```
# descargar la base de datos

dataset_url <- "http://s3.amazonaws.com/practice_assignment/diet_data.zip"
download.file(dataset_url, "diet_data.zip")
unzip("diet_data.zip", exdir = "diet_data")

#listar los archivos
list.files("diet_data")

#ver que hay adentro
andy <- read.csv("diet_data/Andy.csv")
head(andy)

#saber cuantas ffilas hay
length(andy$Day)


# o puedo verlo como data frame
dim(andy)

#mas comandos para obtener info del data frame
str(andy)
# sumary comando apropiado para obtener distintos datos del data frame

summary(andy)

#nombres de las columnas es names
names(andy)

# una vez que tengo los datos generales voy a realizar las distintas operaciones

#saber el primer peso de Andy
andy[1, "Weight"]

#saber el peso en el dia 30
andy[30, "Weight"]

#enfoque alternativo. puedo usar el comando which

andy[which(andy$Day == 30), "Weight"]

# o puedo extraerlo asi. mas facil la anterior
andy[which(andy[,"Day"] == 30), "Weight"]

# tambien puedo usar la funcion subset
subset(andy$Weight, andy$Day==30)

#Let's assign Andy's starting and ending weight to vectors:

andy_start <- andy[1, "Weight"]
andy_end <- andy[30, "Weight"]

#operaciones con los vectores.
#We can find out how much weight he lost by subtracting the vectors:

andy_loss <- andy_start - andy_end
andy_loss

# si queremos ver otro o todos los sujetos a la vez del directorio diet_data

list.files()
files <- list.files("diet_data")
files

#Knowing that 'files' is now a list of the contents of 'diet_data' in alphabetical order,
#we can call a specific file by subsetting it:

files[1]
files[2]
files[3:5]

#Vamos a usar el archivo de John
head(read.csv(files[3]))

#se genera un error. por que: Well, John.csv is sitting inside the diet_data folder.
```

```r
#We just tried to run the equivalent of read.csv("John.csv") and R correctly told us
#that there isn't a file called John.csv in our working directory.
#To fix this, we need to append the directory to the beginning of the file name.

#ESTA ES LA FORMA DE LEER ARCHIVOS SITUADOS DENTRO DE UNA CARPETA

files_full <- list.files("diet_data", full.names=TRUE)
files_full

# ahora si podemos extraer el archivo, data frame, de John
# anteriormente no se abria porque no tenia toda la ruta
read.csv("diet_data/Mike.csv")
#pero ya podemos usar


head(read.csv(files_full[3]))

#So what if we wanted to create one big data frame with everybody's data in it?
#We'd do that with rbind and a loop. Let's start with rbind:

andy_david <- rbind(andy, read.csv(files_full[2]))


#if we wanted to create one big data frame with everybody's data in it?
#We'd do that with rbind and a loop. Let's start with rbind:

andy_david <- rbind(andy, read.csv(files_full[2]))
andy_david

#rbind needs 2 arguments. The first is an
#existing data frame and the second is what you want to append to it

#si quiero ver el inicio y fin del data frame

head(andy_david)
tail(andy_david)

#let's create a subset of the data frame that shows us
#just the 25th day for Andy and David.

day_25 <- andy_david[which(andy_david$Day == 25), ]
day_25

#uso de un loop para imprimir
for (i in 1:5) {print(i)}

#si quiero cargar todas las data frames en una sola que pueda manipular
#este es el procedimiento:


# primero tengo que crear una data frame vacia
dat <- data.frame()

#

for (i in 1:5) {
    dat <- rbind(dat, read.csv(files_full[i]))
}
str(dat)
#chequeo que esten todos los datos contenidos en el directorio "diet_data"

head(dat)
tail(dat)

#operaciones sobre "dat"

median(dat$Weight)

#tenemos un problema porque dat contiene NA, entonces We need to get rid of those NA's
#Approaches.
# 1) complete.cases() or is.na().
#2) look at ?median, you'll see there is an argument called na.rm that will strip
# the NA values out for us.

median(dat$Weight, na.rm=TRUE)
```

```
#other example the median weight of day 30 by taking the median of a subset
#of the data where Day=30

dat_30 <- dat[which(dat[, "Day"] == 30),]
dat_30
median(dat_30$Weight)


##build a function that will return the median weight of a given day

#defining what the arguments of the function . These are the parameters that the user will
#define.
#The first parameter will need to define is the directory that is holding the data.
#The second parameter they need to define is the day for which they want to calculate the
median

weightmedian <- function(directory, day)  {
    files_list <- list.files(directory, full.names=TRUE)   #creates a list of files
    dat <- data.frame()                             #creates an empty data frame
    for (i in 1:5) {
        #loops through the files, rbinding them together
        dat <- rbind(dat, read.csv(files_list[i]))
    }
    dat_subset <- dat[which(dat[, "Day"] == day),]  #subsets the rows that match the 'day'
argument
    median(dat_subset[, "Weight"], na.rm=TRUE)      #identifies the median weight
    #while stripping out the NAs
}

# test this function by running it a few different times:

weightmedian(directory = "diet_data", day = 20)
weightmedian("diet_data", 4)
weightmedian("diet_data", 17)
```