Indicator Variables for Seasonal Time Series

- A simple way to estimate seasonal effects in a time series; e.g., for quarterly data:
  - Set up four indicator ("dummy") variables, one for each quarter;
  - Use them as inputs in a regression model.

- Similarly for monthly data: twelve monthly indicators.
In R, you can create the indicators manually.

– E.g., for a series $x$, Shumway and Stoffer suggest essentially:

$$Q1 = \text{rep}(c(1, 0, 0, 0), \text{length}(x) / 4)$$
etc.

You can then use `lm()` to fit the regression;

– E.g., for the Johnson & Johnson quarterly earnings, fitting a linear trend and the quarterly indicators:

```r
summary(lm(log(jj) ~ time(jj) + Q1 + Q2 + Q3 + Q4))
```

For a monthly series, this would be tedious.
- R has some tools that can help:
  - If \( x \) is a seasonal time series (i.e., \( \text{frequency}(x) > 1 \)), \( \text{cycle}(x) \) creates a companion time series whose value is the corresponding season.
  - E.g. \( \text{cycle}(jj) \):
    
    | Qtr1 | Qtr2 | Qtr3 | Qtr4 |
    |------|------|------|------|
    | 1960 | 1    | 2    | 3    | 4    |
    | 1961 | 1    | 2    | 3    | 4    |
    | 1962 | 1    | 2    | 3    | 4    |
    |      |      |      |      |      |
  - Actually, \( x \) does not need to be seasonal, but if \( \text{frequency}(x) == 1 \) then all the values are 1.
• The time series `cycle(x)` has quantitative values:

```
plot(cycle(jj), xlim = c(1960, 1965))
```
• If you create a `factor()` from this time series and include it in the regression, `lm()` will see that it is a factor, and create one indicator variable for each level.

  – E.g., for the Johnson & Johnson quarterly earnings, fitting a linear trend and the quarterly indicators:

    ```r
    Q = factor(cycle(jj))
    summary(lm(log(jj) ~ time(jj) + Q))
    ```

    gives (almost) the same output as if you used `Q1, Q2, Q3, and Q4`.

  – The difference is that when the model includes an intercept, one indicator needs to be omitted; for a factor, the first indicator is omitted, but for an explicit list of variables, the last is omitted.
Lagged Variables

• Time series models often include lagged variables.

• You can use the R function `lag()` to construct them.

• For example (using just the first five quarters’ earnings):

```r
> x = window(jj, end = 1961)
> x
         Qtr1 Qtr2 Qtr3 Qtr4
1960   0.71 0.63 0.85 0.44
1961   0.61
```
```r
> lag(x, k = -1)
  Qtr1 Qtr2 Qtr3 Qtr4
1960  0.71  0.63  0.85
1961  0.44  0.61
```

- Note that `lag(x, k = -1)` contains the same five values as `x`, but associated with different times.
  - For instance, the value 0.44 of `lag(x, k = -1)` for the first quarter of 1961 is 1960's fourth quarter earnings (from `x`).

- The default is `k = 1`, which changes the times the wrong way for most applications.
• Many R functions use the time structure of a series to “do the right thing”.

• For example, `plot(lag(x, -1), x, xy.labels = FALSE)` plots each quarter’s earnings against the previous quarter’s:

![Plot of lag(x, -1) vs x](image)
• Some functions do not (\texttt{lm()}, \texttt{lowess()}).

• For these, you must first line up the data correctly, e.g., using \texttt{cbind()}:

\begin{verbatim}
> y = cbind(x, lagx = lag(x, -1))
> y

   x lagx
1960 Q1 0.71 NA
1960 Q2 0.63 0.71
1960 Q3 0.85 0.63
1960 Q4 0.44 0.85
1961 Q1 0.61 0.44
1961 Q2 NA 0.61
\end{verbatim}
• For example, if you wanted to include the prior quarter’s earnings into the regression for log earnings:

```r
jjdata = cbind(ljj = log(jj),
              tjj = time(jj),
              cjj = cycle(jj),
              lagljj = lag(log(jj), -1))
jjdata = data.frame(jjdata)
jjdata$Q = factor(jjdata$cjj)
summary(lm(ljj ~ tjj + Q + lagljj, data = jjdata))
```

• If you just include `lag(log(jj), -1)` in the original model, you get a very different result (try it!).