SIOC 290 Homework
August 20, 2015
Sam Shen

The three datasets used for this homework can be downloaded from
http://shen.sdsu.edu/datacuyatmin5110.csv
http://shen.sdsu.edu/datahansen.txt
http://shen.sdsu.edu/dataPrcpRecon.csv

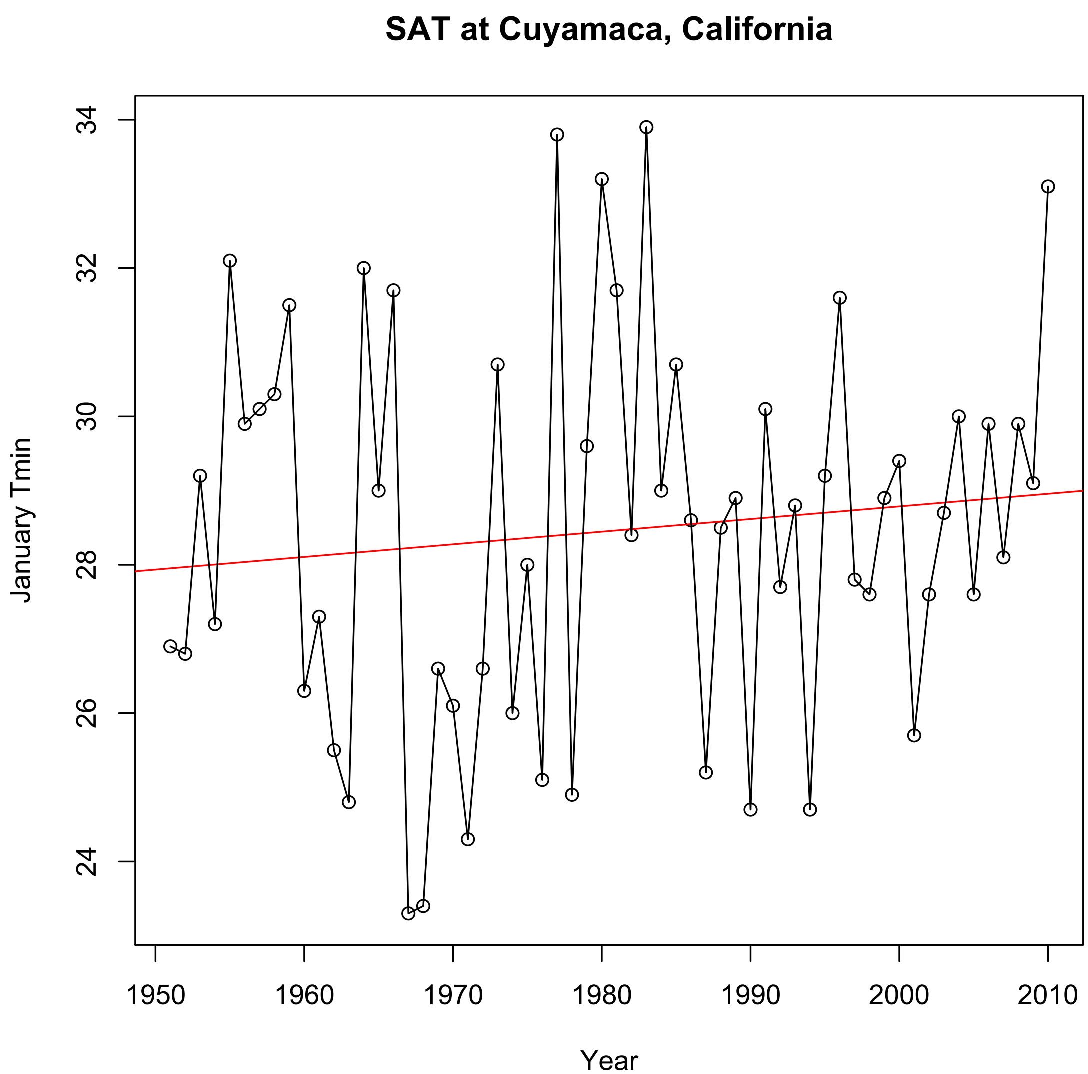
To save the txt file, use the "Save Page" button on your computer download page. The csv file should be automatically saved in your download directory.

Homework #1: Exercise 1.3

```
# Download the Cuyamaca station data and save it into your working directory
#
# /Users/sshen/Desktop/MyDocs/teach/SIOC290-ClimateMath/Hwk-and-
#+Exams/Hwk1.3
#
# The file name for Tmin is
# CA042239_6765.csv
#
# Manually remove the unnecessary head and start from 1893 March
# Or just keep the data in Jan 1951-Dec 2010 only
# Insert a row and add your own file head: stnid, year, month, tminf
# Save this new file as cuyatmin.csv

# Read this csv into r. Make sure that your R console is in the same directory as
# your data files

dtmin <- read.csv("~/Desktop/MyDocs/teach/SIOC290-ClimateMath/Hwk-and-
Exams/Hwk#1/datacuyatmin5110.csv", header=F)

# Check the dimension of data dtmin

dim(dtmin)

# [1] 1463    4, 1463 rows and 4 columns: Stn ID, year, month, Tmin data in degF

# Read out Tmin data only

tmin<- dtmin[ ,4]

# Tmin's properties
```

```
summary(tmin)

#   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#  15.80   29.65   35.60   37.71   45.95   60.80

# Read January Tmin data from 1951-2010

# Extract only the January data out into two vectors: one for year and one for
# Tmin, x5110, y5110

x5110=seq(1:60)
y5110=seq(1:60)

# Extract data from dtmin data matrix
for (i in seq(1, 60)) x5110[i]=dtmin[i*12,2]
for (i in seq(1, 60)) y5110[i]=dtmin[i*12,4]
# Check the data summary of Jan Tmin in these 60 years
summary(y5110)

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  23.30   26.60   28.65   28.46   30.02   33.90

# Linear regression and plotting

# Linear regression and put the result into reg
reg<-lm(y5110 ~ x5110)
summary(reg)

Call:
lm(formula = y5110 ~ x5110)

Residuals:
    Min      1Q  Median      3Q     Max
-4.9250 -1.6719 -0.0146  1.5652  5.4046

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.28339   38.38773  -0.138    0.891
x5110        0.01704    0.01938   0.879    0.383
[The highlighted is the trend 0.017 oF/year, 1.7oF/100a]

Residual standard error: 2.6 on 58 degrees of freedom
Multiple R-squared:  0.01314,      Adjusted R-squared:  -0.003871
F-statistic: 0.7725 on 1 and 58 DF,  p-value: 0.3831

# Plot data into points
```

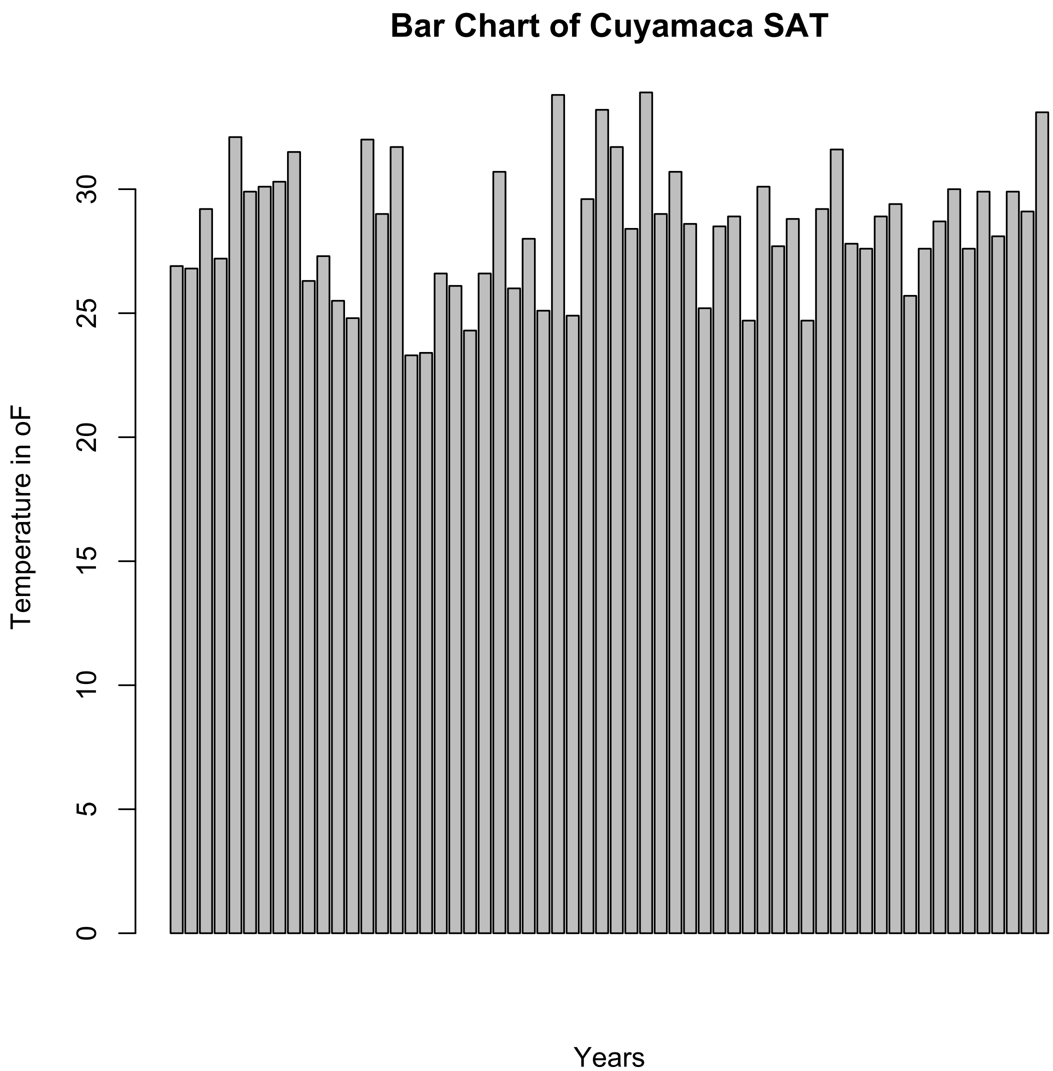
```
plot(x5110,y5110,xlab="Year",ylab="January Tmin",main="SAT at Cuyamaca,
California")

# Connect the points
lines(x5110,y5110)

# Add the trend line
abline(reg,col="red")

# Now one can save the plot in pdf
```



Hwk #2: Exercise 1.4

Use the area under the curve and above zero to explain the warming as an
accumulated temperature time increase.

```
barplot(y5110,main="Bar Chart of Cuyamaca SAT",xlab="Years",
ylab="Temperature in oF")
```



SIOC 290 Homework
August 20, 2015
Sam Shen

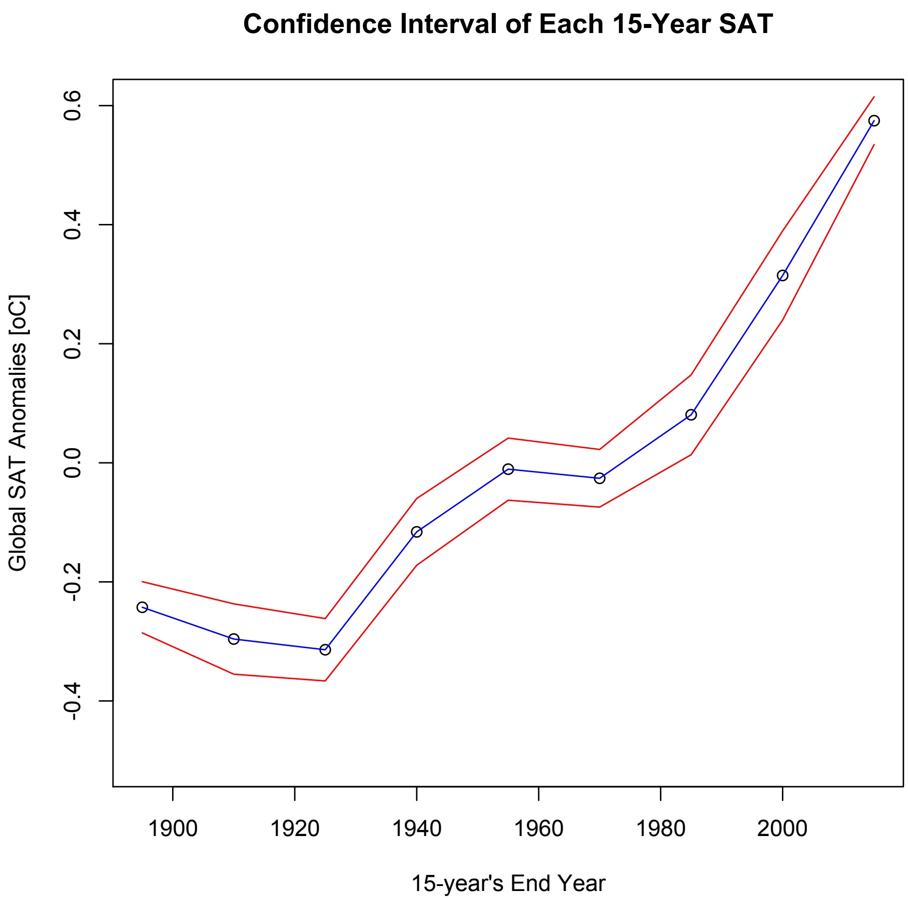
Homework #3: Exercise 3.1

Download the Hansen data and save it into your working directory
You may need to copy and paste it on an excel and save it as a

```
# Tab Delimited txt file
# /Users/sshen/Desktop/MyDocs/teach/SIOC290-ClimateMath/Hwk-and-
+Exams/Hwk3.1
#
# The file name for the Hansen global ocean and land data is
# Hansen.txt
# Manually remove the head and leave only 1880 to 2014 temp.

# Read this txt into r. Make sure that your R console is in the same directory as
# your data files. Apple D can find the R console's directory. However, this may
# vary from computer to computer.

dhan <- read.table("~/Desktop/MyDocs/teach/SIOC290-ClimateMath/Hwk-and-
Exams/Hwk#1/datahansen.txt")

# Check the dimension of data dhan

dim(dhan)

# [1] 135   3     means 135 rows and 3 columns: year, temp data oC, 5-yr ma

# Read out temp data only

htemp<-dhan[ ,2]

# htemp's properties

summary(htemp)

#      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
#-0.470000 -0.215000 -0.070000 -0.003926  0.140000  0.680000

# Covert the 135 years of temp data into a 15 by 9 matrix
# Each column represents a 15-year section of the temp data

dmat=matrix(htemp,15)



dmat
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7] [,8] [,9]
 [1,] -0.21 -0.25 -0.46 -0.21  0.07 -0.12  0.04 0.08 0.40
 [2,] -0.13 -0.17 -0.44 -0.09  0.08 -0.18 -0.07 0.14 0.52
 [3,] -0.16 -0.18 -0.41 -0.18  0.05  0.03  0.02 0.28 0.60
 [4,] -0.19 -0.31 -0.39 -0.16  0.06  0.05  0.15 0.35 0.59
 [5,] -0.27 -0.20 -0.22 -0.31  0.14  0.03 -0.07 0.24 0.51
```

```
 [6,] -0.25 -0.15 -0.16 -0.12  0.01 -0.04 -0.01 0.39 0.65
 [7,] -0.24 -0.21 -0.35 -0.07 -0.07  0.05 -0.12 0.38 0.59
 [8,] -0.31 -0.30 -0.43 -0.11 -0.04  0.04  0.14 0.19 0.62
 [9,] -0.19 -0.36 -0.31 -0.25 -0.10  0.08  0.05 0.20 0.49
[10,] -0.10 -0.44 -0.29 -0.09 -0.11 -0.19  0.11 0.28 0.59
[11,] -0.33 -0.29 -0.27 -0.15 -0.19 -0.10  0.22 0.42 0.66
[12,] -0.27 -0.26 -0.20 -0.10 -0.06 -0.04  0.28 0.32 0.55
[13,] -0.31 -0.42 -0.29  0.03  0.02 -0.01  0.09 0.45 0.57
[14,] -0.36 -0.43 -0.25  0.06  0.09 -0.05  0.27 0.61 0.60
[15,] -0.32 -0.47 -0.24  0.01 -0.11  0.06  0.11 0.39 0.68


# Now take the mean of each column and produce 9 means
dave=seq(9)
dave
# [1] 1 2 3 4 5 6 7 8 9
for(i in seq(9)) dave[i]=mean(dmat[,i])
matrix(dave,9)

# (a) The average of the nine 15-yr sections, oC.
 [1,] -0.24266667
 [2,] -0.29600000
 [3,] -0.31400000
 [4,] -0.11600000
 [5,] -0.01066667
 [6,] -0.02600000
 [7,]  0.08066667
 [8,]  0.31466667
 [9,]  0.57466667
#The coldest is -0.31oC in 1911-1925, and the hottest is 0.57oC in 1999-2014

# Or we can use year as the indicator for every 15-year period
d15<-matrix(1880+15*seq(9),9)
ave <- cbind(d15,dave)
ave
               15-yr ave
 [1,] 1895 -0.24266667
 [2,] 1910 -0.29600000
 [3,] 1925 -0.31400000
 [4,] 1940 -0.11600000
 [5,] 1955 -0.01066667
 [6,] 1970 -0.02600000
 [7,] 1985  0.08066667
 [8,] 2000  0.31466667
 [9,] 2015  0.57466667
```

```
#(b) Confidence interval
ci=seq(9)
for(i in seq(9)) ci[i]=qt(0.975,14)*sd(dmat[,i]/sqrt(15))
confilow <- dave-ci
confihigh <- dave + ci

cimat<-cbind(matrix(confilow,9), matrix(dave,9), matrix(confihigh,9))
cimat
            [,1]        [,2]        [,3]
 [1,] -0.28568818 -0.24266667 -0.19964515    #1881-1895 CI (-0.29, -0.20)
 [2,] -0.35508321 -0.29600000 -0.23691679
 [3,] -0.36636092 -0.31400000 -0.26163908
 [4,] -0.17207783 -0.11600000 -0.05992217
 [5,] -0.06284598 -0.01066667  0.04151265
 [6,] -0.07440444 -0.02600000  0.02240444
 [7,]  0.01360692  0.08066667  0.14772641
 [8,]  0.23981489  0.31466667  0.38951844
 [9,]  0.53452137  0.57466667  0.61481196    #Significantly larger than 0


# Plot the results
d15 =matrix(1880+15*seq(9),9)
intp<-cbind(d15,cimat)
intp
      [,1]        [,2]        [,3]        [,4]
 [1,] 1895 -0.28568818 -0.24266667 -0.19964515
 [2,] 1910 -0.35508321 -0.29600000 -0.23691679
 [3,] 1925 -0.36636092 -0.31400000 -0.26163908
 [4,] 1940 -0.17207783 -0.11600000 -0.05992217
 [5,] 1955 -0.06284598 -0.01066667  0.04151265
 [6,] 1970 -0.07440444 -0.02600000  0.02240444
 [7,] 1985  0.01360692  0.08066667  0.14772641
 [8,] 2000  0.23981489  0.31466667  0.38951844
 [9,] 2015  0.53452137  0.57466667  0.61481196

# Plot the confidence intervals for all the 9 sections
plot(intp[,1], intp[,3],ylim=c(-0.5,0.6), main="Confidence Interval of Each 15-Year
SAT",xlab="15-year's End Year",ylab="Global SAT Anomalies [oC]")

lines(intp[,1],intp[,2],col="red")
lines(intp[,1],intp[,4],col="red")
lines(intp[,1],intp[,3],col="blue")
```



Hwk #6, Exercise 3.2. Significant difference between averages in two 15-year
periods

Examine the largest differences: 2015 (No 9) – 1925 (No 3)

```
dmat=matrix(htemp,15)
dmat
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7] [,8] [,9]
 [1,] -0.21 -0.25 -0.46 -0.21  0.07 -0.12  0.04 0.08 0.40
 [2,] -0.13 -0.17 -0.44 -0.09  0.08 -0.18 -0.07 0.14 0.52
 [3,] -0.16 -0.18 -0.41 -0.18  0.05  0.03  0.02 0.28 0.60
 [4,] -0.19 -0.31 -0.39 -0.16  0.06  0.05  0.15 0.35 0.59
 [5,] -0.27 -0.20 -0.22 -0.31  0.14  0.03 -0.07 0.24 0.51
 [6,] -0.25 -0.15 -0.16 -0.12  0.01 -0.04 -0.01 0.39 0.65
 [7,] -0.24 -0.21 -0.35 -0.07 -0.07  0.05 -0.12 0.38 0.59
 [8,] -0.31 -0.30 -0.43 -0.11 -0.04  0.04  0.14 0.19 0.62
 [9,] -0.19 -0.36 -0.31 -0.25 -0.10  0.08  0.05 0.20 0.49
[10,] -0.10 -0.44 -0.29 -0.09 -0.11 -0.19  0.11 0.28 0.59
```

```
[11,] -0.33 -0.29 -0.27 -0.15 -0.19 -0.10  0.22 0.42 0.66
[12,] -0.27 -0.26 -0.20 -0.10 -0.06 -0.04  0.28 0.32 0.55
[13,] -0.31 -0.42 -0.29  0.03  0.02 -0.01  0.09 0.45 0.57
[14,] -0.36 -0.43 -0.25  0.06  0.09 -0.05  0.27 0.61 0.60
[15,] -0.32 -0.47 -0.24  0.01 -0.11  0.06  0.11 0.39 0.68


tstar<- (intp[9,3]-intp[3,3])/sqrt(var(dmat[,9])/15 + var(dmat[,3])/15)

tstar
# [1] 28.88771

qt(0.95, 14)
 #[1] 1.76131

# This tstar is a very large number, much larger than tc=1.76, showing the
# average of the 2001-2015 is significantly larger than zero at 5% significance
# level, or the 1999-2014 period is significantly hotter than the 1910-1925 period.

# Examine the smallest differences: 1955 (No 5) – 1970 (No 6)

tstar<- (intp[6,3]-intp[5,3])/sqrt(var(dmat[,9])/15 + var(dmat[,3])/15)
tstar
[1] -0.4984377

qt(0.975, 14)
#[1] 2.144787
# Temperature of 1955-1969 is not significantly different from 1940-1954 at
#  5% significance level.
```

```
# SIOC 290S Homework Help
# Problem #5 and 6: Exercise # 4.1 and 4.2
# Sam Shen
# August 24, 2015

# 4.1: SVD for a space-time climate data matrix
```

```
#  Read the csv data. This is a big dataset of 2.2 MB.
mpat<-read.csv("~/Desktop/MyDocs/teach/SIOC290-ClimateMath/Hwk-and-Exams/Hwk#1/dataPrcpRecon.csv")

# Extract only the 2011 and 2012 data
p12<-pmat[,c(1,2,114,115)]

# The first two columns are lat and lon. The long goes from 0 to 360, lat from
#  -72.5 to 72.5

# San Diego lat and lon is (32.7N, 117W), equivalent to (32.7, 243). We choose
# four boxes as one on San Diego, one immediately south of San Diego, and two
# immediately north of San Diego. Then these four boxes cover a region
# (20-40, 240-245). These four boxes correspond to rows 1460-1463. We thus
# extract these four rows out
dim(p12) # Render dimension of the p12 matrix
[1] 2160    4
# Create a row counter
rownames(p12)<-1:2160

# Extract the four rows out
box4<-p12[rownames(p12)%in%c(1460,1461,1462, 1463),  ]
box4
      lat     lon      X2011  X2012
1460 22.5  242.5 -0.1620100 0.62067
1461 27.5  242.5 -0.0611800 0.28499
1462 32.5  242.5  0.0057461 0.57156
1463 37.5  242.5  0.0229020 0.64160
# This is the data matrix of two years for four boxes. The first two columns are lat
# and lon. Remove them.
box4by2<-box4[,3:4]
box4by2
          X2011      X2012
1460 -0.1620100 0.62067
1461 -0.0611800 0.28499
1462  0.0057461 0.57156
1463  0.0229020 0.64160

# Now SVD of this 4by2 matrix
s<-svd(box4by2)
s
$d
[1] 1.1014714 0.1486284

$u
```

```
         [,1]       [,2]
[1,] -0.5738814  0.7342785
[2,] -0.2624958  0.2485962
[3,] -0.5166209 -0.3625604
[4,] -0.5786700 -0.5172862

$v
          [,1]        [,2]
[1,]  0.08426253 -0.99644359
[2,] -0.99644359 -0.08426253
```

```
# Verify SVD decomposition A= U D V' (space*energy*time)

#Step 1: Construct the diagonal matrix D
D<-diag(s$d)
D

[,1]      [,2]
[1,] 1.101471 0.0000000
[2,] 0.000000 0.1486284

#The diagonal values are called eigenvalues of the covariance matrix
# corresponding to the data box4by2

#The decomposition of data box4by2 into  the space*variance*time:
# Data = EOF*Eigen*PC', i.e., A = U D V', U for row variations, the four boxes
# from south to north; V for column variations, the time 2011 and 2012.

# The reconstruction is
r1<-s$u%*%D%*%t(s$v)
r1
         [,1]    [,2]
[1,] -0.1620100 0.62067
[2,] -0.0611800 0.28499
[3,]  0.0057461 0.57156
[4,]  0.0229020 0.64160

box4by2

        X2011  NaN..2
1460 -0.1620100 0.62067
1461 -0.0611800 0.28499
1462  0.0057461 0.57156
1463  0.0229020 0.64160

# We see that the reconstructed matrix based on the SVD matrices is the same
```

as the original data box4by2. That is, the data matrix is decomposed.

There are some properties of the U, D, and V matrices.

```
# D=U' A V
# Get rid of the row head and column head before matrix multiplication.
boxnew<-matrix(seq(8),4)
boxnew[,1]=box4by2[,1]
boxnew[,2]=box4by2[,2]
boxnew
          [,1]    [,2]
[1,] -0.1620100 0.62067
[2,] -0.0611800 0.28499
[3,]  0.0057461 0.57156
[4,]  0.0229020 0.64160

mat1<- t(s$u) %*% boxnew %*% s$v
mat1
            [,1]         [,2]
[1,] 1.101471e+00 2.775558e-17
[2,] 2.445960e-16 1.486284e-01
# This is the diagonal matrix D.

# For the given year, the precipitation on the four grid boxes can be
# reconstructed by U and one row of V. This row corresponds to the year.

c1<-s$u%*%D%*%s$v[1,]
c1
          [,1]
[1,] -0.1620100
[2,] -0.0611800
[3,]  0.0057461
[4,]  0.0229020
# This is the EOF expansion of the year t data, also called spectral expansion.
# Dat(t)= pc1(t) . lamda1. EOF1 +  pc2(t) . lamda2. EOF2

# Vector c1 is the first column (i.e., the first year) of the data
boxnew
          [,1]    [,2]
[1,] -0.1620100 0.62067
[2,] -0.0611800 0.28499
[3,]  0.0057461 0.57156
[4,]  0.0229020 0.64160

# Properties of U matrix
s$u[,1]%*%s$u[,2]
```

```
          [,1]
[1,] 1.665335e-16
s$u[,1]%*%s$u[,1]

 s$u[,1]%*%s$u[,1]
     [,1]
[1,]    1
s$u[,2]%*%s$u[,2]
     [,1]
[1,]    1
```

#Thus the column vectors of U are orthonormal: zero or one. Or in the matrix
form. The column number is the EOF mode number, and row number is the
position number.

```
t(s$u)%*%s$u

             [,1]         [,2]
[1,] 1.000000e+00 1.665335e-16
[2,] 1.665335e-16 1.000000e+00
```

The V matrix's row vectors are orthonormal: zero or one. Matrix form:

```
s$v%*%t(s$v)

     [,1] [,2]
[1,]    1    0
[2,]    0    1
```

Row number is the PC number, or mode number. Column number is time,
year.

4.2: Covariance matrix, eigenvalues and eigenvectors

Construct a covariance matrix C

```
C<-boxnew %*% t(boxnew)
C
          [,1]       [,2]      [,3]      [,4]
[1,] 0.4114785 0.18679652 0.3538192 0.3945115
[2,] 0.1867965 0.08496229 0.1625373 0.1814484
[3,] 0.3538192 0.16253734 0.3267139 0.3668445
[4,] 0.3945115 0.18144844 0.3668445 0.4121751
```

Compute the eigenvalues and eigenvectors of C

```
s1<-eigen(C)
s1

$values
[1] 1.213239e+00 2.209039e-02 1.323093e-16 1.517883e-17

$vectors
          [,1]       [,2]        [,3]       [,4]
[1,] 0.5738814  0.7342785 -0.36262267  0.0000000
[2,] 0.2624958  0.2485962  0.91880715  0.1583961
[3,] 0.5166209 -0.3625604  0.08344434 -0.7711614
[4,] 0.5786700 -0.5172862 -0.13166347  0.6166205

# The first two eigenvalues $values should be the same the SVD eigenvalues,
# and the 3rd and 4th are zero. Compare these with the svd results

svd(boxnew)

$d
[1] 1.1014714 0.1486284

$u
           [,1]       [,2]
[1,] -0.5738814  0.7342785
[2,] -0.2624958  0.2485962
[3,] -0.5166209 -0.3625604
[4,] -0.5786700 -0.5172862

$v
            [,1]        [,2]
[1,]  0.08426253 -0.99644359
[2,] -0.99644359 -0.08426253

# The first two column eigen vectors, i.e., the first two EOFs, are the
# same as the first two svd U vectors.
# The C eigenvalues 1.213239e+00 2.209039e-02 are from the square of
# SVD 1.1014714 0.1486284, because the SVD $d are square root of covariance
# matrix C's eigenvalues $values: 1.1014714^2 =1.213239,
# 0.1486284^2=0.0220904.
#The C eigenvalues can be made accurate using a matrix transpose, i.e.,
# the time-space exchange. Sam Shen developed a detailed theory on this
# method to remove computational errors and reduce uncertainties in data
# reconstruction.

#(a) N=5 boxes, Y=6 years. Now Y>N, the C matrix is of full rank and can be
# inverted.
```

```
pmat<- read.csv("~/Desktop/MyDocs/teach/SIOC290-ClimateMath/Hwk-and-
Exams/Hwk4.1/PrcpRecon.csv")

p5<-pmat[,c(1,2,110,111,112,113,114,115)]
box5<-p5[rownames(p5)%in%c(1459,1460,1461,1462, 1463),  ]
box5
   NaN. NaN..1     X2007     X2008     X2009     X2010      X2011  NaN..2
1459 17.5  242.5 -0.552080 -0.501340 -0.098233 -0.858890 -0.5037900 1.79130
1460 22.5  242.5 -0.082871 -0.086863 -0.254630 -0.359890 -0.1620100 0.62067
1461 27.5  242.5 -0.072606 -0.011015 -0.104730 -0.202700 -0.0611800 0.28499
1462 32.5  242.5 -0.090514 -0.021438  0.053968 -0.269600  0.0057461 0.57156
1463 37.5  242.5 -0.100250  0.180460  0.038779  0.021559  0.0229020 0.64160

box5 <-box5[,3:8]
box5
       X2007     X2008     X2009     X2010      X2011  NaN..2
1459 -0.552080 -0.501340 -0.098233 -0.858890 -0.5037900 1.79130
1460 -0.082871 -0.086863 -0.254630 -0.359890 -0.1620100 0.62067
1461 -0.072606 -0.011015 -0.104730 -0.202700 -0.0611800 0.28499
1462 -0.090514 -0.021438  0.053968 -0.269600  0.0057461 0.57156
1463 -0.100250  0.180460  0.038779  0.021559  0.0229020 0.64160

box56<-matrix(seq(30),5)
for(i in 1:6) box56[,i]=box5[,i]
box56
         [,1]      [,2]      [,3]      [,4]       [,5]    [,6]
[1,] -0.552080 -0.501340 -0.098233 -0.858890 -0.5037900 1.79130
[2,] -0.082871 -0.086863 -0.254630 -0.359890 -0.1620100 0.62067
[3,] -0.072606 -0.011015 -0.104730 -0.202700 -0.0611800 0.28499
[4,] -0.090514 -0.021438  0.053968 -0.269600  0.0057461 0.57156
[5,] -0.100250  0.180460  0.038779  0.021559  0.0229020 0.64160

# Now we have a clean 5 boxes and 6 years data matrix to do SVD,
# covariance matrix, eigenvalues and eigenvectors.

# Covariance matrix
C<- (1/6)*box56%*%t(box56)
C
         [,1]       [,2]       [,3]       [,4]       [,5]
[1,] 0.7943393 0.26947392 0.12855266 0.21798577 0.18005138
[2,] 0.2694739 0.10337475 0.04889789 0.07441114 0.06158515
[3,] 0.1285527 0.04889789 0.02373515 0.03639020 0.02971802
[4,] 0.2179858 0.07441114 0.03639020 0.06849382 0.06138839
[5,] 0.1800514 0.06158515 0.02971802 0.06138839 0.07612659
```

```
# (b) Inverse matrix of C
B<-solve(C)
B
        [,1]      [,2]      [,3]      [,4]      [,5]
[1,]  21.08777  -48.99043   49.98343  -49.69095  10.31492
[2,] -48.99043  505.16593 -962.90545  159.47688 -45.50727
[3,]  49.98343 -962.90545 2200.46670 -376.49208 105.34855
[4,] -49.69095  159.47688 -376.49208  281.63827 -91.62641
[5,]  10.31492  -45.50727  105.34855  -91.62641  58.31607

# Verify the inverse matrix B
B%*%C

             [,1]          [,2]          [,3]          [,4]          [,5]
[1,]  1.000000e+00 -6.994405e-15 -3.219647e-15 -5.329071e-15 -3.441691e-15
[2,]  8.171241e-14  1.000000e+00  1.398881e-14  3.241851e-14  2.620126e-14
[3,] -1.030287e-13 -4.796163e-14  1.000000e+00 -4.263256e-14 -4.618528e-14
[4,]  3.552714e-15  4.440892e-15  3.108624e-15  1.000000e+00  3.552714e-15
[5,]  0.000000e+00  8.881784e-16  2.220446e-16  0.000000e+00  1.000000e+00
# Yes this is an identity matrix. B is the inverse of C. Of course, C is the inverse
# of B.

#Verify eigenvalues and eigenvectors

eigen(C)

$values
[1] 1.0122919098 0.0367593796 0.0125878362 0.0040618975 0.0003686159

$vectors
          [,1]        [,2]        [,3]       [,4]        [,5]
[1,] 0.8835358 -0.25549001  0.35685401  0.1612408  0.02731287
[2,] 0.3043048 -0.11235432 -0.84878297  0.1016091 -0.40499157
[3,] 0.1453906 -0.03717329 -0.38862873 -0.1474359  0.89705635
[4,] 0.2473754  0.29440145  0.02117979 -0.9074722 -0.16786576
[5,] 0.2107846  0.91326190 -0.02723618  0.3441423  0.04844401

#The last two very small eigenvalues at the end means that the C matrix
# is again not invertible, box56 is not full rank, and hence the eigenvalues will
# still have errors. To eliminate the errors, we have to use more years of data,
# say 20 years.

svd(box56)

$d
[1] 2.46449822 0.46963420 0.27482179 0.15611337 0.04702867
```

```
$u
          [,1]        [,2]        [,3]       [,4]        [,5]
[1,] 0.8835358  0.25549001  0.35685401  0.1612408  0.02731287
[2,] 0.3043048  0.11235432 -0.84878297  0.1016091 -0.40499157
[3,] 0.1453906  0.03717329 -0.38862873 -0.1474359  0.89705635
[4,] 0.2473754 -0.29440145  0.02117979 -0.9074722 -0.16786576
[5,] 0.2107846 -0.91326190 -0.02723618  0.3441423  0.04844401

$v
            [,1]        [,2]        [,3]        [,4]        [,5]
[1,] -0.23009909 -0.07422561 -0.35529326 -0.25042627 -0.77209962
[2,] -0.17782573 -0.63187929 -0.38667131 -0.04150974  0.50916999
[3,] -0.06410225 -0.23188927  0.80728203 -0.39650603 -0.01466130
[4,] -0.38952962 -0.44231537  0.25997921  0.68477722 -0.28150150
[5,] -0.20168936 -0.36581085 -0.06911354 -0.55091985 -0.06133029
[6,]  0.84788643 -0.46041940 -0.01348533  0.07690305 -0.24776733
```