LAB 06

TOPICS:

- Test for the mean of paired multivariate Gaussian observations
- Test for repeated measures
- Test for two independent Gaussian populations
- Exercises

```
library(car)
load("mcshapiro.test.RData")
```

1.

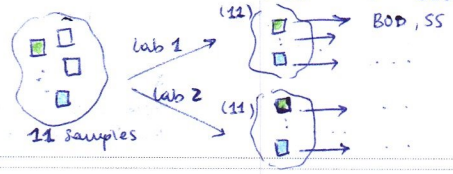
TEST FOR THE MEAN OF PAIRED MULTIVARIATE GAUSSIAN DISTRIBUTION

```
### ------------------------------------------------------------------------
###   Example 6.1 JW + exercise 6.1 JW
###   (n=11, p=2)
### ------------------------------------------------------------------------
# Municipal wastewater treatment plants are required by law to monitor
# their discharges into rivers and streams on a regular basis. Concern
# about the reliability of data from one of these self-monitoring
# programs led to a study in which samples of effluent were divided
# and sent to two laboratories for testing. One-half of each sample
# was sent to the Wisconsin State Laboratory of Hygiene, and one-half
# was sent to a private commercial laboratory routinely used in the
# monitoring program. Measurements of biochemical oxygen demand (BOD)
# and suspended solids (SS) were obtained, for n=11 sample splits,
# from the two laboratories.
# (The experimenter divided each sample by first shaking it and then
# pouring it rapidly into two bottles in order to avoid difference
# in the suspended solids contained in the two half-samples)
### ------------------------------------------------------------------------
# Do the two laboratories' chemical analyses agree?

effluent <- read.table('effluent.dat')
effluent
```

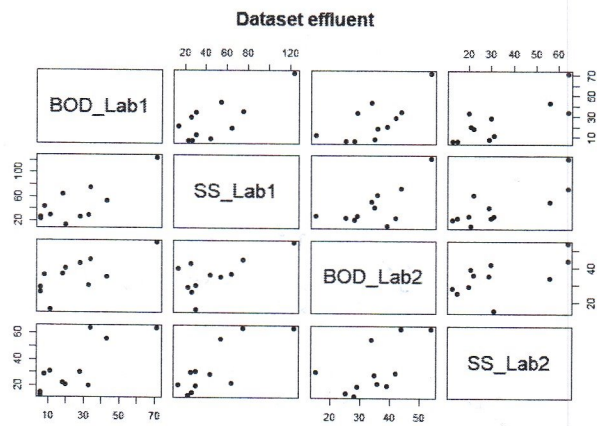
this is important: if it would have been "22 samples splitted in 2 (11, 11) and every set of 11 sample sent to one lab" then they wouldn't be PAIRED DATA (we have to understand how the experiment is made)

```
##    V1  V2 V3 V4
## 1   6  27 25 15
## 2   6  23 28 13
## 3  18  64 36 22
## 4   8  44 35 29
## 5  11  30 15 31
## 6  34  75 44 64
## 7  28  26 42 30
## 8  71 124 54 64
## 9  43  54 34 56
## 10 33  30 29 20
## 11 20  14 39 21
```

each of 2 laboratories have 11 samples from which (each one sample) it is possible to know the concentration of BOD and SS




```
colnames(effluent) <- c('BOD_Lab1','SS_Lab1','BOD_Lab2','SS_Lab2')

x11()
pairs(effluent,pch=19, main='Dataset effluent')
```



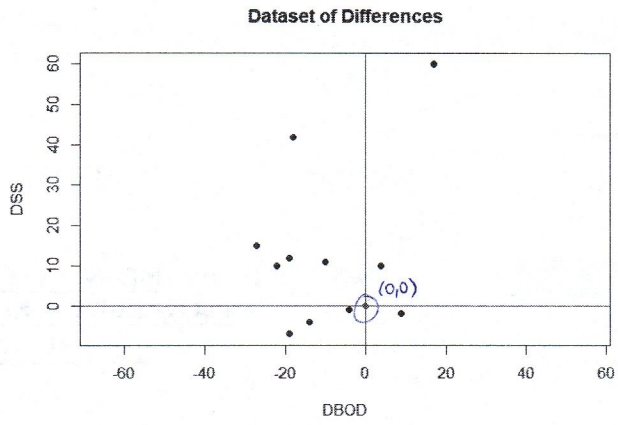

```
dev.off()
```

```
# we compute the sample of differences
D <- data.frame(DBOD=effluent[,1]-effluent[,3], DSS=effluent[,2]-effluent[,4])
D
```

← this is the dataset we want to analyze

```
##    DBOD DSS
## 1   -19  12
## 2   -22  10
## 3   -18  42
## 4   -27  15
## 5    -4  -1
## 6   -10  11
## 7   -14  -4
## 8    17  60
## 9     9  -2
## 10    4  10
## 11  -19  -7
```

```
x11()
plot(D, asp=1, pch=19, main='Dataset of Differences')
abline(h=0, v=0, col='grey35')
points(0,0, pch=19, col='grey35')
```



We want to understand if there is enough statistical evidence to say if the mean of the dataset is different from 0

```
# DBOD: difference in 'biochemical oxygen demand'
# DSS:  difference in 'suspended solids'
#       measured by the two laboratories

dev.off()
```

```
## png
##   2
```

```
# Now we can proceed as we already know, but working on D

### T2 Hotelling Test
# H0: delta == delta.0 vs H1: delta != delta.0
# with delta.0=c(0,0)

# Test the Gaussian assumption (on D!)
mcshapiro.test(D)
```

$$\begin{cases} H_0: \underline{\delta} = \underline{\delta}_0 = [0,0] \\ H_1: \underline{\delta} \neq \underline{\delta}_0 \end{cases}$$

Note that we're not interested in the gaussianity of the original data

```
## $Wmin
## [1] 0.8206964
##
## $pvalue
## [1] 0.0728
##
## $devst
## [1] 0.005196159
##
## $sim
## [1] 2500
```

```
# The p-value isn't very high (but I don't reject for levels 5%, 1%).
# There might be outliers, but we don't remove them because we only have
# very few data

n <- dim(D)[1]  # 11
p <- dim(D)[2]  # 2

D.mean   <- sapply(D,mean)
D.cov    <- cov(D)
D.invcov <- solve(D.cov)

alpha   <- .05
delta.0 <- c(0,0)

D.T2 <- n * (D.mean-delta.0) %*% D.invcov %*% (D.mean-delta.0)
D.T2
```

T^2 statistics

```
##          [,1]
## [1,] 13.63931
```

quantile of the Fisher distribution which defines the rejection region

```
cfr.fisher <- ((n-1)*p/(n-p))*qf(1-alpha,p,n-p)
cfr.fisher
```

the test statistics is bigger than the value that defines the rejection region of 0.95

```
## [1] 9.458877
```

```
D.T2 < cfr.fisher # FALSE: we reject H0 at level 5%
```

← reject H_0

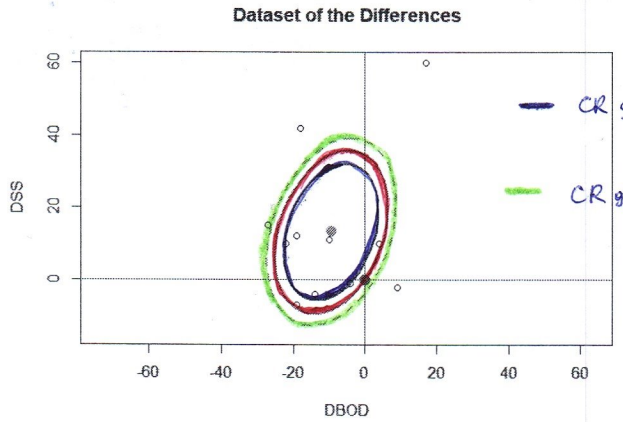
```
##       [,1]
## [1,] FALSE
```

```
# we compute the p-value
P <- 1-pf(D.T2*(n-p)/(p*(n-1)), p, n-p)
P
```

```
##            [,1]
## [1,] 0.02082779
```

← the p-value says that: we have rejected H_0 at 5% (because 5% is larger than the p-value) but if we decide to be more restrictive, for example we chose a level of significance of 1%, we would have had enough statistical evidence to accept H_0.

```
# reject H0 at 5% (don't reject at 1%)

# Ellipsoidal confidence region with confidence level 95%
x11()
plot(D, asp=1, pch=1, main='Dataset of the Differences',ylim=c(-15,60))
ellipse(center=D.mean, shape=D.cov/n, radius=sqrt(cfr.fisher), lwd=2)

points(delta.0[1], delta.0[2], pch=16, col='grey35', cex=1.5)
abline(h=delta.0[1], v=delta.0[2], col='grey35')

# Ellipsoidal confidence region with confidence level 99%
ellipse(center=D.mean, shape=D.cov/n, radius=sqrt((n-1)*p/(n-p)*qf(1-0.01,p,n-p)),lty=2,col='grey',lwd=2)


# What if we set the radius as the quantile of order 1-pval?
ellipse(center=D.mean, shape=D.cov/n, radius=sqrt((n-1)*p/(n-p)*qf(1-as.numeric(P),p,n-p)),lty=1,col='dark grey',lwd=2)
```



CR 95% → (0,0) is out the region (as expected, since we reject H0 at level 5%)

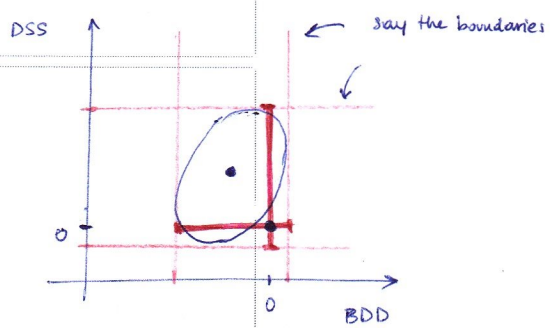
CR 99% → (0,0) is inside the region (again, as expected)

The meaning of the p-value is represented here by the graphical representation of the confidence region taken with confidence level equal to 1 - p.value (— $CR_{1-p.value\,\%}$)

```
dev.off()
```

```
# Now, let's communicate our results to the client.
# Let's build confidence intervals for linear combination of the
# components of the mean vector

### Simultanouse T2 intervals
IC.T2.DBOD <- c( D.mean[1]-sqrt(cfr.fisher*D.cov[1,1]/n) , D.mean[1], D.mean[1]+sqrt(cfr.fisher*D.cov[1,1]/n) )
IC.T2.DSS  <- c( D.mean[2]-sqrt(cfr.fisher*D.cov[2,2]/n) , D.mean[2], D.mean[2]+sqrt(cfr.fisher*D.cov[2,2]/n) )

T2 <- rbind(IC.T2.DBOD, IC.T2.DSS)
dimnames(T2)[[2]] <- c('inf','center','sup')
T2
```

```
##                  inf    center      sup
## IC.T2.DBOD -22.453272 -9.363636  3.72600
## IC.T2.DSS   -5.700119 13.272727 32.24557
```



say the boundaries

```
x11()
plot(D, asp=1, pch=1, main='Dataset of the Differences',ylim=c(-15,60))
ellipse(center=D.mean, shape=D.cov/n, radius=sqrt(cfr.fisher), lwd=2, col='grey')
abline(v = T2[1,1], col='red', lwd=1, lty=2)
abline(v = T2[1,3], col='red', lwd=1, lty=2)
abline(h = T2[2,1], col='red', lwd=1, lty=2)
abline(h = T2[2,3], col='red', lwd=1, lty=2)

points(delta.0[1], delta.0[2], pch=16, col='grey35', cex=1.5)
abline(h=delta.0[1], v=delta.0[2], col='grey35')

segments(IC.T2.DBOD[1],0,IC.T2.DBOD[3],0,lty=1,lwd=2,col='red')
segments(0,IC.T2.DSS[1],0,IC.T2.DSS[3],lty=1,lwd=2,col='red')
```

what happen changing the directions

```
# We don't have enough evidence to reject H0 in any coordinate axes-direction
# But we can reject the global test of (global) level 5%
# => there exists at least one direction along which we are allowed
#    to reject the univariate test

# Recall:
# We reject the global H0 if in at least one direction we observe a 'high'
# value of the statistics T2 (univariate), i.e., we reject the global
# H0 if we reject the univariate test at least in direction (max(T2))

# Hence:
# Worst direction: direction along which the T2 statistics (univariate)
# is maximized

# => From the theory
#    - the maximum is realized (Hotelling T2-statistics)
D.T2
```

we project the ellips along couples of directions. The "worst direction" is the one s.t. the statistics T^2 is max.

```
##          [,1]
## [1,] 13.63931
```

```
#    - the distribution of the maximum is known
#    - the direction along which the maximum is realized is known
worst <- D.invcov %*% (D.mean-delta.0)
worst <- worst/sqrt(sum(worst^2))
worst
```

```
##             [,1]
## DBOD -0.8262423
## DSS   0.5633149
```

```
# Angle with the x-axis:
theta.worst <- atan(worst[2]/worst[1])+pi
theta.worst
```

```
## [1] 2.5432
```

```
# Confidence interval along the worst direction:
IC.worst  <- c( D.mean %*% worst - sqrt(cfr.fisher*(t(worst)%*%D.cov%*%worst)/n),
                D.mean %*% worst,
                D.mean %*% worst + sqrt(cfr.fisher*(t(worst)%*%D.cov%*%worst)/n) )
IC.worst
```

```
## [1]  2.544174 15.213357 27.882540
```

```
delta.0%*%worst
```

← projecting 0 on the worst direction

```
##      [,1]
## [1,]    0
```

```
(IC.worst[1] < delta.0%*%worst) & (delta.0%*%worst < IC.worst[2])
```

```
##       [,1]
## [1,] FALSE
```

Knowing the worst direction we want to project 0 on this direction and see if it's inside the CI along the worst direct. or not

(if $0 \notin CI_{worst} \Rightarrow$ reject $H_0: a^T \mu = 0$)

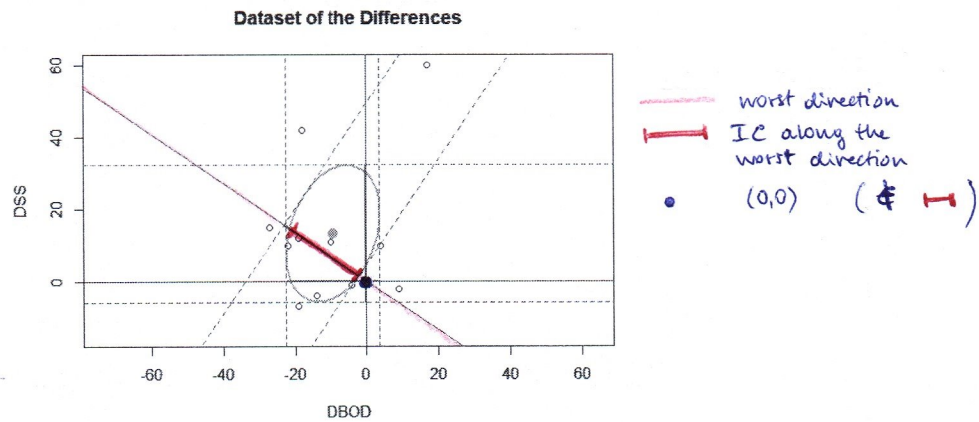
⇓

reject H_0

$H_0: \underline{\delta} = (0,0)$

it's enough one direction

```
# Reject H0: a'mu == a'delta.0 in direction a=worst

# Extremes of IC.worst in the coordinate system (x,y):
x.min <- IC.worst[1]*worst
x.max <- IC.worst[3]*worst
m1.ort <- -worst[1]/worst[2]
q.min.ort <- x.min[2] - m1.ort*x.min[1]
q.max.ort <- x.max[2] - m1.ort*x.max[1]
abline(q.min.ort, m1.ort, col='forestgreen', lty=2,lwd=1)
abline(q.max.ort, m1.ort, col='forestgreen', lty=2,lwd=1)

m1=worst[2]/worst[1] # worst direction
abline(0, m1, col='grey35')
segments(x.min[1],x.min[2],x.max[1],x.max[2],lty=1,lwd=2, col='forestgreen')
```




```
dev.off()
```

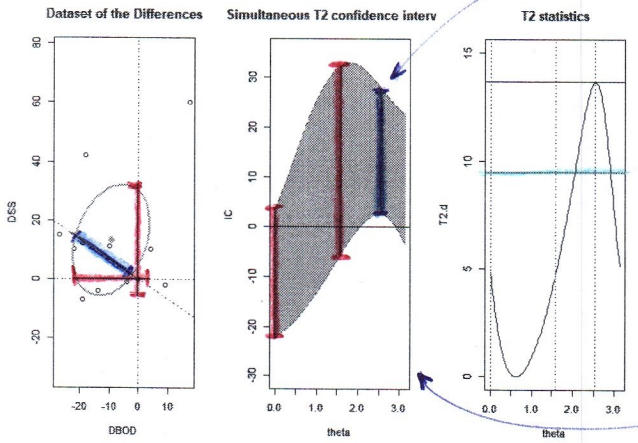
Meaning of the multivariate T^2 statistics = maximum of the univariate T^2 statistics

```
# If we are not convinced yet, let's look at all the directions:
# we compute confidence intervals for a'x where a varies in all the
# directions between 0 and pi, with step pi/180. For each direction
# we compute the T2 statistics (univariate)
D <- as.matrix(D)
theta   <- seq(0, pi - pi/180, by = pi/180)
T2.d      <- NULL
Centerf  <- NULL
Maxf     <- NULL
Minf     <- NULL

for(i in 1:length(theta)){
  a    <- c(cos(theta[i]), sin(theta[i]))
  t2   <- ( mean(D %*% a) - (delta.0 %*% a) )^2 / ( var(D %*% a) / n )
  T2.d  <- c(T2.d, t2)
  centerf  <- D.mean %*% a
  maxf     <- D.mean %*% a + sqrt( t(a) %*% D.cov%*% a / n) * sqrt(cfr.fisher)
  minf     <- D.mean %*% a - sqrt( t(a) %*% D.cov%*% a / n) * sqrt(cfr.fisher)
  Centerf  <- c(Centerf, centerf)
  Maxf     <- c(Maxf, maxf)
  Minf     <- c(Minf, minf)
}

x11(width=21, height=7)
par(mfrow=c(1,3))

plot(D, asp=1, pch=1, main='Dataset of the Differences',ylim=c(-15,60))
abline(h=delta.0[1], v=delta.0[2], col='red', lty=3)
ellipse(center=D.mean, shape=D.cov/n, radius=sqrt(cfr.fisher), lwd=2, col='grey')
segments(IC.T2.DBOD[1],0,IC.T2.DBOD[3],0,lty=1,lwd=2,col='red')
segments(0,IC.T2.DSS[1],0,IC.T2.DSS[3],lty=1,lwd=2,col='red')
x.min <- IC.worst[1]*worst
x.max <- IC.worst[3]*worst
segments(x.min[1],x.min[2],x.max[1],x.max[2],lty=1,lwd=2,col='forestgreen')
abline(0,m1, col='forestgreen',lty=3)
points(delta.0[1], delta.0[2], pch=16, col='black')

plot(theta, Centerf, main = 'Simultaneous T2 confidence intervals', ylim = c(-30,35), col = 'grey25', type='l',ylab='IC')
for(i in 1:length(theta))
{lines(c(theta[i], theta[i]), c(Minf[i], Maxf[i]), col = 'grey75')}
lines(c(theta[1], theta[1]), c(Minf[1], Maxf[1]), col = 'red', lwd=2)
lines(c(theta[91], theta[91]), c(Minf[91], Maxf[91]), col = 'red', lwd=2)
lines(c(theta[which.max(T2.d)], theta[which.max(T2.d)]), c(Minf[which.max(T2.d)], Maxf[which.max(T2.d)]), col = 'forestgree
n', lwd=2)
abline(h=0, col='black')
lines(theta, Minf, col = 'red', lty = 2)
lines(theta, Maxf, col = 'red', lty = 2)

plot(theta, T2.d, main = 'T2 statistics', ylim = c(0,15), col = 'blue', type='l')
abline(v=c(0,pi/2), col = 'red', lty = 3)
abline(v=theta.worst, col = 'forestgreen', lty = 3)

abline(h=cfr.fisher, col = 'grey', lty = 1, lwd=2)

abline(h=D.T2)
```

What happen looking at all the possible directions ?

grid of possible angle values

worst direction



T^2 statistics computed along all the directions

values that defines the rejection region at 5%

CI that we create projecting on the direction which moves with θ moving (these CIs are created by moving θ on the grid we created)

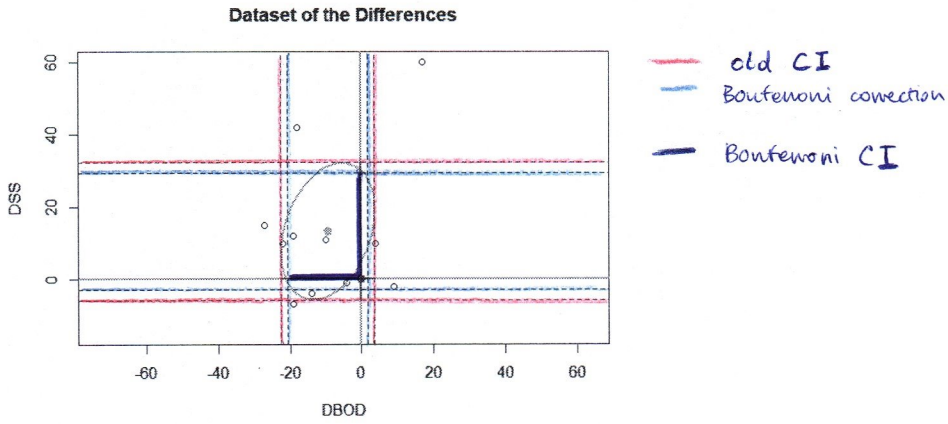
```
dev.off()
```

```
### Bonferroni intervals
k <- p  # 2
cfr.t <- qt(1-alpha/(2*k),n-1)

IC.BF.DBOD <- c( D.mean[1]-cfr.t*sqrt(D.cov[1,1]/n) , D.mean[1], D.mean[1]+cfr.t*sqrt(D.cov[1,1]/n) )
IC.BF.DSS  <- c( D.mean[2]-cfr.t*sqrt(D.cov[2,2]/n) , D.mean[2], D.mean[2]+cfr.t*sqrt(D.cov[2,2]/n) )

Bf <- rbind(IC.BF.DBOD, IC.BF.DSS)
dimnames(Bf)[[2]] <- c('inf','center','sup')
Bf
```

(Bonferroni correction)

```
##                  inf    center       sup
## IC.BF.DBOD -20.573107 -9.363636  1.845835
## IC.BF.DSS   -2.974903 13.272727 29.520358
```

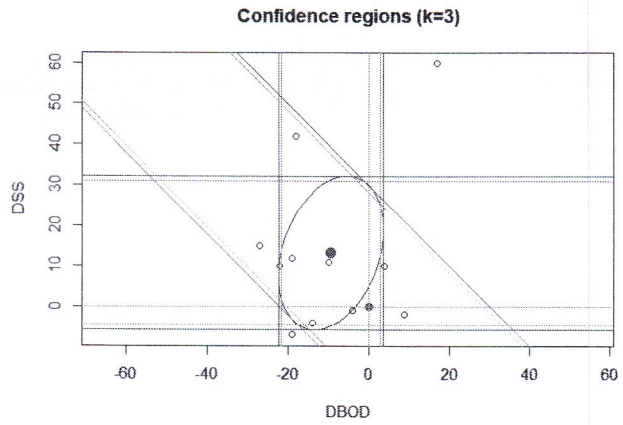
```
x11()
plot(D, asp=1, pch=1, main='Dataset of the Differences',ylim=c(-15,60))
ellipse(center=D.mean, shape=D.cov/n, radius=sqrt(cfr.fisher), lwd=2, col='grey', center.cex=1.25)

abline(h=0, v=0, col='grey', lty=1, lwd=2)
points(delta.0[1], delta.0[2], pch=16, col='grey35', cex=1.25)

abline(v = T2[1,1], col='red', lwd=1, lty=2)
abline(v = T2[1,3], col='red', lwd=1, lty=2)
abline(h = T2[2,1], col='red', lwd=1, lty=2)
abline(h = T2[2,3], col='red', lwd=1, lty=2)
segments(IC.T2.DBOD[1],0,IC.T2.DBOD[3],0,lty=1,lwd=2,col='red')
segments(0,IC.T2.DSS[1],0,IC.T2.DSS[3],lty=1,lwd=2,col='red')

abline(v = Bf[1,1], col='blue', lwd=1, lty=2)
abline(v = Bf[1,3], col='blue', lwd=1, lty=2)
abline(h = Bf[2,1], col='blue', lwd=1, lty=2)
abline(h = Bf[2,3], col='blue', lwd=1, lty=2)
segments(IC.BF.DBOD[1],0,IC.BF.DBOD[3],0,lty=1,lwd=2,col='blue')
segments(0,IC.BF.DSS[1],0,IC.BF.DSS[3],lty=1,lwd=2,col='blue')
```



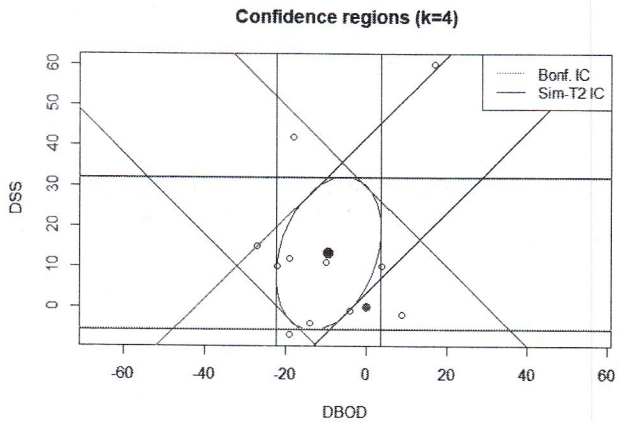

```
dev.off()
```

```
# Note: The simulaneous confidence intervals and the Bonferroni intervals
# are built through different methods
# => they can lead to different conclusions (although in this particular
#    case the conclusion is the same)

# Note 2: Here we have three confidence regions:
# (1) ellipse, (2) T2 rectangle, (3) Bonferroni rectangle
# In general, univariate confidence intervals for k given directions are
# associated with polygonal regions with 2*k sides
# In this example, for k=2 the Bonferroni rectangle is smaller than the
# T2 rectangle. For increasing values of k, the Bonferroni region will
# become larger and larger
```

```
### For instance, let's consider 3 linear combinations:

k <- 3

x11(width=21, height = 7)
# par(mfrow=c(1,3))
plot(D, asp=1, main='Confidence regions (k=3)')
ellipse(center=D.mean, shape=D.cov/n, radius=sqrt(cfr.fisher), lwd=1)
points(delta.0[1], delta.0[2], pch=16, col='grey35', cex=1.25)

theta   <- c(0,pi/4,pi/2)

for(i in 1:length(theta)){
  a       <- c( cos(theta[i]), sin(theta[i]))
  a.orth  <- c(-sin(theta[i]), cos(theta[i]))
  lines( rbind(D.mean + as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a + 100*a.orth,D.mean +
 as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a - 100*a.orth), col='orange', lty=1,lwd=1)
  lines( rbind(D.mean - as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a + 100*a.orth, D.mean -
as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a - 100*a.orth), col='orange', lty=1,lwd=1)
}

abline(h=0, v=0, col='grey')

for(i in 1:length(theta)){
  a       <- c( cos(theta[i]), sin(theta[i]))
  a.orth  <- c(-sin(theta[i]), cos(theta[i]))
  lines( rbind(D.mean + as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a + 100*a.orth, D.mean + as.vect
or(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a - 100*a.orth), col='red', lwd=1, lty=1)
  lines( rbind(D.mean - as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a + 100*a.orth, D.mean - as.vect
or(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a - 100*a.orth), col='red', lwd=1, lty=1)
}
legend('topright', c('Bonf. IC', 'Sim-T2 IC'), col=c('orange', 'red'), lty=1)
```

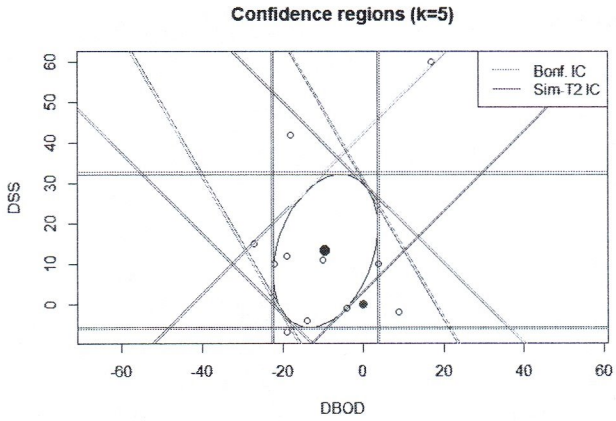



```
### let's add another linear combination
k <- 4

plot(D, asp=1, main='Confidence regions (k=4)')
ellipse(center=D.mean, shape=D.cov/n, radius=sqrt(cfr.fisher), lwd=1)
points(delta.0[1], delta.0[2], pch=16, col='grey35', cex=1.25)

theta   <- c(theta,3*pi/4)

for(i in 1:length(theta)){
  a       <- c( cos(theta[i]), sin(theta[i]))
  a.orth  <- c(-sin(theta[i]), cos(theta[i]))
  lines( rbind(D.mean + as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a + 100*a.orth, D.mean +
as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a - 100*a.orth), col='cyan', lty=1,lwd=1)
  lines( rbind(D.mean - as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a + 100*a.orth, D.mean -
as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a - 100*a.orth), col='cyan', lty=1,lwd=1)
}

for(i in 1:length(theta)){
  a       <- c( cos(theta[i]), sin(theta[i]))
  a.orth  <- c(-sin(theta[i]), cos(theta[i]))
  lines( rbind(D.mean + as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a + 100*a.orth, D.mean + as.vect
or(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a - 100*a.orth), col='blue', lwd=1, lty=1)
  lines( rbind(D.mean - as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a + 100*a.orth, D.mean - as.vect
or(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a - 100*a.orth), col='blue', lwd=1, lty=1)
}
legend('topright', c('Bonf. IC', 'Sim-T2 IC'), col=c('cyan', 'blue'), lty=1)
```



The Bonferroni intervals are becoming longer (they're almost overlapping with the extremes of the simultaneous T^2 CI)

```
### let's add another linear combination
k <- 5

plot(D, asp=1, main='Confidence regions (k=5)')
ellipse(center=D.mean, shape=D.cov/n, radius=sqrt(cfr.fisher), lwd=1)
points(delta.0[1], delta.0[2], pch=16, col='grey35', cex=1.25)

theta   <- c(theta,pi/6)

for(i in 1:length(theta)){
  a       <- c( cos(theta[i]), sin(theta[i]))
  a.orth  <- c(-sin(theta[i]), cos(theta[i]))
  lines( rbind(D.mean + as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a + 100*a.orth, D.mean +
as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a - 100*a.orth), col='green', lty=1,lwd=1)
  lines( rbind(D.mean - as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a + 100*a.orth, D.mean -
as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * qt(1 - alpha/(2*k), n-1)) * a - 100*a.orth), col='green', lty=1,lwd=1)
}

for(i in 1:length(theta)){
  a       <- c( cos(theta[i]), sin(theta[i]))
  a.orth  <- c(-sin(theta[i]), cos(theta[i]))
  lines( rbind(D.mean + as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a + 100*a.orth, D.mean + as.vect
or(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a - 100*a.orth), col='forestgreen', lwd=1, lty=1)
  lines( rbind(D.mean - as.vector(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a + 100*a.orth, D.mean - as.vect
or(sqrt( var(as.matrix(D) %*% a) / n ) * sqrt(cfr.fisher)) * a - 100*a.orth), col='forestgreen', lwd=1, lty=1)
}
legend('topright', c('Bonf. IC', 'Sim-T2 IC'), col=c('green', 'forestgreen'), lty=1)
```



Here the Bonferroni CI are even larger than the simultaneous CI

```
# The Bonferroni region is now bigger than the T2 region

graphics.off()
```

2.

```
###
### Exercise similar to Pb 3 of 14/09/2006
###
# A pharmaceutical company performed a clinical test on 50 rats to investigate
# the effect of a new drug on the blood pressure.
# The blood pressure was measured to each rat four times: just before
# giving the drug, 8, 16, 24 hours after the drug was given.
# (a) Perform a test at level 5% to prove that the drug has influence on
#     the blood pressure during the 24 hours
# (b) Highlight the effect of the drug on the blood pressure

pressure <- read.table ('pressure.txt', col.names=c('h.0','h.8','h.16','h.24'))
head(pressure)
```

TEST FOR REPEATED MEASURES

asks us to provide some CI to see how the pressure changes

Goal: test at 5% to see if the drug has any influence on the blood pressure

```
##          h.0      h.8       h.16      h.24
## 1  3.4588021 7.224847  0.4596315 3.4849473
## 2  2.1210665 5.592171  1.5727843 3.0106627
## 3  2.5967636 4.728428  0.4975863 2.8403774
## 4  3.2415174 5.746075  1.2622631 3.6995064
## 5 -0.2007665 4.952937 -2.5957236 0.4019944
## 6  1.8091951 6.274101  1.0664080 4.0307254
```

```
dim(pressure)
```

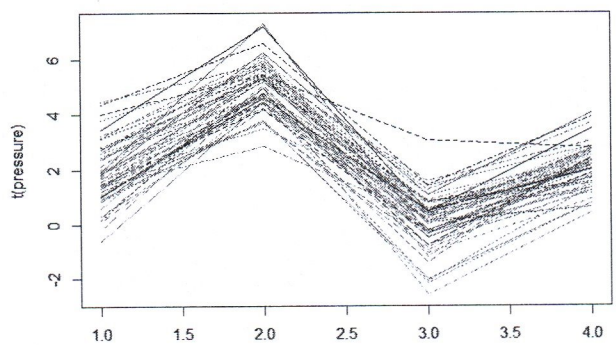
```
## [1] 50  4
```

```
mcshapiro.test(pressure)
```

```
## $Wmin
## [1] 0.9651458
##
## $pvalue
## [1] 0.694
##
## $devst
## [1] 0.009216594
##
## $sim
## [1] 2500
```

→ if the original data is already gaussian we can go on with linear combinations of the variables maintaining the gaussianity

```
x11()
matplot(t(pressure), type='l')
```




```
### question (a)
n <- dim(pressure)[1]
q <- dim(pressure)[2]

M <- sapply(pressure,mean)
M
```

```
##        h.0        h.8       h.16       h.24
## 1.86322042 5.06637474 0.04654578 2.09619753
```

```
S <- cov(pressure)
S
```

```
##            h.0       h.8      h.16      h.24
## h.0  1.3223812 0.5260806 0.6070589 0.5725473
## h.8  0.5260806 0.8545744 0.3124937 0.4342793
## h.16 0.6070589 0.3124937 1.1009830 0.4836133
## h.24 0.5725473 0.4342793 0.4836133 0.6995795
```

```
# we build one of the possible contrast matrices to answer
# the question
C <- matrix(c(-1, 1, 0, 0,
              -1, 0, 1, 0,
              -1, 0, 0, 1), 3, 4, byrow=T)
C
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   -1    1    0    0
## [2,]   -1    0    1    0
## [3,]   -1    0    0    1
```

```
# here we are looking at the effects on the pressure
# after 8, 16 and 24 hours from the instant the drug was
# given

# Test: H0: C%*%mu == 0 vs H1: C%*%mu != 0
alpha   <- .05
delta.0 <- c(0,0,0)

Md <- C %*% M
Sd <- C %*% S %*% t(C)
Sdinv <- solve(Sd)

T2 <- n * t( Md - delta.0 ) %*% Sdinv %*% ( Md - delta.0 )

cfr.fisher <- ((q-1)*(n-1)/(n-(q-1)))*qf(1-alpha,(q-1),n-(q-1))

T2 < cfr.fisher
```

$H_0: C\underline{\mu} = \underline{0}$
$H_1: C\underline{\mu} \neq \underline{0}$

```
##       [,1]
## [1,] FALSE
```

```
T2
```

```
##         [,1]
## [1,] 1029.63
```

$T^2 >> cfr.fisher \Longrightarrow$ we reject H_0

```
cfr.fisher
```

```
## [1] 8.764813
```

we always going to reject H_0 (whatever level since the p-value is 0)

```
# T2 is much higher than cfr.fisher => the p-value will be very small
P <- 1-pf(T2*(n-(q-1))/((q-1)*(n-1)),(q-1),n-(q-1))
P
```

```
##      [,1]
## [1,]    0
```

```
### question (b)

# It is implicitely asking for confidence intervals on the components
# (for the mean of the increments after 8 hours, 16 hours and 24 hours)

# Simultaneous T2 intervals
IC.T2 <- cbind( Md - sqrt(cfr.fisher*diag(Sd)/n) , Md, Md + sqrt(cfr.fisher*diag(Sd)/n) )
IC.T2
```

```
##             [,1]       [,2]       [,3]
## [1,]  2.7591135  3.2031543  3.6471952
## [2,] -2.2770835 -1.8166746 -1.3562657
## [3,] -0.1590833  0.2329771  0.6250375
```

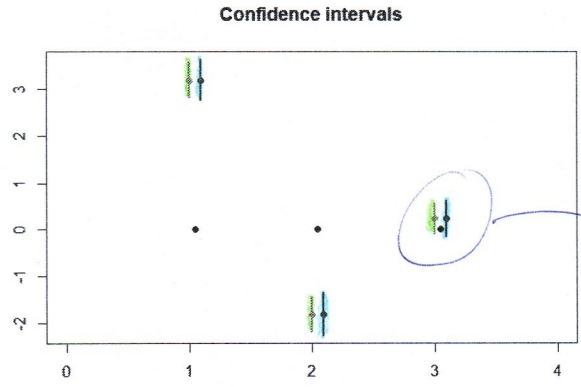
$\longrightarrow$ 2 out of 3 do not contain 0

```
# Bonferroni intervals
k      <- q - 1   # number of increments (i.e., dim(C)[1])
cfr.t <- qt(1-alpha/(2*k),n-1)

IC.BF <- cbind( Md - cfr.t*sqrt(diag(Sd)/n) , Md, Md + cfr.t*sqrt(diag(Sd)/n) )
IC.BF
```

```
##             [,1]       [,2]       [,3]
## [1,]  2.83134533  3.2031543  3.5749633
## [2,] -2.20218911 -1.8166746 -1.4311602
## [3,] -0.09530707  0.2329771  0.5612613
```

$\longrightarrow$ 2 out of 3 do not contain 0

```
x11()
matplot(t(matrix(1:3,3,3)),t(IC.BF), type='b',pch='',xlim=c(0,4),xlab='',ylab='', main='Confidence intervals')
segments(matrix(1:3,3,1),IC.BF[,1],matrix(1:3,3,1),IC.BF[,3], col='orange', lwd=2)
points(1:3, IC.BF[,2], col='orange', pch=16)
points(1:3+.05, delta.0, col='black', pch=16)
segments(matrix(1:3+.1,3,1),IC.T2[,1],matrix(1:3+.1,3,1),IC.T2[,3], col='blue', lwd=2)
points(1:3+.1,IC.T2[,2], col='blue', pch=16)
```



Simultaneous T^2 CI

Bonferroni correction CI (smaller than the simultaneous)

In this (only) case 0 belongs to the CI ⇒ we don't have enough statistical evidence to reject H_0 looking only at this component of the dataset

```
### what happens if we change the constrast matrix?
Cbis <- matrix(c(-1, 1, 0, 0,
                 0, -1, 1, 0,
                 0, 0, -1, 1), 3, 4, byrow=T)
Cbis
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   -1    1    0    0
## [2,]    0   -1    1    0
## [3,]    0    0   -1    1
```

```
# in this way we are looking at the mean increment of the pressure every 8 hours

Mdbis <- Cbis %*% M
Sdbis <- Cbis %*% S %*% t(Cbis)
Sdinvbis <- solve(Sdbis)

T2bis <- n * t( Mdbis ) %*% Sdinvbis %*% Mdbis

T2bis < cfr.fisher
```

```
##       [,1]
## [1,] FALSE
```

→ there is enough statistical evidence against H_0

```
# compare the T2 test statistics associated with C and Cbis
T2bis
```

```
##         [,1]
## [1,] 1029.63
```

```
T2
```

```
##         [,1]
## [1,] 1029.63
```

they're the same! That's because the test statistic has the same meaning not depending on the specific matrix (contrast matrix) that we've considering

```
# What is changed?
# The confidence intervals on the constrasts
# (because we are looking at different contrasts!)

IC.BFbis <- cbind( Mdbis - cfr.t*sqrt(diag(Sdbis)/n) , Mdbis, Mdbis + cfr.t*sqrt(diag(Sdbis)/n) )
IC.T2bis <- cbind( Mdbis - sqrt(cfr.fisher*diag(Sdbis)/n) , Mdbis, Mdbis + sqrt(cfr.fisher*diag(Sdbis)/n) )

IC.BFbis
```

```
##           [,1]      [,2]      [,3]
## [1,]  2.831345  3.203154  3.574963
## [2,] -5.424221 -5.019829 -4.615437
## [3,]  1.729620  2.049652  2.369684
```

```
IC.BF
```

```
##             [,1]       [,2]       [,3]
## [1,]  2.83134533  3.2031543  3.5749633
## [2,] -2.20218911 -1.8166746 -1.4311602
## [3,] -0.09530707  0.2329771  0.5612613
```

```
IC.T2bis
```

```
##           [,1]      [,2]      [,3]
## [1,]  2.759113  3.203154  3.647195
## [2,] -5.502782 -5.019829 -4.536876
## [3,]  1.667447  2.049652  2.431857
```

```
IC.T2
```

```
##             [,1]       [,2]       [,3]
## [1,]  2.7591135  3.2031543  3.6471952
## [2,] -2.2770835 -1.8166746 -1.3562657
## [3,] -0.1590833  0.2329771  0.6250375
```

different values but same conclusion (we're looking in different ways at the data (C vs. Cbis) but the question is always: is the drug making an effect?)

```
### what if we want to verify the following hypothesis:
### "the drug decreases the pressure of two units with respect to
### the baseline at both 8 and 16 hours, and its effect vanishes in 24 hours
### from the drug administration"

C <- matrix(c(-1, 1, 0, 0,
              -1, 0, 1, 0,
              -1, 0, 0, 1), 3, 4, byrow=T)
delta.0 <- c(-2,-2,0)
```

```
# or
C <- matrix(c(-1, 1, 0, 0,
              0, -1, 1, 0,
              0, 0, -1, 1), 3, 4, byrow=T)
delta.0 <- c(-2,0,2)

Md <- C %*% M
Sd <- C %*% S %*% t(C)
Sdinv <- solve(Sd)

T2 <- n * t( Md - delta.0 ) %*% Sdinv %*% ( Md - delta.0 )

cfr.fisher <- ((q-1)*(n-1)/(n-(q-1)))*qf(1-alpha,(q-1),n-(q-1))
T2 < cfr.fisher
```

```
##      [,1]
## [1,] FALSE
```

→ again we reject H_0 (expected, because of the old plot)

```
T2
```

```
##          [,1]
## [1,] 2010.284
```

```
cfr.fisher
```

```
## [1] 8.764813
```

```
# p-value
P <- 1-pf(T2*(n-(q-1))/((q-1)*(n-1)),(q-1),n-(q-1))
P
```

```
##      [,1]
## [1,]    0
```

---

3. Ex.

TEST FOR THE **MEAN OF 2 INDEPENDENT GAUSSIAN POPULATION**

```
# we build the data
t1 <- matrix(c(3,3,1,6,2,3),2)
t1 <- data.frame(t(t1))
t2 <- matrix(c(2,3,5,1,3,1,2,3),2)
t2 <- data.frame(t(t2))

t1
```

```
##   X1 X2
## 1  3  3
## 2  1  6
## 3  2  3
```

```
t2
```

```
##   X1 X2
## 1  2  3
## 2  5  1
## 3  3  1
## 4  2  3
```

```
n1 <- dim(t1)[1] # n1=3
n2 <- dim(t2)[1] # n2=4
p  <- dim(t1)[2] # p=2

# we compute the sample mean, covariance matrices and the matrix
# Spooled

t1.mean <- sapply(t1,mean)
t2.mean <- sapply(t2,mean)
t1.cov  <-  cov(t1)
t2.cov  <-  cov(t2)
Sp      <- ((n1-1)*t1.cov + (n2-1)*t2.cov)/(n1+n2-2)
# we compare the matrices
list(S1=t1.cov, S2=t2.cov, Spooled=Sp)
```

needed for the test of independent populations

```
## $S1
##      X1   X2
## X1  1.0 -1.5
## X2 -1.5  3.0
##
## $S2
##           X1        X2
## X1  2.000000 -1.333333
## X2 -1.333333  1.333333
##
## $Spooled
##      X1   X2
## X1  1.6 -1.4
## X2 -1.4  2.0
```

```
# Test H0: mu1 == mu2  vs  H1: mu1 != mu2
# i.e.,
# Test H0: mu1-mu2 == c(0,0)  vs  H1: mu1-mu2 != c(0,0)

alpha   <- .01
delta.0 <- c(0,0)
Spinv   <- solve(Sp)

T2 <- n1*n2/(n1+n2) * (t1.mean-t2.mean-delta.0) %*% Spinv %*% (t1.mean-t2.mean-delta.0)

cfr.fisher <- (p*(n1+n2-2)/(n1+n2-1-p))*qf(1-alpha,p,n1+n2-1-p)
T2 < cfr.fisher # TRUE: no statistical evidence to reject H0 at level 1%
```

```
##      [,1]
## [1,] TRUE
```

← we have NO statistical evidence to reject H_0

```
P <- 1 - pf(T2/(p*(n1+n2-2)/(n1+n2-1-p)), p, n1+n2-1-p)
P
```

```
##          [,1]
## [1,] 0.317686
```

```
# P-value high (we don't reject at 1%,5%,10%)

# Simultaneous T2 intervals
IC.T2.X1 <- c(t1.mean[1]-t2.mean[1]-sqrt(cfr.fisher*Sp[1,1]*(1/n1+1/n2)), t1.mean[1]-t2.mean[1]+sqrt(cfr.fisher*Sp[1,1]*(1/n
1+1/n2)) )
IC.T2.X2 <- c(t1.mean[2]-t2.mean[2]-sqrt(cfr.fisher*Sp[2,2]*(1/n1+1/n2)), t1.mean[2]-t2.mean[2]+sqrt(cfr.fisher*Sp[2,2]*(1/n
1+1/n2)) )
IC.T2 <- rbind(IC.T2.X1, IC.T2.X2)
dimnames(IC.T2)[[2]] <- c('inf','sup')
IC.T2
```

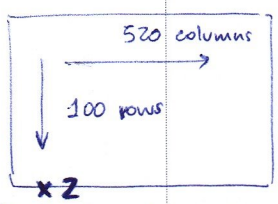
```
##                inf      sup
## IC.T2.X1 -7.480741 5.480741
## IC.T2.X2 -5.245688 9.245688
```

} in both cases the CI contains the 0 (CI for the difference)

```
graphics.off()
```

```
###
### Pb 2 of 4/07/2013
### ---------------------------------------------------------------------------
# The Chinese Institute of Genomics has reconstructed the genetic map of 200
# Chinese individuals, among which 100 were allergic to almonds and 100 were not
# allergic to almonds. The files hatingalmonds.txt e lovingalmonds.txt collect
# the presence (1) or absence (0) of 520 mutations, for the two groups respectively.
# The genetists suspect that some of these mutations might be positively associated
# with the allergy to almonds, and for this reason they decide to compare the
# incidence of the 520 mutations in the two populations.
# a) Report the significant mutations, imposing a probability of at most 1%
#    that the single mutation is judged as influent if it isn't.
# b) Report the significant mutations, imposing a probability of at most 1%
#    that at Least one of the non-influent mutations is judged as influent.
# c) Report the significant mutations, imposing that the expected proportion
#    of mutations erroneously judged to be influent among all those judged to be
#    influent is lower than or equal to 1%
### ---------------------------------------------------------------------------
# You'll have to correct the univariate p-values of multiple tests
# R function to correct p-values: p.adjust

allergy <- read.table('hatingalmonds.txt')
#head(allergy)
dim(allergy)
```

→ **Goal**: which mutation is associated with the allergy to almonds?



(one for allergic and one for not all.)

control over single mutation (impose probability for single mutations, not the overall test) (ONE-AT-TIME)

here we have to have an overall control (SIMULTANEOUS-TESTING/ FINITE-NUMB. OF TESTING/ BONFERRONI CORRECTION)

FALSE DISCOVERY RATE

```
## [1] 100 520
```

```
noallergy <- read.table('lovingalmonds.txt')
#head(noallergy)
dim(noallergy)
```

```
## [1] 100 520
```

```
n1 <- dim(allergy)[1]
n2 <- dim(noallergy)[1]
p <- dim(noallergy)[2]

x.mean1 <- sapply(allergy, mean)
x.mean2 <- sapply(noallergy, mean)

p.hat <- (x.mean1*n1+x.mean2*n2)/(n1+n2)
x.var <- (p.hat*(1-p.hat))

# Test: H0.i: mu.i1 == mu.i2  vs H1.i: mui1 != mu.i2

z.i <- (x.mean1-x.mean2)/sqrt(x.var*(1/n1+1/n2))
p.i <- ifelse(z.i<0, 2*pnorm(z.i),2*(1-pnorm(z.i)))

which(p.i<.01)   (α = 0,01)
```

```
## M152 M160 M174 M188 M227 M258 M291 M336 M345 M355 M372 M446
##  152  160  174  188  227  258  291  336  345  355  372  446
```

```
# Bonferoni test
k <- 520

which(p.i*k<.01)
```

```
## M188 M372 M446
##  188  372  446
```

```
# or
p.Bf <- p.adjust(p.i, method='bonferroni')

which(p.Bf<.01)
```

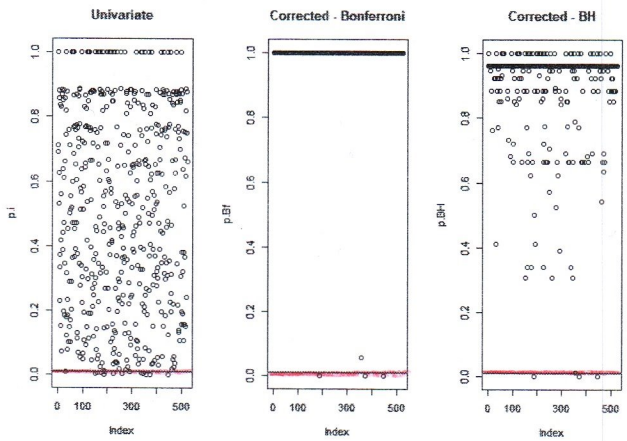
```
## M188 M372 M446
##  188  372  446
```

```
# Benjamini-Hockberg (control the false discovery rate)
p.BH <- p.adjust(p.i, method='BH')

which(p.BH<.01)
```

this function adjust p-values in different ways, we need the **BH** correction

```
## M188 M372 M446
##  188  372  446
```

```
x11(width=21, height=7)
par(mfrow=c(1,3))
plot(p.i, main='Univariate')
abline(h=.01, lwd=2, col='red')

plot(p.Bf, main='Corrected - Bonferroni')
abline(h=.01, lwd=2, col='red')

plot(p.BH, main='Corrected - BH')
abline(h=.01, lwd=2, col='red')
```



0.01
(threshold we wanted to impose)

LAB 06 - Additional Exercices

```
load("mcshapiro.test.RData")
library(car)
```

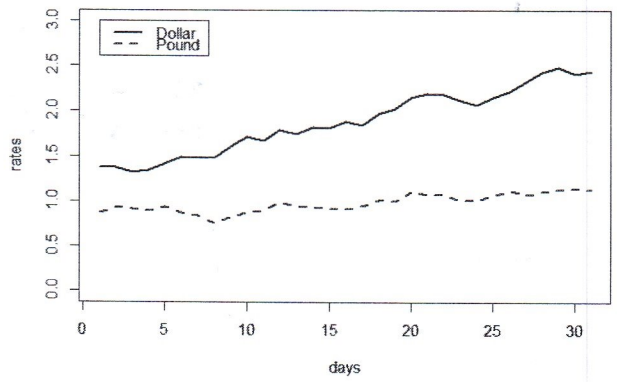
```
### ------------------------------------------------------------------------
### Pb 1 of 10/02/10
### ------------------------------------------------------------------------
# The file exchange.txt collects the daily exchange rates dollar/euro and
# pound/euro of Jan 2010. Assume that the 30 daily increments are independent
# realization of a bivariate Gaussian distribution.
# a) Is there statistical evidence to state that during January the exchange
#    rate has changed in mean?
# b) Using the Bonferroni inequality, provide four confidence intervals of
#    global confidence 90% for the mean of the increments and for their
#    variances.
### ------------------------------------------------------------------------

rates <- read.table(file='exchange.txt', header=T)
head(rates)
```

```
##   dollar  pound
## 1 1.3700 0.8700
## 2 1.3726 0.9261
## 3 1.3156 0.9088
## 4 1.3413 0.8939
## 5 1.4141 0.9291
## 6 1.4847 0.8636
```

31 × 2

```
x11()
matplot(rates, type='l', ylim=c(0,3), lwd = 2, xlab='days')
legend(1,3,legend=c('Dollar','Pound'),col=1:2,lwd=2,lty=1:2)
```

→ because we have a dimension that can be interpreted as time (so the x axes can be interpreted as time)



remember:

if the data has time as columns (like features) then we have to transpose the data we give as input

```
### question a)
# we need to build the correct data matrix:
# we know that the increments are independent realizations of a
# bivariate Gaussian

diffrates <- matrix(NA, 30, 2)
for(i in 1:30)
  diffrates[i,] <- as.numeric(rates[i+1,] - rates[i,])

head(diffrates)
```

} we're looking at the increments

```
##         [,1]    [,2]
## [1,]  0.0026  0.0561
## [2,] -0.0570 -0.0173
## [3,]  0.0257 -0.0149
## [4,]  0.0728  0.0352
## [5,]  0.0706 -0.0655
## [6,] -0.0061 -0.0289
```

```
# we assume that the sample is now iid from a bivariate normal distr.
# H0: rates did not change => mean of increments == zero

# we first need to verify the Gaussian assumption
plot(diffrates, asp=1, pch=1)
mcshapiro.test(diffrates)
```

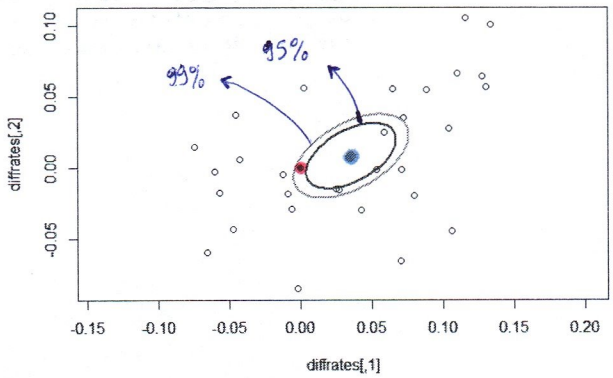
$H_0: \mu = 0$ μ = mean of increments

```
## $Wmin
## [1] 0.9750956
##
## $pvalue
## [1] 0.7788
##
## $devst
## [1] 0.008301098
##
## $sim
## [1] 2500
```

```
mu0      <- c(0, 0)
x.mean   <- colMeans(diffrates)
x.cov    <- cov(diffrates)
x.invcov <- solve(x.cov)
n <- 30
p <- 2

x.T2       <- n * (x.mean-mu0) %*% x.invcov %*% (x.mean-mu0)
Pb <- 1-pf(x.T2*(n-p)/(p*(n-1)), p, n-p)
Pb
```

```
##             [,1]
## [1,] 0.02303093
```

```
# mean under H0 (blue)
points(mu0[1], mu0[2], col='blue', pch=16)

# sample mean (black)
points(x.mean[1], x.mean[2], col='black', pch=16)

# we represent the confidence region of level 95%: where does mu0 fall?
alpha <- .05
cfr.fisher <- (p*(n-1)/(n-p))*qf(1-alpha,p,n-p)
ellipse(center=x.mean, shape=x.cov/n, radius=sqrt(cfr.fisher), lwd=2)

# what about the region of level 99%?
alpha <- .01
cfr.fisher <- (p*(n-1)/(n-p))*qf(1-alpha,p,n-p)
ellipse(center=x.mean, shape=x.cov/n, radius=sqrt(cfr.fisher), lwd=2, col='orange', add=TRUE)
```



● = mean under H_0
● = sample mean

95% → we reject H_0
99% → we accept H_0

It doesn't matter that we've computing 2 CI for the mean and 2 CI for the variance, they are totally 4 ⇒ the correction will be with: $k = 4$

```
dev.off()
```

```
### question b): Bonferroni method (intervals for the components of the mean AND on the variances)

k <- 4
alpha <- 0.1

ICmean <- cbind(inf=x.mean - sqrt(diag(x.cov)/n) * qt(1 - alpha/(2*k), n-1),
                center= x.mean,
                sup= x.mean + sqrt(diag(x.cov)/n) * qt(1 - alpha/(2*k), n-1))

ICvar <- cbind(inf=diag(x.cov)*(n-1) / qchisq(1 - alpha/(2*k), n-1),
               center=diag(x.cov),
               sup=diag(x.cov)*(n-1) / qchisq(alpha/(2*k), n-1))

ICmean
```

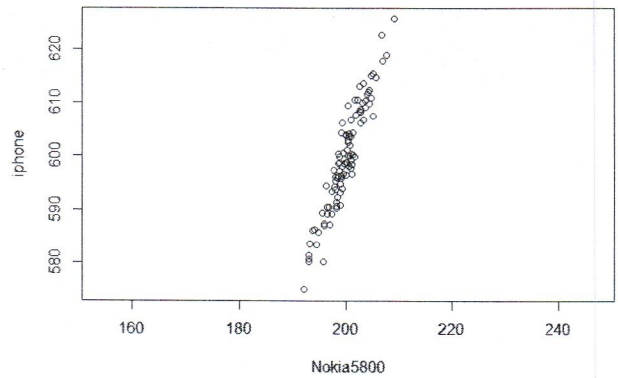
This is from Stat 101 (the formula of the variance CI)

```
##              inf     center        sup
## [1,]  0.00691055 0.03554667 0.06418278
## [2,] -0.01210906 0.00849000 0.02908906
```

```
ICvar
```

```
##              inf      center         sup
## [1,] 0.002623096 0.004402619 0.008710100
## [2,] 0.001357315 0.002278125 0.004507021
```

```
### ---------------------------------------------------------------------------
### Pb 3 of 10/02/10
### ---------------------------------------------------------------------------
# An association of Milanese technologists collected in the file mobile.txt
# the prices of iphone and Nokia 5800 recorded for 100 shops in the province
# of Milan
# a) Build three simultaneous T2 intervals with global confidence 90%
#    for the mean of the two prices and for their difference
# b) Nokia states that - at a worldwide level - Nokia 5800 costs in mean
#    one third of the price for an iphone. Based on the data, can we
#    deny this statement for the province of Milan?
### ---------------------------------------------------------------------------

mobile <- read.table('mobile.txt', header=T)
plot(mobile, asp=1)
```




```
dev.off()
```

```
mcshapiro.test(mobile)
```

```
## $Wmin
## [1] 0.9913581
##
## $pvalue
## [1] 0.8472
##
## $devst
## [1] 0.007195892
##
## $sim
## [1] 2500
```

```
### question a)
n <- dim(mobile)[1]
p <- dim(mobile)[2]
x.mean   <- sapply(mobile,mean)
x.cov    <- cov(mobile)

alpha <- 0.10
cfr.fisher <- (p*(n-1)/(n-p))*qf(1-alpha,p,n-p)

A <- rbind(c(1,0), c(0,1), c(-1,1))

ICT2 <- cbind(A %*% x.mean - sqrt(diag(A %*% x.cov %*% t(A))/n * cfr.fisher),
              A %*% x.mean,
              A %*% x.mean + sqrt(diag(A %*% x.cov %*% t(A))/n * cfr.fisher))
ICT2
```

```
##          [,1]     [,2]     [,3]
## [1,] 199.2544 200.0107 200.7670
## [2,] 597.6723 599.9071 602.1419
## [3,] 398.3600 399.8964 401.4328
```

```
plot(mobile, asp=1)
ellipse(x.mean, x.cov/n, cfr.fisher, add=T)

### question b)
# Test:
# H0: mu_nokia == mu_iphone/3 vs H1: mu_nokia != mu_iphone/3
# i.e.
# H0: mu_iphone - 3* mu_nokia == 0 vs H1: mu_iphone - 3* mu_nokia != 0
# (test a particular linear combination)

M   <- mobile[,2] - 3*mobile[,1]
t.M <- abs(mean(M))/sqrt(var(M)/n)
t.M
```

```
## [1] 0.3767702
```

```
P   <- (1 - pt(t.M, n-1))*2
P
```

```
## [1] 0.7071507
```

```
# or: (it is exactly analogous)
a <- c(-3, 1)
t.M.bis <- abs(x.mean%*%a)/sqrt((t(a)%*%x.cov%*%a)/n)
t.M.bis
```

```
##          [,1]
## [1,] 0.3767702
```

```
P   <- (1 - pt(t.M.bis, n-1))*2
P
```

```
##          [,1]
## [1,] 0.7071507
```

```
### ---------------------------------------------------------------------------
### Pb 3 of 24/09/10
### ---------------------------------------------------------------------------
# The PoliPharma has given a drug to 10 students in order to increase the levels
# of Matina and decrease the level of Fisina in their blood. In the files before.txt
# and after.txt the levels of Matina, Fisina, Chimina and Elettrina are reported
# for 10 students, before and after the administration of the drug.
# a) Having introduced and tested the appropriate hypotheses of Gaussianity,
#    perform a test of level 1% to prove the existence of an effect of the
#    drug on the mean levels of the four enzymes.
# b) Provide four simultaneous T2 intervals for the mean of the four
#    increments
# c) Perform a test of level 1% to confirm/deny what stated by the Polipharma,
#    that is: the ingestion of the drag causes a mean increment of 2 units of
#    the Matina, a decrease of 1 unit in the Fisina and a mean increment in the
#    Chimina equal to the mean decrease in the Elettrina
### ---------------------------------------------------------------------------

before <- read.table('before.txt', header=T)
after <- read.table('after.txt', header=T)

### question a)
D <- after - before

# verificy normality
mcshapiro.test(D)
```

```
## $Wmin
## [1] 0.7632154
##
## $pvalue
## [1] 0.41
##
## $devst
## [1] 0.009836666
##
## $sim
## [1] 2500
```

```
# test
alpha <- 0.01
n <- dim(D)[1]
p <- dim(D)[2]

D.mean   <- sapply(D,mean)
D.cov    <- cov(D)
D.invcov <- solve(D.cov)

delta.0 <- c(0,0,0,0)

D.T2 <- n * (D.mean-delta.0) %*% D.invcov %*% (D.mean-delta.0)

cfr.fisher <- (p*(n-1)/(n-p))*qf(1-alpha,p,n-p)
D.T2 < cfr.fisher
```

```
##      [,1]
## [1,] TRUE
```

```
# pvalue (not requested)
P <- 1-pf(D.T2 * (n-p)/(p*(n-1)), p, n-p)
P
```

```
##            [,1]
## [1,] 0.01427774
```

```
### question b)
# Simultaneous T2 intervals for the components
T2 <- cbind( "Inf"= D.mean-sqrt(cfr.fisher*diag(D.cov)/n) , D.mean, Sup=D.mean+sqrt(cfr.fisher*diag(D.cov)/n) )
T2
```

```
##                Inf  D.mean      Sup
## matina    -1.020980  2.2063 5.433580
## fisina    -5.065534  0.8020 6.669534
## chimina   -4.935618  1.3602 7.656018
## elettrina -5.076943 -1.2858 2.505343
```

```
### question c)
# we want to perform a test of global level 1% for the three linear
# combination of the mean of D (datasets of the differences)
# H0: delta1 = 2, delta2 = -1, delta3 = - delta4
# i.e.,
# H0: delta1 = 2, delta2 = -1, delta3 + delta4 = 0

A <- rbind(c(1,0,0,0),c(0,1,0,0),c(0,0,1,1))

## Way 1: we build a new dataset
D.new <- data.frame(as.matrix(D) %*% t(A))

p.new <- dim(D.new)[2]
n <- dim(D.new)[1]

alpha <- 0.01

D.new.mean   <- sapply(D.new,mean)
D.new.cov    <-  cov(D.new)
D.new.invcov <- solve(D.new.cov)

delta.0 <- c(2,-1,0)

D.T2 <- n * (D.new.mean-delta.0) %*% D.new.invcov %*% (D.new.mean-delta.0)
D.T2
```

```
##          [,1]
## [1,] 5.775267
```

```
cfr.fisher.new <- (p.new*(n-1)/(n-p.new))*qf(1-alpha,p.new,n-p.new)
D.T2 < cfr.fisher.new
```

```
##      [,1]
## [1,] TRUE
```

```
# pvalue (not requested)
P <- 1-pf(D.T2 * (n-p.new)/(p.new*(n-1)), p.new, n-p.new)
P
```

```
##           [,1]
## [1,] 0.2964435
```

```
## Way 2 (totally analogous):
## We perform a test for
## H0: A%*%mu.D == delta.0 vs H1: A%*%mu.D != delta.0
A <- rbind(c(1,0,0,0),c(0,1,0,0),c(0,0,1,1))
delta.0 <- c(2,-1,0)

T2.A <- n * t(A %*% D.mean - delta.0) %*% solve(A %*% D.cov %*% t(A)) %*% (A %*% D.mean - delta.0)
T2.A
```

```
##          [,1]
## [1,] 5.775267
```

```
T2.A < cfr.fisher.new
```

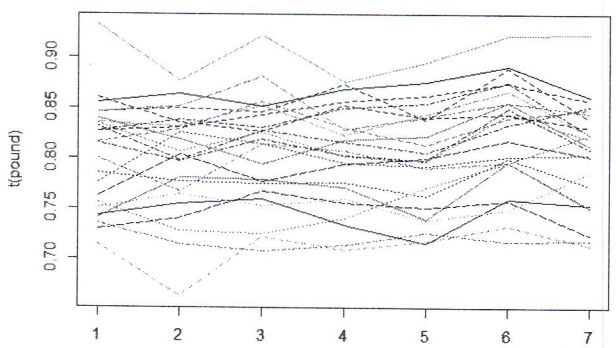
```
##      [,1]
## [1,] TRUE
```

```
# WARNING: here, the dimension 'p' that we use for cfr.fisher
#          is p.new=3, it isn't the initial dimension of the data
#          (that was p=4) [ if I kept p=4 I would have obtained a
#          global level lower than 1% ]
```

```
### ------------------------------------------------------------------------
### Pb 2 of 10/09/10
### ------------------------------------------------------------------------
# The file pound.txt collects the exchange rates pound/euro used by 24 banks in
# UK during the week between 30th Aug and 5th Sept 2010.
# a) Having framed the problem in the context of repeated measures, perform
#    a test of level 5% to verify the hypothesis that the mean of the exchange
#    rate is constant along time
# b) Provide 6 confidence intervals of global level 95% for the daily increments
#    in the exchange rate
# c) Comment the 6 intervals computed at point b).
### ------------------------------------------------------------------------

pound <- read.table('pound.txt', header=T)
head(pound)
```

```
##         X1       X2       X3       X4       X5       X6       X7
## 1 0.742886 0.754892 0.759030 0.733303 0.714801 0.759761 0.753529
## 2 0.846103 0.849913 0.845708 0.874489 0.838584 0.889250 0.839849
## 3 0.834855 0.806727 0.825796 0.801886 0.796831 0.844263 0.819960
## 4 0.827320 0.838127 0.831164 0.850420 0.854521 0.875602 0.824354
## 5 0.835563 0.831413 0.852715 0.822876 0.844074 0.867555 0.824301
## 6 0.741329 0.780281 0.778861 0.770540 0.739152 0.795981 0.751003
```

```
matplot(t(pound), type='l')
```




```
### question a)
n <- dim(pound)[1]
q <- dim(pound)[2]

M <- sapply(pound,mean)
S <- cov(pound)

# matrix of contrasts that looks at the daily increments
C   <-  matrix(c(-1, 1, 0, 0, 0, 0, 0,
                 0, -1, 1, 0, 0, 0, 0,
                 0, 0, -1, 1, 0, 0, 0,
                 0, 0, 0, -1, 1, 0, 0,
                 0, 0, 0, 0, -1, 1, 0,
                 0, 0, 0, 0, 0, -1, 1), 6, 7, byrow=T)
C
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,]   -1    1    0    0    0    0    0
## [2,]    0   -1    1    0    0    0    0
## [3,]    0    0   -1    1    0    0    0
## [4,]    0    0    0   -1    1    0    0
## [5,]    0    0    0    0   -1    1    0
## [6,]    0    0    0    0    0   -1    1
```

```
# I could choose any contrast matrix but if I look at the following
# questions I can save energy and time!
mcshapiro.test(pound)
```

```
## $Wmin
## [1] 0.8502836
##
## $pvalue
## [1] 0.6936
##
## $devst
## [1] 0.009219957
##
## $sim
## [1] 2500
```

```
# Test: H0: C%*%mu=0 vs H1: C%*%mu!=0
alpha   <- .05
delta.0 <- c(0, 0, 0, 0, 0, 0)

Md <- C %*% M
Sd <- C %*% S %*% t(C)
Sdinv <- solve(Sd)

T2 <- n * t( Md - delta.0 ) %*% Sdinv %*% ( Md - delta.0 )

cfr.fisher <- ((q-1)*(n-1)/(n-(q-1)))*qf(1-alpha,(q-1),n-(q-1))
# attention to the multiplying factor!
T2 < cfr.fisher
```

```
##       [,1]
## [1,] FALSE
```

```
P <- 1-pf(T2*(n-(q-1))/((q-1)*(n-1)),(q-1),n-(q-1))
P
```

```
##             [,1]
## [1,] 0.000115767
```

```
### questions b) and c)
k <- q-1
ICmean <- cbind(Md - sqrt(diag(Sd)/n) * qt(1 - alpha/(2*k), n-1),
                Md,
                Md + sqrt(diag(Sd)/n) * qt(1 - alpha/(2*k), n-1))
ICmean
```

```
##              [,1]         [,2]        [,3]
## [1,] -0.021006674 -0.004772292  0.011462090
## [2,] -0.004359247  0.009401250  0.023161747
## [3,] -0.018334396 -0.004834792  0.008664813
## [4,] -0.012540259 -0.002738042  0.007064175
## [5,]  0.014009720  0.024156333  0.034302947
## [6,] -0.032226776 -0.017427958 -0.002629141
```

```
### questions b) e c)
for (i in 1:(q-1))
  print(paste('Reject H0 in direction ',i,': ', !(delta.0[i]>ICmean[i,1] & delta.0[i]<ICmean[i,3]),sep=''))
```

```
## [1] "Reject H0 in direction 1: FALSE"
## [1] "Reject H0 in direction 2: FALSE"
## [1] "Reject H0 in direction 3: FALSE"
## [1] "Reject H0 in direction 4: FALSE"
## [1] "Reject H0 in direction 5: TRUE"
## [1] "Reject H0 in direction 6: TRUE"
```

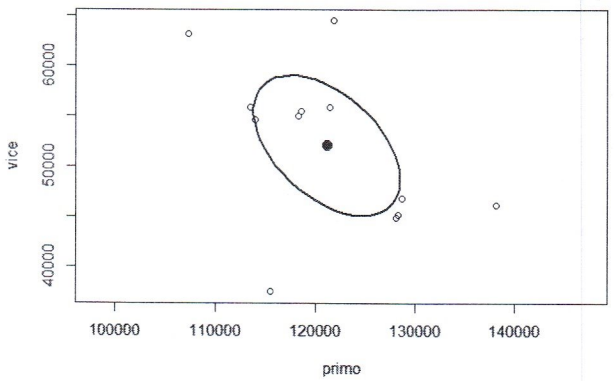
```
### ---------------------------------------------------------------------------
### Pb 3 of 10/09/10
### ---------------------------------------------------------------------------
# The file extra.txt reports the representation expenses [$] of the English
# first minister and of his vice during the first 12 months of 2009. Assume
# those data to be independent realizations of a bivariate Gaussian.
# a) Build an ellipsoidal region of confidence 90% for the mean of the
#    representation expenses
# b) Is there evidence of the fact that the prime minister spends in mean
#    more than twice the expences of its vice?
# c) build a confidence interval of level 90% for the mean of the sum of
#    the expenses.
### ---------------------------------------------------------------------------

extra <- read.table('extra.txt', header=T)

plot(extra, asp=1)
extra
```

```
##     primo  vice
## 1  115544 37428
## 2  113516 55842
## 3  114009 54626
## 4  128119 44841
## 5  128325 45165
## 6  107285 63207
## 7  118304 54975
## 8  118618 55427
## 9  121905 64466
## 10 121501 55848
## 11 128744 46769
## 12 138115 46130
```

```
### question a)
mcshapiro.test(extra)
```

```
## $Wmin
## [1] 0.8548324
##
## $pvalue
## [1] 0.1224
##
## $devst
## [1] 0.006554944
##
## $sim
## [1] 2500
```

```
n <- dim(extra)[1]
p <- dim(extra)[2]

x.mean <- sapply(extra,mean)
x.cov <- cov(extra)
x.inv <- solve(x.cov)

cfr.fisher <- (n-1)*p/(n-p)*qf(1-alpha,p,n-p)

ellipse(center=x.mean, shape=x.cov/n, radius=sqrt(cfr.fisher), lwd=2)
```




```
# centre:
x.mean
```

```
##     primo      vice
## 121165.42  52060.33
```

```
# direction of the axes
eigen(x.cov)$vector[,1]
```

```
## [1] -0.7460693  0.6658683
```

```
eigen(x.cov)$vector[,2]
```

```
## [1] -0.6658683 -0.7460693
```

```
# radius
sqrt(cfr.fisher)
```

```
## [1] 3.004365
```

```
# lenght of the semi-axes
sqrt(eigen(x.cov/n)$values)*sqrt(cfr.fisher)
```

```
## [1] 8754.029 5118.409
```

```
### question b)
# Test:   H0: mu1<=2*mu2   vs H1: mu1>2*mu2
# i.e. H0: mu1-2*mu2<=0 vs H1: mu1-2*mu2>0
# i.e H0: a'mu<=0      vs H1: a'mu>0 con a=c(1,-2)

a <- c(1,-2)
delta.0 <- 0

extra <- as.matrix(extra)
t.stat <- (mean(extra %*% a) - delta.0 ) / sqrt( var(extra %*% a) / n ) # t-statistics (statistica 1!)

# Reject for large values of t
# => compute the p-value as the probability of the right tail (i.e., of values >tstat)
P <- 1-pt(t.stat, n-1)
P
```

```
##              [,1]
## [1,] 0.009578172
```

```r
### question c)
a2 <- c(1,1)
alpha <- .1
cfr.t <- qt(1-alpha/2, n-1)

c(inf = mean(extra %*% a2) - cfr.t * sqrt( var(extra %*% a2) / n ),
  center = mean(extra %*% a2),
  sup = mean(extra %*% a2) + cfr.t * sqrt( var(extra %*% a2) / n ))
```

```
##      inf   center      sup
## 168885.5 173225.8 177566.0
```

```r
# Otherwise one can use the function t.test()
lc <- extra[,1] + extra[,2]
t.test(lc, alternative = 'two.sided', mu = 0, conf.level = 0.90)
```

```
##
##  One Sample t-test
##
## data:  lc
## t = 71.676, df = 11, p-value = 4.844e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  168885.5 177566.0
## sample estimates:
## mean of x
##  173225.8
```

```r
### ------------------------------------------------------------------------
### Pb 1 of 29/06/11
### ------------------------------------------------------------------------
# The Spanish Authority for the paella measured the content of clams (vongole) [g] and
# shrimps [g] in a number of packs of "Paella of Cantabro" of 200g
# (file cantabro.txt). They aim to verify if the mean content of clams and shrimps
# isn't significantly different from the nominal one: 30g of clams and 50g
# of shrimps.
# a) Perform a test to prove the former hypothesis.
# b) Comment the test at point a) using three simultaneous T2 intervals (global
#    level 90%) for the mean content of clams, of shrimps and of their sum
# c) Do you deem necessary an action of the Authority? Motivate the response.
### ------------------------------------------------------------------------

cantabro <- read.table('cantabro.txt', header=T)
head(cantabro)
```

```
##   vongole gamberetti
## 1  27.269     51.827
## 2  28.324     49.956
## 3  28.795     50.166
## 4  29.813     49.127
## 5  31.284     48.106
## 6  27.840     50.876
```

```r
### question a)
mcshapiro.test(cantabro)
```

```
## $Wmin
## [1] 0.987412
##
## $pvalue
## [1] 0.5868
##
## $devst
## [1] 0.009848162
##
## $sim
## [1] 2500
```

```r
p <- dim(cantabro)[2]
n <- dim(cantabro)[1]

# Test: H0: mu=c(30,50) vs H1: mu!=c(30,50)
alpha    <- 0.01
mu0      <- c(30,50)
x.mean   <- colMeans(cantabro)
x.cov    <- cov(cantabro)
x.invcov <- solve(x.cov)

x.T2       <- n * (x.mean-mu0) %*% x.invcov %*% (x.mean-mu0)
cfr.fisher <- ((n-1)*p/(n-p))*qf(1-alpha,p,n-p)
x.T2 < cfr.fisher
```

```
##       [,1]
## [1,] FALSE
```

```r
P <- 1-pf(x.T2*(n-p)/(p*(n-1)), p, n-p)
P
```

```
##      [,1]
## [1,]    0
```

```r
### question b)
a1<-c(1,0)
a2<-c(0,1)
a3<-c(1,1)

ICT21<-data.frame(L=t(a1)%*%x.mean-sqrt(t(a1)%*%x.cov%*%a1/n)*sqrt(cfr.fisher),C=t(a1)%*%x.mean,U=t(a1)%*%x.mean+sqrt(t(a1)%
*%x.cov%*%a1/n)*sqrt(cfr.fisher))
ICT22<-data.frame(L=t(a2)%*%x.mean-sqrt(t(a2)%*%x.cov%*%a2/n)*sqrt(cfr.fisher),C=t(a2)%*%x.mean,U=t(a2)%*%x.mean+sqrt(t(a2)%
*%x.cov%*%a2/n)*sqrt(cfr.fisher))
ICT23<-data.frame(L=t(a3)%*%x.mean-sqrt(t(a3)%*%x.cov%*%a3/n)*sqrt(cfr.fisher),C=t(a3)%*%x.mean,U=t(a3)%*%x.mean+sqrt(t(a3)%
*%x.cov%*%a3/n)*sqrt(cfr.fisher))
ICT2<-data.frame(rbind(ICT21,ICT22,ICT23))
ICT2
```

```
##          L        C        U
## 1 29.03980 29.45348 29.86716
## 2 49.13399 49.52972 49.92545
## 3 78.84331 78.98320 79.12309
```

```
### question c)
for (i in 1:3)
  print(paste('Reject H0 for a',i,': ', !(0>ICT2[i,1] & 0<ICT2[i,3]),sep=''))
```

```
## [1] "Reject H0 for a1: TRUE"
## [1] "Reject H0 for a2: TRUE"
## [1] "Reject H0 for a3: TRUE"
```

```
### ----------------------------------------------------------------------------
### Pb 2 of 18/07/11
### ----------------------------------------------------------------------------
# The file uranium.txt reports the quotations of uranium [$/kg] in 32
# stock exchange, for the days between 1st and 10th July 2011.
# Having frames the problem in the context of repeated measures,
# answer the following questions.
# a) Is there evidence at 90% that the mean price was not constant
#    along the 10 days?
# b) Build 9 Bonferroni confidence intervals (global confidence 90%) for the
#    increments/decrements of the mean daily prices.
# c) Comment the conclusions deduced from the analyses (a) and (b)
# d) The IMF states that between the 8th and 9th July, the prices had a
#    mean decrease of 1$/kg. Perform a test to confirm/deny this statement.
### ----------------------------------------------------------------------------

uranium <- read.table('uranium.txt', header=T)
head(uranium)
```

```
##    day1  day2  day3  day4  day5  day6  day7  day8  day9 day10
## 1 8.221 7.701 7.790 7.138 7.467 7.301 7.589 7.726 6.518 6.462
## 2 8.378 7.176 7.562 7.382 7.340 7.773 7.625 7.993 6.176 6.217
## 3 6.385 6.463 7.220 5.848 6.168 6.379 6.911 6.455 5.040 5.638
## 4 6.016 5.834 5.549 6.824 6.348 6.001 6.140 6.083 4.596 5.246
## 5 3.977 3.861 4.511 4.396 4.499 4.572 4.249 4.777 3.643 3.024
## 6 6.998 7.239 7.715 6.740 7.197 7.175 6.900 7.420 6.555 6.601
```

```
### question a) (with a look to b)!)
C <- matrix (0, nrow=9, ncol=10)
for(i in 1:9)
  C[i,c(i,i+1)]<-c(-1,1)

mcshapiro.test(uranium)
```

```
## $Wmin
## [1] 0.7841037
##
## $pvalue
## [1] 0.3504
##
## $devst
## [1] 0.009541904
##
## $sim
## [1] 2500
```

```
n <- dim(uranium)[1]
q <- dim(uranium)[2]

alpha    <- 0.10
delta.0      <- rep(0,9)
x.mean   <- colMeans(uranium)
x.cov    <- cov(uranium)
x.invcov <- solve(x.cov)

Md <- C %*% x.mean
Sd <- C %*% x.cov %*% t(C)
Sdinv <- solve(Sd)

T2 <- n * t( Md - delta.0 ) %*% Sdinv %*% ( Md - delta.0 )

cfr.fisher <- ((q-1)*(n-1)/(n-(q-1)))*qf(1-alpha,(q-1),n-(q-1))
T2 < cfr.fisher
```

```
##       [,1]
## [1,] FALSE
```

```
P <- 1-pf(T2*(n-(q-1))/((q-1)*(n-1)),(q-1),n-(q-1))
P
```

```
##              [,1]
## [1,] 5.991532e-10
```

```
### question b)
alpha <- 0.10
k<-q-1
cfr.t <- qt(1-alpha/(2*k),n-1)

ICB <- cbind(L=Md-cfr.t*sqrt(diag(Sd)/n),C=Md,U=Md+cfr.t*sqrt(diag(Sd)/n))
ICB
```

```
##              [,1]        [,2]       [,3]
##  [1,] -0.29097336  0.04765625  0.3862859
##  [2,] -0.29167278 -0.01793750  0.2557978
##  [3,] -0.27815287  0.04365625  0.3654654
##  [4,] -0.20039582  0.01700000  0.2343958
##  [5,] -0.38806782 -0.10337500  0.1813178
##  [6,] -0.37097392 -0.09012500  0.1907239
##  [7,] -0.06437765  0.22303125  0.5104401
##  [8,] -1.35138472 -1.11825000 -0.8851153
##  [9,] -0.18104255  0.11153125  0.4041051
```

```
### question c)
for (i in 1:k)
  print(paste('Reject H0 for the increment ',i,': ', !(0>ICB[i,1] & 0<ICB[i,3]),sep=''))
```

```
## [1] "Reject H0 for the increment 1: FALSE"
## [1] "Reject H0 for the increment 2: FALSE"
## [1] "Reject H0 for the increment 3: FALSE"
## [1] "Reject H0 for the increment 4: FALSE"
## [1] "Reject H0 for the increment 5: FALSE"
## [1] "Reject H0 for the increment 6: FALSE"
## [1] "Reject H0 for the increment 7: FALSE"
## [1] "Reject H0 for the increment 8: TRUE"
## [1] "Reject H0 for the increment 9: FALSE"
```

```
### question d)
a <- c(0,0,0,0,0,0,0,-1,1,0)  # combination that gives the increment between day9 and day8
a
```

```
##  [1]  0  0  0  0  0  0  0 -1  1  0
```

```
# Test: a'mu==-1 vs a'mu!=-1  (decrease of 1 == increment of -1)
alpha <- 0.10
delta.0 <- -1

uranium <- as.matrix(uranium)
t2 <- (mean(uranium %*% a) - delta.0 )^2 / ( var(uranium %*% a) / n )
cfr.fisher <- qf(1-alpha,1,n-1)
t2<cfr.fisher
```

```
##      [,1]
## [1,] TRUE
```

```
P <- 1-pf(t2,1,n-1)
p
```

```
##           [,1]
## [1,] 0.1805744
```

```
## N.B. this test is univariate!
##      it is totally equivalent to:
t.test(uranium%*%a, alternative = 'two.sided', mu = delta.0, conf.level = 1-alpha)
```

```
##
##  One Sample t-test
##
## data:  uranium %*% a
## t = -1.3699, df = 31, p-value = 0.1806
## alternative hypothesis: true mean is not equal to -1
## 90 percent confidence interval:
##  -1.2646127 -0.9718873
## sample estimates:
## mean of x
##  -1.11825
```

```
t2
```

```
##          [,1]
## [1,] 1.876493
```

```
(-1.3699)^2
```

```
## [1] 1.876626
```

```
### ------------------------------------------------------------------------------
### Pb 1 of 4/07/2004
### ------------------------------------------------------------------------------
# A lab is conducing a clinical trial to test the effectiveness of a new drug
# for diabetes. To each patient, the following quantities are measured 2h before
# and 2h after the administration of the drug: glycemia, body temperature,
# min and max blood pressure. The data are collected in the file diabete.txt.
# The pharmaceutical company that produces the drug declares that the drug
# is able to decrease of 60 units the glycemia, without any side effect on
# the body temperature, min and max pressure.
# a) perform a test to verify the statement of the producer.
# b) Using Bonferroni intervals of global confidence 95%, analyse the results
#    of the test at point (a).
# c) Verify the assumptions needed to execute the test at point (a).
### ------------------------------------------------------------------------------

diabetes <- read.table('diabetes.txt', header=TRUE)
head(diabetes)
```

```
##            glicemia_prima temperatura_prima pressione_min_prima
## paziente 1         206.69             35.76                81.84
## paziente 2         186.13             35.97                95.62
## paziente 3         177.42             35.98                90.98
## paziente 4         186.35             35.71                97.02
## paziente 5         186.91             36.35                98.65
## paziente 6         187.28             35.75                92.40
##            pressione_max_prima glicemia_dopo temperatura_dopo
## paziente 1              116.85        138.49            37.03
## paziente 2              112.30        119.73            37.25
## paziente 3              117.43        118.29            37.37
## paziente 4              120.29        100.65            36.99
## paziente 5              123.96        105.83            36.92
## paziente 6              125.16        134.52            36.94
##            pressione_min_dopo pressione_max_dopo
## paziente 1              94.95             115.01
## paziente 2              87.66             125.34
## paziente 3              92.88             128.61
## paziente 4              84.35             122.55
## paziente 5              83.80             110.46
## paziente 6              89.80             119.39
```

```r
# question a)
D <- diabetes[,5:8]-diabetes[,1:4]
names(D) <- c('glycemia_diff', 'temperature_diff',
              'pressure_min_diff', 'pressure_max_diff')

# Test: H0: mu_D=c(-60,0,0,0) vs H1: mu_D!=0
n <- dim(D)[1]
p <- dim(D)[2]
M <- sapply(D,mean)
S <- cov(D)
Sinv <- solve(S)

delta0 <- c(-60,0,0,0)

T2 <- n*t(M-delta0)%*%Sinv%*%(M-delta0)

pvalue <- 1-pf(T2*(n-p)/(p*(n-1)), p, n-p)
pvalue
```

```
##              [,1]
## [1,] 4.211076e-13
```

```r
# question b)
alpha <- .05
k <- 4

cfr.t <- qt(1-alpha/(2*k),n-1)

ICB<-data.frame(L=M-sqrt(diag(S)/n)*cfr.t, C=M, U=M+sqrt(diag(S)/n)*cfr.t)
ICB
```

```
##                              L           C          U
## glycemia_diff     -69.9792558 -63.1380000 -56.296744
## temperature_diff    0.8937209   1.0636667   1.233612
## pressure_min_diff  -4.1569526  -0.3046667   3.547619
## pressure_max_diff  -2.8587865   0.2160000   3.290786
```

```r
# question c)
mcshapiro.test(D)
```

```
## $Wmin
## [1] 0.9464517
##
## $pvalue
## [1] 0.7364
##
## $devst
## [1] 0.008811698
##
## $sim
## [1] 2500
```

```r
### ------------------------------------------------------------------------
### Pb 1 of 26/02/2008
### ------------------------------------------------------------------------
# Let X=(X1 X2 X3)'~N(mu, Sigma) a Gaussian random vector with
# mu=(1 1 1)' and Sigma=cbind(c(5,3,1),c(3,5,1),c(1,1,1)).
# a) Identify a region A such that P((X1 X2)' \in A)=0.9
# b) Identify a region A2 such that P((X1 X2)' \in A2 | X3=1)=0.9
# c) Having reported in a plot the graphs of the two regions, order in increasing
#    order the following probabilities:
#    P((X1 X2)' \in A )
#    P((X1 X2)' \in A2)
#    P((X1 X2)' \in A  | X3=1)
#    P((X1 X2)' \in A2 | X3=1)
### ------------------------------------------------------------------------

mu=c(1,1,1)
Sigma=cbind(c(5,3,1),c(3,5,1),c(1,1,1))

### a) Consider only (X1 X2)'
eigen(Sigma[1:2,1:2])
```

```
## eigen() decomposition
## $values
## [1] 8 2
##
## $vectors
##            [,1]       [,2]
## [1,] 0.7071068 -0.7071068
## [2,] 0.7071068  0.7071068
```

```r
# Direction of the axes:
eigen(Sigma[1:2,1:2])$vectors
```

```
##            [,1]       [,2]
## [1,] 0.7071068 -0.7071068
## [2,] 0.7071068  0.7071068
```

```r
# Center:
M <- mu[1:2]

# Radius of the ellipse:
r <- sqrt(qchisq(0.9,2))

# Length of the semi-axes:
r*sqrt(eigen(Sigma[1:2,1:2])$values)
```

```
## [1] 6.069709 3.034854
```

```
# Plot
plot(M[1],M[2],xlim=c(-10,15),ylim=c(-10,15),col='blue',pch=19,xlab='X.1',ylab='X.2',asp=1)
ellipse(center=M, shape=cbind(Sigma[1:2,1:2]), radius=r, col = 'blue')
abline(h=0, v=0, lty=2, col='grey')
abline(a=0,b=1,col='grey',lty=2)
abline(a=2,b=-1,col='grey',lty=2)

### b) Consider the conditional distribution (X1 X2)'|X3=1

# Functions to compute the mean and the covariance matrix of the conditional
# distribution
mu.cond <- function(mu1,mu2,Sig11,Sig12,Sig22,x2)
{
  return(mu1+Sig12%*%solve(Sig22)%*%(x2-mu2))
}

Sig.cond <- function(Sig11,Sig12,Sig22)
{
  Sig21=t(Sig12)
  return(Sig11-Sig12%*%solve(Sig22)%*%Sig21)
}

M.c <- mu.cond(mu1=mu[1:2],mu2=mu[3],Sig11=Sigma[1:2,1:2],Sig12=Sigma[1:2,3],Sig22=Sigma[3,3],x2=1)
M.c
```

```
##      [,1]
## [1,]    1
## [2,]    1
```

```
Sigma.c <- Sig.cond(Sig11=Sigma[1:2,1:2],Sig12=Sigma[1:2,3],Sig22=Sigma[3,3])
Sigma.c
```

```
##      [,1] [,2]
## [1,]    4    2
## [2,]    2    4
```

```
eigen(Sigma.c[1:2,1:2])
```

```
## eigen() decomposition
## $values
## [1] 6 2
##
## $vectors
##           [,1]       [,2]
## [1,] 0.7071068 -0.7071068
## [2,] 0.7071068  0.7071068
```

```
# Direction of the axes:
eigen(Sigma.c[1:2,1:2])$vectors
```

```
##           [,1]       [,2]
## [1,] 0.7071068 -0.7071068
## [2,] 0.7071068  0.7071068
```

```
# Center:
M.c
```

```
##      [,1]
## [1,]    1
## [2,]    1
```

```
# Radius of the ellipse:
r <- sqrt(qchisq(0.9,2))

# Length of the semi-axes:
r*sqrt(eigen(Sigma.c[1:2,1:2])$values)
```

```
## [1] 5.256522 3.034854
```

```
# Plot
ellipse(center=as.vector(M.c), shape=Sigma.c, radius=r, col = 'red')
```