Any Strictly Recursive Response

The binary variable *any* strictly recursive response for a condition is interesting.

```
> rm(list=ls()); options(scipen=999) # To avoid scientific notation
> rec1 = read.table("recursion-horizontal.data.txt",header=T)
>
> # Again adult vs child (between) by Condition ABCE, with the binary dependent
> # variable being any strictly recursive response.
>
> # Extract the variables we need.
> Group = rec1$Group
> Strict = rec1[,20:23]
> head(Strict); attach(Strict)
   A.Strict  B.Strict  C.strict E.Strict
1 0.1666667 0.3333333 0.0000000        0
2 0.5714286 0.5000000 0.1666667        0
3 0.2857143 0.0000000 0.1666667        0
4 0.2857143 0.0000000 0.0000000        0
5 0.4285714 0.1666667 0.5000000        0
6 0.2857143 0.0000000 0.0000000        0
> dim(Strict)
[1] 84  4
>
> table(A.Strict)
A.Strict
          0 0.142857143 0.166666667 0.285714286 0.333333333 0.428571429
         24          15           1          15           1           9
0.571428571         0.6 0.714285714 0.857142857           1
          8           1           5           3           2
> # Make binary variables.
> n = length(Group)
> anyA = anyB = anyC = anyE = numeric(n) # All zeros to start
> anyA[A.Strict>0] = 1; anyB[B.Strict>0] = 1
> anyC[C.strict>0] = 1; anyE[E.Strict>0] = 1
> table(Group,anyA)
       anyA
Group    0  1
  Adult  1 12
  Child 23 48
> chisq.test(table(Group,anyA))

      Pearson's Chi-squared test with Yates' continuity correction

data:  table(Group, anyA)
X-squared = 2.1865, df = 1, p-value = 0.1392

Warning message:
In chisq.test(table(Group, anyA)) :
  Chi-squared approximation may be incorrect
> chisq.test(table(Group,anyA),simulate.p.value=T)

      Pearson's Chi-squared test with simulated p-value (based on 2000
      replicates)

data:  table(Group, anyA)
X-squared = 3.2854, df = NA, p-value = 0.09795
```

```
> chisq.test(table(Group,anyA),simulate.p.value=T)

      Pearson's Chi-squared test with simulated p-value (based on 2000
      replicates)

data:  table(Group, anyA)
X-squared = 3.2854, df = NA, p-value = 0.09795

> fisher.test(table(Group,anyA),simulate.p.value=T)

      Fisher's Exact Test for Count Data

data:  table(Group, anyA)
p-value = 0.09716
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.003906053 1.331662116
sample estimates:
odds ratio
 0.1765267

> anyStrict = cbind(anyA, anyB, anyC, anyE)
> cor(anyStrict)
          anyA       anyB       anyC       anyE
anyA 1.0000000 0.31622777 0.39131190 0.10758287
anyB 0.3162278 1.00000000 0.30304576 0.06804138
anyC 0.3913119 0.30304576 1.00000000 0.07216878
anyE 0.1075829 0.06804138 0.07216878 1.00000000
>
> # Look at means (sample proportions of Yes responses)
> aggregate(anyStrict,by=list(Group),FUN=mean)
  Group.1      anyA      anyB      anyC       anyE
1   Adult 0.9230769 0.7692308 0.6153846 0.46153846
2   Child 0.6760563 0.4507042 0.2816901 0.08450704
> Means = as.matrix(aggregate(anyStrict,by=list(Group),FUN=mean)[,2:5])
> rownames(Means) = c("Adult","Child")
> addmargins(Means,FUN=mean, quiet=T)
           anyA      anyB      anyC       anyE      mean
Adult 0.9230769 0.7692308 0.6153846 0.46153846 0.6923077
Child 0.6760563 0.4507042 0.2816901 0.08450704 0.3732394
mean  0.7995666 0.6099675 0.4485374 0.27302275 0.5327736
>
> # Calculate marginal Condition means and Adult-Child differences
> Conmeans = apply(Means,2,mean); Conmeans
     anyA      anyB      anyC      anyE
0.7995666 0.6099675 0.4485374 0.2730228
> D = Means[1,]-Means[2,]; D # Adult minus Child
     anyA      anyB      anyC      anyE
0.2470206 0.3185265 0.3336945 0.3770314
> rbind(Means,D)
           anyA      anyB      anyC       anyE
Adult 0.9230769 0.7692308 0.6153846 0.46153846
Child 0.6760563 0.4507042 0.2816901 0.08450704
D     0.2470206 0.3185265 0.3336945 0.37703142
>
> # Marginal Condition means and Mean Adult-Child differences
> Conmeans = apply(Means,2,mean); Conmeans
     anyA      anyB      anyC      anyE
0.7995666 0.6099675 0.4485374 0.2730228
> D = Means[1,]-Means[2,]; D # Row 1 minus row 2
     anyA      anyB      anyC      anyE
0.2470206 0.3185265 0.3336945 0.3770314
>
> # Test statistics
> # Main effect for condition
```

```
> Vmeans0 = var(Conmeans) # Variance of the marginal means
> # Group by Condition interaction
> Vdiff0 = var(D) # Variance of the differences
>
> # Randomize
>
> set.seed(9999); nsim = 10000
> Vmeans = Vdiff = numeric(nsim) # To hold simulated test statistics
> n = length(Group); n
[1] 84
>
> for(j in 1:nsim)
+     {
+     strict = Strict
+     for(i in 1:n) strict[i,] = sample(strict[i,]) # Shuffle row i
+     # Make a table of means for the shuffled data
+     means = as.matrix(aggregate(strict,by=list(Group),FUN=mean)[,2:5])
+     # Compute test statistics
+     conmeans = apply(means,2,mean)
+     d = means[1,]-means[2,]
+     # Store the test statistics
+     Vmeans[j] = var(conmeans)
+     Vdiff[j]  = var(d)
+     } # Next simulation j
>
> #  p-values
> # Main effect for condition
> p1 = length(Vmeans[Vmeans>=Vmeans0])/nsim
> c(Vmeans0,p1)
[1] 0.05056789 0.00000000
>
> # Group by Condition interaction
> p2 = length(Vdiff[Vdiff>=Vdiff0])/nsim
> c(Vdiff0,p2) # 0.7686
[1] 0.002921605 0.768600000
>
> # p-value for proportion Strict recursion was 0.1344
>
```