

```

#####
# read data from two files which have been downloaded from
# http://www1.ncdc.noaa.gov/pub/data/ghcn/v2/
# and decompressed by an external program

#v2.mean.Z
#v2.mean.adj.Z

v2.mean = readLines("v2.mean")
v2.madj = readLines("v2.mean_adj")

length(v2.mean) # 595759
length(v2.madj) # 422373

#last ten lines of adjusted file are identical and contain no information
#remove 9 of them
v2.madj = v2.madj[1:422364]

#identify matching station and year lines in both sets
#extract identifying info

idv2 = substr(v2.mean,1,16)
idv2adj = substr(v2.madj,1,16)

sum(idv2[-length(idv2)] > idv2[-1]) #0
sum(idv2adj[-length(idv2adj)] > idv2adj[-1]) #0

#check to see if both sets are in alphabetical order
#if so the pairing process is faster

#function to pair lines
reconcile= function(dat1,dat2) {
  leng1 = length(dat1)
  leng2 = length(dat2)
  id.pos = rep(NA, leng2)
  curr = 1
  for (i in 1:leng2) { j = curr
    while (dat2[i] >= dat1[j]) {j=j+1}
    if (dat2[i]==dat1[j-1]) {
      id.pos[i]=j-1
      curr = j}}
  id.pos }

inds = reconcile(idv2,idv2adj)

#check to see if there are adjusted lines without originals in the raw data
#remove if necessary

```

```

sum(is.na(inds) #31

v2.madjx = v2.madj[-which(is.na(inds))]
indsx = inds[-which(is.na(inds))]
v2.meanx = v2.mean[indsx]
idv2x = idv2[indsx]
idv2adjx = idv2adj[-which(is.na(inds))]

identical(idv2x,idv2adjx) # TRUE

#function to calculate individual monthly differences

diff.calc = function(dat1,dat2) {
  len = length(dat1)
  outmat = matrix(NA,len,13)
  st = 17 + (5*(0:11))
  en = st+4
  x1 = x2 = rep(NA,12)
  for (i in 1:len) {chx1 = dat1[i]
  chx2=dat2[i]
  outmat[i,1] = as.numeric(substr(chx1,13,16))
  if (outmat[i,1] != as.numeric(substr(chx2,13,16))) return("Error")
  for (j in 1:12) {
    x1[j] = as.numeric(substr(chx1,st[j],en[j]))
    x2[j] = as.numeric(substr(chx2,st[j],en[j]))}
  x1[x1==--9999]=NA
  x2[x2==--9999]=NA
  outmat[i,2:13] = (x2-x1)/10}
  outmat}

#adjustment = adjusted - unadjusted
adjs = diff.calc(v2.meanx,v2.madjx)

#some statistics
12*422342 # 5068104 total number of monthly values
sum(is.na(adjs[,-1])) # 205985 (4.06%) NAs
sum( adjs[,-1]==0,na.rm=T) # 1631153 (32.18%) unadjusted values

#calculate annual average for each station in a given year
year=adjs[,1]
ann.mean = rowMeans(adjs[,2:13],na.rm=T)

#calculate average of all adjustments in a given year
annadj = data.frame(year,ann.mean)

aveadj = c(by(annadj,annadj$year, function(x) mean(x$ann.m)))

```

```
plot(year,ann.mean,cex=.25,main = "Annual Averages for Individual Stations",  
      xlab="Year", ylab="Degrees (C)" )
```

```
plot(as.numeric(names(aveadj)),aveadj, main = "Mean Annual GHCN Adjustment",  
      xlab = "Year",ylab = "Degrees (C)")
```