ARIMA

- : 1956 1 1995 8 (fma::elec)

1956 1 1995 8 fma::elec . Figure 1 .

```r
autoplot(elec) +
  labs(x = NULL, y = NULL)
```

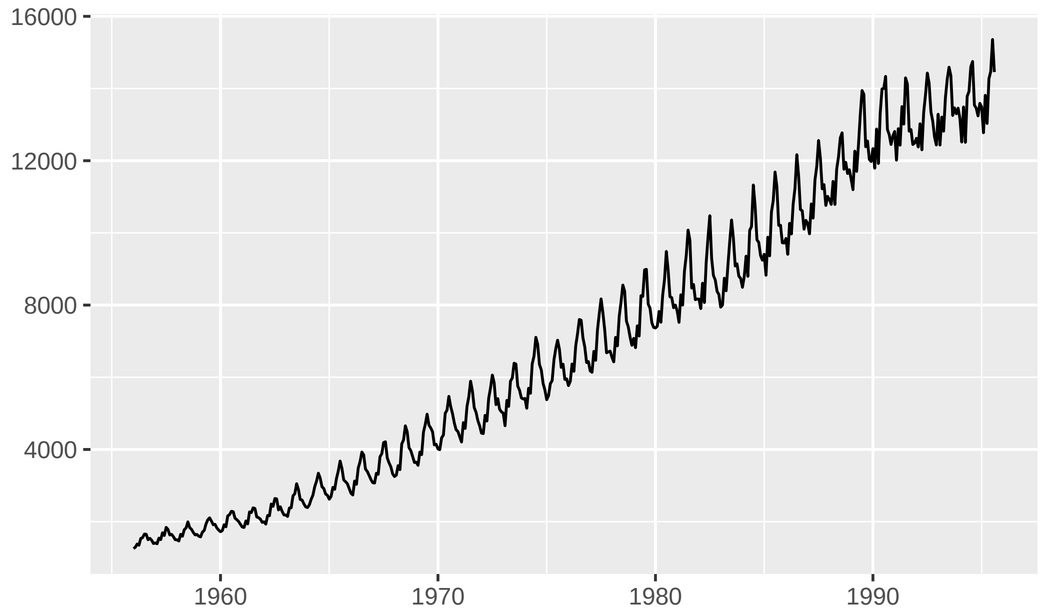


Figure 1: elec

, . .

Box-Cox forecast BoxCox.lambda() BoxCox() . BoxCox.lambda() λ , BoxCox() λ .

elec Box-Cox .

```
(lambda <- BoxCox.lambda(elec))
## [1] 0.2654076
```

Figure 2 .

```
autoplot(BoxCox(elec, lambda)) +
  labs(x = NULL, y = NULL)
```

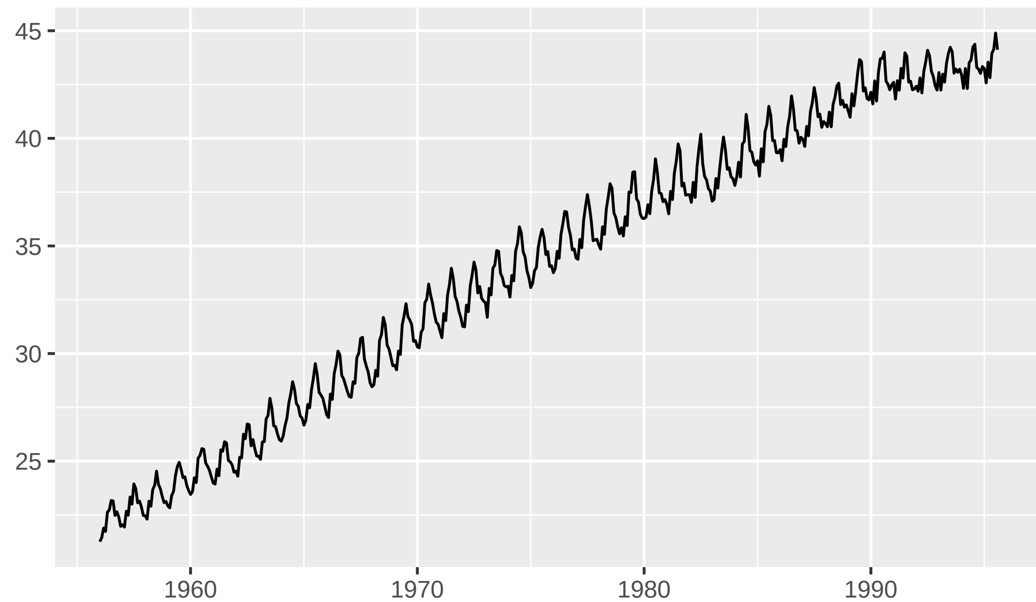


Figure 2: elec Box-Cox

Figure 2 Box-Cox . $y_t^{0.2654}$.
, .

```
library(patchwork)
```

```
p1 <- autoplot(BoxCox(elec, BoxCox.lambda(elec))) +
  labs(x = NULL, y = NULL, title = "Box-Cox transformation")
p2 <- autoplot(log(elec)) +
  labs(x = NULL, y = NULL, title = "log transformation")
```

```r
p1 + p2
```

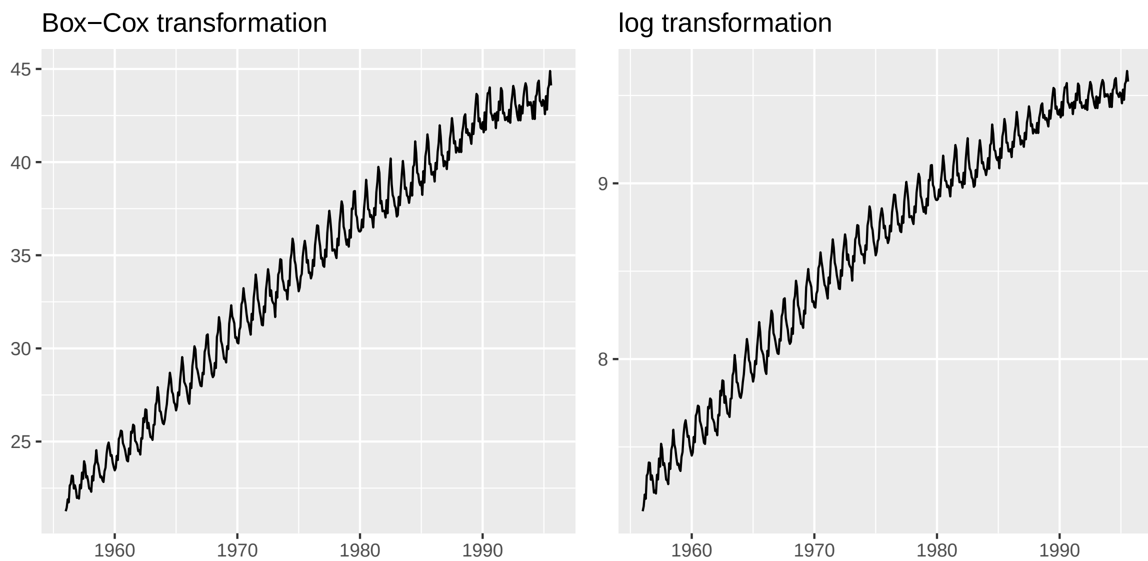


Figure 3: `elec`

Figure 3 . .

- : Google (`fpp2::goog200`)

Google `goog200` ACF . , ACF .

```r
p1 <- autoplot(goog200) +
  labs(x = NULL, y = NULL, title = "Google stock price")
p2 <- ggAcf(goog200) + ggtitle("")
p1 + p2
```

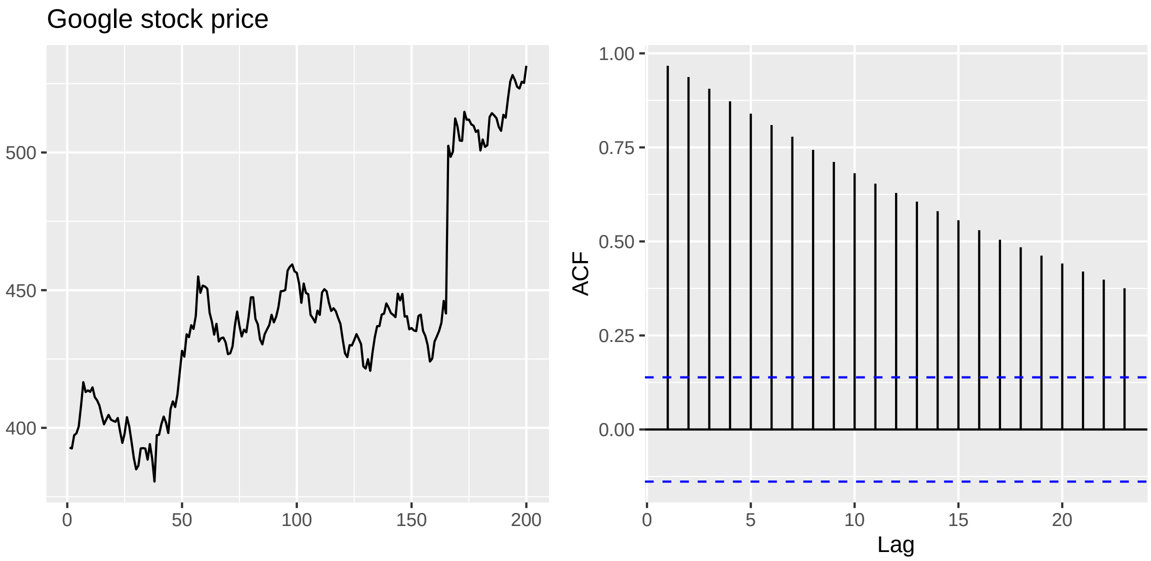


Figure 4: goog200 ACF

goog200 1 .

```
goog200_1 <- diff(goog200)
p3 <- autoplot(goog200_1) +
  labs(x = NULL, y = NULL, title = "Google stock price")
p4 <- ggAcf(goog200_1) + ggtitle("")
p3 + p4
```

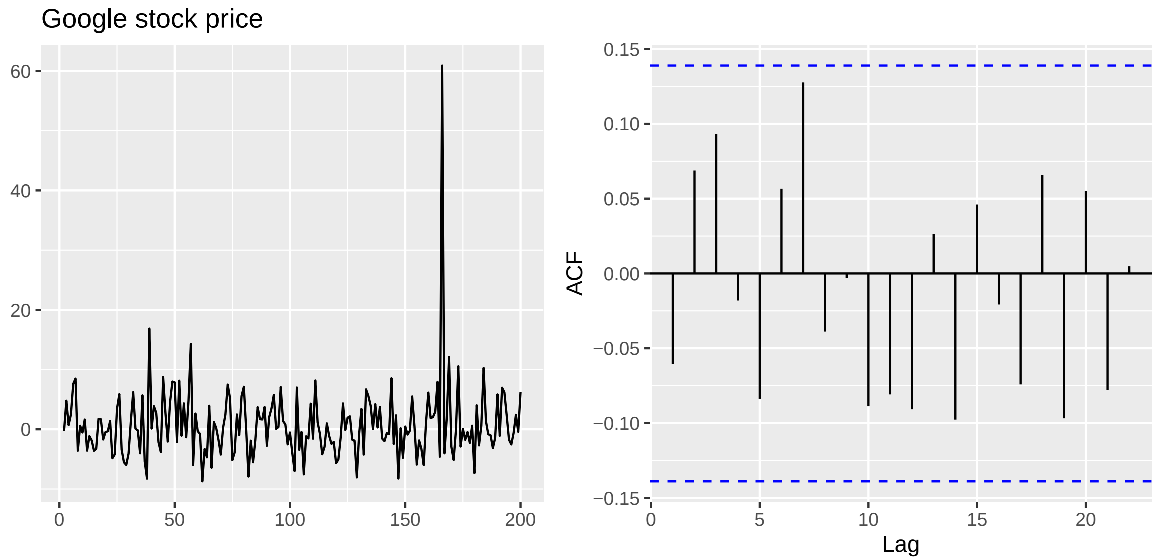


Figure 5: goog200 1 ACF

level , ACF . ,
.

goog200 . urca::ur.kpss() , forecast::ndiffs()
.

```
ndiffs(goog200)
## [1] 1
```

, 1 . urca::ur.kpss() . |> base R pipe
, %>% .

```
library(urca)
goog200 |>
  ur.kpss() |>
  summary()
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
```

```
## Value of test-statistic is: 2.7441
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

1% . 1 , 1
.

```
goog200 |>
  diff() |>
  ur.kpss() |>
  summary()
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.1163
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

- : 1956 1 1995 8 (`fma::elec`)

Figure 1 `elec` , . .
ACF Figure 6 .

```
ln_elec <- log(elec)
p1 <- autoplot(ln_elec) +
  labs(x = NULL, y = NULL, title = "log transformed data")
p2 <-  ggAcf(ln_elec) + ggtitle("")
p1 + p2
```

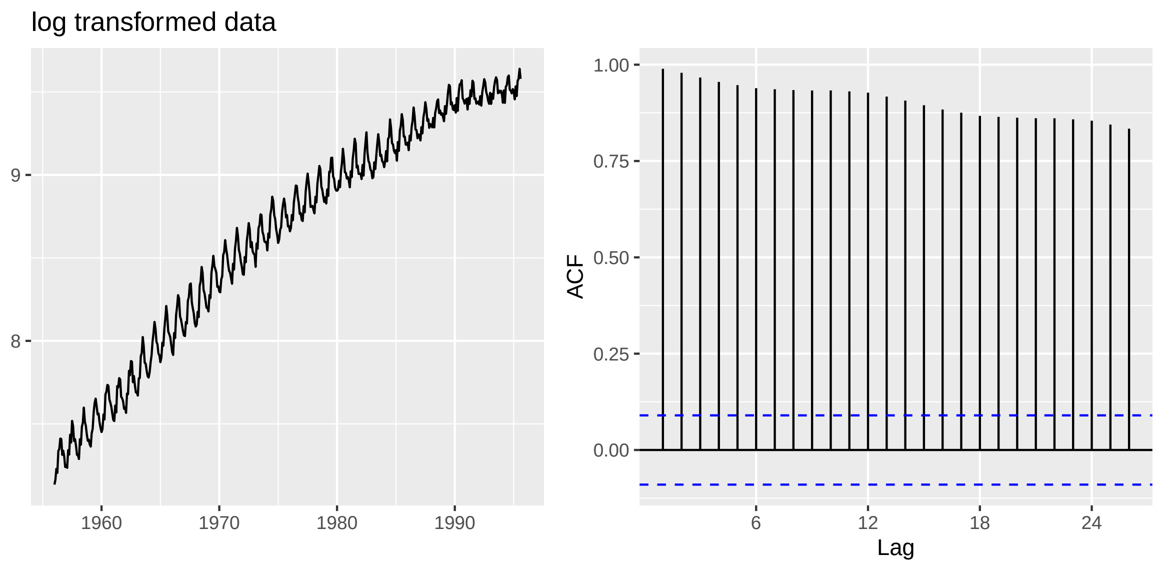


Figure 6: `elec` ACF

. `forecast::nsdiffs()`
. `nsdiffs()` seasonal strength .

```
elec |>
  log() |>
  nsdiffs()
## [1] 1
```

ACF . Figure 7 .

```
ln_elec_m <- log(elec) %>%
  diff(lag = 12)
p3 <- autoplot(ln_elec_m) +
  labs(x = NULL, y = NULL, title = "log transformed and seasonally differenced data")
p4 <-  ggAcf(ln_elec_m) + ggtitle("")
p3 + p4
```

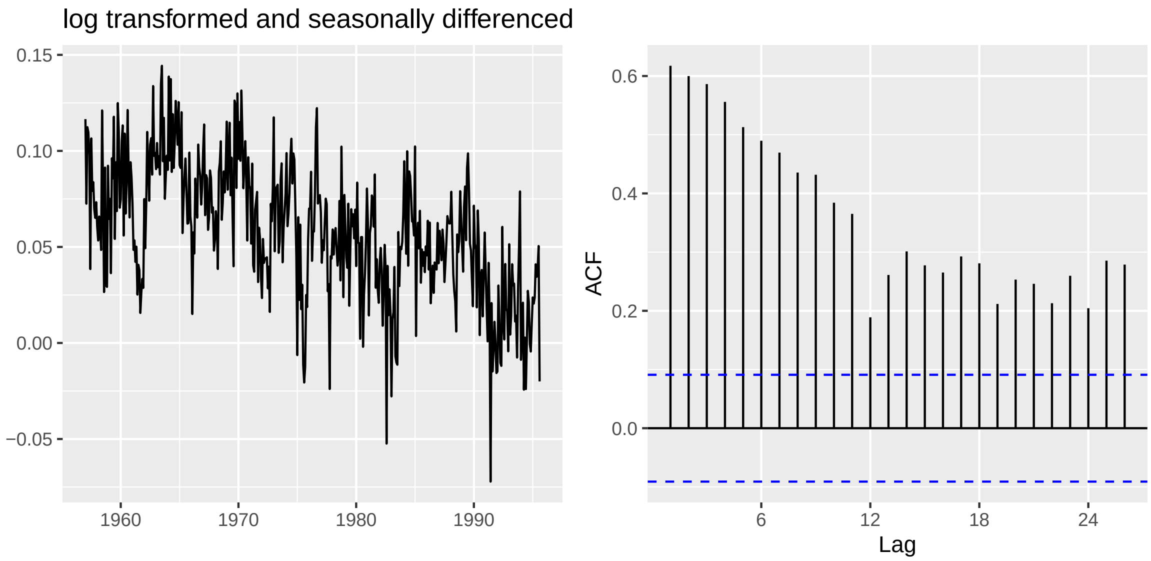


Figure 7: `elec` ACF

Figure 7 . 1 . Figure 8 ,
.

```
ln_elec_m_1 <- log(elec) %>%
  diff(lag = 12) %>%
  diff()
p5 <- autoplot(ln_elec_m_1) +
  labs(x = NULL, y = NULL, title = "log transformed and doubly differenced data")
p6 <-  ggAcf(ln_elec_m_1) + ggtitle("")
p5 + p6
```

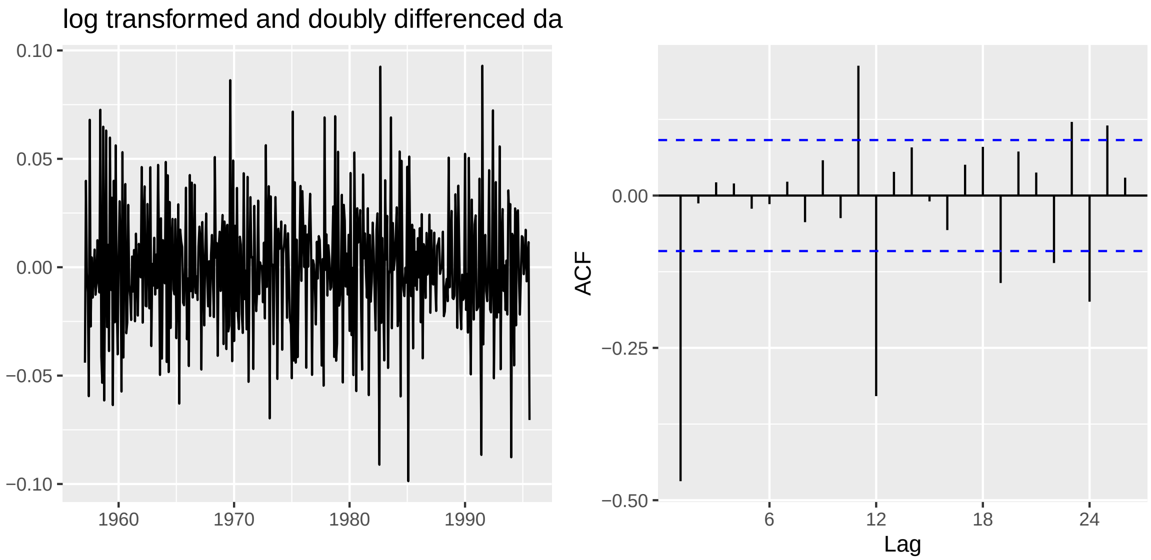


Figure 8: `elec` 1 ACF

ARIMA

- 1: gas

`gas.csv` 9 (`rate`) (`co2`) . .

```
gas <- readr::read_csv("https://raw.githubusercontent.com/yjyjpark/TS-with-R/main/Data/gas
gas %>%
  print(n = 5)
## # A tibble: 296 x 2
##      rate   co2
##     <dbl> <dbl>
## 1 -0.109  53.8
## 2  0      53.6
## 3  0.178  53.5
## 4  0.339  53.5
## 5  0.373  53.4
## # ... with 291 more rows
```

`gas` `rate` `co2` . `rate` `ts` , $t = 1, 2, 3, \ldots$ 1 . `as.ts()` . Figure 9 .

```r
rate.ts <- as.ts(gas$rate)
autoplot(rate.ts) + labs(y = NULL)
```

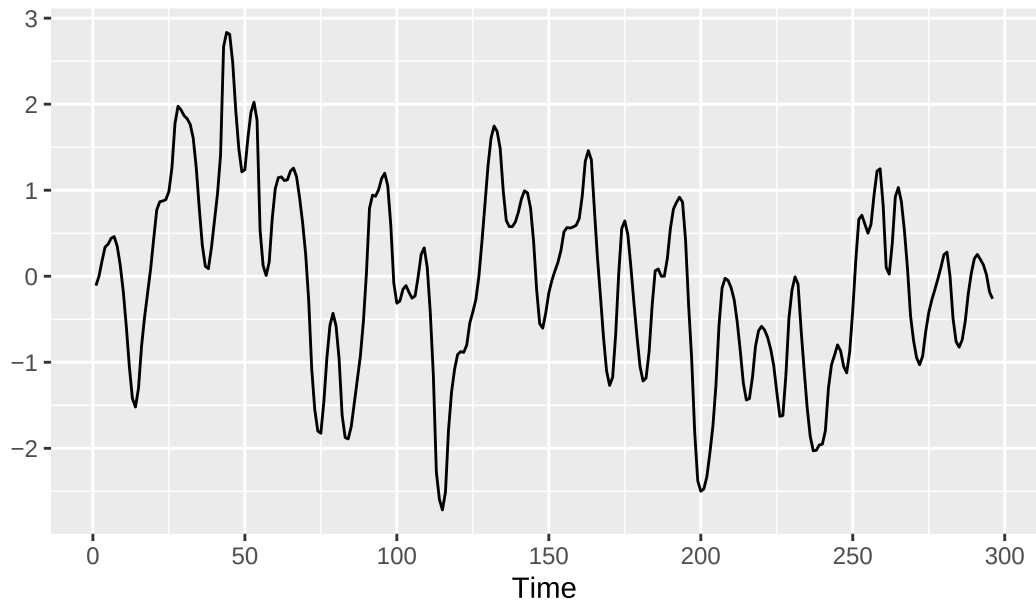


Figure 9: `gas`

`rate.ts` ARIMA . training data test data . Test data 10 .

```r
train_r <- window(rate.ts, end = length(rate.ts) - 10)
test_r <- window(rate.ts, start = length(rate.ts) - 9)
```

Training data . ACF . Figure 10 .

```r
ggtsdisplay(train_r)
```

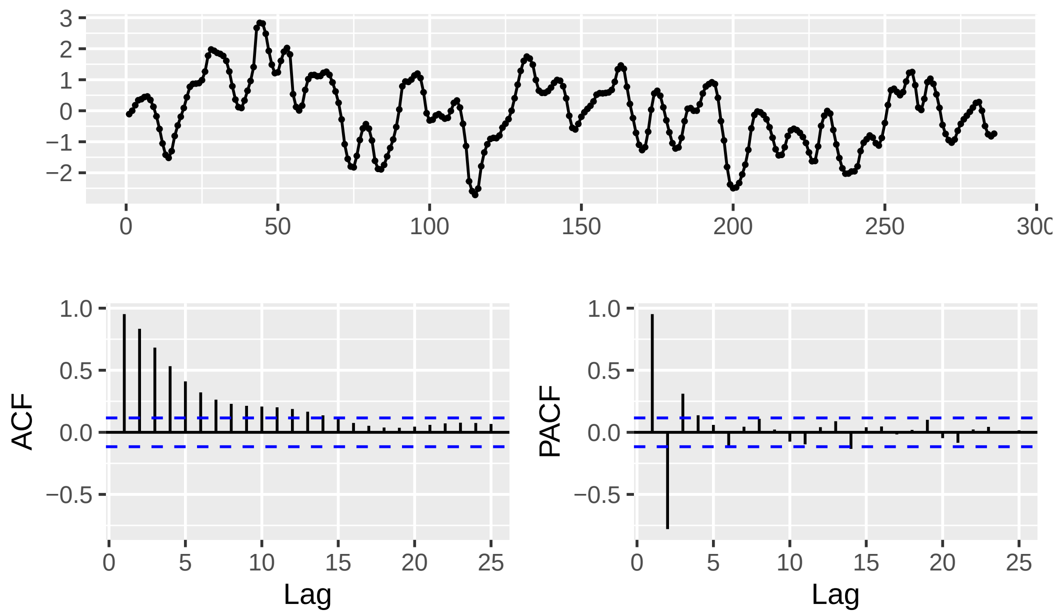


Figure 10: `train_r`

Figure 10 , ACF . .

```r
ndiffs(train_r)
## [1] 1
```

, ACF ,
.

. `d = 0` , `stepwise`, `approximation`, `seasonal`
`FALSE` . `trace = TRUE` AICc .

```r
fit1 <- auto.arima(train_r, d = 0, stepwise = FALSE,
                   approximation = FALSE, seasonal = FALSE,
                   trace = TRUE)
##
##  ARIMA(0,0,0) with zero mean     : 862.9984
##  ARIMA(0,0,0) with non-zero mean : 864.2264
##  ARIMA(0,0,1) with zero mean     : 502.5603
##  ARIMA(0,0,1) with non-zero mean : 503.849
##  ARIMA(0,0,2) with zero mean     : 245.7837
##  ARIMA(0,0,2) with non-zero mean : 247.1783
```

```
## ARIMA(0,0,3) with zero mean     : 81.487
## ARIMA(0,0,3) with non-zero mean : 83.02667
## ARIMA(0,0,4) with zero mean     : -20.65077
## ARIMA(0,0,4) with non-zero mean : -19.01766
## ARIMA(0,0,5) with zero mean     : -84.62715
## ARIMA(0,0,5) with non-zero mean : -82.85418
## ARIMA(1,0,0) with zero mean     : 184.3482
## ARIMA(1,0,0) with non-zero mean : 186.3166
## ARIMA(1,0,1) with zero mean     : -16.72114
## ARIMA(1,0,1) with non-zero mean : -14.73886
## ARIMA(1,0,2) with zero mean     : -74.34891
## ARIMA(1,0,2) with non-zero mean : -72.35703
## ARIMA(1,0,3) with zero mean     : -115.7122
## ARIMA(1,0,3) with non-zero mean : -113.7137
## ARIMA(1,0,4) with zero mean     : -133.8983
## ARIMA(1,0,4) with non-zero mean : -131.9025
## ARIMA(2,0,0) with zero mean     : -89.86221
## ARIMA(2,0,0) with non-zero mean : -88.00051
## ARIMA(2,0,1) with zero mean     : -113.8044
## ARIMA(2,0,1) with non-zero mean : -111.8785
## ARIMA(2,0,2) with zero mean     : -116.6196
## ARIMA(2,0,2) with non-zero mean : -114.6669
## ARIMA(2,0,3) with zero mean     : -130.7106
## ARIMA(2,0,3) with non-zero mean : -128.7274
## ARIMA(3,0,0) with zero mean     : -123.114
## ARIMA(3,0,0) with non-zero mean : -121.1507
## ARIMA(3,0,1) with zero mean     : -125.3355
## ARIMA(3,0,1) with non-zero mean : -123.3294
## ARIMA(3,0,2) with zero mean     : -123.8663
## ARIMA(3,0,2) with non-zero mean : -121.8527
## ARIMA(4,0,0) with zero mean     : -125.2705
## ARIMA(4,0,0) with non-zero mean : -123.2711
## ARIMA(4,0,1) with zero mean     : -123.4833
## ARIMA(4,0,1) with non-zero mean : -121.4641
## ARIMA(5,0,0) with zero mean     : -124.0066
## ARIMA(5,0,0) with non-zero mean : -121.9828
##
##
##
## Best model: ARIMA(1,0,4) with zero mean
```

, ARMA(1,4) .

```r
fit1
## Series: train_r
## ARIMA(1,0,4) with zero mean
##
## Coefficients:
##          ar1     ma1     ma2     ma3     ma4
##       0.7769  1.1456  1.0384  0.7892  0.3022
## s.e.  0.0450  0.0657  0.0922  0.0880  0.0627
##
## sigma^2 = 0.03511:  log likelihood = 73.1
## AIC=-134.2   AICc=-133.9   BIC=-112.26
```

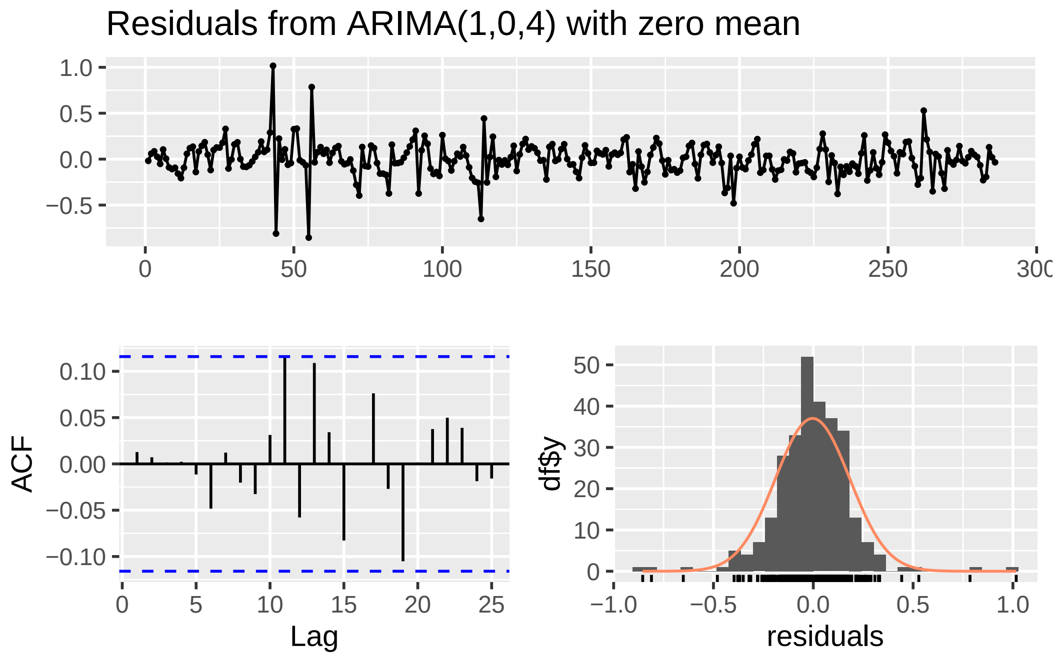
, ARIMA(3,1,1) .

```r
fit2 <- auto.arima(train_r, stepwise = FALSE,
                   approximation = FALSE, seasonal = FALSE)
```

```r
fit2
## Series: train_r
## ARIMA(3,1,1)
##
## Coefficients:
##          ar1      ar2     ar3      ma1
##       1.9589  -1.3503  0.3304  -0.9855
## s.e.  0.0580   0.1032  0.0576   0.0148
##
## sigma^2 = 0.03717:  log likelihood = 65.4
## AIC=-120.81   AICc=-120.59   BIC=-102.55
```

`fit1` . .

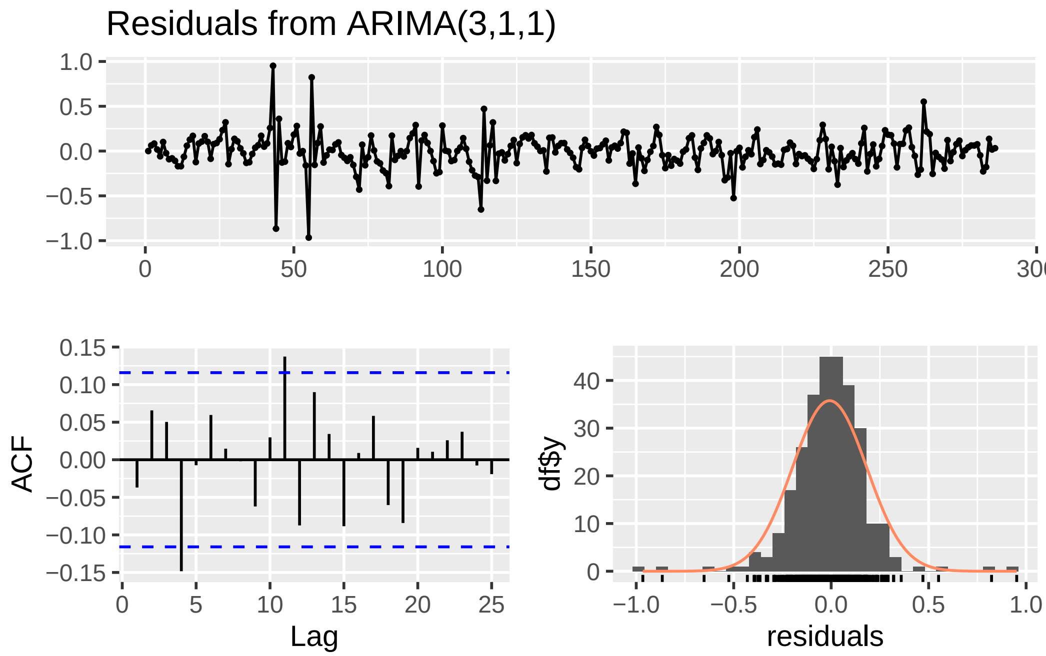
```r
checkresiduals(fit1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,4) with zero mean
## Q* = 1.5628, df = 5, p-value = 0.9057
##
## Model df: 5.   Total lags used: 10
```

fit2 . .

```
checkresiduals(fit2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,1)
## Q* = 11.352, df = 6, p-value = 0.07809
##
## Model df: 4.   Total lags used: 10
```

. , `fit1` , `fit2` .
, , AICc . , test data .
test data .

```r
fc1 <- forecast(fit1)
fc2 <- forecast(fit2)
```

```r
accuracy(fc1, test_r)
##                        ME      RMSE       MAE      MPE     MAPE      MASE
## Training set -0.003485182 0.1857419 0.1308396      NaN      Inf 0.5071232
## Test set      0.197714969 0.2984918 0.2589217 261.7854 282.7786 1.0035587
##                    ACF1 Theil's U
## Training set 0.01278907        NA
## Test set     0.63693511  1.231509
```

```r
accuracy(fc2, test_r)
##                        ME      RMSE       MAE      MPE     MAPE      MASE
## Training set -0.008064075 0.1911149 0.1333927      NaN      Inf 0.5170188
## Test set      0.325580419 0.3713446 0.3255804 390.3076 437.7347 1.2619224
##                     ACF1 Theil's U
## Training set -0.03694615        NA
## Test set      0.57275035  1.400976
```

`accuracy()` , ARMA(1,4) `fit1` . . .

$$(1 - 0.7769B)\; y_t = (1 + 1.145B + 1.038B^2 + 0.789B^3 + 0.302B^4)\; \varepsilon_t$$

. Test data , Figure 11 .

```r
autoplot(fc1, include = 20) +
  autolayer(test_r, color = "red", size = .8) +
  labs(y = "rate")
```

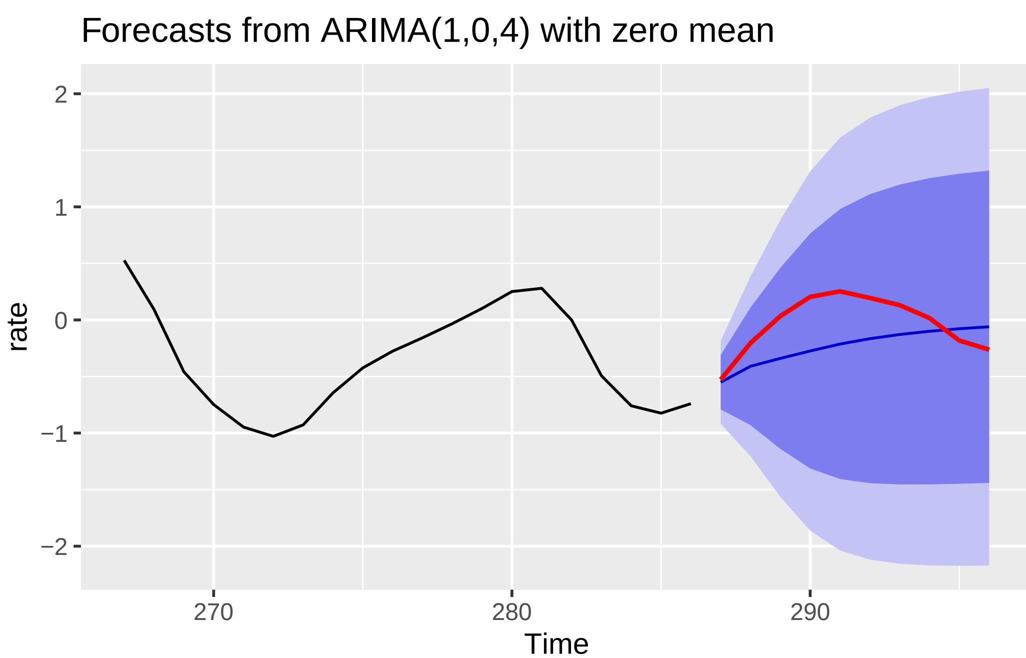


Figure 11: `rate`

- 2 : 1996 1 2012 3 Euro (`fpp2::elecequip`)

elecequip . ARIMA . stl() , forecast::seasadj() .

```r
elecequip_desea <- stl(elecequip, s.window="periodic") %>%
  seasadj()
```

Figure 12 .

```r
autoplot(elecequip, series = "Monthly data") +
  autolayer(elecequip_desea,
            series = "Seasonally adjusted", size = .8) +
  scale_color_manual(values = c("Monthly data" = "blue",
                                "Seasonally adjusted" = "red")) +
  theme(legend.position = "top") +
  labs(y = NULL, color = NULL)
```

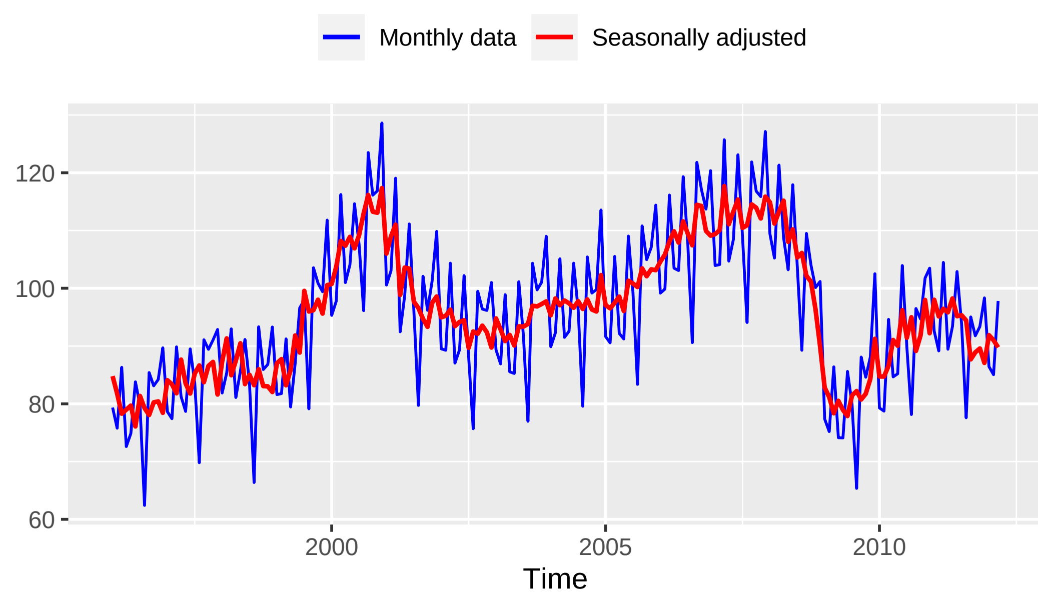


Figure 12: elecequip

. Test data 2 .

```r
train_eq <- window(elecequip_desea, end = c(2010,3))
test_eq <- window(elecequip_desea, start = c(2010,4))
```

Figure 13 .

```r
ggtsdisplay(train_eq)
```

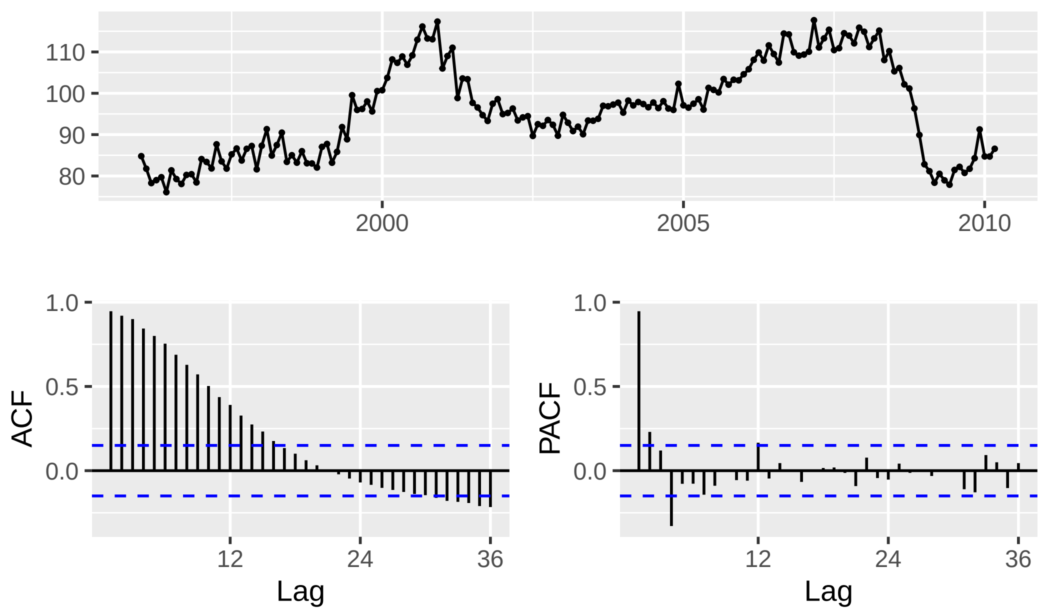


Figure 13: `train_eq`

level , ACF . .

```r
ndiffs(train_eq)
## [1] 1
```

ACF Figure 14 .

```r
train_eq %>%
  diff() %>%
  ggtsdisplay()
```

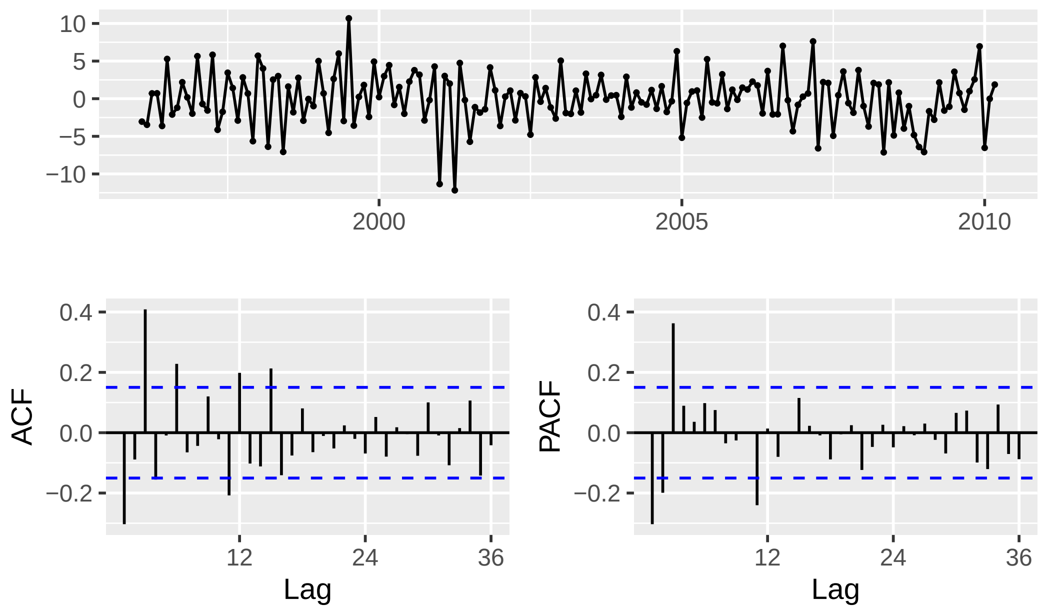


Figure 14: `train_eq`

. ACF , PACF 3 AR(3) , ARIMA(3,1,0)
.

`auto.arima()` .

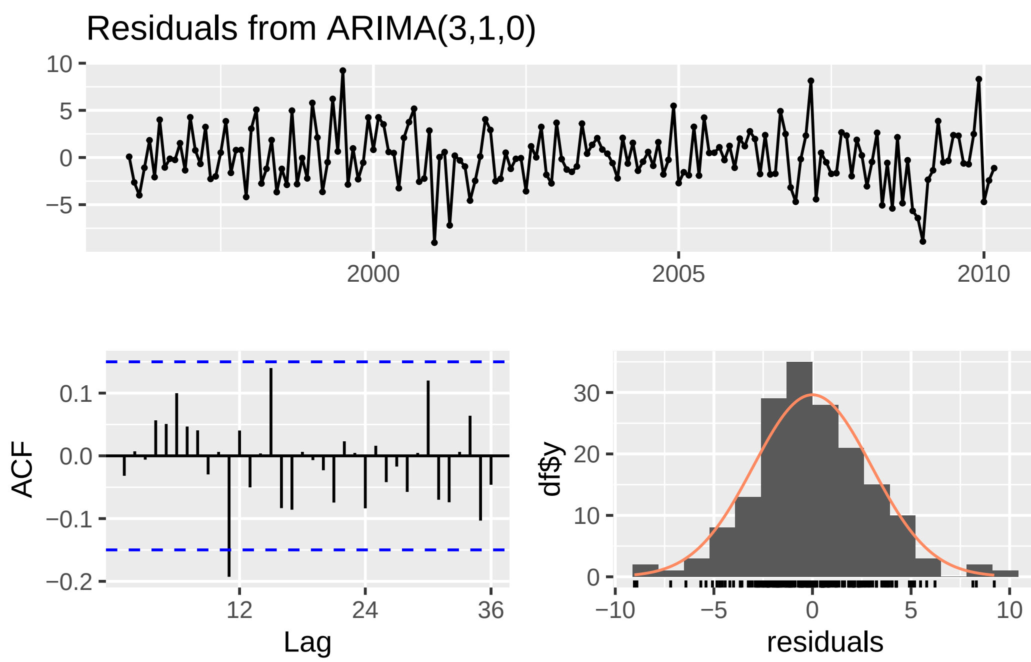
```
fit1 <- auto.arima(train_eq, stepwise = FALSE,
                      approximation = FALSE, seasonal = FALSE)
fit1
## Series: train_eq
## ARIMA(3,1,0)
##
## Coefficients:
##           ar1      ar2     ar3
##       -0.2922  -0.0635  0.3710
## s.e.   0.0713   0.0748  0.0718
##
## sigma^2 = 9.169:  log likelihood = -428.36
## AIC=864.73   AICc=864.97   BIC=877.27
```

ACF PACF ARIMA(3,1,0) . .

$$(1 + 0.292B + 0.064B^2 - 0.371B^3)(1 - B)\ y_t = \varepsilon_t$$

. , .

```r
checkresiduals(fit1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,0)
## Q* = 20.702, df = 21, p-value = 0.4772
##
## Model df: 3.   Total lags used: 24
```

, .

```r
fc1 <- forecast(fit1)
```

```r
accuracy(fc1, test_eq)
##                        ME      RMSE       MAE         MPE      MAPE       MASE
```

```
## Training set -0.002219982 2.992424 2.301026 -0.05081691 2.411491 0.2786977
## Test set      8.727507127 9.330976 8.727507  9.28015517 9.280155 1.0570657
##                     ACF1 Theil's U
## Training set -0.03178277        NA
## Test set      0.38874430  2.481532
```

Figure 15 . test data .

```
autoplot(fc1, include = 20) +
  autolayer(test_eq, color = "red", size = .8) +
  ylab("Electrical equipment manufactured")
```

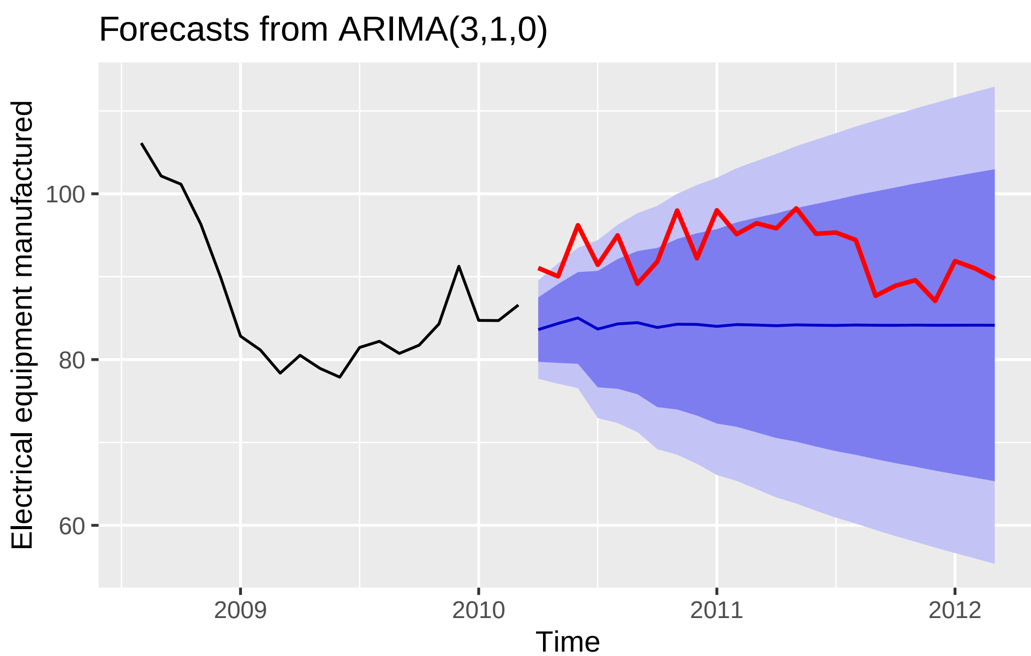


Figure 15: `elecequip`

ARIMA

- 1 : 1984 1 1988 12

1984 1 1988 12 ARIMA . training data test data . `ts` .

```r
depart <- scan("https://raw.githubusercontent.com/yjyjpark/TS-with-R/main/Data/depart.txt")
depart.ts <- ts(depart, start = 1984, freq = 12)
```

`depart.ts` . Figure 16 .
.

```r
autoplot(depart.ts) +
  ylab(NULL)
```

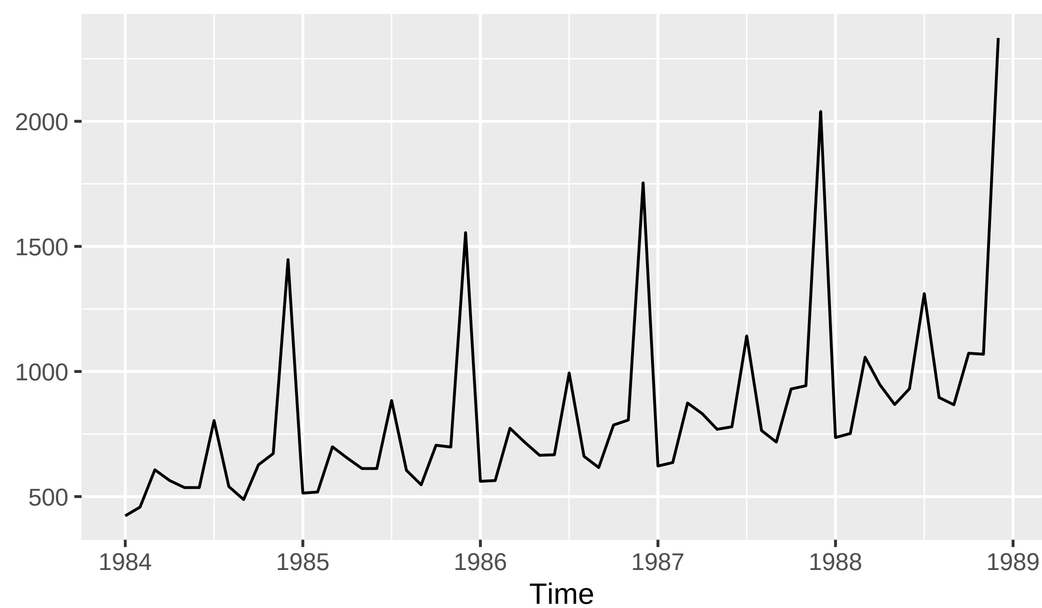


Figure 16:

, . Figure 17 .

```r
lndepart <- log(depart.ts)
autoplot(lndepart) +
  labs(title = "log(depart.ts)", y = NULL)
```

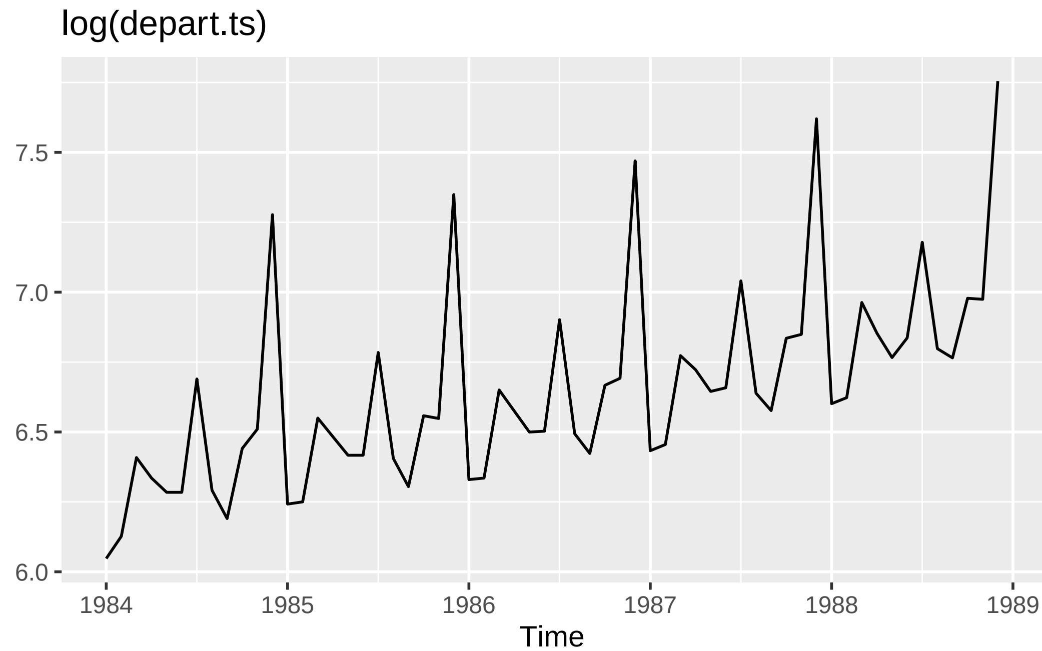


Figure 17:

. . ACF, .
. 1 .

```
ndiffs(lndepart)
## [1] 1
nsdiffs(lndepart)
## [1] 1
```

ACF Figure 17 .

```
ggtsdisplay(lndepart, lag.max = 36,
            main = "lndepart: log transformed data")
```

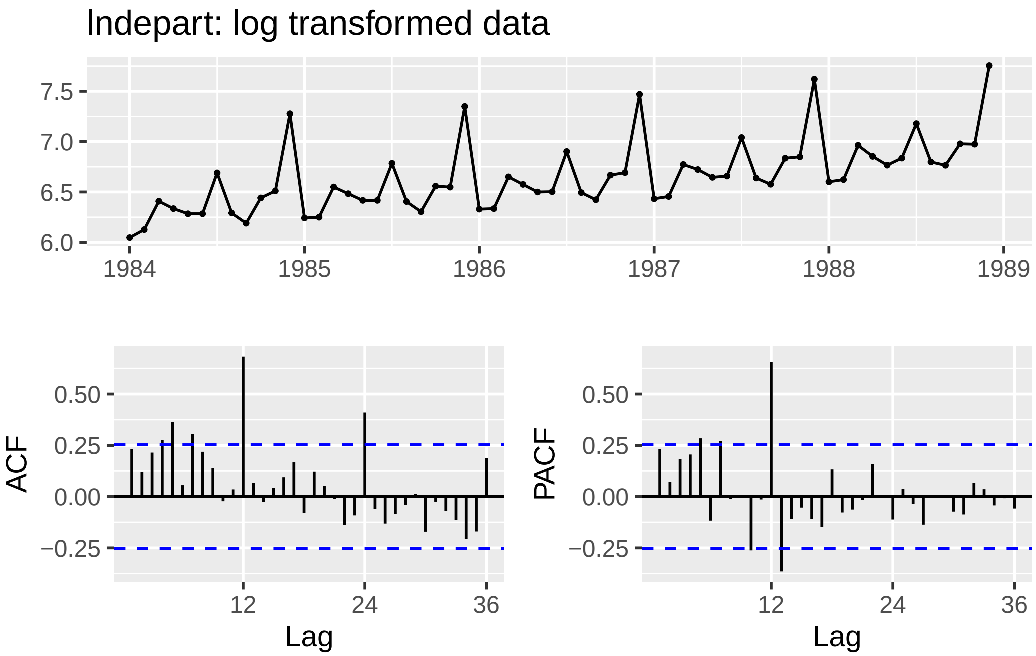


Figure 18: `lndepart` , ACF PACF

. ACF . Figure 19 ACF 1 6 , .

```r
lndepart_12 <- diff(lndepart, lag = 12)
ggtsdisplay(lndepart_12, lag.max = 36,
            main = "Seasonally differenced lndepart")
```

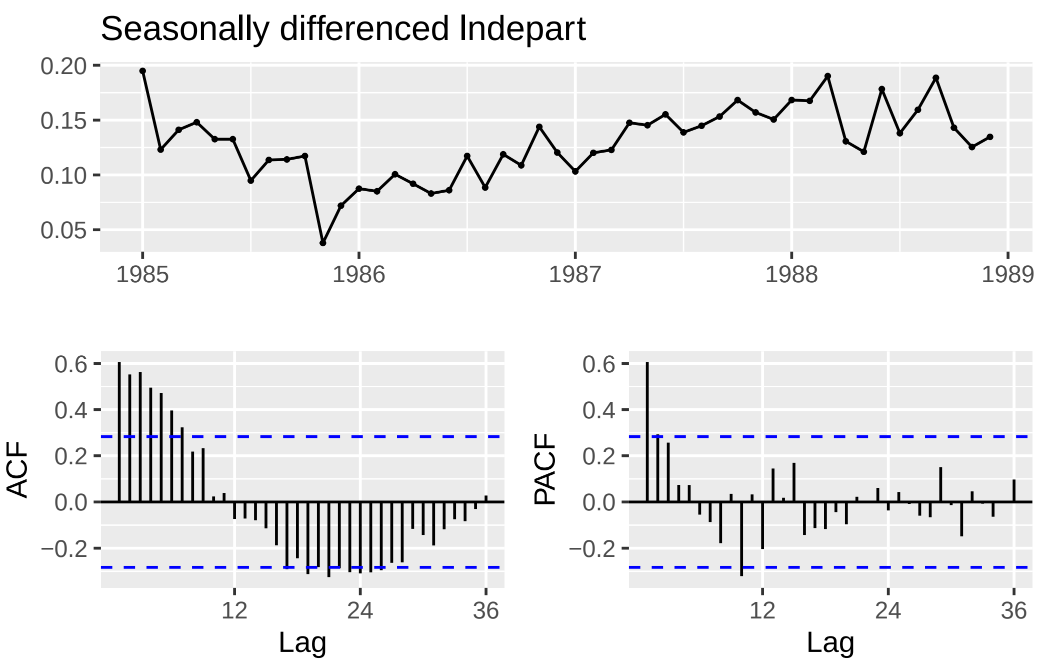


Figure 19: `lndepart` , ACF PACF

1 ACF . Figure 20 . , ,
ACF 1 .

```
lndepart_12_1 <- diff(lndepart_12)
ggtsdisplay(lndepart_12_1, lag.max = 36,
            main = "Doubly differenced lndepart")
```

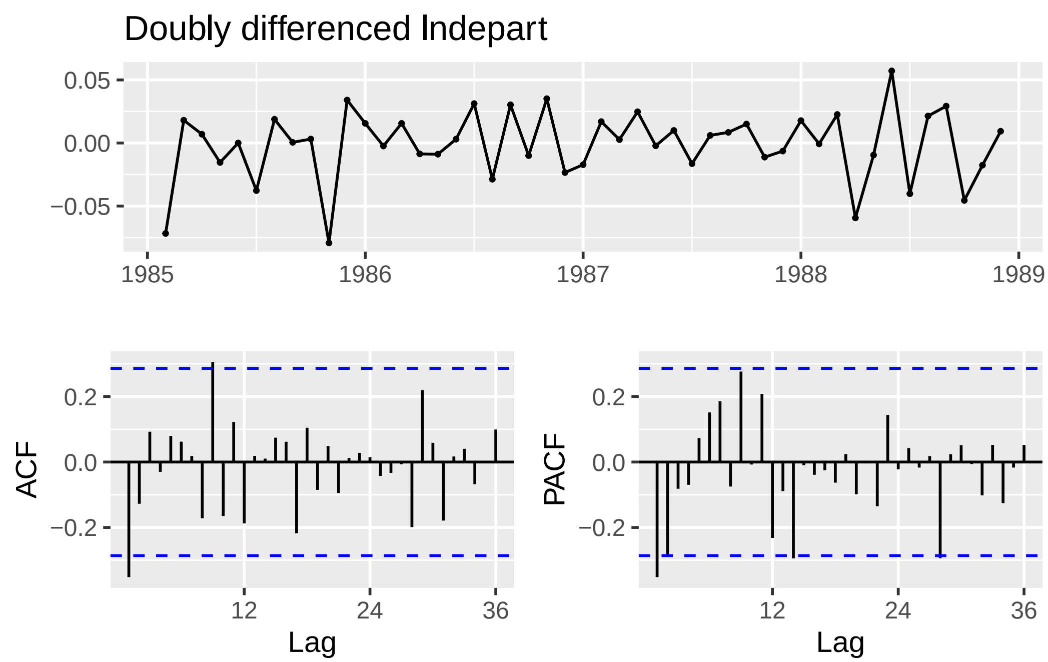


Figure 20: 1 `lndepart` , ACF PACF

1 . Figure 20 ACF PACF . AR MA 1 6 , ACF 1 , , PACF 1, 2 , . ACF 1 , PACF , `p=0, q=1` MA(1) .

12, 24, 36 , ACF PACF 12, 24, 36 . `P=0, Q=0` , `P=1, Q=0` `P=0, Q=1` .

ACF PACF , ARIMA$(0,1,1)(0,1,0)_{12}$ ARIMA$(0,1,1)(1,1,0)_{12}$, ARIMA$(0,1,1)(0,1,1)_{12}$.

`auto.arima()` . `auto.arima()` , , `d` `D` . `lambda` 0 .

```
fit_d <- auto.arima(depart.ts, lambda = 0,
                    stepwise = FALSE, approximation = FALSE)
```

```
summary(fit_d)
## Series: depart.ts
## ARIMA(0,1,1)(0,1,1)[12]
## Box Cox transformation: lambda= 0
```

```
##
## Coefficients:
##           ma1     sma1
##       -0.5840  -0.4159
## s.e.   0.1093   0.1946
##
## sigma^2 = 0.0005401:  log likelihood = 110.29
## AIC=-214.59   AICc=-214.03   BIC=-209.04
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE    MAPE      MASE       ACF1
## Training set -1.937472 16.9307 11.60509 -0.200563 1.36766 0.1084166 0.04358066
```

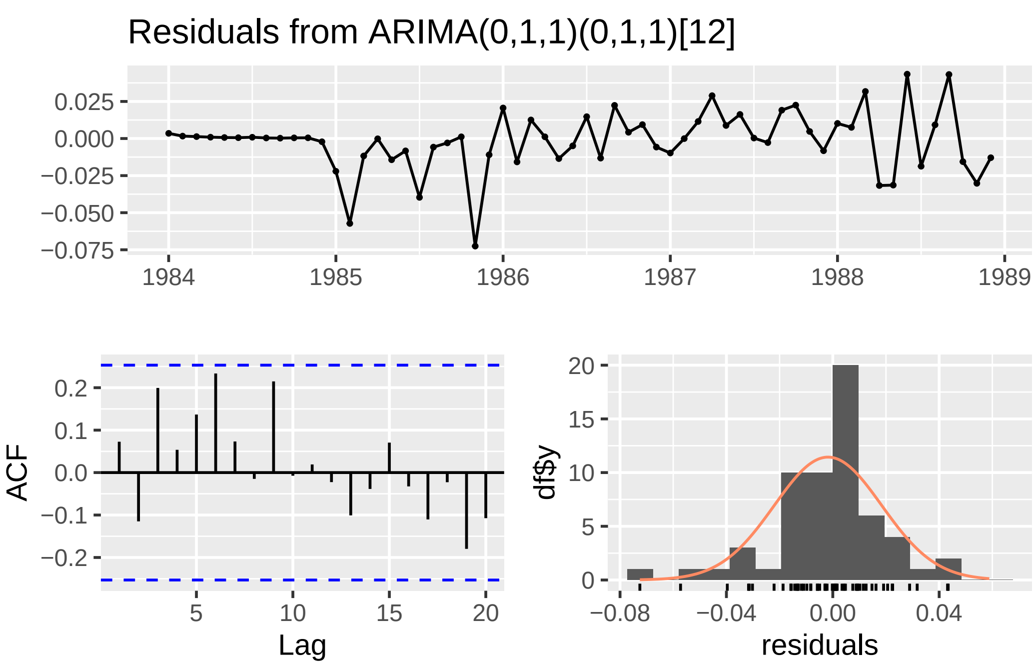
ACF PACF ARIMA$(0,1,1)(0,1,1)_{12}$.

.

$$(1-B^{12})(1-B)\ \log y_t = (1-0.584B)(1-0.4159B^{12})\ \varepsilon_t$$

fit_d . , .

```
checkresiduals(fit_d)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)(0,1,1)[12]
## Q* = 12.817, df = 10, p-value = 0.2341
##
## Model df: 2.   Total lags used: 12
```

Figure 21 . .

```
forecast(fit_d) %>%
  autoplot() + ylab(NULL)
```

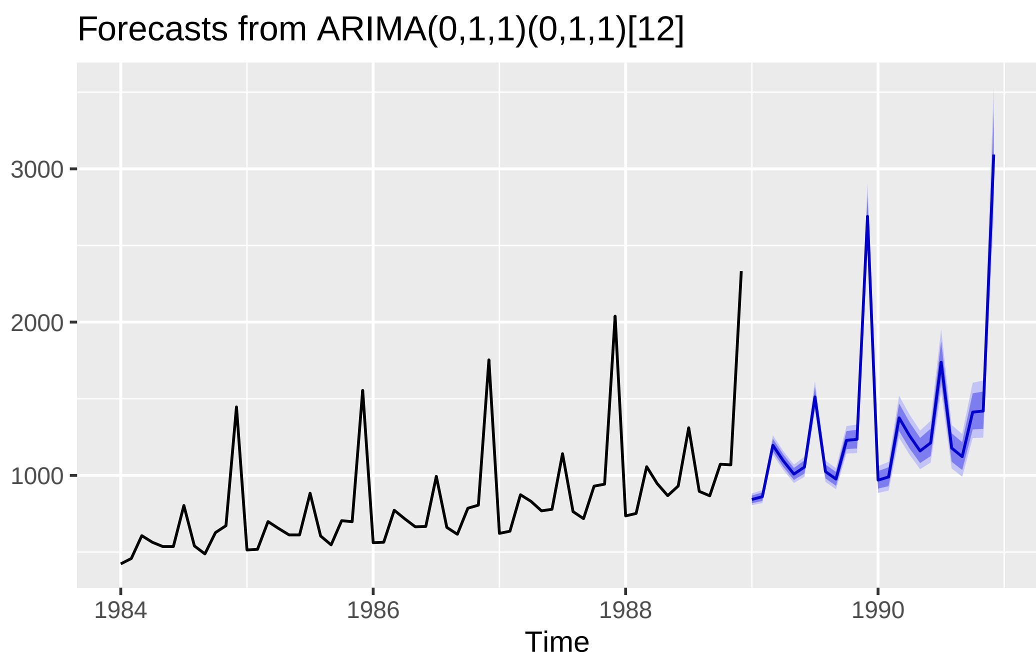


Figure 21:

- 2 : 1981 1 1992 12

1981 1 1992 12 12 ARIMA ETS , . 2 test data , 10 .

```
tour <- scan("https://raw.githubusercontent.com/yjyjpark/TS-with-R/main/Data/Ktour.txt")
tour.ts <- ts(tour, start = 1981, freq = 12)
```

```r
train_K <- window(tour.ts, end = c(1990,12))
test_K <- window(tour.ts, start = c(1991,1))
```

Training data Figure 22 .

```r
autoplot(train_K) +
  ylab(NULL)
```

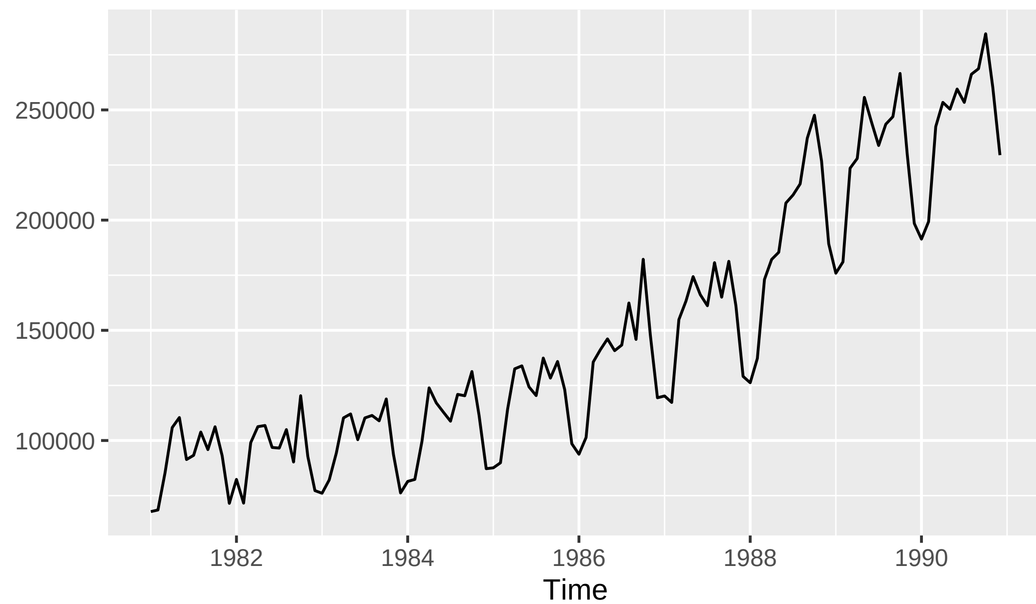


Figure 22:

ARIMA . , , . ,
Box-Cox λ .

```r
BoxCox.lambda(train_K)
## [1] 0.09573094
```

$\hat{\lambda} = 0.096$, $y_t^{0.09}$ λ $\hat{\lambda} = 0$.
. ACF Figure 23 .

```r
lntrain_K <- log(train_K)
ggtsdisplay(lntrain_K,
            main = "log transformed:lntrain_K")
```

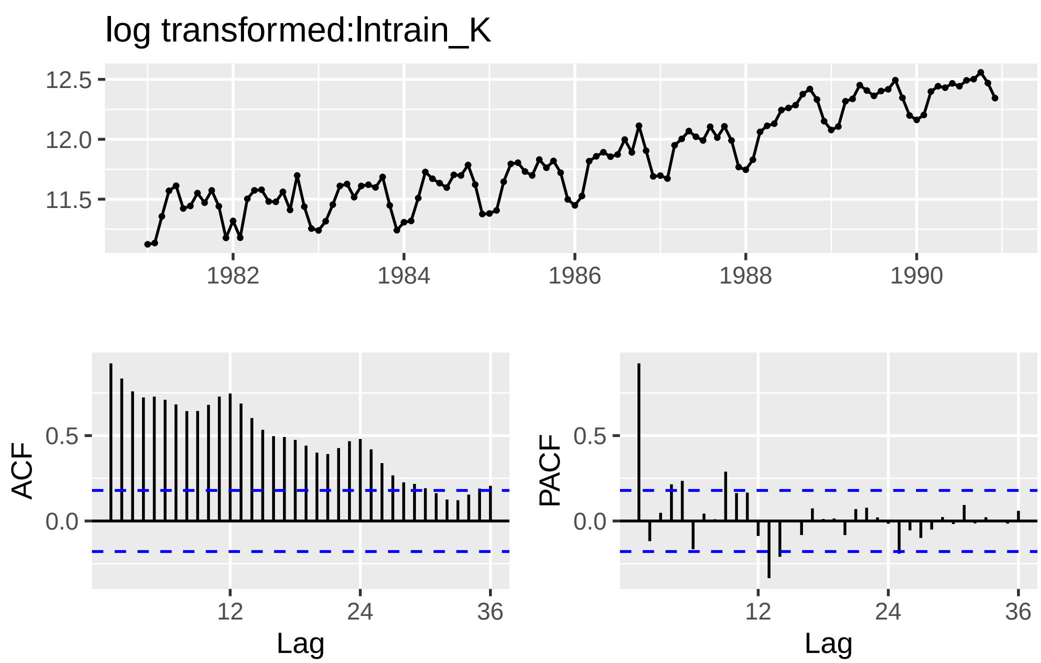


Figure 23: `lntrain_K` , ACF PACF

, . Figure 24 , ACF 1~6 . 1 .

```r
lntrain_K_12 <- diff(lntrain_K, lag = 12)
ggtsdisplay(lntrain_K_12,
            main = "Seasonally differenced")
```

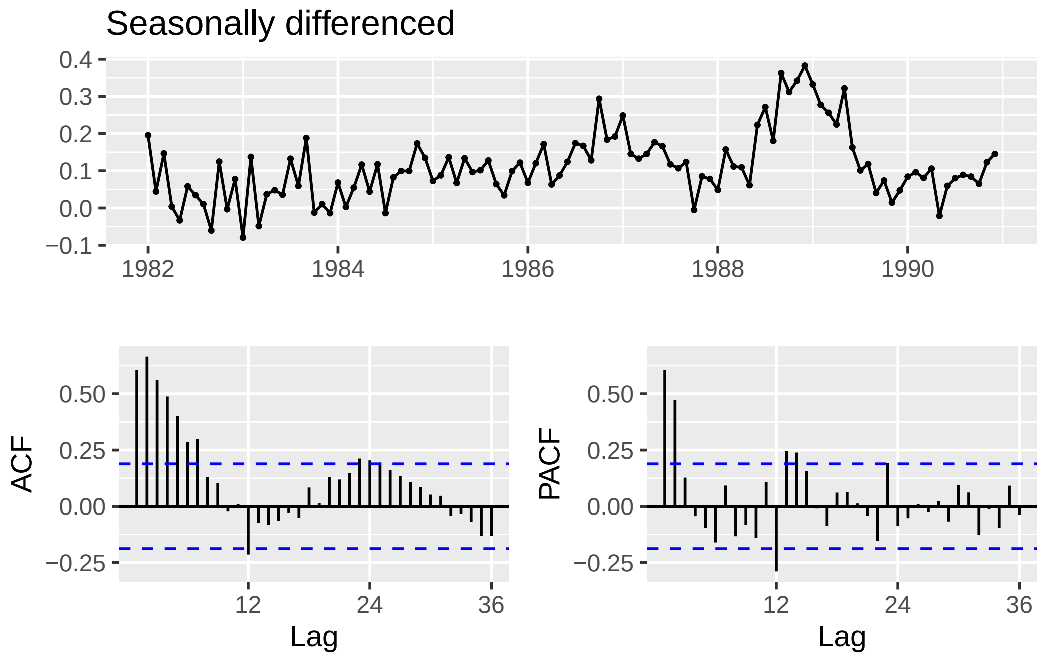


Figure 24: `lntrain_K` , ACF PACF

1 , . Figure 25 .

```
lntrain_K_12_1 <- diff(lntrain_K_12)
ggtsdisplay(lntrain_K_12_1,
            main = "Doubly differenced")
```

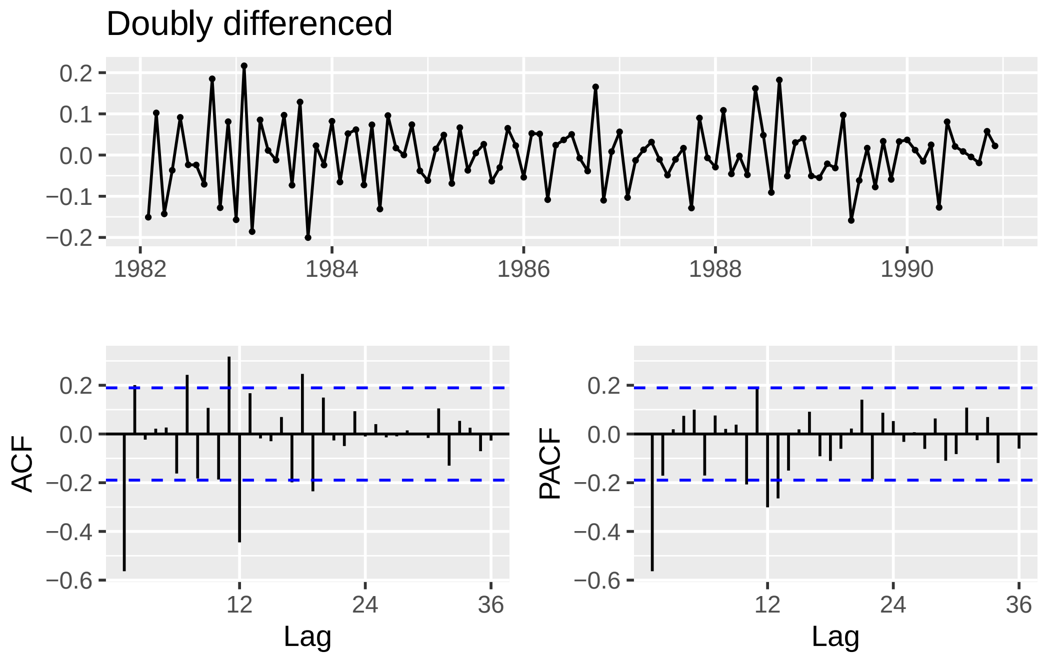


Figure 25: 1 `lntrain_K` , ACF PACF

.

```r
ndiffs(lntrain_K)
## [1] 1
nsdiffs(lntrain_K)
## [1] 1
```

ACF PACF Figure 25 . 1 6 ARIMA . ACF 2 , PACF 1 , 2 . , AR(1), AR(2), MA(2), ARMA(1,1) .

ARIMA ACF PACF 12 , 24 36 ACF PACF , , $AR(1)_{12}$, $MA(1)_{12}$, $ARMA(1,1)_{12}$.

, . $ARIMA(1,1,0)(1,1,0)_{12}$, $ARIMA(1,1,0)(0,1,1)_{12}$, $ARIMA(1,1,0)(1,1,1)_{12}$ $ARIMA(2,1,0)(1,1,0)_{12}$, $ARIMA(2,1,0)(0,1,1)_{12}$, $ARIMA(2,1,0)(1,1,1)_{12}$, $ARIMA(1,1,1)(1,1,0)_{12}$, $ARIMA(1,1,1)(0,1,1)_{12}$, $ARIMA(1,1,1)(1,1,1)_{12}$.

ACF PACF , . , `auto.arima()` ACF PACF .

`auto.arima()` .

```r
fit_K <- auto.arima(train_K, lambda = 0,
                    stepwise = FALSE, approximation = FALSE)
```

```r
summary(fit_K)
## Series: train_K
## ARIMA(2,1,0)(1,1,1)[12]
## Box Cox transformation: lambda= 0
##
## Coefficients:
##           ar1      ar2     sar1     sma1
##       -0.6995  -0.2496  -0.2892  -0.3817
## s.e.   0.0956   0.0947   0.1695   0.1736
##
## sigma^2 = 0.002917:  log likelihood = 159.82
## AIC=-309.63   AICc=-309.04   BIC=-296.27
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -117.8446 8079.833 5752.686 -0.1325567 3.815935 0.3223641
##                   ACF1
## Training set 0.1387895
```

ARIMA(2,1,0)(1,1,1)$_{12}$, ACF PACF . .

$$(1 + 0.69B + 0.25B^2)(1 + 0.29B^{12})(1 - B)(1 - B^{12})\ \log y_t = (1 - 0.382^{12})\ \varepsilon_t$$

ETS . ETS . ETS ETS
.

.

```r
fit_K_ets1 <- ets(train_K, lambda = 0)
```

```r
summary(fit_K_ets1)
## ETS(A,Ad,A)
##
## Call:
##  ets(y = train_K, lambda = 0)
##
##   Box-Cox transformation: lambda= 0
##
```

```
##   Smoothing parameters:
##     alpha = 0.3998
##     beta  = 0.0242
##     gamma = 1e-04
##     phi   = 0.978
##
##   Initial states:
##     l = 11.3617
##     b = 0.0098
##     s = -0.2089 -0.0012 0.1585 0.0577 0.1123 0.035
##            0.0516 0.1154 0.0989 0.0121 -0.2092 -0.2221
##
##   sigma:  0.0548
##
##        AIC       AICc        BIC
## -104.72171  -97.94943  -54.54685
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
## Training set 1116.089 8322.678 5764.589 0.4909287 3.788076 0.3230311 0.09314822
```

ETS(A,Ad,A) .

.

```
fit_K_ets2 <- ets(train_K)
```

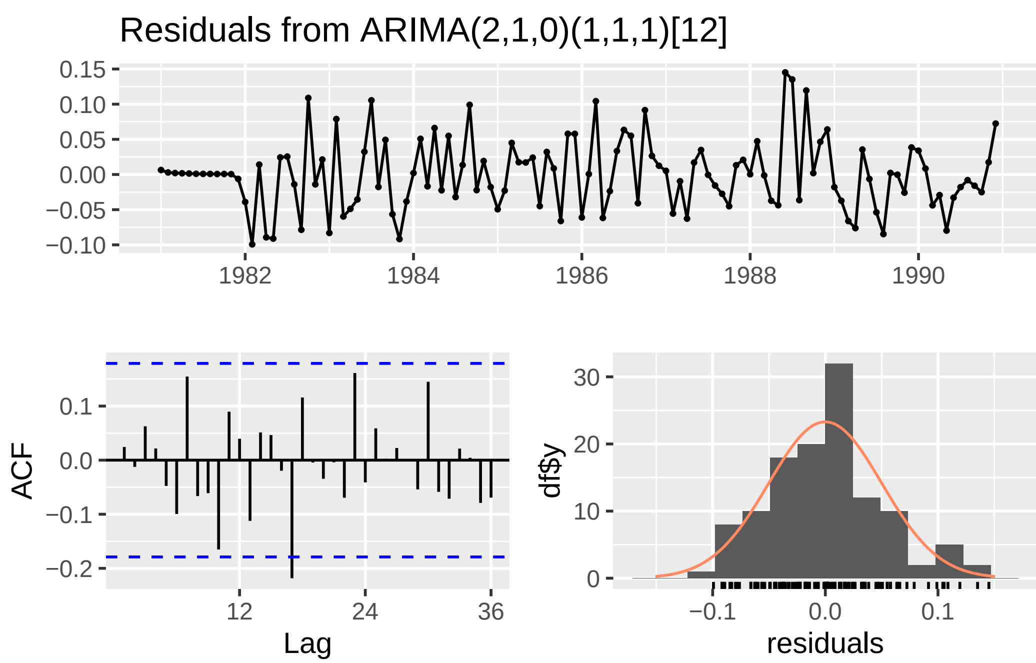
```
summary(fit_K_ets2)
## ETS(M,Ad,M)
##
## Call:
##  ets(y = train_K)
##
##   Smoothing parameters:
##     alpha = 0.4275
##     beta  = 0.0306
##     gamma = 1e-04
##     phi   = 0.9799
##
##   Initial states:
##     l = 89459.3204
```

```
##      b = 291.8495
##      s = 0.808 0.9953 1.172 1.0557 1.1073 1.0353
##             1.0453 1.1046 1.0949 0.9939 0.7992 0.7886
##
##   sigma:  0.0562
##
##      AIC     AICc      BIC
## 2738.255 2745.028 2788.430
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
## Training set 1147.204 8155.819 5774.677 0.5763675 3.839131 0.3235964 0.02357097
```

ETS(M,Ad,M) . ETS 'damped additive' , 'additive' , 'multiplicative' .

ARIMA ETS . ARIMA .

```r
checkresiduals(fit_K)
```



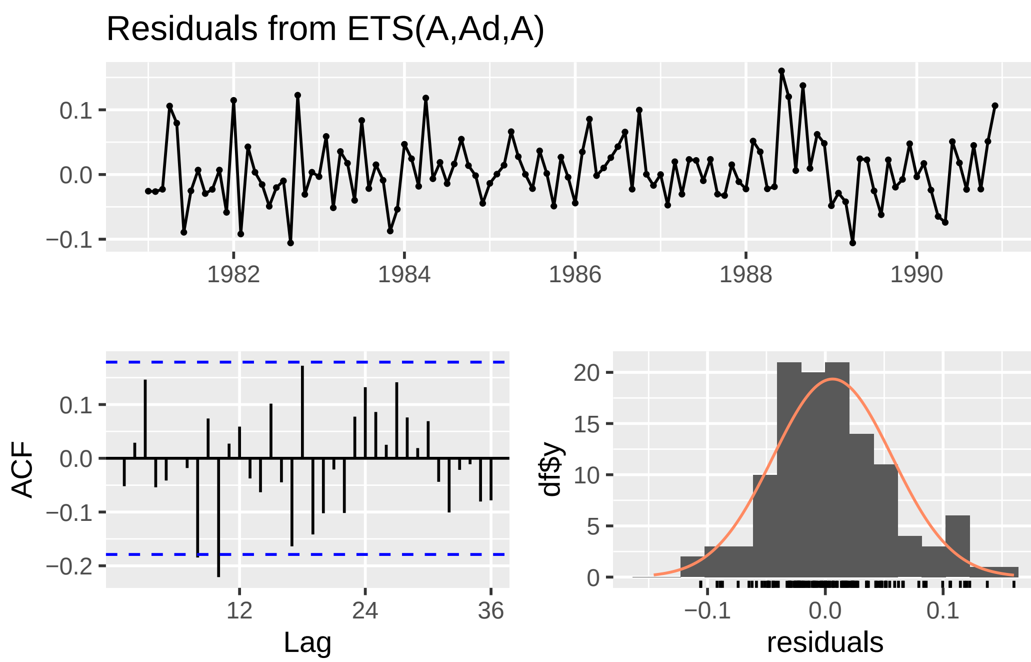

```
##
##  Ljung-Box test
```

```
##
## data:  Residuals from ARIMA(2,1,0)(1,1,1)[12]
## Q* = 27.495, df = 20, p-value = 0.1219
##
## Model df: 4.   Total lags used: 24
```

ETS Ljung-Box p-value , . ETS ,
.

. ETS(A,Ad,A) `fit_K_ets1` .

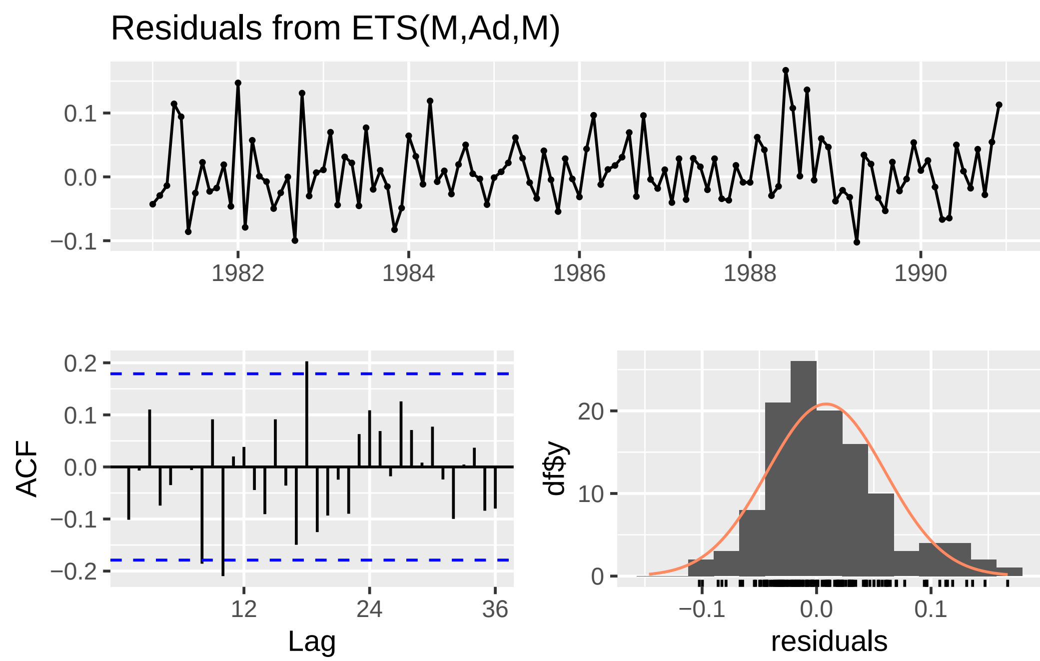
```
checkresiduals(fit_K_ets1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,Ad,A)
## Q* = 36.158, df = 7, p-value = 6.769e-06
##
## Model df: 17.   Total lags used: 24
```

ETS(M,Ad,M) `fit_K_ets2` .

```r
checkresiduals(fit_K_ets2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,Ad,M)
## Q* = 34.451, df = 7, p-value = 1.418e-05
##
## Model df: 17.   Total lags used: 24
```

test data   .

```r
fc_K <- forecast(fit_K)
accuracy(fc_K, test_K)
##                       ME       RMSE         MAE         MPE      MAPE      MASE
## Training set   -117.8446   8079.833   5752.686  -0.1325567  3.815935 0.3223641
## Test set     -33473.2740 45118.350 37348.980 -12.4581745 13.967241 2.0929303
##                   ACF1 Theil's U
## Training set 0.1387895        NA
## Test set     0.6467465   1.67741
```

```r
fc_K_ets1 <- forecast(fit_K_ets1)
accuracy(fc_K_ets1, test_K)
##                       ME      RMSE       MAE        MPE      MAPE      MASE
## Training set    1116.089  8322.678  5764.589  0.4909287  3.788076 0.3230311
## Test set      -26763.496 38813.714 33571.599 -9.9007782 12.507782 1.8812566
##                     ACF1 Theil's U
## Training set 0.09314822        NA
## Test set     0.55156580  1.469369
```

```r
fc_K_ets2 <- forecast(fit_K_ets2)
accuracy(fc_K_ets2, test_K)
##                       ME      RMSE       MAE        MPE      MAPE      MASE
## Training set    1147.204  8155.819  5774.677  0.5763675  3.839131 0.3235964
## Test set      -25539.311 37954.952 32640.675 -9.4299420 12.153867 1.8290903
##                     ACF1 Theil's U
## Training set 0.02357097        NA
## Test set     0.54315176  1.439223
```

ETS(M,Ad,M) `fit_K_ets2` . ARIMA ETS(M,Ad,M) Figure 26 .

```r
p1 <- autoplot(fc_K, include = 24) +
  autolayer(test_K, color = "red", size = .8) +
  ylab(NULL)

p2 <- autoplot(fc_K_ets2, include = 24) +
  autolayer(test_K, color = "red", size = .8) +
  ylab(NULL)
p1 / p2
```

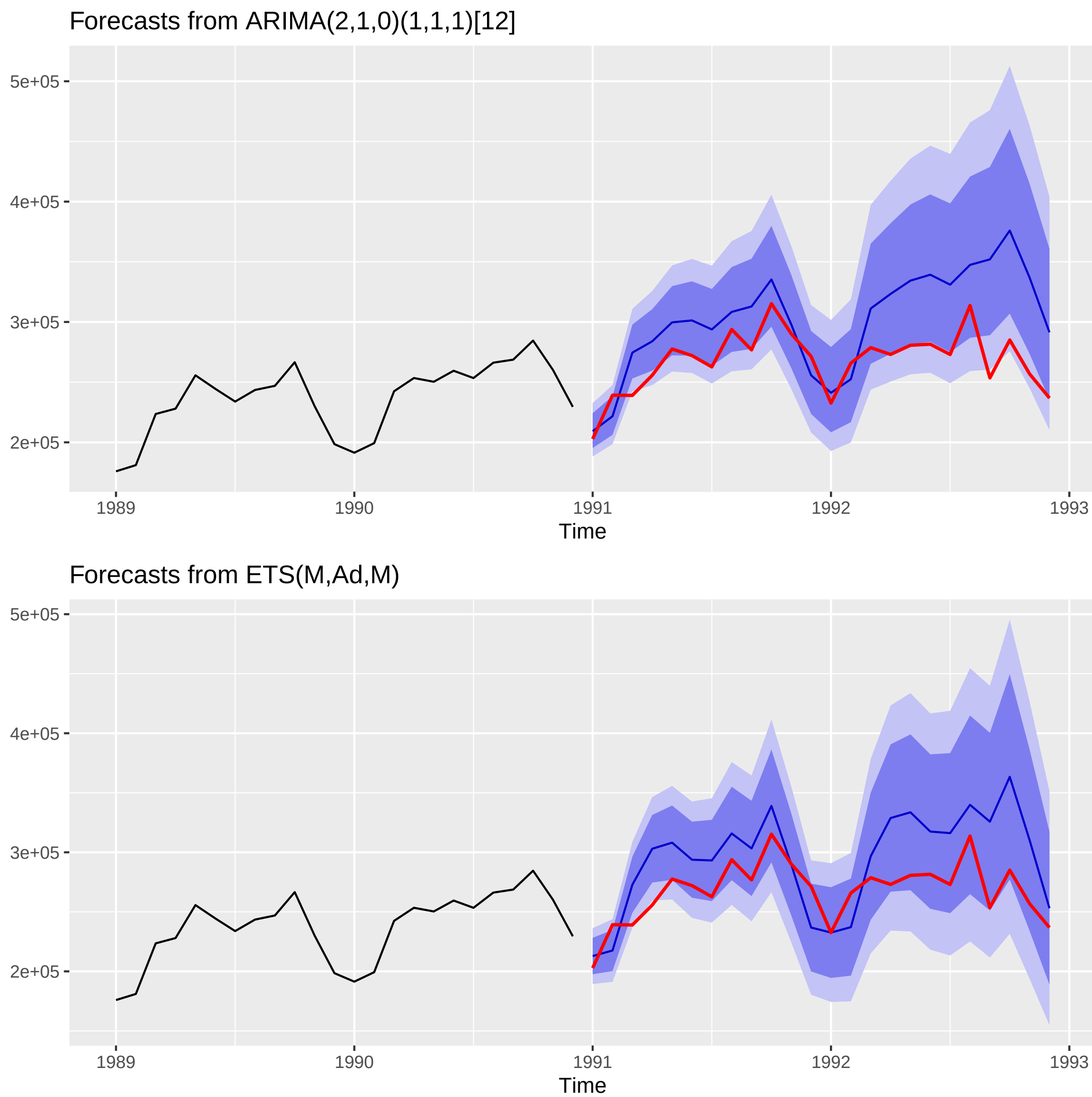


Figure 26:

1. ACF , ACF PACF .

2. ARIMA ETS . Test data 10 . 4
`https://raw.githubusercontent.com/yjyjpark/TS-with-R/main/Data/` .

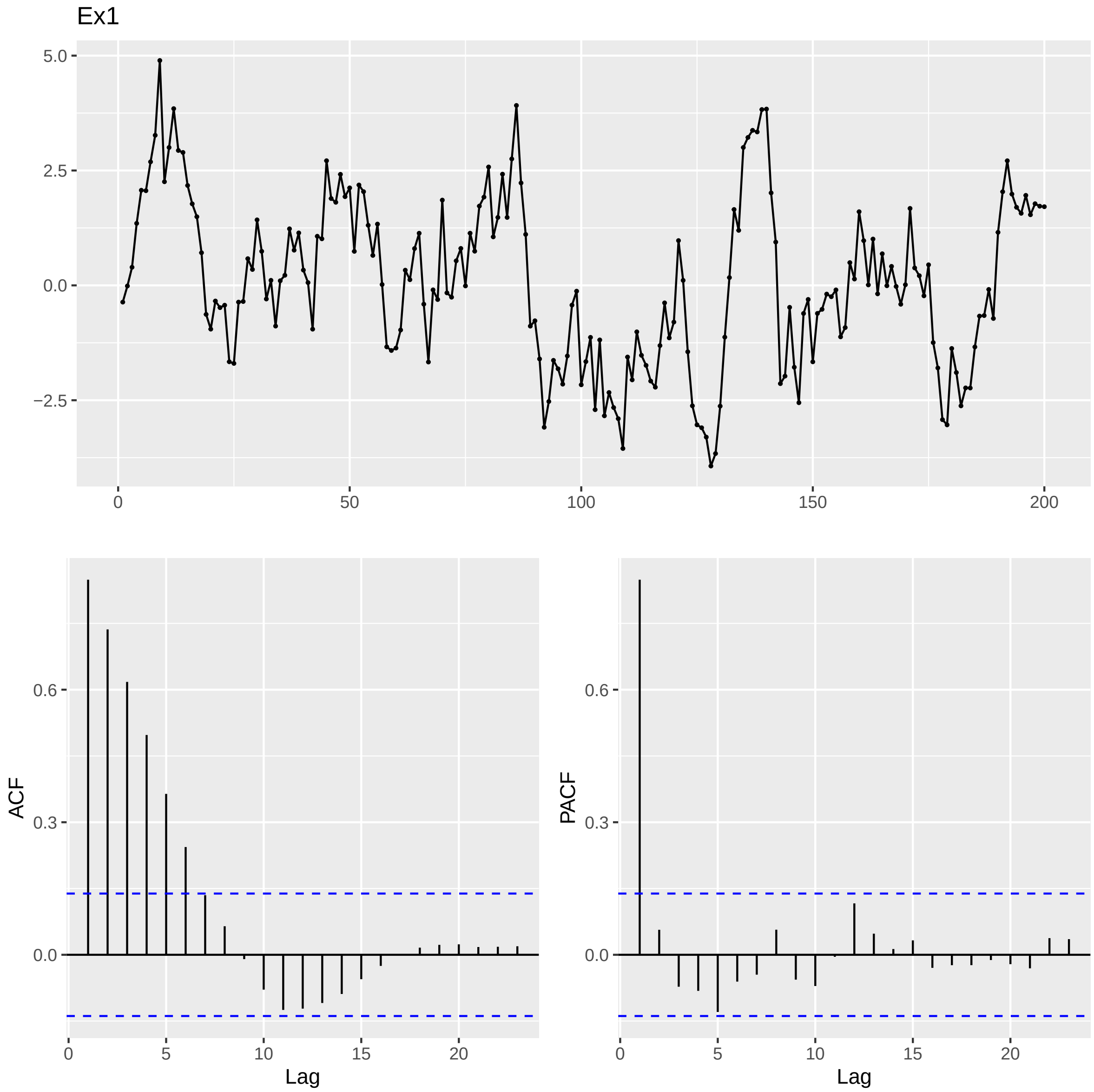


Figure 27: EX 1)

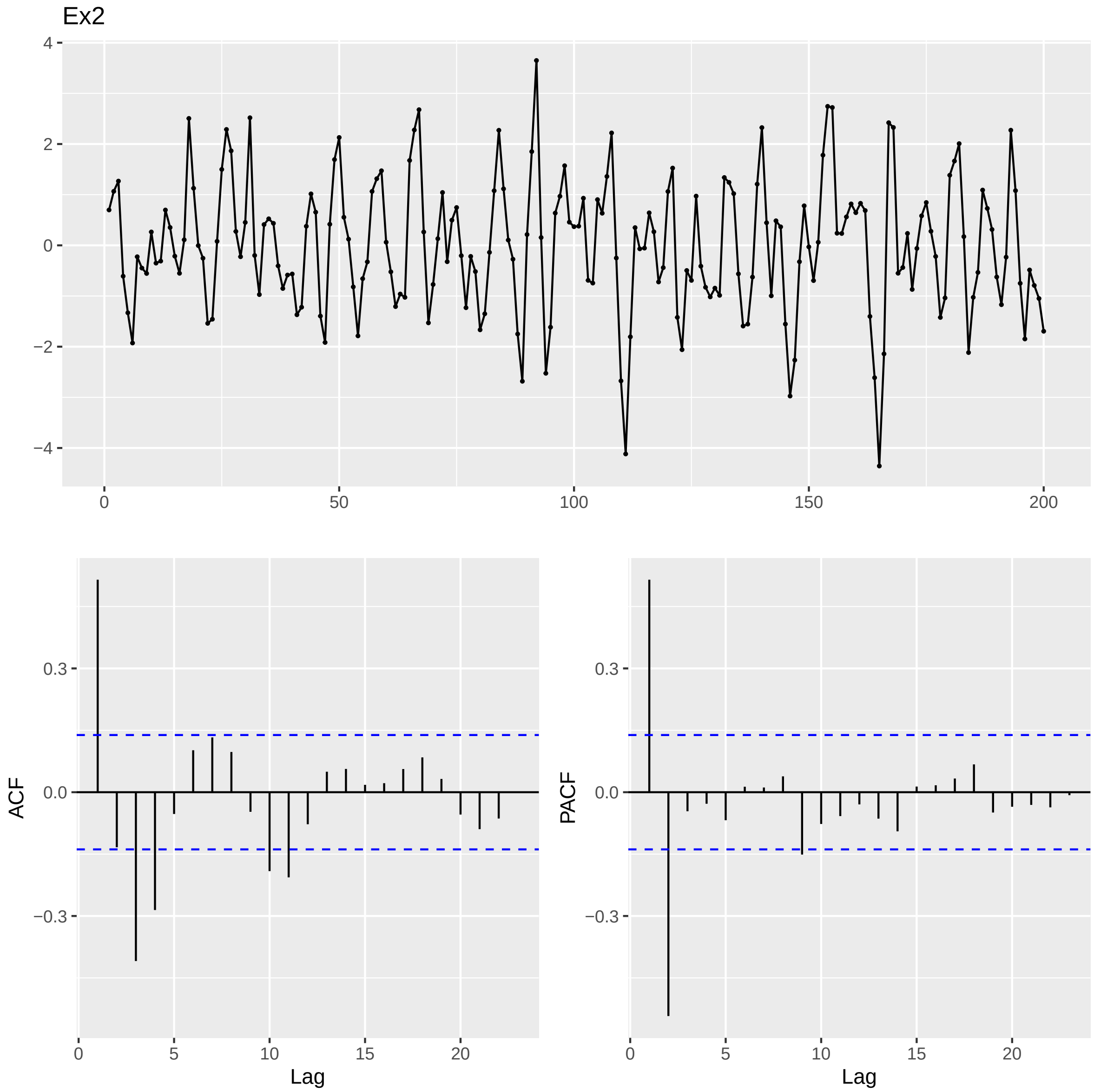


Figure 28: Ex 2)

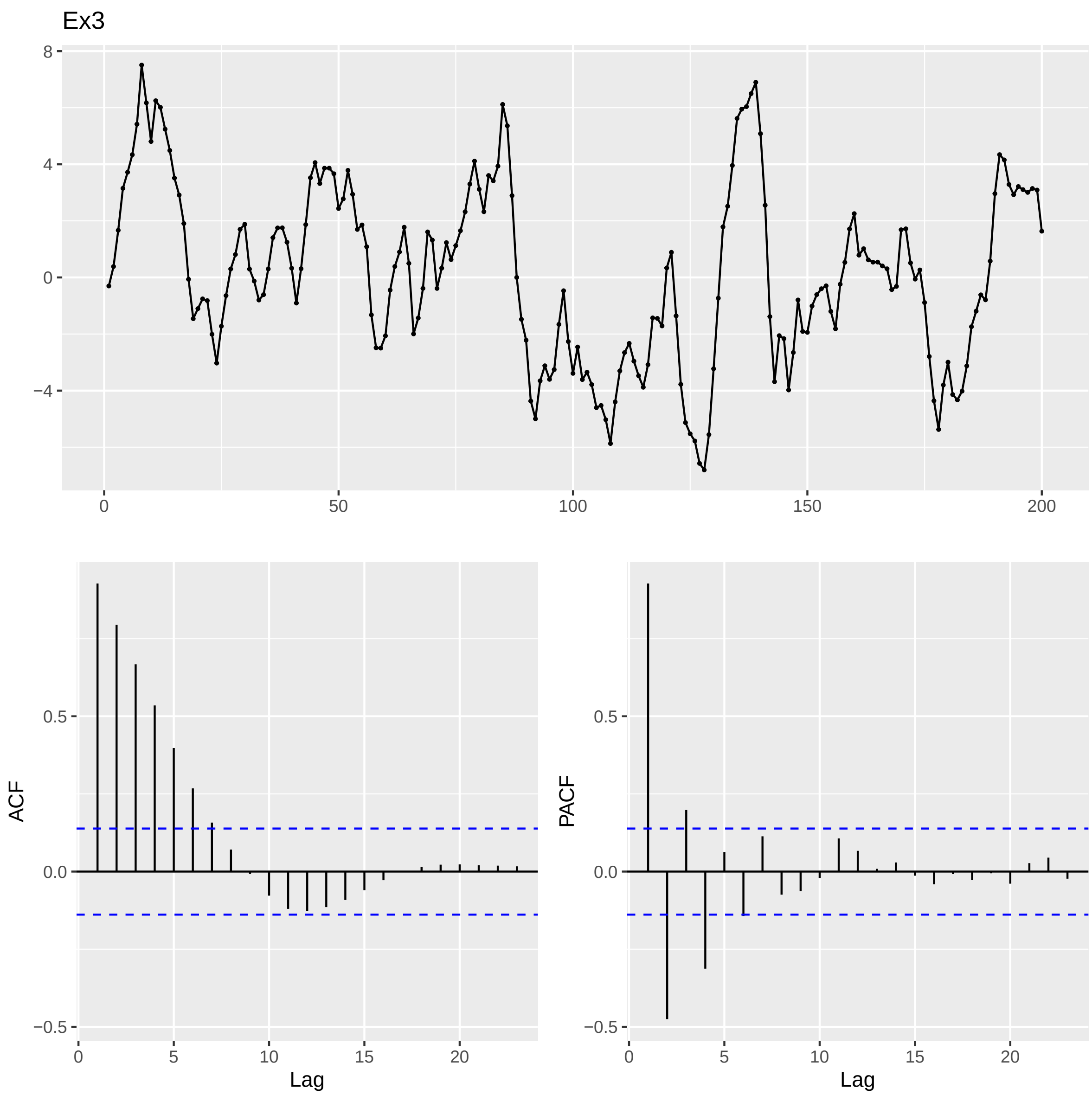


Figure 29: Ex 3)

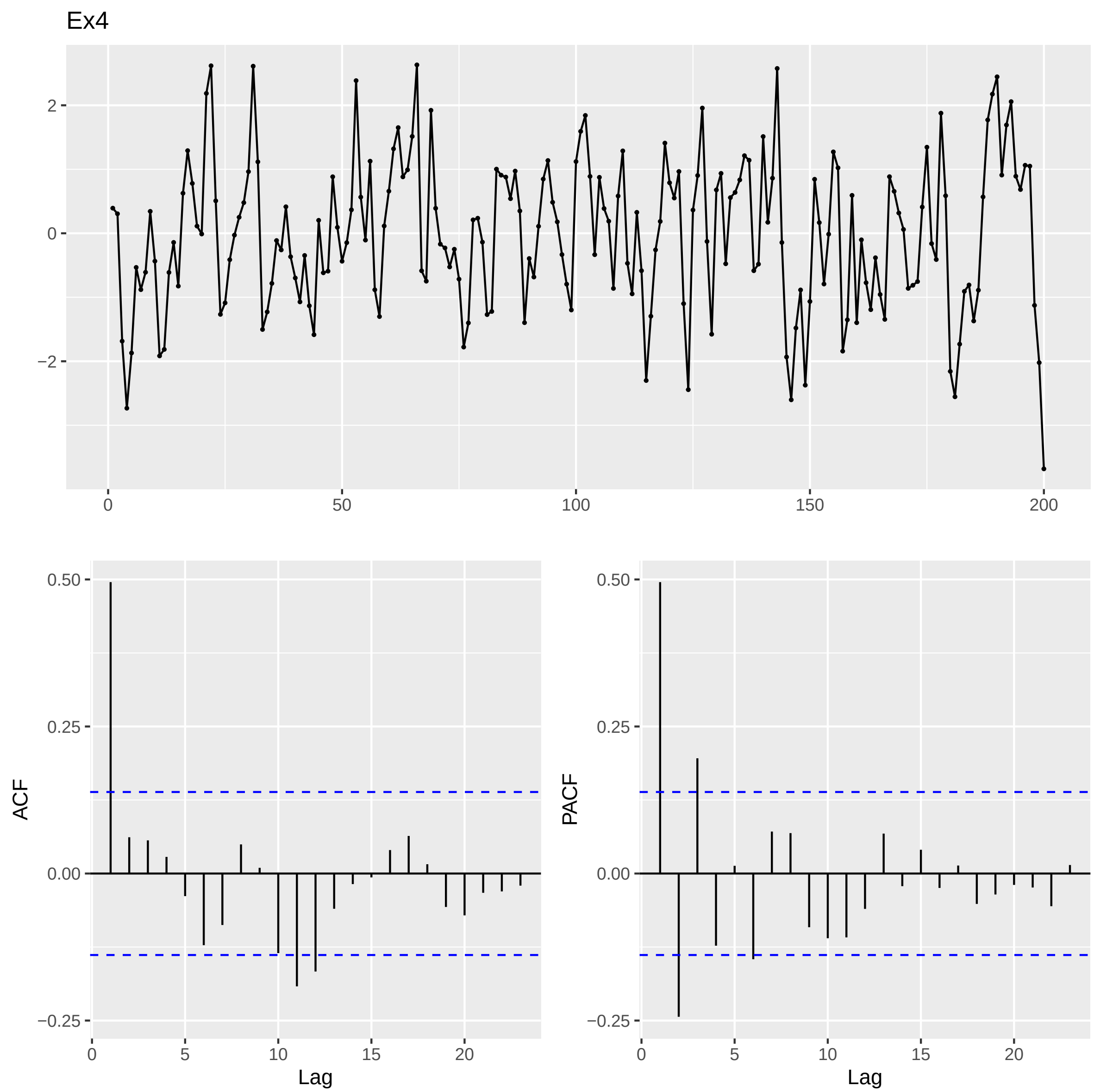


Figure 30: Ex 4)

- arima_ex2_1.txt
- arima_ex2_2.txt
- arima_ex2_3.txt
- arima_ex2_4.txt
- fpp2::ausair