Lab 2 - Examining Temporal Dependencies

Benjamin M. Taylor, Kevin Hayes

January 20, 2023

1 Autocorrelation in Time Series Data

1.1 Luteinizing Hormone in Blood Samples

Diggle (1990) reports on a regular time series giving the Luteinizing hormone in blood samples taken from a healthy woman at 10 mins intervals over an eight-hour period. The data were collected from the early follicular phase of the menstrual cycle and can be obtained from the R time series object `lh`.

First, plot the data.

[ ]:
```
options(repr.plot.width=15, repr.plot.height=15) # makes plots bigger in the␣
↪webpage
```

[ ]:
```
plot(lh,type="o")
```

Look at the correlation between current values and previous values (`lh` at lag 1).

[ ]:
```
plot(x=lag(lh,k=1),y=lh,xy.lines=FALSE,xy.labels=FALSE)
title(main=paste("r =",round(cor(lh[-1],lh[-48]),digits=3)))
```

Look at the correlation between current values and values before last (`lh` at lag 2).

[ ]:
```
plot(x=lag(lh,k=2),y=lh,xy.lines=FALSE,xy.labels=FALSE)
title(main=paste("r =",round(cor(lh[-(1:2)],lh[-(47:48)]),digits=3) ))
```

Let's do something a bit more adventurous, and repeat the previous exercise for a variety of lags.

[ ]:
```
layout(mat=matrix(seq(6),ncol=3,byrow=TRUE))
for(i in 1:6){
    plot(x=lag(lh,k=i),y=lh,xy.lines=FALSE,xy.labels=FALSE,ylab=paste("lh␣
↪Series at lag",i))
    title(main=paste("r=",round(cor(lh[-(1:i)],lh[-((49-i):48)]),digits=3)))
}
```

It would be nice if there was a mechanism to harvest the correlations calculated in the previous plot and show them simultaneously on a single plot, for a range of time lags. The built in R function `acf()` calculates the sample autocorrelation function (a.k.a. correlogram) for a time series and is a key exploratory tool for investigating the autocorrelation structures in time series data. Execute the R command:

[ ]:
```
acf(lh)
```

The acf plot is consistent with the previous calculations, as can be verified as follows:

[ ]:
```
acf(lh)
for (i in 1:6){
    r = round(cor(lh[-(1:i)],lh[-((49-i):48)]), digits=3)
    points(i, r, pch=16, col="blue")
}
```

For this time series the magnitude of the sample autocorrelation at lag 1 is strong, but this inter dependency disappears immediately for higher lags. The horizontal blue bars act as defecto control limits for the sample autocorrelations and are based on the behaviour we would expect to observe if indeed we were examining a white noise process.

1.2 Sales of Blue Jeans

The R object `BJsales` reports the sales of blue jeans over 150 time periods (Box & Jenkins, 1976). Plot the data. Plot the acf of the data. How does the autocorrelation structure of these data contrast with autocorrelation structure of the `lh` series?

[ ]:
```

```

1.3 Gaussian Noise

Consider a time series of pure i.i.d Gaussian (identically independently distributed normal) errors.

[ ]:
```
w=as.ts(rnorm(100))
```

Consider a time series of white noise errors.

[ ]:
```
w=as.ts(rt(100, df=5))
```

Plot the data. Plot the acf of the data. How does the autocorrelation structure of these data contrast with previous series considered?

[ ]:
```

```

2 Partial Correlation

Definition (sort of)

For a time series, the partial autocorrelation τ_k between y_t and y_{t-k} is defined as the conditional correlation between y_t and y_{t-k}, conditional on the separating set of observations $y_{t-k+1}, \cdots, y_{t-1}$.

The 1st order partial autocorrelation $\tau 1$ will be defined to equal the 1st order autocorrelation ρ_1

The 2nd order (lag) partial autocorrelation is

$$\tau_2 = \frac{\text{Cov}(y_t, y_{t-2}|y_{t-1})}{\sqrt{\text{Var}(y_t|y_{t-1})\text{Var}(y_{t-2}|y_{t-1})}}$$

This is the correlation between values two time periods apart conditional on knowledge of the value in between.

The 3rd order (lag) partial autocorrelation is

$$\tau_3 = \frac{\text{Cov}(y_t, y_{t-3}|y_{t-1}, y_{t-2})}{\sqrt{\text{Var}(y_t|y_{t-1}, y_{t-2})\text{Var}(y_{t-3}|y_{t-1}, y_{t-2})}}$$

and, so on, for any lag.

In the context of time series, τ_k is conceptually measuring the extra predictive information about y_t contained in y_{t-k} that is not already contained in the separating set of observations $y_{t-k+1}, \cdots, y_{t-1}$. Matrix manipulations are used to determine estimates $\hat{\tau}_k$ of the partial autocorrelations at higher lags.

The Partial Autocorrelation Function in R

Work your way through the following R commands. For each time series data set contrast the patterns exhibited by the acf and pacf plots.

[ ]:
```
acf(lh, lag.max = 24)
pacf(lh, lag.max = 24)
```

[ ]:
```
acf(BJsales)
pacf(BJsales)
```

[ ]:
```
ldax=log(EuStockMarkets[,"DAX"])
acf(ldax)
pacf(ldax)
```

[ ]:
```
ldax.d1=diff(ldax)
acf(ldax.d1)
```

2.1 Model Identification

Later in the course we will define a versatile family of stochastic models for time series modelling and forecasting. Specifically, autoregressive integrated moving average models (ARIMA). This family of models also have their seasonal counterparts (S-ARIMA) that can be used to model seasonal effects. The question will arise as regarding which model to use for a given data set? We will show later that the ARIMA models leave their footprints in the acf and pacf plots and we reverse engineer – that is we use the acf plot and pacf plot to identify the most plausible ARIMA model for the data under consideration.

3 Differencing

In the lectures, and again in the `EuStockMarkets` example above, we saw that differencing successive terms reduces a random walk process to independent white noise process. We can take this strategy a step further and deploy it to annihilate trend and seasonal effects in real time series data sets.

3.1 Differencing (Stochastic Trend)

Consider the (non-stationary) random walk time-series process $X_t = X_{t-1} + e_t$. Differencing the series generates a new time-series process $Y_t \equiv X_t - X_{t-1} = e_t$, which is stationary. This annihilates the slow meandering trend within the process X_t, often referred to as stochastic trend.

3.2 Differencing (Linear Trend)

Consider the non-stationary process

$$Y_t = \underbrace{\alpha + \beta t}_{\mu_t} + e_t.$$

The once-differenced series U_t defined by

$$\begin{aligned} U_t &= Y_t - Y_{t-1} \\ &= \underbrace{\alpha + \beta t - [\alpha + \beta(t-1)]}_{=\beta} + \underbrace{e_t - e_{t-1}}_{=\eta_t} \end{aligned}$$

satisfies $\mathrm{E}(U_t) = \beta, \ \forall\ t$, and therefore is mean-stationary.

3.3 Differencing (Quadratic Trend)

Consider the non-stationary process

$$Y_t = \underbrace{\beta_2 t^2 + \beta_1 t + \beta_0}_{\mu_t} + e_t.$$

Define the twice-differenced series

$$\begin{aligned} U_t &= [Y_t - Y_{t-1}] - [Y_{t-1} - Y_{t-2}] \\ &= [(\beta_2 t^2 + \beta_1 t + 1) - (\beta_2 (t-1)^2 + \beta_1 (t-1) + \beta_0)] \\ &\qquad [(\beta_2 (t-1)^2 + \beta_1 (t-1) + 1) \\ &\qquad\qquad -(\beta_2 (t-2)^2 + \beta_1 (t-2) + \beta_0)] \\ &\qquad\qquad\quad + \underbrace{e_t - e_{t-1} - (e_{t-1} - e_{t-2})}_{=\eta_t} \\ &= \beta_2 + \eta_t \end{aligned}$$

The series U_t is mean-stationary and satisfies $\mathrm{E}(U_t) = \beta_2 \; \forall \; t$.

Seasonal Differencing

The seasonal difference operator is

$$\nabla^s Y_t = Y_t - Y_{t-s}$$

where s is the period of the seasonal cycle. Seasonal differencing will remove seasonality in the same way that ordinary differencing will remove a polynomial trend.

Atmospheric CO2 revisited

Execute the following R commands...

[ ]:
```
co2.d1 = diff(co2)
co2.D1 = diff(co2, lag = 12)
co2.d1D1 = diff(co2.d1, lag = 12)
```

[ ]:
```
par(mfrow=c(1,2))

plot(co2)
acf(co2)
```

[ ]:
```
par(mfrow=c(1,2))

plot(co2.d1)
acf(co2.d1)
```

[ ]:
```
par(mfrow=c(1,2))

plot(co2.D1)
acf(co2.D1)
```

[ ]:
```
par(mfrow=c(1,2))

plot(co2.d1D1)
acf(co2.d1D1)
```

After differencing at lag 1 (to remove trend), then at lag 12 (to remove seasonal effects), the `co2` series appears stationary. That is, the series

$\nabla\nabla^{12}Y_t$, or equivalently the series $(1 - B)(1 - B^{12})Y_t$, appears stationary. We could now look to fit a model to $\nabla\nabla^{12}Y_t$.

3.4 Exercise

Reduce the AirPassengers time series to a stationary series.

[ ]:
```

```

3.5 Suggested Homework

Visit the web site [https://nsidc.org/cryosphere/sotc/sea_ice.html] Links to an external site.. Download a data set, any data set (e.g. sea ice), load it into R. Write a report of maximum one page in length (including graphs) describing your data set and the steps taken to reduce it to stationarity.

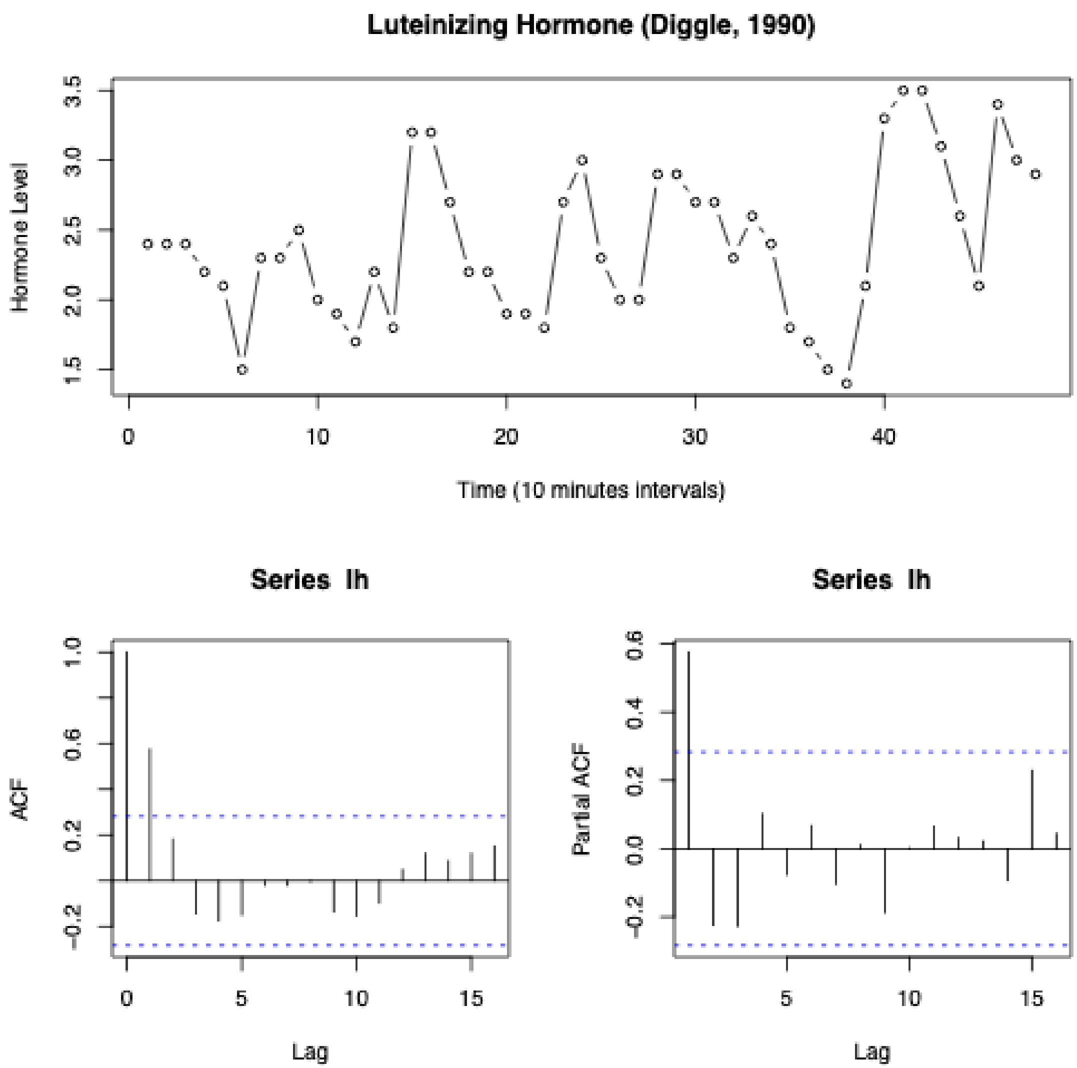
[ ]:

```

```

4 A Trio of Boilerplate Time Series Plots

Write R code to reproduce your own version of the following plot. Hint: use the function layout( ). Feel free to improve the plot if you want to. Upload your plot to Canvas.



As an additional exercise (don't upload this), you could think about writing an R function to take as input a time series object and output these three plots. You'll find this function useful during the course. You could include as optional arguments a title, and axes labels for the main plot.