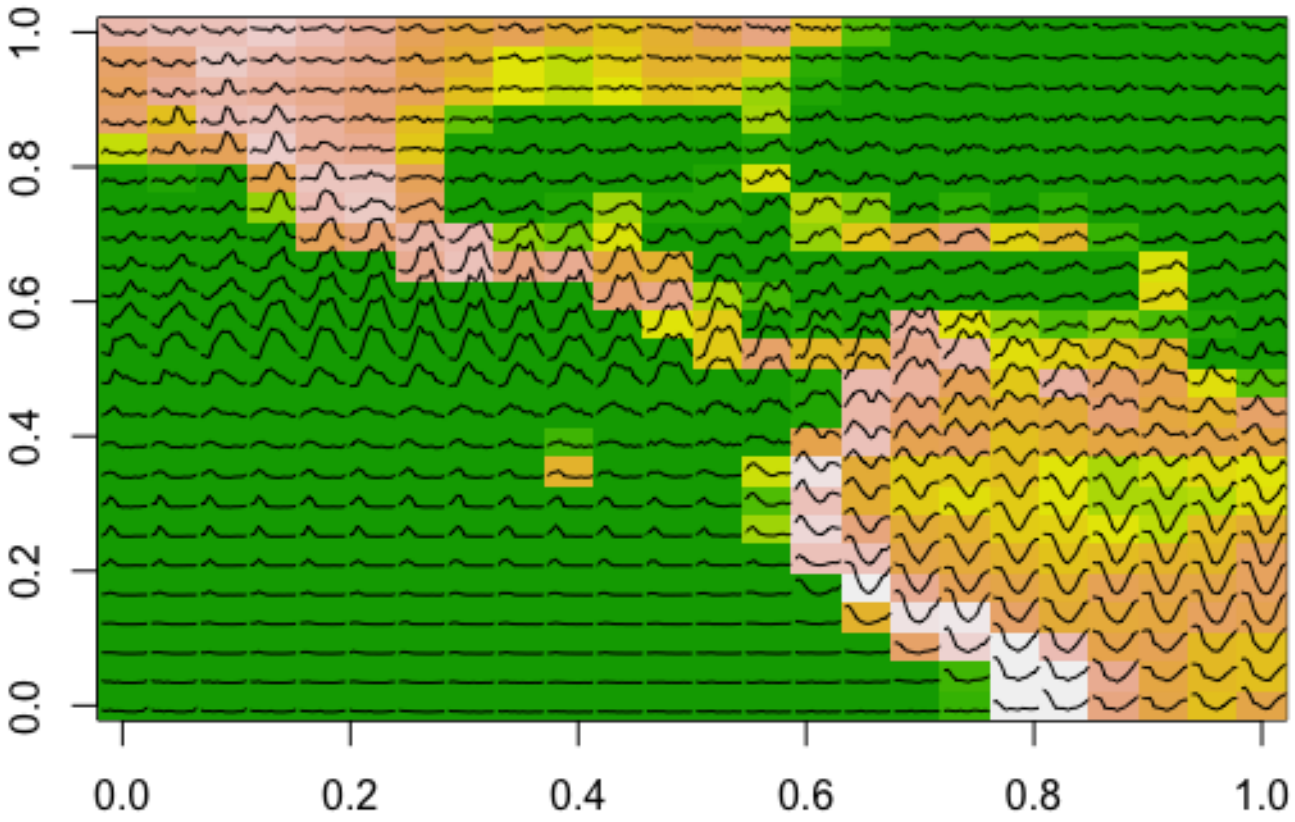
Homework 4

Yiwei He (yh9vhg), Da Lin (dl2de), Ziyue Jin (zj5qj)



Appendix

Build graph corresponding to 24 x 24 grid

```
ele <- read.table("~/Desktop/Homework 4/NASA/Files/elevation.dat",header=TRUE)
dim(ele)
colnames(ele)
rownames(ele)

elemat <- as.matrix(ele)
n <- nrow(elemat)
image(t(elemat[n:1,]))
hist(as.numeric(elemat))
image(log(t(elemat[n:1,])+1))

Lon <- colnames(ele)
Lon <- unlist(lapply(Lon,function(x)gsub("X.","",x)))
Lon
names(ele) <- Lon

ele.Lon <- as.numeric(Lon)
ele.Lon
ele.Lat <- as.numeric(rownames(ele))
ele.Lat
```

```
metric.Lon <- scan("~/Desktop/Homework 4/NASA/Files/Lon.txt",what="",sep="\t")
metric.Lon <- unlist(lapply(metric.Lon,function(x) gsub("W","",x)))
metric.Lon <- as.numeric(metric.Lon)
metric.Lon

metric.Lat <- scan("~/Desktop/Homework 4/NASA/Files/Lat.txt",what="",sep="\n")
metric.Lat <- unlist(lapply(metric.Lat,function(x){
   x <- gsub("N","",x)
   x <- gsub("S","",x)
})                          )
metric.Lat <- as.numeric(metric.Lat)
metric.Lat[16:24] <- -metric.Lat[16:24]
metric.Lat

######## use the elevation data set to find approximate elevation for the 24 x 24 grid
## (approximation is needed because the positions of the
## measurements are not the same as the elevation data )

### use closet location for approximation
approx.elevation <- matrix(0,24,24)
approx.elevation
for(i in 1:24){
```

```
    lat <- metric.Lat[i]
    dist.seq.lat <- abs(ele.Lat-lat)
    lat.index <- which.min(dist.seq.lat)
    Latitude=lat.index

    for(j in 1:24){
        lon <- metric.Lon[j]
        dist.seq.lon <- abs(ele.Lon-lon)
        lon.index <- which.min(dist.seq.lon)
        Longitude=lon.index
        approx.elevation[i,j] <- ele[lat.index,lon.index]
    }
}

image(log(approx.elevation+1))
image(t(log(approx.elevation+1)[24:1,]),xaxt="n",yaxt="n")

points(x=c(0.5,0.6),y=c(0.6,0.5),pch=16,col="red")
lines(x=c(0.5,0.6),y=c(0.6,0.5),lwd=2,col="blue")
x.seq <- runif(10)
y.seq <- runif(10)
points(x=x.seq,y=y.seq,pch=1,col="red")
```

```
x.seq <- 1:100

y.seq <- sin(x.seq)

plot(x.seq,y.seq,type="b")


x.scaled <- (x.seq-50)/101*(1/23)

y.scaled <- (y.seq)/2*(1/23)


plot(x.scaled,y.scaled,"b")


x.new.scaled <- x.scaled + 3/23

y.new.scaled <- y.scaled + 23/23

image(t(log(approx.elevation+1)[24:1,]))

points(x.new.scaled,y.new.scaled,pch=20,lwd=0.6,cex=0.3)


elevation <- approx.elevation

GridTimeSeries <- list()

#### remark: bad notation. Do not use t, because it is overwriting the matrix transpose


for(tt in 1:72){

pressure <- read.table(paste("~/Desktop/Homework
4/NASA/Files/pressure",tt,".txt",sep=""),header=FALSE,skip=7,na.strings="....")

pressure <- as.numeric(as.matrix(pressure[,-(1:3)]))
```

```
ozone <- read.table(paste("~/Desktop/Homework 4/NASA/Files/ozone",tt,".txt",sep=""),header=FALSE,skip=7,na.strings="....")

ozone <- as.numeric(as.matrix(ozone[,-(1:3)]))


surftemp <- read.table(paste("~/Desktop/Homework 4/NASA/Files/surftemp",tt,".txt",sep=""),header=FALSE,skip=7,na.strings="....")

surftemp <- as.numeric(as.matrix(surftemp[,-(1:3)]))


cloudhigh <- read.table(paste("~/Desktop/Homework 4/NASA/Files/cloudhigh",tt,".txt",sep=""),header=FALSE,skip=7,na.strings="....")

cloudhigh <- as.numeric(as.matrix(cloudhigh[,-(1:3)]))



cloudmid <-  read.table(paste("~/Desktop/Homework 4/NASA/Files/cloudmid",tt,".txt",sep=""),header=FALSE,skip=7,na.strings="....")

cloudmid <- as.numeric(as.matrix(cloudmid[,-(1:3)]))


cloudlow <- read.table(paste("~/Desktop/Homework 4/NASA/Files/cloudlow",tt,".txt",sep=""),header=FALSE,skip=7,na.strings="....")

cloudlow <- as.numeric(as.matrix(cloudlow[,-(1:3)]))


temperature <- read.table(paste("~/Desktop/Homework 4/NASA/Files/temperature",tt,".txt",sep=""),header=FALSE,skip=7,na.strings="....")

temperature <- as.numeric(as.matrix(temperature[,-(1:3)]))


#load("ApproxElevation.Rda")
```

```
elevation <- as.numeric(elevation)

X <-
data.frame(pressure=pressure,ozone=ozone,surftemp=surftemp,cloudhigh=cloudhigh,cl
oudmid=cloudmid,cloudlow=cloudlow,temperature=temperature,elevation=elevation)

with.missing <- sum(is.na(X)) > 0

GridTimeSeries[[tt]] <- list(X=X,with.missing=with.missing)

}

GridTimeSeries[[2]]

dt <- GridTimeSeries[[15]]$X

dim(dt)

head(GridTimeSeries)

save(GridTimeSeries,file="NASAGridTimeSeries.Rda")


##################################data

Lon <- data.frame(matrix(ncol = 1,nrow = 576))

colnames(Lon) <- c('lon')

count <- 1

for (i in 1:24) {

    item <- metric.Lon[i]

    for (j in 1:24) {

        Lon$lon[count] <- item

        count <- count + 1

    }

}
```

```
Lat <- data.frame(matrix(ncol = 1,nrow = 576))
colnames(Lat) <- c('lat')
lat_count <- 1
for (i in 1:24) {
   for (j in 1:24) {
      thing <- metric.Lat[j]
      Lat$lat[lat_count] <- thing
      lat_count <- lat_count + 1
   }
}
lon_lat <- cbind(Lon,Lat)

## January
jan <- c(1,13,25,37,49,61)
January <- data.frame(matrix(ncol = 2, nrow = 0))
head(January)
col_name <- c("X.cloud","year")
colnames(January) <- col_name
for (i in jan){
   target_frame <- as.data.frame(GridTimeSeries[[i]][1])
   target_col <- target_frame['X.cloudhigh']
   if( i == 1) {
```

```
        target_col['year'] <- 1
    }
    else if (i == 13) {
        target_col['year'] <- 2
    }
    else if (i == 25) {
        target_col['year'] <- 3
    }
    else if (i == 37) {
        target_col['year'] <- 4
    }
    else if (i == 49) {
        target_col['year'] <- 5
    }
    else {
        target_col['year'] <- 6
    }
    January <- rbind(January,target_col)
}
January_avg <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(January_avg) <- c("avg_cloudhigh")
January_avg
dim(January_avg)
```

```
for (x in 1:576){
    counter <- x
    sum_val <- 0
    for (y in 1:6) {
        sum_val <- sum_val + January$X.cloudhigh[counter]
        counter <- counter + 576
    }
    jan_avg_val <- sum_val/6
    January_avg <- rbind(January_avg,data.frame(jan_avg_val))
}


## FEB
feb <- c(2,14,26,38,50,62)
February <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(February) <- col_name
for (i in feb){
    target_frame <- as.data.frame(GridTimeSeries[[i]][1])
    target_col <- target_frame['X.cloudhigh']
    if( i == 2) {
        target_col['year'] <- 1
    }
    else if (i == 14) {
        target_col['year'] <- 2
```

```
    }
    else if (i == 26) {
        target_col['year'] <- 3
    }
    else if (i == 38) {
        target_col['year'] <- 4
    }
    else if (i == 50) {
        target_col['year'] <- 5
    }
    else {
        target_col['year'] <- 6
    }
    February <- rbind(February,target_col)
}
February_avg <- data.frame(matrix(ncol = 1, nrow = 0))
colnames(February_avg) <- c("feb_avg_cloudhigh")
for (x in 1:576){
    counter <- x
    sum_val <- 0
    for (y in 1:6) {
        sum_val <- sum_val + February$X.cloudhigh[counter]
        counter <- counter + 576
```

```
    }
    feb_avg_val <- sum_val/6
    February_avg <- rbind(February_avg,data.frame(feb_avg_val))
}

# March
mar <- c(3,15,27,39,51,63)
March <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(March) <- col_name
for (i in mar){
    target_frame <- as.data.frame(GridTimeSeries[[i]][1])
    target_col <- target_frame['X.cloudhigh']
    if( i == 3) {
        target_col['year'] <- 1
    }
    else if (i == 15) {
        target_col['year'] <- 2
    }
    else if (i == 27) {
        target_col['year'] <- 3
    }
    else if (i == 39) {
        target_col['year'] <- 4
```

```
    }
    else if (i == 51) {
        target_col['year'] <- 5
    }
    else {
        target_col['year'] <- 6
    }
    March <- rbind(March,target_col)
}

March_avg <- data.frame(matrix(ncol = 1, nrow = 0))
colnames(March_avg) <- c("mar_avg_cloudhigh")
for (x in 1:576){
    counter <- x
    sum_val <- 0
    for (y in 1:6) {
        sum_val <- sum_val + March$X.cloudhigh[counter]
        counter <- counter + 576
    }
    mar_avg_val <- sum_val/6
    March_avg <- rbind(March_avg,data.frame(mar_avg_val))
}
```

```
# April
apr <- c(4,16,28,40,52,64)
April <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(April) <- col_name
for (i in apr){
    target_frame <- as.data.frame(GridTimeSeries[[i]][1])
    target_col <- target_frame['X.cloudhigh']
    if( i == 4) {
        target_col['year'] <- 1
    }
    else if (i == 16) {
        target_col['year'] <- 2
    }
    else if (i == 28) {
        target_col['year'] <- 3
    }
    else if (i == 40) {
        target_col['year'] <- 4
    }
    else if (i == 52) {
        target_col['year'] <- 5
    }
    else {
```

```
        target_col['year'] <- 6
    }
    April <- rbind(April,target_col)
}

Apr_avg <- data.frame(matrix(ncol = 1, nrow = 0))
colnames(Apr_avg) <- c("apr_avg_cloudhigh")
for (x in 1:576){
    counter <- x
    sum_val <- 0
    for (y in 1:6) {
        sum_val <- sum_val + April$X.cloudhigh[counter]
        counter <- counter + 576
    }
    apr_avg_val <- sum_val/6
    Apr_avg <- rbind(Apr_avg,data.frame(apr_avg_val))
}

# May
ma <- c(5,17,29,41,53,65)
May <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(May) <- col_name
for (i in ma){
```

```
    target_frame <- as.data.frame(GridTimeSeries[[i]][1])
    target_col <- target_frame['X.cloudhigh']
    if( i == 5) {
        target_col['year'] <- 1
    }
    else if (i == 17) {
        target_col['year'] <- 2
    }
    else if (i == 29) {
        target_col['year'] <- 3
    }
    else if (i == 41) {
        target_col['year'] <- 4
    }
    else if (i == 53) {
        target_col['year'] <- 5
    }
    else {
        target_col['year'] <- 6
    }
    May <- rbind(May,target_col)
}
```

```
May_avg <- data.frame(matrix(ncol = 1, nrow = 0))
colnames(May_avg) <- c("may_avg_cloudhigh")
for (x in 1:576){
    counter <- x
    sum_val <- 0
    for (y in 1:6) {
        sum_val <- sum_val + May$X.cloudhigh[counter]
        counter <- counter + 576
    }
    may_avg_val <- sum_val/6
    May_avg <- rbind(May_avg,data.frame(may_avg_val))
}

# June
ju <- c(6,18,30,42,54,66)
June <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(June) <- col_name
for (i in ju){
    target_frame <- as.data.frame(GridTimeSeries[[i]][1])
    target_col <- target_frame['X.cloudhigh']
    if( i == 6) {
        target_col['year'] <- 1
    }
```

```
    else if (i == 18) {
        target_col['year'] <- 2
    }
    else if (i == 30) {
        target_col['year'] <- 3
    }
    else if (i == 42) {
        target_col['year'] <- 4
    }
    else if (i == 54) {
        target_col['year'] <- 5
    }
    else {
        target_col['year'] <- 6
    }
    June <- rbind(June,target_col)
}

June_avg <- data.frame(matrix(ncol = 1, nrow = 0))
colnames(June_avg) <- c("jun_avg_cloudhigh")
for (x in 1:576){
    counter <- x
    sum_val <- 0
```

```
    for (y in 1:6) {
        sum_val <- sum_val + June$X.cloudhigh[counter]
        counter <- counter + 576
    }
    jun_avg_val <- sum_val/6
    June_avg <- rbind(June_avg,data.frame(jun_avg_val))
}


#July
july <- c(7,19,31,43,55,67)
July <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(July) <- col_name
for (i in july){
    target_frame <- as.data.frame(GridTimeSeries[[i]][1])
    target_col <- target_frame['X.cloudhigh']
    if( i == 7) {
        target_col['year'] <- 1
    }
    else if (i == 19) {
        target_col['year'] <- 2
    }
    else if (i == 31) {
        target_col['year'] <- 3
```

```
    }
    else if (i == 43) {
        target_col['year'] <- 4
    }
    else if (i == 55) {
        target_col['year'] <- 5
    }
    else {
        target_col['year'] <- 6
    }
    July <- rbind(July,target_col)
}

July_avg <- data.frame(matrix(ncol = 1, nrow = 0))
colnames(July_avg) <- c("july_avg_cloudhigh")
for (x in 1:576){
    counter <- x
    sum_val <- 0
    for (y in 1:6) {
        sum_val <- sum_val + July$X.cloudhigh[counter]
        counter <- counter + 576
    }
    july_avg_val <- sum_val/6
```

```
    July_avg <- rbind(July_avg,data.frame(july_avg_val))
}


#August
august <- c(8,20,32,44,56,68)
August <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(August) <- col_name
for (i in august){
    target_frame <- as.data.frame(GridTimeSeries[[i]][1])
    target_col <- target_frame['X.cloudhigh']
    if( i == 8) {
        target_col['year'] <- 1
    }
    else if (i == 20) {
        target_col['year'] <- 2
    }
    else if (i == 32) {
        target_col['year'] <- 3
    }
    else if (i == 44) {
        target_col['year'] <- 4
    }
    else if (i == 56) {
```

```
        target_col['year'] <- 5
    }
    else {
        target_col['year'] <- 6
    }
    August <- rbind(August,target_col)
}

Aug_avg <- data.frame(matrix(ncol = 1, nrow = 0))
colnames(Aug_avg) <- c("aug_avg_cloudhigh")
for (x in 1:576){
    counter <- x
    sum_val <- 0
    for (y in 1:6) {
        sum_val <- sum_val + August$X.cloudhigh[counter]
        counter <- counter + 576
    }
    aug_avg_val <- sum_val/6
    Aug_avg <- rbind(Aug_avg,data.frame(aug_avg_val))
}

#September
sep <- c(9,21,33,45,57,69)
```

```
September <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(September) <- col_name
for (i in sep){
    target_frame <- as.data.frame(GridTimeSeries[[i]][1])
    target_col <- target_frame['X.cloudhigh']
    if( i == 9) {
        target_col['year'] <- 1
    }
    else if (i == 21) {
        target_col['year'] <- 2
    }
    else if (i == 33) {
        target_col['year'] <- 3
    }
    else if (i == 45) {
        target_col['year'] <- 4
    }
    else if (i == 57) {
        target_col['year'] <- 5
    }
    else {
        target_col['year'] <- 6
    }
```

```
    September <- rbind(September,target_col)
}

Sep_avg <- data.frame(matrix(ncol = 1, nrow = 0))
colnames(Sep_avg) <- c("sep_avg_cloudhigh")
for (x in 1:576){
    counter <- x
    sum_val <- 0
    for (y in 1:6) {
        sum_val <- sum_val + September$X.cloudhigh[counter]
        counter <- counter + 576
    }
    sep_avg_val <- sum_val/6
    Sep_avg <- rbind(Sep_avg,data.frame(sep_avg_val))
}

#October
oct <- c(10,22,34,46,58,70)
October <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(October) <- col_name
for (i in oct){
    target_frame <- as.data.frame(GridTimeSeries[[i]][1])
    target_col <- target_frame['X.cloudhigh']
```

```
    if( i == 10) {
        target_col['year'] <- 1
    }
    else if (i == 22) {
        target_col['year'] <- 2
    }
    else if (i == 34) {
        target_col['year'] <- 3
    }
    else if (i == 46) {
        target_col['year'] <- 4
    }
    else if (i == 58) {
        target_col['year'] <- 5
    }
    else {
        target_col['year'] <- 6
    }
    October <- rbind(October,target_col)
}

Oct_avg <- data.frame(matrix(ncol = 1, nrow = 0))
colnames(Oct_avg) <- c("Oct_avg_cloudhigh")
```

```
for (x in 1:576){
    counter <- x
    sum_val <- 0
    for (y in 1:6) {
        sum_val <- sum_val + October$X.cloudhigh[counter]
        counter <- counter + 576
    }
    oct_avg_val <- sum_val/6
    Oct_avg <- rbind(Oct_avg,data.frame(oct_avg_val))
}

#November
nov <- c(11,23,35,47,59,71)
November <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(November) <- col_name
for (i in nov){
    target_frame <- as.data.frame(GridTimeSeries[[i]][1])
    target_col <- target_frame['X.cloudhigh']
    if( i == 11) {
        target_col['year'] <- 1
    }
    else if (i == 23) {
        target_col['year'] <- 2
```

```
    }
    else if (i == 35) {
        target_col['year'] <- 3
    }
    else if (i == 47) {
        target_col['year'] <- 4
    }
    else if (i == 59) {
        target_col['year'] <- 5
    }
    else {
        target_col['year'] <- 6
    }
    November <- rbind(November,target_col)
}


Nov_avg <- data.frame(matrix(ncol = 1, nrow = 0))
colnames(Nov_avg) <- c("Nov_avg_cloudhigh")
for (x in 1:576){
    counter <- x
    sum_val <- 0
    for (y in 1:6) {
        sum_val <- sum_val + November$X.cloudhigh[counter]
```

```
        counter <- counter + 576
    }
    nov_avg_val <- sum_val/6
    Nov_avg <- rbind(Nov_avg,data.frame(nov_avg_val))
}

#December
dec <- c(12,24,36,48,60,72)
December <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(December) <- col_name
for (i in dec){
    target_frame <- as.data.frame(GridTimeSeries[[i]][1])
    target_col <- target_frame['X.cloudhigh']
    if( i == 12) {
        target_col['year'] <- 1
    }
    else if (i == 24) {
        target_col['year'] <- 2
    }
    else if (i == 36) {
        target_col['year'] <- 3
    }
    else if (i == 48) {
```

```
        target_col['year'] <- 4
    }
    else if (i == 60) {
        target_col['year'] <- 5
    }
    else {
        target_col['year'] <- 6
    }
    December <- rbind(December,target_col)
}

Dec_avg <- data.frame(matrix(ncol = 1, nrow = 0))
for (x in 1:576){
    counter <- x
    sum_val <- 0
    for (y in 1:6) {
        sum_val <- sum_val + December$X.cloudhigh[counter]
        counter <- counter + 576
    }
    dec_avg_val <- sum_val/6
    Dec_avg <- rbind(Dec_avg,data.frame(dec_avg_val))
}
```

```
result_set <-
cbind(January_avg,February_avg,March_avg,Apr_avg,May_avg,June_avg,July_avg,Au
g_avg,Sep_avg,Oct_avg,Nov_avg,Dec_avg,lon_lat)

############################graphing test

xaxis=c(1:12)

plot(c(1:12),result_set[1,1:12],type="b")

image(t(log(approx.elevation+1)[24:1,]))

points(x.new.scaled,y.new.scaled,pch=20,lwd=0.6,cex=0.3)

x.scaled <- (x-6)/101*(1/23)

y.scaled <- (result_set[1,1:12])/2*(1/23)


x.seq <- 1:12

y.seq <- result_set[1,1:12]

plot(x.seq,y.seq,type="b")

x.scaled <- (x.seq)/12*(1/23)

y.scaled <- (y.seq)/12*(1/23)

plot(x.scaled,y.scaled,"b")

image(t(log(approx.elevation+1)[24:1,]))

points(x.scaled,y.scaled,pch=20,lwd=0.6,cex=0.3)

lines(x.scaled,y.scaled,lwd=0.5,col="blue")

############################3graphing

approx.elevation <- matrix(0,24,24)

for(i in 1:24){

   lat <- metric.Lat[i]
```

```
    dist.seq.lat <- abs(ele.Lat-lat)
    lat.index <- which.min(dist.seq.lat)
    Latitude=lat.index

    for(j in 1:24){
        lon <- metric.Lon[j]
        dist.seq.lon <- abs(ele.Lon-lon)
        lon.index <- which.min(dist.seq.lon)
        Longitude=lon.index
        approx.elevation[i,j] <- ele[lat.index,lon.index]
    }
}

image(log(approx.elevation+1))
image(t(log(approx.elevation+1)[24:1,]),xaxt="n",yaxt="n")

##############

######## use the elevation data set to find approximate elevation for the 24 x 24 grid
## (approximation is needed because the positions of the measurements are not the same as the elevation data )
approx.elevation <- matrix(0,24,24)
### use closet location for approximation
for(i in 1:24){
```

```
    lat <- metric.Lat[i]
    dist.seq.lat <- abs(ele.Lat-lat)
    lat.index <- which.min(dist.seq.lat)

    for(j in 1:24){
        lon <- metric.Lon[j]
        dist.seq.lon <- abs(ele.Lon-lon)
        lon.index <- which.min(dist.seq.lon)
        approx.elevation[i,j] <- ele[lat.index,lon.index]
    }
}

image(log(approx.elevation+1))
image(t(log(approx.elevation+1)[24:1,]))
image(t(log(approx.elevation+1)[24:1,]),col = terrain.colors(100))
image(t(log(approx.elevation+1)[24:1,]),col = terrain.colors(100),xaxt="n",yaxt="n")

x.scaled <- (x.seq-50)/101*(1/23)
y.scaled <- (y.seq)/2*(1/23)

image(t(log(approx.elevation+1)[24:1,]),col = terrain.colors(100))
lines(x.scaled,y.scaled)
```

```
image(t(log(approx.elevation+1)[24:1,]),col =
terrain.colors(100),ylim=c(-1/23,1/23),xlim=c(-1/23,1/23))

lines(x.scaled,y.scaled)


x.scaled <- (x.seq-50)/101*(1/23)

y.scaled <- (y.seq)/2*(1/23)/2


image(t(log(approx.elevation+1)[24:1,]),col =
terrain.colors(100),ylim=c(-1/23,1/23),xlim=c(-1/23,1/23))

lines(x.scaled,y.scaled)

image(t(log(approx.elevation+1)[24:1,]),col = terrain.colors(100))

lines(x.scaled,y.scaled)


#### now move it to cell with the first lat and fourth lon

x.new.scaled <- x.scaled + 3/23

y.new.scaled <- y.scaled + 23/23

image(t(log(approx.elevation+1)[24:1,]),col = terrain.colors(100))

points(x.new.scaled,y.new.scaled,pch=20,lwd=0.6,cex=0.3)


for (i in 1:576) {

   month <- 1:12

   HCC_1 <-
as.numeric(c(January_avg$jan_avg_val[i],February_avg$feb_avg_val[i],March_avg$ma
r_avg_val[i],
```

```
Apr_avg$apr_avg_val[i],May_avg$may_avg_val[i],June_avg$jun_avg_val[i],

July_avg$july_avg_val[i],Aug_avg$aug_avg_val[i],Sep_avg$sep_avg_val[i],

Oct_avg$oct_avg_val[i],Nov_avg$nov_avg_val[i],Dec_avg$dec_avg_val[i]))
    month <- 1:12
    month.scaled <- (month-6)/12*(1/23)
    HCC_1.scaled <- (HCC_1-9)/2*(1/23)/10/2
    month.new.scaled <- month.scaled + 0/23 + (i-1)%/%24*(1/23)
    HCC_1.new.scaled <- HCC_1.scaled + 23/23 - ((i-1)%%24*(1/23))
    lines(x=month.new.scaled,y=HCC_1.new.scaled,lwd=1)
}

head(January_avg)
```