Homework 2 - Answer Key

Mark Scheuerell

Here are the answers for the homework problems from the second week of class based on the material in the introductory time series in R. All of the questions were based on the following scenario:

You have been asked by a colleauge to help analyze some time series data she collected as part of an experiment on the effects of light and nutrients on the population dynamics of phytoplankton. Specifically, after controlling for differences in light and temperature, she wants to know if the natural log of population density can be modeled with some form of ARMA(p,q) model. The data are expressed as the number of cells per milliliter recorded every hour for one week beginning at 8:00 AM on December 1, 2014; you can find them here:

```
# get phytoplankton data
pp <- "http://faculty.washington.edu/scheuerl/phytoDat.txt"
pDat <- read.table(pp)
```

Question 1

Convert pDat, which is a data.frame object, into a ts object.

Answer

We need the `ts()` function to do this, but the trick is getting the right starting day to pass in. Using the bit of code provided, here's how to do it:

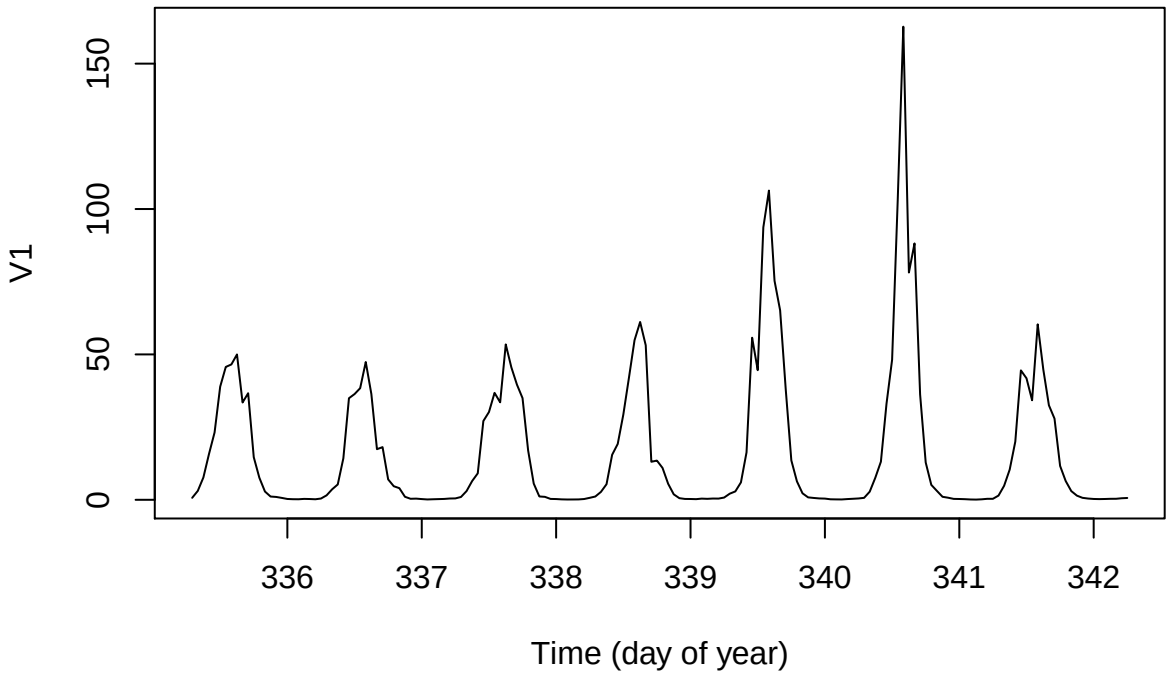
```
# what day of 2014 is Dec 1st?
dBegin <- as.Date("2014-12-01")
dayOfYear <- (dBegin - as.Date("2014-01-01") + 1)
# ts object
pDat <- ts(pDat, freq=24, start=c(dayOfYear,8))
```

Question 2

Plot the time series of phytoplankton density and provide a brief description of any notable features.

Answer

```
# plot the ts
plot.ts(pDat, xlab="Time (day of year)")
```



The time series appear non-stationary in that they have an obvious seasonal signal corresponding to each 24-hr period. The data also appear to be non-Gaussian (ie, they occur on the positive, real interval).

Question 3

Although you do not have the actual measurements for the specific temperature and light regimes used in the experiment, you have been informed that they follow a regular light/dark period with accompanying warm/cool temperatures. Thus, estimating a fixed seasonal effect is justifiable. Also, the instrumentation is precise enough to preclude any systematic change in measurements over time (i.e., you can assume m_t = 0 for all t). Obtain the time series of the estimated log-density of phytoplankton absent any hourly effects caused by variation in temperature or light.

Answer

The key here is to recognize that what we're really after here are the random errors (residuals) $\{e_t\}$ from the classical decomposition model $x_t = m_t + s_t + e_t$.

Let's begin by calculating the log of phytoplankton density, which is what we're ultimately after.

```
lDat <- log(pDat)
```

Next, we can assume that the trend m_t is zero for all t, and therefore we can estimate the seasonal effect from our decomposition model $x_t = m_t + s_t + e_t$ by rearranging the terms as follows.

$$
\begin{aligned}
s_t &= x_t - m_t + e_t \\
\hat{s}_t &= x_t - m_t \\
\hat{s}_t &= x_t - 0 \\
\hat{s}_t &= x_t
\end{aligned}
$$

This means that our estimate of the seasonal effects (including the random errors) equals the data themselves (lDat), which saves us the first step of having to subtracting off the trend. Thus, we can use the code in Section 1.2.2 of the lab handout to estimate the mean seasonal effect for each hour of the day.

```
# length of ts
ll <- length(lDat)
# frequency (ie, 24)
ff <- frequency(lDat)
# number of periods (days); %/% is integer division
periods <- ll %/% ff
# index of cumulative hours
index <- seq(1,ll,by=ff) - 1
# get mean by hour
mm <- numeric(ff)
for(i in 1:ff) {
  mm[i] <- mean(lDat[index+i], na.rm=TRUE)
}
# subtract mean to make overall mean=0
mm <- mm - mean(mm)
# create entire ts of seasonal (hourly) values
sDat <- ts(rep(mm, periods+1)[seq(ll)],freq=24,start=start(lDat))
```

Finally, we can use our estimate of $\{s_t\}$ that we just generated (sDat) to obtain the log-density of phytoplankton after controlling for the seasonal (hourly) effect. Again, from our decomposition model $x_t = m_t + s_t + e_t$, we can get

$$e_t = x_t - m_t - s_t$$
$$e_t = x_t - 0 - s_t$$
$$e_t = x_t - s_t$$

```
# estimate of log-density
dens <- lDat - sDat
```

Question 4

Use diagnostic tools to identify the possible order(s) of ARMA model(s) that most likely describes the log of population density for this particular experiment. Note that at this point you should be focusing your analysis on the results obtained in Question 3.

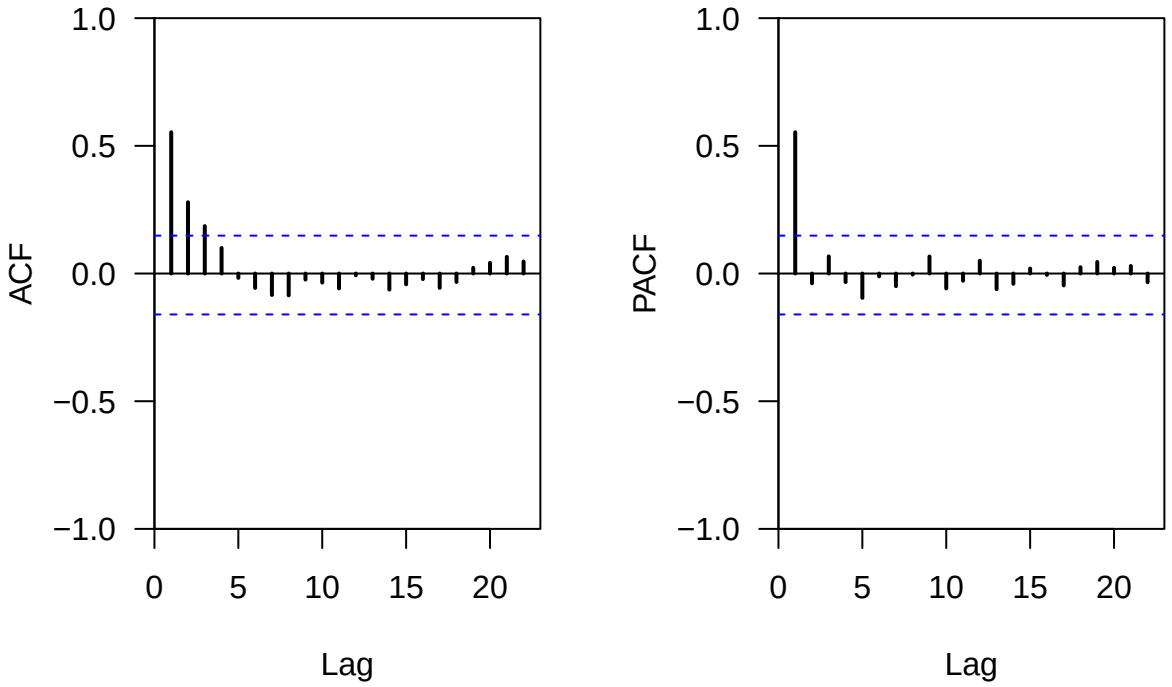
Answer

The ACF and PACF are obvious choices here. Let's use the better plot functions we defined in Secs 1.4.1 & 1.4.2 of the lab handout.

```
# better ACF plot
plot.acf <- function(ACFobj) {
  rr <- ACFobj$acf[-1]
  kk <- length(rr)
  nn <- ACFobj$n.used
  plot(seq(kk),rr,type="h",lwd=2,yaxs="i",xaxs="i",
       ylim=c(floor(min(rr)),1),xlim=c(0,kk+1),
       xlab="Lag",ylab="ACF",las=1)
  abline(h=-1/nn+c(-2,2)/sqrt(nn),lty="dashed",col="blue")
```

```
  abline(h=0)
}
# better PACF plot
plot.pacf <- function(PACFobj) {
  rr <- PACFobj$acf
  kk <- length(rr)
  nn <- PACFobj$n.used
  plot(seq(kk),rr,type="h",lwd=2,yaxs="i",xaxs="i",
       ylim=c(floor(min(rr)),1),xlim=c(0,kk+1),
       xlab="Lag",ylab="PACF",las=1)
  abline(h=-1/nn+c(-2,2)/sqrt(nn),lty="dashed",col="blue")
  abline(h=0)
}
```

```
# plots of ACF & PACF
par(mfrow=c(1,2))
plot.acf(acf(dens, plot=FALSE))
plot.pacf(pacf(dens, plot=FALSE))
```



It looks like the ACF (left plot) tails off slowly, but the PACF (right plot) tails off after lag=1, which suggests that an AR(1) model might be most appropriate.

Question 5

Use some form of search to identify what form of ARMA(p,q) model best describes the log of population density for this particular experiment. Use what you learned in Question 4 to inform possible orders of p and q.

Answer

The first option would be a manual search over various orders of p and q in ARMA models. In Question 4 we were led to believe that an AR(1) model might be most appropriate, so let's try orders from 0-2 for both p and q in case we've made a mistake.

```
# range of orders for p & q
pLo <- qLo <- 0
pHi <- qHi <- 2
# empty list to store model fits
ARMA.res <- list()
# set counter
cc <- 1
# loop over AR
for(p in pLo:pHi) {
  # loop over MA
  for(q in qLo:qHi) {
    ARMA.res[[cc]] <- arima(x=dens,order=c(p,0,q))
    cc <- cc + 1
  }
}
# get AIC values for model evaluation
ARMA.AIC <- sapply(ARMA.res,function(x) x$aic)
# model with lowest AIC is the best
ARMA.res[[which(ARMA.AIC==min(ARMA.AIC))]]
```

```
##
## Call:
## arima(x = dens, order = c(p, 0, q))
##
## Coefficients:
##          ar1  intercept
##       0.5576     1.1474
## s.e.  0.0641     0.0505
##
## sigma^2 estimated as 0.08493:  log likelihood = -31.43,  aic = 68.85
```

The results support our intuition from Question 4 in that an AR(1) model seems most parsimonious. Let's see if an auto-search yields the same answer.

```
library(forecast)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: timeDate
```

```
## This is forecast 7.3
```

```
# search over various ARMA models
auto.arima(dens, max.p=2, max.q=2, seasonal=FALSE)
```

```
## Series: dens
```

```
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1  intercept
##       0.5576     1.1474
## s.e.  0.0641     0.0505
##
## sigma^2 estimated as 0.08595:  log likelihood=-31.43
## AIC=68.85   AICc=69   BIC=78.23
```

And we get the same answer here as well.

Question 6

Write out the best model in the form of Equation (1.26) using the underscore notation to refer to subscripts (e.g., write x_t for xt). You can round any parameters/coefficients to the nearest hundreth. (Hint: if the mean of the time series is not zero, refer to Eqn 1.27 in the lab handout).

Answer

Our "best" model is an AR(1) model that includes a non-zero intercept, so it will look exactly like Eqn 1.27. We can get the parameter values directly from the output in Question 5. So, our AR(1) model becomes:

$$x_t = 1.15 + 0.56(x_{t-1} - 1.15) + w_t \text{ with } w_t \sim \mathsf{N}(0, 0.085).$$

If you wanted to write it out using a more R-friendly notation (i.e., something that you can read in a text editor), you could write something like

```
x_t = 1.15 + 0.56*(x_{t-1} - 1.15) + w_t  with  w_t ~ N(0,0.085)
```