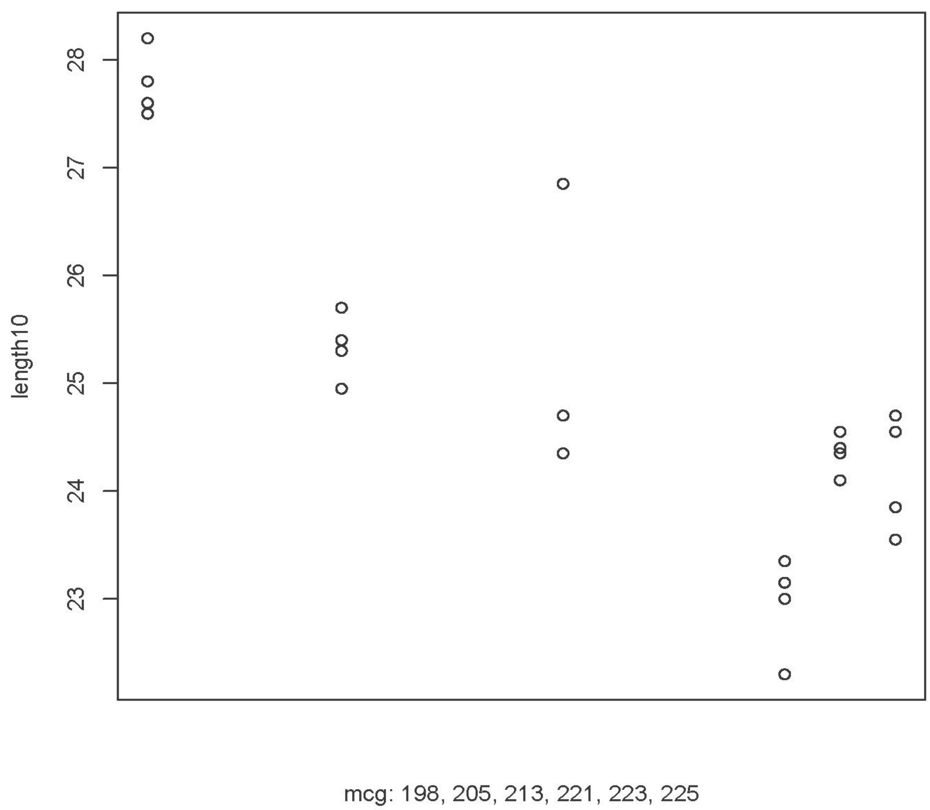
Randomization tests on the tubes data (between cases)

Fungus of different genetic types was grown in a test tubes for 14 days. Twice a day, the investigators' graduate students measured the length of the fungus in the tube and other variables. In this analysis, the independent variable is type of fungus (mycelial compatibility group or MCG), and the dependent variable is the length of the fungus in the tube at day 10: average of morning and evening measurements.

```
> rm(list=ls()); options(scipen=999) # To avoid scientific notation
> # install.packages("car") # Just do this once
> library(car)  # For F-tests, likelihood ratio and Wald chi-squared tests
> tubes =
read.table("http://www.utstat.toronto.edu/~brunner/data/legal/LittleTubes2.data.txt")
> # tubes = read.table("LittleTubes2.data.txt") # Local copy
> head(tubes); attach(tubes)
  id mcg length10 sclr10 weight  rate
1  1 198    27.80   26.0 0.5996 3.145
2 15 198    28.20   27.0 0.6040 3.235
3 29 198    27.60   28.0 0.6172 3.160
4 43 198    27.50   24.5 0.6053 3.140
5 57 205    24.95   28.0 0.6769 2.975
6 71 205    25.70   29.0 0.7057 2.975
> table(mcg)
mcg
198 205 213 221 223 225
  4   4   4   4   4   4
>
> plot(mcg,length10,xaxt='n', xlab = 'mcg: 198, 205, 213, 221, 223, 225')
```

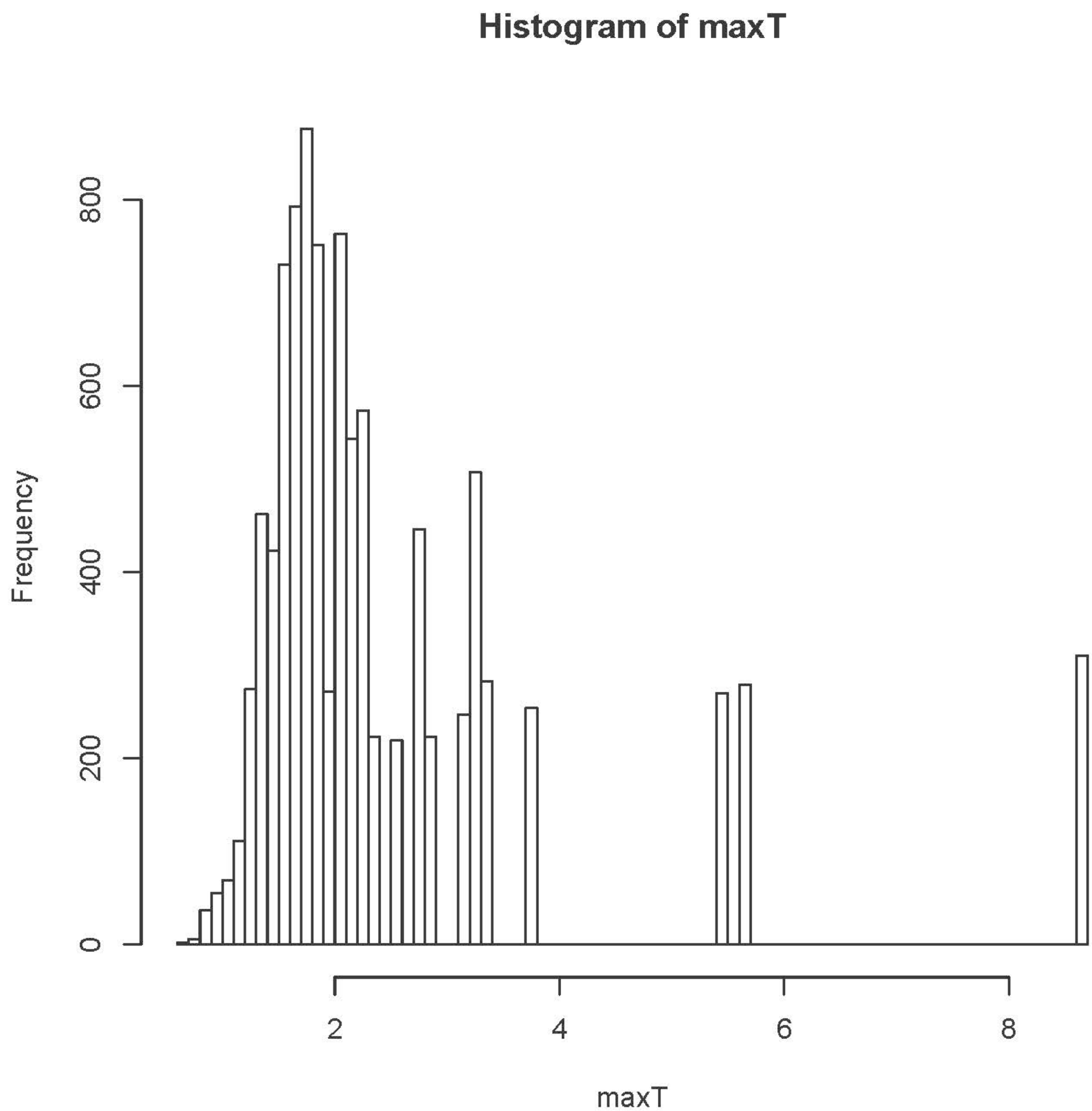



```
> # Note outlier, and also that mcg=198 is a monster. Set it aside and compare the others.
```

```
> length10=length10[mcg!=198]; mcg=mcg[mcg!=198]
> MCG=factor(mcg)
> aggregate(length10,by=list(MCG),FUN=mean) # Look at means
  Group.1       x
1     205 25.3375
2     213 25.0625
3     221 22.9500
4     223 24.3500
5     225 24.1625
> # Ordinary one-way ANOVA -- questionable because of the outlier.
> mod0 = lm(length10~MCG); anova(mod0)
Analysis of Variance Table

Response: length10
          Df  Sum Sq Mean Sq F value    Pr(>F)    
MCG        4 13.9017  3.4754  8.3136 0.0009646 ***
Residuals 15  6.2706  0.4180                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> Fstat0 = summary(mod0)$fstatistic[1]  # Just the F statistic
> Fstat0
   value 
8.313615 
> # Multiple comparisons
> TukeyHSD(aov(mod0),ordered=T)
  Tukey multiple comparisons of means
    95% family-wise confidence level
    factor levels have been ordered

Fit: aov(formula = mod0)

$MCG
          diff        lwr      upr     p adj
225-221 1.2125 -0.1992619 2.624262 0.1100107
223-221 1.4000 -0.0117619 2.811762 0.0524505
213-221 2.1125  0.7007381 3.524262 0.0026214
205-221 2.3875  0.9757381 3.799262 0.0008387
223-225 0.1875 -1.2242619 1.599262 0.9934146
213-225 0.9000 -0.5117619 2.311762 0.3263124
205-225 1.1750 -0.2367619 2.586762 0.1267851
213-223 0.7125 -0.6992619 2.124262 0.5434644
205-223 0.9875 -0.4242619 2.399262 0.2467446
205-213 0.2750 -1.1367619 1.686762 0.9726917

> # MCG 221 is different from (lower than) 213 and 205
> 
> # Randomization test
> # There are 305,540,235,000 ways to scramble the data
> set.seed(9999); nsim=10000; Fstat=numeric(nsim)
> for(j in 1:nsim)
+     {
+     y = sample(length10) # Sample with replacement = random permutation
+     Fstat[j] = summary(lm(y~mcg))$fstatistic[1]
+     } # Next simulation
> Fstat[1:10] # Look at the first 10
 [1] 0.45587198 0.07947568 0.38558024 2.60447372 0.39070831 2.39021682 0.57327407 0.61473615
 [9] 0.18128631 0.79134788
> length(Fstat[Fstat>=Fstat0])/nsim
[1] 0.0094
> # Compare p = 0.0009646    Still significant, but 10 times as large
> # Doing it a few more times, got p = 0.0091, 0.0109, 0.0093, 0.0091
```

```
> kruskal.test(length10~MCG) # Kruskal-Wallis one way analysis of variance by ranks

	Kruskal-Wallis rank sum test

data:  length10 by MCG
Kruskal-Wallis chi-squared = 13.924, df = 4, p-value = 0.007543

>
> # The coin package (COnditionalINference) has various permutation tests.
> # install.packages("coin") # Only need to do this once
> library(coin)

> # help(oneway_test)
> oneway_test(length10~MCG) # p = 0.01083

	Asymptotic K-Sample Fisher-Pitman Permutation Test

data:  length10 by MCG (205, 213, 221, 223, 225)
chi-squared = 13.094, df = 4, p-value = 0.01083

>
> # Multiple comparisons via randomization
> # Compute all pairwise t-tests (unequal variance assumption)
> tvals = matrix(0,5,5)
> mcgval = sort(unique(mcg))
> rownames(tvals) = colnames (tvals) = mcgval
> for(Row in 1:4)
+     {
+     for(Col in (Row+1):5)
+         {
+         A = length10[mcg==mcgval[Row]]
+         B = length10[mcg==mcgval[Col]]
+         tvals[Row,Col] = abs( t.test(A,B)$statistic )
+         }
+     }
> tvals
    205       213      221      223       225
205   0 0.4427835 8.660966 5.464544 3.7176130
213   0 0.0000000 3.283566 1.170437 1.3601864
221   0 0.0000000 0.000000 5.676194 3.3880756
223   0 0.0000000 0.000000 0.000000 0.6441071
225   0 0.0000000 0.000000 0.000000 0.0000000

> # Find critical values of maximum absolute t statistic. If null hypothesis
> # is true, the probability that all are below equals 0.95, and the
> # probability of one or more is (wrongly) above is 0.05.
>
> simtvals = tvals-tvals # Hold simulated t-values in matrix simtvals
> set.seed(8888)
> nsim = 10000; maxT = numeric(nsim)
> for(j in 1:nsim)
+     {
+     for(Row in 1:4)
+         {
+         for(Col in (Row+1):5)
+             {
+             datta = c(length10[mcg==mcgval[Row]],
+                       length10[mcg==mcgval[Col]])   # Put data in one pot
+             scramble = sample(datta)                # Mix it up
+             A = scramble[1:4]; B = scramble[5:8]
+             simtvals[Row,Col] = abs( t.test(A,B)$statistic )
+             } # Next Col in the table
+         } # Next Row in the table
+     maxT[j] = max(simtvals)
+     } # Next simulation j
```

```
> hist(maxT, breaks='fd')
```



```
> length(unique(maxT))
[1] 136
> # There are only 136 unique values, because the sample sizes are so small.
> table(maxT)
maxT
0.631393473296017 0.686061604809112 0.710909286751898 0.799021429604816 0.799144411695704 
                1                 1                 1                 3                 2 
0.804032108985107  0.82092803241773 0.820928032417733 0.826767381914332 0.833472299723665 
                1                 1                 2                 2                 1 
0.837218358278921 0.837860948440368 0.842489683950126 0.850230301897703 0.878365973360015 
                1                 1                 3                 3                 5 
0.878431740526561 0.884390199085696 0.886681576989414 0.893064637781637 0.903994441358834 
                5                 3                 5                 4                 2 
0.917044355745298 0.931652523402094 0.932448967957128 0.933256525257381 0.933256525257395 
                3                16                 9                 3                 2 
0.938253160996928 0.939393939393947 0.942912927702791 0.957591317878072 0.977461894333811 
                6                 2                 1                 3                 8 
 1.00994121985473  1.01923755806376  1.02417347552625   1.0372715186439  1.04582256735224 
                3                 5                17                 3                 6 
 1.07185598641969  1.08359046574615  1.09906734049839  1.10074141558049  1.10959399278472 
               20                 8                 7                14                12 
 1.12058367048475  1.12757446635522  1.14627454820992  1.15470053837926  1.15794745626015 
                9                 9                11                 8                10 
 1.16075889974966  1.17043696905226  1.21434602004128  1.21722333737947  1.21722333737948 
               13                25                25                17                15 
 1.22150329338751  1.22874585642994  1.23596912797325  1.24393264329409  1.26116828954168 
               41                16                16                45                38 
 1.26577242682571  1.26724522422693  1.30060972317916  1.31710976071664  1.32389270974355 
               45                16                31                20                22 
 1.33520587983849  1.33672278941056  1.33916314673729  1.35896476250056  1.36018640868858 
               31                23                51                51                65 
 1.37201584386816  1.38297872340425  1.38304473424466  1.40175993827342  1.40954724433621 
               32                68                68                42                46 
   1.439944828299  1.44265656103573  1.44564436859466  1.44783958657606  1.45087383528466 
               24                57                56                47                53 
 1.49037128976401  1.50961311293385  1.50961311293386  1.51627586174069  1.52687907035753 
               98                49                54                61                59 
 1.54303349962092  1.56913026454941  1.59399690361129  1.60951003943655  1.61235856741763 
               57               234               216                72                74 
 1.64473347565137  1.65344827962325  1.68166904394841  1.68167564648372  1.69282947806063 
              101                96                87                77                96 
 1.69616651560716  1.71636003902653  1.71652792061004  1.72806282553573  1.74340243627067 
              189                87               192               118               241 
 1.77281052085584  1.79106694237798  1.80578779628654  1.82915960136083  1.83482698437864 
              105               133               124               136               120 
 1.86917960927774  1.89467013843749  1.94494475178437  1.96937014151971  2.00713014739239 
              115               256               121               151               163 
 2.01504408998011  2.03930016560806  2.05815659595423  2.07763008319333  2.14269564023653 
              143               142               170               145               172 
 2.15890677135333  2.22408237769114   2.2514481561646  2.26668570332764  2.35392939710549 
              371               182               185               206               223 
  2.5988129407631  2.72378086566882  2.74562589193457  2.89071939090841  3.18177346164191 
              219               219               227               223               247 
  3.2305139641303  3.28356581394543  3.38807563773945  3.71761301197124  5.46454426454198 
              254               253               283               254               270 
 5.67619397855569  8.66096629812587 
              279               310 
> 0.05*10000
[1] 500
> # Critical value should cut off the top 500 values
> # 310/10,000 = 0.031, and (279+310)/10,000 = 0.0589, so t > 5.68
> # is required for significance. Only values of t = 8.660966 are
> # significant, at the 0.031 level. Alpha = 0.05 is not available.
> # That's mcg 205 versus 221 (highest versus lowest)
```