STATS 207: Time Series Analysis
Autumn 2020

Lecture 4: Trend and Data Wrangling.

Dr. Alon Kipnis

September 23th 2020

Genera Info

- Home assignment is out. Due Monday 10/5/2020.
- Option to drop the final assessment.
- No lecture on Monday (9/28/2020).
- I would like to give the missing lecture at 10:00-11:20 on Friday 10/2/2020. **Please let us know if you cannot attend**. stats207-aut2021-staff@lists.stanford.edu

Outline

Data Wrangling

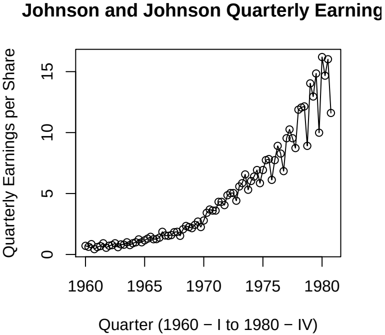
Smoothing

Motivation

Typically, data is does not follow a stationary model. It has

- Trend components
- Seasonality and periodic components

In this lecture: techniques for estimating and removing trend and periodic components.



Data Wrangling

Useful Transformations

- Detrending

$$y_t = x_t - \beta_0 - \beta_1 t$$

- Differencing

$$y_t = \nabla x_t = x_t - x_{t-1}$$

- Backshift

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

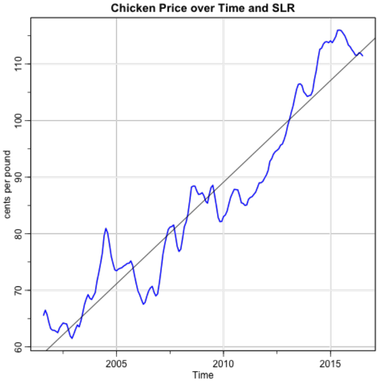
- Differencing of order d

$$\nabla^d x_t = (I - B)^d x_t$$

- Power transformations

$$y_t = \begin{cases} (x_t^\lambda - 1)/\lambda & \lambda > 0 \\ \log(x_t) & \lambda = 0. \end{cases}$$

Trend Model



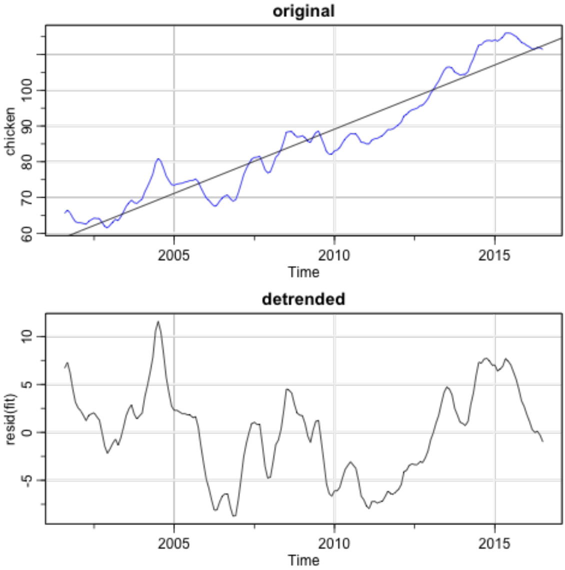
- Suppose

$$x_t = y_t + m_t$$

where (y_t) is stationary and (m_t) is a deterministic **trend**.

- **Ideology:** Remove trend, so that data exhibits steady behavior over time. Then assume stationarity for estimation and prediction.

Detrending Chicken Prices (Example 2.4)



Parametric Trend Estimation

- Assume a **parametric** model: $m_t = f(t; \beta)$. Estimate β.
- The *detrended* series is $\hat{y}_t = x_t - f(t; \hat{\beta})$.
- Examples:
    - **Polynomial** regression

$$f(t;\beta) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3$$

    (recall Example 2.4: chicken prices).

    - **Periodic** regression (period T is known)

$$f(t;\alpha) = \alpha_1 \cos(2\pi t/T) + \alpha_2 \sin(2\pi t/T)$$

    - Hybrid: Polynomial + Periodic

$$f(t;\beta,\alpha) = \beta_0 + t\beta_1 + \alpha_1 \cos(2\pi t/T) + \alpha_2 \sin(2\pi t/T).$$

- **Advantages**:
    - Gives an **accurate estimate** when model assumptions are correct.
    - Easy to **predict future observations**.
- **Disadvantages**:
    - **Selecting the correct model** might be difficult.
    - **Parametric form might be unrealistic** in practice.

Differencing

- First order **differencing**:

$$\begin{aligned}\hat{y}_t &= \nabla x_t = x_t - x_{t-1}\\ &= y_t - y_{t-1} + m_t - m_{t-1}\end{aligned}$$

  ($\nabla = 1 - B$ where B is the backshift operator: $Bx_t = x_{t-1}$).

- Definition: *Differences* of order d are

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

  (useful when m_t is approximately polynomial).

- **Advantage**:
  - No **parameters** are estimated.
  - Especially useful if data behaves as a random walk (cf. Example 2.6).
- **Disadvantage**:
  - Does not yield an **estimate** of the stationary process y_t.

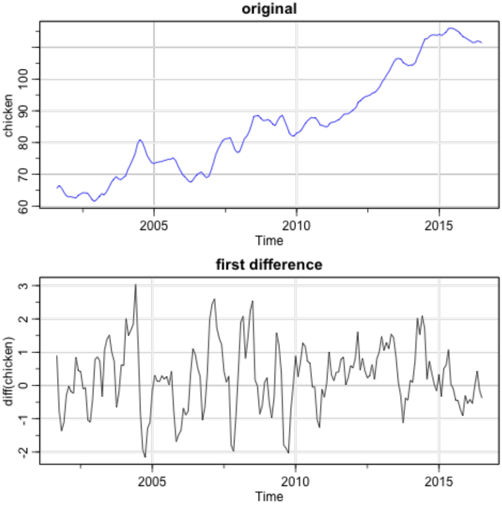
Examples

- $x_t = \beta_0 + t\beta_1 + y_t$:

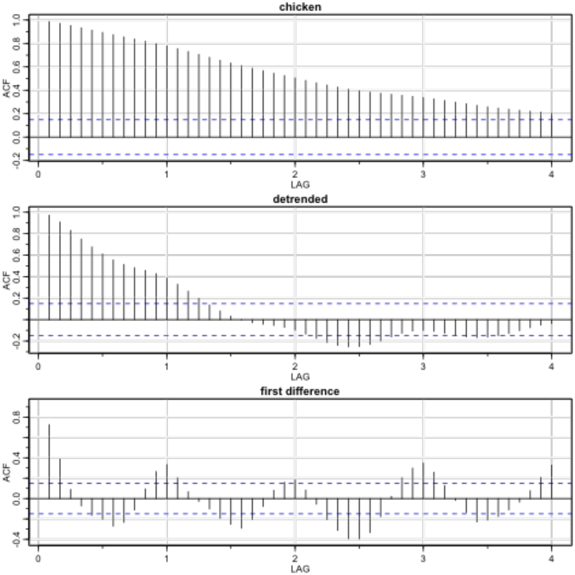
- $x_t = t^2 + y_t$:

- Prediction:

Example 2.5: Chicken Prices



Example 2.5: Detrending and Differencing ACF



Log and Power Transformations

- Log transformation

$$y_t = \log x_t.$$

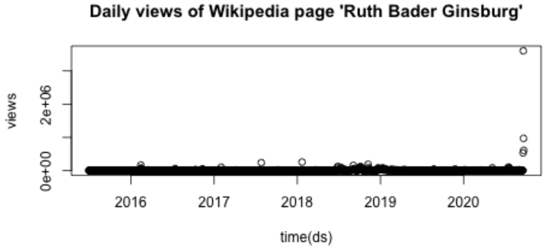
  Applies to **non-negative** data. Tends to **suppress large fluctuations** occurring over portions of the series.

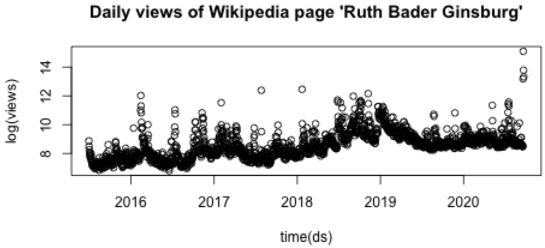
- Generalization: *Box-Cox power transformation*

$$y_t = \begin{cases} (x_t^\lambda - 1)/\lambda & \lambda > 0 \\ \log(x_t) & \lambda = 0. \end{cases}$$

- **Goals**:
  - **Equalize variability** over time.
  - Improve **approximation to normality**.
  - Improve **linearity** in predicting based on another series.

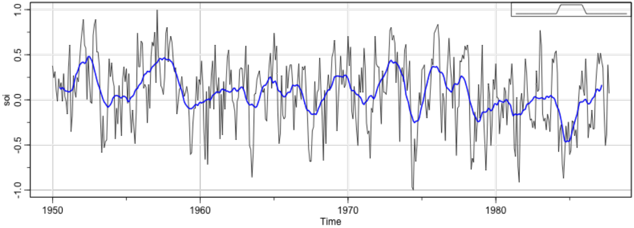
Example 2.7: Daily Views of RBG's Wikipedia Page





Smoothing

Motivation: Discovering El-Ninõ Effect in SOI Data



Smoothing

- Estimating trend m_t by a **weighted average in a neighborhood**:

$$\hat{m}_t = \frac{1}{2q+1}\sum_{j=-q}^{q} a_j x_{t-j} = \frac{1}{2q+1}\sum_{j=-q}^{q} a_j m_{t-j} + \frac{1}{2q+1}\sum_{j=-q}^{q} a_j y_{t-j}$$

  (typically $\sum_{j=-q}^{q} a_j = 1$, $a_j \geq 0$)

- Useful in **discovering** long-term trend and seasonal components.

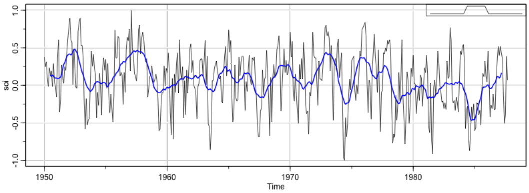
- In the following examples we smooth SOI and discover the periodic El-Ninõ effect.

Moving Average Smoothing (Example 2.11)

$$m_t = \sum_{i=-6}^{6} a_i x_{t-i}$$

$a_0 = a_{\pm 1} = \ldots = a_{\pm 5} = 1/12$, and $a_{\pm 6} = 1/24$ ("boxcar").

```
wgts = c(.5, rep(1,11), .5)/12
soif = filter(soi, sides=2, filter=wgts)
tsplot(soi)
lines(soif, lwd=2, col=4)
par(fig = c(.75, 1, .75, 1), new = TRUE) # the insert
nwgts = c(rep(0,20), wgts, rep(0,20))
plot(nwgts, type="l", ylim = c(-.02,.1), xaxt='n', yaxt='n',
ann=FALSE, main="Smoothed SOI")
```

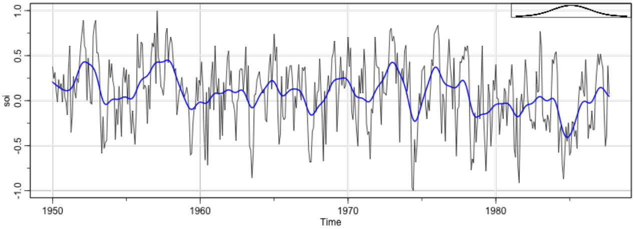


Kernel Method (Example 2.12)

$$m_t = \sum_{i=1}^{n} w_i(t) x_i, \qquad w_i(t) = K\left(\frac{t-i}{b}\right) \Big/ \sum_{j=1}^{n} K\left(\frac{t-j}{b}\right).$$

In the following $K(t) = \frac{1}{\sqrt{2\pi}} \exp\{-t^2/2\}$.

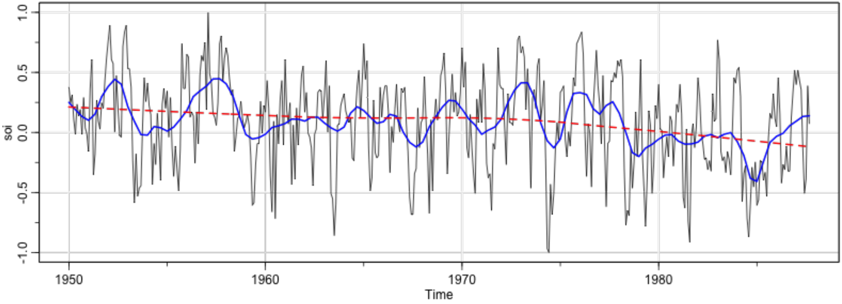
```
tsplot(soi)
lines(ksmooth(time(soi), soi, "normal", bandwidth=1), lwd=2, col=4)
par(fig = c(.75, 1, .75, 1), new = TRUE) # the insert
gauss = function(x) { 1/sqrt(2*pi) * exp(-(x^2)/2) }
x = seq(from = -3, to = 3, by = 0.001)
plot(x, gauss(x), type ="l", ylim=c(-.02,.45), xaxt='n', yaxt='n', ann=FALSE
```



Lowess (Example 2.13)

: Estimate $\hat{x}_t$ based on its k-nearest neighbors $\{x_{t-k/2}, \ldots, x_t, x_{t+k/2}\}$. Set $m_t = \hat{x}_t$.

```
tsplot(soi)
lines(lowess(soi, f=.05), lwd=2, col=4) # El Nino cycle
lines(lowess(soi), lty=2, lwd=2, col=2) # trend (with default span
#                                              = 2/3 of data)
```



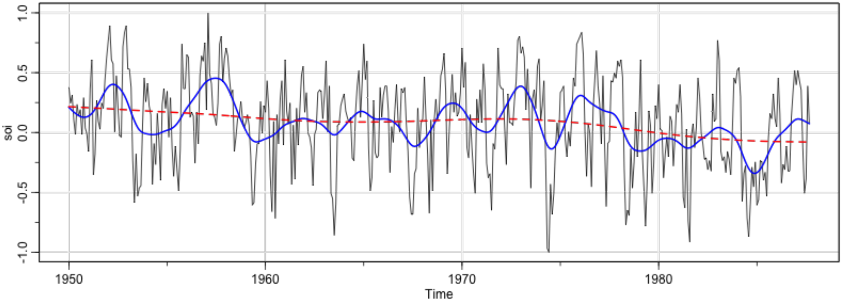
- *Smoothing Splines*: Find $f(t)$ that minimizes

$$\sum_{t=1}^{n}(x_t - f(t))^2 + \lambda \int (f'')^2 dt,$$

  over the class of **twice differentiable functions**.
- The minimizer $\hat{f}(t)$ is a **piecewise cubic polynomial** with knots at $t = 1, \ldots, n$.
- λ trades-off between fitting the data and roughness of the function estimate
  - $\lambda = 0$ leads to $m_t = x_t$ (no smoothing).
  - $\lambda \to \infty$ leads to $m_t = c + vt$.
- Also useful for interpolation when **time grid is non-uniform** or when there are **missing values**.

Smoothing Splines (Example 2.14)

```r
plot(soi)
lines(smooth.spline(time(soi), soi, spar=.5), lwd=2, col=4)
lines(smooth.spline(time(soi), soi, spar= 1), lty=2, lwd=2, col=2)
```



Isotonic Trend Estimation

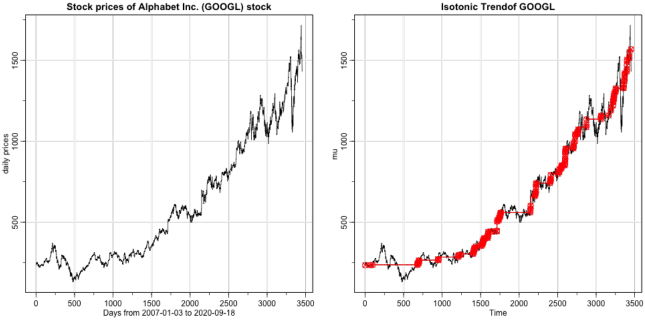
Estimate *monotone* trend where $m_1 \leq \ldots \leq m_n$ by solving the **convex optimization problem**

$$\min_{a_1,\ldots,a_n} \sum_{t=1}^{n} (x_t - a_t)^2 \quad \text{subject to} \quad a_1 \leq \ldots \leq a_n.$$

- **Advantages:**
  1. Non parametric.
  2. No smoothing parameters.
  3. Gives estimate also for **end-points**
- **Disadvantages:**
  - **Monotonicity assumption** might be too strong.
  - No straightforward approach for **predicting future values**.
  - Not helpful if **data is already monotone**.

Example: Stock Price of Alphabet Inc.

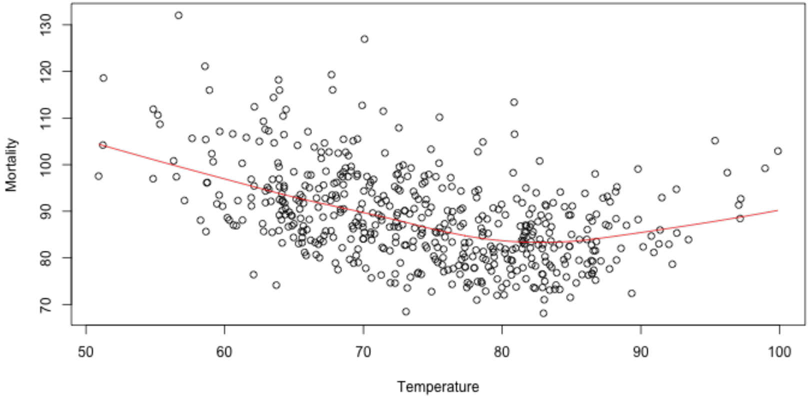
```
getSymbols("GOOGL") #GOOGL is the ticker or stock symbol for Alphabet Inc.
tsplot(as.numeric(GOOGL$GOOGL.Close), ylab = "daily prices",
        xlab = "Days from 2007-01-03 to 2020-09-18", type = "l",
        main = "Stock prices of Alphabet Inc. (GOOGL) stock")
mu <- isoreg(GOOGL$GOOGL.Close) #Isotronic trend estimation
par(par(mfrow=c(2,1)))
tsplot(mu, main = "Isotonic Trend")
```



Smoothing one series as a function of another

Recall Example 2.2: Temperature, Mortality and Pollution.

```
plot(tempr, cmort, xlab="Temperature", ylab="Mortality")
lines(lowess(tempr, cmort), col = 2)
```



Recap

- First:
  1. Transform data for **constant variance**.
  2. **Remove trend** components.
  3. **Remove seasonality/periodic** components.
- If the residuals exhibit steady behavior over time, assume stationarity.

In the next unit we will work with models for stationary data.

Note: in many cases (when the SNR is low), the fitted deterministic model resulting from steps 1-3 is already useful.