DSC 383: Advanced Predictive Models for Complex Data

Section: Time Series Analysis > Subsection: EDA and Classical Models

INSTRUCTOR:
Catherine (Kate) Calder
Department of Statistics & Data Sciences
University of Texas at Austin

COMPUTING DEMO: Assessing Stationarity

LECTURE: Exploratory Data Analysis (EDA)

FORECASTING

- Approaches:
  1) regression (including nonparametric methods) $\rightarrow$ extrapolation
  2) machine learning/supervised learning methods $\rightarrow$ forecasting is a prediction problem
  3) time series analysis $\rightarrow$ exploiting serial dependence

  Or some combination of the three...

- How do we determine whether there is serial dependence in an observed time series?
    1) If the data are stationary, we can look at the sample autocorrelation function

    2) If the data are not stationary, we can try to coerce the data to stationarity

STRATEGIES FOR COERCING TIMES SERIES TO STATIONARITY

1) Remove a trend or cycles through regression

   GOAL: residuals are stationary

2) Differencing

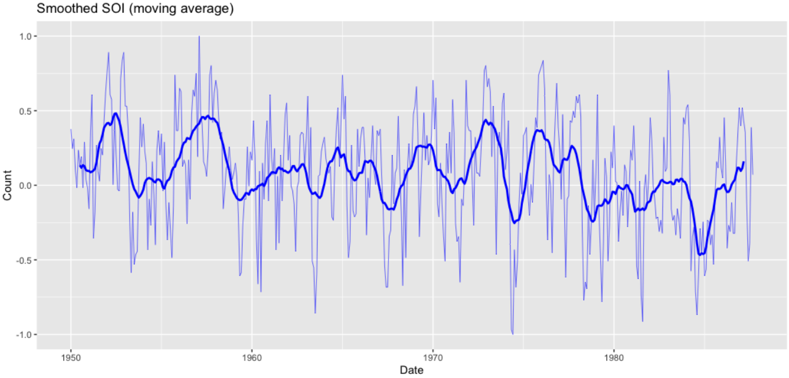
3) Transformation

SMOOTHING

- Moving average smoothing

$$m_t = \sum_{j=-k}^{k} a_j x_{t-j},$$

where $a_j = a_{-j} \geq 0$ and $\sum_{j=-k}^{k} a_j = 1$




```
# Apply kernel smoothing to the soi data in the astsa library
a <- c(.5, rep(1, 11), .5)/12
ma_smoothed_data <- data.frame(
  SOI_ma = stats::filter(as.numeric(soi),
                         sides = 2,
                         filter = a),
  Time = as.Date(time(soi)))
ma_smoothed_data <- ma_smoothed_data[!is.na(ma_smoothed_data$SOI), ]
```

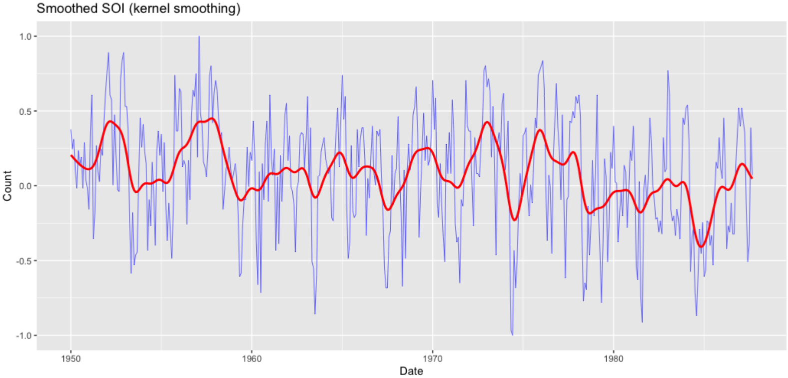
- Kernel smoothing

$$m_t = \sum_{i=1}^{n} w_i(t) x_t,$$

where

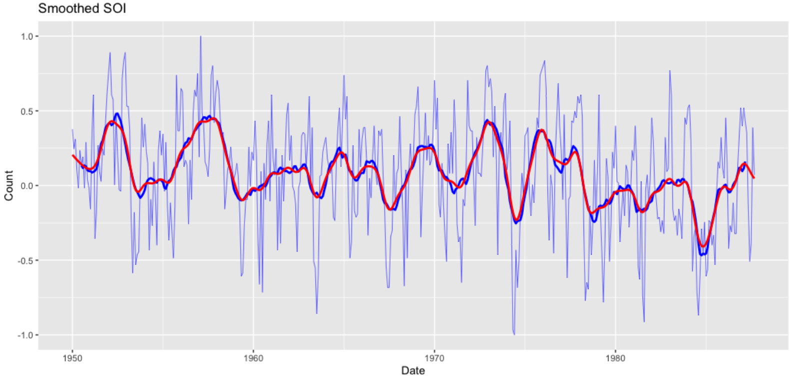
$$w_i(t) = \frac{K\left(\frac{t-t_i}{b}\right)}{\sum_{k=1}^{n} K\left(\frac{t-t_k}{b}\right)},$$

$K(\cdot)$ is a kernel function, and b is the bandwidth

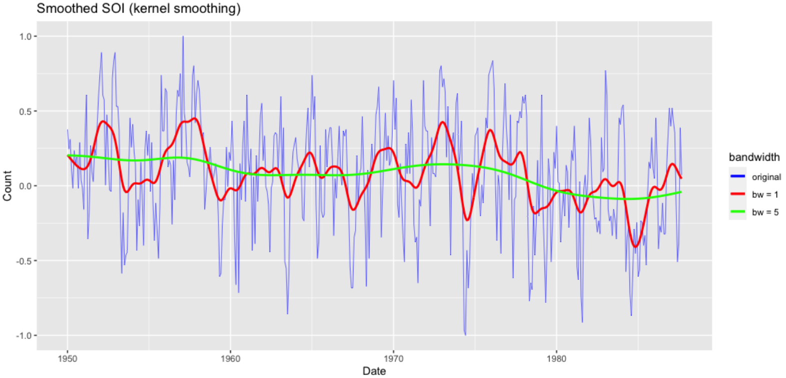



```
# Apply kernel smoothing to the soi data in the astsa library
k_smoothed_data <- data.frame(
  SOI = ksmooth(time(soi),
                as.numeric(soi),
                kernel = "normal",
                bandwidth = 1)$y,
  Time = as.Date(time(soi)))
```

- Moving average vs. kernel smoothing



- **Question:** What's the correct amount of smoothing?



LECTURE: Classical Time-Series Models I

(S)ARIMA MODELS

- Class of models introduced by Box and Jenkins (1970) and are still widely used

  S -

  A -

  R -

  I -

  M -

  A -

- **Idea:** Very general statistical modeling framework that combines ideas discussed previously $\rightarrow$ focus is on inference on model parameters and forecasting with uncertainty

SPECIAL CASE: ARMA MODELS

- **Definition:** An autoregressive model of order p (AR(p)) can be written as

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t,$$

where x_t is stationary and w_t is white noise

$\rightarrow$ Unknown parameters:

- **Question:** Is an AR(p) a multiple linear regression model?

- Consider the zero-mean AR(1) model:

$$x_t = \phi x_{t-1} + w_t$$

Question: What are valid values of ϕ?

- **Definition:** A moving average model of order q (MA(q)) can be written as

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q w_{t-q}$$

  w_t is white noise

  $\rightarrow$ Unknown parameters:

- The moving average process is stationary for any values of parameters $\theta_1, \ldots, \theta_q$

- **Definition:** A autoregressive moving average model of order p, q (ARMA(p,q)) can be written as

$$x_t = \alpha + \sum_{i=1}^{p} \phi_p x_{t-p} + \sum_{j=1}^{q} \theta_j w_{t-j} + w_t,$$

for $\phi_p \neq 0$, $\theta_q \neq 0$, and $\sigma_w^2 > 0$ and the model is *causal* and *invertible*

- ARMA models are not unique

- Model fitting:
  1) Method-of-moments estimator (e.g. Yule-Walker estimator)
  2) Maximum Likelihood Estimation (MLE) estimator
  3) Ordinary Least Squares (OLS) estimator

  * differences only expected when the process is far from stationary or n is small

COMPUTING DEMO: Fitting ARMA Models

LECTURE: Classical Time-Series Model II

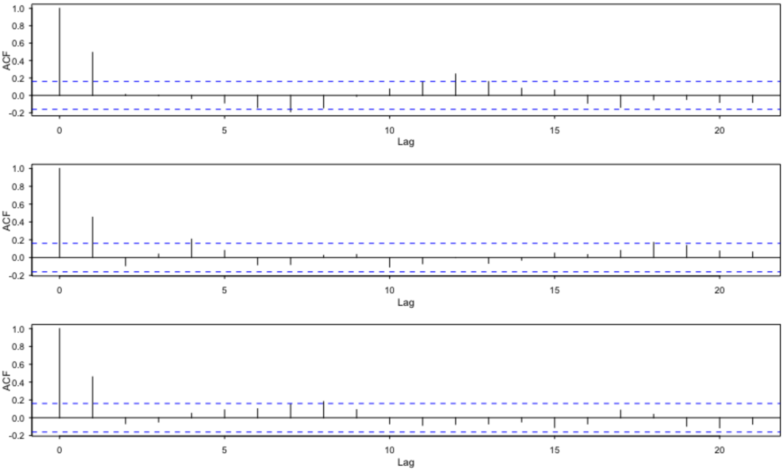
MODEL SELECTION

- How do we pick p and q?
    1) Prior to fitting models: exploratory analysis

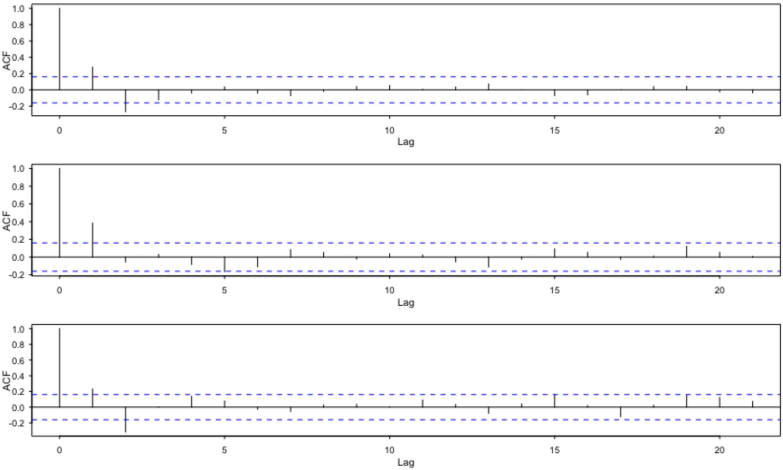
    2) Model comparison: imformation criteria, goodness-of fit measures, visual assessment of model fit

- ACF of the MA(q) process:

ACFs for three realizations of a MA(1) process



ACFs for three realizations of a MA(2) process



- Unlike the MA(q) process, the ACF for the AR(p) and AR(p, q) do not cut off at a particular lag

  $\rightarrow$ the SACF is not particularly helpful in identifying the order for an AR or ARMA process

- **Definition:** The partial autocorrelation function (PACF) of a stationary process, x_t, denoted by ϕ_{hh}, for $h = 1, 2 \ldots,$ is

$$\phi_{11} = \mathrm{Cor}[x_1, x_0] = \rho(1)$$

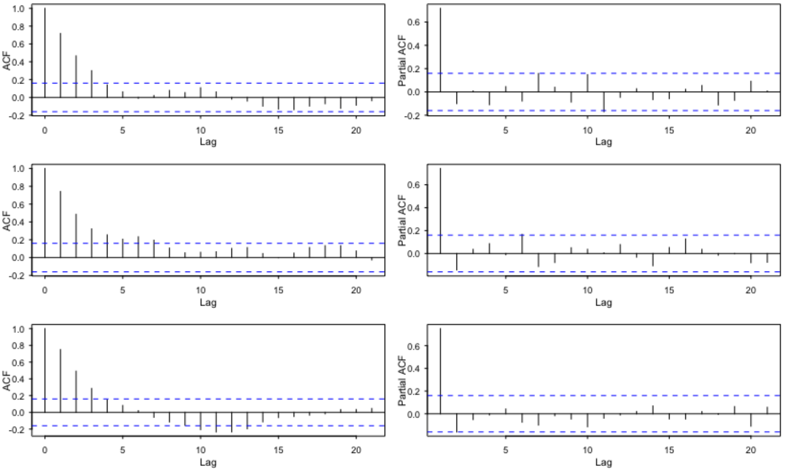
and

$$\phi_{hh} = \mathrm{Cor}[x_h - \hat{x}_h, x_0 - \hat{x}_0], \quad h \geq 2$$

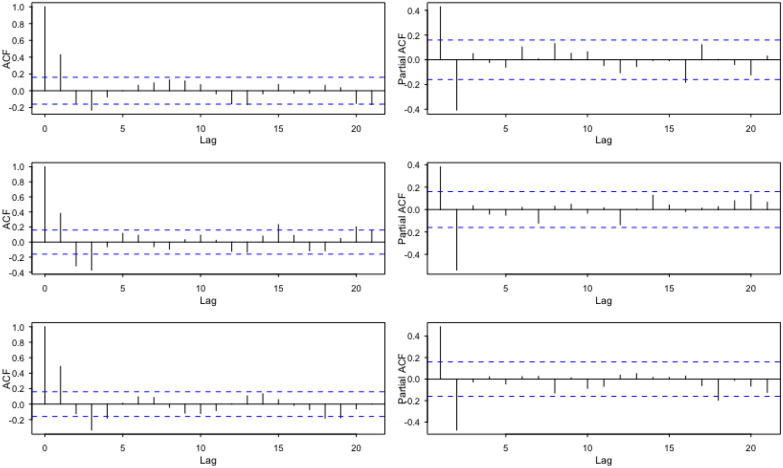
where $\hat{x}_h$ is the regression on x_h on $\{x_1, x_2, \ldots, x_{h-1}\}$ and $\hat{x}_0$ is the regression on x_0 on $\{x_1, x_2, \ldots, x_{h-1}\}$

Intuition...

ACFs and PACFs for three realizations of a AR(1) process



ACFs and PACFs for three realizations of a AR(2) process



- General behaviors of the ACF and PACF:

	AR(p)	MA(q)	ARMA(p, q)
ACF	tails off	cuts off after lag q	tails off
PACF	cuts off after lag p	tails off	tails off

- An alternative strategy:

  - Fit ARMA models for different p and q

  - Compare model fit using an information criterion: AIC (Akaike Information Criterion), AICc (corrected AIC) and BIC (Bayesian Information Criterion)

    Warning: use automated tools with caution

  * remember that p and q are not unique

SPECIAL CASE: ARIMA MODELS

- **Definition:** The backshift operator, B, is defined as

$$Bx_t = x_{t-1}$$

  and this notation is extended so that, in general,

$$B^k x_t = x_{t-k}$$

- **Definition:** The forward-shift operator, B^{-1}, is the inverse of the backshift operator, so that

$$x_t = B^{-1}Bx_t = B^{-1}x_{t-1}$$

- Note that

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

- Differences of order d are defined as

$$\nabla^d = (1 - B)^d$$

- **Definition:** An ARIMA(p, d, q) model can be written as

$$\phi(B)(1-B)^d x_t = \alpha + \theta(B) w_t,$$

  where
$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$$

  is the autoregressive operator,

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$$

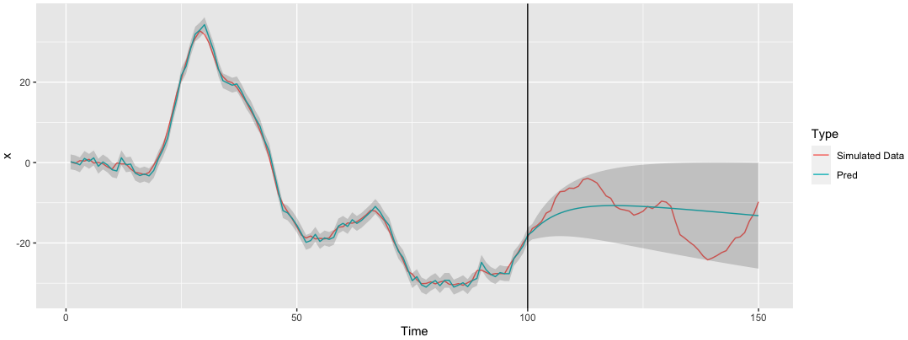
  is the moving average operator, $\alpha = \delta(1 - \phi_1 - \cdots - \phi_p)$, and $\delta = E[\nabla^d x_t]$

- Fitting an ARIMA(p, d, q) is equivalent to fitting an ARMA(p,q) model to the data that are differenced first

- **Example:** Consider the ARIMA(1,1,0) model:

$$\nabla x_t = 0.9 \nabla x_{t-1} + w_t$$

$\rightarrow$




```
# Generate a realization from a ARIMA (1,1,0), fit the model to the training
# data, and compare the forecast to the true values
x <- arima.sim(list(order = c(1, 1, 0), ar = .9), n = 150)[-1]
x_train <- window(x, start = 1, end = n_train)
fit_for <- sarima.for(x_train, n.ahead = 50, 1, 1, 0, plot = F)
```