

# Trabajo Práctico Final - TAA

Adrian Pablo Cafa

7/12/2019

## # Ejercicio 1 - Aprendizaje Supervisado

En este ejercicio se pide comparar 2 modelos de predicción sobre los movimientos de un robot. Primero se realiza un Análisis Exploratorio de los Datos para entender la base, luego se particiona la base en un conjunto de entrenamiento y uno de testeo, después con estos conjuntos se realiza un modelo de SVM y un modelo de Naive Bayes (con la librería e1071) y finalmente se comparan las mediciones de la performance de ambos modelos (con la librería caret).

### Parte A - Análisis Exploratorio de los Datos

1) Abra la base sensor\_readings\_24.data del repositorio UCI en la página <https://archive.ics.uci.edu/ml/datasets/Wall-Following+Robot+Navigation+Data> Indique de qué trata el problema

El dataset contiene los valores brutos de las mediciones de 24 sensores de ultrasonido y la etiqueta de clase correspondiente. Las lecturas del sensor se muestrean en una tasa de 9 muestras por segundo.

2) Renombre el archivo sensor\_readings\_24.data como sensor\_readings\_24.data.csv y ábralo en R como "base"

```
base=read.csv("sensor_readings_24.data.csv", sep=",", header=FALSE)
```

```
base=read.csv("sensor_readings_24.data.csv", sep=",", header=FALSE)
dim(base)
```

```
## [1] 5456 25
```

3) Muestre un head de la base

```
head(base)
```

```
##      V1      V2      V3      V4 V5      V6 V7      V8      V9      V10      V11      V12      V13      V14
## 1 0.438 0.498 3.625 3.645 5 2.918 5 2.351 2.332 2.643 1.698 1.687 1.698 1.717
## 2 0.438 0.498 3.625 3.648 5 2.918 5 2.637 2.332 2.649 1.695 1.687 1.695 1.720
## 3 0.438 0.498 3.625 3.629 5 2.918 5 2.637 2.334 2.643 1.696 1.687 1.695 1.717
## 4 0.437 0.501 3.625 3.626 5 2.918 5 2.353 2.334 2.642 1.730 1.687 1.695 1.717
## 5 0.438 0.498 3.626 3.629 5 2.918 5 2.640 2.334 2.639 1.696 1.687 1.695 1.717
## 6 0.439 0.498 3.626 3.629 5 2.918 5 2.633 2.334 2.645 1.705 1.686 1.694 1.719
##      V15      V16      V17      V18      V19      V20      V21      V22      V23      V24
## 1 1.744 0.593 0.502 0.493 0.504 0.445 0.431 0.444 0.440 0.429 Slight-Right-Turn
## 2 1.744 0.592 0.502 0.493 0.504 0.449 0.431 0.444 0.443 0.429 Slight-Right-Turn
## 3 1.744 0.593 0.502 0.493 0.504 0.449 0.431 0.444 0.446 0.429 Slight-Right-Turn
## 4 1.744 0.593 0.502 0.493 0.504 0.449 0.431 0.444 0.444 0.429 Slight-Right-Turn
## 5 1.744 0.592 0.502 0.493 0.504 0.449 0.431 0.444 0.441 0.429 Slight-Right-Turn
## 6 1.744 0.589 0.502 0.493 0.504 0.446 0.431 0.444 0.444 0.430 Slight-Right-Turn
```

4) Renombre la variable categórica V25 como "Mov"  
`names(base)[names(base)=="V25"]="Mov"`  
 Muestre un head de la base con la variable renombrada.

```
names(base)[names(base)=="V25"]="Mov"
head(base)
```

```
##      V1      V2      V3      V4 V5      V6 V7      V8      V9      V10      V11      V12      V13      V14
## 1 0.438 0.498 3.625 3.645 5 2.918 5 2.351 2.332 2.643 1.698 1.687 1.698 1.717
## 2 0.438 0.498 3.625 3.648 5 2.918 5 2.637 2.332 2.649 1.695 1.687 1.695 1.720
## 3 0.438 0.498 3.625 3.629 5 2.918 5 2.637 2.334 2.643 1.696 1.687 1.695 1.717
## 4 0.437 0.501 3.625 3.626 5 2.918 5 2.353 2.334 2.642 1.730 1.687 1.695 1.717
## 5 0.438 0.498 3.626 3.629 5 2.918 5 2.640 2.334 2.639 1.696 1.687 1.695 1.717
## 6 0.439 0.498 3.626 3.629 5 2.918 5 2.633 2.334 2.645 1.705 1.686 1.694 1.719
##      V15      V16      V17      V18      V19      V20      V21      V22      V23      V24      Mov
## 1 1.744 0.593 0.502 0.493 0.504 0.445 0.431 0.444 0.440 0.429 Slight-Right-Turn
## 2 1.744 0.592 0.502 0.493 0.504 0.449 0.431 0.444 0.443 0.429 Slight-Right-Turn
## 3 1.744 0.593 0.502 0.493 0.504 0.449 0.431 0.444 0.446 0.429 Slight-Right-Turn
## 4 1.744 0.593 0.502 0.493 0.504 0.449 0.431 0.444 0.444 0.429 Slight-Right-Turn
## 5 1.744 0.592 0.502 0.493 0.504 0.449 0.431 0.444 0.441 0.429 Slight-Right-Turn
## 6 1.744 0.589 0.502 0.493 0.504 0.446 0.431 0.444 0.444 0.430 Slight-Right-Turn
```

5) ¿Cuántas variables y registros tiene la base? ¿Cuántos registros por clase hay de la variable a predecir? ¿De qué tipo son las variables?

```
dim(base)
```

```
## [1] 5456    25
```

La base tiene 5456 registros o filas con 25 variables o columnas

Registros por clase de la variable a predecir:

```
summary(base$Mov)
```

```
##      Move-Forward  Sharp-Right-Turn  Slight-Left-Turn  Slight-Right-Turn
##              2205              2097              328              826
```

La única variable categórica es "Mov", variables discriminadas por tipo:

```
str(base)
```

```
## 'data.frame':    5456 obs. of  25 variables:
## $ V1 : num  0.438 0.438 0.438 0.437 0.438 0.439 0.44 0.444 0.451 0.458 ...
## $ V2 : num  0.498 0.498 0.498 0.501 0.498 ...
## $ V3 : num  3.62 3.62 3.62 3.62 3.63 ...
## $ V4 : num  3.64 3.65 3.63 3.63 3.63 ...
## $ V5 : num  5 5 5 5 5 5 5 5 5 5 ...
## $ V6 : num  2.92 2.92 2.92 2.92 2.92 ...
```

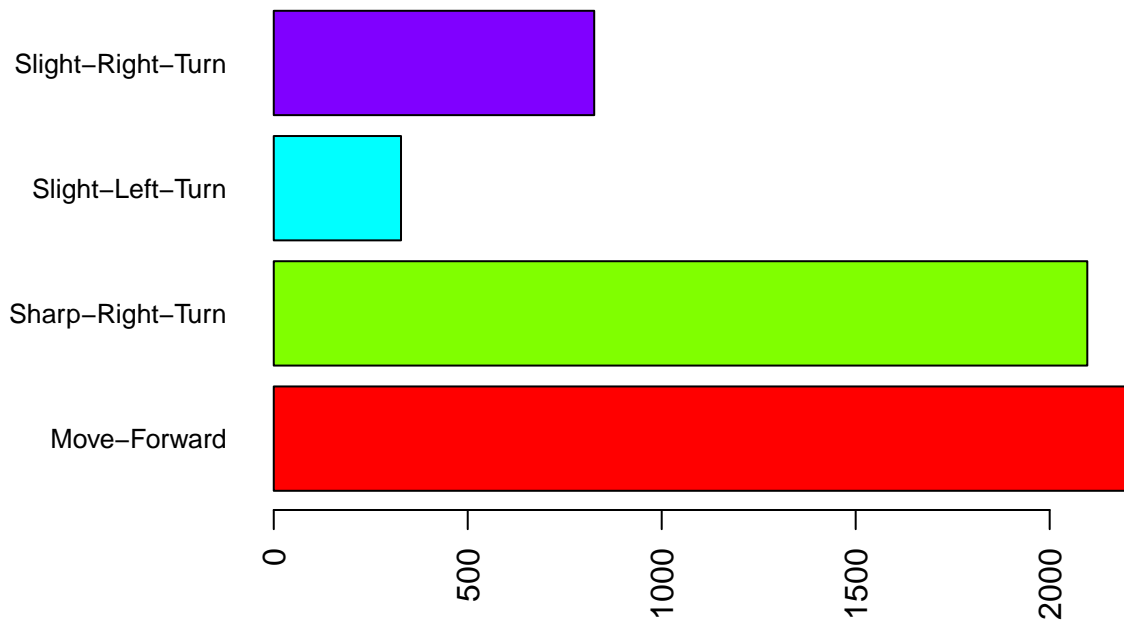
```
## $ V7 : num 5 5 5 5 5 ...
## $ V8 : num 2.35 2.64 2.64 2.35 2.64 ...
## $ V9 : num 2.33 2.33 2.33 2.33 2.33 ...
## $ V10: num 2.64 2.65 2.64 2.64 2.64 ...
## $ V11: num 1.7 1.7 1.7 1.73 1.7 ...
## $ V12: num 1.69 1.69 1.69 1.69 1.69 ...
## $ V13: num 1.7 1.7 1.7 1.7 1.7 ...
## $ V14: num 1.72 1.72 1.72 1.72 1.72 ...
## $ V15: num 1.74 1.74 1.74 1.74 1.74 ...
## $ V16: num 0.593 0.592 0.593 0.593 0.592 0.589 0.588 0.595 0.595 0.59 ...
## $ V17: num 0.502 0.502 0.502 0.502 0.502 0.502 0.501 0.5 0.499 0.496 ...
## $ V18: num 0.493 0.493 0.493 0.493 0.493 0.493 0.492 0.491 0.491 0.49 ...
## $ V19: num 0.504 0.504 0.504 0.504 0.504 0.504 0.504 0.504 0.503 0.502 0.498 ...
## $ V20: num 0.445 0.449 0.449 0.449 0.449 0.449 0.446 0.451 0.453 0.457 0.462 ...
## $ V21: num 0.431 0.431 0.431 0.431 0.431 0.431 0.431 0.433 0.436 0.44 0.444 ...
## $ V22: num 0.444 0.444 0.444 0.444 0.444 0.444 0.444 0.446 0.448 0.453 0.458 ...
## $ V23: num 0.44 0.443 0.446 0.444 0.441 0.444 0.444 0.444 0.454 0.461 ...
## $ V24: num 0.429 0.429 0.429 0.429 0.429 0.43 0.432 0.436 0.442 0.449 ...
## $ Mov: Factor w/ 4 levels "Move-Forward",...: 4 4 4 4 4 4 4 4 2 2 ...
```

6) Realice un gráfico de barras de la variable a predecir.

```
table(base$Mov)
```

```
##
##      Move-Forward  Sharp-Right-Turn  Slight-Left-Turn  Slight-Right-Turn
##              2205              2097              328              826
```

```
par(las=2) # make label text perpendicular to axis
par(mar=c(5,8,4,2)) # increase y-axis margin
plot(x = base$Mov,col = rainbow(length(unique(base$Mov))), horiz=TRUE,cex.names=0.8)
```

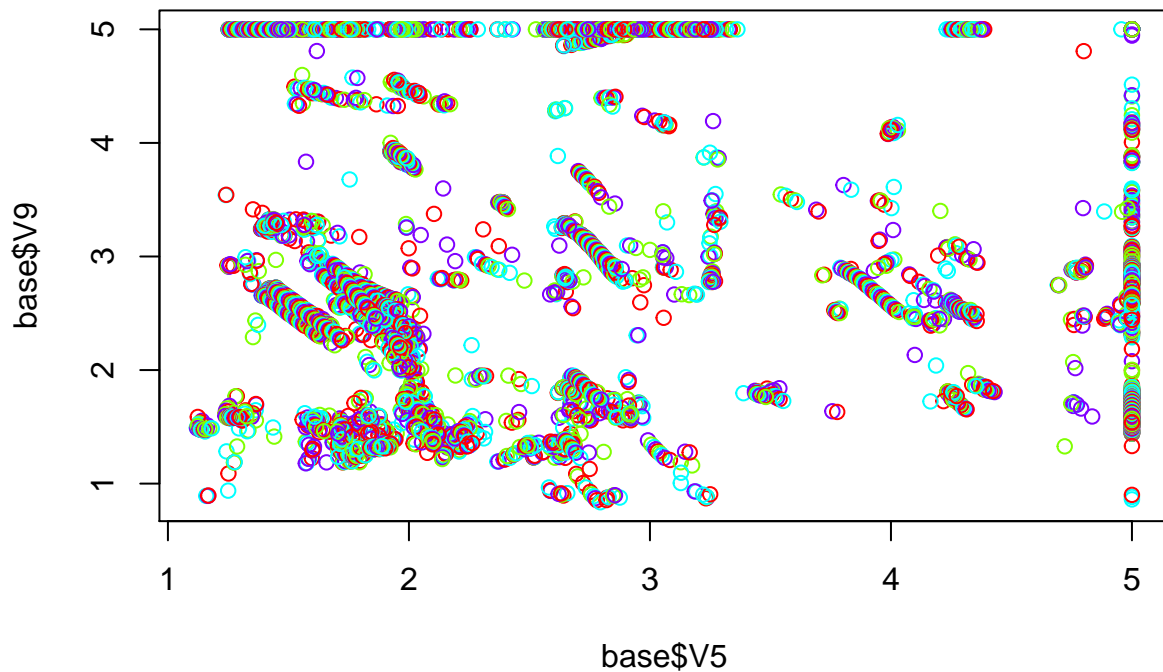


7) Realice un gráfico de dispersión con 2 variables, y coloréelo según la variable Mov

```
names(base)
```

```
## [1] "V1" "V2" "V3" "V4" "V5" "V6" "V7" "V8" "V9" "V10" "V11" "V12"
## [13] "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23" "V24"
## [25] "Mov"
```

```
plot(base$V5,base$V9,col = rainbow(length(unique(base$Mov))))
```



### #Parte B - Conjuntos

1) Cargue la librería caret, setee la semilla=101 y particione la base en un conjunto de entrenamiento y uno de testeo, utilizando la instrucción createDataPartition de la librería caret. Setee  $p=0.70$ . Indique el código R utilizado.

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
set.seed(101);particion=createDataPartition(y=base$Mov, p=0.7,list=FALSE)
train=base[particion, ]
test=base[-particion, ]
```

2) Muestre un head, str y un summary del conjunto de entrenamiento y del conjunto de testeo.

head, str y summary del conjunto train:

```
head(train)
```

```
##      V1      V2      V3      V4 V5      V6 V7      V8      V9      V10      V11      V12      V13      V14
## 1 0.438 0.498 3.625 3.645 5 2.918 5 2.351 2.332 2.643 1.698 1.687 1.698 1.717
## 2 0.438 0.498 3.625 3.648 5 2.918 5 2.637 2.332 2.649 1.695 1.687 1.695 1.720
## 3 0.438 0.498 3.625 3.629 5 2.918 5 2.637 2.334 2.643 1.696 1.687 1.695 1.717
## 4 0.437 0.501 3.625 3.626 5 2.918 5 2.353 2.334 2.642 1.730 1.687 1.695 1.717
## 6 0.439 0.498 3.626 3.629 5 2.918 5 2.633 2.334 2.645 1.705 1.686 1.694 1.719
## 8 0.444 5.021 3.631 3.634 5 2.919 5 2.626 2.327 2.638 1.698 1.680 1.688 1.709
##      V15      V16      V17      V18      V19      V20      V21      V22      V23      V24      Mov
## 1 1.744 0.593 0.502 0.493 0.504 0.445 0.431 0.444 0.440 0.429 Slight-Right-Turn
## 2 1.744 0.592 0.502 0.493 0.504 0.449 0.431 0.444 0.443 0.429 Slight-Right-Turn
## 3 1.744 0.593 0.502 0.493 0.504 0.449 0.431 0.444 0.446 0.429 Slight-Right-Turn
## 4 1.744 0.593 0.502 0.493 0.504 0.449 0.431 0.444 0.444 0.429 Slight-Right-Turn
## 6 1.744 0.589 0.502 0.493 0.504 0.446 0.431 0.444 0.444 0.430 Slight-Right-Turn
## 8 1.740 0.595 0.500 0.491 0.503 0.453 0.436 0.448 0.444 0.436 Slight-Right-Turn
```

```
str(train)
```

```
## 'data.frame': 3821 obs. of 25 variables:
## $ V1 : num 0.438 0.438 0.438 0.437 0.439 0.444 0.451 0.473 0.481 0.481 ...
## $ V2 : num 0.498 0.498 0.498 0.501 0.498 ...
## $ V3 : num 3.62 3.62 3.62 3.62 3.63 ...
## $ V4 : num 3.64 3.65 3.63 3.63 3.63 ...
## $ V5 : num 5 5 5 5 5 ...
## $ V6 : num 2.92 2.92 2.92 2.92 2.92 ...
## $ V7 : num 5 5 5 5 5 ...
## $ V8 : num 2.35 2.64 2.64 2.35 2.63 ...
## $ V9 : num 2.33 2.33 2.33 2.33 2.33 ...
## $ V10: num 2.64 2.65 2.64 2.64 2.64 ...
## $ V11: num 1.7 1.7 1.7 1.73 1.71 ...
## $ V12: num 1.69 1.69 1.69 1.69 1.69 ...
## $ V13: num 1.7 1.7 1.7 1.7 1.69 ...
## $ V14: num 1.72 1.72 1.72 1.72 1.72 ...
## $ V15: num 1.74 1.74 1.74 1.74 1.74 ...
## $ V16: num 0.593 0.592 0.593 0.593 0.589 0.595 0.595 0.578 0.581 0.53 ...
## $ V17: num 0.502 0.502 0.502 0.502 0.502 0.5 0.499 0.496 0.495 0.492 ...
## $ V18: num 0.493 0.493 0.493 0.493 0.493 0.491 0.491 0.487 0.486 0.482 ...
## $ V19: num 0.504 0.504 0.504 0.504 0.504 0.503 0.502 0.498 0.497 0.492 ...
## $ V20: num 0.445 0.449 0.449 0.449 0.446 0.453 0.457 0.469 0.477 0.513 ...
## $ V21: num 0.431 0.431 0.431 0.431 0.431 0.436 0.44 0.454 0.459 0.462 ...
## $ V22: num 0.444 0.444 0.444 0.444 0.444 0.448 0.453 0.467 0.472 0.486 ...
## $ V23: num 0.44 0.443 0.446 0.444 0.444 0.444 0.454 0.476 0.484 0.483 ...
## $ V24: num 0.429 0.429 0.429 0.429 0.43 0.436 0.442 0.465 0.472 0.473 ...
## $ Mov: Factor w/ 4 levels "Move-Forward",...: 4 4 4 4 4 4 2 2 2 2 ...
```

```
summary(train)
```

```
##      V1      V2      V3      V4
## Min.   :0.400   Min.   :0.438   Min.   :0.470   Min.   :0.848
## 1st Qu.:0.918   1st Qu.:1.359   1st Qu.:1.540   1st Qu.:1.729
## Median :1.326   Median :1.890   Median :2.064   Median :2.454
```

```
## Mean :1.456 Mean :2.314 Mean :2.475 Mean :2.797
## 3rd Qu.:1.805 3rd Qu.:2.664 3rd Qu.:2.714 3rd Qu.:4.110
## Max. :5.000 Max. :5.025 Max. :5.029 Max. :5.016
## V5 V6 V7 V8
## Min. :1.123 Min. :1.114 Min. :1.122 Min. :0.863
## 1st Qu.:1.767 1st Qu.:1.787 1st Qu.:1.926 1st Qu.:1.613
## Median :2.668 Median :2.683 Median :3.224 Median :2.151
## Mean :2.944 Mean :2.907 Mean :3.343 Mean :2.520
## 3rd Qu.:4.285 3rd Qu.:4.013 3rd Qu.:5.000 3rd Qu.:3.177
## Max. :5.000 Max. :5.000 Max. :5.008 Max. :5.000
## V9 V10 V11 V12
## Min. :0.854 Min. :0.817 Min. :0.783 Min. :0.778
## 1st Qu.:1.798 1st Qu.:1.636 1st Qu.:1.577 1st Qu.:1.293
## Median :2.798 Median :2.671 Median :2.003 Median :1.692
## Mean :3.123 Mean :2.810 Mean :2.537 Mean :2.091
## 3rd Qu.:5.000 3rd Qu.:3.378 3rd Qu.:3.163 3rd Qu.:2.317
## Max. :5.000 Max. :5.000 Max. :5.019 Max. :5.000
## V13 V14 V15 V16
## Min. :0.770 Min. :0.756 Min. :0.495 Min. :0.424
## 1st Qu.:1.199 1st Qu.:1.037 1st Qu.:0.862 1st Qu.:0.695
## Median :1.613 Median :1.506 Median :1.331 Median :0.803
## Mean :2.129 Mean :2.216 Mean :2.234 Mean :1.198
## 3rd Qu.:2.360 3rd Qu.:2.940 3rd Qu.:5.000 3rd Qu.:1.168
## Max. :5.003 Max. :5.000 Max. :5.000 Max. :5.000
## V17 V18 V19 V20
## Min. :0.373 Min. :0.3540 Min. :0.340 Min. :0.355
## 1st Qu.:0.583 1st Qu.:0.5350 1st Qu.:0.525 1st Qu.:0.543
## Median :0.741 Median :0.6920 Median :0.693 Median :0.693
## Mean :1.004 Mean :0.9236 Mean :1.067 Mean :1.088
## 3rd Qu.:0.917 3rd Qu.:0.8400 3rd Qu.:0.862 3rd Qu.:0.866
## Max. :5.000 Max. :5.0000 Max. :5.000 Max. :5.000
## V21 V22 V23 V24
## Min. :0.380 Min. :0.370 Min. :0.367 Min. :0.380
## 1st Qu.:0.567 1st Qu.:0.742 1st Qu.:0.792 1st Qu.:0.884
## Median :0.762 Median :1.031 Median :1.071 Median :1.291
## Mean :1.015 Mean :1.785 Mean :1.557 Mean :1.575
## 3rd Qu.:0.999 3rd Qu.:2.081 3rd Qu.:1.574 3rd Qu.:1.661
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000
## Mov
## Move-Forward :1544
## Sharp-Right-Turn :1468
## Slight-Left-Turn : 230
## Slight-Right-Turn: 579
##
##
```

```
head, str y summary del conjunto test:
```

```
head(test)
```

```
## V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12
## 5 0.438 0.498 3.626 3.629 5.000 2.918 5.000 2.640 2.334 2.639 1.696 1.687
## 7 0.440 5.000 3.627 3.628 5.000 2.919 3.028 2.346 2.330 2.638 1.727 1.684
```

```
## 10 0.458 5.022 3.640 3.644 5.000 2.922 5.000 2.346 2.321 2.628 1.688 1.666
## 11 0.465 0.525 3.646 3.670 5.000 2.923 5.000 2.611 2.315 2.631 1.674 1.658
## 14 0.484 0.544 3.661 3.665 5.000 2.928 5.000 2.321 2.304 5.022 1.647 1.639
## 15 0.484 0.532 3.669 3.662 2.945 2.926 5.000 2.326 2.306 2.620 1.648 1.639
##      V13   V14   V15   V16   V17   V18   V19   V20   V21   V22   V23   V24
## 5   1.695 1.717 1.744 0.592 0.502 0.493 0.504 0.449 0.431 0.444 0.441 0.429
## 7   1.692 1.714 1.745 0.588 0.501 0.492 0.504 0.451 0.433 0.446 0.444 0.432
## 10  1.674 1.696 0.744 0.590 0.496 0.490 0.498 0.462 0.444 0.458 0.461 0.449
## 11  1.666 1.688 0.735 0.593 0.495 0.488 0.497 0.467 0.449 0.462 0.469 0.457
## 14  1.646 1.270 0.757 0.623 0.493 0.484 0.495 0.480 0.461 0.474 0.485 0.476
## 15  1.646 1.270 0.760 0.533 0.493 0.483 0.494 0.507 0.461 0.473 0.486 0.476
##                               Mov
## 5   Slight-Right-Turn
## 7   Slight-Right-Turn
## 10  Sharp-Right-Turn
## 11  Sharp-Right-Turn
## 14  Sharp-Right-Turn
## 15  Sharp-Right-Turn
```

```
str(test)
```

```
## 'data.frame':   1635 obs. of  25 variables:
## $ V1 : num  0.438 0.44 0.458 0.465 0.484 0.484 0.482 0.479 0.479 0.48 ...
## $ V2 : num  0.498 5 5.022 0.525 0.544 ...
## $ V3 : num  3.63 3.63 3.64 3.65 3.66 ...
## $ V4 : num  3.63 3.63 3.64 3.67 3.66 ...
## $ V5 : num  5 5 5 5 5 ...
## $ V6 : num  2.92 2.92 2.92 2.92 2.93 ...
## $ V7 : num  5 3.03 5 5 5 ...
## $ V8 : num  2.64 2.35 2.35 2.61 2.32 ...
## $ V9 : num  2.33 2.33 2.32 2.31 2.3 ...
## $ V10: num  2.64 2.64 2.63 2.63 5.02 ...
## $ V11: num  1.7 1.73 1.69 1.67 1.65 ...
## $ V12: num  1.69 1.68 1.67 1.66 1.64 ...
## $ V13: num  1.7 1.69 1.67 1.67 1.65 ...
## $ V14: num  1.72 1.71 1.7 1.69 1.27 ...
## $ V15: num  1.744 1.745 0.744 0.735 0.757 ...
## $ V16: num  0.592 0.588 0.59 0.593 0.623 0.533 0.533 0.617 0.688 0.687 ...
## $ V17: num  0.502 0.501 0.496 0.495 0.493 0.493 0.492 0.491 0.534 0.553 ...
## $ V18: num  0.493 0.492 0.49 0.488 0.484 0.483 0.482 0.479 0.475 0.475 ...
## $ V19: num  0.504 0.504 0.498 0.497 0.495 0.494 0.492 0.491 0.475 0.474 ...
## $ V20: num  0.449 0.451 0.462 0.467 0.48 0.507 0.513 0.575 0.489 0.558 ...
## $ V21: num  0.431 0.433 0.444 0.449 0.461 0.461 0.459 0.461 0.461 0.462 ...
## $ V22: num  0.444 0.446 0.458 0.462 0.474 0.473 0.474 0.465 0.456 0.453 ...
## $ V23: num  0.441 0.444 0.461 0.469 0.485 0.486 0.485 0.484 0.465 0.465 ...
## $ V24: num  0.429 0.432 0.449 0.457 0.476 0.476 0.474 0.473 0.474 0.476 ...
## $ Mov: Factor w/ 4 levels "Move-Forward",...: 4 4 2 2 2 2 2 2 4 4 ...
```

```
summary(test)
```

```
##           V1           V2           V3           V4
## Min.      :0.400   Min.      :0.437   Min.      :0.471   Min.      :0.833
## 1st Qu.:0.935   1st Qu.:1.372   1st Qu.:1.534   1st Qu.:1.737
```



##	Median :1.348	Median :1.936	Median :2.064	Median :2.478
##	Mean :1.508	Mean :2.357	Mean :2.523	Mean :2.796
##	3rd Qu.:1.833	3rd Qu.:2.724	3rd Qu.:2.848	3rd Qu.:4.061
##	Max. :5.000	Max. :5.022	Max. :5.028	Max. :5.017
##	V5	V6	V7	V8
##	Min. :1.120	Min. :1.115	Min. :1.130	Min. :0.859
##	1st Qu.:1.794	1st Qu.:1.784	1st Qu.:1.958	1st Qu.:1.635
##	Median :2.663	Median :2.682	Median :3.233	Median :2.346
##	Mean :2.992	Mean :2.861	Mean :3.370	Mean :2.588
##	3rd Qu.:4.755	3rd Qu.:3.269	3rd Qu.:5.000	3rd Qu.:3.227
##	Max. :5.000	Max. :5.005	Max. :5.000	Max. :5.087
##	V9	V10	V11	V12
##	Min. :0.836	Min. :0.810	Min. :0.790	Min. :0.779
##	1st Qu.:1.804	1st Qu.:1.636	1st Qu.:1.582	1st Qu.:1.294
##	Median :2.813	Median :2.724	Median :1.991	Median :1.683
##	Mean :3.132	Mean :2.884	Mean :2.578	Mean :2.048
##	3rd Qu.:5.000	3rd Qu.:3.769	3rd Qu.:3.325	3rd Qu.:2.219
##	Max. :5.000	Max. :5.022	Max. :5.017	Max. :5.000
##	V13	V14	V15	V16
##	Min. :0.778	Min. :0.757	Min. :0.5040	Min. :0.429
##	1st Qu.:1.183	1st Qu.:1.026	1st Qu.:0.8565	1st Qu.:0.685
##	Median :1.600	Median :1.476	Median :1.3140	Median :0.800
##	Mean :2.119	Mean :2.130	Mean :2.1401	Mean :1.212
##	3rd Qu.:2.339	3rd Qu.:2.357	3rd Qu.:3.5315	3rd Qu.:1.140
##	Max. :5.000	Max. :5.000	Max. :5.0000	Max. :5.000
##	V17	V18	V19	V20
##	Min. :0.3760	Min. :0.3550	Min. :0.3400	Min. :0.3670
##	1st Qu.:0.5720	1st Qu.:0.5210	1st Qu.:0.5205	1st Qu.:0.5385
##	Median :0.7290	Median :0.6730	Median :0.6870	Median :0.6930
##	Mean :0.9573	Mean :0.8792	Mean :1.0374	Mean :1.0486
##	3rd Qu.:0.9020	3rd Qu.:0.8270	3rd Qu.:0.8520	3rd Qu.:0.8605
##	Max. :5.0000	Max. :5.0000	Max. :5.0000	Max. :5.0000
##	V21	V22	V23	V24
##	Min. :0.380	Min. :0.370	Min. :0.367	Min. :0.3770
##	1st Qu.:0.569	1st Qu.:0.745	1st Qu.:0.792	1st Qu.:0.8835
##	Median :0.765	Median :1.030	Median :1.070	Median :1.2840
##	Mean :1.018	Mean :1.763	Mean :1.550	Mean :1.5858
##	3rd Qu.:1.009	3rd Qu.:2.050	3rd Qu.:1.502	3rd Qu.:1.6435
##	Max. :5.000	Max. :5.000	Max. :5.000	Max. :5.0000
##	Mov			
##	Move-Forward	:661		
##	Sharp-Right-Turn	:629		
##	Slight-Left-Turn	: 98		
##	Slight-Right-Turn	:247		
##				
##				

3) ¿Cuántos registros quedaron en cada conjunto (entrenamiento y testeo)? ¿Cuántos registros por clase de la variable Mov quedaron en cada conjunto?

# Cantidad de registros y registros por clase en conjunto de entrenamiento:

```
dim(train)
```

```
## [1] 3821 25
```

```
summary(train$Mov)
```

```
##      Move-Forward Sharp-Right-Turn Slight-Left-Turn Slight-Right-Turn
##              1544              1468              230              579
```

*# Cantidad de registros y registros por clase en conjunto de testeo:*

```
dim(test)
```

```
## [1] 1635 25
```

```
summary(test$Mov)
```

```
##      Move-Forward Sharp-Right-Turn Slight-Left-Turn Slight-Right-Turn
##              661              629              98              247
```

### *#Parte C - SVM*

1) Cargue la librería e1071 y modele el problema planteado con una Support Vector Machine con un kernel = (asignado1) , costo = (asignado2) y los parámetros restantes con los valores por defecto sin cambiar. Indique el código R utilizado.

2) Calcule la matriz de confusión utilizando la instrucción confusionMatrix de la librería caret. Muestre una captura de pantalla de los resultados obtenidos.

3) ¿Cuántos registros quedaron bien y mal clasificados?

4) Calcule el accuracy según la cantidad de registros bien **clasificados** (indique la fórmula que usó) y verifique que coincida con el accuracy obtenido por confusionMatrix.

5) ¿Cuál fue la sensibilidad y especificidad de cada clase?

6) ¿Cuál clase presenta mayor sensibilidad y cuál clase presenta menor sensibilidad?

### *#Parte D - Naive Bayes*

1) Cargue la librería e1071 y modele el problema planteado utilizando Naive Bayes. Indique el código R utilizado.

2) Calcule la matriz de confusión utilizando la instrucción confusionMatrix de la librería caret. Muestre una captura de pantalla de los resultados obtenidos.

3) ¿Cuántos registros quedaron bien y mal clasificados?

4) Calcule el accuracy según la cantidad de registros bien **clasificados** (indique la fórmula que usó) y verifique que coincida con el accuracy obtenido por confusionMatrix.

5) ¿Cuál fue la sensibilidad y especificidad de cada clase?

6) ¿Cuál clase presenta mayor sensibilidad y cuál clase presenta menor sensibilidad?

### *#Parte E - Comparación*

1) Compare los resultados obtenidos con la SVM y Naive Bayes. ¿Cuál modelo le parece que resultó mejor?

### *# Ejercicio 2 - Aprendizaje No Supervisado*

#### *# Parte A - Análisis Exploratorio de Datos*

*# 1) Abra la base rock y renómbrela como "base". Indique de qué trata el problema.*

```
base <- rock
```

Este dataset consiste de mediciones de 48 rocas de reservorios de petróleo.

*# 2) Muestre un head, str y summary de la base. Comente las variables, cantidad de registros, etc*

```
head(base)
```

```
##   area   peri   shape perm
## 1 4990 2791.90 0.0903296  6.3
## 2 7002 3892.60 0.1486220  6.3
## 3 7558 3930.66 0.1833120  6.3
## 4 7352 3869.32 0.1170630  6.3
## 5 7943 3948.54 0.1224170 17.1
## 6 7979 4010.15 0.1670450 17.1
```

```
str(rock)
```

```
## 'data.frame':   48 obs. of  4 variables:
## $ area : int  4990 7002 7558 7352 7943 7979 9333 8209 8393 6425 ...
## $ peri : num  2792 3893 3931 3869 3949 ...
## $ shape: num  0.0903 0.1486 0.1833 0.1171 0.1224 ...
## $ perm : num  6.3 6.3 6.3 6.3 17.1 17.1 17.1 17.1 119 119 ...
```

```
summary(base)
```

```
##      area      peri      shape      perm
## Min.   : 1016   Min.   : 308.6   Min.   :0.09033   Min.   :  6.30
```

```
## 1st Qu.: 5305    1st Qu.:1414.9    1st Qu.:0.16226    1st Qu.: 76.45
## Median : 7487    Median :2536.2    Median :0.19886    Median : 130.50
## Mean   : 7188    Mean   :2682.2    Mean   :0.21811    Mean   : 415.45
## 3rd Qu.: 8870    3rd Qu.:3989.5    3rd Qu.:0.26267    3rd Qu.: 777.50
## Max.   :12212    Max.   :4864.2    Max.   :0.46413    Max.   :1300.00
```

La base de datos consiste en Mediciones de 48 muestras de reservorios de rocas de yacimientos de petróleo. Es un dataset de 48 filas con 4 columnas, siendo las siguientes:

```
base[, 1] area: área de poros, en píxeles de 256 por 256
base[, 2] peri: perímetro en píxeles
base[, 3] shape: perímetro / sqrt (área)
base[, 4] perm: permeabilidad en mili-Darcies
```

## Parte B - Agrupamiento k-means

1) Setee la semilla=101 y realice un agrupamiento k-means con Cantidad de Grupos = (asignado3) (con nstart por defecto). Indique el código R utilizado.

Nota:

ASIGNADO 3:

Grupos 4

```
set.seed(101);km=kmeans(base,4)
```

2) Muestre una captura de pantalla de los centroides.

```
km$centers #centroides
```

```
##      area      peri      shape      perm
## 1  8441.059 3667.6371 0.1793699 167.97059
## 2  2685.143  928.9131 0.2360283 500.00000
## 3  6148.706 1650.9830 0.2468348 763.84118
## 4 11169.857 4546.7486 0.2245176  85.82857
```

3) ¿Cuántos elementos quedaron en cada grupo?

```
km$size
```

```
## [1] 17  7 17  7
```

```
GRUPO_1 = km$size[1]
GRUPO_2 = km$size[2]
GRUPO_3 = km$size[3]
GRUPO_4 = km$size[4]
```

```
GRUPO_1
```

```
## [1] 17
```

```
GRUPO_2
```

```
## [1] 7
```

```
GRUPO_3
```

```
## [1] 17
```

```
GRUPO_4
```

```
## [1] 7
```

Como se verifica son 4 grupos de clusters con cantidades 17, 7, 17, 7

4) ¿A qué grupo pertenece el tercer elemento de la base?

Observando los elementos por grupo de cluster:

```
km$cluster
```

```
## [1] 3 1 1 1 1 1 1 1 1 3 1 1 4 1 1 1 4 4 4 4 1 4 4 1 3 2 3 3 3 3 3 3 3 3 2 2
## [39] 2 3 3 2 3 1 2 2 3 1
```

Vemos que el tercer elemento corresponde al grupo 1, si quisieramos obtenerlo directamente podemos directamente ejecutar lo siguiente:

```
km$cluster[3]
```

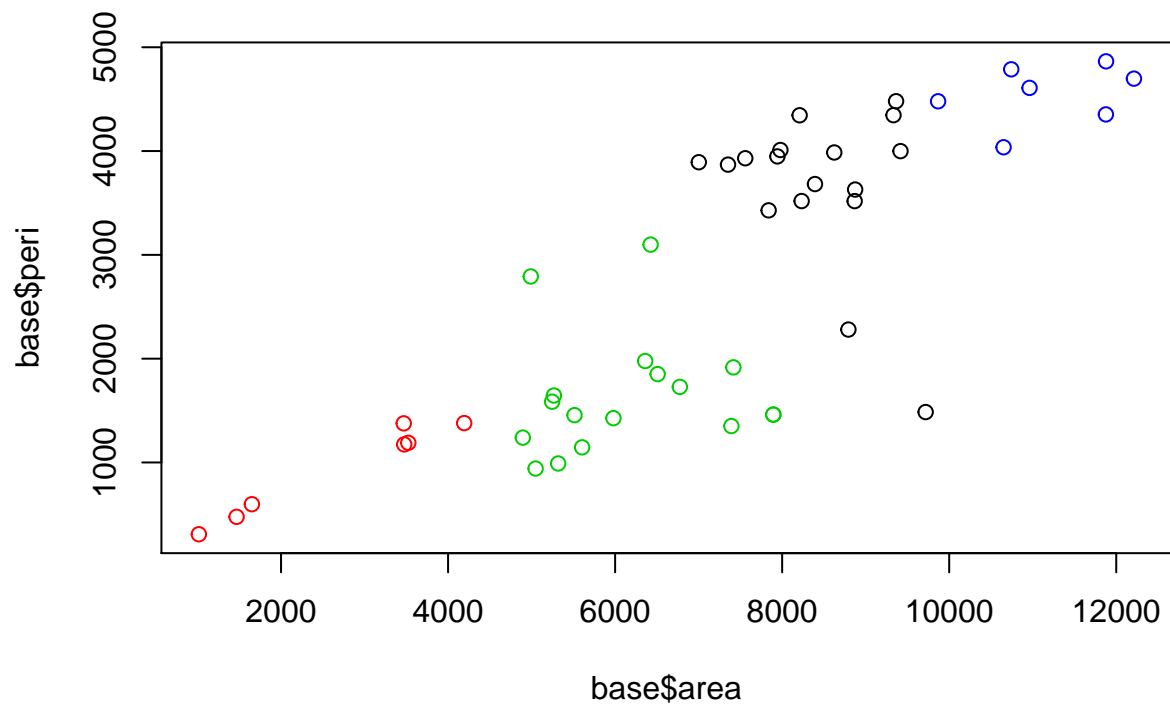
```
## [1] 1
```

5) Realice un gráfico con dos variables coloreado por los grupos formados.

```
nueva=cbind(base,km$cluster)
head(nueva)
```

```
##   area    peri    shape perm km$cluster
## 1 4990 2791.90 0.0903296 6.3      3
## 2 7002 3892.60 0.1486220 6.3      1
## 3 7558 3930.66 0.1833120 6.3      1
## 4 7352 3869.32 0.1170630 6.3      1
## 5 7943 3948.54 0.1224170 17.1     1
## 6 7979 4010.15 0.1670450 17.1     1
```

```
plot(base$area,base$peri,col=km$cluster)
```

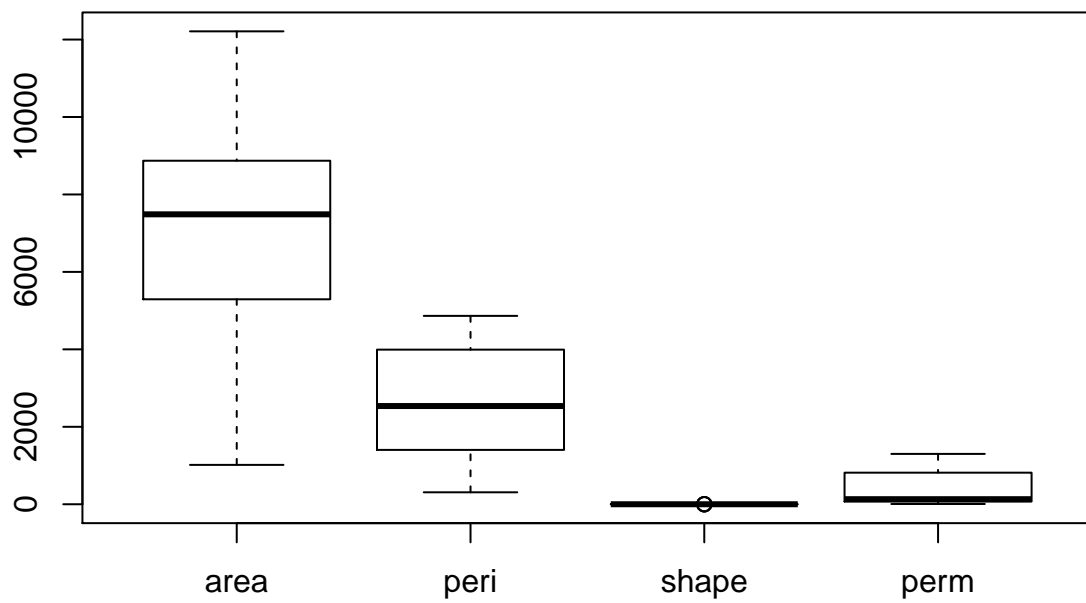


6) Determine alguna característica de alguno de los grupos

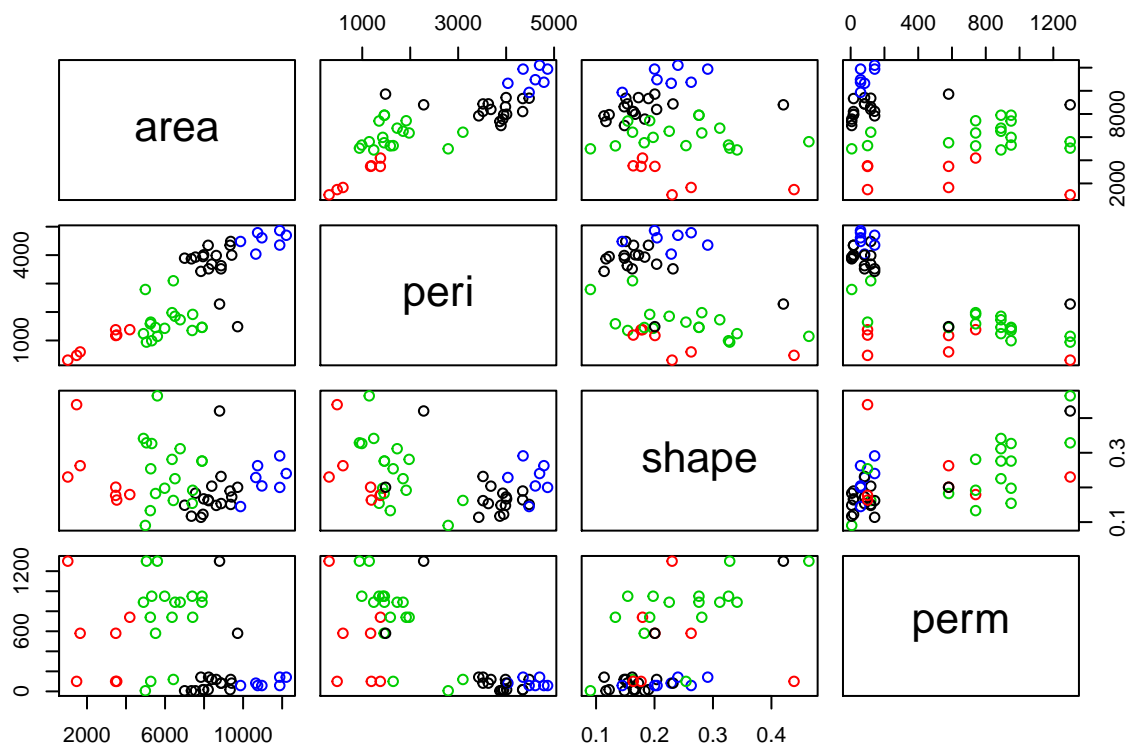
```
km$centers
```

```
##      area      peri      shape      perm
## 1  8441.059 3667.6371 0.1793699 167.97059
## 2  2685.143  928.9131 0.2360283 500.00000
## 3  6148.706 1650.9830 0.2468348 763.84118
## 4 11169.857 4546.7486 0.2245176  85.82857
```

```
boxplot(base, km$cluster)
```



```
pairs(base,col=km$cluster)
```



```
summary(base)
```

```
##      area      peri      shape      perm
##  Min.   : 1016   Min.   : 308.6   Min.   :0.09033   Min.   :  6.30
## 1st Qu.: 5305   1st Qu.:1414.9   1st Qu.:0.16226   1st Qu.: 76.45
## Median : 7487   Median :2536.2   Median :0.19886   Median : 130.50
## Mean   : 7188   Mean   :2682.2   Mean   :0.21811   Mean   : 415.45
## 3rd Qu.: 8870   3rd Qu.:3989.5   3rd Qu.:0.26267   3rd Qu.: 777.50
## Max.   :12212   Max.   :4864.2   Max.   :0.46413   Max.   :1300.00
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##   %+%, alpha
```

```
describeBy(base, km$cluster)
```

```
##
## Descriptive statistics by group
```



```

## group: 1
##      vars n    mean      sd median trimmed    mad    min    max    range
## area      1 17 8441.06 779.38 8393.00 8451.87 713.13 7002.00 9718.00 2716.00
## peri      2 17 3667.64 747.31 3892.60 3758.95 390.71 1485.58 4480.05 2994.47
## shape     3 17   0.18   0.07   0.16   0.17   0.03   0.11   0.42   0.31
## perm      4 17  167.97 321.25   82.40  103.28  96.81    6.30 1300.00 1293.70
##      skew kurtosis    se
## area -0.09   -1.16 189.03
## peri -1.66    2.15 181.25
## shape 2.33    5.60   0.02
## perm  2.65    6.25  77.91
## -----
## group: 2
##      vars n    mean      sd median trimmed    mad    min    max    range
## area      1 7 2685.14 1261.78 3469.00 2685.14 1073.40 1016.00 4193.00 3177.00
## peri      2 7  928.91  452.88 1174.11  928.91  304.29  308.64 1379.35 1070.71
## shape     3 7   0.24   0.10   0.20   0.24   0.04   0.16   0.44   0.28
## perm      4 7  500.00  445.42  580.00  500.00  711.65  100.00 1300.00 1200.00
##      skew kurtosis    se
## area -0.18   -2.01 476.91
## peri -0.24   -2.01 171.17
## shape 1.23   -0.04   0.04
## perm  0.54   -1.22 168.35
## -----
## group: 3
##      vars n    mean      sd median trimmed    mad    min    max    range
## area      1 17 6148.71 1031.45 5980.00 6115.93 1088.23 4895.00 7894.00 2999.00
## peri      2 17 1650.98  572.25 1461.06 1601.77  395.97  941.54 3098.65 2157.11
## shape     3 17   0.25   0.09   0.25   0.24   0.11   0.09   0.46   0.37
## perm      4 17  763.84  374.42  890.00  778.60  222.39    6.30 1300.00 1293.70
##      skew kurtosis    se
## area  0.41   -1.37 250.16
## peri  1.15    0.65 138.79
## shape 0.38   -0.42   0.02
## perm -0.74   -0.40  90.81
## -----
## group: 4
##      vars n    mean      sd median trimmed    mad    min    max    range
## area      1 7 11169.86 844.42 10962.00 11169.86 1355.10 9867.00 12212.00 2345.00
## peri      2 7  4546.75 285.11  4608.66  4546.75  265.33 4036.54  4864.22  827.68
## shape     3 7   0.22   0.05   0.23   0.22   0.04   0.14   0.29   0.15
## perm      4 7   85.83  39.34   58.60   85.83    0.00   58.60   142.00   83.40
##      skew kurtosis    se
## area -0.14   -1.69 319.16
## peri -0.56   -1.20 107.76
## shape -0.23   -1.23   0.02
## perm  0.64   -1.68  14.87

```

Como observamos, shape y perm no pueden usarse para discriminar diferencias entre las rocas de los yacimientos dado de que sus medias son muy similares ademas de que la variabilidad con respecto al resto es prácticamente nula.

## Parte C - Red Neuronal Kohonen

1) Transforme la base en una matriz llamada "basesom"  
`basesom=as.matrix(base)`

```
basesom=as.matrix(rock)
basesom
```

##		area	peri	shape	perm
##	[1,]	4990	2791.900	0.0903296	6.3
##	[2,]	7002	3892.600	0.1486220	6.3
##	[3,]	7558	3930.660	0.1833120	6.3
##	[4,]	7352	3869.320	0.1170630	6.3
##	[5,]	7943	3948.540	0.1224170	17.1
##	[6,]	7979	4010.150	0.1670450	17.1
##	[7,]	9333	4345.750	0.1896510	17.1
##	[8,]	8209	4344.750	0.1641270	17.1
##	[9,]	8393	3682.040	0.2036540	119.0
##	[10,]	6425	3098.650	0.1623940	119.0
##	[11,]	9364	4480.050	0.1509440	119.0
##	[12,]	8624	3986.240	0.1481410	119.0
##	[13,]	10651	4036.540	0.2285950	82.4
##	[14,]	8868	3518.040	0.2316230	82.4
##	[15,]	9417	3999.370	0.1725670	82.4
##	[16,]	8874	3629.070	0.1534810	82.4
##	[17,]	10962	4608.660	0.2043140	58.6
##	[18,]	10743	4787.620	0.2627270	58.6
##	[19,]	11878	4864.220	0.2000710	58.6
##	[20,]	9867	4479.410	0.1448100	58.6
##	[21,]	7838	3428.740	0.1138520	142.0
##	[22,]	11876	4353.140	0.2910290	142.0
##	[23,]	12212	4697.650	0.2400770	142.0
##	[24,]	8233	3518.440	0.1618650	142.0
##	[25,]	6360	1977.390	0.2808870	740.0
##	[26,]	4193	1379.350	0.1794550	740.0
##	[27,]	7416	1916.240	0.1918020	740.0
##	[28,]	5246	1585.420	0.1330830	740.0
##	[29,]	6509	1851.210	0.2252140	890.0
##	[30,]	4895	1239.660	0.3412730	890.0
##	[31,]	6775	1728.140	0.3116460	890.0
##	[32,]	7894	1461.060	0.2760160	890.0
##	[33,]	5980	1426.760	0.1976530	950.0
##	[34,]	5318	990.388	0.3266350	950.0
##	[35,]	7392	1350.760	0.1541920	950.0
##	[36,]	7894	1461.060	0.2760160	950.0
##	[37,]	3469	1376.700	0.1769690	100.0
##	[38,]	1468	476.322	0.4387120	100.0
##	[39,]	3524	1189.460	0.1635860	100.0
##	[40,]	5267	1644.960	0.2538320	100.0
##	[41,]	5048	941.543	0.3286410	1300.0
##	[42,]	1016	308.642	0.2300810	1300.0
##	[43,]	5605	1145.690	0.4641250	1300.0
##	[44,]	8793	2280.490	0.4204770	1300.0
##	[45,]	3475	1174.110	0.2007440	580.0
##	[46,]	1651	597.808	0.2626510	580.0
##	[47,]	5514	1455.880	0.1824530	580.0
##	[48,]	9718	1485.580	0.2004470	580.0

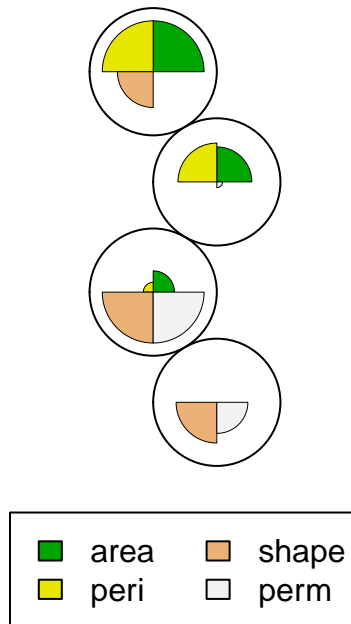
2) Cargue la librería kohonen, setee la semilla=101 y realice un agrupamiento con una Red Neuronal Kohonen con Cantidad de Grupos = (asignado3) y forma "hexagonal"  
`somgrid(1,cantGrupos,"hexagonal")`  
Indique el código R utilizado

```
library(kohonen)
set.seed(101);som=som(basesom,grid=somgrid(1,4,"hexagonal"))
```

3) Realice un gráfico con los vectores de pesos de la red

```
plot(som,type="codes")
```

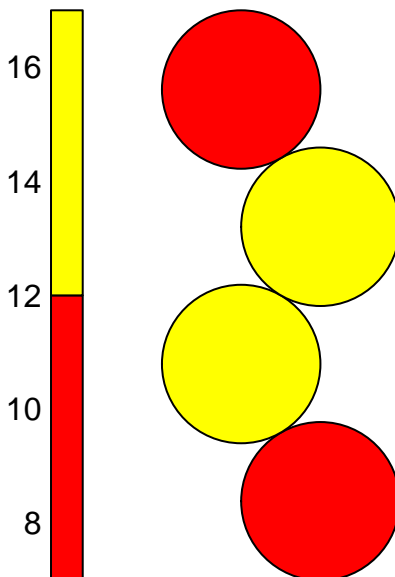
## Codes plot



4) Realice un gráfico con la cantidad de elementos que quedaron en cada grupo.

```
cantidad=plot(som,type="count")
```

## Counts plot



5) Muestre una captura de pantalla de los vectores de pesos.

```
som$codes
```

```
## [[1]]
##      area      peri      shape      perm
## V1  2589.166  905.3349  0.2298054  514.7821
## V2  6145.699  1602.8568  0.2427577  788.9125
## V3  8442.303  3664.8685  0.1780742  164.0987
## V4  11105.425  4521.5074  0.2235116   85.5087
```

6) ¿Cuántos elementos quedaron en cada grupo?

```
cantidad
```

```
## [1]  7 17 17  7
```

Como podemos observar quedaron conformados:

```
Grupo1: 7
Grupo2: 17
Grupo3: 17
Grupo4: 7
```

7) ¿A qué grupo pertenece el tercer elemento de la base?

```
names(som)
```

```
## [1] "data"          "unit.classif"    "distances"      "grid"
## [5] "codes"         "changes"         "alpha"          "radius"
## [9] "user.weights"  "distance.weights" "whatmap"        "maxNA.fraction"
## [13] "dist.fcts"
```

```
som$unit.classif[3]
```

```
## [1] 3
```

Como vemos el tercer elemento de la base pertenece al grupo 3

8) Determine alguna característica de alguno de los grupos

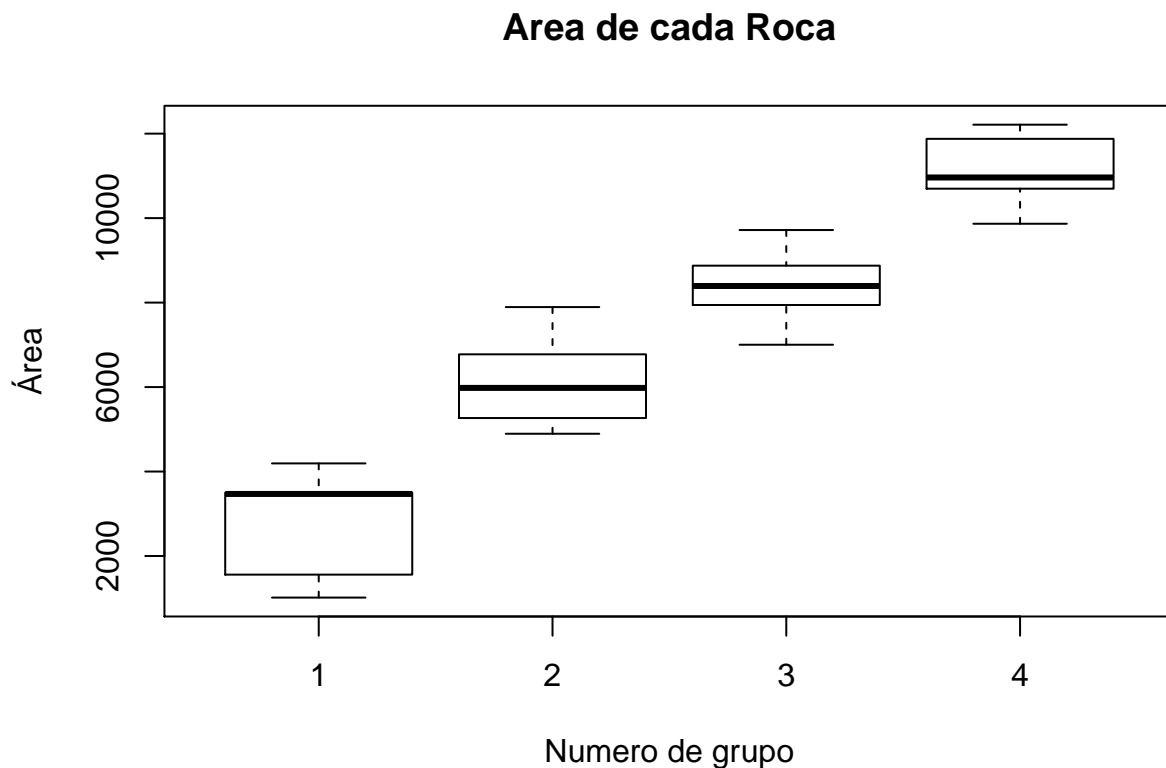
```
describeBy(base,som$unit.classif)
```

```
##
## Descriptive statistics by group
## group: 1
##      vars n    mean      sd median trimmed    mad    min    max    range
## area    1 7 2685.14 1261.78 3469.00 2685.14 1073.40 1016.00 4193.00 3177.00
## peri    2 7  928.91  452.88 1174.11  928.91  304.29  308.64 1379.35 1070.71
## shape   3 7    0.24    0.10    0.20    0.24    0.04    0.16    0.44    0.28
## perm    4 7  500.00  445.42  580.00  500.00  711.65  100.00 1300.00 1200.00
##      skew kurtosis    se
## area -0.18    -2.01 476.91
## peri -0.24    -2.01 171.17
## shape 1.23    -0.04  0.04
## perm  0.54    -1.22 168.35
## -----
## group: 2
##      vars n    mean      sd median trimmed    mad    min    max    range
## area    1 17 6148.71 1031.45 5980.00 6115.93 1088.23 4895.00 7894.00 2999.00
## peri    2 17 1650.98  572.25 1461.06 1601.77  395.97  941.54 3098.65 2157.11
## shape   3 17    0.25    0.09    0.25    0.24    0.11    0.09    0.46    0.37
## perm    4 17  763.84  374.42  890.00  778.60  222.39    6.30 1300.00 1293.70
##      skew kurtosis    se
## area  0.41    -1.37 250.16
## peri  1.15     0.65 138.79
## shape 0.38    -0.42  0.02
## perm -0.74    -0.40  90.81
## -----
## group: 3
##      vars n    mean      sd median trimmed    mad    min    max    range
## area    1 17 8441.06 779.38 8393.00 8451.87 713.13 7002.00 9718.00 2716.00
## peri    2 17 3667.64 747.31 3892.60 3758.95 390.71 1485.58 4480.05 2994.47
## shape   3 17    0.18    0.07    0.16    0.17    0.03    0.11    0.42    0.31
## perm    4 17  167.97 321.25  82.40  103.28  96.81    6.30 1300.00 1293.70
```

```
##      skew kurtosis      se
## area -0.09    -1.16 189.03
## peri -1.66     2.15 181.25
## shape 2.33     5.60  0.02
## perm  2.65     6.25  77.91
## -----
## group: 4
##      vars n      mean      sd  median trimmed      mad      min      max      range
## area    1 7 11169.86 844.42 10962.00 11169.86 1355.10 9867.00 12212.00 2345.00
## peri    2 7 4546.75 285.11 4608.66 4546.75 265.33 4036.54 4864.22 827.68
## shape    3 7   0.22  0.05   0.23   0.22   0.04   0.14   0.29   0.15
## perm    4 7   85.83 39.34   58.60   85.83   0.00   58.60  142.00  83.40
##      skew kurtosis      se
## area -0.14    -1.69 319.16
## peri -0.56    -1.20 107.76
## shape -0.23    -1.23  0.02
## perm  0.64    -1.68  14.87
```

Como podemos observar viendo la media del area el grupo 1 tiene las rocas mas pequeñas, el grupo 2 medianas, el grupo 3 grandes y el 4 gigantes

```
boxplot(base$area~som$unit.classif, main="Area de cada Roca",
        xlab="Numero de grupo", ylab="Área")
```



9) Optativo: Realice un gráfico con dos variables coloreado por los grupos formados.

```
plot(base$area,base$shape,col=som$unit.classif,pch=19,
     main="Gráfico de 2 variables - agrupamiento por Kohonen",
     xlab="Área", ylab="Shape")
```

## Gráfico de 2 variables – agrupamiento por Kohonen

