Applied Time Series Analysis
ARIMA

Prof. Dr. Stephan Trahasch

Offenburg University of Applied Sciences

Outline

- Stationarity and differencing
- Non-seasonal ARIMA models
- Estimation and order selection
- ARIMA modelling in R
- Forecasting
- Seasonal ARIMA models
- ARIMA vs ETS

Stochastic Model for Time Series

A time series process is a set $\{X_t, t \in T\}$ of random variables, where T is the set of times. Each of the random variables $X_t, t \in T$ has a univariate probability distribution F_t.
$P[X_1 \leq x_1, \dots, X_t \leq x_t]$ for all t and $x_1, \dots, x_t$

- If we exclusively consider time series processes with equidistant time intervals, we can enumerate $\{T = 1,2,3, \dots\}$
- An observed time series is a realization of $X = (X_1, \dots, X_n)$ and is denoted with small letters as $\mathrm{x} = (x_1, \dots, x_n)$.
- We have a multivariate distribution, but only 1 observation (i.e. 1 realization from this distribution) is available. In order to perform "statistics", we require some additional structure.

Strict Stationarity

For being able to do statistics with time series, we require that the series "doesn't change its probabilistic character" over time. Shifting the time axis does not affect the distribution.

A time series $\{X_t, t \in T\}$ is **strictly stationary**, if the joint distribution of the random vector $\{X_t, \dots, X_{t+h}\}$ is equal to the one of $\{X_s, \dots, X_{s+h}\}$ for all combinations of t, s and h.

$X_t \sim F$	all X_t are identically distributed
$E[X_t] = \mu$	all X_t have identical expected value
$Var(X_t) = \sigma^2$	all X_t have identical variance
$Cov(X_t, X_{t+h}) = \gamma_h$	the autocov depends only on the lag h

A stationary series is:
roughly horizontal, constant variance, no patterns predictable in the long-term

Stationarity

It is impossible to „prove“ the theoretical concept of stationarity from data. We can only search for evidence in favor or against it.

However, with strict stationarity, even finding evidence only is too difficult. We thus resort to the concept of weak stationarity.

A time series $\{X_t, t \in T\}$ is said to be **weakly stationary**, if

$$E[X_t, t \in T] = \mu$$

$$Cov(X_t, X_{t+h}) = \gamma_h \text{ for all lags h}$$

$$Var(X_t) = \sigma^2.$$

Note that weak stationarity is sufficient for „practical purposes“.

Testing Stationarity

- In time series analysis, we need to verify whether the series has arisen from a stationary process or not.
  Be careful: stationarity is a property of the process, and not of the data.
- Treat stationarity as a hypothesis! We may be able to reject it when the data strongly speak against it. However, we can never prove stationarity with data. At best, it is plausible.
- Formal tests for stationarity do exist. We discourage their use due to their low power for detecting general non-stationarity, as well as their complexity.

→Use the time series plot for deciding on stationarity!

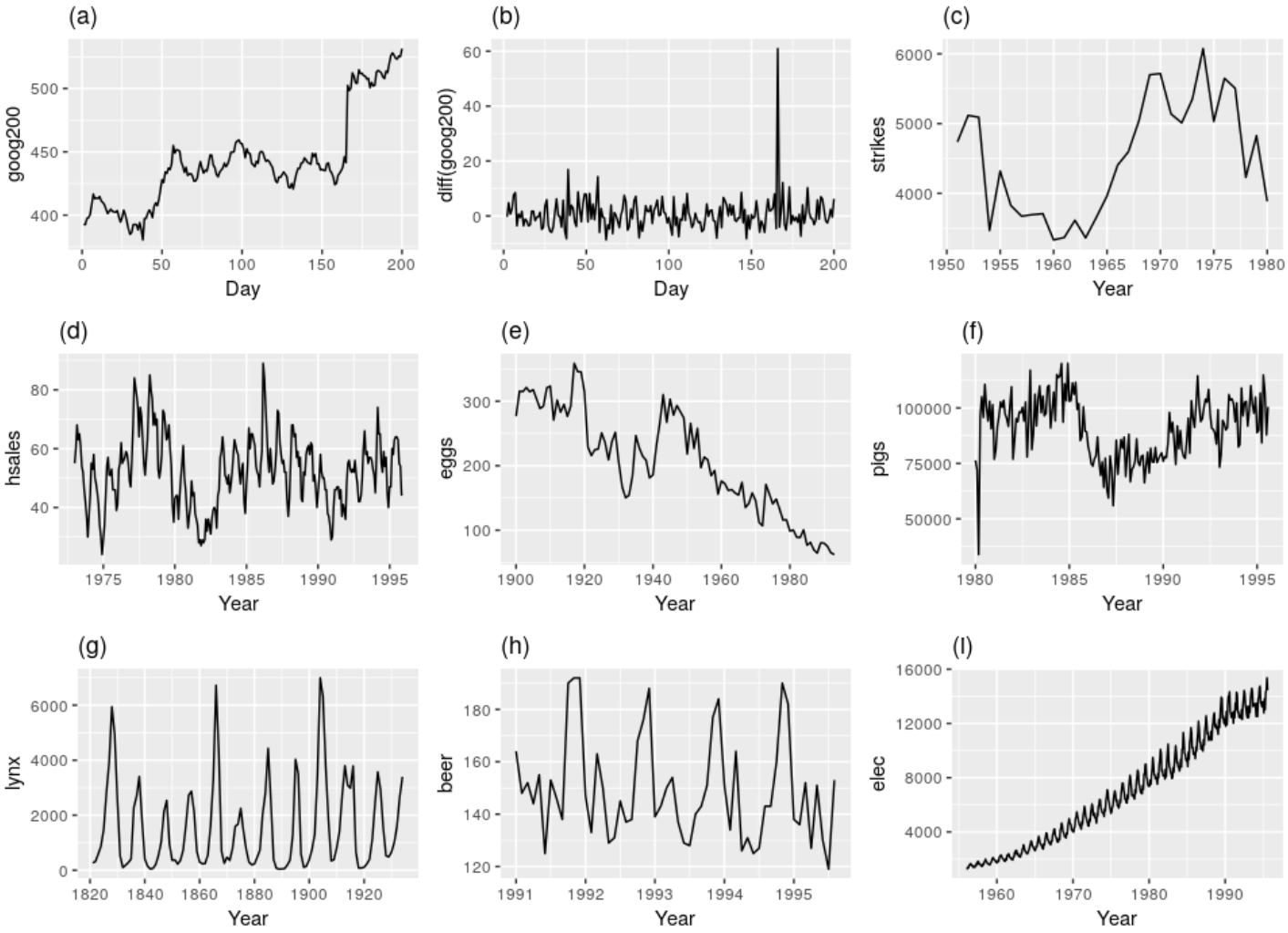
Evidence for Non-Stationarity

- Trend, i.e. non-constant expected value
- Seasonality, i.e. deterministic, periodical oscillations
- Non-constant variance, i.e. multiplicative error
- Non-constant dependency structure

Remark:

Note that some periodical oscillations, as for example in the lynx trappings data, can be stochastic and thus, the underlying process is assumed to be stationary. However, the boundary between the two is fuzzy.

Stationarity?



Stationary

If $\{y_t\}$ is a stationary time series, then for all s, the distribution of $(y_t, \dots, y_{t+s})$ does not depend on t.

Transformations help to stabilize the variance.

For ARIMA modelling, we also need to stabilize the mean.

Strategies for Detecting Non-Stationarity

- Time series plot
  - non-constant expected value (trend/seasonal effect)
  - changes in the dependency structure
  - non-constant variance
- Correlogram
  - non-constant expected value (trend/seasonal effect)
  - changes in the dependency structure

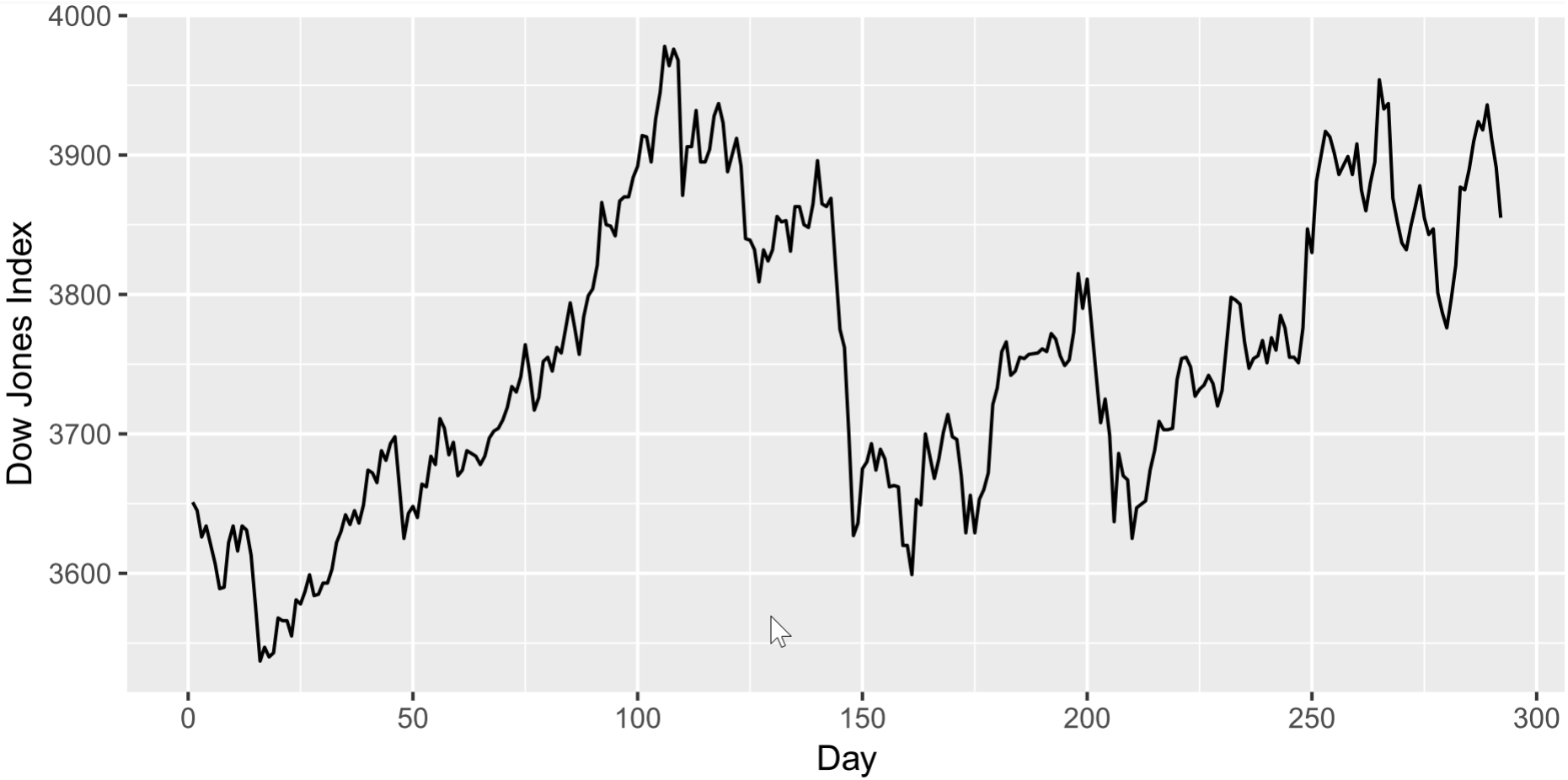
A (sometimes) useful trick, especially when working with the correlogram, is to split up the series in two or more parts, and producing plots for each of the pieces separately.

Non-stationarity in the mean

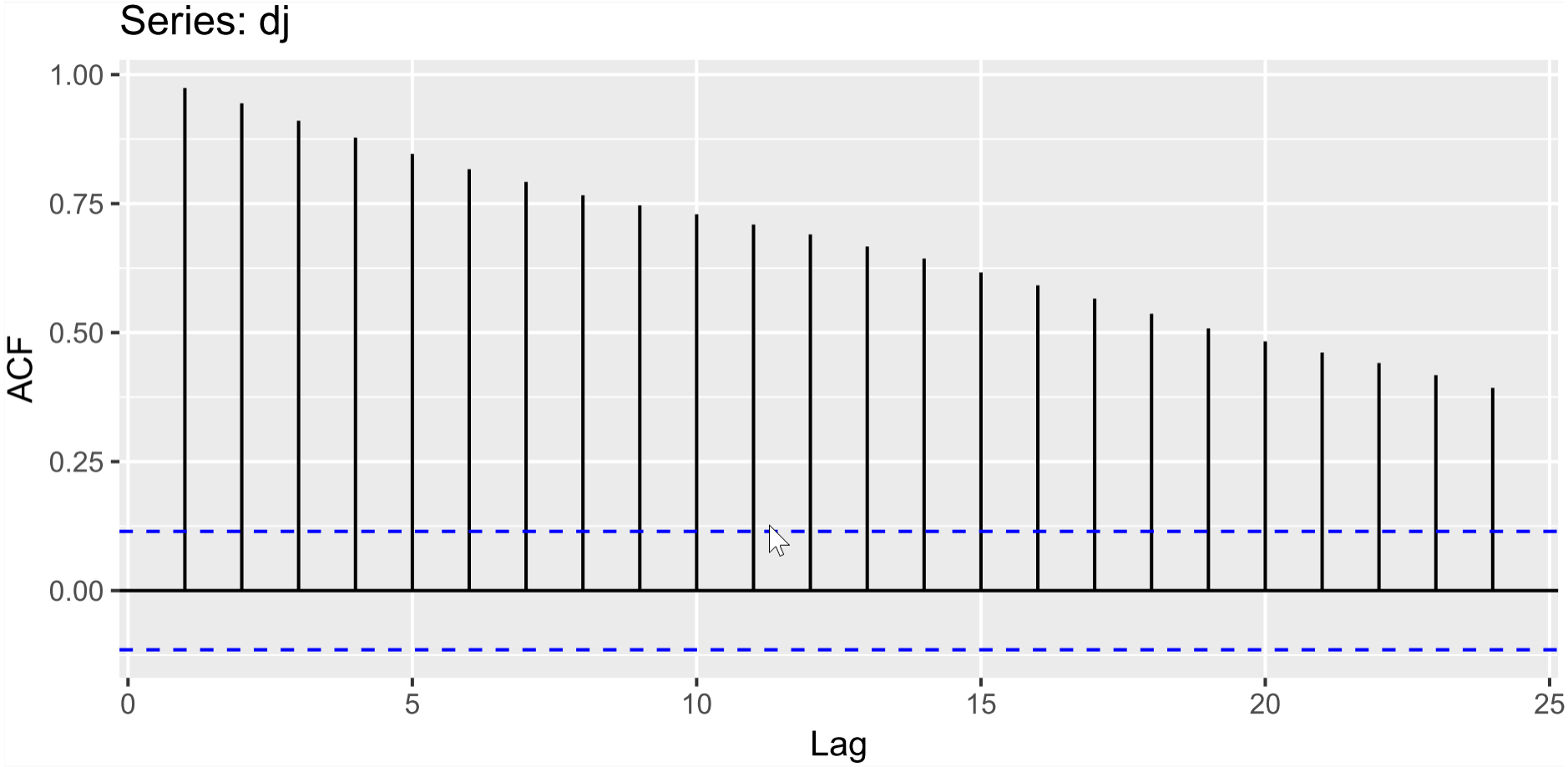
Identifying non-stationary series

- Time plot
- The ACF of stationary data drops to zero relatively quickly
- The ACF of non-stationary data decreases slowly.
- For non-stationary data, the value of r_1 is often large and positive.

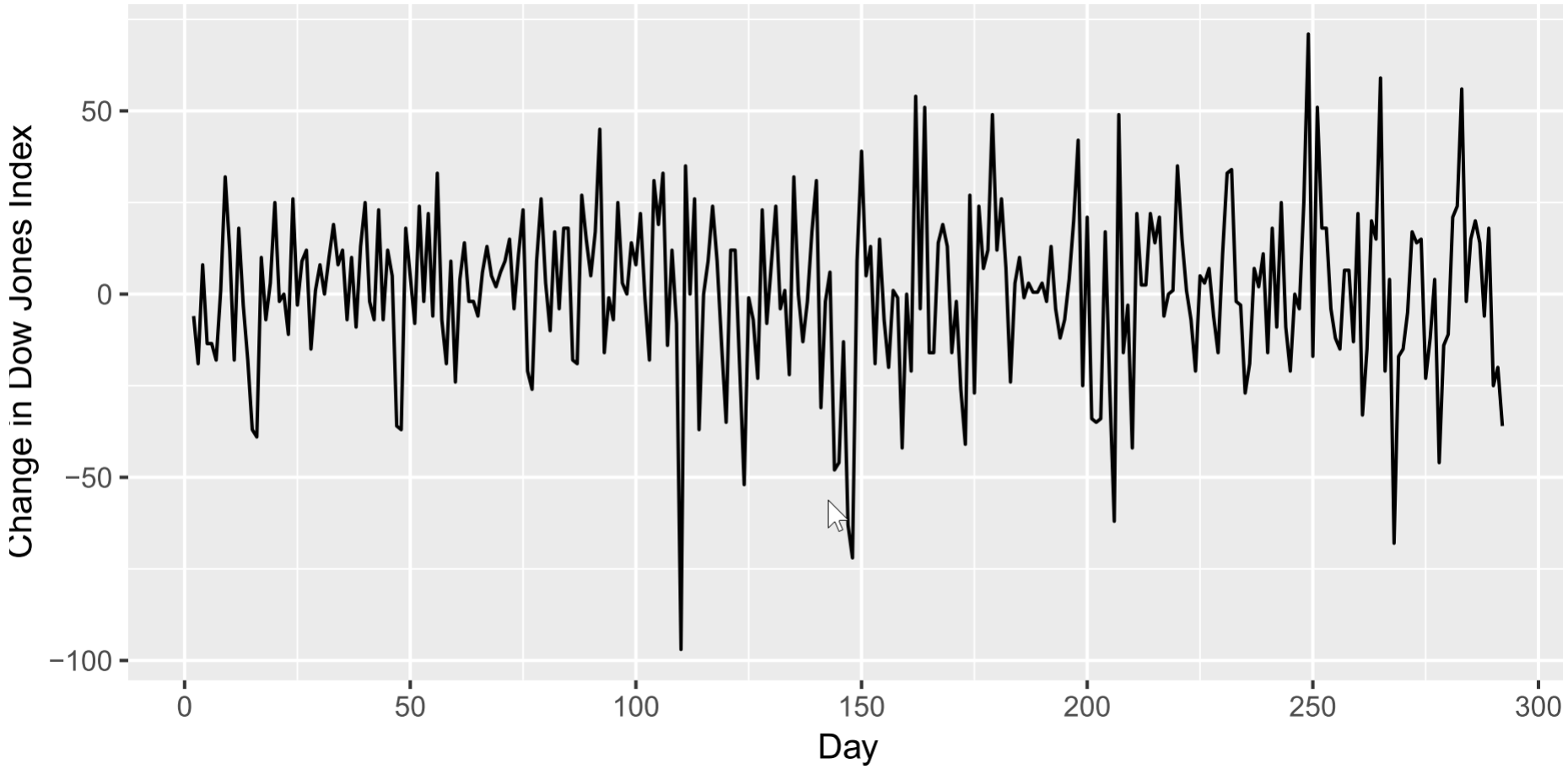
Example: Dow-Jones index



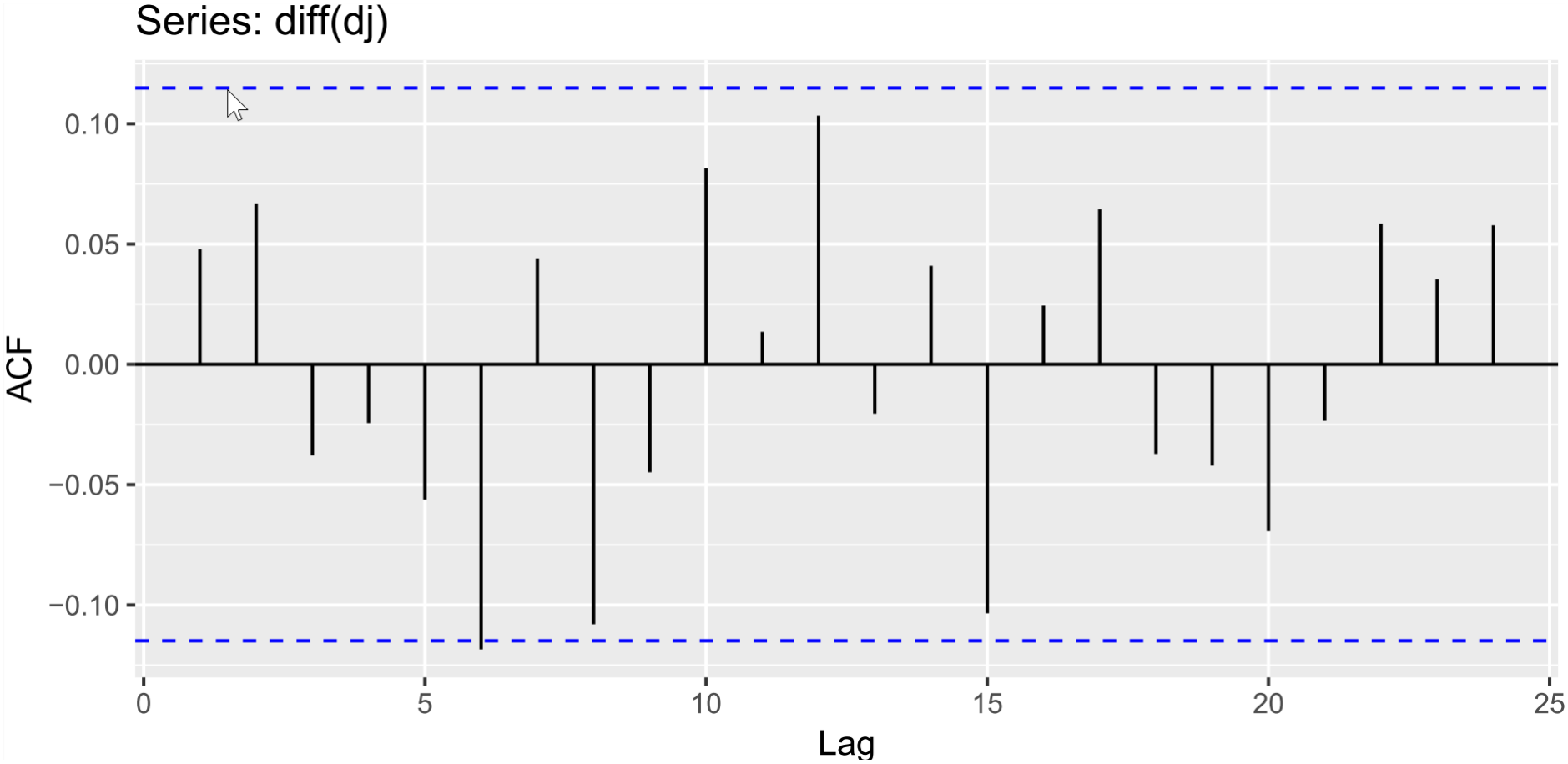
Example: Dow-Jones index



Example: Dow-Jones index



Example: Dow-Jones index



Differencing

- Differencing helps to stabilize the mean.
- The differenced series is the between each observation in the original series: $y'_t = y_t - y_{t-1}$.
- The differenced series will have only $T - 1$ values since it is not possible to calculate a difference y_1' for the first observation.

Second-order differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time:

$$\begin{aligned} y_t'' &= y_t' - y_{t-1}' \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2} \end{aligned}$$

- y_t'' will have $T - 2$ values.
- In practice, it is almost never necessary to go beyond second-order differences.

Seasonal differencing

A seasonal difference is the difference between an observation and the corresponding observation from the previous year.

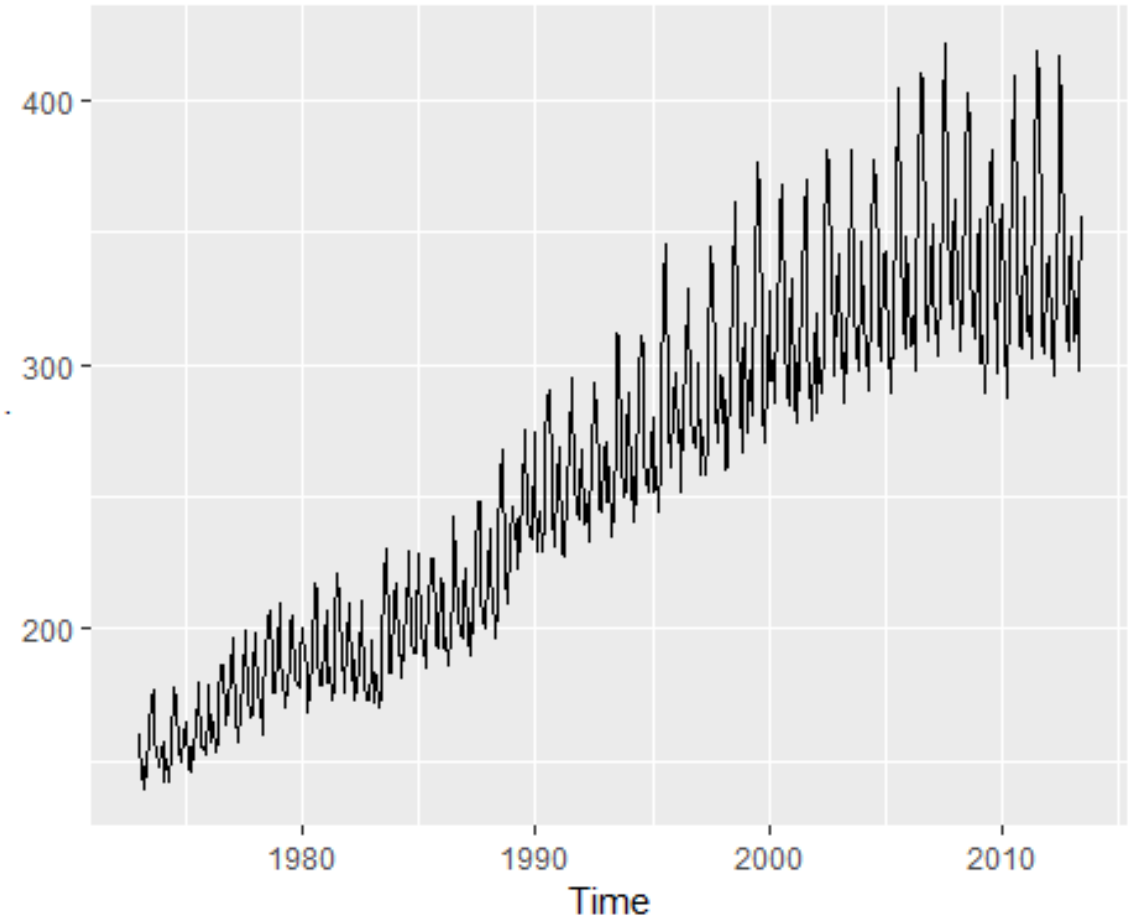
$$y'_t = y_t - y_{t-m}$$

where $m =$ number of seasons.

- For monthly data $m = 12$.
- For quarterly data $m = 4$.

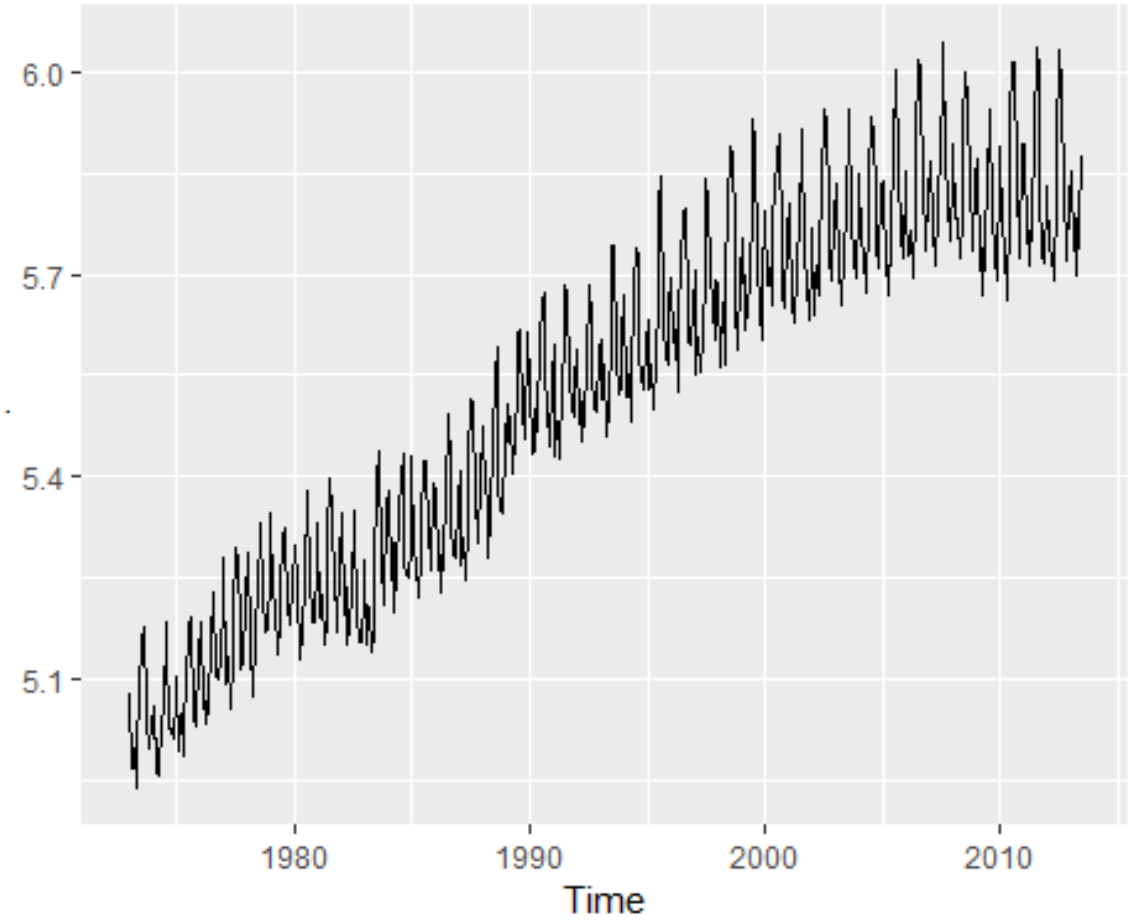
Electricity production

```
usmelec %>% autoplot()
```



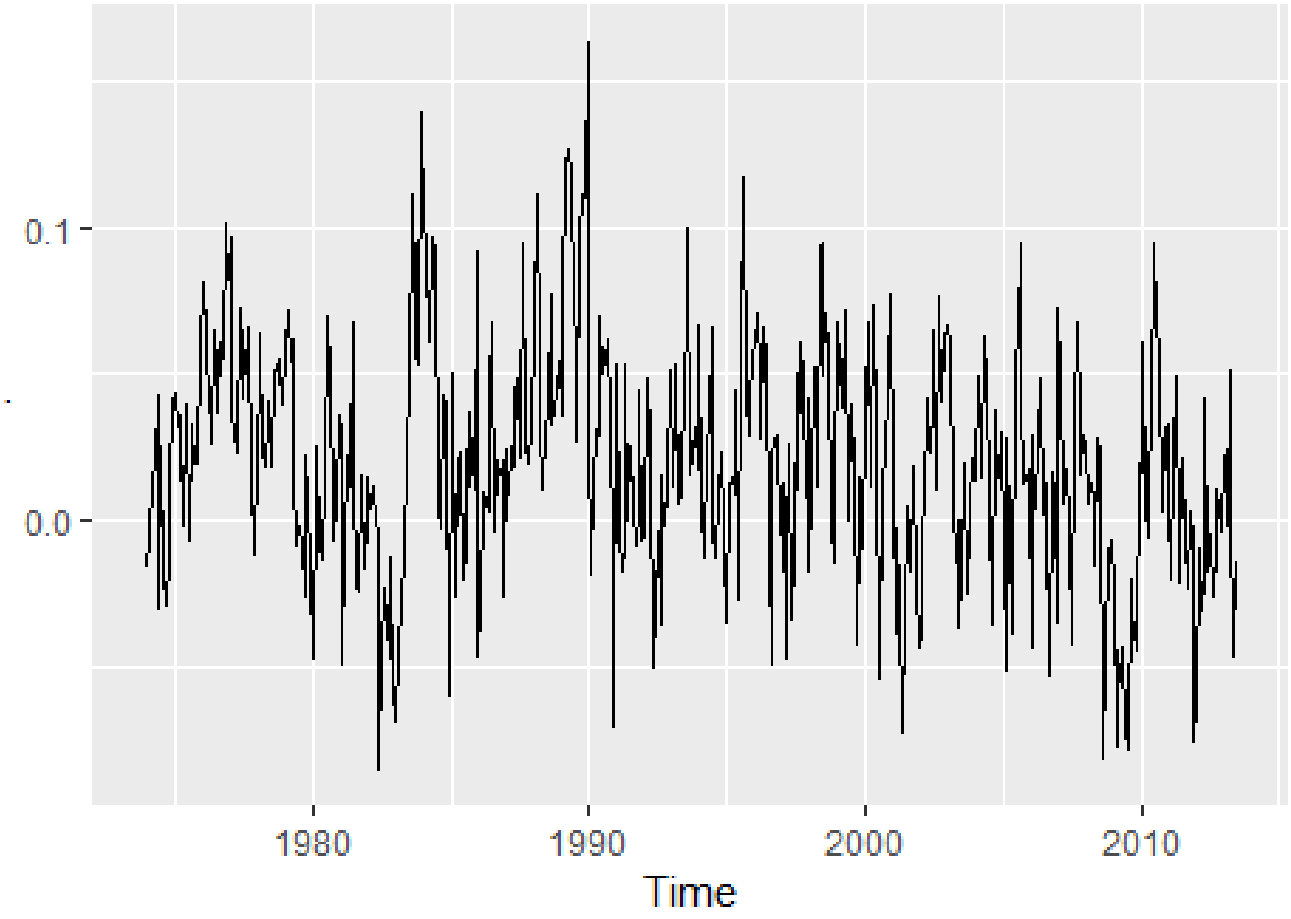
Electricity production

```
usmelec %>% log() %>% autoplot()
```



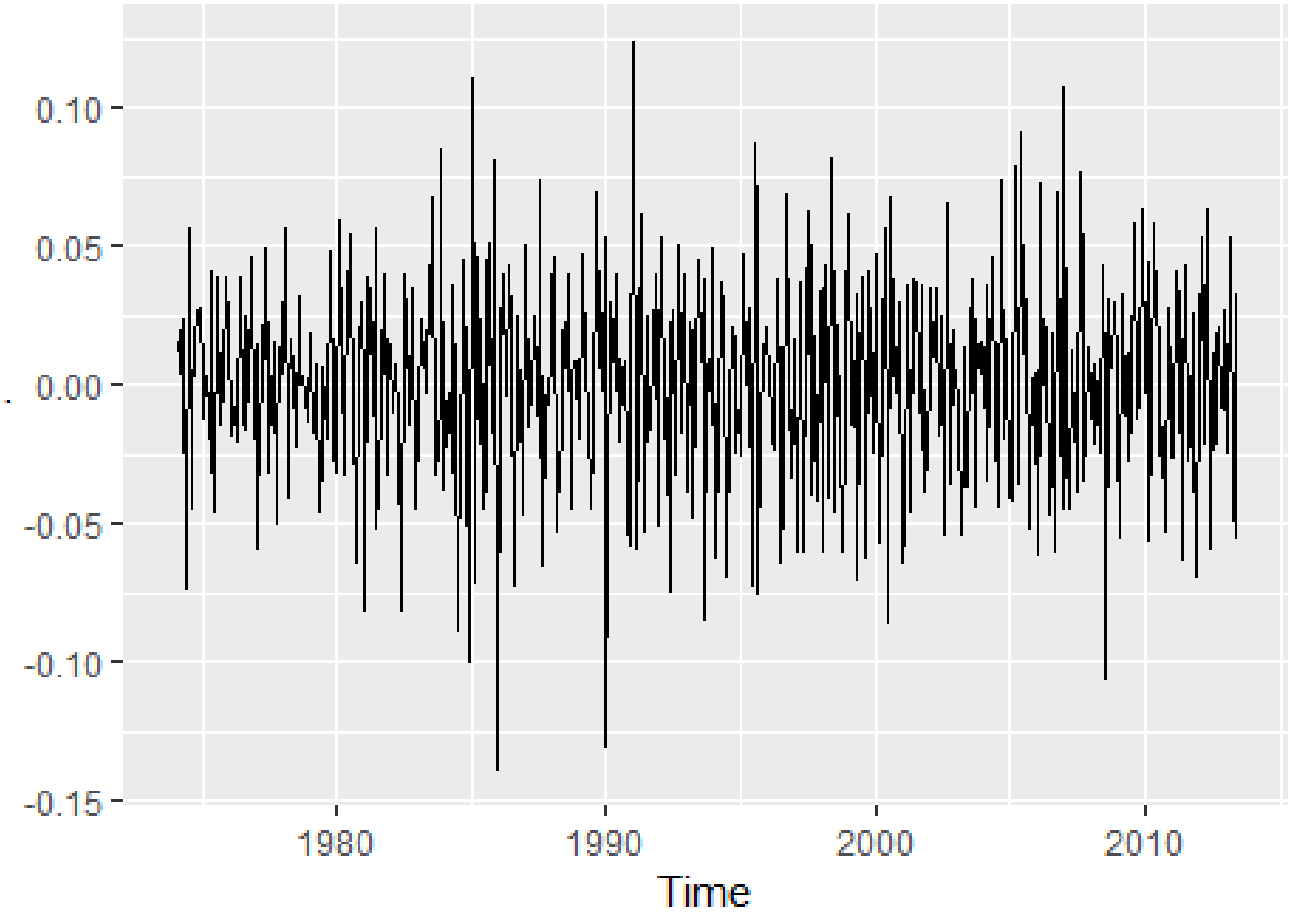
Electricity production

```
usmelec %>% log() %>% diff(lag=12) %>% autoplot()
```



Electricity production

```
usmusmelec %>% log() %>% diff(lag=12) %>% diff(lag=1) %>% autoplot()
```



Electricity production

- Seasonally differenced series is closer to being stationary.
- Remaining non-stationarity can be removed with further first difference.

If $y'_t = y_t - y_{t-12}$ denotes seasonally differenced series, then twice-differenced series i

$$
\begin{aligned}
y_t^* &= y'_t - y'_{t-1} \\
&= (y_t - y_{t-12}) - (y_{t-1} - y_{t-13}) \\
&= y_t - y_{t-1} - y_{t-12} + y_{t-13} \cdot
\end{aligned}
$$

Seasonal differencing

When both seasonal and first differences are applied

- it makes no difference which is done first—the result will be the same.
- If seasonality is strong, we recommend that seasonal differencing be done first because sometimes the resulting series will be stationary and there will be no need for further first difference.

It is important that if differencing is used, the differences are interpretable.

Interpretation of differencing

- First differences are the change between one observation and the next;
- seasonal differences are the change between one year to the next.

But taking lag 3 differences for yearly data, for example, results in a model which cannot be sensibly interpreted.

Unit root tests

Statistical tests to determine the required order of differencing.

1. Augmented Dickey Fuller test:
   null hypothesis is that the data are non-stationary and non-seasonal.
2. Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test:
   null hypothesis is that the data are stationary and non-seasonal.
3. Other tests available for seasonal data.

KPSS test

```
library(urca)
summary(ur.kpss(goog))
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 10.7223
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
ndiffs(goog)
## [1] 1
```

Automatically selecting differences

STL decomposition: $y_t = T_t + S_t + R_t$

Seasonal strength $F_s = \max(0,1 - \frac{\mathrm{Var}(R_t)}{\mathrm{Var}(S_t+R_t)})$

If $F_s > 0.64$,do one seasonal difference

```
usmelec %>% log() %>% nsdiffs()
## [1] 1
usmelec %>% log() %>% diff(lag=12) %>% ndiffs()
## [1] 1
```

Backshift notation

A very useful notational device is the backward shift operator, B, which is used as follows:

$$By_t = y_{t-1}$$

In other words, B, operating on y_t ,has the effect of **shifting the data back one period.** Two applications of B to y_t **shifts the data back two periods:**

$$B(By_t) = B^2y_t = y_{t-2}$$

For monthly data, if we wish to shift attention to „the same month last year“, then B^{12} is used, and the notation is $B^{12}y_t = y_{t-12}$.

Backshift notation

The backward shift operator is convenient for describing the process of . A first difference can be written as

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1 - B)y_t$$

Note that a first difference is represented by $(1 - B)$.

Similarly, if second-order differences (i.e., first differences of first differences) have to be computed, then:

$$y''_t = y_t - 2y_{t-1} + y_{t-2} = (1 - B)^2 y_t$$

Backshift notation

Second-order difference is denoted $(1-B)^2$.

is not the same as a , which would be denoted $1-B^2$;

In general, a dth-order difference can be written as

$$(1-B)^d y_t$$

A seasonal difference followed by a first difference can be written as

$$(1-B)(1-B^m)y_t$$

Backshift notation

The „backshift“ notation is convenient because the terms can be multiplied together to see the combined effect.

$$(1-B)(1-B^m)y_t = (1-\mathrm{B}-B^m+B^{m+1})\,y_t$$
$$= y_t - y_{t-1} - y_{t-m} + y_{t-m-1}$$

For monthly data, $m = 12$ and we obtain the same result as earlier.

Outline

- Stationarity and differencing
- Non-seasonal ARIMA models
- Estimation and order selection
- ARIMA modelling in R
- Forecasting
- Seasonal ARIMA models
- ARIMA vs ETS

There is a wealth of time series models

- AR autoregressive model
- MA moving average model
- ARMA combination of AR & MA
- ARIMA non-stationary ARMAs
- SARIMA seasonal ARIMAs

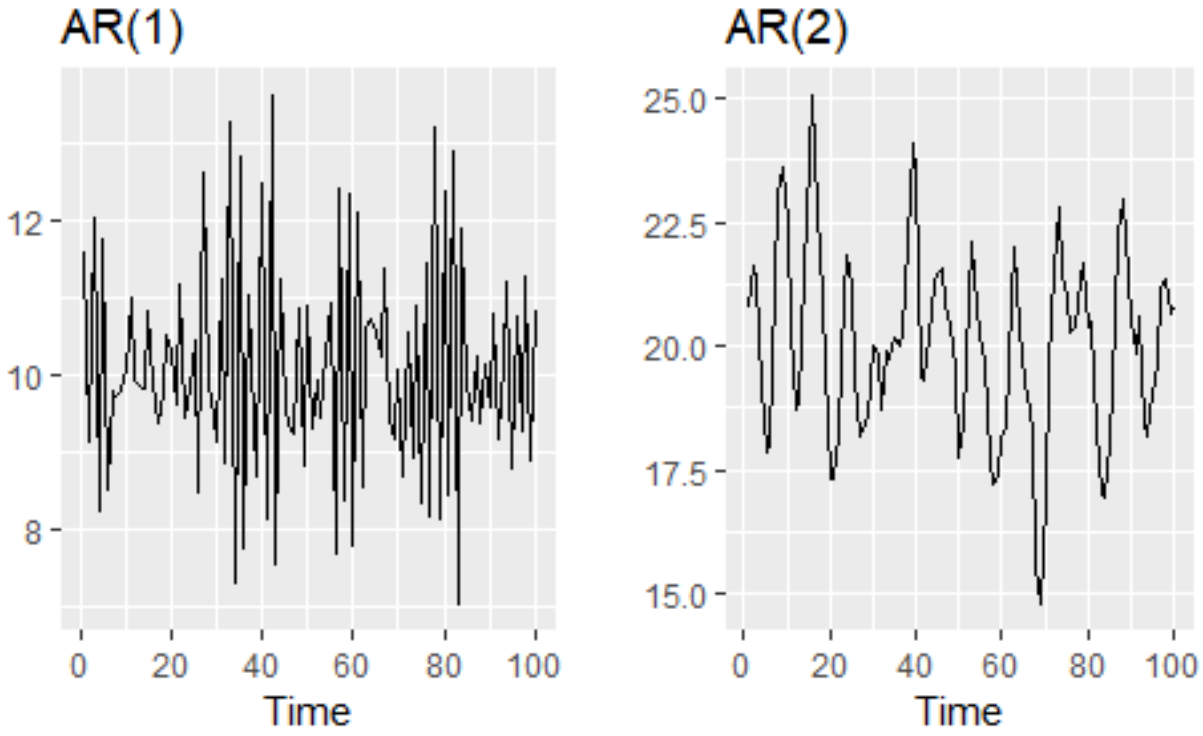
…

We start by discussing autoregressive models. They are perhaps the simplest and most intuitive time series models that exist.

Autoregressive models

$$y_t = c + \phi y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

Where ε_t is white noise. This is a multiple regression with lagged values of y_t as predictors.



The simplest model is the AR(1)-model

$$y_t = c + \phi y_{t-1} + \varepsilon_t$$

- When $\phi_1 = 0$,y_t is **equivalent to WN**
- When $\phi_1 = 1$ and $c = 0$, y_t is **equivalent to a RW**
- When $\phi_1 = 1$ and $c \neq 0$, y_t is **equivalent to a RW with drift**
- When $\phi_1 < 0$,y_t tends to **oscillate between positive and negative values**.

Stationarity conditions

We normally restrict autoregressive models to stationary data, and then some constraints on the values of the parameters are required.

General condition for stationarity:

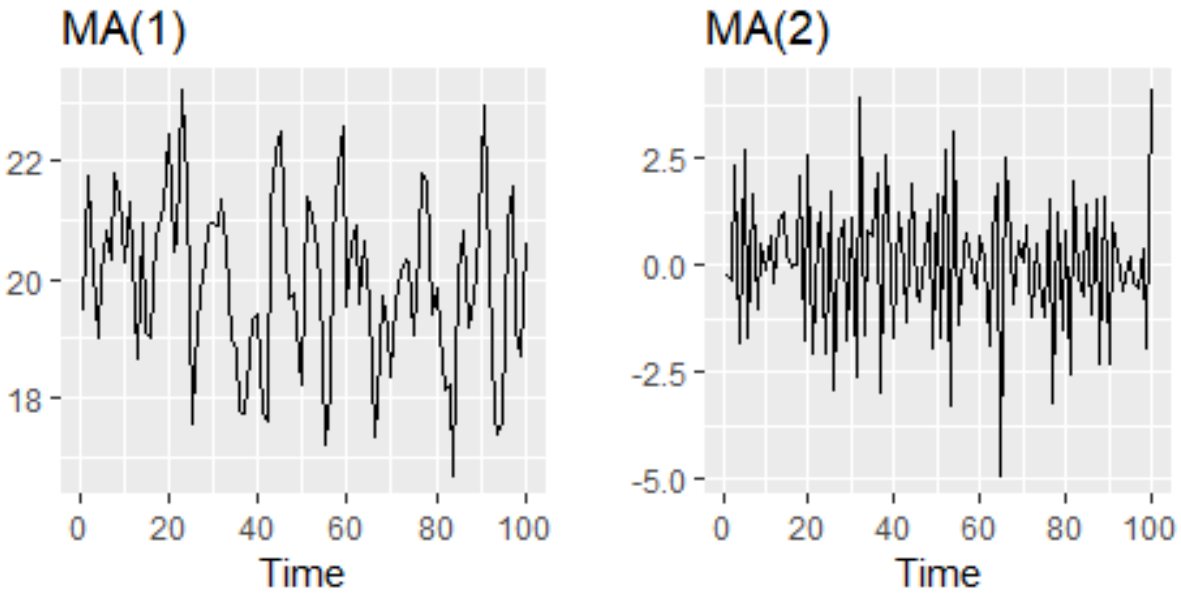
Complex roots of $1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p$ lie outside the unit circle on the complex plane.

- For $p = 1$: $-1 < \phi_1 < 1$.
- For $p = 2$: $-1 < \phi_2 < 1 \qquad \phi_2 + \phi_1 < 1 \qquad \phi_2 - \phi_1 < 1$.
- More complicated conditions hold for $p \geq 3$.
- Estimation software takes care of this.

Moving Average (MA) models

$$y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} \ldots + \theta_q \epsilon_{t-q}$$

where ϵ_t is white noise. This is a multiple regression with past errors as predictors. Don't confuse this with moving average smoothing!



ARIMA models

Autoregressive Moving Average models:

$$y_t = c + \phi y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} \ldots + \theta_q \epsilon_{t-q} + \varepsilon_t$$

- Predictors include both **lagged values of y_t and lagged errors.**
- Conditions on coefficients ensure stationarity.
- Conditions on coefficients ensure invertibility.

Autoregressive Integrated Moving Average models

- Combine ARMA model with **differencing**.
- $(1 - B)^d y_t$ follows an ARMA model.

ARIMA models

Autoregressive Integrated Moving Average models

ARIMA(p, d, q) model

AR: p = order of the autoregressive part

I: d = degree of first differencing involved

MA: q = order of the moving average part.

- White noise model: ARIMA(0,0,0)
- Random walk: ARIMA(0,1,0) with no constant
- Random walk with drift: ARIMA(0,1,0) with
- AR(p): ARIMA(p,0,0)
- MA(q): ARIMA(0,0,q)

Backshift notation for ARIMA

ARMA model:

$$y_t = c + \phi_1 B y_t + \cdots + \phi_p B^p y_t + \varepsilon_t + \theta_1 B \varepsilon_t + \cdots + \theta_q B^q \varepsilon_t$$

or $$(1 - \phi_1 B - \cdots - \phi_p B^p) y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q) \varepsilon_t$$

ARIMA(1,1,1) model:

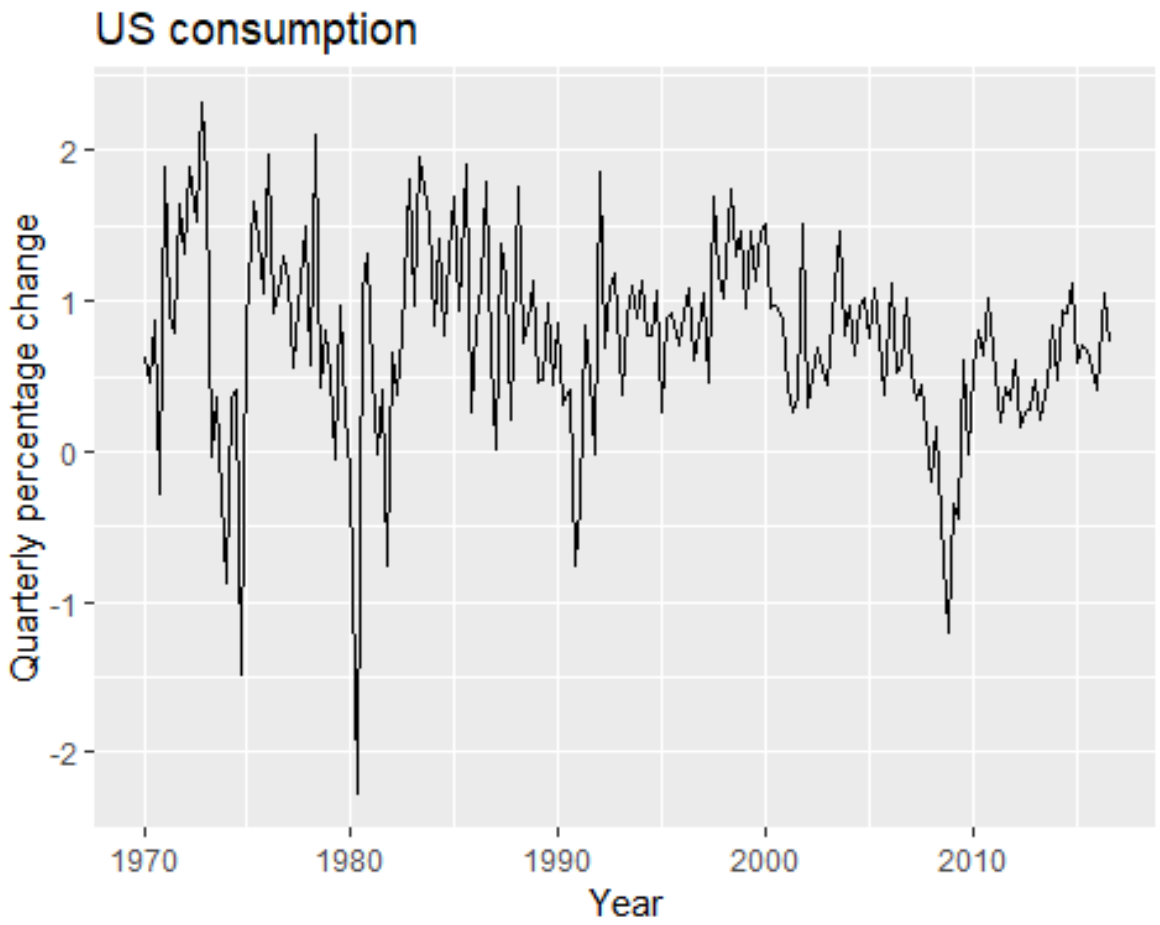
$$(1 - \phi_1 B)\ (1 - B) y_t = c + (1 + \theta_1 B) \varepsilon_t$$

↑ AR(1) ↑ First difference ↑ MA(1)

Written out:

$$y_t = c + y_{t-1} + \phi_1 y_{t-1} - \phi_1 y_{t-2} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

US personal consumption



US personal consumption

```
(fit <- auto.arima(uschange[,"Consumption"]))
## Series: uschange[, "Consumption"]
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2    mean
##       1.3908  -0.5813  -1.1800  0.5584  0.7463
## s.e.  0.2553   0.2078   0.2381  0.1403  0.0845
##
## sigma^2 estimated as 0.3511:  log likelihood=-165.14
## AIC=342.28   AICc=342.75   BIC=361.67
```

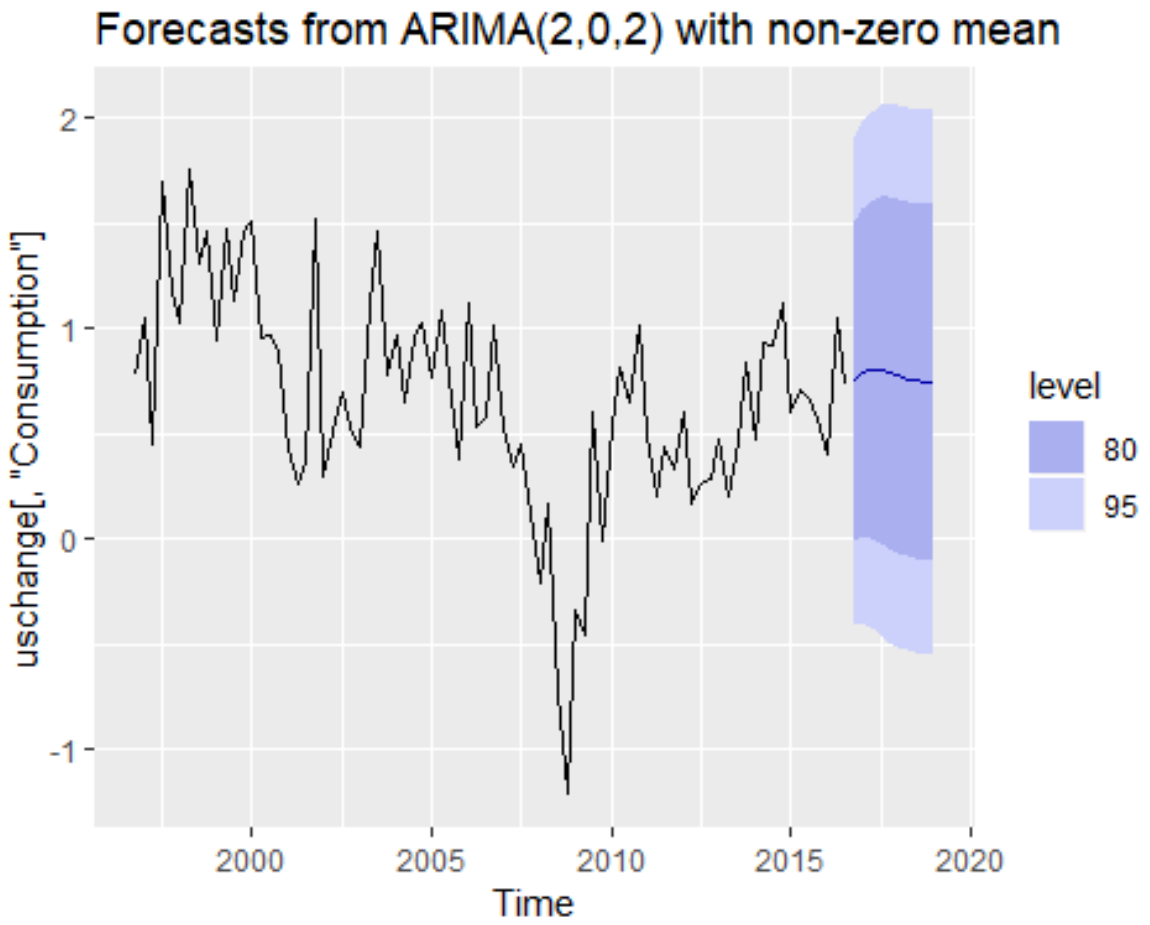
ARIMA(2,0,2) model:

$$y_t = c + 1.391y_{t-1} - 0.581y_{t-2} - 1.180\varepsilon_{t-1} + 0.558\varepsilon_{t-2} + \varepsilon_t$$

where $c = 0.746 \times (1 - 1.391 + 0.581) = 0.142$ and ε_t is white noise with a standard deviation of $0.593 = \sqrt{0.351}$.

US personal consumption

```
fit %>% forecast(h=10) %>% autoplot(include=80)
```



Understanding ARIMA models

- If $c = 0$ and $d = 0$, the long-term forecasts will go to zero.
- If $c = 0$ and $d = 1$, the long-term forecasts will go to a non-zero constant.
- If $c = 0$ and $d = 2$, the long-term forecasts will follow a straight line.
- If $c \neq 0$ and $d = 0$, the long-term forecasts will go to the mean of the data.
- If $c \neq 0$ and $d = 1$, the long-term forecasts will follow a straight line.
- If $c \neq 0$ and $d = 2$, the long-term forecasts will follow a quadratic trend.

Understanding ARIMA models

Forecast variance and d

- The higher the value of d, the more rapidly the prediction intervals increase in size.
- For $d = 0$, the long-term forecast standard deviation will go to the standard deviation of the historical data.

Cyclic behaviour

- For cyclic forecasts, $p \geq 2$ and some restrictions on coefficients are required.
- If $p = 2$, we need $\phi_1^2 + 4\phi_2 < 0$ .Then average cycle of length
- $(2\pi)/[\arccos(-\phi_1(1-\phi_2)/(4\phi_2))]$.

Outline

- Stationarity and differencing
- Non-seasonal ARIMA models
- Estimation and order selection
- ARIMA modelling in R
- Forecasting
- Seasonal ARIMA models
- ARIMA vs ETS

Maximum likelihood estimation

Having identified the model order, we need to estimate the parameters $c, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$.

- MLE is very similar to least squares estimation obtained by minimizing
- $\sum_{t-1}^{T} e_t^2$.
- The `Arima()` command allows CLS or MLE estimation.
- Non-linear optimization must be used in either case.
- Different software will give different estimates.

Partial autocorrelations

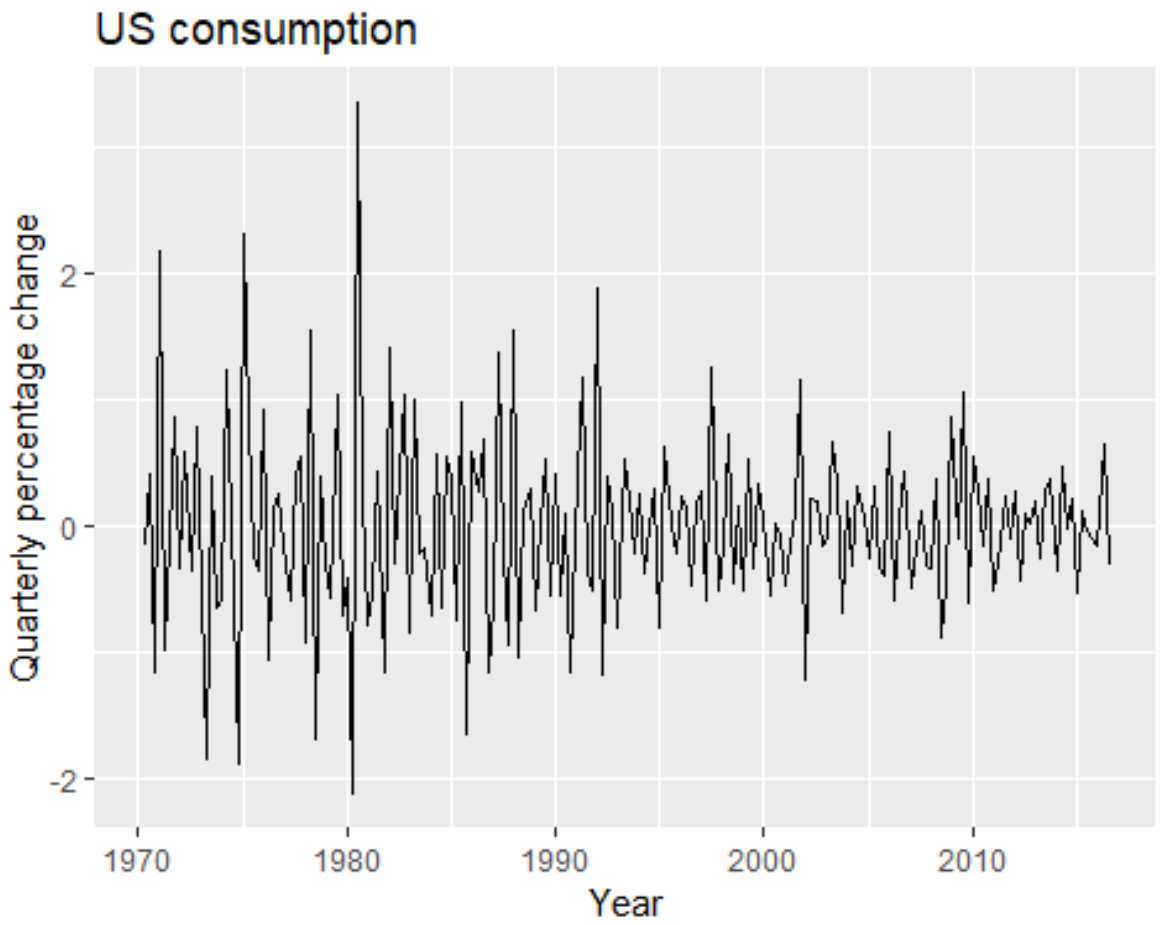
measure relationship between y_t and y_{t-k} ,when the effects of other time lags — $1,2,3,\ldots,k-1$ — are removed.

α_k = kth partial autocorrelation coefficient
= equal to the estimate of β_k in regression:

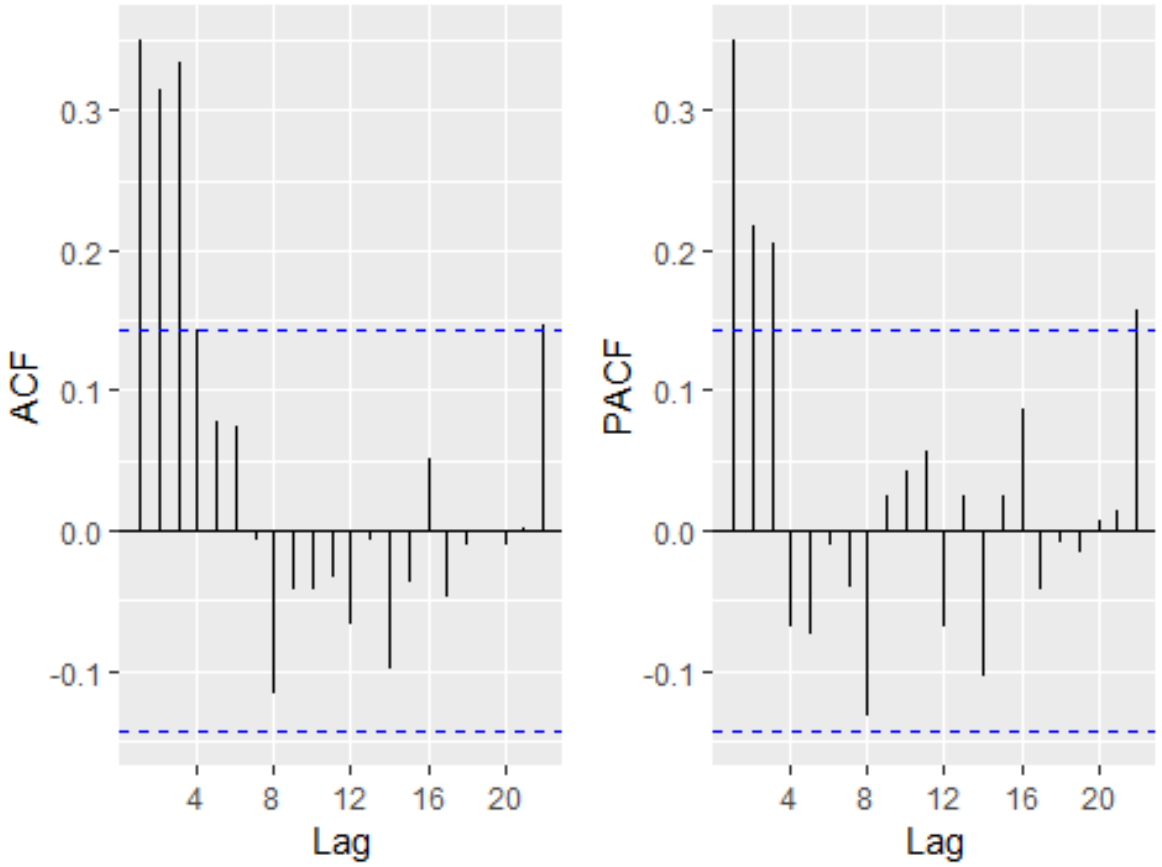
$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-k}$$

- Varying number of terms on RHS gives α_k for different values of k.
- There are more efficient ways of calculating α_k.k
- $\alpha_1 = \rho_1$
- same critical values of $\pm 1.96/\sqrt{T}$ as for ACF.

Example: US consumption



Example: US consumption



ACF and PACF interpretation

AR(1)

$$\rho_k = \phi_1^k \qquad \text{for } k = 1,2,\dots;$$
$$\alpha_1 = \phi_1 \qquad \alpha_k = 0 \qquad \text{for } k = 2,3,\dots;$$

So we have an AR(1) model when

- autocorrelations exponentially decay
- there is a single significant partial autocorrelation.

ACF and PACF interpretation

AR(p)

- ACF dies out in an exponential or damped sine-wave manner
- PACF has all zero spikes beyond the pth spike

So we have an AR(p) model when

- the ACF is exponentially decaying or sinusoidal
- there is a significant spike at lag p in PACF, but none beyond p

ACF and PACF interpretation

MA(1)

$$\rho_1 = \theta_1 \qquad \rho_k = 0 \qquad \text{for } k = 2,3,\ldots;$$
$$\alpha_k = -(-\theta_1)^k$$

So we have an MA(1) model when

- the PACF is exponentially decaying and
- there is a single significant spike in ACF

ACF and PACF interpretation

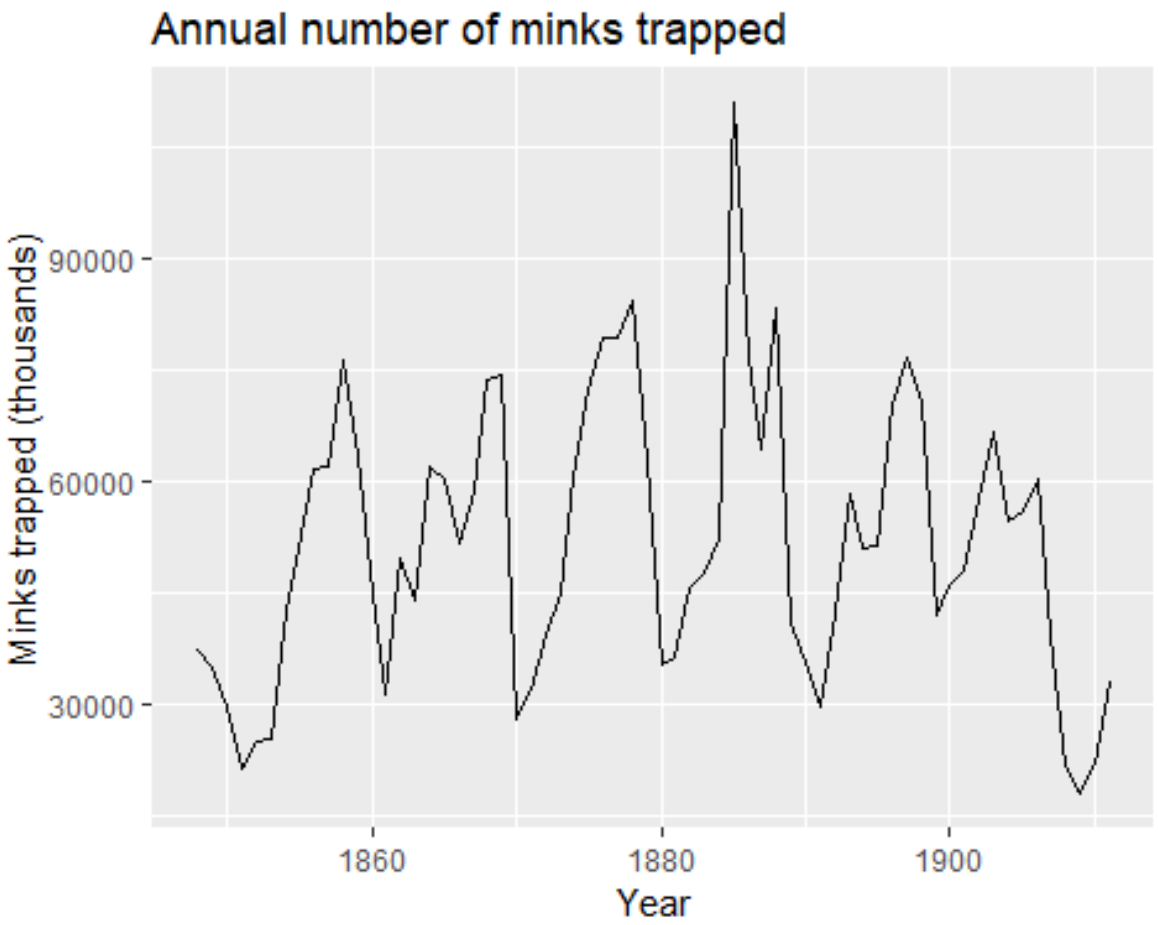
MA(q)

- PACF dies out in an exponential or damped sine-wave manner
- ACF has all zero spikes beyond the qth spike

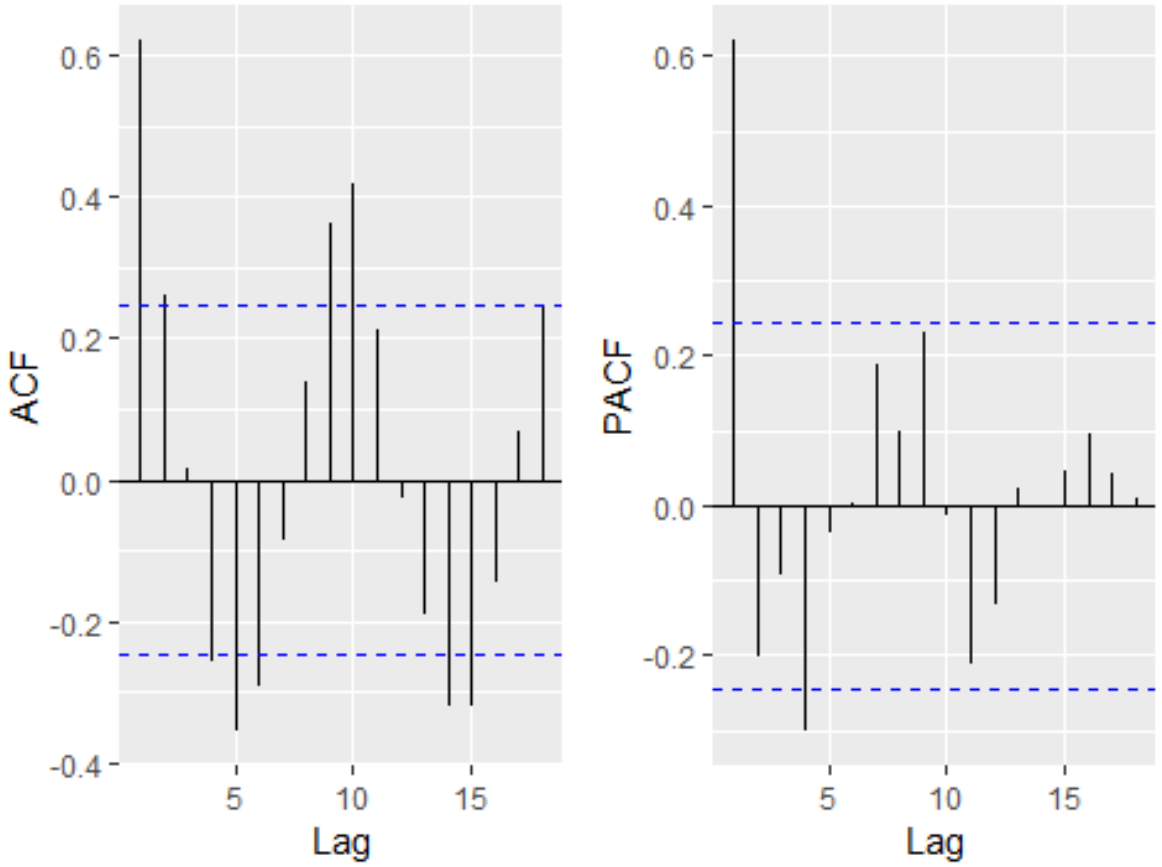
So we have an MA(q) model when

- the PACF is exponentially decaying or sinusoidal
- there is a significant spike at lag q in ACF, but none beyond q

Example: Mink trapping



Example: Mink trapping



Information criteria

Akaike's Information Criterion (AIC):

$$\text{AIC} = -2\log L + 2(p + q + k + 1)$$

where L is the likelihood o the data,

$k = 1$ if $c \neq 0$ and $k = 0$ if $c = 0$.

Corrected AIC:

$$\text{AICc} = \text{AIC} + \frac{2(p + q + k + 1)(\text{p+q+k+2})}{T - p - q - k - 2}$$

Bayesian Information Criterion:

BIC = AIC + [log(T) − 2](p + q + k − 1).

Good models are obtained by minimizing either the AIC, AICc or BIC. Our preference is to use the AICc.

Outline

- Stationarity and differencing
- Non-seasonal ARIMA models
- Estimation and order selection
- ARIMA modelling in R
- Forecasting
- Seasonal ARIMA models
- ARIMA vs ETS

How does auto.arima() work?

The auto.arima() function in R uses a variation of the Hyndman-Khandakar algorithm (Hyndman & Khandakar, 2008), which combines unit root tests, minimisation of the AICc and MLE to obtain an ARIMA model. The arguments to auto.arima() provide for many variations on the algorithm.

Hyndman-Khandakar algorithm for automatic ARIMA modelling

1. The number of differences 0<=d<=2 is determined using repeated KPSS tests.
2. The values of p and q are then chosen by minimising the AICc after differencing the data d times. Rather than considering every possible combination of p and q, the algorithm uses a stepwise search to traverse the model space.
   a) Four initial models are fitted:
      1. ARIMA(0,d,0)
      2. ARIMA(2,d,2)
      3. ARIMA(1,d,0)
      4. ARIMA(0,d,1)

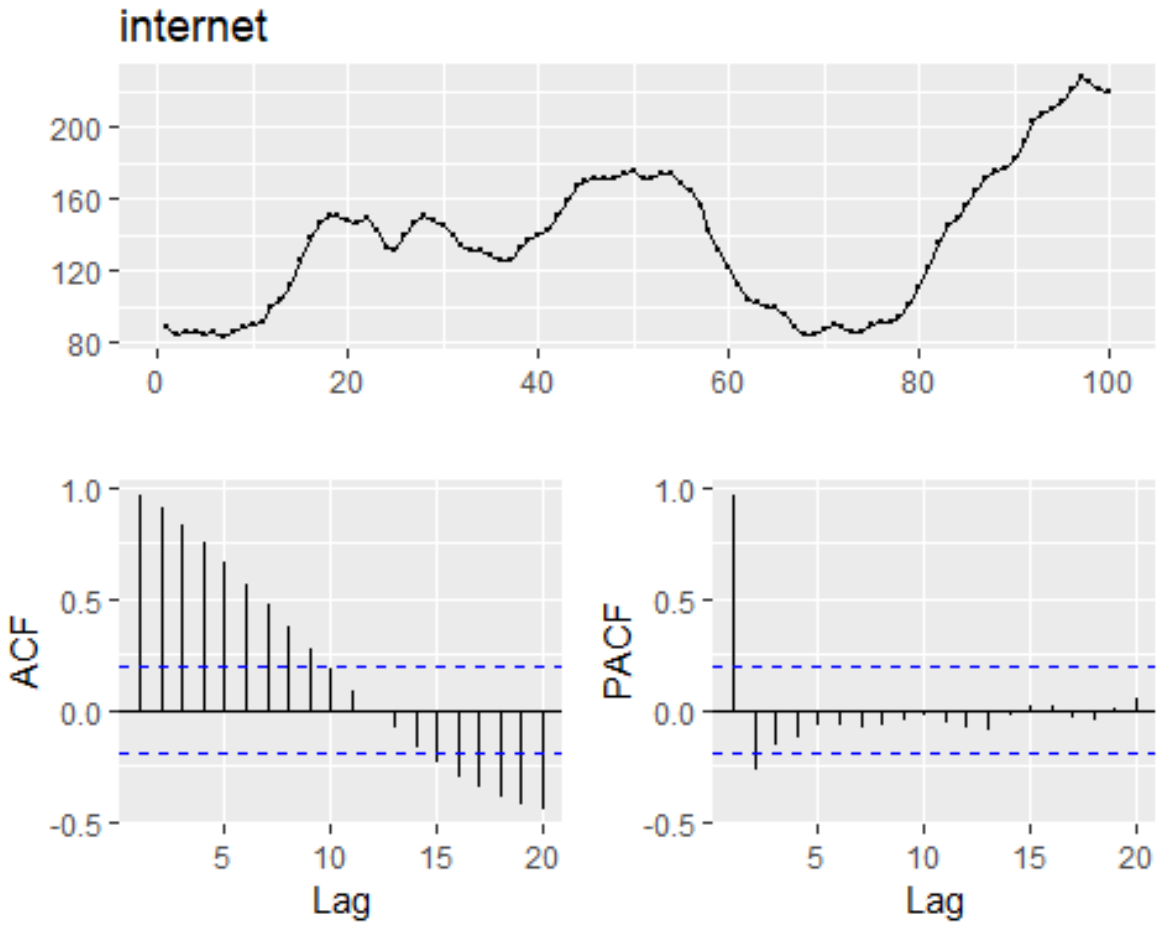
      A constant is included unless d=2. If d<=1, an additional model is also fitted: ARIMA(0,d,0) without a constant.
   b) The best model (with the smallest AICc value) fitted in step (a) is set to be the "current model".
   c) Variations on the current model are considered:
      i. vary p and/or q from the current model by ±1
      ii. include/exclude c from the current model.

   The best model considered so far (either the current model or one of these variations) becomes the new current model.

Repeat Step 2(c) until no lower AICc can be found.

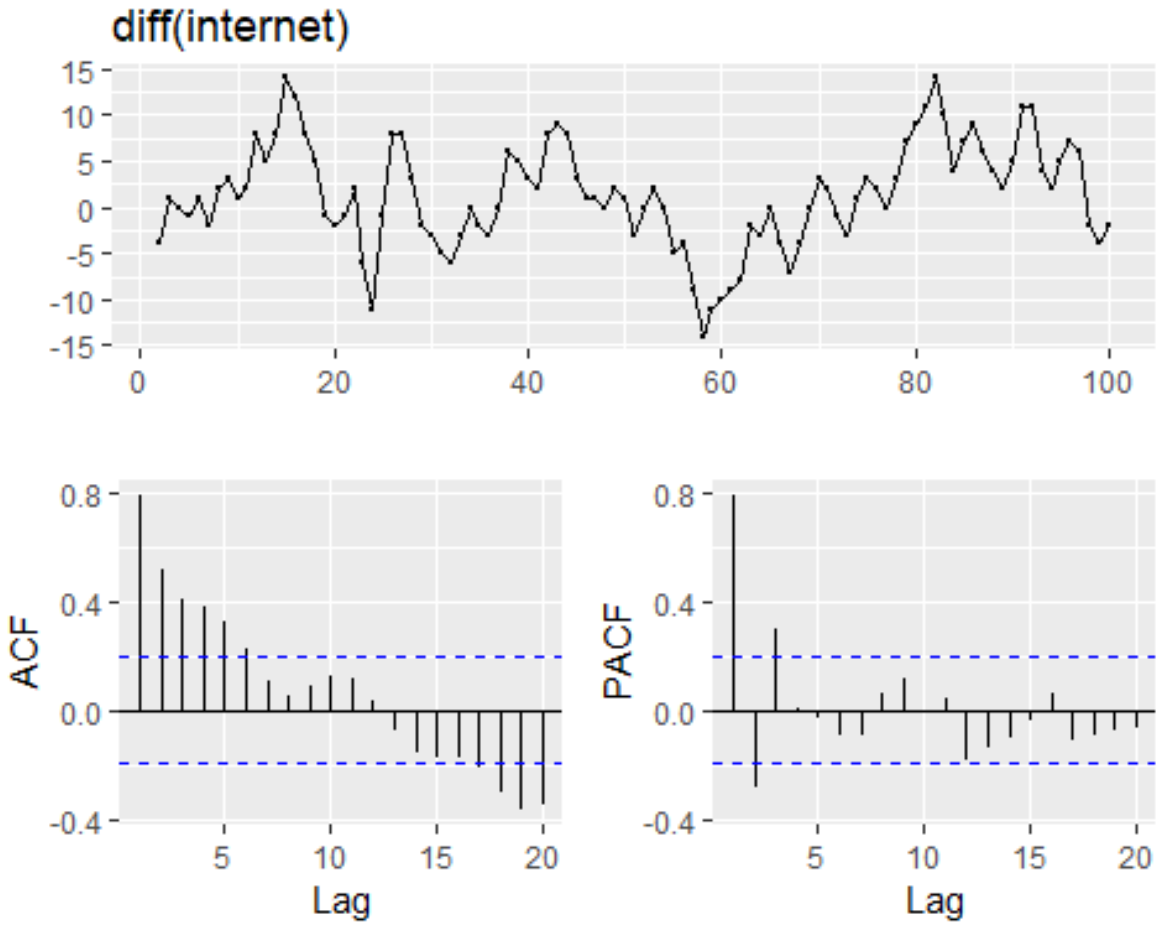
Choosing your own model

```
ggtsdisplay(internet)
```



Choosing your own model

```
ggtsdisplay(diff(internet))
```



Choosing your own model

```
(fit <- Arima(internet,order=c(3,1,0)))
## Series: internet
## ARIMA(3,1,0)
##
## Coefficients:
##          ar1      ar2     ar3
##       1.1513  -0.6612  0.3407
## s.e.  0.0950   0.1353  0.0941
##
## sigma^2 estimated as 9.656:  log
likelihood=-252
## AIC=511.99   AICc=512.42   BIC=522.37
```

Choosing your own model

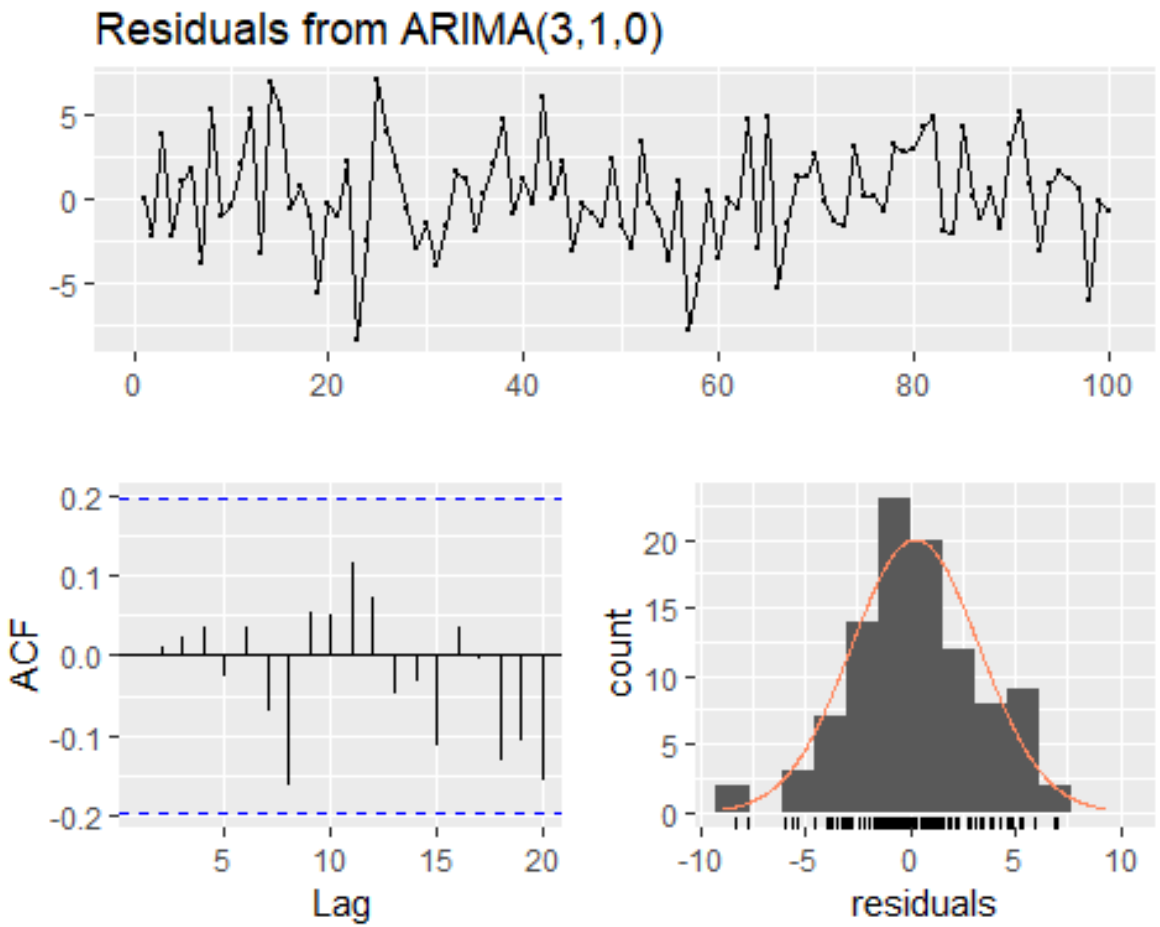
```
auto.arima(internet)
## Series: internet
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1     ma1
##       0.6504  0.5256
## s.e.  0.0842  0.0896
##
## sigma^2 estimated as 9.995:  log
likelihood=-254.15
## AIC=514.3   AICc=514.55   BIC=522.08
```

Choosing your own model

```
auto.arima(internet, stepwise=FALSE,
  approximation=FALSE)
## Series: internet
## ARIMA(3,1,0)
##
## Coefficients:
##          ar1      ar2     ar3
##       1.1513  -0.6612  0.3407
## s.e.  0.0950   0.1353  0.0941
##
## sigma^2 estimated as 9.656:  log likelihood
```

Choosing your own model

```
checkresiduals(fit)
```

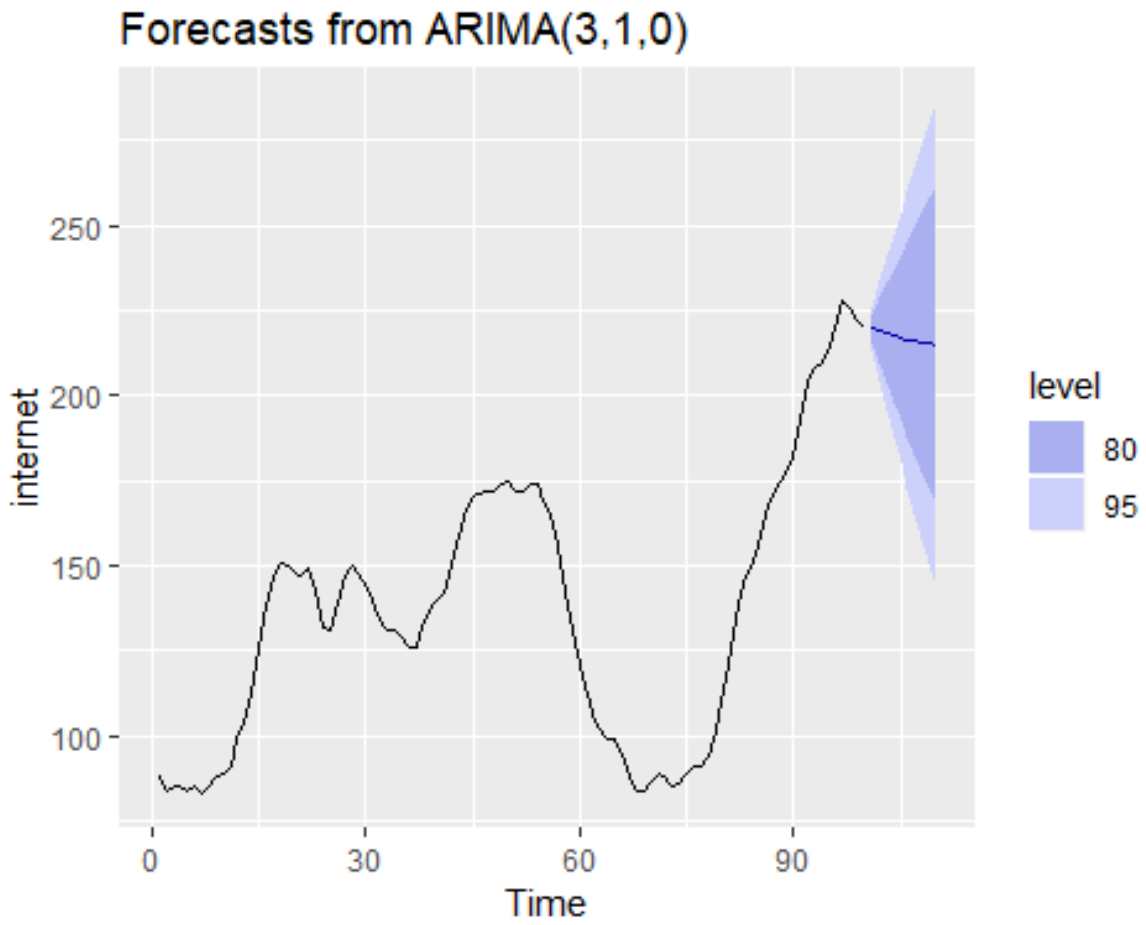


Choosing your own model

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,0)
## Q* = 4.4913, df = 7, p-value = 0.7218
##
## Model df: 3.   Total lags used: 10
```

Choosing your own model

```
fit %>% forecast %>% autoplot
```



Modelling procedure with ARIMA

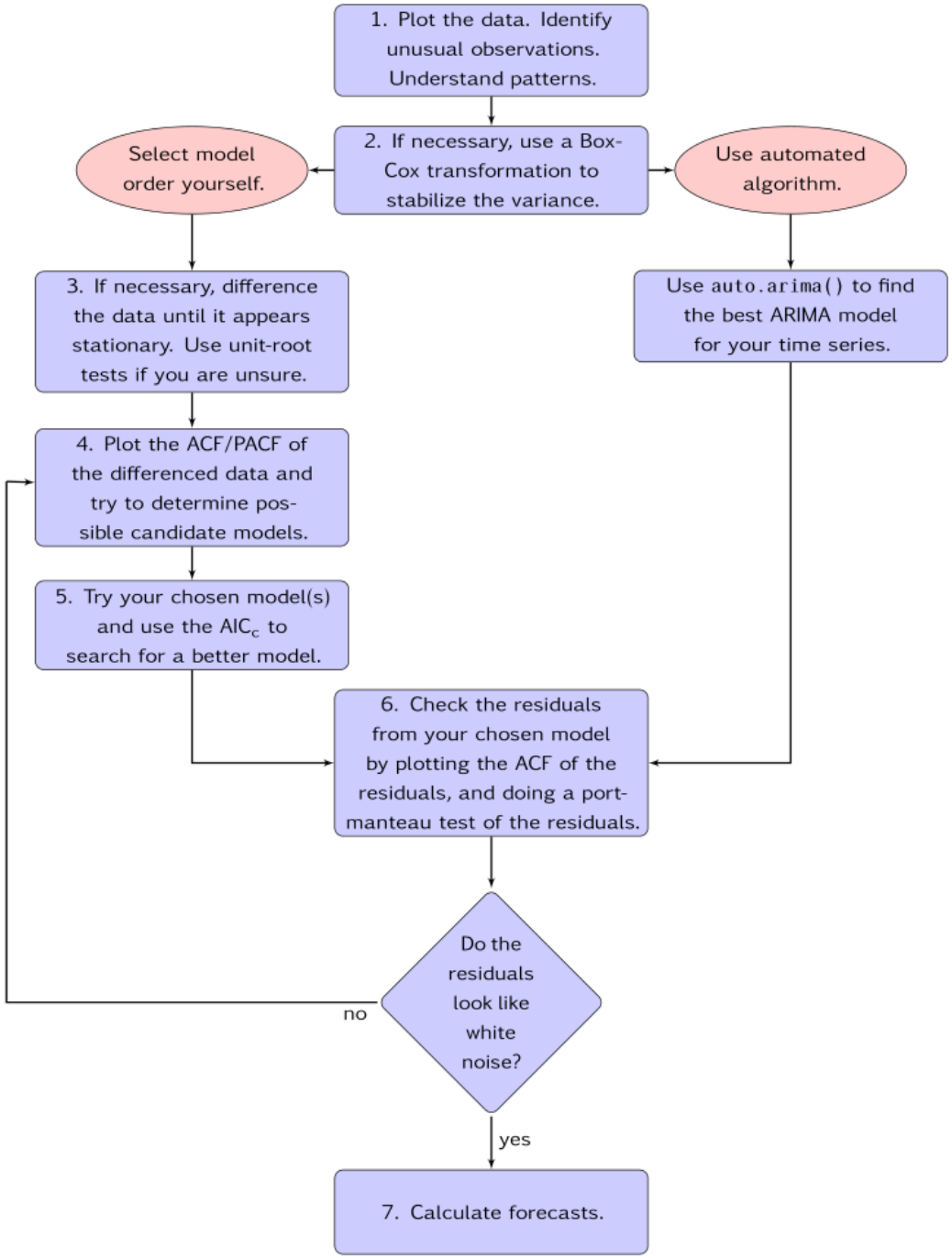
1. Plot the data. Identify any unusual observations.
2. If necessary, transform the data (using a Box-Cox transformation) to stabilize the variance.
3. If the data are non-stationary: take first differences of the data until the data are stationary.
4. Examine the ACF/PACF: Is an AR(p) or MA(q) model appropriate?
5. Try your chosen model(s), and use the to search for a better model.
6. Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals. If they do not look like white noise, try a modified model.
7. Once the residuals look like white noise, calculate forecasts.

Modelling procedure with auto.arima

1. Plot the data. Identify any unusual observations.
2. If necessary, transform the data (using a Box-Cox transformation) to stabilize the variance.

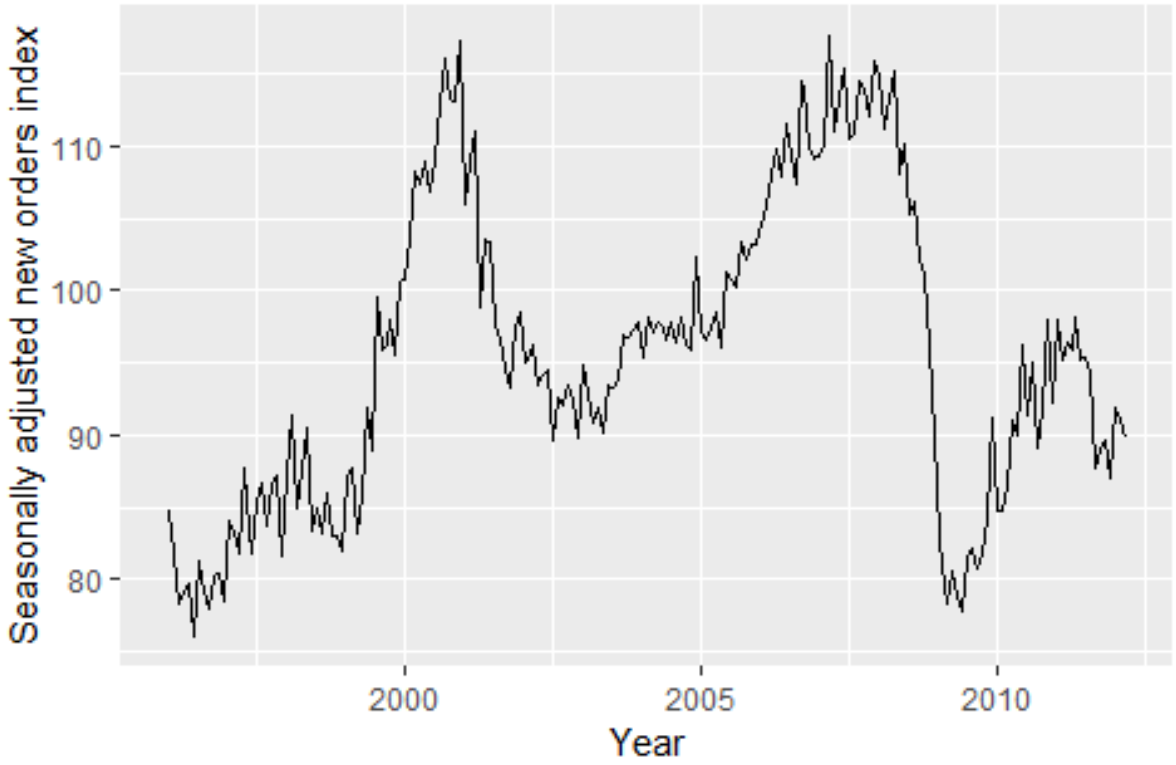
3. Use `auto.arima` to select a model.

6. Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals. If they do not look like white noise, try a modified model.
7. Once the residuals look like white noise, calculate forecasts.


1. Plot the data. Identify unusual observations. Understand patterns.
2. If necessary, use a Box-Cox transformation to stabilize the variance.
Select model order yourself.
Use automated algorithm.
3. If necessary, difference the data until it appears stationary. Use unit-root tests if you are unsure.
Use auto.arima() to find the best ARIMA model for your time series.
4. Plot the ACF/PACF of the differenced data and try to determine possible candidate models.
5. Try your chosen model(s) and use the AIC$_c$ to search for a better model.
6. Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals.
Do the residuals look like white noise?
no
yes
7. Calculate forecasts.

Seasonally adjusted electrical equipment

```
eeadj <- seasadj(stl(elecequip, s.window="periodic"))
autoplot(eeadj) + xlab("Year") +
  ylab("Seasonally adjusted new orders index")
```

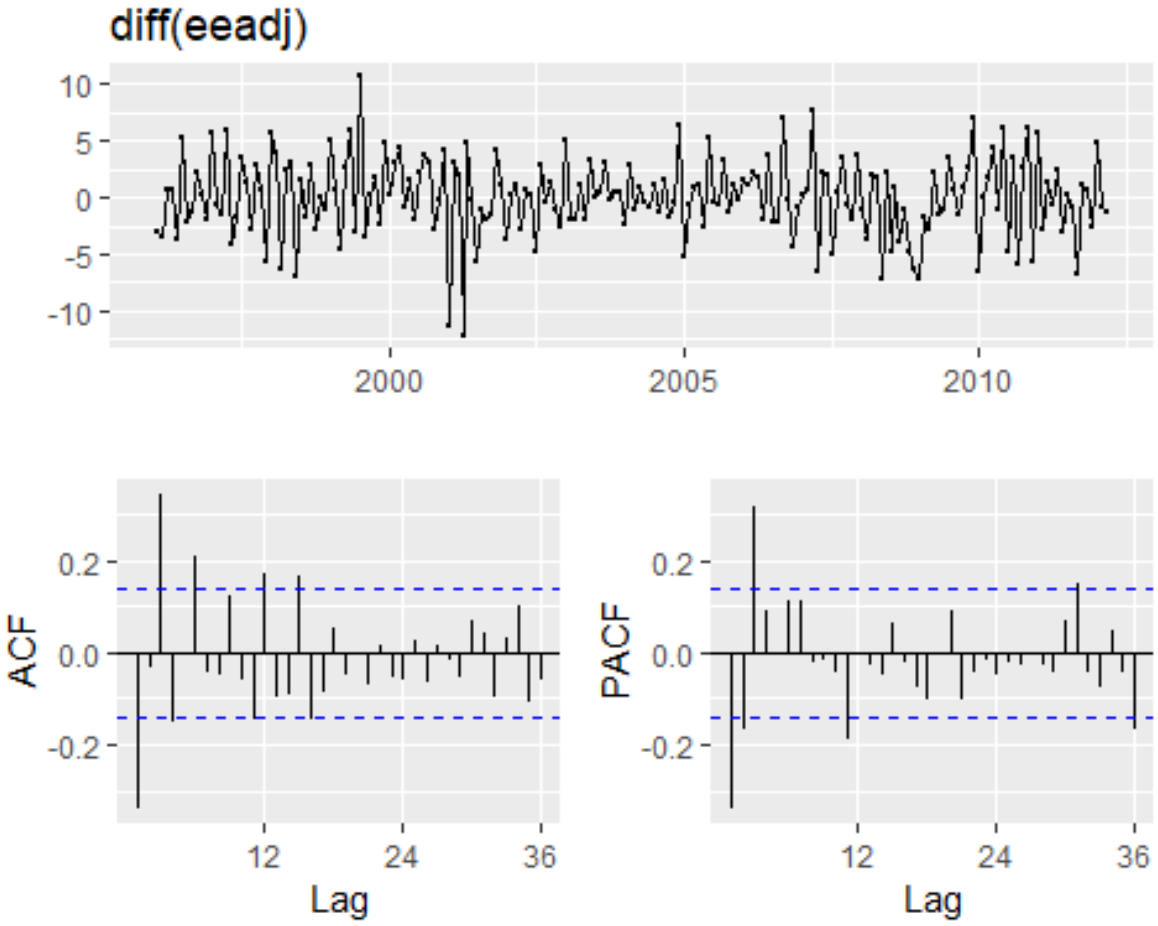


Seasonally adjusted electrical equipment

1. Time plot shows sudden changes, particularly big drop in 2008/2009 due to global economic environment. Otherwise nothing unusual and no need for data adjustments.
2. No evidence of changing variance, so no Box-Cox transformation.
3. Data are clearly non-stationary, so we take first differences.

Seasonally adjusted electrical equipment

```
ggtsdisplay(diff(eeadj))
```



Seasonally adjusted electrical equipment

4. PACF is suggestive of AR(3). So initial candidate model is ARIMA(3,1,0). No other obvious candidates.

5. Fit ARIMA(3,1,0) model along with variations: ARIMA(4,1,0), ARIMA(2,1,0), ARIMA(3,1,1), etc. ARIMA(3,1,1) has smallest value.

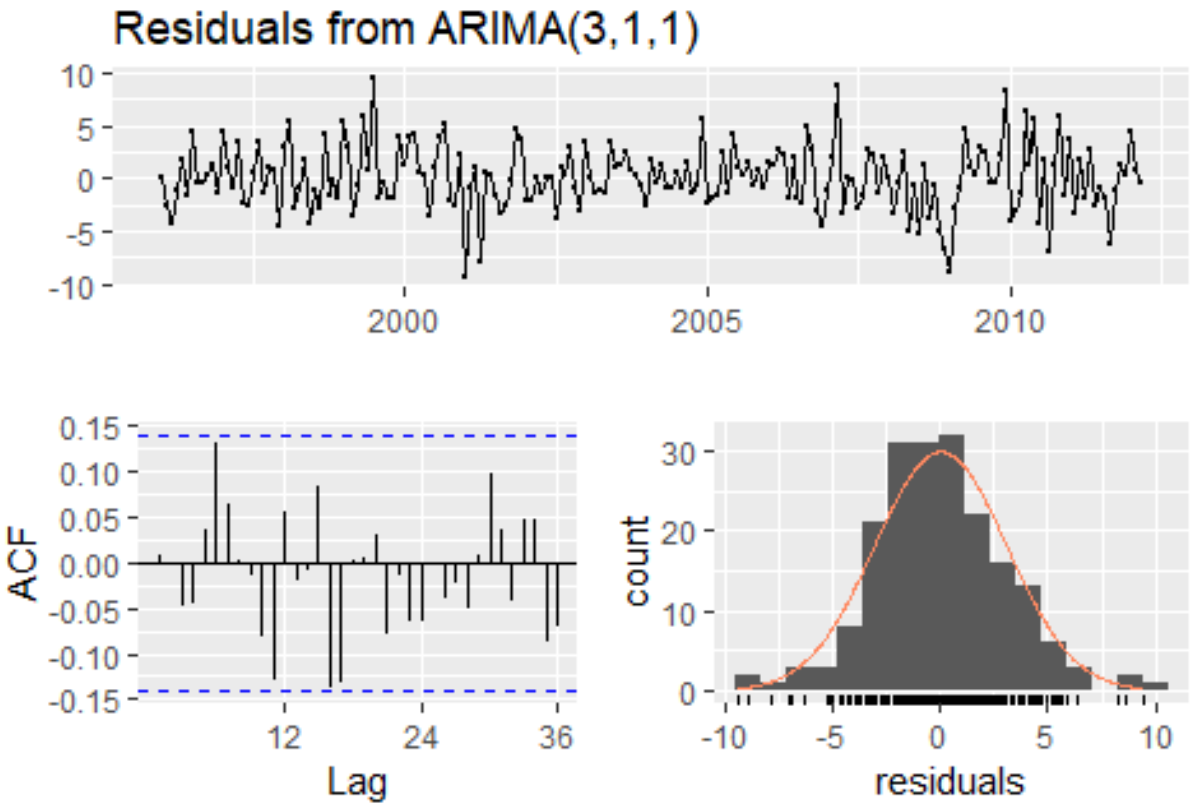
Seasonally adjusted electrical equipment

```
(fit <- Arima(eeadj, order=c(3,1,1)))
## Series: eeadj
## ARIMA(3,1,1)
##
## Coefficients:
##          ar1     ar2     ar3      ma1
##       0.0044  0.0916  0.3698  -0.3921
## s.e.  0.2201  0.0984  0.0669   0.2426
##
## sigma^2 estimated as 9.577:  log likelihood=-
492.69
## AIC=995.38   AICc=995.7   BIC=1011.72
```

Seasonally adjusted electrical equipment

6. CF plot of residuals from ARIMA(3,1,1) model look like white noise.

```
checkresiduals(fit)
```



Seasonally adjusted electrical equipment

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,1)
## Q* = 24.034, df = 20, p-value = 0.2409
##
## Model df: 4.   Total lags used: 24
```

Seasonally adjusted electrical equipment

```
fit %>% forecast %>% autoplot
```