SOLUCIONES LAB2: Análisis exploratorio

Objetivos

Tratamos de establecer las herramientas para realizar análisis exploratorios de conjuntos de datos para cada variable de manera individual y para la relación entre dos variables. Es especialmente importante la interpretación de los resultados obtenidos.

Los objetivos específicos de esta sesión son,

- Describir correctamente una variable atendiendo a su tipo.
- Evaluar la relación entre dos variables y los posibles efectos de terceras variables sobre estas relaciones.
- Utilizar bucles *for* y sentencias condicionales.

Se van a manejar dos conjuntos de datos distintos y estructuramos la sesión teniendo en cuenta cada uno de ellos.

Dataset1: dds.dat

El estado de California asigna fondos para personas discapacitadas residentes a través del *California Department of Developmental Services* (DDS). Los individuos que reciben este tipo de fondos se denominan "beneficiarios". Se analiza el gasto medio anual por beneficiario con el objetivo de evaluar si pudiera existir cierta discriminación racial. El conjunto de datos dds.dat contiene información de una muestra de 1000 beneficiarios de este tipo de subvenciones e incluye las siguientes variables:

Variable	Descripción
id	Identificador del individuo.
age	Edad en años.
gender	Sexo del individuo. Es una variable cualitativa con 2 niveles: `Male` y `Female`.
expenditures	Cantidad de subvención anual (en dólares)
ethnicity	Grupo étnico del individuo. Es una variable cualitativa con 8 niveles: `American Indian`, `Asian`, `Black`, `Hispanic`, `Multi Race`, `Native Hawaiian`, `White not hispanic` y `Other`.

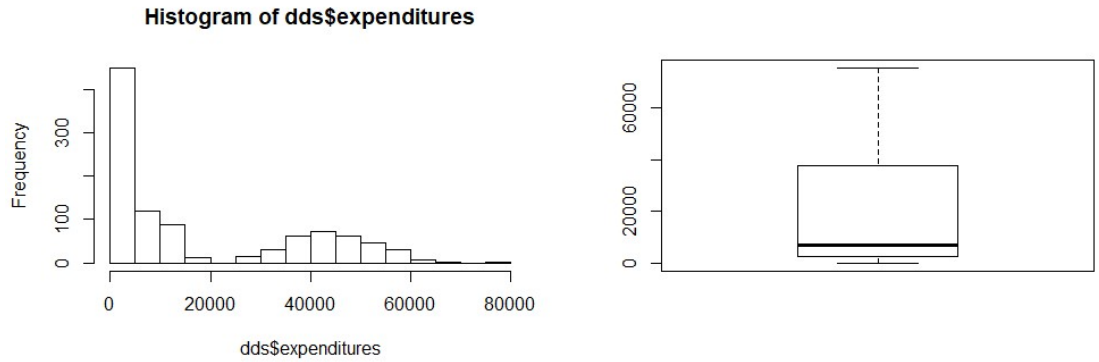
Cargamos los datos en R utilizando el código,

```
workingDir <- "D:/Ingenieria
Biomedica/Bioestadistica/Material/Laboratorios"
setwd(workingDir)
dds <- read.table(file.path(workingDir, "datos/dds.dat"), header=TRUE,
sep = "\t")
```

1. Usando los resúmenes numéricos y gráficos apropiados examina la distribución de cada una de las variables de este conjunto de datos para contestar las siguientes cuestiones.

a) Describe la distribución de la subvención anual (variable `expenditures`). Para la mayoría de los beneficiarios, ¿cómo es la cantidad de financiación recibida, relativamente alta o baja?

```
# resumen gráfico: variable cuantitativa, histograma y boxplot
par(mfrow = c(1, 2))
hist(dds$expenditures)
boxplot(dds$expenditures)
```




```
# resumen numérico: descriptivos
summary(dds$expenditures)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    222    2899    7026   18066   37713   75098
```

La distribución de la variable `expenditures` es bastante asimétrica a la derecha, lo que significa que para la mayoría de los beneficiarios, la subvención es baja, entre 0 y 5000 dólares por año. Vemos que la media está muy desplazada a la derecha respecto de la mediana. En el histograma observamos una segunda moda alrededor de los 40000 dólares/año. Existe un conjunto de beneficiarios para los que la subvención es bastante alta, en el rango de entre los 60000 y 80000 dólares anuales. Los cuartiles son 2899, 7026, y 37710 doláres/año.

b) La variable age recoge directamente la edad del beneficiario. Utiliza el siguiente código para crear una nueva variable, cualitativa, que categoriza esta variable en 6 grupos de edad:

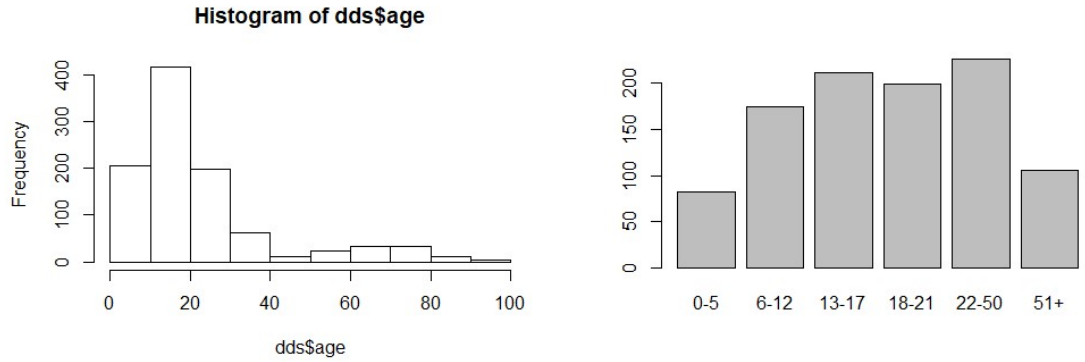
```
# crear la variable grupo de edad
age.cohort <- cut(dds$age,
                  breaks= c(-0.5,5,12,17,21,50,100), # intervalos
                  labels = c("0-5", "6-12", "13-17", "18-21", "22-50",
"51+")) # labels: etiquetas de cada intervalo
# añadimos esta variable al dataset
dds <- data.frame(dds, age.cohort)
head(dds)
```

```
     id age gender expenditures          ethnicity age.cohort
1 10210  17 Female         2113 White not Hispanic      13-17
2 10409  37   Male        41924 White not Hispanic      22-50
```

```
3 10486   3   Male          1454           Hispanic        0-5
4 10538  19 Female          6400           Hispanic      18-21
5 10568  13   Male          4412 White not Hispanic      13-17
6 10690  15 Female          4566           Hispanic      13-17
```

Describe las distribuciones de la variable age y age.cohort. ¿Cómo de jóvenes son los beneficiarios de esta muestra?

```
# resumen gráfico: histograma para la edad y barras para el grupo
par(mfrow = c(1, 2))
hist(dds$age)
plot(dds$age.cohort) # o barplot(table(dds$age.cohort))
```




```
# resumen numérico: descriptivos para la edad y frecuencias para el grupo
summary(dds$age)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0    12.0    18.0    22.8    26.0    95.0
```

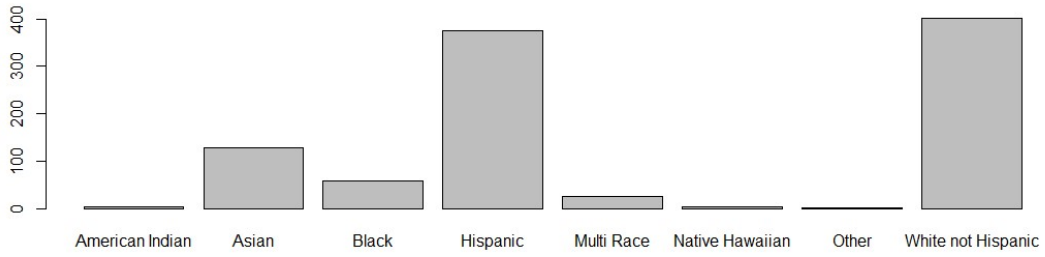
```
table(dds$age.cohort) # o summary(dds.discr$age.cohort)
```

```
  0-5  6-12 13-17 18-21 22-50   51+
   82   175   212   199   226   106
```

Como nos indica el histograma, la edad tiene una distribución asimétrica a la derecha, lo que indica que la mayoría de los beneficiarios son jóvenes, menores de 30 años. La edad mediana son 18 años y la media prácticamente 23 años. El 50% central de los beneficiarios se encuentran entre los 12 y 26 años, siendo 0 y 95 años los valores mínimo y máximo respectivamente. Aproximadamente 200 individuos están en cada uno de los 3 grupos de edad centrales, entre los 13 y 50 años, siendo los menos representados el primer y último grupo de edad: hasta los 5 años y a partir de los 51.

c) ¿Existe algún grupo racial sobre- o infra- representado en esta muestra respecto de los demás?

```
# resumen gráfico: variable cualitativa, diagrama de barras
plot(dds$ethnicity)
```




```
# resumen numérico: frecuencias
table(dds$ethnicity)
```

```
   American Indian              Asian              Black       Hispanic
                 4                129                 59            376
     Multi Race    Native Hawaiian          Other   White not Hispanic
             26                  3              2                401
```

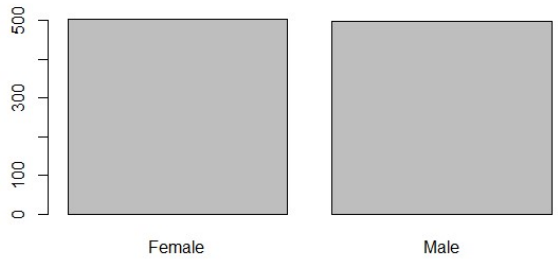
```
# frecuencias relativas
prop.table(table(dds$ethnicity))
```

```
  American Indian              Asian              Black       Hispanic
            0.004              0.129              0.059          0.376
     Multi Race    Native Hawaiian          Other   White not Hispanic
          0.026              0.003              0.002              0.401
```

Hay 8 grupos raciales en esta muestra y no están igualmente representados. Los dos grupos más grandes son hispanos y blancos no hispánicos, juntos representan aproximadamente el 80% de la muestra.

d) ¿Está la muestra balanceada respecto del sexo de los beneficiarios?

```
# resumen gráfico: variable cualitativa diagrama de barras
plot(dds $gender)
```




```
# resumen numérico: frecuencias
table(dds$gender)
```

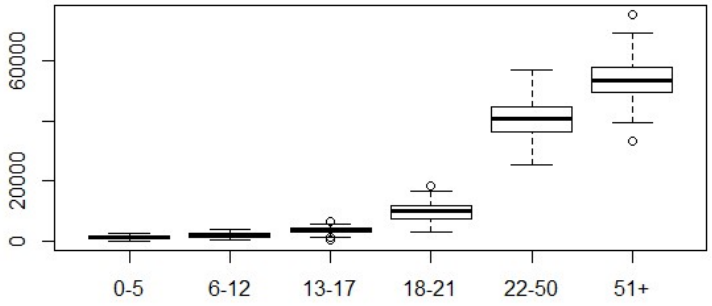
```
Female   Male
   503    497
```

Sí, aproximadamente la mitad de los beneficiarios son hombres y la otra mitad mujeres, aproximadamente 500 de cada sexo.

2. Después de examinar las variables individualmente, vamos a explorar las relaciones entre la variable respuesta, en este caso `expenditures`, y cada una de las variables explicativas.

a) ¿Cómo varían las subvenciones respecto de la edad de los beneficiarios? Utiliza el grupo de edad (variable `age.cohort`).

```
# resumen numérico: cuantitativa vs cualitativa, diagrama de cajas
boxplot(dds$expenditures ~ dds$age.cohort)
```



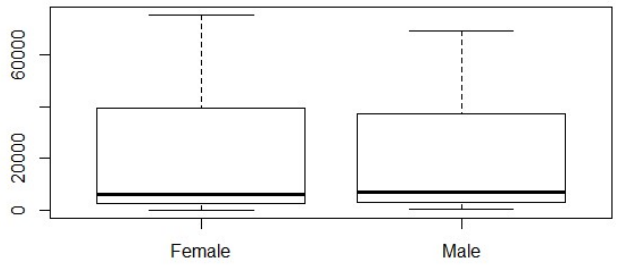

```
# resumen numérico: descriptivos en cada grupo de edad
summary(dds$expenditures[dds$age.cohort == "0-5"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    222    1034    1380    1415    1739    2750
summary(dds$expenditures[dds$age.cohort == "6-12"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    620    1602    2191    2227    2846    4163
summary(dds$expenditures[dds$age.cohort == "13-17"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    386    3306    3952    3923    4666    6798
summary(dds$expenditures[dds$age.cohort == "18-21"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3153    7588    9979    9889   11806   18435
summary(dds$expenditures[dds$age.cohort == "22-50"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  25348   36447   40456   40209   44721   56716
summary(dds$expenditures[dds$age.cohort == "51+"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  33110   49515   53509   53522   57746   75098
```

Observamos una clara relación entre la cantidad de subvención recibida y el grupo de edad: individuos mayores reciben una asignación superior. Para los 3 primeros grupos de edad, hasta los 17 años, la subvención media varía entre 1400 y 4000 dólares por año. En los dos últimos grupos de edad, beneficiarios de más de 22 años, esta media está en torno a los 40-50 mil dólares al año. Esta tendencia es consistente con el objetivo de la subvención, que pretende ayudar a las personas discapacitadas a mantener una calidad de vida aceptable. Los

beneficiarios más jóvenes vivirán en el domicilio familiar y/o pueden recibir soporte económico de sus familias. En cambio, los beneficiarios mayores, probablemente vivirán sin ese soporte familiar, por lo que sus necesidades serán mayores.

b) ¿Varían las subvenciones respecto del sexo del beneficiario?

```
# resumen numérico: cuantitativa vs cualitativa, diagrama de cajas
boxplot(dds$expenditures ~ dds$gender)
```

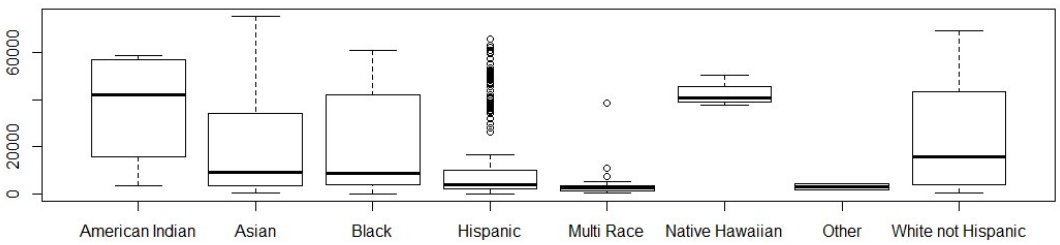



```
# resumen numérico: descriptivos en cada sexo
summary(dds$expenditures[dds$gender == "Female"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    222    2872    6400   18130   39488   75098
summary(dds$expenditures[dds$gender == "Male"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    386    2954    7219   18001   37201   68890
```

No, la distribución de la subvención es muy similar en hombres y mujeres.

c) ¿Cómo varía la subvención respecto de la raza del beneficiario? ¿Parece que existen diferencias entre la subvención anual que una persona recibe, de media, dependiendo de su raza?

```
# resumen numérico: cuantitativa vs cualitativa, diagrama de cajas
boxplot(dds$expenditures ~ dds$ethnicity)
```




```
# resumen numérico: descriptivos en cada grupo étnico
summary(dds$expenditures[dds$ethnicity == "American Indian"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
   3726   22085   41818   36438   56171   58392
summary(dds$expenditures[dds$ethnicity == "Asian"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    374    3382    9369   18392   34274   75098
summary(dds$expenditures[dds$ethnicity == "Black"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    240    3870    8687   20885   41857   60808
summary(dds$expenditures[dds$ethnicity == "Hispanic"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    222    2331    3952   11066   10292   65581
summary(dds$expenditures[dds$ethnicity == "Multi Race"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    669    1690    2622    4457    3750   38619
summary(dds$expenditures[dds$ethnicity == "Native Hawaiian"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  37479   39103   40727   42782   45434   50141
summary(dds$expenditures[dds$ethnicity == "Other"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2018    2667    3316    3316    3966    4615
summary(dds$expenditures[dds$ethnicity == "White not Hispanic"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    340    3977   15718   24698   43134   68890
```

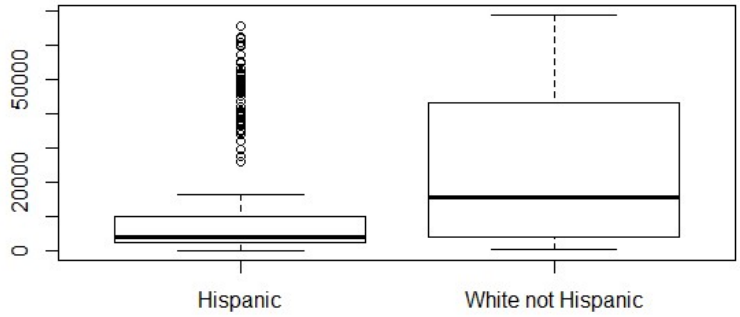
La distribución de las subvenciones es bastante diferente para cada grupo étnico. Por ejemplo, hay muy poca variabilidad en algunos grupos como el muti-racial, los nativos hawaianos y otras razas. En contraste, grupos como los blancos no hispanos tienen una variabilidad muy grande. En cuanto a la subvención media recibida parece que hay bastantes diferencias. Los indios americanos y los nativos hawaianos son los que más subvención reciben con cantidades medias en torno a los 40000 dólares al año. En contraste, los hispanos sólo reciben 11000 dólares anuales de media.

3. Como se vio en el apartado 1.c, los grupos raciales `Hispanic` and `White non-Hispanic` son los mayoritarios en este conjunto de datos. En este punto vamos a enfocar nuestros análisis en estos dos grupos raciales. Construimos el *dataset* `dds.white` seleccionando estos dos grupos étnicos con el siguiente código,

```
dds.white <- dds[dds$ethnicity == "Hispanic" | dds$ethnicity == "White
not Hispanic", ]
# eliminamos los niveles vacíos de la variable ethnicity
dds.white$ethnicity <- factor(dds.white$ethnicity, levels=
c("Hispanic", "White not Hispanic"))
```

a) Comparar gráfica y numéricamente la distribución de las subvenciones en estos dos grupos raciales. De media, los beneficiarios hispanos ¿reciben menos fondos que los blancos no hispanos?

```
# resumen numérico: cuantitativa vs cualitativa, diagrama de cajas
boxplot(dds.white$expenditures ~ dds.white$ethnicity)
```



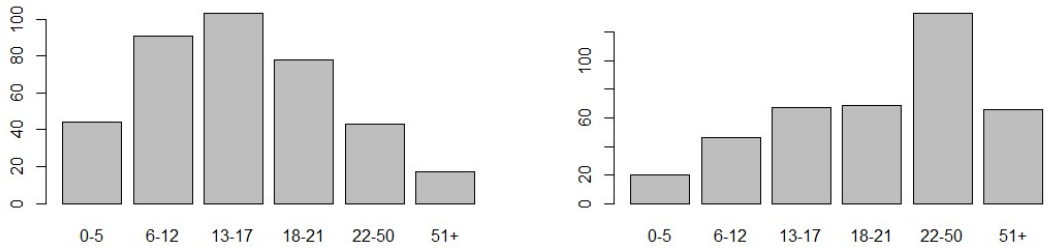

```
# resumen numérico: descriptivos en cada grupo racial
summary(dds.white$expenditures[dds.white$ethnicity == "Hispanic"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    222    2331    3952   11066   10292   65581
summary(dds.white$expenditures[dds.white$ethnicity == "White not Hispanic"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    340    3977   15718   24698   43134   68890
```

Basándonos en el diagrama de cajas, la mayoría de los beneficiarios hispanos reciben una subvención menor que la subvención mediana de los blancos no hispanos. En el grupo de hispanos subvenciones por encima de los 20000 dólares anuales son considerados *outliers*. El 50% central de los hispanos reciben entre 2000 y 11000 dólares al año frente a los blancos no hispanos que esa misma proporción recibe entre 4000 y 43000 dólares por año. La subvención media para los hispanos es de 11066 dólares / año y para los blancos no hispanos 24698 dólares / año. En esta muestra parece que los blancos hispanos reciben más fondos que los hispanos.

b) En el apartado 2a comprobamos que la cantidad de fondos recibidos estaba influenciada por la edad del beneficiario, siendo los mayores los que más fondos reciben. ¿Existe una relación entre la edad y la raza en estos dos grupos raciales? Explora la relación entre la variable `ethnicity` y `age.cohort`. *Pista*: para el resumen gráfico construye un diagrama de barras de los grupos de edad en cada raza por separado. Para el resumen numérico calcula la tabla de frecuencias condicionada que mejor resuma la asociación entre estas variables.

```
# resumen grafico: diagrama de barras por separado para cada raza
par(mfrow = c(1, 2))
plot(dds.white$age.cohort[dds.white$ethnicity == "Hispanic"])
```

```
plot(dds.white$age.cohort[dds.white$ethnicity == "White not Hispanic"])
```




```
# resumen numérico: tabla de contingencia
table(dds.white$age.cohort,dds.white$ethnicity)
```

```
        Hispanic White not Hispanic
  0-5          44                 20
  6-12         91                 46
  13-17       103                 67
  18-21        78                 69
  22-50        43                133
  51+          17                 66
```

```
# condicionamos por columna
prop.table(table(dds.white$age.cohort,dds.white$ethnicity),margin = 2)
```

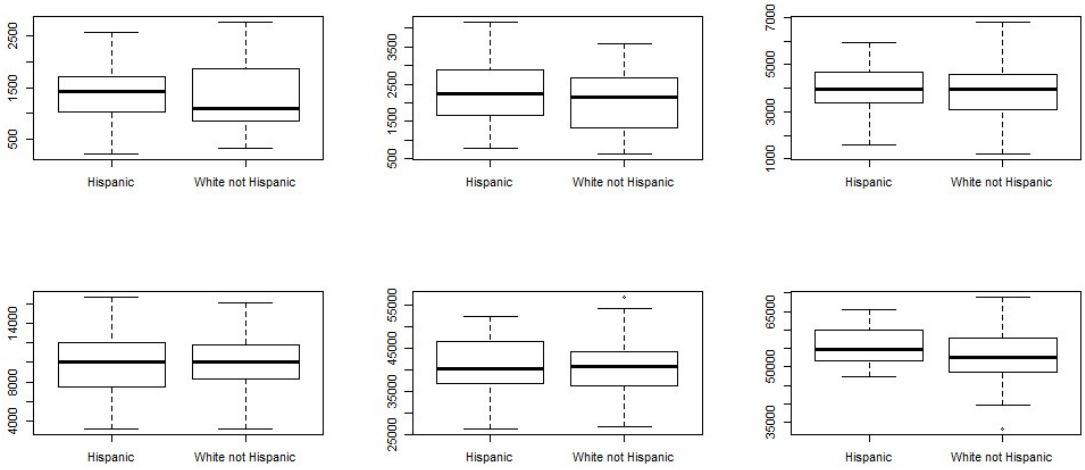
```
          Hispanic White not Hispanic
  0-5    0.11702128         0.04987531
  6-12   0.24202128         0.11471322
  13-17  0.27393617         0.16708229
  18-21  0.20744681         0.17206983
  22-50  0.11436170         0.33167082
  51+    0.04521277         0.16458853
```

El grupo de hispanos es más joven, con una mayor proporción de beneficiarios en los grupos de edad entre 6 y 21 años. El grupo de edad mayoritario para los blancos no hispanos es el de los 22-50 años. El grupo de mayores de 51 años que reciben más fondos, supone un 17% del grupo de blancos no hispánicos frente a un 5% de los hispanos.

c) Es muy importante explorar las posibles relaciones entre las variables explicativas, ya que una relación entre ellas podría confundir la relación con la variable respuesta. En este conjunto de datos, la edad es un factor de confusión para la relación entre la subvención y la raza, puesto que la edad y la raza están fuertemente relacionadas en esta muestra. Construye un diagrama de cajas para cada grupo de edad, que evalúe la relación entre la subvención y el grupo étnico. ¿Observas el mismo tipo de relación que el visto en el apartado 3a?

```
# resumen grafico: boxplot en cada grupo de edad
par(mfrow = c(2, 3))
boxplot(dds.white$expenditures [dds.white$age.cohort == "0-5"] ~
```

```
        dds.white$ethnicity[dds.white$age.cohort == "0-5"])
boxplot(dds.white$expenditures [dds.white$age.cohort == "6-12"] ~
        dds.white$ethnicity[dds.white$age.cohort == "6-12"])
boxplot(dds.white$expenditures [dds.white$age.cohort == "13-17"] ~
        dds.white$ethnicity[dds.white$age.cohort == "13-17"])
boxplot(dds.white$expenditures [dds.white$age.cohort == "18-21"] ~
        dds.white$ethnicity[dds.white$age.cohort == "18-21"])
boxplot(dds.white$expenditures [dds.white$age.cohort == "22-50"] ~
        dds.white$ethnicity[dds.white$age.cohort == "22-50"])
boxplot(dds.white$expenditures [dds.white$age.cohort == "51+"] ~
        dds.white$ethnicity[dds.white$age.cohort == "51+"])
```



Cuando comparamos las subvenciones en cada uno de los grupos de edad por separado, observamos que las distribuciones entre hispanos y blancos no hispanos son bastante similares. Claramente en esta muestra la edad es un factor de confusión para la relación entre la raza y los fondos recibidos.

Dataset2: golub.dat

En 1999, Golub y colaboradores[1] publicaron un estudio sobre leucemia que representó una de las primeras aplicaciones de la tecnología de microarrays para fines de diagnóstico. En aquel momento, no se realizaba ninguna prueba diagnóstica que ayudara a distinguir entre leucemia mieloide aguda (AML) y leucemia linfoblástica aguda (ALL). En este trabajo, Golub trataba de investigar si el perfil de expresión génica podría ser una herramienta útil para hacer esta clasificación. Utilizaron microarrays de DNA de Affymetrix para medir el nivel de expresión de 7129 genes en niños con leucemia aguda en los que se conocía el tipo. El objetivo del estudio

[1] Golub TR et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286 (1999): 531-537.

fue desarrollar un procedimiento para distinguir entre *AML* o *ALL* a partir de los niveles de expresión génica de un paciente. Específicamente se trataba de abordar dos cuestiones importantes:

1. ¿Qué genes son los más informativos? Si un gen se expresa diferencialmente entre individuos con cada tipo de leucemia, el nivel de expresión de ese gen puede ser informativo a la hora diagnosticar el tipo de leucemia.

2. ¿Cómo se puede predecir el tipo de leucemia a partir de los datos de expresión? El perfil de expresión de un individuo son sus niveles de expresión en un grupo de genes. Buscamos un perfil que nos permita traducir los datos de expresión en una predicción del tipo de leucemia.

Los datos golob.dat contienen los niveles de expresión de 72 pacientes, 62 de ellos se utilizarán para identificar los genes informativos o diferencialmente expresados y los otros 10 para comprobar lo buena o mala que es la estrategia de predicción.

Además de la cantidad de expresión de cada uno de los genes se recogen las siguientes variables,

Variable	Descripción
Samples	Identificador del individuo, único para cada paciente
BM.PB	Tipo de tejido del que se ha conseguido extraer el perfil génico. Puede tomar dos valores `BM` para médula espinar y `PB` para sangre periférica.
Gender	Sexo del individuo. Puede tomar dos valores `M` para varones y `F` para mujeres.
Source	Hospital donde se trata al paciente.
tissue.mf	Combinación de las variables BM.PB y Gender
cancer	Tipo de leucemia. Puede tomar tres valores: `aml` para leucemia aguda mieloide, `allB` para leucemia aguda limfoblástica con origen en células B y `allT` para leucemia aguda limfoblástica con origen en células T.
gr	Grupo de muestra. Puede tomar dos valores: `train` para indicar los 62 individuos utilizados en la identificación de los genes diferencialmente expresados y `test` para indicar los 10 individuos utilizados en la fase de predicción.

Cargamos los datos en R utilizando el código,

```
workingDir <- "D:/Ingenieria
Biomedica/Bioestadistica/Material/Laboratorios"
setwd(workingDir)
golub <- read.table(file.path(workingDir, "datos/golub.dat"),
header=TRUE, sep = "\t")
```

Identificar los genes informativos

Aunque este conjunto de datos es pequeño en relación al tamaño que se maneja en estos momentos en el ámbito de la expresión génica, vamos a simplificar esta primera parte, examinando una muestra aleatoria de 100 genes para todos los pacientes en el grupo `train`

del *data frame* golub. Para manejar todos la misma muestra vamos a hacer esta selección con el siguiente código,

```
# nuestro data frame tiene 7136 variables.
# Podemos consultar qué variables con el comando
colnames(golub)
```

```
   [1] "Samples"                    "BM.PB"
   [3] "Gender"                     "Source"
   [5] "tissue.mf"                  "cancer"
   [7] "gr"                         "AFFX.BioB.5_at"
   [9] "AFFX.BioB.M_at"             "AFFX.BioB.3_at"
  [11] "AFFX.BioC.5_at"             "AFFX.BioC.3_at"
  [13] "AFFX.BioDn.5_at"            "AFFX.BioDn.3_at"
  [15] "AFFX.CreX.5_at"             "AFFX.CreX.3_at"
  [17] "AFFX.BioB.5_st"             "AFFX.BioB.M_st" …
```

```
# notar que las 7 primeras variables son variables clínicas
# los genes que se han medido son colnames(golub)[-(1:7)]
# establecemos una semilla para poder reproducir los resultados
set.seed(2401)
# creamos un vector con los genes seleccionados
gene.selected <- sample(colnames(golub)[-(1:7)], size = 100, replace =
FALSE) # sin reemplazamiento para no repetir genes!
# creamos una data frame con los genes seleccionados
# variables: las 7 primeras variables y los genes seleccionados
# individuos: los 62 que tienen el valor train en la variable gr
golub.train <- golub[golub$gr== "train",
c(colnames(golub)[1:7],gene.selected)]
# comprobamos las dimensiones de este data frame
dim(golub.train)
```

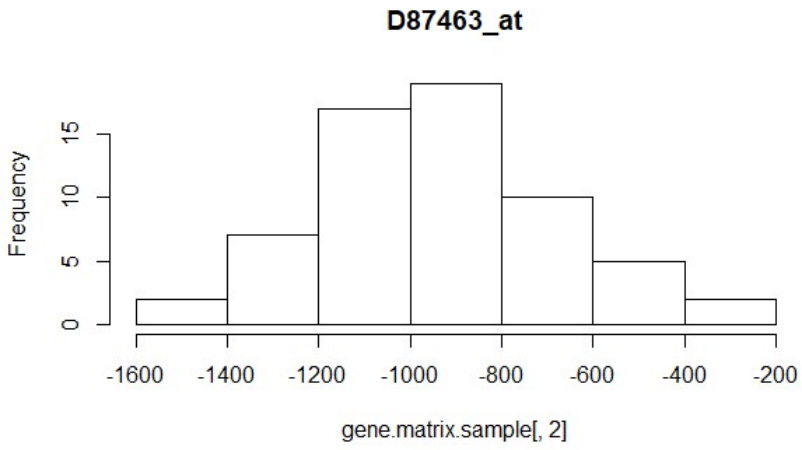
```
[1]  62 107
```

```
# creamos una matriz numérica únicamente con los genes
gene.matrix.sample <- as.matrix(golub.train[ ,-(1:7)])
class(gene.matrix.sample)
```

```
[1] "matrix"
```

1. Construye un histograma que muestre la distribución de la expresión del segundo gen. Describe la forma de esta distribución.

```
# histograma. Usamos el argumento main para poner el nombre de gen en el título
hist(gene.matrix.sample[ ,2], main=colnames(gene.matrix.sample)[2])
```



La distribución es bastante simétrica, algo asimétrica a la izquierda. Todos los valores son negativos y centrados en torno a -900.

2. Crea un vector lógico, `leuk.type`, que tenga el valor `TRUE` para la leucemia de tipo AML y `FALSE` en caso contrario, es decir si la variable `cancer` toma los valores `allT` o `allB`. ¿Cuántos pacientes hay de cada tipo en esta muestra?

```
# el vector lógico
leuk.type <- (golub.train$cancer == "aml")
# para ver cuántos niños hay de cada tipo utilizamos la función table
table(leuk.type)
```

```
leuk.type
FALSE  TRUE
   42    20
```

Hay 42 pacientes con ALL y 20 con AML.

3. Calcula el nivel medio de expresión para cada uno de los genes en cada grupo de leucemia. Utiliza la función `apply()` que repite una operación para cada fila o columna de una matriz (*más información sobre su uso en los apuntes del laboratorio 2*). Almacena los resultados en dos vectores, `aml.mean.expression` para los pacientes con *AML*, y `all.mean.expression` para los pacientes con *ALL*.

```
# nivel de expresión medio para los pacientes con aml
aml.mean.expression <- apply(gene.matrix.sample[leuk.type == TRUE, ],
2, mean)
# nivel de expresión medio para los pacientes con all
all.mean.expression <- apply(gene.matrix.sample[leuk.type == FALSE, ],
2, mean)
```

Cada uno de estos objetos es un vector de longitud 100, uno por cada gen, que contiene en cada posición el nivel promedio de expresión observado en cada tipo de paciente.

4. Para cada gen, compara la expresión promedio entre pacientes con *AML* y pacientes con *ALL*. Para ello, calcula las diferencias en el nivel de expresión promedio y almacénalo en el vector `diff.mean.expression`.

```
# diferencia, R hace las operaciones elemento a elemento luego
diff.mean.expression <- (aml.mean.expression - all.mean.expression)
```

a) ¿Cuál es la diferencia de expresión media para el primer gen de la lista entre los dos tipos de leucemia? ¿En cuál de los dos tipos de paciente está sobre-expresado este gen? Teniendo en cuenta esta diferencia, ¿podrías decir, sin utilizar más información, que este gen es un gen informativo para la identificación del tipo de leucemia?

```
# diferencia de expresión media para el primer gen
diff.mean.expression[1]
```

```
M86752_at
-1131.455
```

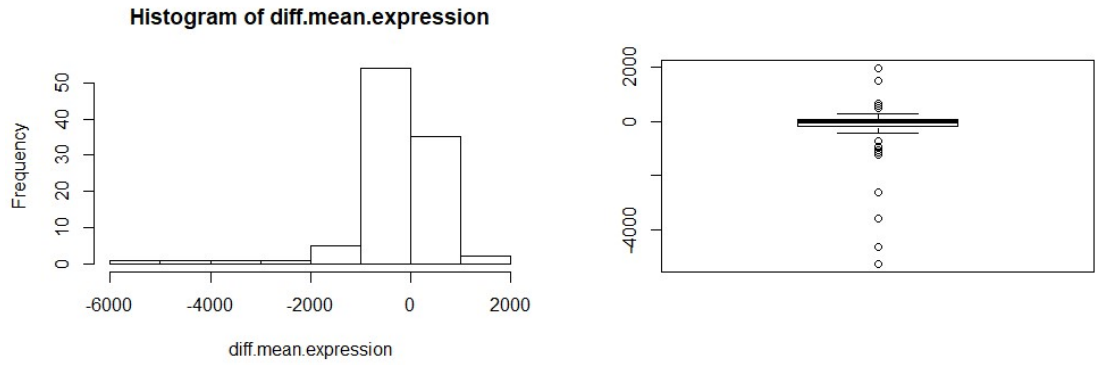
La diferencia es negativa, por lo tanto el gen M86752_at está sobre-expresado en el grupo de pacientes con ALL. No es posible saber, utilizando únicamente esta diferencia si este gen es o no un gen informativo para la identificación del tipo de leucemia, puesto que no sabemos si una diferencia entorno a 1100 es la diferencia que nos encontraremos en el resto de genes o es muy grande. Necesitaremos por tanto conocer la distribución de la diferencia de expresiones promedio.

b) Describe la distribución de las diferencias en los niveles medios de expresión utilizando herramientas numéricas y gráficas. Describe esta distribución. Puedes contestar ahora sí, es esperable que el primer gen de la lista sea un gen informativo para la identificación del tipo de leucemia.

```
# resumen numérico: descriptivos
summary(diff.mean.expression)
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-5274.84  -181.47   -30.31  -215.81    71.63  1968.01
```

```
# resumen gráfico: histograma y boxplot
par(mfrow = c(1, 2))
hist(diff.mean.expression)
boxplot(diff.mean.expression)
```



Observamos que para la gran mayoría de genes las diferencias están entre -1000 y 1000. Para el 50% central de las observaciones (genes) estas diferencias se encuentran entre -181.47 and 71.63 (primer y tercer cuartil). En el *boxplot* observamos que hay genes con diferencias consideradas extremas tanto por arriba como por abajo, y serían precisamente estos genes los que mejor diferenciarían el tipo de leucemia, ya que se caracterizan por tener diferencias en los niveles medios de expresión entre pacientes con *AML* y *ALL* extremas. En el apartado anterior vimos que el gen M86752_at tiene una diferencia de medias en torno a -1100, que es un valor menor que el primer cuartil, es, por tanto pequeño, pero aún no sabemos si es lo suficientemente pequeño para poder considerarlo un *outlier*.

5. Vamos a identificar los *outliers* para la diferencia de niveles de expresión promedio. Estos *outliers* serán los genes que muestran diferencias más extremas y por tanto los genes a priori más informativos. Identificamos los *outliers* utilizando el criterio de los cuartiles utilizado en los diagramas de cajas. El siguiente código permite identificar los valores extremos por arriba, es decir por diferencias extremadamente grandes.

```
# calculamos el primer y tercer cuartil
quart.1 <- quantile(diff.mean.expression, 0.25, na.rm = TRUE)
quart.3 <- quantile(diff.mean.expression, 0.75, na.rm = TRUE)
# rango intercuartílico
iqr <- quart.3 - quart.1 # o el comando IQR(diff.mean.expression)
# definimos los límites para determinar los outliers
lb.outlier <- quart.1 - 1.5*iqr # inferior
ub.outlier <- quart.3 + 1.5*iqr # superior
# lista de los outliers por arriba (expresión > límite superior)
large.out <- (diff.mean.expression > ub.outlier)
diff.mean.expression[large.out]
  U57592_at   J03040_at U05681_s_at   D38535_at M13452_s_at
   653.9938    568.2183   1493.6025    504.2603   1968.0096
# ordenados de mayor a menor
sort(diff.mean.expression [large.out], decreasing = TRUE)
M13452_s_at U05681_s_at   U57592_at   J03040_at   D38535_at
```

```
  1968.0096    1493.6025     653.9938     568.2183     504.2603
```

a) ¿Cuáles son los valores de los límites superior e inferior?

```
# inferior
lb.outlier
```

```
      25%
-561.1334
```

```
# superior
ub.outlier
```

```
     75%
451.2931
```

b) ¿Cuántos genes son *outliers* por tener diferencias en los niveles de expresión promedio extremadamente altos?

```
table(large.out)
```

```
large.out
FALSE  TRUE
   95     5
```

```
# podríamos también
sum(large.out)
```

```
[1] 5
```

c) Encuentra los *outliers* por tener diferencias en los niveles de expresión promedio extremadamente bajos. ¿Cuántos outliers hay de este tipo?

```
# lista de los outliers por abajo (expresión < límite inferior)
small.out <- (diff.mean.expression < lb.outlier)
sum(small.out)
```

```
[1] 12
```

```
diff.mean.expression[small.out]
```

```
          M86752_at            D79205_at   AFFX.HUMRGE.M10098_5_at
         -1131.4553           -3590.3593                -5274.8429
          U11863_at            U05340_at               X98482_r_at
         -1220.0197            -933.1331                -2637.0440
          V00563_at            U30825_at                 U27460_at
         -4635.7772           -1150.3442                 -742.7816
          X52882_at            U83843_at                 Z30426_at
         -1065.0601            -727.2129                -1217.0467
```

```
# ordenados de menor a mayor
sort(diff.mean.expression [small.out], decreasing = FALSE)
```

```
AFFX.HUMRGE.M10098_5_at           V00563_at            D79205_at
             -5274.8429         -4635.7772           -3590.3593
            X98482_r_at           U11863_at            Z30426_at
             -2637.0440         -1220.0197           -1217.0467
              U30825_at           M86752_at            X52882_at
             -1150.3442         -1131.4553           -1065.0601
              U05340_at           U27460_at            U83843_at
              -933.1331           -742.7816            -727.2129
```

Hay 12 genes con diferencias extremadamente pequeñas.

d) ¿Cuál es el gen más informativo sobre-expresado para el tipo de leucemia *AML*? ¿Y el sobre-expresado para el tipo *ALL*?

Las diferencias se han calculado como AML-ALL, por tanto, el gen más sobre-expresado en AML es el que tiene una diferencia más positiva, el que ocupa la posición 65,

```
# posición
which.max(diff.mean.expression)
```

```
M13452_s_at
         65
```

```
# valor de la diferencia de medias
diff.mean.expression [which.max(diff.mean.expression)]
```

```
M13452_s_at
    1968.01
```

El gen más sobre-expresado en pacientes con ALL será el que tenga la diferencia más negativa.

```
# posición
which.min(diff.mean.expression)
```

```
AFFX.HUMRGE.M10098_5_at
                     21
```

```
# valor de la diferencia de medias
diff.mean.expression [which.min(diff.mean.expression)]
```

```
AFFX.HUMRGE.M10098_5_at
              -5274.843
```

e) El primer gen de nuestra muestra, ¿será un gen informativo para la identificación del tipo de leucemia?

En los apartados anteriores vimos que el gen M86752_at tiene una diferencia de medias en torno a -1100, un valor pequeño en relación al resto. Además, la diferencia de medias está por debajo del límite inferior, por lo tanto es extremo y sí será un gen informativo.

6. Hasta aquí hemos identificado los genes informativos en una muestra de 100 genes. Modifica ligeramente el código para obtener los genes informativos de los 7129 genes analizados en el estudio original. El punto de partida será la matriz de expresión de los genes,

```
gene.matrix <- as.matrix(golub[golub$gr== "train",-(1:7)])
```

y el vector lógico calculado anteriormente como,

```
leuk.type <- (golub.train$cancer == "aml")
# calcula la expresión media para cada gen en cada tipo de paciente
aml.mean.expression <- apply(gene.matrix[leuk.type == TRUE, ], 2,
mean)
all.mean.expression <- apply(gene.matrix[leuk.type == FALSE, ], 2,
mean)
# diferencias en los niveles medios de expresión
diff.mean.expression <- (aml.mean.expression - all.mean.expression)
```

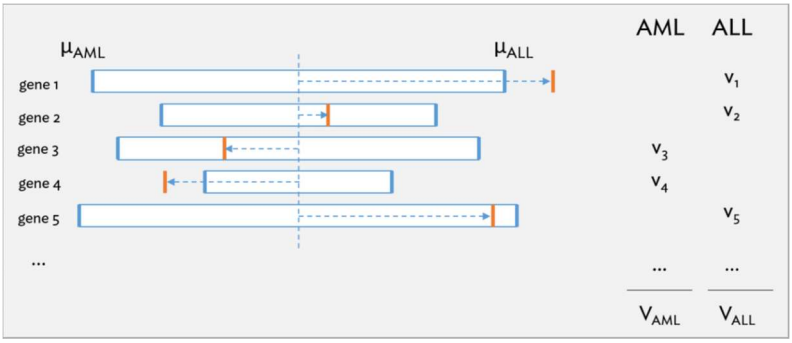
```
# definimos los límites para determinar los outliers
quart.1 <- quantile(diff.mean.expression, 0.25, na.rm = TRUE)
quart.3 <- quantile(diff.mean.expression, 0.75, na.rm = TRUE)
iqr <- quart.3 - quart.1
lb.outlier <- quart.1 - 1.5*iqr
ub.outlier <- quart.3 + 1.5*iqr
# outliers por arriba
large.out <- (diff.mean.expression > ub.outlier)
sum(large.out)
[1] 593
#los 5 más extremos
sort(diff.mean.expression[large.out], decreasing = TRUE)[1:5]
M19507_at      M27891_at      M11147_at M96326_rna1_at    Y00787_s_at
19820.86       17657.05       16235.83       15914.76       15845.77
# outliers por abajo
small.out <- (diff.mean.expression < lb.outlier)
sum(small.out)
[1] 668
# los 5 más extremos
sort(diff.mean.expression[small.out], decreasing = FALSE)[1:5]
M14483_rna1_s_at X82240_rna1_at  X58529_at  M33680_at U05259_rna1_at
      -11293.653     -10415.906  -9629.464  -9296.219      -8383.484
```

Predecir el tipo de leucemia

La estrategia que utilizó Golub y colaboradores para predecir el tipo de leucemia a partir de los datos de expresión fue la siguiente. Para cada individuo en el grupo `test` se compara el nivel de expresión de cada uno de los genes identificados como informativos, con el nivel promedio de dicho gen para pacientes de tipo AML y de tipo ALL en el grupo `train`, denotados como μ_{AML} y μ_{ALL}, respectivamente. Un gen determinado vota por AML o ALL, dependiendo de si su nivel de expresión está más cerca de μ_{AML} o μ_{ALL}. El paciente se clasifica en el tipo que más votos obtenga en todos los genes identificados como informativos. La siguiente figura representa un ejemplo de este sistema de votación en un paciente. La línea naranja es el nivel de expresión del paciente y las azules los niveles de expresión promedio en cada tipo del grupo `train`, es decir, μ_{AML} y μ_{ALL}. El gen 1 vota por el tipo ALL, puesto que su nivel de expresión está más cerca de μ_{ALL}. El gen 3 vota por AML puesto que su nivel de expresión está más cerca de μ_{AML}.

Los votos no serán dicotómicos, sino que se ponderan para tener en cuenta cómo de lejos estamos del punto medio entre los dos tipos (línea de puntos vertical). Pesará más el voto que se encuentre más lejos de este punto. Así, el voto del gen 2 (v_2) no es tan claro como el voto del gen 1 (v_1) por el tipo ALL, por lo que v_1 será mayor que v_2. Para asignar finalmente el paciente a

un tipo de leucemia, se suman todos los votos, ganando el tipo que obtenga una mayor puntuación.



En el artículo original utilizaron 50 genes informativos para hacer esta clasificación. Por simplicidad, nosotros sólo vamos a utilizar los 10 genes más atípicos en la diferencia entre las medias de expresión de AML y ALL, 5 atípicos por arriba y 5 por abajo. Se va a utilizar la expresión en estos 10 genes en los pacientes del grupo `test` para hacer la predicción del tipo de leucemia.

```
# creamos el dataframe golub.test
golub.test <- golub[golub$gr== "test", ]
# los 10 genes más informativos (punto 6) son:
inf.genes <- c("M19507_at", "M27891_at", "M11147_at",
"M96326_rna1_at", "Y00787_s_at", "M14483_rna1_s_at", "X82240_rna1_at",
"X58529_at", "M33680_at", "U05259_rna1_at")
```

Recordad que podemos acceder a las columnas de un *data frame* (o matriz) por sus nombres, así el comando `golub.test[ , inf.genes]` devolverá la expresión de los genes informativos del *data frame* `golub.test`.

7. Calcular μ_{AML} y μ_{ALL} para cada gen informativo en el *data frame* `golub.train`. Almacenamos estos resultados en los vectores `mu.ALL` y `mu.ALL`. ¿Cuáles son los valores de estas medias para el primer gen informativo?

```
# vector lógico con TRUE para aml y FALSE en el resto de pacientes del
grupo train
leuk.type <- (golub.train$cancer == "aml")
golub.train.aml <- golub.train[leuk.type == TRUE, ] #o
golub.train[leuk.type, ]
golub.train.all <- golub.train[leuk.type == FALSE, ] #o
golub.train[!leuk.type, ]
# calcular la media para AML
mu.aml <- apply(golub.train.aml[, inf.genes], 2, mean)
# calcular la media para ALL
mu.all <- apply(golub.train.all[, inf.genes], 2, mean)
# medias para el primer gen informativo
mu.aml[1]
```

```
M19507_at
 20142.84
mu.all[1]
M19507_at
 321.9871
```

8. Utiliza el siguiente código para determinar la dirección del voto de cada uno de los genes en los 10 pacientes del grupo `test`. El valor 0 representa el voto para AML y 1 para ALL.

Usamos dos bucles `for()` y una estructura condicional `if()` para asignar las direcciones del voto de cada uno de los 10 genes informativos y cada uno de los 10 pacientes del grupo `test`.

```
# creamos la matriz con la expresión de los genes informativos
test.gene.matrix <- golub.test[, inf.genes]
# creamos una matriz vacía, votes, en la que almacenaremos los votos
num.genes <- ncol(test.gene.matrix)
num.patients <- nrow(test.gene.matrix)
votes <- matrix(nrow = num.patients, ncol = num.genes)
# calculamos la dirección y distancia para cada gen
dist.from.aml <- vector("numeric", num.genes)
dist.from.all <- vector("numeric", num.genes)
for(i in 1:dim(votes)[1]){ # repetir desde 1 hasta el nº de filas de
votes (paciente)
   for(j in 1:dim(votes)[2]) { # repetir desde 1 hasta el nº de
columnas de votes (genes)
      # distancias a mu del paciente i (valor absoluto: abs)
      dist.from.aml[i] <- abs(test.gene.matrix[i, j] - mu.aml[j])
      dist.from.all[i] <- abs(test.gene.matrix[i, j] - mu.all[j])
      if(dist.from.aml[i] <= dist.from.all[i]){
         votes[i, j] <- 0  # 0 si la distancia a mu.AML<= que a mu.ALL
      } else { votes[i, j] = 1 # 1 en otro caso
      }
   }
}
```

a) Almacenamos los resultados en una matriz llamada `votes`. ¿Cuáles son las dimensiones de esta matriz? Cada fila de esta matriz, ¿contiene los votos de un paciente o los votos de un gen?

```
# dimensiones de votes
dim(votes)
[1] 10 10
```

La matriz votes tiene 10 filas y 10 columnas. La sintaxis que hemos utilizado para construirla especifica que el número de filas es el número de pacientes y el de columnas el número de genes. Por tanto, la primera fila serán los votos de los 10 genes para el paciente número 1.

b) Los dos vectores `dist.from.aml` y `dist.from.all` almacenan las distancias entre la expresión en el paciente y las medias para los tipos AML y ALL respectivamente. Esta distancia

se calcula como el valor absoluto de la diferencia. Se necesitan dos bucles for(), uno que varía el índice i desde 1 hasta 10 y el otro que varía j desde 1 hasta 10 también.

i) Cuando i=1 y j=1, describe los cálculos que se realizan y almacenan en `dist.from.aml` y `dist.from.all`.

Cuando i = 1 y j = 1, se calcula la distancia entre la expresión del primer gen informativo y μ_{AML} o μ_{ALL} respectivamente en el primer paciente. `test.gene.matrix[1,1]` contiene el nivel de expresión del paciente 1 en el gen 1, `mu.aml[1]` y `mu.aml[1]` contiene los valores promedios para ese gen. `abs()` calcula el valor absoluto. Estos dos cálculos se almacenan en la primera posición de los vectores `dist.from.aml` y `dist.from.all`.

ii) ¿Qué se calcula y almacena cuando i=2 y j=1?

Cuando i = 2 y j = 1, se calcula la distancia entre la expresión del primer gen informativo y μ_{AML} o μ_{ALL} respectivamente en el segundo paciente. `test.gene.matrix[2,1]` contiene el nivel de expresión del paciente 2 en el gen 1, `mu.aml[1]` y `mu.aml[1]` contiene los valores promedios para ese gen. `abs()` calcula el valor absoluto. Estos dos cálculos se almacenan en la segunda posición de los vectores `dist.from.aml` y `dist.from.all`.

iii) ¿Qué se calcula y almacena cuando i=1 y j=2?

Cuando i = 1 y j = 2, se calcula la distancia entre la expresión del segundo gen informativo y μ_{AML} o μ_{ALL} respectivamente, en el primer paciente. `test.gene.matrix[1,2]` contiene el nivel de expresión del paciente 1 en el gen 2, `mu.aml[2]` y `mu.aml[2]` contiene los valores promedios para ese gen. `abs()` calcula el valor absoluto. Estos dos cálculos se almacenan en la primera posición de los vectores `dist.from.aml` y `dist.from.all`.

En general, la secuencia es calcular las distancias de un gen para todos los pacientes (i de 1 a 10) y después cambiamos al siguiente gen. Cada vez que cambiamos de gen los vectores `dist.from.aml` y `dist.from.all` se re-calculan.

c) Observa los resultados de la primera fila de la matriz `votes`, del primer paciente. ¿Cuáles son los genes para este paciente que votan por el tipo AML y cuales votan por el tipo ALL?

```
# la primera fila de votes
votes[1, ]
[1] 1 0 0 1 1 0 0 1 0 0
# nombre de los genes que votan AML
colnames(test.gene.matrix)[votes[1, ] == 0]
[1] "M27891_at"   "M11147_at"      "M14483_rna1_s_at" "X82240_rna1_at"
[5] "M33680_at"        "U05259_rna1_at"
# nombre de los genes que votan ALL
colnames(test.gene.matrix)[votes[1, ] == 1]
[1] "M19507_at"      "M96326_rna1_at" "Y00787_s_at"    "X58529_at"
```

Los genes 2, 3, 6, 7, 9 y 10 votan AML (votes=0), mientras que los genes 1, 4, 5 y 8 votan ALL (votes=1). Los nombres de los genes los tenemos en las columnas de la matriz test.gene.matrix.

9. Calcula $v_1,\ldots, v_{10}$, como las desviaciones del punto medio entre las dos medias. El siguiente código almacenan estas desviaciones en una matriz llamada deviation.magnitude.

```
deviation.magnitude <- matrix(nrow = num.patients, ncol = num.genes)
for(i in 1:dim(deviation.magnitude)[1]){
   for(j in 1:dim(deviation.magnitude)[2]) {
      midpoint <- (mu.aml - mu.all)/2
      deviation.magnitude[i,j] <- abs(test.gene.matrix[i, j] -
midpoint[j])
   }
}
# añadimos los genes como nombre de columna
colnames(deviation.magnitude) <- colnames(test.gene.matrix)
```

a) ¿Cuál es la desviación del gen M19507_at para el paciente 1? ¿Qué paciente es el que tiene una desviación mayor en este gen?

```
# deviation.magnitude para el paciente 1 en el gen M19507_at
deviation.magnitude[1, "M19507_at"]
M19507_at
5429.313
# desviación mayor en el gen M19507_at, ¿qué paciente?
which.max(deviation.magnitude[ ,"M19507_at"])
[1] 3
max(deviation.magnitude[ ,"M19507_at"]) # cuanto se desvía
[1] 11383.1
```

b) Para el paciente 1, ¿qué desviaciones deberían sumarse para calcular el voto total para el tipo AML?

El valor del voto total para el tipo AML seria la suma de las desviaciones de genes que han votado por este tipo. Para el paciente 1 las desviaciones de los genes 2, 3, 6, 7, 9, y 10 (apartado 8c).

10. Calcular V_{AML} y V_{ALL} para cada paciente del grupo test como la suma de las desviaciones de los genes que votan por cada uno de los tipos. Determinamos el tipo de cada paciente como el tipo cuyo voto sea mayor. El código es el siguiente,

```
# sumamos las desviaciones que correspondan
v.aml <- vector("numeric", num.patients)
v.all <- vector("numeric", num.patients)
for(i in 1:num.patients){
   v.aml[i] <- sum(deviation.magnitude[i, which(votes[i,] == 0)])
```

```
   V.all[i] <- sum(deviation.magnitude[i, which(votes[i,] == 1)])
}
# el tipo de leucemia para cada paciente
predicted.leuk.type <- vector("numeric", num.patients)
for(i in 1:num.patients){
   if (V.aml[i] > V.all[i]){predicted.leuk.type[i] = "aml"}
   if (V.aml[i] < V.all[i]){predicted.leuk.type[i] = "all"}
   if (V.aml[i] == V.all[i]){predicted.leuk.type[i] = "tie"}
}
predicted.leuk.type
[1] "aml" "aml" "aml" "all" "aml" "all" "all" "all" "all" "all"
```

a) Describe brevemente como el primer bucle `for()` crea los vectores `V.aml` y V.all.

Para cada paciente, el bucle crea cada uno de estos vectores sumando las desviaciones correspondientes a los genes que han votado por el grupo correspondiente. Por ejemplo, cuando i=1 V.aml se calcula sumando las desviaciones de la primera fila en los genes en los que hay un 0 en la primera fila de la matriz votes.

b) Describe brevemente como el segundo bucle `for()` y la sentencia condicional `if()` crea el vector `predicted.leuk.type.`

Para cada uno de los pacientes (`i` en `for`) la sentencia `if()` compara `V.aml` y `V.all` devolviendo un valor diferente dependiendo del resultado de la compasión. Si `V.aml` es menor que `V.all`, la predicción es ALL. Si `V.aml` es mayor que `V.all` entonces la predicción es AML. Si los dos valores son iguales no podemos hacer la clasificación.

11. Comprueba la precisión de la predicción del tipo de leucemia comparando dicha predicción con el tipo real almacenado en la variable `cancer`. Da alguna medida que permita evaluar como de buena es esta predicción.

```
# comparamos la predicción con los tipos reales
table(golub.test$cancer, predicted.leuk.type)
      predicted.leuk.type
       all aml
  allB   5   0
  allT   0   0
  aml    1   4
```

Sólo 1 paciente del tipo aml está mal clasificado. El porcentaje de bien clasificados es del 90% en general, el 100% para el tipo allB y del 80% para el tipo aml. En esta muestra ninguno de los 10 pacientes es del tipo allT por lo que no podemos valorar la precisión en este grupo de pacientes.