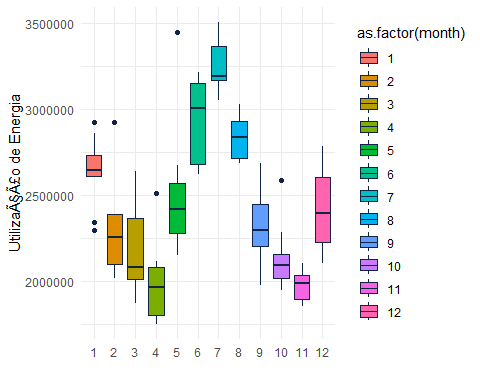
**Resolução dos exercícios**

Thomaz Rodrigues Lima

# Questão 1

### a)

ggplot(energy) +  
 aes(x = as.factor(month), y = energy\_demand, fill = as.factor(month)) +  
 geom\_boxplot(colour = "#112446") +  
 xlab("") +  
 ylab(iconv("Utilização de Energia", from = "latin1", to = "UTF-8")) +  
 theme\_minimal()

 ### b) De acordo com o gráfico podemos vizualizar que o maior consumo da-se nos mezes de verão Europeu, com um especial destaque para o mês de julho. Outrem, sendo o mês que menos se consome energia, abril.

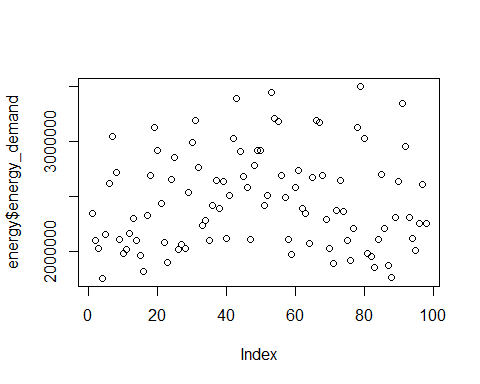
### c)

class(energy)

## [1] "data.frame"

### d)

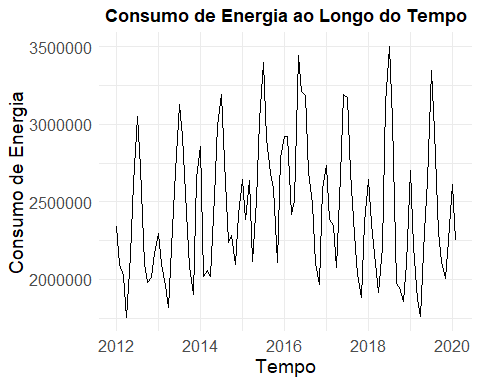
plot(energy$energy\_demand)

 ### e)

ts.energy<-ts(energy$energy\_demand, frequency = 12, start = c(2012,1))

### f)

ggplot2::autoplot(ts.energy) +  
 ggtitle("Consumo de Energia ao Longo do Tempo") +  
 xlab("Tempo") +  
 ylab("Consumo de Energia") +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(hjust = 0.5, face = "bold"),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 14)  
 )



# Questão 2

### a)

ts.gas <-ts(NATURALGAS, frequency = 12, start = c(2000,1))

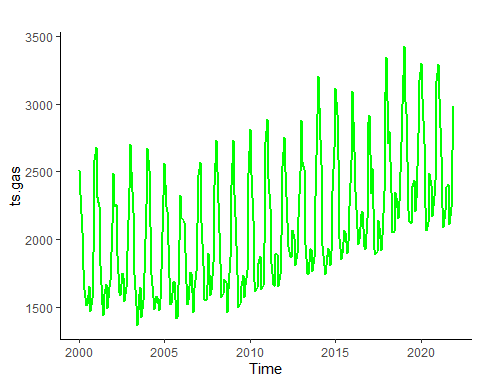
### b)

length(gas)

## [1] 476

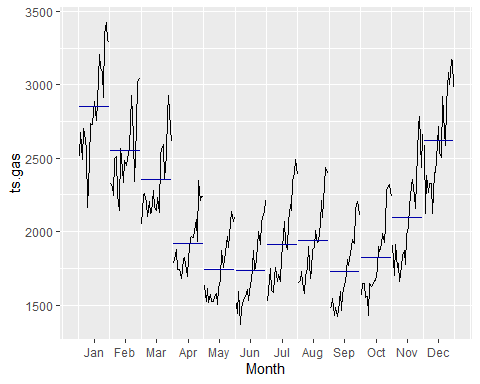
### c)

autoplot(ts.gas, size=0.8, col="green")+  
 theme\_classic()

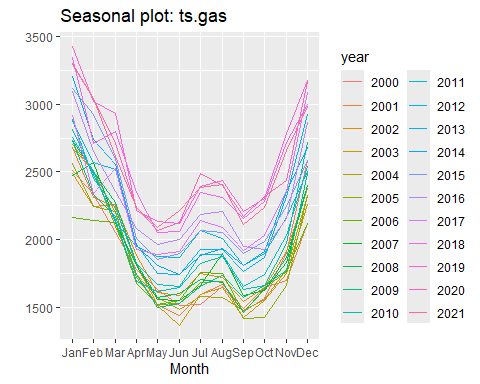
 Apresenta Sazonalidade e uma certa tendência Crescente

### d)

ggsubseriesplot(ts.gas)

 ### d2)

ggseasonplot(ts.gas)

 Observa-se que de longe o período que mais se usa o gás é em Janeiro, que coincide com inverno no Hemisfério Sul. Já o período que menos se usa é no Verão Europeu

# Questão 3

### a)

ts.ph <- ts(ph, frequency = 4, start = c(1988,4))

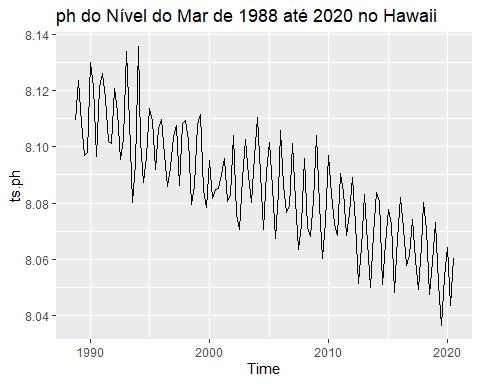
### b)

length(ts.ph)

## [1] 128

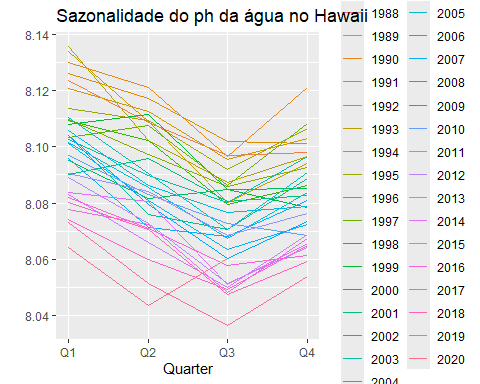
### c)

autoplot(ts.ph)+  
 ggtitle("ph do Nível do Mar de 1988 até 2020 no Hawaii")

 Podemos observar que a série temporal apresenta uma tendência decrescente e uma sazonalidade.

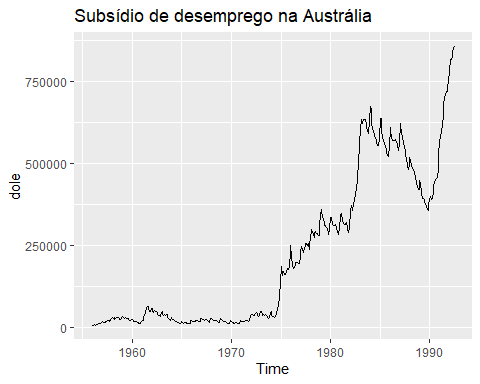
###d)

ggseasonplot(ts.ph)+  
 ggtitle("Sazonalidade do ph da água no Hawaii")



# Questão 4

autoplot(dole)+  
 ggtitle("Subsídio de desemprego na Austrália")

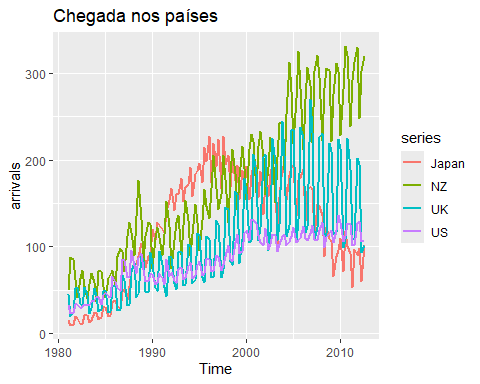


xlab("")

## $x  
## [1] ""  
##   
## attr(,"class")  
## [1] "labels"

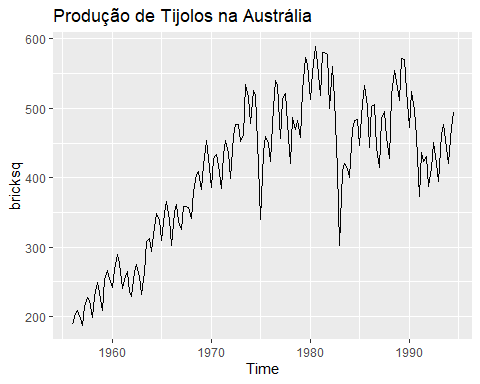
# Questão 5

autoplot(arrivals, size = 0.8) +  
 ggtitle("Chegada nos países")

 Observa-se que o gráfico tem aumentado em média desde de o primeiro trimestre de 1981.

Enquanto os outros 3 países, o Japão há um aumento até os anos 90 e depois há uma queda.

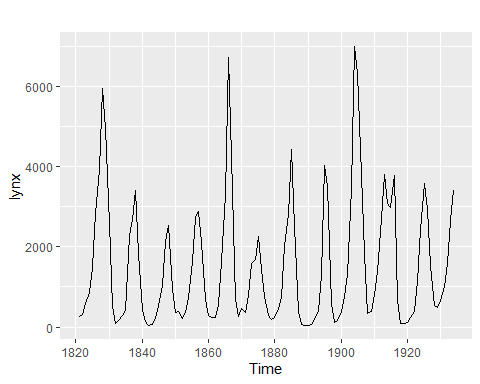
autoplot(bricksq)+  
 ggtitle("Produção de Tijolos na Austrália")



xlab("")

## $x  
## [1] ""  
##   
## attr(,"class")  
## [1] "labels"

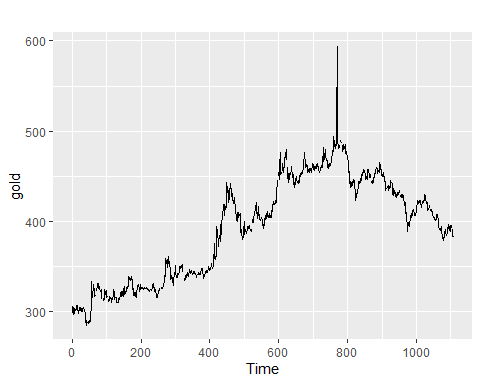
autoplot(lynx)



# Questão 6

### b)

autoplot(gold)

 Podemos observar uma tendência crescente até meados do dia 800, após essa data, uma certa tendência decrescente, uma falta de sazonalidade. Um grande Outlayer e alguns dias sem dados

### c)

frequency(gold)

## [1] 1

### d)

which.max(gold)

## [1] 770

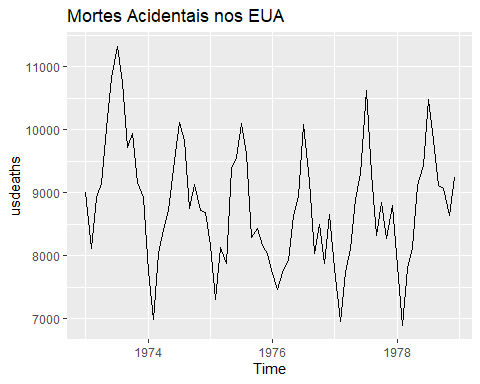
### e)

anyNA(gold)

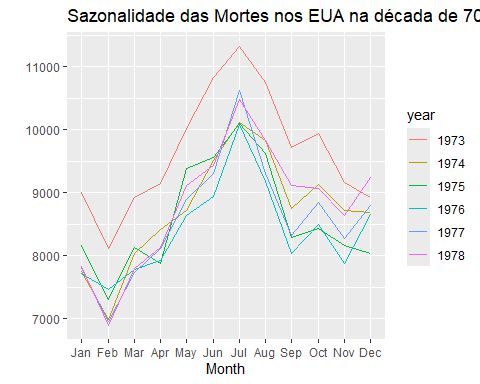
## [1] TRUE

# Questão 7

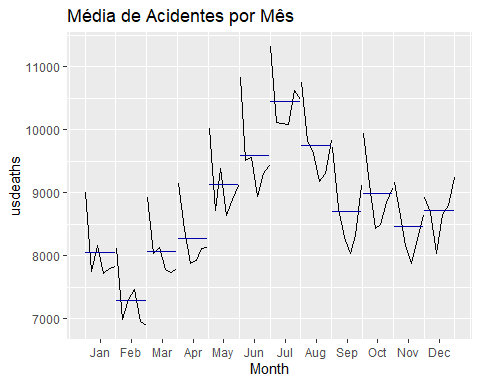
autoplot(usdeaths)+  
 ggtitle("Mortes Acidentais nos EUA")



ggseasonplot(usdeaths)+  
 ggtitle("Sazonalidade das Mortes nos EUA na década de 70")



ggsubseriesplot(usdeaths)+  
 ggtitle("Média de Acidentes por Mês")

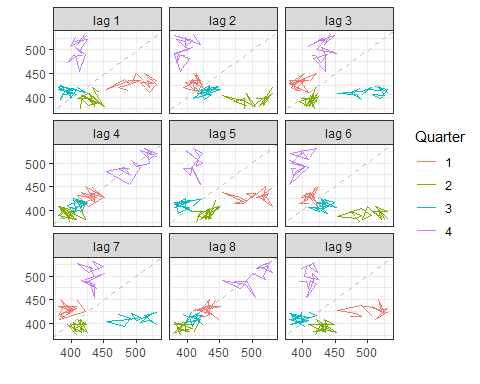


title: “**Resolução dos exercícios 2**”

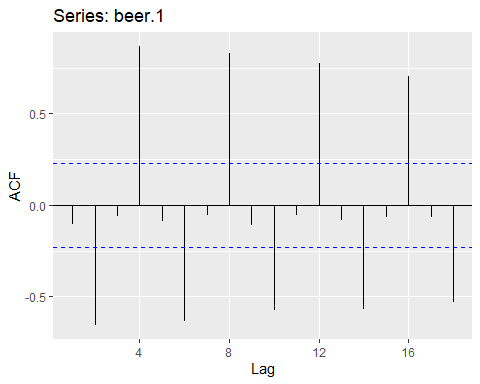
# Questão 1

###) a

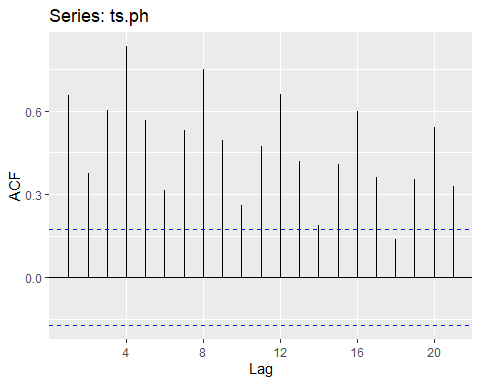
beer.1 <- window(ausbeer, start = 1992)  
gglagplot(beer)+theme\_bw()



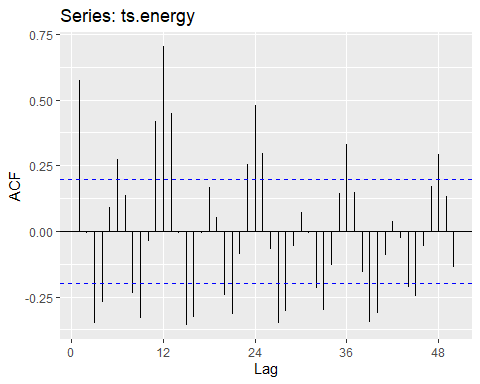
ggAcf(beer.1)

 É uma série trimestral, com uma correlação de 4 em 4

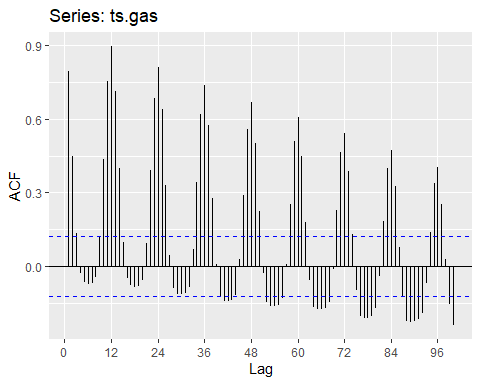
ggAcf(ts.ph)



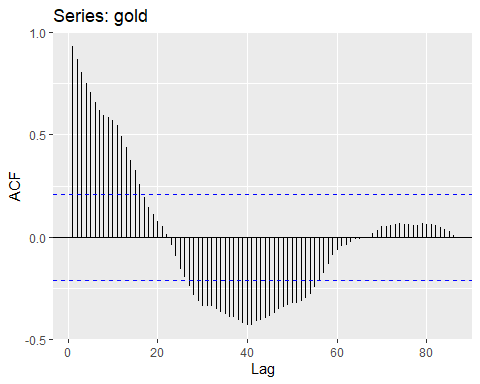
ggAcf(ts.energy, lag.max = 50)

 Aqui de 12 em 12, que indica que a série é Mensal

ggAcf(ts.gas, lag.max = 100)

 Mais uma vez de 12 em 12, ou seja, mensal

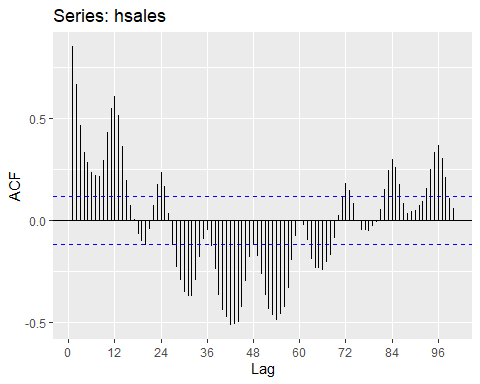
ggAcf(gold,lag.max = 200)

 Claramente a série não tem sazonalidade

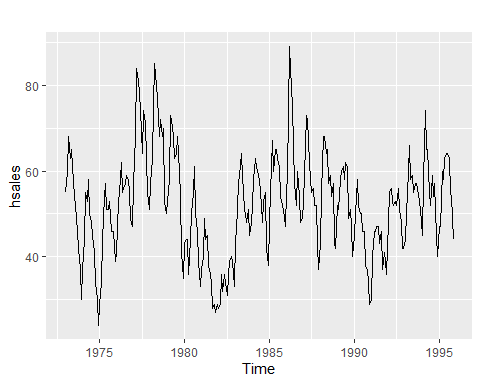
# Questão 1

### a)

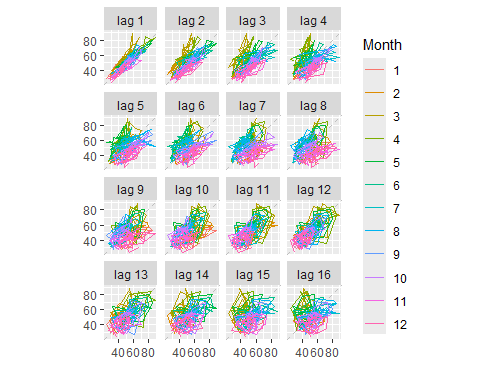
ggAcf(hsales, lag.max = 100)



autoplot(hsales)

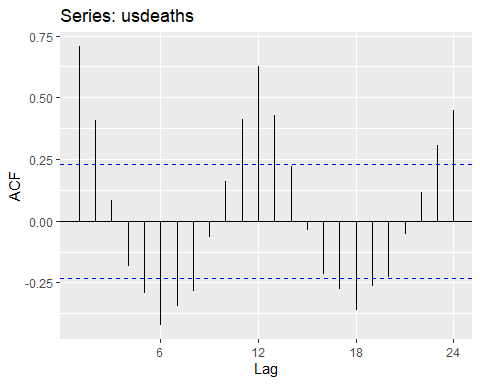


gglagplot(hsales)

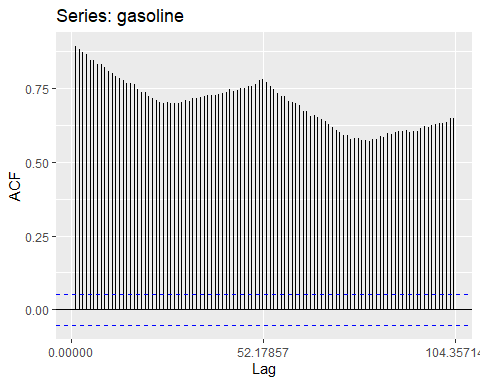


É um fenômeno ciclico, mas não sazonal

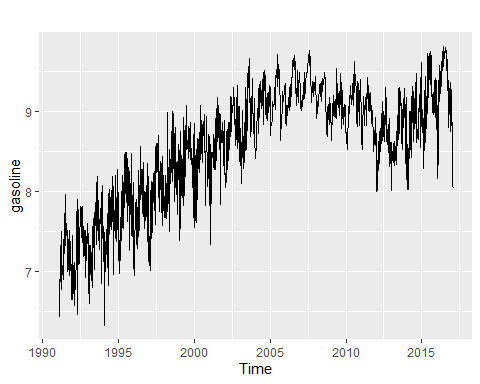
ggAcf(usdeaths)

 Tem sazonalidade, mas não uma tendência muito forte

ggAcf(gasoline)



autoplot(gasoline)



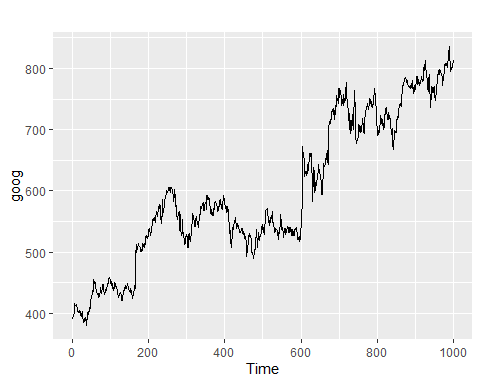
Tem uma forte tendência, mas não uma sazonalidade

# Questão 2

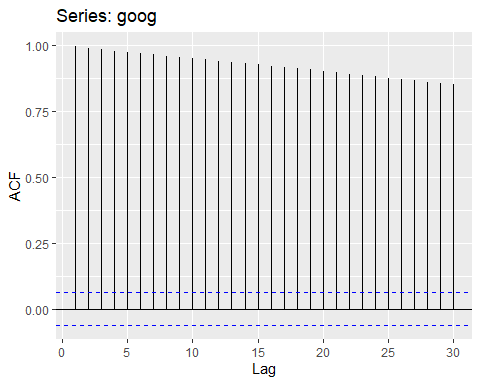
1-B, Sazonalidade com uma certa tendência, mas ambas mais fortes 2-A, Sazonalidade com uma certa tendência 3-D, Tendência muito forte 4-C, Pouca sazonalidade e tendência

# Questão 3

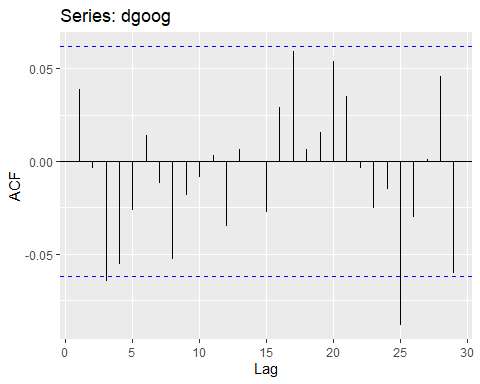
autoplot(goog)



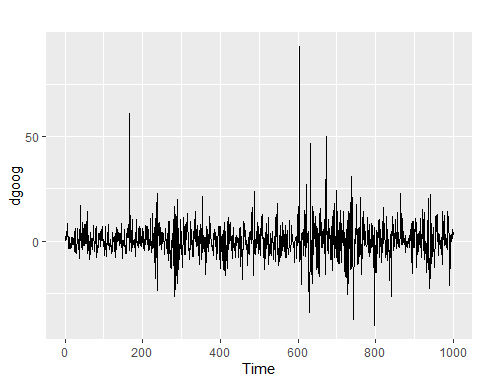
ggAcf(goog)



dgoog <- diff(goog)  
ggAcf(dgoog)



autoplot(dgoog)



Box.test(dgoog,lag=10,type="Ljung-Box")

##   
## Box-Ljung test  
##   
## data: dgoog  
## X-squared = 13.123, df = 10, p-value = 0.2169

A serie Dgoog não apresenta auto-correlação

# Questão 4

### a) Será que todas as figuras indicam que os dados são ruído branco?

Sim

### b) Existem diferenças entre as figuras? Comente.

Sim, podemos observar que da série x1 até a x2 está a diminuir a amplitude desse intervalo, o mesmo de x2 para x3.

### c) Porque é que os valores críticos estão a distâncias diferentes da média de

zero? Porque são as autocorrelações diferentes em cada figura quando cada uma delas se refere ao ruído branco?

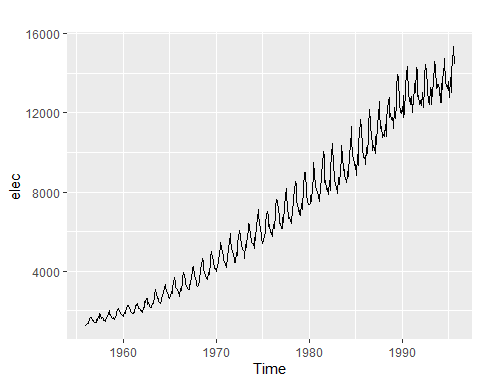
A dimensão é diferente, o intervalo diminui pois a dimensão da série está a aumentar

# Questão 5

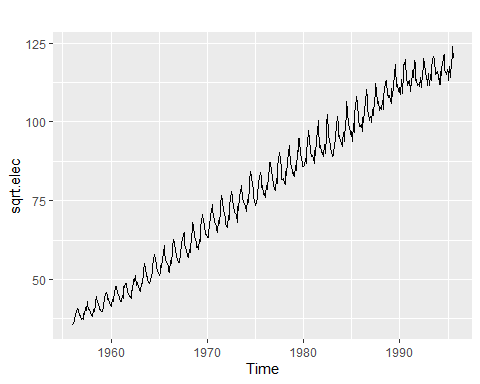
1-C 2-E 3-A 4-B 5-D

title: “**Resolução dos exercícios 3**”

autoplot(elec)

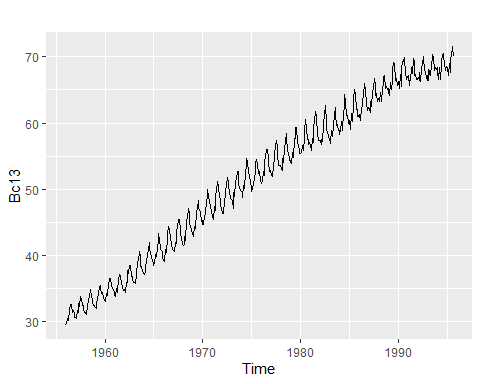
 -aplicando a raiz quadrada

sqrt.elec<-sqrt(elec)  
autoplot(sqrt.elec)

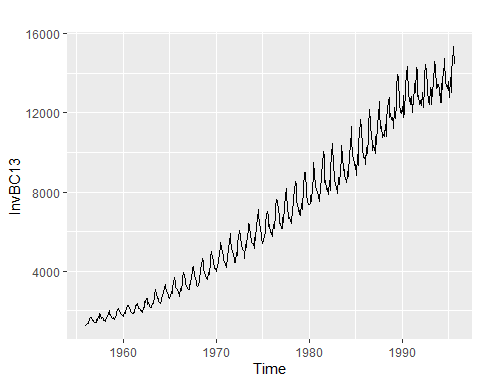
 - A aplitude sazonal ficou mais constante. Serve exatamente para estabilizar a amplitude sazonal, estabilizar a variância

## BoxCox

Bc13<-BoxCox(elec,lambda=1/3)  
autoplot(Bc13)



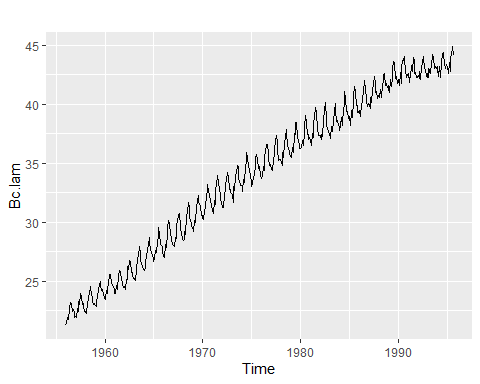
InvBC13<-InvBoxCox(Bc13, lambda = 1/3)  
autoplot(InvBC13)

 ## Estimar o melhor lambda

lam<-BoxCox.lambda(elec)  
lam

## [1] 0.2654076

Bc.lam<-BoxCox(elec, lambda = lam)  
autoplot(Bc.lam)

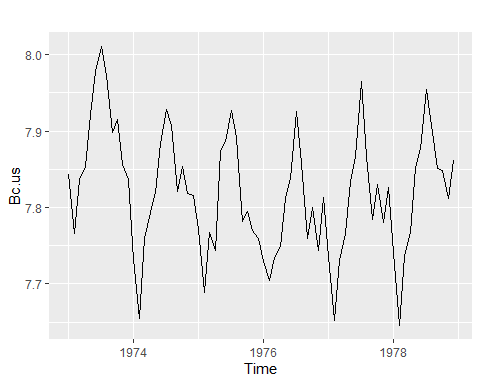


# Questão 1

lam.us<-BoxCox.lambda(usdeaths)  
lam.us

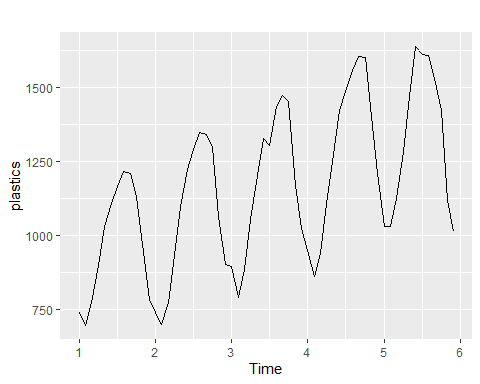
## [1] -0.03363775

Bc.us<-BoxCox(usdeaths, lambda = lam.us)  
autoplot(Bc.us)

 # Questão 2

### a)

autoplot(plastics)

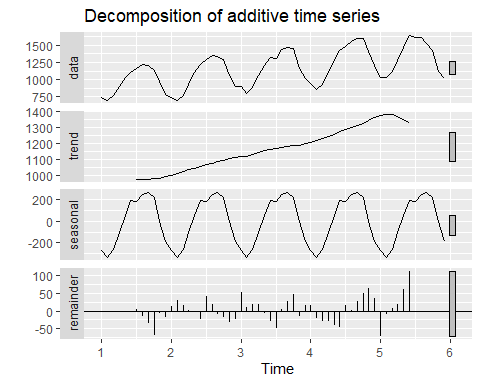
 Sim! com um S=12 meses, com uma tendência positiva

### b)

Aplicar a função decompose(serie, type=“additive”)

plastics.decAd <- decompose(plastics, type = "additive")

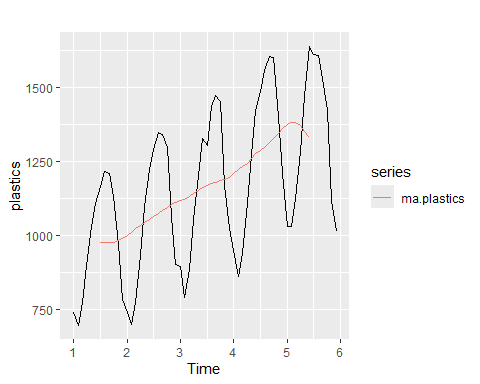
autoplot(plastics.decAd)

 ### Estimando a tendência por médias móveis de ordem 4

ma.plastics <- ma(plastics, 12)

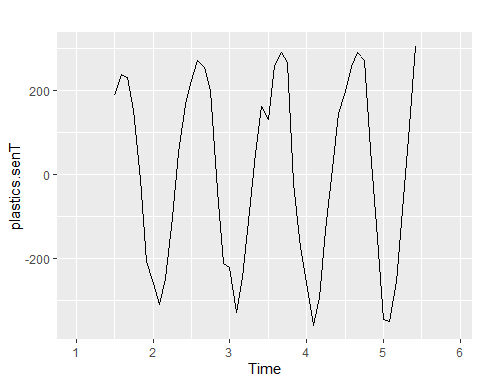
autoplot(plastics)+  
 autolayer(ma.plastics)

## Warning: Removed 12 rows containing missing values or values outside the scale range  
## (`geom\_line()`).



### Eliminar a tendência obtida por ma

plastics.senT <- plastics - ma.plastics  
  
autoplot(plastics.senT)

 ### Para a decomposição aditiva para que a variáve

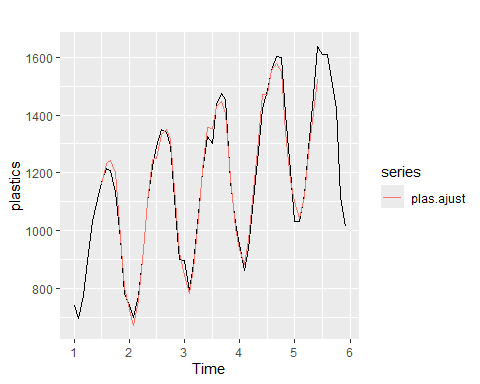
head(plastics.decAd)

## $x  
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1 742 697 776 898 1030 1107 1165 1216 1208 1131 971 783  
## 2 741 700 774 932 1099 1223 1290 1349 1341 1296 1066 901  
## 3 896 793 885 1055 1204 1326 1303 1436 1473 1453 1170 1023  
## 4 951 861 938 1109 1274 1422 1486 1555 1604 1600 1403 1209  
## 5 1030 1032 1126 1285 1468 1637 1611 1608 1528 1420 1119 1013  
##   
## $seasonal  
## Jan Feb Mar Apr May Jun  
## 1 -273.203993 -339.933160 -263.099826 -104.943576 56.504340 193.316840  
## 2 -273.203993 -339.933160 -263.099826 -104.943576 56.504340 193.316840  
## 3 -273.203993 -339.933160 -263.099826 -104.943576 56.504340 193.316840  
## 4 -273.203993 -339.933160 -263.099826 -104.943576 56.504340 193.316840  
## 5 -273.203993 -339.933160 -263.099826 -104.943576 56.504340 193.316840  
## Jul Aug Sep Oct Nov Dec  
## 1 183.441840 254.952257 265.316840 221.139757 -4.953993 -188.537326  
## 2 183.441840 254.952257 265.316840 221.139757 -4.953993 -188.537326  
## 3 183.441840 254.952257 265.316840 221.139757 -4.953993 -188.537326  
## 4 183.441840 254.952257 265.316840 221.139757 -4.953993 -188.537326  
## 5 183.441840 254.952257 265.316840 221.139757 -4.953993 -188.537326  
##   
## $trend  
## Jan Feb Mar Apr May Jun Jul  
## 1 NA NA NA NA NA NA 976.9583  
## 2 1000.4583 1011.2083 1022.2917 1034.7083 1045.5417 1054.4167 1065.7917  
## 3 1117.3750 1121.5417 1130.6667 1142.7083 1153.5833 1163.0000 1170.3750  
## 4 1208.7083 1221.2917 1231.7083 1243.2917 1259.1250 1276.5833 1287.6250  
## 5 1374.7917 1382.2083 1381.2500 1370.5833 1351.2500 1331.2500 NA  
## Aug Sep Oct Nov Dec  
## 1 977.0417 977.0833 978.4167 982.7083 990.4167  
## 2 1076.1250 1084.6250 1094.3750 1103.8750 1112.5417  
## 3 1175.5000 1180.5417 1185.0000 1190.1667 1197.0833  
## 4 1298.0417 1313.0000 1328.1667 1343.5833 1360.6250  
## 5 NA NA NA NA NA  
##   
## $random  
## Jan Feb Mar Apr May Jun Jul  
## 1 NA NA NA NA NA NA 4.599826  
## 2 13.745660 28.724826 14.808160 2.235243 -3.046007 -24.733507 40.766493  
## 3 51.828993 11.391493 17.433160 17.235243 -6.087674 -30.316840 -50.816840  
## 4 15.495660 -20.358507 -30.608507 -29.348090 -41.629340 -47.900174 14.933160  
## 5 -71.587674 -10.275174 7.849826 19.360243 60.245660 112.433160 NA  
## Aug Sep Oct Nov Dec  
## 1 -15.993924 -34.400174 -68.556424 -6.754340 -18.879340  
## 2 17.922743 -8.941840 -19.514757 -32.921007 -23.004340  
## 3 5.547743 27.141493 46.860243 -15.212674 14.453993  
## 4 2.006076 25.683160 50.693576 64.370660 36.912326  
## 5 NA NA NA NA NA  
##   
## $figure  
## [1] -273.203993 -339.933160 -263.099826 -104.943576 56.504340 193.316840  
## [7] 183.441840 254.952257 265.316840 221.139757 -4.953993 -188.537326  
##   
## $type  
## [1] "additive"

Fazer o gráfico com a série temoral e os valores ajustados/estimados

plas.ajust<-plastics.decAd$seasonal+plastics.decAd$trend  
  
autoplot(plastics)+  
 autolayer(plas.ajust)

## Warning: Removed 12 rows containing missing values or values outside the scale range  
## (`geom\_line()`).



Como determinar a componente residual, a diferença entre a série temporal e os valores ajustados

rt.plastic<-plastics-plas.ajust  
rt.plastic

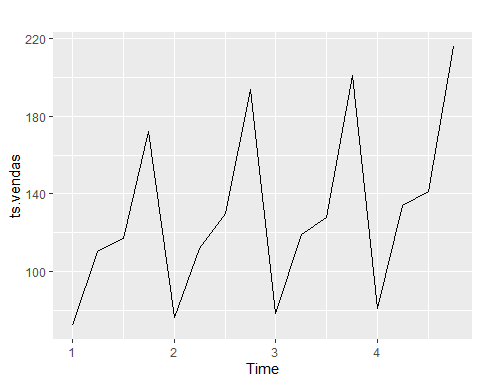
## Jan Feb Mar Apr May Jun Jul  
## 1 NA NA NA NA NA NA 4.599826  
## 2 13.745660 28.724826 14.808160 2.235243 -3.046007 -24.733507 40.766493  
## 3 51.828993 11.391493 17.433160 17.235243 -6.087674 -30.316840 -50.816840  
## 4 15.495660 -20.358507 -30.608507 -29.348090 -41.629340 -47.900174 14.933160  
## 5 -71.587674 -10.275174 7.849826 19.360243 60.245660 112.433160 NA  
## Aug Sep Oct Nov Dec  
## 1 -15.993924 -34.400174 -68.556424 -6.754340 -18.879340  
## 2 17.922743 -8.941840 -19.514757 -32.921007 -23.004340  
## 3 5.547743 27.141493 46.860243 -15.212674 14.453993  
## 4 2.006076 25.683160 50.693576 64.370660 36.912326  
## 5 NA NA NA NA NA

# Exercício 3

vendas<-c(72,110,117,172,76,112,130,194,78,119,128,201,81,134,141,216)  
  
ts.vendas <- ts(vendas, frequency=4)  
ts.vendas

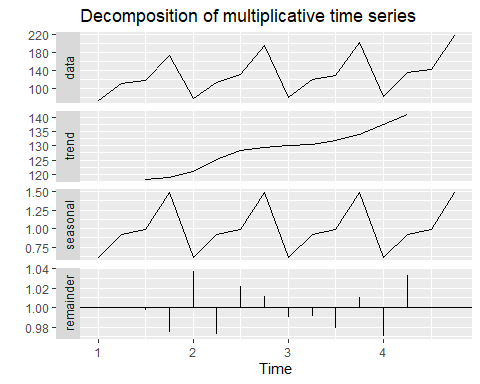
## Qtr1 Qtr2 Qtr3 Qtr4  
## 1 72 110 117 172  
## 2 76 112 130 194  
## 3 78 119 128 201  
## 4 81 134 141 216

autoplot(ts.vendas)

 Sazonalidade e uma pequena tendência ligereiramente crescente

### b)

vendas.decM <- decompose(ts.vendas, type = "multiplicative")  
  
autoplot(vendas.decM)



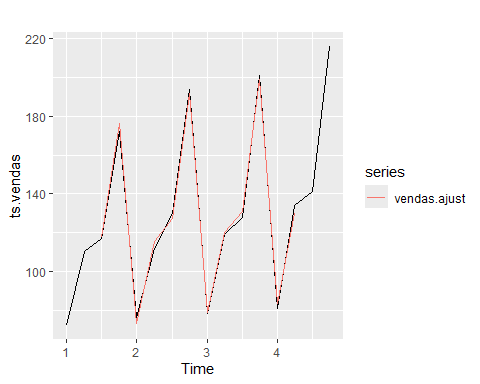
vendas.decM

## $x  
## Qtr1 Qtr2 Qtr3 Qtr4  
## 1 72 110 117 172  
## 2 76 112 130 194  
## 3 78 119 128 201  
## 4 81 134 141 216  
##   
## $seasonal  
## Qtr1 Qtr2 Qtr3 Qtr4  
## 1 0.6063128 0.9190686 0.9921213 1.4824973  
## 2 0.6063128 0.9190686 0.9921213 1.4824973  
## 3 0.6063128 0.9190686 0.9921213 1.4824973  
## 4 0.6063128 0.9190686 0.9921213 1.4824973  
##   
## $trend  
## Qtr1 Qtr2 Qtr3 Qtr4  
## 1 NA NA 118.250 119.000  
## 2 120.875 125.250 128.250 129.375  
## 3 130.000 130.625 131.875 134.125  
## 4 137.625 141.125 NA NA  
##   
## $random  
## Qtr1 Qtr2 Qtr3 Qtr4  
## 1 NA NA 0.9972865 0.9749617  
## 2 1.0370039 0.9729541 1.0216948 1.0114804  
## 3 0.9895882 0.9912261 0.9783240 1.0108633  
## 4 0.9707133 1.0331251 NA NA  
##   
## $figure  
## [1] 0.6063128 0.9190686 0.9921213 1.4824973  
##   
## $type  
## [1] "multiplicative"  
##   
## attr(,"class")  
## [1] "decomposed.ts"

### c)

vendas.ajust<-vendas.decM$seasonal\*vendas.decM$trend  
  
autoplot(ts.vendas)+  
 autolayer(vendas.ajust)

## Warning: Removed 4 rows containing missing values or values outside the scale range  
## (`geom\_line()`).



rt.vendas<-ts.vendas/vendas.ajust  
rt.vendas

## Qtr1 Qtr2 Qtr3 Qtr4  
## 1 NA NA 0.9972865 0.9749617  
## 2 1.0370039 0.9729541 1.0216948 1.0114804  
## 3 0.9895882 0.9912261 0.9783240 1.0108633  
## 4 0.9707133 1.0331251 NA NA

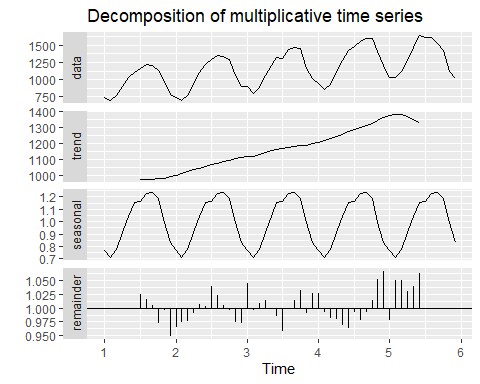
vendas.decM$random

## Qtr1 Qtr2 Qtr3 Qtr4  
## 1 NA NA 0.9972865 0.9749617  
## 2 1.0370039 0.9729541 1.0216948 1.0114804  
## 3 0.9895882 0.9912261 0.9783240 1.0108633  
## 4 0.9707133 1.0331251 NA NA

# Questão 2

plastics\_decM<-decompose(plastics, type="multiplicative")

autoplot(plastics\_decM)

 ### c)

round(plastics\_decM$figure,2)

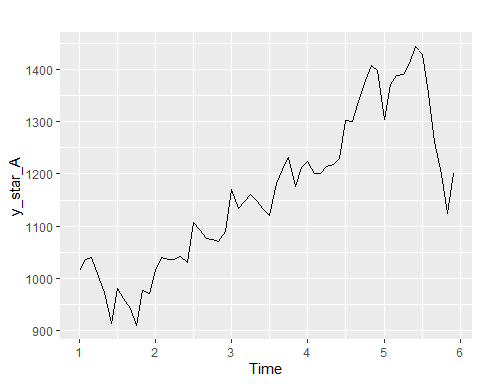
## [1] 0.77 0.71 0.78 0.91 1.04 1.16 1.16 1.23 1.23 1.19 0.99 0.83

Verificou-se em Agosto e setembro um aumento de 23%

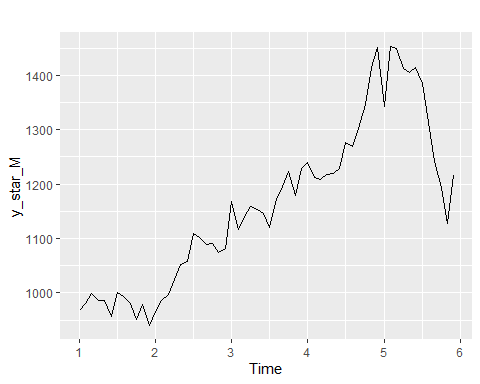
Houve um decrescimo de 29% no mês de fevereiro

### d)

y\_star\_A<-plastics-plastics.decAd$seasonal  
autoplot(y\_star\_A)

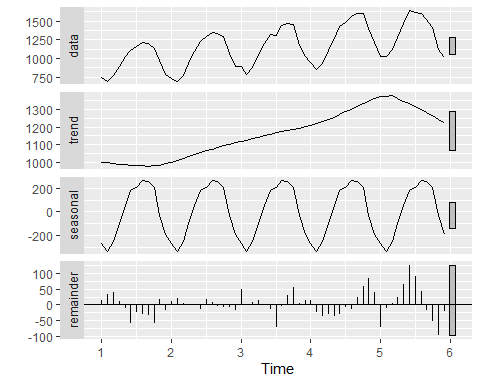


y\_star\_M<-plastics/plastics\_decM$seasonal  
  
autoplot(y\_star\_M)



### e)

plastics.stl<-stl(plastics, s.window = "periodic")  
  
autoplot(plastics.stl)

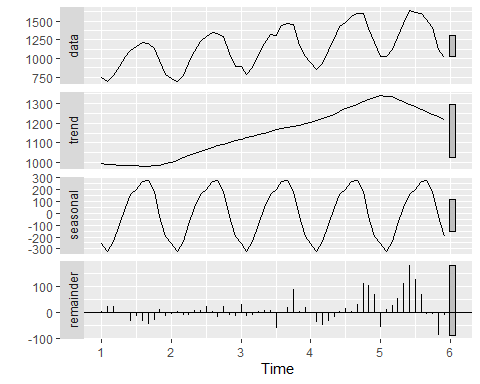


test.Outliers.STL<-function (x)   
{  
 stlR <- stl(x, s.window = "per", robust = TRUE)  
 iO <- which(stlR$weights < 1e-08)  
 out <- ifelse(length(iO) == 0, FALSE, TRUE)  
 return(out)  
}  
  
test<-test.Outliers.STL(plastics)  
  
test

## [1] TRUE

Para testar se existem outliers vamos usar a função “test.Outliers.ST”

rob<-test.Outliers.STL(plastics)  
plastics.stlR<-stl(plastics,s.window = "periodic", robust = rob)  
  
autoplot(plastics.stlR)



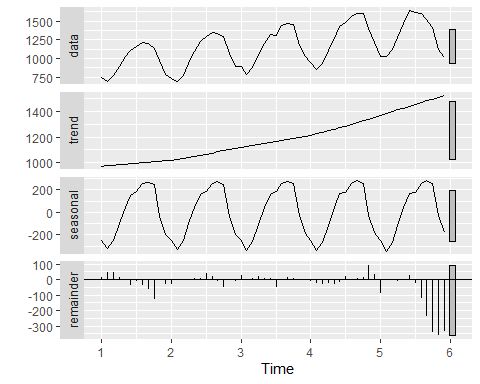
Com a FUNÇÃO STL FIT

stl.fit <- function(y,rob,k){  
 nextodd <- function(x){  
 x <- round(x)  
 if (x%%2 == 0)   
 x <- x + 1  
 as.integer(x)  
 }  
 aux <- c()  
 fit <- stl(y, s.window = "periodic", robust = rob)  
 fit2 <- fit$time.series[,"seasonal"] + fit$time.series[,"trend"]  
 m1 <- accuracy(fit2,y)[k]   
 aux$measure <- m1  
 aux$stl <- fit  
 len <- min(5\*frequency(y), length(y))  
 i\_range <- seq(7,len,2)  
 for (i in i\_range){  
 t.win <- nextodd(ceiling(1.5\*frequency(y)/(1-1.5/i)))  
 kk\_range <- seq(t.win,len,2)  
 for (kk in kk\_range){  
 for (t in 0:1){  
 for (w in 0:1){  
 fit <- stl(y,  
 s.window = i,  
 t.window = kk,  
 s.degree = t,  
 t.degree = w,  
 robust=rob)  
 fit2 <- fit$time.series[,"seasonal"] + fit$time.series[,"trend"]  
 m2 <- accuracy(fit2,y)[k]   
 if (m2 < m1){  
 m1 <- m2  
 aux$measure <- m1  
 aux$stl <- fit  
 }  
 }  
 }  
 }  
 }  
 aux  
}

plastics.stlfit <- stl.fit(plastics, rob = rob, k = 1)  
  
plastics.stlfit$measure

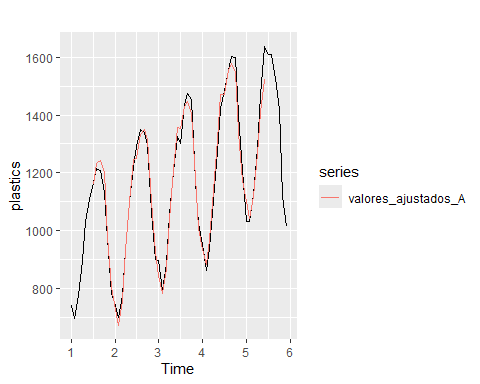
## [1] -28.55759

autoplot(plastics.stlfit$stl)

 ### f)

valores\_ajustados\_A<-plastics-remainder(plastics.decAd)  
  
autoplot(plastics)+  
 autolayer(valores\_ajustados\_A)

## Warning: Removed 12 rows containing missing values or values outside the scale range  
## (`geom\_line()`).

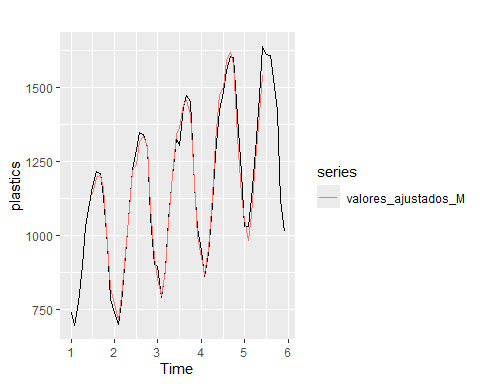


accuracy(plastics, valores\_ajustados\_A)[,c("RMSE", "MAE")]

## RMSE MAE   
## 35.41956 27.82429

valores\_ajustados\_M<-plastics/remainder(plastics\_decM)  
  
autoplot(plastics)+  
 autolayer(valores\_ajustados\_M)

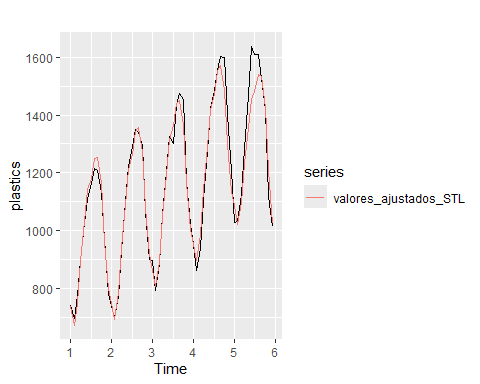
## Warning: Removed 12 rows containing missing values or values outside the scale range  
## (`geom\_line()`).



accuracy(plastics, valores\_ajustados\_M)[,c("RMSE", "MAE")]

## RMSE MAE   
## 34.78578 27.53742

valores\_ajustados\_STL<-plastics-remainder(plastics.stlR)  
  
autoplot(plastics)+  
 autolayer(valores\_ajustados\_STL)



accuracy(plastics, valores\_ajustados\_STL)[,c("RMSE", "MAE")]

## RMSE MAE   
## 48.54421 31.88731

autoplot(plastics)+  
 autolayer(valores\_ajustados\_A)+  
 autolayer(valores\_ajustados\_M)+  
 autolayer(valores\_ajustados\_STL)

## Warning: Removed 12 rows containing missing values or values outside the scale range  
## (`geom\_line()`).  
## Removed 12 rows containing missing values or values outside the scale range  
## (`geom\_line()`).

