HW09 Confidence sets

Doug Nychka

2024-04-12

Setup

Load the climate model ensemble results for the Boulder grid box.

```
suppressMessages(library( fields))
setwd("~/Dropbox/Home/Teaching/MATH531/MATH-531/MATH531S2024/Assignments")
load("HW09.rda")
dim(MAMTempBoulder )
```

```
## [1] 75 30
```

```
dim( TempGlobal)
```

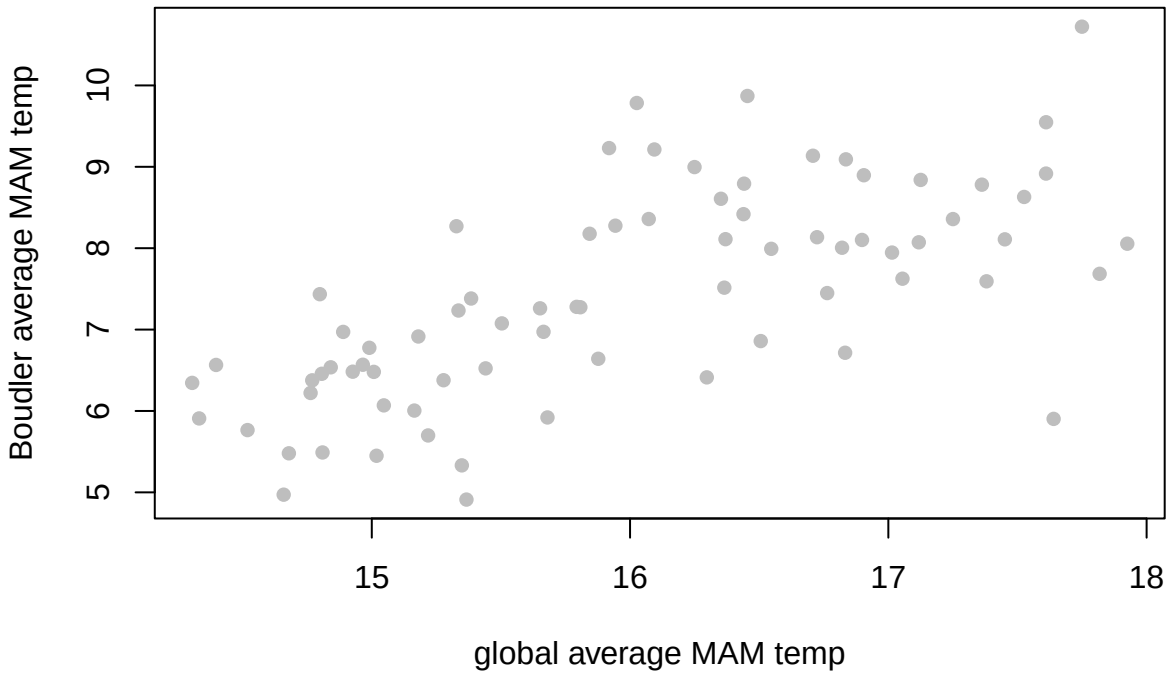
```
## [1] 75 30
```

These model output data sets are a very small summary of a large climate model experiment (LENS) conducted by the National Center for Atmospheric research. In the model the surface of the earth is divided up into roughly 50K grid boxes that that are about 60 miles on a side. The data here is the grid box containing Boulder, CO. In each data set are the simulated average spring daily temperatures (March, April and May) (in C) for the (model) years 2006- 2080. The data set **TempGlobal** are the global average temperatures and **MAMTempBoulder** the average MAM temperatures for the Boulder grid box.

Thirty (30) independent runs of the model are available and these are the columns in both of the data sets. In climate science the 30 runs are known as an *ensemble* and the individual runs are refered to as the ensemble members.

The goal is to quantify the relationship between the global temperature (the "X") and the local MAM temperature in Boulder (the "Y"). Below is a plot for the 10th member of the ensemble, a quadratic fit to the relationship and the predicted polynomial at 40 points. Note that the call to predict also computes the standard errors for the predicted values

```
y10<- MAMTempBoulder[,10]
x10<- TempGlobal[,10]
plot(x10, y10,
     xlab="global average MAM temp", ylab="Boudler average MAM temp",
     pch=16, col="grey")
df<- data.frame( y10= y10, x10=x10)
fit10<- lm(y10 ~ x10 + I(x10^2), data=df )
x10Grid<-seq( 5,11, length.out=40)
dfGrid<- data.frame( x10 = x10Grid)
hPredict<- predict( fit10, dfGrid,se.fit = TRUE )
lines( x10Grid, hPredict$fit, lwd=2, col="orange3")
```




```
# components from predict
names( hPredict)
```

```
## [1] "fit"            "se.fit"         "df"
## [4] "residual.scale"
```

Problem 1

- Using the code above reproduce the figure and add lines that are upper and lower bounds for 95% confidence intervals for predicting the mean. (use a solid line in black)
- Add lines that are upper and lower bounds for 95% simultaneous envelpoe using the Bonferoni adjustment. (use a dashed line in black)
- compute the predicted standard errors reported in **hPredict$sefit** "by hand" use only basic matrix/vector computations to find the standard errors. For this computation start with the **X** matrix. Also for convenience I have included is the matrix for prediction **XPred**, although you may want to a **for** loop on the rows to find the individual prediction standard errors.
- **Extra credit** Instead of just adding lines for the Bonferroni adjustment with a shaded envelope to indicate this region. See the handy fields function **envelopePlot** to help with this. You will have to set up the plot, draw the envelope first then add the other information as the envelope will obscure the other graphics elements if drawn last.

```
X<- cbind( 1, x10, x10^2)
XPred<- cbind( 1, x10Grid, x10Grid^2)
```

Problem 2

Below is code to create a dataframe combining the individual ensemble members and the corresponding global temperatures. The member ID is formatted a *factor*. Note that way the data is "unrolled" into data

frame columns is fragile in that it depends exactly on how the **c** operates on matrices and also the result of the **rep** function.

```
LENSExample<- data.frame( BMAM= c(MAMTempBoulder),
                          GMAM= c(TempGlobal),
                          member= as.factor((rep( 1:30, rep(75,30) )))
)

#  a regression where every intercept and slope are estimated separately for each ensemble member.
lmFull<- lm(BMAM ~ GMAM:member  + member - 1  , LENSExample )

intercepts<- lmFull$coefficients[1:30]
slopes<- lmFull$coefficients[(1:30) + 30]

#Extracting the standard errors for the slopes
 summaryFit<- summary(lmFull )
 tableCoefficients<- summaryFit$coefficients
 SESlopes<- tableCoefficients[30 + (1:30), 2]
# the monster covariance matrix  sigma^2 inverse(X^TX)
 sigmaHat2<- (summaryFit$sigma)^2
bigCov<- sigmaHat2* summaryFit$cov.unscaled
# check that things are correct
# diag elements of bigCov should the SE squared.
test.for.zero( diag( bigCov), tableCoefficients[, 2]^2 )
```

```
## PASSED test at tolerance  1e-08
```

- Linear model theory gives a covariance matrix for the OLS estimates of beta: $\sigma^2(X^TX)^{-1}$. For the particular fit above (**lmFull**) describe the structure of the "X" matrix and explain why the slope estimates for different ensemble members are uncorrelated.
- Explain why the diagonal elements of the covariance matrix created above are the square of the standard errors.
- Find a 95% confidence interval for the difference of the first and second slopes.
- Find all pairwise differences among the slopes. Note that there are `choose( 30,2) = 435` of these. Use a double **for** loop to compute all of them. To standardize these for comparison divide each difference by the square root of the of the sum of the two standard errors. I.e. for the `i` , `j` pair: `sqrt(SESlopes[i]^2 +SESlopes[j]^2  )` Based on linear model theory what will be the distribution for a particular standardized difference in slopes under the assumption that they are the same?
- For your 435 standardized pairs square them and compare to $F_\alpha(1, 75-2)$ with $\alpha = .05$. Explain why this is the same as comparing the standardized pairs to a t-distribution $\alpha = .025$ and $75-2$ degrees of freedom. How many pairs are significantly different from zero at the 95% confidence level?
- Now adjust your inference using Bonferroni. In this case work with the standardized differences and compare to a t-distribution with $\alpha = .025/435$ and $75-2$. How much different would your results be if you compared to just a normal distribution?

Below is code I used to find all the standized pairs.

```
stdPairs<- matrix( NA, (30*29)/2  )
kk<- 0
for( i in 1:29){
  for( j in (i+1):30 ){
    kk<- kk +1
    stdPairs[kk]<- (slopes[i]- slopes[j])/
      sqrt( SESlopes[i]^2 + SESlopes[j]^2)
}
}
```

Problem 3

For the pairs described above apply the Scheffe's S-Method (Section 5.1.1 in Seber/Lee and specifically the simultanous intervals in 5.12) to test for nonzero pairs. Note that that rejecting the contrast pair being significantly different from zero is the same as the confidence interval not covering zero.

Problem 4

Assume that the slopes are the same. Use a Monte Carlo simulation to determine the distribution of the maximum of the absolute value of the standardized pairs.

- Assume the "true" model is one fit to all the data with a common slope and intercept and use for σ the estimate from the data.
- Generate 5000 synthetic data sets and for each find the absolute value of the maximum absolute difference for the standardized slopes.
- Make a histogram of these maxima and locate on this a vertical line that is the maximum found for the actual data.
- Using these Monte Carlo results, what is the (approximate) p-value for the hypothesis test that the slopes are equal based on the statistic of the maximum absolute slopes.