

RBeginnersClass2

Gregory S Gilbert

11/12/2016

Read in data from the web We will use these dataframes for different things today.

```
Rsite<-"http://greggilbertlab.sites.ucsc.edu/wp-content/uploads/sites/276/2015/09/"
a<-read.csv(paste(Rsite,"RegressionDataset.csv",sep=""))
three<-read.csv(paste(Rsite,"ThreeTreatmentDataset.csv",sep=""))
ferp<-read.csv(paste(Rsite,"FERP07data.csv",sep=""))
ferp<-ferp[,1:6] #trim to just the first six columns
logdat<-read.csv(paste(Rsite,"logistregdata.csv",sep=""))
facblock<-read.csv(paste(Rsite,"FactorialBlockDataset.csv",sep=""))
allom<-read.csv(paste(Rsite,"heightdbh.csv",sep=""))
```

Some more common things you often need to do with data ##Sorting data **Sort a vector**

```
head(a)
(temp<-a$temp) #make a vector of just the temperature data
sort(temp) #sort it
sort(temp, decreasing=T) #sort it descending
```

But what if you want to sort the dataframe by one or more variables?

```
head(a) #look at the dataframe
head(a[order(a$temp),]) #sort all by temp
head(a[order(-a$temp),]) #sort all by decreasing temp
a[order(-a$temp,a$precip),] #sort all by decr temp & incr ppt
a[with(a, order(-temp, precip)),] #same in different syntax
```

A crude way to shuffle data in specific columns

```
head(a) #look at the dataframe
a2<-a; head(a2) #make a copy
#want to sort num_spp by rannum, but not temp and precip
a3<-as.data.frame(a2[,c('num_spp')]); colnames(a3)<-"num_spp"
a3$rannum<-runif(dim(a3)[1]) #add col of uniform random #s
a3<-a3[order(a3$rannum),] #sort by that #
a2$num_spp<-a3$num_spp #replace the randomized column
head(a2)
```

Subsetting Data

```
head(ferp) #take a look at the FERP data
subf<-ferp[,c("tag","code","date","dbh","east","north")] #another way
subf<-subf[subf$code%in%c("QUERPA","SEQUESE","TOXIDI"),] #keep 3 spp
head(subf)
table(subf$code) # look at how many of each species
subf<-droplevels(subf) #get rid of the extra levels
table(subf$code) #look again at the number of each species
```

Combining Data

Let's make a small dataframe with descriptors of those three species

Use `cbind` to join vectors side by side

```
code<-c("QUERPA","SEQUESE","TOXIDI","PSEUME") #names of species
color<-c("red","blue","yellow","green") # a color for each
shape<-c(1,2,19,21) #number that indicate symbol types
other<-cbind(code,color,shape) #join using cbind
otherdf<-as.data.frame(cbind(code,color,shape)) #make a dataframe
otherdf #take a peek
```

put dataframes (or matrices) on top of each other with `cbind`

```
otherdf2<-otherdf #just make a copy
(otherdf3<-rbind(otherdf,otherdf2)) #stack two dataframes
```

`cbind` and `rbind` just glue together regardless of content **Use `merge` to combine based on content**

```
head(subf) #main dataframe
head(otherdf) #auxiliary dataframe
subf2<-merge(subf,otherdf,by='code',all.x=T,all.y=F) #merge them
head(subf2); tail(subf2)
```

Summarizing data

We will use the first six columns of the `ferp` data here

```
head(ferp)
mean(ferp$dbh) #global mean of dbh
```

What if you want means for each species? Three approaches: `by`, `aggregate`, `tapply`

```
ferpby<-by(data=ferp$dbh,INDICES=ferp$code,FUN=mean) #creates a list
ferpagg<-aggregate(x=ferp$dbh,by=list(ferp$code),FUN=mean) # create df
ferptapply<-tapply(X=ferp$dbh,INDEX=ferp$code,FUN=mean) #create vector
head(ferpby)
head(ferpagg)
head(ferptapply)
```

to make ferpagg prettier

```
ferpagg<-setNames(aggregate(x=ferp$dbh,by=list(ferp$code),FUN=mean),c("code","mean_dbh"))
#or after the fact
ferpagg<-aggregate(x=ferp$dbh,by=list(ferp$code),FUN=mean) # create df
colnames(ferpagg)<-c("code","mean_dbh") #add column names
```

Make your own function

```
NameOfMyFunction<-function(dataobject)
{
a<-function1(dataobject) #do something to your dataobject
b<-function2(dataobject) #do something else
a*b #the result of the last line is what is returned by the function
}
# note that dataobject is a placeholder name
# it takes on the value of whatever you put in the ()
```

Make your own function

```
myStats<-function(mydata)
{
mean<-mean(mydata)
sd<-sd(mydata)
se<-sd(mydata)/sqrt(length(mydata))
n<-length(mydata)
med<-median(mydata)
q<-quantile(mydata,prob=c(0.025,0.975))
aa<-c(mean,sd,se,n,med,q[1],q[2])
names(aa)<-c("mean","sd","se","n","median","l95CI","u95CI")
aa
}
```

You now have a function `myStats` that takes an object `mydata`

Apply it to some data!

Create complex (or simple, but useful) functions at the start of your code, then use them at will!

Let's make some pretty pictures.

Refer to Class6 handout. Let's use the subset of the ferp data `subf`

Poke around those data

Make a simple plot

```
plot(north~east,data=subf)
plot(north~east,data=subf[subf$code=='TOXIDI',])
plot(north~east,
     data=subf[subf$code=='TOXIDI',],
     pch=19,
     col="red",
     cex=.5,
     las=1,
     asp=1,
     xlim=c(0,200),
     ylim=c(0,300)
     )
```

```
pdf("~/Desktop/ferplot.pdf",width=6,height=8)
plot(north~east,
     data=subf[subf$code=='TOXIDI',],
     pch=19,
     col="blue",
     cex=.5,
     las=1,
     asp=1,
     xlim=c(0,200),
     ylim=c(0,300)
     )
dev.off()
```

Let's overlay another species and add a legend

```
pdf("~/Desktop/ferplot.pdf",width=6,height=8)
plot(north-east,
     data=subf[subf$code=='TOXIDI',],
     pch=19,
     col="blue",
     cex=.5,
     las=1,
     asp=1,
     xlim=c(0,200),
     ylim=c(0,300)
     )
points(north-east,
       data=subf[subf$code=='SEQUESE',],col="red")

legend(-20,310,c('TOXIDI','SEQUESE'),pch=c(19,1),col=c('blue','red'),bg="yellow")
dev.off()
```

A more elegant way Recall subf2?

```
head(subf2); tail(subf2)
plot(north-east,data=subf2,col=color,pch=shape)
subf2$shape<-as.numeric(subf2$shape)
plot(north-east,data=subf2,col=color,pch=shape)

pdf("~/Desktop/ferplot2.pdf",width=6,height=8)
plot(north-east,data=subf2,col=color,pch=shape)
dev.off()
```

Let's explore par ?par