AgeGroup by Condition

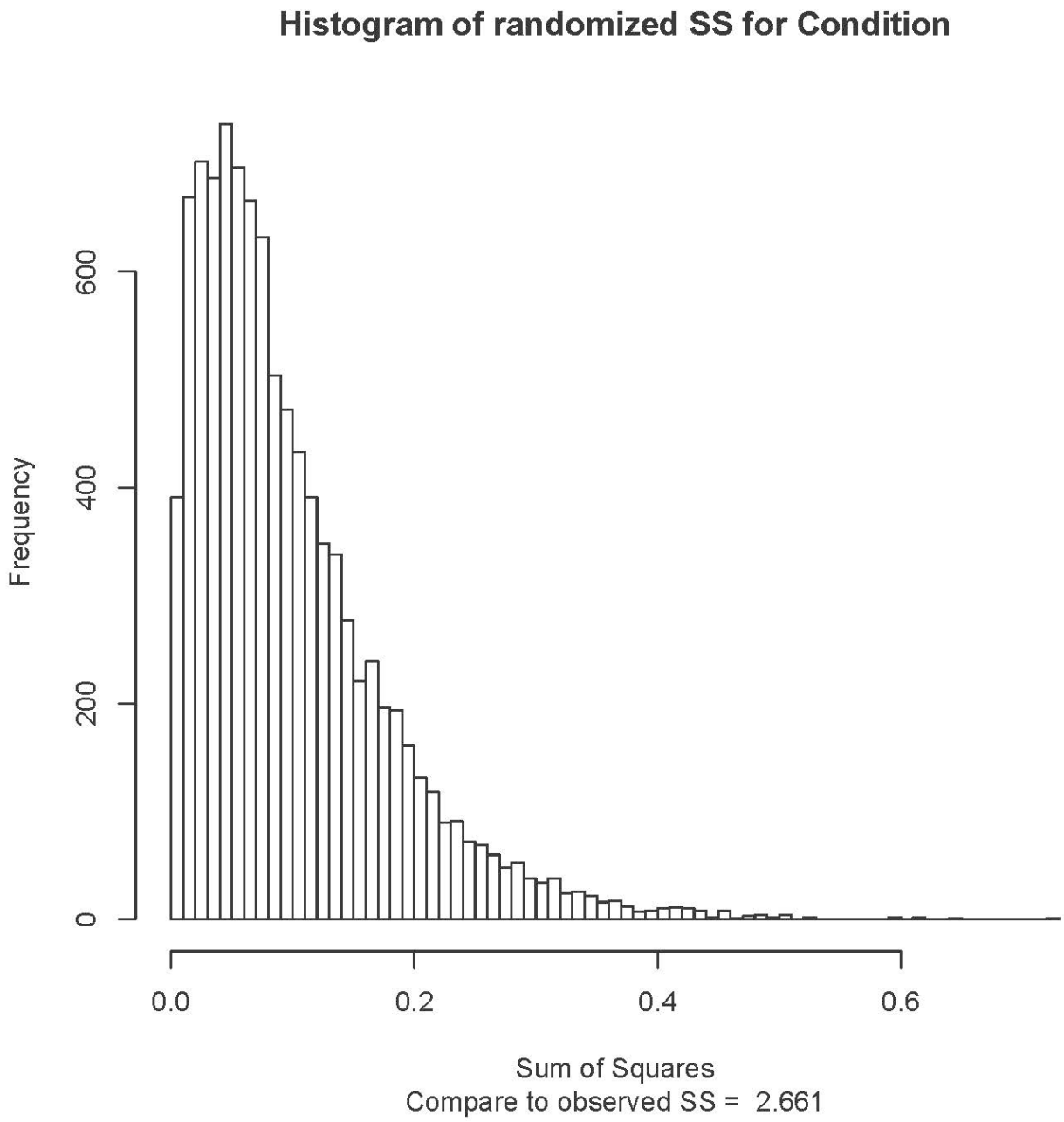
```
>
> # Age group rather than just adult-child is interesting, and more challenging.
> # Analyze All Targets for comparability with the mixed model results.
> # Throw out adults, though we don't need to because there are no
> # convergence problems.
>
> rm(list=ls()); options(scipen=999) # To avoid scientific notation
> rec1 = read.table("recursion-horizontal.data.txt",header=T)
> head(rec1)
  Participant AgeinMonths Group AgeGroups KABC_Concept KABC_Face KABC_Tri
1           1        48.5 Child      4yos            7        13       13
2           2        63.9 Child      5yos           19        12       16
3           4        64.2 Child      5yos           11         8       18
4           5        71.6 Child      5yos           14        11       24
5           6        53.1 Child      4yos           14         5       13
6           7        55.5 Child      4yos           13        13       11
  KABC_Pattern KABC_Hand CELF_SS CELF_WS CELF_EV CELF_ConDir CELF_Sent
1           NA         4      16      13      22           8        21
2            7         7      19      22      29          16        34
3            4         5      16      15      31          14        26
4            9         5      21      19      30          12        27
5           NA         5      19      21      36          19        34
6           NA         5      18      17      34          15        19
  CELF_Basic PPVT KBIT RecursionStrictTarget RecursionALLtargets
1         16   92   NA             0.10344828           0.10344828
2         NA   96   NA             0.35483871           0.35483871
3         NA  108   NA             0.12903226           0.35483871
4         NA  102   NA             0.06451613           0.16129032
5         16  111   NA             0.29032258           0.45161290
6         17   99   NA             0.06451613           0.06451613
   A.Strict  B.Strict  C.strict E.Strict A.alltargets B.alltargets
1 0.1666667 0.3333333 0.0000000        0    0.1666667    0.3333333
2 0.5714286 0.5000000 0.1666667        0    0.5714286    0.5000000
3 0.2857143 0.0000000 0.1666667        0    0.2857143    0.5000000
4 0.2857143 0.0000000 0.0000000        0    0.4285714    0.0000000
5 0.4285714 0.1666667 0.5000000        0    0.4285714    0.6666667
6 0.2857143 0.0000000 0.0000000        0    0.2857143    0.0000000
  C.alltargets E.alltargets
1    0.0000000    0.0000000
2    0.1666667    0.0000000
3    0.3333333    0.1666667
4    0.1666667    0.0000000
5    0.5000000    0.1666667
6    0.0000000    0.0000000
>
> # Extract the variables we need.
> rec1 = rec1[1:71,] # Omit last 13 rows, the adults
> Participant = rec1$Participant
> AgeGroup = factor(rec1$AgeGroups) # Eliminating empty Adult category
> AllTargets = rec1[,24:27]
> head(AllTargets)
  A.alltargets B.alltargets C.alltargets E.alltargets
1    0.1666667    0.3333333    0.0000000    0.0000000
2    0.5714286    0.5000000    0.1666667    0.0000000
3    0.2857143    0.5000000    0.3333333    0.1666667
4    0.4285714    0.0000000    0.1666667    0.0000000
5    0.4285714    0.6666667    0.5000000    0.1666667
6    0.2857143    0.0000000    0.0000000    0.0000000
> dim(AllTargets)
[1] 71  4
```

```
>
> # Look at means
> # aggregate(AllTargets,by=list(AgeGroup),FUN=mean)
> Means = as.matrix(aggregate(AllTargets,by=list(AgeGroup),FUN=mean)[,2:5])
> rownames(Means) = sort(unique(AgeGroup))
> addmargins(Means,FUN=mean, quiet=T)
     A.alltargets B.alltargets C.alltargets E.alltargets      mean
4yos    0.1809524    0.1866667   0.06666667   0.02666667 0.1152381
5yos    0.2011429    0.2533333   0.07333333   0.02666667 0.1386190
6yos    0.4421769    0.4841270   0.23015873   0.16666667 0.3307823
mean    0.2747574    0.3080423   0.12338624   0.07333333 0.1948798
>
> # To test main effect of Age Group, use the variable RecursionALLtargets (a
proportion) and proceed as in the analysis of the fungi.
>
> # Let the responses from each participant stay together. They are what they are.
Participants stay in their age groups, too. Perhaps any apparent main effect for
Condition and any apparent AgeGroup by Condition interaction came from the random
ordering of presentation of the conditions.
>
> # The challenge is that it would be tough to come up with an ad hoc measure of
the interaction and then program it. It's preferable to just use an F statistic.
But calculating F requires a long (univariate) data setup, with one line for each
response. Set up the independent variables with repeats. Use them over and over,
repeatedly placing the shuffled responses (within conditions) beside them. There
are 4 conditions: ABCE.
>
> AgeGroupNames = as.character(sort(unique(AgeGroup)))
> n = dim(AllTargets)[1]; n
[1] 71
>
> Subject = rep(Participant,each=4) # 1,1,1,1, 2,2,2,2, ...
> Agegroup = rep(AgeGroup,each=4) # age1,age1,age1,age1, age2,age2,age2,age2, ...
> Condition = rep(c("A","B","C","E"),times=n) # A,B,C,E,  A,B,C,E, ...
> IVs = data.frame(Subject,Agegroup,Condition)
> head(IVs,10)
   Subject Agegroup Condition
1        1     4yos         A
2        1     4yos         B
3        1     4yos         C
4        1     4yos         E
5        2     5yos         A
6        2     5yos         B
7        2     5yos         C
8        2     5yos         E
9        4     5yos         A
10       4     5yos         B
> head(AllTargets)
  A.alltargets B.alltargets C.alltargets E.alltargets
1    0.1666667    0.3333333    0.0000000    0.0000000
2    0.5714286    0.5000000    0.1666667    0.0000000
3    0.2857143    0.5000000    0.3333333    0.1666667
4    0.4285714    0.0000000    0.1666667    0.0000000
5    0.4285714    0.6666667    0.5000000    0.1666667
6    0.2857143    0.0000000    0.0000000    0.0000000
```

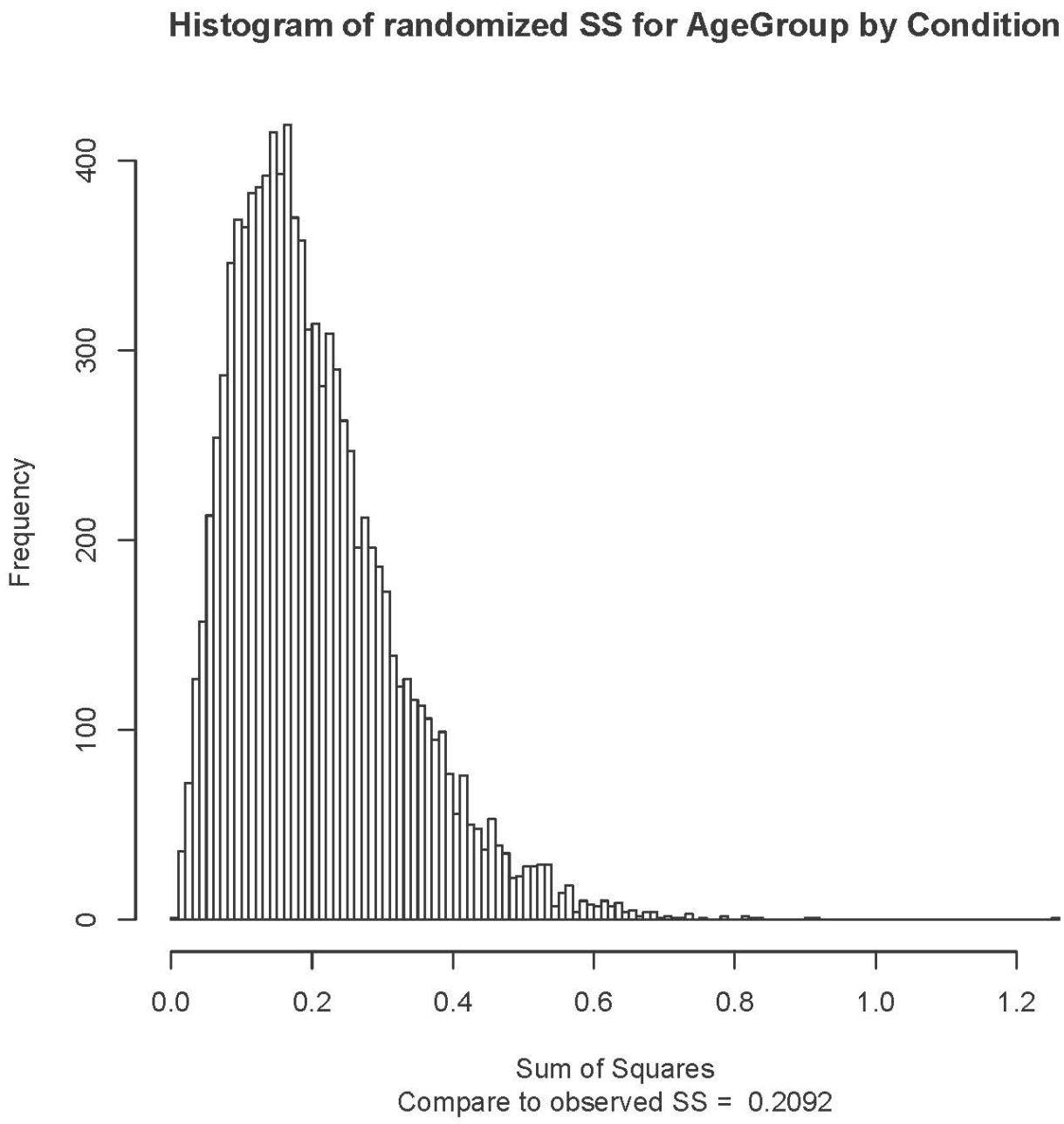
```
>
> # Get a column of proportions in the original order
> totalobs = prod(dim(AllTargets)) # 84*4 = 336
> Alltargets = t(AllTargets); dim(Alltargets) = c(totalobs,1)
> Truedata = cbind(IVs,Alltargets)
> head(Truedata,10)
   Subject Agegroup Condition Alltargets
1        1     4yos         A  0.1666667
2        1     4yos         B  0.3333333
3        1     4yos         C  0.0000000
4        1     4yos         E  0.0000000
5        2     5yos         A  0.5714286
6        2     5yos         B  0.5000000
7        2     5yos         C  0.1666667
8        2     5yos         E  0.0000000
9        4     5yos         A  0.2857143
10       4     5yos         B  0.5000000
>
> Truemodel = lm(Alltargets ~ Agegroup*Condition)
> Ssdf = aov(Truemodel) ; Ssdf # Use numerator SS as test statistics
Call:
   aov(formula = Truemodel)

Terms:
                Agegroup Condition Agegroup:Condition Residuals
Sum of Squares  2.485596  2.660661           0.209185  8.929444
Deg. of Freedom        2         3                  6       272

Residual standard error: 0.1811873
Estimated effects may be unbalanced
> Sumssdf = summary(Ssdf); Sumssdf
                    Df Sum Sq Mean Sq F value               Pr(>F)
Agegroup             2  2.486  1.2428  37.857 0.00000000000000313 ***
Condition            3  2.661  0.8869  27.015 0.00000000000000253 ***
Agegroup:Condition   6  0.209  0.0349   1.062                0.386
Residuals          272  8.929  0.0328
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> # Better compare a proper analysis of variance with repeated measures
> library(lme4) # For lmer function
> library(car)
> anova( lmer(Alltargets ~ Agegroup*Condition + (1 | Subject)) )
Analysis of Variance Table
                   Df  Sum Sq Mean Sq F value
Agegroup            2 0.72189 0.36095 17.7103
Condition           3 2.66066 0.88689 43.5163
Agegroup:Condition  6 0.20918 0.03486  1.7107
> Anova( lmer(Alltargets ~ Agegroup*Condition + (1 | Subject)), test="F" )
Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)

Response: Alltargets
                         F Df Df.res                Pr(>F)
Agegroup           17.7103  2     68          0.0000006437 ***
Condition          43.5163  3    204 < 0.00000000000000022 ***
Agegroup:Condition  1.7107  6    204                  0.12
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> # But I'm not going to randomize that.
>
```

```
> # Extract test statistics: Sums of squares for condition and interaction
> SScond = Sumssdf[[1]][[2]][2]; SScond # Had to experiment to locate this
[1] 2.660661
> SSinter = Sumssdf[[1]][[2]][3]; SSinter
[1] 0.2091848
>
> # Randomize. Shuffle each row of AllTargets, make it a single long column,
> #            and append it to IVs.
>
> set.seed(7777); nsim = 10000
> sscond = ssinter = numeric(nsim) # To hold simulated test statistics
>
> for(j in 1:nsim)
+     {
+     alltargets = AllTargets # Matrix of proportions
+     for(i in 1:n) alltargets[i,] = sample(alltargets[i,]) # Shuffle row i
+     alltargets = t(alltargets); dim(alltargets) = c(totalobs,1)
+     # Now alltargets is a single long column
+     summarytable = summary(aov( lm(alltargets ~ Agegroup*Condition) ))
+     sscond[j] = summarytable[[1]][[2]][2]  # SS for condition
+     ssinter[j] = summarytable[[1]][[2]][3] # SS for interaction
+     } # Next simulation j
>
>
> summary(cbind(sscond,ssinter))
     sscond             ssinter
 Min.   :0.0001491   Min.   :0.008545
 1st Qu.:0.0407810   1st Qu.:0.117295
 Median :0.0771445   Median :0.180532
 Mean   :0.0984432   Mean   :0.204123
 3rd Qu.:0.1346993   3rd Qu.:0.267536
 Max.   :0.7224512   Max.   :1.255891
> c(SScond,SSinter) # Compare randomized test statistics to these
[1] 2.6606614 0.2091848
```

```
> title1 = paste("Histogram of randomized SS for Condition")
> hist(sscond, breaks = 'fd', xlab = 'Sum of Squares', main=title1)
> sub1 = paste("Compare to observed SS = ",round(SScond,3))
> title(sub=sub1)
```



```
> title2 = paste("Histogram of randomized SS for AgeGroup by Condition")
> hist(ssinter, breaks = 'fd', xlab = 'Sum of Squares', main=title2)
> sub2 = paste("Compare to observed SS = ",round(SSinter,4))
> title(sub=sub2)
```




```
> # p-value
> length(ssinter[ssinter>=SSinter])/nsim
[1] 0.4057
```

For comparison, the p-value for this interaction with a mixed model on the binary data was 0.721.