Time Series

Time series
- series of data points indexed in time order
- sequence of observations over regularly spaced intervals of time
- Examples: daily conversion rates for the last three months

Time series analysis
- methods for analyzing time series data to extract meaningful statistics and characteristics

Time series forecasting
- use of a model to predict future values based on previously observed values

Methods: Two Classes
- Time-domain: auto-correlation and cross-correlation analysis
- Frequency-domain: spectral analysis, wavelet analysis

Panel data
- multidimensional data set
- time series data set is a one-dimensional panel
- difference: concerns Uniqueness of data-record - unique / differentiated by…
  - Time-series: time-field
  - Panel data: time-field and additional identifier unrelated to time :
  - Cross-sectional: non-time identifier

EDA
- Line Chart
- Autocorrelation analysis: examine serial dependence
- Decomposition: trend, seasonality,..
- Segmentation: splitting into sequence of individual segments, each with its own characteristic properties

Run chart
- run-sequence plot is a graph that displays observed data in a time sequence
- graph that displays observed data in a time sequence.

=> normal time series plot

Summary: Models

Models

Different models represent different underlying stochastic processes

AR, I, MA and combinations
- model variations in the level of a process
- depend linearly on previous data points

From ARIMA to VAR and ARIMAX
1. **VAR**:
   - for vector-valued data
   - to model multivariate time-series that effect each othe
2. **ARIMAX**:
   - to model that observed time-series is driven by "forcing" time-series
   - used if other series is causally related or at least predictive for observed series
   - uses information form independent variable to forecast series of interest
   - "X" for „exogenous" - also called dynamic regression

VAR vs ARIMAX
- ARIMAX: explanatory variables affect only variable of interest
  => forcing series is deterministic
- VAR: multiple series that affect each other

Exponential Smoothing
- characterized by
  - number of smoothing parameters (0-1) - data(level), trend, season
  - smoothing equations
- **Simple**: describes level => straight horizontal line forecast
- **Double**: describes level + trend
  - Holt's linear: => straight line with constant slope
  - Holt's damed: => line dampens to horizontal
- **Triple**: describes level + trend + seasonality
  - Additive: seasonal variations are roughly constant throughout series
  - Multiplicative: seasonal variations are changing proportional to level of series

Dynamic Harmonic Regression

Uses Fourier terms to model series with long seasonal periods

TBATS

Uses components of different models-types and in fully automated procedure

ARCH & GARCH
- volatility models
- non-constant variance / non-linear time series models
- represent the changes of variance over time - heteroskedasticity
- changes in variability are related / predicted by recent past values
- **ARCH**: error variance follows AR model
- **GARCH**: error variance follows ARMA model

HMM
- model in which system being modeled is assumed to be Markov process with unobserved (hidden) states

Summary: Tests

Stationarity

Dickey–Fuller test
- tests the null hypothesis that a unit root is present in autoregressive model

Augmented Dickey–Fuller test
- augmented version of the Dickey–Fuller test
- used for a larger and more complicated models

Autocorrelation

Durbin–Watson test
- tests for presence of autocorrelation at lag 1 in the residuals

Ljung–Box test
- tests whether any of a group of autocorrelations of a time series are different from zero
- instead of testing sat each distinct lag: tests the "overall" randomness based on a number of lags

Serial Correlation

Breusch–Godfrey test
- used for regression models
- tests for presence of serial correlation that has not been included in proposed model

Notes

Determine which parts are predictive
- During training: which part of series is predictive of the future
  => only include that in during training
- Inspecting the time series important
  - eg. are there several distinct parts in the series?!
    => might exhibit different behavior in middle than in the end - maybe due to different process
    => should not be used for training
- Tools:
  - Visual Inspection: which parts seem to be predictive
  - ACF: gives information of how long in the past provides information for forecasting
  - Scatterplot: with current vs lag 1, lag 2, etc => more intuitive

How would you forecast X?!
- is information from past available
- what carries information to predict:
  - yesterday
  - shock of yesterday
  - other variables - can they be predicted?
- does it make sense to forecast next value based yesterday or other variables?

Problem n-step ahead / large horizon forecasting
- forecast predicted based on values predicting it
- in long forecasts these are not available
  => Always ask: What information do I really have?!

Problem using independent variables
- Question: why not just use explanatory variables?
- Problem: independent vars need to be forecasted themselves for those indices
  => explanatory vars needs to be forecasted before the variable of interest can be forecasted
- Easy: for planned events like christmas
- Hard: essentially every meaningful metrics

Histogram vs Runs-Plot
- histogram has no information about time

Random component
- can be analyzed for for eg. mean location, or mean squared size (variance)
- or whether the component is random or might be modeled with ARIMA model

ARIMA vs Exponential Smoothing

- complementary approaches to forecasting
  - ES models: describe trend and seasonality in the data,
  - ARIMA models: describes. autocorrelations in the data

  => trend, seasonal effects are treated as nuisance parameters and effects removed before analysis
- linear exponential smoothing models are all special cases of ARIMA models,
- non-linear exponential smoothing models have no equivalent ARIMA counterparts
- but also many ARIMA models have no exponential smoothing counterparts
- Examples for expressing one as the other ETS(A,N,N) = ARIMA(0,1,1) with $\theta_1 = \alpha - 1$

- ETS: modlsf trend and seasonal com- ponents. Such components are explicitly modelled in the state space approach. In the Box–Jenkins approach, trend and seasonal effects are treated as nuisance parameters. These effects are removed from the series before any analysis can begin.

Stationarity: ETS vs ARIMA

- ARIMA models requires (differenced) time series to be stationary
- sometimes the series cannot be made stationary however much differencing is done

  => Question: How close to stationary is close enough
- Contrast ETS: stationarity of the time series is not required

Basic Concepts

Time series

- sequence of observations over regularly spaced intervals of time
- Examples:
  - Monthly unemployment rates for the previous five years
  - Daily production at a manufacturing plant for a month

Stochastic process

- linear regression tries model true relationship between indep. and dep. var using linear regression model
- time-series model models the underlying stochastic process
- observed time series can be considered a realization of a stochastic process

Regression vs Time Series

- Normal regression: independence of serial errors are presumed, or at least minimized
- Time series regression: accounts for autocorrelation between time events

Parsimonious model

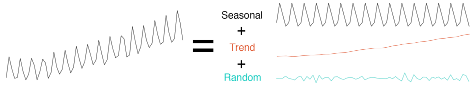
Simple models with great explanatory predictive power.

=> explain data with a minimum number of parameters and predictor variables

Lag

number of time periods that separate the time series data

Components



Motivation: Components & Decomposition
- time series data can exhibit variety of patterns
- split series into several components - each representing an underlying pattern category

Systematic & Non-Systematic Components
- Given series can be thought to consist of..
- 3 systematic components:
  Consistent / recurrent components of series that can be described and modeled
- 1 non-Systematic components:
  Components of series that cannot be directly modeled

Four Components

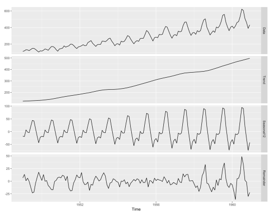
0. (**Level)**
- average value of the series

1. **Trend**
- trend component **captures changes in level over time around which** the seasonal and irregular fluctuate
- long-term tendency of series to increase or fall - upward / downward trend
- increasing / decreasing value in the series

2. **Seasonality**
- seasonal component captures **cyclical effects due to the time of year**
- periodic pattern mostly due to the calendar (e.g. the quarter, month, or day of the week)
- periodic fluctuation that form pattern that tends to repeat from one seasonal period to the next

3. **Cycles**
- pattern of regular fluctuations that is not of fixed period (duration usually > 2 years)
  => lengths of time between successive peaks or troughs of a cycle not necessarily the same
- long departures from the trend due to factors others than seasonality
  => usually occur along a large time interval

4. **Non-Systematic Component: Noise**
- irregular / error component captures those influences not described by the trend and seasonal effects.
- movement left after explaining trend, seasonal and cyclical movements
- random noise or error in a time series



Seasonal vs Cyclic Pattern
- Seasonal pattern constant length vs. cyclic pattern variable length
- both together: average length of cycle longer than length of seasonal pattern
- magnitude of cycle more variable than magnitude of seasonal pattern
  => timing of peaks and troughs is predictable with seasonal data,
  but unpredictable in the long term with cyclic data

Decomposition

Methods combine trend and cycle component
- trend and cycle combined into a single trend-cycle component = trend component
- meaning:
  - if see current current downward trend, then
    1. could be trend
    2. only seems like a trend
       but could part of of long cycle, if longer series was available
- Three components: a trend-cycle component, a seasonal component, remainder component

Decomposition
- used to describe the trend and seasonal factors in a time series
- extensive decompositions also include long-run cycles, holiday effects, day of week effects and so on

Goal
- estimate seasonal effects
- used to create and present seasonally adjusted values
- seasonally adjusted value removes seasonal effect from value so that trends can be seen more clearly

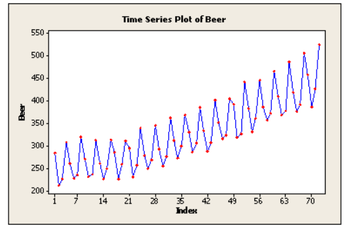
Two types of decompositions

1. **Additive**:

   x_t = Trend + Seasonal + Random

   Used if the seasonal variation is relatively constant over time
   = magnitude of the seasonal fluctuations
   = **variation around the trend-cycle does not vary with level of series**

   Effects of the components are additive



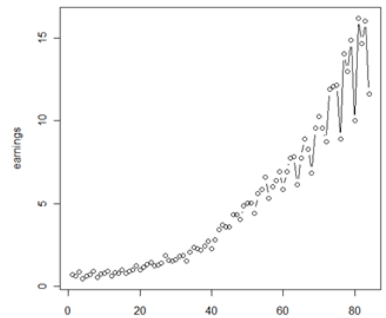
2. **Multiplicative**:

   x_t = Trend * Seasonal * Random

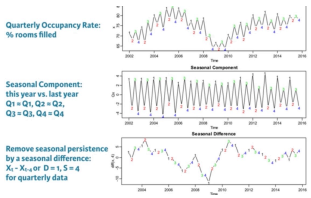
   Used if seasonal variation changes over time
   = when variation in seasonal pattern = variation around trend is proportional to level of series
   = **Variability of the season component is proportional to the level**
   As level increases => so does the variability

   Effects of the components are multiplicative

   By taking the log multiplicative series ,
   it can be decomposed using an additive model!



Intuition: Additive vs Multiplicative in nature
- Additive: Every year in December 10,000 more apartments are sold than in November
  => seasonality represented by absolute amount
- Multiplicative: Sell 10% more apartments in summer months than in winter months
  => seasonality represented by constant factor
  = > **CANNOT be represented by absolute amount**



Methods components extractions

Moving average of order m: $\hat{T}_t = \frac{1}{m}\sum_{j=-k}^{k} y_{t+j}$ where $m = 2k + 1$

=> estimate of the trend-cycle at time t obtained by averaging values of series within k periods of t

A: **Classical Decomposition**: Moving Average

1. **Estimate trend**
   Two approaches to estimate trend:
   a) Estimate trend with a smoothing procedure
      Centered moving averages:
      1. Find periodicity - length of season
      2. Use centered moving average with length of season to smooth and estimate trend
      => no equation is used to describe trend
   b) Model the trend with a regression equation

2. **De-trend the series**
   Additive decomposition: by subtracting the trend estimates from the original data
   Multiplicative decomposition: by dividing original data by trend values

3. **Estimate seasonal factors using de-trended series**
   - Average the de-trended values for each season
     e.g. monthly data: average all values for january, feb, etc
       get a seasonal effect for January, by averaging de-trended values for all Januarys in the series
       => Result: estimate effect for each month
   - Minitab: uses medians not means
   - seasonal effects usually adjusted: average to 0 for an additive / 1 for a multiplicative decomposition

4. **Determine irregular component**
   - by removing trend and seasonality form original data
   - Additive model, random = series – trend – seasonals
   - Multiplicative model, random = series / (trend*seasonal)

Problem: Classical decomposition
1. assume seasonal component repeats from year to year
   => sometimes unreasonable assumption
      eg. electricity demand patterns changed over time as air conditioning has become more widespread
2. trend-cycle estimate tends to over-smooth rapid rises and falls in the data

B: **STL: Seasonal and Trend decomposition using Loess**
- versatile and robust method for decomposing
- solves several problems of classical decomposition
  1. seasonal component: allowed to change over time
     => rate of change can be controlled by the user
  2. trend-cycle: smoothness of be controlled
  3. robustness: outlier effect can be mitigated
- Two main parameters:
  Trend-cycle window: how rapidly trend-cycle component can change
  Seasonal window: how rapidly seasonal components can change
  => smaller values allow for more rapid changes

C: **SEATS and X11**

Stationary Processes

Intuition
Series is stationary if it is stable, meaning:
- No trend: mean is constant over time
- Constant variance: correlation structure remains constant over time

=> stationary time series' properties do not depend on the time at which series is observed

Importance
- Allows to estimate mean: because mean is constant => use sample average
- Covariance between two observations only depends on the lag (and not the indices)
  => Correlation structure constant
  use pairs $(x_1, x_2), (x_2, x_3), \ldots$ to estimate Lag-1 Correlation
  use pairs $(x_1, x_3), (x_2, x_4), \ldots$ to estimate Lag-2 Correlation
  => covariance between two observations depends only on the lag,
     the time distance between observations, not the indices t or s directly

Identifying Stationarity
Runs Plot: should show constant location and scale
ACF plot: non-stationarity processes often show very slow decay

Behavior of stationary processes
- distributional stability over time
- Fluctuate randomly but behave similarly from one time period to the next

- Intuitive:
  - Effect of an observation dissipates as time goes on
    What happens today has less and less effect on the data the further we go in the future
    => best estimate for long term prediction for stationary series is the mean of the series
    => if series goes up or down the, the mean would not be good predictor for the future

Strictly stationary
Process is strictly stationary, distribution of a sequence of n observations unchanged by shifts in time
=> strong assumption

Weakly stationary
- Also: covariance stationary / wide-sense stationary
- Weakly stationary, if
  1. mean μ, variance σ^2 are time-independent = unchanged by time shifts
  2. autocorrelation function only depends only on lag between t_1 and t_2
- Definition
  The series $Y_1, Y_2, \ldots$ is weakly stationary if for all t and s and some function γ(h)

$$E(Y_t) = \mu \qquad Var(Y_t) = \sigma^2 \qquad Cov(Y_t, Y_s) = \gamma(|t - s|)$$

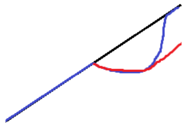
  => covariance between two observations depends only on the lag, the time distance lt – sl

Mean-reversion
- property of stationary processes: randomly oscillate around some fixed level

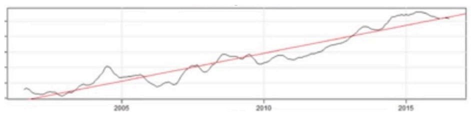
Violation of stationarity

1. **Trend in the mean**
Causes: 2 possibilities for the trend
1. Unit root: shocks have permanent effects. => process is not mean-reverting
2. Deterministic trend: = trend stationary process => (shocks have only transitory effects)
=> differencing can make both stationary

1a: **Unit Root**
- is a stochastic trend in series
- can be seen as random walk with drift within the series
- Intuition:
  - shocks have permanent effect
  - thus, process is not mean reverting
- Example:
  - Blue Line: drops and reverts back to mean, continues
    => no unit root and trend stationary
  - Red Line: drops and does not revert back and continues trend
    => unit root
- Details: process can be written as a series of monomials = expressions with single term
  each monomial corresponds to a root. If a monomial = 1 => then unit-root

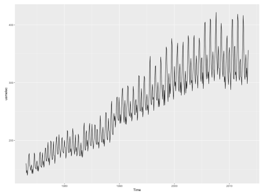


1b. **Trend stationary process**
- Definition:
  1. shocks have only transitory effect
  2. after which variable follows a deterministically evolving (non-constant) mean



- Not strictly stationary
  But can be transformed to stationary process by removing underlying trend
  Trend = function of time

2. **Non-Constant Variance** the variance increases / decreases with time
=> Solution: log or box-cox



Identification
- TS Plot
- ACF plot:
  Stationary: ACF will drop to zero relatively quickly
  Non-stationary: ACF will decreases slowly

Tests
- Dickey Fuller Test: based on linear regression => sometimes problem with autocorrelation
  H0: unit root is present in an autoregressive model
  H1: depending on version of the test - stationarity or trend-stationarity
- Augmented Dickey-Fuller: solves problem => can handle bigger, more complex models depending on

Notes: Stationarity

Differencing
- transformation applied to series data in order to make it stationary
- **Removes changes in level:** it **stabilizes the mean** by **eliminating trend and seasonality** of series

- Computation: difference between consecutive observations

$$y'_t = y_t - y_{t-1}$$

- Second order differencing:Sometimes necessary: difference data a second time

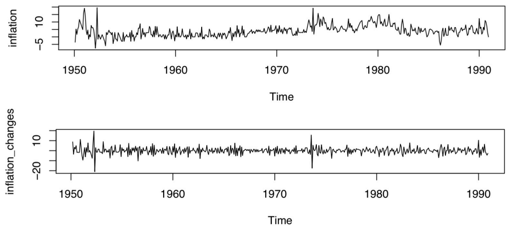
$$y^*_t = y'_t - y_{t-1}$$

- Seasonal differencing
  Computes difference between observation and corresponding observation in previous year.

$y'_t = y_t - y_{t-m}$ where m = duration of season

Models & Stationarity
- Stationary: WN, AR for $|\varphi| < 1$, MA
- Non-Stationary:
  - RW is always non-stationary - with and without drift => not stationary after differencing?

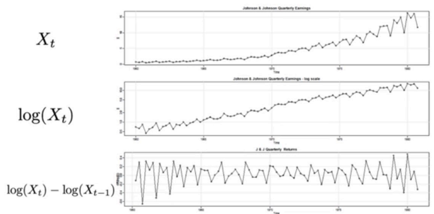
Change Series
- many time series do not exhibit stationarity
- but: difference of the series = changes = change series often approximately stationary
- Example:: Inflation vs Inflation changes
  Not stationary: Inflation: does not return naturally to a fixed level
  Stationary: Change in inflation: quick mean reversion to zero and no clear patterns over time



Example: Violation of both no-trend & constant variance
- investing money into bank with a fixed interest p
- X_0 = initial deposit
- X_t = value of account at time period t

Solution: 1. Log, 2. Differennce

Generated $X_t = (1 + p_t) * X_{t-1}$
Approximation of growth rate p_t:
$Y_t = \log X_t - \log X_{t-1} \approx p_t$



Autocorrelation

Autocorrelation
- also: serial correlation
- is correlation of series with delayed copy of itself as a function of delay
- tool to find repeating patterns, which are obscured by noise

Intuition
- If observations can be correlated with different observations k time units later
  - can be estimated at many lags to better assess how a time series relates to its past
  - typically most interested in how series relates to its most recent past

  => similarity between observations as a function of the time lag between them

Sample Autocorrelation function s
- measures correlation between observations that are separated by k time units (Y_t and Y_{t-k})
- R: acf(…, lag.max = ..., plot = FALSE) estimates all autocorrelations from 0, 1, 2,..., to lag.max
- Definition

$$\hat{p}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)},$$

where γ = auto-covariance function, ρ = autocorrelation function, for each lag h

ACF test bounds
- t-test:
  - H0: autocorrelation coefficient for specific lag = 0

    = sampling distribution based on the assumption that data is white noise

    Common rule: absolute value of t-statistic > 2 indicates that autocorrelation is not equal to 0
  - blue lines: show 95%-CI

    => 95% of all autocorrelations for white noise should lie between the blue lines

    => expect 1 out of 20 sample autocorrelations outside the test bounds simply by chance

Ljung-Box Q: (LBQ)
- test whether a series of observations over time are random and independent
- to determine whether all the autocorrelations up to and including a specific lag are equal to 0.

  => LBQ is greater than alpha, then you can conclude that the autocorrelation is not equal to 0
- H0: autocorrelations up to lag k are equal zero

  = >data values are random and independent up to a certain number of lags
- is Portmanteau test and modified version of Box-Pierce-Test

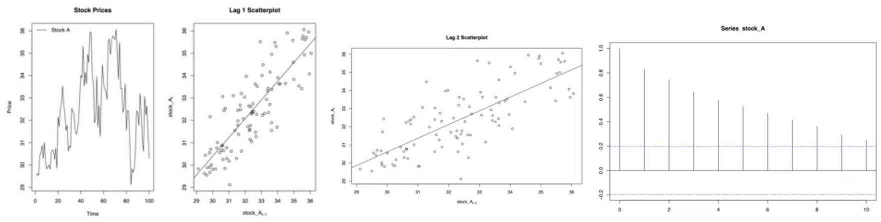
  => box-pierce only good for series with at least 100 observations

- portmanteau test is a type of statistical hypothesis test in which the null hypothesis is well specified, but the alternative hypothesis is more loosely specified

Visualizing Autocorrelation

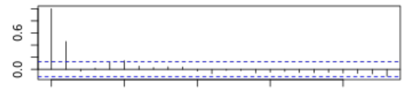
Important: Look for trend
- look for **trend in autocorrelation across** the plot instead just at spikes

Series **Series vs Lag 1** **Series vs Lag 2** **ACF: Lag 0 to 10**

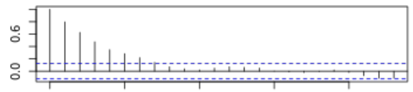


Examples of ACF Plots for different kinds of models

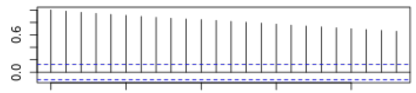
Moving average:
Autocorrelation for
the first lag only



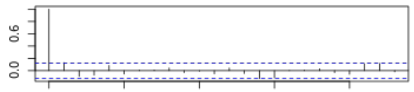
Autoregressive
Dissipating autocorrelation
across several lags



Random walk
considerable autocorrelation
for many lags. Finally



White Noise
No autocorrelation
with any lags



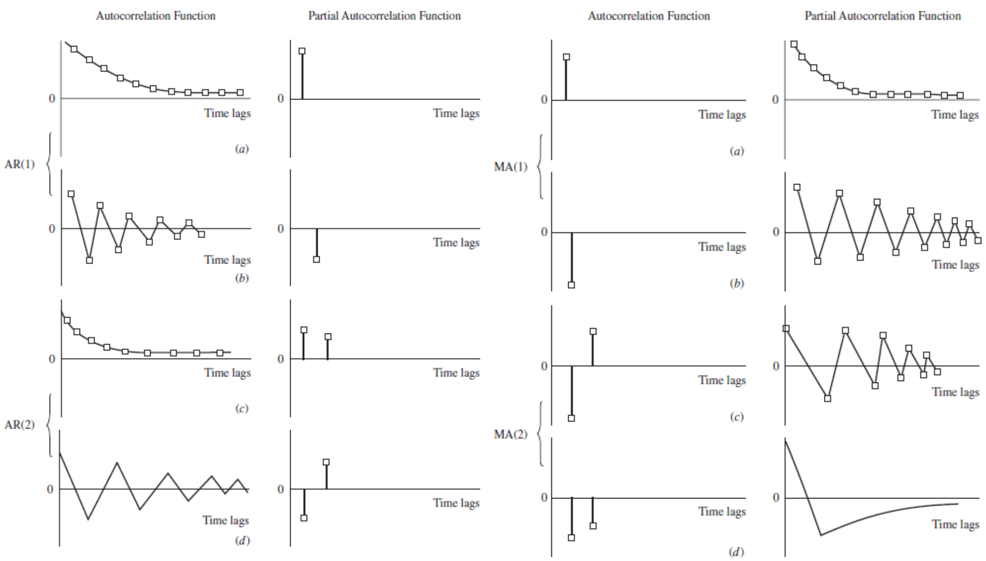
Partial Autocorrelation

Partial Autocorrelation
- correlation between observations in series that is not accounted for by previous, shorter intervals between observations
- Example:
  Partial autocorrelation for a lag of 6
  => only correlation that is not accounted for by lags 1 to 5

PACF: Partial autocorrelation function
- measures correlation between observations that are
  - separated by k time units (Y_t and Y_{t-k})
  - after adjusting for the presence of all the other terms of shorter lag $(y_{t-1}, y_{t-2}, \ldots, y_{t-k-1})$

Partial correlation is conditional Correlation
- correlation between two variables under the assumption that we know and take into account the values of some other set of variables
- regression context: y = response; x1, x2, and x3 are predictors.
  Partial correlation between y and x3 is the correlation between the variables determined taking into account how both y and x3 are related to x1 and x2

Orders
- 1st order partial autocorrelation: 1st order autocorrelation
- 2nd Order partial autocorrelation = $\dfrac{\textbf{Covariance}(x_t, x_{t-2} \mid x_{t-1})}{\sqrt{\textbf{Variance}(x_t \mid x_{t-1})\textbf{Variance}(x_{t-2} \mid x_{t-1})}}$



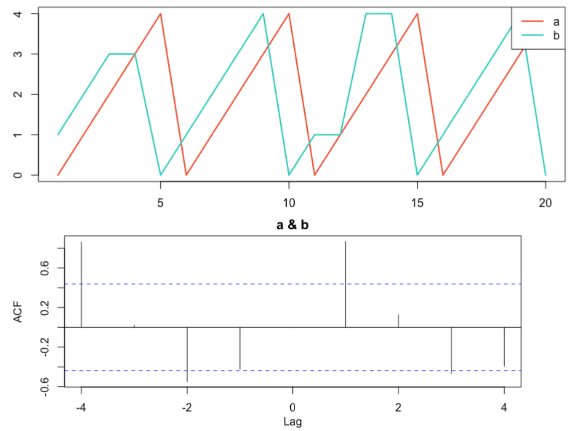
Cross-correlation

Used
- to describe and model of the relationship between two time series
- two series y_t and x_t, may be related to past lags of the x-series the series

Cross-correlation
Is measure of similarity of two series as a function of the displacement of one relative to the other

CCF: Sample cross correlation function
- used to identify lags of the x-variable that might be useful predictors of y_t
- sample CCF: defined as set of sample correlations between x_{t+h} and y_t
- R: ccf(x-variable name, y-variable name)

Example
Series are correlated at lag 1 and 4



-4	-3	-2	-1	0	1	2	3	4
0.862	0.021	-0.547	-0.423	0.000	0.867	0.127	-0.466	-0.393

White Noise Process

White Noise
- random process that is regarded as
  sequence of serially uncorrelated random variables with zero mean and constant variance
- variables are independent and identically distributed with a mean of zero and constant variance
  => each value has a zero correlation with all other values in series
- is sequence of random numbers and cannot be predicted
- Gaussian white noise: if distribution is normal
- Series residuals not white noise: there is still information / signal => improve model

Shock
A single realization of white noise is a random shock

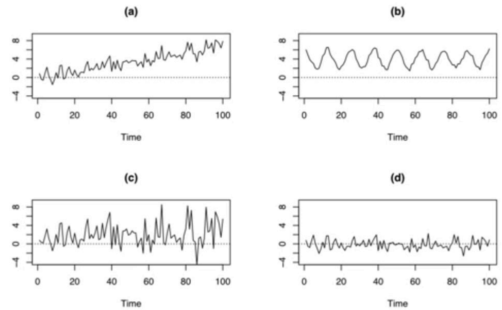
Definition
Series $X_1, X_2, ...$ is white noise process with mean μ and variance σ^2 - $WN(\mu, \sigma^2)$
if for all t

$$X_t = \mu + \epsilon_t$$

where
$E(X_t) = \mu \qquad Var(X_t) = \sigma^2 \qquad Cov(X_t, X_s) = 0$ for all $t \neq s$

Properties
- past values contain no information that allows better forecast
- Best predictor: mean μ
- is ARIMA(0, 0, 0) model

Examples of (not) whitenoise
a = upward trend
b = periodic
c = increasing variance
d = white noise



Random Walk Process

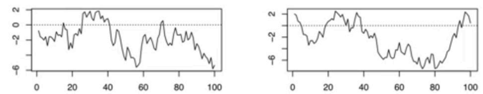
Random walk
- basic time series model
- random process, that describes a path consisting of succession of random steps
- eg. Random walk on integer number line: starts at 0; at each step moves +1 or −1 with equal probability

Intuition
Random Walk: Today = Yesterday + Noise
Random Walk with Drift: Today = Constant + Yesterday + Noise

Definition
Series $X_1, X_2, ...$ is Random Walk, if for all t

$$X_t = X_{t-1} + \epsilon_t$$

,where ϵ_t is mean zero white noise



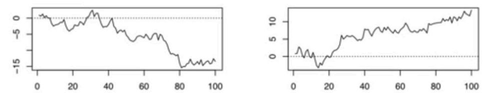
Simulation
Requires initial point Y_0 and only one parameter - WN variance σ_ϵ^2

Random walk <=> White noise
- White noise => Random walk: integration / cumulative sum of white noise series yields random walk
- Random walk => White noise: first difference series of random walk yields white noise series with mean c
  Random walk is recursive white noise data
  => removing long-term trend results in white noise

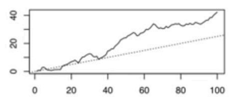
Properties
- is ARIMA(0, 1, 0) = order of integration = 1 => needs differencing one to be white-noise

Random walk with drift
- Drift: = time trend = upward or downward trajectory
- First difference: also yields white noise proces
- Example: Mean = for drift ;   arima.sim(model = list(order = c(0, 1, 0)), n = 100, mean = 1)
- Model:
  Includes intercept in model = slope of the time-trend

$$X_t = c + X_{t-1} + \epsilon_t$$



Drift & Coefficient
Bigger coefficient = steeper trend



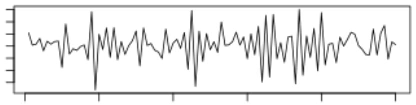
Persistence

Persistence
- Persistence: is high correlation between an observation and its lag
  = how close are observations to their neighbors
- Anti-Persistence: defined by a large amount of variation between an observation and its lag

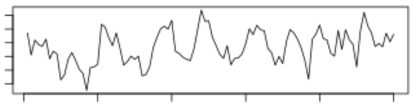
- AR processes can exhibit varying levels of persistence as well as anti-persistence or oscillatory behavior.

Examples

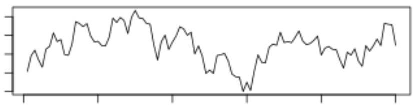
Anti-persistence & oscillation



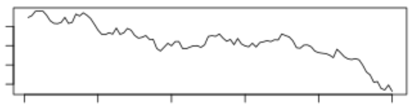
Low persistence



Higher persistence



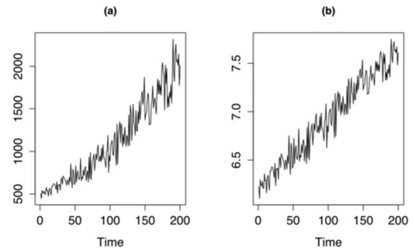
Very high persistence,
with downward trend



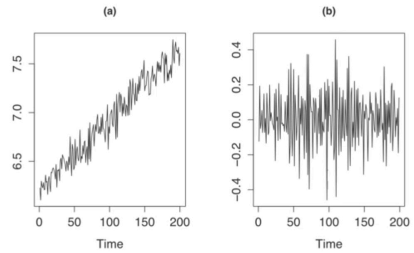
Transformations

Simple transformations

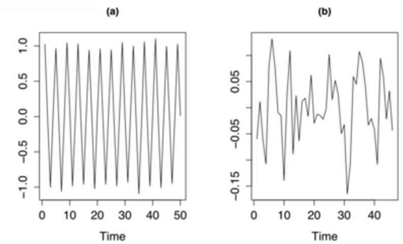
Exponential growth => linear growth : log()



Remove Linear Trend : diff()
First-Difference-Series
Values represent the difference between
one value and the next in original series
=> Result: difference series / change series



Remove seasonal trend : diff(x, s=4)
Seasonal difference transformation
Example: Cycle-Length = 4



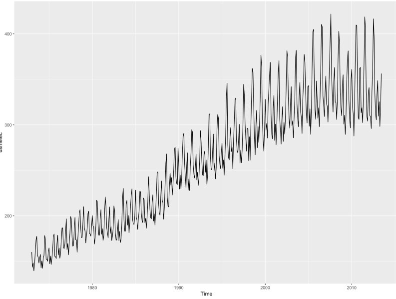
Box-Cox Transformation

Used in order to **stabilize variance**

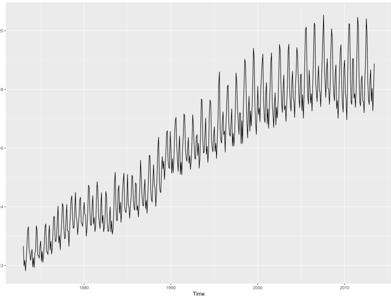
$$w_t = \begin{cases} \log(y_t) & \textbf{if } \lambda = 0 \\ (y_t^\lambda - 1)/\lambda & \text{otherwise} \end{cases}$$

λ = 1: no transformation (subtracts 1)
λ = 1/2: Square Root
λ = 1/3: Cube root
λ = 0: Natural logarithm
λ = -1: Inverse transformation

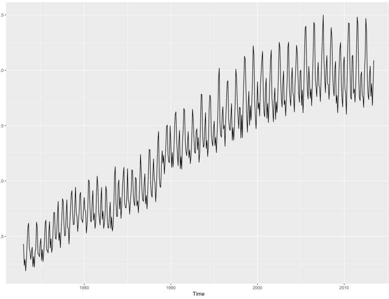
=> possibly needs bias-adjustment (biasadj=TRUE)



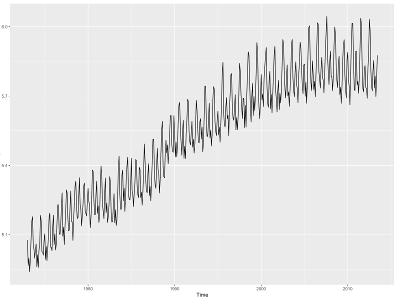
Square-Root = λ = 1/2 => Not enough



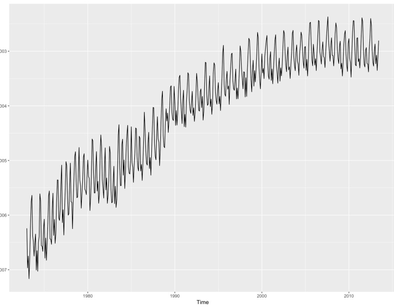
Cube-Root = λ = 1/3 => Not enough



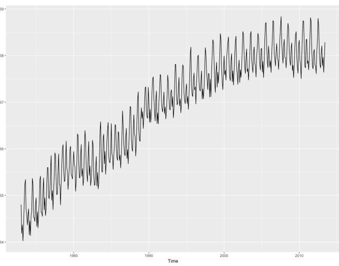
Log Transform = λ = 0 => Still too much on the right



Inverse Transform = λ = 1 ==> too much



Needs something between log and Inverse
Box-Cox => λ = -0.5738331 = stable variance



Mathematical transformations for stabilizing variation		
Square Root	$w_t = \sqrt{y_t}$	↓
Cube Root	$w_t = \sqrt[3]{y_t}$	Increasing
Logarithm	$w_t = \log(y_t)$	strength
Inverse	$w_t = -1/y_t$	↓

Forecasting

Forecasting
- process of making predictions of future based on patterns in past and present data
- most commonly by analysis of trends
- eg. warehouse manager models how much product to order for the next 3 months based on previous 12
- different methods: trend analysis, decomposition, single exponential smoothing to extrapolate those patterns to the future
- Choosing analysis method by:
  - nature of the trend seasonal components
  - how far ahead to forecast
- vs prediction: similar, but more general term

Qualitative vs. quantitative Forecasting
- Qualitative: subjective
  - based on the opinion and judgment of consumers and experts
  - appropriate when past data are not available
  - applied to intermediate- or long-range decisions

- Quantitative: models
  - uses models to forecasts future data as a function of past data
  - appropriate when past numerical data available
  - reasonable to assume that some patterns in data expected to continue into future
  - applied to short- & intermediate-range decisions

CI & Four sources of uncertainty for n-step ahead forecast in AR models
1. is autoregressive model the correct model
2. true values of the autoregressive coefficients
3. accuracy n to n-k steps forcasts used as lagged values for forecast n+k+1
4. value of the error term ε_t for each for the period being predicted

=> Confidence interval constructed by using 2,3,4 for the n-step-ahead predictions
CI becomes wider as n increases - because of increasing number of estimated values

Forecasting Accuracy

Residual / Forecast Error
Is the difference between the actual value and the forecast value for the corresponding period

$$e = y_t - \hat{y}_t$$

=> good forecasting methods yield uncorrelated, zero mean, constant variance & normally distributed resids

Residuals: Necessary Properties
- Uncorrelated:
  Otherwise still information in data, which should be captured by model
- Mean of Zero:
  Otherwise forecast would be biased and easily be fixed by adjusting forecast

Residuals: Optional Properties
- properties important for computing the prediction interval
- point-forecast can still be good & used
  => prediction interval might be too wide or narrow
- Constant variance
- Normally distributed

Adjustment
- Correlation: between residual values
  - then still information left in residuals which should be used in computing forecasts
  - computing expected value of residual as function of the known past residuals
  - adjusting forecast by amount by which expected value differs from zero
- residuals have a mean other than zero, then the forecasts are biased

Forecasting accuracy measures
- Scale-dependent errors:
  Forecast error is on the same scale as the data => cannot compare between two different series

  MAE: Mean Absolute Error: $MAE = average(|e_t|)$
  MSE: Mean Squared Error: $MSE = average(e_t^2)$

- Percentage Errors
  Forecast error is scale-independent => can compare between different series
  Disadvantage: if Y is close to or equal to zero, then extremely large / undefined

  MAPE: Mean Absolute Percentage Error: $MAPE = 100 \times average(|\frac{e_t}{y_t}|)$
  => only if values are positive and have no zeros or small values
  Assumes there is a natural 0 (eg. not celsius or fahrenheit because of arbitrary 0 points)

- Scaled Errors:
  As alternative to percentage errors

  MASE: Mean Absolute Scaled Error: $MASE = MAE/Q$
  => like MAE, but scaled using Q and so can be used across series

Backtesting: Forecast model Evaluation

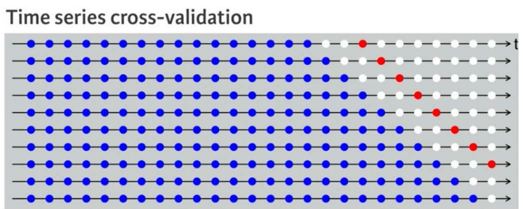
Backtesting
- name for model evaluation in time series forecasting
- also hindcasting

Two types of forecasts performances
1. one-step-ahead: all observations prior to the except the one being predicted used for training
2. n-step-ahead:
   - obtain forecasts for each of the out-of-sample observations
   - evaluate the quality of n-step-ahead forecasts using the CI

Growing Uncertainty: problem **n-step-ahead forecasts**
- each of the n-steps is forecast and uses previous observations
- step k+1 in forecast horizon is based on the estimates of the k previous periods
  => eg. more and more values on the right side of the AR-equation are estimates
=> uncertainty grows with each step

Train-Test Split
- split has to respect temporal order of observations
- train-set: used for parameter estimation
- test-set: held back for out-of-sample testing.

Cross Validation
- also: Forecast Evaluation on a rolling origin)
- point of origin = last data-point in each
- train on all data before point of origin and predict
- also with multiple steps in the future



Forecasting methods

Forecasting categories
- Naive Forecasting
- Mean Forecasting
- Time Series Models (ARIMA,ETS)
- Neural Networks

Naive Forecasting
- simplest form of forecasting methods
- use most recent observation
  => forecast for time t is the data value at time t-1
- used as benchmark
  - to establish benchmark for time series model
  - compare accuracy measures of the naive model to model using different method
    => naive model is better fit, then shouldn't use the other model since naive is a better fit & more simple
- vs others:
  - with moving average with moving average length = 1
  - single exponential smoothing with weight=1
- Variations
  - Drift method
    - allows forecasts to increase or decrease over time
    - amount of change over time = drift set to average change in historical data
    - equivalent to drawing line between first and last observation and extrapolating into future
  - Seasonal naïve approach
    - accounts for seasonality
    - sets prediction to be equal to the last observed value of the same season
    - eg. prediction for all subsequent months of April will be equal to previous value observed for April

Mean Forecasting
- predictions of all future values equal to average of all observations

Autoregressive Model

Autoregressive model
Specifies that output variable depends linearly on
1. its own previous value
2. stochastic term / imperfectly predictable term
=> term autoregression indicates that it is a regression of the variable against itself
- includes the WN and RW (slope parameter = 1) models examined as special cases.

Multiple regression model vs autoregressive model
Multiple regression: response is linear combination of predictors
Autoregressive model: forecast is linear combination of past values of the variable
=> similar interpretation ass linear regression, but each observation is regressed on previous observation

Used to model long-run autocorrelation

Definition
$AR(p)$ = autoregressive model of order p, defined as

$$X_t = c + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \ldots + \varphi_p X_{t-p} + \varepsilon_t$$
$$= c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

where:
$\varphi_1, \ldots, \varphi_p$ = parameters / slope of the model
c = constant / mean
ε_t = white noise = independent and normally distributed with mean 0 and const. variance
=> **like multiple regression, but with lagged values of the X_t**

Different Notation
Backshift operator: $X_t = c + \sum_{i=1}^{p} \varphi_i B^i X_t + \varepsilon_t$; Polynomial: $\phi[B]X_t = c + \varepsilon_t$

Stationarity
- parameter constraints necessary for the model to be weakly stationary
- eg. AR(1) model with $|\varphi_1| \geq 1$ are not stationary
- Formal constraint: each unit root z_i must satisfy: $|z_i| > 1$

Effect of random shocks(!)
- one-time shocks affect values of variable infinitely far into the future
- Example: AR(1) model
  - non-zero value for ε_t at t=1 affects X_1 by the amount ε_1
  - Then: by AR equation for
    - this affects X_2 in terms of X_1 by amount of $\varphi_1 \varepsilon_1$
    - this affects X_3 in terms of X_2 by amount of $\varphi_1^2 \varepsilon_1$
    - …..
    => effect of ε_1 never ends
    but if the process is stationary => effect diminishes toward zero in the limit

Determining maximum lag p
- PACF cuts off at lag p => last lag carries information
Parameter estimation: OLS

AR(1) model: Process & Series

Intuition Today = Constant + Slope * Yesterday + Noise

Definition
Value of X_t is a linear function of the value of X at time $t-1$

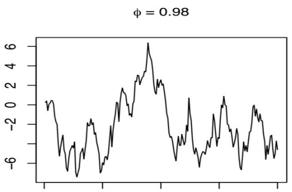
$$X_t = c + \varphi_1 X_{t-1} + \varepsilon_t \quad \text{where } \varphi_1 \text{ is slope}$$

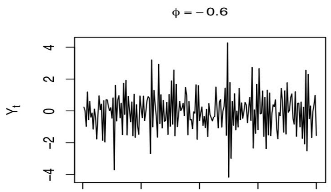
Expected Value $\phi < 1$ => $E(Y_t) = \mu$ $\qquad Var(Y_t) = \sigma_Y^2 = \dfrac{\sigma_\epsilon^2}{1-\sigma^2}$

Stationary: for slope parameter -1 < ϕ < 1
if ϕ =1 then variance of X_t depends on lag t, so that variance diverges and goes to ∞ as t goes to ∞

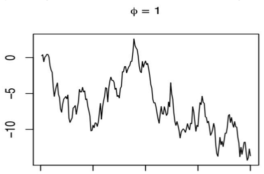
Slope & Properties of the process / series
Zero vs Non-Zero
ϕ_1 = 0 X = white noise
ϕ_1 <> 0 X = autocorrelated and depends on ϵ_t and X_{t-1}
=> X_{t-1} fed forward in X_t
ϕ determines amount of feedback - larger values of $|\phi|$ = more feedback

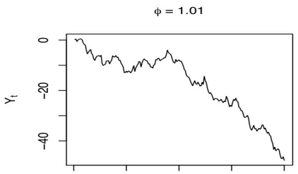
Positive vs Negative
ϕ_1 > 0 greater value = series has higher persistence = each observation close to its neighbors
= ACF slower decaying
Closer to 1 higher persistence
but process reverts to its mean quickly
=> ACF also decays to 0 at quick (geometric) rate
indicating that values far in the past have little impact on future values of the process
ϕ_1 < 0 X = oscillates between positive and negative values





Greater > 1
> 1 => very high dependence that series is quickly diverges





Special Cases
ϕ_1 = 0 , c = 1 X = random walk
ϕ_1 = 1, c <> 1 X = random walk with drift => not stationary

AR(1) model: Autocorrelation, ACF, & PACF

Exponential Decay of autocorrelation
Correlation between observations that are h time periods apart follows an exponential decay

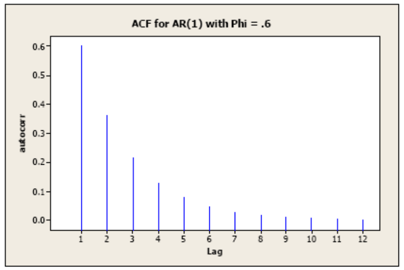
$$Corr(X_t, X_{t-h}) = \rho(h) = \varphi_1^{|h|} \text{ for all h}$$

Explanation: Effect of shocks
==> see exponential decay in ACF plots (in terms of positive and negative)

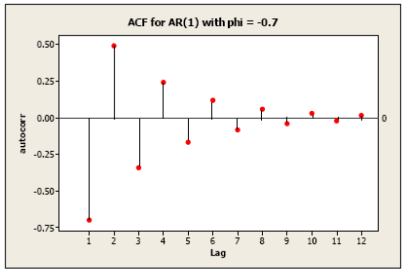
Exponential decay
A quantity is subject to exponential decay if it decreases at a rate proportional to its current value

ACF: Positive & negative slope values
- positive ϕ_1: ACF exponentially decreases to 0 as lag h increases
- negative ϕ_1: ACF exponentially decreases to 0 as lag h increases & signs alternate
- Example:

ϕ = **0.6**

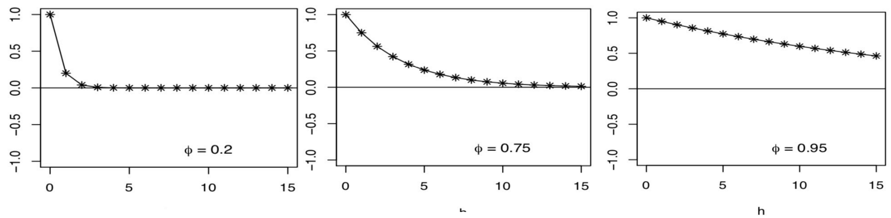


ϕ = **-0.7**

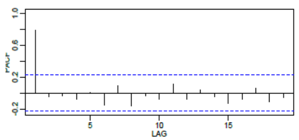


ACF: Positive Values
Nearer 1 ==> Slower decaying ACF

ϕ = **0.2** slower than ϕ = **0.75** slower than ϕ = **0.9**

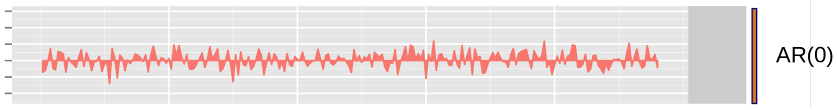


PACF
Will show only cuts off at lag 1 at height of ϕ => because only last information carries information

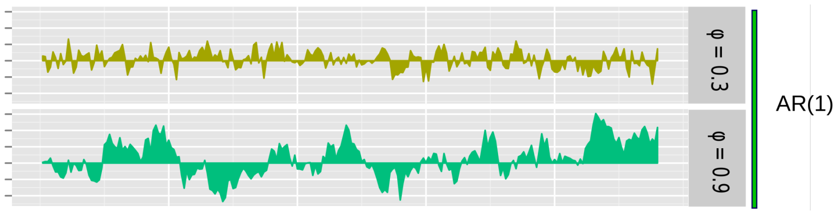


AR(p) Models: for p = 0,1,2

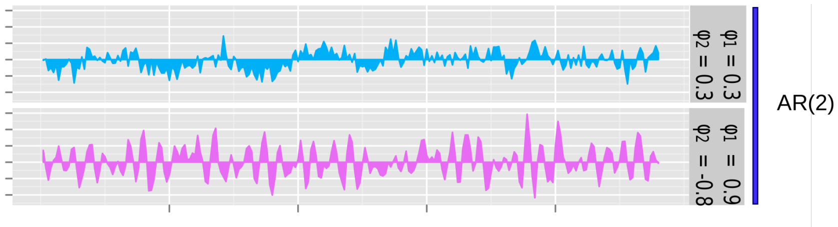
AR(0)
- AR(0) corresponds to white noise
- has no dependence between the terms
- only the error/innovation/noise term contributes to output of process



AR(1)
- Positive φ: only previous term and noise term contribute to output
- φ close to 0: process still looks like white noise, but
- φ approaches 1: larger contribution from previous term relative to noise
  => results in smoothing / integration of the output, similar to low pass filter
  => higher persistence
- $\varphi < 0$: acts has a high-pass filter



AR(2)
- previous two terms and noise term contribute to output
- both φ_1 and φ_2 positive: output resembles low pass filter, with high frequency part of noise decreased
- φ_1 positive and φ_2 negative:
  - process favors changes in sign between terms of the process => output oscillates
  - seen as detection of change in direction
- φ_1 negative acts as high-pass filter



Moving-average model

Moving-average model
Specifies that output variable depends linearly on
1. current value
2. various past values of a stochastic term = imperfectly predictable = error

Used to model very short-run autocorrelation

Intuition
- is conceptually a linear regression, in which current value is regressed on:
  - current - ε_t - and previous (observed) $\varepsilon_{t-1,2,...}$ white noise error terms / random shocks
    eg. MA(1) = regressed on yesterdays error
- Random shocks: at each point
  - assumed to be mutually independent
  - come from same distribution - normal distribution, with location at zero and constant scale = variance
- X can be thought of as weighted moving average of past few forecast errors

Defintion
MA(q) = moving average model of order q

$$X_t = \mu + \varepsilon_t + \theta_1\varepsilon_{t-1} + \cdots + \theta_q\varepsilon_{t-q}$$

where
μ = mean of series
$\theta_1, \ldots, \theta_q$ = parameters of model
$\varepsilon_t, \varepsilon_{t-1}, \ldots, \varepsilon_{t-q}$ = white noise error terms

Written using backshift operator $X_t = \mu + (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t$

Stationarity MA models are always stationary

Effect of random shocks(!)
Effect of random shocks differs in MA from AR model in 2 ways:
1. Errors propagated to future directly:
   e.g. ε_{t-1} appears directly on the right hand side of MA equation for X_t
   AR: ε_{t-1} only appears on RHS of X_{t-1}, which itself is in X_t equation
   => only an indirect effect of ε_{t-1} on X_t
2. Shocks only affect current period and q periods into future
   AR: shock affects values infinitely far into future, because ε_t affects $X_t, X_{t+1}, X_{t+2} \cdots$

Determining order q
- ACF of an MA(q) process is zero at lag q + 1 and greater
  => no correlation beyond what error-terms are included
- ACF for MA(2): only 2 non-zero value - for lag 1 and 2

Parameter estimation:
Problematic: lagged error terms are not observable
Solution: iterative non-linear fitting procedures

MA(1) model: ACF, Autocorrelation & PACF

Intuition Today = Constant + Slope * Yesterday's Noise + Noise

Definition
Value of X_t is a linear function of the random error of X at time $t-1$

$$X_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1}$$

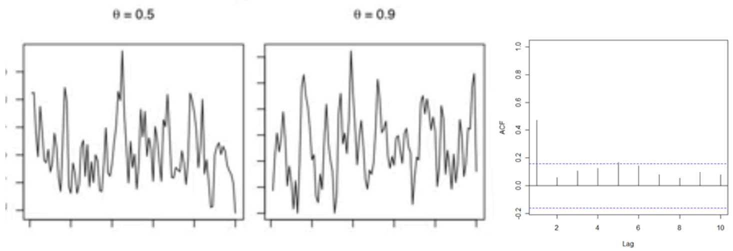
Expected Value $E(y_t) = \mu$ $\quad Var(y_t) = \sigma_\epsilon^2(1 + \theta_1^2)$

Slope & Properties of the process / series
<u>Zero vs Non-Zero</u>
$\theta = 0$ $\quad X_t = X_{t-1} + \epsilon_t$=> White noise
$\theta \neq 0,$ $\quad X_t$ depends on both ϵ_t and ϵ_{t-1} and process <u>X is autocorrelated</u>
=> ϵ_{t-1} is fed forward into X_t , where θ determines impact
larger values = greater impact

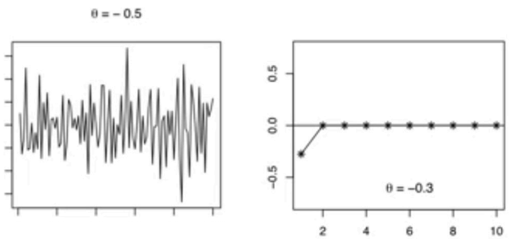
<u>Greater or Less than 0</u>
$\theta > 0$ $\quad$ larger values θ
=> higher autocorrelation (but still only at lag 1)
=> higher persistence
$\theta < 0$ $\quad$ oscillatory series

Series & ACF

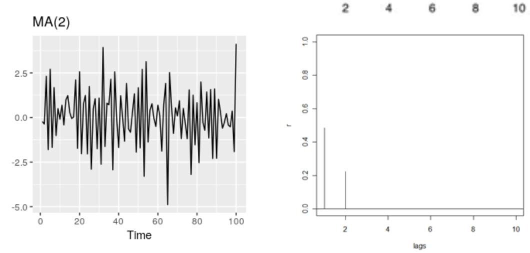
<u>Positive θ</u>
(0.9 =
higher persistence)



<u>Negative θ</u>
(oscillation)



MA(2)



Autoregressive VS Moving Average Models

Differ in
- Regression:
  Todays values depends on yesterdays value vs yesterdays noise
- How they handle shocks
- Autocorrelation
-

Main distinction: Regression
AR: uses past values of the forecast variable in regression model
MA: uses past forecast errors in regression model

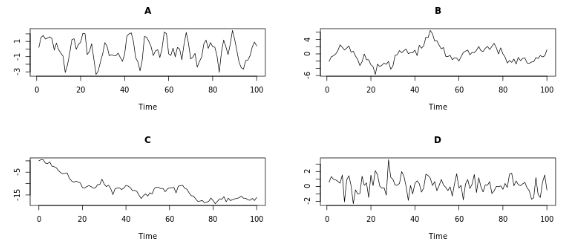
Example: First Order Models
<u>Todays observation</u> regressed on:
<u>AR</u>: yesterdays observation X_{t-1}
<u>MA</u>: yesterdays noise ϵ_{t-1}

Autocorrelation
<u>AR</u>: to model <u>long-run</u> autocorrelation
<u>MA</u>: to model <u>very short-run</u> autocorrelation

Short vs long-term models
Concerns: for how long does a value influence future values
=> how abnormal yesterdays value was compared to what was predicted has residual effect on today
=> effect of shocks
AR: Long Memory Models: effect dissipates slowly
MA: Short-memory models: effects quickly disappear completely
=> MA often shows short run depends but quickly reverts to the mean

Model Examples
Grafik: (A) MA, (B) AR, (C) RW, (D) WN
MA
shows short-run dependence
but reverts quickly to the mean

AR and RW, respectively. Series D does not show any clear patterns, so it must be the WN model.



ARMA

ARMA
- **Autoregression with correlated errors**
- combination of the AR and MA model

=> natural for time series because they are usually autocorrelated
- ARMA model needs stationary data

Usage
If process is function of:
1. series of unobserved shocks - MA,
2. and its own behavior

Example Stock prices
1. shocked by fundamental information
2. exhibiting technical trending and mean-reversion effects due to market participants

Definition
ARMA(p, q) = model with p autoregressive terms and q moving-average terms

$$X_t = c + \varepsilon_t + \underbrace{\sum_{i=1}^{p} \varphi_i X_{t-i}}_{\textbf{Autoregression}} + \underbrace{\sum_{i=1}^{q} \theta_i \varepsilon_{t-i}}_{\textbf{with correlated errors}}$$

ACF Decay, that starts after a few lags => Mix of AR and MR components (p,q ≥ 1)

ARMAX
- Autoregressive–moving-average model **with exogenous inputs** model
- R implements ARMAX model through use of exogenous / independent variables
- contains AR(p) and MA(q) models and a linear combination of the last b terms of a known and external time series d_t

ARMAX(p, q, b) = model with p AR terms, q MA terms and b exogenous inputs terms

$$X_t = \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \sum_{i=1}^{b} \eta_i d_{t-i}$$

where $\eta_1, \ldots, \eta_b$ are the parameters of the exogenous input d_t

VARMA
Extensions for the multivariate case:
VAR = vector autoregression
VARMA = Vector Autoregression Moving-Average

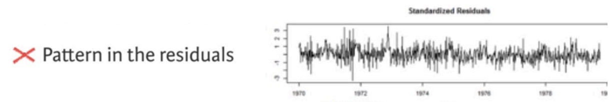
Residual Analysis

Residual Analysis
- residuals should be gaussian white noise - otherwise improve model

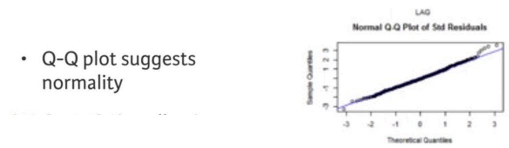
Importance: Forecast intervals
- Forecast intervals based on assumptions that residuals
  1. uncorrelated
  2. normally distributed
- If not => forecast intervals may be incorrect

1. **Checks: Normality**

Standardized residual plot: white noise sequence with mean 0 and variance 1

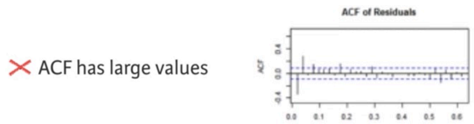


Histogram of the residuals: check for normality
QQ-Plot: check for normality of residuals

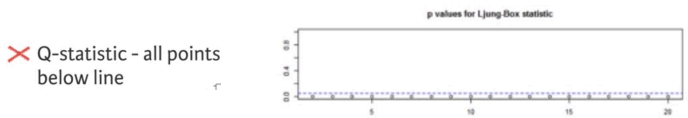


1. **Checks: Autocorrelation**

Sample-ACF of residuals: check for autocorrelation 95% of values should be between blue lines



Box-Ljung-Pierce Q-statistic: tests for whiteness of residuals - 95% should be **ABOVE** blue line



ARIMA

ARIMA = AutoRegressive Integrated Moving Average

(where "integration" = reverse of differencing)

Autoregressive (AR) models
$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t$, $e_t \sim$ white noise
Multiple regression with *lagged observations* as predictors

Moving Average (MA) models
$y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q}$, $e_t \sim$ white noise
Multiple regression with *lagged errors* as predictors

Autoregressive Moving Average (ARMA) models
$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t$
Multiple regression with *lagged observations* and *lagged errors* as predictors

ARIMA(p, d, q) models
Combine ARMA model with d - lots of *differencing*

ARiMA(p,d,q)-Model

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

where
p = order of the autoregressive part
d = degree of first differencing involved
q = order of the moving average part

y'_t is the differenced series (may have been differenced more than once)

Backshift Notation:

$$\underbrace{(1 - \phi_1 B - \cdots - \phi_p B^p)}_{AR(p)} \underbrace{(1 - B)^d y_t}_{d-differences} = \underbrace{c + (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t}_{MA(q)}$$

Special Cases

White Noise:	ARIMA(0,0,0)
Random Walk:	ARIMA(0,1,0) with no constant
Random Walk with drift:	ARIMA(0,1,0) with a constant
Autoregression:	ARIMA(p,0,0)
Moving Average:	ARIMA(0,0,q)

ARIMA vs ARMA
- ARIMA = Integrated ARMA
- Time series is ARIMA(p,d,q) if the differenced series (of order d) is ARMA(p,q)
  => series exhibits ARIMA behavior if the differenced data has ARMA behavior
- to model non-stationary time-series

auto.arima()
- Number of differences d: via unit root tests
- p and q: by minimizing AICc
- Estimate parameters: maximum likelihood estimation
- Save time by using stepwise search to traverse model space, to

Unit Root
- model has 1 unit root if it needs 1 level of differencing to make it stationary
- model has 2 unit root if it needs 2 levels of differencing to make it stationary

Box-Jenkins Method

Idea behind ARMA & ARIMA model fitting in 3 stages

1. **Model identification and model selection**

In ARIMA: trend and seasonal effects are treated as nuisance parameters
=> need to be removed from series before analysis

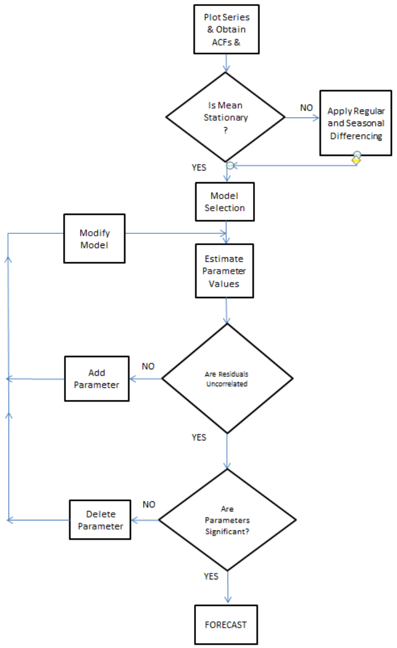
1a. **Make series stationary & Detect seasonality**
1. Detect stationarity: using runs, ACF
   => use differencing to make stationary
2. Detect & Identify seasonality: using ACF and seasonal subseries plot
   - in model identification phase: only detect seasonality & identify order of seasonal AR and MA terms
   - not explicitly remove the seasonality before fitting the model - instead just specify for software
   - BUT: seasonal differencing helpful for model identification of non-seasonal component of model

- Which AR,MA and order of terms: ACF & PACF to decide which should be used

1b. **Identify p and q**
Parsimony principle
Choose the simplest scientific explanation that fits the evidence

2. **Parameter estimation**
Use MLE to estimate coefficients

3. **Model checking**
- Check: if resulting estimated model is a stationary univariate process
- Residuals Analysis: identify misspecification
  => if estimation is inadequate, return to step one and attempt to build a better model



Model-Identification: Finding p and q

Problem: Model-Identification
Goal: Identify model by looking at the raw time series data
Problem: MA und AR model plots very similar

Solution: ACF and PACF
Idea:
Use sample ACF, PACF plots and compare to
theoretical behavior of these plots when the order is known
=> usually only able to see model type - hard to determine concrete order - especially for mixed models

Choosing p and q: automatically
Gridsearch & AIC/BIC

Choosing p and q: manually
Basic Approach
1: First estimate:
  p using PACF
  q using ACF
2. Check residuals using PACF and ACF to refine

Usual problem with approach: multiple models reasonable
- mixed models particularly difficult to identify
  => even with experience, using ACF & PACF requires trial and error
- Solution:
  - build models and compare using AIC, BIC
  - Extreme case: use gridsearch and & start small - ARMA(1,1) and add parameters as needed

AR
- ACF: often mixture of exponentially decreasing and damped sinusoidal components
- PACF: spike at p, none beyond lag p + 1

MA
- ACF: spike at lag q, but none beyond lag q
- PACF: exponentially decaying or sinusoidal
  => PACF not helpful for determining order

ACF Plot
Exponential, decaying to 0: AR => use PACF for order
Alternating, decaying to 0: AR => use PACF for order
>=1 spikes, rest essentially 0: MA => order where plot becomes 0
Decay starting after few lags: ARMA model
All 0 or close to 0: Random data
High values at fixed intervals: Seasonal AR
No decay to 0 Not stationary

	AR(p)	MA(q)	ARMA(p, q)
ACF	Tails off	Cuts off lag q	Tails off
PACF	Cuts off lag p	Tails off	Tails off

Seasonality

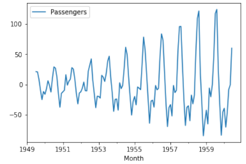
Seasonality
- presence of variations that occur at specific regular intervals - eg. weekly, monthly, or quarterly
- consists of periodic, repetitive, regular and predictable patterns in the levels of a time series
- caused by various factors - eg. weather, vacation, and holidays

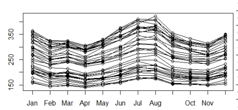
Seasonality vs cyclical patterns
- Cyclical patterns: data exhibits rises and falls that are not of a fixed period
  eg. due to economic conditions and often related to the business cycle

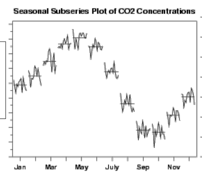
Motivation
- often interested in knowing performance relative to the normal seasonal variation
- regular changes are of less interest than underlying trend
  => seasonal adjustment by removing = subtracting the seasonal component (additive model)
- used in forecasting

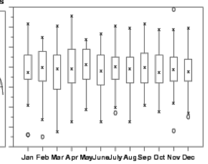
Example:
- ice-cream vendors interested in knowing their performance relative to the normal seasonal variation

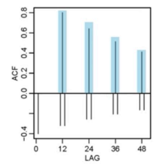
Detecting seasonality
- Good approach for Inspection: Detrending
  1. to find periodicity in series of data => find freq
  2. remove any overall trend first and inspect time periodicity => ma & remove trend
- Run sequence plot
- Seasonal plot: shows season overlapped - picture
- Seasonal subseries: specialized technique for showing seasonality
- Box plots: as alternative to seasonal subseries
- ACF: spikes at seasonal lags

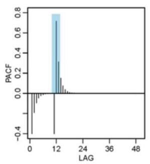




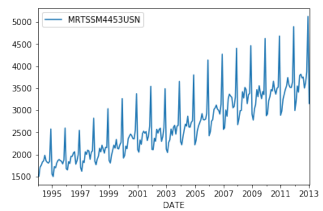


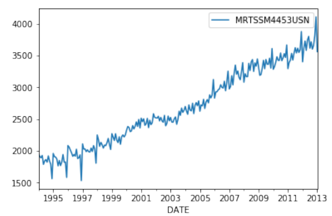






Seasonal adjustment
- removing seasonal component from the original data
- Goal: analyze trend, and cyclical deviations from trend independently of seasonal components
- seasonal_adj = decomp.observed - decomp.seasonal
  seasonal_adj.plot();





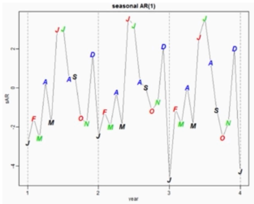
Seasonal variation: Seasonal index
- Seasonal variation measured in terms of the seasonal index
- is average used to compare actual observation relative to situation of no seasonal variation
- index value: is attached to each period of the time series within a season
  => 12 separate seasonal indices for monthy data
- several methods use seasonal indices to measure seasonal variations
- Ratio-to-moving-average method:
  - provides an index to measure the degree of the seasonal variation
  - index based on a mean of 100
  - degree of seasonality measured by variations away from base
  - express original data value of series as percentage of corresponding centered moving average value

Regression analysis & seasonailty
- if seasonally varying dependent variable is influenced by one or more independent variables
- seasonality accounted for by including n-1 dummy variable - one for each of the seasons - reference

Handling multiple seasonalities
- adjusting for muliple seasonalities,
- get each of them separately using decomposition

```
daily_components = sm.decompose(raw_series, freq=24)
weekly_components= sm.decompose(raw_series, freq=24*7)
adjusted = raw_series - daily_components['seasonal'] - weekly_components['seasonal']
```

Pure Seasonal Model
- pure seasonal ARMA time series
  is correlated at the seasonal lags only
- Seasonal AR: SAR(1)$_{s=12}$ => $X_t = \Phi X_{t-12} + W_t$
  = AR(1) model at the seasonal lag of 12
- Example**:** Average monthly temperature
  Temperature in march this year only depends
  on temperature in march last year plus noise
- SAR(1): value this month is related to last year's value X_{t-12}



Seasonal ARIMA Models

Seasonal ARIMA
- seasonal ARIMA model formed by including additional seasonal terms in the ARIMA models

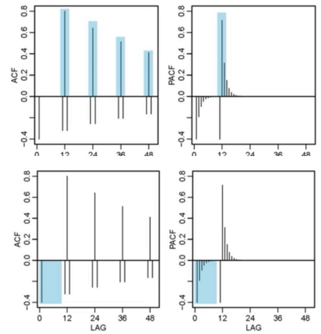
$$\text{ARIMA} \quad \underbrace{(p,d,q)}_{\text{non-seasonal part}} \quad \underbrace{(P,D,Q)_m}_{\text{seasonal part}}$$

where m = number of observations per year

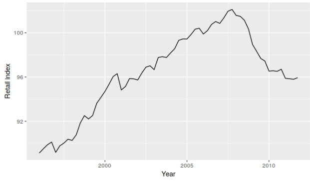
Lowercase = non-seasonal; Uppercase = seasonal

- Example: monthly data with yearly season $\text{SARIMA}(0,0,1)x(1,0,0)_{12}$

Determining seasonal part: P and Q
- FIRST: use differencing and seasonal differencing to make series stationary
- seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF
- similar interpretation as in non-seasonal model
- but at season-thresholds
  = 1 * s, 2 * s = 0,12,24…. (x-ticks)
- Interpretation according to pq-rules at:
  - Non-Seasonal Compont: 1-12
  - Seasonal Compont: 12,24,36
    => find appropriate seasonal orders by looking at seasonal lags
- Examples:
  - $\text{ARIMA}(0,0,0)(0,0,1)_{12}$
    ACF: spike at lag 12, but no other significant spikes
    PACF: exponential decay in the seasonal lags - i.e., at lags 12, 24, 36, …
  - $\text{ARIMA}(0,0,0)(1,0,0)_{12}$
    ACF: exponential decay in the seasonal lags
    PACF: single significant spike at lag 12 in the PACF

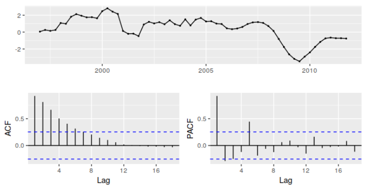


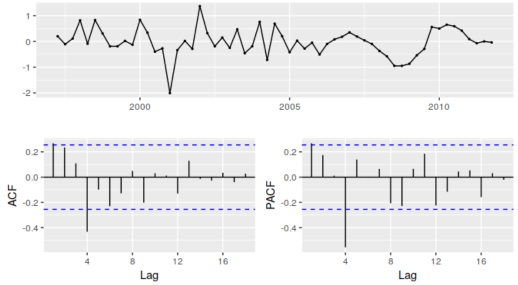
Modeling Seasonal ARIMA



Model with seasonal component & non-seasonal component

1. **Seasonal & First Differencing**
   Seasonal Diff: euretail %>% diff(lag=4) %>% diff() %>% ggtsdisplay()
   => seasonally differenced data still
   First Diff: now stationary





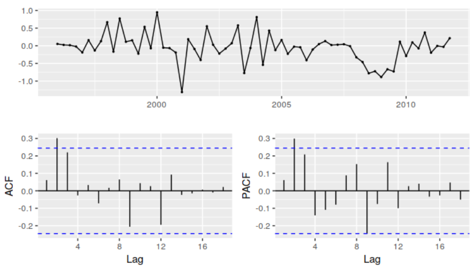
2. **Determine model components for initial model**
   ACF:
   significant spike at lag 1 => non-seasonal MA(1)
   significant spike at lag 4 => seasonal MA(1)
   (with same logic applied to PACF - could also started with $\text{ARIMA}(1,1,0)(1,1,0)_4$)
   => $\text{ARIMA}(0,1,1)(0,1,1)_4$

3. **Check Residuals & Find better model using AICc**
   a) Check resisuals: significant spikes at lag 2 and almost at 3
   => additional non-seasonal terms needed
   aa) Check Ljung-Box test if residuals have autocorrelations
   b) Add terms & Check AIC
   $\text{ARIMA}(0,1,2)(0,1,1)_4$ => AIC = 74.36
   $\text{ARIMA}(0,1,3)(0,1,1)_4$ => AIC = 68.53 => choose model!
   c) Check residuals again = > better!

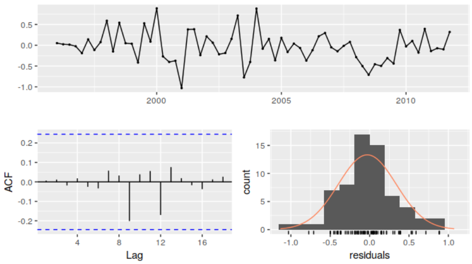
```
Ljung-Box test

data:  Residuals from ARIMA(0,1,3)(0,1,1)[4]
Q* = 0.51, df = 4, p-value = 1

Model df: 4.   Total lags used: 8
```

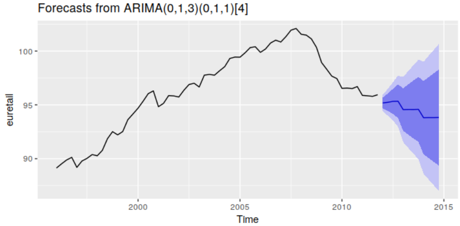


Residuals from ARIMA(0,1,3)(0,1,1)[4]



4. **Forecast**
   - point forecasts follow recent trend the data, because of the double differencing
   - prediction interval shows that it can increase or decrease at any time => allow upward

Forecasts from ARIMA(0,1,3)(0,1,1)[4]



Dynamic Regression

Motivation
- ARIMA and ETS uses only information from past observations
- Dynamic regression allows to incorporate of external information

From normal to dynamic regression

General Regression Equation: $y_t = \beta_0 + \beta_1 x_{1,t} + \ldots + \beta_k x_{k,t} + \varepsilon_t,$

In
Normal Regression: ε_t is assumed to be uncorrelated error-term - whitenoise (show Breusch-Godfrey test)
Dynamic Regression: ε_t follows ARIMA process

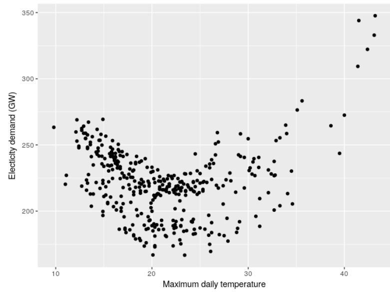
Example: ARIMA(1,1,1):

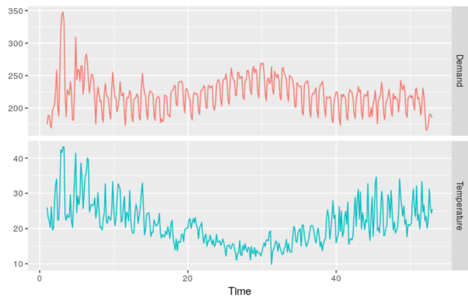
$$y_t = \beta_0 + \beta_1 x_{1,t} + \ldots + \beta_k x_{k,t} + \eta_t$$,where ε_t is whitenoise series.

Two error-terms:
1. Error-term of regression model: η_t
2. Error-term of ARIMA model: ε_t

Example: Forecasting demand of electricity
Instead of simple ARIMA model for demand use dynamic regression to include temperature information
=> Demand is high if cold, because of heater and high if hot, because of air-conditioning

```
xreg <- cbind(MaxTemp = elecdaily[, "Temperature"],
              MaxTempSq = elecdaily[, "Temperature"]^2,
              Workday = elecdaily[, "WorkDay"])
fit <- auto.arima(elecdaily[, "Demand"], xreg = xreg)
```

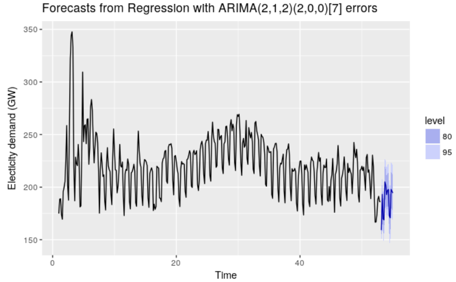




Forecast with temperature = 26 and working day = 1

```
fcast <- forecast(fit,
  xreg = cbind(rep(26,14),
               rep(26^2,14),
               c(0,1,0,0,1,1,1,1,1,0,0,1,1,1)))
```

Forecasts from Regression with ARIMA(2,1,2)(2,0,0)[7] errors



Dynamic harmonic regression

Used
If there are long seasonal periods, then dynamic regression with Fourier terms is often better

Arima & ETS vs haromic regression
- ARIMA & ETS: designed for shorter periods - eg. 12 for monthly, 4 for quarterly data
- Problem ETS:
  - Problem: m-1 parameters to be estimated for the initial seasonal states
    => estimation becomes almost impossible for larger seasonal periods
  - function restricts seasonality to be a maximum period of 24
- Problem: auto.arima
  - usually runs out of memory, when seasonal period m > 200
- Harmonic
  - seasonal pattern modeled using Fourier terms
  - short-term dynamics handled by an ARMA
    => allows any length seasonality
  - Disadvantage: seasonality is assumed to be fixed (vs ARIMA)

Handling Periodic seasonality by pairs of Fourier terms
Periodic seasonality handled using pairs of Fourier terms
=> every periodic function can be modeled by a numbers of pairs of Fourier terms

$$s_k(t) = sin\left(\frac{2\pi kt}{m}\right) \qquad c_k(t) = cos\left(\frac{2\pi kt}{m}\right)$$

$$y_t = \beta_0 + \sum_{k=1}^{k} [\alpha_k s_k(t) + \gamma_k c_k(t)] + e_t$$

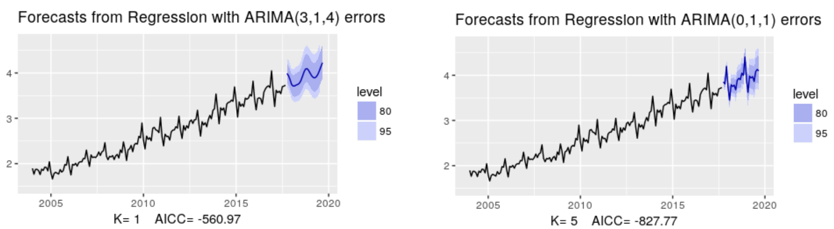
where:
k = **harmonic frequency** = number of pairs of fourier terms
m = seasonal period
α_k, γ_k = regression coefficients
e_t = modeled as a non-seasonal ARIMA process
=> assumes seasonal pattern is unchanging

Other predictor variables can be added: $\beta_1 x_{t,1}, \ldots, \beta_r x_{t,r}$

$$y_t = \beta_0 + \beta_1 x_{t,1} + \ldots + \beta_r x_{t,r} + \sum_{k=1}^{k} [\alpha_k s_k(t) + \gamma_k c_k(t)] + e_t$$

Choosing k
- determines how „wiggly“ line can be
- smoother for lower values of k
- chosen by minimize the AIC
- number of pairs of fourier terms

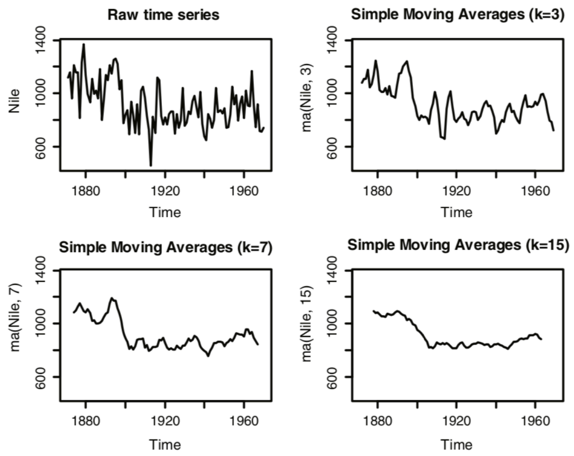
Example **K = 1** **K = 5**



Smoothing

Smoothing
- time series usually have significant irregular component
- Goal: discern patterns in data
- Problem: irregular components possibly hides pattern
- Solution: plot a smoothed curve that damps down fluctuations

Simple average smoothing
- simplest method of smoothing time series
- uses centered moving average: each data point replaced with the mean of surrounding observations
  = rolling mean(centered)
- k increases: plot becomes increasingly smoothed
- Challenge:
  Find value of k that highlights the major patterns in the data, without under- or over- smoothing



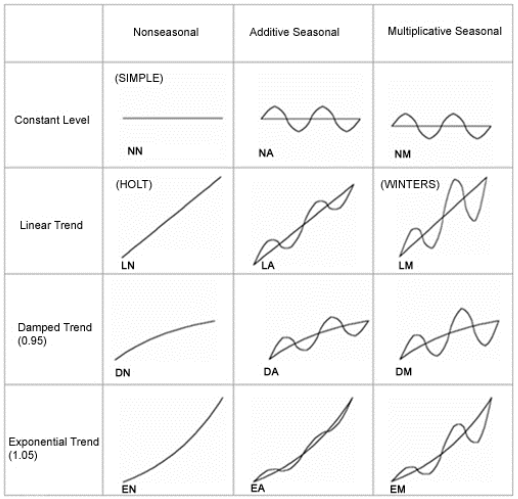
Exponential smoothing
- technique for smoothing time series data using an exponential window function
- vs simple moving average
  - simple moving average: past observations are weighted equally
  - exponential smoothing: exponential func. used to assign exponentially decreasing weights over time

Exponential smoothing acts as low-pass filter
- Exponential smoothing often applied to smooth signal in signal processing
- acts as low-pass filter to remove high frequency noise

Exponential Smoothing: Overview

Exponential Smoothing => Exponential Smoothing Methods => Exponential Smoothing Models
- <u>Exp. Smoothing</u>: itself does not predict values
- <u>Exp. Smoothing methods</u>: generate point forecasts
- <u>Exp. Smoothing models:</u> generate <u>distribution for the forecasts</u> and and hence also <u>confidence intervals</u>
  => yield good short-term predictions

Overview: Simple, Double & Triple Exponential Smoothing
- <u>**Differ in**</u>
  - **which components of series are modeled**
  - what they can describe
  - number of smoothing parameter and smoothing equations
    = how often smoothing is applied (for the different parameters)
- **Simple exponential**:
  - fits series that has constant level and irregular component at time i
    but has neither a trend nor a seasonal component
  - Smoothing: Data smoothing factor α
  - only has level equation => describes only the level of the ts - outputs same value for all forecasts
- **Double exponential**
  - also called a Holt exponential smoothing
    <u>Holt's linear trend</u>: trend is linear
    <u>Holt's damped trend</u>: uses damping factor
  - fits a time series with both a level and a trend
  - 2 parameters: α + trend smoothing factor, 0 < β < 1
- **Triple exponential**
  - also Holt-Winters exponential smoothing
    <u>Holt-Winters' additive method</u>: for additive time series
    <u>Holt-Winters' multiplicative method</u>: for multiplicative time series
  - fits time series with level, trend, and seasonal components
  - 3 parameters:
    α = Data smoothing factor, β = trend smoothing factor, γ= seasonal change smoothing factor 0< γ< 1
    => all parameters between 0 and 1



Simple exponential smoothing

*Modeling the **level** component of series*

Intuition
- can be seen as mix of naive and mean forecasting
- forecast based on all observations but recent ones are more heavily weighted

Weighted average
- uses weighted average of existing time-series values to make prediction
- weights chosen so that obs. have exponentially decreasing impact on average going back in time

= exponentially weighted moving average EWMA
= Exponentially weighted forecasting

One smoothing parameter - One smoothing equation
- <u>one smoothing factor</u> α: with $0 < \alpha < 1$
- <u>models only level</u> of series: $Y_t = \text{level} + \text{irregular}_t$
- <u>one smoothing equation</u>: in component form => level

Definition: Intuitive Forecast Equation using weighted average
- Notation: $\hat{Y}_{t+h|t}$ = point forecast of Y_{t+h} given data $y_1, \dots, y_t$
- Intuitive Forecast Equation:

$$\hat{Y}_{t+h|t} = \alpha y_t + \alpha(1-\alpha)t_{t-1} + \alpha(1-\alpha)t_{t-2} + \dots \quad \text{where} \quad 0 \le \alpha \le 1$$

=> weighted average all the data up to time t with weights decrease exponentially as it goes back in time

Definition: In terms of time series components
- <u>Forecast Equation</u>: $\hat{Y}_{t+h|t} = \ell_t$
  - forecast value at time t +1 is the estimated level at time t
  - independent of h => straight line
- <u>Smoothing Equation</u>: $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$
  - **models level** of the time series
  - gives estimated level of the series at each period t
    => how level changes over time as function of most recent obs and previous estimate of the levels
- <u>Unobserved level component ℓ_t</u>:- level of the series <u>given by the smoothed value</u> at time t
  => unobserved because cannot be determined what part of current value is level, trend, seas., irreg.

Smoothing Equation vs weighted average form
Replace: ℓ_t with $\hat{y}_{t+1|t}$ and ℓ_{t-1} with $\hat{y}_{t|t-1}$

Two parameters
- smoothing parameter <u>α</u> and <u>ℓ_0</u>
- <u>found</u> by <u>minimizing SSE</u> using MLE: $SSE = \Sigma_{t=1}^{T}(Y_t - \hat{Y}_{t|t-1})^2$

vs ARIMA
- can also be classified as autoregressive integrated moving average
- (ARIMA) (0,1,1) model with no constant term

(Data-) Smoothing Factor α

Name is misnomer
- bigger values of α: reduce the level of smoothing
- & Limiting case with α = 1: just returns current observation

Intuition α-parameter
- Determines
  - how much weight is placed on the most recent observation
  - how quickly the weights decay away

Values for α
- Values of α
  - α close to one = less of a smoothing effect = greater weight to recent changes in the data
  - α closer to zero have a greater smoothing effect and are less responsive to recent changes

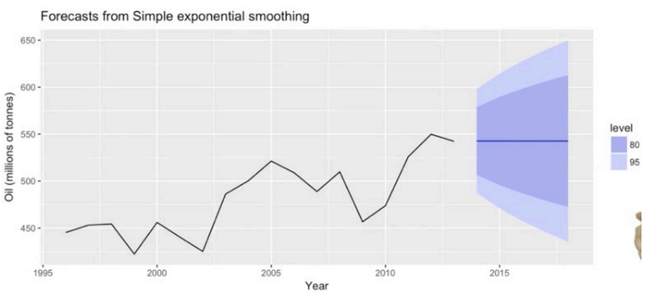
=> Greater α: more weight is placed on most recent observation and weights decay away quickly
=> Smaller α: less weight is placed on most recent observations and weights decay away slower

Observation	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
y_t	0.2	0.4	0.6	0.8
y_{t-1}	0.16	0.24	0.24	0.16
y_{t-2}	0.128	0.144	0.096	0.032
y_{t-3}	0.1024	0.0864	0.0384	0.0064
y_{t-4}	$(0.2)(0.8)^4$	$(0.4)(0.6)^4$	$(0.6)(0.4)^4$	$(0.8)(0.2)^4$
y_{t-5}	$(0.2)(0.8)^5$	$(0.4)(0.6)^5$	$(0.6)(0.4)^5$	$(0.8)(0.2)^5$

Choosing α
1. Experience
2. Estimate from observed data by minimizing sum of squared errors

Example
α = 0.83 (high)
=> value of forecast near most recent observation
83% of forecast is based on the most recent observation
14% on observation before that and 3% on remaining earlier observations

=> Note: same value for all forecasts - is estimated mean



Double exponential smoothing
*Modeling **level + trend** component of series*

Problem: Simple Exponential Smoothing
Only suitable if no trend and no seasonality

Intuition
- recursive application of an exponential filter twice
- Idea:
  - add term to account for the trend
  - slope component is itself updated via exponential smoothing

  => forecast function is no longer flat but trending
- 2 main methods: Hold's linear trend method & Hold's damped trend method

Two smoothing parameter & Two smoothing equations
- two smoothing factors: α and β

  α = smoothing parameter controlling exponential decay for the level

  β = smoothing parameter controlling exponential decay for the slope

  => larger values give more weight to most recent observations
- models level + trend of series: $Y_t = \text{level} + \text{trend} + \text{irregular}_t$
- two smoothing equation: in component form => level + trend

1. Hold's Linear Trend method

Forecast Equation: $\hat{Y}_{t+h|t} = \ell_t + h b_t$
=> forecast function is no longer flat but trending
The h-step-ahead forecast is equal to the last estimated level plus h times the last estimated trend value
=> hence forecast function is linear function of forecast horizon h, gives trended forecasts with slope = b_t

Smoothing Equation: $\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1})$
weighted average of observation y_t and the one-step-ahead training forecast for time t given by $\ell_{t-1} + b_{t-1}$
=> data is trended

Trend: $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$
b_t = weighted average of estimated trend at time t based on $\ell_t - \ell_{t-1}$ and b_{t-1} = previous estimate trend
=> how slope(!) changes over time
Since it changes, also called local linear trend

ℓ_t = estimate of the level of the series at time t
b_t = estimate of the trend (slope) of the series at time t
α = smoothing parameter for the level
β^* = smoothing parameter for the trend = how quickly slope can change
=> Small β^* = slope hardly changes => trend almost linear throughout series
=> Large β^* = slope changes rapidly => allows highly non-linear trends

where $0 \leq \alpha, \beta^* \leq 1$ are tuning-parameters
Goal: Estimate parameters => by minimizing SSE
α, β^* = smoothing parameters, ℓ_0, b_0 = **state parameters**

2. Hold's Damped Trend method

Motivation: Problem of Holt's linear method
- Holt's linear method
  - produces forecasts with linear trend
  - slope is constant and does not change over time
  - Result: trend is increasing / decreasing indefinitely
- Problem:
  - tends over-forecast
  - especially for longer forecast horizons
- Solution: damping the trend

Hold's Damped Trend method
- Idea: introduces parameter ϕ that dampens trend to a flat line
  => ϕ dampens trend so that trend gradually becomes flat = constant

- Result
  - short-run forecasts: trended
  - long-run forecasts: constant

- Converges to: $\ell_T + \phi b_T / (1-\phi)$ for $0 < \phi < 1$
- Definition

$$\hat{y}_{t+h|t} = \ell_t + h b_t \qquad \hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \ldots + \phi^h) b_t$$

$$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \quad \text{== damping ==>} \quad \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$$

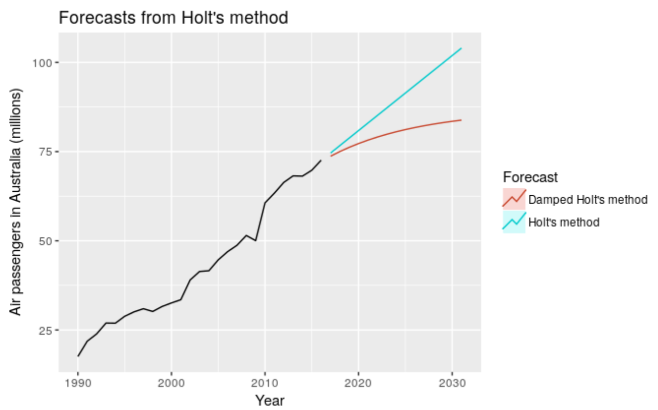
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*) b_{t-1} \qquad b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$$

$\phi = 1$ => identical to Holt's linear method

Values for damping parameter ϕ

ϕ rarely less than 0.8 => damping has a very strong effect for smaller values
ϕ close to 1 => damped model very similiar to non-damped
$\phi = 1$ => identical to Holt's linear method

Usually: ϕ = 0.8 to 0.98 (Example = 0.9, h = 15)



Triple Exponential smoothing
*Modeling **level + trend + seasonality** component of series*

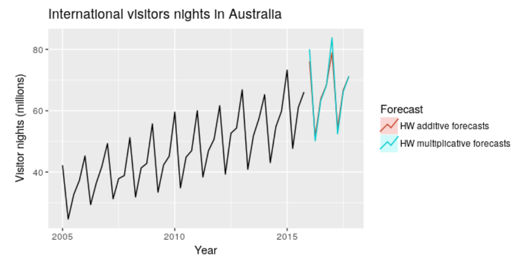
Holt-Winters' seasonal method
- method by Holt and Winters that extends Holt's method to capture seasonality

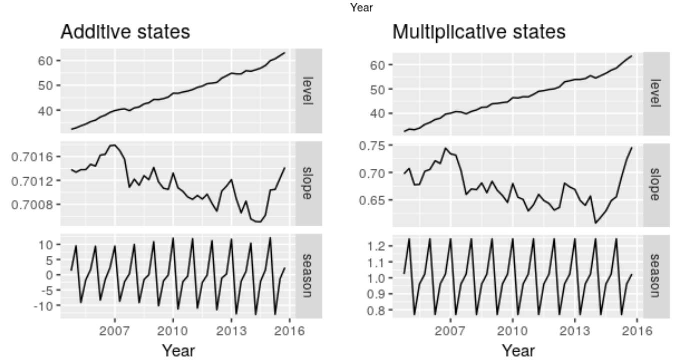
Three smoothing parameter
- three smoothing factors: α, β, γ => all ranging from 0 to 1
  α = smoothing parameter controlling exponential decay for the level
  β = smoothing parameter controlling exponential decay for the slope
  γ = smoothing parameter controlling exponential decay for seasonal component
  => larger values give more weight to most recent observations
- 1 forecast equation and
  3 smoothing equations: one for the level ℓ_t, one for the trend b_t and for seasonal component s_t
- with corresponding smoothing parameter
- models level + trend + seasonality of series: $Y_t = \text{level} + \text{trend} + \text{season} + \text{irregular}_t$
- small value γ in multiplicative model: seasonal component hardly changes over times

2 variations
Differ in the nature of the seasonal component
- Additive Method: for additive series
  Multiplicative method: for multiplicative series
- Example:

Example
- multiplicative model fits better
- multiplicative seasonality display larger and increasing seasonal variation
- Smoothing parameters:
  -
  - β: small value of for additive model => slope component hardly changes over time (vertical scale)

=> increasing size of seasonal component for the additive model suggests
that the addtive model is less appropriate than the multiplicative model





Holt-Winters' additive method

Intuition

- used if additive series = seasonal variations are roughly constant through
- seasonal component expressed in absolute terms in scale of observed series
- level equation of series is seasonally adjusted by subtracting seasonal component
- within each year, seasonal component adds up to approximately zero

Definition

Forecast equation	$\hat{y}_{t+h\|t} = \ell_t + h\,b_t$	==>	$\hat{y}_{t+h\|t} = \ell_t + h\,b_t + s_{t+h-m(k+1)}$
Level equation	$\ell_t = \alpha\,y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$		$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$
Trend equation	$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$		$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$
			$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$

m = frequency of the seasonality - eg. monthly data: m = 12

ℓ_t = level, b_t = trend,

s_t = seasonal component; represents seasonal influence at time t, γ = smoothing parameter for seasonality

Holt-Winters' multiplicative method

Intuition

- used if muliplicative series = seasonal variations changing proportional to level of series
- seasonal component is expressed in relative term = percentages
- seasonally adjusted by dividing through by seasonal component
- each year, seasonal component will sum up to approximately m

Definition

$$\hat{y}_{t+h|t} = (\ell_t + h\,b_t)s_{t+h-m(k+1)}$$
$$\ell_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1})$$
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$$
$$s_t = \gamma\frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m}$$

Holt-Winters' damped method

- Damping possible with both additive and multiplicative methods
- Accurate and robust forecasts for seasonal data:
  Holt-Winters method with a damped trend and multiplicative seasonality

Taxonomy of exponential smoothing methods

Combining variations of trend and seasonal components yields 9 exponential smoothing methods

Each method

Labelled by (T,S) defining type of 'Trend' and 'Seasonal' component

eg. (N,N) = Simple Exponential Smoothing

		Seasonal Component		
		N (None)	**A (Additive)**	**M (Multiplicative)**
Trend Component	**N (None)**	Simple Exponential	-	-
	A (Additive)	**Double:** Holt's linear	**Triple**: Additive Holt-Winters	**Triple:** Multiplicative Holt-Winters
	A_d **(Additive damped)**	**Double:** Additive damped trend	**Triple** Additve Holt-Winters' damped	**Triple**: Multiplicative Holt-Winters' damped

Trend	**Seasonal** N	A	M
N	$\hat{y}_{t+h\|t} = \ell_t$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1}) + (1-\gamma)s_{t-m}$
A	$\hat{y}_{t+h\|t} = \ell_t + hb_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1-\gamma)s_{t-m}$
$\mathbf{A_d}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + \phi_h b_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1})) + (1-\gamma)s_{t-m}$

Type	Parameters fit	Functions
`simple`	`level`	`ets(ts, model="ANN")` `ses(ts)`
`double`	`level, slope`	`ets(ts, model="AAN")` `holt(ts)`
`triple`	`level, slope, seasonal`	`ets(ts, model="AAA")` `hw(ts)`

Innovations state space models for exponential smoothing

From exponential smoothing methods => state space models
- Smoothing methods: algorithm for producing point forecasts only
- each exponential smoothing method has underlying state space model
- statistical model gives same point forecasts, but **also generate prediction / forecast intervals**

State space model
- state space model: is a statistical model
  = stochastic data generating process that can produce an entire forecast distribution
- state = unobserved level, trend, seasonal component
- model consists of:
  - measurement equation: describes observed data
  - state equations: describing the states change over time
    => hence referred to as state space models

Two types of Errors
- for each method there exists 2 possible innovations state space models
- Additive Errors: one corresponding to a model with additive errors
- Multiplicative Errors: sone corresponding to model with multiplicative errors

E,T,S-Model
- Described by: (E,T,S)
  - E = Error = {A,M}
  - T = Trend = {A,A,A_d}
  - S = Seasonal = {N,A,M}
- Example:
  - ETS(A,A,N)
    = model with additive errors, additive trend, no seasonality = Holt's linear method with additive errors
  - ETS(M,M_d,M)
    = model with multiplicative errors, a damped multiplicative trend and multiplicative seasonality

=> computed by R using maximum likelihood and not by minimized SSE (like in holt)

ETS(A,N,N): simple exponential smoothing with additive errors

Component Form: Forecast Equation: $\hat{y}_{t+1|t} = \ell_t$ Smoothing Equation: $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$

Re-arranging smoothing equation "error correction" form

$$\begin{aligned}\ell_t &= \alpha y_t + \ell_{t-1} - \alpha \ell_{t-1} \\ &= \ell_{t-1} + \alpha(y_t - \ell_{t-1}) \\ &= \ell_{t-1} + \alpha e_t\end{aligned}$$

where $e_t = y_t - \ell_{t-1} = y_t - \hat{y}_{t|t-1}$ is residual at time t

For additive errors assume: residuals $e_t = \varepsilon_t \sim NID(0,\sigma^2)$):

Equations
- Measurement Equation: $y_t = \ell_{t-1} + \varepsilon_t$
  => shows relationship between the observations and unobserved states (linear)
- State / Transition equation: $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$
  => shows the evolution of the state through time -
    here: α governs change - $\alpha = 0$ => no change, $\alpha = 1$ => random walk
- Together :
  - with statistical distribution of errors this forms fully specified statistical model
  - Specifically: an innovations state space model underlying simple exponential smoothing

Innovations
- "innovations", because all equations use the same random error process ε_t
- also referred to as a "single source of error" model

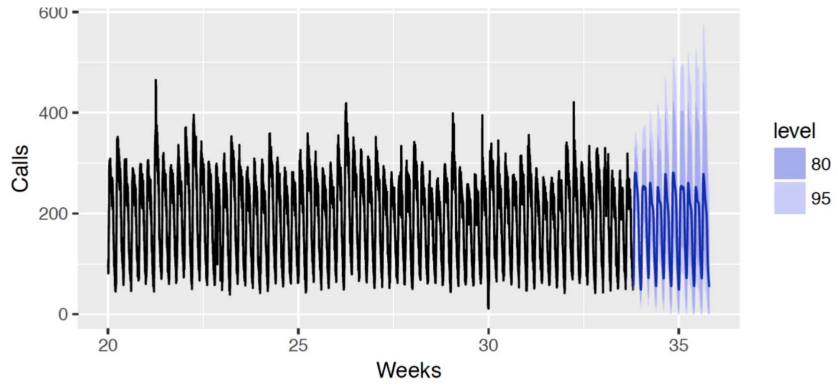
TBATS model

Idea
Use components of different models-types and in fully automated procedure to build forecast models

TBATS
T = Trigonometric terms for seasonality
B = Box-Cox transformations for heterogeneity
A = ARMA errors for short-term dynamics
T = Trend - possibly damped
S = Seasonal - including multiple and non-integer periods

Properties
- handles non-integer seasonality
- multiple seasonal periods entirely automated
- prediction intervals often too wide
- very slow on long series

Example
- good point forecast
- wide prediction intervals



VAR: Vector autoregression

VAR: Vector autoregression
- model used to capture the linear interdependencies among multiple time series
- generalize univariate AR model by allowing for more than one evolving variable
- summarized & interpreted using Granger causality

Used
- forecast collection of related variables where no explicit interpretation is required
- testing whether one variable is useful in forecasting another

Granger Causality
- a variable X
  - that evolves over time
  - Granger-causes another evolving variable Y
- if predictions of the value of Y
  - based on its own past values and
  - on the past values of X
  - are **better than predictions of Y** based **only on its own past values**
- Underlying principles of causal relationship
  1. cause happens prior to its effect.
  2. cause has unique information about the future values of its effect

Granger causality test
Used to test whether one variable is useful in forecasting another

Needed Knowledge
- VAR modeling does not require knowledge about forces influencing each variable
- only prior knowledge required is a list of variables which can be hypothesized to affect each other

Only two decisions
when using VAR to forecast:
1. How many variables
2. How many lags

Allow feedback relationships
- Variables modeled as if they all influence each other equally
  => all variables are now treated as "endogenous"
- Each variable has equation explaining its evolution based on:
  1. its own lagged values
  2. lagged values of the other model variables
  3. error term

Model
- one equation per variable
- RHS: a constant and lags of all of the variables
- 2-dimensional VAR(1)

$$y_{1,t} = c_1 + \phi_{11,1} y_{1,t-1} + \phi_{12,1} y_{2,t-1} + e_{1,t}$$
$$y_{2,t} = c_2 + \phi_{21,1} y_{1,t-1} + \phi_{22,1} y_{2,t-1} + e_{2,t}$$

Example:
- government issues cash payments in December in time for Christmas spending
- retailers reported strong sales and the economy was stimulated - consequently, incomes increased.

ARCH: Autoregressive conditional heteroskedasticity model

ARCH: Autoregressive conditional heteroskedasticity
- belongs to family of stochastic volatility models
- model that describes the variance of current error term
  as a function of the actual sizes of previous time periods' error terms
- variance is often related to the squares of the previous innovations

Used
- if error variance in a time series is assumed to follow AR model
- eg. financial time series that exhibit time-varying volatility and volatility clustering
  => periods of swings interspersed with periods of relative calm

Definition
Let ϵ_t = error terms (return residuals, with respect to a mean process), i.e. the series terms.

These ϵ_t are split into
1. stochastic piece z_t and
2. time-dependent standard deviation σ_t characterizing the typical size of the terms so that

$$\epsilon_t = \sigma_t z_t$$

where z_t is a strong white noise process
The series σ_t^2 is modeled by

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \cdots + \alpha_q \epsilon_{t-q}^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2$$

where $\alpha_0 > 0$ and $\alpha_i \geq 0,\ i > 0$

GARCH: Generalized autoregressive conditional heteroskedasticity
If error variance in a time series is assumed to follow an ARMA model

Hierarchical Models

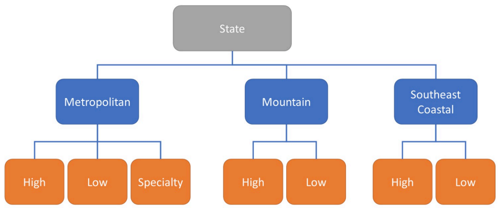
Motivation: Hierarchical time series
- time series can often be naturally disaggregated by various attributes of interest
- eg. total number of bicycles sold
  - into road, mountain and children's bikes
  - each of these can be disaggregated into finer categories -eg mountain - trecking & cross
  - categories are nested within the larger group categories

  => collection of time series follow a hierarchical aggregation structure

  Therefore: hierarchical time series

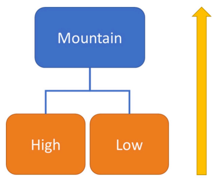
Hierarchical Models
- used if different items that need to be forecasted that can be arranged in a logical hierarchy

  => eg. by geographic divisions
- Forecasts need to be reconciled up and down the hierarchy



Three Types of Hierarchical Forecasting

1. **Bottom-up Forecasting:**
- generate forecasts for each series at the bottom-level
- sum up to produce forecasts

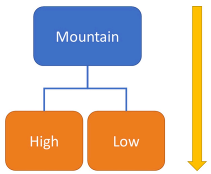


2. **Top-down Forecasting**
- works only with strictly hierarchical aggregation structures
  (not with grouped structures)
- generate forecasts for total series
- then disaggregating these down the hierarchy

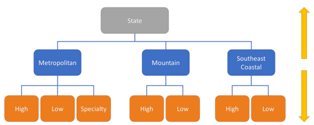
- Two Techniques
  1. Average of historical proportions
  2. Proportion of historical averages

  Set of disaggregation proportions, which dictate how the forecasts of total series are to be distributed to obtain forecasts

  => reconciled forecasts at lower level not as accurate as directly forecasting



3. **Middle-out Forecasting**
- combines bottom-up and top-down
- "middle level" is chosen and forecasts are generated for all the series at this level
- series above: forecasts are generated using the bottom-up approach
- series below: forecasts are generated using a top-down approach



Price elasticity

Price elasticity is a measure of how much demand changes - „reacts" - to changes in price

As price changes, it is expected that demand changes as well, but how much:

$$\text{Price Elasticity} = \frac{\%\text{Change in Demand}}{\%\text{Change in Price}}$$

If price of brand of bread goes up by factor 3, you re-consider buying it. => demand goes down

Elastic vs Inelastic
How much demand changes with changes in price determines if product is called elastic / inelastic

Elastic Products
- Price Elasticity > 1
- Demand changes are larger than price changes in terms of percentages
- Elastic products are ones that have % changes in demand larger than the % change in price
- Example: Brand of bread: if price goes up (while other brands stay the same), demand goes down

Inelastic products
- Price Elasticity < 1
- opposite: even drastic changes in price don't change the demand very much
- are ones that have % changes in demand smaller than the % change in price
- Example: Gasoline - even if price goes up, the demand will no change drastically

Unit elastic products
- Price Elasticity = 1
- change is the same
- are ones that have % changes in demand equal to the % change in price
- Example: if bread price would go up by 10%, the demand would drop by 10% in reponse

How to measure: Linear regression
Use natural log of price to predict natural log of sales
Why: Transforms the coefficient from units of demand to percentage-change in demand given a percentage change in price
model_M_hi <- lm(log_sales ~ log_price, data = M_hi_train)
=> value of absolute of coefficient between 0 and 1 = price elasticity => -0.7138 = inelastic

Notation

Backshift Operator

- B before value of the series x_t or error term w_t means to move that element back one time
  "Power" of B means to repeatedly apply the backshift

$$Bx_t = x_{t-1} \qquad B^2 x_t = x_{t-2} \qquad B^k x_t = x_{t-k}$$

AR polynomial

= Different way of writing AR-Model

$\Phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$ Model: $(1 - \phi_1 B)x_t = \delta + w_t$

Example

AR(1) $x_t = \delta + \phi x_{t-1} + w_t$ AR-Polynomial: $\Phi(B) = 1 - \phi_1 B$ Model: $(1 - \phi_1 B)x_t = \delta + w_t$

AR(2) $x_t = \delta + \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$ AR-Polynomial: $\Phi(B) = 1 - \phi_1 B - \phi_2 B^2$

Differencing

$x_t - x_{t-1}$ can be expressed as $(1 - B)x_t$

Alternative Notation: $\nabla = 1 - B$

$\nabla x_t = (1 - B)x_t = x_t - x_{t-1}$

=> subscript defines a difference of a lag equal to the subscript

Example: $\nabla_{12} x_t = x_t - x_{t-12}$

Notes

Financial asset returns:
measure changes in price as fraction of the initial price over given time horizon, eg. 1 day

The revenue or loss from investing depends on the amount invested and changes in prices, and high revenue relative to the size of an investment is of central interest. This is what financial asset returns measure, changes in price as a fraction of the initial price over a given time horizon, for example, one business day.

Log returns
Also called continuously compounded returns,
are log of gross returns, or equivalently, the changes (or first differences) in the logarithm of prices

Log returns, also called continuously compounded returns, are also commonly used in financial time series analysis. They are the log of gross returns, or equivalently, the changes (or first differences) in the logarithm of prices. The change in appearance between daily prices and daily returns is typically substantial, while the difference between daily returns and log returns is usually small. As you'll see later, one advantage of using log returns is that calculating multi-period returns from individual periods is greatly simplified - you just add them together!

Efficient Market Hypothesis
There is a well-known result in economics called the "Efficient Market Hypothesis" that states that asset prices reflect all available information. A consequence of this is that the daily changes in stock prices should behave like white noise (ignoring dividends, interest rates and transaction costs). The consequence for forecasters is that the best forecast of the future price is the current price.