Lab 5 - Case Studies 1

Benjamin M. Taylor, Kevin Hayes

January 20, 2023

1 Luteinizing Hormone in Blood Samples

First things first. Set up options:

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

[ ]:
```
options(digits = 5, show.signif.stars = FALSE)
```

Diggle (1990) Time Series: A Biostatistical Introduction. Oxford. Diggle (1990) reports on a regular time series giving the luteinizing hormone in blood samples at 10 mins intervals from a human female, 48 samples. This time series is pre-loaded in R can can be viewed by typing `lh` at the R console

[ ]:
```
######################################
# code 01: look at the data

lh
```

[ ]:
```
######################################
# code 02: plot the data. Think: AR or MA(1)? Why?

layout(mat=matrix(c(1,1,2,3),byrow=TRUE,ncol=2))

plot(lh, type = "o", xlab = "Time (10 minutes intervals)", ylab= "Hormone Level")

acf(lh) ; pacf(lh)
```

[ ]:
```
######################################
# code 03: fit an AR(1) model

lh.ar1 = ar(lh, aic = FALSE, order.max = 1)

lh.ar1
```

[ ]:
```
names(lh.ar1)
```

[ ]:
```
lh.ar1$method
```

[ ]:
```
########################################
# code 04: exploting, we see that for
# an AR(1) phi.hat is just:

acf(lh)$acf
```

[ ]:
```
round(acf(lh)$acf[2], 4)
```

[ ]:
```
########################################
# code 05: criticise fitted AR(1)
# Think: are you satisfied with the behaviour of the residuals?

resids = c(lh.ar1$resid)

try(acf(resids)) #doesn't work, why not and how to get round this?
```

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

acf(resids[-1])

pacf(resids[-1])

qqnorm(resids)

cpgram(resids)
```

[ ]:
```
########################################
# code 06: criticise fitted AR(1)
# Using the arima( ) function in R...

try(tsdiag(lh.ar1)) # doesn't work!   TRY...
```

[ ]:
```
lh.ar1 = arima(lh, order = c(1,0,0))
```

[ ]:
```
lh.ar1 # how does the output differ???
```

[ ]:
```
tsdiag(lh.ar1)
```

[ ]:
```
cpgram(residuals(lh.ar1))
```

[ ]:
```
########################################
# code 07: use arima function to investigate
# AR(2), AR(3), and AR(4) models. e.g.

lh.ar2 = arima(lh, order = c(2,0,0))
```

```
lh.ar2 # how does the output differ???
```

[ ]:
```
tsdiag(lh.ar2)
```

[ ]:
```
cpgram(residuals(lh.ar2))
```

[ ]:
```
######################################
# code 08: explore AR(p) p=1,...,8
# The "aic" option in ar( ) is a logical flag.
# If TRUE then the Akaike Information Criterion
# is used to choose the order of the AR model.
# If FALSE, the model of order "order.max" is fitted.
# For example, "order.max = 8" fits an AR(8) model.
# Also collects the AICs for AR(1),AR(2),...,AR(8) models.
# Think: which model? (think very carefully!)

lh.ARmods = ar(lh, order.max=8, aic=FALSE)

plot(seq(0,8), lh.ARmods$aic, type="b", pch=16, col="blue",
     main = "AIC for LH", xlab="Order of AR model", ylab="log[ AIC / AIC of␣
 ↪AR(3) ]")
```

[ ]:
```
######################################
# code 09: Summary of AR(3) model..
# Compare the output...

lh.ar3 = ar(lh, order.max=3)

lh.ar3
```

[ ]:
```
#   y[t] = 0.6534*y[t-1] -0.0636*y[t-2] -0.2269*y[t-3] + e[t]
#   with var{ e[t] } = sigma^2 = 0.1959

lh.ar3 = arima(lh, order = c(3,0,0))

lh.ar3 #   what has changed?
```

[ ]:
```
lh.ar3 = arima(lh, order = c(3,0,0), method = "CSS-ML")

lh.ar3 #   what has changed?
```

[ ]:
```
lh.ar3 = arima(lh, order = c(3,0,0), method = "ML")

lh.ar3 #   what has changed?
```

```
[ ]: lh.ar3 = arima(lh, order = c(3,0,0), method = "CSS")

     lh.ar3 # what has changed?
```

```
[ ]: #
     ######################################
     # code 10: Best ARMA subset regression
     # Try this at home. You will need the TSA package.
     # Think: which model? (think carefully!)

     require(TSA)

     lh.mods = armasubsets(y = lh, nar=4, nma=4)
```

```
[ ]: plot(lh.mods)
```

2 Level of Lake Huron

```
[ ]: # PREAMBLE

     options(digits = 5, show.signif.stars = FALSE)
```

Brockwell & Davis (1991). Time Series and Forecasting Methods. The R object "LakeHuron" describes Annual measurements of the level, in feet, of Lake Huron 1875-1972. The time series has 98 values.

```
[ ]: ######################################
     # code 01: look at the data
     # Think: AR or MA? What order?

     LakeHuron
```

```
[ ]: layout(mat=matrix(c(1,1,2,3),byrow=TRUE,ncol=2))

     plot(LakeHuron, type = "o", ylab = "level (feet)")

     title(main= "Annual measurements of the level of Lake Huron 1875-1972.")

     acf(LakeHuron)

     pacf(LakeHuron)
```

```
[ ]: ######################################
     # code 02: fit some starting models!
     # Think: Which model? Why??
```

```
lake1 = arima(LakeHuron, order = c(1,0,0))

tsdiag(lake1)
```

[ ]:
```
lake2 = arima(LakeHuron, order = c(2,0,0))

tsdiag(lake2)
```

[ ]:
```
lake3 = arima(LakeHuron, order = c(3,0,0))

tsdiag(lake3)
```

[ ]:
```
lake4 = arima(LakeHuron, order = c(1,0,1))

tsdiag(lake4)
```

[ ]:
```
lake1$aic
lake2$aic
lake3$aic
lake4$aic
```

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

cpgram(residuals(lake1))
cpgram(residuals(lake2))
cpgram(residuals(lake3))
cpgram(residuals(lake4))
```

[ ]:
```
######################################
# code 03: Select a working model

lake.fit = lake4
```

[ ]:
```
######################################
# code 04: produce forecasts...

lake.pred = predict(lake.fit, n.ahead = 12)
```

[ ]:
```
names(lake.pred)
```

[ ]:
```
lake.pred
```

[ ]:
```
# Here we have fitted an ARIMA(1,0,1) model to the
# Lake Huron data. We have predicted the levels of
# Lake Huron for the next 12 years. In this case,
# "Lake.pred" is a list containing two entries, the
# predicted values "Lake.pred$pred" and the standard
```

```
# errors of the prediction "Lake.pred$se"}. Using a
# rule of thumb ``prediction +/- 2*standard error,"
# an approximate 95% prediction interval can be
# calculated for these future values.

#######################################
# code 05: plot forecasts...

plot(LakeHuron, xlim = c(1875, 2000), ylim = c(575,584),main="Lake Hurron␣
 ↪levels and predicted values")

# First the levels of Lake Huron are plotted.
# To leave space for adding the predicted values,
# the x-axis is set from 1875 to 2000 with
# "xlim=c(1875,2000)"; the use of "ylim" is purely
# for cosmetic purposes here:
# Execute the following code line-by-line...

lake.pred = predict(lake.fit, n.ahead = 28)

lines(lake.pred$pred,col="red")

lines(lake.pred$pred+2*lake.pred$se,col="red",lty=3)

lines(lake.pred$pred-2*lake.pred$se,col="red",lty=3)

abline(h = mean(LakeHuron), col="grey", lty = 2)
```

```
[ ]: # The final command draws a horizontal line at
# the mean of the Lake Huron values. What does
# this lead you to conclude you about the
# potential for long term forecasting?

#######################################
# code 06: plot forecasts using one line of R code :) but less colour :(

plot(lake.fit, n.ahead = 12)
```

3 Nile River Flow

Balke (1993) Detecting level shifts in time series. J Business and Economic Stats 11, 8192.

Cobb (1978) The problem of the Nile: conditional solution to a change-point problem. Biometrika 65, 24351.

Measurements of the annual flow of the river Nile at Ashwan 18711970. A time series of length 100.

[ ]:
```r
########################################
# code 01: look at the data

layout(mat=matrix(c(1,1,2,3),byrow=TRUE,ncol=2))

plot(Nile, type = "o")

lines(lowess(Nile, f=1/3), col="red", lwd=2)

acf(Nile)
pacf(Nile)
```

[ ]:
```r
########################################
# code 02: look through possible AR models

ar.aic = ar(Nile)

ar.aic # selects order 2
```

[ ]:
```r
#how did it decide on AR(2)? try...

plot(seq(ar.aic$aic), ar.aic$aic,type = "b", ylab = "AIC", xlab = "order")
```

[ ]:
```r
########################################
# code 03: fit a preliminary model

myfit = arima(Nile, c(2, 0, 0))

# Can you explain why the pvalues of the Ljung-Box
# Portmanteau statistic are approaching statistical
# significance at higher lags?
# Examine the data further. Can you do better??
```

[ ]:
```r
########################################
# code 99: Best ARMA subset regression
# Try this at home. You will need the TSA package.
# Think: which model? (think carefully!)
require(TSA)

lh.mods = armasubsets(y = Nile, nar=4, nma=4)

plot(lh.mods)
```

4 Monthly Air Temperatures at Recife

Average Monthly Air Temperatures at Recife in Brazil, over a 10 year period. The data are here `recifetemps.txt`

Download `recifetemps.txt`

[ ]:
```
########################################
# code 01: read in the data

recife = ts(data = scan("data/recifetemps.txt"), start = 1953, frequency = 12)
```

[ ]:
```
########################################
# code 02: look at the data
# Think: in words, describe the trend and seasonal patterns in the data.

recife
```

[ ]:
```
plot(recife, ylab="air temperature")

# and tag the months...
require(miscFuncs)
tag = function(i){
    return(substr(monthnames(),1,1)[i])
}

points(y= recife, x=time(recife), pch = tag(as.vector(cycle(recife))), cex=0.7)
```

[ ]:
```
########################################
# code 03: look at the data again

plot(stl(recife, s.window=11))
```

[ ]:
```
# Rethink: in words, describe the trend and seasonal patterns in the data.

########################################
# code 04: differencing

y = diff(recife)

y = diff(y)

y = diff(y, lag = 12)

layout(mat=matrix(c(1,1,2,3),byrow=TRUE,ncol=2))

plot(y, main= expression((1-B)^{2}*(1-B^{12})*X[t]))

points(y, x=time(y), pch = tag(as.vector(cycle(recife))), cex=0.7)

acf(y, lag.max = 30); pacf(y, lag.max = 30)
```

```
[ ]: ########################################
     # code 05: fit an ARIMA(p,d,q)(P,D,Q)

     myfit = arima(recife, order=c(3,2,0), seasonal = c(2,1,0))

     myfit
```

```
[ ]: resids = residuals(myfit)

     par(mfrow=c(1,2))

     acf(resids, lag.max = 36)

     pacf(resids, lag.max = 36)
     # If working in R GUI, kill plot before proceeding
```

```
[ ]: cpgram(resids)

     ########################################
     # code 06: model criticism
     # Think: are you satisfied with the behaviour of the residuals?
     # Think: can you produce a better model?
     #
```

5 Government Securities

Yield (%) on British short term government securities in successive months from about 1950 to about 1971.

```
[ ]: ########################################
     # code 01: load the data

     yield = ts(start=1950, frequency = 12, data = c(
     2.22,2.23,2.22,2.20,2.09,1.97,2.03,1.98,1.94,1.79,1.74,1.86,
     1.78,1.72,1.79,1.82,1.89,1.99,1.89,1.83,1.71,1.70,1.97,2.21,
     2.36,2.41,2.92,3.15,3.26,3.51,3.48,3.16,3.01,2.97,2.88,2.91,
     3.45,3.29,3.17,3.09,3.02,2.99,2.97,2.94,2.84,2.85,2.86,2.89,
     2.93,2.93,2.87,2.82,2.63,2.33,2.22,2.15,2.28,2.28,2.06,2.54,
     2.29,2.66,3.03,3.17,3.83,3.99,4.11,4.51,4.66,4.37,4.45,4.58,
     4.58,4.76,4.89,4.65,4.51,4.65,4.52,4.52,4.57,4.65,4.74,5.10,
     5.00,4.74,4.79,4.83,4.80,4.83,4.77,4.80,5.38,6.18,6.02,5.91,
     5.66,5.42,5.06,4.70,4.73,4.64,4.62,4.48,4.43,4.33,4.32,4.30,
     4.26,4.02,4.06,4.08,4.09,4.14,4.15,4.20,4.30,4.26,4.15,4.27,
     4.69,4.72,4.92,5.10,5.20,5.56,6.08,6.13,6.09,5.99,5.58,5.59,
     5.42,5.30,5.44,5.32,5.21,5.47,5.96,6.50,6.48,6.00,5.83,5.91,
     5.98,5.91,5.64,5.49,5.43,5.33,5.22,5.03,4.74,4.55,4.68,4.53,
     4.67,4.81,4.98,5.00,4.94,4.84,4.76,4.67,4.51,4.42,4.53,4.70,
```

```
4.75,4.90,5.06,4.99,4.96,5.03,5.22,5.47,5.45,5.48,5.57,6.33,
6.67,6.52,6.60,6.78,6.79,6.83,6.91,6.93,6.65,6.53,6.50,6.69,
6.58,6.42,6.79,6.82,6.76,6.88,7.22,7.41,7.27,7.03,7.09,7.18,
6.69,6.50,6.46,6.35,6.31,6.41,6.60,6.57,6.59,6.80,7.16,7.51,
7.52,7.40,7.48,7.42,7.53,7.75,7.80,7.63,7.51,7.49,7.64,7.92,
8.10,8.18,8.52,8.56,9.00,9.34,9.04,9.08,9.14,8.99,8.96,8.86,
8.79,8.62,8.29,8.05,8.00,7.89,7.48,7.31,7.42,7.51,7.71,7.99))

yield
```

[ ]:
```
########################################
# code 02: plot the data

layout(mat=matrix(c(1,1,2,3),byrow=TRUE,ncol=2))

plot(yield, type = "o", ylab= "Yield")

acf(yield)

pacf(yield)

    # [1] There is no discernible seasonal variation.
    # [2] There is a marked trend, from about 2% at
    # the start of the series to about 7% at the end.
    # [3] The trend is by no means regular, and to
    # assume and to fit a straight line would be a gross
    # over simplification. The use of a trend-and-season
    # model, or linear regression seems inappropriate.
    # [4] These observations are confirmed by the
    # spectrum, ACF, and PACF.
```

[ ]:
```
cpgram(yield)

# suggests a Random Walk Model.
# The non-stationarity indicates that
# some sort of differencing is required.
# Kill the last plot. Then...
```

[ ]:
```
########################################
# code 03: difference the data, and plot...

yield.diff = diff(yield)

layout(mat=matrix(c(1,1,2,3),byrow=TRUE,ncol=2))

plot(yield.diff, type = "o", ylab = "Difference",
```

```
main= "Month-to-Month Changes in Yield")

acf(yield.diff)

pacf(yield)
```

[ ]:
```
cpgram(yield.diff)

# These plots tell us quite a lot.
# First, the differenced series is
# now stationary, and no additional
# differencng is required. The fact
# that there is only one significant
# coefficient at lag 1 indicates an
# ARIMA(0,1,1) model.
```

[ ]:
```
########################################
# code 04: Fit and contrast some ARIMA fits...

fit1 = arima(yield,order=c(0,1,1))

tsdiag(fit1)

cpgram(residuals(fit1))
```

[ ]:
```
fit2 = arima(yield,order=c(0,1,2))

tsdiag(fit2)

cpgram(residuals(fit2))
```

[ ]:
```
fit2 = arima(yield,order=c(0,1,2))

tsdiag(fit2)

cpgram(residuals(fit2))
```

[ ]:
```
fit3 = arima(yield,order=c(0,1,3))

tsdiag(fit3)

cpgram(residuals(fit3))
```

[ ]:
```
########################################
# code 05: plot forecasts...

plot(yield, xlim = c(1950, 1972), ylim = c(0,10),
```

```
    main="Yield (%) on British short term government securities")

yield.pred = predict(fit3, n.ahead = 20)

lines(yield.pred$pred,col="red")

lines(yield.pred$pred+2* yield.pred$se,col="red",lty=3)

lines(yield.pred$pred-2* yield.pred$se,col="red",lty=3)

# What do you conclude about the potential for long term forecasting here?
```

6 WWW Usage

A time series of the numbers of users connected to the Internet through a server every minute.

[ ]:
```
######################################
# code 01: look at the data

WWWusage
```

[ ]:
```
######################################
# code 02: plot the data. Think: AR or MA(1)? Why?

layout(mat=matrix(c(1,1,2,3),byrow=TRUE,ncol=2))

plot(WWWusage, type = "o", main = "Internet Usage per Minute",

xlab = "Time (minute intervals)", ylab= "Number of Users")

acf(WWWusage)

pacf(WWWusage)
```

[ ]:
```
######################################
# code 03: Dicky-Fuller t=Test
# If "tseries" package is available, then try:

require(tseries)

adf.test(WWWusage)
```

[ ]:
```
######################################
# code 04: take differences

work <- diff(WWWusage)
```

```r
par(mfrow = c(2, 1))

plot(WWWusage)

plot(work)
```

[ ]:
```r
#######################################
# code 05: Model Search

aics <- matrix(, 6, 6, dimnames = list(p = 0:5, q = 0:5))

for(q in 1:5){
    aics[1, 1+q] <- arima(WWWusage, c(0, 1, q),optim.control = list(maxit =␣
 ↪500))$aic
}

for(p in 1:5){
   for(q in 0:5){
       aics[1+p, 1+q] <- arima(WWWusage, c(p, 1, q),optim.control = list(maxit␣
 ↪= 500))$aic
    }
}

round(aics - min(aics, na.rm = TRUE), 2)
```

[ ]:
```r
#
#######################################
# code 06: Investigate the following models.
# ARIMA(1,1,0),  ARIMA(2,1,0), ARIMA(3,1,0), ARIMA(1,1,1),
```