November 2, 2020

1: Please describe the two main classes of multiple comparisons procedures. When should they be used?

The two main classes of multiple comparison are: Planned multiple comparison: If a multiple comparison scheduled and run before looking into of knowing the data not presuming the results after running the test is called as planned multiple comparison. Test is decided before conducting anova omnibus test no matter what result will be obtained. For instance, in several groups only few are selected, and means are compared. This type of comparison is used to test hypothesis and it has no control on experiment-based results. This is done even if F is not significant, degree of freedom is more than comparison.

Post-hoc multiple comparison: A multiple post-hoc comparison is more prospective meaning test can be decided after conducting anova omnibus test and reject null hypothesis. When need to look for something specific or interesting after looking at the details of the data then the comparison is called as post hoc comparison. This is used to see which mean is different in group. It is used to see summary. This test is used when F is significant, and degree of freedom is less than comparison in it. Thus, this is done when comparison is not planned and an unforeseen anova effect is discovered

2: Please describe the rationale and behind the False Discovery Rate procedures. The, specifically describe how the Holm and the Benjamini-Hochberg (BH) procedures work. While doing more than one planned multiple comparison there is chance of incorrect rejection of null hypothesis then it is type 1 error (inflated alpha) and the expected proportion of that is false discovery ratio (FDR). For instance, when type 1 error occurs then a person will be false positive with a disease which mean there is no disease in real. An adjustment for the multiplicity of experiments is also required to minimize the total number of false observations when several theories are simultaneously evaluated.

Holm-Bonferroni Procedure: When conducting multiple tests, the Bonferroni correction lowers the probability of Type I error. While it is easy to measure the Bonferroni, it suffers from a lack of statistical capacity. Modified Bonferroni correction is Holm-Bonferroni method (Holm’s Sequential Bonferroni Procedure) for multiple hypothesis tests is a way to deal with family error rates (FWER). It is a change to the correction of Bonferroni. It is also fairly easy to calculate the Holm-Bonferroni process, but it is more efficient than the Bonferroni single-step. It has least power than BH procedure. It can be done by simply dividing target alpha by number of tests. i.e. ai = a / J Benjamini-Hochberg Procedure: it is a more powerful tool that lower the false discovery rate. In this procedure individual p-values are put into ascending order, then ranks are assigned and critical value is calculated by dividing individual p-value rank by number of tests and then multiplied by FDR . i.e. (i/m)Q

3: Please find the data set drugTx.csv on Canvas. This describes an experiment where three different doses of a drug were given (one was placebo) and tumor volume was measured. Please evaluate data these to determine if the drug influenced tumor volume and which specific dose might have had an effect. There is no strong a prior hypothesis to suggest which dose should work. Don’t forget to test all assumptions and present your conclusions in APA accepted format.

#loading libraries equired  
library(ggplot2)  
library(pgirmess)

## Warning: package 'pgirmess' was built under R version 4.0.3

library(DescTools)

## Warning: package 'DescTools' was built under R version 4.0.3

## Registered S3 method overwritten by 'DescTools':  
## method from   
## reorder.factor gdata

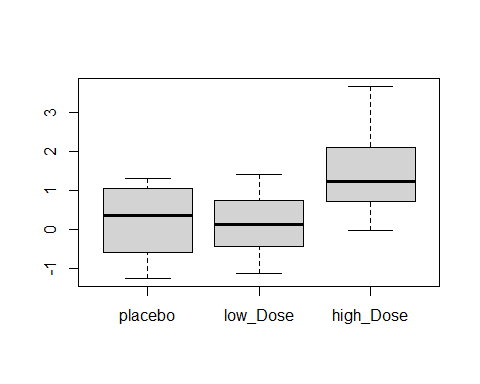
# Reading file  
data = read.csv("drugTx.csv" )  
  
data #looking into data , seems there is NA

## placebo low\_Dose high\_Dose  
## 1 1.12468414 -0.500174495 1.9995809  
## 2 0.52358165 -1.143150211 0.7296775  
## 3 -1.11031233 1.149536456 1.1157366  
## 4 -1.26172273 -0.055869102 1.4537068  
## 5 0.35107508 0.006137063 2.9248627  
## 6 0.69150048 0.375916509 -0.0331015  
## 7 1.24062324 -0.472048423 0.6965438  
## 8 -0.43563091 -0.391721381 3.6724432  
## 9 -0.86116184 1.054743168 3.1661001  
## 10 1.30585117 0.375499071 1.2324254  
## 11 0.04710268 -0.313344826 2.2023778  
## 12 0.98398272 0.780540801 0.3396470  
## 13 -0.68196023 1.404602289 0.7471397  
## 14 1.26410573 0.715439012 0.3307367  
## 15 -0.52883046 -0.641362218 1.6826386  
## 16 NA 0.250000000 NA

colnames(data)

## [1] "placebo" "low\_Dose" "high\_Dose"

# initial plotting to check distribution  
  
boxplot(data)



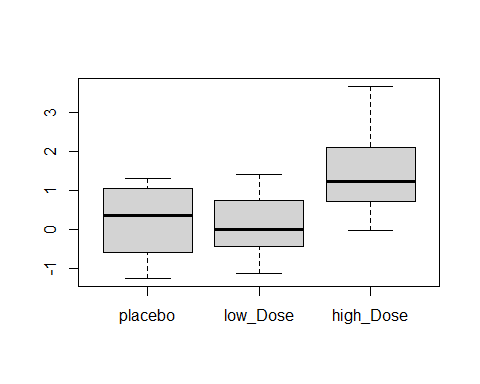
# checking NA  
  
data[!complete.cases(data),]

## placebo low\_Dose high\_Dose  
## 16 NA 0.25 NA

# removing NA  
  
clean\_data<-na.omit(data)  
  
clean\_data #checking again

## placebo low\_Dose high\_Dose  
## 1 1.12468414 -0.500174495 1.9995809  
## 2 0.52358165 -1.143150211 0.7296775  
## 3 -1.11031233 1.149536456 1.1157366  
## 4 -1.26172273 -0.055869102 1.4537068  
## 5 0.35107508 0.006137063 2.9248627  
## 6 0.69150048 0.375916509 -0.0331015  
## 7 1.24062324 -0.472048423 0.6965438  
## 8 -0.43563091 -0.391721381 3.6724432  
## 9 -0.86116184 1.054743168 3.1661001  
## 10 1.30585117 0.375499071 1.2324254  
## 11 0.04710268 -0.313344826 2.2023778  
## 12 0.98398272 0.780540801 0.3396470  
## 13 -0.68196023 1.404602289 0.7471397  
## 14 1.26410573 0.715439012 0.3307367  
## 15 -0.52883046 -0.641362218 1.6826386

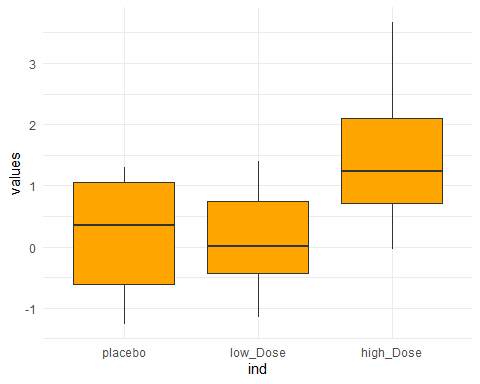
# plotting again to check distribution  
  
boxplot(clean\_data)



# stacking data  
  
stacked\_data<-stack(clean\_data)  
stacked\_data

## values ind  
## 1 1.124684136 placebo  
## 2 0.523581647 placebo  
## 3 -1.110312328 placebo  
## 4 -1.261722727 placebo  
## 5 0.351075078 placebo  
## 6 0.691500478 placebo  
## 7 1.240623243 placebo  
## 8 -0.435630913 placebo  
## 9 -0.861161843 placebo  
## 10 1.305851167 placebo  
## 11 0.047102684 placebo  
## 12 0.983982715 placebo  
## 13 -0.681960228 placebo  
## 14 1.264105726 placebo  
## 15 -0.528830464 placebo  
## 16 -0.500174495 low\_Dose  
## 17 -1.143150211 low\_Dose  
## 18 1.149536456 low\_Dose  
## 19 -0.055869102 low\_Dose  
## 20 0.006137063 low\_Dose  
## 21 0.375916509 low\_Dose  
## 22 -0.472048423 low\_Dose  
## 23 -0.391721381 low\_Dose  
## 24 1.054743168 low\_Dose  
## 25 0.375499071 low\_Dose  
## 26 -0.313344826 low\_Dose  
## 27 0.780540801 low\_Dose  
## 28 1.404602289 low\_Dose  
## 29 0.715439012 low\_Dose  
## 30 -0.641362218 low\_Dose  
## 31 1.999580929 high\_Dose  
## 32 0.729677528 high\_Dose  
## 33 1.115736568 high\_Dose  
## 34 1.453706788 high\_Dose  
## 35 2.924862707 high\_Dose  
## 36 -0.033101504 high\_Dose  
## 37 0.696543817 high\_Dose  
## 38 3.672443234 high\_Dose  
## 39 3.166100124 high\_Dose  
## 40 1.232425388 high\_Dose  
## 41 2.202377751 high\_Dose  
## 42 0.339647028 high\_Dose  
## 43 0.747139674 high\_Dose  
## 44 0.330736663 high\_Dose  
## 45 1.682638604 high\_Dose

## plotting  
ggplot(stacked\_data) + aes(x = ind, y = values) + geom\_boxplot(fill = "orange") + theme\_minimal()



## Planned comparisons (no protection for alpha inflation)  
pairwise.t.test(stacked\_data$values,stacked\_data$ind,paired=FALSE,p.adjust.method="none")

##   
## Pairwise comparisons using t tests with pooled SD   
##   
## data: stacked\_data$values and stacked\_data$ind   
##   
## placebo low\_Dose  
## low\_Dose 0.95266 -   
## high\_Dose 0.00046 0.00038   
##   
## P value adjustment method: none

#pairwise.t.test(stacked\_data$values,stacked\_data$ind,paired=TRUE,p.adjust.method="none")  
## Bonferroni  
pairwise.t.test(stacked\_data$values,stacked\_data$ind,paired=FALSE,p.adjust.method="bonferroni")

##   
## Pairwise comparisons using t tests with pooled SD   
##   
## data: stacked\_data$values and stacked\_data$ind   
##   
## placebo low\_Dose  
## low\_Dose 1.0000 -   
## high\_Dose 0.0014 0.0012   
##   
## P value adjustment method: bonferroni

## Holm  
pairwise.t.test(stacked\_data$values,stacked\_data$ind,paired=FALSE,p.adjust.method="holm")

##   
## Pairwise comparisons using t tests with pooled SD   
##   
## data: stacked\_data$values and stacked\_data$ind   
##   
## placebo low\_Dose  
## low\_Dose 0.9527 -   
## high\_Dose 0.0012 0.0012   
##   
## P value adjustment method: holm

# Running ANOVA using aov  
x=aov(values~ind,stacked\_data)  
  
# summary of ANOVA results  
summary(x)

## Df Sum Sq Mean Sq F value Pr(>F)   
## ind 2 17.36 8.680 9.783 0.000325 \*\*\*  
## Residuals 42 37.26 0.887   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## post-hocs   
## Simple Tukey's HSD  
TukeyHSD(x)

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = values ~ ind, data = stacked\_data)  
##   
## $ind  
## diff lwr upr p adj  
## low\_Dose-placebo -0.02054298 -0.8561537 0.8150678 0.9980352  
## high\_Dose-placebo 1.30717513 0.4715644 2.1427859 0.0013115  
## high\_Dose-low\_Dose 1.32771811 0.4921074 2.1633289 0.0010985

TukeyHSD(x, conf.level=0.99)

## Tukey multiple comparisons of means  
## 99% family-wise confidence level  
##   
## Fit: aov(formula = values ~ ind, data = stacked\_data)  
##   
## $ind  
## diff lwr upr p adj  
## low\_Dose-placebo -0.02054298 -1.0796663 1.038580 0.9980352  
## high\_Dose-placebo 1.30717513 0.2480518 2.366298 0.0013115  
## high\_Dose-low\_Dose 1.32771811 0.2685947 2.386841 0.0010985

## Dunnett's Test  
stacked\_data$ind <- as.factor(stacked\_data$ind)  
#DunnettTest(x = stacked\_data$values, g = stacked\_data$ind, control = 'None')  
  
  
# Nonparametrics  
  
## Multiple comparisons with wilcox rank sum   
  
pairwise.wilcox.test(clean\_data$placebo,clean\_data$low\_Dose,paired=FALSE,p.adjust.method="none",exact=TRUE)

##   
## Pairwise comparisons using Wilcoxon rank sum exact test   
##   
## data: clean\_data$placebo and clean\_data$low\_Dose   
##   
## -1.143150211 -0.641362218 -0.500174495 -0.472048423 -0.391721381  
## -0.641362218 1 - - - -   
## -0.500174495 1 1 - - -   
## -0.472048423 1 1 1 - -   
## -0.391721381 1 1 1 1 -   
## -0.313344826 1 1 1 1 1   
## -0.055869102 1 1 1 1 1   
## 0.006137063 1 1 1 1 1   
## 0.375499071 1 1 1 1 1   
## 0.375916509 1 1 1 1 1   
## 0.715439012 1 1 1 1 1   
## 0.780540801 1 1 1 1 1   
## 1.054743168 1 1 1 1 1   
## 1.149536456 1 1 1 1 1   
## 1.404602289 1 1 1 1 1   
## -0.313344826 -0.055869102 0.006137063 0.375499071 0.375916509  
## -0.641362218 - - - - -   
## -0.500174495 - - - - -   
## -0.472048423 - - - - -   
## -0.391721381 - - - - -   
## -0.313344826 - - - - -   
## -0.055869102 1 - - - -   
## 0.006137063 1 1 - - -   
## 0.375499071 1 1 1 - -   
## 0.375916509 1 1 1 1 -   
## 0.715439012 1 1 1 1 1   
## 0.780540801 1 1 1 1 1   
## 1.054743168 1 1 1 1 1   
## 1.149536456 1 1 1 1 1   
## 1.404602289 1 1 1 1 1   
## 0.715439012 0.780540801 1.054743168 1.149536456  
## -0.641362218 - - - -   
## -0.500174495 - - - -   
## -0.472048423 - - - -   
## -0.391721381 - - - -   
## -0.313344826 - - - -   
## -0.055869102 - - - -   
## 0.006137063 - - - -   
## 0.375499071 - - - -   
## 0.375916509 - - - -   
## 0.715439012 - - - -   
## 0.780540801 1 - - -   
## 1.054743168 1 1 - -   
## 1.149536456 1 1 1 -   
## 1.404602289 1 1 1 1   
##   
## P value adjustment method: none

pairwise.wilcox.test(clean\_data$placebo,clean\_data$high\_Dose,paired=FALSE,p.adjust.method="none",exact=TRUE)

##   
## Pairwise comparisons using Wilcoxon rank sum exact test   
##   
## data: clean\_data$placebo and clean\_data$high\_Dose   
##   
## -0.033101504 0.330736663 0.339647028 0.696543817 0.729677528  
## 0.330736663 1 - - - -   
## 0.339647028 1 1 - - -   
## 0.696543817 1 1 1 - -   
## 0.729677528 1 1 1 1 -   
## 0.747139674 1 1 1 1 1   
## 1.115736568 1 1 1 1 1   
## 1.232425388 1 1 1 1 1   
## 1.453706788 1 1 1 1 1   
## 1.682638604 1 1 1 1 1   
## 1.999580929 1 1 1 1 1   
## 2.202377751 1 1 1 1 1   
## 2.924862707 1 1 1 1 1   
## 3.166100124 1 1 1 1 1   
## 3.672443234 1 1 1 1 1   
## 0.747139674 1.115736568 1.232425388 1.453706788 1.682638604  
## 0.330736663 - - - - -   
## 0.339647028 - - - - -   
## 0.696543817 - - - - -   
## 0.729677528 - - - - -   
## 0.747139674 - - - - -   
## 1.115736568 1 - - - -   
## 1.232425388 1 1 - - -   
## 1.453706788 1 1 1 - -   
## 1.682638604 1 1 1 1 -   
## 1.999580929 1 1 1 1 1   
## 2.202377751 1 1 1 1 1   
## 2.924862707 1 1 1 1 1   
## 3.166100124 1 1 1 1 1   
## 3.672443234 1 1 1 1 1   
## 1.999580929 2.202377751 2.924862707 3.166100124  
## 0.330736663 - - - -   
## 0.339647028 - - - -   
## 0.696543817 - - - -   
## 0.729677528 - - - -   
## 0.747139674 - - - -   
## 1.115736568 - - - -   
## 1.232425388 - - - -   
## 1.453706788 - - - -   
## 1.682638604 - - - -   
## 1.999580929 - - - -   
## 2.202377751 1 - - -   
## 2.924862707 1 1 - -   
## 3.166100124 1 1 1 -   
## 3.672443234 1 1 1 1   
##   
## P value adjustment method: none

pairwise.wilcox.test(clean\_data$low\_Dose,clean\_data$high\_Dose,paired=FALSE,p.adjust.method="none",exact=TRUE)

##   
## Pairwise comparisons using Wilcoxon rank sum exact test   
##   
## data: clean\_data$low\_Dose and clean\_data$high\_Dose   
##   
## -0.033101504 0.330736663 0.339647028 0.696543817 0.729677528  
## 0.330736663 1 - - - -   
## 0.339647028 1 1 - - -   
## 0.696543817 1 1 1 - -   
## 0.729677528 1 1 1 1 -   
## 0.747139674 1 1 1 1 1   
## 1.115736568 1 1 1 1 1   
## 1.232425388 1 1 1 1 1   
## 1.453706788 1 1 1 1 1   
## 1.682638604 1 1 1 1 1   
## 1.999580929 1 1 1 1 1   
## 2.202377751 1 1 1 1 1   
## 2.924862707 1 1 1 1 1   
## 3.166100124 1 1 1 1 1   
## 3.672443234 1 1 1 1 1   
## 0.747139674 1.115736568 1.232425388 1.453706788 1.682638604  
## 0.330736663 - - - - -   
## 0.339647028 - - - - -   
## 0.696543817 - - - - -   
## 0.729677528 - - - - -   
## 0.747139674 - - - - -   
## 1.115736568 1 - - - -   
## 1.232425388 1 1 - - -   
## 1.453706788 1 1 1 - -   
## 1.682638604 1 1 1 1 -   
## 1.999580929 1 1 1 1 1   
## 2.202377751 1 1 1 1 1   
## 2.924862707 1 1 1 1 1   
## 3.166100124 1 1 1 1 1   
## 3.672443234 1 1 1 1 1   
## 1.999580929 2.202377751 2.924862707 3.166100124  
## 0.330736663 - - - -   
## 0.339647028 - - - -   
## 0.696543817 - - - -   
## 0.729677528 - - - -   
## 0.747139674 - - - -   
## 1.115736568 - - - -   
## 1.232425388 - - - -   
## 1.453706788 - - - -   
## 1.682638604 - - - -   
## 1.999580929 - - - -   
## 2.202377751 1 - - -   
## 2.924862707 1 1 - -   
## 3.166100124 1 1 1 -   
## 3.672443234 1 1 1 1   
##   
## P value adjustment method: none

# Multiple comparions with kruskalmc   
  
kruskalmc(clean\_data$placebo~clean\_data$high\_Dose, data=clean\_data, probs = 0.01)

## Multiple comparison test after Kruskal-Wallis   
## p.value: 0.01   
## Comparisons  
## obs.dif critical.dif difference  
## -0.033101504-0.330736663 4 24.68103 FALSE  
## -0.033101504-0.339647028 1 24.68103 FALSE  
## -0.033101504-0.696543817 3 24.68103 FALSE  
## -0.033101504-0.729677528 1 24.68103 FALSE  
## -0.033101504-0.747139674 6 24.68103 FALSE  
## -0.033101504-1.115736568 8 24.68103 FALSE  
## -0.033101504-1.232425388 5 24.68103 FALSE  
## -0.033101504-1.453706788 9 24.68103 FALSE  
## -0.033101504-1.682638604 5 24.68103 FALSE  
## -0.033101504-1.999580929 2 24.68103 FALSE  
## -0.033101504-2.202377751 3 24.68103 FALSE  
## -0.033101504-2.924862707 2 24.68103 FALSE  
## -0.033101504-3.166100124 7 24.68103 FALSE  
## -0.033101504-3.672443234 4 24.68103 FALSE  
## 0.330736663-0.339647028 3 24.68103 FALSE  
## 0.330736663-0.696543817 1 24.68103 FALSE  
## 0.330736663-0.729677528 5 24.68103 FALSE  
## 0.330736663-0.747139674 10 24.68103 FALSE  
## 0.330736663-1.115736568 12 24.68103 FALSE  
## 0.330736663-1.232425388 1 24.68103 FALSE  
## 0.330736663-1.453706788 13 24.68103 FALSE  
## 0.330736663-1.682638604 9 24.68103 FALSE  
## 0.330736663-1.999580929 2 24.68103 FALSE  
## 0.330736663-2.202377751 7 24.68103 FALSE  
## 0.330736663-2.924862707 6 24.68103 FALSE  
## 0.330736663-3.166100124 11 24.68103 FALSE  
## 0.330736663-3.672443234 8 24.68103 FALSE  
## 0.339647028-0.696543817 2 24.68103 FALSE  
## 0.339647028-0.729677528 2 24.68103 FALSE  
## 0.339647028-0.747139674 7 24.68103 FALSE  
## 0.339647028-1.115736568 9 24.68103 FALSE  
## 0.339647028-1.232425388 4 24.68103 FALSE  
## 0.339647028-1.453706788 10 24.68103 FALSE  
## 0.339647028-1.682638604 6 24.68103 FALSE  
## 0.339647028-1.999580929 1 24.68103 FALSE  
## 0.339647028-2.202377751 4 24.68103 FALSE  
## 0.339647028-2.924862707 3 24.68103 FALSE  
## 0.339647028-3.166100124 8 24.68103 FALSE  
## 0.339647028-3.672443234 5 24.68103 FALSE  
## 0.696543817-0.729677528 4 24.68103 FALSE  
## 0.696543817-0.747139674 9 24.68103 FALSE  
## 0.696543817-1.115736568 11 24.68103 FALSE  
## 0.696543817-1.232425388 2 24.68103 FALSE  
## 0.696543817-1.453706788 12 24.68103 FALSE  
## 0.696543817-1.682638604 8 24.68103 FALSE  
## 0.696543817-1.999580929 1 24.68103 FALSE  
## 0.696543817-2.202377751 6 24.68103 FALSE  
## 0.696543817-2.924862707 5 24.68103 FALSE  
## 0.696543817-3.166100124 10 24.68103 FALSE  
## 0.696543817-3.672443234 7 24.68103 FALSE  
## 0.729677528-0.747139674 5 24.68103 FALSE  
## 0.729677528-1.115736568 7 24.68103 FALSE  
## 0.729677528-1.232425388 6 24.68103 FALSE  
## 0.729677528-1.453706788 8 24.68103 FALSE  
## 0.729677528-1.682638604 4 24.68103 FALSE  
## 0.729677528-1.999580929 3 24.68103 FALSE  
## 0.729677528-2.202377751 2 24.68103 FALSE  
## 0.729677528-2.924862707 1 24.68103 FALSE  
## 0.729677528-3.166100124 6 24.68103 FALSE  
## 0.729677528-3.672443234 3 24.68103 FALSE  
## 0.747139674-1.115736568 2 24.68103 FALSE  
## 0.747139674-1.232425388 11 24.68103 FALSE  
## 0.747139674-1.453706788 3 24.68103 FALSE  
## 0.747139674-1.682638604 1 24.68103 FALSE  
## 0.747139674-1.999580929 8 24.68103 FALSE  
## 0.747139674-2.202377751 3 24.68103 FALSE  
## 0.747139674-2.924862707 4 24.68103 FALSE  
## 0.747139674-3.166100124 1 24.68103 FALSE  
## 0.747139674-3.672443234 2 24.68103 FALSE  
## 1.115736568-1.232425388 13 24.68103 FALSE  
## 1.115736568-1.453706788 1 24.68103 FALSE  
## 1.115736568-1.682638604 3 24.68103 FALSE  
## 1.115736568-1.999580929 10 24.68103 FALSE  
## 1.115736568-2.202377751 5 24.68103 FALSE  
## 1.115736568-2.924862707 6 24.68103 FALSE  
## 1.115736568-3.166100124 1 24.68103 FALSE  
## 1.115736568-3.672443234 4 24.68103 FALSE  
## 1.232425388-1.453706788 14 24.68103 FALSE  
## 1.232425388-1.682638604 10 24.68103 FALSE  
## 1.232425388-1.999580929 3 24.68103 FALSE  
## 1.232425388-2.202377751 8 24.68103 FALSE  
## 1.232425388-2.924862707 7 24.68103 FALSE  
## 1.232425388-3.166100124 12 24.68103 FALSE  
## 1.232425388-3.672443234 9 24.68103 FALSE  
## 1.453706788-1.682638604 4 24.68103 FALSE  
## 1.453706788-1.999580929 11 24.68103 FALSE  
## 1.453706788-2.202377751 6 24.68103 FALSE  
## 1.453706788-2.924862707 7 24.68103 FALSE  
## 1.453706788-3.166100124 2 24.68103 FALSE  
## 1.453706788-3.672443234 5 24.68103 FALSE  
## 1.682638604-1.999580929 7 24.68103 FALSE  
## 1.682638604-2.202377751 2 24.68103 FALSE  
## 1.682638604-2.924862707 3 24.68103 FALSE  
## 1.682638604-3.166100124 2 24.68103 FALSE  
## 1.682638604-3.672443234 1 24.68103 FALSE  
## 1.999580929-2.202377751 5 24.68103 FALSE  
## 1.999580929-2.924862707 4 24.68103 FALSE  
## 1.999580929-3.166100124 9 24.68103 FALSE  
## 1.999580929-3.672443234 6 24.68103 FALSE  
## 2.202377751-2.924862707 1 24.68103 FALSE  
## 2.202377751-3.166100124 4 24.68103 FALSE  
## 2.202377751-3.672443234 1 24.68103 FALSE  
## 2.924862707-3.166100124 5 24.68103 FALSE  
## 2.924862707-3.672443234 2 24.68103 FALSE  
## 3.166100124-3.672443234 3 24.68103 FALSE

kruskalmc(clean\_data$placebo~clean\_data$high\_Dose, cont="two-tailed") # restricts number of comparisons

## Multiple comparison test after Kruskal-Wallis, treatments vs control (two-tailed)   
## p.value: 0.05   
## Comparisons  
## obs.dif critical.dif difference  
## -0.033101504-0.330736663 4 18.42802 FALSE  
## -0.033101504-0.339647028 1 18.42802 FALSE  
## -0.033101504-0.696543817 3 18.42802 FALSE  
## -0.033101504-0.729677528 1 18.42802 FALSE  
## -0.033101504-0.747139674 6 18.42802 FALSE  
## -0.033101504-1.115736568 8 18.42802 FALSE  
## -0.033101504-1.232425388 5 18.42802 FALSE  
## -0.033101504-1.453706788 9 18.42802 FALSE  
## -0.033101504-1.682638604 5 18.42802 FALSE  
## -0.033101504-1.999580929 2 18.42802 FALSE  
## -0.033101504-2.202377751 3 18.42802 FALSE  
## -0.033101504-2.924862707 2 18.42802 FALSE  
## -0.033101504-3.166100124 7 18.42802 FALSE  
## -0.033101504-3.672443234 4 18.42802 FALSE

# linear regression  
model = lm(stacked\_data$values~stacked\_data$ind)  
summary(model)

##   
## Call:  
## lm(formula = stacked\_data$values ~ stacked\_data$ind)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.5171 -0.7369 -0.1298 0.7183 2.1884   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.17686 0.24321 0.727 0.47114   
## stacked\_data$indlow\_Dose -0.02054 0.34394 -0.060 0.95266   
## stacked\_data$indhigh\_Dose 1.30718 0.34394 3.801 0.00046 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9419 on 42 degrees of freedom  
## Multiple R-squared: 0.3178, Adjusted R-squared: 0.2853   
## F-statistic: 9.783 on 2 and 42 DF, p-value: 0.0003251

# plotting #  
  
barplot(stacked\_data$values,col=c("gray", "red","blue"))  
legend("topleft", legend=c("placebo","high\_Dose", "low\_Dose"),fill = c("gray", "red","blue"))

Conclusion: Planned comparisons (no protection for alpha inflation) independent t test: high\_dose p-value=0.00046 <0.05 and hence there is difference between placemo and high\_dose. P value adjustment method: bonferroni : p-value=0.0014<0.05 P value adjustment method: holm : p-value=0.0012<0.05 show more precise significance Anova: p-value=0.000325 \*\*\* p<0.05 post-hocs:TukeyHSD: high\_Dose-placebo1.30717513 high\_Dose-low\_Dose 1.32771811 There is significance difference by all tests applicable and hence it is clear by plot of regression model to that there is certainly difference between placebo and drugs used as well as high dose have more effect than low dose.