ESPM 205 - Lab 7 & Homework Assignment 4

Jennifer Natali, with reference to Julia Nicholson's 2020 code and help from chatgpt for R coding questi

29 October 2024

Instructions

In this lab and associated homework assignment, we will advance the final project. We will get the data ready, we will specify a model to fit your own data, we will fit the model, and we will interpret the results. This lab is meant to build on Homework Assignment 2 (HW2), and get you closer to the final model or set of models that you will be using for your project.

Most of you will be fitting MAR or MARSS models. If so, answer questions 1-5 below. If in our grading of HW2 we recommended that you use some method other than MAR on MARSS (e.g., ARIMA), please see question 6 first.

Please submit the completed lab on bCourses as a knitted R Markdown (html or pdf) by next Tuesday, Oct 29th before midnight. This submission will be the 4th and last Homework Assignment (HW4).

Questions

Question 1 In your individual project, what is (are) your variate(s), also known as response(s)?

Create the object (or recover it from HW2), and name it 'dat'. If you have many variates, you do not need to analyze the whole dataset in this lab/HW4. However, the closer the resemblance of this data set to the one you will end up analyzing, the better. E.g., if your question is at the community level, then include several species; if you would like to compare a particular physical variable across different sites, then include several sites.

If you have multivariate responses, name rows appropriately so that you can keep track of each state. Do you need to further clean and/or transform these data for analysis (e.g., log-transform, z-score)? If so, do it below (and name this new object 'transformed_dat'). Remember time needs to go over columns (use tidyr's 'pivot_wider' if necessary), and you need a 'matrix' object–you can check that using the function 'class()' [1 point]

```r
# Load libraries
library(MARSS)

# Setup directories and filepaths
home_dir='/Volumes/SANDISK_SSD_G40/GoogleDrive/GitHub/'
repository_dir = paste(home_dir,'sagehen_meadows/', sep='')
groundwater_data_dir = 'data/field_observations/groundwater/biweekly_manual/'
groundwater_weekly_matrix_filepath = paste(repository_dir, groundwater_data_dir,
                                'groundwater_weekly_matrix.csv',
                                sep='')

# Load response data: weekly groundwater measurements
dat <- as.matrix(read.csv(groundwater_weekly_matrix_filepath), header=TRUE)

# Check matrix dimensions
dim(dat)
```

```
## [1] 54 79
```

```r
# Is it a 'matrix' object? check using the function 'class()'
class(dat)
```

```
## [1] "matrix" "array"
```

Your text here (<100 words). Groundwater variate data has 54 wells, with observations spaced weekly over 68 weeks in 2018, 2019 and 2021 (2024 to be added).

Question 2 What is (are) your covariate(s), aka driver(s), if any? Z-score them and make sure they have no missing data (MARSS does not allow NA's in covariate data). You can name them 'transformed_covar'. Remember time needs to go over columns (use tidyr's 'pivot_wider' if necessary), and you need a 'matrix' object–you can check that using the function 'class()' [1 point]

```r
# Load covariate data
# --- Discharge

# TODO: Add covariate:
# --- Greenness values for vegetation plant functional types
#     (proxy for seasonality of photosynthesis rates)
#     (source: sub-daily images from tree-mounted, time-lapse camera)

# TODO: Consider Potential Hydro Covariates:
# --- Previous winter snowpack (how express?)
# --- Multi-year snowpack (know that groundwater at spring dated to be 30yo)
# --- Topographic Wetness Index
# --- Distance to Sagehen Creek or tributary stream channel, following topo

# TODO: Consider Potential Meteorological Covariates:
# --- Temperature (at east only vs near each site; consider daily mean, max, cumulative)
# --- Precip (total per day)
# --- Relative Humidity (consider daily mean)
# --- PAR (consider cumulative, daily mean, max)

# TODO: Consider how covariates correlate with each other??

### --- Load DISCHARGE from USGS ---

# Load NWIS data retrieval package for
# USGS and EPA Hydro and Water Quality Data
library(dataRetrieval)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(ISOweek)

# Sagehen Creek NWIS site number and discharge code
sagehen_NWIS_site <- "10343500"
discharge_code <- "00060"

sagehen_NWIS_data_available <- whatNWISdata(siteNumber=sagehen_NWIS_site,
                                            parameterCd=discharge_code,
                                            service="dv",
                                            statCd="00003")

# Get time range from groundwater time series data
# (start and end date from week_year column labels)
colnames(dat) <- sub("^X", "", colnames(dat))
isoweek_range <- colnames(dat)[-1]
isoweek_list <- as.list(isoweek_range)

# get start date of the groundwater time series
start_week <- isoweek_list[1]
iso_start_year <- substr(start_week, 1, 4)
iso_start_week <- substr(start_week, 5, 6)
start_date <- ISOweek2date(paste0(iso_start_year, "-W", iso_start_week, "-1"))

# get end date of the groundwater time series
end_week <- isoweek_list[length(isoweek_list)]
iso_end_year <- substr(end_week, 1, 4)
# add one to isoweek so I capture the last date of the time series
# (not the first day of the last week)
iso_end_week <- as.integer(substr(end_week, 5, 6)) + 1
iso_end_week <- sprintf("%02d", iso_end_week)
end_date <- ISOweek2date(paste0(iso_end_year, "-W", iso_end_week, "-1"))

# get the mean daily discharge for the appropriate date range
# reported in cfs according to USGS
discharge <- readNWISdv(sagehen_NWIS_site,
                    discharge_code, start_date, end_date)
# limit to two columns: date and flow
discharge <- discharge[, 3:4] %>%
  rename(date = Date, flow_cfs = X_00060_00003)

# add new column with year and isoweek combined
# (e.g. 201820 for year 2018, week 20)
discharge <- discharge %>%
  mutate(year_week = as.character(paste0(year(as.Date(date)), sprintf("%02d", isoweek(date)))))
```

```r
# average flow (7 days) for each year_week
# --- TODO: is averaging the best representation of the data?
discharge <- discharge %>%
  group_by(year_week) %>%
  summarise(
    flow_cfs = mean(flow_cfs, na.rm = TRUE),
    .groups = "drop"
  )

# create grid to match groundwater time series
discharge_grid <- expand_grid(
  year_week = as.character(isoweek_list)
  )

# join the grid with discharge data
discharge_grid <- discharge_grid %>%
  left_join(discharge, by = "year_week")

# transform to timesteps as columns
discharge_matrix <- discharge_grid %>%
  pivot_wider(
    names_from = year_week,
    values_from = flow_cfs
  )

# check completeness
discharge_matrix %>%
  summarize(
    na_sum = sum(across(everything(), ~ is.na(.))),
    na_percent = 100 * na_sum / (n() * ncol(.))
  )
```

```
## # A tibble: 1 x 2
##   na_sum na_percent
##    <int>      <dbl>
## 1      0          0
```

```r
# Convert to matrix, validate and check dimensions
discharge_matrix <- as.matrix(discharge_matrix)
class(discharge_matrix)
```

```
## [1] "matrix" "array"
```

```r
dim(discharge_matrix)
```

```
## [1]  1 78
```

```r
# Z-score flow values
discharge_covariate <- zscore(discharge_matrix)
```

Your text here (<100 words). So far, covariate is Sagehen Creek discharge (at East meadow where "E" wells located). Data is complete, validated (it's a matrix of same time series as groundwater data), and zscored. Other covariates being considered (noted in TODO comments).

Question 3 Is each observation suppposed to be modeled as a different state, or do you have 'replicate' observations, i.e. more than one observation being funneled into a state (via the Z matrix)? What are the dimensions of your Y's (observations x time steps) and X's (states x time steps)? Build the Z matrix you need, or specify it using a shortcut (e.g., Z = “identity”). [1 point]

```
# H1: Every well is its own state; Z is the identity matrix
Z_1 = "identity"

### MEADOW SITES ###
# H2: 3 meadow sites (Kiln, East, Lo)
Z_2 = matrix(NA, nrow=0, ncol=3)
# --- loop thru well_ids to determine meadow site
for (i in 1:nrow(dat)) {
  well_id <- dat[i,1]
  meadow <- substring(well_id, 1, 1)
  #print(meadow)
  if (meadow == "E") {
    row <- c(1, 0, 0)
  } else if (meadow == "K") {
    row <- c(0, 1, 0)
  } else if (meadow == "L") {
    row <- c(0, 0, 1)
  } else print("WARNING: Well Meadow Site is NOT E, K, or L")
  Z_2 <- rbind(Z_2, row)
}
dim(Z_2)
```

```
## [1] 54  3
```

```
### PLANT FUNCTIONAL TYPES ###
# H3: PFT x 4 (sedge, willow, mixed herbaceous, pine)
Z_3 = matrix(NA, nrow=0, ncol=4)
# --- loop thru well_ids to determine PFT
for (i in 1:nrow(dat)) {
  well_id <- dat[i,1]
  pft <- substring(well_id, 2, 2)
  #print(pft)
  if (pft == "E") {
    row <- c(1, 0, 0, 0)
  } else if (pft == "W") {
    row <- c(0, 1, 0, 0)
  } else if (pft == "H") {
    row <- c(0, 0, 1, 0)
  } else if (pft == "F") {
    row <- c(0, 0, 0, 1)
  } else print("WARNING: Well PFT is NOT E, W, H or F")
  Z_3 <- rbind(Z_3, row)
}
dim(Z_3)
```

```
## [1] 54  4
```

```
### HYDROGEOMORPHIC ZONES ###
# H4: HGMZ x 3 (riparian, terrace, fan)
Z_4 = matrix(NA, nrow=0, ncol=3)
# --- loop thru well_ids to determine PFT
for (i in 1:nrow(dat)) {
  well_id <- dat[i,1]
  hgmz <- substring(well_id, 3, 3)
  #print(hgmz)
  if (hgmz == "R") {
    row <- c(1, 0, 0)
  } else if (hgmz == "T") {
    row <- c(0, 1, 0)
  } else if (hgmz == "F") {
    row <- c(0, 0, 1)
  } else print("WARNING: Well HGMZ is NOT R, T, or F")
  Z_4 <- rbind(Z_4, row)
}
dim(Z_4)
```

```
## [1] 54  3
```

```
### STRATIFIED COMBO ###
# H5: PFT x HGMZ (3x4 = 12 combos)
Z_5 = matrix(0, nrow=dim(dat)[1], ncol=12)
colnames(Z_5) <- c("ER", "ET", "EF", "WR", "WT", "WF", "HR", "HT", "HF", "FR", "FT", "FF")
# --- loop thru well_ids to determine PFT x HGMZ
for (i in 1:nrow(dat)) {
  well_id <- dat[i,1]
  combo <- substring(well_id, 2, 3)
  # Find the column that matches the PFT x HGMZ combo characters
  col_index <- which(colnames(Z_5) == combo)
  if (length(col_index) == 1) { # if combo is valid
    Z_5[i, col_index] <- 1 # assign to 1
  }
}
dim(Z_5)
```

```
## [1] 54 12
```

```
# TODO: Consider SITE x PFT x HGMZ combo
# TODO: Consider well distance from Sagehen Creek (well across a transect)

# TODO: Use dynamic factor analysis to improve categorization of HGMZ / PFT
#       address uncertainty about categorization; see MARSS User Guide Ch 10.
```

Your text here (<100 words). The data has 54 rows which are replicate observations, different wells. Half the wells were inherited, placed across transects at 1, 5, 10, 20, to 100 m from Sagehen Creek. Half were distributed in a stratified random sample across 3 meadows, mapped by plant functional type (pft) and hydrogeomorphic zone (hgmz). Most wells are in East or Kiln meadow, only three wells in Lo meadow site. Each meadow is about 2-3 km apart along Sagehen Creek. Well replicates stratifications considered: by meadow site, PFT, HGMZ, the cross PFT x HGMZ, or possibly site x PFT x HGMZ.

Question 4 Specify the rest of your MAR/MARSS parameters using a model list, like we have been doing so far: R (based on the number of observations), and U, B, C, Q (based on the number of states).

If you would like to fit MAR instead of MARSS, then set R to “zero”. Remember what we learned over the past few weeks, e.g. if you want to focus on the B matrix (e.g. species interactions) then it is best to fix U (long-term trend) to zero, which you can do after demeaning the variate data.

If you are building custom matrices, remember that R and Q need to be symmetrical, B does not need to be. Also, R, Q, and B need to be square; all other matrices may be rectangular. If you have covariate data, assign it here as well to the model list (“c”).

If you plan on comparing models, it is best practice to start with a simple model structure (e.g., Q = “diagonal and equal" instead of "unconstrained"), and make it progressively more complex. [1 point]

```
### --- MARSS MODEL PARAMETERS --- ###
# NOTE: Defaults from MARSS User Guide, p 30-31
R.1 <- 'diagonal and equal' # observations are iid
U.1 <- 'unequal' # all different
B.1 <- 'identity' # no interaction among the x's in x
C.1 <- 'zero' # no inputs
Q.1 <- 'diagonal and equal' # process errors independent, but different variances
A.1 <- 'zero' # ALERT: I don't know what this is!

# NOTE: Julia's params
R.2 <- "zero" # Was not converging otherwise, but had less data
U.2 <- "zero" # Assume no overall trend in data
B.2 <- "identity" #Assume wells do not interact with oneanother, and no density dependence, since we zs
Q.2 <- "diagonal and unequal"
C.2 <- "unequal" #Let each state respond to covariates in a different way
c.2  <- discharge_covariate
A.2 <- "zero" # ALERT: I don't know what this is!

# NOTE: Simpler?
R.3 <- 'zero'
U.3 <- 'equal'
B.3 <- 'identity' # no interaction among the x's in x
C.3 <- 'zero' # no inputs
Q.3 <- 'diagonal and equal' # process errors independent, but different variances
A.3 <- 'zero' # ALERT: I don't know what this is!
```

Your text here (<100 words). Not sure what I'm doing yet, went with defaults to get started.

Question 5 Fit the model. If you get errors or warnings, first check that the model specification is right (e.g., number of dimensions of each matrix). If dimensions are right, but the model does not converge, increase number of iterations using the agument 'maxit'. If it still does not converge, check if the model you are fitting does not make sense given the data (e.g. perhaps you are fitting a stationary model to a non-stationary process), and re-specify the model accordingly, in step 5.

If you are fitting MARSS and one of the variances goes to zero (Q or R), try fitting a MAR model instead.

If errors persist, check the MARSS User Guide: https://cran.r-project.org/web/packages/MARSS/vignettes/UserGuide.pdf ("Appendix A - Warnings and errors", page 309).

Once it does work: bootstrap the model(s). What do you obtain? Is it what you expected? What are the next steps to complete analyses for your final project? [1 point]

```
### Clean up dat ###
# Remove header row and well_id column from dat
colnames(dat) <- NULL
dat <- dat[,-1]
# Ensure it's all numeric
dat <- apply(dat, 2, as.numeric)

# FIT THE MODEL
# # Simplest possible?
# model.list = list(Z=Z_1)
# model_1 <- MARSS(dat, model = model.list, control=list(maxit=10000))
#
# # Better?
# model.list = list(Z=Z_1,B=B.1,U=U.1,Q=Q.1,A=A.1,R=R.1,C=C.1)
# model_2 <- MARSS(dat, model = model.list, control=list(maxit=10000))

# # Is this simpler?
# model.list = list(Z=Z_1,B=B.3,U=U.3,Q=Q.3,A=A.3,R=R.3,C=C.3)
# model_3 <- MARSS(dat, model = model.list, control=list(maxit=10000))

# -----model_3 returns----
# Estimation converged in 141 iterations.
# Log-likelihood: -3889.696
# AIC: 7891.393   AICc: 7899.353
#
#          Estimate
# U.1        -0.113
# Q.diag    173.356
# x0.X.Y1     5.015
# x0.X.Y2    17.480
# x0.X.Y3     9.135
# x0.X.Y4    83.979
# x0.X.Y5    40.208
# x0.X.Y6    22.197
# x0.X.Y7    11.660
# x0.X.Y8     7.851
# x0.X.Y9    34.674
# x0.X.Y10   26.758
# x0.X.Y11   41.021
# x0.X.Y12   53.413
# x0.X.Y13   24.088
# x0.X.Y14    3.610
# x0.X.Y15   29.781
# x0.X.Y16   66.858
# x0.X.Y17   -0.927
# x0.X.Y18   19.523
# x0.X.Y19   59.771
# x0.X.Y20    0.000
# x0.X.Y21   10.397
# x0.X.Y22    5.194
# x0.X.Y23   38.808
# x0.X.Y24   37.042
# x0.X.Y25   35.767
# x0.X.Y26    7.909
# x0.X.Y27   21.070
# x0.X.Y28    2.502
# x0.X.Y29   -4.492
# x0.X.Y30  -15.114
# x0.X.Y31   18.905
# x0.X.Y32   -2.020
# x0.X.Y33   43.822
# x0.X.Y34    2.569
# x0.X.Y35   64.425
# x0.X.Y36   33.650
# x0.X.Y37    0.000
# x0.X.Y38   19.402
# x0.X.Y39   -6.575
# x0.X.Y40   37.941
# x0.X.Y41   27.834
# x0.X.Y42   54.970
# x0.X.Y43   39.210
# x0.X.Y44   27.256
# x0.X.Y45   47.828
# x0.X.Y46    7.970
# x0.X.Y47    4.783
# x0.X.Y48   55.355
# x0.X.Y49   24.500
# x0.X.Y50   41.825
# x0.X.Y51   60.994
# x0.X.Y52   35.783
# x0.X.Y53   41.607
# x0.X.Y54   31.409

# TODO: BOOTSTRAP THE MODEL
```

Model_3 did return, finally. Others were taking too long, no errors yet. Model_3 AICc score seems astronomical. Interesting that there's a slight negative trend, U=-0.1 Q seems high at 178, but it includes observation error. Considering that groundwater data is <50% complete at the weekly timestep, I'm hopeful because at least the model returned. Next steps: - better understand model params and how to align with my hypotheses. - consider if incompleteness is a problem - when/how to incorporate discharge (or temp/precip/snowpack) covariates - consider using B matrix to test interaction between groundwater and discharge

Question 6 [ONLY FOR THOSE OF YOU WHO ARE NOT USING MAR/MARSS] Discuss with Albert/Robert and find an appropriate model to use. Follow the steps 1-5 above by reading in your response and driver data (if any), transforming it (if necessary), and specifying a model (e.g, ARIMA, DFA, DLM) that gets at your question. Fit the model, if you can (if we have not seen this in class yet, let's troubleshoot together). What did you learn? What are the next steps to complete analyses for your final project? [5 points]

```
# your code here
```

Your text here (<100 words).

Any notes (optional)

Did not get model to return until 11:40pm on Tues night, and it takes >10 minutes to return results. Will keep working on it!