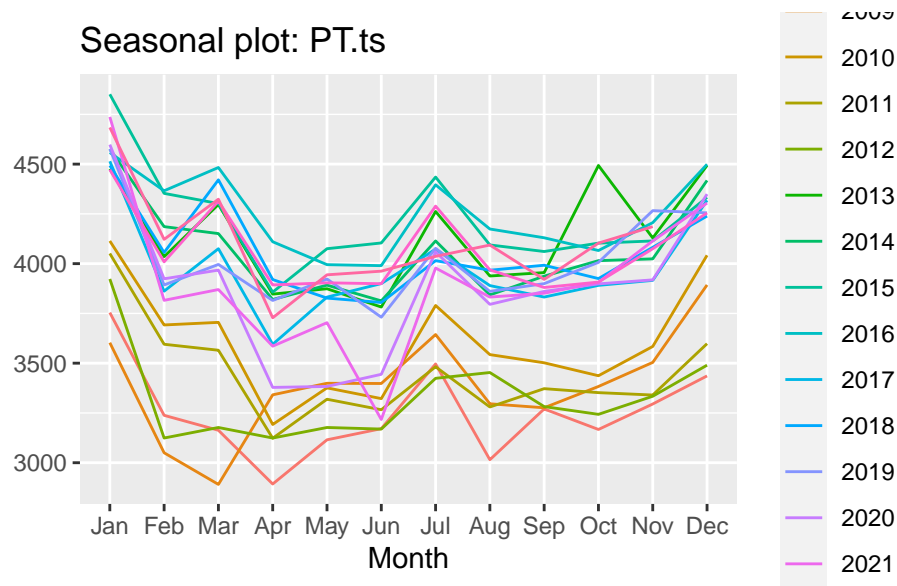
Quanto a Portugal, até por volta de 2013 observa-se uma tendência crescente não muito significativa, por volta de 2013 observa-se um aumento significativo do consumo de eletricidade, e que a partir daí parece haver uma estabilização e ligeiro decréscimo, ao longo do tempo, do consumo de eletricidade. Para além disso também se observa sazonalidade, uma queda por volta de 2020, tal como os outros países, e ainda uma outra queda à volta de 2022.

Q2: Evolução da componente sazonal ao longo dos anos (graficamente)

```
ggsubseriesplot(PT.ts)
```



```
ggseasonplot(PT.ts)
```



Atendendo a Portugal, através do primeiro gráfico observa-se que, em média, em janeiro atingimos o maior consumo de eletricidade e em abril o menor. De janeiro para fevereiro há uma queda, março sobe ligeiramente, abril volta a descer matendo o nível até junho. Julho tem um aumento do consumo de eletricidade, mas agosto tem uma queda voltando o nível de consumo de eletricidade a estar, em média, ligeiramente acima do nível de abril/maio/junho. Este nível vai aumentando ligeiramente até novembro, tendo uma subida mais significativa em dezembro. Através do segundo gráfico é mais notório a diferença dos consumos entre os anos, mas não sendo isto necessário para esta questão.

Q3: Decomposição clássica aditiva e multiplicativa

```
PT.dec.adt <- decompose(PT.ts, type="additive")
PT.dec.mul <- decompose(PT.ts, type="multiplicative")
```

Q4: Índices sazonais

```
round(PT.dec.adt$figure, 2)
```

```
## [1] 542.34 -5.13 87.12 -264.59 -183.37 -249.35 125.22 -118.74 -114.38
## [10] -73.74 -4.46 259.08
```

```
round(PT.dec.mul$figure, 2)
```

```
## [1] 1.14 1.00 1.02 0.93 0.95 0.94 1.03 0.97 0.97 0.98 1.00 1.07
```

Quanto a Portugal, observa-se que o mês com maior e menor consumo são janeiro e abril, respetivamente. Pela decomposição aditiva observa-se que o valor em janeiro é superior à média em 542 kW e o de abril é inferior em 265 kW. Pela dec. multiplicativa observa-se que janeiro tem um acréscimo de 14% em relação à média e abril um decréscimo de 7%. Isto está em concordância com a análise efetuada na questão 2.

Q5: Transformação necessária? (BoxCox)

```
#autoplot(PT.ts)
BoxCox.lambda(PT.ts)
```

```
## [1] 1.111554
```

Uma transformação nesta série temporal seria importante se fosse possível suavizar o aumento repentino que surge à volta de 2013 e também suavizar os picos da sazonalidade ao longo dos anos que possam estar a originar outliers, como parece acontecer em vários anos, tais como à volta de 2015, 2020 e 2021. O λ é próximo de 1, o que não sugere grande transformação, mas seria interessante analisar.

Q6: Decomposição de Loess

```
test.Outliers.STL(PT.ts)
```

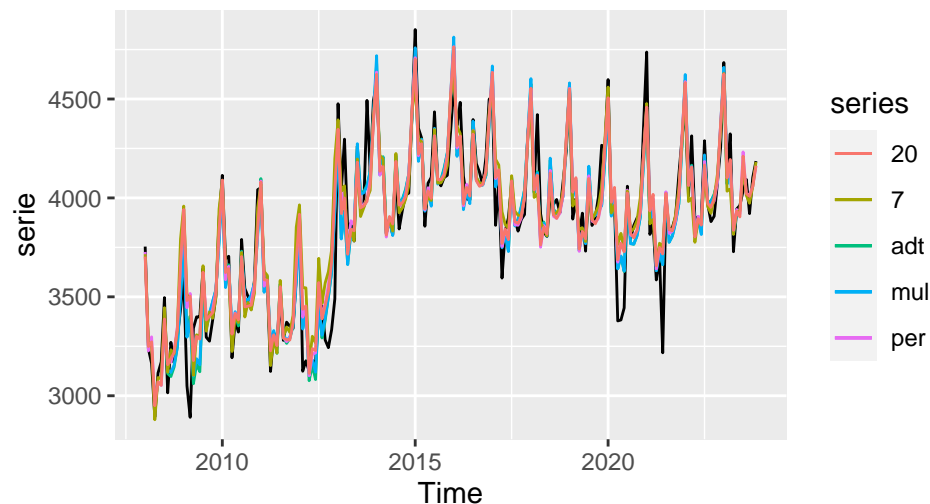
```
## [1] TRUE
```

A presença de outliers implica o uso de decomposições robustas (`robust = TRUE`)

```
PT.dec.stl11 <- stl(PT.ts, s.window = "periodic", robust = TRUE)
PT.dec.stl7 <- stl(PT.ts, s.window = 7, robust = TRUE)
PT.dec.stl20 <- stl(PT.ts, s.window = 20, robust = TRUE)
```

Q7: Representação gráfica das séries ajustadas

```
serie <- PT.ts
autoplot(serie) +
  autolayer(serie - PT.dec.adt$random, series="adt") +
  autolayer(serie / PT.dec.mul$random, series="mul") +
  autolayer(serie - PT.dec.stl11$time.series[, "remainder"], series="per") +
  autolayer(serie - PT.dec.stl7$time.series[, "remainder"], series="7") +
  autolayer(serie - PT.dec.stl20$time.series[, "remainder"], series="20")
```



Q8: Medidas de erro para cada decomposição

```
serie <- PT.ts
accuracy(serie - PT.dec.adt$random, PT.ts)[,c("ME", "RMSE", "MAE")]
```

```
##           ME           RMSE           MAE
## 0.1522433 136.8908090 100.1481843
```

```
accuracy(serie / PT.dec.mul$random, PT.ts)[,c("ME", "RMSE", "MAE")]
```

```
##           ME           RMSE           MAE
## 0.3187023 136.5617277 99.6676257
```

```
accuracy(serie - PT.dec.stl1$time.series[, "remainder"], PT.ts)[,c("ME", "RMSE", "MAE")]
```

```
##          ME          RMSE          MAE
## -16.84662 141.89044   96.24274
```

```
accuracy(serie - PT.dec.stl7$time.series[, "remainder"], PT.ts)[,c("ME", "RMSE", "MAE")]
```

```
##          ME          RMSE          MAE
## -22.61293 155.61014   94.40975
```

```
accuracy(serie - PT.dec.stl20$time.series[, "remainder"], PT.ts)[,c("ME", "RMSE", "MAE")]
```

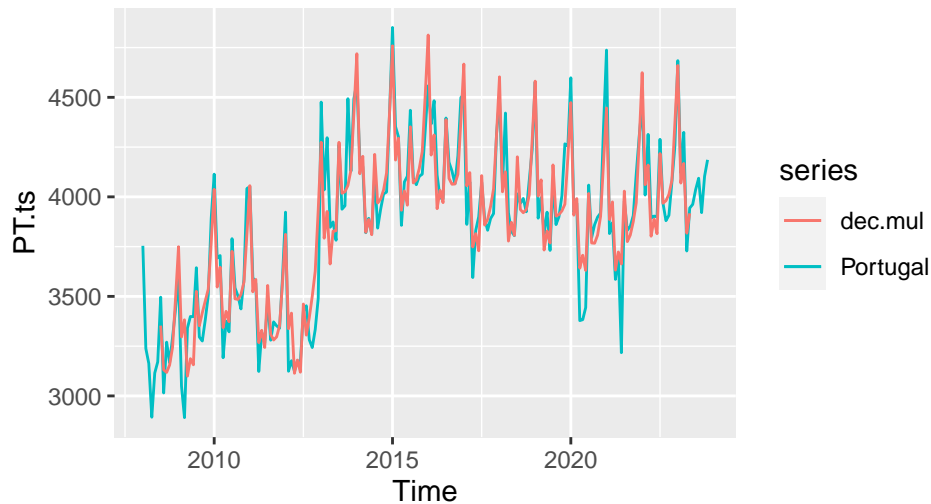
```
##          ME          RMSE          MAE
## -15.51186 141.61694   96.35864
```

O melhor método de decomposição para Portugal é: decomposição clássica multiplicativa. Contudo, esta decomposição está bastante próxima dos valores da aditiva pelo que ambos os modelos serão bastante bons.

Q9: Melhor ajustamento face aos dados reais

```
autoplot(PT.ts, series="Portugal") +
  autolayer(PT.ts / PT.dec.mul$random, series="dec.mul")
```

```
## Warning: Removed 12 rows containing missing values (`geom_line()`).
```



Tarefa 2 de 4

Pré-requisitos

Definição de variáveis

```
PT.train <- window(PT.ts, end = c(2022, 12))
PT.test <- window(PT.ts, start = c(2023, 1))
```

Melhor decomposição

Utilizei a decomposição clássica (aditiva e multiplicativa), decomposição de Loess (`s.window` igual a “periodic”, 7, 15, 20) e a função `stl.fit`.

```
tseries <- PT.train

# clássica
accuracy(tseries - decompose(tseries, type = "additive")$random, tseries)
accuracy(tseries / decompose(tseries, type = "multiplicative")$random, tseries)

# Loess
accuracy(tseries - remainder(stl(tseries, s.window = ...,
                                robust = test.Outliers.STL(tseries))), tseries)

# stl.fit
accuracy(tseries - remainder(stl.fit(tseries, k = 2,
                                     rob = test.Outliers.STL(tseries))$stl), tseries)
```

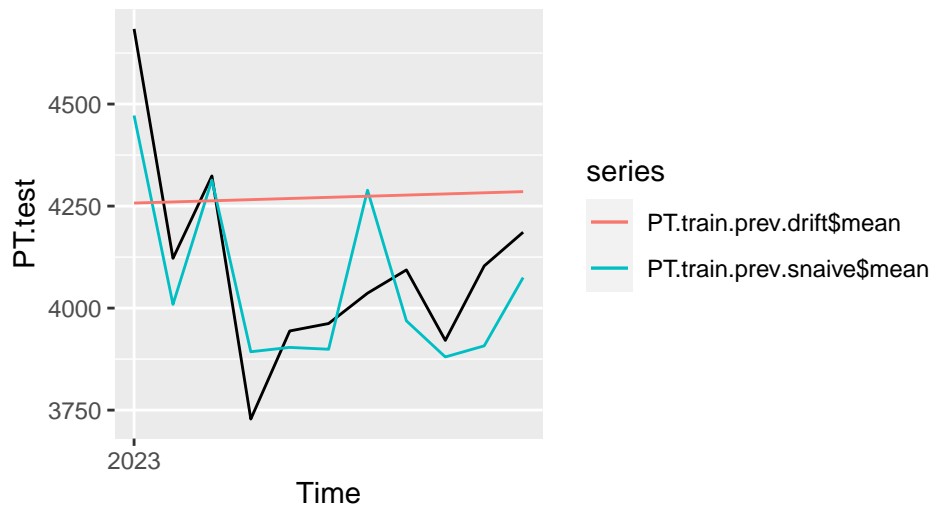
Decomposição	adt	mul	stlper	stl7	stl15	stl20	stl.fit
PT.train	139.503	139.355	142.7396	157.1439	146.3441	145.8309	140.1014

```
PT.dec.classic.mul <- decompose(PT.train, type = "multiplicative")
PT.dec.stl <- stl(PT.train, s.window = "periodic", robust = test.Outliers.STL(PT.train))
PT.dec.stlfit <- stl.fit(PT.train, k = 2, rob = test.Outliers.STL(PT.train))$stl
```

Q1: Previsões através de métodos simples

```
n <- length(PT.test)
PT.train.prev.snaive <- snaive(PT.train, h=n)
PT.train.prev.drift <- rwf(PT.train, drift=TRUE, h=n)

autoplot(PT.test) + autolayer(PT.train.prev.snaive$mean) + autolayer(PT.train.prev.drift$mean)
```

Os dados iniciais apresentavam tendência e sazonalidade, daí a escolha destas duas previsões.

Q2: Previsões a partir da decomposição clássica

```
tserie.test <- PT.test
n <- length(tserie.test)
dec <- PT.dec.classic.mul

A.drift <- rwf(seasadj(dec), drift = TRUE, h=n)$mean
A.meanf <- meanf(seasadj(dec), h=n)$mean
A.naive <- naive(seasadj(dec), h=n)$mean
S <- snaive(seasonal(dec), h = n)$mean

AS.sdrift <- A.drift * S
accuracy(AS.sdrift,tserie.test)[,"RMSE"]

## [1] 160.2161

AS.smeanf <- A.meanf * S
accuracy(AS.smeanf,tserie.test)[,"RMSE"]

## [1] 301.1516

AS.snaive <- A.naive * S
accuracy(AS.snaive,tserie.test)[,"RMSE"]

## [1] 177.768

#autoplot(tserie.test) + autolayer(AS.sdrift) + autolayer(AS.smeanf) + autolayer(AS.snaive)

PT.classic.prev <- A.drift * S
```

PT, com decomposição clássica - snaive (para a componente sazonal) * drift (para a componente sem sazonalidade) é a melhor previsão

Q3: Previsões a partir da decomposição de Loess

```
tserie.test <- PT.test
n <- length(tserie.test)
dec <- PT.dec.stl
```

```

A.drift <- rwf(seasadj(dec), drift = TRUE, h=n)$mean
A.meanf <- meanf(seasadj(dec), h=n)$mean
A.naive <- naive(seasadj(dec), h=n)$mean
S <- snaive(seasonal(dec), h = n)$mean

AS.sdrift <- A.drift + S
accuracy(AS.sdrift,tserie.test)[,"RMSE"]

## [1] 167.3951

AS.smeanf <- A.meanf + S
accuracy(AS.smeanf,tserie.test)[,"RMSE"]

## [1] 302.3336

AS.snaive <- A.naive + S
accuracy(AS.snaive,tserie.test)[,"RMSE"]

## [1] 188.012

#autoplot(tserie.test) + autolayer(AS.sdrift) + autolayer(AS.smeanf) + autolayer(AS.snaive)

PT.stl.prev <- A.drift + S

```

PT, com decomposição de Loess - drift + snaive

Q4: Previsões a partir do uso da função stl.fit

```

tserie.test <- PT.test
n <- length(tserie.test)
dec <- PT.dec.stlfit

A.drift <- rwf(seasadj(dec), drift = TRUE, h=n)$mean
A.meanf <- meanf(seasadj(dec), h=n)$mean
A.naive <- naive(seasadj(dec), h=n)$mean
S <- snaive(seasonal(dec), h = n)$mean

AS.sdrift <- A.drift + S
accuracy(AS.sdrift,tserie.test)[,"RMSE"]

## [1] 165.6724

AS.smeanf <- A.meanf + S
accuracy(AS.smeanf,tserie.test)[,"RMSE"]

## [1] 301.7216

AS.snaive <- A.naive + S
accuracy(AS.snaive,tserie.test)[,"RMSE"]

## [1] 185.81

#autoplot(tserie.test) + autolayer(AS.sdrift) + autolayer(AS.smeanf) + autolayer(AS.snaive)

PT.stlfit.prev <- A.drift + S

```

PT, com a decomposição através da função stl.fit - snaive + drift

Q5 e Q6: Transformação BoxCox

```
PT.lam <- BoxCox.lambda(PT.train)
PT.transf <- BoxCox(PT.train, lambda = PT.lam)
```

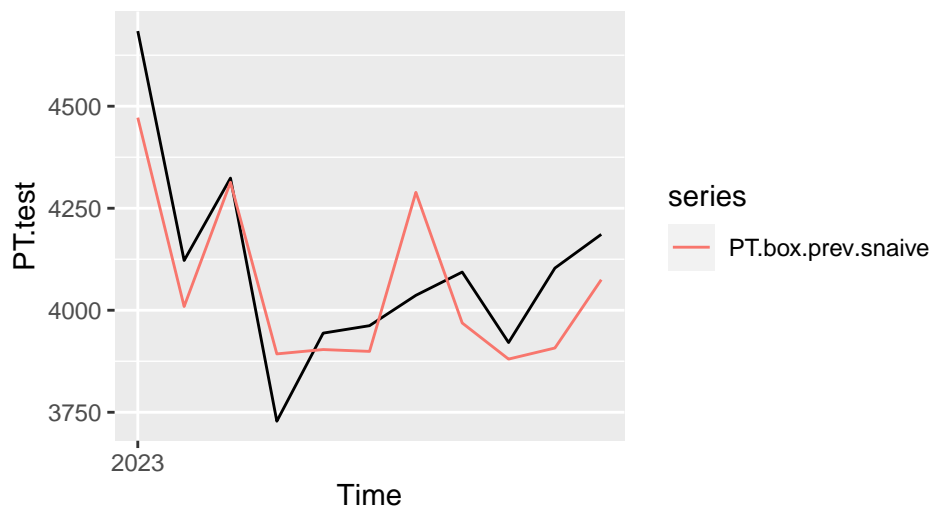
Q7: Previsões simples a partir da transf. BoxCox

```
#autoplot(PT.transf)

n <- length(PT.test)
PT.transf.prev.snaive <- snaive(PT.transf , h = n)$mean

PT.box.prev.snaive <- InvBoxCox(PT.transf.prev.snaive, lambda = PT.lam)

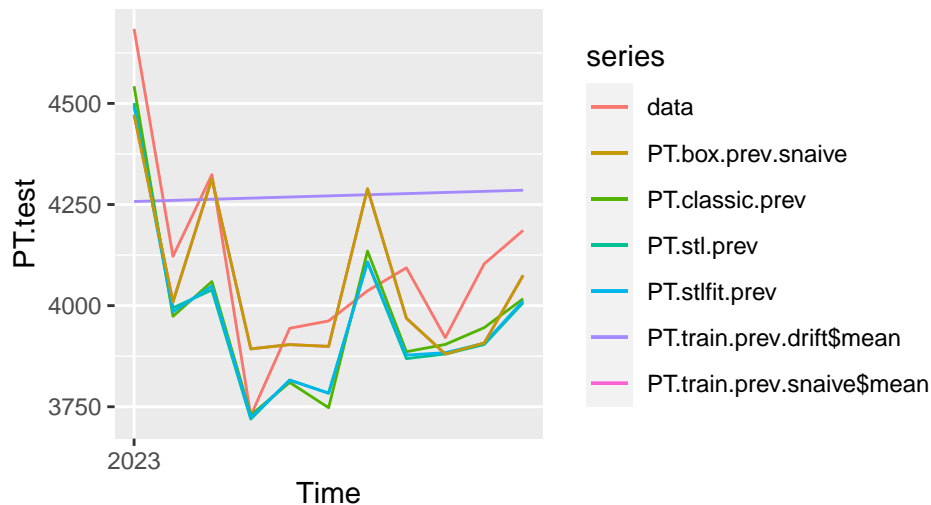
autoplot(PT.test) + autolayer(PT.box.prev.snaive)
```



É importante referir que todos os gráficos apresentaram tendência nula ou de baixa significância, mas uma sazonalidade bem significativa, daí ter usado sempre o método snaive de forma a prever os dados de teste.

Q8: Representação gráfica das previsões obtidas

```
autoplot(PT.test, series="data") +
  autolayer(PT.train.prev.snaive$mean) + autolayer(PT.train.prev.drift$mean) +
  autolayer(PT.classic.prev) +
  autolayer(PT.stl.prev) +
  autolayer(PT.stlfit.prev) +
  autolayer(PT.box.prev.snaive)
```



Q9: Qualidade das previsões

```
ttest <- PT.test

accuracy(PT.train.prev.snaive$mean , ttest)[,"RMSE"]
```

```
## [1] 142.1451
```

```
accuracy(PT.train.prev.drift$mean , ttest)[,"RMSE"]
```

```
## [1] 294.8171
```

```
accuracy(PT.classic.prev , ttest)[,"RMSE"]
```

```
## [1] 160.2161
```

```
accuracy(PT.stl.prev , ttest)[,"RMSE"]
```

```
## [1] 167.3951
```

```
accuracy(PT.stlfit.prev , ttest)[,"RMSE"]
```

```
## [1] 165.6724
```

```
accuracy(PT.box.prev.snaive , ttest)[,"RMSE"]
```

```
## [1] 142.1451
```

PT - PT.train.prev.snaive & PT.box.prev.snaive são as melhores

Q10: Transformar ou não transformar? (BoxCox)

Tendo em conta a alínea anterior, as transformações BoxCox utilizadas não obtiveram as melhores previsões com exceção no caso dos dados de Portugal, mas ainda assim ficou em empatado com a previsão simples snaive.

Pode-se observar que há exceção de Portugal, as melhores previsões foram obtidas a partir de aplicações às decomposições stl (Loess e stl.fit).

Tarefa 3 de 4

Pré-Requisitos

Definição das variáveis de treino e de teste

```
PT.train <- window(PT.ts, end = c(2022, 12))
PT.test <- window(PT.ts, start = c(2023, 1))
```

Encontrar as melhores decomposições utilizando a medida RMSE

Decomposição	adt	mul	stlper	stl7	stl15	stl20	stl.fit	
PT.train	139.503	139.355	142.7396	157.1439	146.3441	145.8309	140.1014	mul

```
PT.dec.mul <- decompose(PT.train, type = "multiplicative")
```

Definir os melhores ajustamentos

```
PT.fitT2 <- seasonal(PT.dec.mul) * trendcycle(PT.dec.mul)
```

Em que países (–.train) deve ser usado BoxCox?

Nenhuma das séries parece beneficiar significativamente de uma transformação BoxCox, com exceção de Portugal. Pelo facto da sua melhor decomposição ser multiplicativa é importante aplicar esta transformação caso se realize uma decomposição aditiva, por exemplo `stl`.

Q1: Modelo ets, com o critério aic

```
PT.dec.ets <- ets(PT.train, ic="aic")
```

Q2: Obter as previsões a partir do modelo ets

```
PT.prev.ets <- forecast(PT.dec.ets, h = length(PT.test))
```

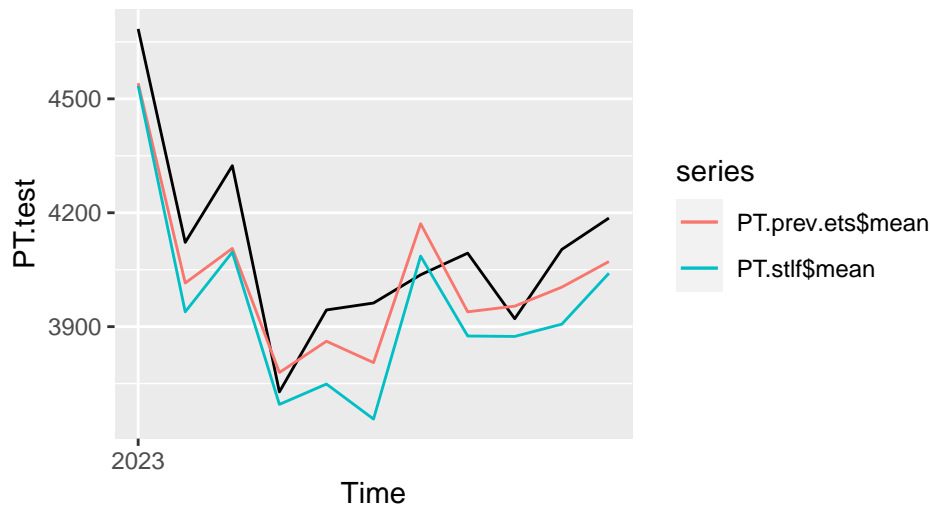
Q3: Previsões a partir da função stlf

```
PT.stlf <- stlf(PT.train, h = length(PT.test), lambda = "auto")
```

Utilizei unicamente a transformação BoxCox na previsão de Portugal pelas razões indicadas nos Pré-Requisitos.

Q4: Qualidade das previsões (ets vs stlf)

```
autoplot(PT.test) + autolayer(PT.prev.ets$mean) + autolayer(PT.stlf$mean)
```



```
t(data.frame(ets = accuracy(PT.prev.ets$mean, PT.test)[, "RMSE"],
              stlf = accuracy(PT.stlf$mean, PT.test)[, "RMSE"])))
```

```
##           [,1]
## ets  127.7184
## stlf 179.0117
```

No caso de Portugal, pela primeira vez, a função `ets + forecast` produziu a melhor previsão. Pergunto-me se terá sido pelo facto de ter sido utilizado uma transformação BoxCox na previsão com a função `stlf`.

Assim sendo vou verificar a qualidade da previsão caso a mesma fosse realizada sem a transformação:

```
accuracy(stlf(PT.train, h = length(PT.test))$mean,
          PT.test)[, "RMSE"]
```

```
## [1] 178.9882
```

Verifica-se que de qualquer forma, no caso de Portugal, a função `stlf` não produz as melhores previsões.

Q5: Comparar os resultados com a tarefa anterior

Selecione os resultados obtidos na Tarefa 2, questão 9:

```
PT.T2.train.prev.snaive <- 142.1451
```

```
t(data.frame(ets = accuracy(PT.prev.ets$mean, PT.test)[, "RMSE"],
              T2 = PT.T2.train.prev.snaive))
```

```
##           [,1]
## ets 127.7184
## T2  142.1451
```

No caso de Portugal, as melhores previsões foram obtidas através da aplicação da decomposição ETS aos dados de treino, com o modelo ETS(A,A,A), e posterior utilização da função `forecast` para realizar as previsões.

Tarefa 4 de 4

Pré-Requisitos

Melhores previsões para cada país das tarefas anteriores

- No caso de **Portugal**, as melhores previsões foram obtidas através da aplicação da decomposição ETS aos dados de treino, com o modelo ETS(A,A,A), e posterior utilização da função forecast para realizar as previsões (Tarefa 3). [RMSE = 127.7184]

```
PT.T3.prev.ets <- 127.7184
```

T3, Q1: decomposição ets para cada país

```
PT.dec.ets <- ets(PT.train, ic="aic")
```

T3, Q2: previsões a partir do ets para cada país

```
PT.prev.ets <- forecast(PT.dec.ets, h = length(PT.test))
```

Q1: Ajustar um modelo ARIMA

```
PT.ari <- auto.arima(PT.train, ic="aic")
```

```
PT.ari
```

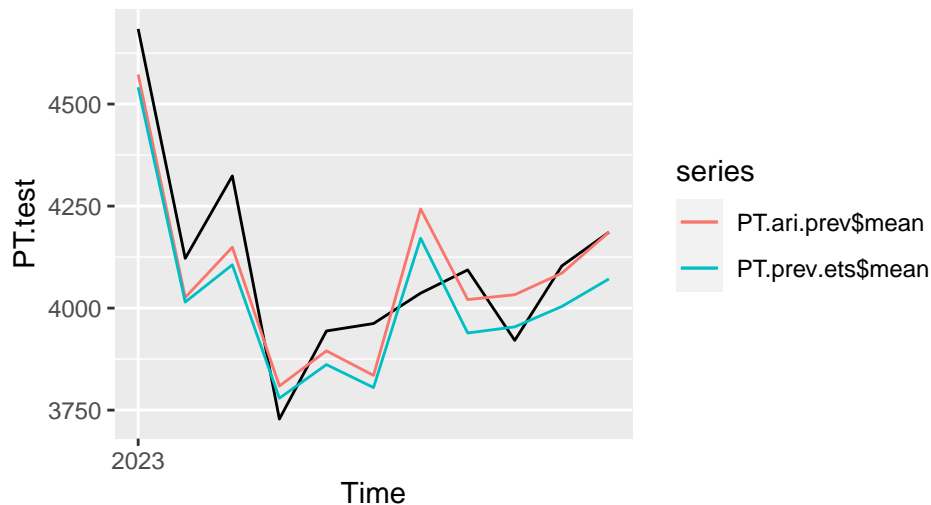
```
## Series: PT.train
## ARIMA(1,0,1)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1          ma1          sma1      drift
##          0.9133    -0.2761    -0.8832    4.3409
## s.e.    0.0415     0.1041     0.0909    1.8385
##
## sigma^2 = 26218:  log likelihood = -1100.22
## AIC=2210.44   AICc=2210.82   BIC=2226.06
```

Para a Portugal, o modelo ARIMA adequado, é na verdade um SARIMA, visto que os dados têm sazonalidade. Para além disso verifica-se que o modelo é ARIMA(1,0,1)(0,1,1)[12], ou seja, $p = 1$, $d = 0$, $q = 1$, $P = 0$, $D = 1$ e $Q = 1$.

Q2: Previsões a partir do modelo ARIMA

```
PT.ari.prev <- forecast(PT.ari, h = length(PT.test))
```

```
autoplot(PT.test) + autolayer(PT.ari.prev$mean) + autolayer(PT.prev.ets$mean)
```



Mais uma vez, ambas as previsões bastante próximas dos dados de teste, mas desta vez é mais difícil visualmente escolher a candidata a ser a melhor previsão.

Q3: Melhor previsão

```
accuracy(PT.ari.prev$mean, PT.test)[,c("RMSE", "MAE", "MAPE")]
```

```
##      RMSE      MAE      MAPE
## 112.025775 95.294679 2.322986
```

```
accuracy(PT.prev.ets$mean, PT.test)[,c("RMSE", "MAE", "MAPE")]
```

```
##      RMSE      MAE      MAPE
## 127.718429 117.673263 2.838124
```

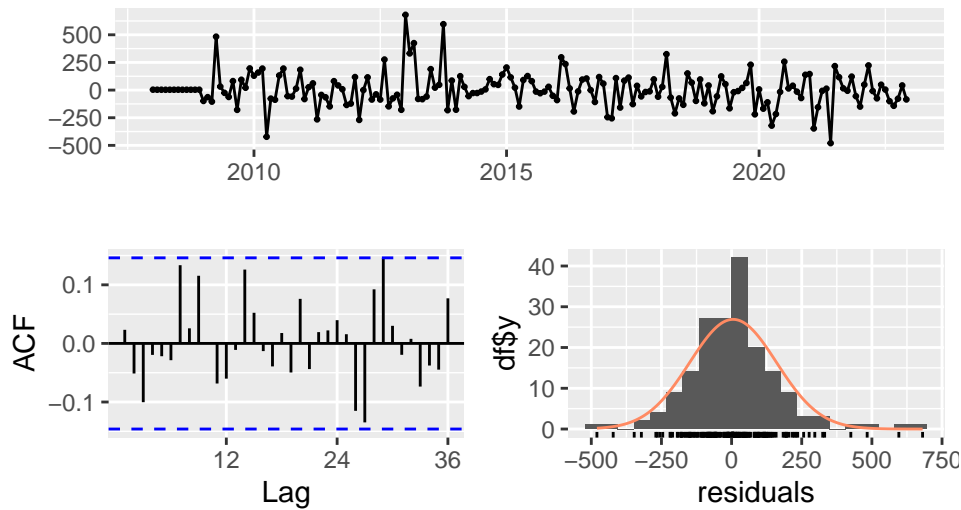
A partir das medidas de erro indicadas no enunciado, verifica-se, que no caso de Portugal, as melhores previsões foram obtidas a partir da previsão com o modelo ARIMA do exercício 1.

Q4: Previsões credíveis? (análise aos resíduos)

Analisar os resíduos dos modelos que originaram as melhores previsões:

```
checkresiduals(PT.ari)
```

Residuals from ARIMA(1,0,1)(0,1,1)[12] with drift



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1)(0,1,1)[12] with drift
## Q* = 17.131, df = 21, p-value = 0.7031
##
## Model df: 3.    Total lags used: 24
```

Pelo facto dos resíduos resultantes dos modelos que originaram as melhores previsões serem todos white noise (nível de significância de 1%), com exceção da Polónia, é de esperar que as melhores previsões sejam credíveis para esses.

Q5: Melhor ajustamento

```
accuracy(fitted(PT.ari), PT.train)[,c("RMSE", "MAE", "MAPE")]
```

```
##      RMSE      MAE      MAPE
## 154.556398 108.486564  2.815188
```

```
accuracy(fitted(PT.dec.ets), PT.train)[,c("RMSE", "MAE", "MAPE")]
```

```
##      RMSE      MAE      MAPE
## 151.394824 110.589361  2.920849
```

No caso de Portugal, o melhor modelo ajustado é o modelo ARIMA (MAE e MAPE menores) (T4 Q1). O melhor modelo ajustado corresponde às melhores previsões.

Repara-se que o melhor modelo ajustado, para estes países, foi sempre o modelo ARIMA, realizado na questão 1 desta tarefa. Contudo, também se repara que nem sempre o melhor modelo ajustado leva às melhores previsões.

Q6: Previsões para 2024, sem dados de teste

Relembrando:

País | melhor prev | melhor ajust

PT | ari | ari

vou ter por base as melhores previsões na escolha do modelo, contudo, no caso da Polónia, irei verificar se o modelo ARIMA produz resíduos que sejam ruído branco e irei usar o mesmo.

```
PT.ts.ari <- auto.arima(PT.ts, ic = "aic")
```

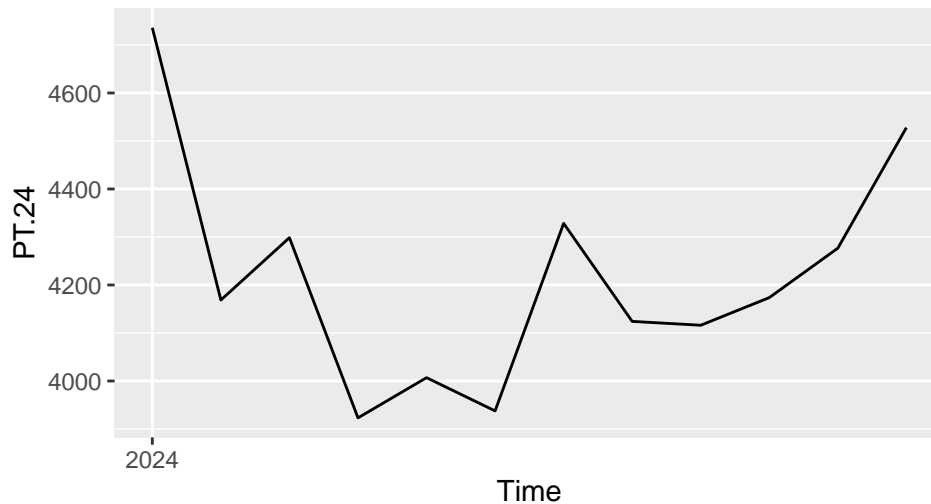
```
#tail(PT.ts)
```

```
PT.ts.prev <- forecast(PT.ts.ari, h = 12 + 1) #incluir DEZ23
```

```
(PT.24 <- window(PT.ts.prev$mean, start=c(2024,1), end = c(2024, 12)))
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2024 4735.860 4168.619 4298.528 3923.127 4006.924 3937.715 4328.313 4124.111
##           Sep      Oct      Nov      Dec
## 2024 4116.134 4173.683 4276.729 4527.690
```

```
autoplot(PT.24)
```



Atendendo ao consumo de eletrecidade em Portugal no ano de 2024, prevê-se que o mesmo atinja o seu máximo do ano em janeiro, provavelmente devido ao inverno, uma descida nos meses seguintes até junho, havendo uma subida mais significativa em julho e por fim uma tendência de aumento ligeiro até ao final do ano.