Outline: bringing together what we've covered so far with an example

- read in data
- see if it makes sense
- clean up or filter it
- do some statistical tests
- explore the results
- manuscript()

Today's example starts with pre-processed data

- we're using a specialised program to preprocess array data
  - gcos
  - dChip
  - RMAexpress
- text file, one column per chip

read in data

you really want informative row and column names

you don't want extraneous columns

```
kk <- read.delim('data.txt',sep='\t',row.names=1)
str(kk)
kk <- as.matrix(kk)
```

explore the data: what does it look like?

```
image(kk)
image(is.na(kk))
plot(density(kk))
plot(density(log(kk),na.rm=T)
mean(kk)
rowmeans<-apply(kk,1,mean,na.rm=T)
```

explore the data: any odd men out?

```
colmeans<-apply(kk,2,mean,na.rm=T)
(kkcor<-cor(kk))
image(kkcor)
plot.cor(kkcor)
min(kkcor)
```

filtering the data

get rid of uninteresting or problem data

reduce number of tests

various ways of generating a logical vector

```
high <- rowmeans > 300
complete <- apply(kk,1, function(x) !any(is.na(x)) )
```

making factors for statistical tests

```
fac<-strsplit(colnames(kkm),'_')
fac<-data.frame(fac)
fac<-t(fac)
rownames(fac)<-NULL

colnames(fac)<-c('chip','date','sampno','geno','treat','tissue')
fac<-data.frame(fac)
```

try some tests on one row

```
t.test(kkm[1,]~fac$tissue)
t.test(kkm[1,]~fac$tissue,var.equal=T) ->tt
str(tt)
tt$p.value
```

make a wrapper function and apply it to all the rows

```
ttrow <- function (x) {
    tt<-t.test(x ~ fac$tissue)
    tt$p.value }
tttispval<-apply(kk,1,ttrow)
```

look at the p-values

```
sort(tttispval)[1:10]
sum(ttispval < .05/1999)
library(GeneTS)
fdr.control(tttispval)
```

exercise for next time

do the same thing using anova

do some other anova tests

```
Outline: bringing together what we've covered so far with an example
        read in data
        see if it makes sense
        clean up or filter it
        do some statistical tests
        explore the results
        manuscript()
Today's example starts with pre-processed data
        we're using a specialised program to preprocess array data
                gcos
                dChip
                RMAexpress
        text file, one column per chip
read in data
        you really want informative row and column names
        you don't want extraneous columns
        kk <- read.delim('data.txt',sep='\t',row.names=1)
        str(kk)
        kk <- as.matrix(kk)
explore the data: what does it look like?
        image(kk)
        image(is.na(kk))
        plot(density(kk))
        plot(density(log(kk),na.rm=T)
        mean(kk)
        rowmeans<-apply(kk,1,mean,na.rm=T)
explore the data: any odd men out?
        colmeans<-apply(kk,2,mean,na.rm=T)
        (kkcor<-cor(kk))
        image(kkcor)
        plot.cor(kkcor)
        min(kkcor)
filtering the data
        get rid of uninteresting or problem data
        reduce number of tests
        various ways of generating a logical vector
                high     <- rowmeans > 300
                complete <- apply(kk,1, function(x) !any(is.na(x)) )
making factors for statistical tests
        fac<-strsplit(colnames(kkm),'_')
        fac<-data.frame(fac)
        fac<-t(fac)
        rownames(fac)<-NULL
        colnames(fac)<-c('chip','date','sampno','geno','treat','tissue')
        fac<-data.frame(fac)
try some tests on one row
        t.test(kkm[1,]~fac$tissue)
        t.test(kkm[1,]~fac$tissue,var.equal=T) ->tt
        str(tt)
        tt$p.value
make a wrapper function and apply it to all the rows
        ttrow <- function (x) {
                tt<-t.test(x ~ fac$tissue)
                tt$p.value  }
        tttispval<-apply(kk,1,ttrow)
look at the p-values
        sort(tttispval)[1:10]
        sum(ttispval < .05/1999)
        library(GeneTS)
        fdr.control(tttispval)
```