Ejercicio1

Marta Verona Almeida

1. Comprobar la dimensión y los nombres de las columnas del dataset. ¿Qué dimensión tiene? ¿qué datos alberga?

```
dim(hip)
colnames(hip)
sapply(hip,class)
```

2. Muestra por pantalla la columna de la variable RA

A continuación, se muestra por pantalla los primeros seis valores de la variable RA.

```
RA <- hip$RA
head(RA)
```

```
## [1] 0.003797 0.111047 0.135192 0.151656 0.221873 0.243864
```

Para mostrar el listado completo de valores, bastaría con ejecutar:

```
hip$RA
# ó
RA
```

3. Calcula las tendencias centrales de todos los datos del dataset (mean, media) utilizando la function apply

```
apply(hip, MARGIN = 2, mean, na.rm = TRUE)
```

```
##           HIP          Vmag            RA            DE           Plx
## 56549.4828981     8.2593858   173.4529975    -0.1397663    22.1980213
##          pmRA          pmDE         e_Plx           B.V
##     5.3761346   -63.9419934     1.6267929     0.7615299
```

```
apply(hip, MARGIN = 2, median, na.rm = TRUE)
```

```
##          HIP         Vmag           RA           DE          Plx
## 56413.000000     8.280000   173.369788     3.254234    22.100000
##         pmRA         pmDE        e_Plx          B.V
##    10.550000   -49.480000     1.140000     0.710500
```

4. Haz lo mismo para las medidas de dispersión mínimo y máximo. ¿Seria posible hacerlo con un único comando?¿Que hace la función range()?

Es posible calcular estas medidas de dispersión utilizando la función *range*.

```
apply(hip, 2, range, na.rm = TRUE)
```

```
##         HIP  Vmag         RA        DE Plx    pmRA     pmDE e_Plx    B.V
## [1,]      2  0.45   0.003797 -87.20273  20 -868.01 -1392.30  0.45 -0.158
## [2,] 120003 12.74 359.954685  88.30268  25  781.34   481.19 46.91  2.800
```

5. Sin embargo las medidas mas populares de dispersión son la varianza (var()), su desviación standard (sd()) y la desviación absoluta de la mediana o MAD. Calcula estas medidas para los valores de RA

```
  var(RA, na.rm = TRUE)
```

```
## [1] 11566.32
```

```
  sd(RA, na.rm = TRUE)
```

```
## [1] 107.5468
```

```
  mad(hip$RA)
```

```
## [1] 146.9334
```

6. Imagina que quieres calcular dos de estos valores de una sola vez. ¿Te serviría este código? ¿Cuál sería el resultado de aplicar apply(hip,2,f)?

Efectovamente, el siguiente código calcula la mediana y la desviación absoluta de la mediana para la variable de entrada.

```
  f = function(x) c(median(x), mad(x))
  f(hip[,3])
```

```
## [1] 173.3698 146.9334
```

Al utilizar la función *apply*, se calculan estos valores para cada una de las columnas de hip.

```
  apply(hip,2,f)
```

```
##           HIP     Vmag       RA        DE       Plx     pmRA      pmDE
## [1,] 56413.00 8.280000 173.3698  3.254234 22.100000  10.5500 -49.48000
## [2,] 49090.37 1.882902 146.9334 43.984032  1.764294 141.6476  99.49729
##         e_Plx B.V
## [1,] 1.140000  NA
## [2,] 0.489258  NA
```

7. Vamos a medir la dispersión de la muestra utilizando el concepto de cuartiles. El percentil 90 es aquel dato que excede en un 10% a todos los demás datos. El cuartil (quantile) es el mismo concento, solo que habla de proporciones en vez de porcentajes. De forma que el percentil 90 es lo mismo que el cuartil 0.90. La mediana "median" de un dataset es el valor más central, en otras palabras exactamente la mitad del dataset excede la media. Calcula el cuartil .10 y .50 para la columna RA del dataset hip. Sugerencia: quantile()

```
  quantile(RA, .10)
```

```
##      10%
## 28.92324
```

```
quantile(RA, .50)
```

```
##      50%
## 173.3698
```

8. Los cuantiles 0.25 y 0.75 se conocen como el first quartile y el third quartile, respectivamente. Calcula los cuatro cuartiles para RA con un único comando.

```
quantile(RA, probs = seq(0.25,1,0.25))
```

```
##       25%       50%       75%      100%
##  70.14137 173.36979 266.92332 359.95468
```

9. Otra medida de dispersion es la diferencia entre el primer y el tercer cuartil conocida como rango intercuartil (IQR) Inter Quantile Range. ¿Obtienes ese valor con la función summary()?

```
summary(hip)
```

```
##       HIP              Vmag              RA                DE
##  Min.   :     2   Min.   : 0.450   Min.   :  0.0038   Min.   :-87.2027
##  1st Qu.: 21770   1st Qu.: 7.050   1st Qu.: 70.1414   1st Qu.:-31.3635
##  Median : 56413   Median : 8.280   Median :173.3698   Median :  3.2542
##  Mean   : 56549   Mean   : 8.259   Mean   :173.4530   Mean   : -0.1398
##  3rd Qu.: 87096   3rd Qu.: 9.610   3rd Qu.:266.9233   3rd Qu.: 28.0705
##  Max.   :120003   Max.   :12.740   Max.   :359.9547   Max.   : 88.3027
##
##       Plx             pmRA              pmDE             e_Plx
##  Min.   :20.00   Min.   :-868.010   Min.   :-1392.30   Min.   : 0.450
##  1st Qu.:20.98   1st Qu.: -91.980   1st Qu.: -130.79   1st Qu.: 0.870
##  Median :22.10   Median :  10.550   Median :  -49.48   Median : 1.140
##  Mean   :22.20   Mean   :   5.376   Mean   :  -63.94   Mean   : 1.627
##  3rd Qu.:23.36   3rd Qu.: 103.870   3rd Qu.:    8.57   3rd Qu.: 1.680
##  Max.   :25.00   Max.   : 781.340   Max.   :  481.19   Max.   :46.910
##
##       B.V
##  Min.   :-0.1580
##  1st Qu.: 0.5600
##  Median : 0.7105
##  Mean   : 0.7615
##  3rd Qu.: 0.9530
##  Max.   : 2.8000
##  NA's   :41
```

Como puede verse, no se obtiene el valor del IQR directamente. Sin embargo, es posible calcularlo utilizando el siguiente comando.

```
IQR(RA)
```

```
## [1] 196.782
```

10. Hasta ahora has ignorado la presencia de valores perdidos NA. La función any() devuelve TRUE si se encuentra al menos un TRUE en el vector que damos como argumento. Su combinación con is.na es muy útil. ¿qué obtienes cuando ejecutas el siguiente comando? ¿Cómo lo interpretas?

```
hasNA = function(x) any(is.na(x))
apply(hip,2,hasNA)
```

```
##   HIP  Vmag    RA    DE   Plx  pmRA  pmDE e_Plx   B.V
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
```

El resultado obtenido es un booleano que indica la presencia de elementos nulos en cada variable. En este caso, la única que presenta este tipo de valores es "B.V".

11. Prueba a ejecutar el siguiente comando.

```
min(hip$B.V)
```

```
## [1] NA
```

Podemos observar que la función *min* no es capaz de tratar con los valores ausentes.

12. Como has observado nos devuelve NA para toda la columna, normalmente querríamos poder usar la función sobre el resto de datos que no son NA: Para ello podemos utilizar la función na.omit. ¿Que ocurre cuando lo hacemos?. Usando apply calcula la media para hip y hip1. Intenta calcular la media de forma que solo cambie la de B.V cuando ignores los valores NA.

Al utilizar la función na.omit se eliminan las filas en las que se encuentre algún valor nulo.

```
hip1<- na.omit(hip)
dim(hip1)
```

```
## [1] 2678    9
```

A continuación, calculamos la media de hip y hip1 para cada variable.

```
apply(hip, MARGIN = 2, mean)
```

```
##           HIP          Vmag            RA            DE           Plx
## 56549.4828981     8.2593858   173.4529975    -0.1397663    22.1980213
##          pmRA          pmDE         e_Plx           B.V
##     5.3761346   -63.9419934     1.6267929            NA
```

```
apply(hip1, MARGIN = 2, mean)
```

```
##           HIP          Vmag            RA            DE           Plx
## 56575.8050784     8.2147797   173.5284087    -0.2743560    22.1954033
##          pmRA          pmDE         e_Plx           B.V
##     5.5370575   -63.5345892     1.5449552     0.7615299
```

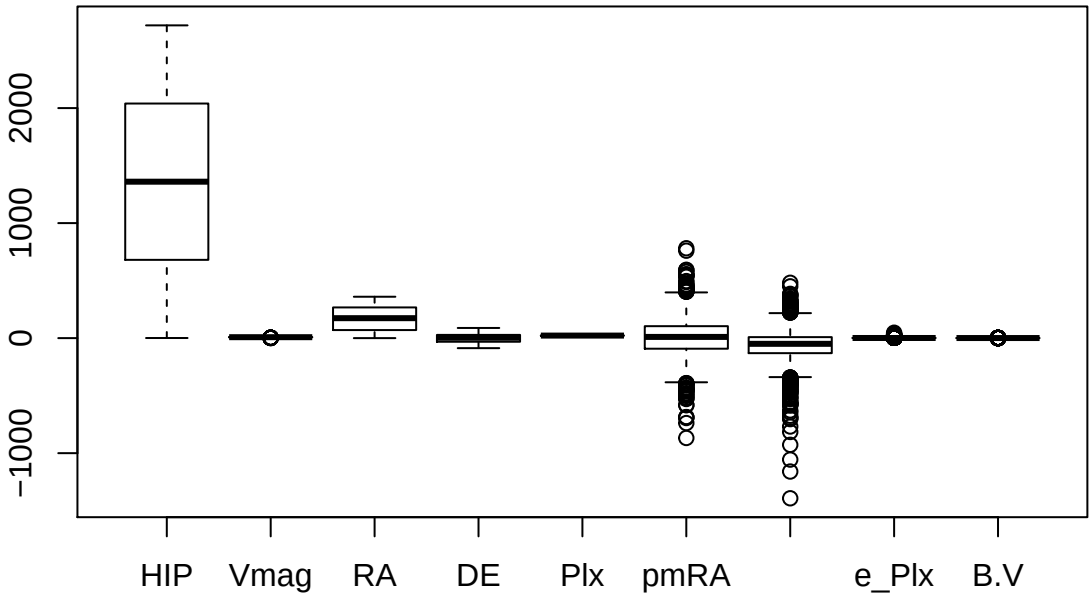
Por último, utilizando el parámetro **na.rm = TRUE** obtenemos la media de manea que solo cambia la de B.V, puesto que se ignoran los valores nulos.

```r
apply(hip, MARGIN = 2, mean, na.rm = TRUE)
```

```
##           HIP          Vmag            RA           DE          Plx
## 56549.4828981     8.2593858   173.4529975    -0.1397663    22.1980213
##          pmRA          pmDE         e_Plx          B.V
##     5.3761346   -63.9419934     1.6267929     0.7615299
```

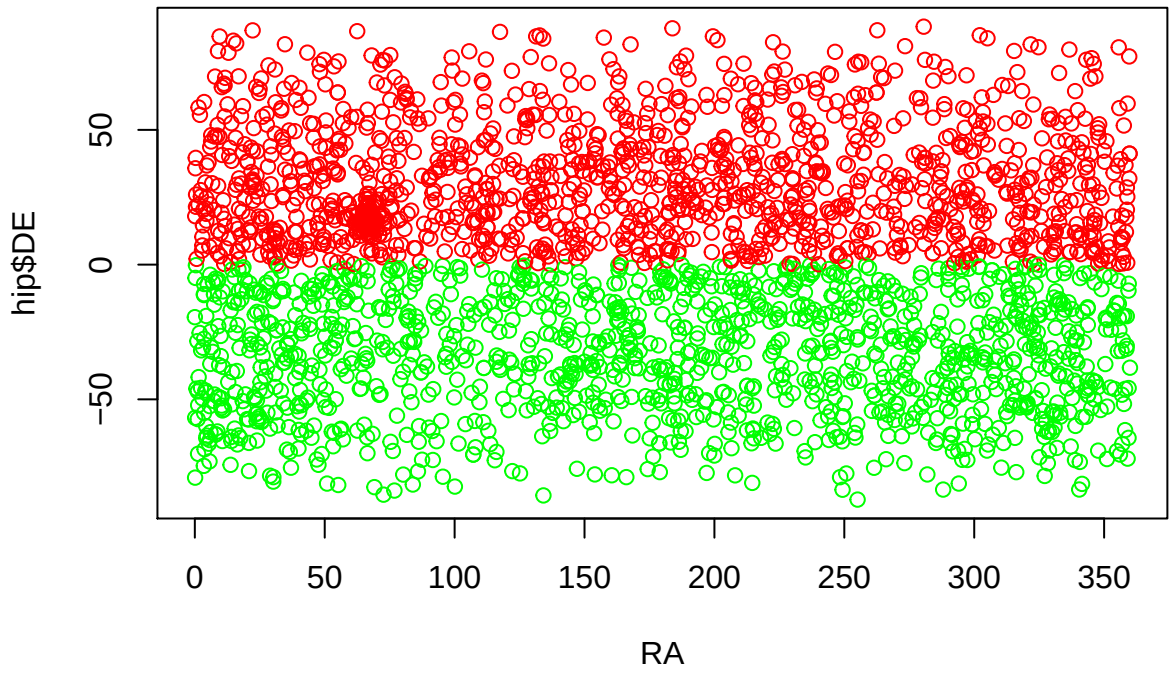
13. Obten una idea aproximada de tus datos mediante la creación de un boxplot del hop dataset

```r
hip$HIP <- as.factor(hip$HIP)
boxplot(hip)
```



14. Crea un scatterplot que te compare los valores de RA y DE. Representa los puntos con el símbolo '.' Y que estos puntos sean de color rojo si DE excede de 0. Sugerencia ifelse()

```r
plot(RA, hip$DE, type = "p", col=ifelse(hip$DE>0,"red","green"))
```



15. Haz un scatterplot de RA y pmRA. ¿Ves algún patrón?

```
plot(RA, hip$pmRA, type = "p", col="purple")
```