

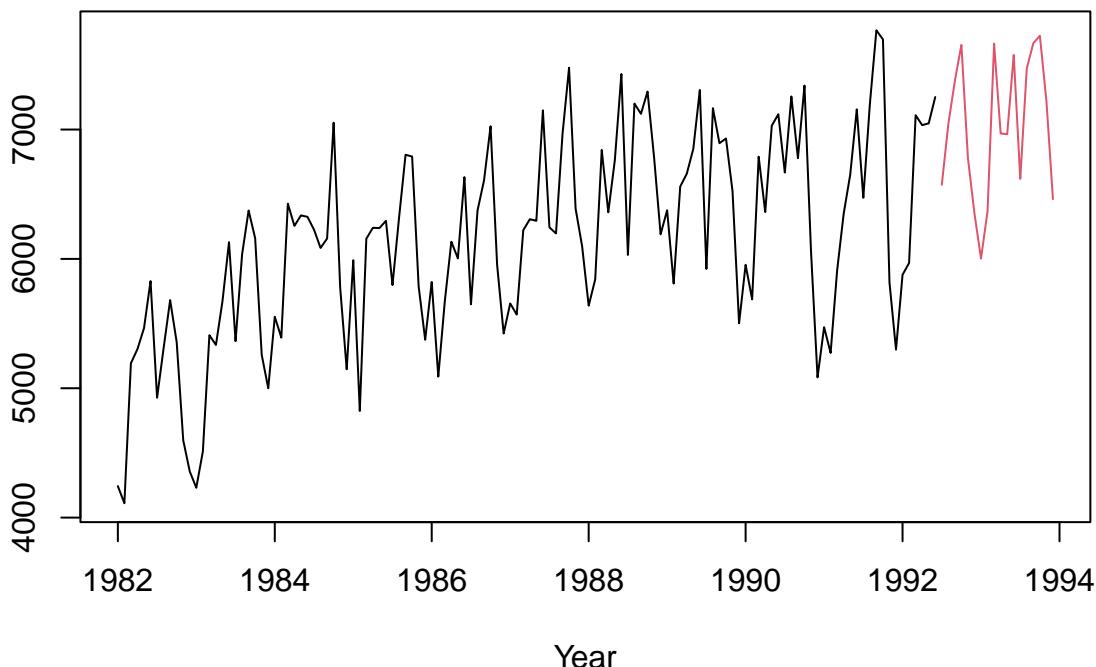
trabalho_series_1

Introducao

A série M489, classificada como industrial e de periodicidade mensal, representa os embarques totais de pneus pneumáticos (pneumatic casings) ao longo do tempo. O período histórico cobre de janeiro de 1982 a junho de 1992, seguido por projeções (“future data”) até dezembro de 1993.

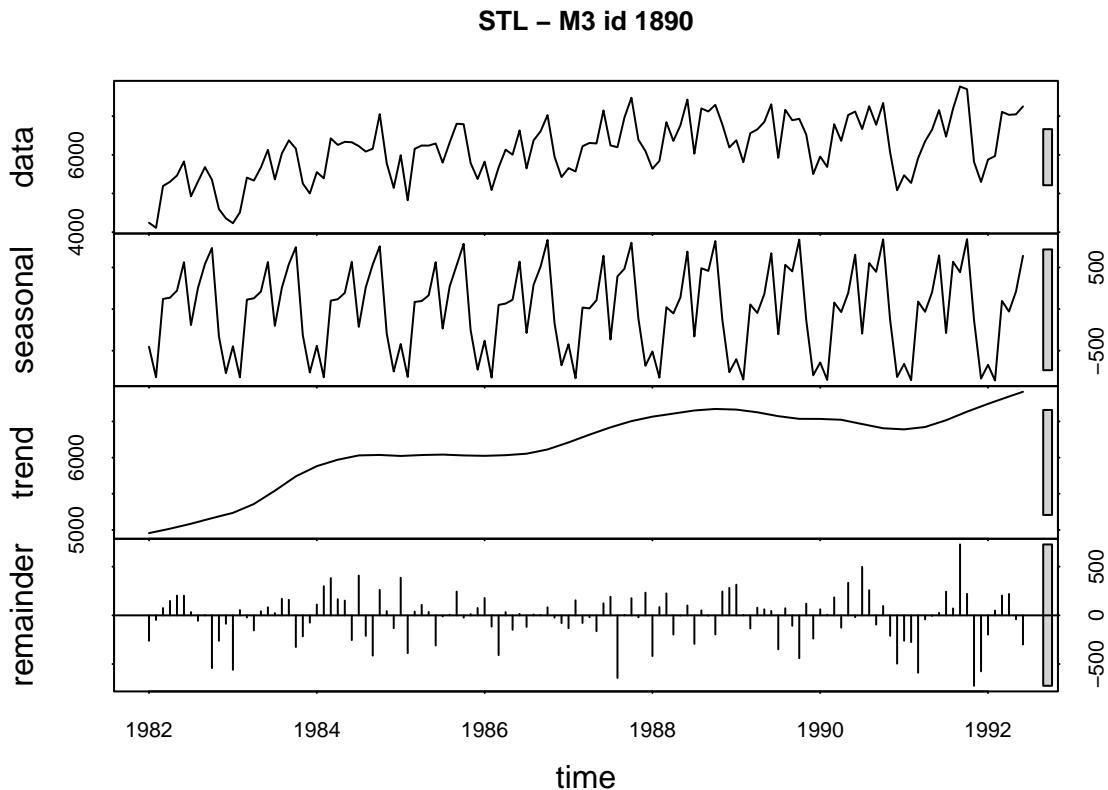
```
M3[[id]] %>% plot() # gráfico do treinamento e do teste
```

N1890



Observa-se uma tendência geral de crescimento moderado ao longo da década de 1980, partindo de cerca de 4.200 mil unidades em 1982 para valores próximos de 7.000 mil unidades em anos posteriores, com flutuações sazonais marcadas: os embarques tendem a atingir picos entre meados e o final do ano (junho a outubro) e quedas nos meses iniciais, padrão típico de sazonalidade anual. Há também oscilações cíclicas de médio prazo, refletindo possíveis variações na demanda industrial e no setor automotivo. As projeções para 1992-1993 sugerem manutenção dos níveis elevados de produção, em torno de 6.500 a 7.700 mil unidades mensais, indicando estabilidade após o crescimento observado na década anterior.

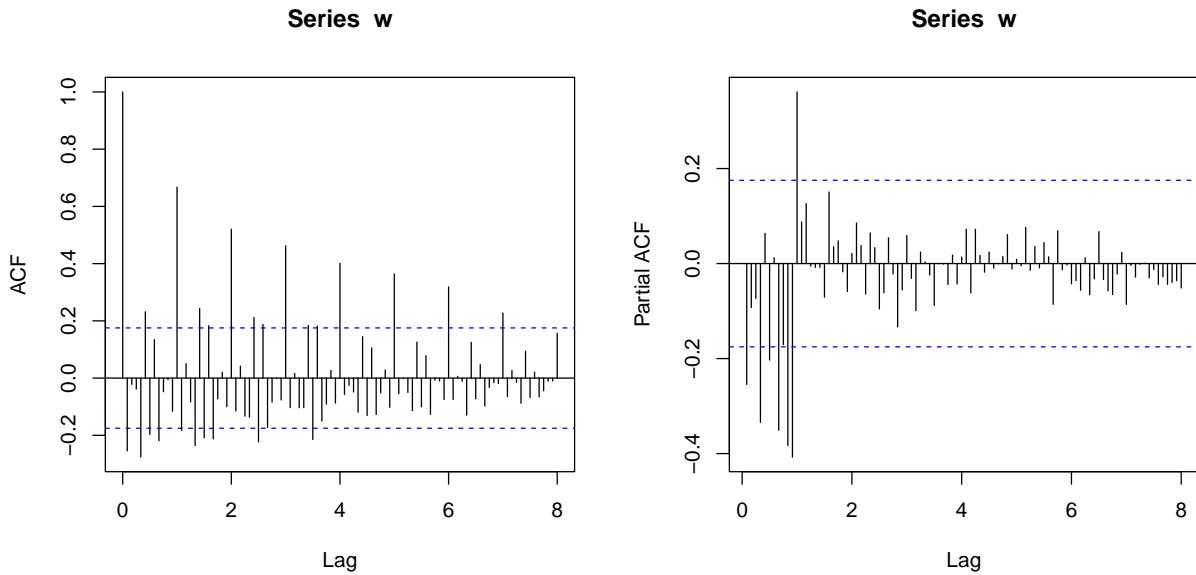
Decomposição STL



Aplicando a decomposicao, podemos notar que conseguimos capturar bem a sazonalidade e observamos uma tendencia de crescimento

Seleção dos modelos

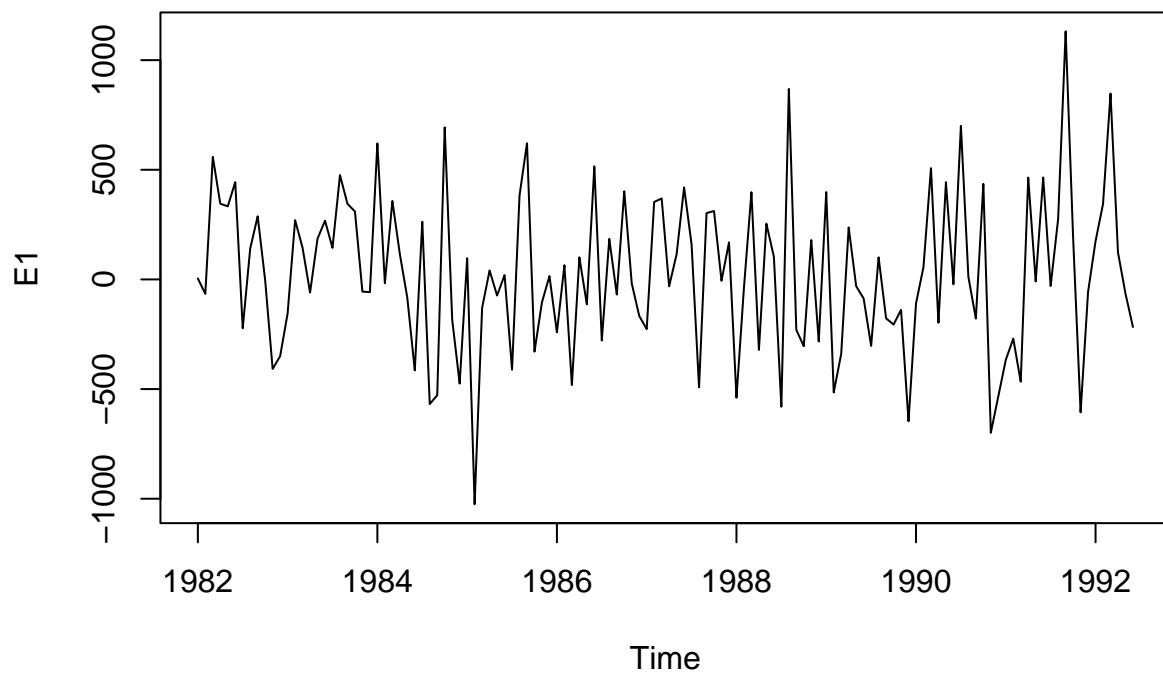
SARIMA (Manual) Para a selecao do modelo, foi observado que a serie necessita somente de uma integracao simples e sem uma sazonal, com isso podemos analisar os graficos FAC e FACP da serie integrada



Nos gráficos de autocorrelação da série diferenciada w , observa-se que os *lags* intermediários não estão bem definidos, especialmente na FACP, indicando que valores pequenos de p e q devem ser testados inicialmente. Contudo, nota-se uma estrutura sazonal evidente: na FAC, os *lags* sazonais exibem um decaimento gradual, enquanto na FACP há um corte mais claro no primeiro *lag* sazonal, sugerindo um comportamento autorregressivo na parte sazonal. Assim, até este ponto, o modelo indicado é um **SARIMA**($p, 1, q$) \times ($1, 0, 0$), sendo conveniente agora variar os valores de p e q para identificar a melhor combinação e alcançar um ajuste mais adequado ao comportamento da série.

SARIMA(1,1,1)x(1,0,1)

```
fit <- Arima(x, order = c(1,1,1), seasonal = c(1,0,0))
```



Resíduos do modelo ARIMA com Drift

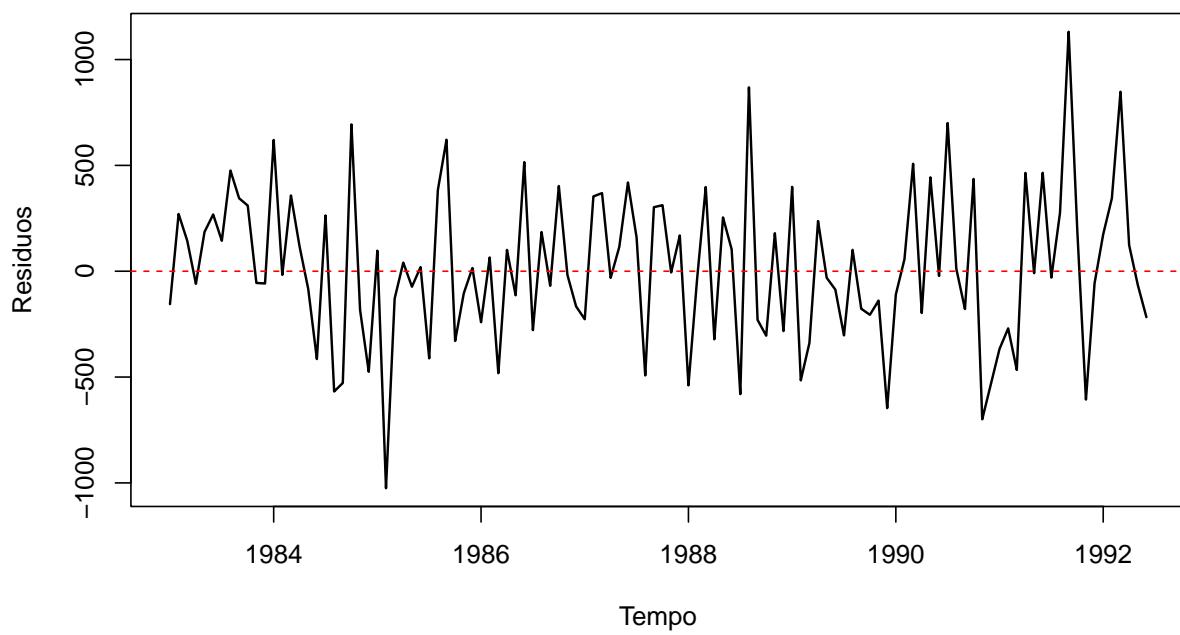
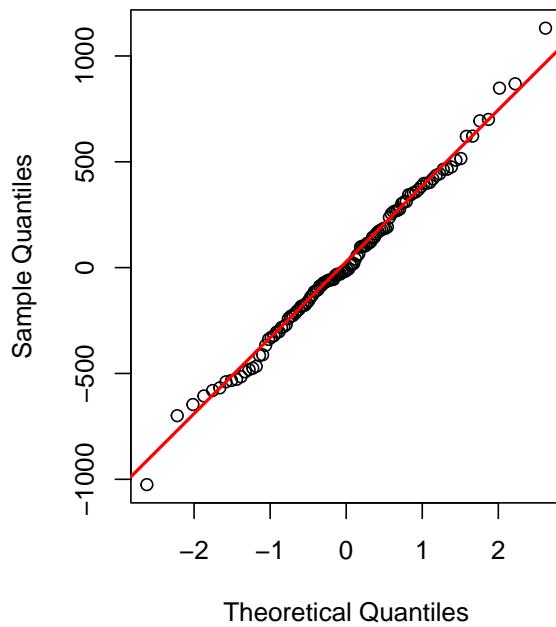
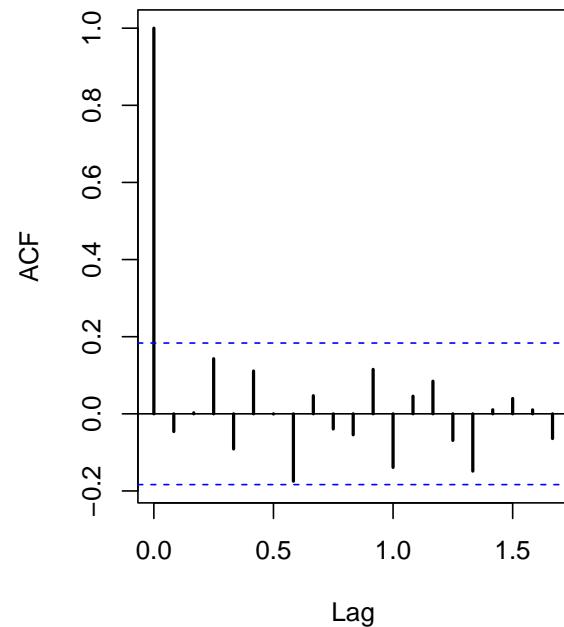


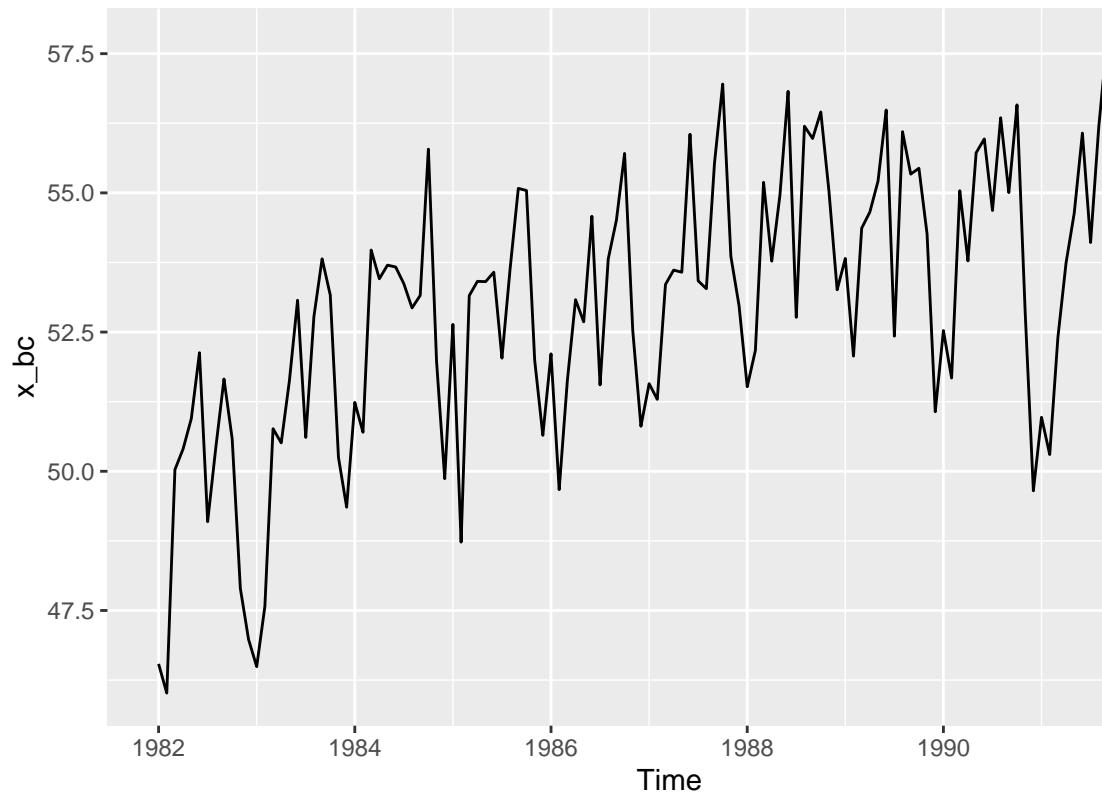
Grafico QQ-Normal dos Residuos



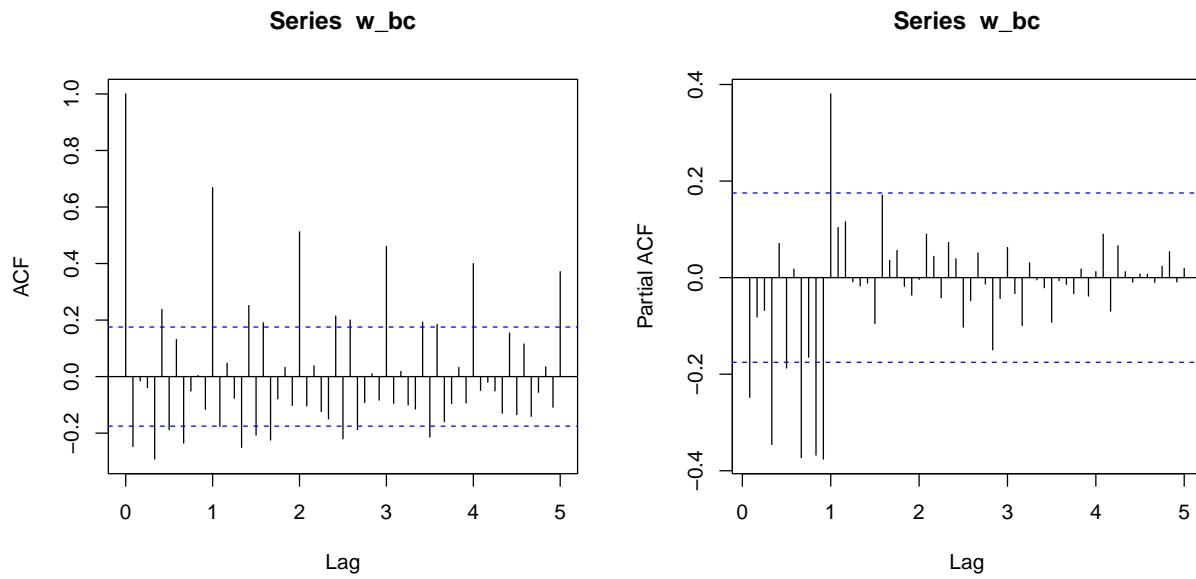
FAC dos Residuos



Série transformada por Box-Cox



SARIMA com Box-Cox



Resíduos do modelo ARIMA com Drift

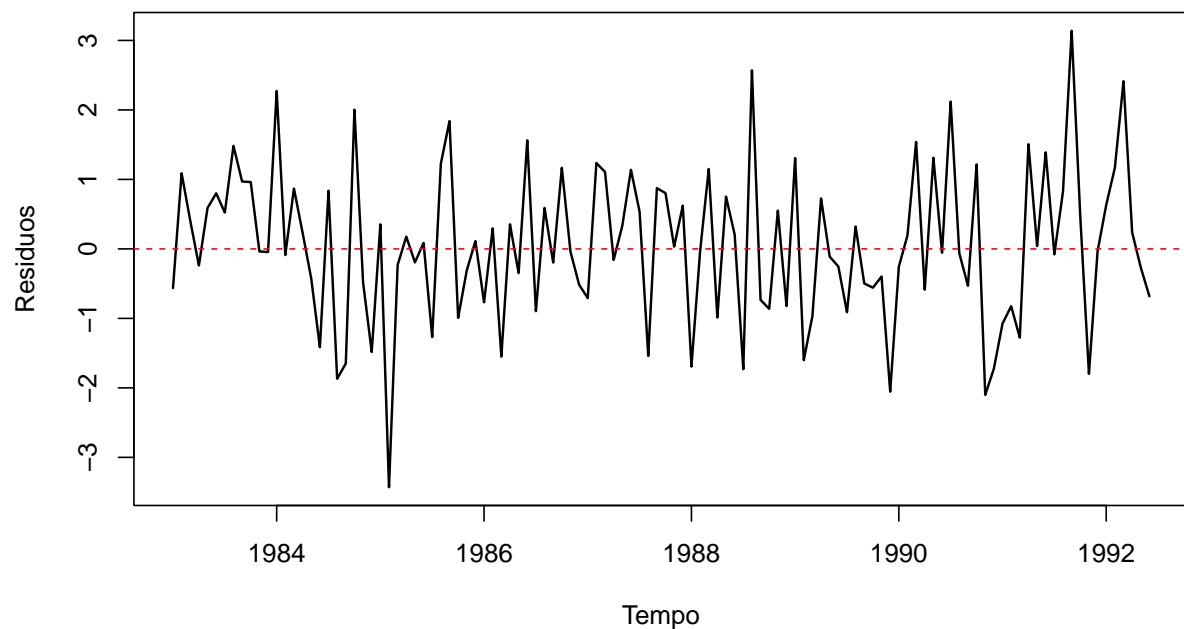
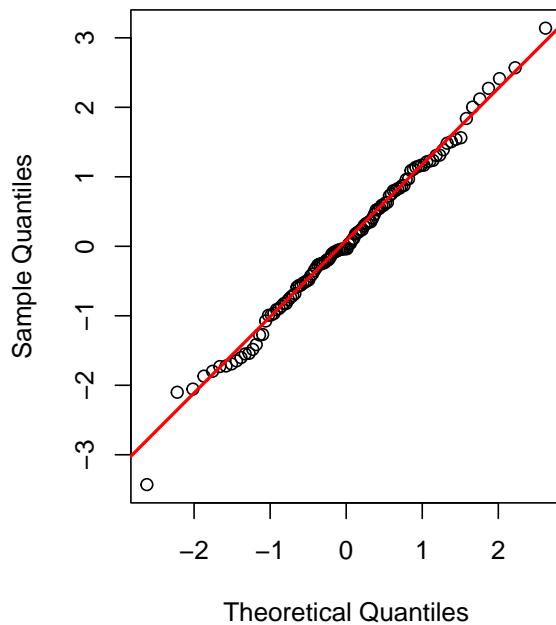
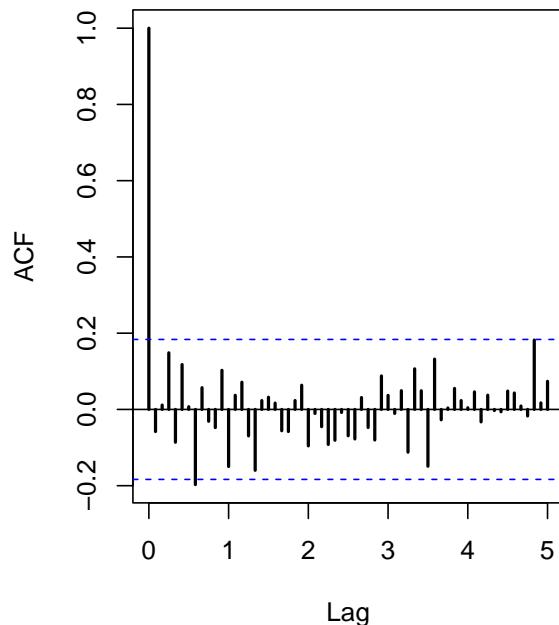


Grafico QQ-Normal dos Residuos



FAC dos Residuos

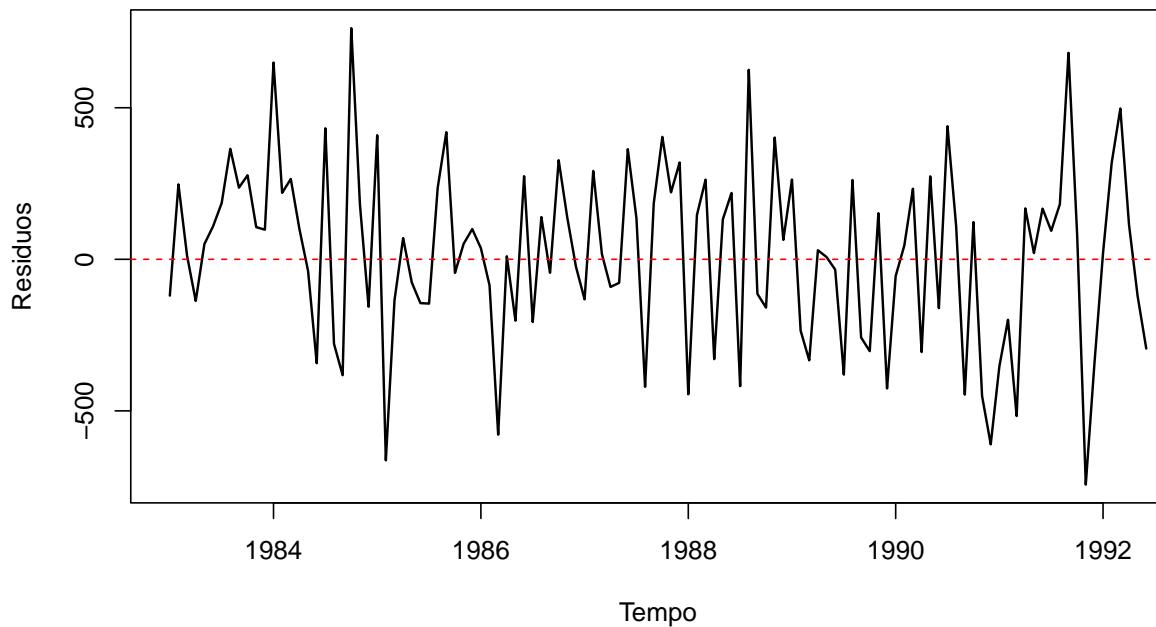


```
#### ARIMA com Drift
fit_drift <- Arima(x, order = c(1,0,1), seasonal = c(1,1,1), include.drift = TRUE)

# Resíduos (a partir de 1983)
E <- fit_drift$residuals %>% window(start = c(1983, 1))
```

```
# Série dos resíduos
plot(E,
      main = "Resíduos do modelo ARIMA com Drift",
      ylab = "Resíduos",
      xlab = "Tempo",
      lwd = 1.5)
abline(h = 0, col = "red", lty = 2)
```

Resíduos do modelo ARIMA com Drift



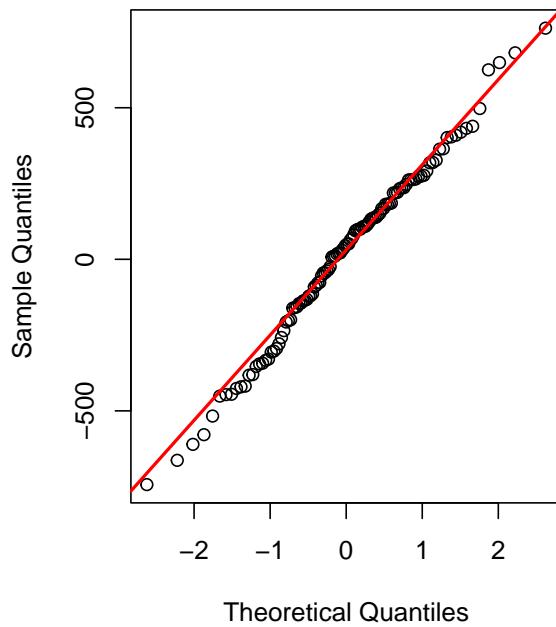
SARIMA com Drift

```
# QQ-plot e ACF lado a lado
par(mfrow = c(1, 2))

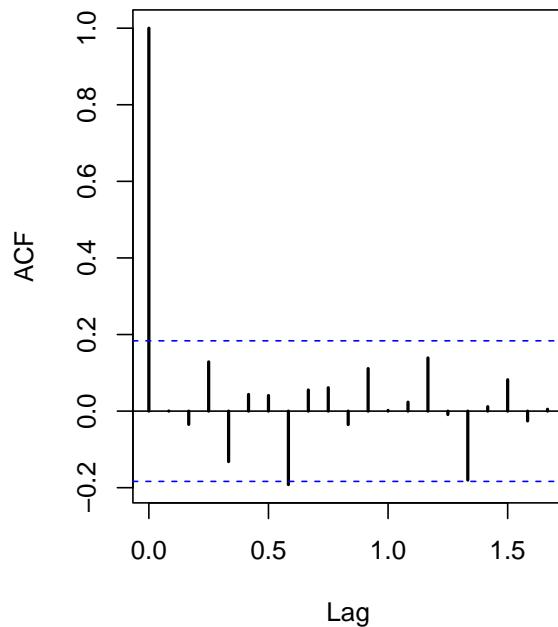
qqnorm(E,
       main = "Grafico QQ-Normal dos Resíduos")
qqline(E, col = "red", lwd = 2)

acf(E,
     main = "FAC dos Resíduos",
     lwd = 2)
```

Grafico QQ-Normal dos Residuos



FAC dos Residuos



```
par(mfrow = c(1, 1)) # volta para 1 gráfico por vez
```

Parâmetros Estimados

```
summary(fit)
```

SARIMA (Manual)

```
## Series: x
## ARIMA(1,1,1)(1,0,0)[12]
##
## Coefficients:
##          ar1      ma1     sar1
##        0.1662  -0.7749  0.8106
## s.e.  0.1915   0.1385  0.0572
##
## sigma^2 = 135489: log likelihood = -921.11
## AIC=1850.21  AICc=1850.55  BIC=1861.52
##
## Training set error measures:
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 25.79928 362.1984 286.3816 0.1531367 4.693914 0.7814409 -0.0131891
```

```
summary(fit_bc)
```

SARIMA com Box-Cox

```
## Series: x
## ARIMA(1,1,1)(1,0,0)[12]
## Box Cox transformation: lambda= 0.3369062
##
## Coefficients:
##          ar1      ma1     sar1
##        0.1828 -0.7675  0.8132
## s.e.  0.2099  0.1553  0.0581
##
## sigma^2 = 1.294: log likelihood = -198.74
## AIC=405.48  AICc=405.82  BIC=416.8
##
## Training set error measures:
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 28.71617 363.6663 286.5944 0.2184702 4.68873 0.7820214 -0.03418845
```

```
summary(fit_drift)
```

SARIMA com Drift

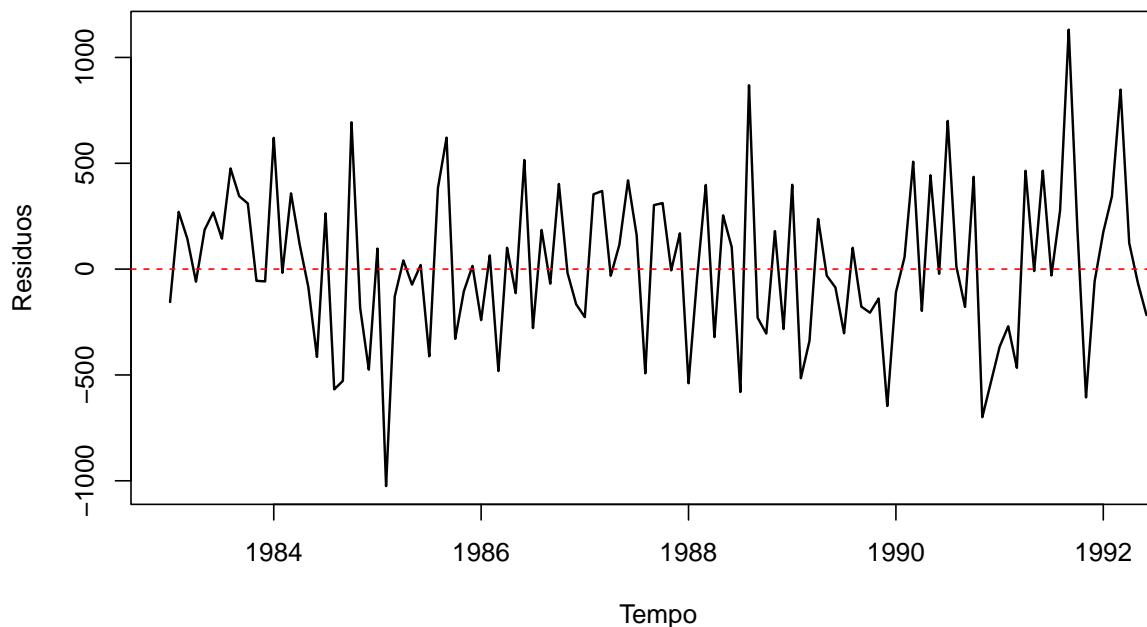
```
## Series: x
## ARIMA(1,0,1)(1,1,1)[12] with drift
##
## Coefficients:
##          ar1      ma1     sar1     sma1     drift
##        0.8980 -0.5810  0.2914 -1.0000  13.3067
## s.e.  0.0669  0.1186  0.1104  0.1236  3.2589
##
## sigma^2 = 91257: log likelihood = -821.26
## AIC=1654.52  AICc=1655.3  BIC=1670.93
##
## Training set error measures:
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 16.30541 280.9705 215.9047 0.1027304 3.493005 0.5891327
##           ACF1
## Training set 0.000707696
```

Análise dos Resíduos

```
# Resíduos (a partir de 1983)
E <- fit$residuals %>% window(start = c(1983, 1))
```

```
# Série dos resíduos
plot(E,
  main = "Resíduos do modelo SARIMA",
  ylab = "Resíduos",
  xlab = "Tempo",
  lwd = 1.5)
abline(h = 0, col = "red", lty = 2)
```

Resíduos do modelo SARIMA



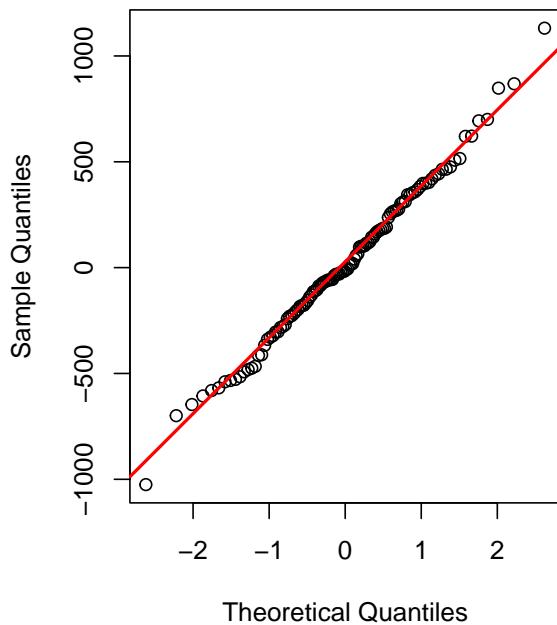
SARIMA (Manual)

```
# QQ-plot e ACF lado a lado
par(mfrow = c(1, 2))

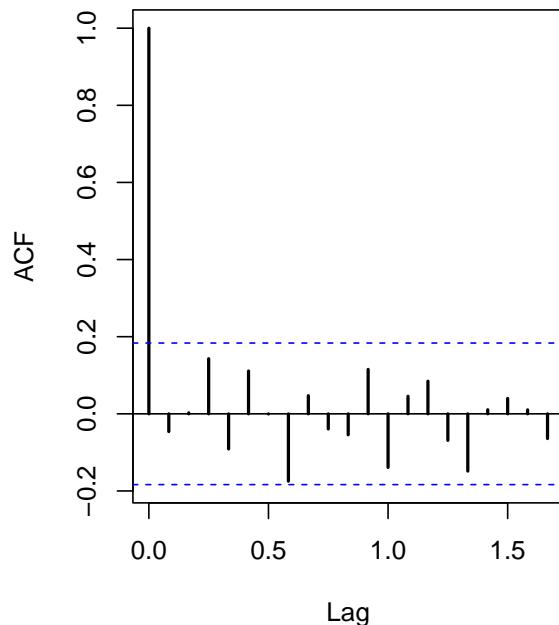
qqnorm(E,
       main = "Grafico QQ-Normal dos Resíduos")
qqline(E, col = "red", lwd = 2)

acf(E,
    main = "FAC dos Resíduos",
    lwd = 2)
```

Grafico QQ-Normal dos Resíduos



FAC dos Resíduos

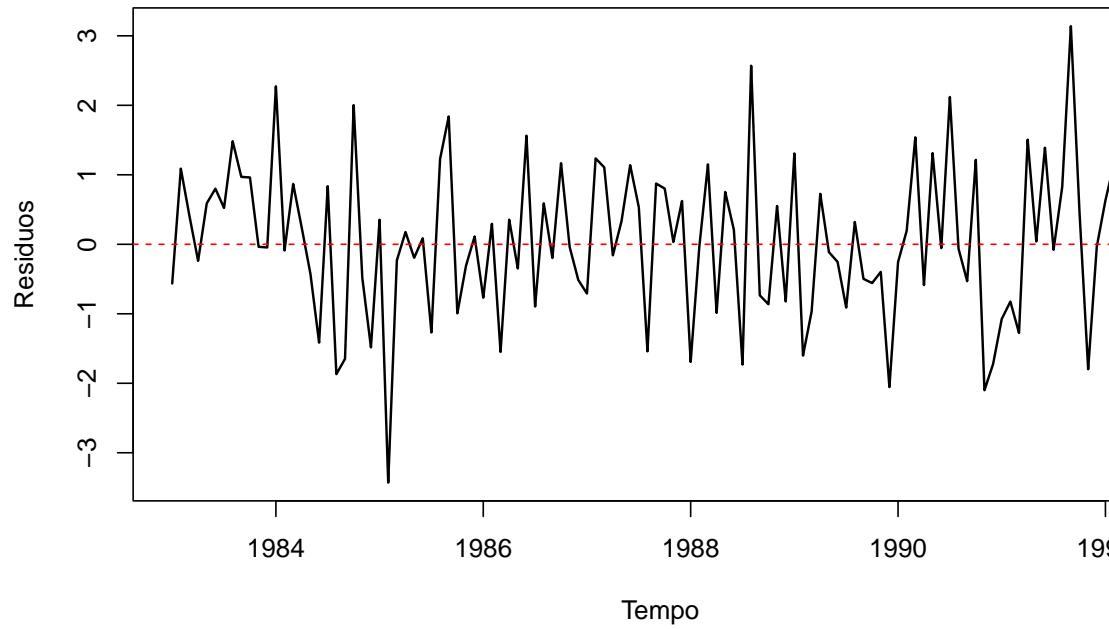


```
par(mfrow = c(1, 1)) # volta para 1 gráfico por vez
```

```
# Resíduos (a partir de 1983)
E <- fit_bc$residuals %>% window(start = c(1983, 1))

# Série dos resíduos
plot(E,
      main = "Resíduos do modelo SARIMA com Box-Cox",
      ylab = "Resíduos",
      xlab = "Tempo",
      lwd = 1.5)
abline(h = 0, col = "red", lty = 2)
```

Resíduos do modelo SARIMA com Box-Cox



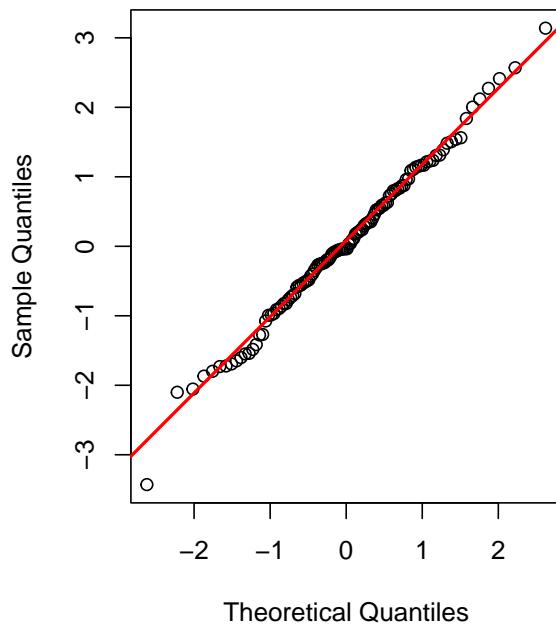
SARIMA com Box-Cox

```
# QQ-plot e ACF lado a lado
par(mfrow = c(1, 2))

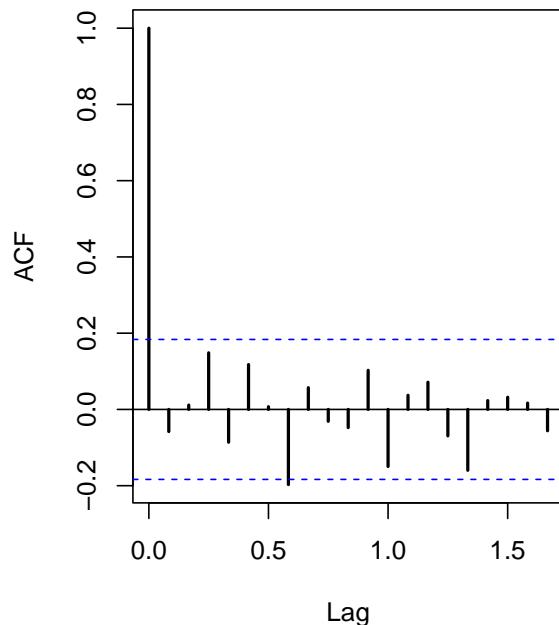
qqnorm(E,
       main = "Grafico QQ-Normal dos Resíduos")
qqline(E, col = "red", lwd = 2)

acf(E,
     main = "FAC dos Resíduos",
     lwd = 2)
```

Grafico QQ-Normal dos Resíduos



FAC dos Resíduos

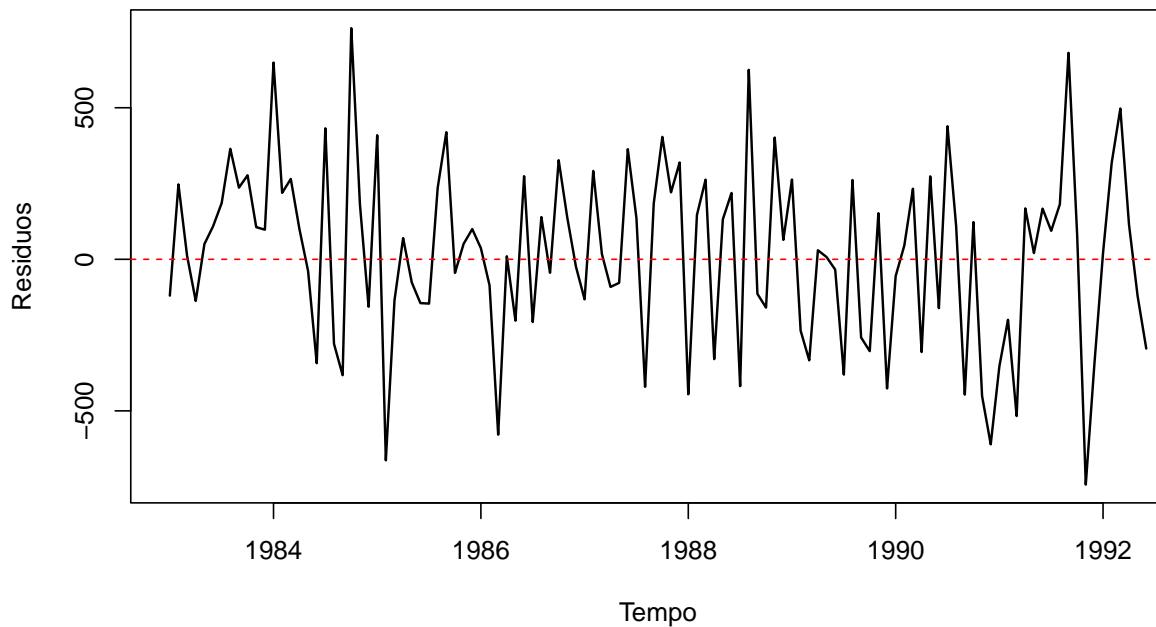


```
par(mfrow = c(1, 1)) # volta para 1 gráfico por vez
```

```
# Resíduos (a partir de 1983)
E <- fit_drift$residuals %>% window(start = c(1983, 1))

# Série dos resíduos
plot(E,
      main = "Resíduos do modelo SARIMA com Drift",
      ylab = "Resíduos",
      xlab = "Tempo",
      lwd = 1.5)
abline(h = 0, col = "red", lty = 2)
```

Resíduos do modelo SARIMA com Drift



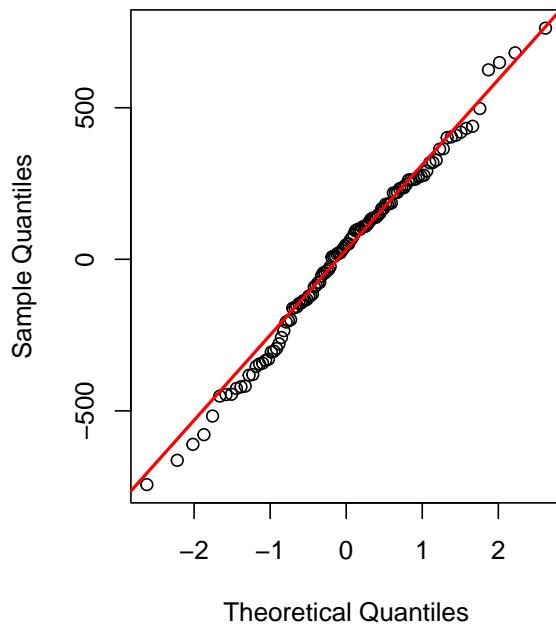
SARIMA com Drift

```
# QQ-plot e ACF lado a lado
par(mfrow = c(1, 2))

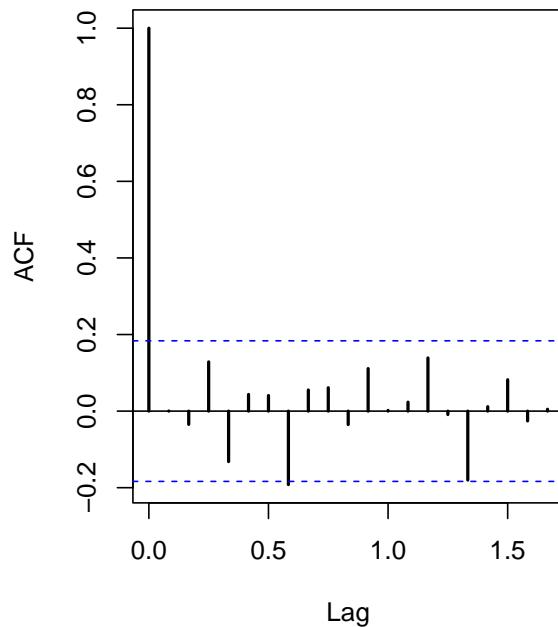
qqnorm(E,
       main = "Grafico QQ-Normal dos Resíduos")
qqline(E, col = "red", lwd = 2)

acf(E,
     main = "FAC dos Resíduos",
     lwd = 2)
```

Grafico QQ-Normal dos Residuos



FAC dos Residuos



```
par(mfrow = c(1, 1)) # volta para 1 gráfico por vez
```

Estudo Janela Deslizante

```
#SARIMA manual
f_arima_manual <- function(y, h){
  fit = Arima(y, order=c(1,1,1), seasonal=c(1,0,1))
  forecast(fit, h)
}

f_arima_bc <- function(y, h){
  fit = Arima(y, order=c(1,1,1), seasonal=c(1,0,1), lambda = BoxCox.lambda(y))
  forecast(fit, h)
}

f_arima_drift <- function(y, h){
  fit = Arima(x, order = c(1,0,1), seasonal = c(1,1,1), include.drift = TRUE)
  forecast(fit, h)
}

f_holt_damped <- function(y, h){
  fit = holt(y, h, damped = TRUE )
  forecast(fit, h)
}
```

```

f_holt_winters <- function(y, h){
  fit = hw(y, h=, seasonal = "m")
  forecast(fit, h)
}

CV_arima_manual = tsCV(y=x, forecastfunction=f_arima,h=12, initial=length(x) - 36)
CV_arima_bc = tsCV(y=x, forecastfunction=f_arima_bc, h=12, initial=length(x) - 36)
CV_arima_drift = tsCV(y=x, forecastfunction=f_arima_drift, h=12, initial=length(x) - 36)
CV_holt_damped = tsCV(y=x, forecastfunction=f_holt_damped, h=12, initial=length(x) - 36)
CV_holt_winters = tsCV(y=x, forecastfunction=f_holt_winters, h=12, initial=length(x) - 36)

# ultimas 10 linhas de 7 colunas da saida da tsCV()
CV_arima_manual[,1:7] %>% round(2) %>% tail(10)

```

```

##          h=1 h=2 h=3 h=4 h=5 h=6 h=7
## Sep 1991  NA  NA  NA  NA  NA  NA  NA
## Oct 1991  NA  NA  NA  NA  NA  NA  NA
## Nov 1991  NA  NA  NA  NA  NA  NA  NA
## Dec 1991  NA  NA  NA  NA  NA  NA  NA
## Jan 1992  NA  NA  NA  NA  NA  NA  NA
## Feb 1992  NA  NA  NA  NA  NA  NA  NA
## Mar 1992  NA  NA  NA  NA  NA  NA  NA
## Apr 1992  NA  NA  NA  NA  NA  NA  NA
## May 1992  NA  NA  NA  NA  NA  NA  NA
## Jun 1992  NA  NA  NA  NA  NA  NA  NA

MAE_arima_manual = CV_arima_manual %>% abs() %>% colMeans(na.rm=T)
MAE_arima_bc = CV_arima_bc %>% abs() %>% colMeans(na.rm=T)
MAE_arima_drift = CV_arima_drift %>% abs() %>% colMeans(na.rm=T)
MAE_holt_damped = CV_holt_damped %>% abs() %>% colMeans(na.rm=T)
MAE_holt_winters = CV_holt_winters %>% abs() %>% colMeans(na.rm=T)
tab = cbind(MAE_arima_manual,MAE_arima_bc,MAE_arima_drift,MAE_holt_damped,MAE_holt_winters)
print(tab)

```

	MAE_arima_manual	MAE_arima_bc	MAE_arima_drift	MAE_holt_damped
## h=1	NaN	304.7086	608.9171	573.9212
## h=2	NaN	405.2826	851.4834	734.8541
## h=3	NaN	458.6039	1150.0308	893.2316
## h=4	NaN	471.9360	1253.9906	971.0608
## h=5	NaN	422.8283	657.5510	990.6827
## h=6	NaN	435.0938	790.1266	908.8965
## h=7	NaN	455.0158	686.5895	818.0837
## h=8	NaN	417.3419	737.9126	709.3122
## h=9	NaN	401.5053	720.8481	606.7625
## h=10	NaN	387.5213	708.5031	485.3540
## h=11	NaN	395.7700	809.6809	506.4933
## h=12	NaN	432.8981	1106.2787	571.1332
## MAE_holt_winters				

```

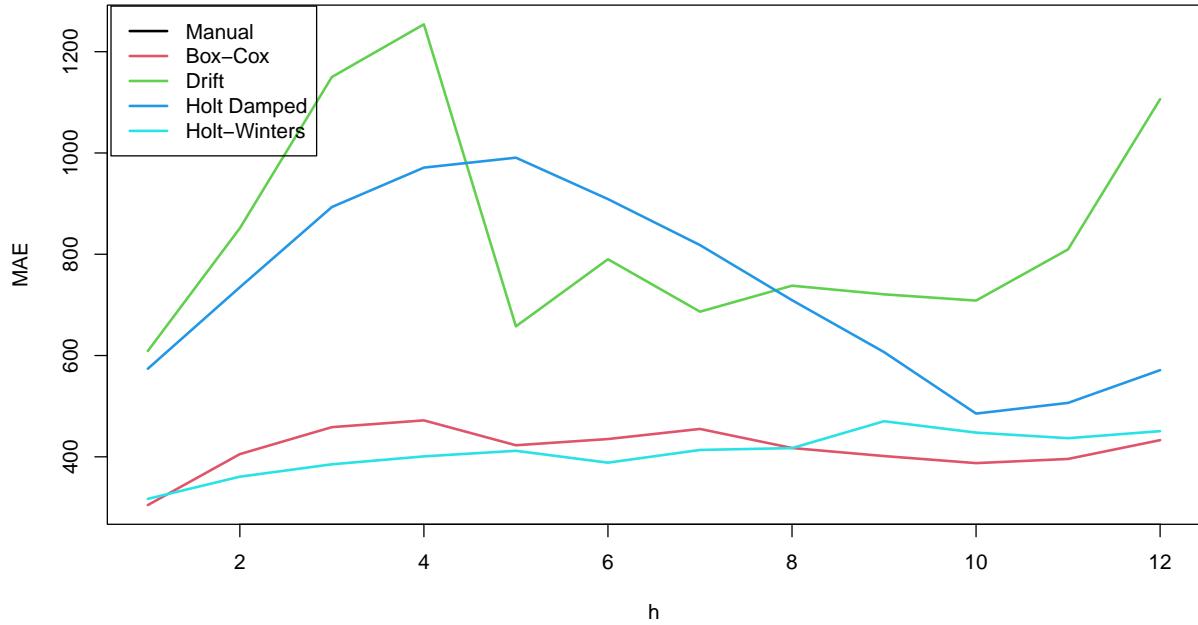
## h=1      316.9257
## h=2      360.7795
## h=3      385.2131
## h=4      400.8344
## h=5      411.8618
## h=6      388.4344
## h=7      413.5620
## h=8      417.1587
## h=9      470.2785
## h=10     447.7575
## h=11     436.7623
## h=12     450.6965

```

```

plot.ts(tab, plot.type='s', col=1:5, lwd=c(2,2), xlab="h", ylab="MAE")
legend(x=0.6, y=1290, legend=c("Manual", "Box-Cox", "Drift", "Holt Damped", "Holt-Winters"), col=1:5, lwd=c(2,2))

```



Previsao

```

# Horizonte de previsão (ex: 12 meses)
h <- 18

# ---- 1) Previsões individuais ----
fc_fit      <- forecast(fit, h = h)          # modelo base
fc_fit_bc   <- forecast(fit_bc, h = h)        # com Box-Cox (já volta à escala original)
fc_fit_drift <- forecast(fit_drift, h = h)    # com drift

```

```

# Modelo ARIMA base
p1 <- autoplot(fc_fit) +
  ggtitle("Previsao - ARIMA base") +
  ylab("Shipments") + xlab("Ano") +
  theme_minimal(base_size = 12)

# Modelo com Box-Cox
p2 <- autoplot(fc_fit_bc) +
  ggtitle("Previsao - ARIMA com Box-Cox") +
  ylab("Shipments") + xlab("Ano") +
  theme_minimal(base_size = 12)

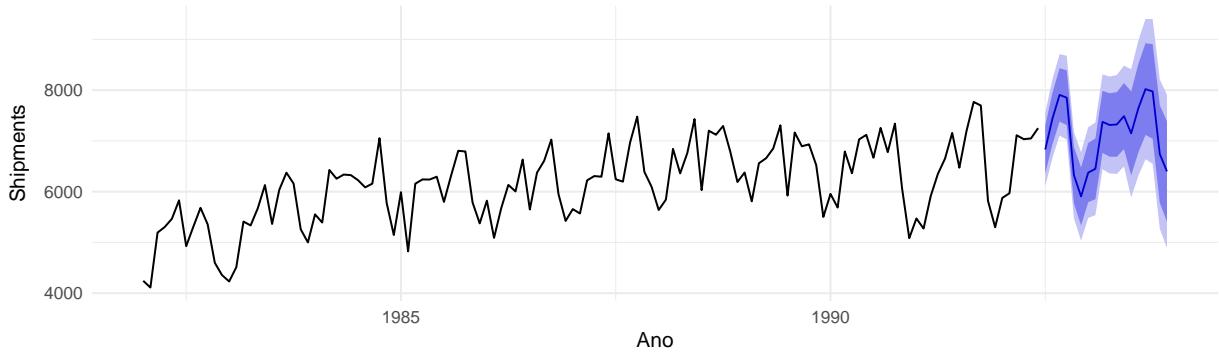
# Modelo com Drift
p3 <- autoplot(fc_fit_drift) +
  ggtitle("Previsao - ARIMA com Drift") +
  ylab("Shipments") + xlab("Ano") +
  theme_minimal(base_size = 12)

# Combina os 3 lado a lado
(p1 / p2 / p3) +
  plot_annotation(title = "Comparacao das previsoes - Serie M489")

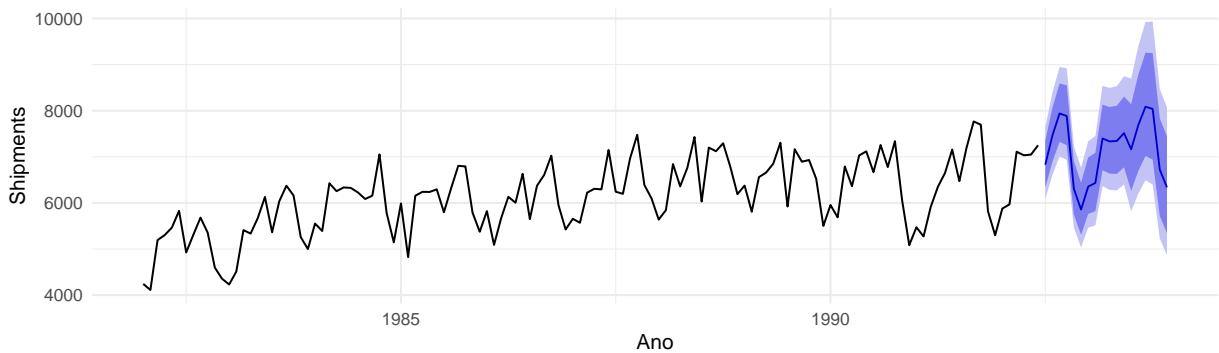
```

Comparacao das previsões — Serie M489

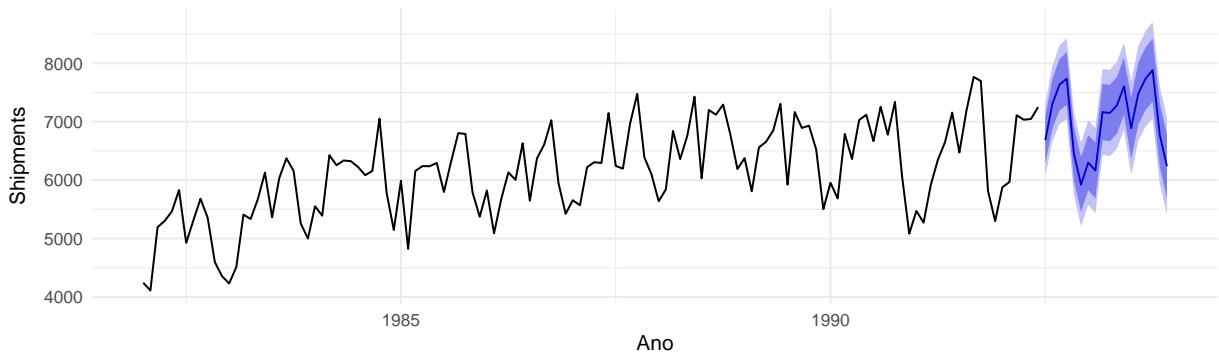
Previsão – ARIMA base



Previsão – ARIMA com Box–Cox



Previsão – ARIMA com Drift



```
tab_fit <- data.frame(
  Data = as.yearmon(time(fc_fit$mean)),
  Teste = teste,
  Previsto = as.numeric(fc_fit$mean),
  Lower95 = as.numeric(fc_fit$lower[,2]),
  Upper95 = as.numeric(fc_fit$upper[,2])
)

tab_bc <- data.frame(
  Data = as.yearmon(time(fc_fit_bc$mean)),
  Teste = teste,
  Previsto = as.numeric(fc_fit_bc$mean),
  Lower95 = as.numeric(fc_fit_bc$lower[,2]),
  Upper95 = as.numeric(fc_fit_bc$upper[,2])
```

```

)

tab_drift <- data.frame(
  Data = as.yearmon(time(fc_fit_drift$mean)),
  Teste = teste,
  Previsto = as.numeric(fc_fit_drift$mean),
  Lower95 = as.numeric(fc_fit_drift$lower[,2]),
  Upper95 = as.numeric(fc_fit_drift$upper[,2])
)

# Cria um título (textGrob) para cada tabela
titulo1 <- textGrob("ARIMA Manual", gp = gpar(fontsize = 12, fontface = "bold"))
titulo2 <- textGrob("ARIMA com Box-Cox", gp = gpar(fontsize = 12, fontface = "bold"))
titulo3 <- textGrob("ARIMA com Drift", gp = gpar(fontsize = 12, fontface = "bold"))

# Empilha cada título com sua respectiva tabela
tabela1 <- arrangeGrob(titulo1, tableGrob(tab_fit, rows = NULL), ncol = 1, heights = c(0.2, 1))
tabela2 <- arrangeGrob(titulo2, tableGrob(tab_bc, rows = NULL), ncol = 1, heights = c(0.2, 1))
tabela3 <- arrangeGrob(titulo3, tableGrob(tab_drift, rows = NULL), ncol = 1, heights = c(0.2, 1))

# Mostra tudo lado a lado
grid.arrange(tabela1, tabela2, tabela3, ncol = 3,
             top = textGrob("Previsões Pontuais e Intervalares",
                            gp = gpar(fontsize = 13, fontface = "bold")))

```

Previsões Pontuais e Intervalares

ARIMA Manual

ARIMA com Box-Cox

ARIMA com Drift

Data	Teste	Previsto	Lower95	Upper95
jul 1992	6573.0	6830.610	6109.170	7552.050
ago 1992	7041.6	7439.164	6664.462	8213.866
set 1992	7376.7	7906.103	7103.620	8708.587
out 1992	7654.2	7849.816	7023.459	8676.174
nov 1992	6778.8	6326.702	5477.606	7175.798
dez 1992	6348.6	5905.792	5034.626	6776.958
jan 1993	6002.7	6374.870	5482.191	7267.548
fev 1993	6370.5	6449.037	5535.354	7362.720
mar 1993	7664.1	7374.785	6440.570	8308.999
abr 1993	6969.6	7312.776	6358.472	8267.081
mai 1993	6963.9	7323.719	6349.739	8297.700
jun 1993	7576.8	7488.588	6495.322	8481.855
jul 1993	6618.6	7147.834	5885.183	8410.486
ago 1993	7473.6	7641.108	6309.295	8972.920
set 1993	7668.0	8019.593	6638.814	9400.372
out 1993	7725.0	7973.969	6548.377	9399.560
nov 1993	7223.1	6739.384	5270.740	8208.028
dez 1993	6461.4	6398.208	4887.802	7908.615

Data	Teste	Previsto	Lower95	Upper95
jul 1992	6573.0	6829.434	6080.720	7636.874
ago 1992	7041.6	7462.490	6604.357	8391.499
set 1992	7376.7	7941.513	7012.342	8948.883
out 1992	7654.2	7884.516	6931.174	8921.011
nov 1992	6778.8	6304.397	5462.002	7228.753
dez 1992	6348.6	5856.168	5033.685	6763.190
jan 1993	6002.7	6355.340	5464.018	7338.133
fev 1993	6370.5	6433.624	5513.458	7450.290
mar 1993	7664.1	7397.919	6363.149	8538.569
abr 1993	6969.6	7334.011	6283.074	8495.396
mai 1993	6963.9	7345.296	6271.519	8534.445
jun 1993	7576.8	7514.975	6402.800	8748.290
jul 1993	6618.6	7162.582	5819.209	8697.083
ago 1993	7473.6	7690.746	6203.221	9397.525
set 1993	7668.0	8086.526	6490.925	9922.787
out 1993	7725.0	8039.598	6400.051	9936.061
nov 1993	7223.1	6719.616	5228.530	8468.609
dez 1993	6461.4	6337.495	4866.633	8076.648

Data	Teste	Previsto	Lower95	Upper95
jul 1992	6573.0	6683.798	6065.117	7302.479
ago 1992	7041.6	7309.741	6660.831	7958.650
set 1992	7376.7	7636.525	6964.298	8308.752
out 1992	7654.2	7736.610	7046.235	8426.985
nov 1992	6778.8	6453.507	5748.938	7158.076
dez 1992	6348.6	5923.786	5208.097	6639.475
jan 1993	6002.7	6301.389	5578.727	7024.052
fev 1993	6370.5	6164.885	5435.104	6894.666
mar 1993	7664.1	7166.567	6431.110	7902.023
abr 1993	6969.6	7148.540	6408.557	7888.523
mai 1993	6963.9	7278.433	6534.841	8022.024
jun 1993	7576.8	7608.034	6861.572	8354.496
jul 1993	6618.6	6889.083	6087.743	7690.424
ago 1993	7473.6	7483.549	6672.387	8294.712
set 1993	7668.0	7736.240	6917.335	8555.144
out 1993	7725.0	7883.432	7058.451	8708.413
nov 1993	7223.1	6771.758	5942.049	7601.466
dez 1993	6461.4	6236.979	5403.650	7070.307

```

# ====== Config ======
h <- 18

# ===== Ajuste dos modelos =====
m_auto      <- auto.arima(x, seasonal = TRUE, stepwise = FALSE, approximation = FALSE)
m_ses        <- ses(x, h = h)
m_holt       <- holt(x, h = h)
m_holt_d    <- holt(x, damped = TRUE, h = h)
m_hw         <- hw(x, seasonal = "multiplicative", h = h)
m_ets        <- ets(x) |> forecast(h = h)

```

```

# ===== Função auxiliar =====
make_tab <- function(fc) {
  # Se não for forecast, gera previsão
  if (!inherits(fc, "forecast")) fc <- forecast(fc, h = h)

  data.frame(
    Data = as.yearmon(time(fc$mean)),
    Previsto = as.numeric(fc$mean)
  ) %>%
    mutate(across(where(is.numeric), round, 2))
}

# ===== Tabelas individuais =====
tab_auto <- make_tab(forecast(m_auto, h = h))

```

MAE

```

## Warning: There was 1 warning in 'mutate()'.
## i In argument: 'across(where(is.numeric), round, 2)'.
## Caused by warning:
## ! The '...' argument of 'across()' is deprecated as of dplyr 1.1.0.
## Supply arguments directly to '.fns' through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \((x) mean(x, na.rm = TRUE))

tab_ses   <- make_tab(m_ses)
tab_holt  <- make_tab(m_holt)
tab_holt_d <- make_tab(m_holt_d)
tab_hw    <- make_tab(m_hw)
tab_ets   <- make_tab(m_ets)

h_eval <- length(teste)

# Garante que o objeto é um 'forecast'; se não for, gera com horizonte h_eval

```

```

ensure_forecast <- function(obj, h) {
  if (inherits(obj, "forecast")) obj else forecast(obj, h = h)
}

# Monta a lista de previsões (cada item vira/é um forecast com h_eval passos)
fc_list <- list(
  "auto.arima()"      = ensure_forecast(m_auto,      h_eval),
  "ses()"             = ensure_forecast(m_ses,       h_eval),
  "holt()"            = ensure_forecast(m_holt,      h_eval),
  "holt(damped=TRUE)" = ensure_forecast(m_holt_d,    h_eval),
  "hw()"              = ensure_forecast(m_hw,        h_eval),
  "ets()"              = ensure_forecast(m_ets,       h_eval),
  "ARIMA base (fit)" = ensure_forecast(fit,         h_eval),
  "Box-Cox (fit_bc)" = ensure_forecast(fit_bc,      h_eval),
  "Com Drift (fit_drift)" = ensure_forecast(fit_drift, h_eval)
)

# Função MAE segura (corta ao menor comprimento disponível)
MAE <- function(fc, y) {
  yhat <- as.numeric(fc$mean)
  n <- min(length(yhat), length(y))
  mean(abs(yhat[1:n] - as.numeric(y)[1:n]), na.rm = TRUE)
}

# Calcula MAE para cada modelo
mae_vec <- sapply(fc_list, MAE, y = teste)

# Tabela final (ordenada do menor MAE para o maior)
tabela_mae <- data.frame(
  Modelo = names(mae_vec),
  MAE = round(unname(mae_vec), 2),
  row.names = NULL
)
tabela_mae <- tabela_mae[order(tabela_mae$MAE), ]

print(tabela_mae)

##           Modelo     MAE
## 1      auto.arima() 211.47
## 9  Com Drift (fit_drift) 232.26
## 5          hw() 245.40
## 6          ets() 260.44
## 7  ARIMA base (fit) 313.92
## 8   Box-Cox (fit_bc) 336.14
## 4  holt(damped=TRUE) 475.10
## 2          ses() 475.85
## 3          holt() 500.10

```