Análisis multivariado. Tarea 4

José Antonio García Ramirez

Marzo 12, 2018

Ejercicio 1

El departamento de control de calidad de un fabricante de hornos de microondas es requerido por el gobierno federal para monitorear la cantidad de radiación emitida por los hornos que fabrican. Se realizaron mediciones de la radiación emitida por 42 hornos seleccionados al azar con las puertas cerradas y abiertas. Los datos están en el archivo *datosradiacion*

a) Construye un elipse de confianza del 95% para μ, considerando la transformación de las variables:

$$x_1 = \sqrt[4]{radiacion\_con\_puerta\_cerrada}$$

$$x_2 = \sqrt[4]{radiacion\_con\_puerta\_abierta}$$

Primero exporte los datos a formato .csv para que el código sea más portable y no dependa de la configuración de Java de la máquina, que involucra leer con archivos .xlsx con el package xlsx.

Luego leemos los datos y realizamos la transformación sugerida sobre las variables posteriormente realizamos los test de Shapiro-Wilks para descartar la no normalidad univariada de cada variable (aunque se cuenta con 42 observaciones y podríamos emplear los resultados asintóticos para las hipótesis de media y el TLC para el caso multivariado).

```
setwd('C:\\Users\\fou-f\\Desktop\\MCE\\Second\\EstadisticaMultivariada\\Tarea4')
radiacion <- read.csv('datos_radiacion.csv')
colnames(radiacion) <- c('Puerta.cerrada', 'Puerta.abierta') #cambio de nombres de columnas
                                                              #para agilizar el tipeo de codigo
radiacion.sqrt <-apply(radiacion, 2, function(x){ x**(1/4) }) #realizo la transformación
radiacion.sqrt <- as.data.frame(radiacion.sqrt)
shapiro.test(radiacion.sqrt$Puerta.cerrada)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  radiacion.sqrt$Puerta.cerrada
## W = 0.96481, p-value = 0.2188
```

```
shapiro.test(radiacion.sqrt$Puerta.abierta)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  radiacion.sqrt$Puerta.abierta
## W = 0.94282, p-value = 0.03589
```

Por los resultados de la prueba podemos ver que la variable transformada que mide el nivel de radiación con la puerta cerrada pasa la prueba de normalidad, sin embargo la otra variable tiene un p-value menor a 0.05, sin embargo al realizar un test de Kolmogorov sobre la variable transformada que mide el nivel de radiación con la puerta abierta sí se tiene evidencia para no descartar su normalidad.

```r
ks.test(radiacion.sqrt$Puerta.abierta, "pnorm", mean(radiacion.sqrt$Puerta.abierta),
        sd(radiacion.sqrt$Puerta.abierta ))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  radiacion.sqrt$Puerta.abierta
## D = 0.19043, p-value = 0.09508
## alternative hypothesis: two-sided
```

Posteriormente calculamos el vector de medias y la matriz de covarianzas, esta última para invertirla y dar una expresión del elipsoide de confianza.

```r
alpha <- 1-.95
p <- dim(radiacion.sqrt)[2]
n <- dim(radiacion.sqrt)[1]
      #se calcula la media y la matriz de covarianza
media <- apply(radiacion.sqrt, 2, mean)
(media <- matrix(media, ncol = 1))
```

```
##           [,1]
## [1,] 0.5642575
## [2,] 0.6029812
```

```r
S <- cov(radiacion.sqrt)
(solve(S))
```

```
##                Puerta.cerrada Puerta.abierta
## Puerta.cerrada       203.4981      -163.9069
## Puerta.abierta      -163.9069       200.7691
```

Sabemos que el elipsoide de confianza esta dado por:

$$n(\bar{x}-\mu)^t S^{-1}(\bar{x}-\mu) \leq \frac{p(n-1)}{n-p} F_{p,n-p}(\alpha)$$

Y en nuestro caso queda (redondeando las operaciones a tres decimales):

$$\begin{aligned}
42(0.564-\mu_1, 0.6029-\mu_2)\begin{pmatrix} 203.018 & -163.906 \\ -163.906 & 200.77 \end{pmatrix}\begin{pmatrix} 0.564-\mu_1 \\ 0.6029-\mu_2 \end{pmatrix} &\leq \frac{2*41}{40} * 3.232 \\
\Leftrightarrow 42(0.564-\mu_1, 0.6029-\mu_2)\begin{pmatrix} 15.667-203.018x_1+163.906x_2 \\ 22.229+163.96x_1-200.77x_2 \end{pmatrix} &\leq 6.625 \\
\Leftrightarrow 42(22.24-31.334x_1-50.85x_2-327.812x_1x_2+200.77x_2^2+203.018x_1^2) &\leq .158
\end{aligned}$$

Entonces la elipse de confianza al 95% está dada por la forma cuadrática:

$$203.018x_1^2 - 31.334x_1 - 327.812x_1x_2 - 50.85x_2 + 200.77x_2^2 + 22.083 = 0 \tag{1}$$

b) Prueba si $\mu = (.562, .589)$ está en la region de confianza.

Calculamos el estadistico basado en la muestra

```r
mu <- matrix(c(.562,.589), byrow = TRUE, ncol = 1)
(nivel <- qf(1-alpha, df1=p, df2=n-p)*(p*(n-1)/(n-p)))
```

```
## [1] 6.62504
```

```r
(T.hotelling <- n*t(media-mu)%*%solve(S)%*%(media-mu))
```

```
##          [,1]
## [1,] 1.2573
```

Como el estadistico calculado $T^2 = 1.2573005$ es menor al cuantil teórico 6.6250403, aceptamos la hipótesis nula de la prueba, por lo que es factible que la media poblacional sea 0.562, 0.589.

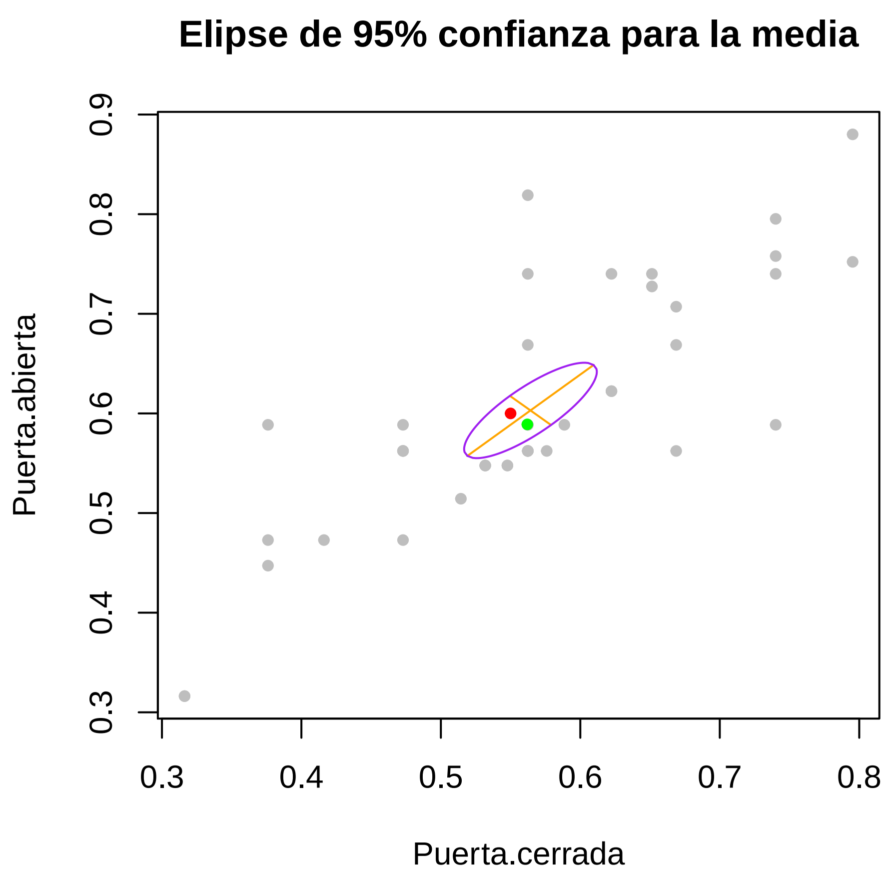
c) Calcula los valores y vectores propios de S y obten la gráfica del elipsoide de confianza.

Defino una función que construye la elipse y dibuja los datos y la utilizo, posteriormente dibujo el punto propuesto en el inciso anterior con color verde y la media propuesta en el inciso siguiente en color rojo.

Para lo anterior, en lugar de despejar de (1) una variable en función de la otra, lo cual es posible rotando la cónica con una matriz pertinente que resulta ser la de los vectores propios de la matriz de covarianzas, el procedimiento con el que grafico es generar los ejes y la elipse considerando su forma canónica (con centro en el origen y sin rotación) posteriormente roto usando la matriz de vectores propios (la ortogonalidad de los ejes se preserva bajo rotaciones, además es un resultado que los vectores propios que definen los ejes de la elipse, son ortogonales) y por último translado los puntos con el vector de media muestral.

```r
dibuja.elipse <- function(alpha, n, p, media, S, xlim, ylim)
{
  #alpha (numeric): nivel de signiificancia
  #n (numeric): numero de observaciones
  #p (numeric): numero de variables
  #media (vector): vector-columna de medias
  #S (matrix): matriz de covarianzas
  ejes <- eigen(S)
  eje_x <- ejes$vectors[,1]
  eje_y <- ejes$vectors[,2]
  nivel <- qf(1-alpha, df1 = p, df2 = n-p ) #cuantil de la distribucion f con confianza 1-alpha
  c <- ((p*(n-1))/(n*(n-p)) )*nivel #cuantil de la T de Hotelling
  longitud_x <- ejes$values[1]**.5*( c)**.5 #formulas vistas en clase
  longitud_y <- ejes$values[2]**.5*( c)**.5 #formulas vistas en clase
      #se dibujan los ejes
  g <- 100 #numero de puntos a dibujar, primero los genero centrados en el origen y
          #luego roto y translado
  x <- seq(-longitud_x, longitud_x, length=g)
  y <- seq(-longitud_y, longitud_y, length=g)
  ejes.puntos <- matrix(c(x, rep(0,g )), byrow = TRUE, nrow = 2)
  ejes.puntos <- (ejes$vectors)%*%ejes.puntos + matrix(rep(media, g), nrow = 2)
  plot(ejes.puntos[1,], ejes.puntos[2,], type='l', col='orange',
       main='Elipse de 95% confianza para la media',
       xlab=colnames(S)[1], ylab = colnames(S)[2], xlim=xlim, ylim=ylim)
  ejes.puntos <- matrix(c(rep(0,g ), y), byrow = TRUE, nrow = 2)
  ejes.puntos <- (ejes$vectors)%*%ejes.puntos + matrix(rep(media, g), nrow = 2)
  points(ejes.puntos[1,], ejes.puntos[2,], type='l', col='orange')
      #se genera el perimetro, primero centrado y luego se rota y translada
  y <- ((c-x**2/ejes$values[1])*ejes$values[2])**.5 #funcion inversa de elipse con
                                                    #centro en elorigen
  elipse.puntos <- matrix(c(x,y), ncol = g, byrow = TRUE)
  elipse.puntos <- ejes$vectors%*%(elipse.puntos) + matrix(rep(media, g), nrow = 2)
  points(elipse.puntos[1,], elipse.puntos[2,], col='purple', type='l')
  y <- -((c-x**2/ejes$values[1])*ejes$values[2])**.5
```

```
  elipse.puntos <- matrix(c(x,y), ncol = g, byrow = TRUE)
  elipse.puntos <- ejes$vectors%*%(elipse.puntos) + matrix(rep(media, g), nrow = 2)
  points(elipse.puntos[1,], elipse.puntos[2,], col='purple', type='l')
}
dibuja.elipse(S=cov(radiacion.sqrt), alpha=alpha, n = dim(radiacion.sqrt)[1],
              p=dim(radiacion.sqrt)[2],
              media=matrix(apply(radiacion.sqrt,2, mean), ncol = 1),
              xlim=range(radiacion.sqrt[,1]), ylim=range(radiacion.sqrt[,2]))
points(radiacion.sqrt[,1], radiacion.sqrt[,2], col='gray', pch=20)
points(c(mu[1],.55),c(mu[2],.6), col=c('green','red'), pch=20)
```



d) Realiza una prueba para la hipótesis $H_0 : \mu' = (.55, .60)$ en un nivel de significancia de $\alpha = .05$. Es consistente el resultado con la gráfica de la elipse de confianza del 95% para μ obtenida en el inciso anterior? Explica.

Realizamos la prueba para μ', calculamos el estadistico:

```
mu <- matrix(c(.55,.6), byrow = TRUE, ncol = 1)
(nivel <- qf(1-alpha, df1=p, df2=n-p)*(p*(n-1)/(n-p)))
```

```
## [1] 6.62504
```

```
(T.hotelling <- n*t(media-mu)%*%solve(S)%*%(media-mu))
```

```
##          [,1]
## [1,] 1.227116
```

Al igual que en el inciso anterior el estadistico calculado $T^2 = 1.2271163$ es menor al cuantil teórico 6.6250403, aceptamos la hipótesis nula de la prueba, por lo que es factible que la media poblacional sea 0.55, 0.6.

Éste resultado es consistente con la gráfica anterior pues este nuevo punto de media propuesto también esta dentro de la región de confianza, de hecho cualquier punto dentro de la región es un candidato plausible de ser la media poblacional.

Ejercicio 2

Sabemos que T^2 es igual al t-valor cuadrado univariado más grande construido a partir de la combinación lineal $a^t x_j$ con $a = S^{-1}(\bar{x} - \mu_0)$ (ver notas de la semana6 2).

a) Usando los resultados del ejercicio anterior y la misma hipotesis nula H_0 del inciso (d), evalua a para los datos transformados de radiaciones de los hornos.

Evaluamos a:

```
mu <- matrix(c(.55,.6), byrow = TRUE, ncol = 1)
(a <- solve(S)%*%(media-mu))
```

```
##                    [,1]
## Puerta.cerrada  2.412731
## Puerta.abierta -1.738364
```

b) Verifica que el valor t^2 calculado con esta a es igual a la T^2 del ejercicio anterior.

Entonces estamos interesados en la media de la combinación lineal dada por:

$$z = a^t x = a_1 x_1 + a_2 x_2 = 2.412731 x_1 - 1.738364 x_2$$

Es decir que nuestra hipótesis nula $H_0 : \mu_x = (.55, .6)$ es quivalente a $H_0 : \mu_z = a^t \mu_x = a_1 \mu_{x_1} + a_2 \mu_{x_2} = 2.412731 \mu_{x_1} - 1.738364 \mu_{x_2} = (2.412731)(.55) - (1.738364)(.6) = 0.2839837$

Con una significancia de $\alpha = 0.05$, calculamos t^2 para $H_0 : \mu_z = a^t \mu_x = 0.2839837$, calculamos el estadístico $t^2 = (\frac{\sqrt{n}(\bar{z} - \mu_z)}{S_z})^2 = n\frac{(\bar{z} - \mu_z)^2}{S_z^2}$, donde sabemos que $S_z = a^t S_x a$ pero no es nesesario evaluarla de esa manera (por ello en el codígo siguiente la calculo como la varianza de las observaciones z_i).

```
z <- t(a)%*%t(as.matrix(radiacion.sqrt))
media.z <- mean(z)
(t2 <- n*(media.z-t(a)%*%mu)^2/var(t(z)))
```

```
##          [,1]
## [1,] 1.227116
```

Y sí efectivamente el estadístico T^2 que vale 1.2271163 es identico al estadístico t^2 que vale 1.2271163. Es importante destacar que ésta igualdad solo se da por la construcción de a que se da en el ejercicio y que justamente esta construcción de a es la que maximiza al estadístico t^2 igualándolo al estadistico T^2

Ejercicio 3

Los datos en el archivo **datososos** representan las longitudes en centímetros de siete osos hembras a los 2, 3, 4 y 5 años de edad.

a) Obtener los intervalos de confianza simultaneos T^2 del 95% para las cuatro medias poblacionales de la longitud por año.

Al igual que en el ejercicio 1 exporte los datos a un formato .csv por las mismas razones, portabilidad y agilidad en el código.

Como se esta volviendo repetitivo (y tal vez me sirva en el examen) construyo una función que evalue el estadistico T^2 sobre una muestra, otra que lo calcule teoricamente para contrastarlo con el muestral y otra que estime los T^2-intervalos simultaneos.

```
T.hotelling <- function(data, alpha)
{
  #data (data.frame): Conjunto de datos cuyas observaciones son una m.a con distribucion
  #                   normal multivariada
  #alpha (numeric): Nivel de significancia requerido
  n <- dim(data)[1]
  p <- dim(data)[2]
  T.hotelling.escalar <- ((p*(n-1)) / (n-p))*qf(1-alpha, df1=p, df2=n-p)
  return(T.hotelling.escalar)
}
T.hotelling.muestral <- function(data, mu)
{
  #data (data.frame): Conjunto de datos cuyas observaciones son una m.a con distribucion
  #                   normal multivariada
  # mu (vector numeric): vector columna de dimension p con la media a testear
  n <- dim(data)[1]
  p <- dim(data)[2]
  medias <- matrix(apply(data, 2, mean), byrow = TRUE, ncol = 1)
  S <- cov(data)
  T.muestral <- n*t(medias-mu)%*%solve(S)%*%(medias-mu)
  return(T.muestral)
}
T2.intervalos <- function(S,media, alpha, a, n, p)
{
  #alpha (numeric): nivel de signiificancia
  #n (numeric): numero de observaciones
  #p (numeric): numero de variables
  #media (vector): vector-columna de medias
  #S (matrix): matriz de covarianzas
  nivel <- sqrt(((p*(n-1)/(n*(n-p))))*(qf(1-alpha, df1 = p, df2=n-p)))
  errores <- nivel*sqrt(t(a)%*%S%*%a)
  #errores <- nivel*sqrt(ss) #es importante el orden
  inf.intervalos <- media - errores
  sup.intervalos <- media + errores
  intervalos <- as.data.frame(cbind(inf.intervalos,media, sup.intervalos))
  colnames(intervalos) <- paste0(c('lim.inferior.T2intervalo.media.al',
                                   'media.muestral',
                                   'lim.superior.T2intervalo.media.al'),
                                 c(as.character(1-alpha), '',as.character(1-alpha)  ))
```

```
  #row.names(intervalos) <- colnames(data)
  return(intervalos)
}
```

Aunque la muestra es pequeña probamos la normalidad univariada de cada variable, con el test de Shapiro-Wilks.

```
osos <- read.csv('datos_osos.csv')
colnames(osos) <- paste0(rep("Longitud", 4), 2:5)
apply(osos, 2, shapiro.test)
```

```
## $Longitud2
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.93317, p-value = 0.5782
##
##
## $Longitud3
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.89901, p-value = 0.325
##
##
## $Longitud4
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.89612, p-value = 0.3081
##
##
## $Longitud5
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.91, p-value = 0.3959
```

Por los resultados del test no rechazamos la hipótesis de que los variables provengan de una normal univariada.

Por lo que asumimos normalidad y podemos usar el estadistico T^2, para calcular los intervalos simultáneos para las medias (a.k.a. T^2-intervalos). Los cuales se presentan en la siguiente tabla:

```
media <- matrix(apply(osos, 2, mean), ncol = 1)
n <- dim(osos)[1]
p <- dim(osos)[2]
a <- matrix(c(1,0,0,0), ncol=1 )
media1 <- t(a)%*%media
intervalos.osos1 <- T2.intervalos(S=cov(osos), media= media1, alpha=0.05,
                                  a = a, n=n, p=p)
a <- matrix(c(0,1,0,0), ncol=1 )
media2 <- t(a)%*%media
```

```r
intervalos.osos2 <- T2.intervalos(S=cov(osos), media= media2, alpha=0.05,
                                  a = a, n=n, p=p)
a <- matrix(c(0,0,1,0), ncol=1 )
media3 <- t(a)%*%media
intervalos.osos3 <- T2.intervalos(S=cov(osos), media= media3, alpha=0.05,
                                  a = a, n=n, p=p)
a <- matrix(c(0,0,0,1), ncol=1 )
media4 <- t(a)%*%media
intervalos.osos4 <- T2.intervalos(S=cov(osos), media= media4, alpha=0.05,
                                  a = a, n=n, p=p)

intervalos.osos <- rbind(intervalos.osos1,intervalos.osos2, intervalos.osos3,
                         intervalos.osos4)
#library(xtable)
#xtable(intervalos.osos)
```

	lim.inferior.T2intervalo.media.al0.95	media.muestral	lim.superior.T2intervalo.media.al0.95
Longitud2	130.69	143.29	155.89
Longitud3	127.02	159.29	191.55
Longitud4	160.31	173.14	185.98
Longitud5	155.37	177.14	198.91

b) Respecto al inciso (a), obtener los intervalos de confianza simultaneos T^2 del 95% para los tres aumentos anuales sucesivos en la longitud media.

Notemos que podemos utilizar los vectores $a_1^* = (0,0,-1,1)$, $a_2^* = (0,-1,1,0)$ y $a_3^* = (-1,1,0,0)$ para obtener los incrementos entre los años 4 a 5, 3 a 4 y de 2 a 3. Reutilizando la función que tengo para los T^2-intervalos.

```r
a <- matrix(c(0,0,-1,1), ncol = 1 )
media1 <- t(a)%*%media
intervalos.incremento.osos.4a5 <- T2.intervalos(S=cov(osos), media= media1, alpha=0.05,
                                  a = a, n=n, p=p)
a <- matrix(c(0,-1,1,0), ncol = 1 )
media2 <- t(a)%*%media
intervalos.incremento.osos.3a4 <- T2.intervalos(S=cov(osos), media= media2, alpha=0.05,
                                  a = a, n=n, p=p)
a <- matrix(c(-1,1,0,0), ncol = 1 )
media3 <- t(a)%*%media
intervalos.incremento.osos.2a3 <- T2.intervalos(S=cov(osos), media= media3, alpha=0.05,
                                  a = a, n=n, p=p)
incrementos <- rbind(intervalos.incremento.osos.2a3, intervalos.incremento.osos.3a4,
                     intervalos.incremento.osos.4a5)
row.names(incrementos) <- c('incremento.osos.2a3', 'incremento.osos.3a4',
                            'incremento.osos.4a5')
#xtable(incrementos)
```

Los intervalos de confianza de 95% para la media de los incrementos anuales sucesivos, son reportados en la siguiente tabla:

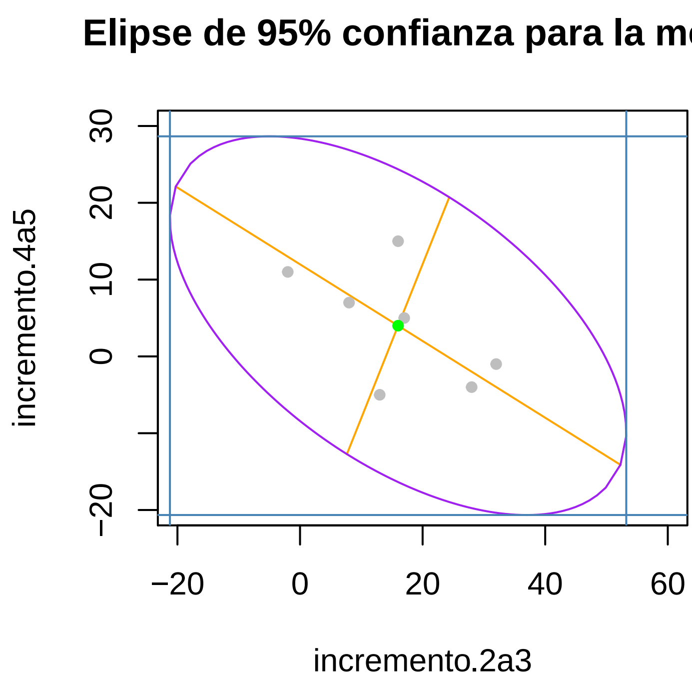
Nótese que los tres intervalos contienen al cero, así que se podría concluir que en promedio los osos no crecen año con año, suponemos que esto se debe al pequeño tamaño de muestra.

	lim.inferior.T2intervalo.media.al0.95	media.muestral	lim.superior.T2intervalo.media.al0.95
incremento.osos.2a3	-21.23	16.00	53.23
incremento.osos.3a4	-22.73	13.86	50.45
incremento.osos.4a5	-20.65	4.00	28.65

c) Obtener la elipse de confianza T^2 del 95% para el aumento medio de la longitud de 2 a 3 años y el aumento medio de la longitud de 4 a 5 años.

Reutilizo la función que implemente en el ejercicio 1 para dibujar las elipses de confianza de medias. La media muestral de los incrementos se muestra en verde en la siguiente gráfica.

```
a <- t(matrix(c(-1,1,0,0, 0,0,-1,1), ncol = 4, byrow = TRUE ))
incrementos.osos <-  as.matrix(osos)%*%a
incrementos.osos <- as.data.frame(incrementos.osos)
colnames(incrementos.osos) <- c('incremento.2a3', 'incremento.4a5')
S_z <- t(a)%*%cov(osos)%*%a
colnames(S_z) <- c('incremento.2a3', 'incremento.4a5')
media.z <- matrix(apply(incrementos.osos, 2, mean))
dibuja.elipse(S=S_z, alpha=0.05, n=dim(osos)[1], p=dim(osos)[2],
              media=media.z, xlim=c(-20,60),
              ylim=c(-20,30))
points(incrementos.osos[,1], incrementos.osos[,2], col='gray', pch=20)
points(media.z[1], media.z[2], col ='green', pch=20)
abline(v = incrementos$lim.inferior.T2intervalo.media.al0.95[1], col ='steelblue' )
abline(v = incrementos$lim.superior.T2intervalo.media.al0.95[1], col ='steelblue' )
abline(h = incrementos$lim.inferior.T2intervalo.media.al0.95[3], col ='steelblue' )
abline(h = incrementos$lim.superior.T2intervalo.media.al0.95[3], col ='steelblue' )
```



Podemos apreciar el gran tamaño de esta elipse ello es consecuencia de la gran varianza en el incremento del año 2 al 3, lo que induce altos valores propios en la matriz de covarianzas, por lo que es natural ver una elipse tan grande (aunado al bajo número de observaciones).

Nótese también que la proyección de la elipse en los ejes cartesianos corresponde a los T^2-intervalos simultáneos para las a_i propuestas.

d) Construir los intervalos de confianza de 95% de Bonferroni para el conjunto formado por las cuatro longitudes medias y los tres aumentos anuales sucesivos en la longitud media, compara los resultados con los obtenidos en (a) y (b).

Implemento una función que estime los intervalos de Bonferroni

```
Bonferroni.intervalos <- function(S,media, alpha, n, p)
{
  #alpha (numeric): nivel de signiificancia conjunto
  #n (numeric): numero de observaciones
  #p (numeric): numero de variables
  #media (vector): vector-columna de medias
  #S (matrix): matriz de covarianzas
  ss <- diag(S)
  nivel <- qt(1-(alpha/(2*p)), df = n-1)
  errores <- nivel*sqrt(ss/n)
  inf.intervalos <- media - errores
  sup.intervalos <- media + errores
  alpha_ <- alpha/(2*p)
  a <- round(1-alpha_,3)
  a <- as.character(a)
  intervalos <- as.data.frame(cbind(inf.intervalos,media, sup.intervalos))
  colnames(intervalos) <- paste0(c('lim.inferior.Bonferri.media.al',
                                   'media.muestral',
                                   'lim.superior.Bonferri.media.al'),
                                 c(a,'',a  ))
  #row.names(intervalos) <- colnames(data)
  return(intervalos)
}
```

Y la utilizo para estimar los intervalos de Bonferri para las 4 longitudes medias

```
medias.osos.len <- matrix(apply(osos, 2, mean))
S <- cov(osos)
n <- dim(osos)[1]
p <- dim(osos)[2]
alpha <- 0.05
intervalos.bonferri.len <- Bonferroni.intervalos(S,medias.osos.len, alpha=alpha,n=n,p=p )
row.names(intervalos.bonferri.len) <- colnames(osos)
#xtable(intervalos.bonferri.len)
```

En la siguiente tabla se reportan los intervalos de Bonferroni para las longitudes medias, notemos en este caso que efectivamente los de Bonferroni son más estrechos.

	lim.inferior.Bonferri.media.al0.994	media.muestral	lim.superior.Bonferri.media.al0.994
Longitud2	138.09	143.29	148.48
Longitud3	145.98	159.29	172.59
Longitud4	167.85	173.14	178.43
Longitud5	168.17	177.14	186.12

Y realizamos lo análogo para los tres aumentos anuales sucesivos en la longitud media

```
a <- t(matrix(c(-1,1,0,0,
                0,-1,1,0,
                0,0,-1,1), ncol = 4, byrow = TRUE ))
incrementos.osos <-  as.matrix(osos)%*%a
medias.osos.incremento <- t(a)%*%matrix(apply(osos, 2, mean))
S <- cov(osos)
S_z <- t(a)%*%S%*%a
n <- dim(incrementos.osos)[1]
m <- dim(incrementos.osos)[2]
alpha <- 0.05
intervalos.bonferri.incrememento <- Bonferroni.intervalos(S=S_z,
                                        media = medias.osos.incremento,
                                          alpha= alpha, n=n, p=m )
row.names(intervalos.bonferri.incrememento) <- c('incremento.2a3', 'incremento.3a4',
                                                 'incremento.4a5')
#xtable(intervalos.bonferri.incrememento)
```

	lim.inferior.Bonferri.media.al0.992	media.muestral	lim.superior.Bonferri.media.al0.992
incremento.2a3	1.67	16.00	30.33
incremento.3a4	-0.23	13.86	27.94
incremento.4a5	-5.49	4.00	13.49