STATS 207: Time Series Analysis
Autumn 2020

Lecture 9: Regression with Autocorrelated Errors and Lagged Regression

Dr. Alon Kipnis

October 9th 2020

Genera Info

- HW2 is out. Due on Monday October 19.
- Solutions to HW1 are out.
- Grades to HW1 are out. Regreding requests through Gradescope.
- **Midquarter Feedback Survey** `https://canvas.stanford.edu/courses/123058/quizzes/84145` (Canvas). **Visit now!**.
- Wednesday: Prof. David Donoho will talk about ARCH, GARCH, and stochastic volatility models.

Outline

Regression with Correlated Errors

Lagged Regression

Overview

So far:

- Regression with **uncorrelated** errors:

$$y_t = \sum_{j=0}^{r} \beta_j z_{tj} + w_t, \qquad w_t \text{ is white noise.}$$

Today:

1. Regression with **correlated** errors:

$$y_t = \sum_{j=0}^{r} \beta_j z_{tj} + x_t,$$

   x_t is independent of (z_t); not assumed white noise.

2. Lagged regression

$$y_t = \sum_{j=0}^{p} \alpha_j x_{t-j} + \eta_t,$$

   η_t is independent of (x_t); not assumed white noise.

Slogan: Use ARMA-style modelling for **residuals** of standard regression.

Regression with Correlated Errors

Regression with Correlated Errors (Section 3.8)

- General form:

$$y_t = \mathrm{z}_t'\boldsymbol{\beta} + x_t \qquad t = 1, \ldots, n,$$

  where $\boldsymbol{\beta} \equiv (\beta_1, \ldots, \beta_r)' \in \mathbb{R}^r$.

- Matrix form:

$$\boldsymbol{y} = Z\boldsymbol{\beta} + \boldsymbol{x},$$

  where:

  - $\boldsymbol{y} = (y_1, \ldots, y_n)' \in \mathbb{R}^n$
  - $\boldsymbol{x} = (x_1, \ldots, x_n)' \in \mathbb{R}^n$
  - $Z = [\mathrm{z}_1|\mathrm{z}_2|\cdots|\mathrm{z}_n]' \in \mathbb{R}^{r \times n}$.

- Also: $\mathrm{Cov}(x_t, x_s) = \gamma(t, s)$. Set

$$\Gamma \equiv \{\gamma(t, s)\}_{1 \le t, s \le n}$$

  (for future use).

Exact LS Solution (Γ is known)

- Use transformation $\boldsymbol{x} \to \tilde{\boldsymbol{x}} = \Gamma^{-1/2}\boldsymbol{x}$:

$$\Gamma^{-1/2}\boldsymbol{y} = \Gamma^{-1/2}Z\boldsymbol{\beta} + \Gamma^{-1/2}\boldsymbol{x}$$
$$\tilde{\boldsymbol{y}} = \tilde{Z}\boldsymbol{\beta} + \tilde{\boldsymbol{x}}.$$

- The covariance matrix of $\tilde{x}$:

$$\begin{aligned}
\mathrm{Cov}(\tilde{\boldsymbol{x}}) &= \mathbb{E}\left[(\tilde{\boldsymbol{x}} - \mu_{\tilde{x}})'(\tilde{\boldsymbol{x}} - \mu_{\tilde{x}})\right] \\
&= \mathbb{E}\left[\Gamma^{-1/2}(\boldsymbol{x} - \mu_x)'(\boldsymbol{x} - \mu_x)\Gamma^{-1/2}\right] \\
&= \Gamma^{-1/2}\mathbb{E}\left[(\boldsymbol{x} - \mu_x)'(\boldsymbol{x} - \mu_x)\right]\Gamma^{-1/2} = \mathrm{I}
\end{aligned}$$

- From OLS:

$$\hat{\boldsymbol{\beta}} = \left(\tilde{Z}'\tilde{Z}\right)^{-1}\tilde{Z}'\tilde{y} = \left(Z'\Gamma^{-1}Z\right)^{-1}Z'\Gamma^{-1}\boldsymbol{y}$$

  and

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \left(Z'\Gamma^{-1}Z\right)^{-1}$$

- Note: $\hat{\boldsymbol{\beta}}$ is MLE if $x_t \sim \mathcal{N}(0, \Gamma)$.

How to get Γ?

- **Issue**: Sample vectors $\boldsymbol{y}$ and Z; at best **one realization** of $\boldsymbol{x}$.
- **Idea 1**: Assume stationarity of x_t. Implies a special structure (Toeplitz) on Γ.
- **Idea 1.1**: Assume ARMA(p, q) model for x_t.

- Interpretation of transform $\boldsymbol{x} \to \tilde{\boldsymbol{x}}$: **whitening**

$$\mathrm{Cov}(\boldsymbol{x}) = \Gamma, \qquad \mathrm{Cov}(\tilde{\boldsymbol{x}}) = \mathrm{I}.$$

- Example: If x_t is AR(p), the **whitening transformation** is

$$\phi(B)x_t = w_t.$$

- Example: If x_t is (invertible) ARMA(p, q), the **whitening transformation** is

$$\pi(B)x_t = w_t.$$

AR(p) **Whitening I**

- Use

$$\phi(B)x_t = w_t, \qquad \phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p,$$

in the regression equation to obtain

$$\begin{aligned} \phi(B)y_t &= \sum_{j=1}^{r} \beta_j \phi(B) \mathrm{z}'_{tj} + \phi(B)x_t \\ &= \sum_{j=1}^{r} \beta_j \tilde{\mathrm{z}}'_{tj} + w_t. \end{aligned}$$

- Find $\boldsymbol{\beta}$ and ϕ that minimize

$$S(\phi, \boldsymbol{\beta}) = \sum_{t=1}^{n} w_t^2 = \sum_{t=1}^{n} \left[\phi(B)y_t - \sum_{j=1}^{r} \beta_j \phi(B) \mathrm{z}_{tj} \right]^2$$

(non-linear; use numerical methods; see [Shumway & Stoffer]).

- For ARMA(p, q) error modelling $\phi(B)x_t = \theta(B)w_t$, use $[\theta(B)]^{-1}\phi(B)x_t = w_t$.

AR(p) **Whitening II**

Explicit whitening transformation for AR(p) errors:

- Define $A = \{A_{t,s}\} \in \mathbb{R}^{n \times n}$

$$A_{t,t} = 1, \qquad A_{t,t-h} = -\phi_h, \qquad 1 \leq h \leq p.$$

- Claim:

$$\mathrm{Cov}(\sigma_w^{-1} A \boldsymbol{x}) \approx \mathsf{I}$$

"except for edge effects" ($\sigma_w^{-1} A$ is whitening)

- Namely,

$$\tilde{\boldsymbol{y}} \approx \sigma_w^{-1} A \boldsymbol{y}, \qquad \tilde{Z} \approx \sigma_w^{-1} A Z, \qquad \tilde{\boldsymbol{x}} \approx \sigma_w^{-1} A \boldsymbol{x}.$$

If we have ϕ, we can solve the OLS exactly. Otherwise, we estimate it along with β coefficients.

General ARMA Whitening Algorithm

STEP I **Fit** $\hat{\beta}$ by **OLS** of y_t on $z_{t1}, \ldots, z_{tr}$ (assuming uncorrelated errors). **Get residuals**

$$\hat{x}_t = y_t - \sum_{j=1}^{r} \hat{\beta}_j z_{tj}.$$

STEP II **Identify** an ARMA model for the **residuals**:

$$\phi(B)\hat{x}_t = \theta(B)w_t.$$

STEP III **Run** weighted least squares (or MLE) on the **specified model** (with autocorrelated errors):

$$\tilde{\boldsymbol{y}} = \tilde{Z}\beta + \tilde{\boldsymbol{x}}$$

STEP IV **Inspect** the residuals $\hat{w}_t$ for **whiteness** and adjust the model if necessary.

Example 3.44: Mortality, Temperature and Pollution I

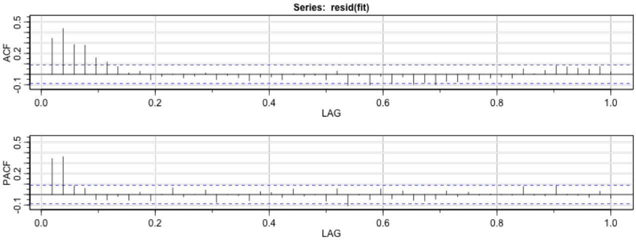
$$M_t = \beta_1 + \beta_2 t + \beta_3 T_t + \beta_4 T_t^2 + \beta_5 P_t + x_t.$$

STEP I: OLS while assuming that x_t is white noise (as in Example 2.2):

```
trend   = time(cmort)
temp    = tempr - mean(tempr)
temp2   = temp^2
fit <- lm(cmort~trend + temp + temp2 + part, na.action=NULL)
```

- ACF and PACF of residuals

```
acf2(resid(fit), 52) # implies AR2
```



Example 3.44: Mortality, Temperature and Pollution II

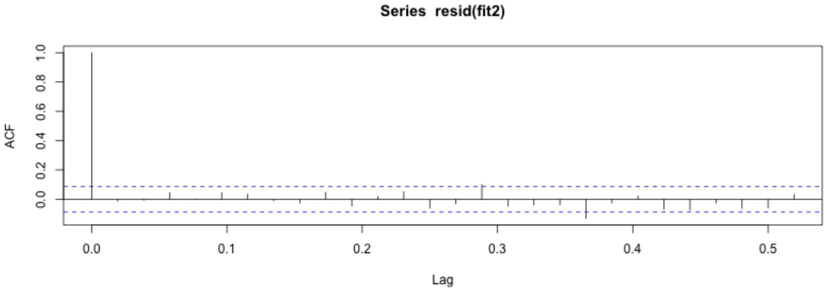
STEP II: We identify an AR(2) model for the residuals.

STEP III: Fit model with AR(2) modelling of residuals:

```
fit.w <- sarima(cmort, 2,0,0, xreg=cbind(trend,temp,temp2,part) )
acf(resid(fit.w$fit))
```

```
$ttable
           Estimate      SE t.value p.value
ar1          0.3848  0.0436  8.8329  0.0000
ar2          0.4326  0.0400 10.8062  0.0000
```

STEP IV: Inspect residuals

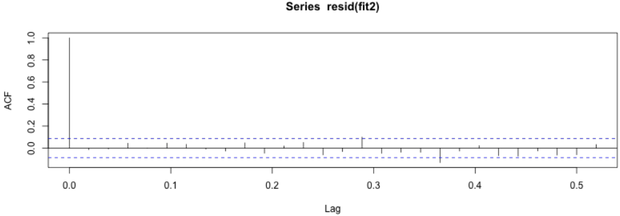


Prewhitened Mortality Modelling I

$$\tilde{M}_t = \beta_1 + \beta_2 t + \beta_3 \tilde{T}_t + \beta_4 \tilde{T}_t^2 + \beta_5 \tilde{P}_t + \tilde{x}_t$$

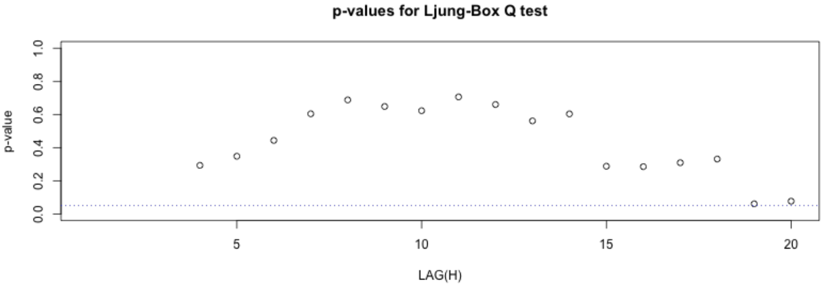
(here $\tilde{u}_t \equiv \pi(B)u_t$ for any given series (u_t))

```
my.whiten = function(x,filt=c(1,-.3848,-.4326),drop=2){
y=filter(x,filt,method="conv",sides=1);
y[(drop+1):length(y)]
}
cmort.w = my.whiten(cmort)
trend.w = my.whiten(trend)
temp.w = my.whiten(temp)
temp2.w = my.whiten(temp2)
part.w = my.whiten(part)
fit.w = lm(cmort.w ~trend.w + temp.w + temp2.w + part.w, na.action=NULL)
acf(resid(fit.w))
```



Prewhitened Mortality Modelling II

```
H.max = 20
QQ = rep(0.5, H.max)
for (h in 4:H.max) {
QQ[h] <- Box.test(resid(fit.w), lag = h, fitdf = 2)$p.value
    }
plot(QQ, main = 'p-values for Ljung-Box Q test',
            xlab='LAG(H)', ylim=c(0,1), ylab='p-value')
abline(a = 0.05, b= 0, col=4, lty = 3)
```



Lagged Regression

Lagged Regression (Section 5.5)

- Basic time series regression model

$$y_t = \sum_{j=0}^{p} \alpha_j x_{t-j} + \eta_t$$

where

- (η_t) and (x_t) are independent.
- (x_t) is available for predicting (y_t)
- "**Slogan**: y_t can be predicted from **current and past** values of (x_t)"

ARIMA-Inspired Lagged Regression I

- Write
$$y_t = \alpha(B)x_t + \eta_t,$$
- Assume:
  - (x_t) and (η_t) are **ARMA**:
  $$\phi(B)x_t = \theta(B)w_t, \qquad \phi_\eta(B)\eta_t = \theta_\eta(B)z_t,$$
  (w_t) and (z_t) are independent white noise processes.
  - $\alpha(B)$ has an **ARMA-style factorization** (Box-Jenkins formulation):
  $$\alpha(B) = \sum_{j=0}^{\infty} \alpha_j B^j, \qquad \alpha(B) = [\omega(B)]^{-1}\,\delta(B)B^d,$$
  $$\omega(B) = 1 - \omega_1 B - \ldots - \omega_r B^r, \qquad \delta(B) = \delta_0 + \delta_1 B + \ldots + \delta_s B^s,$$
  - $\alpha(B)$ is denoted the **transfer function** of the system with input x_t and output y_t.
- We want to determine $\alpha_0, \alpha_1, \ldots$.
  The challenge: "Determining a parsimonious model involving a simple form for $\alpha(B)$"

ARIMA-Inspired Lagged Regression II

Prescription for lagged regression (Part I):

STEP 1: **Fit** an ARMA model $(\hat{\phi}, \hat{\theta}, \hat{\sigma}_w^2)$ to (x_t).

STEP 2: **Apply** model to both y_t and x_t, getting $\tilde{y}_t$ and $\tilde{x}_t$:

$$\hat{\phi}(B)x_t \equiv \hat{\theta}(B)\tilde{x}_t, \qquad \hat{\phi}(B)y_t \equiv \hat{\theta}(B)\tilde{y}_t.$$

STEP 3: **Cross-correlate $\tilde{\boldsymbol{y}}$** with $\tilde{\boldsymbol{x}}$:

$$\hat{\alpha}_h \equiv \gamma_{\tilde{\mathbf{y}}\tilde{\mathbf{x}}}(h) = \mathrm{Cov}\left(\tilde{y}_{t+h}, \tilde{x}_t\right)/\hat{\sigma}_w^2, \quad h = 1, 2, \ldots.$$

Terminology:

- $(\tilde{x}_t)$ is the **prewhitened** version of (x_t)
- $(\tilde{y}_t)$ is the **filtered** version of (y_t).

Rationale for Prescription

- Suppose

$$\phi(B)x_t = \theta(B)w_t.$$

- From

$$y_t = \alpha(B)x_t + \eta_t,$$

we have

$$\begin{aligned}\tilde{y}_t \equiv [\theta(B)]^{-1}\phi(B)y_t &= \alpha(B)[\theta(B)]^{-1}\phi(B)x_t + [\theta(B)]^{-1}\phi(B)\eta_t \\ &= \alpha(B)w_t + \tilde{\eta}_t,\end{aligned}$$

where $\tilde{\eta} = [\theta(B)]^{-1}\,\phi(B)\eta_t$ is independent of (w_t).
Meanwhile

$$\gamma_{\tilde{y}\tilde{x}}(h) = \mathrm{Cov}(\tilde{y}_{t+h}, \tilde{x}_t) = \mathrm{Cov}(\tilde{y}_{t+h}, w_t) = \sigma_w^2\alpha_h, \quad h = 1, 2, \ldots.$$

ARMA-style $\alpha(B)$

- From $\alpha(B) = [\omega(B)]^{-1}\,\delta(B)B^d$, write

$$y_t = \alpha(B)x_t + \eta_t$$

or

$$\omega(B)y_t = \delta(B)B^d x_t + \omega(B)\eta_t$$

or

$$y_t = \sum_{k=1}^{r} \omega_k y_{t-k} + \sum_{k=0}^{s} \delta_k x_{t-d-k} + u_t.$$

- Suggest regression

$$\boldsymbol{y} = Z'\boldsymbol{\beta} + u$$

where

  - $\boldsymbol{y} = (y_1, \ldots, y_n)$
  - $\mathrm{z}_t = (y_{t-1}, \ldots, y_{t-r}, x_{t-d}, x_{t-d-1}, \ldots, x_{t-d-s})$
  - $\boldsymbol{u} = (u_1, \ldots, u_n)$ is stationary (possibly correlated).

ARIMA-Inspired Lagged Regression III

Prescription for lagged regression:

STEP 1: **Fit** an ARMA model $(\hat{\phi}, \hat{\theta}, \hat{\sigma}_w^2)$ to (x_t).

STEP 2: **Apply** model to both y_t and x_t, getting $\tilde{y}_t$ and $\tilde{x}_t$:

$$\hat{\phi}(B)x_t \equiv \hat{\theta}(B)\tilde{x}_t, \qquad \hat{\phi}(B)y_t \equiv \hat{\theta}(B)\tilde{y}_t.$$

STEP 3: **Cross-correlate $\tilde{\boldsymbol{y}}$ with $\tilde{\boldsymbol{x}}$**:

$$\hat{\alpha}_h \equiv \mathrm{Cov}\left(\tilde{y}_{t+h}, \tilde{x}_t\right)/\hat{\sigma}_w^2, \quad h = 1, 2, \ldots.$$

STEP 4: Suppose $\alpha(B) = [\omega(B)]^{-1}\,\delta(B)B^d$. **Fit** coefficients $\hat{\omega}$, $\hat{\delta}$ by OLS according to

$$y_t = \sum_{k=1}^{r} \omega_k y_{t-k} + \sum_{k=0}^{s} \delta_k x_{t-k-d} + u_t.$$

STEP 5: **Apply** fitted MA transformation to u_t

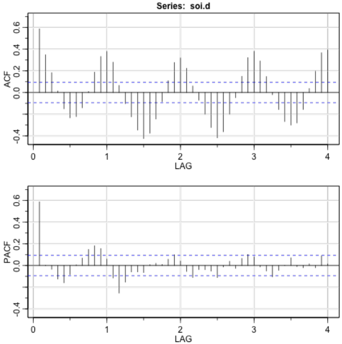
$$[\hat{\omega}(B)]^{-1}\hat{u}_t = \hat{\eta}_t.$$

and **fit** an ARMA model for $\hat{\eta}_t$.

Example 5.8: Relating Prewhitened SOI to the Filtered Recruitment Series (Steps 1-3)

Step 1: Fit ARMA model to $x_t = SOI_t$

```
soi.d = resid(lm(soi~time(soi), na.action=NULL)) # detrended SOI
acf2(soi.d)
```




```
(fit = arima(soi.d, xreg=time(soi), order=c(1, 0, 0))) # fit AR(1)
ar1 = as.numeric(fit$coef[1]) # = 0.5875387
```

Fitted model: $(1 - 0.588B)\hat{x}_t = w_t$, $\hat{\sigma}_w^2 = 0.092$.

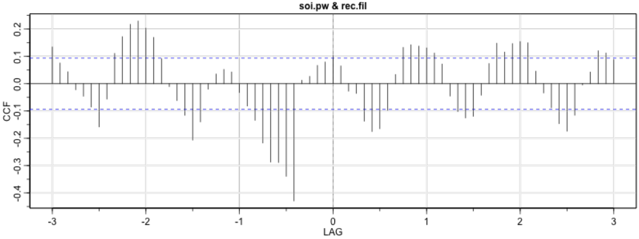
Example 5.8: Relating Prewhitened SOI to the Filtered Recruitment Series (Steps 1-3) II

Step 2: Apply fitted model to $x_t = SOI_t$ and $y_t = rec_t$:

```
soi.pw  = resid(fit)
rec.fil = filter(rec, filter=c(1, -ar1), sides=1)
```

Step 3 Cross-correlate $\tilde{y}$ with $\tilde{x}$:

```
ccf2(soi.pw, rec.fil, na.action=na.omit)
```



Example 5.9: Predict Recruitment from SOI (Steps 4-5) I

Step 4: Suggest $d = 5$, with $\omega(B)$ and $\delta(B)$ of order 1:

$$rec_t = \alpha_0 + \omega_1 rec_{t-1} + \delta_0 SOI_{t-5} + u_t.$$

Fit:

```
rec.d = resid(lm(rec~time(rec), na.action=NULL))
soi.d = resid(lm(soi~time(soi), na.action=NULL))
fish = ts.intersect(rec.d, rec.d1=lag(rec.d,-1), soi.d5=lag(soi,-5),
dframe=TRUE)
summary(fish.fit <- lm(rec.d~0+rec.d1+soi.d5, data=fish))
```

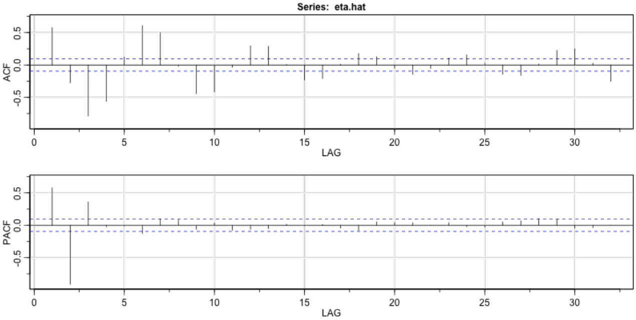
```
Call: lm(formula = rec.d ~ 0 + rec.d1 + soi.d5, data = fish)
Residuals:
    Min       1Q   Median       3Q      Max
-49.441   -2.492    1.806    6.052   29.932
Coefficients:
       Estimate Std. Error t value Pr(>|t|)
rec.d1   0.8531     0.0144   59.23   <2e-16 ***
soi.d5 -18.8627     1.0038  -18.79   <2e-16 ***
---

Residual standard error: 8.02 on 446 degrees of freedom
Multiple R-squared:  0.9145,    Adjusted R-squared:  0.9141
F-statistic:  2386 on 2 and 446 DF,  p-value: < 2.2e-16
```

Example 5.9: Predict Recruitment from SOI (Steps 4-5) II

Step 5: Fit ARMA to $\hat{\eta}_t = [\hat{\omega}(B)]^{-1}\hat{u}_t$:

```
om1 = as.numeric(fish.fit$coef[1])
eta.hat = filter(resid(fish.fit), filter=c(1,-om1),
                                         method="recur", sides=1)
acf2(eta.hat)    # looks like AR(3)
```



Step 5: Fit AR(3) to $\hat{\eta}_t = [\hat{\omega}(B)]^{-1}\hat{u}_t$:

```
eta.fit <- arima(eta.hat, order=c(3,0,0))
```

```
Call:
arima(x = eta.hat, order = c(3, 0, 0))

Coefficients:
         ar1      ar2     ar3  intercept
      1.4261  -1.3017  0.3557     1.6015
s.e.  0.0441   0.0517  0.0441     0.6483

sigma^2 estimated as 50.85:  log likelihood = -1517.91,  aic = 3045.82
```

Final model:

$$(1 - 0.588B)SOI_t = w_t$$

$$rec_t = 0.583 \cdot rec_{t-1} - 18.86 SOI_{t-5} + u_t$$

$$u_t = (1 - 0.853B)\eta_t$$

$$(1 - 1.426B + 1.3B^2 - 0.356B^3)\eta_t = z_t$$

where (w_t) is white noise $\sigma_w^2 = 0.092$, and (z_t) is white noise $\sigma_z^2 = 50.85$.