Ejercicios Introducción a la ciencia de datos - Mª Cristina Heredia Gómez

1. Exploratory Data Analysis

a. Ejemplo 1, hip dataset

*** Descargate el dataset hip con el siguiente commando**

```
hip  <-read.table("http://astrostatistics.psu.edu/datasets/HIP_star.dat", header=T,fill=T)
```

*** Una vez descargado comprueba la dimensión y los nombres de las columnas del dataset. ¿Qué dimensión tiene? ¿qué datos alberga?**

Comprobamos la dimensión del dataset con `nrow` y `ncol`:

```
ncol(hip)
```

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

```
nrow(hip)
```

```
## [1] 2719
```

La dimensión del dataset es de 9 columnas y 2719 filas. El dataset alberga datos numéricos, casi todos tipo double pero la columna HIP

*** Muestra por pantalla la columna de la variable RA**

```
head(hip$RA, 110)
```

```
##   [1]  0.003797  0.111047  0.135192  0.151656  0.221873  0.243864  0.348708
##   [8]  0.426746  0.455182  0.478685  0.612287  0.696411  0.972063  1.099309
##  [15]  1.102623  1.244275  1.281668  1.369764  1.423333  1.468617  1.843365
##  [22]  1.966150  2.261459  2.315143  2.352249  2.431558  2.768701  2.878592
##  [29]  2.898287  2.906145  3.125492  3.136756  3.287636  3.470117  3.499989
##  [36]  3.517613  3.542473  3.561294  3.584257  3.588865  3.768990  4.047722
##  [43]  4.101227  4.179702  4.342874  4.375348  4.376308  4.386210  4.387165
##  [50]  4.401851  4.457699  4.582106  4.608357  4.901928  5.280108  5.398303
##  [57]  5.604551  5.644953  5.735995  5.750338  5.770342  5.907064  5.913140
##  [64]  5.947756  6.036948  6.091006  6.256391  6.266819  6.285408  6.667964
##  [71]  6.930394  6.957396  7.292119  7.368555  7.427643  7.465869  7.575263
##  [78]  7.608228  7.689831  7.801700  7.885670  8.033768  8.182476  8.613389
##  [85]  8.622360  8.712882  8.739422  8.807178  8.908298  8.967503  8.970857
##  [92]  9.007174  9.016931  9.028761  9.156804  9.342176  9.487066  9.561303
##  [99]  9.638220  9.803365  9.919210  9.934979  9.989965 10.182398 10.214448
## [106] 10.281968 10.313306 10.393825 10.505973 10.617905
```

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

```
apply(hip,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
```

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

Con apply:

```
hip.mins<- apply(hip,2,min)
hip.mins
```

```
##          HIP         Vmag           RA           DE          Plx
##     2.000000     0.450000     0.003797   -87.202730    20.000000
##         pmRA         pmDE        e_Plx          B.V
##  -868.010000 -1392.300000     0.450000           NA
```

```
hip.max<- apply(hip,2,max)
hip.max
```

```
##          HIP         Vmag           RA           DE          Plx
## 120003.00000     12.74000    359.95468     88.30268     25.00000
##         pmRA         pmDE        e_Plx          B.V
##    781.34000    481.19000     46.91000           NA
```

Range devuelve un vector que contiene el mínimo y el máximo de un argumento dado. Por lo que podríamos sacar el mínimo y el máximo con un único comando usando **range**:

```
hip.min.max<-apply(hip,2,range)
hip.min.max
```

```
##         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  NA
## [2,] 120003 12.74 359.954685  88.30268  25  781.34   481.19 46.91  NA
```

*** 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**

```
hip.var<-var(hip$RA)
hip.var
```

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

```
hip.sd<-sd(hip$RA)
hip.sd
```

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

```
hip.mad<-mad(hip$RA)
hip.mad
```

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

*** Imagina que quieres calcular dos de estos valores de una sola vez. ¿Te serviría este código?**

```
f = function(x) c(median(x), mad(x))
f(hip[,1])
```

```
## [1] 56413.00 49090.37
```

No exactamente, ya que en el apartado anterior no lo hacíamos sobre la priemra columna del dataset ni calculábamos su media. Pero si serviría ese código un poco modificado, por ejemplo:

```
f = function(x) c(sd(x), mad(x))
f(hip$RA)
```

```
## [1] 107.5468 146.9334
```

Sí nos calcularía simuláneamente la desviación estándar y la desviación absoluta de la mediana de la columna **RA**, tal como hacíamos en el apartado anterior.

*** ¿Cuál sería el resultado de aplicar apply(hip,2,f)?**

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

```
##           HIP     Vmag       RA       DE      Plx     pmRA      pmDE
## [1,] 35587.31 1.884730 107.5468 38.93039 1.417193 160.9799 140.89042
## [2,] 49090.37 1.882902 146.9334 43.98403 1.764294 141.6476  99.49729
##         e_Plx B.V
## [1,] 2.212867  NA
## [2,] 0.489258  NA
```

Aplica '**f** a cada columna del dataset hip, obteniendo así la desviación estándar y desviación absoluta de la mediana para cada columna.

*** 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()**