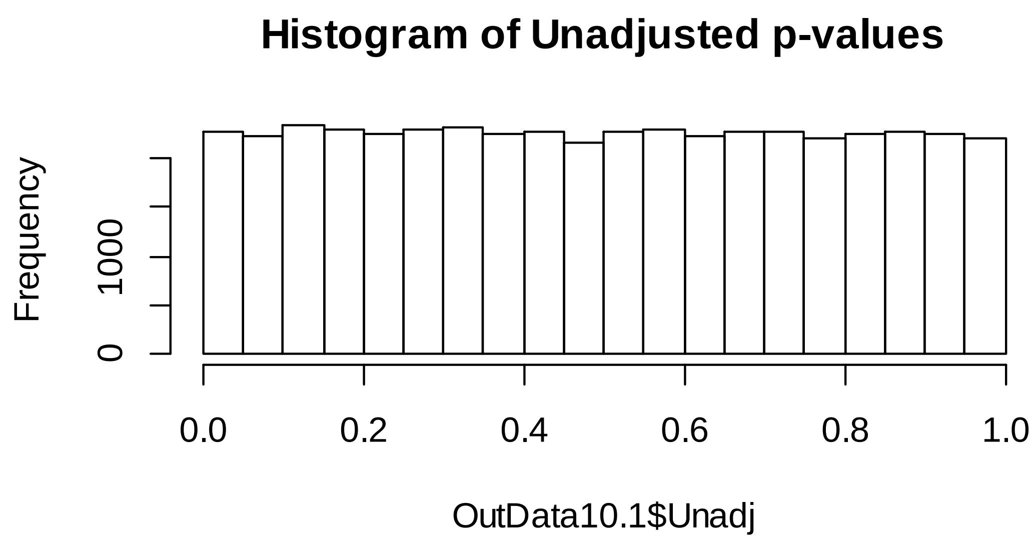
#Multiple Testing Simulation

For Illustration: This is not a basic data analysis example! We simulate data where H0 is true (no difference between means) with t = 5 or 10 groups and n = 10 observations per group. For each scenario, we consider 1000 “runs” where 1 run = 1 ANOVA = 1 “experiment”. Then compare CER (comparison-wise error rate) and EER (experiment-wise error rate) of unadjusted (LSD), Tukey (HSD) and Bonferroni pairwise comparisons.

```
> library(emmeans)
> library(plyr)
> library(tidyverse)
> library(broom)
> #t = 10  groups
> #n = 10 observations per group
> t = 10; n = 10
> Groups <- sort(rep(seq(1, t), n))
> Groups <- as.factor(Groups)
>
> set.seed(5729)
> SimData <- data.frame(SampleID = sort(rep(seq(1, 1000), length
(Groups))),
+                        Group = rep(Groups, 1000),
+                        Y = rnorm(1000*length(Groups)))
>
> summary(SimData)
    SampleID          Group             Y
 Min.   :   1.0   1      :10000   Min.   :-4.251903
 1st Qu.: 250.8   2      :10000   1st Qu.:-0.671399
 Median : 500.5   3      :10000   Median : 0.004226
 Mean   : 500.5   4      :10000   Mean   :-0.000067
 3rd Qu.: 750.2   5      :10000   3rd Qu.: 0.675631
 Max.   :1000.0   6      :10000   Max.   : 4.558262
                  (Other):40000
> str(subset(SimData, SampleID == 1))
'data.frame':      100 obs. of  3 variables:
 $ SampleID: int  1 1 1 1 1 1 1 1 1 1 ...
 $ Group   : Factor w/ 10 levels "1","2","3","4",..: 1 1 1 1 ...
 $ Y       : num  -1.003 0.811 -1.898 -0.339 -0.578 ...
>
> OutData10.1 <- ddply(SimData, c("SampleID"),
+                 function(d) {
+                 oneway <- lm(Y ~ Group, data = d)
+                 emout <- emmeans(oneway, ~ Group)
+                 emout1 <- tidy(pairs(emout))
+                 emout2 <- tidy(pairs(emout, adjust = "none"))
+                 emout3 <- tidy(pairs(emout, adjust = "bonferron
i"))
```

```
+                 data.frame(Tukey = emout1$p.value,
+                            Unadj = emout2$p.value,
+                            Bon   = emout3$p.value)
+                       }
+ )

> str(OutData10.1)
'data.frame':      45000 obs. of  4 variables:
 $ SampleID: int  1 1 1 1 1 1 1 1 1 1 ...
 $ Tukey   : num  1 1 1 1 0.994 ...
 $ Unadj   : num  0.621 0.869 0.538 0.938 0.347 ...
 $ Bon     : num  1 1 1 1 1 1 1 1 1 1 ...
> hist(OutData10.1$Unadj, main = "Histogram of Unadjusted p-valu
es")
```




```
> OutData10.2 <- OutData10.1 %>%
+                 group_by(SampleID) %>%
+                 summarise(
+                   minTukey = min(Tukey),
+                   minUnadj = min(Unadj),
+                   minBon   = min(Bon))
> str(OutData10.2)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':    1000 obs. of  4 vari
ables:
 $ SampleID: int  1 2 3 4 5 6 7 8 9 10 ...
 $ minTukey: num  0.812 0.923 0.194 0.892 0.15 ...
 $ minUnadj: num  0.09952 0.1637 0.00852 0.13931 0.00611 ...
 $ minBon  : num  1 1 0.383 1 0.275 ...
```

```
> #CER t = 10
> CER <- (OutData10.1[,-1] < 0.05)
> colSums(CER)/nrow(CER)
       Tukey        Unadj          Bon
0.0012222222 0.0502444444 0.0008444444
>
> #EER t = 10
> EER <- (OutData10.2[,-1] < 0.05)
> colSums(EER)/nrow(EER)
minTukey minUnadj   minBon
   0.040    0.610    0.028

#Similarly for t = 5 groups
#Simulation Code not shown
> t = 5; n = 10
> #CER t = 5
> CER <- (OutData5.1[,-1] < 0.05)
> colSums(CER)/nrow(CER)
 Tukey  Unadj    Bon
0.0075 0.0546 0.0051
>
> #EER t = 5
> EER <- (OutData5.2[,-1] < 0.05)
> colSums(EER)/nrow(EER)
minTukey minUnadj   minBon
   0.055    0.293    0.039
```