Script:

# Ejercicio 1. Búsqueda y Selección.

#Cargar los datos de “student\_census.csv” del repositorio haciendo uso de read.csv. Para una celebración

#deportiva en el campus, se quiere saber si es posible formar un equipo de 5 estudiantes que jueguen al

#Baloncesto. Para ello se pide localizar a todos los estudiantes que practiquen dicho deporte y de ellos

#seleccionar 5 mediante una selección aleatoria. Para seleccionarlos se implementará una función que genera

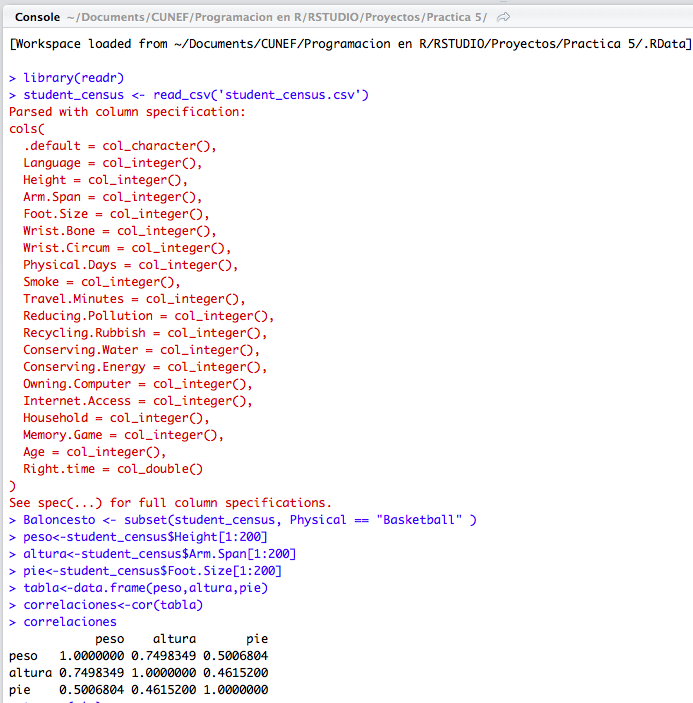
#números aleatorios uno a uno en el rango adecuado, si el estudiante aún no forma parte del equipo se integrará

#y si ya está se generará otro número aleatorio hasta que estén seleccionados los cinco.

library(readr)

student\_census <- read\_csv('student\_census.csv')

Baloncesto <- subset(student\_census, Physical == "Basketball" )



# Ejercicio 2. Ordenación y Regresión.

#Con los datos de “student\_census.csv” del ejercicio anterior, se pide filtrar en un data.frame las variables Height,

#Arm.Span, Foot.Size y realizar correlaciones dos a dos. Ordenar los datos de acuerdo con la estatura para

#determinar la procedencia de los alumnos más altos y hacer un diagrama de barras que indique los resultados

#por provincias.

peso<-student\_census$Height[1:200]

altura<-student\_census$Arm.Span[1:200]

pie<-student\_census$Foot.Size[1:200]

tabla<-data.frame(peso,altura,pie)

correlaciones<-cor(tabla)

correlaciones

#creamos tres vectores y un nuevo objeto definido por nuestros tres vectores denominada ii lo ordeamos por la funcion

#order poniendo como primer elemento aquel que queremos que ordene el objeto.

toe<-c(pie)

tall<-c(altura)

height<-c(peso)

ii<-order(tall,height,toe)

rbind(height,tall,toe)[,ii]

cbind(tall,height,toe)[ii, ]

#creamos un histograma de barras con barplot

provincia<-student\_census$Province[1:200]

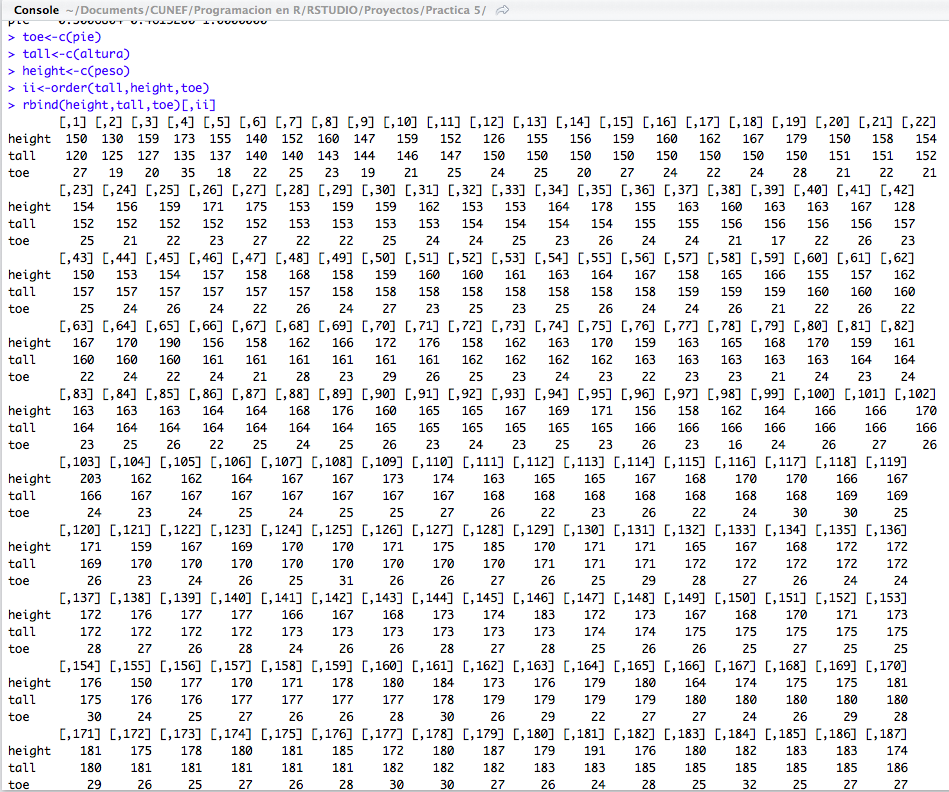
tabla2<-data.frame(peso,altura,pie,provincia)

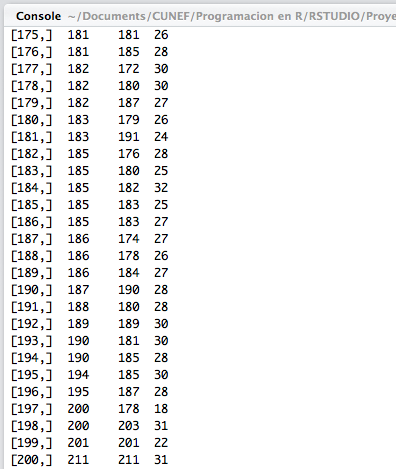
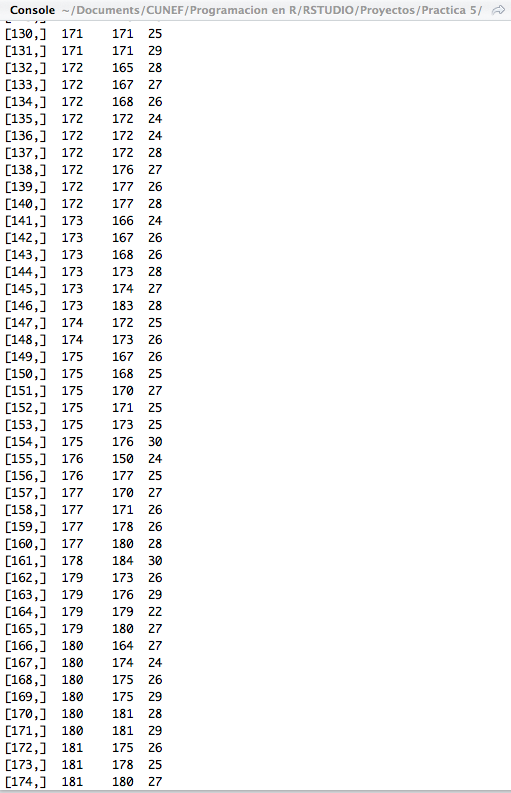
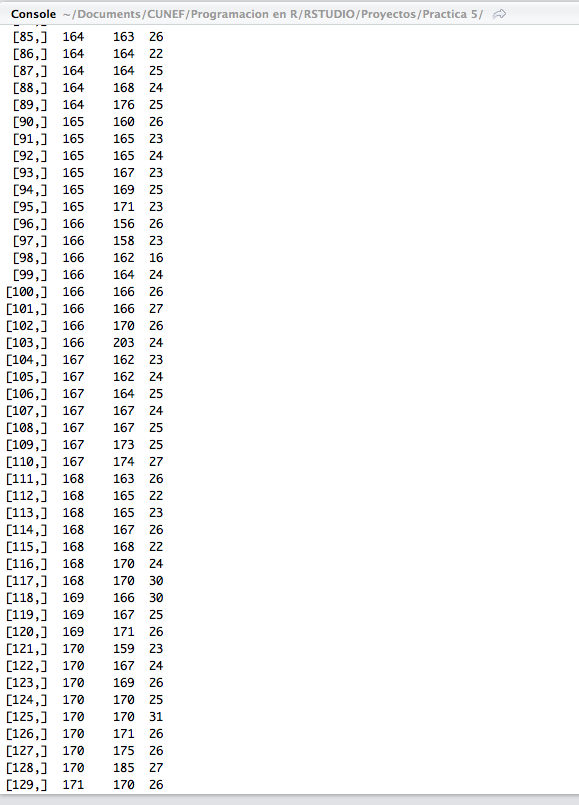
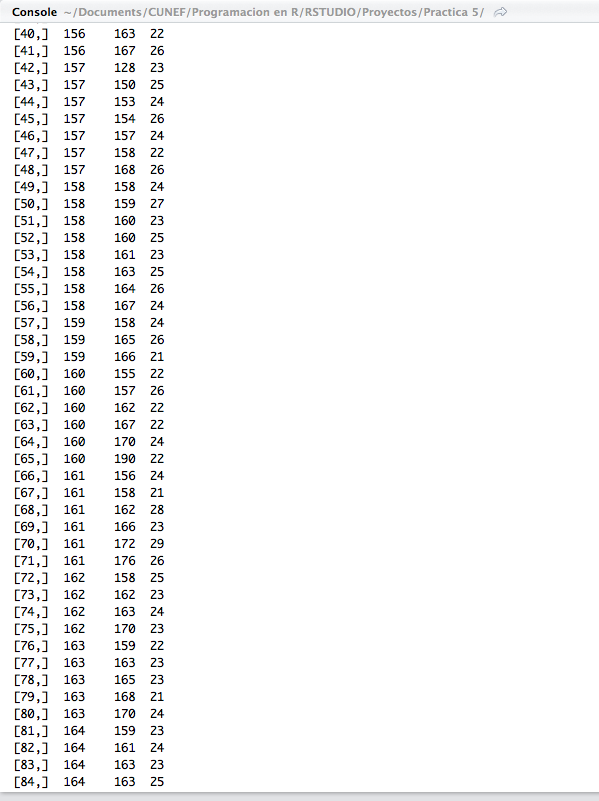
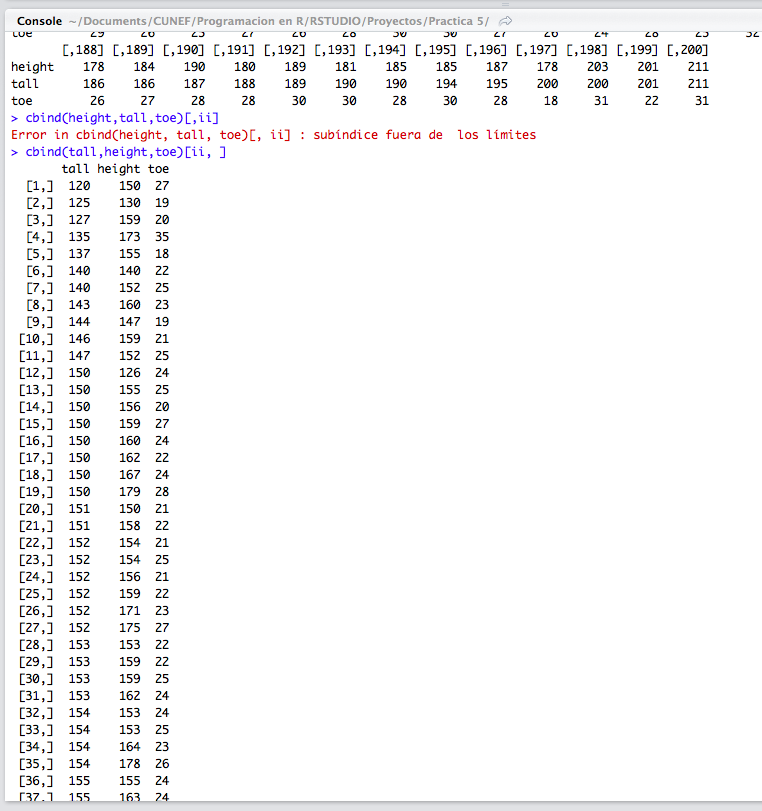
#metodo 1:

barplot(prop.table(table(student\_census$Province)))

#metodo 2:

barplot(prop.table(table(tabla2$provincia)))





## EJERCICIO 3

#se divide el data.frame en 20 listas diferentes aleatoriamente con la funcion sample

l<-split(student\_census, sample(rep(1:20,10,replace=F)))

l[[1]]

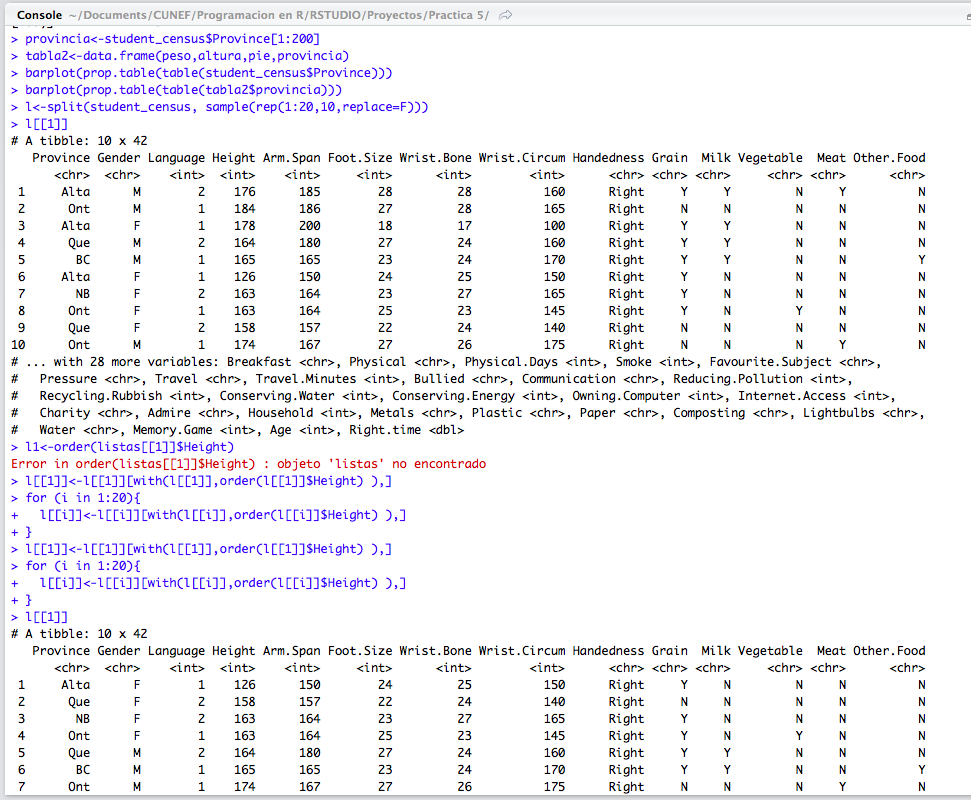
l1<-order(listas[[1]]$Height)

l[[1]]<-l[[1]][with(l[[1]],order(l[[1]]$Height) ),]

for (i in 1:20){

l[[i]]<-l[[i]][with(l[[i]],order(l[[i]]$Height) ),]

}



##EJERCICIO 5

x<- c(1:80, 78:34, 1,2,3,4,5,6,7,5,4,3,2)

n<-5

divideenBloques <- function( x, n ){

tam<-length(x)%/%n

list<-NULL

for (i in 0:(n-2)){

list<-c(list,c(i\*tam+1, (i+1)\*tam))

}

list<-c(list,c(((i+1)\*tam)+1, length(x)))

return(list)

}

lista<-divideenBloques(x,n)

########################

i<-1

pares\_sumas<-function (lista, x, n){

sumas<-NULL

tams<-NULL

for (i in 0:(n-1)){

ini<-lista[i\*2+1]

fin<-lista[i\*2+2]

sumas<-c(sumas, sum(x[ini:fin]))

tams<-c(tams,fin-ini+1)

}

return(data.frame(sumas, tams))

}

ps<-pares\_sumas(lista, x, n)

reduce<-function(ps){

resultado<- as.double(sum(ps$sumas)/sum(ps$tams))

return(resultado)

}

resultado\_media<-reduce(ps)

##EJERCICIO 6

busca <- function (l, h, st){

if (l<=st) {if (sum(l:h) ==st) {print(c(l, h))

return(busca(l+1,l+1,st))}

else {if (sum(l:h)<st) return(busca(l, h+1, st))

else return(busca(l+1, l+1, st))}

}

}

busca(0,100, 100)

