Este trabalho pode ser realizado em equipes de no máximo 5 integrantes

Estudante: Clístenes Grizafis Bento

O que deve ser entregue:

- Um arquivo compactado com os documentos e arquivos
- A lista de comandos R que foi executada, com suas respectivas saídas
- Um texto com o resultado e justificativa do porque
- Outros arquivos pedidos (ex, modelo gerado)

1 Pesquisa com Dados de Satélite (Satellite)

O banco de dados consiste nos valores multi-espectrais de pixels em vizinhanças 3x3 em uma imagem de satélite, e na classificação associada ao pixel central em cada vizinhança. O objetivo é prever esta classificação, dados os valores multi-espectrais.

Um quadro de imagens do Satélite Landsat com MSS (Multispectral Scanner System) consiste em quatro imagens digitais da mesma cena em diferentes bandas espectrais. Duas delas estão na região visível (correspondendo aproximadamente às regiões verde e vermelha do espectro visível) e duas no infravermelho (próximo). Cada pixel é uma palavra binária de 8 bits, com 0 correspondendo a preto e 255 a branco. A resolução espacial de um pixel é de cerca de 80m x 80m. Cada imagem contém 2340 x 3380 desses pixels. O banco de dados é uma subárea (minúscula) de uma cena, consistindo de 82 x 100 pixels. Cada linha de dados corresponde a uma vizinhança quadrada de pixels 3x3 completamente contida dentro da subárea 82x100. Cada linha contém os valores de pixel nas quatro bandas espectrais (convertidas em ASCII) de cada um dos 9 pixels na vizinhança de 3x3 e um número indicando o rótulo de classificação do pixel central.

As classes são: solo vermelho, colheita de algodão, solo cinza, solo cinza úmido, restolho de vegetação, solo cinza muito úmido.

Os dados estão em ordem aleatória e certas linhas de dados foram removidas, portanto você não pode reconstruir a imagem original desse conjunto de dados. Em cada linha de dados, os quatro valores espectrais para o pixel superior esquerdo são dados primeiro, seguidos pelos quatro valores espectrais para o pixel superior central e, em seguida, para o pixel superior direito, e assim por diante, com os pixels lidos em sequência, da esquerda para a direita e de cima para baixo. Assim, os quatro valores espectrais para o pixel central são dados pelos atributos 17,

18, 19 e 20. Se você quiser, pode usar apenas esses quatro atributos, ignorando os outros. Isso evita o problema que surge quando uma vizinhança 3x3 atravessa um limite.

O banco de dados se encontra no pacote mlbench e é completo (não possui dados faltantes).

Tarefas:

- Treine modelos RandomForest, SVM e RNA para predição destes dados.
- Escolha o melhor modelo com base em suas matrizes de confusão.
- Treine o modelo final com todos os dados e faça a predição na base completa.
- Analise o resultado.
- Salve este modelo final

In [2]:
```
#Carregando os dados
library("caret")
library("mlbench")
data(Satellite)
dataset <- Satellite
head(dataset)
```

x.1	x.2	x.3	x.4	x.5	x.6	x.7	x.8	x.9	x.10	...	x.28	x.29	x.30	x.31	x.32	x.33	x.34	x.35	x.36	classes
92	115	120	94	84	102	106	79	84	102	...	104	88	121	128	100	84	107	113	87	grey soil
84	102	106	79	84	102	102	83	80	102	...	100	84	107	113	87	84	99	104	79	grey soil
84	102	102	83	80	102	102	79	84	94	...	87	84	99	104	79	84	99	104	79	grey soil
80	102	102	79	84	94	102	79	80	94	...	79	84	99	104	79	84	103	104	79	grey soil
84	94	102	79	80	94	98	76	80	102	...	79	84	103	104	79	79	107	109	87	grey soil
80	94	98	76	80	102	102	79	76	102	...	79	79	107	109	87	79	107	109	87	grey soil

In [3]:
```
tail(dataset)
```

	x.1	x.2	x.3	x.4	x.5	x.6	x.7	x.8	x.9	x.10	...	x.28	x.29	x.30	x.31	x.32	x.33	x.34	x.35	x.36	classes
6430	84	116	128	103	92	116	133	103	84	112	...	85	74	83	104	92	78	96	112	96	red soil
6431	60	83	96	85	64	87	100	88	64	83	...	92	66	87	108	89	63	83	104	85	red soil
6432	64	79	100	85	56	71	96	85	56	68	...	85	66	83	100	85	63	83	100	81	red soil
6433	56	68	91	81	56	64	91	81	53	64	...	81	59	87	96	81	63	83	92	74	vegetation stubble
6434	56	68	87	74	60	71	91	81	60	64	...	74	59	83	92	74	59	83	92	70	vegetation stubble
6435	60	71	91	81	60	64	104	99	56	64	...	74	59	83	92	70	63	79	108	92	vegetation stubble

In [4]:
```
#Separando dados de treino e de teste
indices <- createDataPartition(dataset$classes, p=0.8, list=F)
treino <- dataset[indices,]
teste <- dataset[-indices,]
```

Treinar e testar modelo com random forest

In [5]:
```
# criando semente pseudo aleatória para verificação futura
set.seed(1)
# creinando o modelo
rf <- train(classes ~ x.17 + x.18 + x.19 + x.20, data=treino, method="rf")
```

In [6]:
```
# realizando predição com a base de teste
predicao.rf <- predict(rf, teste)
```

In [8]:
```
# Gerar matriz de confusão do modelo
confusionMatrix(predicao.rf, teste$classes)
```

Confusion Matrix and Statistics

Prediction	Reference: red soil	cotton crop	grey soil	damp grey soil
red soil	298	0	5	3
cotton crop	0	129	0	1
grey soil	4	0	241	29
damp grey soil	0	2	19	51
vegetation stubble	4	8	1	0
very damp grey soil	0	1	5	41

Prediction	Reference: vegetation stubble	very damp grey soil
red soil	11	1
cotton crop	6	2
grey soil	0	14
damp grey soil	0	30
vegetation stubble	113	8
very damp grey soil	11	246

Overall Statistics

```
               Accuracy : 0.8396
                 95% CI : (0.8183, 0.8592)
    No Information Rate : 0.2383
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8012

 Mcnemar's Test P-Value : NA
```

Statistics by Class:

	Class: red soil	Class: cotton crop	Class: grey soil
Sensitivity	0.9739	0.9214	0.8893
Specificity	0.9796	0.9921	0.9536
Pos Pred Value	0.9371	0.9348	0.8368
Neg Pred Value	0.9917	0.9904	0.9699
Prevalence	0.2383	0.1090	0.2111
Detection Rate	0.2321	0.1005	0.1877
Detection Prevalence	0.2477	0.1075	0.2243
Balanced Accuracy	0.9767	0.9568	0.9215

	Class: damp grey soil	Class: vegetation stubble
Sensitivity	0.40800	0.80142
Specificity	0.95600	0.98163

```
Pos Pred Value                    0.50000                 0.84328
Neg Pred Value                    0.93739                 0.97565
Prevalence                        0.09735                 0.10981
Detection Rate                    0.03972                 0.08801
Detection Prevalence              0.07944                 0.10436
Balanced Accuracy                 0.68200                 0.89152
                     Class: very damp grey soil
Sensitivity                             0.8173
Specificity                             0.9410
Pos Pred Value                          0.8092
Neg Pred Value                          0.9439
Prevalence                              0.2344
Detection Rate                          0.1916
Detection Prevalence                    0.2368
Balanced Accuracy                       0.8791
```

Avaliação do resultado: O modelo obteve acurácia de 83,96% o que é algo positivo pois indica que o modelo não está sofrendo de overfitting. O modelo encontrou dificuldade em reconhecer a classe damp grey soil.

Treinar e testar modelo com SVM

In [9]:
```r
# criando semente pseudo aleatória para verificação futura
set.seed(1)
# creinando o modelo
svm <- train(classes ~ x.17 + x.18 + x.19 + x.20, data=treino, method="svmRadial")
```

In [10]:
```r
# realizando predição com a base de teste
predicao.svm <- predict(svm, teste)
```

In [11]:
```r
# Gerar matriz de confusão do modelo
confusionMatrix(predicao.svm, teste$classes)
```

```
Confusion Matrix and Statistics

                    Reference
Prediction           red soil cotton crop grey soil damp grey soil
  red soil                300           1         4              2
  cotton crop               0         129         0              1
  grey soil                 4           0       257             38
  damp grey soil            0           2         9             53
  vegetation stubble        2           7         0              0
  very damp grey soil       0           1         1             31
                    Reference
Prediction           vegetation stubble very damp grey soil
  red soil                           12                   0
  cotton crop                         3                   1
  grey soil                           0                  14
  damp grey soil                      0                  35
  vegetation stubble                111                  10
  very damp grey soil                15                 241

Overall Statistics

               Accuracy : 0.8497
                 95% CI : (0.829, 0.8688)
    No Information Rate : 0.2383
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8136

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: red soil Class: cotton crop Class: grey soil
Sensitivity                   0.9804             0.9214           0.9483
Specificity                   0.9806             0.9956           0.9447
Pos Pred Value                0.9404             0.9627           0.8211
Neg Pred Value                0.9938             0.9904           0.9856
Prevalence                    0.2383             0.1090           0.2111
Detection Rate                0.2336             0.1005           0.2002
Detection Prevalence          0.2484             0.1044           0.2438
Balanced Accuracy             0.9805             0.9585           0.9465
                     Class: damp grey soil Class: vegetation stubble
Sensitivity                        0.42400                   0.78723
Specificity                        0.96031                   0.98338
```

```
Pos Pred Value                    0.53535                 0.85385
Neg Pred Value                    0.93924                 0.97400
Prevalence                        0.09735                 0.10981
Detection Rate                    0.04128                 0.08645
Detection Prevalence              0.07710                 0.10125
Balanced Accuracy                 0.69216                 0.88531
                     Class: very damp grey soil
Sensitivity                             0.8007
Specificity                             0.9512
Pos Pred Value                          0.8339
Neg Pred Value                          0.9397
Prevalence                              0.2344
Detection Rate                          0.1877
Detection Prevalence                    0.2251
Balanced Accuracy                       0.8759
```

Avaliação do resultado: O modelo obteve acurácia de 84,97% o que é algo positivo pois indica que o modelo não está sofrendo de overfitting. O modelo encontrou menor dificuldade em reconhecer a classe damp grey soil.

Treinar e testar modelo com RNA

In [13]:
```r
# criando semente pseudo aleatória para verificação futura
set.seed(1)
# creinando o modelo
rna <- train(classes ~ x.17 + x.18 + x.19 + x.20, data=treino, method="nnet")
```

In [14]:
```r
# realizando predição com a base de teste
predicao.rna <- predict(rna, teste)
```

In [15]:
```r
# Gerar matriz de confusão do modelo
confusionMatrix(predicao.rna, teste$classes)
```

```
Confusion Matrix and Statistics

                    Reference
Prediction           red soil cotton crop grey soil damp grey soil
  red soil                296           2         7              3
  cotton crop               0         127         0              0
  grey soil                 6           0       254             45
  damp grey soil            0           0         8             17
  vegetation stubble        4           7         0              0
  very damp grey soil       0           4         2             60
                    Reference
Prediction           vegetation stubble very damp grey soil
  red soil                           21                   1
  cotton crop                         2                   0
  grey soil                           0                  17
  damp grey soil                      0                  11
  vegetation stubble                101                  12
  very damp grey soil                17                 260

Overall Statistics

               Accuracy : 0.8217
                 95% CI : (0.7996, 0.8422)
    No Information Rate : 0.2383
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7768

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: red soil Class: cotton crop Class: grey soil
Sensitivity                   0.9673            0.90714           0.9373
Specificity                   0.9652            0.99825           0.9329
Pos Pred Value                0.8970            0.98450           0.7888
Neg Pred Value                0.9895            0.98874           0.9823
Prevalence                    0.2383            0.10903           0.2111
Detection Rate                0.2305            0.09891           0.1978
Detection Prevalence          0.2570            0.10047           0.2508
Balanced Accuracy             0.9663            0.95270           0.9351
                     Class: damp grey soil Class: vegetation stubble
Sensitivity                        0.13600                   0.71631
Specificity                        0.98361                   0.97988
```

```
Pos Pred Value                    0.47222                  0.81452
Neg Pred Value                    0.91346                  0.96552
Prevalence                        0.09735                  0.10981
Detection Rate                    0.01324                  0.07866
Detection Prevalence              0.02804                  0.09657
Balanced Accuracy                 0.55980                  0.84809
                     Class: very damp grey soil
Sensitivity                               0.8638
Specificity                               0.9156
Pos Pred Value                            0.7580
Neg Pred Value                            0.9564
Prevalence                                0.2344
Detection Rate                            0.2025
Detection Prevalence                      0.2671
Balanced Accuracy                         0.8897
```

Avaliação do resultado: O modelo obteve acurácia de 82,17% o que é algo positivo pois indica que o modelo não está sofrendo de overfitting. O modelo encontrou maior dificuldade em reconhecer a classe damp grey soil.

O modelo escolhido com base na matriz de confusão foi o SVM devido a sua acurácia

Treinando o modelo final com todos os dados e fazendo a predição na base completa

In [34]:
```
# identificando sigma e C de svm
print(svm)
```

```
Support Vector Machines with Radial Basis Function Kernel

5151 samples
   4 predictor
   6 classes: 'red soil', 'cotton crop', 'grey soil', 'damp grey soil', 'vegetation stubble', 'very damp grey soil'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 5151, 5151, 5151, 5151, 5151, 5151, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.8601569  0.8265153
  0.50  0.8628120  0.8298780
  1.00  0.8651450  0.8328261

Tuning parameter 'sigma' was held constant at a value of 0.8788543
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.8788543 and C = 1.
```

In [35]:
```r
# Treinando o modelo final
library(kernlab)

modelo_final <- ksvm(classes~x.17+x.18+x.19+x.20,
                     data=dataset,
                     type="C-svc",
                     kernel="rbfdot",
                     C=1.0,
                     kpar=list(sigma=0.8788543)
                    )
```

In [36]:
```r
# Realizando predição do modelo final com toda base
predicao_final.svm <- predict(modelo_final, dataset)

# Exibindo a matrix de confusão
confusionMatrix(predicao_final.svm, dataset$classes)
```

```
Confusion Matrix and Statistics

                    Reference
Prediction           red soil cotton crop grey soil damp grey soil
  red soil               1499           2        10              5
  cotton crop               2         637         0              1
  grey soil                19           0      1302            161
  damp grey soil            0           6        41            309
  vegetation stubble       12          48         0              4
  very damp grey soil       1          10         5            146
                    Reference
Prediction           vegetation stubble very damp grey soil
  red soil                           48                   0
  cotton crop                         9                   2
  grey soil                           3                  49
  damp grey soil                      4                 165
  vegetation stubble                571                  29
  very damp grey soil                72                1263

Overall Statistics

               Accuracy : 0.8673
                 95% CI : (0.8588, 0.8755)
    No Information Rate : 0.2382
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8355

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: red soil Class: cotton crop Class: grey soil
Sensitivity                   0.9778            0.90612           0.9588
Specificity                   0.9867            0.99756           0.9543
Pos Pred Value                0.9584            0.97849           0.8488
Neg Pred Value                0.9930            0.98859           0.9886
Prevalence                    0.2382            0.10925           0.2110
Detection Rate                0.2329            0.09899           0.2023
Detection Prevalence          0.2430            0.10117           0.2384
Balanced Accuracy             0.9823            0.95184           0.9565
                     Class: damp grey soil Class: vegetation stubble
Sensitivity                        0.49361                   0.80764
Specificity                        0.96282                   0.98376
```

```
Pos Pred Value                    0.58857                  0.85994
Neg Pred Value                    0.94636                  0.97643
Prevalence                        0.09728                  0.10987
Detection Rate                    0.04802                  0.08873
Detection Prevalence              0.08159                  0.10319
Balanced Accuracy                 0.72821                  0.89570
                     Class: very damp grey soil
Sensitivity                               0.8375
Specificity                               0.9525
Pos Pred Value                            0.8437
Neg Pred Value                            0.9504
Prevalence                                0.2343
Detection Rate                            0.1963
Detection Prevalence                      0.2326
Balanced Accuracy                         0.8950
```

Avaliação do resultado: O modelo obteve acurácia de 86,73% o que é algo positivo pois indica que o modelo não está sofrendo de overfitting. Foi o melhor modelo obtido até o momento, mas isso deve ser devido a utilizar toda a base para treino e teste.

Salvando o modelo

In [38]:
```
saveRDS(modelo_final, "satelites_svm.rds")
```

2 Estimativa de Volumes de Árvores

Modelos de aprendizado de máquina são bastante usados na área da engenharia florestal (mensuração florestal) para, por exemplo, estimar o volume de madeira de árvores sem ser necessário abatê-las.

O processo é feito pela coleta de dados (dados observados) através do abate de algumas árvores, onde sua altura, diâmetro na altura do peito (dap), etc, são medidos de forma exata. Com estes dados, treina-se um modelo de AM que pode estimar o volume de outras árvores da população.

Os modelos, chamados **alométricos**, são usados na área há muitos anos e são baseados em regressão (linear ou não) para encontrar uma equação que descreve os dados. Por exemplo, o modelo de Spurr é dado por:

Volume = b0 + b1 * dap2 * Ht

Onde **dap** é o diâmetro na altura do peito (1,3metros), **Ht** é a altura total. Tem-se vários modelos alométricos, cada um com uma determinada característica, parâmetros, etc. Um modelo de regressão envolve aplicar os dados observados e encontrar b0 e b1 no modelo apresentado, gerando assim uma equação que pode ser usada para prever o volume de outras árvores.

Dado o arquivo **Volumes.csv**, que contém os dados de observação, escolha um modelo de aprendizado de máquina com a melhor estimativa, a partir da estatística de correlação.

Tarefas:

- Carregar o arquivo Volumes.csv (http://www.razer.net.br/datasets/Volumes.csv)
- Eliminar a coluna NR, que só apresenta um número sequencial
- Criar partição de dados: treinamento 80%, teste 20%
- Usando o pacote "caret", treinar os modelos: Random Forest (rf), SVM (svmRadial), Redes Neurais (neuralnet) e o modelo alométrico de SPURR
- O modelo alométrico é dado por: Volume = b0 + b1 *dap2* Ht

  alom <- nls(VOL ~ b0 + b1*DAP*DAP*HT, dados, start=list(b0=0.5, b1=0.5))

- Efetue as predições nos dados de teste
- Crie funções e calcule as seguintes métricas entre a predição e os dados observados
  - Coeficiente de determinação: R2
  - Erro padrão da estimativa: Syx
  - Syx%
- Escolha o melhor modelo

In [11]:
```
#Carregando os dados

library("caret")

#carregando arquivo Volumes.csv e removendo a coluna NR
df <- read.csv("http://www.razer.net.br/datasets/Volumes.csv",sep=";", dec=",")
df$NR <- NULL
head(df)
```

DAP	HT	HP	VOL
34.0	27.00	1.80	0.8971441
41.5	27.95	2.75	1.6204441
29.6	26.35	1.15	0.8008181
34.3	27.15	1.95	1.0791682
34.5	26.20	1.00	0.9801112
29.9	27.10	1.90	0.9067022

In [12]:
```
tail(df)
```

	DAP	HT	HP	VOL
95	31.5	23.50	2.50	0.9221653
96	33.1	25.75	2.65	1.0966956
97	31.0	25.70	2.60	1.0514350
98	43.0	27.90	2.70	2.0090605
99	40.0	24.20	1.10	1.7411209
100	38.0	27.65	2.45	1.5336724

In [13]:
```
# Repartir is dadis em 80% e 20%
indices_reg <- createDataPartition(df$VOL, p=0.8, list=F)
treino_reg <- df[indices_reg,]
teste_reg <- df[-indices_reg,]
```

Treinar e testar modelo com random forest

In [14]:
```
# criando semente pseudo aleatória para verificação futura
set.seed(1)
# treinando o modelo
rf_reg <- train(VOL ~ ., data=treino_reg, method="rf")
```

```
note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .
```

In [26]:
```
# realizando predição com a base de teste
predicao_reg.rf <- predict(rf_reg, teste_reg)
```

In [31]:
```
summary(predicao_reg.rf)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.7527  1.1287  1.3114  1.3478  1.4095  2.2681 
```

Treinar e testar modelo com SVM

In [16]:
```
# criando semente pseudo aleatória para verificação futura
set.seed(1)
# treinando o modelo
svm_reg <- train(VOL ~ ., data=treino_reg, method="svmRadial")
```

In [17]:
```
# realizando predição com a base de teste
predicao_reg.svm <- predict(svm_reg, teste_reg)
```

Treinar e testar modelo com redes neurais

In [138…
```
# criando semente pseudo aleatória para verificação futura
set.seed(1)
# treinando o modelo
library(neuralnet)
rna_reg <- neuralnet(VOL ~ ., data=treino_reg, hidden=c(2,1), linear.output=FALSE, threshold=0.01)
```

In [140…
```
# realizando predição com a base de teste
predicao_reg.rna <- predict(rna_reg, teste_reg)
```

Treinar e testar modelo com modelo alométrico de SPURR

In [28]:
```
# criando semente pseudo aleatória para verificação futura
set.seed(1)
# treinando o modelo
alom_reg <- nls(VOL ~ b0 + b1*DAP*DAP*HT, data=treino_reg, start=list(b0 = 0.5, b1=0.5))
```

In [29]:
```
# realizando predição com a base de teste
predicao_reg.alom <- predict(alom_reg, teste_reg)
```

Criando funções para calcular métricas entre as predições dos dados observados

Coeficinete de determinação: R^2

In [116…
```r
r2 <- function (valor_observado, valor_predito){
    return(1-(sum((valor_observado-valor_predito)^2)/sum((valor_observado-mean(valor_observado))^2)))
}
```

In [121…
```r
#Testando a função
x = c(1,2,3)
y = c(1.5,2.5,3.5)
r2(x, y)
```

0.625

Erro padrão da estimativa: S_yx

In [99]:
```r
syx <- function(valor_observado, valor_predito){
    return(sqrt(sum((valor_observado-valor_predito)^2)/(length(valor_observado)-2)))
}
```

In [122…
```r
#Testando a função
x = c(1,2,3)
y = c(1.5,2.5,3.5)
syx(x, y)
```

0.866025403784439

Erro padrão da estimativa em %: $S_yx\%$

In [110…
```r
syx_porc <- function(valor_observado, valor_predito){
    return(syx(valor_observado, valor_predito)/mean(valor_observado)*100)
}
```

In [123…
```r
#Testando a função
x = c(1,2,3)
y = c(1.5,2.5,3.5)
syx_porc(x, y)
```

43.3012701892219

Utilizando funções criadas para calcular métricas entre as predições dos dados observados

Coeficinete de determinação: R^2

In [117…
```
# Random forest
r2(teste_reg$VOL, predicao_reg.rf)
```

0.889632210301559

In [118…
```
# SVM
r2(teste_reg$VOL, predicao_reg.svm)
```

0.597887498177314

In [142…
```
# rna
r2(teste_reg$VOL, predicao_reg.rna)
```

-0.663964969377359

In [125…
```
# modelo Alometrico
r2(teste_reg$VOL, predicao_reg.alom)
```

0.901891629360885

Avaliação: O modelo alométrico se saiu melhor que os outros modelos, pois está mais próximo de 1. Nota: ainda não consegui interpretar o sinal negativo em RNA.

Erro padrão da estimativa: S_yx

In [126…
```
# Random forest
syx(teste_reg$VOL, predicao_reg.rf)
```

0.177310703060105

In [127…
```
# SVM
syx(teste_reg$VOL, predicao_reg.svm)
```

0.338444582805524

In [143…
```
# rna
syx(teste_reg$VOL, predicao_reg.rna)
```

0.688471344498325

In [129…
```
# modelo Alometrico
syx(teste_reg$VOL, predicao_reg.alom)
```

0.167173259699492

Avaliação: O modelo alométrico se saiu melhor que os outros modelos, pois está mais próximo de 0. Nota: RNA parece ter tido o pior resultado.

Erro padrão da estimativa em %: $S_yx\%$

In [130…
```
# Random forest
syx_porc(teste_reg$VOL, predicao_reg.rf)
```

12.5551066908747

In [131…
```
# SVM
syx_porc(teste_reg$VOL, predicao_reg.svm)
```

23.9647566262908

In [144…
```
# rna
syx_porc(teste_reg$VOL, predicao_reg.rna)
```

48.749630082152

In [133…
```
# modelo Alometrico
syx_porc(teste_reg$VOL, predicao_reg.alom)
```

11.8372894312925

Avaliação: O modelo alométrico se saiu melhor que os outros modelos, pois está mais próximo de 0. RNA indica ser a pior.

Após avaliar os resultados obtidos foi escolhido o modelo Alométrico de SPURR