ARMA

- : (fpp2::ausbeer)

1.

ausbeer 1956 1 2010 2 . 1975 1 . 2 test data .

```r
train_b <- window(ausbeer, start = 1975, end = c(2008, 2))
test_b <- window(ausbeer, start = c(2008, 3))
```

Figure 1 . Test data .

```r
autoplot(train_b) +
  autolayer(test_b, color = "red", size = 0.8) +
  labs(y = NULL, x = NULL)
```

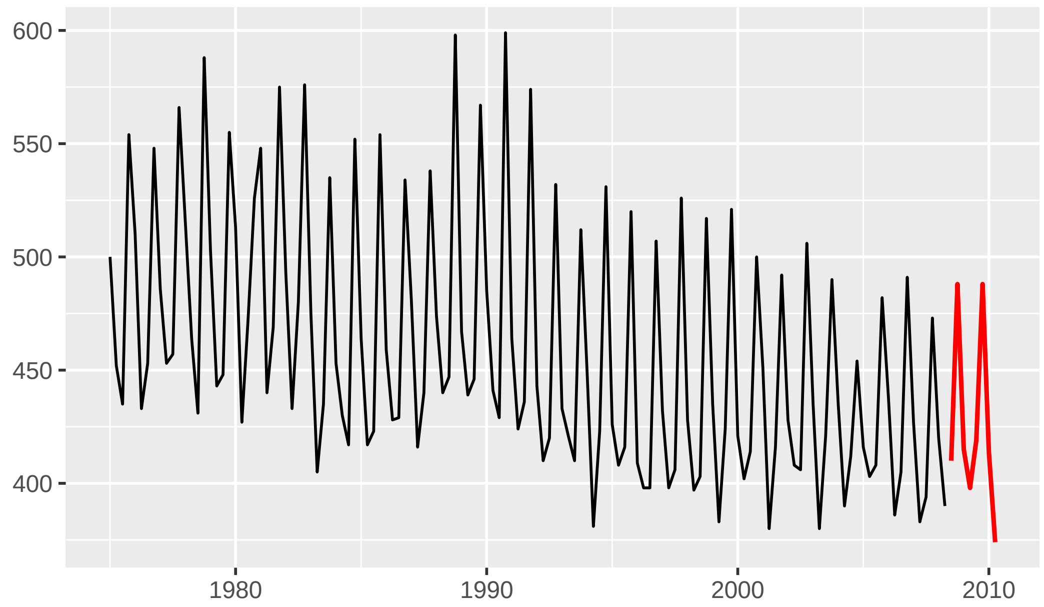


Figure 1: `ausbeer`

`tslm()` . `tslm()` `lm()` , `ts` . `ts` , .

t . `time()` . `train_b` `time()` . 1975 1 1975.00 , 1/4 .

```
time(train_b)[1:9]
## [1] 1975.00 1975.25 1975.50 1975.75 1976.00 1976.25 1976.50 1976.75 1977.00
```

$t = 1, 2, 3, \ldots$, `1:length(train_b)` , `tslm()` `trend` .

dummy `forecast::seasonaldummy()` , `tslm()` `season` . dummy .

```
seasonaldummy(train_b)[1:5,]
##      Q1 Q2 Q3
## [1,]  1  0  0
## [2,]  0  1  0
## [3,]  0  0  1
## [4,]  0  0  0
## [5,]  1  0  0
```

Fourier series `forecast::fourier()` . fourier series `K` .

```
fourier(train_b, K = 2)[1:5,]
##      S1-4 C1-4 C2-4
## [1,]    1    0   -1
## [2,]    0   -1    1
## [3,]   -1    0   -1
## [4,]    0    1    1
## [5,]    1    0   -1
```

S1-4 C1-4 $K = 1$ $\sin(2\pi t/4)$ $\cos(2\pi t/4)$, C2-4 $K = 2$ $\cos(2\pi 2t/4)$.

`time()` `seasonaldummy()` , .

```
fit1 <- tslm(train_b ~ time(train_b) + seasonaldummy(train_b))
```

```
summary(fit1)
##
## Call:
## tslm(formula = train_b ~ time(train_b) + seasonaldummy(train_b))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.776 -11.771  -0.738  10.842  63.468
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               5335.5613   325.0641   16.41   <2e-16 ***
## time(train_b)               -2.4112     0.1632  -14.78   <2e-16 ***
## seasonaldummy(train_b)Q1   -74.4299     4.4641  -16.67   <2e-16 ***
## seasonaldummy(train_b)Q2 -118.3271     4.4639  -26.51   <2e-16 ***
## seasonaldummy(train_b)Q3 -105.7240     4.4973  -23.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.27 on 129 degrees of freedom
## Multiple R-squared:  0.8911, Adjusted R-squared:  0.8877
## F-statistic: 263.9 on 4 and 129 DF,  p-value: < 2.2e-16
```

`trend` `season` , .

```
fit2 <- tslm(train_b ~ trend + season)
```

```
summary(fit2)
##
## Call:
## tslm(formula = train_b ~ trend + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.776 -11.771  -0.738  10.842  63.468
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 499.6812     4.1577 120.181  < 2e-16 ***
## trend        -0.6028     0.0408 -14.775  < 2e-16 ***
## season2     -43.8972     4.4306  -9.908  < 2e-16 ***
## season3     -31.2941     4.4639  -7.011 1.19e-10 ***
## season4      74.4299     4.4641  16.673  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.27 on 129 degrees of freedom
## Multiple R-squared:  0.8911, Adjusted R-squared:  0.8877
## F-statistic: 263.9 on 4 and 129 DF,  p-value: < 2.2e-16
```

fit1 fit2 . . , 4 fit1 1/4 fit2 1 .

, . , fit1 1975.00 , fit2 1 , seasonaldummy() , season , dummy .

.

```
tibble(fit1 = fit1$fitted, fit2 = fit2$fitted)
## # A tibble: 134 x 2
##     fit1  fit2
##    <dbl> <dbl>
##  1  499.  499.
##  2  455.  455.
##  3  467.  467.
##  4  572.  572.
##  5  497.  497.
##  6  452.  452.
##  7  464.  464.
## 8 569. 569.
## 9 494. 494.
## 10 450. 450.
## # ... with 124 more rows
```

time() fourier() Fourier series , . K=2 .

```
fit3 <- tslm(train_b ~ time(train_b) + fourier(train_b, K=2))
```

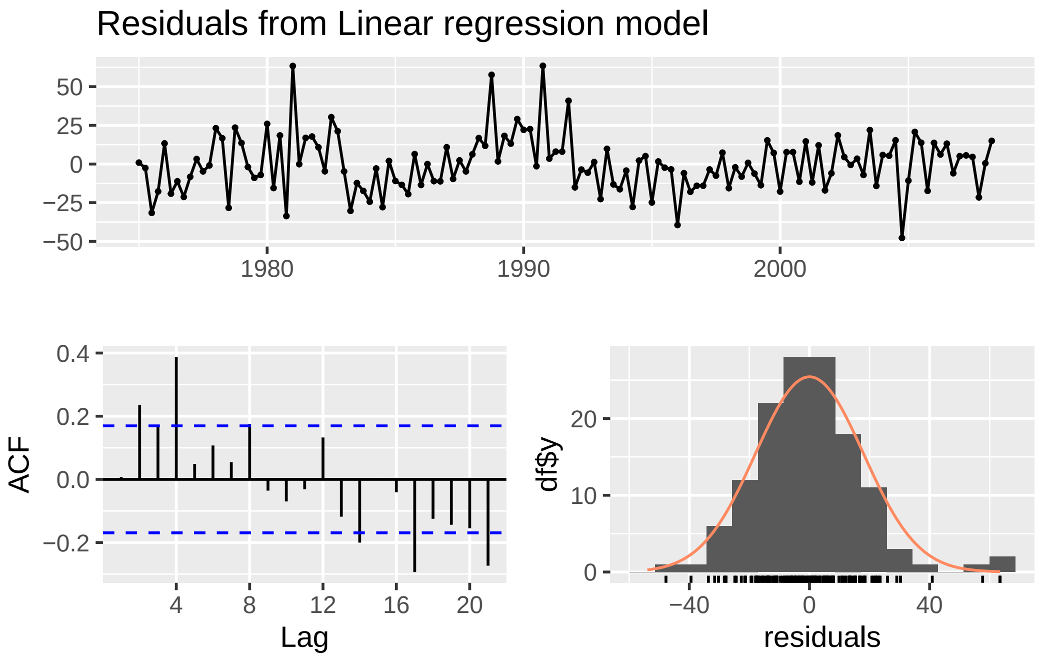
```
summary(fit3)
##
## Call:
## tslm(formula = train_b ~ time(train_b) + fourier(train_b, K = 2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.776 -11.771  -0.738  10.842  63.468
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  5260.9410   325.0320  16.186  < 2e-16 ***
## time(train_b)                  -2.4112     0.1632 -14.775  < 2e-16 ***
## fourier(train_b, K = 2)S1-4    15.6471     2.2319   7.011 1.19e-10 ***
## fourier(train_b, K = 2)C1-4    59.1635     2.2319  26.508  < 2e-16 ***
## fourier(train_b, K = 2)C2-4    15.4567     1.5784   9.793  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.27 on 129 degrees of freedom
## Multiple R-squared:  0.8911, Adjusted R-squared:  0.8877
## F-statistic: 263.9 on 4 and 129 DF,  p-value: < 2.2e-16
```

Fourier series dummy . fit1 fit3 .

```
tibble(fit1 = fit1$fitted, fit3 = fit3$fitted)
## # A tibble: 134 x 2
##     fit1  fit3
##    <dbl> <dbl>
##  1  499.  499.
##  2  455.  455.
##  3  467.  467.
##  4  572.  572.
##  5  497.  497.
##  6  452.  452.
##  7  464.  464.
##  8  569.  569.
##  9  494.  494.
## 10  450.  450.
## # ... with 124 more rows
```

, . checkresiduals() lm() tslm() Breusch-Godfrey . , ACF . . , .

```
checkresiduals(fit1)
```



```
##
##  Breusch-Godfrey test for serial correlation of order up to 8
##
## data:  Residuals from Linear regression model
## LM test = 30.158, df = 8, p-value = 0.0001982
```

2. ARMA

`ausbeer` . ARMA ARMA .

```r
train_b <- window(ausbeer, start = 1975, end = c(2008, 2))
test_b <- window(ausbeer, start = c(2008, 3))
```

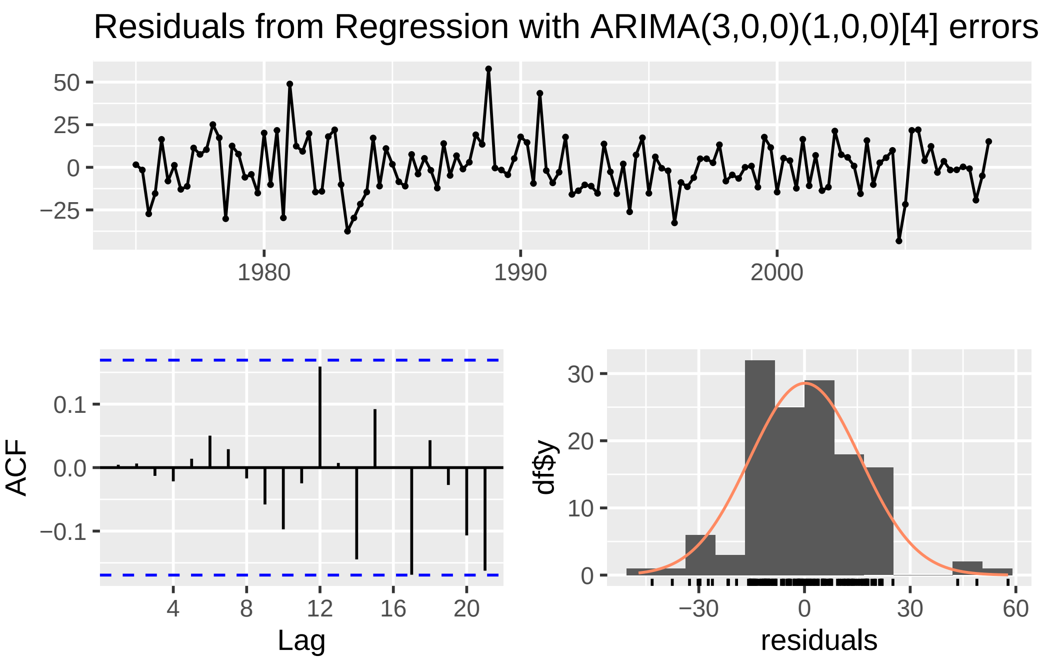
```r
fit4 <- auto.arima(train_b,
                   xreg = cbind(Time = time(train_b),
                                Qtr = seasonaldummy(train_b)),
                   stepwise = FALSE)
```

```r
summary(fit4)
## Series: train_b
## Regression with ARIMA(3,0,0)(1,0,0)[4] errors
##
## Coefficients:
##           ar1     ar2     ar3    sar1  intercept     Time     Qtr.Q1     Qtr.Q2
##       -0.1233  0.1645  0.2440  0.4155  5227.3160  -2.3573  -73.4170  -117.2944
## s.e.   0.0852  0.0831  0.0849  0.0817   611.6432   0.3071    6.0565     5.2917
##         Qtr.Q3
##       -105.5713
## s.e.    6.0999
##
## sigma^2 = 264.2:  log likelihood = -559.59
## AIC=1139.19   AICc=1140.97   BIC=1168.16
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 0.1677325 15.69822 12.11679 -0.07293605 2.618274 0.7416116
##                     ACF1
## Training set 0.004529308
```

ARIMA(3,0,0)$(1,0,0)_4$. , .

ARMA `fit4` . .

```r
checkresiduals(fit4)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0)(1,0,0)[4] errors
## Q* = 0.64935, df = 4, p-value = 0.9574
##
## Model df: 4.   Total lags used: 8
```

ARMA forecast() . forecast(object, xreg, ...) , object Arima() auto.arima() , xreg .

Test data test_b .

```
fc4 <- forecast(fit4,
                xreg = cbind(Time = time(test_b),
                             Qtr = seasonaldummy(test_b))
                )
```

ARMA . tslm() trend season fit2 .

```
fit2 <- tslm(train_b ~ trend + season)
fc2 <- forecast(fit2, h = length(test_b))
```

fc2$mean fc4$mean . . .

```
tibble(fc2 = fc2$mean, fc4 = fc4$mean)
## # A tibble: 8 x 2
##     fc2   fc4
##   <dbl> <dbl>
## 1  387.  383.
## 2  492.  485.
## 3  417.  420.
## 4  373.  378.
## 5  385.  384.
## 6  490.  487.
## 7  415.  416.
## 8  370.  373.
```

95% . fc2$upper fc4$upper , fc2$lower fc4$lower . 80% 95% , 95% . ARMA fc4 .

```
tibble(fc2 = fc2$upper[,2] - fc2$lower[,2],
       fc4 = fc4$upper[,2] - fc4$lower[,2])
## # A tibble: 8 x 2
##     fc2   fc4
##   <dbl> <dbl>
## 1  74.2  63.7
## 2  74.2  64.2
## 3  74.2  65.2
## 4  74.2  66.5
## 5  74.3  71.0
## 6  74.3  71.0
## 7  74.3  71.3
## 8  74.3  71.5
```

. .

```
accuracy(fc2, test_b)
##                        ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 8.491860e-16 17.92314 13.57971 -0.1317235 2.924979 0.8311500
## Test set     9.745836e+00 17.30833 11.90530  2.4185316 2.886254 0.7286673
##                     ACF1 Theil's U
## Training set 0.007149994        NA
## Test set     0.048860146 0.3056324
```

```
accuracy(fc4, test_b)
##                      ME     RMSE      MAE         MPE     MAPE      MASE
## Training set  0.1677325 15.69822 12.11679 -0.07293605 2.618274 0.7416116
## Test set     10.0161277 17.40591 11.67960  2.42511984 2.826267 0.7148528
##                     ACF1 Theil's U
## Training set 0.004529308        NA
## Test set     0.049644862 0.2895782
```

.

```
library(patchwork)
```

```
p1 <- autoplot(fc2, include=8) +
  autolayer(test_b, color = "red", size=.8) +
  labs(y = NULL, x = NULL)

p2 <- autoplot(fc4, include=8) +
  autolayer(test_b, color = "red", size=.8) +
  labs(y = NULL, x = NULL)
p1 + p2
```

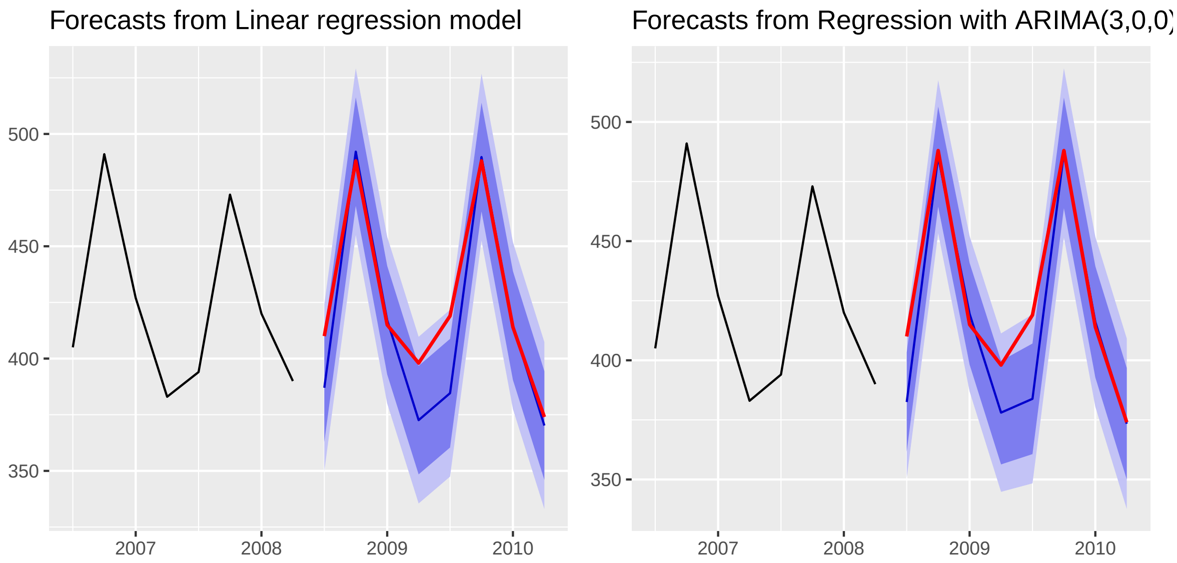


Figure 2: `ausbeer`

- : 1970 1 2005 12 (`global.txt`)

global.txt 1856 1 2005 12 , 1970 ARMA ARIMA , ETS .

training data test data .

```
global <- scan("https://raw.githubusercontent.com/yjyjpark/TS-with-R/main/Data/global.txt"
global.ts <- ts(global, start = c(1856, 1), frequency = 12)
train_g <- window(global.ts, start = 1970, end = c(2003,12))
test_g <- window(global.ts, start = 2004)
```

Figure 3 1970 . Test data .

```
autoplot(train_g) +
  autolayer(test_g, color = "red", size = .8) +
  labs(x = NULL, y = NULL)
```

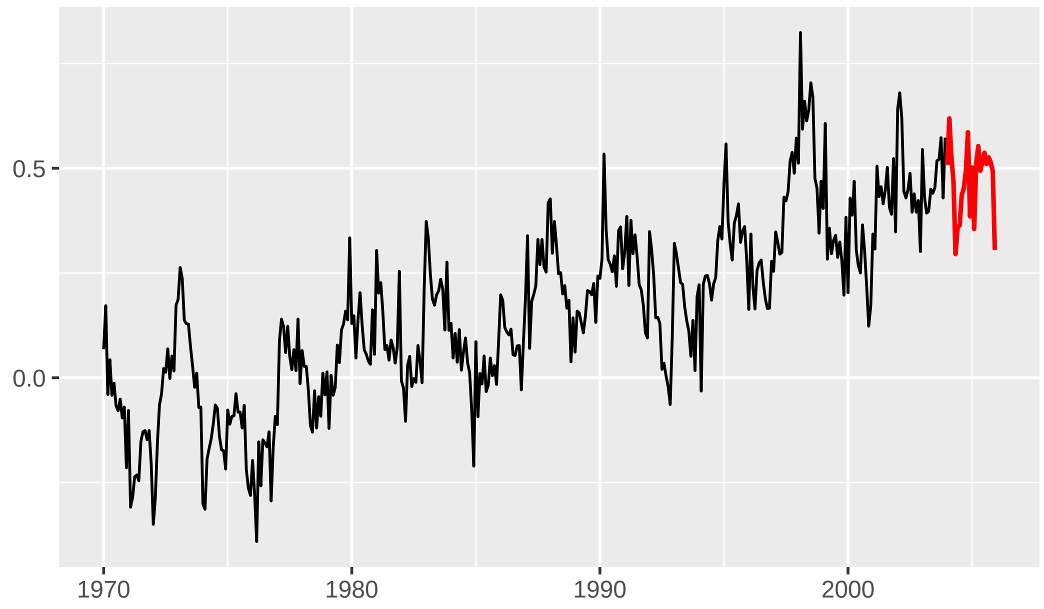


Figure 3: global.txt training data test data

1. ARMA

dummy , Fourier series . dummy . time() , seasonaldummy() .

```r
Time <- time(train_g)
Month <- seasonaldummy(train_g)
```

fit1 stepwise = FASLE approximation = FALSE , .

```r
fit1 <- auto.arima(train_g, xreg = cbind(Time, Month))
```

```r
summary(fit1)
## Series: train_g
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##          ar1     ar2  intercept    Time  Month.Jan  Month.Feb  Month.Mar
##       0.4932  0.3204   -34.5510  0.0175     0.0409     0.0522     0.0268
## s.e.  0.0469  0.0472     4.2613  0.0021     0.0161     0.0171     0.0195
##       Month.Apr  Month.May  Month.Jun  Month.Jul  Month.Aug  Month.Sep
##          0.0250     0.0093     0.0162     0.0137     0.0151     0.0002
## s.e.     0.0204     0.0211     0.0213     0.0211     0.0204     0.0194
##       Month.Oct  Month.Nov
##          -0.015    -0.0304
## s.e.      0.017     0.0160
##
## sigma^2 = 0.007112:  log likelihood = 437.21
## AIC=-842.41   AICc=-841.02   BIC=-778.23
##
## Training set error measures:
##                         ME       RMSE        MAE      MPE     MAPE      MASE
## Training set -0.0007778394 0.08276884 0.06379682 22.62755 81.57672 0.4285951
##                      ACF1
## Training set 0.006984407
```

AR(2) .

Fourier series . Fourier series .

```r
Time <- time(train_g)
res <- vector("numeric", 6)
for(i in seq(res)){
  xreg <- cbind(Time, fourier(train_g, K = i))
  fit <- auto.arima(train_g, xreg = xreg)
  res[i] <- fit$aicc
}
```

res 6 AICc . AICc .

```
res
## [1] -840.8765 -851.6310 -850.3429 -846.7538 -842.5771 -841.0206
```

```
(k_min <- which.min(res))
## [1] 2
```

$K = 2$ Fourier series .

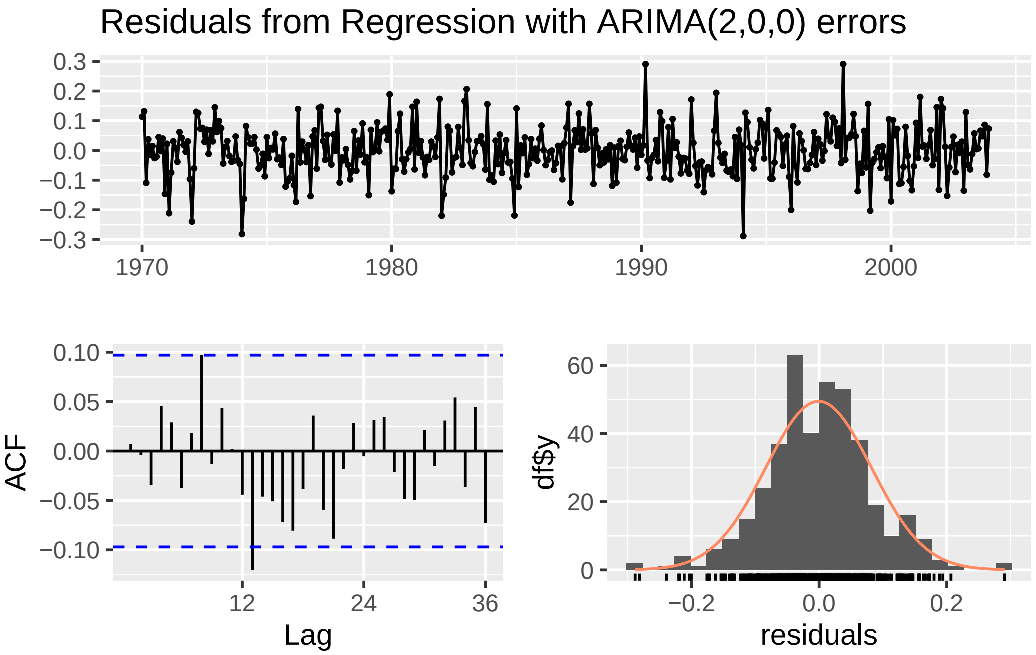
```
Fourier <- fourier(train_g, K = k_min)
fit2 <- auto.arima(train_g, xreg = cbind(Time, Fourier))
```

```
summary(fit2)
## Series: train_g
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##          ar1     ar2  intercept    Time  Fourier.S1-12  Fourier.C1-12
##       0.4919  0.3200   -34.4729  0.0174         0.0210        -0.0047
## s.e.  0.0469  0.0472     4.2484  0.0021         0.0088         0.0087
##       Fourier.S2-12  Fourier.C2-12
##              0.0192        -0.0054
## s.e.         0.0051         0.0051
##
## sigma^2 = 0.007062:  log likelihood = 435.04
## AIC=-852.08   AICc=-851.63   BIC=-815.98
##
## Training set error measures:
##                         ME       RMSE       MAE      MPE     MAPE      MASE
## Training set -0.0007984029 0.08320983 0.0640202 22.23643 80.56552 0.4300958
##                     ACF1
## Training set 0.004973903
```

fit1 fit2 AR(2) .

. dummy fit1 . fit1 .

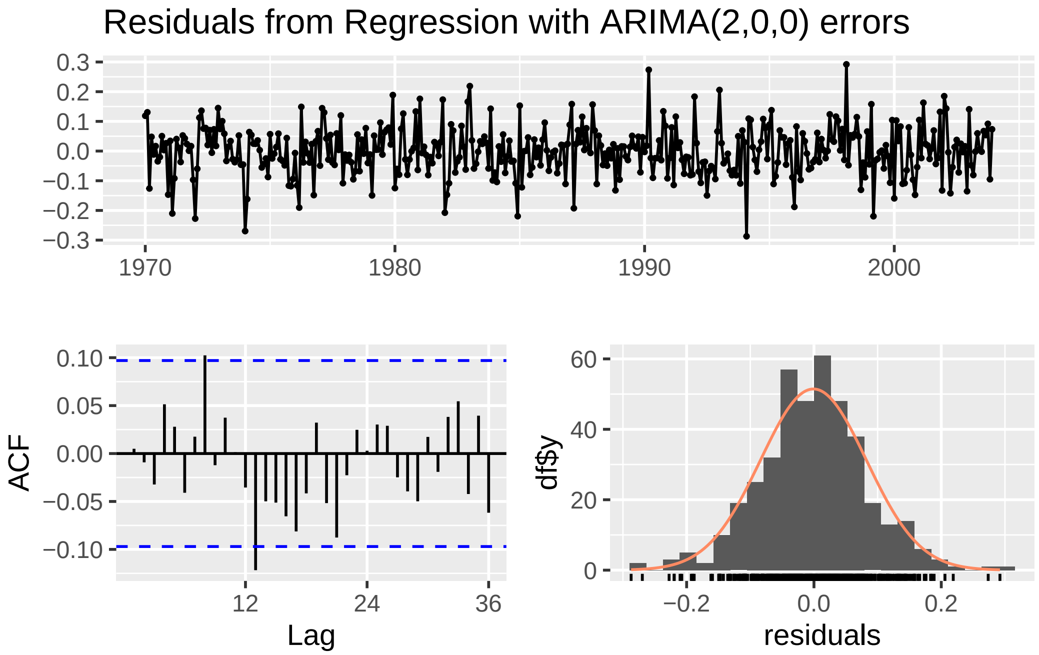
```
checkresiduals(fit1)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 27.817, df = 22, p-value = 0.1818
##
## Model df: 2.   Total lags used: 24
```

Fourier series fit2 . fit2 .

```
checkresiduals(fit2)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0) errors
## Q* = 27.53, df = 22, p-value = 0.1918
##
## Model df: 2.   Total lags used: 24
```

. AICc .

```
c(fit1$aicc, fit2$aicc)
## [1] -841.0206 -851.6310
```

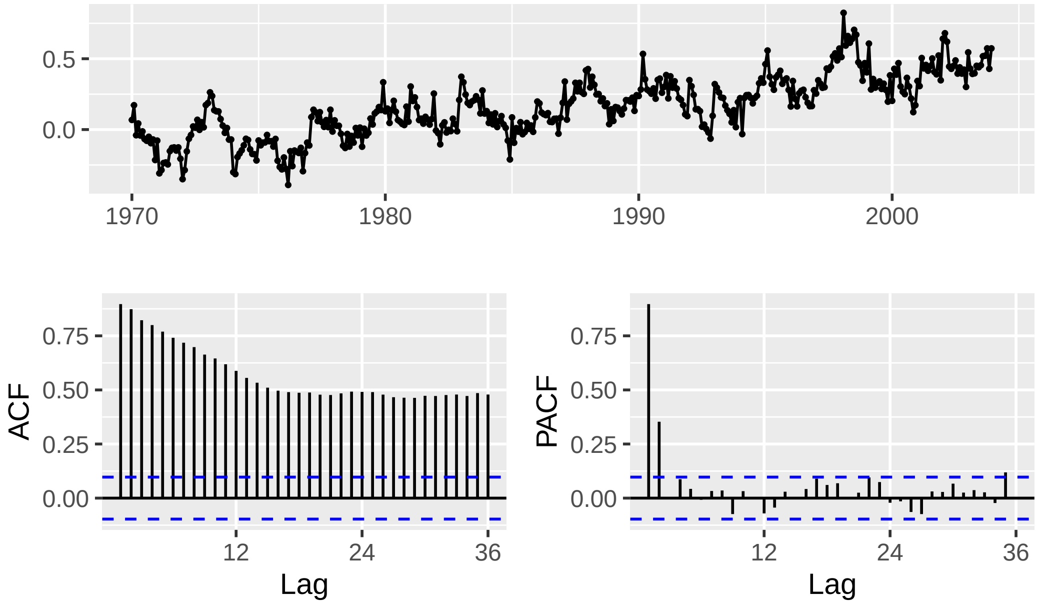
fit2 AICc , .

```
fit_reg <- fit2
```

2. ARIMA

.

```
ggtsdisplay(train_g)
```

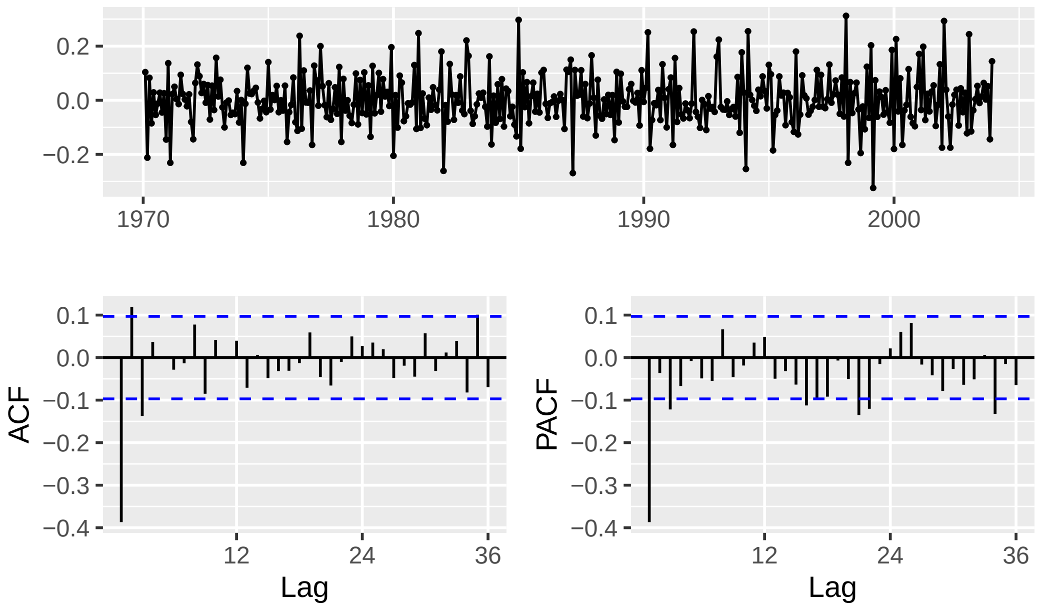


, ACF r_1 1 . .
.

```
ndiffs(train_g)
## [1] 1
nsdiffs(train_g)
## [1] 0
```

1 , . 1 ACF, PACF . .

```
train_g %>%
  diff() %>%
  ggtsdisplay()
```



ACF PACF . 1 6 , ACF 3 , PACF 1 3 . ACF , PACF 3 , ARMA AR . 12, 24, 36 ACF PACF , , AR(1)$_{12}$ MA(1)$_{12}$.

`auto.arima()` AICc .

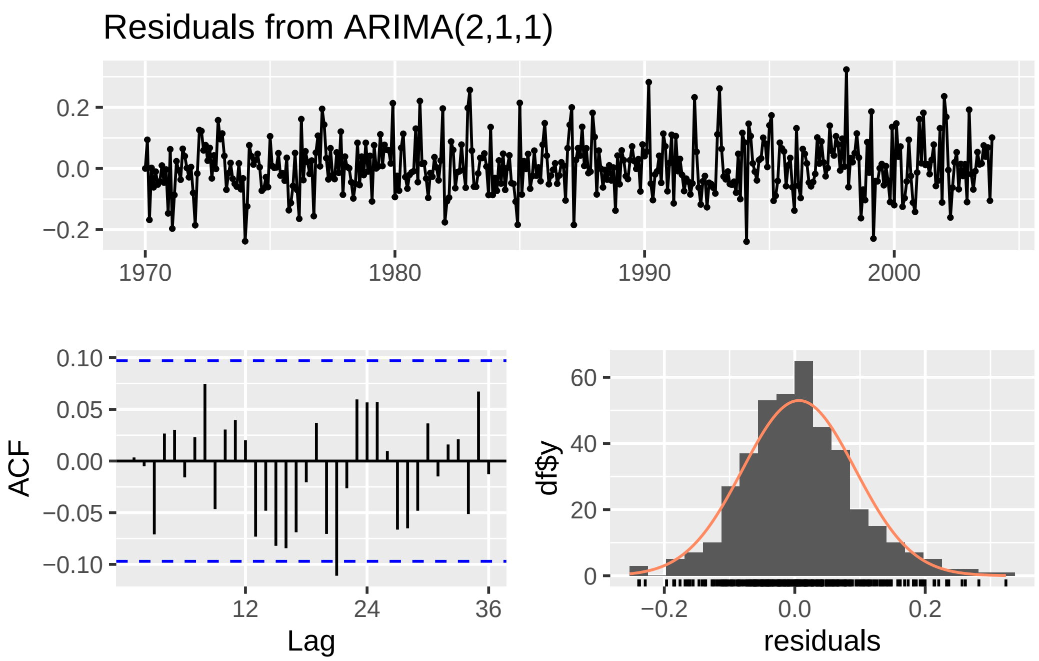
```
fit_arima <- auto.arima(train_g,
                        stepwise = FALSE,
                        approximation = FALSE)
```

ARIMA(2,1,1) .

```
fit_arima
## Series: train_g
## ARIMA(2,1,1)
##
## Coefficients:
##          ar1     ar2      ma1
##       0.5198  0.3008  -0.9703
## s.e.  0.0498  0.0492   0.0132
##
## sigma^2 = 0.007576:  log likelihood = 417.27
## AIC=-826.55   AICc=-826.45   BIC=-810.51
```

fit_arima . .

```
checkresiduals(fit_arima)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,1)
## Q* = 30.228, df = 21, p-value = 0.08751
##
## Model df: 3.   Total lags used: 24
```

3. ETS

ets() AICc .

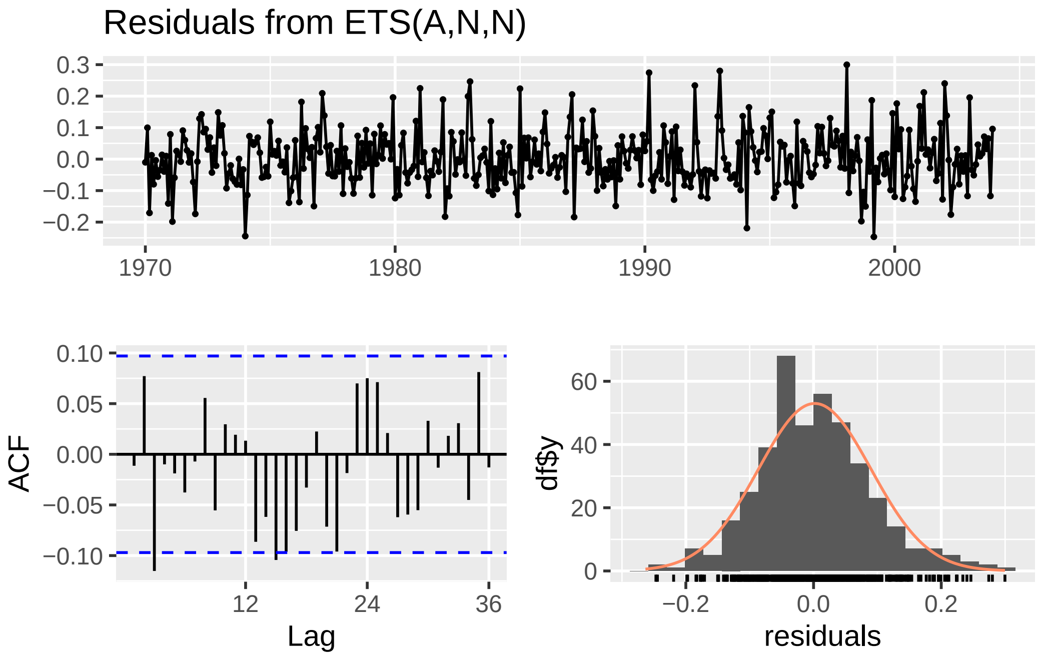
```
fit_ets <- ets(train_g)
```

ETS(A,N,N) .

```
fit_ets
## ETS(A,N,N)
##
## Call:
##  ets(y = train_g)
##
##   Smoothing parameters:
##     alpha = 0.5868
##
##   Initial states:
##     l = 0.0782
##
##   sigma:  0.0885
##
##       AIC     AICc      BIC
## 478.0785 478.1379 490.1123
```

fit_ets .

```
checkresiduals(fit_ets)
```



```
##
```

```
##  Ljung-Box test
##
## data:  Residuals from ETS(A,N,N)
## Q* = 39.376, df = 22, p-value = 0.01277
##
## Model df: 2.   Total lags used: 24
```

. , (point forecast) .
, .

fit_reg, fit_arima, fit_ets , .

```
new_reg <- cbind(Time = time(test_g),
                 Fourier = fourier(test_g, K = k_min))
fc_reg <- forecast(fit_reg, xreg = new_reg)
fc_arima <- forecast(fit_arima, h = length(test_g))
fc_ets <- forecast(fit_ets, h = length(test_g))
```

.

```
accuracy(fc_reg, test_g)
##                         ME       RMSE        MAE       MPE     MAPE      MASE
## Training set -0.0007984029 0.08320983 0.06402020 22.236428 80.56552 0.4300958
## Test set     -0.0266870191 0.09047869 0.06864433 -9.691571 17.39724 0.4611613
##                     ACF1 Theil's U
## Training set 0.004973903        NA
## Test set     0.300995351  1.015694
```

```
accuracy(fc_arima, test_g)
##                       ME       RMSE        MAE       MPE     MAPE      MASE
## Training set 0.006369769 0.08661271 0.06601014 24.330149 79.34066 0.4434645
## Test set     0.004650257 0.08557693 0.07319499 -2.829173 17.33066 0.4917332
##                     ACF1 Theil's U
## Training set 0.003457635        NA
## Test set     0.335253668 0.9396727
```

```
accuracy(fc_ets, test_g)
##                        ME       RMSE        MAE       MPE     MAPE      MASE
## Training set  0.001901676 0.08829198 0.06862716  20.00303 84.32056 0.4610460
## Test set     -0.062218811 0.10436651 0.07564587 -17.55797 19.80559 0.5081986
##                     ACF1 Theil's U
## Training set -0.01131321         NA
## Test set       0.26877094  1.191604
```

Test data , . MASE ARMA ARIMA , RMSE MAPE ARIMA . test data Figure 4 . Y .

```
y_lim <- c(-.06, 1.06)
p1 <- autoplot(fc_reg, include = 12) +
  autolayer(test_g, color = "red", size = .8) +
  labs(x = NULL, y = NULL) + ylim(y_lim[1], y_lim[2])
p2 <- autoplot(fc_arima, include = 12) +
  autolayer(test_g, color = "red", size = .8) +
  labs(x = NULL, y = NULL) + ylim(y_lim[1], y_lim[2])
p3 <- autoplot(fc_ets, include=12) +
  autolayer(test_g, color = "red", size = .8) +
  labs(x = NULL, y = NULL) + ylim(y_lim[1], y_lim[2])

p1 / p2 / p3
```

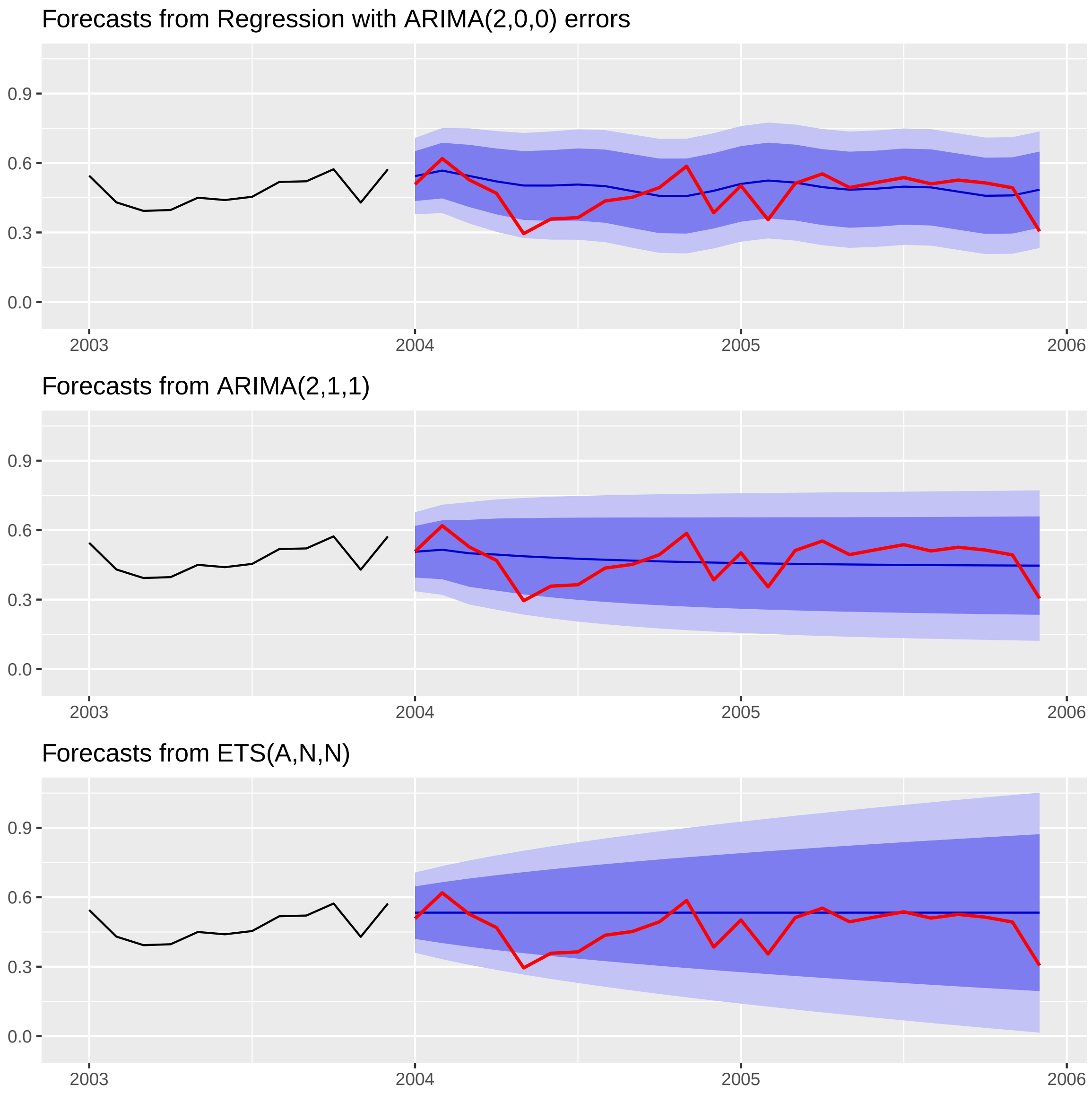


Figure 4: `global.txt`

- : 1949 1960 (`AirPassengers`)

`AirPassengers` 1949 1 1960 12 . ETS ARIMA , ARMA , . 2 test data .

```
train_AP <- window(AirPassengers, end = c(1958, 12))
test_AP <- window(AirPassengers, start = c(1959, 1))
```

Figure 5 . Test data .

```
autoplot(train_AP) +
  autolayer(test_AP, color = "red", size = .8) +
  labs(x = NULL, y = NULL)
```

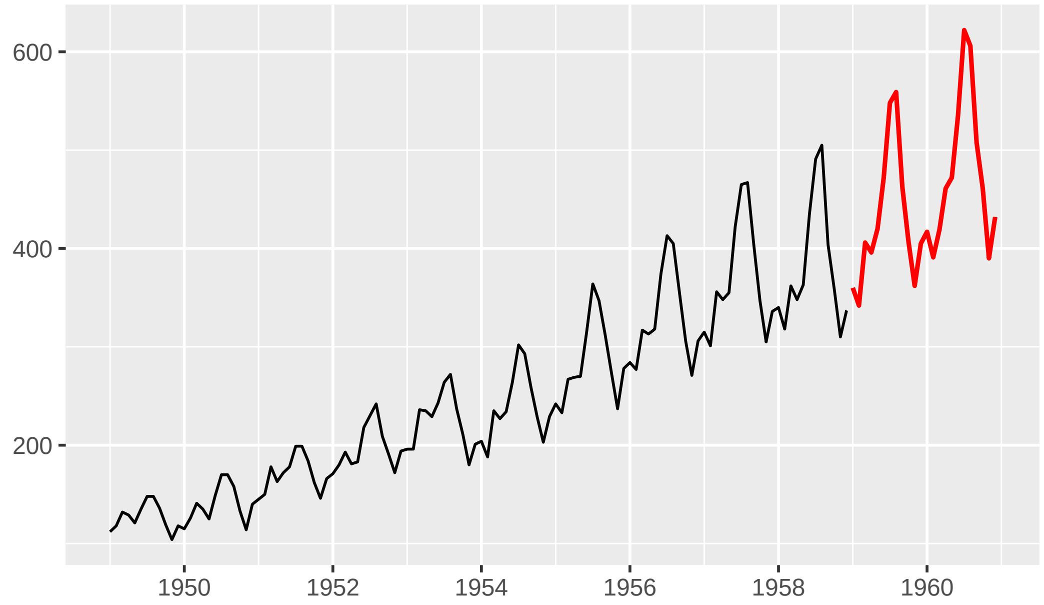


Figure 5: AirPassengers

, . . . Box-Cox .

```
(lam <- BoxCox.lambda(train_AP))
## [1] -0.3096628
```

λ = -0.3096628 . . Figure 6 .

.

```
p1 <- BoxCox(train_AP, lambda = lam) %>%
  autoplot() +
  labs(title = paste("Box-Cox", "lambda = ", signif(lam, 3)), x = NULL)
p2 <- train_AP %>%
  log() %>%
  autoplot() + labs(title = "Log", x = NULL)
```

```
p1+p2
```

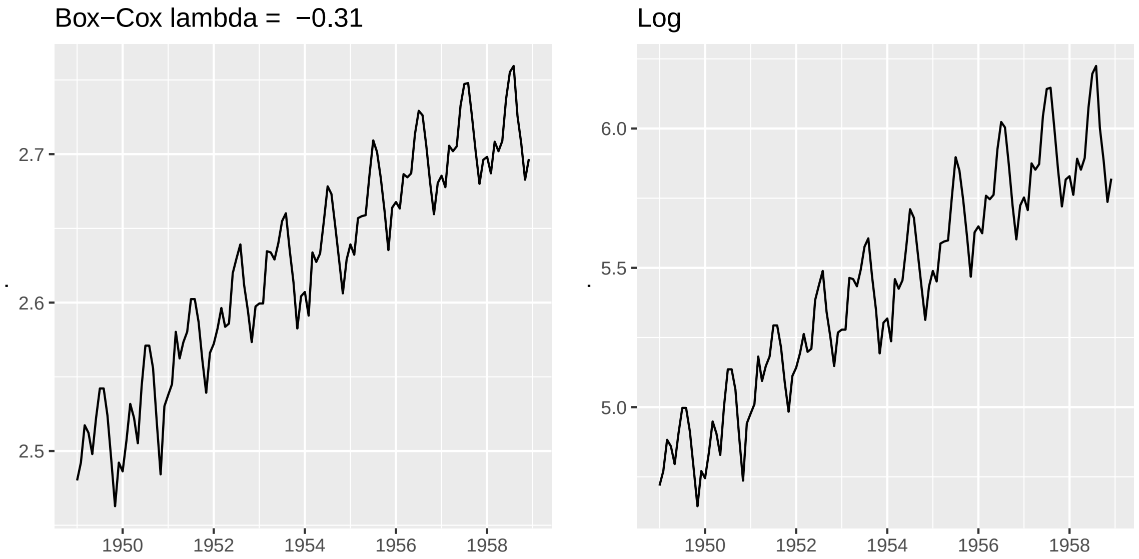


Figure 6:

1. ETS

ETS , . , . ETS Box-Cox ETS , ETS .

```
ets_1 <- ets(train_AP, lambda = lam)
ets_fc1 <- forecast(ets_1, h = length(test_AP))

ets_2 <- ets(train_AP, lambda = 0)
ets_fc2 <- forecast(ets_2, h = length(test_AP))

ets_3 <- ets(train_AP)
ets_fc3 <- forecast(ets_3, h = length(test_AP))
```

Figure 7 ETS test data . Box-Cox ETS test data . Test data , .

```
p1 <- autoplot(ets_fc1, include = 0) +
  autolayer(test_AP, color = "red", size = .8) + labs(y = NULL, subtitle = "ets_fc1")
p2 <- autoplot(ets_fc2, include = 0) +
  autolayer(test_AP, color = "red", size = .8) + labs(y = NULL, subtitle = "ets_fc2")
p3 <- autoplot(ets_fc3, include = 0) +
  autolayer(test_AP, color = "red", size = .8) + labs(y = NULL, subtitle = "ets_fc3")
p1/p2/p3
```

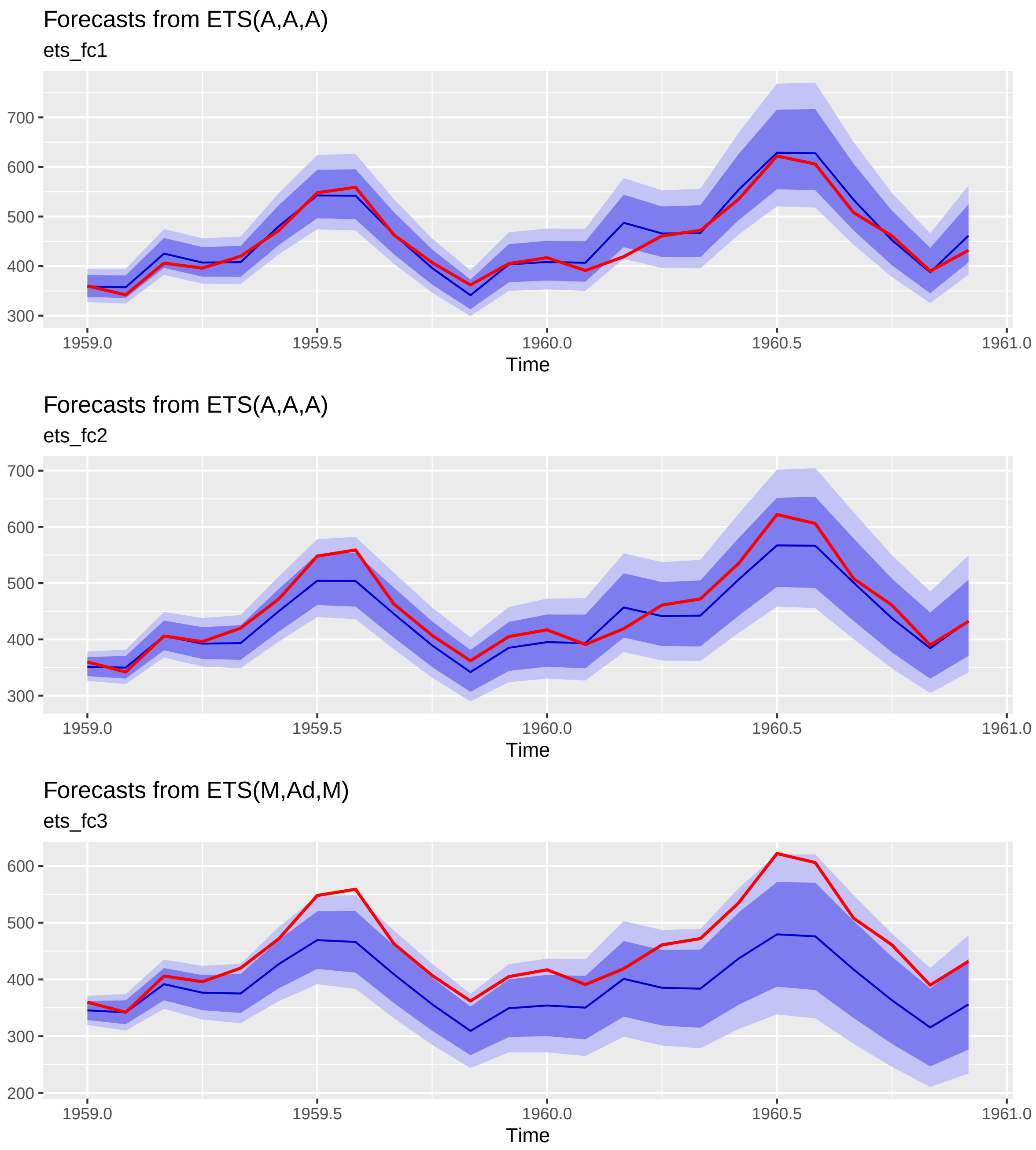
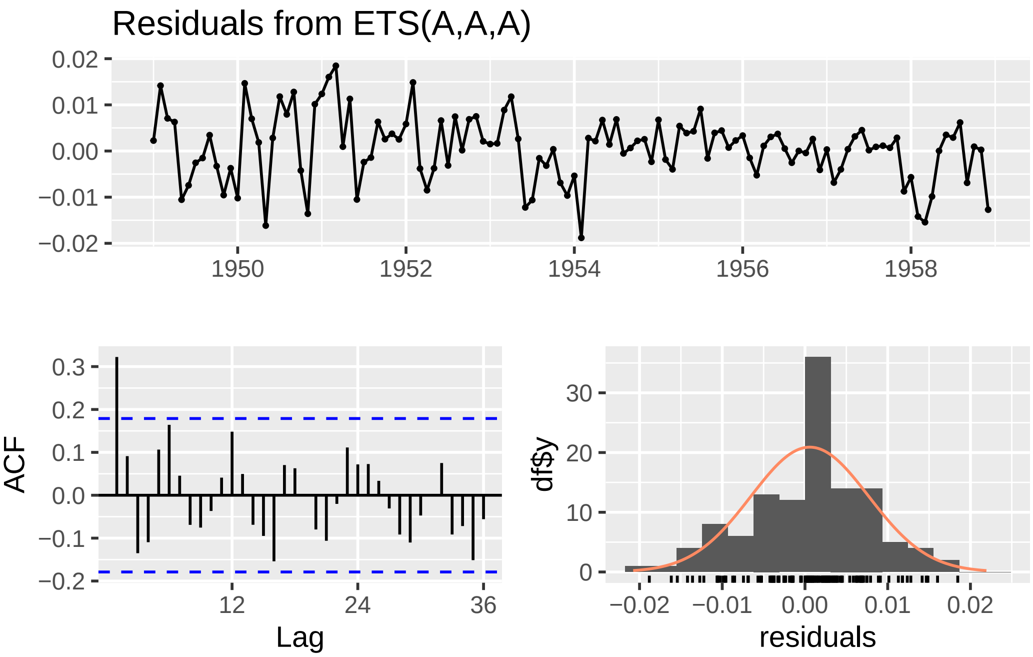


Figure 7: `AirPassengers` ETS

Box-Cox ETS ETS , . .

```
fit_ets <- ets(train_AP, lambda = lam)
```

```
checkresiduals(fit_ets)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,A)
## Q* = 39.722, df = 8, p-value = 3.61e-06
##
## Model df: 16.   Total lags used: 24
```

2. ARMA

, . .

dummy .

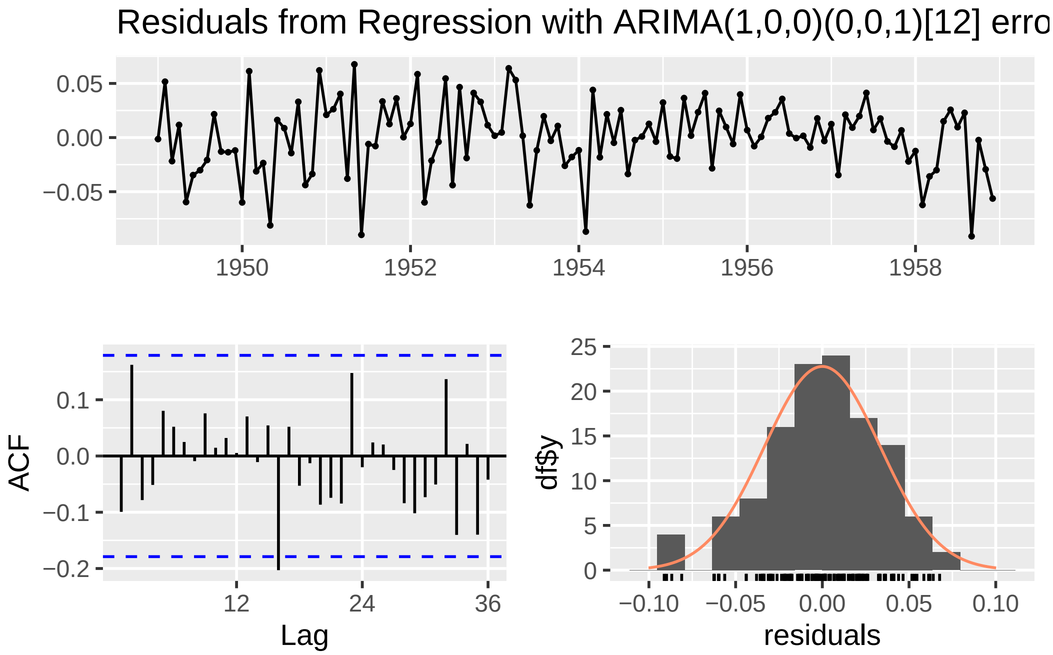
```
Time <- time(train_AP)
Month <- seasonaldummy(train_AP)
fit_r1 <- auto.arima(train_AP, xreg = cbind(Time, Month),
                     lambda = 0)
```

stepwise = FALSE . .

```
fit_r1
## Series: train_AP 
## Regression with ARIMA(1,0,0)(0,0,1)[12] errors 
## Box Cox transformation: lambda= 0 
## 
## Coefficients:
##          ar1    sma1  intercept   Time  Month.Jan  Month.Feb  Month.Mar
##       0.7766  0.1651  -236.9229  0.124     0.0127     0.0017     0.1369
## s.e.  0.0674  0.0994     9.7959  0.005     0.0133     0.0173     0.0198
##       Month.Apr  Month.May  Month.Jun  Month.Jul  Month.Aug  Month.Sep
##          0.0956     0.0872     0.2124     0.3096     0.3018     0.1660
## s.e.     0.0212     0.0220     0.0223     0.0219     0.0211     0.0195
##       Month.Oct  Month.Nov
##          0.0254    -0.1166
## s.e.     0.0170     0.0128
## 
## sigma^2 = 0.001265:  log likelihood = 237.46
## AIC=-442.92   AICc=-437.64   BIC=-398.32
```

. . .

```
checkresiduals(fit_r1)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0)(0,0,1)[12] errors
## Q* = 21.798, df = 22, p-value = 0.472
##
## Model df: 2.   Total lags used: 24
```

Fourier series . .

```
Time <- time(train_AP)
res <- vector("numeric", 6)
for(i in seq(res)){
  xreg <- cbind(Time, fourier(train_AP, K = i))
  fit <- auto.arima(train_AP, xreg = xreg,
                    lambda = 0)
  res[i] <- fit$aicc
}
```

```
(min_k <- which.min(res))
## [1] 5
```

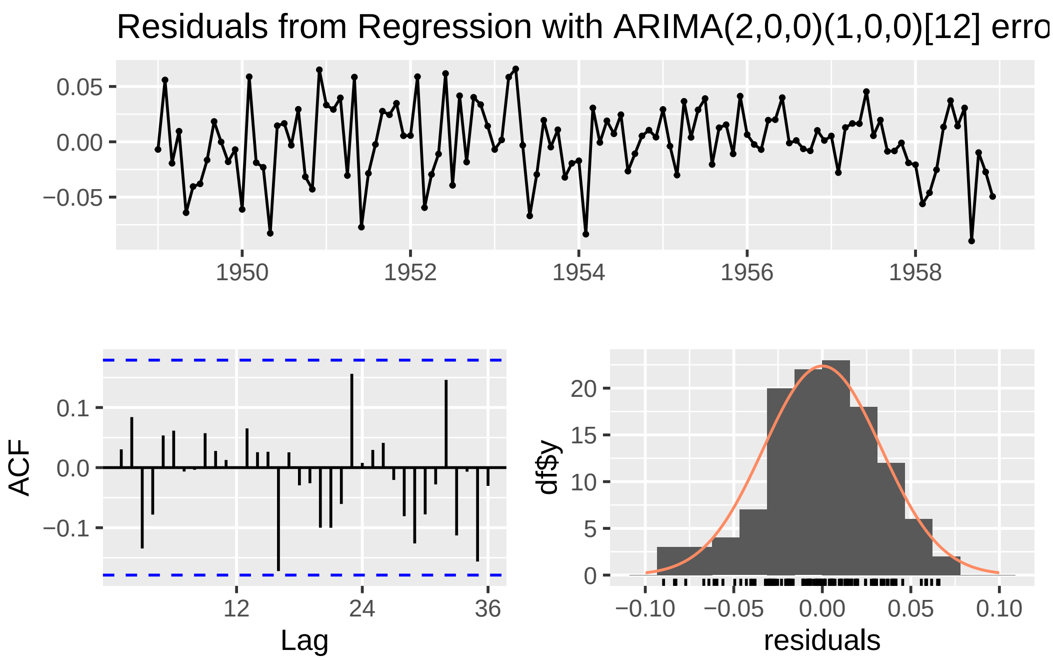
$K = 5$. Fourier series .

```
Time <- time(train_AP)
Fourier <- fourier(train_AP, K = min_k)
fit_r2 <- auto.arima(train_AP, xreg = cbind(Time, Fourier),
                     lambda = 0)
```

```
summary(fit_r2)
## Series: train_AP
## Regression with ARIMA(2,0,0)(1,0,0)[12] errors
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1     ar2    sar1  intercept    Time  Fourier.S1-12  Fourier.C1-12
##       0.6384  0.1782  0.2060  -233.8237  0.1224        -0.0464        -0.1379
## s.e.  0.0910  0.0964  0.1057    12.3191  0.0063         0.0090         0.0090
##       Fourier.S2-12  Fourier.C2-12  Fourier.S3-12  Fourier.C3-12  Fourier.S4-12
##              0.0773        -0.0259        -0.0107         0.0264         0.0245
## s.e.         0.0051         0.0051         0.0039         0.0039         0.0036
##       Fourier.C4-12  Fourier.S5-12  Fourier.C5-12
##              0.0261         0.0206         0.0060
## s.e.         0.0036         0.0036         0.0036
##
## sigma^2 = 0.001256:  log likelihood = 237.75
## AIC=-443.49   AICc=-438.21   BIC=-398.89
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 0.1302685 8.156184 6.052862 -0.05051831 2.597827 0.2118306
##                    ACF1
## Training set 0.09399082
```

. . .

```
checkresiduals(fit_r2)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,0)(1,0,0)[12] errors
## Q* = 17.88, df = 21, p-value = 0.6566
##
## Model df: 3.   Total lags used: 24
```

. . 11 Fourier series 10 , $K = 6$
11 Fourier series dummy . AICc .

```
c(fit_r1$aicc, fit_r2$aicc)
## [1] -437.6420 -438.2119
```

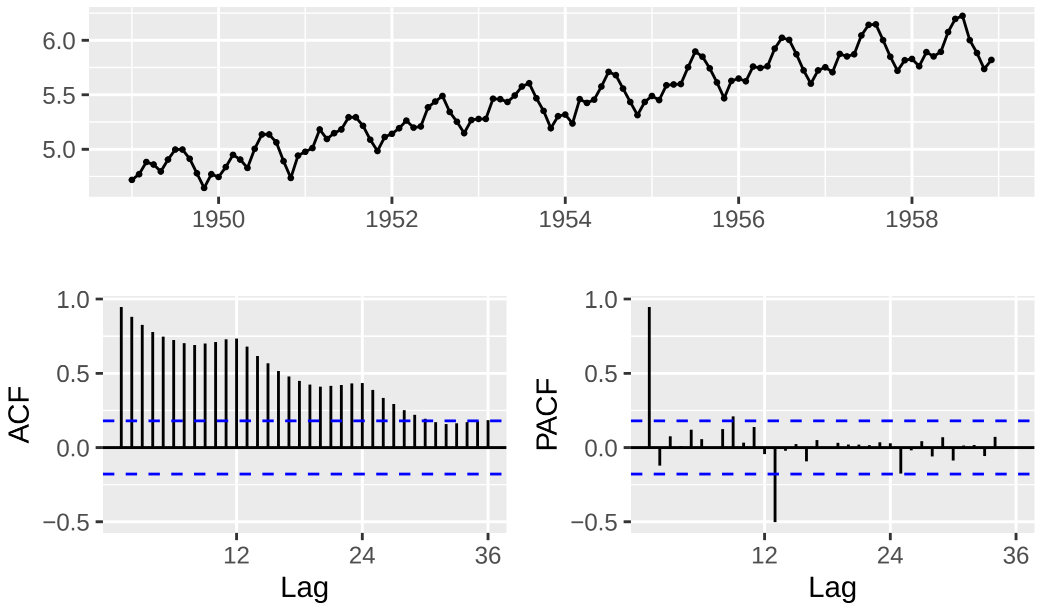
AICc . Fourier series .

```
fit_reg <- fit_r2
```

3. ARIMA

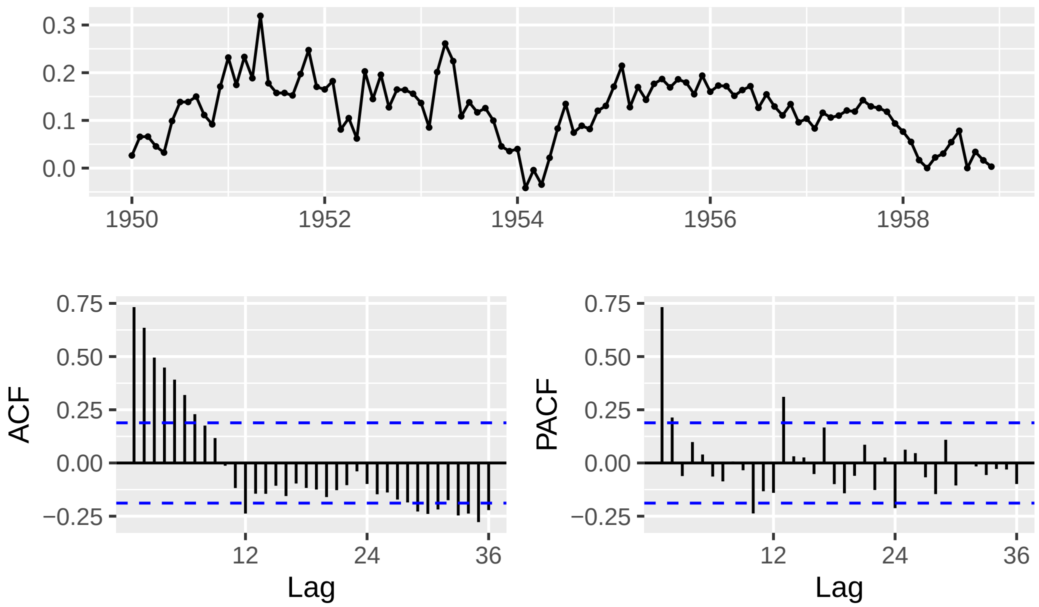
ARIMA , . .
ACF .

```r
train_AP %>%
  log() %>%
  ggtsdisplay()
```



. , .

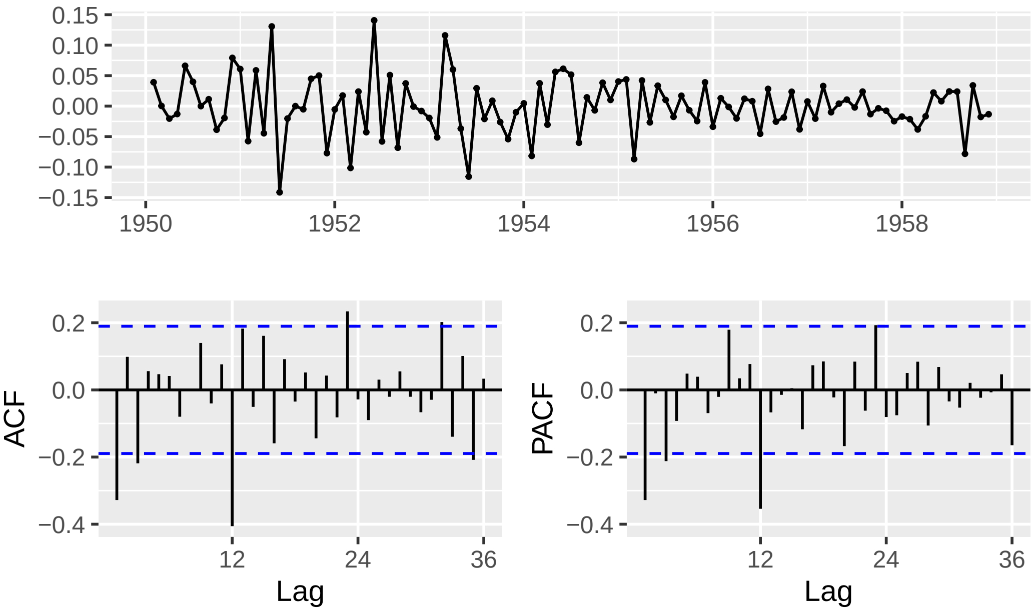
```r
train_AP %>%
  log() %>%
  diff(lag = 12) %>%
  ggtsdisplay()
```



ACF 1 . ARIMA .

1 , . .

```
train_AP %>%
  log() %>%
  diff(lag = 12) %>%
  diff() %>%
  ggtsdisplay()
```



. 1 .

```
train_AP %>%
  log() %>%
  ndiffs()
## [1] 1
```

```
train_AP %>%
  log() %>%
  nsdiffs()
## [1] 1
```

1 , 1 ARIMA . 1
ARIMA .

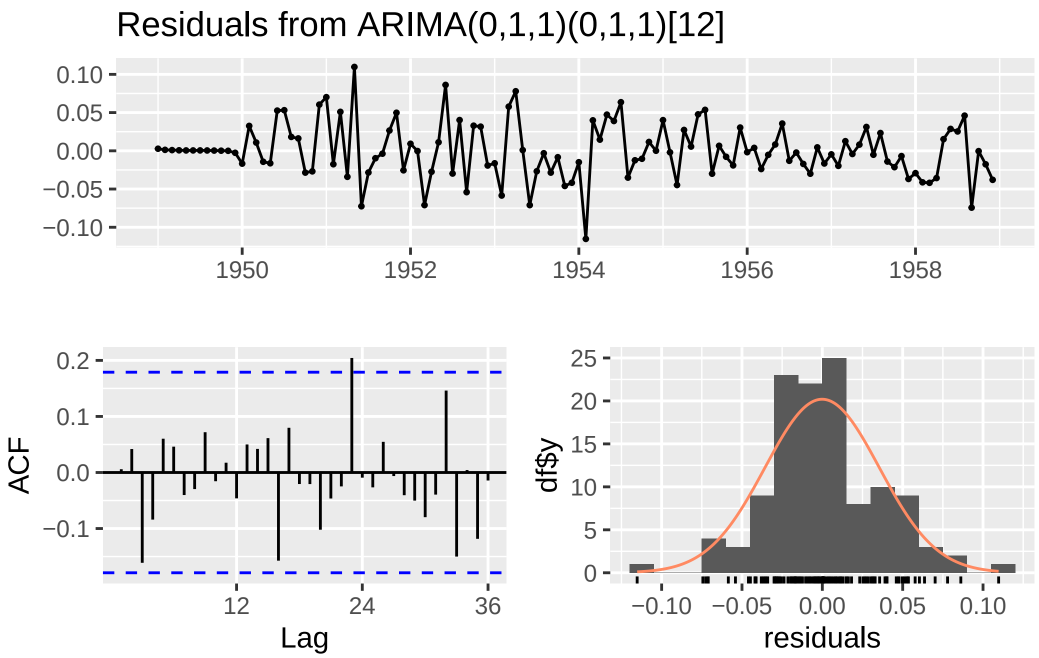
```
fit_a1 <- auto.arima(train_AP, lambda = 0,
                     stepwise = FALSE)
```

.

```
summary(fit_a1)
## Series: train_AP
## ARIMA(0,1,1)(0,1,1)[12]
## Box Cox transformation: lambda= 0
##
## Coefficients:
##           ma1     sma1
##       -0.3424  -0.5405
## s.e.   0.1009   0.0877
##
## sigma^2 = 0.001432:  log likelihood = 197.51
## AIC=-389.02   AICc=-388.78   BIC=-381
##
## Training set error measures:
##                       ME     RMSE     MAE         MPE     MAPE      MASE
## Training set -0.2372088 8.835339 6.51704 -0.07508532 2.637955 0.2280753
##                     ACF1
## Training set 0.04249699
```

.

```
checkresiduals(fit_a1)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)(0,1,1)[12]
## Q* = 20.34, df = 22, p-value = 0.5618
##
## Model df: 2.   Total lags used: 24
```

ARIMA .

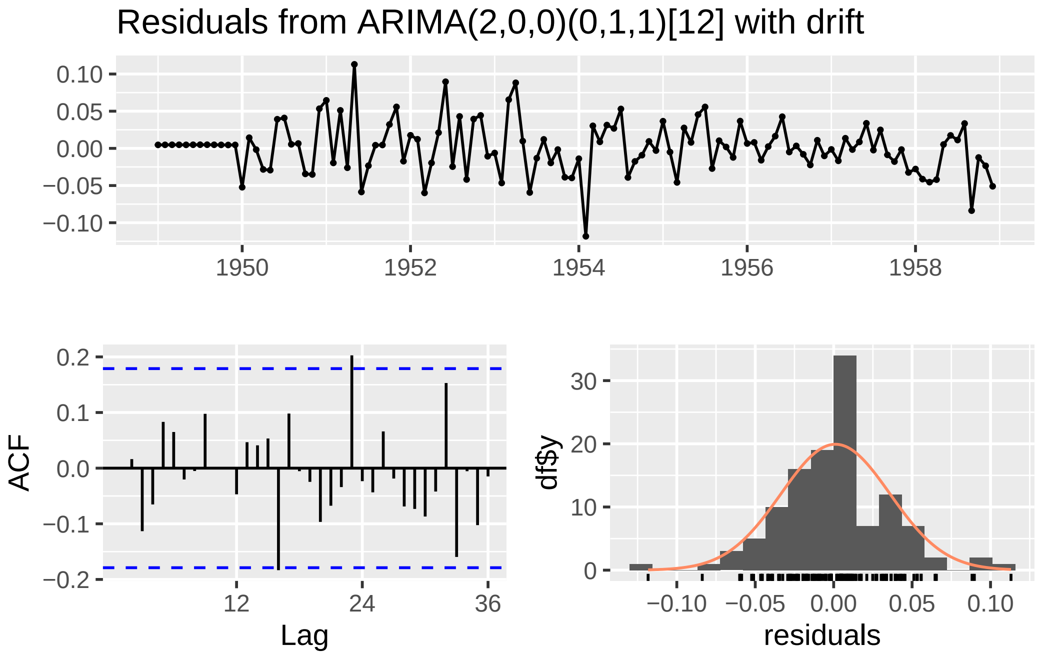
```
fit_a2 <- auto.arima(train_AP, d = 0, lambda = 0,
                     stepwise = FALSE)
```

.

```
fit_a2
## Series: train_AP
## ARIMA(2,0,0)(0,1,1)[12] with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1     ar2     sma1   drift
##       0.6159  0.2356  -0.5562  0.0101
## s.e.  0.0944  0.0965   0.0898  0.0010
##
## sigma^2 = 0.001382:  log likelihood = 201.77
## AIC=-393.53   AICc=-392.95   BIC=-380.12
```

, .

```
checkresiduals(fit_a2)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,0)(0,1,1)[12] with drift
## Q* = 20.83, df = 21, p-value = 0.4694
##
## Model df: 3.   Total lags used: 24
```

. AICc . Test data
.

```
fc_a1 <- forecast(fit_a1, h = length(test_AP))
fc_a2 <- forecast(fit_a2, h = length(test_AP))
```

```
p1 <- autoplot(fc_a1, include = 0) +
  autolayer(test_AP, color = "red", size = 1) +
  labs(y = NULL, subtitle = "fc_a1")
p2 <- autoplot(fc_a2, include = 0) +
  autolayer(test_AP, color = "red", size = 1) +
  labs(y = NULL, subtitle = "fc_a2")

p1 + p2
```

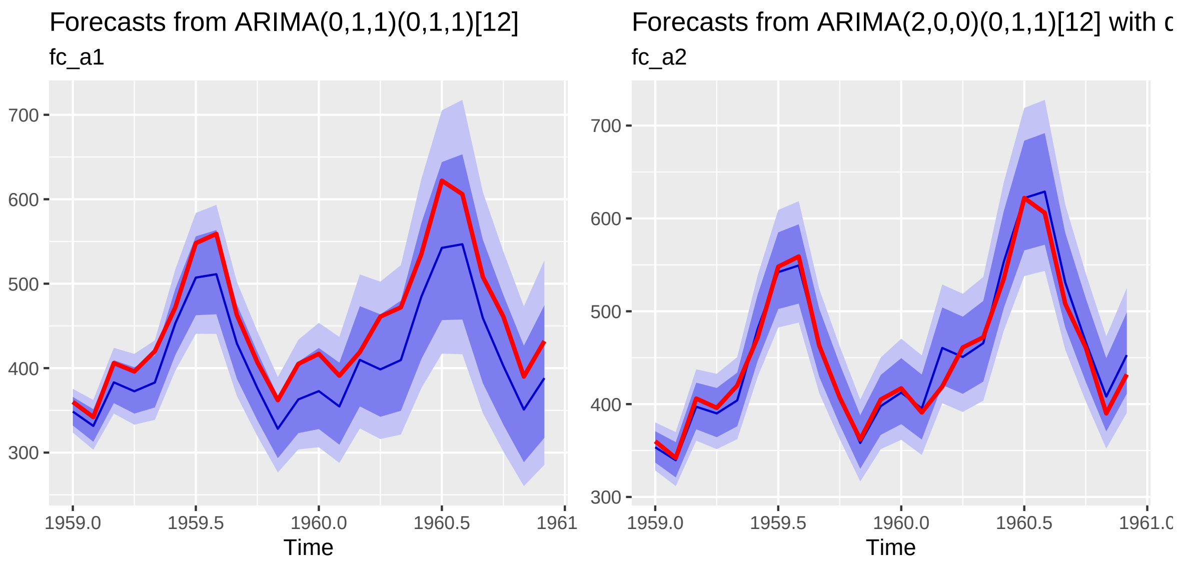


Figure 8: ARIMA

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

```r
fit_arima <- fit_a2
```

ETS ARIMA , ARMA .

```r
new_t <- cbind(Time = time(test_AP),
               Fourier = fourier(test_AP, K = min_k))
fc_reg <- forecast(fit_r2, xreg = new_t)

fc_ets <- forecast(fit_ets, h = length(test_AP))
fc_arima <- forecast(fit_arima, h = length(test_AP))
```

```r
accuracy(fc_reg, test_AP)
##                      ME      RMSE       MAE         MPE     MAPE      MASE
## Training set   0.1302685  8.156184  6.052862 -0.05051831 2.597827 0.2118306
## Test set     -10.8769172 22.234136 15.437231 -2.58040950 3.488570 0.5402531
##                    ACF1 Theil's U
## Training set 0.09399082        NA
## Test set     0.39683635 0.4785114
```

```
accuracy(fc_ets, test_AP)
##                      ME      RMSE       MAE        MPE     MAPE      MASE
## Training set  0.1851383  9.164883  6.647767  0.1873228 2.862835 0.2326503
## Test set     -6.3756691 19.752432 14.199026 -1.4099460 3.219620 0.4969199
##                   ACF1 Theil's U
## Training set 0.3728555        NA
## Test set     0.1682524 0.4355005
```

```
accuracy(fc_arima, test_AP)
##                       ME      RMSE       MAE         MPE     MAPE      MASE
## Training set  0.05350955  8.616827  6.212913  0.05341167 2.560180 0.2174318
## Test set     -3.33508240 14.125641 10.626775 -0.66418334 2.356519 0.3719027
##                    ACF1 Theil's U
## Training set 0.04419876        NA
## Test set     0.14856147 0.2945598
```

ARIMA . .

```
p1 <- autoplot(fc_reg, include = 0) +
  autolayer(test_AP, color = "red", size = 1) +
  labs(x=NULL, y=NULL)
p2 <- autoplot(fc_arima, include = 0) +
  autolayer(test_AP, color = "red", size = 1) +
  labs(x=NULL, y=NULL)
p3 <- autoplot(fc_ets, include = 0) +
  autolayer(test_AP, color = "red", size = 1) +
  labs(x=NULL, y=NULL)

p1 + p2 + p3
```

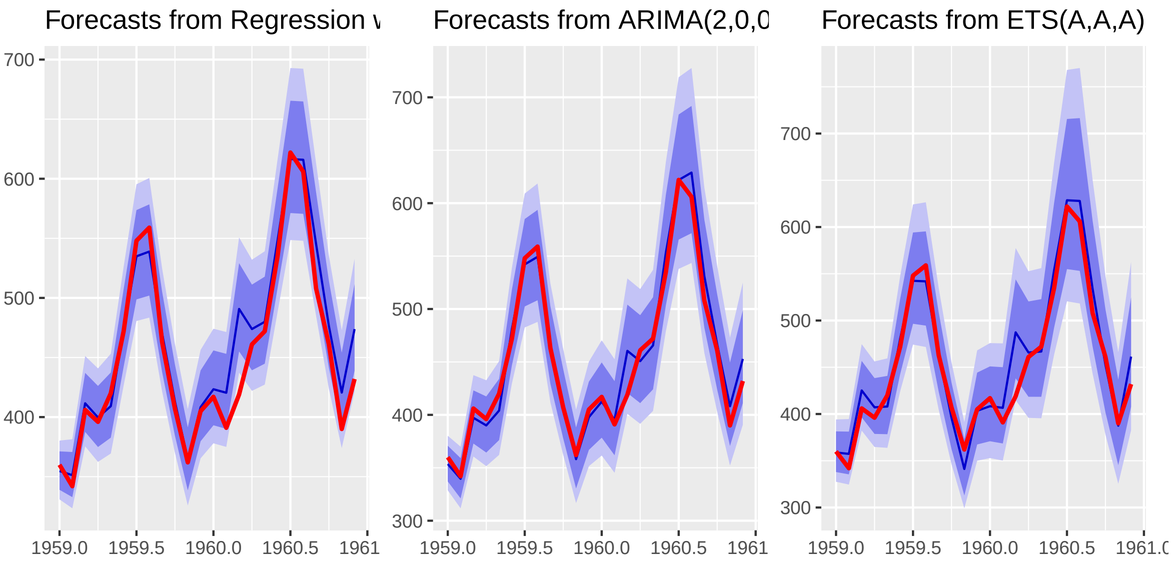


Figure 9: `AirPassengers`

ARMA Dynamic

- : 2014 (`fpp2::elecdaily`)

`elecdaily` 365 × 3 ts . 3 .

```
elecdaily[1:3,]
##          Demand WorkDay Temperature
## [1,] 174.8963       0        26.0
## [2,] 188.5909       1        23.0
## [3,] 188.9169       1        22.2
```

`Demand` , `WorkDay` 0, 1 , `Temperature` . Figure 10 .

```
autoplot(elecdaily, facets = TRUE)
```

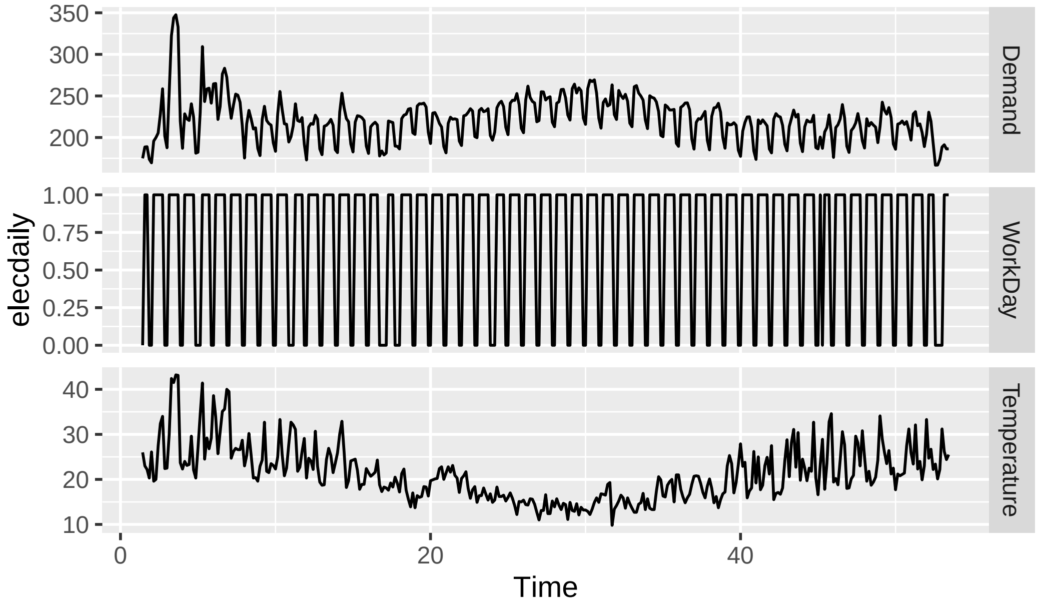


Figure 10: elecdaily

elecdaily .

```r
Demand <- elecdaily[,1]
Work <- elecdaily[,2]
Temp <- elecdaily[,3]
```

, Demand .

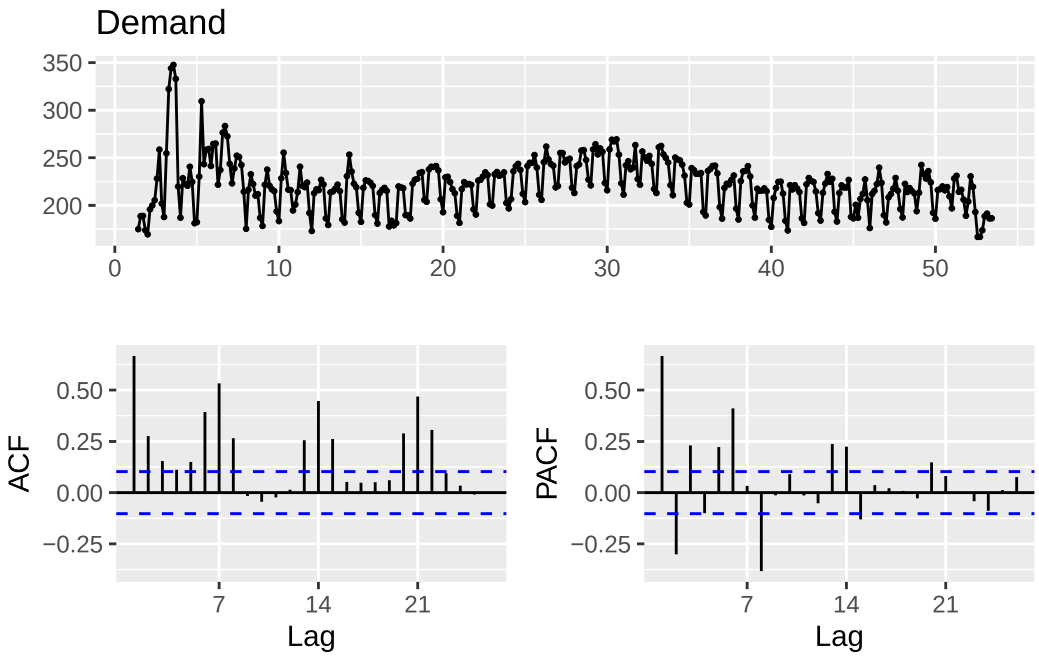
```r
start(Demand); end(Demand); frequency(Demand)
## [1] 1 4
## [1] 53  4
## [1] 7
```

2014 , 2014 53 . 7 . lubridate wday() .

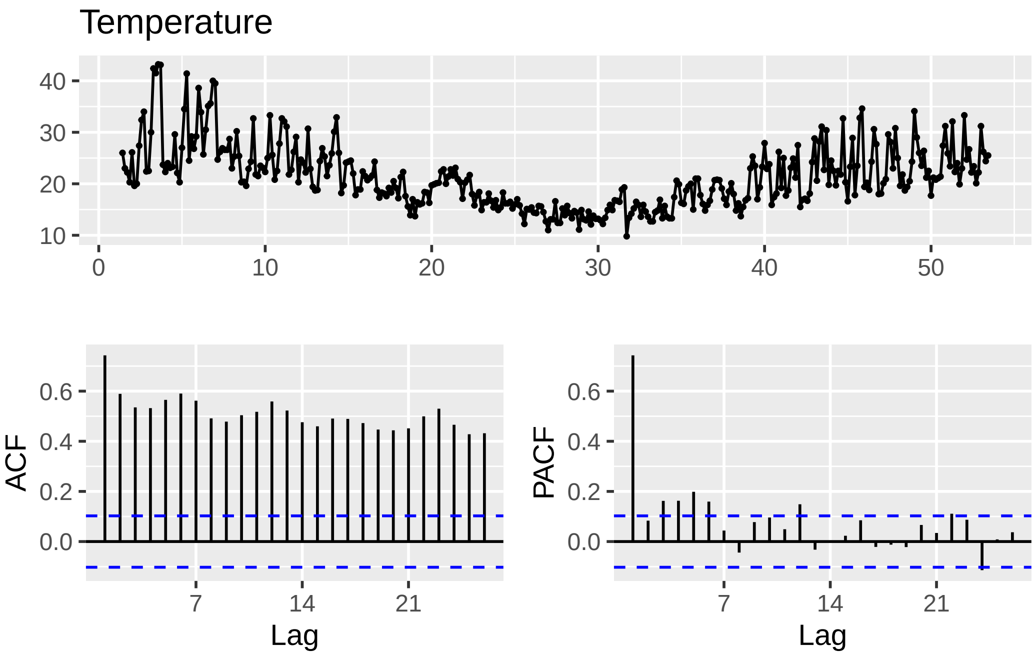
```r
library(lubridate)
wday(ymd("2014-1-1"), label = TRUE)
## [1]
## Levels:  <  <  <  <  <  <
```

Demand Temp . .

```
Demand %>%
  ggtsdisplay(main = "Demand")
```



```
Temp %>%
  ggtsdisplay(main = "Temperature")
```



.

```
ndiffs(Demand)
## [1] 1
ndiffs(Temp)
## [1] 1
```

. Figure 11 2 . `Temp` .

```
tibble(Demand, Temp) %>%
  ggplot(aes(x = as.numeric(Temp), y = as.numeric(Demand))) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(x = "Temperature", y = "Demand")
```

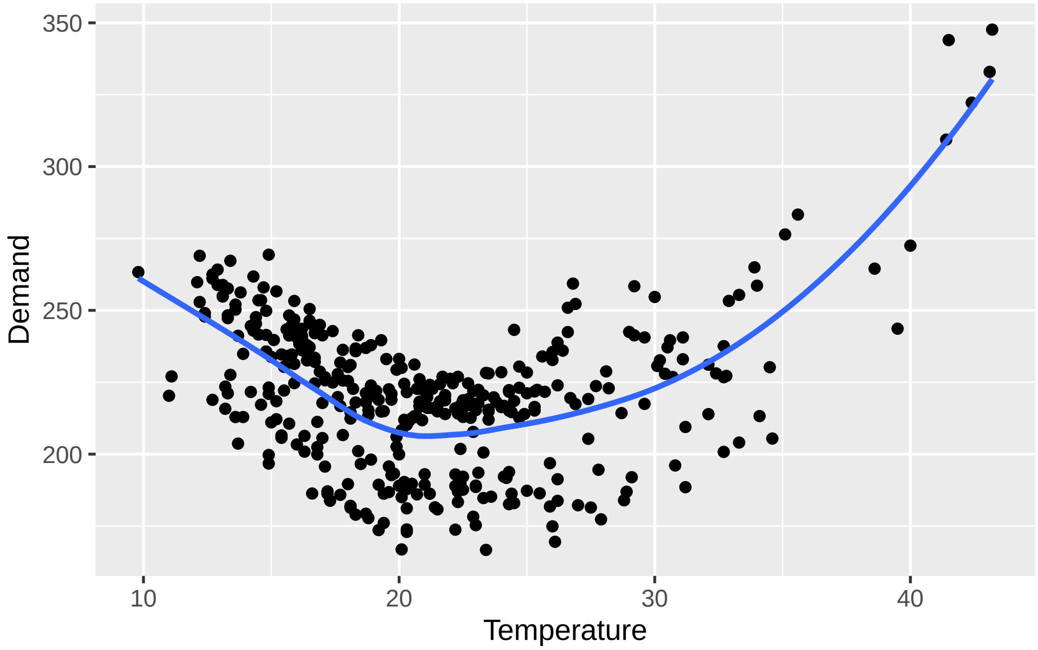


Figure 11: `Demand` `Temperature`

Dynamic . `auto.arima()` `Temp` `Temp^2`, `Work` `xreg` .

```r
xreg <- cbind(Temp, Temp2 = Temp^2, Work)
fit <- auto.arima(Demand, xreg = xreg, stepwise = FALSE)
```

.

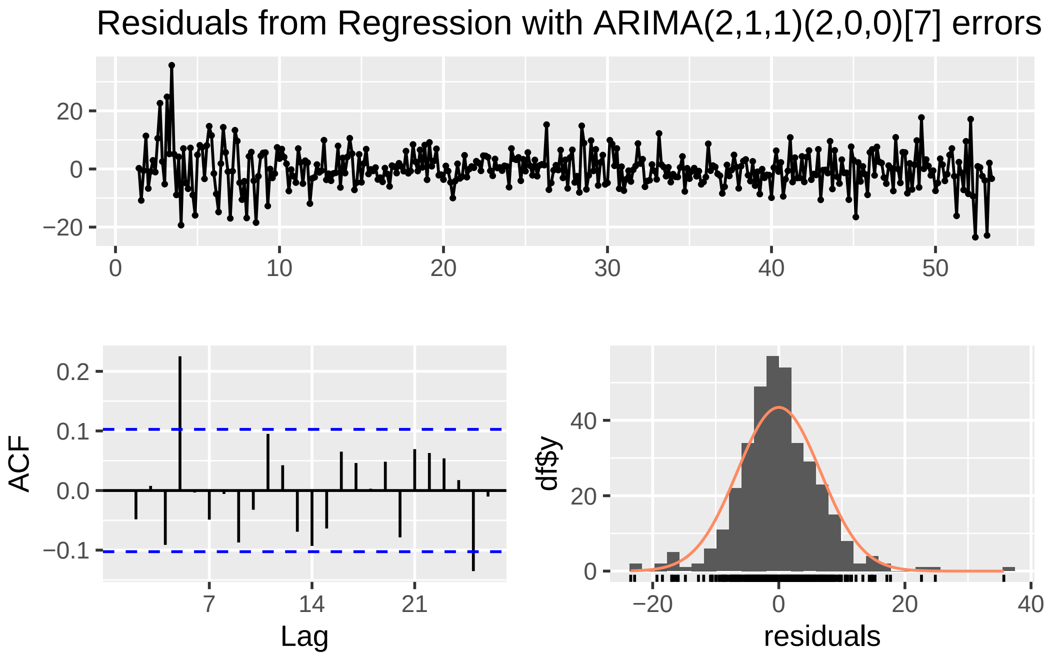
```r
summary(fit)
## Series: Demand
## Regression with ARIMA(2,1,1)(2,0,0)[7] errors
##
## Coefficients:
##          ar1      ar2      ma1    sar1    sar2     Temp   Temp2     Work
##       0.8247  -0.0225  -0.9806  0.2216  0.4008  -7.8847  0.1849  30.3192
## s.e.  0.0700   0.0666   0.0203  0.0552  0.0566   0.4457  0.0088   1.3390
##
## sigma^2 = 44.7:  log likelihood = -1205.77
## AIC=2429.54   AICc=2430.04   BIC=2464.61
##
## Training set error measures:
```

```
##                       ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 0.01290229 6.602876 4.767955 -0.09519977 2.159123 0.3273729
##                      ACF1
## Training set -0.000989946
```

, .

. .

```
checkresiduals(fit)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,1,1)(2,0,0)[7] errors
## Q* = 36.219, df = 9, p-value = 3.625e-05
##
## Model df: 5.   Total lags used: 14
```

Dynamic 2015 1 1 1 10 . Temp . Temp , Demand .

2014 1 1 1 10 `Temp` . `Work` 2015 1 1 1 10 .

```
old_T <- Temp[1:10]
new_x <- cbind(Temp = old_T, Temp2 = old_T^2,
               Work = c(0,1,0,0,1,1,1,1,1,0))
fc <- forecast(fit, xreg = new_x)
```

.

```
autoplot(fc)
```

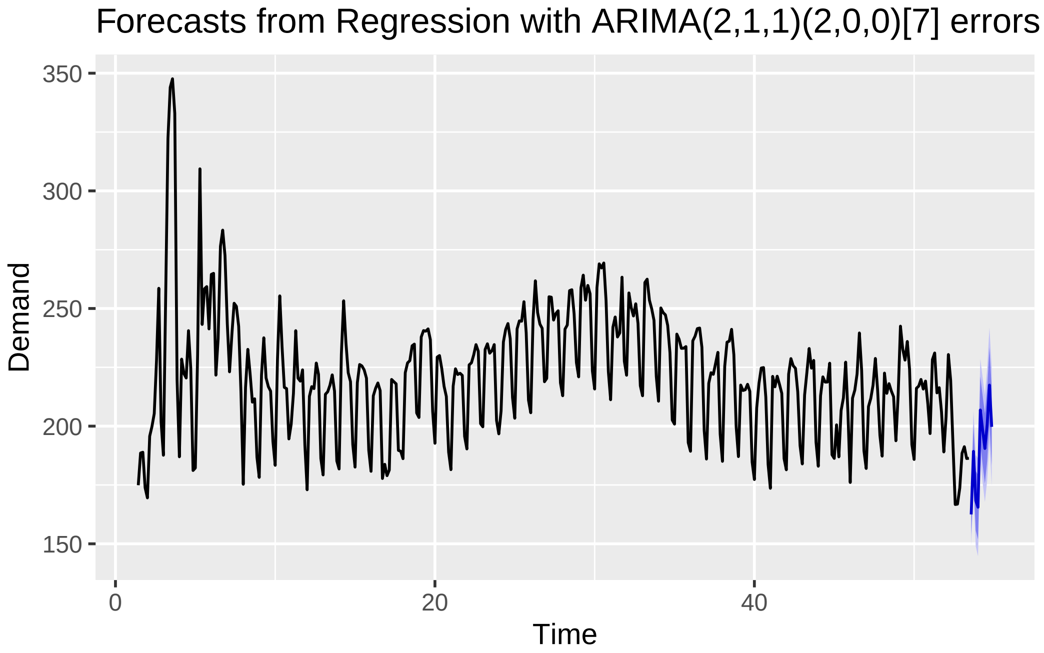


Figure 12: `Demand`

- : , 1970 1 2016 3 (`fpp2::uschange`)

`uschange` 187×5 ts .

```
uschange[1:3,]
##      Consumption   Income Production  Savings Unemployment
## [1,]   0.6159862 0.972261 -2.4527003 4.810312          0.9
## [2,]   0.4603757 1.169085 -0.5515251 7.287992          0.5
## [3,]    0.8767914 1.553271 -0.3587079 7.289013            0.5
```

, , . 4 , 1970 1 , 2016 3 .

```
start(uschange[,1])
## [1] 1970    1
end(uschange[,1])
## [1] 2016    3
frequency(uschange[,1])
## [1] 4
```

Consumption . ARIMA ETS , ARMA , uschange dynamic .

uschange Figure 13 . .

```
autoplot(uschange, facets=TRUE) +
  labs(y = NULL, x = NULL)
```

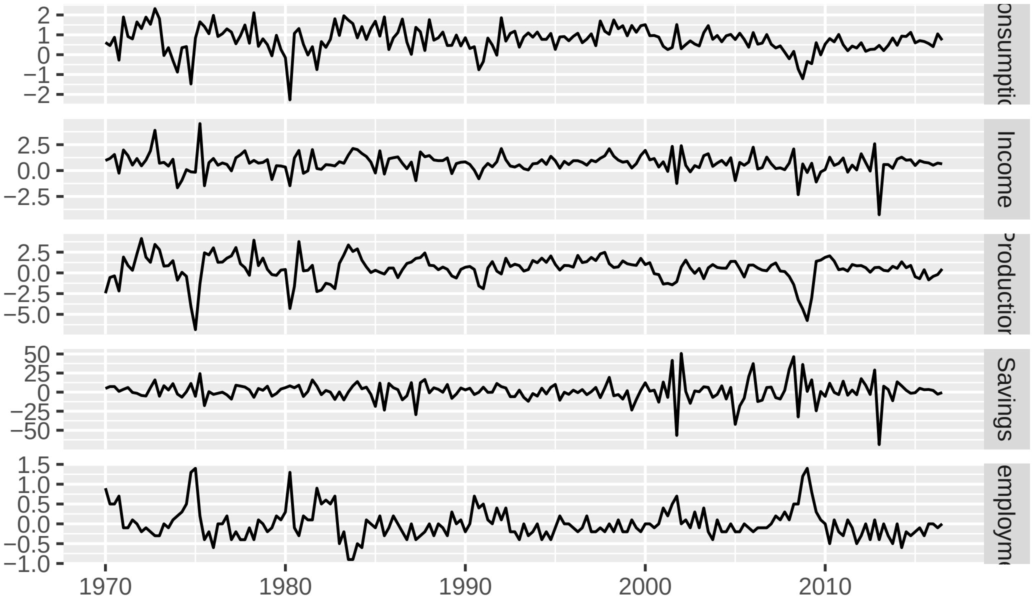


Figure 13: uschange

ACF ggAcf() , . ggAcf() uschange

, ACF .

```r
ggAcf(uschange)
```

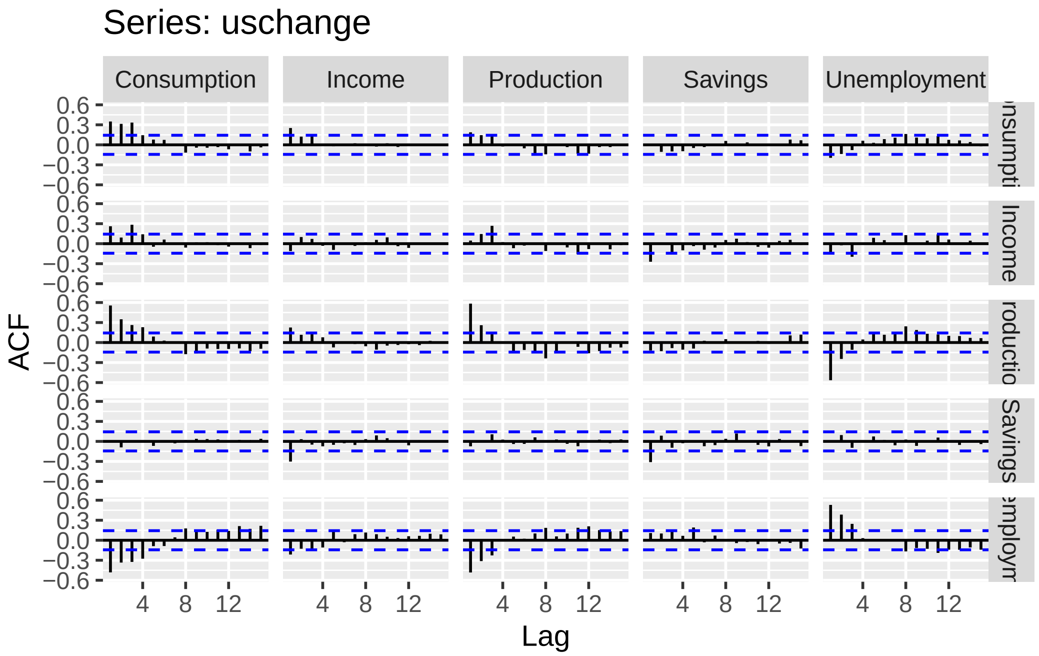


Figure 14: uschange ACF

Figure 13 Figure 14 .

Dynamic Income, Production, Savings, Unemployment . GGally::ggpairs() .

```r
GGally::ggpairs(as_tibble(uschange) %>%
                  relocate(Consumption, .after=last_col()),
                lower=list(continuous="smooth_loess"))
```

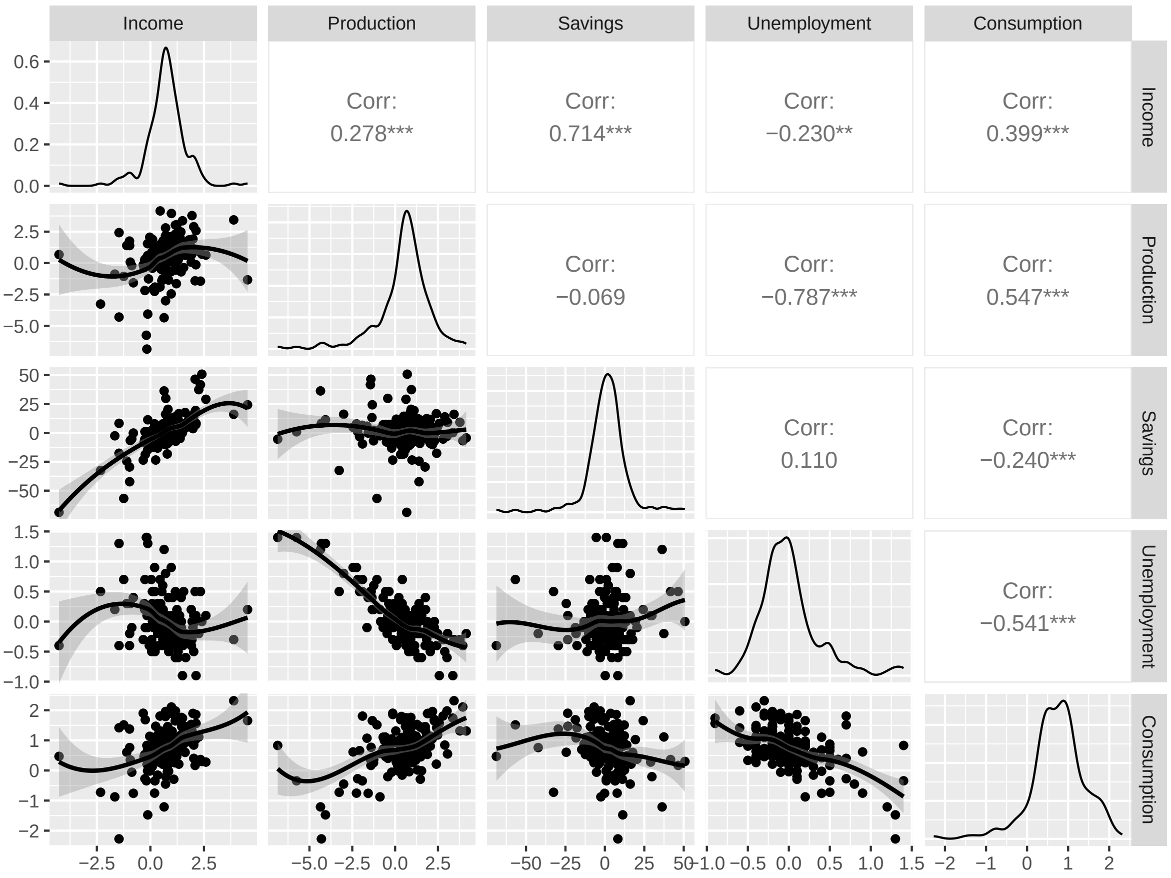


Figure 15: `uschange`

`(Income, Savings)` `(Production, Uneployment)` , `Income` `Production`
. .

.

```
uschange_te <- tail(uschange, n = 8)
uschange_tr <- head(uschange, n = nrow(uschange)-8)
```

ARIMA , .

```
fit_arima <- auto.arima(uschange_tr[,1],
                        stepwise = FALSE, approximation = FALSE)
```

```
fit_arima
## Series: uschange_tr[, 1]
## ARIMA(3,0,0)(2,0,0)[4] with non-zero mean
##
## Coefficients:
##          ar1     ar2     ar3      sar1      sar2    mean
##       0.2267  0.1771  0.2218  -0.0351  -0.1792  0.7482
## s.e.  0.0739  0.0738  0.0726   0.0774   0.0745  0.0951
##
## sigma^2 = 0.3544:  log likelihood = -158.39
## AIC=330.78   AICc=331.44   BIC=353.09
```

ETS , .

```
fit_ets <- ets(uschange_tr[,1])
```

```
fit_ets
## ETS(A,N,N)
##
## Call:
##  ets(y = uschange_tr[, 1])
##
##   Smoothing parameters:
##     alpha = 0.3315
##
##   Initial states:
##     l = 0.6877
##
##   sigma:  0.633
##
##      AIC     AICc      BIC
## 768.8004 768.9376 778.3626
```

ARMA . `time()` , dummy .

```
Time <-  time(uschange_tr[,1])
Qtr <-  seasonaldummy(uschange_tr[,1])
```

```
fit_reg <- auto.arima(uschange_tr[,1],
                      xreg = cbind(Time, Qtr),
                      stepwise = FALSE, approximation = FALSE)
```

```
fit_reg
## Series: uschange_tr[, 1] 
## Regression with ARIMA(3,0,0)(0,0,2)[4] errors 
## 
## Coefficients:
##           ar1     ar2     ar3     sma1     sma2   Time  Qtr.Q1  Qtr.Q2  Qtr.Q3
##        0.2564  0.1608  0.2482  -0.1066  -0.1942  3e-04  0.0554  0.0122   0.172
## s.e.   0.0742  0.0737  0.0729   0.0773   0.0707  1e-04  0.0700  0.0748   0.070
## 
## sigma^2 = 0.349:  log likelihood = -155.52
## AIC=331.03   AICc=332.34   BIC=362.91
```

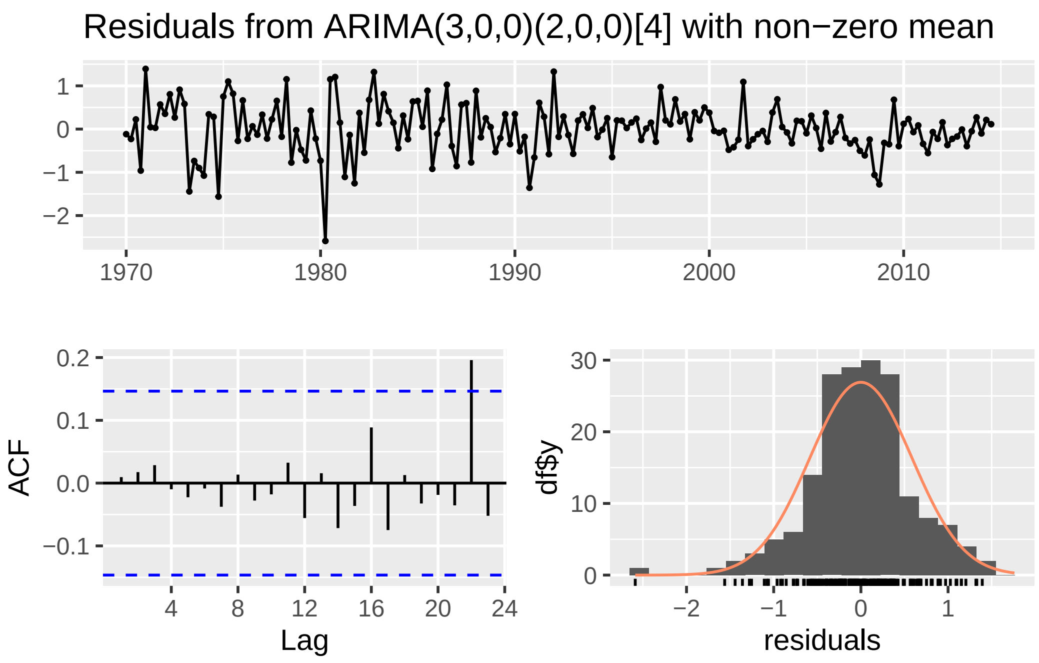
Dynamic .

```
fit_dyn <- auto.arima(uschange_tr[,1], d = 0,
                   xreg = uschange_tr[,c(2,3)],
                   stepwise = FALSE, approximation = FALSE)
```

```
fit_dyn
## Series: uschange_tr[, 1] 
## Regression with ARIMA(3,0,0) errors 
## 
## Coefficients:
##           ar1     ar2     ar3  intercept  Income  Production
##        0.0060  0.1960  0.1890     0.5288  0.1741      0.1758
## s.e.   0.0813  0.0734  0.0735     0.0708  0.0457      0.0262
## 
## sigma^2 = 0.2696:  log likelihood = -133.71
## AIC=281.41   AICc=282.07   BIC=303.72
```

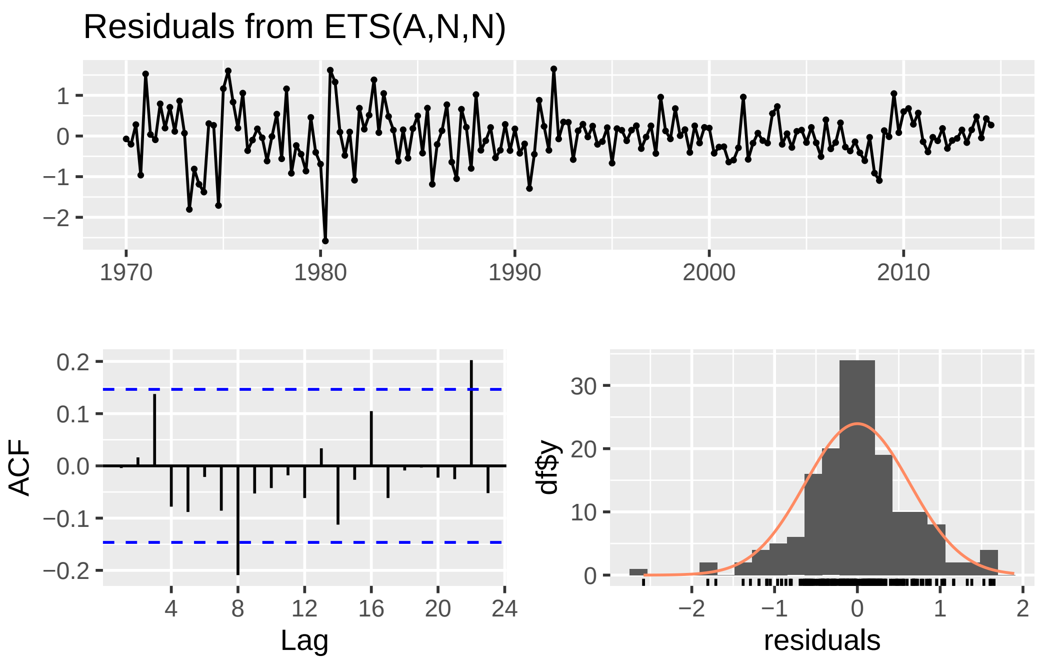
. , Ljung-Box ETS .

```
checkresiduals(fit_arima)
```



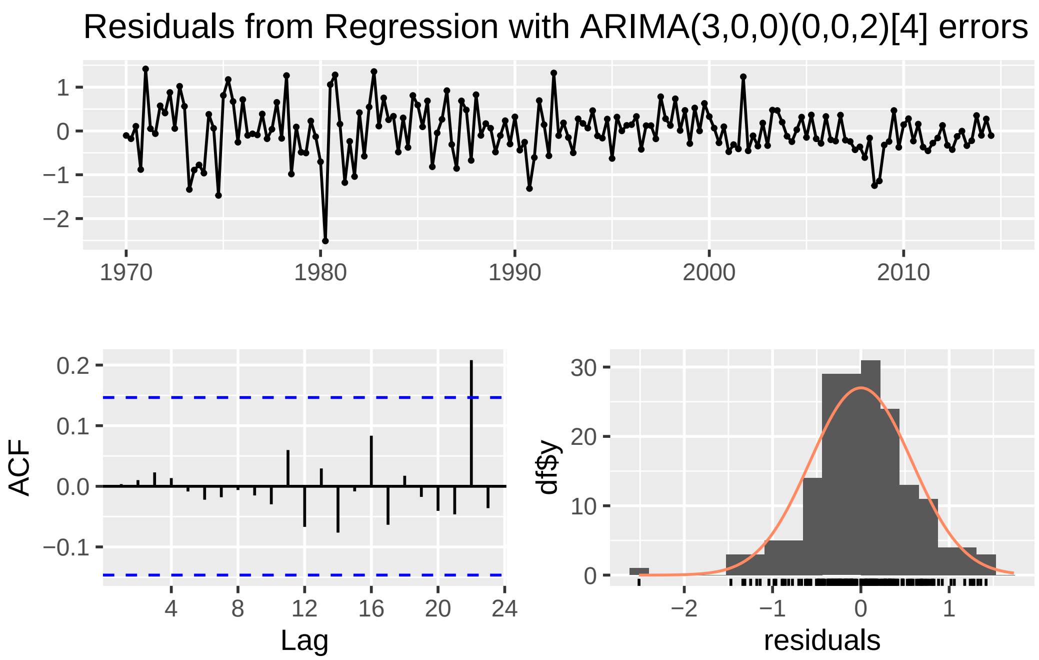
```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,0)(2,0,0)[4] with non-zero mean
## Q* = 0.6507, df = 3, p-value = 0.8847
##
## Model df: 5.   Total lags used: 8
```

```
checkresiduals(fit_ets)
```



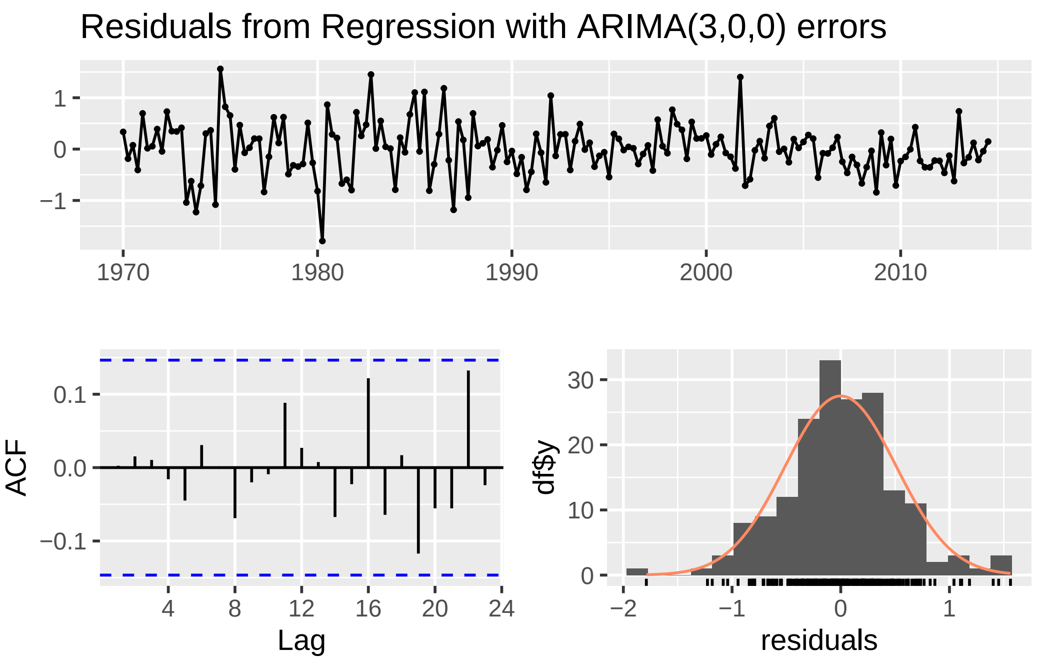
```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,N,N)
## Q* = 15.865, df = 6, p-value = 0.0145
##
## Model df: 2.   Total lags used: 8
```

```
checkresiduals(fit_reg)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0)(0,0,2)[4] errors
## Q* = 0.32544, df = 3, p-value = 0.9552
##
## Model df: 5.   Total lags used: 8
```

```
checkresiduals(fit_dyn)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 1.5616, df = 5, p-value = 0.9059
##
## Model df: 3.   Total lags used: 8
```

test data . `fit_dyn` test data , .
`fit_dyn` .

```
fc_arima <- forecast(fit_arima, h = 8)
fc_ets <- forecast(fit_ets, h = 8)
fc_dyn <- forecast(fit_dyn,
                   xreg = uschange_te[,c(2,3)])
```

```
Time <-  time(uschange_te[,1])
Qtr <-  seasonaldummy(uschange_te[,1])
fc_reg <- forecast(fit_reg,
                   xreg = cbind(Time, Qtr))
```

test data . `fit_dyn` . ETS .

```
accuracy(fc_arima, uschange_te[,1])
##                         ME      RMSE       MAE       MPE      MAPE     MASE
## Training set  0.0002559072 0.5852267 0.4396851  65.90037 189.15841 0.670526
## Test set     -0.0475341277 0.2500710 0.2185770 -17.75795  33.64265 0.333333
##                     ACF1 Theil's U
## Training set  0.00952753        NA
## Test set     -0.21075717 0.6797118
```

```
accuracy(fc_ets, uschange_te[,1])
##                         ME      RMSE       MAE       MPE      MAPE      MASE
## Training set  0.0008292707 0.6294144 0.4622752  15.94476 163.66594 0.7049764
## Test set     -0.0071405456 0.2275948 0.1819332 -11.07730  27.12135 0.2774507
##                      ACF1 Theil's U
## Training set -0.004088847        NA
## Test set     -0.208694082 0.5901059
```

```
accuracy(fc_reg, uschange_te[,1])
##                         ME      RMSE       MAE       MPE      MAPE      MASE
## Training set  0.0008586772 0.5757214 0.4336545  62.73093 184.11947 0.6613293
## Test set     -0.0460208936 0.2877620 0.2542229 -18.22233  37.33168 0.3876936
##                      ACF1 Theil's U
## Training set  0.003604112        NA
## Test set     -0.360236167 0.7215638
```

```
accuracy(fc_dyn, uschange_te[,1])
##                         ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.0006526438 0.5104141 0.3844923 43.024517 189.5903 0.5863563
## Test set      0.1004867937 0.2061923 0.1498566  8.058113  18.8564 0.2285334
##                      ACF1 Theil's U
## Training set  0.002548898        NA
## Test set     -0.413909940 0.6493148
```

. ETS , ,

.

```
y_lim <- c(-1, 2.5)
p1 <- autoplot(fc_arima, include = 8) +
  autolayer(uschange_te[,1], color = "red", size = .8) +
  ylab(NULL) + ylim(y_lim[1], y_lim[2])
p2 <- autoplot(fc_ets, include = 8) +
  autolayer(uschange_te[,1], color = "red", size = .8) +
  ylab(NULL) + ylim(y_lim[1], y_lim[2])
p3 <- autoplot(fc_reg, include = 8) +
  autolayer(uschange_te[,1], color = "red", size = .8) +
  ylab(NULL) + ylim(y_lim[1], y_lim[2])
p4 <- autoplot(fc_dyn, include = 8) +
  autolayer(uschange_te[,1], color = "red", size = .8) +
  ylab(NULL) + ylim(y_lim[1], y_lim[2])

(p1 + p2) / (p3 + p4)
```

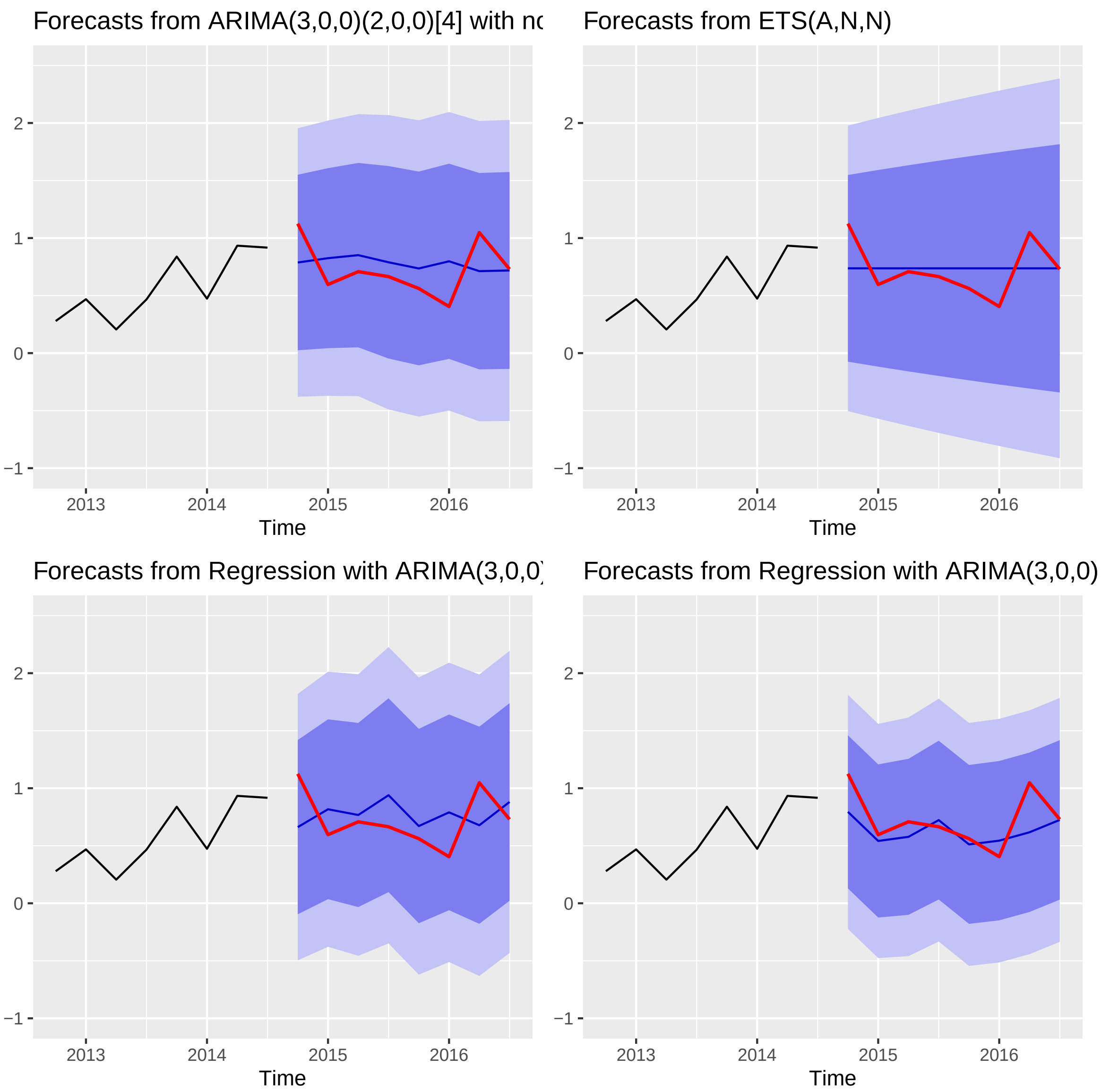


Figure 16: `uschange Consumption`