Homework 2

Jing Leng

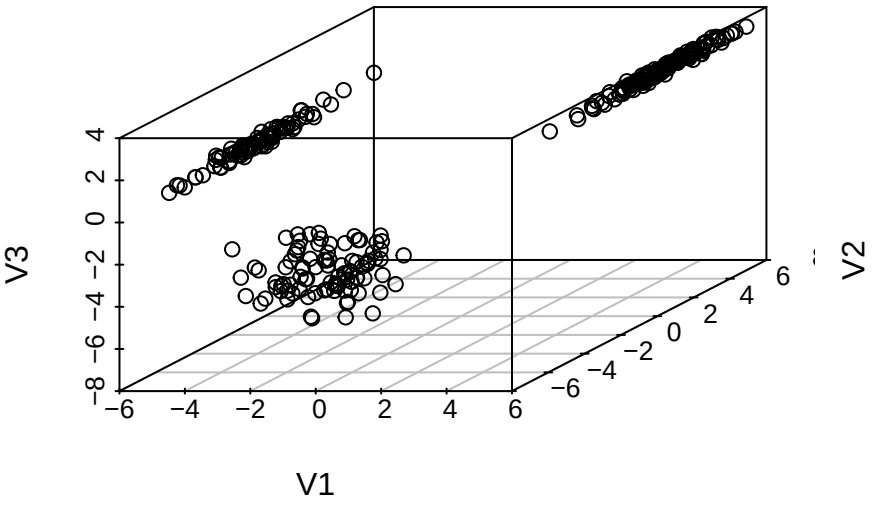
September 24, 2014

```
# STATS 406 HW2
setwd("~/Documents/GoBlue/F14/F14R")
mix <- read.table("mixgauss.dat")
head(mix)
dim(mix)
library(rgl)

# 1
## a
plot3d(mix, type = 'p', size = 10)
## b
pairs(mix)
## c
c1 <- c(-2, -2, -4) # observed centers
c2 <- c(-4, -2, 2)
c3 <- c(4, 6, 2)

centers <- rbind(c1, c2, c3)

classify <- function(x) {
  dis <- apply(centers, 1, function(y) sqrt(sum((x - y)^2)));
  which(dis == min(dis))
}
label <- apply(mix, 1, classify )

mix$class <- label
plot3d(mix, type = 'p', col = mix$class + 1, size = 10)
plot3d(centers, type = 'p', col = 'blue', size = 20, add = T)
pairs(mix[, 1:3], col = mix$class + 1, pch = ifelse(mix$class < 3, mix$class -1, mix$class + 2))

p1 = sum(mix$class == 1)/dim(mix)[1]
p2 = sum(mix$class == 2)/dim(mix)[1]
p3 = 1 - p1 -p2
p1; p2; p3

# 2
## a
lai <- read.csv('LAI.csv', header = F)
agg <- apply(lai, 1, sum)
which(agg == max(agg))

maxmonths <- sapply(0:5, function(i) which.max(agg[(12*i+1):(12*i+12)])) #use sapply to get the months w
maxmonths
minmonths <- sapply(0:5, function(i) which.min(agg[(12*i+1):(12*i+12)])) #use sapply to get the months w
minmonths

yearcol = matrix(agg,nrow=12)
#now with the above, using apply(), how do you find the yearly aggregated LAI over the continent?
 yragg = apply(yearcol, 2, sum) #along which axis?(2-column), by which function?(sum)
```

```
maxyear <- which(yragg == max(yragg))
minyear <- which(yragg == min(yragg))

lai<-as.matrix(lai)
maxt<-dim(lai)[1] # 72 time points
LAI<-sapply(1:maxt,function(i) matrix(lai[i,],nrow=120,ncol=60,byrow=T),simplify="array")

## b
LAImean <- apply(LAI, 1:2, mean)
LAIsd <- apply(LAI, 1:2, sd)
dim(LAI)
dim(LAImean)


## c
image.plot(LAImean, horizontal = T)
image.plot(LAIsd, horizontal = T)

image.plot(LAI[80:90,25:45,7],horizontal=T) ## used to determine a location within Michigan.
image.plot(LAI[48:80,10:30,7],horizontal=T) ## used to determine a location within Michigan.
image.plot(LAI[,,7],horizontal=T) ## used to determine a location within Michigan.

# Michigan [87, 33]
# Arizona [57,28]
plot(LAI[87, 33,], type = 'l', ylim = c(0, 4), xlab = 'month', ylab = 'lai')
par(new = T)
plot(LAI[57, 28,], type = 'l', ylim = c(0, 4), xlab = '', ylab = '', lty = 2)
## d
maxloc <- which(LAImean == max(LAImean))
minloc <- which(LAImean == min(LAImean[LAImean != 0]))
maxloc <- c(maxloc%/%120, maxloc%%120)
minloc <- c(minloc%/%120, minloc%%120)
maxloc; minloc;

## e
leftlim <- (2005-2000)*12+1
rightlim <- (2005-2000)*12+12
par(ask = T)
for (i in leftlim:rightlim) {
  image.plot(LAI[65:100,15:35,i],horizontal=T)
}
```

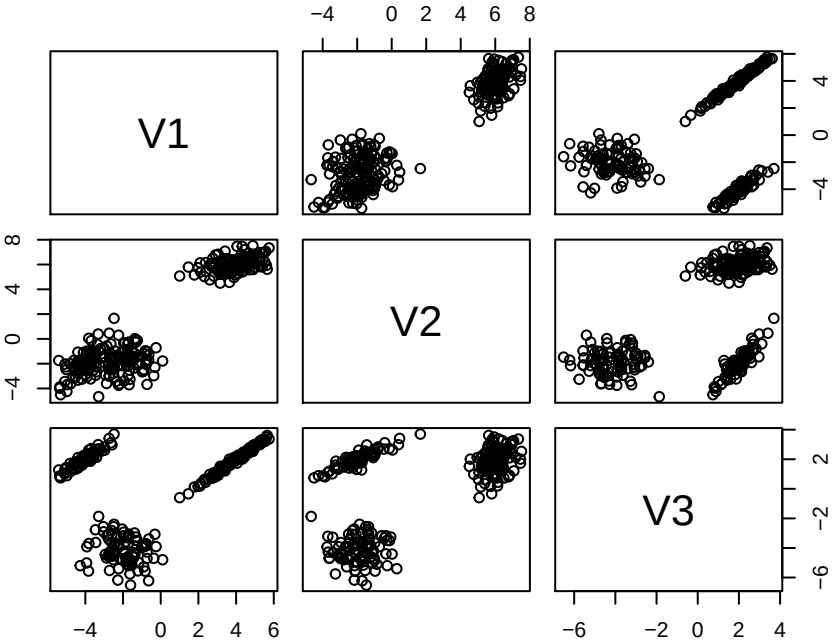
1

a).



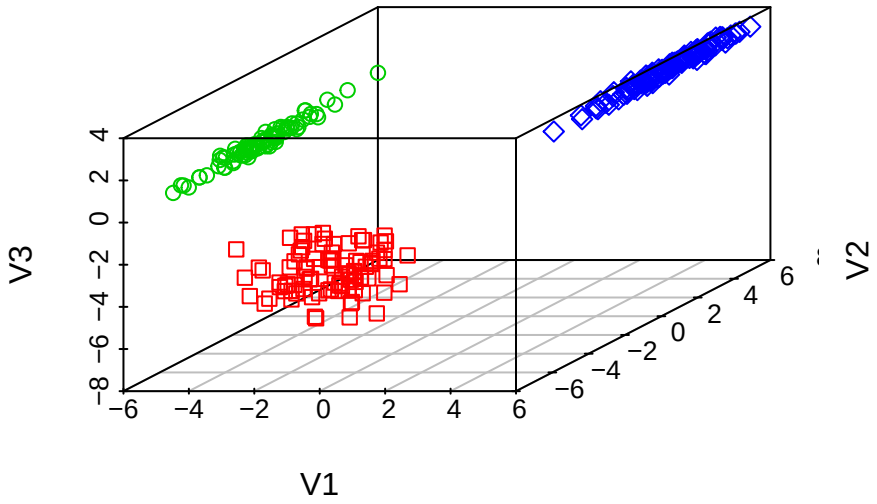
Yes, there are three clusters of data points.

b).

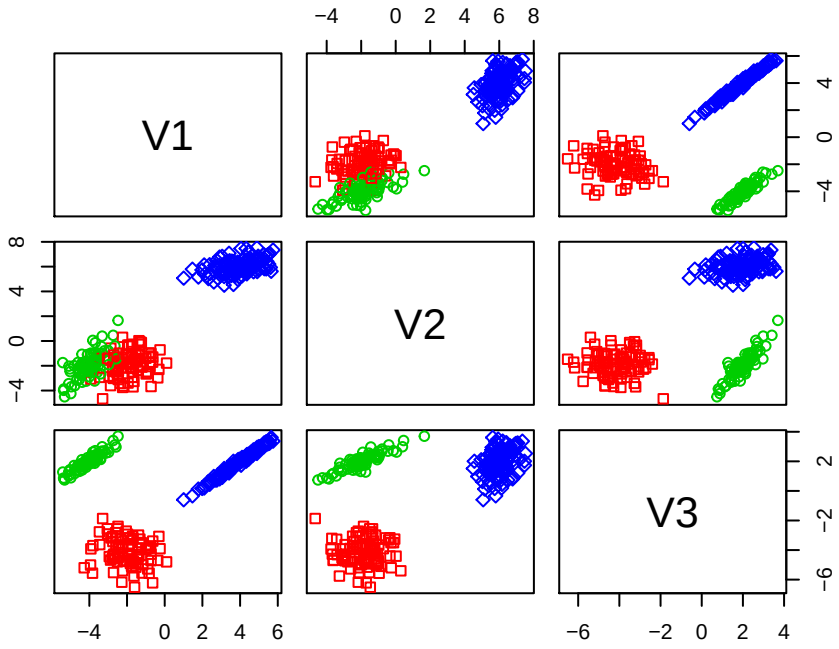


The coordinates for the three cluster centers are estimated to be: (-2, -2, -4), (-4, -2, 2) and (4, 6, 2)

c).



3-d



2-d

The proportion of the three clusters are: 0.3067, 0.2833, 0.41.

2

a)

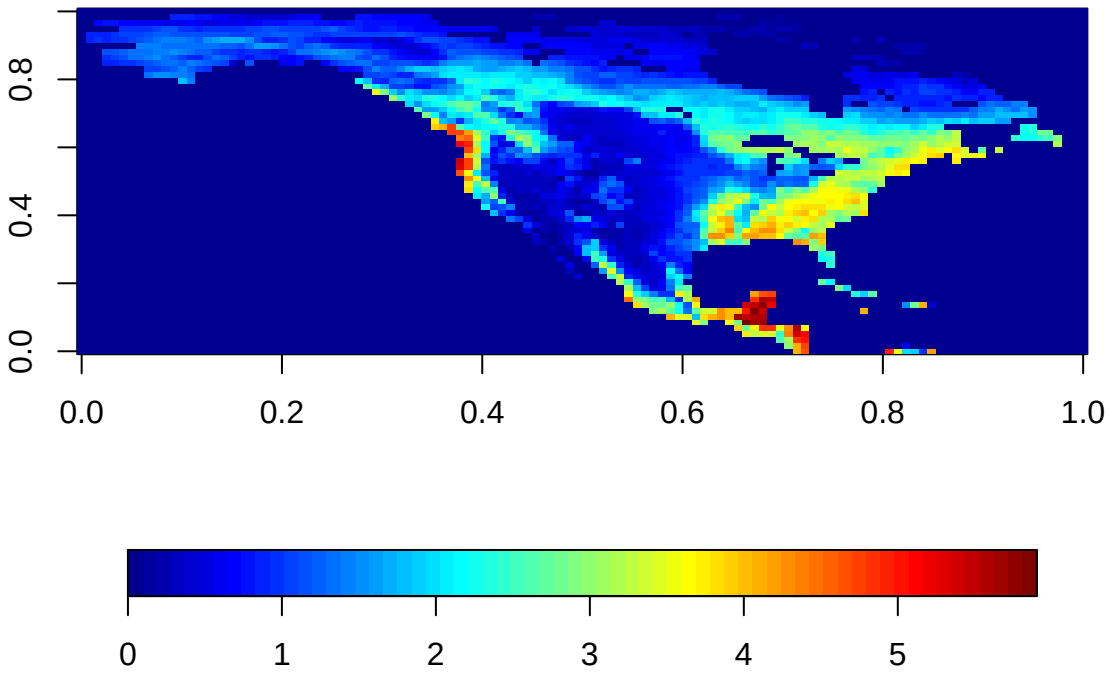
For each year from 2000 to 2005, the month with maximum LAI is (respectively): 7, 7, 7, 7, 7, 7. The month with minimum LAI is (respectively): 12, 1, 2, 2, 2, 2. The year with maximun aggregated LAI is: 2004, with minimum: 2003.

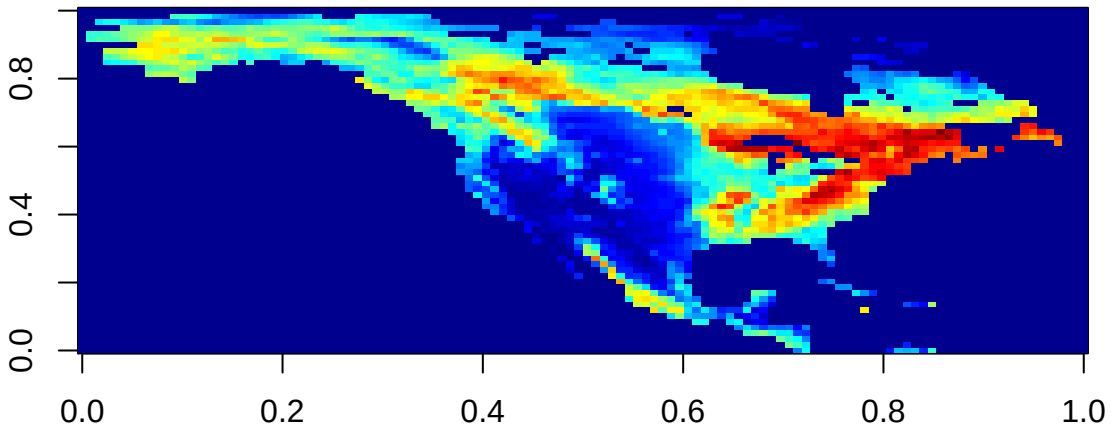
b)

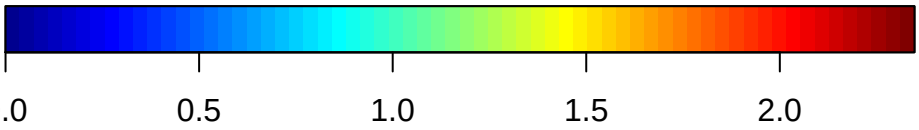
mean

```
## Loading required package: spam
```

```
## Loading required package: grid
## Spam version 1.0-1 (2014-09-09) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
##
## Attaching package: 'spam'
##
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
##
## Loading required package: maps
```

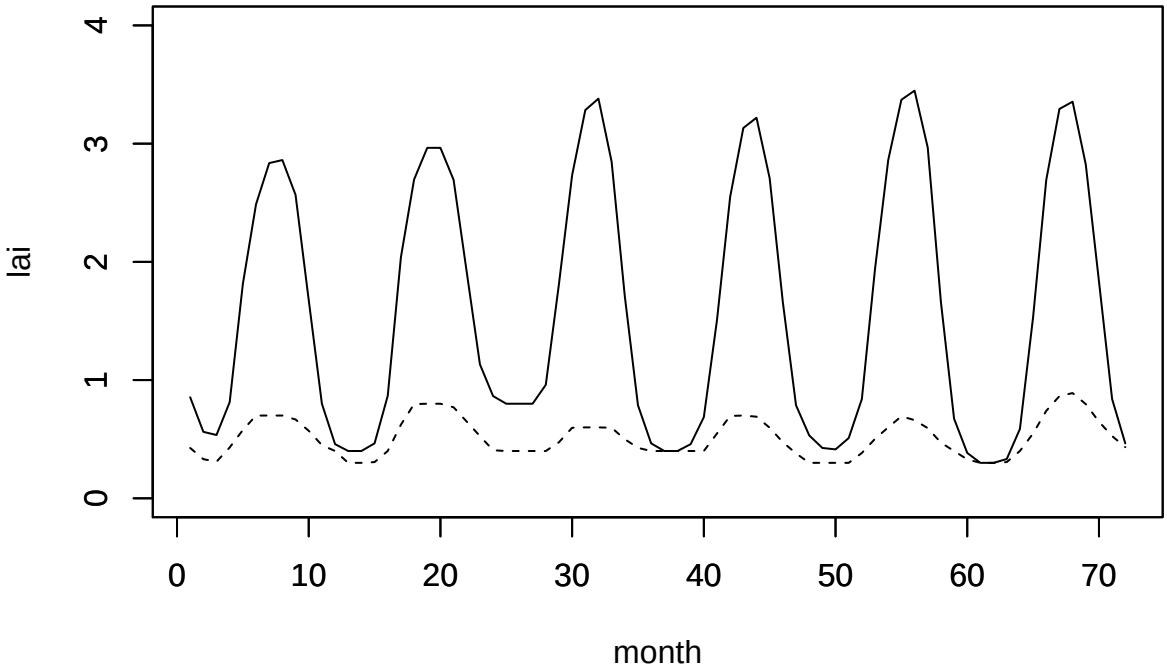






sd

c)

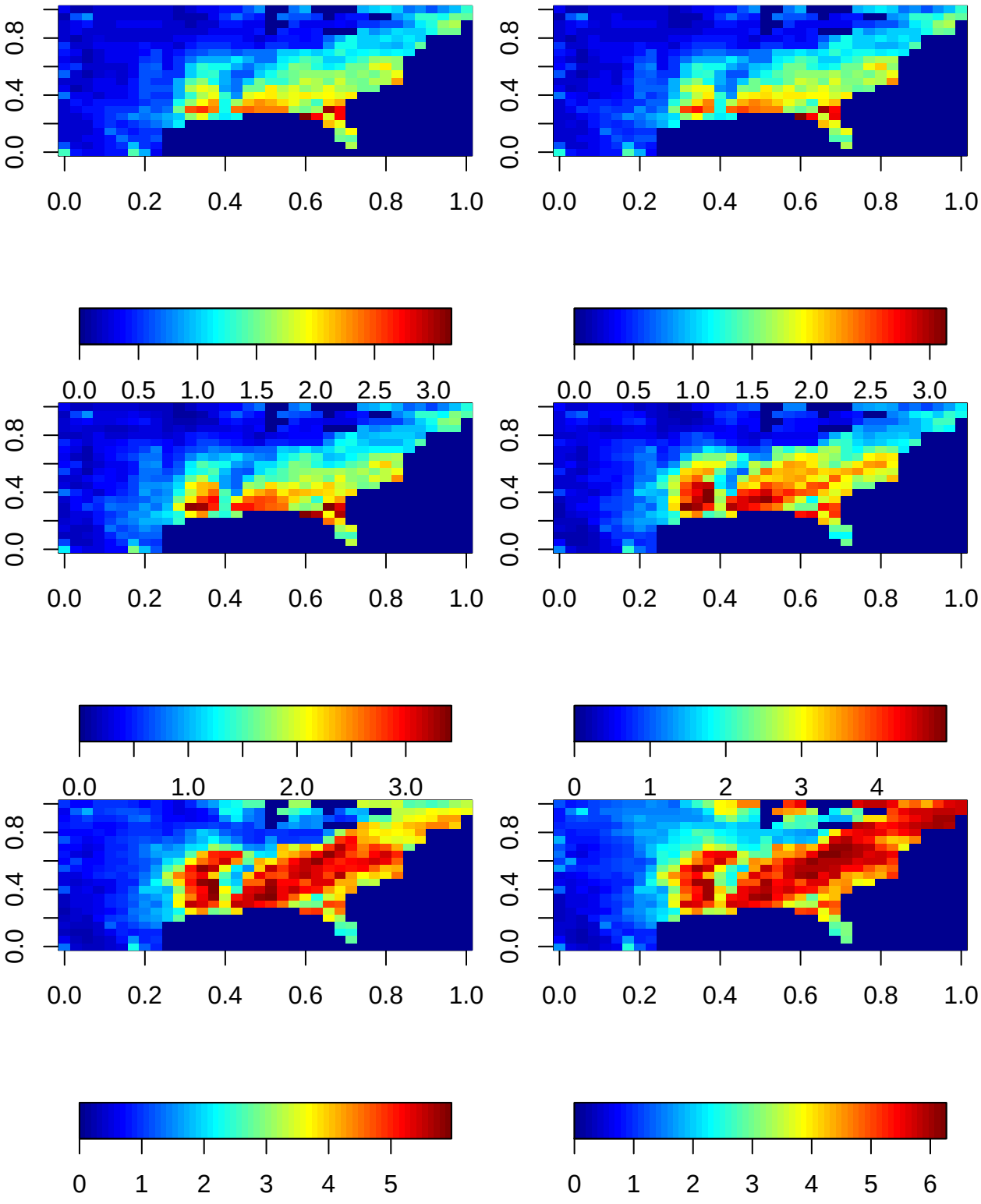


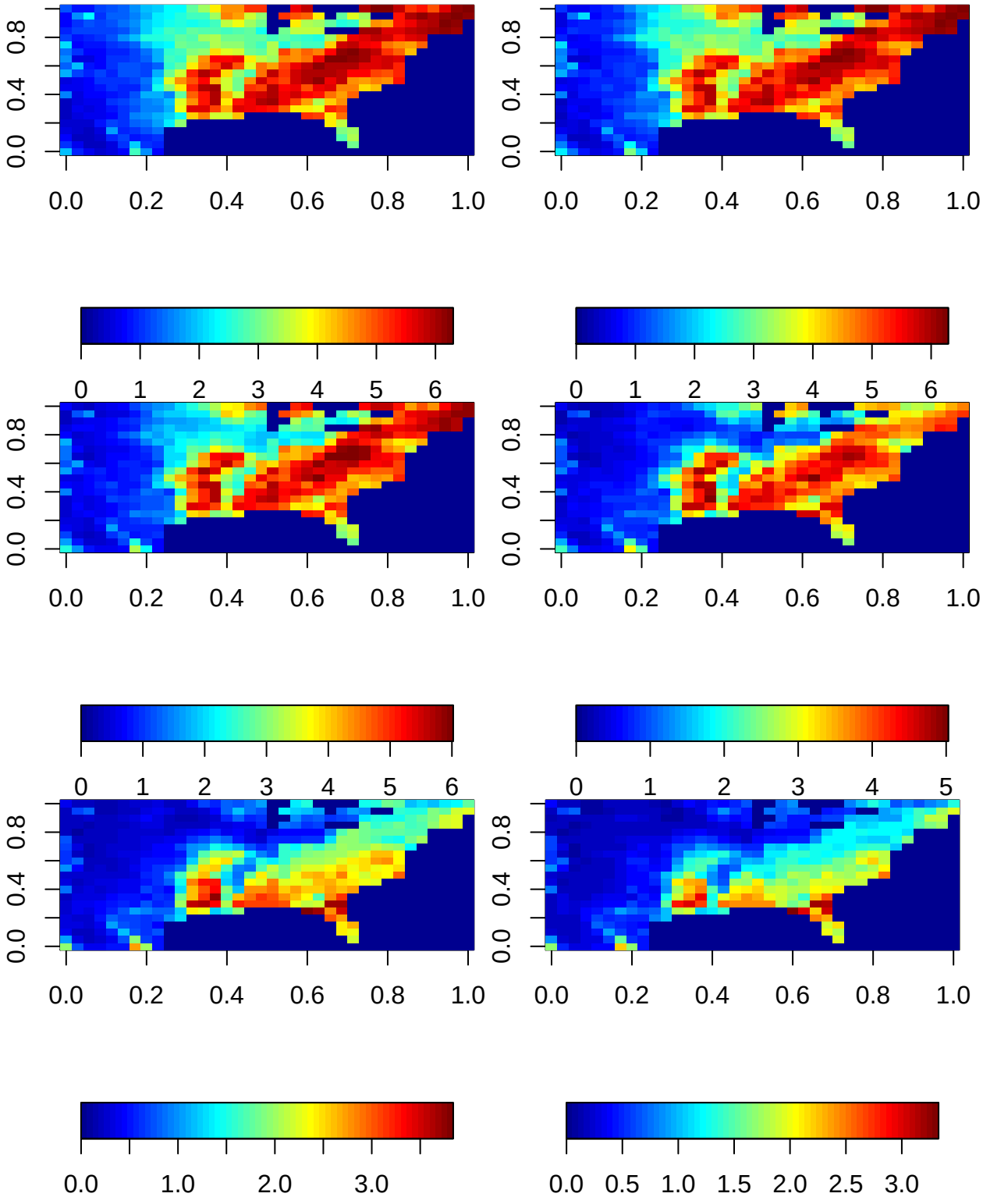
SE Michigan (location: 87, 33) has higher LAI than Arizona (location: 57, 28).

d)

Most leafy: 7, 81.
Least leafy: 26, 60.

e)


0.0 0.4 0.8
0.0 0.2 0.4 0.6 0.8 1.0
0.0 0.5 1.0 1.5 2.0 2.5 3.0
0.0 1.0 2.0 3.0
0 1 2 3 4
0 1 2 3 4 5
0 1 2 3 4 5 6


0.0 0.4 0.8
0.0 0.2 0.4 0.6 0.8 1.0
0 1 2 3 4 5 6
0.0 1.0 2.0 3.0
0.0 0.5 1.0 1.5 2.0 2.5 3.0