Test-script for applying a model to subsets of dataset

Andrew Nguyen

2016-March-07

Contents

Play script to show how to apply a stat model to a dataset

load in libraries

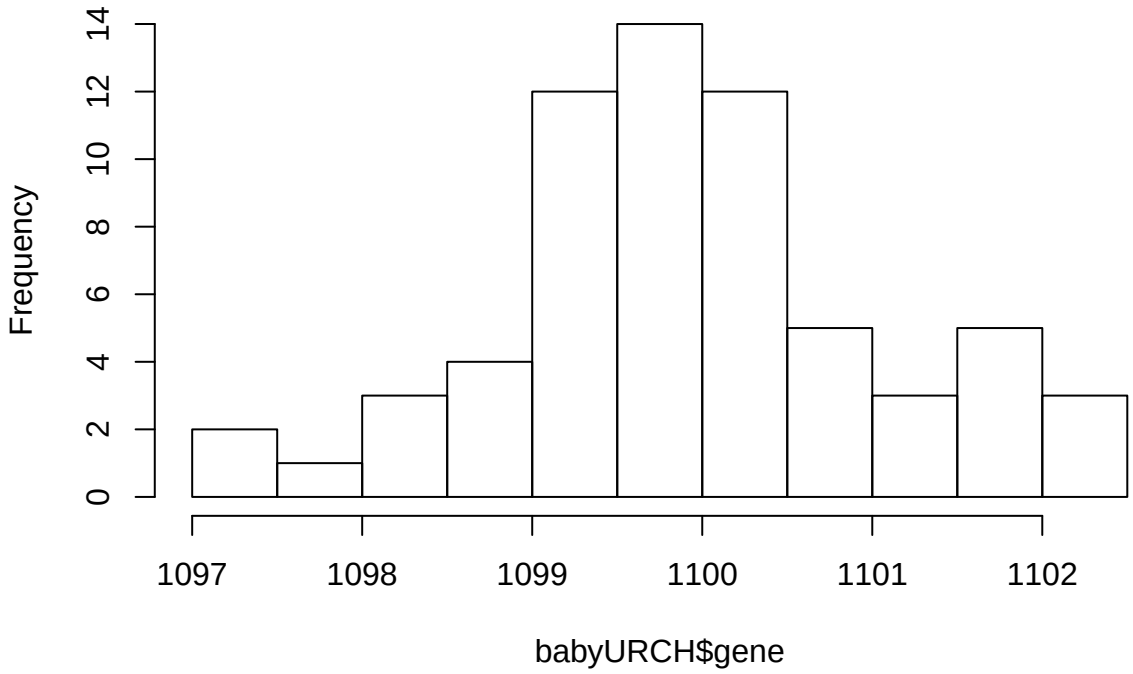
```
#library(dplyr)
library(ggplot2)
library(plyr)
library(MASS)
```

Loading in mock dataset

```
babyURCH<-read.csv("20160307_playing_around.csv")
str(babyURCH)
```

```
## 'data.frame':    64 obs. of  5 variables:
##  $ Gene          : Factor w/ 2 levels "a","b": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Splice_variant: Factor w/ 4 levels "g","t","x","z": 4 4 4 4 4 4 4 4 4 4 ...
##  $ rep           : int  1 1 1 1 2 2 2 2 3 3 ...
##  $ Day           : int  1 7 1 7 1 7 1 7 1 7 ...
##  $ acid          : Factor w/ 2 levels "high","low": 1 1 2 2 1 1 2 2 1 1 ...
```

```
#randomly generating expression data
babyURCH$gene_xp<-rnorm(length(babyURCH$Gene),100)+1000
hist(babyURCH$gene)
```




```
#visualize dataset
str(babyURCH)
```

```
## 'data.frame':    64 obs. of  6 variables:
##  $ Gene          : Factor w/ 2 levels "a","b": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Splice_variant: Factor w/ 4 levels "g","t","x","z": 4 4 4 4 4 4 4 4 4 4 ...
##  $ rep           : int  1 1 1 1 2 2 2 2 3 3 ...
##  $ Day           : int  1 7 1 7 1 7 1 7 1 7 ...
##  $ acid          : Factor w/ 2 levels "high","low": 1 1 2 2 1 1 2 2 1 1 ...
##  $ gene_xp       : num  1100 1101 1099 1100 1099 ...
```

```
#check replicates per treatment
ddply(babyURCH,.(Gene,Splice_variant,Day,acid),summarize,counts=length(rep))
```

```
##    Gene Splice_variant Day acid counts
## 1     a              x   1 high      4
## 2     a              x   1  low      4
## 3     a              x   7 high      4
## 4     a              x   7  low      4
## 5     a              z   1 high      4
## 6     a              z   1  low      4
## 7     a              z   7 high      4
## 8     a              z   7  low      4
## 9     b              g   1 high      4
## 10    b              g   1  low      4
## 11    b              g   7 high      4
## 12    b              g   7  low      4
## 13    b              t   1 high      4
```

```
## 14    b              t   1  low      4
## 15    b              t   7 high      4
## 16    b              t   7  low      4
```

Fittign ANOVA model

```
#fitting models
#specifying the model with 3 way interaction
aovmod<-function(df){
  aov(gene_xp~Splice_variant*acid*Day,data=mutate(df))
}


#specifying the model with forward selection (can do backward or both too)
step.aovmod<-function(df){
  summary(stepAIC(aov(gene_xp~Splice_variant*acid*Day,data=mutate(df)),direction="forward"))
  #stepAIC(aov(gene_xp~Splice_variant*acid*Day,data=mutate(df)),direction="forward")
}

#applying the model for every gene
#.(Gene) splits the data based on gene
#models<-dlply(babyURCH,.(Gene),step.aovmod)

#alternative way of doing it with lapply(this is better)
models2<-sapply(split(babyURCH,list(babyURCH$Gene)),step.aovmod)
```

```
## Start:  AIC=2.22
## gene_xp ~ Splice_variant * acid * Day
##
## Start:  AIC=17.02
## gene_xp ~ Splice_variant * acid * Day
```

```
models2
```

```
## $a
##                         Df  Sum Sq Mean Sq F value  Pr(>F)
## Splice_variant           1  0.7426  0.7426  0.8567 0.36388
## acid                     1  1.6508  1.6508  1.9045 0.18030
## Day                      1  0.1135  0.1135  0.1309 0.72064
## Splice_variant:acid      1  0.3917  0.3917  0.4519 0.50784
## Splice_variant:Day       1  0.6167  0.6167  0.7114 0.40729
## acid:Day                 1  0.0182  0.0182  0.0210 0.88586
## Splice_variant:acid:Day  1  3.4841  3.4841  4.0196 0.05638 .
## Residuals               24 20.8027  0.8668
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $b
##                         Df Sum Sq Mean Sq F value   Pr(>F)
## Splice_variant           1 12.095 12.0946  8.7854 0.006759 **
```

```
## acid                     1  1.534  1.5336  1.1140 0.301727
## Day                      1  2.160  2.1597  1.5688 0.222453
## Splice_variant:acid      1  0.065  0.0654  0.0475 0.829340
## Splice_variant:Day       1  0.438  0.4376  0.3179 0.578127
## acid:Day                 1  1.028  1.0281  0.7468 0.396050
## Splice_variant:acid:Day  1  0.150  0.1499  0.1089 0.744258
## Residuals               24 33.040  1.3767
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Parsing output of model

```
#putting the model in a dataframe
full<-ldply(models2,data.frame)

#this line of code took the longest...grabbing the predictor names
#and just adding it to the dataframe!!
full$variables<-c(t(ldply(models2,row.names))[-1,])

#adding a new dataframe to indicate whether predictor
#was significant at 0.05 level. Can toggle to lower if you want
full$sig<-ifelse(full$Pr..F.<0.05,"sig","nonsig")

#let's nicely visualize the dataframe!!
knitr::kable(full)
```

.id	Df	Sum.Sq	Mean.Sq	F.value	Pr..F.	variables	sig
a	1	0.7425664	0.7425664	0.8566966	0.3638789	Splice_variant	nonsig
a	1	1.6507688	1.6507688	1.9044870	0.1802982	acid	nonsig
a	1	0.1134806	0.1134806	0.1309222	0.7206441	Day	nonsig
a	1	0.3917165	0.3917165	0.4519222	0.5078444	Splice_variant:acid	nonsig
a	1	0.6166606	0.6166606	0.7114395	0.4072948	Splice_variant:Day	nonsig
a	1	0.0182426	0.0182426	0.0210464	0.8858640	acid:Day	nonsig
a	1	3.4841439	3.4841439	4.0196463	0.0563783	Splice_variant:acid:Day	nonsig
a	24	20.8026895	0.8667787	NA	NA	Residuals	NA
b	1	12.0945864	12.0945864	8.7854020	0.0067585	Splice_variant	sig
b	1	1.5335857	1.5335857	1.1139833	0.3017273	acid	nonsig
b	1	2.1596773	2.1596773	1.5687707	0.2224534	Day	nonsig
b	1	0.0653719	0.0653719	0.0474856	0.8293400	Splice_variant:acid	nonsig
b	1	0.4375895	0.4375895	0.3178612	0.5781266	Splice_variant:Day	nonsig
b	1	1.0280814	1.0280814	0.7467893	0.3960497	acid:Day	nonsig
b	1	0.1499257	0.1499257	0.1089047	0.7442578	Splice_variant:acid:Day	nonsig
b	24	33.0400448	1.3766685	NA	NA	Residuals	NA

```
#writing out data frame into csv file :-)!
write.csv(full,"Test_output_model_dataframe.csv")
```

Let's visualize gene "a" expression

Grab my ggplot defaults: http://adnguyen.github.io/blog/2016/02/12/ggplot_defaults

The gray background and grids are so annoying.

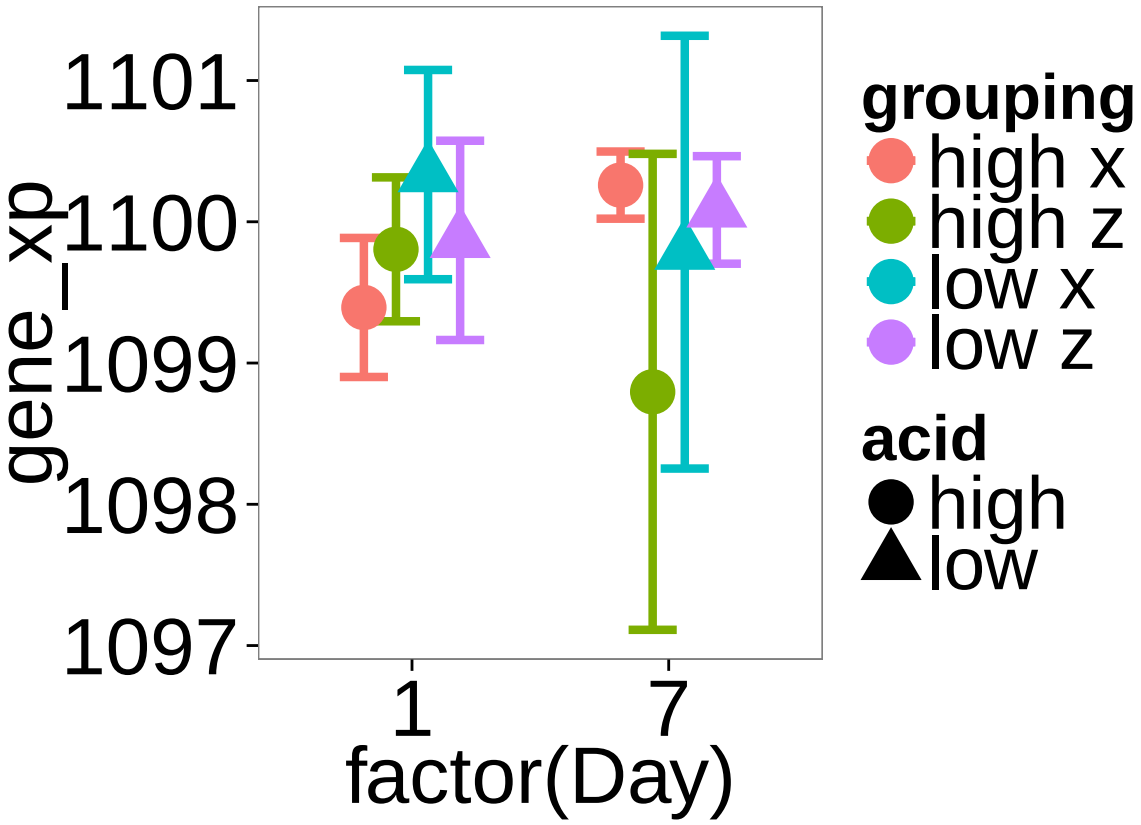
```
default<-theme_bw()+theme(text=element_text(size=30),axis.text=element_text(size=30), legend.text=eleme
library(Rmisc)
```

```
## Loading required package: lattice
```

```
#grab averages
tc<-summarySE(babyURCH,measurevar="gene_xp",groupvars=c("Gene","Day","acid","Splice_variant"))
#grouping splice variant and acid levels together for coloring purposes
tc$grouping<-paste(tc$acid,tc$Splice_variant)
#visualize it
knitr::kable(tc)
```

Gene	Day	acid	Splice_variant	N	gene_xp	sd	se	ci	grouping
a	1	high	x	4	1099.393	0.4922160	0.2461080	0.7832256	high x
a	1	high	z	4	1099.805	0.5094212	0.2547106	0.8106028	high z
a	1	low	x	4	1100.334	0.7406555	0.3703277	1.1785481	low x
a	1	low	z	4	1099.868	0.7053037	0.3526518	1.1222955	low z
a	7	high	x	4	1100.260	0.2376525	0.1188262	0.3781581	high x
a	7	high	z	4	1098.796	1.6846759	0.8423379	2.6806953	high z
a	7	low	x	4	1099.785	1.5319707	0.7659853	2.4377072	low x
a	7	low	z	4	1100.084	0.3806229	0.1903114	0.6056559	low z
b	1	high	g	4	1101.357	0.6684302	0.3342151	1.0636216	high g
b	1	high	t	4	1100.121	1.0788632	0.5394316	1.7167121	high t
b	1	low	g	4	1100.788	1.1570969	0.5785484	1.8411993	low g
b	1	low	t	4	1099.097	1.1165316	0.5582658	1.7766510	low t
b	7	high	g	4	1100.382	1.1683093	0.5841546	1.8590408	high g
b	7	high	t	4	1099.340	1.3737656	0.6868828	2.1859676	high t
b	7	low	g	4	1100.256	1.7579498	0.8789749	2.7972904	low g
b	7	low	t	4	1099.307	0.6888551	0.3444276	1.0961222	low t

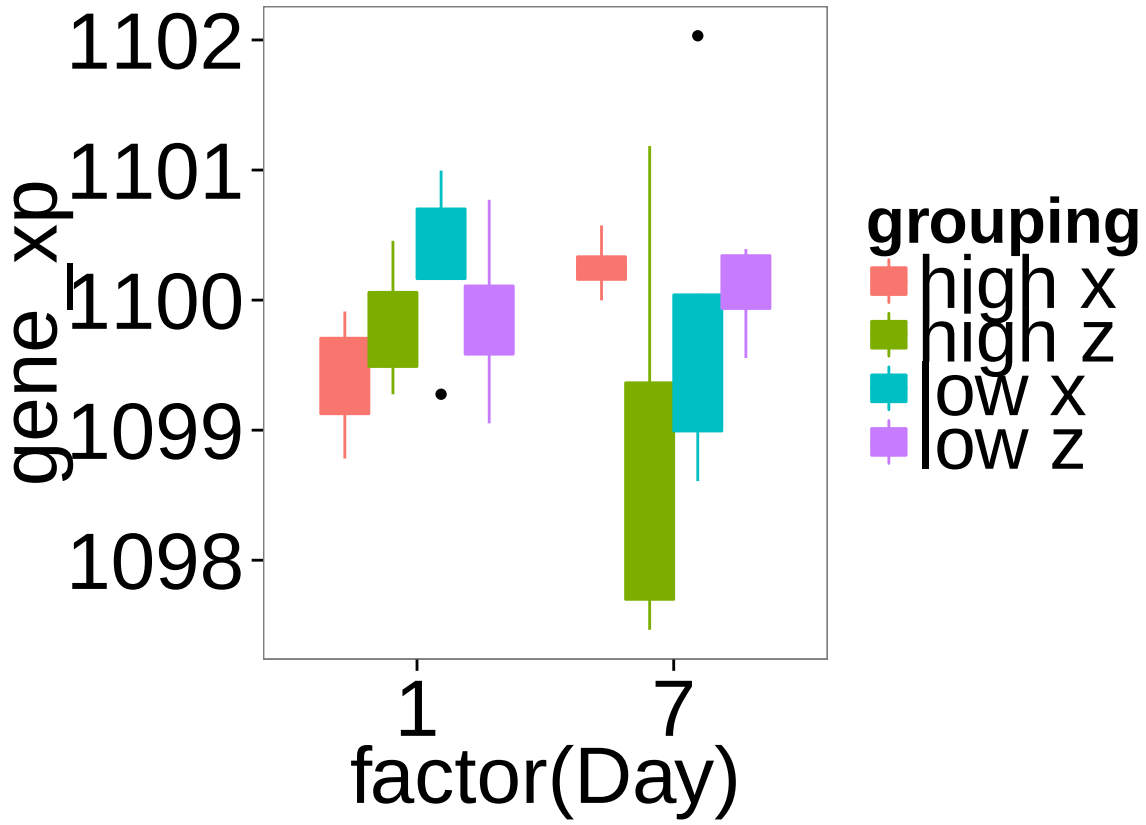
```
pd <- position_dodge(0.5) #offset parameter in plotting points
#only looking at the "a" gene
a.gene<-subset(tc,tc$Gene=="a")

#mock figure
ggplot(a.gene,aes(y=gene_xp,x=factor(Day),color=grouping,shape=acid))+default+geom_errorbar(aes(ymin=ge
geom_line(position=pd,size=1.5,stat="identity")+geom_point(position=pd,size=8)
```



You'll probably want boxplots because it better represents non-normal expression data (I think John assumed his expression data to be a negative binomial distribution).

let's try a boxplot

```
a.gene2<-subset(babyURCH,babyURCH$Gene=="a")
a.gene2$grouping<-paste(a.gene2$acid,a.gene2$Splice_variant)
ggplot(a.gene2,aes(y=gene_xp,x=factor(Day),color=grouping,fill=grouping))+default+geom_boxplot()
```


1102
1101
1100
1099
1098
gene_xp
1
7
factor(Day)
grouping
high x
high z
low x
low z