MTH 4230 Lab 8

Due Wed., Apr. 29

1 Part A: Time Series Analysis

1.1 Lake Huron Data Set

R has several built-in data sets. One of them is called `LakeHuron`. It contains the levels of Lake Huron for the years 1875-1972.

1. Type:

   ```
   LakeHuron
   ```

   to view the data, and type:

   ```
   ? LakeHuron
   ```

   to view its help file. `LakeHuron` is a *time series*, or `"ts"` object. Type:

   ```
   class(LakeHuron)
   ```

   to verify this. The function `time()` will extract the "time" component from a *ts* object. Type:

   ```
   time(LakeHuron)
   ```

   to see the years of the `LakeHuron` time series.

2. When a *ts* object like `LakeHuron` is passed to `plot()`, it makes a ***time-series*** plot of the data. Make the plot by typing:

   ```
   plot(LakeHuron)
   ```

3. First, fit a linear regression model to the data using the usual least squares method:

   ```
   my.reg <- lm(LakeHuron ~ time(LakeHuron))
   ```

   and then use `summary()` to look at the results.

4. Now add the fitted regression line to the time-series plot in the usual manner:

```
  abline(my.reg)
```

5. We can perform a ***Durbin-Watson*** test to decide if the the serial correlation in the data is statistically significant. We'll need the `dwtest()` function, which is in the `"lmtest"` package.

After downloading and installing the `"lmtest"` package:

```
install.packages("lmtest")
```

load it into the current R session by typing:

```
library(lmtest)
```

Now the `dwtest()` function should be available for use. It takes an `"lm"` object as its argument, and carries out a **Durbin-Watson** test on the residuals. Carry out the test, for example by typing:

```
dwtest(my.reg)
```

6. We can carry out a ***Cochrane-Orcutt*** procedure for fitting a model to time series data with autocorrelated errors by estimating the AR(1) correlation parameter ρ, fitting a regression model to $Y'_t = Y_t - rY_{t-1}$ and $X'_t = X_t - rX_{t-1}$, where r is the estimate of ρ, and then "backtransforming" the parameter estimates.

We'll need the `cochrane.orcutt()` function, which is in the `"orcutt"` package.

After installing and loading the `"orcutt"` package:

```
install.packages("orcutt")

library("orcutt")
```

carry out the **Cochrane-Orcutt** procedure by passing `cochrane.orcutt()` your `"lm"` object `my.reg`:

```
my.cochrane <- cochrane.orcutt(my.reg)

my.cochrane
```

Then look at the summary by typing:

```
summary(my.cochrane)
```