Homework 2

Forecasting: Principles and Practice - The Forecasters Toolbox

Paul Perez

2/21/2021

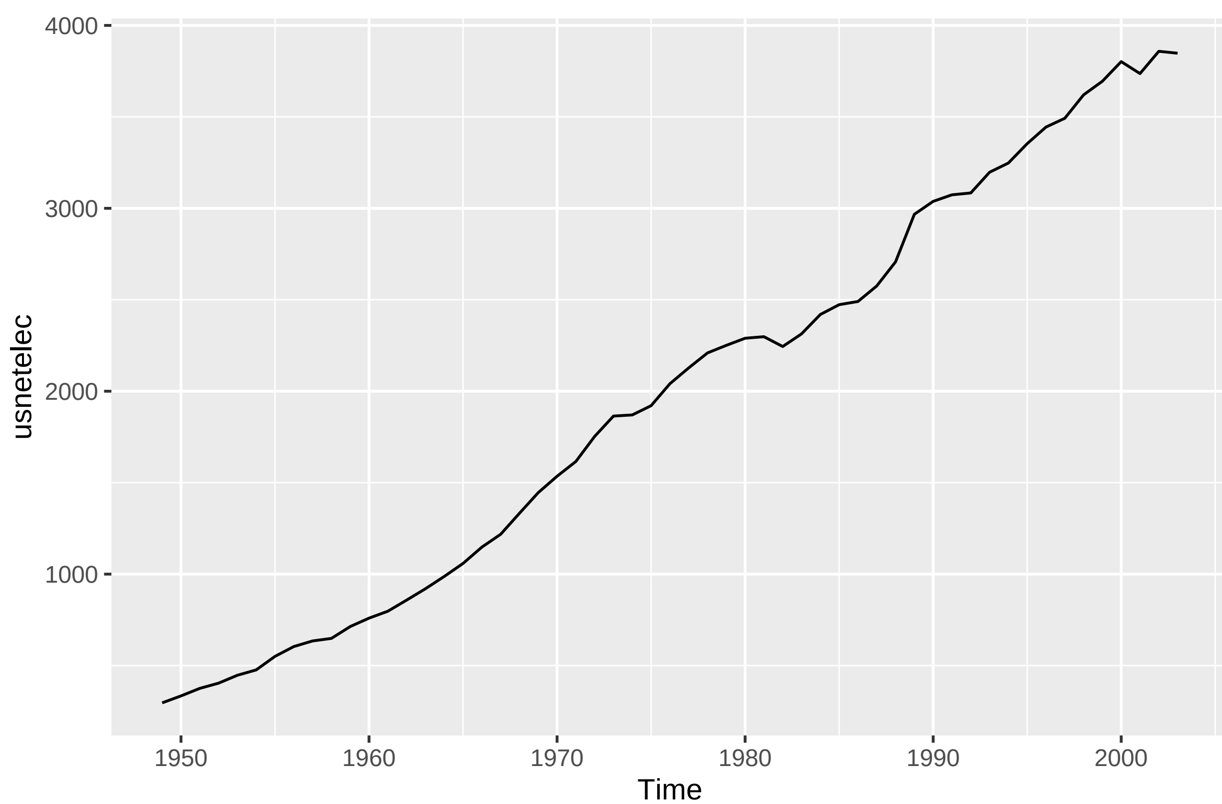
Exercise 3.7 - 1

For the following series, find an appropriate Box-Cox transformation in order to stabilise the variance.

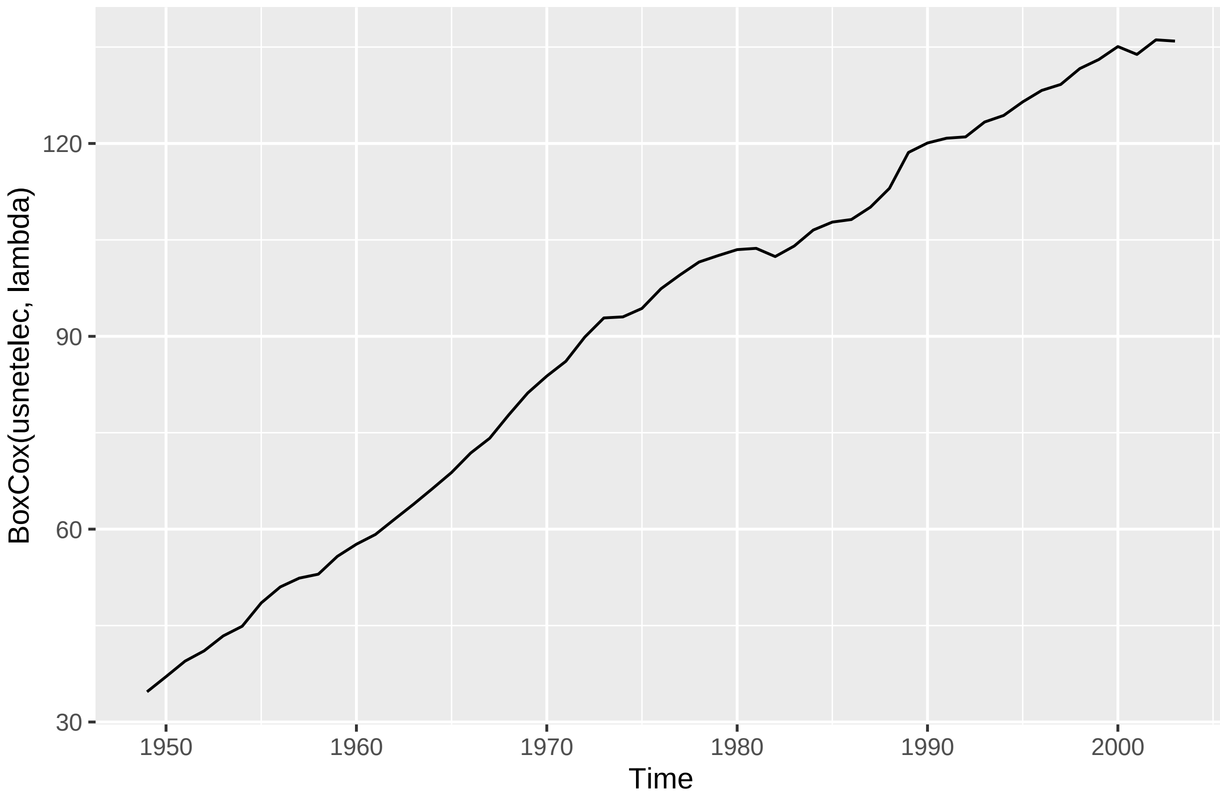
- `usnetelec`
- `usgdp`
- `mcopper`
- `enplanements`

usnetelec

```r
autoplot(usnetelec)
```



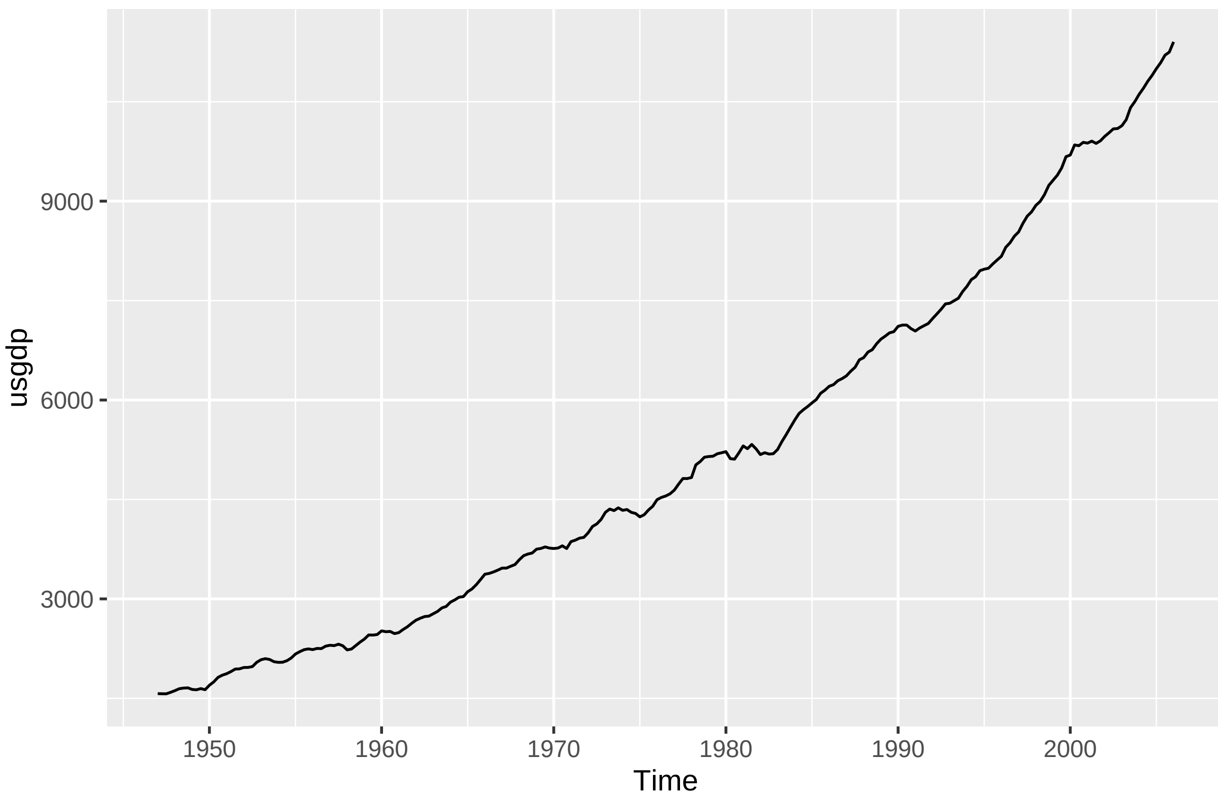
```r
lambda <- BoxCox.lambda(usnetelec)
autoplot(BoxCox(usnetelec, lambda))
```



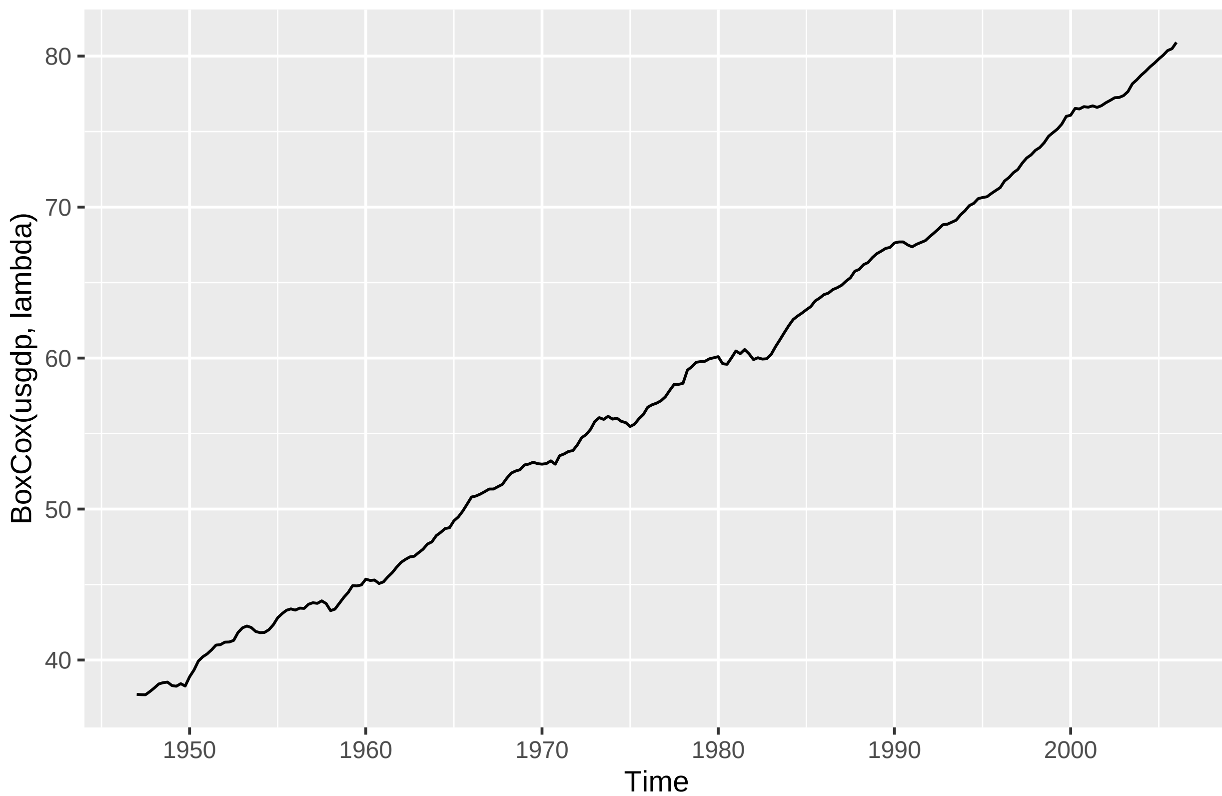
The lambda for the `usnetelec` Box-Cox is 0.5167714.

usgdp

```
autoplot(usgdp)
```



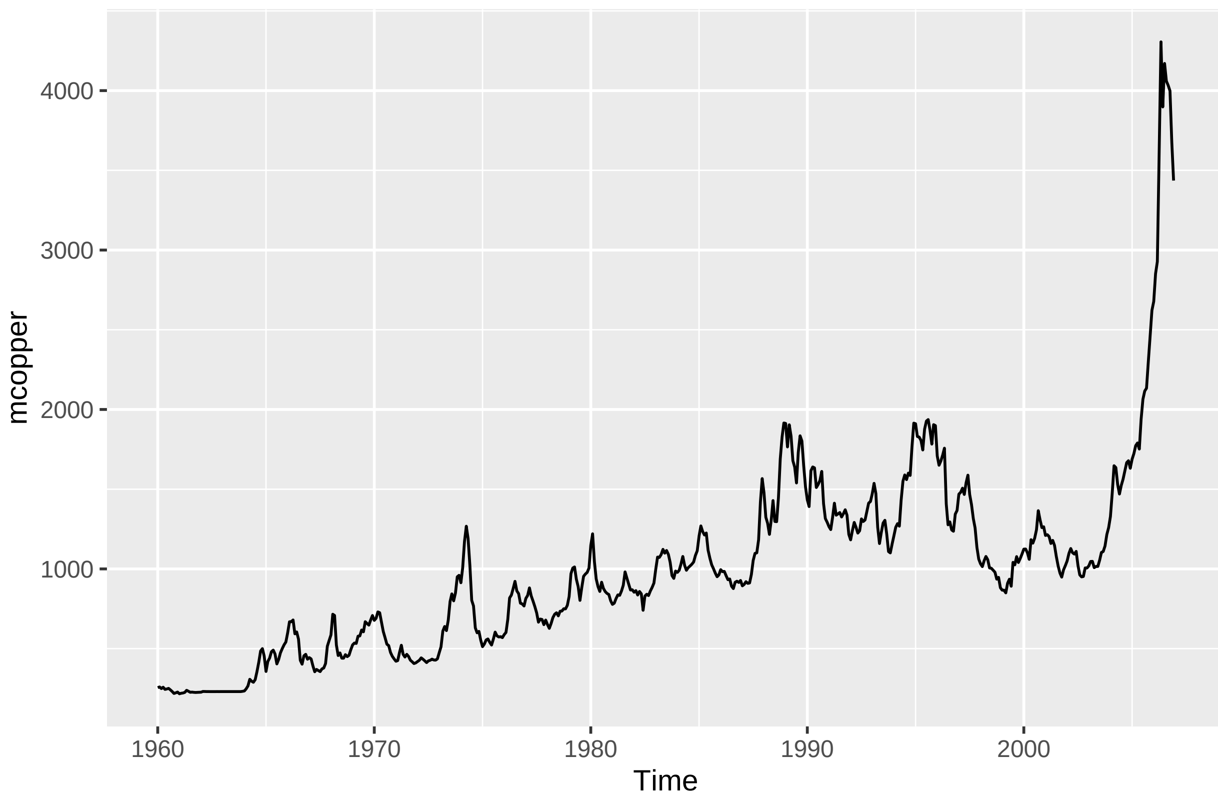

```
lambda <- BoxCox.lambda(usgdp)
autoplot(BoxCox(usgdp, lambda))
```



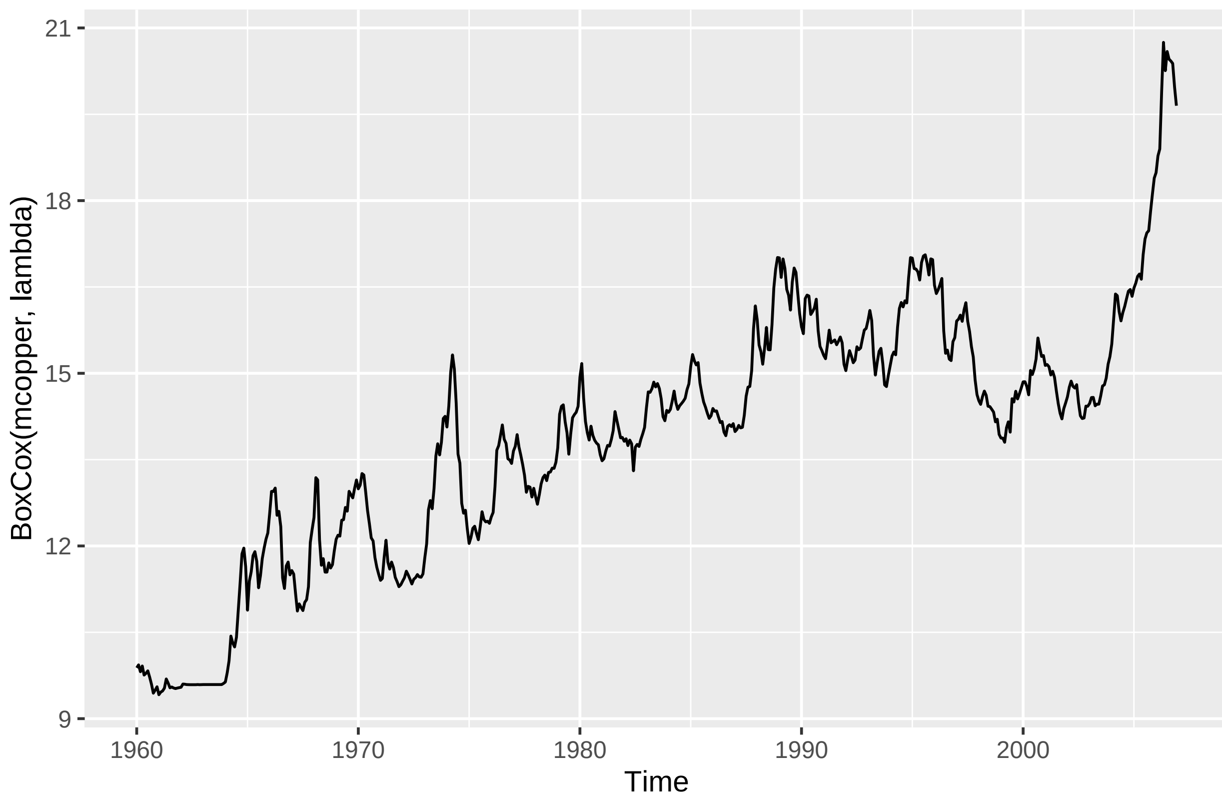
The lambda for the `usgdp` Box-Cox is 0.366352.

mcopper

```r
autoplot(mcopper)
```



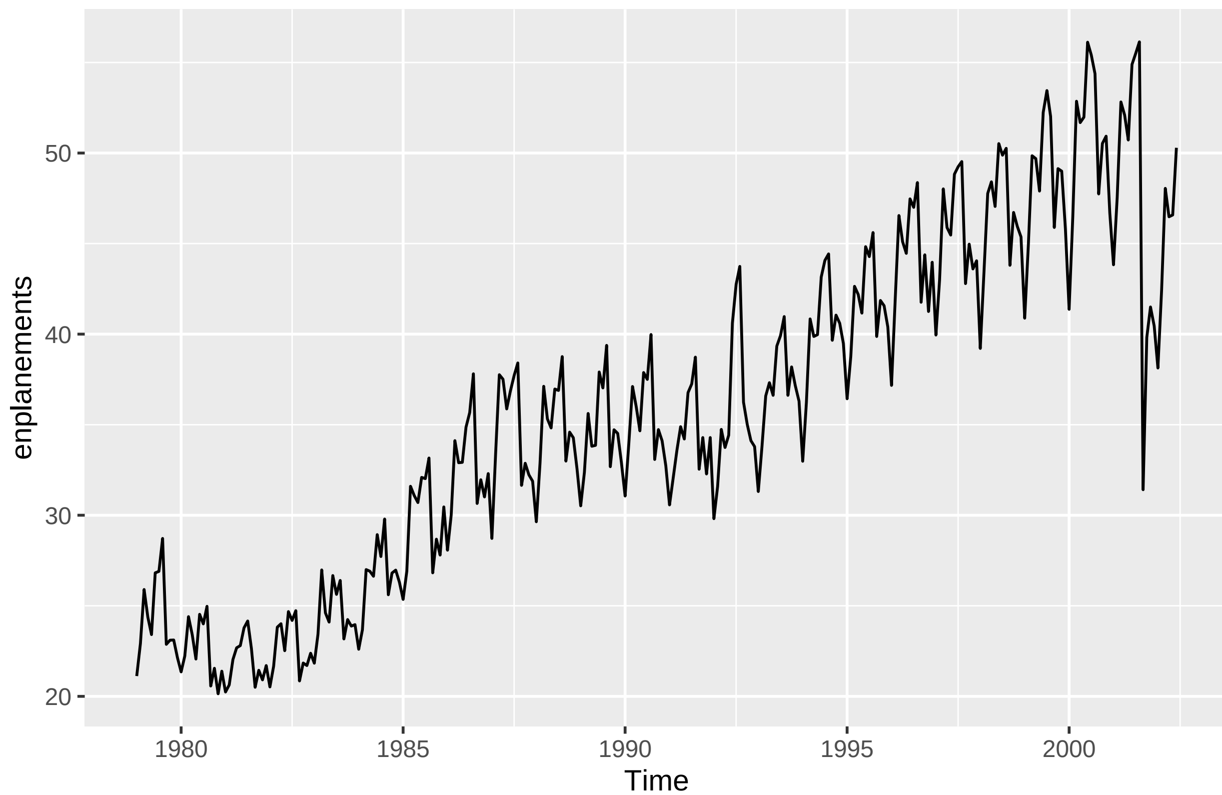

```r
lambda <- BoxCox.lambda(mcopper)
autoplot(BoxCox(mcopper, lambda))
```



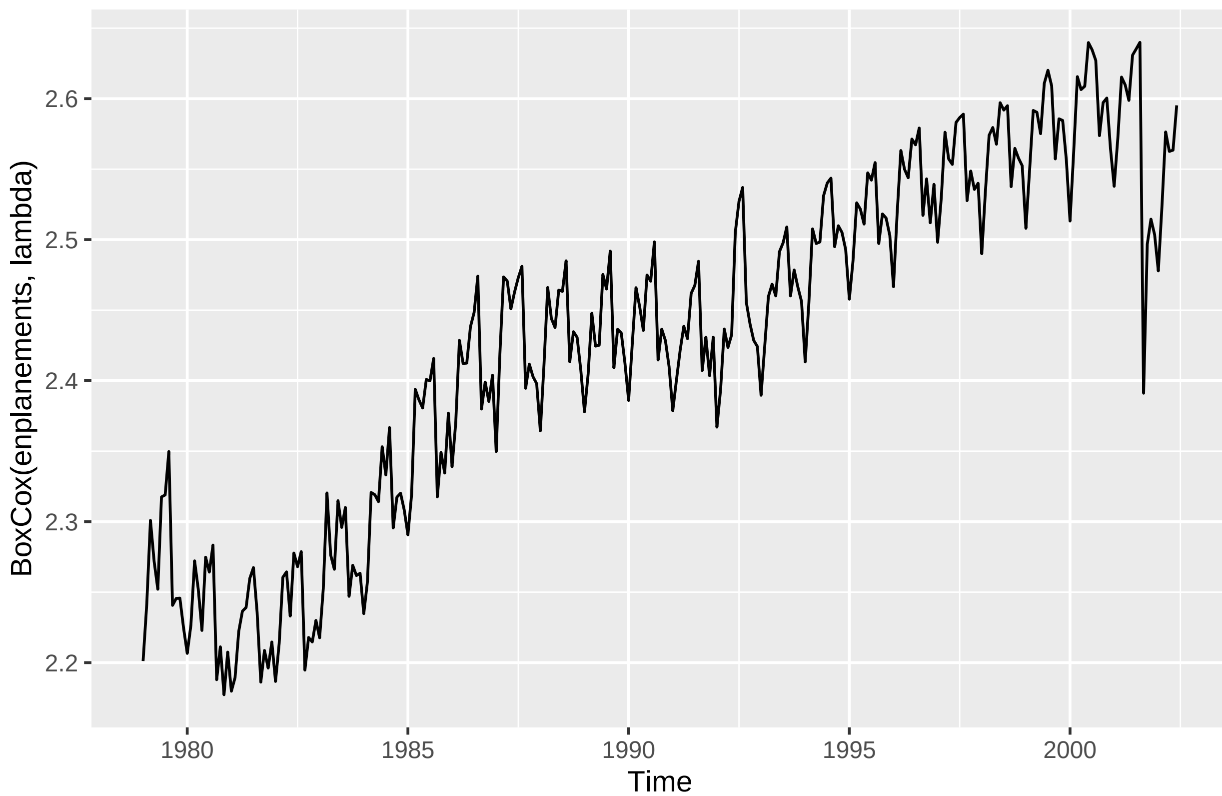
The lambda for the `mcopper` Box-Cox is 0.1919047.

enplanements

```r
autoplot(enplanements)
```




```r
lambda <- BoxCox.lambda(enplanements)
autoplot(BoxCox(enplanements, lambda))
```

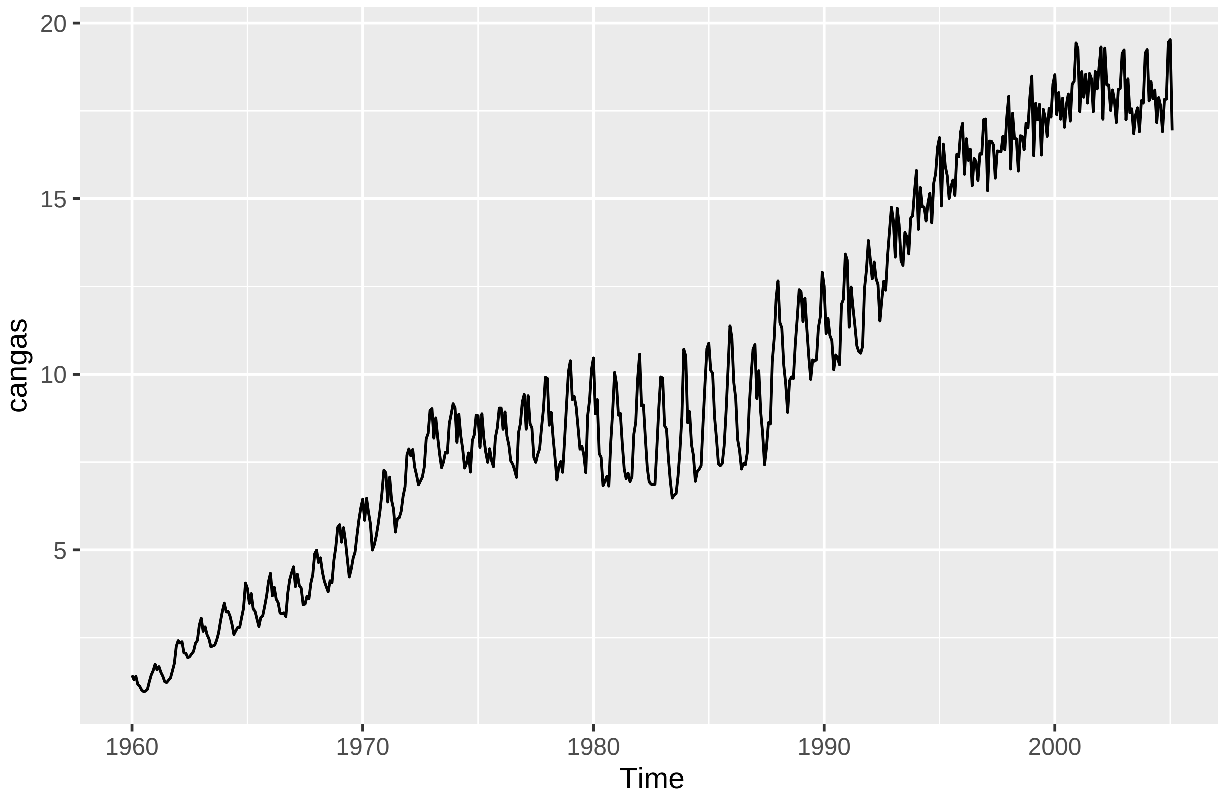


The lambda for the `enplanements` Box-Cox is -0.2269461.

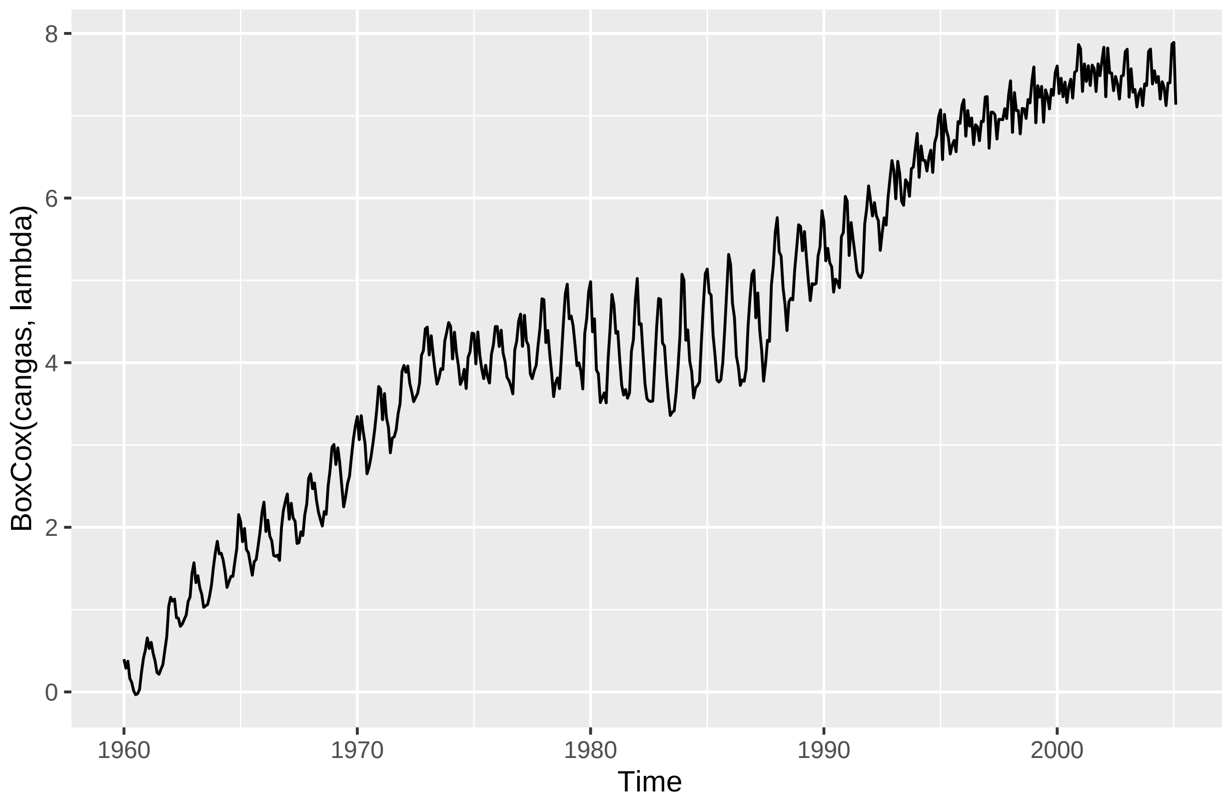
Exercise 3.7 - 2

Why is a Box-Cox transformation unhelpful for the `cangas` data?

```
autoplot(cangas)
```




```
lambda <- BoxCox.lambda(cangas)
autoplot(BoxCox(cangas, lambda))
```



Looking at both plots, the original and Box-Cox transformation, there does not appear to be much of a change. Specifically looking at the window of the mid 1970's through to 1990, the variance seems to increase on both plots. We'll need to use a different transformation to handle this dataset.

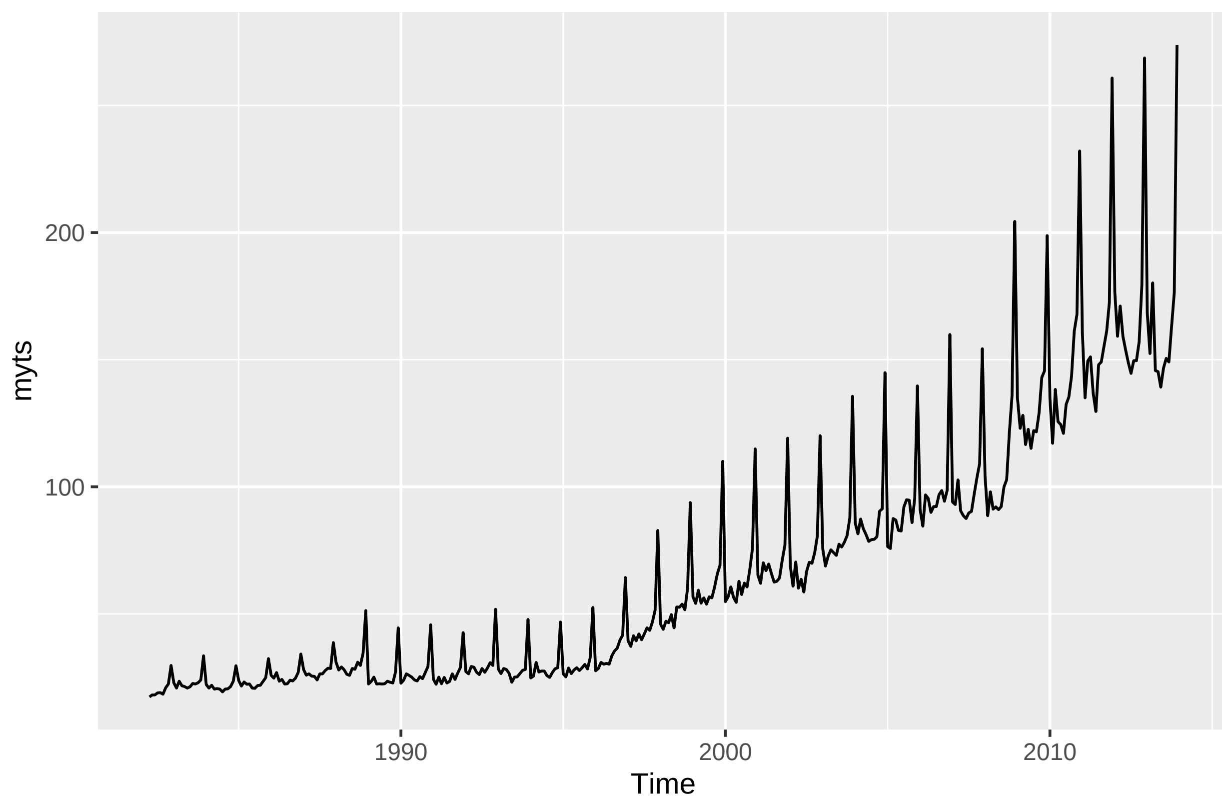
Exercise 3.7 - 3

What Box-Cox transformation would you select for your retail data (From Exercise 3 in Section 2.10)?

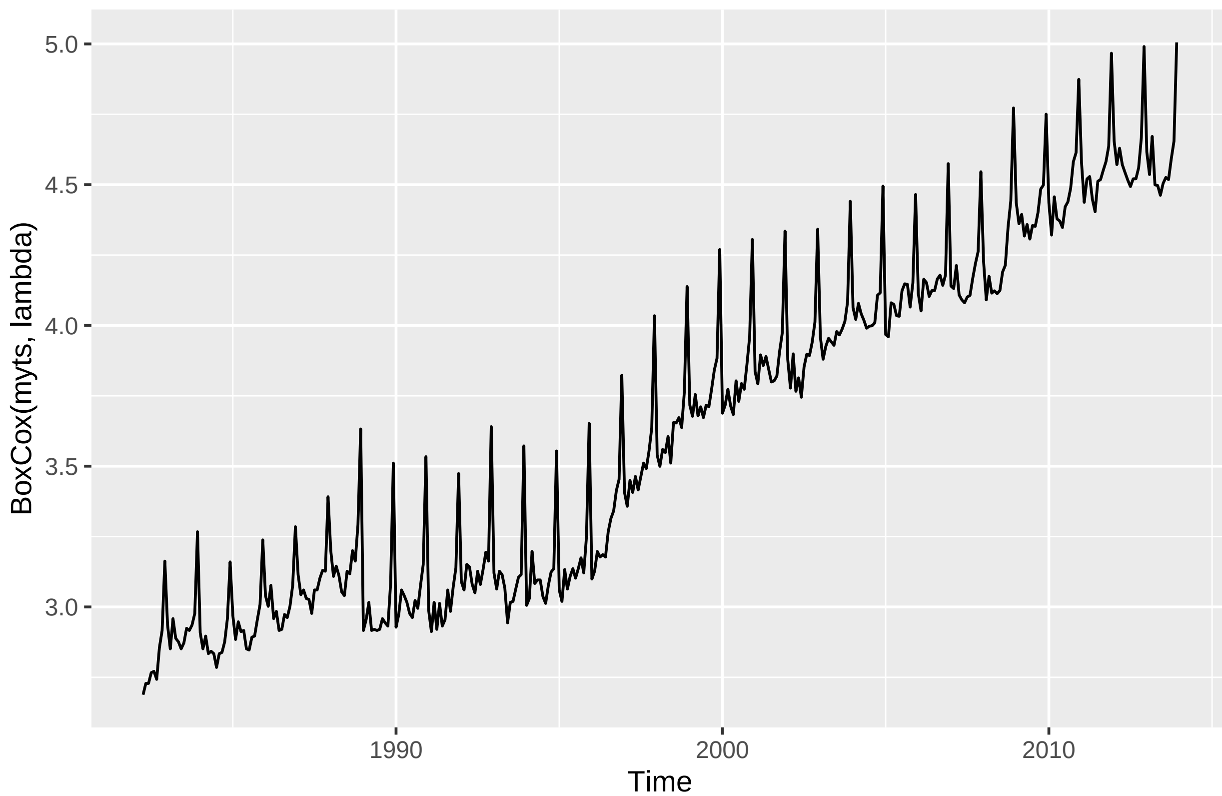
```
retaildata <- readxl::read_excel("retail.xlsx", skip=1)

myts <- ts(retaildata[,"A3349414R"],
  frequency=12, start=c(1982,4))
```

```
autoplot(myts)
```




```
lambda <- BoxCox.lambda(myts)
autoplot(BoxCox(myts, lambda))
```



Looking into my retail data from exercise 3 in section 2.10, the `BoxCox.lamnda()` function provides us with an ideal lambda of `-0.04159144`. Being that this is negative, and our `BoxCox()` function allows for negative values, we're going to use a Power Transformation using a sign function in the formula. Please see below.

$$w_t = \begin{cases} \log(y_t) & \text{if } \lambda = 0; \\ \text{sign}(y_t)(|y_t|^\lambda - 1)/\lambda & \text{otherwise.} \end{cases}$$

Exercise 3.7 - 8

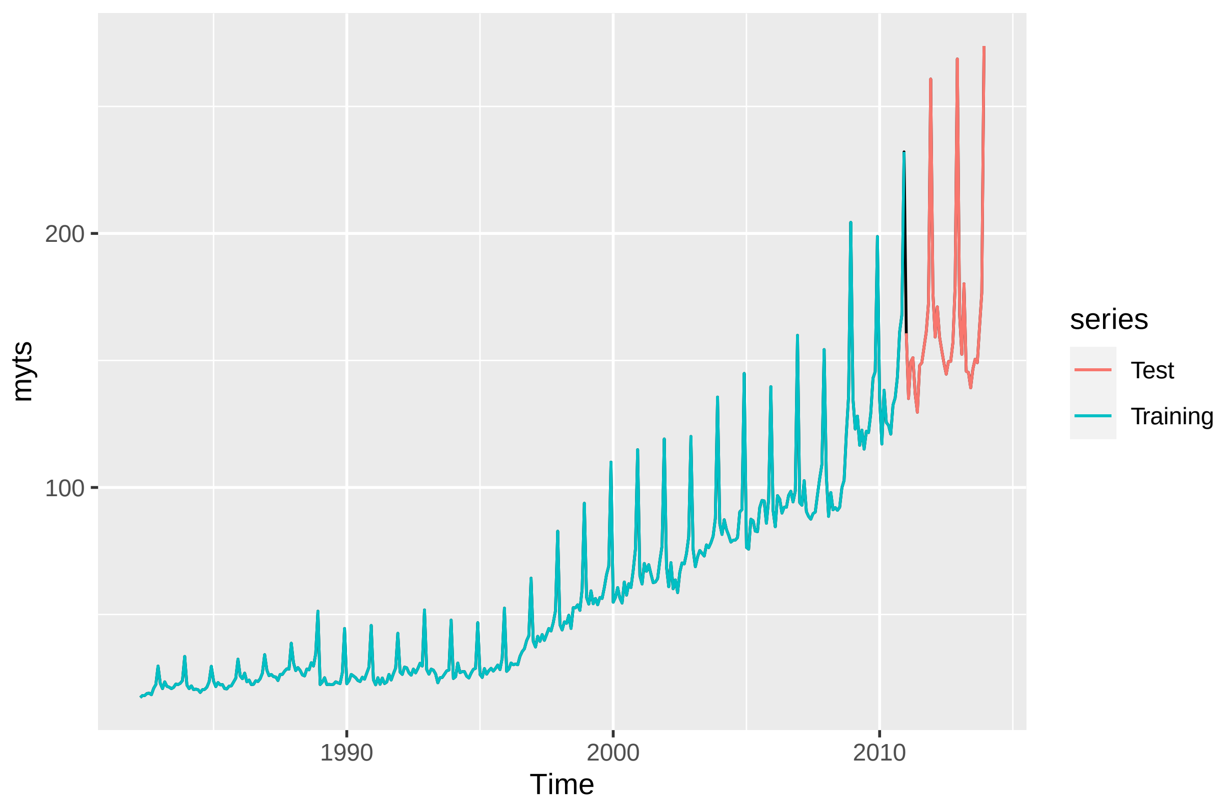
For your retail time series (from Exercise 3 in Section 2.10):

a. Split the data into two parts using

```
myts.train <- window(myts, end=c(2010,12))
myts.test <- window(myts, start=2011)
```

b. Check that your data have been split appropriately by producing the following plot.

```
autoplot(myts) +
  autolayer(myts.train, series="Training") +
  autolayer(myts.test, series="Test")
```



c. Calculate forecasts using `snaive` applied to `myts.train`.

```r
fc <- snaive(myts.train)
```

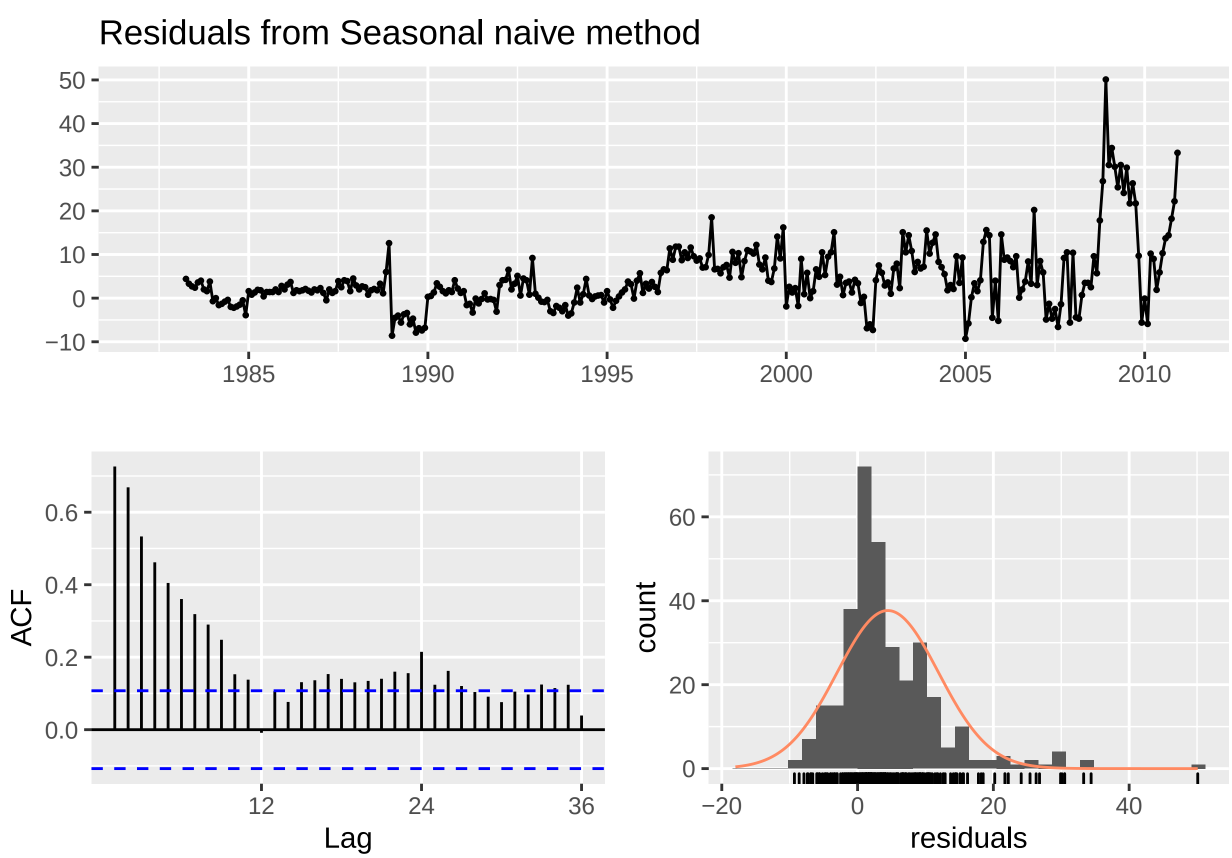
d. Compare the accuracy of your forecasts against the actual values stored in `myts.test`.

```r
accuracy(fc,myts.test)
```

```
##                     ME      RMSE       MAE      MPE      MAPE     MASE
## Training set  4.455255  8.699864  5.818619  6.15400  9.948117 1.000000
## Test set     19.170833 22.956217 19.520833 11.59039 11.813322 3.354891
##                   ACF1 Theil's U
## Training set 0.7261600        NA
## Test set     0.5801161 0.7479721
```

e. Check the residuals.

```r
checkresiduals(fc)
```




```
##
```

```
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 783.91, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

Do the residuals appear to be uncorrelated and normally distributed?

While there is a slight skew to the right on the residuals distribution plot, there seems to be strong significant of correlation in the lags of the ACF plot.

f. How sensitive are the accuracy measures to the training/test split?

```r
myts.train <- window(myts, end=c(2010,12))
myts.test <- window(myts, start=2011)

myts.train2 <- window(myts, end=c(2006,12))
myts.test2 <- window(myts, start=2007)

myts.train3 <- window(myts, end=c(2008,12))
myts.test3 <- window(myts, start=2009)

fc <- snaive(myts.train)
fc2 <- snaive(myts.train2)
fc3 <- snaive(myts.train3)

accuracy(fc, myts.test)
```

```
##                     ME      RMSE       MAE      MPE      MAPE     MASE
## Training set  4.455255  8.699864  5.818619  6.15400  9.948117 1.000000
## Test set     19.170833 22.956217 19.520833 11.59039 11.813322 3.354891
##                   ACF1 Theil's U
## Training set 0.7261600        NA
## Test set     0.5801161 0.7479721
```

```r
accuracy(fc2, myts.test2)
```

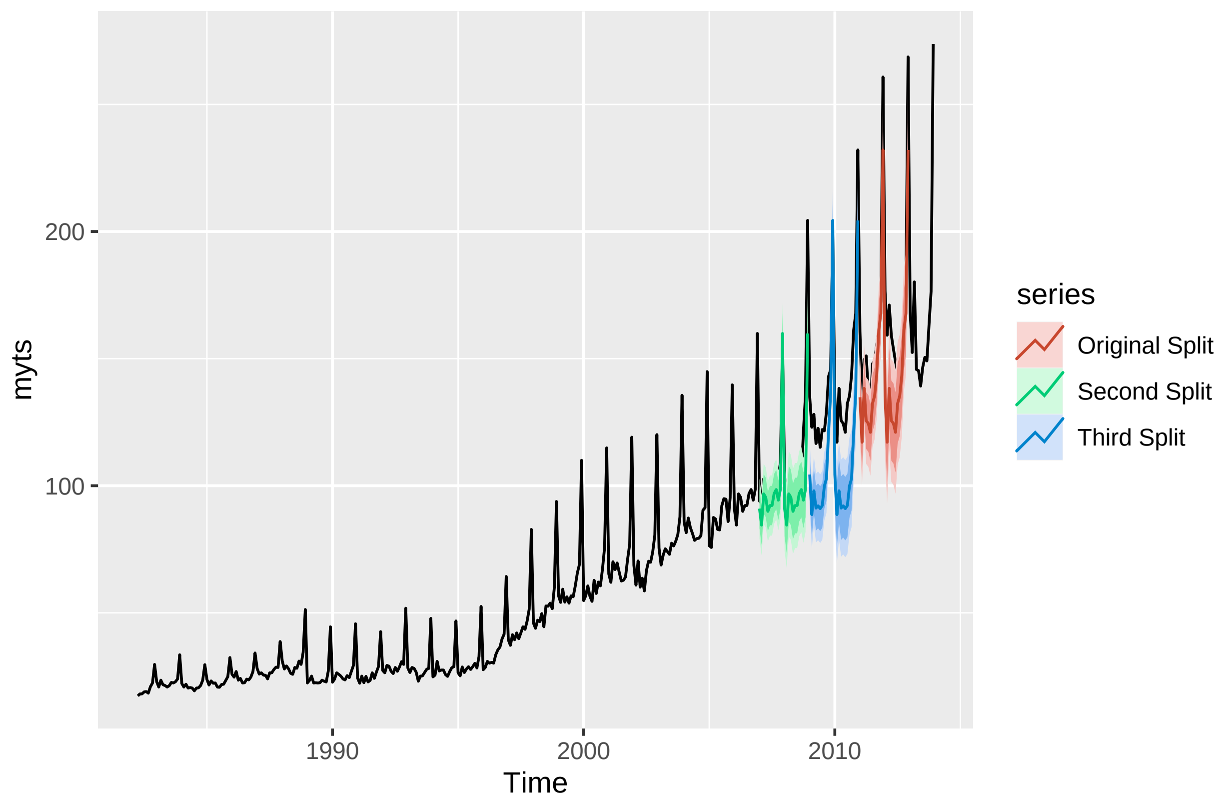
```
##                    ME      RMSE      MAE      MPE     MAPE     MASE      ACF1
## Training set 3.298947  6.037997 4.557193 5.784881 9.901433 1.000000 0.6028602
## Test set     5.904167 14.129800 8.604167 4.290980 7.069659 1.888041 0.5801372
##              Theil's U
## Training set        NA
## Test set     0.7066347
```

```r
accuracy(fc3, myts.test3)
```

```
##                     ME      RMSE       MAE       MPE      MAPE     MASE
## Training set  3.468608  6.893273  4.862783  5.641569  9.679051 1.000000
## Test set     28.770833 30.430926 29.237500 21.592543 21.827284 6.012503
```

```
##                    ACF1 Theil's U
## Training set 0.5475324        NA
## Test set     0.4978109  1.549617
```

```r
autoplot(myts) +
  autolayer(fc, series = "Original Split") +
  autolayer(fc2, series = "Second Split") +
  autolayer(fc3, series = "Third Split")
```



The second split created looks to be the best predicition in this example. I think given these various timeframes to predict against and the varied results, we can say that this accuracy mesaures between the training and test split are sensitive.