Proyecto R CIFF 2015 - Alfonso Campos

Proyecto I

Configuracion

```
# Variables de entorno
# setwd("D:\\Dropbox\\Doc\\Actual\\CIFF\\R\\Entregas") # MODIFICAR!
getwd()
```

```
## [1] "D:/Dropbox/Doc/Actual/CIFF/R/Entregas"
```

```
# Paquetes
# install.packages("caTools") # INSTALAR!
library(caTools) # Permite utilizar la funcion sample.split
```

```
## Warning: package 'caTools' was built under R version 3.1.3
```

```
# Semilla
set.seed(2000)
```

Cargar los datos en R.

```
housing = read.table("housing")
```

Realizar un analisis estadistico de las variables: calcular la media, varianza, rangos, etc. ¿Tienen las distintas variables rangos muy diferentes?

```
# Media y rangos
summary(housing)
```

```
##        V1              V2                V3                V4         
##  Min.   : 5.00   Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.0000  
##  1st Qu.:17.02   1st Qu.:-0.9983   1st Qu.:-1.0000   1st Qu.:-0.6532  
##  Median :21.20   Median :-0.9944   Median :-1.0000   Median :-0.3233  
##  Mean   :22.53   Mean   :-0.9189   Mean   :-0.7727   Mean   :-0.2172  
##  3rd Qu.:25.00   3rd Qu.:-0.9175   3rd Qu.:-0.7500   3rd Qu.: 0.2933  
##  Max.   :50.00   Max.   : 1.0000   Max.   : 1.0000   Max.   : 1.0000  
##        V5                V6                 V7                 V8          
##  Min.   :-1.0000   Min.   :-1.00000   Min.   :-1.00000   Min.   :-1.0000  
##  1st Qu.:-1.0000   1st Qu.:-0.73663   1st Qu.:-0.10922   1st Qu.:-0.1323  
##  Median :-1.0000   Median :-0.37037   Median : 0.01456   Median : 0.5366  
##  Mean   :-0.8617   Mean   :-0.30167   Mean   : 0.04374   Mean   : 0.3527  
##  3rd Qu.:-1.0000   3rd Qu.:-0.01646   3rd Qu.: 0.17360   3rd Qu.: 0.8780  
##  Max.   : 1.0000   Max.   : 1.00000   Max.   : 1.00000   Max.   : 1.0000
```

```
##        V9                V10               V11               V12
##  Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.00000
##  1st Qu.:-0.8235   1st Qu.:-0.7391   1st Qu.:-0.6489   1st Qu.: 0.02128
##  Median :-0.6221   Median :-0.6522   Median :-0.4542   Median : 0.37234
##  Mean   :-0.5152   Mean   :-0.2566   Mean   :-0.1556   Mean   : 0.24586
##  3rd Qu.:-0.2618   3rd Qu.: 1.0000   3rd Qu.: 0.8282   3rd Qu.: 0.61702
##  Max.   : 1.0000   Max.   : 1.0000   Max.   : 1.0000   Max.   : 1.00000
##       V13               V14
##  Min.   :-1.0000   Min.   :-1.0000
##  1st Qu.: 0.8915   1st Qu.:-0.7119
##  Median : 0.9725   Median :-0.4685
##  Mean   : 0.7971   Mean   :-0.3972
##  3rd Qu.: 0.9966   3rd Qu.:-0.1598
##  Max.   : 1.0000   Max.   : 1.0000
```

```
# Varianza
apply(housing, 2, var)
```

```
##          V1          V2          V3          V4          V5          V6
## 84.58672359  0.03738755  0.21757473  0.25296715  0.25805189  0.22739816
##          V7          V8          V9         V10         V11         V12
##  0.07249747  0.33615782  0.14666168  0.57328053  0.41379799  0.21217686
##         V13         V14
##  0.21197814  0.15531348
```

Podemos ver que las Variables V2 a V14 estan normalizadas. La variable V1 no lo esta! Los rangos de las variables V2 a V14 son similares (-1 a 1), V1 tiene un rango mayor (5 a 50).

Escalar los datos para que tengan media 0 y varianza 1, es decir, restar a cada variable su media y dividir por la desviacion tipica.

```
# Antes de escalar
colMeans(housing)
```

```
##          V1          V2          V3          V4          V5          V6
## 22.53280632 -0.91891182 -0.77272727 -0.21724497 -0.86166008 -0.30166642
##          V7          V8          V9         V10         V11         V12
##  0.04373805  0.35272711 -0.51523744 -0.25657317 -0.15558347  0.24585816
##         V13         V14
##  0.79713566 -0.39718195
```

```
apply(housing, 2, var)
```

```
##          V1          V2          V3          V4          V5          V6
## 84.58672359  0.03738755  0.21757473  0.25296715  0.25805189  0.22739816
##          V7          V8          V9         V10         V11         V12
##  0.07249747  0.33615782  0.14666168  0.57328053  0.41379799  0.21217686
##         V13         V14
##  0.21197814  0.15531348
```

```r
# Escalado
scaledHousing <- scale(housing)

scaledHousing <- data.frame(lapply(housing, function(x) scale(x)))

# Despues de escalar
colMeans(scaledHousing) # estos valores son 0
```

```
##            V1            V2            V3            V4            V5
## -1.379311e-16 -6.240205e-17 -4.667983e-17  1.952764e-17  4.816086e-17
##            V6            V7            V8            V9           V10
## -2.951087e-17  7.329721e-18  3.871245e-17  5.641622e-17 -7.953673e-17
##           V11           V12           V13           V14
## -4.059116e-17  1.590735e-17  1.256543e-16 -1.971962e-17
```

```r
apply(scaledHousing, 2, var) # la varianza es 1
```

```
##  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1
```

Podemos comprobar que hemos escalado correctamente los datos.

La variable de respuesta se encuentra en la primera columna, separarla del resto y calcular la correlacion de dicha variable con el resto.

```r
cor(scaledHousing[,1], scaledHousing[,2:14])
```

```
##              V2        V3         V4        V5         V6      V7
## [1,] -0.3883046 0.3604453 -0.4837251 0.1752602 -0.4273208 0.69536
##              V8        V9        V10        V11        V12       V13
## [1,] -0.3769546 0.2499287 -0.3816263 -0.4685359 -0.5077867 0.3334608
##             V14
## [1,] -0.7376627
```

Podemos ver que existen variables con mayor grado de correlacion positiva o negativa que otras. La variable V14 tiene la mayor correlacion, mientras que la variable V5 tiene la menor (hablando en valores absolutos).

Separar el conjunto de datos en dos, el primero (entrenamiento) conteniendo un 80% de los datos y el segundo (test) un 20%, de forma aleatoria.

```r
spl = sample.split(scaledHousing$V1, SplitRatio = 0.8)
scaledHousingTrain = subset(housing, spl==TRUE)
scaledHousingTest = subset(housing, spl==FALSE)
```

Proyecto II

Realizar un modelo de regresion lineal de la variable de respuesta sobre el resto y ajustarlo por minimos cuadrados usando unicamente los datos del conjunto de entrenamiento.

```
housingModel <- lm(V1 ~ ., data = scaledHousingTrain)
summary(housingModel)
```

```
##
## Call:
## lm(formula = V1 ~ ., data = scaledHousingTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4053  -2.8795  -0.5853   1.8236  26.5386
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.1040     1.6111   7.513 3.51e-13 ***
## V2           -4.9059     1.4874  -3.298 0.001056 **
## V3            2.0795     0.7410   2.806 0.005245 **
## V4            0.6818     0.8782   0.776 0.437990
## V5            1.7000     0.4550   3.736 0.000213 ***
## V6           -4.7640     1.0157  -4.690 3.69e-06 ***
## V7           10.0721     1.1556   8.716  < 2e-16 ***
## V8           -0.1854     0.6868  -0.270 0.787293
## V9           -7.7274     1.1908  -6.489 2.43e-10 ***
## V10           3.5273     0.8191   4.306 2.07e-05 ***
## V11          -3.0670     1.0460  -2.932 0.003551 **
## V12          -4.7976     0.6653  -7.212 2.60e-12 ***
## V13           1.6498     0.5768   2.860 0.004445 **
## V14          -9.2921     0.9597  -9.682  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.717 on 420 degrees of freedom
## Multiple R-squared:  0.7578, Adjusted R-squared:  0.7503
## F-statistic: 101.1 on 13 and 420 DF,  p-value: < 2.2e-16
```

Podemos ver que V4 y V8 no son significativas en nuestro modelo, cabe esperar que suframos algo de sobreajuste.

Calcular el error cuadratico medio de los datos del conjunto de entrenamiento y de los datos del conjunto de test.

```
# Datos de entrenamiento
housingModelPredTrain <- predict.lm(housingModel, newdata=scaledHousingTrain)
mseTrain <- mean((housingModelPredTrain - scaledHousingTrain$V1)^2)
mseTrain
```

```
## [1] 21.534
```

```r
# Datos de test
housingModelPredTest <- predict.lm(housingModel, newdata=scaledHousingTest)
mseTest <- mean((housingModelPredTest - scaledHousingTest$V1)^2)
mseTest
```

```
## [1] 24.68927
```

Los datos obtenidos indican que el error sobre los datos de test es ligeramente mayor que sobre los datos de entrenamiento, lo cual parece razonable maxime teniendo en cuenta algo de sobreajuste del modelo.