Analysis of Monthly Flights in Denmark

R Markdown

```
rm(list=ls())
library(dplyr)
library(ggplot2)
library(fpp3)
library(tsibble)
library(seasonal)
library(urca)
library(strucchange)
library(bsts)
```

Read in the dataset

```
flight_traffic <- read.csv("/Users/michellevonhuth/Documents/Copenhagen/CBS/Semester2/Predictive Analyti
```

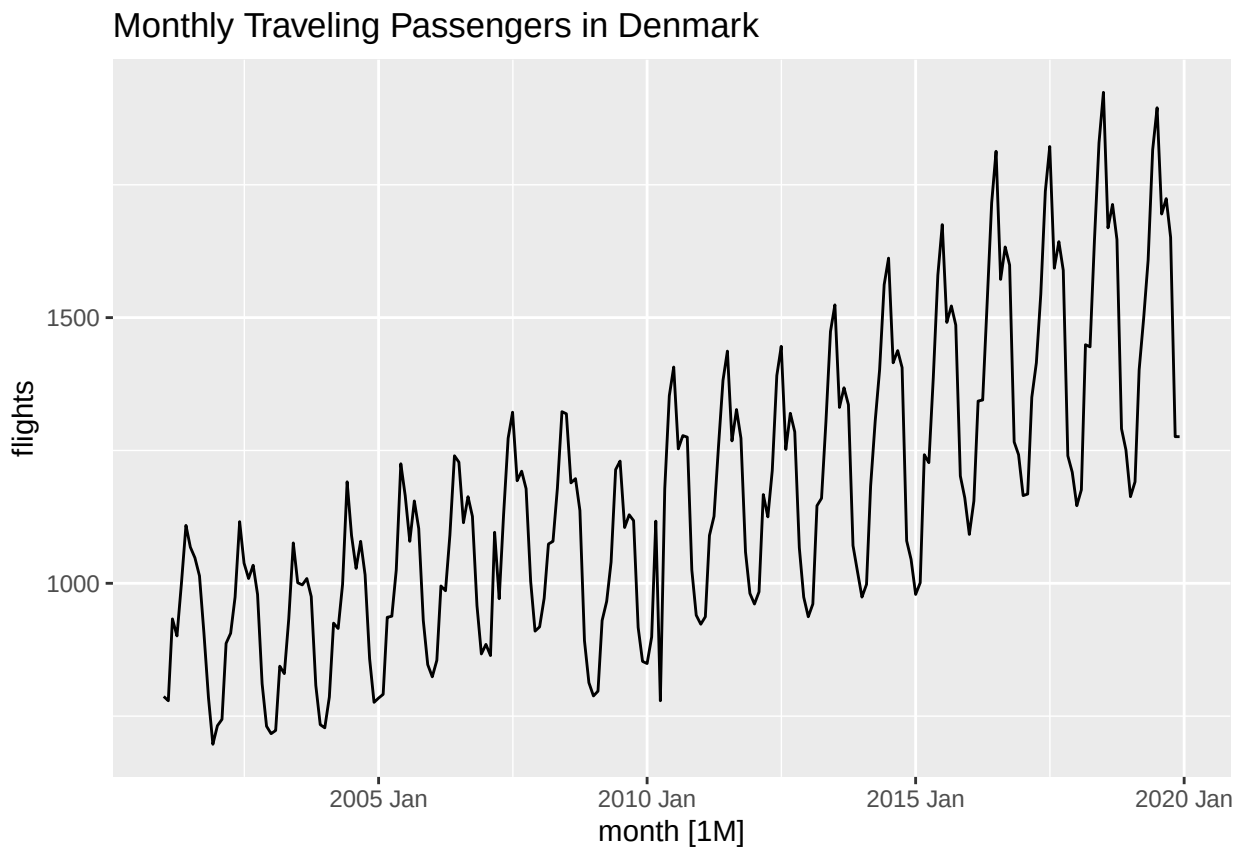
Perform data transformations - convert to a tsibble object

```
flights_reality <- flight_traffic %>%
  mutate(month = yearmonth(as.character(month))) %>%
  as_tsibble(index = month)

# Only use data up intil 2022 Januray
flights <- flights_reality %>%
  filter(yearmonth(month) < yearmonth("2020 January"))
```

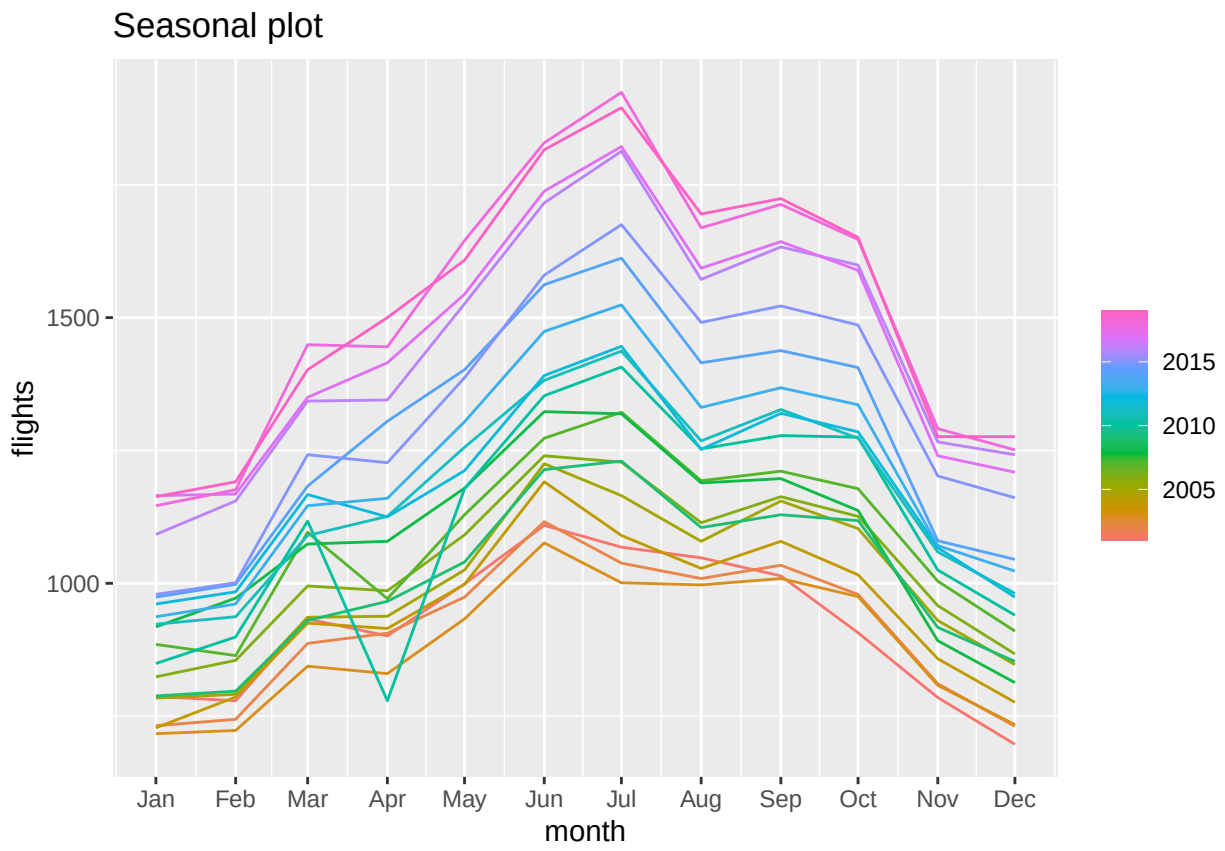
VISUAL INVESTIGATION

Plot data and the perform autocorrelation

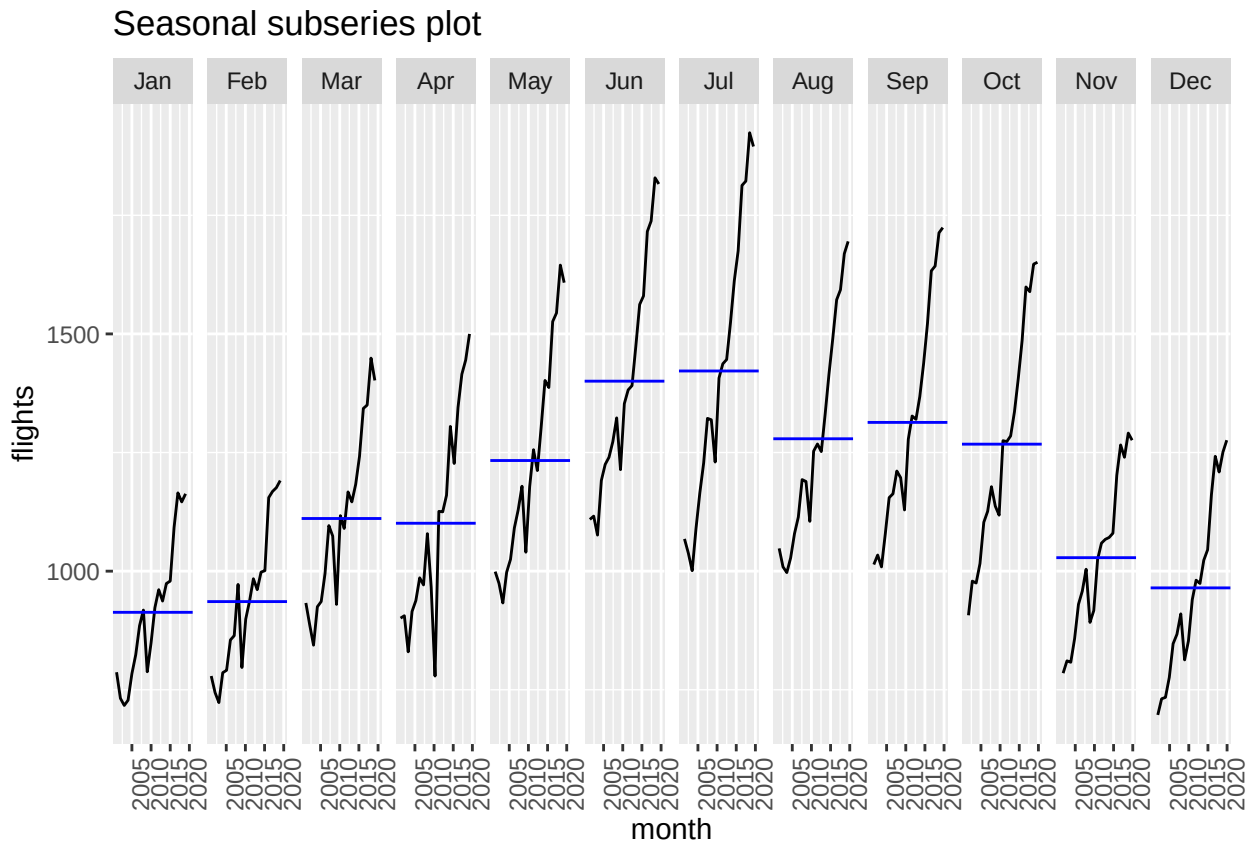
```
flights %>% autoplot(flights) + labs(title = "Monthly Traveling Passengers in Denmark")
```



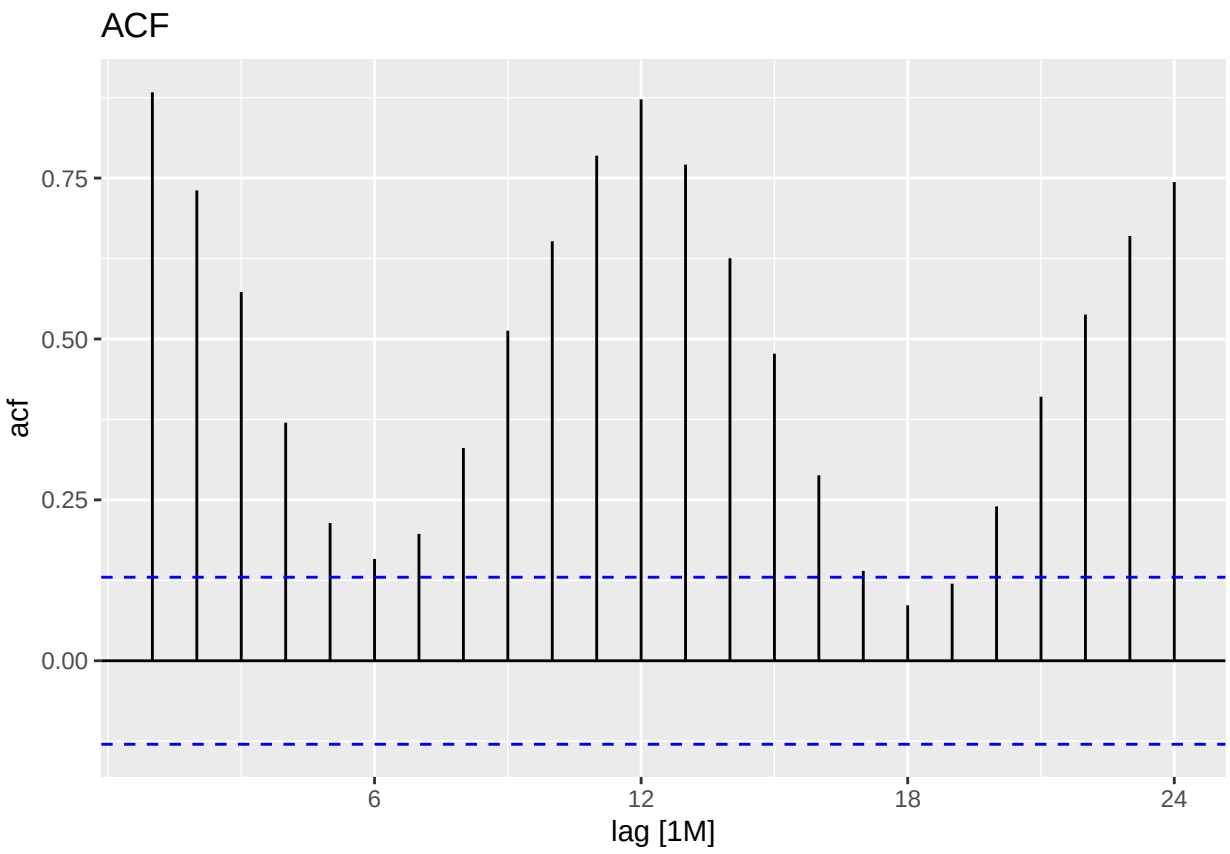

```
flights %>% gg_season(flights) + labs(title = "Seasonal plot")
```




```
flights %>% gg_subseries(flights) + labs(title = "Seasonal subseries plot")
```

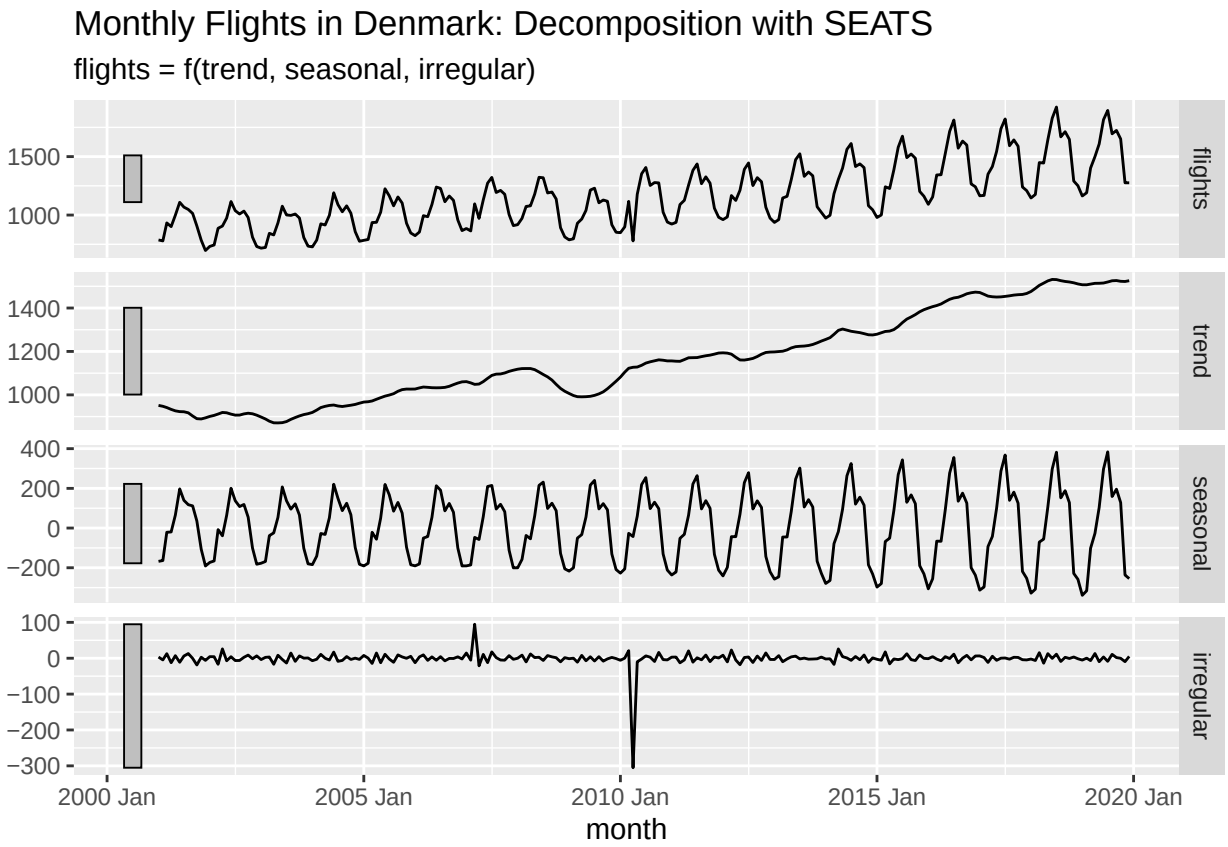



```
flights %>% ACF(flights, lag_max = 24) %>% autoplot() + labs(title = "ACF")
```



Perform SEATS decomposition

```
seats_dcmp <- flights%>%
model(seats = X_13ARIMA_SEATS(flights ~ seats())) %>%
components()

plot_seats <- autoplot(seats_dcmp) +
labs(title = "Monthly Flights in Denmark: Decomposition with SEATS")
print(plot_seats)
```



The SEATS decomposition shows an increasing trend and seasonality throughout the time period analyzed. There is no abrupt dip in the time series. The irregularity in the time series describes variations that are not explained by trend or seasonality.

TRANSFORM DATA

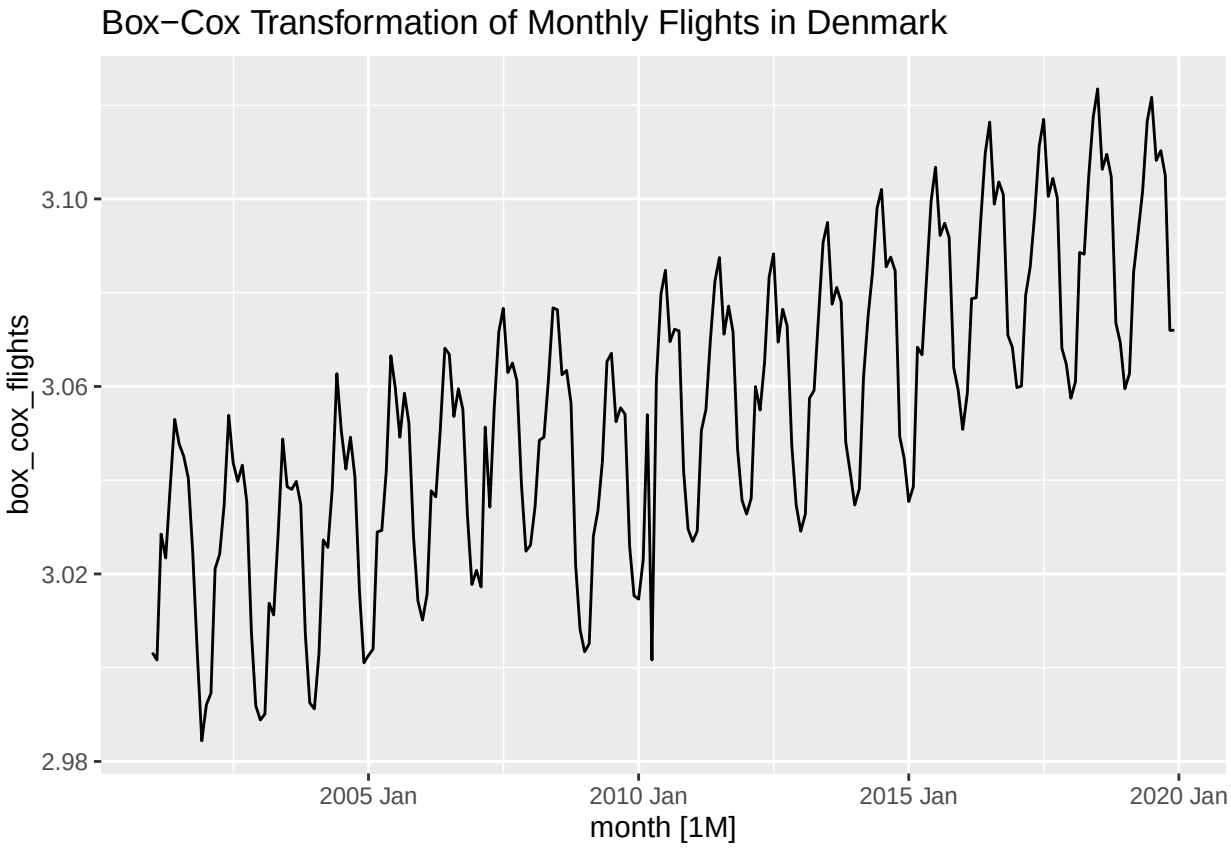
Box-Cox Transformation

```
# Box-Cox Transformation
lambda <- flights %>%
  features(flights, features = guerrero) %>%
  pull(lambda_guerrero)

# Print the value of lambda
print(lambda)
```

```
## [1] -0.2822881
```

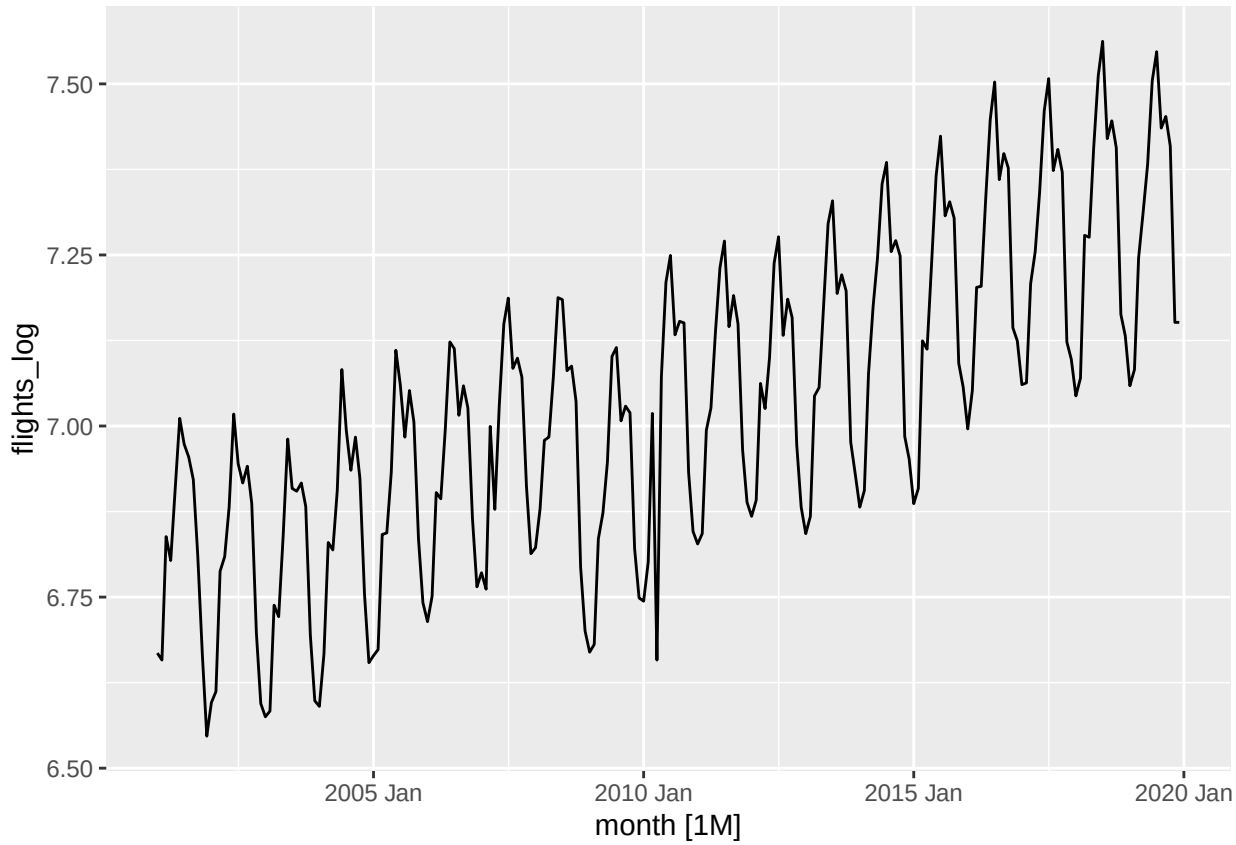
```
# Create new column with transformed data using optimal lambda value
flights <- flights %>%
  mutate(box_cox_flights = box_cox(flights, lambda))

# Visualize transformed data
```

```
flights %>%
  autoplot(box_cox_flights) +
  labs(title = "Box-Cox Transformation of Monthly Flights in Denmark")
```



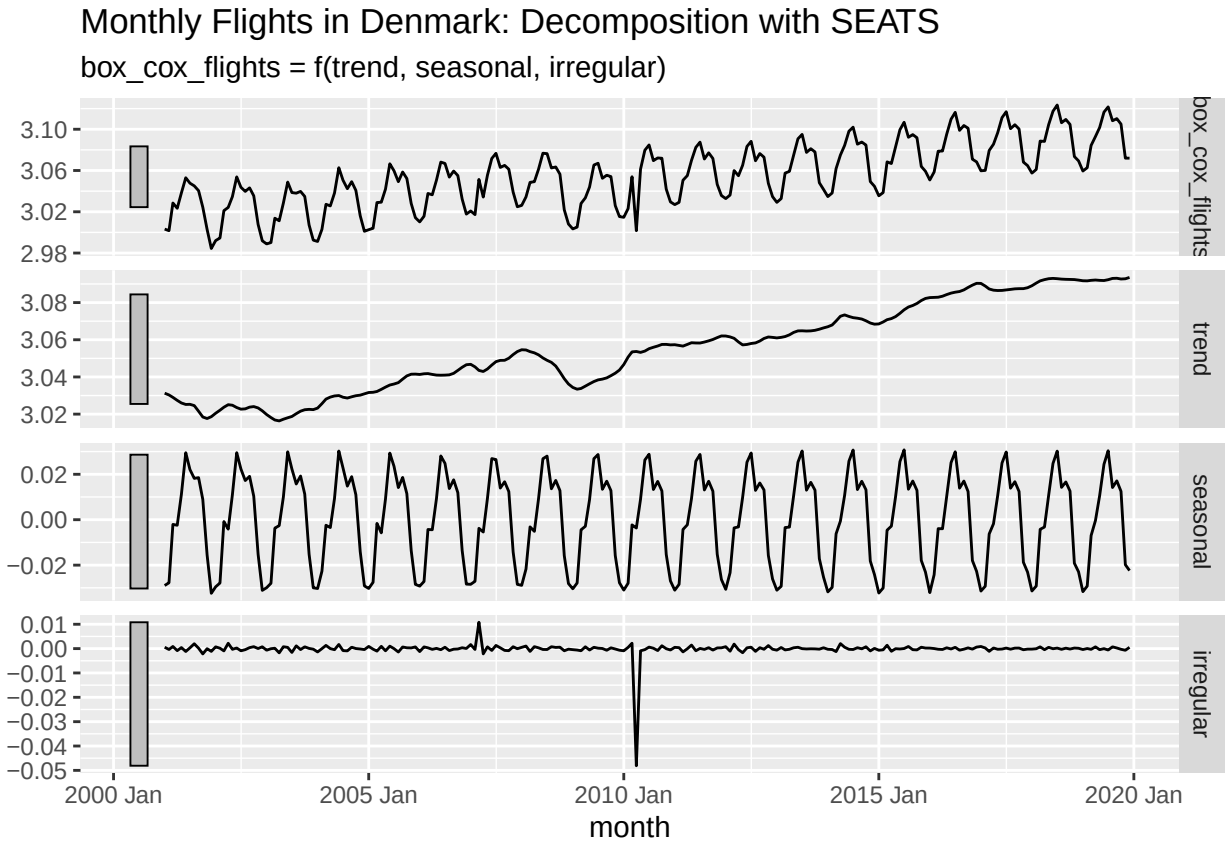

```
flights$flights_log = log(flights$flights)
```

```
flights %>%
  autoplot(flights_log)
```

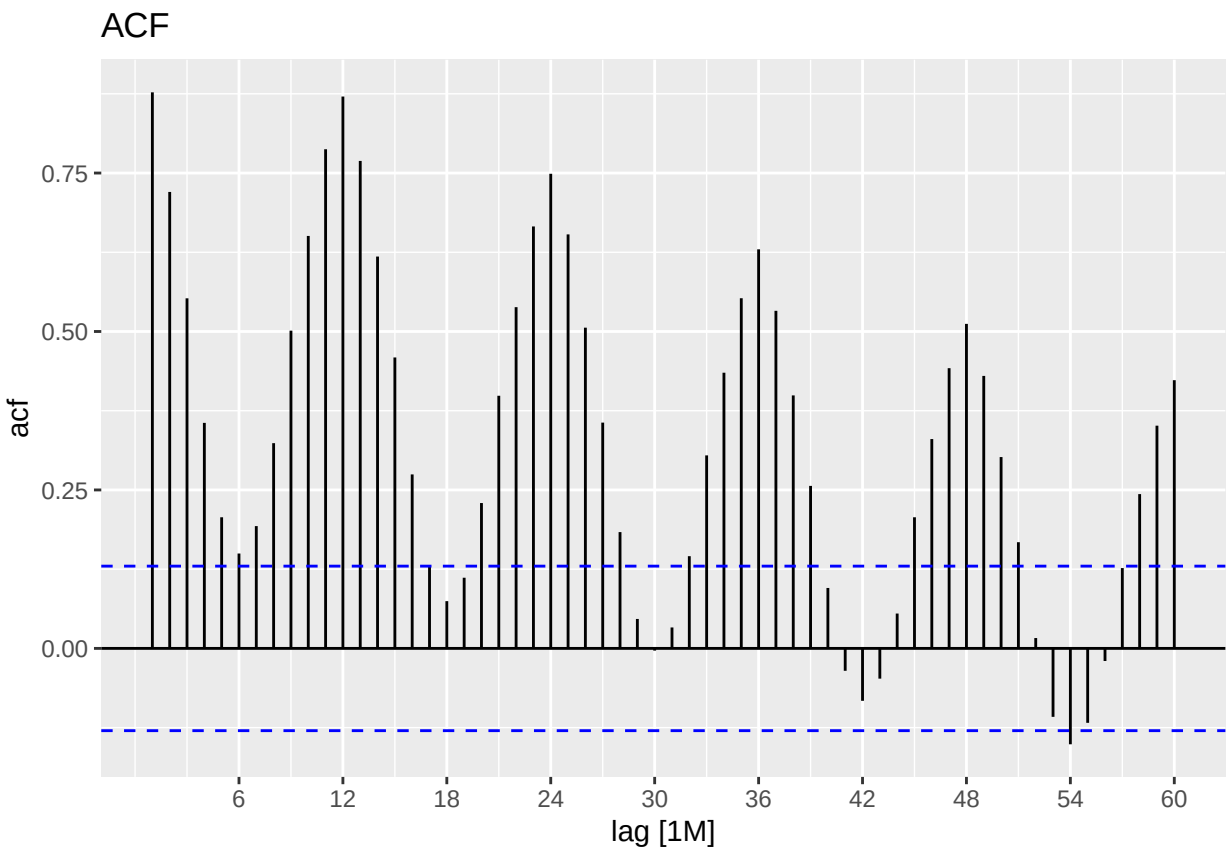


SEATS decomposition and ACF on Box-Cox transformed data

```
seats_dcmp <- flights%>%
model(seats = X_13ARIMA_SEATS(box_cox_flights ~ seats())) %>%
components()

plot_seats <- autoplot(seats_dcmp) +
labs(title = "Monthly Flights in Denmark: Decomposition with SEATS")
print(plot_seats)
```




```
flights %>% ACF(box_cox_flights, lag_max = 60) %>% autoplot() + labs(title = "ACF")
```

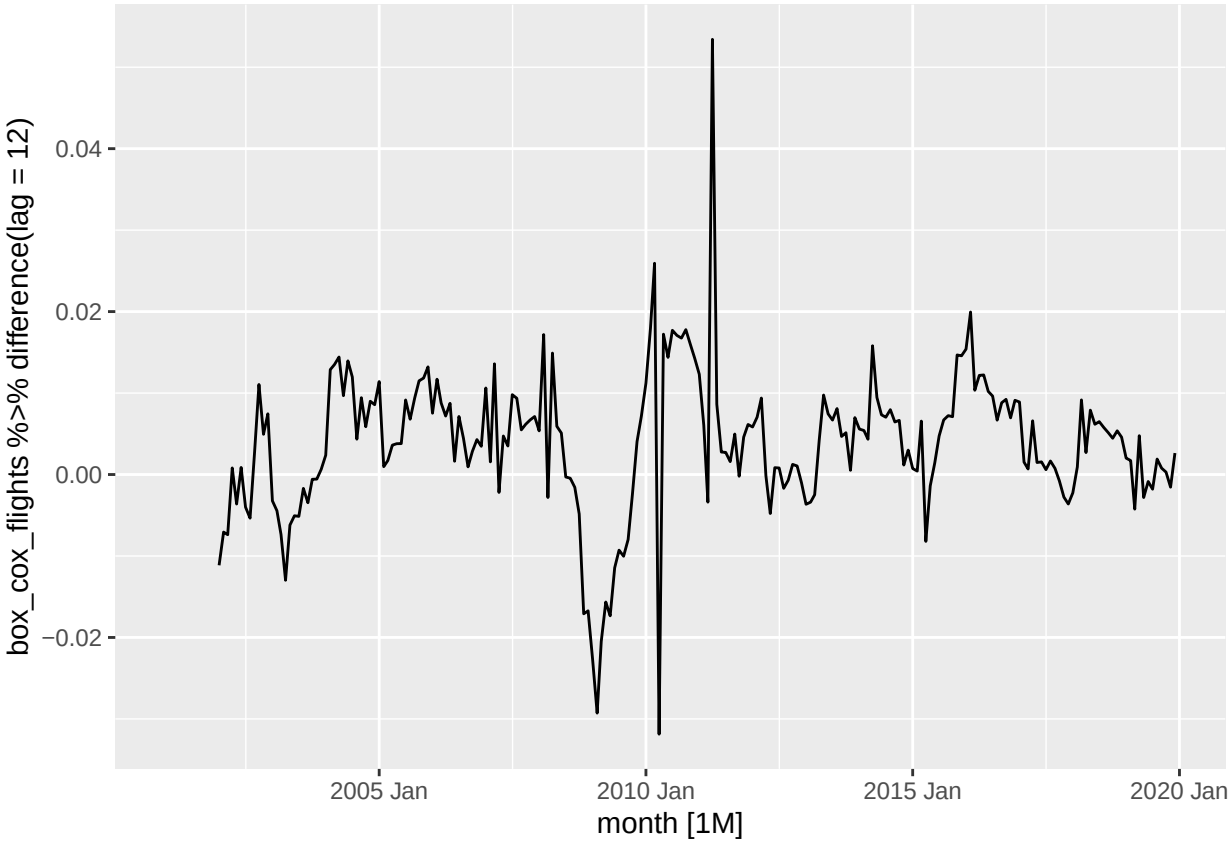


Check for stationarity

```
flights %>% autoplot(box_cox_flights %>% difference(lag = 12))
```

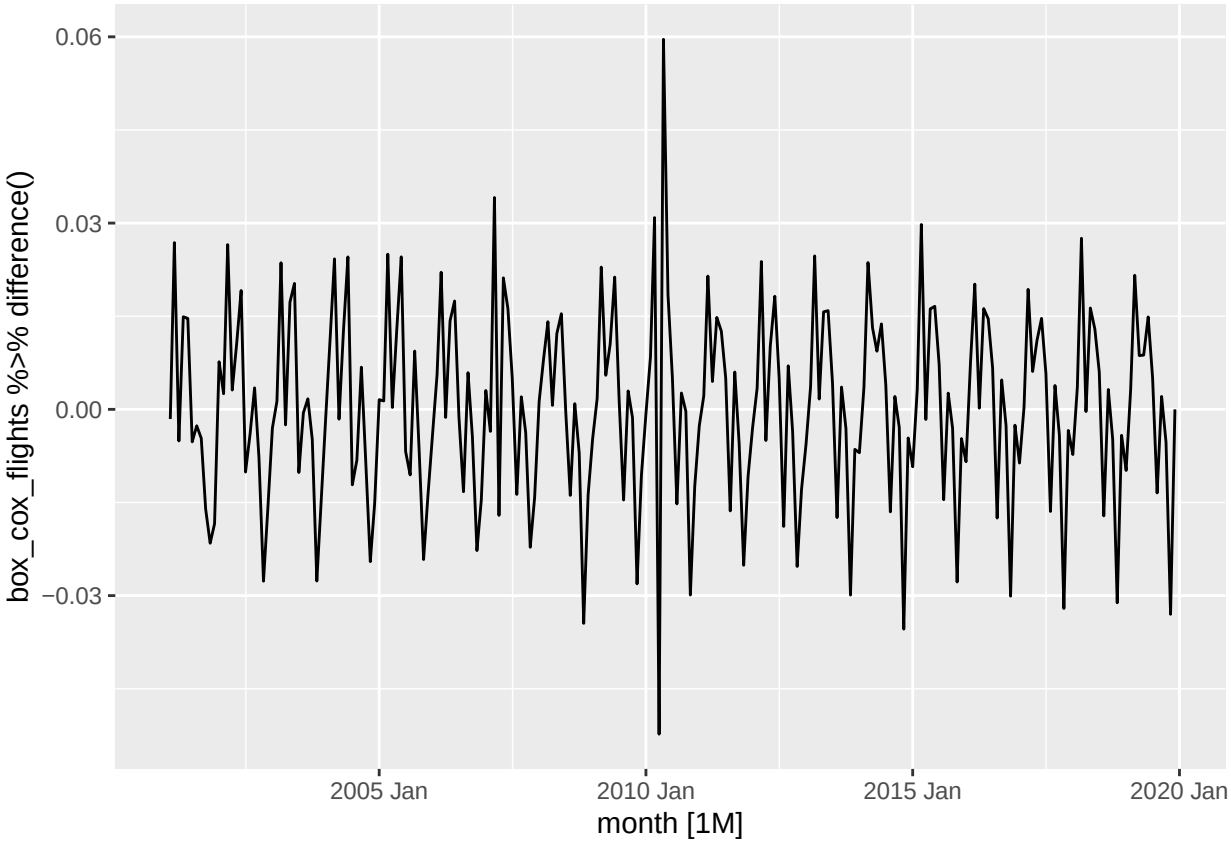
First, check for stationarity in plots

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



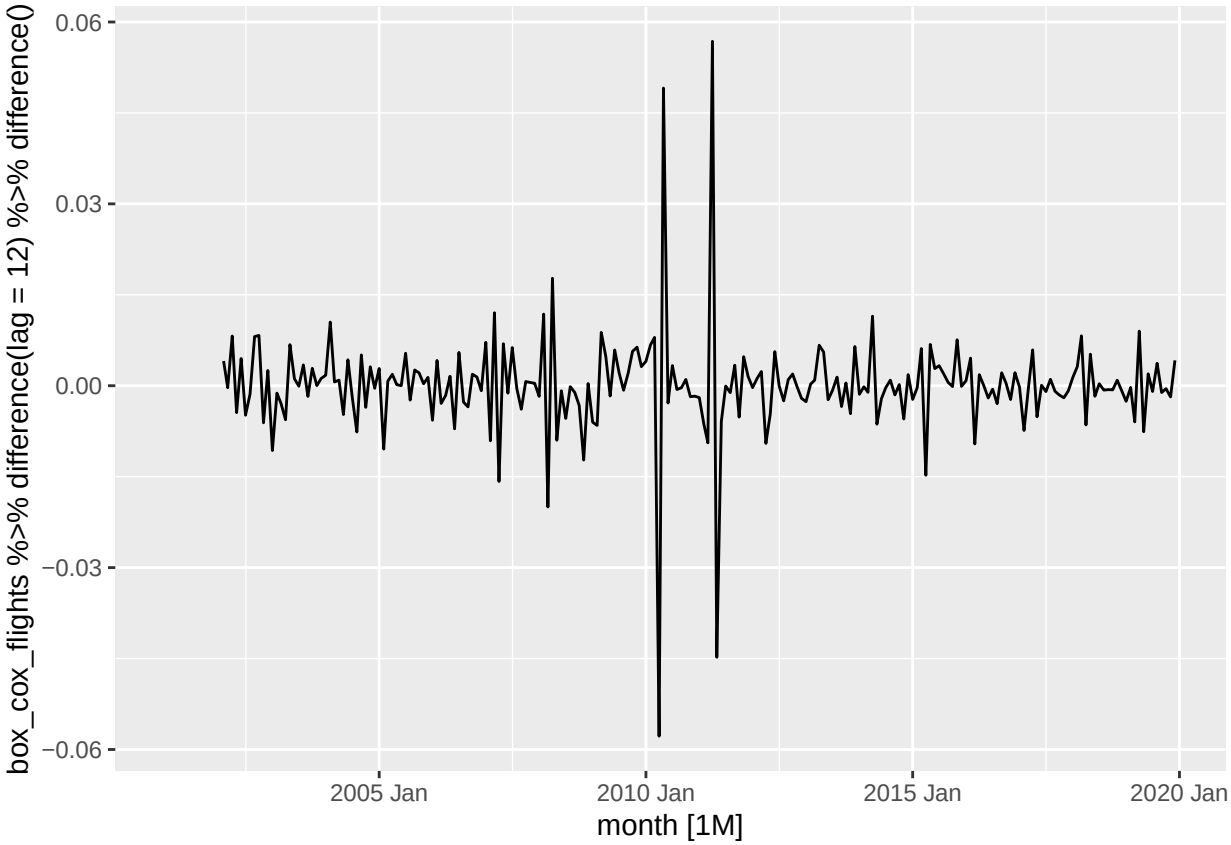

```
flights %>% autoplot(box_cox_flights %>% difference())
```

```
## Warning: Removed 1 row containing missing values (‘geom_line()‘).
```




```
flights %>% autoplot(box_cox_flights %>% difference(lag = 12) %>% difference())
```

```
## Warning: Removed 13 rows containing missing values (‘geom_line()‘).
```



After the application of first differences, the data still shows non-stationarity due to the presence of pronounced seasonality. However, after the application of both first differences and seasonal differences, the data appears to be stationary and has the characteristics of white noise. To validate these assumptions, ADF and KPSS tests are performed.

ADF - no difference

```
summary(ur.df(flights$box_cox_flights, type = "trend", selectlags = c("AIC"), lags = 12))
```

ADF with trend to test for a deterministic trend, drift and unit root:

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.048731 -0.002750 -0.000246  0.003248  0.037144
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.635e-01  3.088e-01   2.148 0.032877 *
## z.lag.1      -2.198e-01  1.025e-01  -2.144 0.033244 *
## tt            8.083e-05  3.737e-05   2.163 0.031741 *
## z.diff.lag1  -3.020e-01  1.147e-01  -2.634 0.009099 **
## z.diff.lag2  -2.514e-01  1.105e-01  -2.274 0.024000 *
## z.diff.lag3  -1.807e-01  1.044e-01  -1.731 0.085026 .
## z.diff.lag4  -2.080e-01  9.897e-02  -2.101 0.036877 *
## z.diff.lag5  -2.952e-01  9.171e-02  -3.219 0.001503 **
## z.diff.lag6  -3.790e-01  8.475e-02  -4.472 1.30e-05 ***
## z.diff.lag7  -3.958e-01  7.902e-02  -5.009 1.20e-06 ***
## z.diff.lag8  -3.932e-01  7.614e-02  -5.165 5.81e-07 ***
## z.diff.lag9  -2.798e-01  7.546e-02  -3.708 0.000271 ***
## z.diff.lag10 -3.521e-01  7.226e-02  -4.872 2.24e-06 ***
## z.diff.lag11 -2.934e-01  7.018e-02  -4.181 4.34e-05 ***
## z.diff.lag12  3.824e-01  6.401e-02   5.973 1.05e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007088 on 200 degrees of freedom
## Multiple R-squared:  0.7985, Adjusted R-squared:  0.7844
## F-statistic:  56.6 on 14 and 200 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -2.144 5.031 2.3485
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

Data may exhibit a deterministic trend or drift, but not a deterministic trend with drift.

tau3: -2.144 > -3.13 (t > cv), do not reject the null hypothesis of a unit root (non-stationarity).

phi2: 5.031 > 4.07 (t > cv), reject the null hypothesis of no deterministic trend and no drift, indicating the presence of a deterministic trend or drift in the data.

phi3: 2.3485 < 5.47 (t < cv), do not reject the null hypothesis of a unit root and no trend.

In summary -> we cannot reject the null hypothesis of non-stationarity, suggesting that our data might be non-stationary. The tau3 test does not reject the null hypothesis of a unit root, while the phi3 test does not reject the null hypothesis of no deterministic trend with drift. However, the phi2 test rejects the null hypothesis of no deterministic trend and no drift, indicating evidence of a deterministic trend or drift in the data. For this reason, we can conclude that the data may exhibit a deterministic trend or drift, but not a deterministic trend with drift. For this reason, we move on to test for only drift and unit root.

```
summary(ur.df(flights$box_cox_flights, type = "drift",selectlags = c("AIC"), lags = 12))
```

ADF with drift to test for drift and unit root:

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.046699 -0.003425 -0.000374  0.003264  0.037944
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.010796   0.066159   0.163    0.871
## z.lag.1      -0.002979   0.021663  -0.138    0.891
## z.diff.lag1  -0.500843   0.069191  -7.239 9.41e-12 ***
## z.diff.lag2  -0.430562   0.073789  -5.835 2.13e-08 ***
## z.diff.lag3  -0.341483   0.073971  -4.616 6.95e-06 ***
## z.diff.lag4  -0.353525   0.073224  -4.828 2.73e-06 ***
## z.diff.lag5  -0.424530   0.070153  -6.051 6.92e-09 ***
## z.diff.lag6  -0.490302   0.067973  -7.213 1.09e-11 ***
## z.diff.lag7  -0.486737   0.067515  -7.209 1.12e-11 ***
## z.diff.lag8  -0.464719   0.069217  -6.714 1.90e-10 ***
## z.diff.lag9  -0.333308   0.071942  -4.633 6.47e-06 ***
## z.diff.lag10 -0.390158   0.070721  -5.517 1.05e-07 ***
## z.diff.lag11 -0.315945   0.070031  -4.511 1.09e-05 ***
## z.diff.lag12  0.376136   0.064531   5.829 2.19e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007152 on 201 degrees of freedom
## Multiple R-squared:  0.7938, Adjusted R-squared:  0.7804
## F-statistic: 59.51 on 13 and 201 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -0.1375 5.114
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

There may be evidence of a drift term.

tau2: -0.1375 > -2.57 (t > cv), do not reject the null hypothesis of a unit root (non-stationarity).

phi1: 5.114 > 3.81 (t > cv), reject the null hypothesis of no drift, indicating the presence of drift in the data.

In summary -> in this case, we are unable to reject the null hypothesis of non-stationarity for tau2, which supports the previous test that suggested our data is non-stationary. We can also reject the null hypothesis of no drift for phi1, indicating that there is evidence of a drift term in the data. Despite the presence of drift, there might still be a unit root in the data. For this reason, we should consider testing for only the unit root.

```
summary(ur.df(flights$box_cox_flights, type = "none",selectlags = c("AIC"), lags = 12))
```

ADF test to test for unit root:

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.046645 -0.003338 -0.000299  0.003218  0.037944
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## z.lag.1       0.0005562  0.0001737   3.202  0.00159 **
## z.diff.lag1  -0.5046208  0.0650464  -7.758 4.19e-13 ***
## z.diff.lag2  -0.4342681  0.0700371  -6.201 3.12e-09 ***
## z.diff.lag3  -0.3449566  0.0706705  -4.881 2.14e-06 ***
## z.diff.lag4  -0.3567777  0.0702891  -5.076 8.73e-07 ***
## z.diff.lag5  -0.4274585  0.0676549  -6.318 1.66e-09 ***
## z.diff.lag6  -0.4928579  0.0659832  -7.469 2.38e-12 ***
## z.diff.lag7  -0.4889405  0.0659904  -7.409 3.40e-12 ***
## z.diff.lag8  -0.4666444  0.0680394  -6.858 8.31e-11 ***
## z.diff.lag9  -0.3349994  0.0710195  -4.717 4.46e-06 ***
## z.diff.lag10 -0.3916201  0.0699814  -5.596 7.08e-08 ***
## z.diff.lag11 -0.3170304  0.0695466  -4.559 8.91e-06 ***
## z.diff.lag12  0.3753698  0.0642045   5.846 1.99e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007135 on 202 degrees of freedom
## Multiple R-squared:  0.7939, Adjusted R-squared:  0.7806
## F-statistic: 59.84 on 13 and 202 DF,  p-value: < 2.2e-16
##
##
```

```
## Value of test-statistic is: 3.2017
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

The series most likely has a unit root and is non-stationary (stochastic trend).

tau1: 3.2017 > -1.62 (t > cv), we do not reject the null hypothesis of a unit root (non-stationarity).

A summary of the findings from ADF tests:

Unit root: The series has a unit root, which implies that the data is non-stationary.

Non-stationary: The results indicate that the data is non-stationary as we cannot reject the null hypothesis of non-stationarity in any of the tests.

Deterministic trend: The phi2 test result rejects the null hypothesis of no deterministic trend and no drift, which indicates that there might be a deterministic trend or drift in the data.

Stochastic trend: Since the data is non-stationary and has a unit root, it is likely to have a stochastic trend.

Drift: The phi1 test result rejects the null hypothesis of no drift, suggesting that there may be evidence of a drift term in the data.

In summary -> the ADF tests indicate that the data is non-stationary and has a unit root, which implies the presence of a stochastic trend. There may be a deterministic trend or drift in the data, but not a deterministic trend with drift. The presence of drift is also suggested by the test results.

KPSS - no difference

From here, considering the results obtained from the ADF tests, I will perform the KPSS tests to further investigate the stationary, deterministic trend, and drift in the data. Since my data suggests the presence of a deterministic trend or drift, it makes sense to perform both KPSS tests with trend and drift, and one with only drift.

```
summary(ur.kpss(flights$box_cox_flights, type = "tau"))  # Trend + drift
```

KPSS test with trend and drift:

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: tau with 4 lags.
##
## Value of test-statistic is: 0.0376
##
```

```
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

This implies that the series does not have a unit root and is stationary.

t: 0.0376 < 0.119 (t < cv), we do not reject the null hypothesis of stationarity.

In summary -> this means that we cannot reject the null hypothesis of stationarity, and we cannot conclude that the time series has a unit root or a deterministic trend. This implies that the series does not have a unit root and is stationary.

```
summary(ur.kpss(flights$box_cox_flights, type = "mu")) # Drift
```

KPSS test with drift:

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 3.2009
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

This implies that the series have a unit root and is non-stationary.

t: 3.2009 > 0.347 (t > cv), we reject the null hypothesis of no drift, indicating the presence of drift in the data.

In summary -> this means that we cannot accept the null hypothesis of stationarity, and we can conclude that the time series has a unit root or a deterministic trend. This implies that the series is non-stationary.

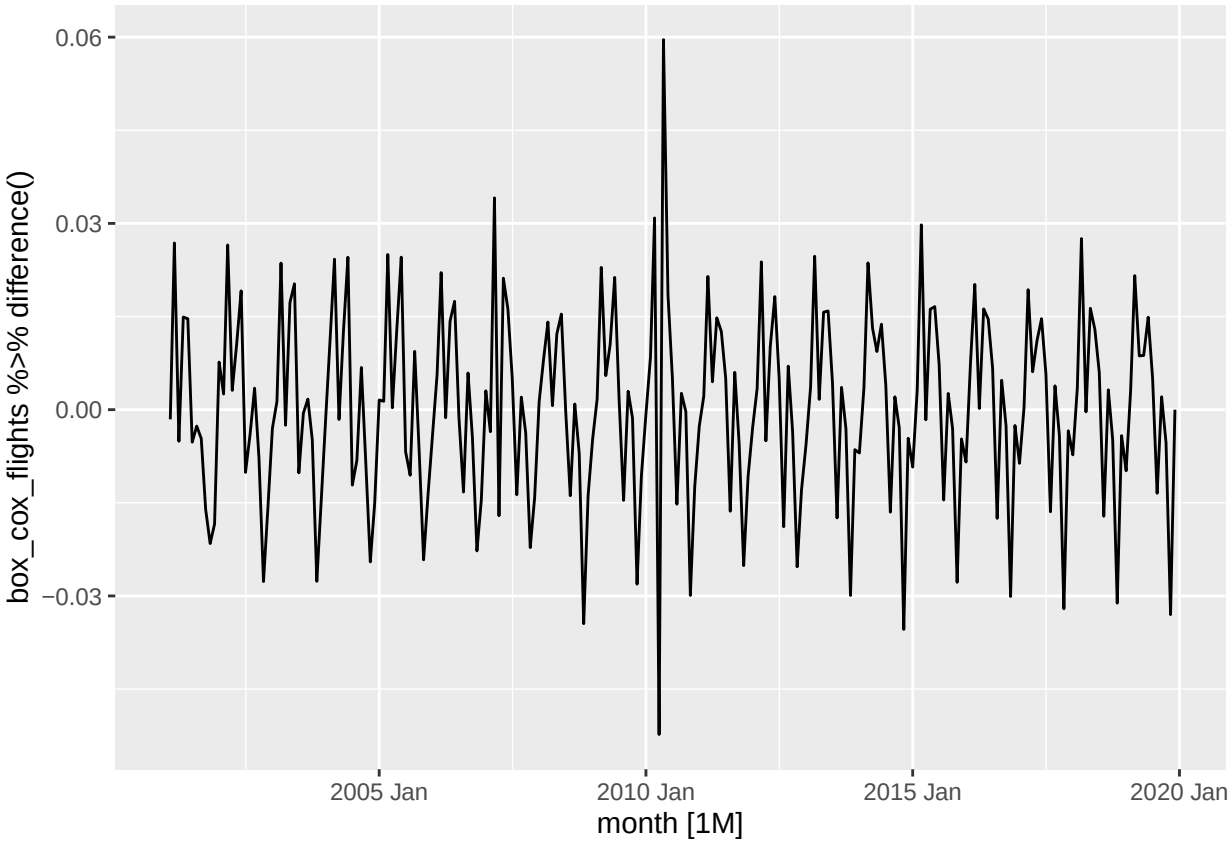
Based on the test results, they indicate that the time series is non-stationary. As a result, differentiation should be used, and the ADF and KPSS tests should be performed again.

Taking 12 month difference and start from the top again:

```
flights %>% autoplot(box_cox_flights %>% difference())
```

First, check for stationarity in plots, take the first difference to try to remove the trending pattern

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```



There isn't a visible trend at the moment. Let's proceed. We compute the ACF and PACF plots for the series to better understand the seasonal pattern; they could indicate whether we should take a seasonal difference in addition to a regular second difference.

```
summary(ur.df(diff(log(flights$flights), lag=12), type = "trend", selectlags = "AIC", lags=12))
```

ADF with first difference (trend):

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.34541 -0.01756 -0.00085  0.01892  0.18583
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.912e-03  7.025e-03   1.269  0.20614
## z.lag.1      -4.130e-01  8.987e-02  -4.596 7.90e-06 ***
## tt            2.981e-05  5.420e-05   0.550  0.58294
## z.diff.lag1  -2.362e-01  9.090e-02  -2.598  0.01011 *
## z.diff.lag2  -1.164e-02  9.236e-02  -0.126  0.89988
## z.diff.lag3   1.114e-01  9.113e-02   1.222  0.22328
## z.diff.lag4   1.990e-01  8.948e-02   2.223  0.02738 *
## z.diff.lag5   2.012e-01  8.899e-02   2.261  0.02493 *
## z.diff.lag6   1.669e-01  8.873e-02   1.881  0.06154 .
## z.diff.lag7   1.912e-01  8.798e-02   2.174  0.03099 *
## z.diff.lag8   2.314e-01  8.764e-02   2.640  0.00898 **
## z.diff.lag9   2.032e-01  8.763e-02   2.318  0.02150 *
## z.diff.lag10  2.082e-01  8.548e-02   2.435  0.01581 *
## z.diff.lag11  3.321e-01  8.103e-02   4.099 6.17e-05 ***
## z.diff.lag12 -1.592e-01  7.076e-02  -2.249  0.02566 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04441 on 188 degrees of freedom
## Multiple R-squared:  0.5145, Adjusted R-squared:  0.4784
## F-statistic: 14.23 on 14 and 188 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -4.5955 7.0605 10.5907
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

The series may have a unit root (non-stationarity) and there is evidence of drift.

tau3: -4.5955 > -3.13 (t > cv) -> do not reject the null hypothesis of a unit root (non-stationarity).

phi2: 7.0605 > 4.07 (t > cv) -> reject the null hypothesis of no drift, indicating the presence of drift in the data.

phi3: 10.5907 > 5.47 (t > cv) -> do not reject the null hypothesis of no linear trend.

In summary -> suggests that the series may have a unit root (non-stationarity) and there is evidence of drift. However, there is no evidence of a linear trend in the data.

```
summary(ur.df(diff(log(flights$flights), lag=12), type = "drift", selectlags = "AIC", lags = 12))
```

ADF with first difference (drift):

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34529 -0.01650 -0.00040  0.01908  0.18561
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.01206    0.00407   2.962  0.00344 **
## z.lag.1      -0.40445    0.08835  -4.578  8.5e-06 ***
## z.diff.lag1  -0.24212    0.09009  -2.688  0.00784 **
## z.diff.lag2  -0.01722    0.09163  -0.188  0.85111
## z.diff.lag3   0.10611    0.09047   1.173  0.24230
## z.diff.lag4   0.19343    0.08875   2.179  0.03053 *
## z.diff.lag5   0.19489    0.08809   2.212  0.02814 *
## z.diff.lag6   0.16042    0.08779   1.827  0.06922 .
## z.diff.lag7   0.18498    0.08708   2.124  0.03496 *
## z.diff.lag8   0.22530    0.08678   2.596  0.01016 *
## z.diff.lag9   0.19691    0.08673   2.270  0.02431 *
## z.diff.lag10  0.20221    0.08463   2.389  0.01786 *
## z.diff.lag11  0.32697    0.08033   4.070  6.9e-05 ***
## z.diff.lag12 -0.16293    0.07030  -2.318  0.02155 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04433 on 189 degrees of freedom
## Multiple R-squared:  0.5138, Adjusted R-squared:  0.4803
## F-statistic: 15.36 on 13 and 189 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -4.5778 10.4782
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

The series may have a unit root (non-stationarity) and there is evidence of drift.

tau2: -4.5778 > -2.57 (t > cv) -> not reject the null hypothesis of a unit root (non-stationarity).

phi1: 10.4782 > 3.81 (t > cv) -> reject the null hypothesis of no drift, indicating the presence of drift in the data.

In summary -> suggests that the series may have a unit root (non-stationarity) and there is evidence of

drift.

```
summary(ur.df(diff(log(flights$flights), lag=12), type = "none", selectlags = "AIC", lags = 12))
```

ADF with first difference (unit root):

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33448 -0.00894  0.00629  0.02707  0.18780
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1      -0.23579    0.06893  -3.421 0.000765 ***
## z.diff.lag1  -0.36183    0.08215  -4.405 1.77e-05 ***
## z.diff.lag2  -0.12219    0.08621  -1.417 0.158017
## z.diff.lag3   0.01130    0.08633   0.131 0.896004
## z.diff.lag4   0.10613    0.08541   1.243 0.215550
## z.diff.lag5   0.10949    0.08493   1.289 0.198891
## z.diff.lag6   0.07636    0.08476   0.901 0.368777
## z.diff.lag7   0.10359    0.08431   1.229 0.220722
## z.diff.lag8   0.14596    0.08421   1.733 0.084675 .
## z.diff.lag9   0.11777    0.08419   1.399 0.163465
## z.diff.lag10  0.13032    0.08272   1.575 0.116825
## z.diff.lag11  0.26587    0.07921   3.356 0.000954 ***
## z.diff.lag12 -0.21034    0.06984  -3.012 0.002952 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04523 on 190 degrees of freedom
## Multiple R-squared:  0.4912, Adjusted R-squared:  0.4564
## F-statistic: 14.11 on 13 and 190 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -3.4207
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

The series may have a unit root (non-stationarity) and there is evidence of drift.

tau1: -3.4207 > -1.62 (t > cv) -> do not reject the null hypothesis of a unit root (non-stationarity).

In summary -> suggests that the series may have a unit root (non-stationarity) and there is evidence of drift.

```
summary(ur.kpss(diff(log(flights$flights), lag=12), type="tau")) # trend + drift
```

KPSS with first difference (trend + drift):

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: tau with 4 lags.
##
## Value of test-statistic is: 0.078
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

Data is stationary!

t: 0.078 < 0.119 (t < cv) -> cannot reject the null hypothesis.

In summary -> result suggests that there is no unit root and the time series is stationary.

```
summary(ur.kpss(diff(log(flights$flights), lag=12), type = "mu")) # drift
```

KPSS with first difference (drift):

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.1464
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Data is stationary!

t: 0.1464 < 0.347 (t < cv) -> cannot reject the null hypothesis.

In summary -> suggests that there is no evidence of a unit root with drift, and the time series may be stationary without a drift component.

Summary of Results from first difference:

The results of the ADF test indicated that the series may have a unit root, suggesting non-stationarity. Additionally, there was evidence of drift in the data. The ADF test with a regression type of "trend" revealed a rejection of the null hypothesis of no drift, indicating the presence of a drift component. Similarly, the ADF test with a regression type of "drift" also indicated the presence of drift in the series. However, no evidence of a linear trend was found.

In contrast, the KPSS tests yielded different outcomes. The KPSS test with a type of "tau" suggested that the time series is stationary, as the test statistic was lower than the critical value at a 10% significance level. Likewise, the KPSS test with a type of "mu" indicated the absence of a unit root with drift, further suggesting stationary behavior.

Let's attempt to combine seasonal and first differences for a smoother outcome.

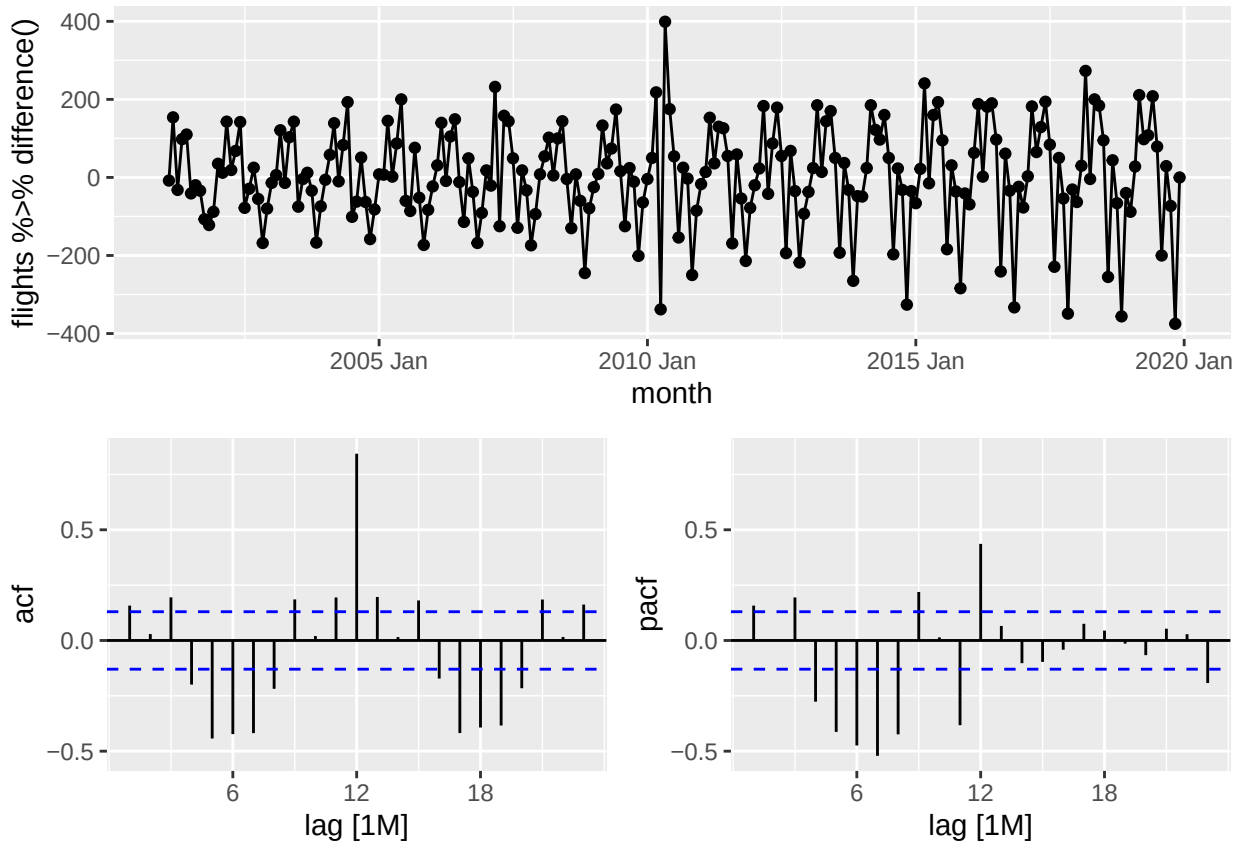
Seasonal and first differences combined.

```
flights %>% gg_tsdisplay(flights %>% difference(), plot_type = "partial")
```

ACF plot of the differenced data to see if we can spot any seasonal pattern:

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

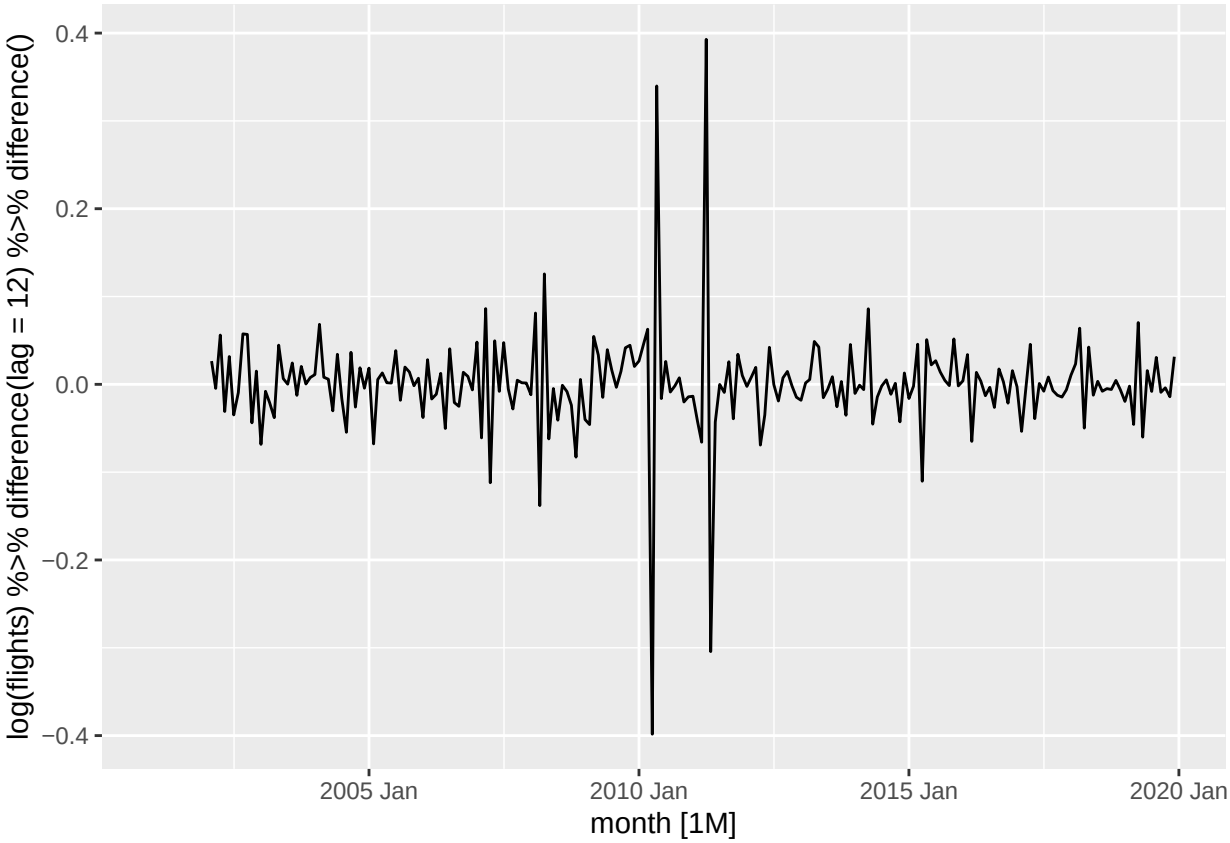


The very high and regular spikes around lag 6, 12, 18 suggest that taking the seasonal difference with lag = 12 would make sense.

```
flights %>% autoplot(log(flights) %>% difference(lag = 12) %>% difference())
```

Plotting seasonal and first differences combined:

```
## Warning: Removed 13 rows containing missing values (`geom_line()`).
```



The resulting plot exhibits a more consistent variation, let's run the ADF and KPPS tests to see if the series is now stationary.

```
summary(ur.df(diff(diff(log(flights$flights),lag=12)), type = "trend", lags = 12))
```

ADF with seasonal and first difference (trend):

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.309177 -0.016862 -0.000217  0.018978  0.151318
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   3.809e-03  6.772e-03   0.562 0.574470
## z.lag.1      -2.723e+00  4.429e-01  -6.149 4.61e-09 ***
## tt           -2.865e-05  5.308e-05  -0.540 0.590025
## z.diff.lag1   1.092e+00  4.264e-01   2.560 0.011266 *
## z.diff.lag2   8.828e-01  4.103e-01   2.152 0.032692 *
## z.diff.lag3   7.731e-01  3.884e-01   1.991 0.047989 *
## z.diff.lag4   7.656e-01  3.644e-01   2.101 0.036999 *
## z.diff.lag5   7.764e-01  3.395e-01   2.287 0.023321 *
## z.diff.lag6   7.353e-01  3.121e-01   2.356 0.019518 *
## z.diff.lag7   7.113e-01  2.822e-01   2.520 0.012565 *
## z.diff.lag8   7.400e-01  2.497e-01   2.964 0.003435 **
## z.diff.lag9   7.347e-01  2.129e-01   3.451 0.000691 ***
## z.diff.lag10  7.197e-01  1.698e-01   4.239 3.53e-05 ***
## z.diff.lag11  8.057e-01  1.206e-01   6.679 2.68e-10 ***
## z.diff.lag12  3.484e-01  6.736e-02   5.173 5.91e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04384 on 187 degrees of freedom
## Multiple R-squared:  0.8445, Adjusted R-squared:  0.8329
## F-statistic: 72.57 on 14 and 187 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -6.1494 12.6142 18.9109
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

Data is stationary!

tau3: -6.1494 < -3.13 (t < cv), reject the null hypothesis of a unit root, the transformed data is stationary.

phi2: 12.6142 > 4.07 (t > cv), do not reject the null hypothesis of no trend or drift, evidence of a deterministic trend or drift.

phi3: 18.9109 > 5.47 (t > cv), do not reject the null hypothesis of no trend or drift, is evidence of a deterministic trend or drift.

In summary -> indicates that the data is stationary, but may exhibit a deterministic trend or drift.

```
summary(ur.df(diff(diff(log(flights$flights), lag=12)), type = "drift", selectlags = "AIC", lags = 12))
```

ADF with seasonal and first difference (drift):

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
```

```
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30918 -0.01778 -0.00113  0.01833  0.15139
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0005555  0.0030800   0.180 0.857068
## z.lag.1      -2.7055950  0.4408097  -6.138 4.85e-09 ***
## z.diff.lag1   1.0744701  0.4244420   2.531 0.012177 *
## z.diff.lag2   0.8661080  0.4083167   2.121 0.035218 *
## z.diff.lag3   0.7569503  0.3865192   1.958 0.051665 .
## z.diff.lag4   0.7500446  0.3626079   2.068 0.039964 *
## z.diff.lag5   0.7621314  0.3378388   2.256 0.025229 *
## z.diff.lag6   0.7229414  0.3107059   2.327 0.021043 *
## z.diff.lag7   0.7009861  0.2810565   2.494 0.013490 *
## z.diff.lag8   0.7315430  0.2487246   2.941 0.003681 **
## z.diff.lag9   0.7282574  0.2121677   3.432 0.000736 ***
## z.diff.lag10  0.7151115  0.1692574   4.225 3.72e-05 ***
## z.diff.lag11  0.8029730  0.1203068   6.674 2.71e-10 ***
## z.diff.lag12  0.3471429  0.0671892   5.167 6.05e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04376 on 188 degrees of freedom
## Multiple R-squared:  0.8443, Adjusted R-squared:  0.8335
## F-statistic: 78.42 on 13 and 188 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -6.1378 18.8467
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

Data is stationary!

tau2: -6.1378 < -2.57 (t < cv), reject the null hypothesis, indicating stationarity.

phi1: 18.8467 > 3.81 (t > cv), not reject null, may exhibit a deterministic trend or drift.

In summary -> suggests that the series is stationary as the test statistic for tau2 is less than the critical value. Therefore, the series can be considered stationary!

```
summary(ur.df(diff(diff(log(flights$flights), lag=12)), type="none", selectlags = "AIC", lags = 12))
```

ADF with seasonal and first difference (unit root):

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.308629 -0.017212 -0.000584  0.018910  0.151946
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1       -2.7051     0.4397  -6.153 4.45e-09 ***
## z.diff.lag1    1.0740     0.4234   2.537 0.011991 *
## z.diff.lag2    0.8655     0.4073   2.125 0.034876 *
## z.diff.lag3    0.7561     0.3855   1.961 0.051295 .
## z.diff.lag4    0.7491     0.3616   2.071 0.039686 *
## z.diff.lag5    0.7611     0.3369   2.259 0.025034 *
## z.diff.lag6    0.7219     0.3099   2.330 0.020877 *
## z.diff.lag7    0.7000     0.2803   2.497 0.013366 *
## z.diff.lag8    0.7306     0.2480   2.946 0.003629 **
## z.diff.lag9    0.7274     0.2116   3.438 0.000721 ***
## z.diff.lag10   0.7144     0.1688   4.233 3.60e-05 ***
## z.diff.lag11   0.8024     0.1200   6.689 2.47e-10 ***
## z.diff.lag12   0.3469     0.0670   5.177 5.73e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04365 on 189 degrees of freedom
## Multiple R-squared:  0.8443, Adjusted R-squared:  0.8336
## F-statistic: 78.82 on 13 and 189 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -6.1526
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

Data is stationary!

tau1 -> t < cv -> reject null (data is stationary!).

tau1: -6.1526 < -1.62 (t < cv), we reject the null hypothesis of a unit root (non-stationarity).

In summary -> suggests that the transformed data is stationary (rejecting the null hypothesis of a unit root) without a deterministic trend or drift component.

```
summary(ur.kpss(diff(diff(log(flights$flights), lag=12)), type = "tau"))
```

KPSS with seasonal and first difference (trend + drift):

```
##
## ######################
## # KPSS Unit Root Test #
## ######################
##
## Test is of type: tau with 4 lags.
##
## Value of test-statistic is: 0.0202
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

Data is stationary!

t: 0.0202 < 0.119 (t < cv) -> cannot reject the null hypothesis.

In summary -> suggests that there is no evidence of a unit root and the time series is stationary.

```
summary(ur.kpss(diff(diff(log(flights$flights), lag=12)), type = "mu"))
```

KPSS with seasonal and first difference (drift):

```
##
## ######################
## # KPSS Unit Root Test #
## ######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.0323
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Data is stationary!

t: 0.0323 < 0.347 (t < cv) -> cannot reject the null hypothesis.

In summary -> suggests that there is no evidence of a unit root, indicating stationarity in the time series even after seasonal and first differences with drift are applied.

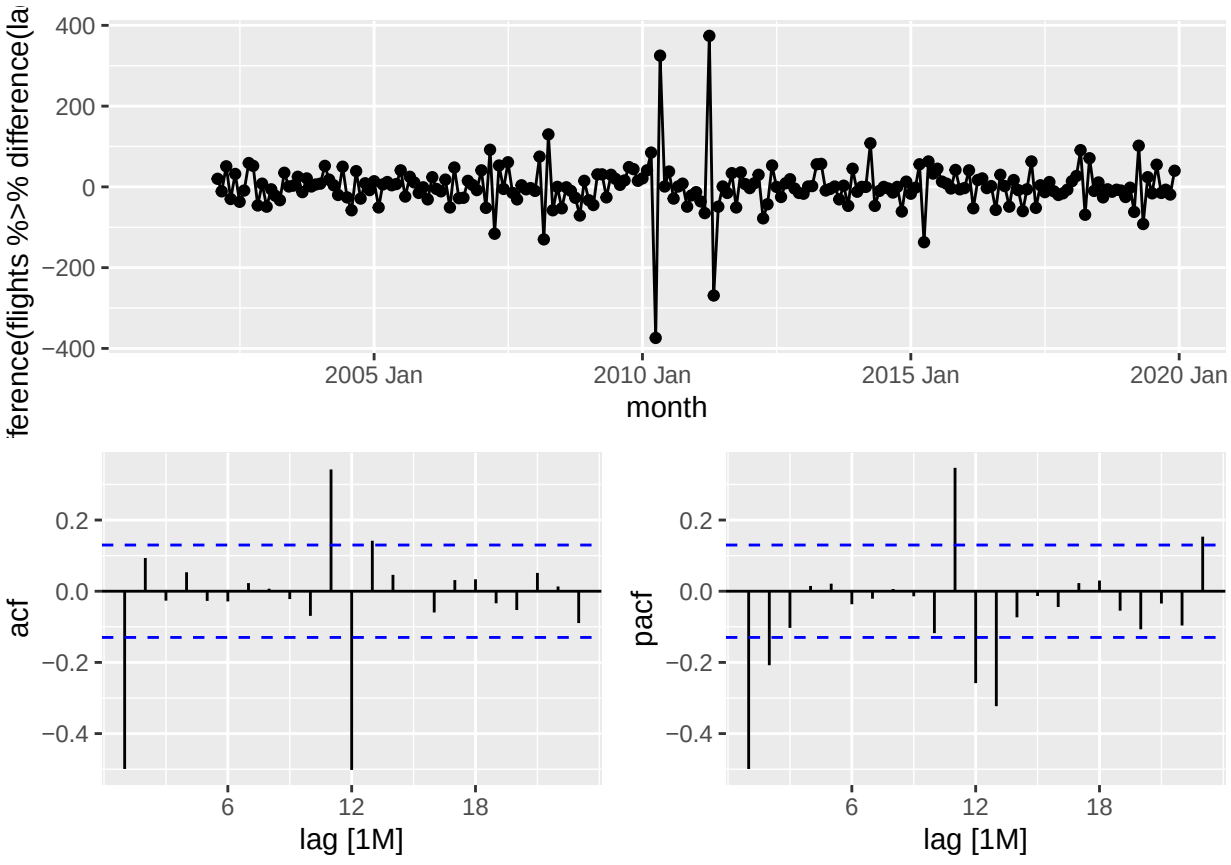
Finally, the difference + seasonal difference with lag = 12 was able to make the data stationary and ready for ARIMA modeling.

As our data is considered stationary, lets have a look at the ACF and PACF plots for the differenced and seasonally differenced data:

```
flights %>%
  gg_tsdisplay(difference(flights %>% difference(lag = 12)), plot_type = "partial")
```

```
## Warning: Removed 13 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```



Based on the provided ACF and PACF plots, here are the interpretations:

Non-seasonal ARIMA component: From ACF, an MA(2) component seems to be required. From PACF, the plot suggests an AR(2) component. PACF typically cuts off after the AR term. Therefore, it might be more appropriate to consider ARIMA(2, 0, 2) as the non-seasonal component.

Seasonal ARIMA component: The ACF and PACF both have significant spikes at lags 11, 12, and 13. This suggests seasonality of order 12. From ACF, there is a significant negative spike at lag 12, suggesting an MA(1) component. From PACF, there are negative spikes at lags 12 and 13, which suggests an AR(2) component.

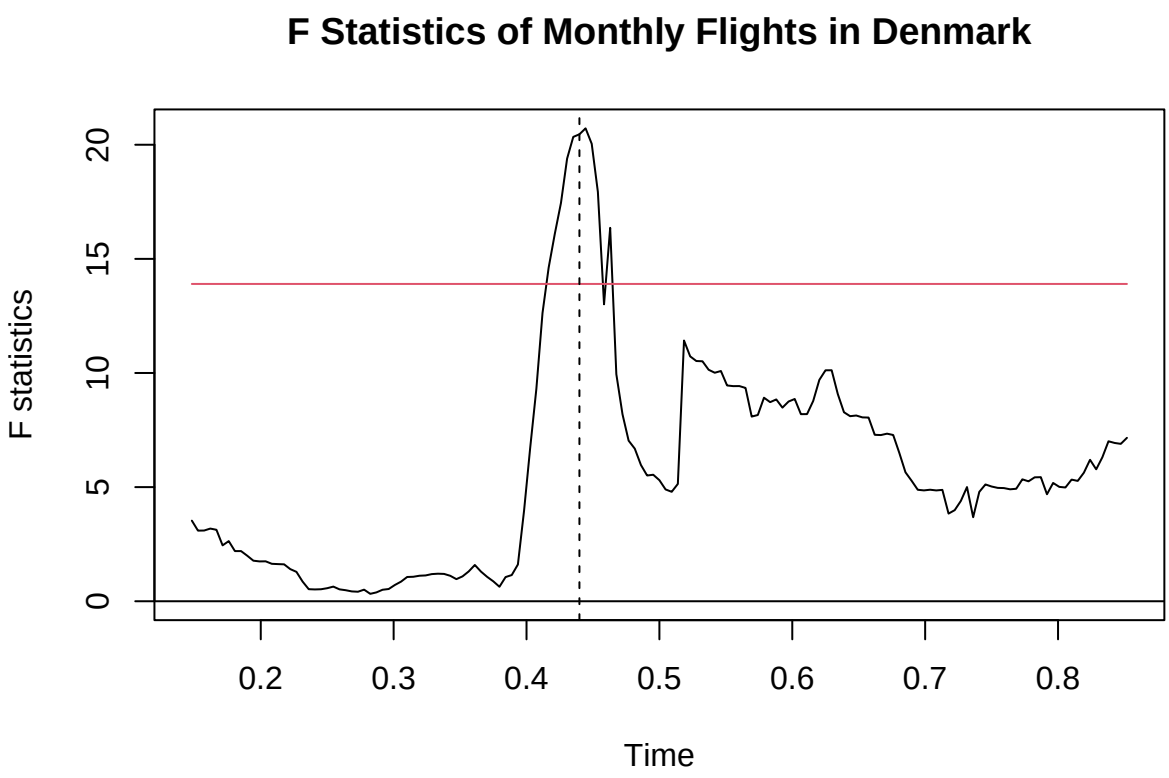
STRUCTURAL BREAKS

QLR Test

```
# First we need to prepare our data for the structural break test
flights <- flights %>%
  mutate(l1.flights = lag(box_cox_flights),
         l12.flights = lag(box_cox_flights, 12))

# QLR test
qlr_flights <- Fstats(box_cox_flights ~ l1.flights + l12.flights, data = as.ts(flights), from = 0.15)
test_flights <- sctest(qlr_flights, type = "supF")
test_flights
```

```
##
##  supF test
##
## data:  qlr_flights
## sup.F = 20.719, p-value = 0.002672
```

```
# Plot it
breakpoints_flights <- breakpoints(qlr_flights, alpha = 0.05)
plot(qlr_flights, alpha = 0.05, main = "F Statistics of Monthly Flights in Denmark")
lines(breakpoints_flights)
```



The QLR test plot shows that the curve goes above the red line, indicating the presence of a structural break in the time series data. The associated p-value of 0.002672 suggests that the null hypothesis of no structural break can be rejected. This implies that there is likely at least one structural break in the data, meaning that the pattern before and after the break is different. Therefore, the Chow test is performed to further detect the presence of a structural break in the data.

Chow test

```
# Identify the breakpoints
breakpoints_flights <- breakpoints(qlr_flights, alpha = 0.05)

# Perform the Chow test
k <- breakpoints_flights$breakpoints[1]
data1 <- data.frame(y = flights$box_cox_flights[1:k], x = flights$l1.flights[1:k], z = flights$l12.flig
data2 <- data.frame(y = flights$box_cox_flights[(k + 1):nrow(flights)],
                    x = flights$l1.flights[(k + 1):nrow(flights)],
                    z = flights$l12.flights[(k + 1):nrow(flights)])

model1 <- lm(y ~ x + z, data = data1)
model2 <- lm(y ~ x + z, data = data2)

residuals_full <- residuals(lm(box_cox_flights ~ l1.flights + l12.flights, data = flights))
residuals_seg1 <- residuals(model1)
residuals_seg2 <- residuals(model2)
```

```r
ssr_full <- sum(residuals_full^2)
ssr_seg1 <- sum(residuals_seg1^2)
ssr_seg2 <- sum(residuals_seg2^2)

chow_test_statistic <- ((ssr_full - (ssr_seg1 + ssr_seg2)) / 2) / ((ssr_seg1 + ssr_seg2) / (nrow(flights
p_value <- 1 - pf(chow_test_statistic, df1 = 2, df2 = nrow(flights) - 4)

# Print the p-value
print(p_value)
```

```
## [1] 0.543486
```

The obtained p-value of 0.543486, it indicates that there is no strong statistical evidence to reject the null hypothesis of no structural break. The p-value is greater than the commonly used significance level of 0.05, suggesting that there is no significant difference in the relationship between the two segments of the data.

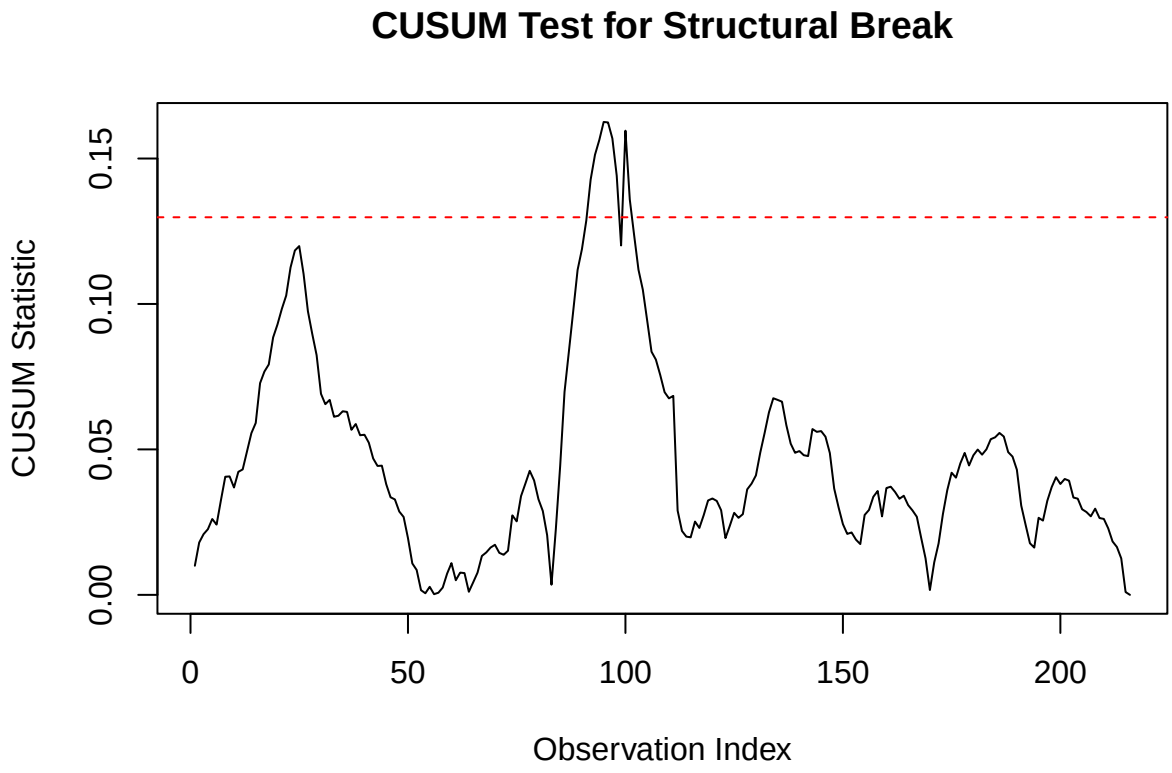
However, the QRL test is more informative and reliable in detecting structural breaks, providing insights into variable relationships and underlying dynamics. The Chow test assesses overall model fit but may miss specific parameter changes. To confirm structural breaks, a cumulative sum test was conducted.

Cumulative sum test

```r
# Fit the linear regression model
model <- lm(box_cox_flights ~ l1.flights + l12.flights, data = flights)

# Calculate the residuals
residuals <- residuals(model)

# Calculate the CUSUM statistic
cusum <- cumsum(residuals)

# Calculate the absolute CUSUM statistic
abs_cusum <- abs(cusum)

# Calculate the critical values for the CUSUM test
h <- nrow(flights)
delta <- 1.96 / sqrt(h)  # Adjust the critical value based on the desired significance level

# Find the index of the potential break point(s)
breakpoints <- which(abs_cusum > delta)

# Check if any breakpoints were found
if (length(breakpoints) > 0) {
  # Print the detected break point(s)
  cat("Potential break point(s) detected at index:", breakpoints, "\n")
} else {
  cat("No structural break detected.\n")
}
```

```
## Potential break point(s) detected at index: 92 93 94 95 96 97 98 100 101
```

```r
# Create the index vector for plotting
index <- 1:length(abs_cusum)

# Plot the CUSUM statistic and critical values
plot(index, abs_cusum, type = "l", main = "CUSUM Test for Structural Break",
     xlab = "Observation Index", ylab = "CUSUM Statistic")
abline(h = delta, col = "red", lty = 2)  # Upper critical value
abline(h = -delta, col = "red", lty = 2)  # Lower critical value
```



The cumulative sum (cusum) analysis detected potential structural breaks at indices 92, 93, 94, 95, 96, 97, 98, 100, and 101. These break points indicate significant shifts or changes in the underlying data patterns. Upon visual inspection of the cusum graph, it is evident that a spike in the cumulative sum crosses the threshold represented by the red line. This suggests a departure from the expected behavior and indicates the presence of structural breaks in the data. Therefore,I will proceed by splitting the data into two subsets, and only use the data after the structural break.

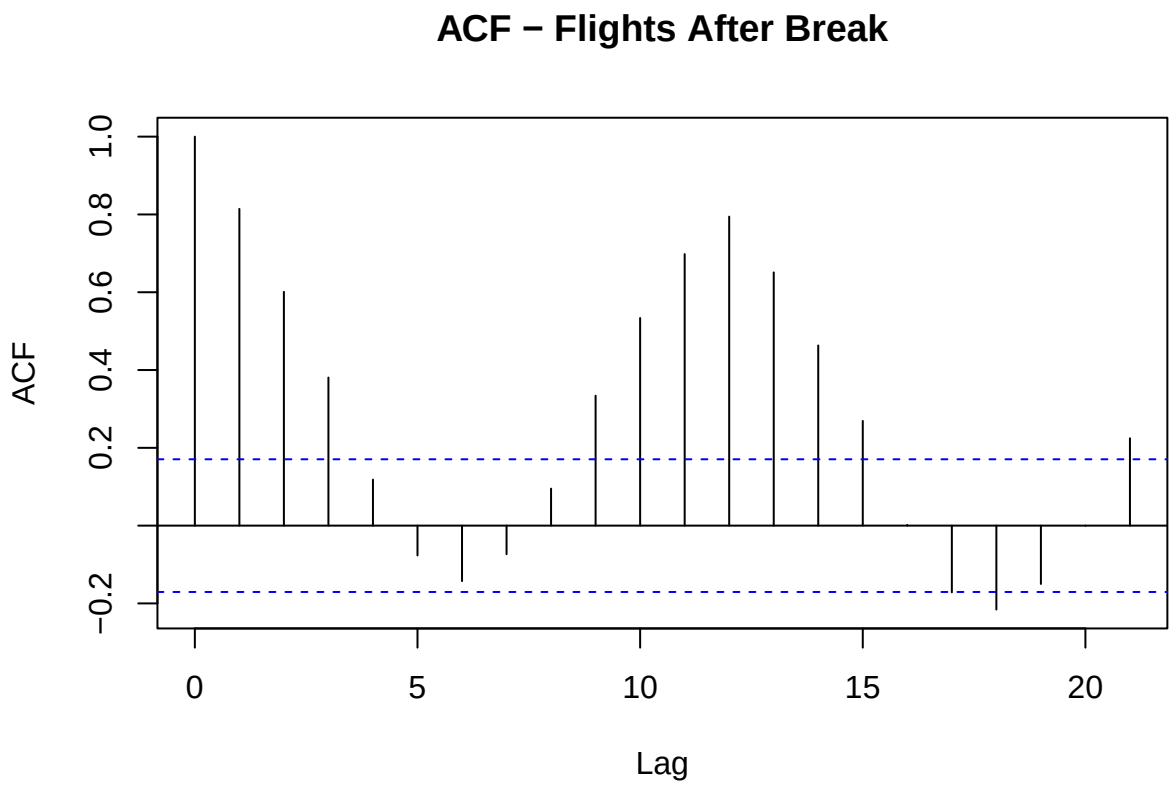
Split the data

```r
# Identify the point of the structural break
breakpoint <- breakpoints_flights$breakpoints[1]

# Split the data at the point of the structural break
flights_before_break <- flights[1:breakpoint,]
flights_after_break <- flights[(breakpoint+1):nrow(flights),]
```

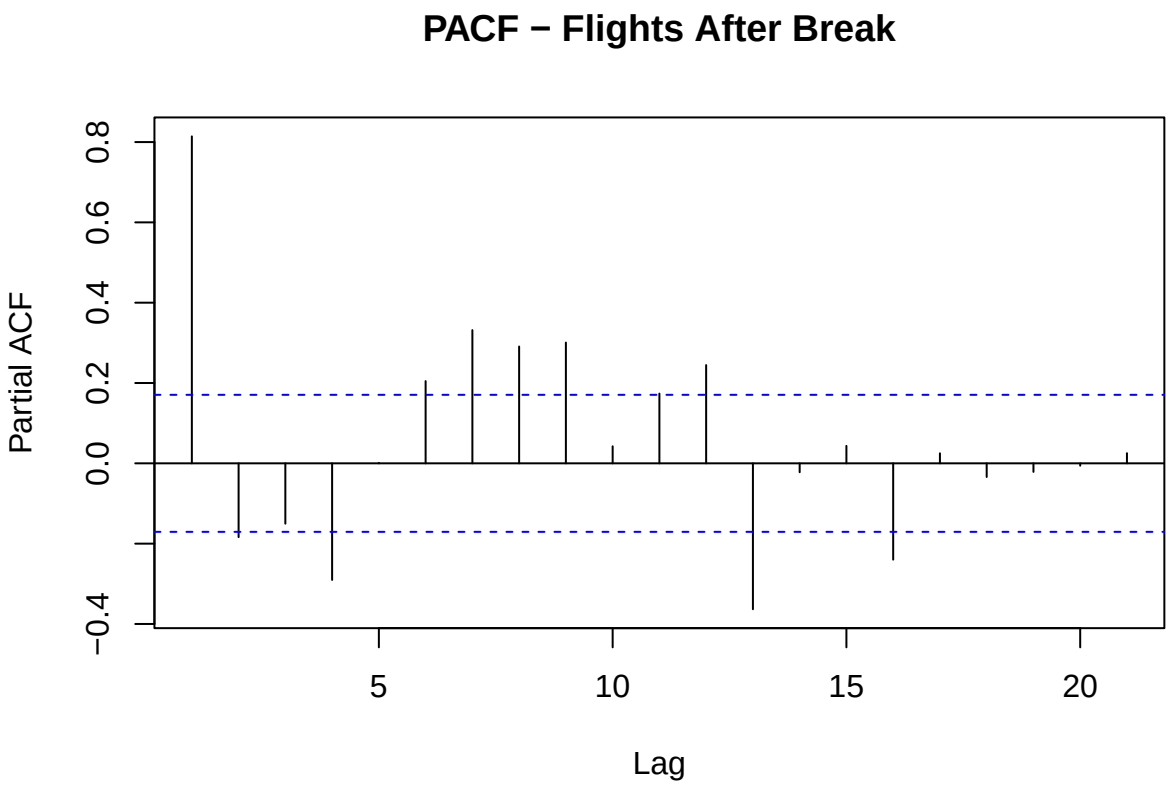
(!) Going further, I will only use the dataset taking place after the structural break. (!)

ACF and PACF plots for flights after structural break:

```
# ACF plot for Flights After Break
acf_after <- acf(flights_after_break$box_cox_flights, main = "ACF - Flights After Break")
```




```
acf_values_after <- acf_after$acf

# PACF plot for Flights After Break
pacf_after <- pacf(flights_after_break$box_cox_flights, main = "PACF - Flights After Break")
```




```
pacf_values_after <- pacf_after$acf
```

ACF: The significant autocorrelation at lag 1 suggests the possibility of an AR(1) model. Additionally, the significant autocorrelation at lag 12 indicates potential seasonal behavior. This suggests the inclusion of a seasonal MA(1) term in the model. Therefore, an ARMA(1,1) model or an AR(1) model with a seasonal MA(1) term could be appropriate for capturing the autocorrelation in the data.

PACF: The significant partial autocorrelation at lag 1, which decreases and becomes non-significant beyond lag 1, suggests the inclusion of an autoregressive term in the model, such as an AR(1) model. However, there are some non-significant spikes at lags 2, 3, 4, 6, and 7, indicating possible residual autocorrelation after removing the effect of the previous lags. This suggests that an AR(1) model or an ARMA(1,0) model may be suitable for capturing the partial autocorrelation in the data.

Considering these observations, it would be reasonable to consider fitting an ARMA(1,1) model or an AR(1) model with a seasonal MA(1) term.

```
train_after_break <- flights_after_break %>% filter(month <= max(month)-12*2)
test_after_break <- flights_after_break %>% filter(month > max(month) - 12 * 2)
train_after_break
```

ARIMA

```
## # A tsibble: 108 x 6 [1M]
```

```
##    flights    month box_cox_flights flights_log l1.flights l12.flights
##      <int>    <mth>           <dbl>       <dbl>      <dbl>       <dbl>
##  1     788 2009 Jan            3.00        6.67       3.01        3.03
##  2     797 2009 Feb            3.01        6.68       3.00        3.03
##  3     930 2009 Mar            3.03        6.84       3.01        3.05
##  4     966 2009 Apr            3.03        6.87       3.03        3.05
##  5    1040 2009 May            3.04        6.95       3.03        3.06
##  6    1214 2009 Jun            3.07        7.10       3.04        3.08
##  7    1230 2009 Jul            3.07        7.11       3.07        3.08
##  8    1105 2009 Aug            3.05        7.01       3.07        3.06
##  9    1129 2009 Sep            3.06        7.03       3.05        3.06
## 10    1118 2009 Oct            3.05        7.02       3.06        3.06
## # ... with 98 more rows
```

```
test_after_break
```

```
## # A tsibble: 24 x 6 [1M]
##    flights    month box_cox_flights flights_log l1.flights l12.flights
##      <int>    <mth>           <dbl>       <dbl>      <dbl>       <dbl>
##  1    1146 2018 Jan            3.06        7.04       3.06        3.06
##  2    1176 2018 Feb            3.06        7.07       3.06        3.06
##  3    1449 2018 Mar            3.09        7.28       3.06        3.08
##  4    1445 2018 Apr            3.09        7.28       3.09        3.09
##  5    1645 2018 May            3.10        7.41       3.09        3.10
##  6    1829 2018 Jun            3.12        7.51       3.10        3.11
##  7    1924 2018 Jul            3.12        7.56       3.12        3.12
##  8    1669 2018 Aug            3.11        7.42       3.12        3.10
##  9    1713 2018 Sep            3.11        7.45       3.11        3.10
## 10    1647 2018 Oct            3.10        7.41       3.11        3.10
## # ... with 14 more rows
```

Build ARIMA models

```
# After break
arima <- train_after_break %>%
  model(
    auto_arima = ARIMA(box_cox_flights, stepwise = FALSE, approx = FALSE),
    model1_arima = ARIMA(box_cox_flights ~ pdq(1,1,1) + PDQ(0,1,0)),
    model2_arima = ARIMA(box_cox_flights ~ pdq(1,1,0) + PDQ(0,1,1))
  )

report(arima %>% dplyr::select(auto_arima))
```

```
## Series: box_cox_flights
## Model: ARIMA(1,0,1)(1,1,1)[12] w/ drift
##
## Coefficients:
##          ar1      ma1     sar1     sma1  constant
##       0.9105  -0.7026  -0.1975  -0.6853     6e-04
## s.e.  0.0757   0.1093   0.2077   0.2156     1e-04
```

```
##
## sigma^2 estimated as 4.152e-05:  log likelihood=346.6
## AIC=-681.19   AICc=-680.25   BIC=-665.81
```

```
report(arima %>% dplyr::select(model1_arima))
```

```
## Series: box_cox_flights
## Model: ARIMA(1,1,1)(0,1,0)[12]
##
## Coefficients:
##           ar1      ma1
##       -0.0983  -0.7693
## s.e.   0.1289   0.0889
##
## sigma^2 estimated as 7.739e-05:  log likelihood=316.15
## AIC=-626.29   AICc=-626.03   BIC=-618.63
```

```
report(arima %>% dplyr::select(model2_arima))
```

```
## Series: box_cox_flights
## Model: ARIMA(1,1,0)(0,1,1)[12]
##
## Coefficients:
##           ar1     sma1
##       -0.5061  -0.8188
## s.e.   0.0877   0.1285
##
## sigma^2 estimated as 5.174e-05:  log likelihood=329.56
## AIC=-653.11   AICc=-652.85   BIC=-645.45
```

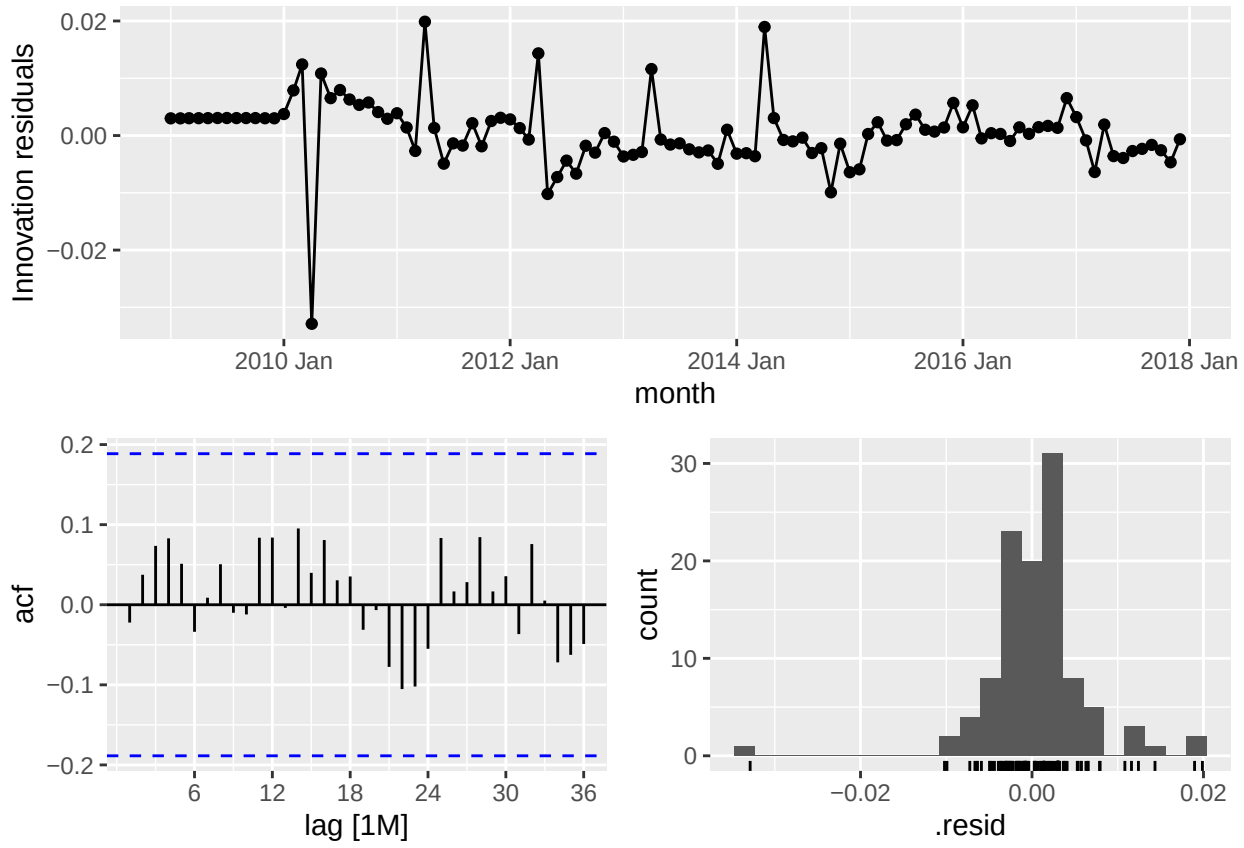
```
arima %>% pivot_longer(everything(), names_to = "Model name",
                       values_to = "Orders")
```

Pivoting model to long format:

```
## # A mable: 3 x 2
## # Key:     Model name [3]
##   `Model name`                                Orders
##   <chr>                                      <model>
## 1 auto_arima   <ARIMA(1,0,1)(1,1,1)[12] w/ drift>
## 2 model1_arima          <ARIMA(1,1,1)(0,1,0)[12]>
## 3 model2_arima          <ARIMA(1,1,0)(0,1,1)[12]>
```

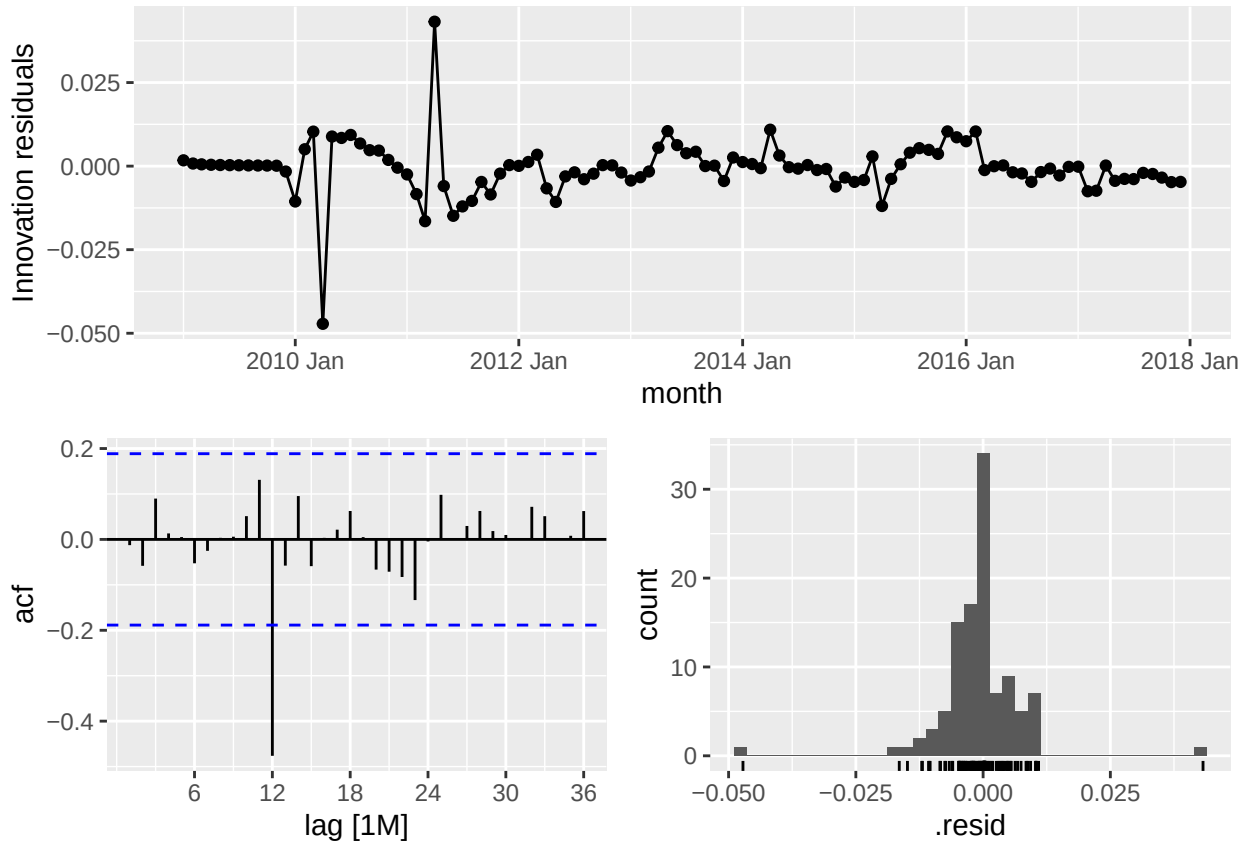
Check the residuals for all three models: Auto ARIMA:

```
arima %>%dplyr::select(auto_arima) %>% gg_tsresiduals(lag=36)
```



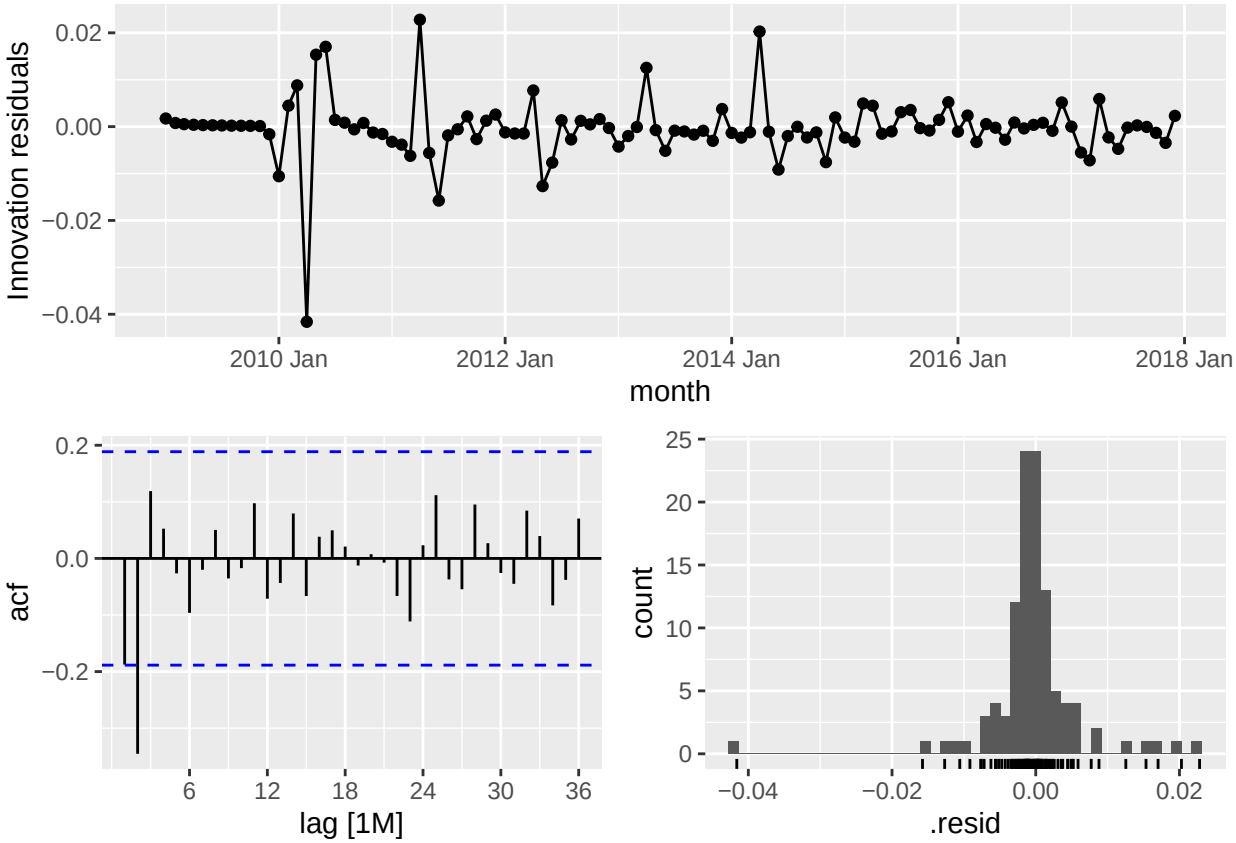
Model1:

```
arima %>%dplyr::select(model1_arima) %>% gg_tsresiduals(lag=36)
```



Model2:

```
arima %>%dplyr::select(model2_arima) %>% gg_tsresiduals(lag=36)
```



The residuals demonstrate some oscillation around the zero line, indicating that our model is, on average, unbiased. Nevertheless, notable anomalies are observed in February from year 2010 until 2014.

The ACF has no significant spikes surpass the boundary, suggesting that the residuals do not display any substantial autocorrelation.

The normal distribution histogram of the residuals exhibits a multimodal pattern with multiple spikes at different counts. The presence of a large spike to the right of zero residuals suggests a cluster of positive residuals, while the spike to the left indicates a cluster of negative residuals. The spike at the center, directly at zero residuals, indicates a cluster of residuals close to zero. The remaining residuals display a symmetric distribution, slightly skewed to the left towards the end. The isolated spikes on the sides could indicate the presence of outliers or anomalies in the data. Overall, the distribution of residuals shows some deviations from a perfect normal distribution, suggesting potential non-normality in the data.

Ljung-box test (ARIMA)

```
arima %>%
residuals() %>%
features(.resid, features = ljung_box, lag = 20)
```

```
## # A tibble: 3 x 3
##   .model       lb_stat lb_pvalue
##   <chr>          <dbl>     <dbl>
## 1 auto_arima      6.73    0.997
## 2 model1_arima   35.4     0.0180
```

```
## 3 model2_arima   24.9     0.207
```

Since the p-value obtained from the Ljung-Box test for the "auto" model is considerably higher than the conventional significance level of 0.05, we lack substantial evidence to reject the null hypothesis. As a result, we can reasonably infer that there is no noteworthy autocorrelation observed in the residuals. This suggests that the residuals of our "auto" model exhibit properties similar to white noise, reinforcing the credibility and appropriateness of our chosen model.

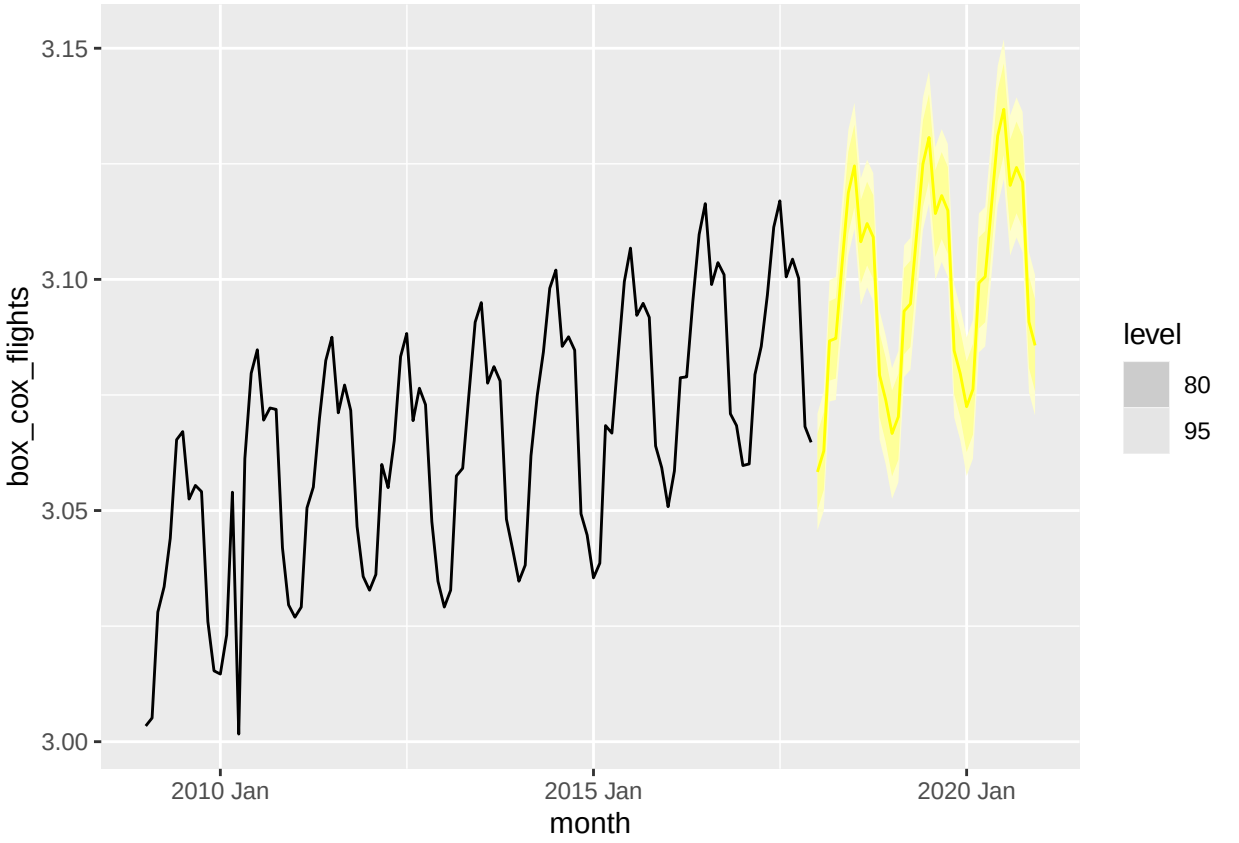
Forecast ARIMA

Train data

```
forecast_1 <- forecast(arima, h=36) %>%
  filter(.model=='auto_arima')

forecast_2 <- forecast(arima, h=36) %>%
  filter(.model=='model1_arima')
```

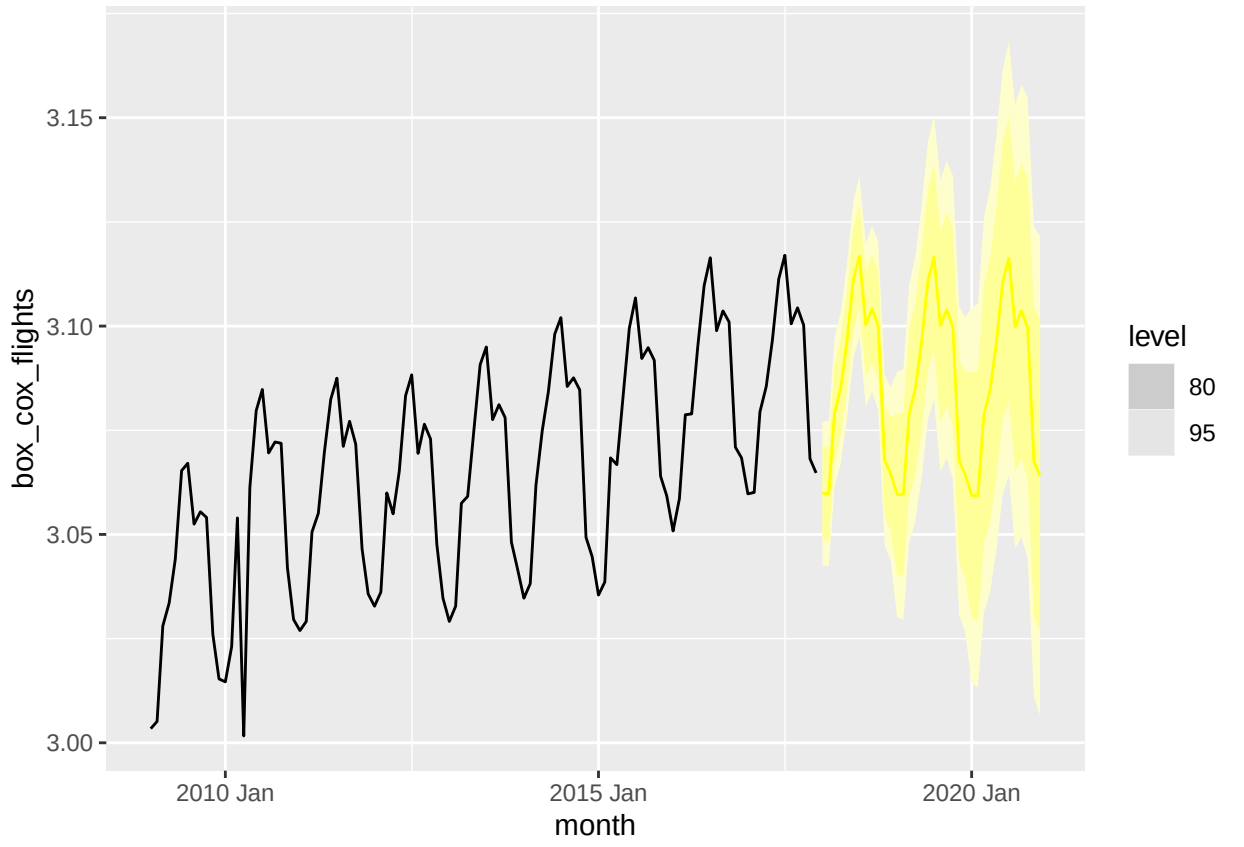
Forecast the auto_arima model and model1: Plotting the 'auto' ARIMA Model with train data:

```
autoplot(forecast_1, train_after_break, color = "yellow")
```



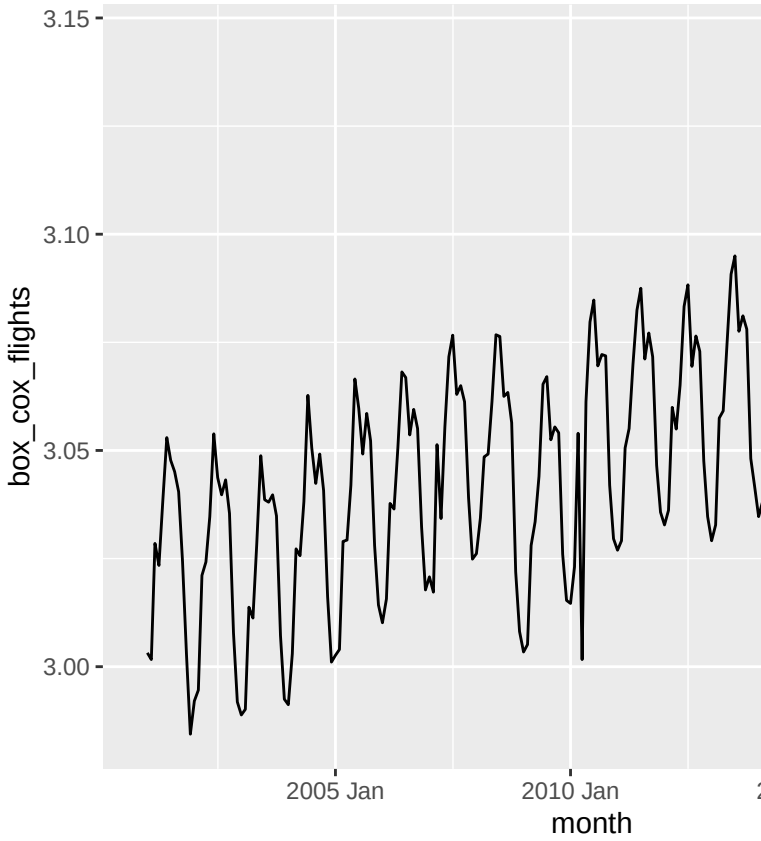
Plotting the 'model1' ARIMA Model with train data:

```
autoplot(forecast_2, train_after_break, color = "yellow")
```



Test data

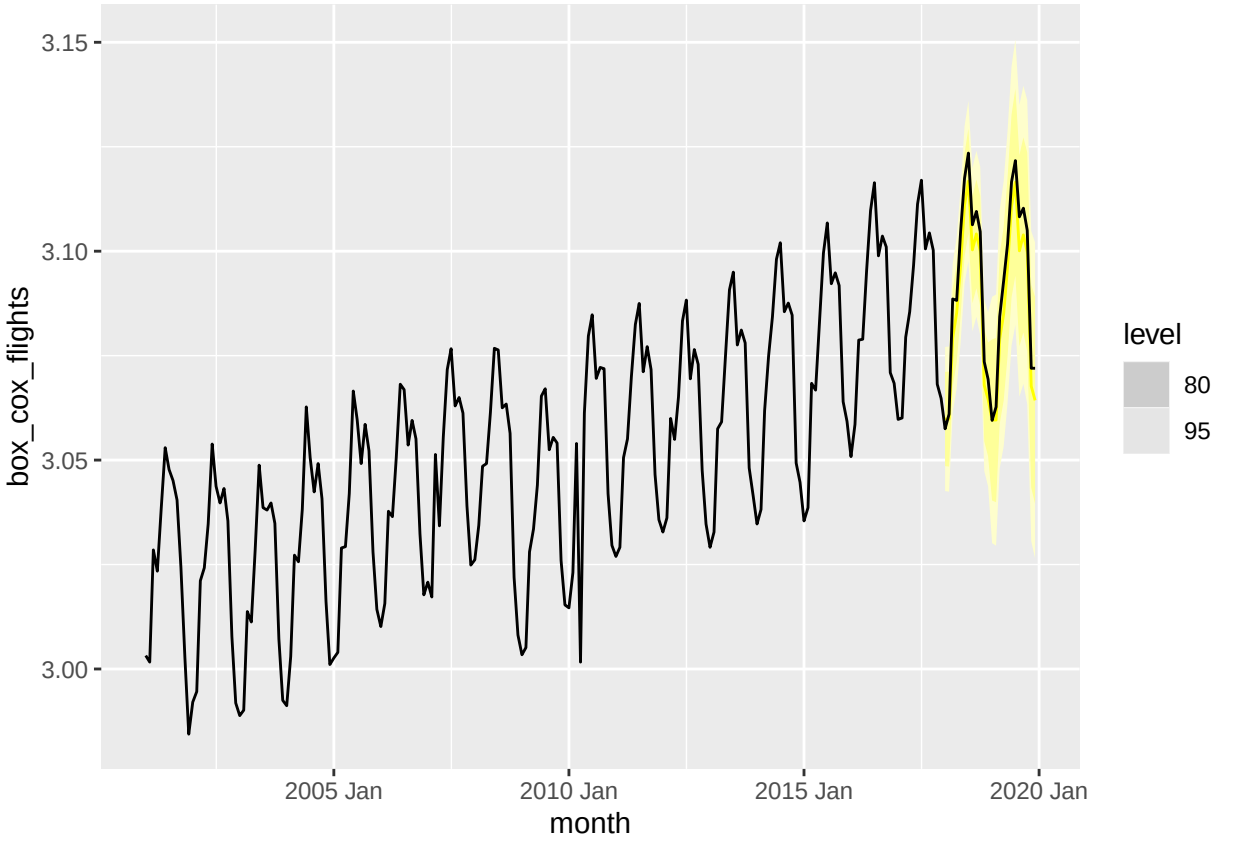
```
plot.arima <- arima %>%
  dplyr::select(auto_arima) %>%
  forecast(test_after_break) %>%
  autoplot(flights, color = "yellow")


print(plot.arima)
```



Plotting the 'auto' ARIMA Model with test data:
The prediction interval for the auto model, shows that the prediction interval ("yellow color") is tightly close to the forecasted values ("black line"). This suggests that the model predicts a very small range of uncertainty, but a tiny bit more uncertainty in the positive direction (values more than the forecast). The presence of seasonality indicates that the model has detected and incorporated a repeating pattern in the data.

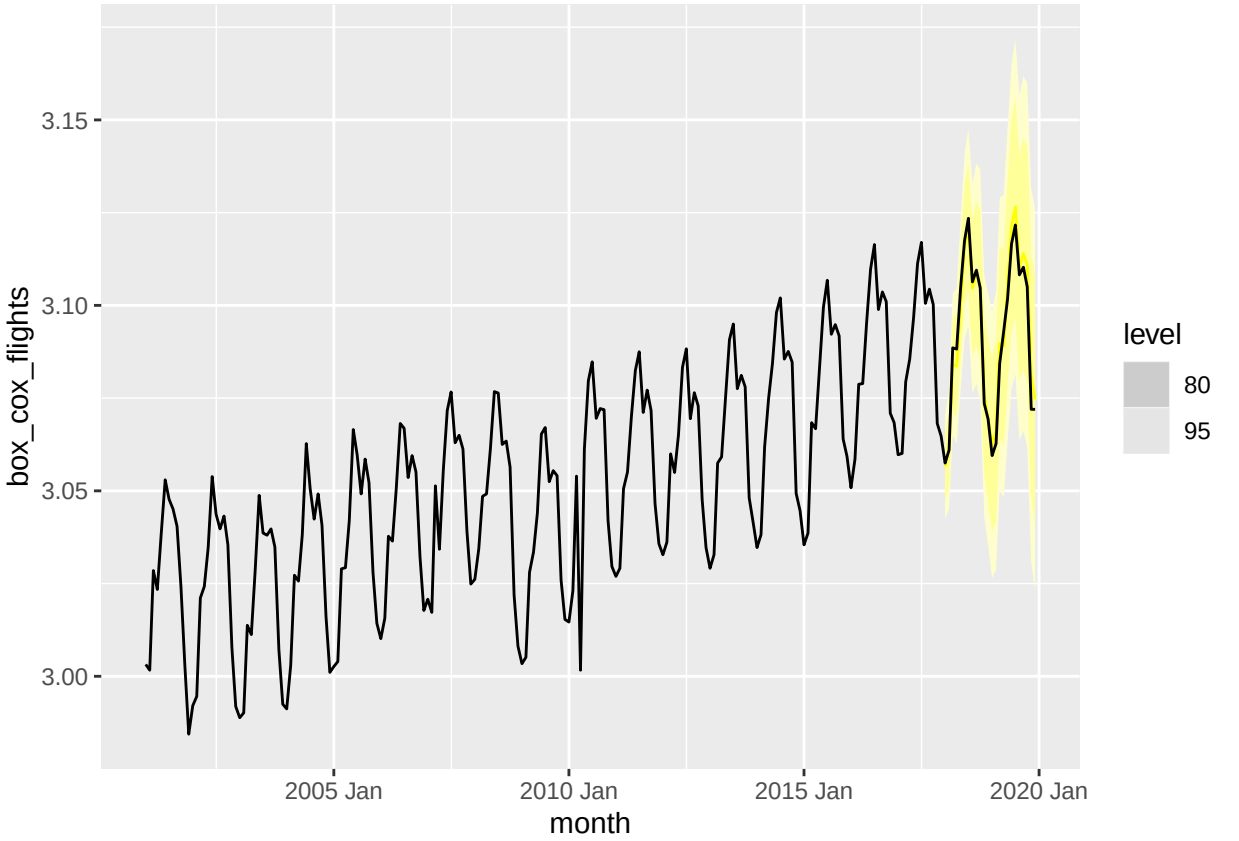
Plotting the 'model1' with test data:

```
plot.arima <- arima %>%
  dplyr::select(model1_arima) %>%
  forecast(test_after_break) %>%
  autoplot(flights, color = "yellow")


print(plot.arima)
```



The prediction interval for the model1 is wider both above and below the forecast line, compared to the auto model. This implies that it is predicting a larger range of uncertainty for future values, both in the positive and negative direction. This might indicate that 'model1' is less certain about future values compared to the 'auto' ARIMA model.

```
plot.arima <- arima %>%
  dplyr::select(model2_arima) %>%
  forecast(test_after_break) %>%
  autoplot(flights, color = "yellow")


print(plot.arima)
```



Calculate the accuracy of these forecasts

```
arima%>%
  forecast(test_after_break) %>%
  accuracy(flights)
```

```
## # A tibble: 3 x 10
##   .model       .type       ME    RMSE     MAE     MPE  MAPE  MASE RMSSE  ACF1
##   <chr>        <chr>    <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 auto_arima   Test  -0.00478 0.00617 0.00513 -0.155  0.166 0.676 0.613 0.727
## 2 model1_arima Test   0.00515 0.00578 0.00535  0.166  0.173 0.704 0.574 0.114
## 3 model2_arima Test  -0.00109 0.00381 0.00321 -0.0353 0.104 0.422 0.379 0.563
```

The auto model has a slightly negative mean error and performs reasonably well in terms of RMSE, MAE, and MAPE. The model1 and model2 models show positive mean errors and have comparable performance in terms of the other metrics. The ACF1 values for all models suggest no significant residual autocorrelation. Overall, the auto model seems to be the best choice based on its performance and lack of autocorrelation in the residuals.

ETS

Auto, guess1 and guess2 models

```
ets <- train_after_break %>%
  model(
    auto_ets = ETS(box_cox_flights),
    guess1_ets = ETS(box_cox_flights ~ error("A") + trend("Ad") + season("A")),
    guess2_ets = ETS(box_cox_flights ~ error("M") + trend("Ad") + season("A"))
  )

report(ets %>%
dplyr::select(auto_ets))
```

```
## Series: box_cox_flights
## Model: ETS(A,A,A)
##   Smoothing parameters:
##     alpha = 0.2232895
##     beta  = 0.0001009399
##     gamma = 0.0001014674
##
##   Initial states:
##      l[0]         b[0]        s[0]      s[-1]      s[-2]      s[-3]      s[-4]
##  3.035156 0.0005083501 -0.02435279 -0.0163582 0.01344665 0.01669082 0.01386019
##       s[-5]      s[-6]      s[-7]        s[-8]        s[-9]     s[-10]
##  0.03087183 0.02535363 0.01080296 -0.007646294 -0.004282987 -0.0275235
##       s[-11]
##  -0.03086229
##
##   sigma^2:  0
##
##       AIC      AICc       BIC
## -583.2383 -576.4383 -537.6421
```

```
report(ets %>%
dplyr::select(guess1_ets))
```

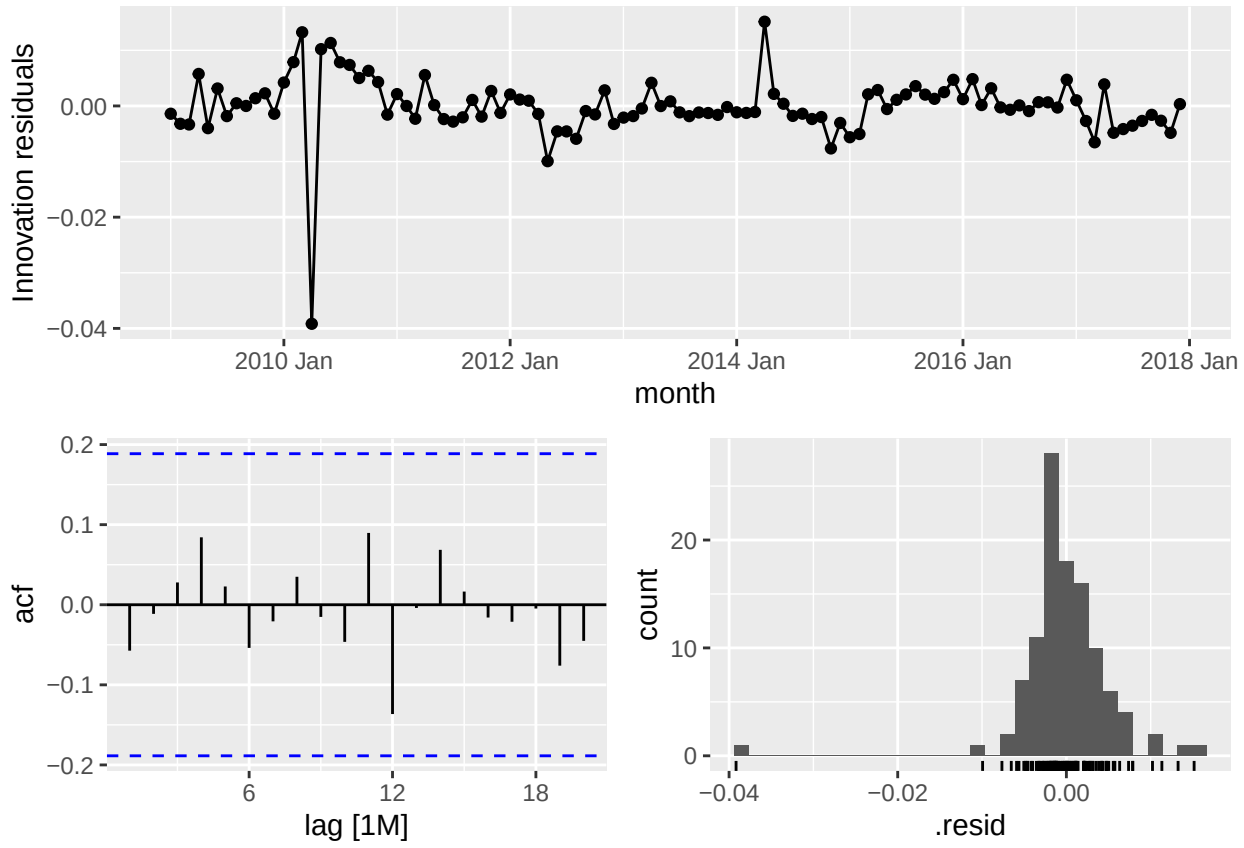
```
## Series: box_cox_flights
## Model: ETS(A,Ad,A)
##   Smoothing parameters:
##     alpha = 0.2101288
##     beta  = 0.0001025252
##     gamma = 0.0001084556
##     phi   = 0.9796453
##
##   Initial states:
##      l[0]       b[0]        s[0]       s[-1]      s[-2]      s[-3]      s[-4]
##  3.030975 0.00117205 -0.02435747 -0.01648814 0.01342321 0.01710208 0.01353076
##       s[-5]      s[-6]       s[-7]        s[-8]       s[-9]      s[-10]
##  0.03000708 0.02561642 0.009976143 -0.007500632 -0.00394759 -0.02714847
##       s[-11]
```

```
##  -0.03021339
##
##   sigma^2:  0
##
##       AIC      AICc       BIC
## -582.8208 -575.1354 -534.5425
```

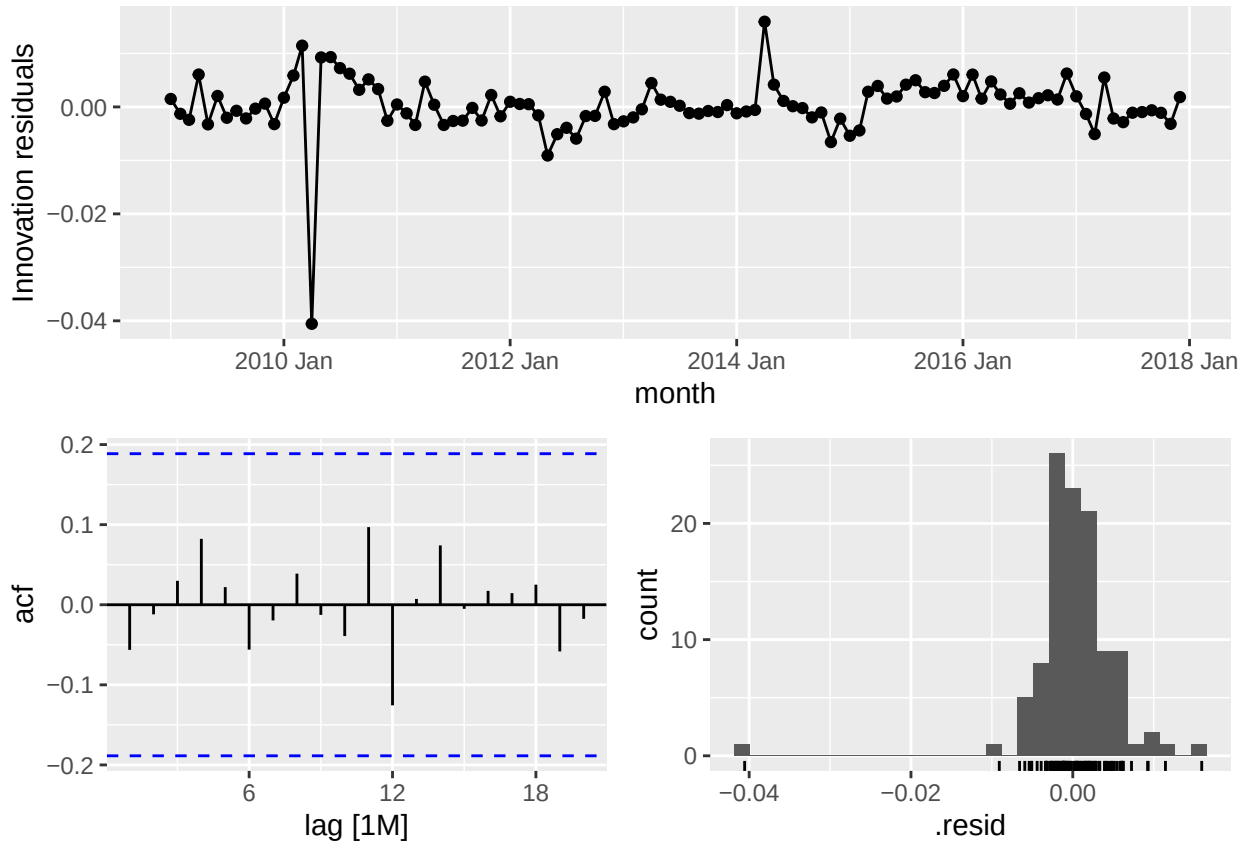
```
report(ets %>%
dplyr::select(guess2_ets))
```

```
## Series: box_cox_flights
## Model: ETS(M,Ad,A)
##   Smoothing parameters:
##     alpha = 0.2268771
##     beta  = 0.0001003358
##     gamma = 0.0001010376
##     phi   = 0.9799977
##
##   Initial states:
##      l[0]        b[0]        s[0]       s[-1]      s[-2]      s[-3]      s[-4]
##  3.029901 0.001185635 -0.02430962 -0.01602243 0.01337368 0.01675668 0.01301458
##       s[-5]     s[-6]      s[-7]        s[-8]        s[-9]      s[-10]
##  0.03026879 0.0255488 0.01029302 -0.007248507 -0.003963619 -0.02758415
##       s[-11]
##  -0.03012724
##
##   sigma^2:  0
##
##       AIC      AICc       BIC
## -581.5828 -573.8974 -533.3045
```

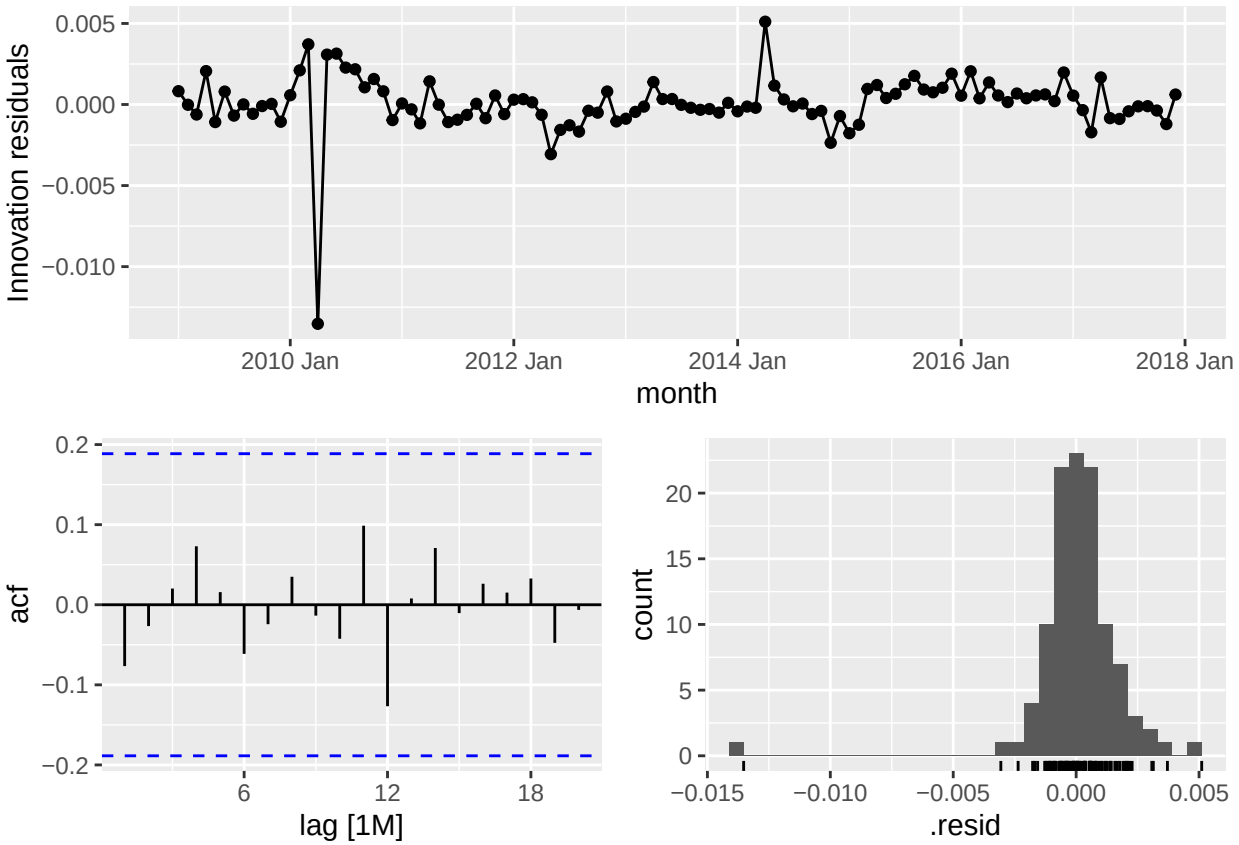
```
ets %>%
dplyr::select(auto_ets) %>%
gg_tsresiduals(type = "innovation")
```




```
ets %>%
dplyr::select(guess1_ets) %>%
gg_tsresiduals(type = "innovation")
```




```
ets %>%
dplyr::select(guess2_ets) %>%
gg_tsresiduals(type = "innovation")
```



The residuals demonstrate a some oscillation around the zero line, indicating that our model is, on average, unbiased. Nevertheless, notable anomalies are observed in February 2012, and February 2014. The ACF has no significant spikes surpass the boundary, suggesting that the residuals do not display any substantial autocorrelation. The histogram plot of the residuals demonstrates some asymmetric distribution. The frequency of residuals approx. gradually decreases on both sides, suggesting a normal distribution.

```
ets %>%
residuals() %>%
features(.resid, features = ljung_box, lag = 20)
```

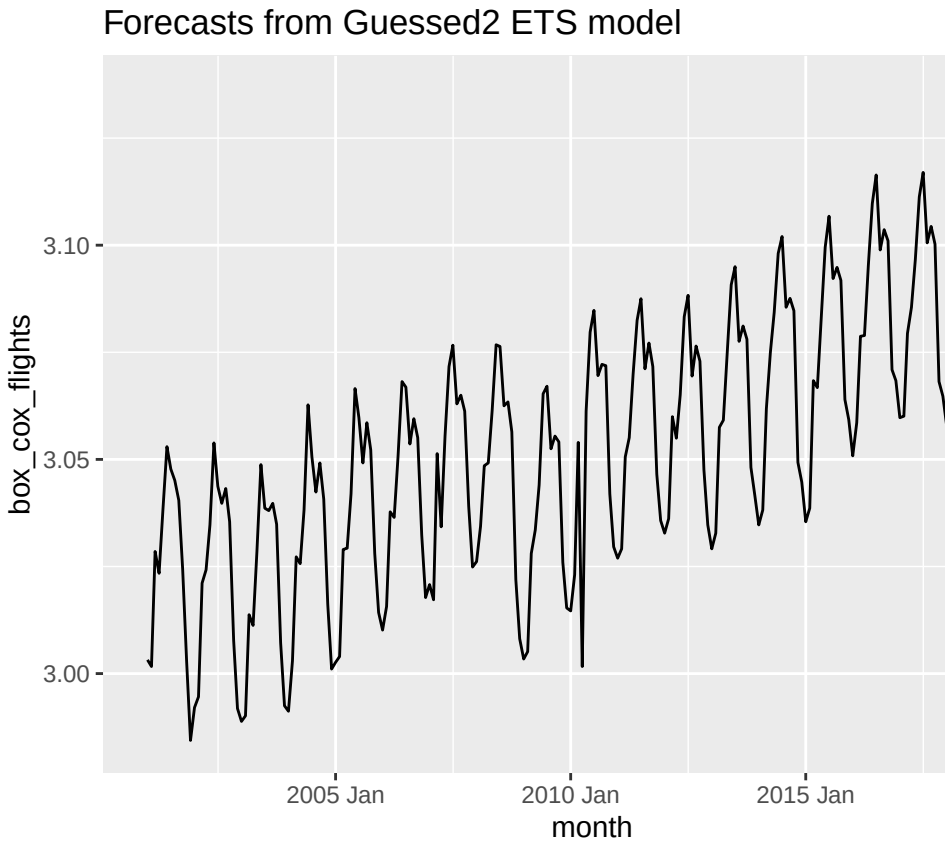
Ljung-box test (ETS)

```
## # A tibble: 3 x 3
##   .model     lb_stat lb_pvalue
##   <chr>        <dbl>     <dbl>
## 1 auto_ets      7.22     0.996
## 2 guess1_ets    6.54     0.998
## 3 guess2_ets    6.71     0.998
```

The Ljung-Box p-values for all three models are high, which suggests that there is no significant autocorrelation in the residuals of either model. This indicates that the ETS models adequately capture the autocorrelation structure in the data.

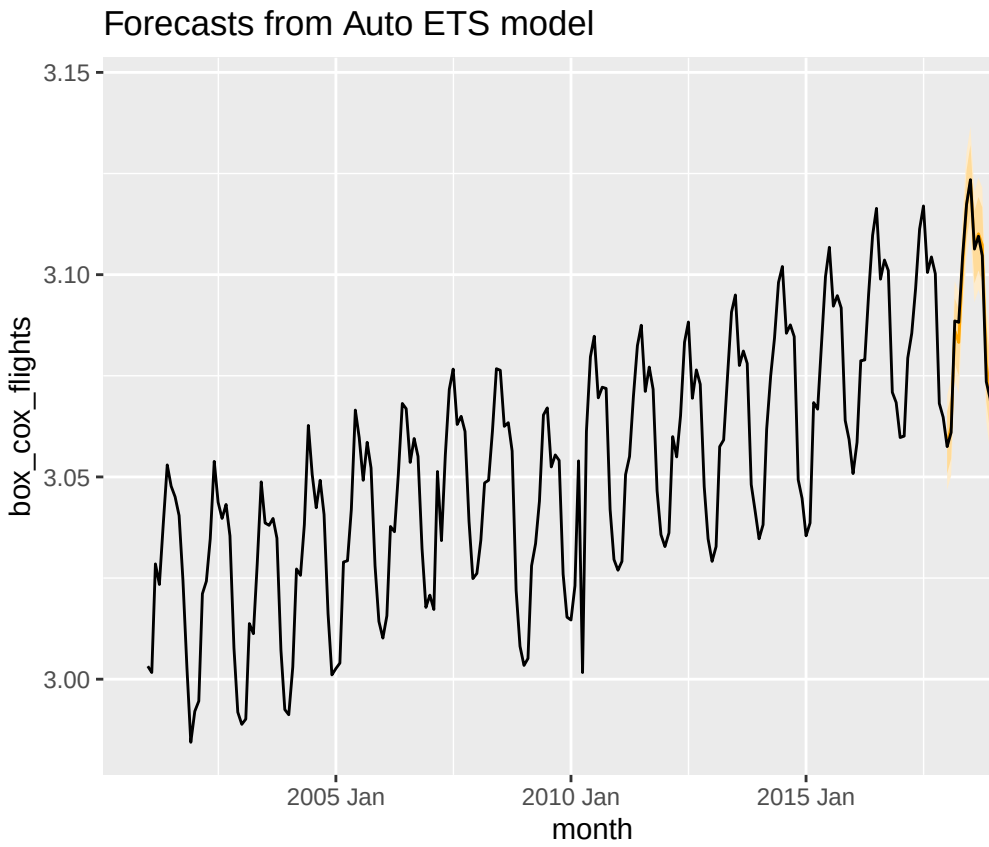
Forecast ETS

```
plot_guess <- ets %>%
  dplyr::select(guess2_ets) %>%
  forecast(test_after_break) %>%
  autoplot(flights, color = "orange") +
  labs(title = "Forecasts from Guessed2 ETS model")

print(plot_guess)
```



Forecasts from Guessed ETS model

```
plot_auto <- ets %>%
  dplyr::select(auto_ets) %>%
  forecast(test_after_break) %>%
  autoplot(flights, color = "orange") +
  labs(title = "Forecasts from Auto ETS model")

print(plot_auto)
```



Forecasts from Auto ETS model

```
ets_forecast <- ets %>%
forecast(test_after_break)

accuracy(ets_forecast, flights %>%
dplyr::select(box_cox_flights))
```

```
## # A tibble: 3 x 10
##   .model     .type       ME    RMSE     MAE     MPE  MAPE  MASE RMSSE    ACF1
##   <chr>      <chr>    <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1 auto_ets   Test  -0.00307 0.00521 0.00422 -0.0995 0.137 0.556 0.517  0.524
## 2 guess1_ets Test   0.00310 0.00422 0.00332  0.100  0.107 0.438 0.419 -0.0318
## 3 guess2_ets Test   0.00308 0.00421 0.00336  0.0995 0.108 0.442 0.418 -0.0611
```

The "auto_ets" model exhibited the best performance among the three models, as it achieved lower values for all matrices, compared to the "guess1_ets" and "guess2_ets" models, indicating higher forecasting accuracy and a closer fit to the test data.

SNAIVE

```
model(train_after_break, snaive = SNAIVE(flights)) %>% report()
```

Forecasting with SNAIVE Model:

```
## Series: flights
## Model: SNAIVE
##
## sigma^2: 4562.6236
```

The SNAIVE model estimated the variance (sigma^2) of the "flights" series to be 4562.6236. This indicates that the model's predictions had a relatively large spread or deviation from the actual data points. A higher variance suggests that the model may not have captured the underlying patterns and dynamics of the data effectively, resulting in less accurate predictions.

```
train_after_break %>% model(SNAIVE(box_cox_flights ~ lag("year")))
```

SNAIVE Model with lagged year variable (train data):

```
## # A mable: 1 x 1
##   `SNAIVE(box_cox_flights ~ lag("year"))`
##                                    <model>
## 1                                 <SNAIVE>
```

```
flights_fit <- train_after_break %>%
  model(
    naive = NAIVE(box_cox_flights),
    drift = RW(box_cox_flights ~ drift()),
    mean = MEAN(box_cox_flights),
    snaive = SNAIVE(box_cox_flights)
  )
```

Fit multiple models on train data (Naive, Random Walk, Mean, and SNAIVE):

```
forecast_flights <- flights_fit %>%
  forecast(h= "2 years")
```

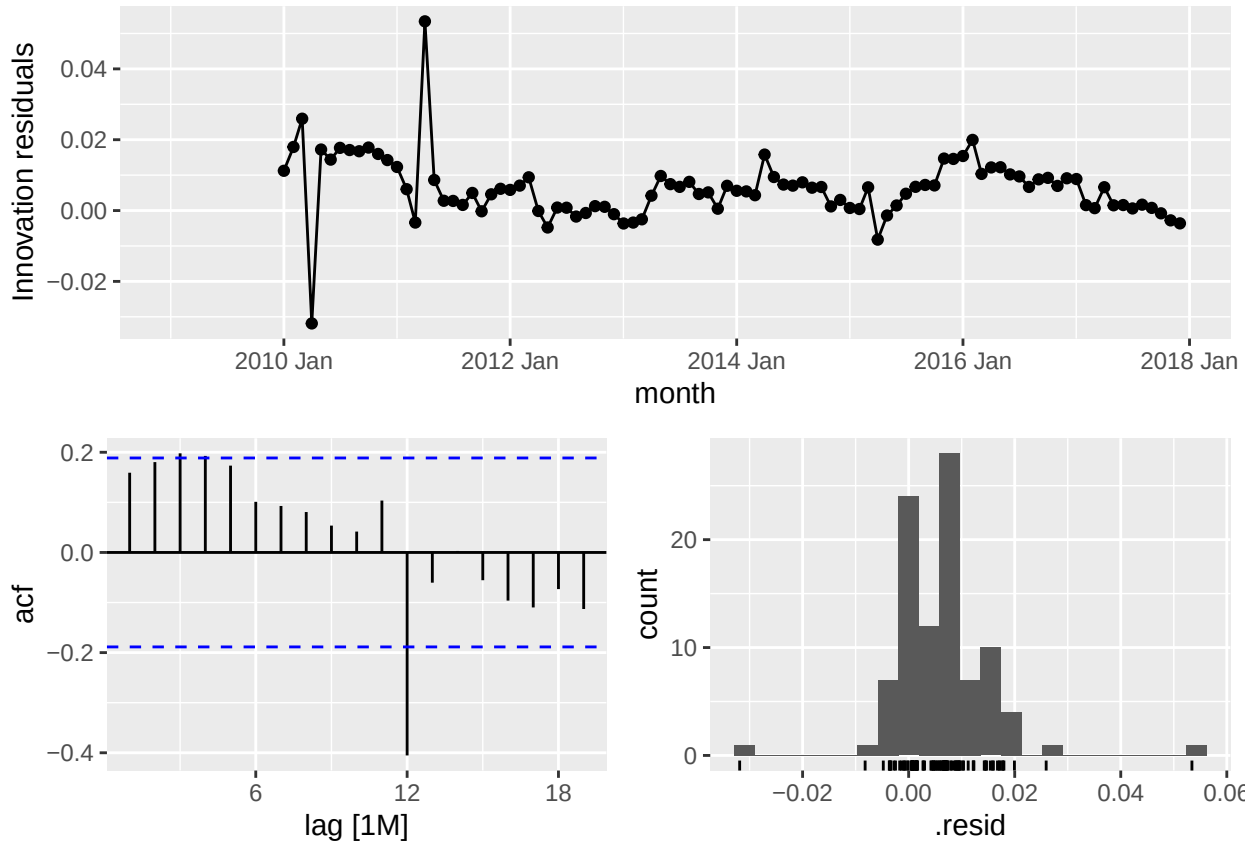
```
flights_fit %>%
  dplyr::select(snaive) %>%
  gg_tsresiduals()
```

Forecasts the models with a horizon of "2 years":

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

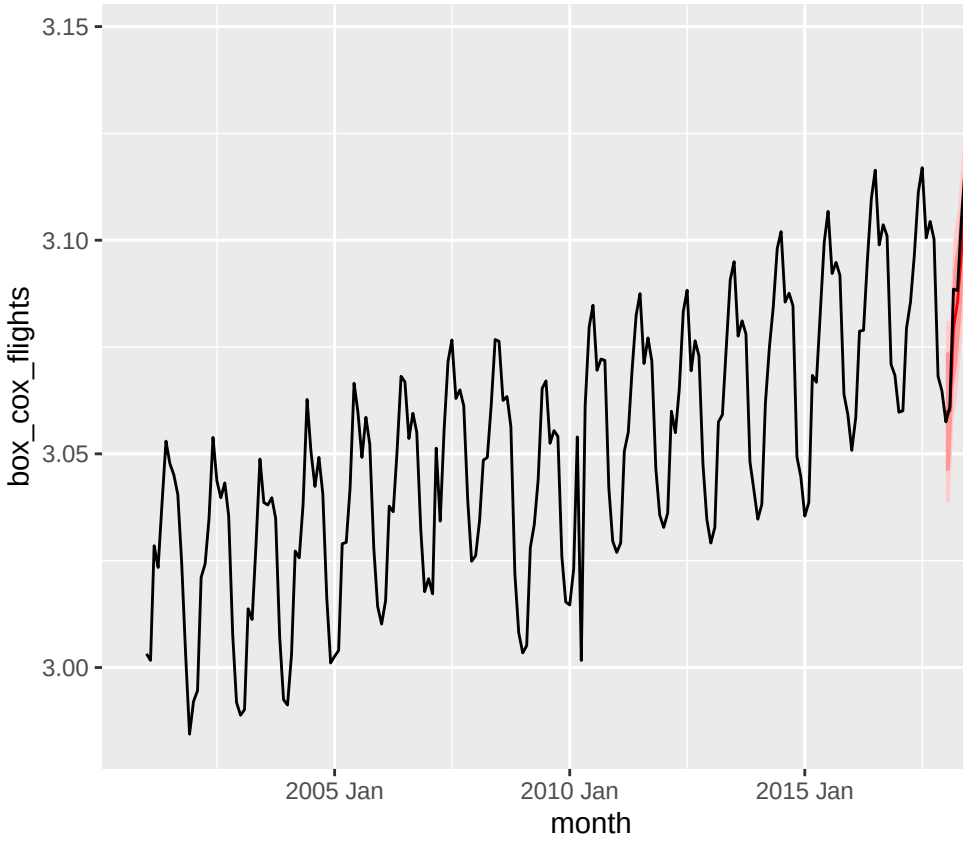
```
## Warning: Removed 12 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 12 rows containing non-finite values (`stat_bin()`).
```



There is one significant spike at lag 12 that goes outside the threshold boundaries in the ACF plot, it suggests the presence of a significant seasonal autocorrelation at that lag. It may imply that the model is not adequately capturing the seasonal patterns in the data. The presence of two high spikes in the histogram suggests a bimodal distribution, indicating the existence of two distinct groups or patterns within the observed values.

```
flights_fit%>%
  dplyr::select(snaive) %>%
  forecast(h = "2 years") %>%
  autoplot(flights,color = "red")
```



Forecasting with SNAIVE Model:
The red curves in the forecast occasionally exceed the black line, indicating higher forecast errors, the overall performance captures the underlying trend and seasonality, suggesting a reasonably good forecast. However,the red curves extend beyond the black line more frequently than the ETS and ARMA models indicating that the snaive model may have more variability and less accuracy in predicting the future values.

```
accuracy(forecast_flights, flights %>%
dplyr::select(box_cox_flights))
```

Accuracy Assessment of Forecast

```
## # A tibble: 4 x 10
##   .model .type      ME    RMSE     MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr>   <dbl>   <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 drift  Test  0.0202  0.0290  0.0244  0.649 0.786 3.21  2.88  0.697
## 2 mean   Test  0.0267  0.0340  0.0284  0.858 0.915 3.74  3.37  0.697
## 3 naive  Test  0.0274  0.0345  0.0289  0.881 0.930 3.80  3.43  0.697
## 4 snaive Test  0.00482 0.00546 0.00503 0.156 0.162 0.662 0.542 0.102
```

The Snaive model outperformed the other models, exhibiting the lowest values for RMSE, MAE, MASE, and RMSSE, suggesting it provides the most accurate forecast for the given data.

Reality

```
flights_reality %>% autoplot(flights) + labs(title = "Flight traffic")
```

Finally, what the actual air traffic looked like both during and after the pandemic: