Práctica 1

Grupo 4

Iria Lago Portela
Mario Picáns Rey
Javier Kniffki
David Bamio Martínez

Ejercicios

En primer lugar vamos a cargar los datos y los paquetes necesarios para la realización de esta práctica:

```
##        rpart   rpart.plot        caret randomForest          pdp      kernlab
##         TRUE         TRUE         TRUE         TRUE         TRUE         TRUE
```

```
##                                Private     Apps   Accept   Enroll Top10perc
## University of Southern Colorado      No 7.244942 7.122060 6.405228        10
## University of San Francisco         Yes 7.743270 7.450661 6.287859        23
## Clarkson University                 Yes 7.684324 7.577122 6.322565        35
## George Washington University        Yes 8.971448 8.529517 7.307873        38
## SUNY College at Buffalo              No 8.578853 8.164795 6.932448         8
## Winona State University              No 8.109225 7.624131 7.170888        20
##                                Top25perc P.Undergrad Outstate Room.Board Books
## University of Southern Colorado        34    6.514713    7.100      4.380   5.4
## University of San Francisco            48    6.308098   13.226      6.452   7.5
## Clarkson University                    68    3.970292   15.960      5.580   7.0
## George Washington University          71    7.293018   17.450      6.328   7.0
## SUNY College at Buffalo                29    7.645398    6.550      4.040   5.5
## Winona State University                45    6.770789    4.200      2.700   3.0
##                                Personal PhD Terminal S.F.Ratio perc.alumni
## University of Southern Colorado    2.948  63       88      19.4           0
## University of San Francisco        2.450  86       86      13.6           8
## Clarkson University                1.300  95       95      15.8          32
## George Washington University       0.950  92       93       7.6          15
## SUNY College at Buffalo            1.230  71       78      18.7          12
## Winona State University            1.200  53       60      20.2          18
##                                Expend Grad.Rate
## University of Southern Colorado  5.389        36
## University of San Francisco     10.074        62
## Clarkson University             11.659        77
## George Washington University    14.745        72
## SUNY College at Buffalo          7.511        42
## Winona State University          5.318        58
```

```
## [1] 500  17
```

Este conjunto de datos está formado por 500 universidades públicas (Private==‘No’) y privadas (Private==‘Yes’) de EE.UU., para las cuales se observan 17 variables.

Para mejorar la interpretación de los resultados modificaremos la variable tipo de Universidad **Private**, de modo que 'Yes' sea Privada y 'No' sea Pública.

```
datos <- College4[,-1]
datos$Tipo <- factor(College4$Private == "Yes", labels = c("Pública", "Privada"))
head(datos)
```

```
##                                  Apps   Accept   Enroll Top10perc Top25perc
## University of Southern Colorado 7.244942 7.122060 6.405228        10        34
## University of San Francisco     7.743270 7.450661 6.287859        23        48
## Clarkson University             7.684324 7.577122 6.322565        35        68
## George Washington University    8.971448 8.529517 7.307873        38        71
## SUNY College at Buffalo         8.578853 8.164795 6.932448         8        29
## Winona State University         8.109225 7.624131 7.170888        20        45
##                                 P.Undergrad Outstate Room.Board Books Personal
## University of Southern Colorado    6.514713    7.100      4.380   5.4    2.948
## University of San Francisco        6.308098   13.226      6.452   7.5    2.450
## Clarkson University                3.970292   15.960      5.580   7.0    1.300
## George Washington University       7.293018   17.450      6.328   7.0    0.950
## SUNY College at Buffalo            7.645398    6.550      4.040   5.5    1.230
## Winona State University            6.770789    4.200      2.700   3.0    1.200
##                                 PhD Terminal S.F.Ratio perc.alumni Expend
## University of Southern Colorado  63       88      19.4           0  5.389
## University of San Francisco      86       86      13.6           8 10.074
## Clarkson University              95       95      15.8          32 11.659
## George Washington University     92       93       7.6          15 14.745
## SUNY College at Buffalo          71       78      18.7          12  7.511
## Winona State University          53       60      20.2          18  5.318
##                                 Grad.Rate    Tipo
## University of Southern Colorado        36 Pública
## University of San Francisco            62 Privada
## Clarkson University                    77 Privada
## George Washington University           72 Privada
## SUNY College at Buffalo                42 Pública
## Winona State University                58 Pública
```

Además, nótese que g.

```
#Proporción privada-pública
table(datos$Tipo)
```

```
##
## Pública Privada
##     143     357
```

1. Obtener un árbol de decisión que permita clasificar las observaciones (universidades)

en privadas (`Private="Yes"`) o públicas (`Private="No"`).

a. Seleccionar el parámetro de complejidad de forma automática, siguiendo el criterio de un error estándar de Breiman et al. (1984).

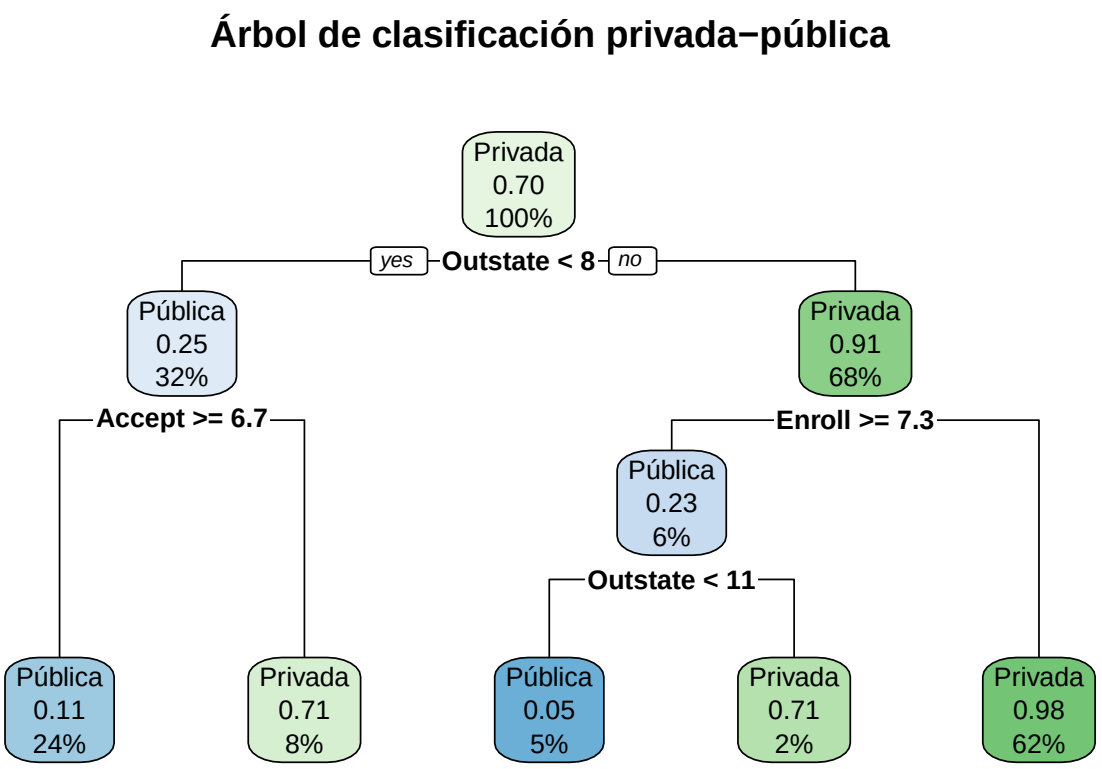
En primer lugar vamos a considerar el 80% de las observaciones como muestra de entrenamiento y el 20% restante como muestra de test.

Establecemos la semilla igual al número de grupo multiplicado por 10, utilizando la función `set.seed` de R:

```r
#Semilla
set.seed(40)
nobs <- nrow(datos) #Filas
itrain <- sample(nobs, 0.8 * nobs)
train <- datos[itrain, ] # M. Entrenamiento
test <- datos[-itrain, ] # M. Test
```

En primer lugar obtendremos un árbol que nos permita clasificar las universidades en privadas y públicas, utilizando la muestra de entrenamiento.

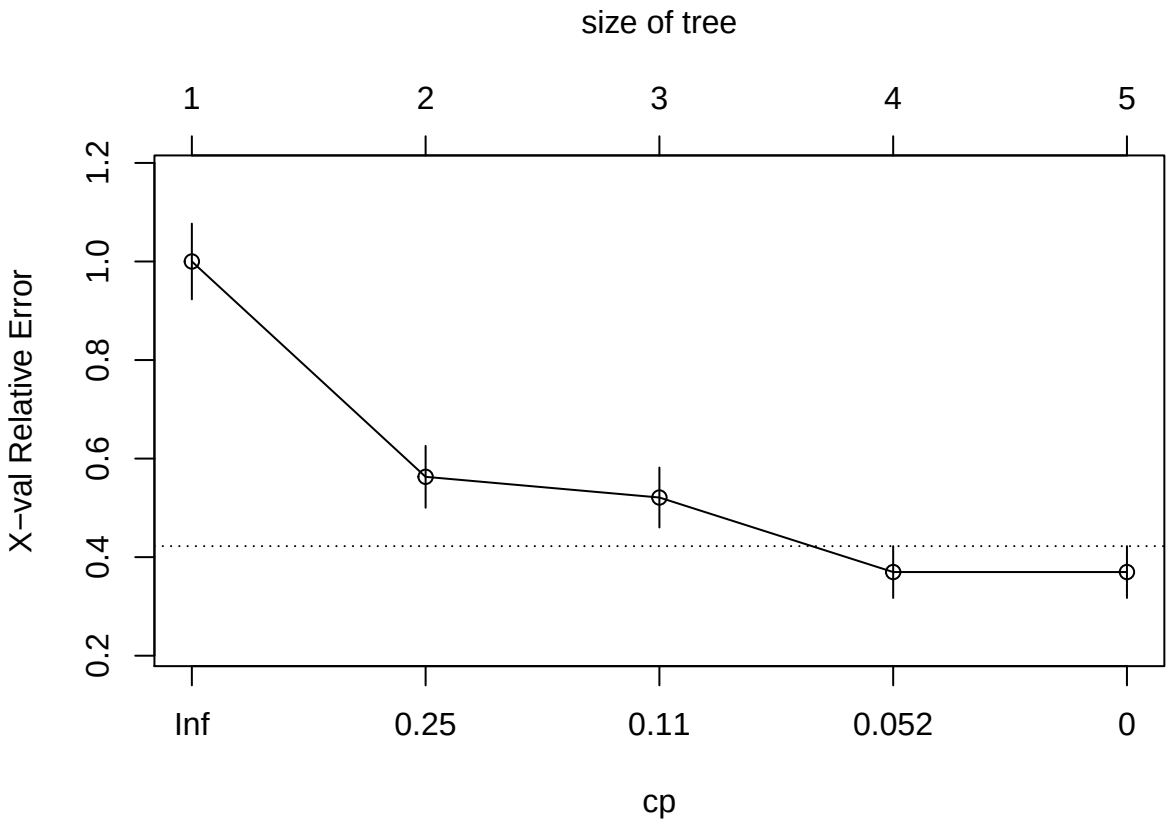
```r
tree<-rpart(Tipo~.,data=train)
rpart.plot(tree,main="Árbol de clasificación privada-pública")
```



El resultado es un árbol con 5 nodos terminales, por lo que puede ser interesante podarlo.

Para el proceso de poda seleccionaremos un paramétro de complejidad de forma automática, siguiendo el criterio de un error estándar de Breiman et al. (1984).

```r
tree <- rpart(Tipo ~ ., data = train, cp = 0)
plotcp(tree)
```




```
xerror <- tree$cptable[,"xerror"]
imin.xerror <- which.min(xerror)
upper.xerror <- xerror[imin.xerror] + tree$cptable[imin.xerror, "xstd"]
icp <- min(which(xerror <= upper.xerror))
cp <- tree$cptable[icp, "CP"]
cp
```

```
## [1] 0.02521008
```

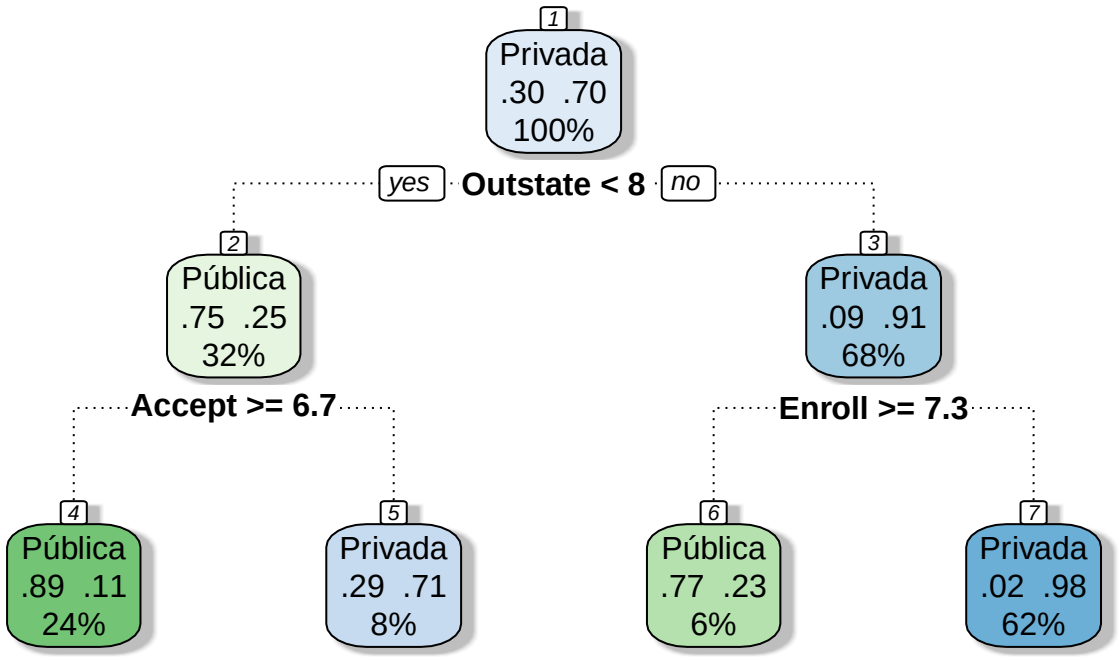
En primer lugar fijamos el parámetro $cp = 0$, es decir, ajustamos el árbol completo. A continuación se calculan los errores de validación cruzada (reescalados) dependiendo del parámetro de complejidad empleado en el ajuste del árbol de decisión. Usando el criterio del error estándar de Breiman nos quedamos con el valor de cp que de lugar al mínimo error, en este caso $cp = 0.02521008$.

b. Representar e interpretar el árbol resultante.

Si podamos el árbol utilizando el valor del parámetro obtenido en el apartado anterior, obtenemos el siguiente árbol:

```
tree<-prune(tree,cp=cp)
rpart.plot(tree,
           extra = 104,          # show fitted class, probs, percentages
           box.palette = "GnBu", # color scheme
           branch.lty = 3,       # dotted branch lines
           shadow.col = "gray",  # shadows under the node boxes
           main="Árbol de clasificación privada-pública",
           nn = TRUE)
```

Árbol de clasificación privada−pública



En este caso obtuvimos un árbol con 4 nodos terminales, que contienen un 24%, 8%, 6% y 62% del total de los datos respectivamente.

El nodo inicial o nodo padre contiene el total de los datos, para los cuales el 70% de los datos son universidades privadas y el 30% son públicas. Dado que la moda o mayoría de universidades son privadas clasifica como privada.

A continuación el árbol se divide en dos ramas teniendo en cuenta la variable **Outstate**, es decir, el número de estudiantes de otro estado (en miles). Si el número de estudiantes de otro estado es menor que 8000 entonces clasificará como universidad pública, mientras que si es mayor clasificará como privada.

En el nodo 2 se encuentra un 32% de los datos, para los cuales el 75% son universidades públicas y el 25% privadas.

En el nodo 3 se encuentra un 68% de los datos, para los cuales el 9% de los datos son universidades públicas y el 91% son privadas.

A continuación el nodo 2 se divide en otras dos ramas teniendo en cuenta la variable **Accept**, es decir, el número de solicitudes aceptadas en escala logarítmica. Si el número de solicitudes aceptadas es mayor o igual que 6.7 entonces clasificará como universidad pública, mientras que si es menor clasificará como privada.

Por otra parte el nodo 3 se divide en dos teniendo en cuenta la variable **Enroll**, es decir, el número de nuevos estudiantes matriculados en escala logarítmica. Si el número de nuevos estudiantes es mayor o igual que 7.3, el árbol clasificará como universidad pública, mientras que si es menor clasificará como universidad privada.

En el primer nodo terminal se encuentra un 24% de los datos, de los cuales el 89% de las universidades son públicas y el 11% restante son privadas. Dado que hay un mayor número de universidades públicas clasifica en públicas.

En el segundo nodo terminal se encuentra un 8% de los datos, de los cuales el 29% de las universidades son públicas y el 71% restante son privadas, por lo que clasifica en privadas.

En el tercer nodo terminal se encuentra un 6% de los datos, de los cuales el 77% de las universidades son públicas y el 23% restante son privadas, por lo que clasifica en públicas.

En el último nodo terminal se encuentra un 62% de los datos, de los cuales el 2% de las universidades son públicas y el 98% restante son privadas, por lo que clasifica en privadas.

Nótese que tanto el primer como el último nodo terminal poseen colores más oscuros, esto indica que en estos nodos la clasificación es mejor.

c. Evaluar la precisión, de las predicciones y de las estimaciones de la probabilidad, en la muestra de test.

Por último nos piden evaluar la precisión de las predicciones y de las estimaciones de la probabilidad en la muestra de test. Para ello debemos obtener las observaciones de la muestra de test y compararlas con las predicciones obtenidas con nuestro modelo.

```
obs <- test$Tipo # Observaciones
pred <- predict(tree, newdata = test, type = "class") #Predicciones
confusionMatrix(pred,obs)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Pública Privada
##    Pública      17       6
##    Privada       7      70
##
##                Accuracy : 0.87
##                  95% CI : (0.788, 0.9289)
##     No Information Rate : 0.76
##     P-Value [Acc > NIR] : 0.004749
##
##                   Kappa : 0.6385
##
##  Mcnemar's Test P-Value : 1.000000
##
##             Sensitivity : 0.7083
##             Specificity : 0.9211
##          Pos Pred Value : 0.7391
##          Neg Pred Value : 0.9091
##              Prevalence : 0.2400
##          Detection Rate : 0.1700
##    Detection Prevalence : 0.2300
##       Balanced Accuracy : 0.8147
##
##        'Positive' Class : Pública
##
```

En primer lugar obtenemos la matriz de confusión, donde enfrentamos observaciones frente a predicciones. En este caso hemos obtenido que el modelo clasifica bien 17 universidades públicas de un total de 24 y 70 universidades privadas de un total de 76. Luego nuestro modelo tiene una precisión de las predicciones de un 87%.

Sin embargo, hay que tener en cuenta que se trata de una muestra desbalanceada, puesto que contiene 143 universidades públicas y 357 universidades privadas. En estos casos conviene fijarse en el Kappa, que posee un valor más bajo, del 63.85%.

Para calcular la precisión de las estimaciones de la probabilidad, debemos utilizar la función `pred`con la opción por defecto 'type="prob":

```
pred_prob <- predict(tree, newdata = test) #Estimaciones de la probabilidad
head(pred_prob)
```

```
##                                  Pública   Privada
## University of San Francisco    0.02016129 0.9798387
## Clarkson University            0.02016129 0.9798387
## Marymount University           0.02016129 0.9798387
## West Virginia Wesleyan College 0.02016129 0.9798387
## Salem-Teikyo University        0.02016129 0.9798387
## Loyola Marymount University    0.02016129 0.9798387
```

Así obtenemos la probabilidad de que cada Universidad sea pública o privada.

2. Realizar la clasificación anterior empleando Bosques Aleatorios mediante

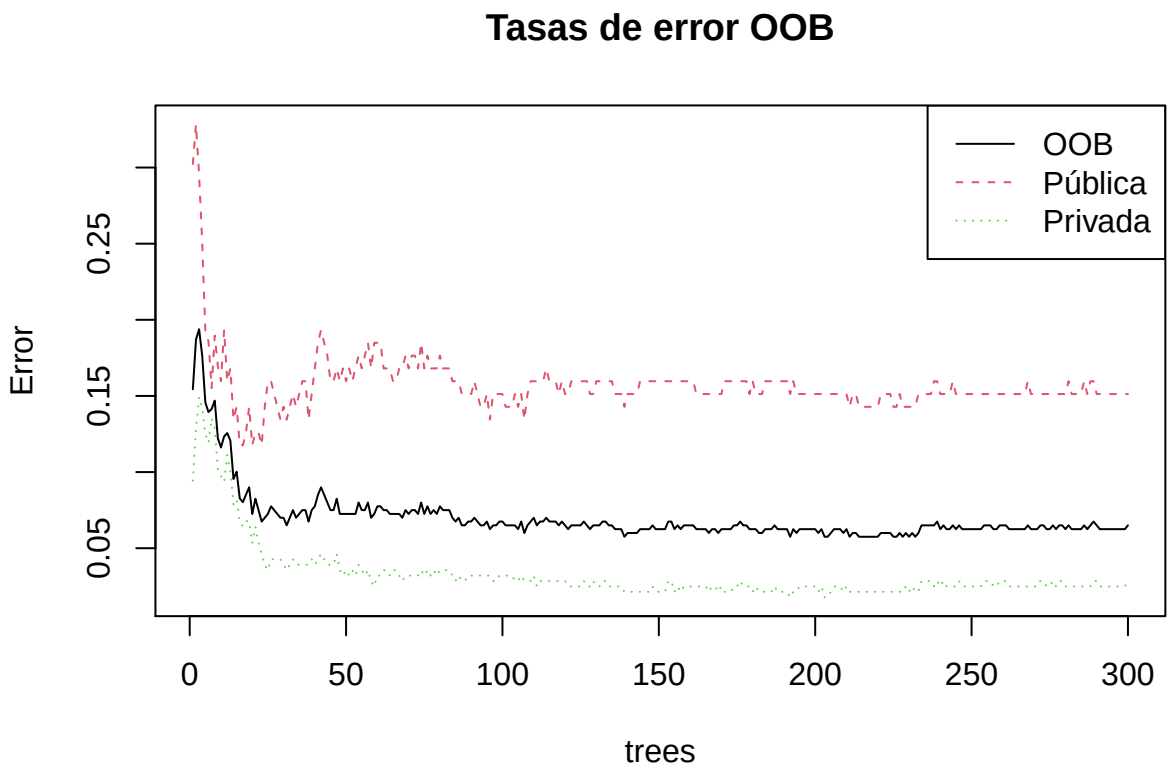
el método `"rf"` del paquete `caret`.

a. Considerar 300 árboles y seleccionar el número de predictores empleados en cada división `mtry = c(1, 2, 4, 6)` mediante validación cruzada, con 10 grupos y empleando el criterio de un error estándar de Breiman.

```
tuneGrid <- data.frame(mtry = c(1, 2, 4, 6))
rf.caret <-
    train(
        Tipo ~ .,
        data = train,
        method = "rf",
        ntree = 300,
        tuneGrid = tuneGrid,
        trControl = trainControl(
            method = "cv",
            number = 10,
            selectionFunction = "oneSE"
        )
    )
final <- rf.caret$finalModel
```

b. Representar la convergencia del error en las muestras OOB en el modelo final.

```
plot(final, main = "Tasas de error OOB")
legend("topright",
       colnames(final$err.rate),
       lty = 1:5,
       col = 1:6)
```

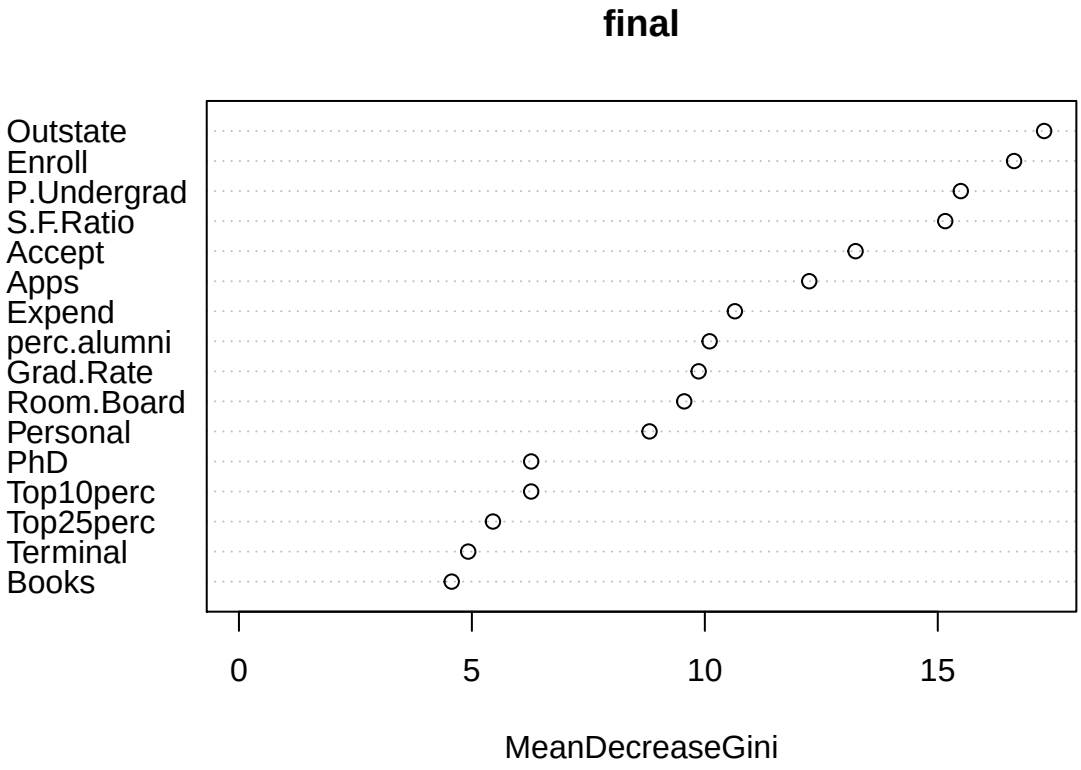


c. Estudiar la importancia de las variables y el efecto de las principales empleando algún método gráfico (para la interpretación del modelo).

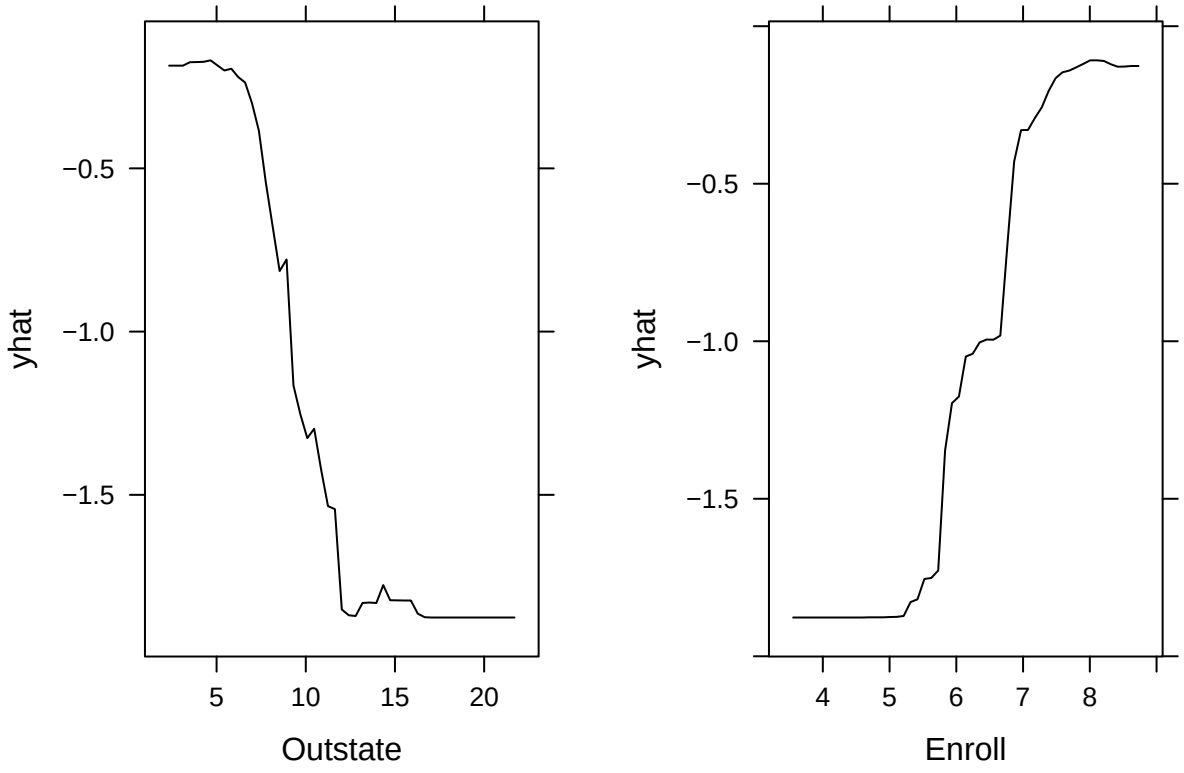
```
importance(final)
```

```
##              MeanDecreaseGini
## Apps                12.241090
## Accept              13.235416
## Enroll              16.638087
## Top10perc            6.272712
## Top25perc            5.452311
## P.Undergrad         15.493902
## Outstate            17.283349
## Room.Board           9.557924
## Books                4.566409
## Personal             8.812833
## PhD                  6.273321
## Terminal             4.920698
## S.F.Ratio           15.160058
## perc.alumni         10.102554
## Expend              10.644972
## Grad.Rate            9.867373
```

```
varImpPlot(final)
```




```
pdp1 <- partial(final, "Outstate", train = train)
p1 <- plotPartial(pdp1)
pdp2 <- partial(final, "Enroll", train = train)
p2 <- plotPartial(pdp2)
grid.arrange(p1, p2, ncol = 2)
```



d. Evaluar la precisión de las predicciones en la muestra de test y comparar los resultados con los obtenidos con el modelo del ejercicio anterior.

```
obs <- test$Tipo
head(predict(final, newdata = test))
```

```
##    University of San Francisco            Clarkson University
##                        Privada                        Privada
##           Marymount University West Virginia Wesleyan College
##                        Privada                        Privada
##        Salem-Teikyo University    Loyola Marymount University
##                        Privada                        Privada
## Levels: Pública Privada
```

```
pred <- predict(final, newdata = test, type = "class")
table(obs, pred)
```

```
##          pred
## obs       Pública Privada
##   Pública      18       6
##   Privada       4      72
```

```
confusionMatrix(pred, obs)
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction Pública Privada
##    Pública      18       4
##    Privada       6      72
##
##                Accuracy : 0.9
##                  95% CI : (0.8238, 0.951)
##     No Information Rate : 0.76
##     P-Value [Acc > NIR] : 0.0003075
##
##                   Kappa : 0.7178
##
##  Mcnemar's Test P-Value : 0.7518296
##
##             Sensitivity : 0.7500
##             Specificity : 0.9474
##          Pos Pred Value : 0.8182
##          Neg Pred Value : 0.9231
##              Prevalence : 0.2400
##          Detection Rate : 0.1800
##    Detection Prevalence : 0.2200
##       Balanced Accuracy : 0.8487
##
##        'Positive' Class : Pública
##
```

3. Realizar la clasificación anterior empleando SVM mediante la función `ksvm()` del paquete `kernlab`,

a. Ajustar el modelo con las opciones por defecto.

```
set.seed(40)
svm <- ksvm(Tipo ~ ., data = train)
svm
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.0543868376683745
##
## Number of Support Vectors : 114
##
## Objective Function Value : -66.589
## Training error : 0.0425
```

```
pred <- predict(svm, newdata = test)
confusionMatrix(pred, test$Tipo)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Pública Privada
##    Pública      19       3
```

```
##    Privada       5      73
##
##                Accuracy : 0.92
##                  95% CI : (0.8484, 0.9648)
##     No Information Rate : 0.76
##     P-Value [Acc > NIR] : 3.001e-05
##
##                   Kappa : 0.7743
##
##  Mcnemar's Test P-Value : 0.7237
##
##             Sensitivity : 0.7917
##             Specificity : 0.9605
##          Pos Pred Value : 0.8636
##          Neg Pred Value : 0.9359
##              Prevalence : 0.2400
##          Detection Rate : 0.1900
##    Detection Prevalence : 0.2200
##       Balanced Accuracy : 0.8761
##
##        'Positive' Class : Pública
##
```

b. Ajustar el modelo empleando validación cruzada con 10 grupos para seleccionar los valores "óptimos" de los hiperparámetros, considerando las posibles combinaciones de `sigma = c(0.01, 0.05, 0.1)` y `C = c(0.5, 1, 10)` (sin emplear el paquete `caret`; ver Ejercicio 3.1 en *03-bagging_boosting-ejercicios.html*).

```
tune.grid <- expand.grid(
    sigma = c(0.01, 0.05, 0.1),
    C = c(0.5, 1, 10),
    error = NA
)
best.err <- Inf
set.seed(40)
for (i in 1:nrow(tune.grid)) {
    fit <-
        ksvm(
            Tipo ~ .,
            data = train[, ],
            cross = 10,
            C = tune.grid$C[i],
            kpar = list(tune.grid$sigma[i])
        )
    fit.error <- fit@cross
    tune.grid$error[i] <- fit.error
    if (fit.error < best.err) {
        final.model <- fit
        best.err <- fit.error
        best.tune <- tune.grid[i,]
    }
}
final.model
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 0.5
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.01
##
## Number of Support Vectors : 145
##
## Objective Function Value : -53.98
## Training error : 0.055
## Cross validation error : 0.0525
```

```
pred2 <- predict(final.model, newdata = test)
confusionMatrix(pred2, test$Tipo)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Pública Privada
##    Pública      19       3
##    Privada       5      73
##
##                Accuracy : 0.92
##                  95% CI : (0.8484, 0.9648)
##     No Information Rate : 0.76
##     P-Value [Acc > NIR] : 3.001e-05
##
##                   Kappa : 0.7743
##
##  Mcnemar's Test P-Value : 0.7237
##
##             Sensitivity : 0.7917
##             Specificity : 0.9605
##          Pos Pred Value : 0.8636
##          Neg Pred Value : 0.9359
##              Prevalence : 0.2400
##          Detection Rate : 0.1900
##    Detection Prevalence : 0.2200
##       Balanced Accuracy : 0.8761
##
##        'Positive' Class : Pública
##
```

c. Evaluar la precisión de las predicciones de ambos modelos en la muestra de test y comparar también los resultados con los obtenidos en el ejercicio anterior.