STAT 436 / 536 - Lecture 9

October 3, 2018

Linear Models and Regression

- With time series models, trends and seasonal patterns can be

- *Deterministic*

- *Stochastic*

- **Q:** how would forecasting differ based on stochastic and deterministic patterns?

- As we have seen time series analysis requires careful consideration due to the serial correlation present in the random error. The same idea holds for time series regression.

Linear Models

- A regression model for a time series data $\{x_t, \ldots, x_1\}$ is a linear model if it can be written as

  where $u_{i,t}$ is the value of the ith explanatory variable (or predictor or covariate) at time t, z_t is the error term at time t, and $\alpha_0, \ldots, \alpha_m$ are the model parameters (or regression covariates).

- **Q:** are the following equations linear models?

$$x_t = \alpha_0 + \alpha_1 \log(u_{1,t}) + \alpha_2 u_{2,t} + z_t$$

$$x_t = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + z_t$$

$$x_t = \alpha_0 + \alpha_1^2 u_{1,t} + \alpha_2^4 u_{2,t} + z_t$$

- The

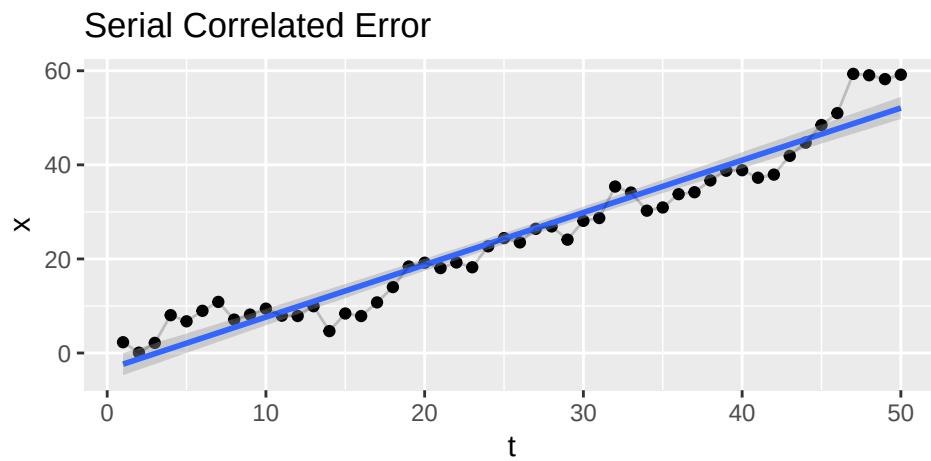
- The typical approach for fitting linear models is to use least squares.

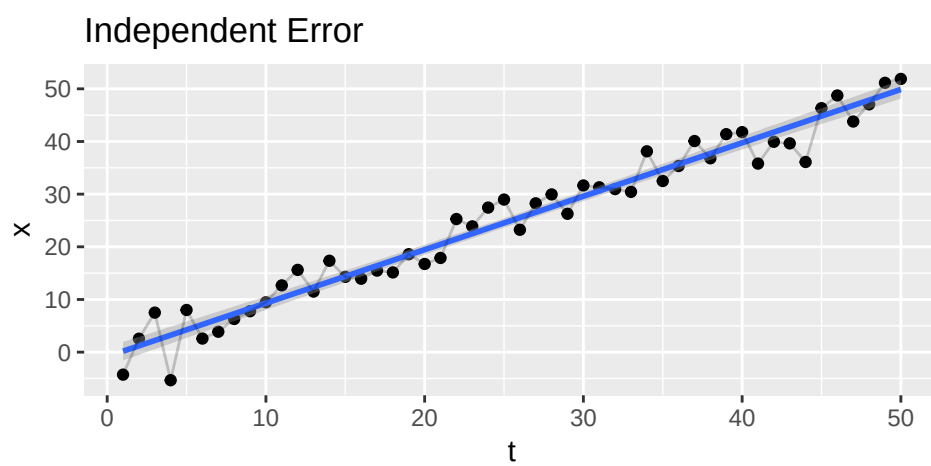
- Consider the following linear model

$$x_t = \alpha_0 + \alpha_1 t + z_t$$

  where z_t has mean zero and follows a AR(1). Code for simulating from this model and one with indpendent errors is included next.

```
set.seed(09242018)
time.pts <- 50
z <- rep(0,time.pts)
t <- 1:time.pts
alpha.0 <- 0; alpha.1 <- 1; sigma.z <- 3; ar.coef <- .95
z[1] <- rnorm(1,0,sd=sigma.z)
for (t.val in 2:time.pts){
  z[t.val] <- ar.coef * z[t.val-1] + rnorm(1,0,sd=sigma.z)
}
x.corr <- alpha.0 + alpha.1 * t + z
x.ind <- alpha.0 + alpha.1 * t + rnorm(time.pts, 0 , sd = sigma.z)
df.corr <- data.frame(x=x.corr, t = t)
df.ind <- data.frame(x=x.ind, t = t)
```





- The `lm()` function in R can be used to fit regression models.

```
lm.ind <- lm(x.ind ~ t)
summary(lm.ind)$coefficients
```

```
##               Estimate Std. Error    t value     Pr(>|t|)
## (Intercept) -0.8285193 0.89835412 -0.9222636 3.610037e-01
## t            1.0142882 0.03066037 33.0814060 1.072357e-34
```

- For the independent errors,

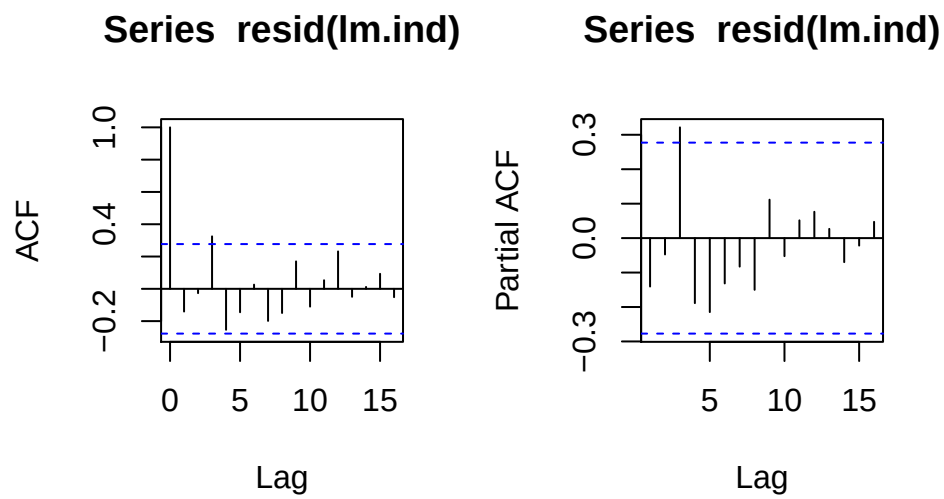
```
lm.corr <- lm(x.corr ~ t)
summary(lm.corr)$coefficients
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -3.472271  1.2041550 -2.883575 5.869025e-03
## t            1.111209  0.0410972 27.038547 1.057006e-30
```

- With the correlated errors,

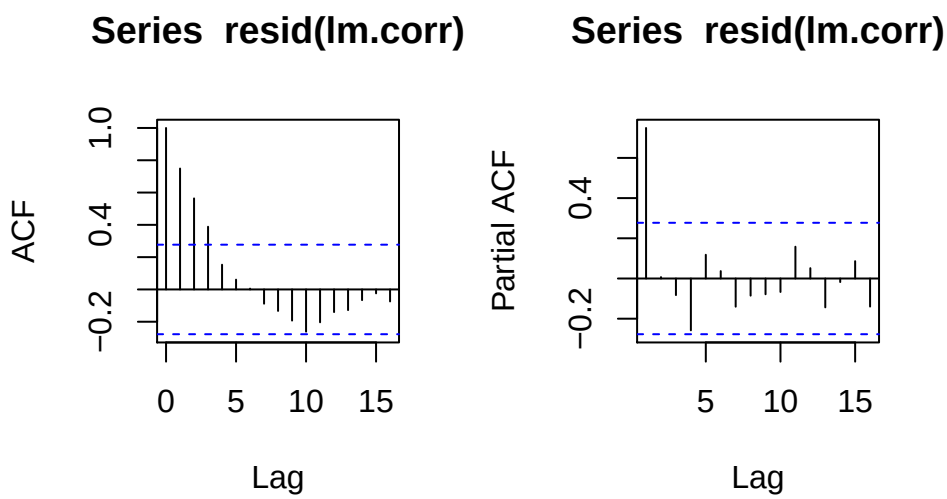
- Diagnostics are in important part of the modeling process, to verify assumptions and make sure the models are "useful". Look at the ACF and PACF plots for the residuals in each setting.

```
par(mfcol=c(1,2))
acf(resid(lm.ind)); pacf(resid(lm.ind))
```



- **Q:** what do we make of the ACF and PACF figures in this case? How does that compare with the "known model"?

```
par(mfcol=c(1,2))
acf(resid(lm.corr)); pacf(resid(lm.corr))
```



- **Q:** what do we make of the ACF and PACF figures in this case? How does that compare with the "known model"?

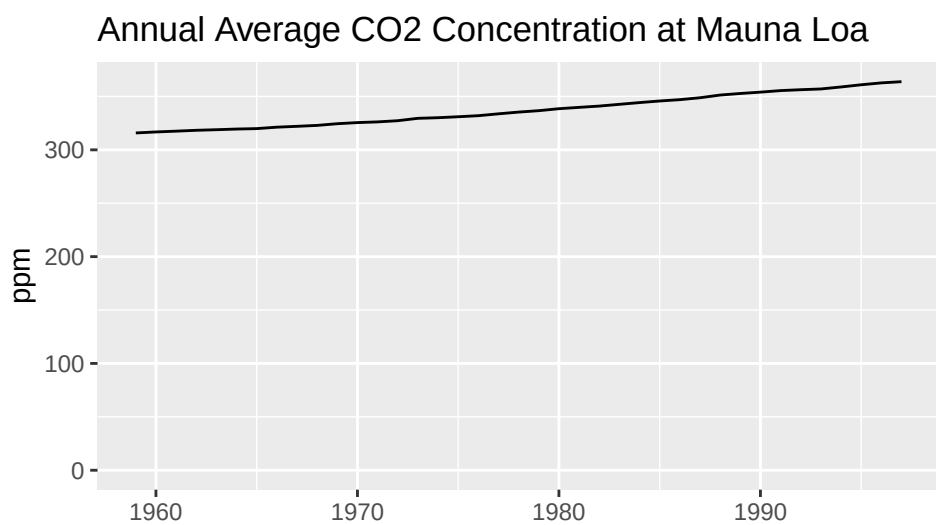
- To illustrate the effect of autocorrelation, consider the simple case of estimating a sample mean from a set of observations from a sample mean with $E[X_t] = \mu$ and $Var(X_t) = \sigma^2$.

- When the X_i^S are independent, $Var(\bar{x}) =$

- When $Cor(x_t, x_{t+k}) = \rho_k$, then $Var(\bar{x}) =$

- Now reconsider the CO_2 dataset.

```
co2.annual.mean <- aggregate(co2, FUN = 'mean')
library(ggfortify)
autoplot(co2.annual.mean) + ylim(0, max(co2.annual.mean)) +
  ggtitle('Annual Average CO2 Concentration at Mauna Loa') + ylab('ppm')
```



- Fit and assess a model for this dataset.

- There is clear evidence of autocorrelation in the residuals.

- So we cannot trust the standard errors on the model coefficients,

- but what is the solution…

Generalised Least Squares

- *Generalised least squares*

- Recall the data we simulated earlier

```r
summary(lm.corr)
```

```
##
## Call:
## lm(formula = x.corr ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4256 -2.4855 -0.7761  1.8925 10.5807
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.4723     1.2042  -2.884  0.00587 **
## t             1.1112     0.0411  27.039  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.194 on 48 degrees of freedom
## Multiple R-squared:  0.9384, Adjusted R-squared:  0.9371
## F-statistic: 731.1 on 1 and 48 DF,  p-value: < 2.2e-16
```

```r
library(nlme)
x.corr.gls <- gls(x ~ t, data=df.corr, correlation = corAR1(ar.coef))
summary(x.corr.gls)
```

```
## Generalized least squares fit by REML
##   Model: x ~ t
##   Data: df.corr
##        AIC      BIC    logLik
##   243.5734 251.0582 -117.7867
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi
## 0.9067894
##
## Coefficients:
##                 Value Std.Error   t-value p-value
## (Intercept) -2.607506  5.512467 -0.473020  0.6383
## t            1.141716  0.170775  6.685491  0.0000
##
##  Correlation:
##   (Intr)
## t -0.79
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -1.39977114 -0.71951437 -0.37987843  0.09780529  1.32986277
##
## Residual standard error: 6.227801
## Degrees of freedom: 50 total; 48 residual
```

- Now use the GLS procedure to refit the model for CO_2.