Model Diagnostics

So far, we've learned how to

- Transform non-stationary data to stationary data,
- After transformation, based on ACF and PACF, suggest (p,d,q) (and/or $(P, D, Q)_S$ if there is seasonal pattern),
- Estimation of parameters,
- Forecasting.

In this lecture, we will talk about

- Unit Root tests,
- Residual diagnostics,
- Evaluating Forecasting accuracy.

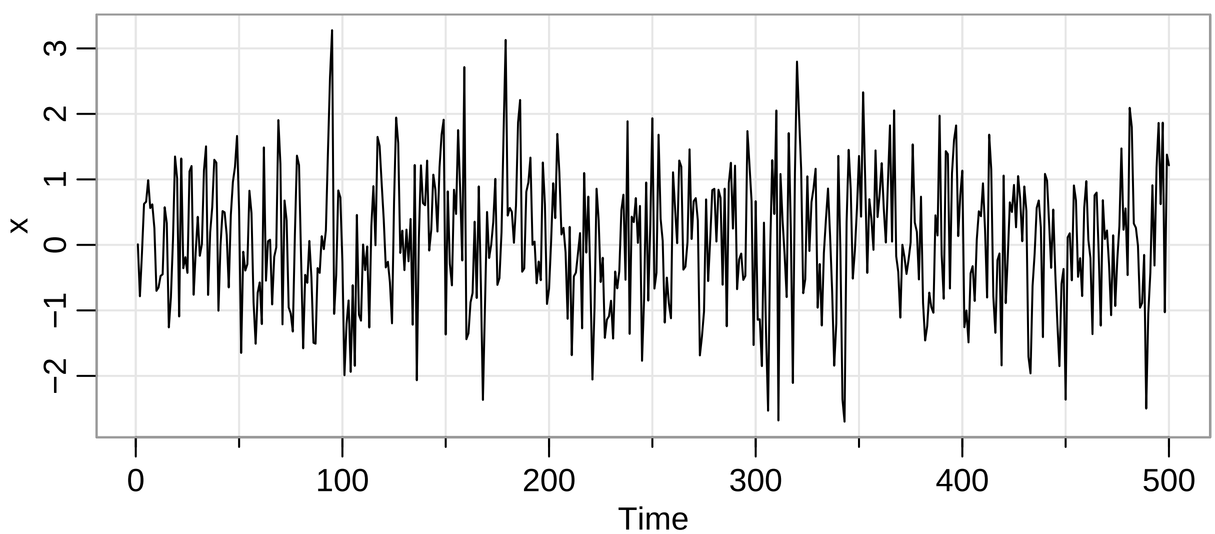
Unit root tests

- One way to determine more objectively whether differencing is required is to use a unit root test.
- These are statistical hypothesis tests of stationarity that are designed for determining whether differencing is required.
- A number of unit root tests are available, which are based on different assumptions and may lead to conflicting answers.
  - Dickey-Fuller Test (1979), Augmented Dickey-Fuller Test (adf test)
  - Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test (Kwiatkowski, Phillips, Schmidt, & Shin, 1992)

(Augmented) Dickey-Fuller Test

- In this test, the null hypothesis is that the data are not stationary.
- Big p-values (e.g., bigger than 0.05) suggest that differencing is required.

```
#library(tseries)
set.seed(429)
x <- arima.sim(list(order = c(1,0,0),ar = 0.2),n = 500)
tsplot(x)
```




```
adf.test(x)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  x
## Dickey-Fuller = -7.7656, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```r
set.seed(42900)
wn=rnorm(500)
adf.test(wn)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  wn
## Dickey-Fuller = -7.5289, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```r
wn.drift=rnorm(500)+1:500
adf.test(wn.drift)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  wn.drift
## Dickey-Fuller = -8.0214, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```r
# adf.test() uses a model that allows an intercept and trend.
rw=cumsum(wn)
adf.test(rw)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  rw
## Dickey-Fuller = -2.3119, Lag order = 7, p-value = 0.4463
## alternative hypothesis: stationary
```

KPSS Test

- In this test, the null hypothesis is that the data are stationary, and we look for evidence that the null hypothesis is false.
- Consequently, small p-values (e.g., less than 0.05) suggest that differencing is required.

```r
kpss.test(wn)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  wn
## KPSS Level = 0.13232, Truncation lag parameter = 5, p-value = 0.1
```

```r
kpss.test(wn.drift)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  wn.drift
## KPSS Level = 8.4311, Truncation lag parameter = 5, p-value = 0.01
```

```
kpss.test(wn.drift,null="Trend")
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  wn.drift
## KPSS Trend = 0.081951, Truncation lag parameter = 5, p-value = 0.1
```

```
kpss.test(rw)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  rw
## KPSS Level = 6.0342, Truncation lag parameter = 5, p-value = 0.01
```

```
kpss.test(rw,null='Trend')
```

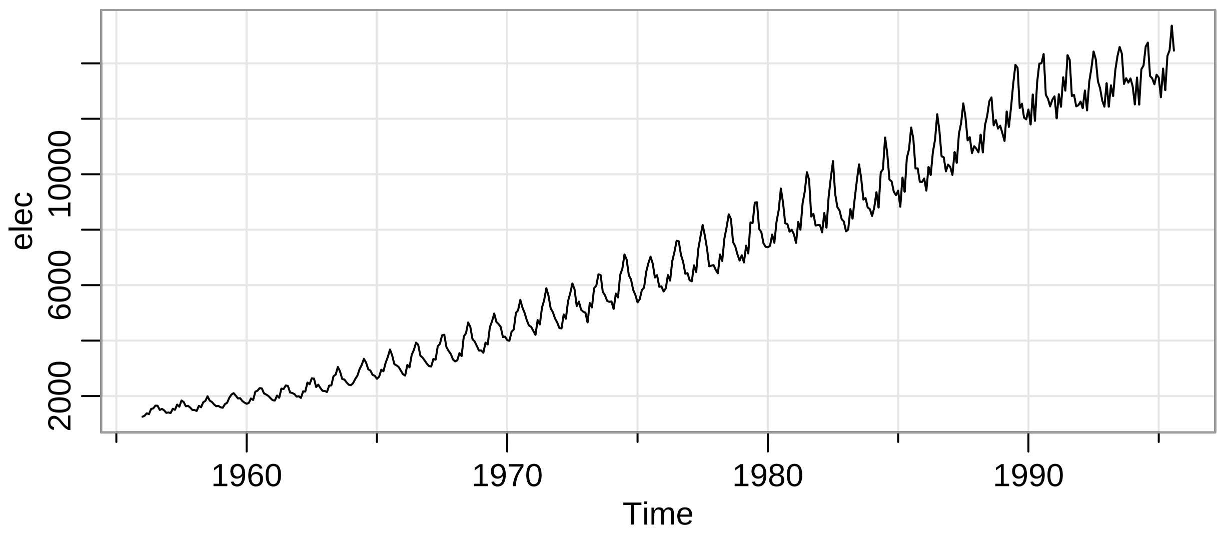
```
##
##  KPSS Test for Trend Stationarity
##
## data:  rw
## KPSS Trend = 1.1658, Truncation lag parameter = 5, p-value = 0.01
```

Power Transformation

$$Y_t = \begin{cases} (X_t^\lambda - 1)/\lambda & \lambda \neq 0, \\ \log X_t & \lambda = 0, \end{cases} \tag{1}$$

Power Transformation Example

```
#library(fpp2)
tsplot(elec)
```

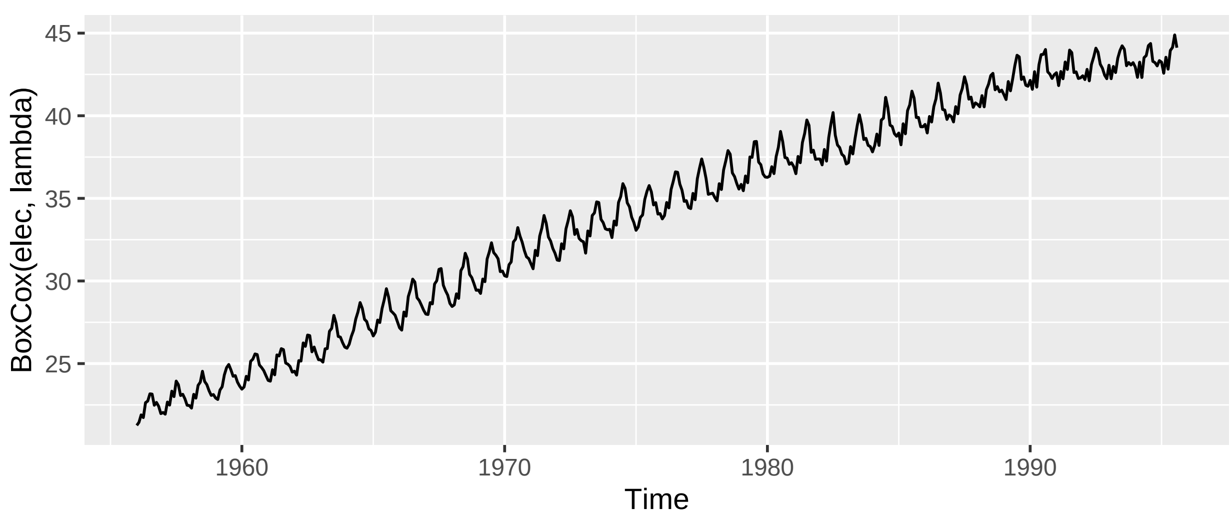



```
(lambda <- BoxCox.lambda(elec))
```

```
## [1] 0.2654076
```

Power Transformation Example

```
autoplot(BoxCox(elec,lambda))
```



More reading: https://otexts.com/fpp2/transformations.html

Residual diagnostics

Reading:

- [TSA4] Chapter 3.7
- https://otexts.com/fpp2/residuals.html

Desirable properties of residuals:

- The residuals are uncorrelated. If there are correlations between residuals, then there is information left in the residuals which should be used in computing forecasts.
- The residuals have zero mean. If the residuals have a mean other than zero, then the forecasts are biased.
- The residuals have constant variance.
- The residuals are normally distributed.

Portmanteau tests for autocorrelation

Box-Pierce Test

$$Q = T\sum_{k=1}^{h}\hat{\rho}^2(k)$$

- h: maximum lag being considered
- T: number of observations

Suggestion for h:

- $h = 10$ for non-seasonal data
- $h = 2s$ for seasonal data (s: seasonality)
- Test is not good when h is large, suggest $h \leq T/5$

Ljung-Box Test

$$Q^* = T(T+2)\sum_{k=1}^{h}(T-k)^{-1}\hat{\rho}^2(k)$$

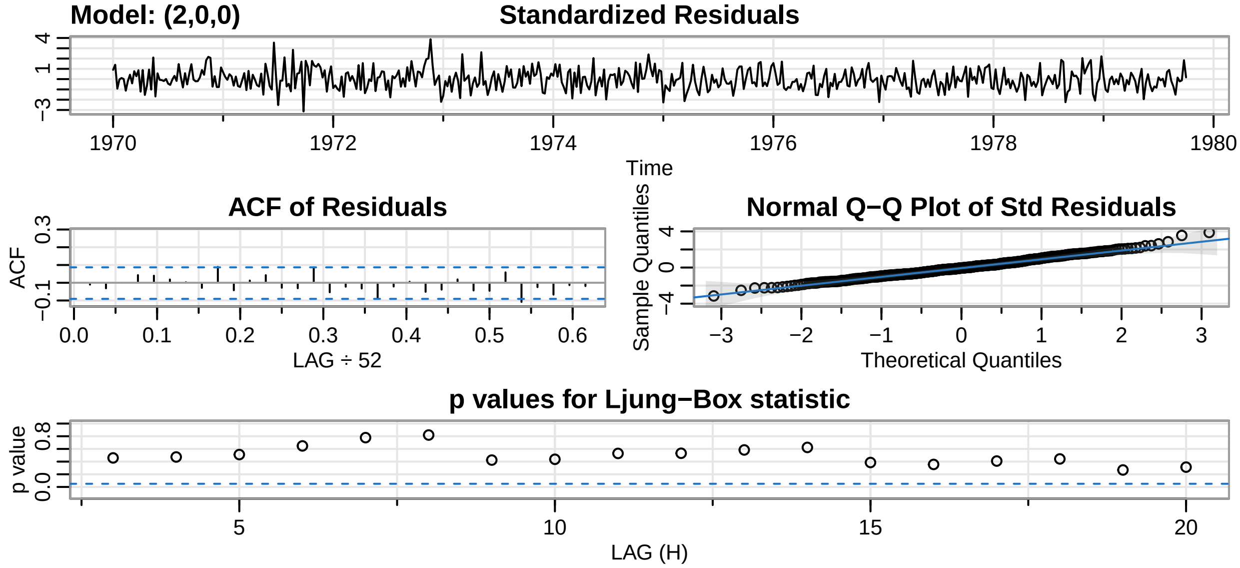
```
#library(astsa)
model.ar.2=arima(cmort,order=c(2,0,0))
res=residuals(model.ar.2)
Box.test(res, lag=10, fitdf=0)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 7.8207, df = 10, p-value = 0.6464
```

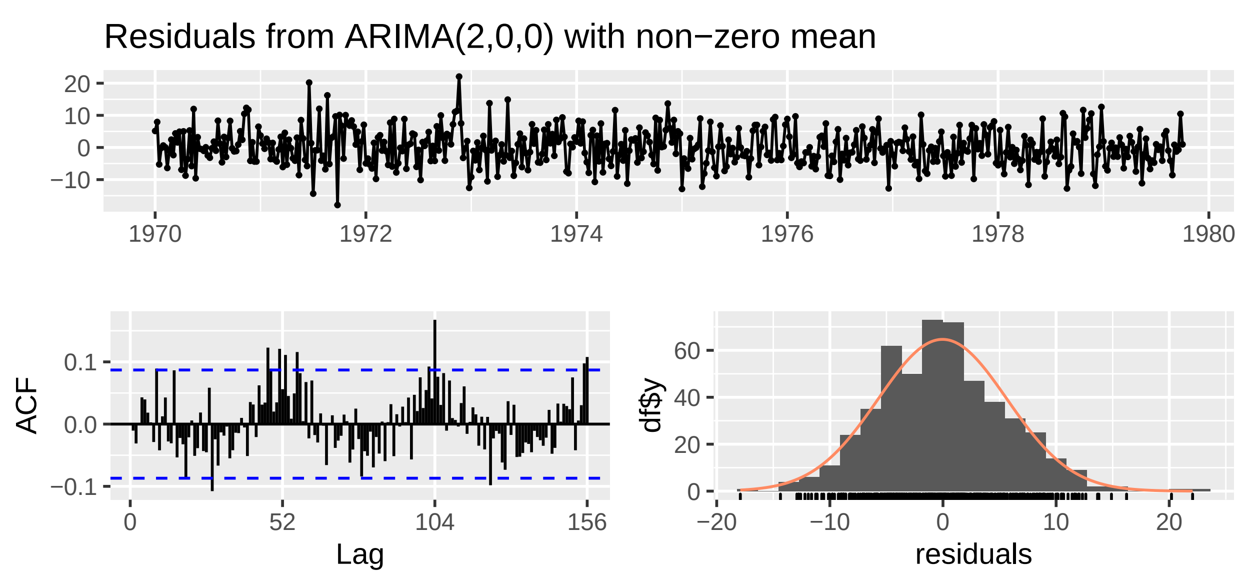
```
Box.test(res,lag=10, fitdf=0, type="Lj")
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 7.9691, df = 10, p-value = 0.6319
```

```
sarima(cmort,p=2,d=0,q=0)
```




```
model.ar2=Arima(cmort,order=c(2,0,0))
checkresiduals(model.ar2)
```




```
##
```

```
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,0) with non-zero mean
## Q* = 138.34, df = 100, p-value = 0.006724
##
## Model df: 2.   Total lags used: 102
```

Evaluating Forecast Accuracy

Reading

https://otexts.com/fpp2/forecasting-on-training-and-test-sets.html

Train and Test sets

- Training: 80% of total sample
- Test: 20% of total sample
- Proportions depends on how long the sample is and how far ahead you want to forecast.
- test set should (ideally) be at least as large as the maximum forecast horizon required.

Note that

- A model which fits the training data well will not necessarily forecast well.
- A perfect fit can always be obtained by using a model with enough parameters.
- Over-fitting a model to data is just as bad as failing to identify a systematic pattern in the data

```
total.length=length(cmort)
test.length=10
train.length=total.length-test.length
cmort.test=subset(cmort,start=train.length)
cmort.train=subset(cmort,end=train.length-1)
```

Forecast Errors:

Difference between an observed value and its forecast. Notation: e_t

Difference between forecast errors and residuals:

- Residuals are from the training set
- Forecast errors are calculated on the test set.
- Residuals are based on one-step forecasts
- Forecast errors can involve multi-step forecasts.

Scale-dependent errors

Mean Absolute Error (MAE): $mean(|e_t|)$

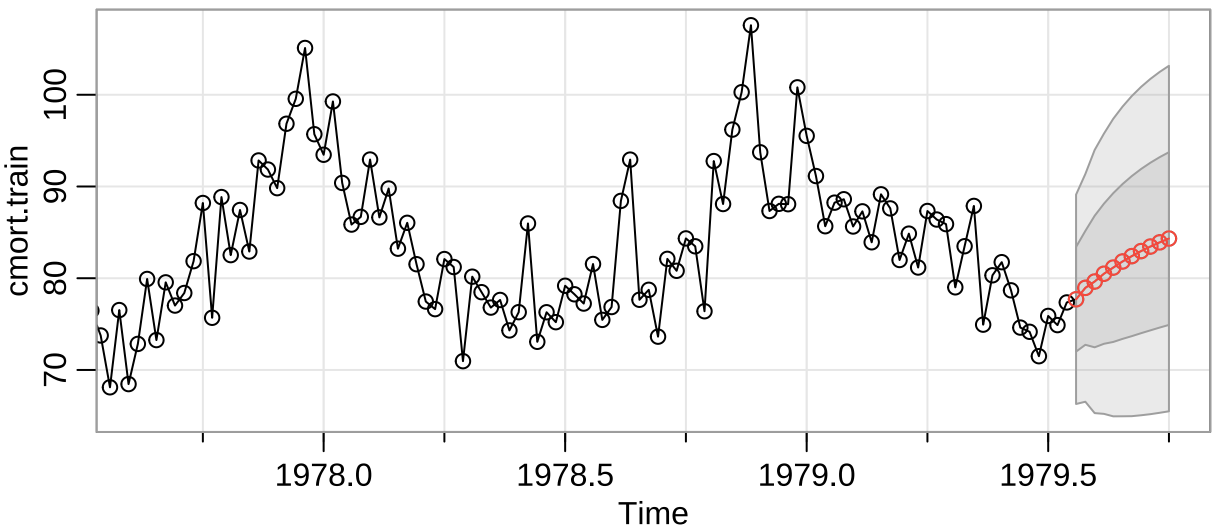
Root mean squared error (RMSE): $\sqrt{mean(e_t^2)}$

Scale-independent errors

Mean Absolute Percentage Error (MAPE): $mean(|p_t|), \quad p_t = 100e_t/y_t$

R code for forecasting evaluation

```
library(fpp2)
model.ar2.forecast=sarima.for(cmort.train,p=2,d=0,q=0,n.ahead = length(cmort.test))
```




```
accuracy(object=model.ar2.forecast$pred,x=cmort.test)
```

```
##                 ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -1.229867 4.510824 3.965355 -1.817486 5.022879 0.3077932 0.8360721
```