

1-way ANOVA ($p=1$, $g=6$)

```

#####
##### One-way ANOVA (p=1, g>2)
#####

load("mcshapiro.test.RData")
data = chickwts

# -----
# Exploration
# (label seconda colonna)
# -----
dim(data)
summary(data)
colnames(data) = c('measure', 'label')
head(data)

g = length(levels(data$label))

attach(data)
plot(label, measure, xlab='treat', ylab='measures')

# -----
# Model
# -----
# measure.ij = mu + tau.i + eps.ij           eps.ij ~ N(0, sigma^2)
#
# H0: tau.1 = tau.2 = ... = tau.g = 0
# H1: there is a tau.j != 0

# Boxplots
par(mfrow=c(1,2))
barplot(rep(mean(measure),g), names.arg=levels(label), ylim=c(0,max(measure)),
       las=2, col='grey85', main='Model under H0')
barplot(tapply(measure, label, mean), names.arg=levels(label), ylim=c(0,max(measure)),
       las=2, col=rainbow(6),main='Model under H1')

# Check the assumptions
n = length(label) # total number of obs.
ng = table(label) # number of obs. in each group
treat = levels(label) # levels of the treatment
g = length(treat) # number of levels (i.e., of groups)

# 1. Gaussianity of the groups
Ps = 0*(1:g)
for(i in 1:g){
  Ps[i] = shapiro.test(measure[ label==treat[i]])$p
}
Ps

# 2. Same covariance structure (= same sigma^2)
Var = 0*(1:g)
for(i in 1:g){
  Var[i] = var(measure[label==treat[i]])
}
Var

# test of homogeneity of variances
# H0: sigma.1 = sigma.2 = ... = sigma.g
# H1: there exist i,j s.t. sigma.i!=sigma.j
bartlett.test(measure, label)

# -----
# One-way ANOVA
# -----
```

```

fit = aov(measure ~ label)
summary(fit)
```

```

#### How to read the summary:
#          Df   Sum Sq   Mean Sq    F value    Pr(>F)
# treat     (g-1) SStreat  SStreat/(g-1)  Fstatistic p-value [H0: tau.i=0 for every i]
# Residuals (n-g) SSRes   SSRes/(n-g)
#### --
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
####
```

```

# -----
# Which supplement is responsible for this?
# -----
# To see this, we need to do g*(g-1)/2 comparisons.
```

```

## Method 1
## Bonferroni
k = g*(g-1)/2
alpha = 0.05
```

Bonferroni

Bout.

Bonferroni

1-at-time + Bonferroni

```

Mediag = tapply(measure, label, mean)
SSres = sum(residuals(fit)^2)
S = SSres/(n-g)

# CI for all the differences
ICrange = NULL
for(i in 1:(g-1)){
  for(j in (i+1):g){
    print(paste(treat[i],"-",treat[j]))
    print(as.numeric(c(Mediag[i]-Mediag[j] - qt(1-alpha/(2*k), n-g) *
                      sqrt(S * (1/ng[i] + 1/ng[j]))),
                    Mediag[i]-Mediag[j] + qt(1-alpha/(2*k), n-g) *
                      sqrt(S * (1/ng[i] + 1/ng[j])))))
    ICrange=rbind(ICrange,as.numeric(c(Mediag[i]-Mediag[j] - qt(1-alpha/(2*k), n-g) *
                                      sqrt(S * (1/ng[i] + 1/ng[j]))),
                                      Mediag[i]-Mediag[j] + qt(1-alpha/(2*k), n-g) *
                                      sqrt(S * (1/ng[i] + 1/ng[j])))))
  }
}

# Comment
# Se per un intervallo di confidenza non c'è lo 0 allora c'è evidenza per dire
# che il trattamento ha avuto effetto nei due gruppi

# Plot
par(mfrow=c(1,2))
plot(label, measure, xlab='treat', ylab='measure', col = rainbow(6), las=2)
h = 1
plot(c(1,g*(g-1)/2),range(ICrange), pch='', xlab='pairs treat',
      ylab='Conf. Int. tau weight', main='Univariate Conf. Int. - Bonf. corrected')
for(i in 1:(g-1)) {
  for(j in (i+1):g) {
    ind = (i-1)*g-i*(i-1)/2+(j-i)
    lines(c(h,h), c(ICrange[ind,1],ICrange[ind,2]), col='grey55');
    points(h, Mediag[i]-Mediag[j], pch=16, col='grey55');
    points(h, ICrange[ind,1], col=rainbow(g)[j], pch=16);
    points(h, ICrange[ind,2], col=rainbow(g)[i], pch=16);
    h <- h+1
  }
}
abline(h=0)

# Comment
# Attenzione ad applicare la transitività
dev.off()

## Method 2
## k one-at-the-time CI (without Bonferroni correction) -> each CI has alpha, not together
## (change criterium to control the univariate rejection (multiple testing))

Auni = matrix(0,g,g)
for(i in 1:g) {
  for(j in i:g) {
    Auni[i,j] = Mediag[i]-Mediag[j]+qt(1-alpha/2,n-g)*sqrt(S*(1/ng[i]+1/ng[j])))
    for(j in 1:i) {
      Auni[i,j] = Mediag[j]-Mediag[i]-qt(1-alpha/2,n-g)*sqrt(S*(1/ng[i]+1/ng[j])))
    }
    Auni[i,i] = 0
  }
}
par(mfrow=c(1,2))
h = 1
plot(c(1,g*(g-1)/2),range(Auni), pch='', xlab='pairs treat',
      ylab='CI delta measure', main='Univariate Conf. Int. - 1-at-the-time', col='grey55')
for(i in 1:(g-1)) {
  for(j in (i+1):g) {lines (c(h,h), c(Auni[i,j],Auni[j,i]));
    points(h, as.numeric(Mediag[i]-Mediag[j]), pch=16, col='grey55');
    points(h, Auni[i,j], col=rainbow(g)[i], pch=16);
    points(h, Auni[j,i], col=rainbow(g)[j], pch=16);
    h <- h+1
  }
}
abline(h=0)

# (Per il confronto aggiungiamo)
h = 1
plot(c(1,g*(g-1)/2),range(ICrange), pch='', xlab='pairs treat',
      ylab='Conf. Int. tau weight', main='Univariate Conf. Int. - Bonf. corrected')
for(i in 1:(g-1)) {
  for(j in (i+1):g) {
    ind = (i-1)*g-i*(i-1)/2+(j-i)
    lines(c(h,h), c(ICrange[ind,1],ICrange[ind,2]), col='grey55');
    points(h, Mediag[i]-Mediag[j], pch=16, col='grey55');
    points(h, ICrange[ind,1], col=rainbow(g)[j], pch=16);
    points(h, ICrange[ind,2], col=rainbow(g)[i], pch=16);
    h <- h+1
  }
}
abline(h=0)

dev.off()

# We compute the p-values of the univariate tests

```

1-at-time
+ Bonferroni

1-at-time + correction

```
# Matrix of tests for the difference between all the pairs
P <- matrix(0,g,g)
for(i in 1:g) {
  for(j in i:g) {
    P[i,j] = (1-pt(abs((Mediag[i]-Mediag[j])/sqrt(S*(1/ng[i]+1/ng[j]))), n-g))*2}
  for(j in 1:i) {
    P[i,j] = (1-pt(abs((Mediag[i]-Mediag[j])/sqrt(S*(1/ng[i]+1/ng[j]))), n-g))*2}
  P[i,i] = 0
}
P

# Vector of p-values
p_values <- c(P[1, 2:6], P[2, 3:6], P[3, 4:6], P[4, 5:6], P[5, 6])

## Plottiamo tutti i pvalues (one-at-time, Bonferroni, FDR)
plot(1:(g*(g-1)/2), p_values, ylim=c(0,1), type='b', pch=16, col='grey55',
      xlab='pairs treat', main='P-values')
abline(h=alpha, lty=2)

# Bonferroni correction
p.bonf <- p.adjust(p, 'bonf')
lines(1:(g*(g-1)/2), p.bonf, col='blue', pch=16, type='b')

# Correction according to the false discovery rate (Benjamini-Hockberg)
p.fdr <- p.adjust(p, 'fdr')
lines(1:(g*(g-1)/2), p.fdr, col='red', pch=16, type='b')

legend('topleft', c('Not corr.', 'Bonf.', 'BH'), col=c('grey55', 'blue', 'red'), pch=16)

## Quali hanno fatto effetto
which(p.bonf < alpha)
which(p.fdr < alpha)

detach(data)
```

1-way MANOVA ($p=4$, $g=3$)

```

#####
#####
### One-way MANOVA (p>1, g>2)
### (da modificare ovunque, nell'esempio g=3)
#####
#####
load("mcshapiro.test.RData")
data = iris

# -----
# Exploration
# (label ultima colonna)
# -----
head(data)
p = dim(data)[2] - 1
v = 'X1'
# Rinominiamo
for(i in 2:p){
  v = c(v, paste('X', i, sep=''))
}
colnames(data) = c(v, 'label')
head(data)

n     = length(data$label)      # total number of obs.
ng    = table(data$label)       # number of obs. in each group
treat = levels(data$label)     # levels of the treatment
g     = length(treat)         # number of levels (i.e., of groups)

data.feats = data[,1:p]

# Exploration
colore = rep(rainbow(p), each=50)
pairs(data.feats, col=colore, pch=16)

# Creiamo g indici
i1 = which(data$label == treat[1])
i2 = which(data$label == treat[2])
i3 = which(data$label == treat[3])

## Plot: different panels -> different group
par(mfrow=c(1,3))
boxplot(data.feats[i1,], main='gruppo 1', ylim=c(0,8), col = rainbow(p))
boxplot(data.feats[i2,], main='gruppo 2', ylim=c(0,8), col = rainbow(p))
boxplot(data.feats[i3,], main='gruppo 3', ylim=c(0,8), col = rainbow(p))

## Plot: different panels -> different feature
par(mfrow=c(1,4))
boxplot(data.feats[,1]~data$label, main='X1', ylim=c(0,8), col = rainbow(3))
boxplot(data.feats[,2]~data$label, main='X2', ylim=c(0,8), col = rainbow(3))
boxplot(data.feats[,3]~data$label, main='X3', ylim=c(0,8), col = rainbow(3))
boxplot(data.feats[,4]~data$label, main='X4', ylim=c(0,8), col = rainbow(3))

graphics.off()

# -----
# Model
# -----
# measure.ij = mu + tau.i + eps.ij           eps.ij ~ N(0, sigma^2) (in R^p)
#
# H0: tau.1 = tau.2 = ... = tau.g = 0
# H1: there is a tau.j != 0

n1 = length(i1)
n2 = length(i2)
n3 = length(i3)
n  = n1+n2+n3

# Check the assumptions
# 1. normality
Ps = NULL
for(i in 1:g){
  Ps = c(Ps, mcshapiro.test(data[get(paste('i',i, sep='')),1:p])$p)
}
Ps

# 2. same covariance structure (homoschedasticity)
S  = cov(data.feats)
S1 = cov(data.feats[i1,])
S2 = cov(data.feats[i2,])
S3 = cov(data.feats[i3,])

# Qualitatively:
round(S1,digits=1)
round(S2,digits=1)
round(S3,digits=1)

```

```

par(mfrow=c(1,3))
image(S1, col=heat.colors(100),main='Cov. S1', asp=1, axes = FALSE,
      breaks = quantile(rbind(S1,S2,S3), (0:100)/100, na.rm=TRUE))
image(S2, col=heat.colors(100),main='Cov. S2', asp=1, axes = FALSE,
      breaks = quantile(rbind(S1,S2,S3), (0:100)/100, na.rm=TRUE))
image(S3, col=heat.colors(100),main='Cov. S3', asp=1, axes = FALSE,
      breaks = quantile(rbind(S1,S2,S3), (0:100)/100, na.rm=TRUE))

# Comment
# Se sono diverse segnalarlo, però proviamo lo stesso a procedere

# -----
# One-way MANOVA
# -----

fit = manova(as.matrix(data.feats) ~ data$label)
summary.manova(fit, test="Wilks")

# Pr(>F) = p-value di H0 vs. H1
# se molto basso -> H1 -> il trattamento ha fatto effetto

# Comment
# If p<=2 and g<=3 we have an exact test (with Wilks)

# -----
# Which supplement is responsible for this?
# (nel caso accettiamo H1)
# -----

## Prima di tutto
## Vediamo su quali variabili il gruppo fa effetto:
## Via ANOVA: per ciascuna feature facciamo un'anova per vedere se l'appartenenza
## al gruppo ha effetto sulla media delle variabili

summary.aov(fit)

# Comment
# Pr(>F) = p-value basso -> il gruppo influisce su quella Xk
# Questa analisi non dice né quali GRUPPI differiscono né le VARIABILI per le quali
# i gruppi differiscono

## Bonferroni CI
## Vogliamo sapere il livello (di labels) che induce la differenza.
## Dobbiamo fare g*(g-1)/2 intervalli

alpha = 0.05
k     = p*g*(g-1)/2
qT    = qt(1-alpha/(2*k), n-g)

W   = summary.manova(fit)$SS$Residuals
m   = sapply(data.feats,mean)          # estimates mu
m1  = sapply(data.feats[i1,],mean)    # estimates mu.1=mu+tau.1
m2  = sapply(data.feats[i2,],mean)    # estimates mu.2=mu+tau.2
m3  = sapply(data.feats[i3,],mean)    # estimates mu.3=mu+tau.3

inf12 = m1-m2 - qT * sqrt( diag(W)/(n-g) * (1/n1+1/n2) )
sup12 = m1-m2 + qT * sqrt( diag(W)/(n-g) * (1/n1+1/n2) )
inf13 = m1-m3 - qT * sqrt( diag(W)/(n-g) * (1/n1+1/n3) )
sup13 = m1-m3 + qT * sqrt( diag(W)/(n-g) * (1/n1+1/n3) )
inf23 = m2-m3 - qT * sqrt( diag(W)/(n-g) * (1/n2+1/n3) )
sup23 = m2-m3 + qT * sqrt( diag(W)/(n-g) * (1/n2+1/n3) )

CI <- list(g1_g2 = cbind(inf12, sup12),
           g1_g3 = cbind(inf13, sup13),
           g2_g3 = cbind(inf23, sup23))
CI

# Now we have a complete frame (intervals for all the components of tau_i)

# Comment
# Da questi intervalli vediamo di chi è colpa del cambiamento:
# se lo zero non è presente in un paragone "g.i_g_j" in un intervallo "X.k"
# allora la variabile X.k è influenzata dai gruppi g.i e g.j

# Plot: different panels -> different features
par(mfrow=c(2,4))
boxplot(data.feats[,1]~data$label, main=paste(v[1]), ylim=c(0,8), col = rainbow(g))
boxplot(data.feats[,2]~data$label, main=paste(v[2]), ylim=c(0,8), col = rainbow(g))
boxplot(data.feats[,3]~data$label, main=paste(v[3]), ylim=c(0,8), col = rainbow(g))
boxplot(data.feats[,4]~data$label, main=paste(v[4]), ylim=c(0,8), col = rainbow(g))

mg      = rbind(m1,m2,m3)
sp.name = v
for(k in 1:p){
  plot(c(1,g*(g-1)/2),ylim=c(-4,4), xlim=c(1,3), pch='',
       xlab='pairs treat', ylab=paste('CI tau',k), main=paste('CI tau',sp.name[k]))
  lines (c(1,1), c(CI[[1]][k,1],CI[[1]][k,2]))
}

```

```
points(1, mg[1,k]-mg[2,k], pch=16);
points(1, CI[[1]][k,1], col=rainbow(g)[2], pch=16);
points(1, CI[[1]][k,2], col=rainbow(g)[1], pch=16);
lines (c(2,2), c(CI[[2]][k,1],CI[[2]][k,2]));
points(2, mg[1,k]-mg[3,k], pch=16);
points(2, CI[[2]][k,1], col=rainbow(g)[3], pch=16);
points(2, CI[[2]][k,2], col=rainbow(g)[1], pch=16);
lines (c(3,3), c(CI[[3]][k,1],CI[[3]][k,2]));
points(3, mg[2,k]-mg[3,k], pch=16);
points(3, CI[[3]][k,1], col=rainbow(g)[3], pch=16);
points(3, CI[[3]][k,2], col=rainbow(g)[2], pch=16);
abline(h=0)
}

# Comment
# If no one contains 0 it means that every group has relevance in every feature
```

Two-way ANOVA ($p=1, g=2, b=2$)

```

#####
#####
##### Two-way ANOVA (p=1, g=2, b=2)
#####
#####
load("mcshapiro.test.RData")

# -----
# Create the dataset
# -----

measure = c(18.7, 16.8, 20.1, 22.4, 14.0, 15.2, 22.0, 23.3)
label_1 = factor(c('Esso','Esso','Esso','Shell','Shell','Shell','Shell'))
label_2 = factor(c('95','95','98','98','95','95','98','98'))

# -----
# Exploration
# -----


g = length(levels(label_1))
b = length(levels(label_2))
n = length(measure)/(g*b)

M = mean(measure)
M_label_1 = tapply(measure, label_1, mean)
M_label_2 = tapply(measure, label_2, mean)

# !Just for visualization!
label_1_2 = factor(c('Esso95','Esso95','Esso98','Esso98','Shell95',
                     'Shell95','Shell98','Shell98'))
M_label_1_2 = tapply(measure, label_1_2, mean)

# Plot
par(mfrow=c(2,3), las=2)
barplot(rep(M,4), names.arg=levels(label_1_2), ylim=c(0,24), main='No factor')
barplot(rep(M_label_1,each=2), names.arg=levels(label_1_2), ylim=c(0,24),
        col=rep(c('blue','red'),each=2), main='label_1')
barplot(rep(M_label_2,times=2), names.arg=levels(label_1_2), ylim=c(0,24),
        col=rep(c('darkgreen','orange'),times=2), main='label_2')
barplot(c(M_label_1[1]+M_label_2[1]-M, M_label_1[1]+M_label_2[2]-M,
          M_label_1[2]+M_label_2[1]-M, M_label_1[2]+M_label_2[2]-M),
        names.arg=levels(label_1_2), ylim=c(0,24),
        col=rep(c('darkgreen','orange'),times=2), density=rep(10,4), angle=135,
        main='Additive model label_1 + label_2')
barplot(c(M_label_1[1]+M_label_2[1]-M, M_label_1[1]+M_label_2[2]-M,
          M_label_1[2]+M_label_2[1]-M, M_label_1[2]+M_label_2[2]-M),
        names.arg=levels(label_1_2), ylim=c(0,24),
        col=rep(c('blue','red'),each=2), density=rep(10,4), add=T)
barplot(M_label_1_2, names.arg=levels(label_1_2), ylim=c(0,24),
        col=rainbow(5)[2:5], main='Model with Interact. label_1 & label_2.')
plot(label_1_2, measure, col=rainbow(5)[2:5], ylim=c(0,24), xlab='')

dev.off()

# -----
# Model
# (if we have to remove rows => one at the time)
# -----


##### Complete model
#   measure.ijk = mu + tau.i + beta.j + gamma.ij + eps.ijk      eps.ijk ~ N(0, sigma^2)
#                                         i=1,2 (label_1)
#                                         j=1,2 (label_2)
fit.aov2.int = aov(measure ~ label_1 + label_2 + label_1:label_2)
summary.aov(fit.aov2.int)

##### Additive model:
#   measure.ijk = mu + tau.i + beta.j + eps.ijk      eps.ijk ~ N(0, sigma^2)
#                                         i=1,2 (label_1)
#                                         j=1,2 (label_2)
fit.aov2.ad = aov(measure ~ label_1 + label_2)
summary.aov(fit.aov2.ad)

##### Reduced additive model (ANOVA one-way, b=2 (or g=2)):
#   measure.jk = mu + beta.j + eps.jk      eps.ijk ~ N(0, sigma^2)
#                                         j=1,2 (label_2)
fit.aov1 = aov(measure ~ label_2)
summary.aov(fit.aov1)

# -----
# IC for the reduced additive model
# -----


SSres = sum(residuals(fit.aov1)^2)
IC <- c(diff(M_label_2)-qt(0.95, (n*g-1)*b) * sqrt(SSres/((n*g-1)*b) * (1/(n*g)+1/(n*g))),
         diff(M_label_2)+qt(0.95, (n*g-1)*b) * sqrt(SSres/((n*g-1)*b) * (1/(n*g)+1/(n*g))))
names(IC) <- c('Inf', 'Sup')

```

meglio
tde
p.2

```
IC # IC for mu(label_2[1]) - mu(label_2[2])

# Note
# Non dovrebbe esserci lo zero (altrimenti non sarebbe)

# -----
# (Approximate) check of the assumptions
# (controlliamo singolarmente e solo per i labels che consideriamo)
# -----

# 1) Gaussianity
treat = levels(label_2)
Ps = 0*(1:b)
for(i in 1:b){
  Ps[i] = shapiro.test(measure[label_2==treat[i]])$p
}
Ps

# 2) homogeneity of variances
bartlett.test(measure, label_2)
```



Two-way MANOVA ($p=3$, $g=2$, $b=2$)

```

#####
#####
### Two-way MANOVA (p=3, g=2, b=2)
#####
#####

load("mcshapiro.test.RData")
data = read.table('T6-4.dat', col.names=c('Ex', 'Ad', 'Tr', 'Gl', 'Op'))

# -----
# Exploration
# (label prime colonne)
# -----


head(data)
p = dim(data)[2] - 2
v = 'X1'

# Rinominiamo
for(i in 2:p){
  v = c(v, paste('X', i, sep=''))
}
colnames(data) = c('label_1', 'label_2', v)
head(data)

Label_1 = factor(data$label_1, labels=c('L', 'H')) # Treat.1
Label_2 = factor(data$label_2, labels=c('L', 'H')) # Treat.2

Label_1_2 = Label_1
levels(Label_1_2) <- c('LL', 'LH', 'HL', 'HH')
Label_1_2[Label_1=="L" & Label_2=="L"] <- 'LL'
Label_1_2[Label_1=="L" & Label_2=="H"] <- 'LH'
Label_1_2[Label_1=="H" & Label_2=="L"] <- 'HL'
Label_1_2[Label_1=="H" & Label_2=="H"] <- 'HH'

data.feats = data[,3:(2+p)]


## Graphical exploration of the data
# effect of the treatments + their interaction on the FIRST variable
layout(matrix(c(1,1,2,3), 2, byrow=T))
boxplot(data.feats[,1]~Label_1_2, main='With Interac. lab_1+lab_2', ylab='Tr', col='grey95')
boxplot(data.feats[,1]~Label_1, main='Only lab_1', ylab='Tr', col=c('red', 'blue'))
boxplot(data.feats[,1]~Label_2, main='Only lab_2', ylab='Tr', col=c('forestgreen', 'gold'))

# effect of the treatments + their interaction on the SECOND variable
layout(matrix(c(1,1,2,3), 2, byrow=T))
boxplot(data.feats[,2]~Label_1_2, main='With Interac. lab_1+lab_2', ylab='Tr', col='grey95')
boxplot(data.feats[,2]~Label_1, main='Only lab_1', ylab='Tr', col=c('red', 'blue'))
boxplot(data.feats[,2]~Label_2, main='Only lab_2', ylab='Tr', col=c('forestgreen', 'gold'))

# effect of the treatments + their interaction on the THIRD variable
layout(matrix(c(1,1,2,3), 2, byrow=T))
boxplot(data.feats[,3]~Label_1_2, main='With Interac. lab_1+lab_2', ylab='Tr', col='grey95')
boxplot(data.feats[,3]~Label_1, main='Only lab_1', ylab='Tr', col=c('red', 'blue'))
boxplot(data.feats[,3]~Label_2, main='Only lab_2', ylab='Tr', col=c('forestgreen', 'gold'))

dev.off()

# -----
# Model
# (if we have to remove rows => one at the time)
# -----


# Assumptions
# 1) Gaussianity
treat = levels(Label_1_2)
Ps = 0*(1:length(treat))
for(i in 1:length(treat)){
  Ps[i] = mcshapiro.test(data.feats[Label_1_2==levels(Label_1_2)[i],])$p
}
Ps

# 2) homogeneity of the covariance (qualitatively)
S1 = cov(data.feats[Label_1_2==levels(Label_1_2)[1], ])
S2 = cov(data.feats[Label_1_2==levels(Label_1_2)[2], ])
S3 = cov(data.feats[Label_1_2==levels(Label_1_2)[3], ])
S4 = cov(data.feats[Label_1_2==levels(Label_1_2)[4], ])
par(mfrow=c(1,4))
image(S1, col=heat.colors(100), main='Cov. S1', asp=1, axes = FALSE,
      breaks = quantile(rbind(S1,S2,S3,S4), (0:100)/100, na.rm=TRUE))
image(S2, col=heat.colors(100), main='Cov. S2', asp=1, axes = FALSE,
      breaks = quantile(rbind(S1,S2,S3,S4), (0:100)/100, na.rm=TRUE))
image(S3, col=heat.colors(100), main='Cov. S3', asp=1, axes = FALSE,
      breaks = quantile(rbind(S1,S2,S3,S4), (0:100)/100, na.rm=TRUE))
image(S4, col=heat.colors(100), main='Cov. S4', asp=1, axes = FALSE,
      breaks = quantile(rbind(S1,S2,S3,S4), (0:100)/100, na.rm=TRUE))

```

```

#### Complete model
# measure.ijk = mu + tau.i + beta.j + gamma.ij + eps.ijk      eps.ijk ~ N(0, sigma^2)
#                                         i=1,2   (label_1)           i=1,2   (label_1)
#                                         j=1,2   (label_2)           j=1,2   (label_2)
fit2.int = manova(as.matrix(data.feats) ~ Label_1 + Label_2 + Label_1:Label_2)
summary.manova(fit2.int, test="Wilks")

#### Additive model:
# measure.ijk = mu + tau.i + beta.j + eps.ijk      eps.ijk ~ N(0, sigma^2)
#                                         i=1,2   (label_1)           i=1,2   (label_1)
#                                         j=1,2   (label_2)           j=1,2   (label_2)
fit2.ad = manova(as.matrix(data.feats) ~ Label_1 + Label_2)
summary.manova(fit2.ad, test="Wilks")

# -----
# Which supplement is responsible for this?
# (nel caso accettiamo H1)
# -----

## Prima di tutto
## Vediamo su quali variabili il gruppo fa effetto:
## Via ANOVA: per ciascuna feature facciamo un'anova per vedere se l'appartenenza
##           al gruppo ha effetto sulla media delle variabili

summary.aov(fit2.ad)

# Comment
# Pr(>F) = p-value basso -> il gruppo influenza su quella Xk
# Questa analisi non dice né quali GRUPPI differiscono né le VARIABILI per le quali
# i gruppi differiscono

## Bonferroni CI
## Vogliamo sapere il livello (di labels) che induce la differenza.
alpha = 0.05
g = 2
b = 2
n = 5          # quante unità per label_1[i], label_2[j]
                # noi supponiamo design simmetrico
N = n*g*b

W = summary.manova(fit2.ad)$SS$Residuals

# how many comparisons?
k = g*(g-1)/2*p + b*(b-1)/2*p
# because we have: g levels on the first treatment on p components
#           b levels on the second treatment on p components
k

qT = qt(1 - alpha / (2 * k), g*b*n-g-b+1)
# the degrees of freedom of the residuals on the additive model are g*b*n-g-b+1

mExL <- sapply(data.feats[Label_1=='L',],mean)
mExH <- sapply(data.feats[Label_1=='H',],mean)
infEx <- mExH-mExL - qT * sqrt( diag(W)/(g*b*n-g-b+1) * (1/10+1/10) )
supEx <- mExH-mExL + qT * sqrt( diag(W)/(g*b*n-g-b+1) * (1/10+1/10) )

mAdL <- sapply(data.feats[Label_2=='L',],mean)
mAdH <- sapply(data.feats[Label_2=='H',],mean)
infAd <- mAdH-mAdL - qT * sqrt( diag(W)/(g*b*n-g-b+1) * (1/10+1/10) )
supAd <- mAdH-mAdL + qT * sqrt( diag(W)/(g*b*n-g-b+1) * (1/10+1/10) )

IC2 <- list(lab1.H_lab1.L=cbind(infEx, supEx), lab2.H_lab2.L=cbind(infAd, supAd))
IC2

## Plot incazzatissimi
Ex = Label_1
Ad = Label_2
plastic3 = data.feats

par(mfrow=c(3,4))
boxplot(plastic3[,1]~Ex, main='Fact.: Extrusion (Tear Resistance)' ,
         ylab='Tr', col=rainbow(2*6)[c(1,2)], ylim=c(-2,10))
plot(c(1,g*(g-1)/2),range(IC2[[1]][1,]), pch='',main='IC (tau.1-tau.2)[1]' ,
      xlab='pairs treat', ylab='IC (tau.1-tau.2)[1]', ylim=c(-2,10))
lines(c(1,1), c(IC2[[1]][1,1],IC2[[1]][1,2]), col='grey55');
points(1, (mExH-mExL)[1], pch=16, col='grey55');
points(1, IC2[[1]][1,1], col=rainbow(2*6)[1], pch=16);
points(1, IC2[[1]][1,2], col=rainbow(2*6)[2], pch=16);
abline(h=0)

boxplot(plastic3[,1]~Ad, main='Fact.: Additive (Tear Resistance)' ,
         ylab='Tr', col=rainbow(2*6)[c(7,8)], ylim=c(-2,10))
plot(c(1,g*(g-1)/2),range(IC2[[2]][1,]), pch='',main='IC (beta.1-beta.2)[1]' ,
      xlab='pairs treat', ylab='IC (beta.1-beta.2)[1]', ylim=c(-2,10))
lines(c(1,1), c(IC2[[2]][1,1],IC2[[2]][1,2]), col='grey55');

```

```

points(1, (mAdH-mAdL)[1], pch=16, col='grey55');
points(1, IC2[[2]][1,1], col=rainbow(2*6)[7], pch=16);
points(1, IC2[[2]][1,2], col=rainbow(2*6)[8], pch=16);
abline(h=0)

boxplot(plastic3[,2]~Ex,   main='Fact.: Extrusion (Gloss)' , ylab='G1',
         col=rainbow(2*6)[c(3,4)], ylim=c(-2,10))
plot(c(1,g*(g-1)/2),range(IC2[[1]][2,]), pch='',main='IC (tau.1-tau.2)[2]',
      xlab='pairs treat', ylab='IC (tau.1-tau.2)[2]', ylim=c(-2,10))
lines(c(1,1), c(IC2[[1]][2,1],IC2[[1]][2,2]), col='grey55');
points(1, (mExH-mExL)[2], pch=16, col='grey55');
points(1, IC2[[1]][2,1], col=rainbow(2*6)[3], pch=16);
points(1, IC2[[1]][2,2], col=rainbow(2*6)[4], pch=16);
abline(h=0)

boxplot(plastic3[,2]~Ex,   main='Fact.: Additive (Gloss)' , ylab='G1',
         col=rainbow(2*6)[c(9,10)], ylim=c(-2,10))
plot(c(1,g*(g-1)/2),range(IC2[[2]][2,]), pch='',main='IC (beta.1-beta.2)[2]',
      xlab='pairs treat', ylab='IC (beta.1-beta.2)[2]', ylim=c(-2,10))
lines(c(1,1), c(IC2[[2]][2,1],IC2[[2]][2,2]), col='grey55');
points(1, (mAdH-mAdL)[2], pch=16, col='grey55');
points(1, IC2[[2]][2,1], col=rainbow(2*6)[9], pch=16);
points(1, IC2[[2]][2,2], col=rainbow(2*6)[10], pch=16);
abline(h=0)

boxplot(plastic3[,3]~Ex,   main='Fact.: Extrusion (Opacity)' , ylab='Op',
         col=rainbow(2*6)[c(5,6)], ylim=c(-2,10))
plot(c(1,g*(g-1)/2),range(IC2[[1]][3,]), pch='',main='IC (tau.1-tau.2)[3]',
      xlab='pairs treat', ylab='IC (tau.1-tau.2)[3]', ylim=c(-2,10))
lines(c(1,1), c(IC2[[1]][3,1],IC2[[1]][3,2]), col='grey55');
points(1, (mExH-mExL)[3], pch=16, col='grey55');
points(1, IC2[[1]][3,1], col=rainbow(2*6)[5], pch=16);
points(1, IC2[[1]][3,2], col=rainbow(2*6)[6], pch=16);
abline(h=0)

boxplot(plastic3[,3]~Ex,   main='Only Factor Additive (Opacity)' , ylab='Op',
         col=rainbow(2*6)[c(11,12)], ylim=c(-2,10))
plot(c(1,g*(g-1)/2),range(IC2[[2]][3,]), pch='',main='IC (beta.1-beta.2)[3]',
      xlab='pairs treat', ylab='IC beta.1[3]', ylim=c(-2,10))
lines(c(1,1), c(IC2[[2]][3,1],IC2[[2]][3,2]), col='grey55');
points(1, (mAdH-mAdL)[3], pch=16, col='grey55');
points(1, IC2[[2]][3,1], col=rainbow(2*6)[11], pch=16);
points(1, IC2[[2]][3,2], col=rainbow(2*6)[12], pch=16);
abline(h=0)

```