STAT 224 Lecture 15
Chapter 8 The Problem of Correlated Errors

Yibi Huang

Chapter 8 Outline

- What Are Correlated Errors and Why Worry About Them?
- Detection of Correlated Errors
  - Time plot of residuals
  - Runs test
  - Durbin-Watson test
  - Lag plots
  - Autocorrelation function and autocrrelation plot
- Remedies to Correct for Autocorrelated Errors
- Autocorrelation Due to Missing Predictors
- Autocorrelation and Seasonality

What Are Correlated Errors and Why Worry About Them?

Correlated Errors

- Recall in MLR Models:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i,$$

  the errors ε_i are assumed to be **independent**.
- In Ch8, we introduce the diagnosis and remedies for models with **correlated errors**.
- Correlated errors can arise when observations have a spatial or temporal order, e.g.,
  - Temporal: In sports, a player may exhibit hot or cold streaks in which he performs above or below expectation for several games.
  - Spatial: In agriculture studies, adjacent plots of land tend to be similar (soil, humidity, sun exposure)

Effects of Correlated Errors

- Least squares estimates, while still unbiased, no longer have minimal variance among unbiased estimators.
- σ^2 and the s.e. of β's would be seriously underestimated (if errors are positively correlated)
- Confidence intervals and significance tests are no longer accurate.

Causes of Autocorrelation

When the observations have a natural sequential order, the correlation is referred to as **autocorrelation**, which may occur for several reasons.

- Autocorrelation may occur due to an unmeasured predictor is associated with time or space.
  - Ex1: Athletes competing against exceptionally good or bad teams.
    This is especially evident in baseball because teams play each other 3-4 times in a row.
  - Ex2: Certain pests which inhibit plant growth may be more prevalent in some areas.
  - In this case, we can remove autocorrelation by accounting for these variables.
- **Pure autocorrelation** is not due to missing variables.

Ex: Consumer Expenditure & Money Stock Data (p.211)

Data: http://www.stat.uchicago.edu/~yibi/s224/data/P211.txt

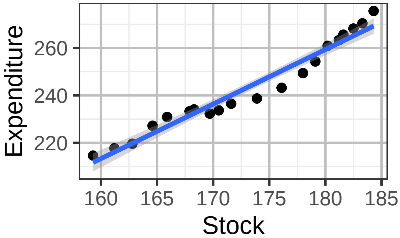
Quarterly data from 1952 to 1956 on consumer expenditure (Y = `Expenditure`) and the stock of money (X = `Stock`), both in millions of current US dollars.

```
p211 = read.table("P211.txt", h=T)
library(ggplot2)
ggplot(p211, aes(x=Stock, y=Expenditure)) +
  geom_point() + geom_smooth(method='lm')
```

If we fit the naive model,

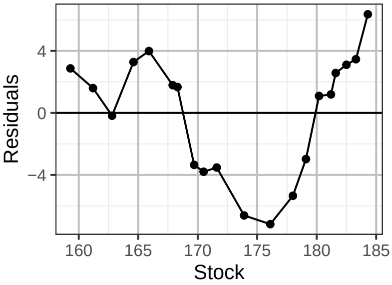
$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t$$

the residual plots looks like ...



Since the observations are ordered in time, we should also **plot the residuals against time (index)**.

```
lmp211 = lm(Expenditure ~ Stock, data=p211)
ggplot(p211, aes(x=Stock, y=lmp211$res)) +
  geom_point() + geom_line() +
  ylab("Residuals")+ geom_hline(yintercept=0)
ggplot(p211, aes(x=1:20, y=lmp211$res)) +
  geom_point() + geom_line() + xlab("Index") +
  ylab("Residuals")+ geom_hline(yintercept=0)
```



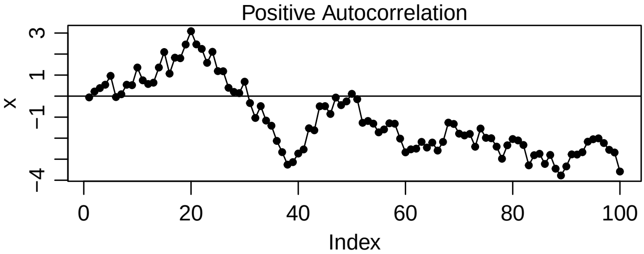
Diagnostic for Autocorrelation

Time Plot/Index Plot of Residuals

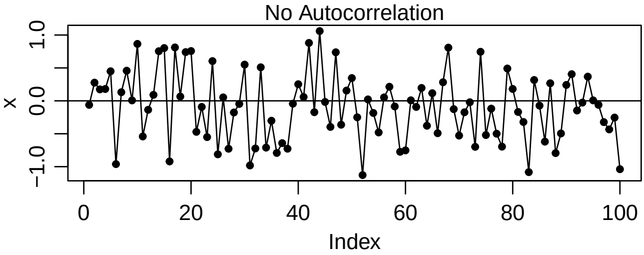
Time Plot/Index Plot of Residuals

A time plot or an index plot of the residuals is a plot of residuals v.s. the (time) order they are recorded. Points in a time-plot are connected by a line.

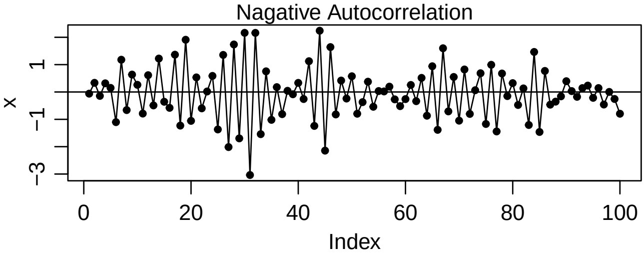
- Better keeping track of the time order observations are recorded so we can make a time-plot
- A *smooth* time-plot is a sign of *positive* autocorrelation, since a smooth time plot means successive residuals are close together



If *no* autocorrelation, the time plot has more up-and-downs.

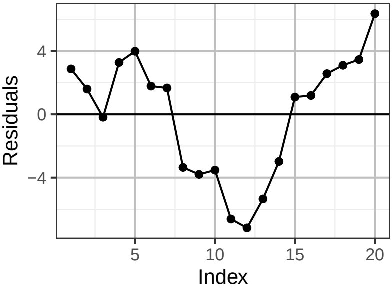


Time plot of data w/ *negative* autocorrelation tend to *alternate* regularly between positive and negative values.



For the Stock Data, the time plot is “smooth” which is a sign of positive autocorrelation.

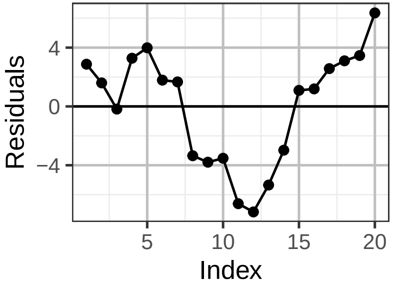
```
lmp211 = lm(Expenditure ~ Stock, data=p211)
ggplot(p211, aes(x=1:20, y=lmp211$res)) + geom_point() + geom_line() +
  labs(x="Index", y="Residuals")+ geom_hline(yintercept=0)
```



Runs Test

Analysis of Runs

Positive autocorrelation results in longer-than-usual *runs* of consecutive positive or negative residuals.

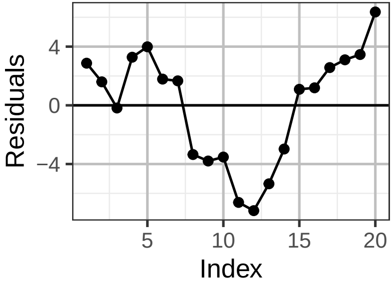


- For the Stocks Data, there are 5 separate runs:

$$+ + - + + + + - - - - - - - + + + + + +$$

- How many runs are expected when the residuals are independent?

Positive autocorrelation results in longer-than-usual *runs* of consecutive positive or negative residuals.



- For the Stocks Data, there are 5 separate runs:

$$\underbrace{++}_{1st\ run}\overbrace{-}^{2nd\ run}\underbrace{++++}_{3rd\ run}\overbrace{------}^{4th\ run}\underbrace{++++++}_{5th\ run}$$

- How many runs are expected when the residuals are independent?

Distribution of Runs

Assuming independence, with n_1 positive and n_2 negative residuals, the expected number μ and variance σ^2 of *runs* are

$$\mu = \frac{2n_1 n_2}{n_1 + n_2} + 1, \quad \text{and} \quad \sigma^2 = \frac{2n_1 n_2(2n_1 n_2 - n_1 - n_2)}{(n_1 + n_1)^2(n_1 + n_2 - 1)}.$$

In the Stock Data example, $n_1 = 12$ and $n_2 = 8$ (text incorrectly says $n_1 = 13$, $n_2 = 7$), and

$$\mu = \frac{2n_1 n_2}{n_1 + n_2} + 1 = \frac{2 \cdot 12 \cdot 8}{12 + 8} + 1 = 10.6,$$

$$\sigma^2 = \frac{2n_1 n_2(2n_1 n_2 - n_1 - n_2)}{(n_1 + n_2)^2(n_1 + n_2 - 1)} = \frac{2 \cdot 12 \cdot 8(2 \cdot 12 \cdot 8 - 12 - 8)}{(12 + 8)^2(12 + 8 - 1)} \approx 4.345$$

We hence expect to see 10.6 runs w/ the SD $\approx \sqrt{4.345} \approx 2.0845$.

- We observed only 5 runs, is this unusual under the null hypothesis of no autocorrelation?

Normal Approx for the Runs Test

If n_1 and n_2 are large (say ≥ 10),

$$\text{Number of Runs is approx.} \sim N(\mu, \sigma^2)$$

then we can use an approximate z-statistic

$$z = \frac{\text{Number of Runs} - \mu}{\sigma} \sim \text{approx. } N(0, 1)$$

For our example,

$$z = \frac{5 - 10.6}{2.0845} \approx -2.6865$$

The two-sided P-value is `2*pnorm(-2.686)` =0.0072

Runs Test in R

The command `runs.test()` in the `tseries` library can perform the runs test. You need to first install the `tseries`.

```
install.packages("tseries") # Only Install ONCE!
```

```
library(tseries)
runs.test(factor(lmp211$res > 0)) # two-sided by default


    Runs Test


data:  factor(lmp211$res > 0)
Standard Normal = -2.686, p-value = 0.00722
alternative hypothesis: two.sided
```

For testing *positive* autocorrelation, use `alternative = "less"` as positive autocorrelation leads to fewer runs.

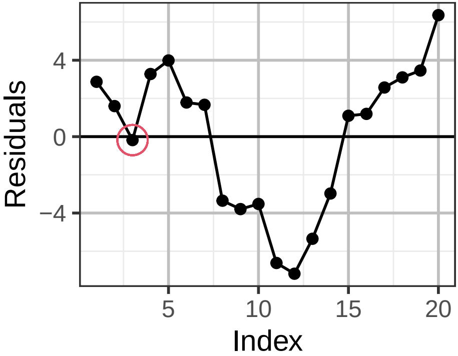
```
runs.test(factor(lmp211$res > 0), alternative = "less")


    Runs Test


data:  factor(lmp211$res > 0)
Standard Normal = -2.686, p-value = 0.00361
alternative hypothesis: less
```

For testing *negative* autocorrelation, use `alternative = "greater"` as negative autocorrelation leads to more runs.

```
runs.test(factor(lmp211$res > 0), alternative = "greater")


    Runs Test


data:  factor(lmp211$res > 0)
Standard Normal = -2.686, p-value = 0.996
alternative hypothesis: greater
```

Pros and Cons of the Runs Test

- Pros: Simple, intuitive
- Cons: It ignores the magnitude of the residuals $|e_i|$.
- In the Stock Data, the 3rd residual is just barely below 0.
  If it was above 0, we'd have 3 runs only, not 5 runs.
  Evidence of correlated errors could be stronger.



Durbin-Watson Test

Durbin-Watson Statistic

- Proceeds from the assumption that successive errors are correlated:

$$\varepsilon_t = \rho\varepsilon_{t-1} + \omega_t, \quad |\rho| < 1$$

- Note: In Time Series analysis, this is called a **first-order autoregressive model**, abbreviated $AR(1)$, or **first-order autocorrelation**
- The actual autocorrelation structure may be more complex (e.g. $AR(2)$, $AR(3)$, etc.) In this case, the first-order structure is a simple approximation.

Theorem (Durbin-Watson statistic)

$$d = \frac{\sum_{t=2}^{n}(e_t - e_{t-1})^2}{\sum_{t=1}^{n} e_t^2},$$

where e_i is the ordinary least square residual.

Properties of d

- $d \approx 2(1-\hat{\rho})$, where $\hat{\rho}$ estimates the aurocorrelation ρ by

$$\hat{\rho} = \frac{\sum_{t=2}^{n} e_t e_{t-1}}{\sum_{t=1}^{n} e_t^2}.$$

- d is a test statistic for testing

$$H_0 : \rho = 0 \quad \text{vs.} \quad H_a : \rho > 0.$$

- The null hypothesis indicates that successive residuals are not correlated.
- Under the H_0 of no autocorrelation, d should be close to 2.

DW Test for Positive Autocorrelation

- Is d is significantly different than 2?
- We use two cut-off values, d_L and d_U, which depend on the number of parameters p, the sample size n, and the desired significance level α of the test.
  - $d < d_L$, reject H_0
  - $d > d_U$, do not reject H_0.
  - $d_L < d < d_U$, the test is inconclusive.
- The values for d_L and d_U are given in Tables A.6 and A.7.
- For the Stock Data, $p = 1, n = 20, d = 0.329$.
  - For an $\alpha = .05$ test, we use Table A.6 to see $(d_L, d_U) = (1.20, 1.41)$. We reject H_0 and infer positive autocorrelation.
  - For an $\alpha = .01$ test, we use Table A.7 to see $(d_L, d_U) = (0.95, 1.15)$. We also reject at the 1% significance level.

Remarks about DW Test

- To test for **negative autocorrelation**, use the test statistic $(4 - d)$ then follow the test for positive autocorrelation.
- When $d_L < d < d_U$, the test is inconclusive.
  A good strategy is to correct for autocorrelation and see if the model changes in a major way.
- Unfortunately, the Durbin-Watson test can be fooled by higher-order autocorrelation structure.
- As always, there is no substitute for diagnostic graphs!

Durbin-Watson Test in R

In R, `durbinWatsonTest()` in `library(car)` can produce an approximate P-value (by simulation) for the DW test.

```
library(car)
durbinWatsonTest(lmp211, alt="positive")
 lag Autocorrelation D-W Statistic p-value
   1        0.750612      0.328211       0
 Alternative hypothesis: rho > 0
durbinWatsonTest(lmp211, alt="negative")
 lag Autocorrelation D-W Statistic p-value
   1        0.750612      0.328211       1
 Alternative hypothesis: rho < 0
durbinWatsonTest(lmp211)  # two-sided by default
 lag Autocorrelation D-W Statistic p-value
   1        0.750612      0.328211       0
 Alternative hypothesis: rho != 0
```

Lag Plots

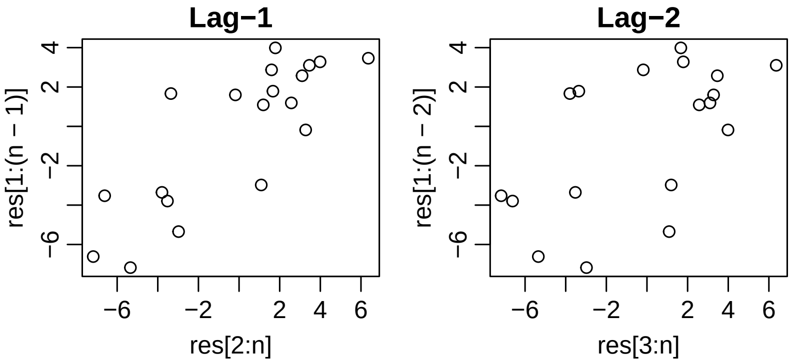
Plotting Residuals Against Lag-k Residuals

If successive residuals are correlated, we would observe a positive correlation when we plot the residuals $(e_1, \ldots, e_{n-1})$ against the next ones $(e_2, \ldots, e_n)$ (Lag 1).

- or $(e_1, \ldots, e_{n-k})$ against the lag-k residuals $(e_{1+k}, \ldots, e_n)$
- Any trend in the plot is a sign of autocorrelation.

Lag 1	Lag k
(e_1, e_2)	(e_1, e_{1+k})
(e_2, e_3)	(e_2, e_{2+k})
(e_3, e_4)	(e_3, e_{3+k})
$\vdots$	$\vdots$
(e_{n-1}, e_n)	(e_{n-k}, e_n)

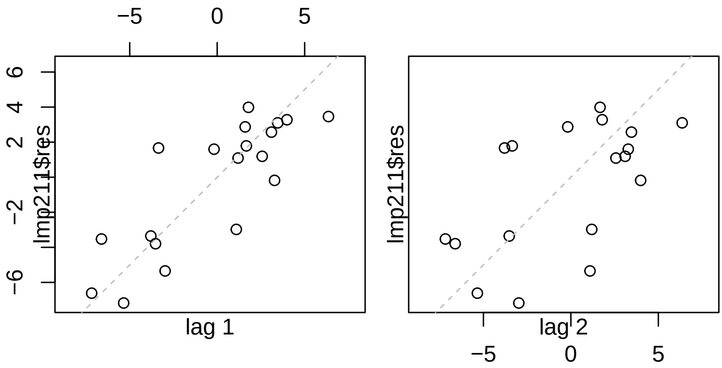
```r
res = lmp211$res
n = length(res)
plot(res[2:n], res[1:(n-1)], main="Lag-1")
plot(res[3:n], res[1:(n-2)], main="Lag-2")
```



Lag Plots in R

The `lag.plot()` command can produce lag plots.

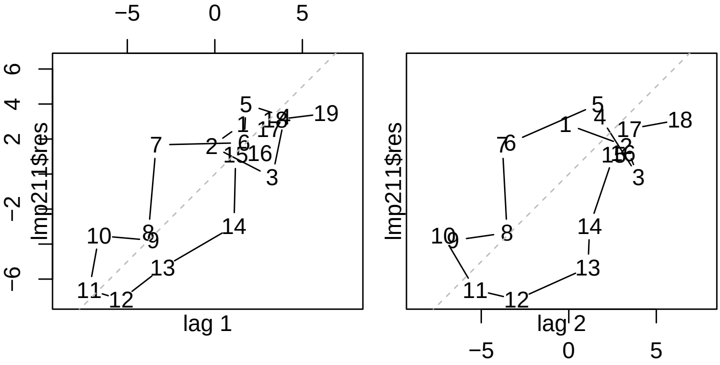
```
lag.plot(lmp211$res, lags=2, layout=c(1,2), do.lines=FALSE)
```



- `lags = k` would produce lag-1 to lag-k plots
- `layout = c(1,2)` arranges the plots in 1 row and 2 columns.

If not specifying `do.lines=FALSE`, the plots would look like the following

```
lag.plot(lmp211$res, lags=2, layout=c(1,2))
```



Autocorrelation Functions

Autocorrelation

The **lag-k autocorrelation** of the residuals $(e_1, \ldots, e_n)$ is defined as

$$\hat{\rho}_k = \frac{\sum_{t=k+1}^{n} e_t e_{t-k}}{\sum_{t=1}^{n} e_t^2}, \quad k = 1, 2, 3, \ldots.$$

which is slightly different from the "correlation" of $(e_1, \ldots, e_{n-k})$ v.s. $(e_{1+k}, \ldots, e_n)$,

$$\frac{\sum_{t=k+1}^{n} e_t e_{t-k}}{\sqrt{\sum_{t=1}^{n-k} e_t^2 \sum_{t=k+1}^{n} e_t^2}}$$

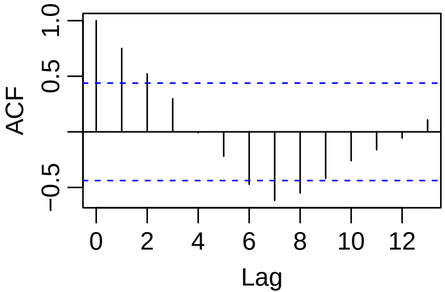
The R command `acf()` (autocorrelation function) in R can calculate lag-k autocorrelation.

```
acf(lmp211$res, lag.max =5, plot=FALSE)

Autocorrelations of series 'lmp211$res', by lag

     0      1      2      3      4      5
 1.000  0.751  0.521  0.297 -0.007 -0.220
```

Autocorrelation Function and the Plot

In time-series analysis, one often plot the lag-k autocorrelations against k to examine the autocorrelation structure of a variable. The `acf()` command can produce such **autocorrelation plot**.

```
acf(lmp211$res)
```



The horizontal dash lines marks the levels autocorrelations to be significantly different from 0.

Remedies to Correct for Autocorrelated Errors

Removal of Autocorrelation with Transformation

Assume the errors ε_t's of the linear model $y_t = \beta_0 + \beta_1 x_t + \varepsilon_t$ have the **first order autocorrelation** AR(1) structure

$$\varepsilon_t = \rho\varepsilon_{t-1} + \omega_t, \quad \text{where } \omega_t \text{ are indep. } \sim N(0, \theta^2)$$

Then

$$\begin{aligned} \overbrace{y_t - \beta_0 - \beta_1 x_t}^{\varepsilon_t} &= \rho(\overbrace{y_{t-1} - \beta_0 - \beta_1 x_{t-1}}^{\varepsilon_{t-1}}) + \omega_t \\ \underbrace{y_t - \rho y_{t-1}}_{y_t^*} &= \beta_0(1-\rho) + \beta_1(\underbrace{x_t - \rho x_{t-1}}_{x_t^*}) + \omega_t \end{aligned}$$

Removal of Autocorrelation with Transformation

Assume the errors ε_t's of the linear model $y_t = \beta_0 + \beta_1 x_t + \varepsilon_t$ have the **first order autocorrelation** AR(1) structure

$$\varepsilon_t = \rho\varepsilon_{t-1} + \omega_t, \quad \text{where } \omega_t \text{ are indep. } \sim N(0, \theta^2)$$

Then

$$\begin{aligned} \overbrace{y_t - \beta_0 - \beta_1 x_t}^{\varepsilon_t} &= \rho(\overbrace{y_{t-1} - \beta_0 - \beta_1 x_{t-1}}^{\varepsilon_{t-1}}) + \omega_t \\ \underbrace{y_t - \rho y_{t-1}}_{y_t^*} &= \beta_0(1-\rho) + \beta_1(\underbrace{x_t - \rho x_{t-1}}_{x_t^*}) + \omega_t \end{aligned}$$

Hence the transformed variables $x_t^* = x_t - \rho x_{t-1}$ and $y_t^* = y_t - \rho y_{t-1}$ satisfy the SLR model

$$y_t^* = \beta_0^* + \beta_1^* x_t^* + \omega_t, \text{ where } \omega_t \text{ are indep.} \sim N(0, \theta^2)$$

The coefficients of the orignal and the transformed models are related as follows

$$\beta_0^* = \beta_0(1 - \rho), \quad \beta_1^* = \beta_1$$

However, we need to estimate ρ!

Cochrane-Orcutt Method

1. Fit the OLS model and obtain the residuals $e_1, \ldots, e_n$
2. Use the residuals $e_1, \ldots, e_n$ to estimate ρ with

$$\hat{\rho} = \frac{\sum_{t=2}^{n} e_t e_{t-1}}{\sum_{t=1}^{n} e_t^2}$$

3. Compute OLS estimates of β_0^* and β_1^* by regressing

$$y_t^* = y_t - \hat{\rho} y_{t-1} \quad \text{on} \quad x_t^* = x_t - \hat{\rho} x_{t-1}$$

   and use them to find coefficients for the original variables.

$$\widehat{\beta}_0 = \frac{\widehat{\beta_0^*}}{1 - \hat{\rho}} \quad \text{and} \quad \widehat{\beta}_1 = \widehat{\beta_1^*}$$

4. Use the new $\widehat{\beta}_0$ and $\widehat{\beta}_1$ to calulate the new residuals $e_1, \ldots, e_n$ and then go back to Step 2.
5. Iterate until the estimates $\widehat{\beta}_0$ and $\widehat{\beta}_1$ converge.

First Iteration

```
x = p211$Stock
y = p211$Expenditure
n = length(y)
fit1 = lm(y ~ x)
res = fit1$res
rho.hat = sum(res[1:(n-1)]*res[2:n]) / sum(res^2 )
rho.hat
[1] 0.7506
ystar = y[2:n] - rho.hat*y[1:(n-1)]
xstar = x[2:n] - rho.hat*x[1:(n-1)]
fit2 = lm(ystar ~ xstar)
b0.hat = fit2$coef[1]/(1-rho.hat)
b1.hat = fit2$coef[2]
c(b0.hat, b1.hat)
(Intercept)       xstar
   -215.311       2.643
```

Second Iteration

```
res = y - b0.hat - b1.hat*x
rho.hat = sum(res[1:(n-1)]*res[2:n]) / sum(res^2 )
rho.hat
[1] 0.79
ystar = y[2:n] - rho.hat*y[1:(n-1)]
xstar = x[2:n] - rho.hat*x[1:(n-1)]
fit2 = lm(ystar ~ xstar)
b0.hat = fit2$coef[1]/(1-rho.hat)
b1.hat = fit2$coef[2]
c(b0.hat, b1.hat)
(Intercept)       xstar
     -225.6         2.7
```

Complete R Codes of Cochrane-Orcutt Method

```
x = p211$Stock
y = p211$Expenditure
n = length(y)
n.iter = 15
rho.iter = vector("numeric", n.iter)
b0.iter = vector("numeric", n.iter)
b1.iter = vector("numeric", n.iter)
fit1 = lm(y ~ x)
res = fit1$res
rho.iter[1] = sum(res[1:(n-1)]*res[2:n]) / sum(res^2 )
for(i in 2:n.iter){
  rho.iter[i] = sum(res[1:(n-1)]*res[2:n]) / sum(res^2 )
  ystar = y[2:n] - rho.iter[i]*y[1:(n-1)]
  xstar = x[2:n] - rho.iter[i]*x[1:(n-1)]
  fit2 = lm(ystar ~ xstar)$coef
  b0.iter[i] = fit2[1]/(1-rho.iter[i])
  b1.iter[i] = fit2[2]
  res = y - b0.iter[i] - b1.iter[i]*x
}
```

```
data.frame(rho.iter,b0.iter,b1.iter)
   rho.iter b0.iter b1.iter
1    0.7506     0.0   0.000
2    0.7506  -215.3   2.643
3    0.7900  -225.6   2.700
4    0.7977  -227.8   2.712
5    0.7996  -228.3   2.715
6    0.8000  -228.5   2.715
7    0.8001  -228.5   2.716
8    0.8002  -228.5   2.716
9    0.8002  -228.5   2.716
10   0.8002  -228.5   2.716
11   0.8002  -228.5   2.716
12   0.8002  -228.5   2.716
13   0.8002  -228.5   2.716
14   0.8002  -228.5   2.716
15   0.8002  -228.5   2.716
```

We can see that the estimates for ρ, β_0, β_1 converge quickly to

$$\hat{\rho} = 0.8002, \quad \hat{\beta}_0 = -228.5212, \quad \hat{\beta}_1 = 2.7157.$$

Checking the Independence Assumption After Transformation

Recall the transformed variables $x_t^* = x_t - \rho x_{t-1}$ and $y_t^* = y_t - \rho y_{t-1}$ satisfy the SLR model with indep. errors

$$y_t^* = \beta_0^* + \beta_1^* x_t^* + \omega_t, \text{ where } \omega_t \text{ are indep.} \sim N(0, \theta^2)$$

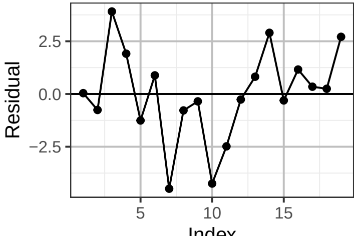
Let's obtain the residuals for the model $y_t^* = \beta_0^* + \beta_1^* x_t^* + \omega_t$ and check if they exhibit any autocorrelation.

Time Plot of Residuals

Time plot of the residuals for the model $y_t^* = \beta_0^* + \beta_1^* x_t^* + \omega_t$.

```
rho.hat = rho.iter[n.iter]
ystar = y[2:n] - rho.hat*y[1:(n-1)]
xstar = x[2:n] - rho.hat*x[1:(n-1)]
xystar = data.frame(xstar, ystar)
fit2 = lm (ystar ~ xstar)
ggplot(xystar, aes(x=1:(n-1), y = fit2$res)) +
  geom_point() + geom_line() +
  labs(x="Index", y="Residual") + geom_hline(yintercept=0)
```

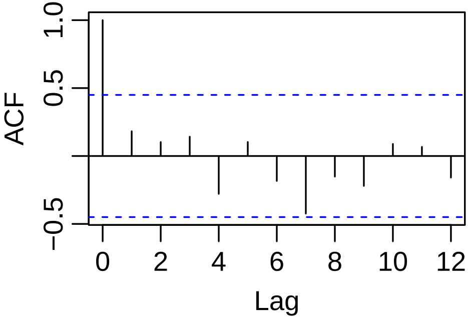
The time plot is no longer "smooth". No longer-than-usual runs of consecutive positive or negative residuals.



Autocorrelation Plot of the Residuals

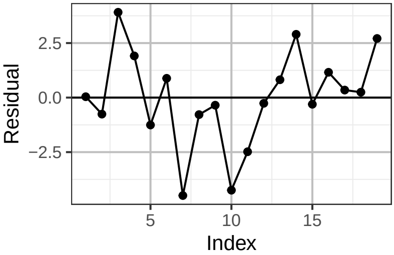
Below is the autocorrelation plot of the residuals for the model $y_t^* = \beta_0^* + \beta_1^* x_t^* + \omega_t$.

```
acf(fit2$res)
```



None of the lag-k autocorrelations is significant. (All are between the two horizontal dash lines), $k = 1, 2, 3, \ldots$

Runs Test



The signs of the 19 residuals are

$$+ - + + - + - - - - - - + + - + + + +$$

There are 9 runs, $n_1 = 10$ positives and $n_2 = 9$ negatives.

The expected value and SD of the number of runs when $n_1 = 10$ positives and $n_2 = 9$ negatives are

$$\mu = \frac{2n_1n_2}{n_1 + n_2} + 1 = \frac{2 \cdot 10 \cdot 9}{10 + 9} + 1 \approx 10.474,$$
$$\sigma = \sqrt{\frac{2n_1n_2(2n_1n_2 - n_1 - n_2)}{(n_1 + n_1)^2(n_1 + n_2 - 1)}} = \sqrt{\frac{2 \cdot 10 \cdot 9(2 \cdot 10 \cdot 9 - 10 - 9)}{(10 + 9)^2(10 + 9 - 1)}} \approx 2.112$$

Then z-statistic is

$$z = \frac{\text{Number of Runs} - \mu}{\sigma} = \frac{9 - 10.474}{2.112} \approx -0.698$$

The two-sided P-value is `2*pnorm(-0.698)` =0.4852.

No significant evidence of autocorrelation.

Durbin-Watson Test

```
library(car)
durbinWatsonTest(fit2, alt="positive")
 lag Autocorrelation D-W Statistic p-value
   1          0.1825         1.549     0.1
 Alternative hypothesis: rho > 0
durbinWatsonTest(fit2)    # default is two-sided
 lag Autocorrelation D-W Statistic p-value
   1          0.1825         1.549   0.174
 Alternative hypothesis: rho != 0
```

The P-values are over 0.05.

No significant evidence of autocorrelation.

Autocorrelation Due to Missing Predictors

Missing Variables and Autocorrelation

- ε_t is variation that cannot be explained by covariates in the model.
- This can be due to non-systematic random errors...
- ... or possibly important predictors missing from the model!
- If the missing predictors are associated with t, then residual analysis will exhibit autocorrelation.
- This type of autocorrelation can be considered “artificial”
- The autocorrelation may disappear when the predictor is included

Pure vs. Artificial Autocorrelation

- There is no foolproof analysis to differentiate pure autocorrelation from missing predictors.
- In general, we should consider both.
- It is better if we can improve the model with new predictors
  - We improve our understanding of the process.
  - We understand what caused the autocorrelation.
  - We avoid relying on structured residuals.
  - It is more satisfying to have nice, independent random errors.
- Techniques to correct pure autocorrelation are a last resort.

Example: Housing Starts (p. 219)

Data: http://www.stat.uchicago.edu/~yibi/s224/data/P219.txt

A construction industry association is interested in forecasting housing construction activity. As a starting place, they gather historical data on the population size of 22- 44-year olds as an estimate of the number of potential buyers.

One can load the data by the command

```
p219 = read.table("P219.txt", h=T)
```

The variables are

- `H`: Housing Starts
- `P`: Population Size of 22- to 45-yr-olds in millions
- `D`: Availability for Mortgage Money Index

Model 1 of Housing Starts

$$H_t = \beta_0 + \beta_1 P_t + \varepsilon_t,$$

```
model1 = lm(H ~ P, data=p219)
```

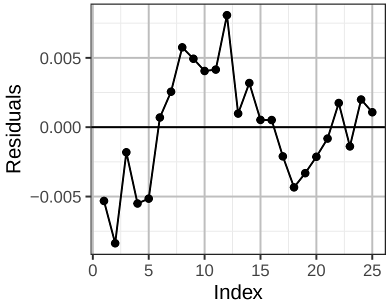
```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.060884   0.010416  -5.845 5.89e-06 ***
P            0.071410   0.004234  16.867 1.91e-14 ***
---
Residual standard error: 0.00408 on 23 degrees of freedom
Multiple R-squared:  0.9252,    Adjusted R-squared:  0.922
F-statistic: 284.5 on 1 and 23 DF,  p-value: 1.911e-14
```

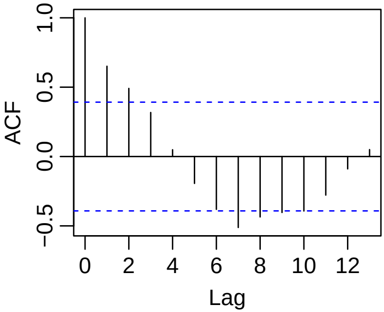
- Naively, the model fits well with $R^2 = .9252$.
- Due to the temporal nature of the data, we must check for autocorrelation.

```
durbinWatsonTest(model1, alt="positive")
 lag Autocorrelation D-W Statistic p-value
   1          0.6511         0.6208       0
 Alternative hypothesis: rho > 0
```

```
Warning: Use of ‘p219$H‘ is discouraged. Use ‘H‘ instead.
```

```
Warning: Use of ‘p219$H‘ is discouraged. Use ‘H‘ instead.
```





The time plot and ACF plot of residuals exhibit clear autocorrelation too.

What's Missing?

- Autocorrelation is suspected…
- But first, there are many reasonable variables we should consider in this case.
  - unemployment rate, social trends, government programs, availability of construction funds…
- Our choice is an index that measures availability of mortgage money, D_t and hence we consider the model

$$H_t = \beta_0 + \beta_1 P_t + \beta_2 D_t + \varepsilon_t$$

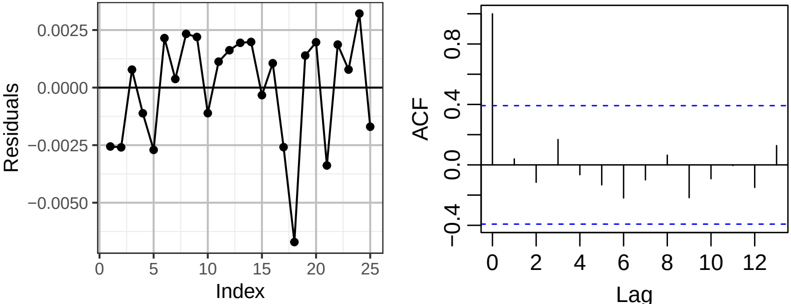
```
model2 = lm(H ~ P + D, data=p219)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.010427   0.010291  -1.013    0.322
P            0.034656   0.006425   5.394 2.04e-05 ***
D            0.760464   0.121588   6.254 2.70e-06 ***
---
Residual standard error: 0.002503 on 22 degrees of freedom
Multiple R-squared:  0.9731,    Adjusted R-squared:  0.9706
```

```
durbinWatsonTest(model2, alt="positive")
 lag Autocorrelation D-W Statistic p-value
   1         0.03957         1.852    0.25
 Alternative hypothesis: rho > 0
```

- Durbin Watson's test shows little sign of autocorrelation (P-value 0.223)

```
Warning: Use of ‘p219$H‘ is discouraged. Use ‘H‘ instead.

Warning: Use of ‘p219$H‘ is discouraged. Use ‘H‘ instead.
```



Comparison of Model 1 and Model 2

- Model 2 has a better adjusted R^2 and no obvious autocorrelation.
- The Mortgage Index accounted for the autocorrelation.

Moral:

1. A high R^2 does not necessarily indicate that the response variation is adequately understood.
2. The Durbin-Watson statistics, residual plot, and ACF plot may indicate autocorrelation when the real problem is one or more important variables unaccounted for in the model.
3. Typically, any two variables measured over long stretches of time seem highly-correlated.

Autocorrelation and Seasonality

Limitation of Durbin-Watson Statistic

- d cannot distinguish between pure and artificial autocorrelation.
- d only measures first-order autocorrelation (i.e. between adjacent observations).
- Sometimes, ε_t is correlated with ε_{t-2} (second-order autocorrelation), or errors even further back (higher-order autocorrelation).
- Time plot of residuals is less helpful when there are higher-order autocorrelation but not first-order autocorrelation
- ACF plots are best for detecting higher-order dependence.

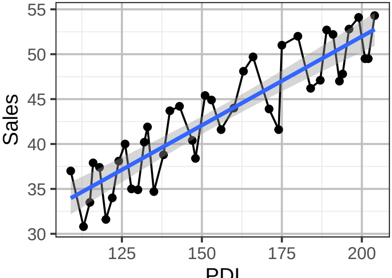
Example: Ski Sales Data (p.149)

Data: http://www.stat.uchicago.edu/~yibi/s224/data/P149.txt

- `Sales`: Sales of skis and related equipment in millions
- `PDI`: personal disposable income

Both variables are measured quarterly for the years 1964-1973

```
ski = read.table("P149.txt", h=T)
ggplot(ski, aes(x=PDI, y=Sales)) + geom_point() +
  geom_line() + geom_smooth(method='lm')
```

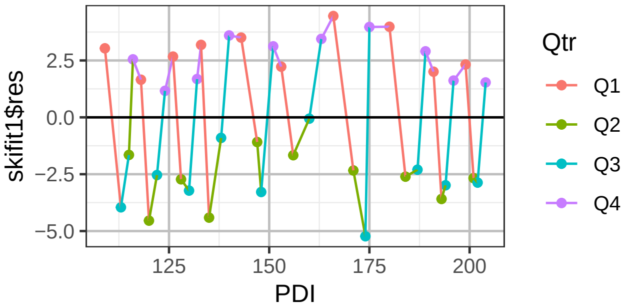


Extracting “Quarter” From “Date”

```
ski$Date
 [1] "Q1/64" "Q2/64" "Q3/64" "Q4/64" "Q1/65" "Q2/65" "Q3/65" "Q4/65" "Q
[10] "Q2/66" "Q3/66" "Q4/66" "Q1/67" "Q2/67" "Q3/67" "Q4/67" "Q1/68" "Q
[19] "Q3/68" "Q4/68" "Q1/69" "Q2/69" "Q3/69" "Q4/69" "Q1/70" "Q2/70" "Q
[28] "Q4/70" "Q1/71" "Q2/71" "Q3/71" "Q4/71" "Q1/72" "Q2/72" "Q3/72" "Q
[37] "Q1/73" "Q2/73" "Q3/73" "Q4/73"
ski$Qtr = substr(ski$Date, start=1, stop=2)
ski$Qtr
 [1] "Q1" "Q2" "Q3" "Q4" "Q1" "Q2" "Q3" "Q4" "Q1" "Q2" "Q3" "Q4" "Q1" "
[16] "Q4" "Q1" "Q2" "Q3" "Q4" "Q1" "Q2" "Q3" "Q4" "Q1" "Q2" "Q3" "Q4" "
[31] "Q3" "Q4" "Q1" "Q2" "Q3" "Q4" "Q1" "Q2" "Q3" "Q4"
```

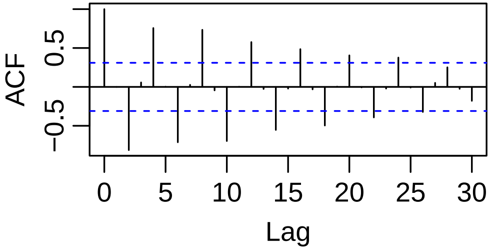
Time Plot of Residuals

```
skifit1 = lm(Sales ~ PDI, data=ski)
ggplot(ski, aes(x=PDI, y=skifit1$res, col=Qtr, group=I(1))) +
  geom_point() + geom_line() +
  geom_hline(yintercept = 0)
```



Autocorrelation Plot

```
acf(skifit1$res, lag.max=30)
acf(skifit1$res, lag.max=30, plot=F)
```




```
Autocorrelations of series 'skifit1$res', by lag

     0      1      2      3      4      5      6      7      8      9
 1.000 -0.001 -0.813  0.058  0.757  0.002 -0.712  0.026  0.734 -0.044 -
    11     12     13     14     15     16     17     18     19     20
 0.002  0.577 -0.026 -0.553 -0.022  0.486 -0.031 -0.497  0.003  0.405 -
    22     23     24     25     26     27     28     29     30
```

- $\hat{\rho}_k \approx 0$ when k are odd numbers
- $\hat{\rho}_k > 0$ for k's that are multiples of 4
- $\hat{\rho}_k < 0$ for $k = 2, 6, 10, 14, \ldots$

Durbin-Watson Test Failed!

```
durbinWatsonTest(skifit1, alt="positive")
 lag Autocorrelation D-W Statistic p-value
   1      -0.0008867         1.968   0.422
 Alternative hypothesis: rho > 0
durbinWatsonTest(skifit1)
 lag Autocorrelation D-W Statistic p-value
   1      -0.0008867         1.968   0.842
 Alternative hypothesis: rho != 0
```

Durbin-Watson test give large P-values even though there exist significant lag-2 autocorrelation

Treating Seasonal Autocorrelation

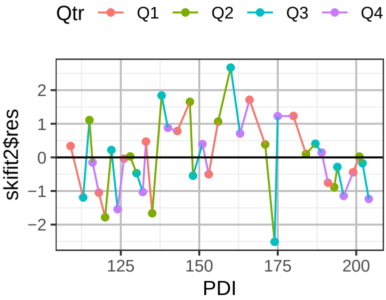
- We can account for seasonality using indicator variables.
- $W_t = 1$ for winter season, (Q1 and Q4).
- De-seasonality model:

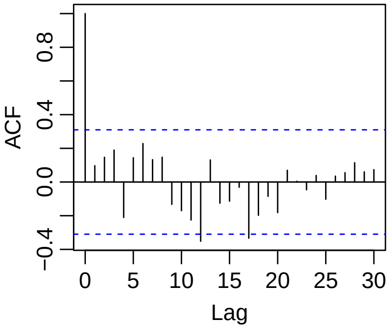
$$\text{Sales}_t = \beta_0 + \beta_1 \text{PDI}_t + \beta_2 W_t + \varepsilon_t$$

```
ski$Winter = (ski$Qtr == "Q1") | ski$Qtr == "Q4"
ski$Winter
 [1]  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
[13]  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
[25]  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
[37]  TRUE FALSE FALSE  TRUE
skifit2 = lm(Sales ~ PDI + Winter, data=ski)
```

Time Plot and ACF Plot After Accounting for Seasonality

```
ggplot(ski, aes(x=PDI, y=skifit1$res, col=Qtr, group=I(1))) +
  geom_point() + geom_line() + geom_hline(yintercept = 0) +
  theme(legend.position="top")
acf(skifit2$res, lag.max=30)
```





```
summary(skifit2)

Call:
lm(formula = Sales ~ PDI + Winter, data = ski)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.5112 -0.7864  0.0263  0.7284  2.6704 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.54020    0.97483    9.79  8.2e-12
PDI          0.19868    0.00604   32.91  < 2e-16
WinterTRUE   5.46434    0.35968   15.19  < 2e-16

Residual standard error: 1.14 on 37 degrees of freedom
Multiple R-squared:  0.972, Adjusted R-squared:  0.971 
F-statistic:  653 on 2 and 37 DF,  p-value: <2e-16
```