Code for Predictive analysis on Hospital Readmission data

```
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(nnet)
library(SDMTools)
library(tree)
library(naivebayes)
library(e1071)
library(Metrics)
```

```
##
## Attaching package: 'Metrics'
```

```
## The following object is masked from 'package:SDMTools':
##
##     auc
```

```
library(neuralnet)
library(class)
library(stats)
library(arulesViz)
```

```
## Loading required package: arules
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
## Loading required package: grid
```

```
classify <- function(x) {
        value = -1
        if (startsWith(x, "E"))
        {
                value = 19
        }
        else if (startsWith(x, "V"))
        {
                value = 20
        }
        else
        {
                x = as.numeric(x)
                if (x < 140) {
                        value = 1
                }
                else if (x >= 140 && x < 240) {
                        value = 2
                }
                else if (x >= 240 && x < 280) {
                        value = 3
                }
                else if (x >= 280 && x < 290) {
                        value = 4
                }
                else if (x >= 290 && x < 320) {
                        value = 5
                }
                else if (x >= 320 && x < 360) {
                        value = 6
                }
                else if (x >= 360 && x < 390) {
                        value = 7
                }
                else if (x >= 390 && x < 460) {
                        value = 8
                }
                else if (x >= 460 && x < 520) {
                        value = 9
                }
                else if (x >= 520 && x < 580) {
                        value = 10
                }
                else if (x >= 580 && x < 630) {
                        value = 11
                }
                else if (x >= 630 && x < 680) {
```

```
                value = 12
            }
            else if (x >= 680 && x < 710) {
                value = 13
            }
            else if (x >= 710 && x < 740) {
                value = 14
            }
            else if (x >= 740 && x < 760) {
                value = 15
            }
            else if (x >= 760 && x < 780) {
                value = 16
            }
            else if (x >= 780 && x < 800) {
                value = 17
            }
            else if (x >= 800 && x < 1000) {
                value = 18
            }
        }
        value
}

maxidx <- function(arr) {
        return(  which(arr == max(arr)) )
}
```

```
#Fully cleaned and imputed dataset
df <- read.csv("diabetes_clean.csv", header = TRUE, strip.white = TRUE, na.strings =
c("NA", "?"," ","."))
df$readmitted <- as.factor(df$readmitted)
#df$readmitted <- ifelse(df$readmitted == df$readmitted[1], 0,1)
```

```
train <- sample(1:nrow(df), 0.8*nrow(df)) # Split the data into 80:20 ratio for cross
validation
train_data <- df[train,] # Training data
test_data <- df[-train,] # Test data
```

```
####### Main Task: Readmitted or not #######
# Model 1: Generalised Logistic Regression
model.logit <- glm(readmitted~., data=train_data[,-1], family=binomial(link='logit'))
pred.logit <- predict(model.logit,test_data, type = "response")
pred.logit <- ifelse(pred.logit > 0.5, 1, 0)
pred.logit.2 <- ifelse(pred.logit == 1, "TRUE", "FALSE")
mean(pred.logit==test_data$readmitted) # Accuracy: 64.6%
```

```
## [1] 0.6415
```

```
# Model 2: Random forest
df$readmitted <- ifelse(df$readmitted == 1, "TRUE", "FALSE")
df$readmitted <- as.factor(df$readmitted)
model.rf1 <-randomForest(readmitted~., data=train_data[,-1], ntree=10, na.action=na.e
xclude, importance=T,proximity=T)
print(model.rf1) #error rate: 42.08%
```

```
##
## Call:
##  randomForest(formula = readmitted ~ ., data = train_data[, -1],      ntree = 10,
importance = T, proximity = T, na.action = na.exclude)
##                Type of random forest: classification
##                      Number of trees: 10
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 41.27%
## Confusion matrix:
##      0    1 class.error
## 0 3210 1524   0.3219265
## 1 1743 1439   0.5477687
```

```
model.rf2 <-randomForest(readmitted~., data=train_data, ntree=20, na.action=na.exclud
e, importance=T,proximity=T)
print(model.rf2) #error rate: 40.21%
```

```
##
## Call:
##  randomForest(formula = readmitted ~ ., data = train_data, ntree = 20,      import
ance = T, proximity = T, na.action = na.exclude)
##                Type of random forest: classification
##                      Number of trees: 20
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 40.33%
## Confusion matrix:
##      0    1 class.error
## 0 3461 1326   0.2770002
## 1 1900 1312   0.5915318
```

```
model.rf3 <-randomForest(readmitted~., data=train_data, ntree=30, na.action=na.exclud
e, importance=T,proximity=T)
print(model.rf3) #error rate: 39.21%
```

```
##
## Call:
##  randomForest(formula = readmitted ~ ., data = train_data, ntree = 30,      import
ance = T, proximity = T, na.action = na.exclude)
##                Type of random forest: classification
##                      Number of trees: 30
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 40.35%
## Confusion matrix:
##      0    1 class.error
## 0 3470 1317   0.2751201
## 1 1911 1302   0.5947712
```

```
model.rf4 <-randomForest(readmitted~., data=train_data, ntree=40, na.action=na.exclud
e, importance=T,proximity=T)
print(model.rf4) #error rate: 38.32%
```

```
##
## Call:
##  randomForest(formula = readmitted ~ ., data = train_data, ntree = 40,      import
ance = T, proximity = T, na.action = na.exclude)
##                Type of random forest: classification
##                      Number of trees: 40
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 39.61%
## Confusion matrix:
##      0    1 class.error
## 0 3595 1192   0.2490077
## 1 1977 1236   0.6153128
```

```
model.rf5 <-randomForest(readmitted~., data=train_data, ntree=50, na.action=na.exclud
e, importance=T,proximity=T)
print(model.rf5) #error rate: 38.57%
```

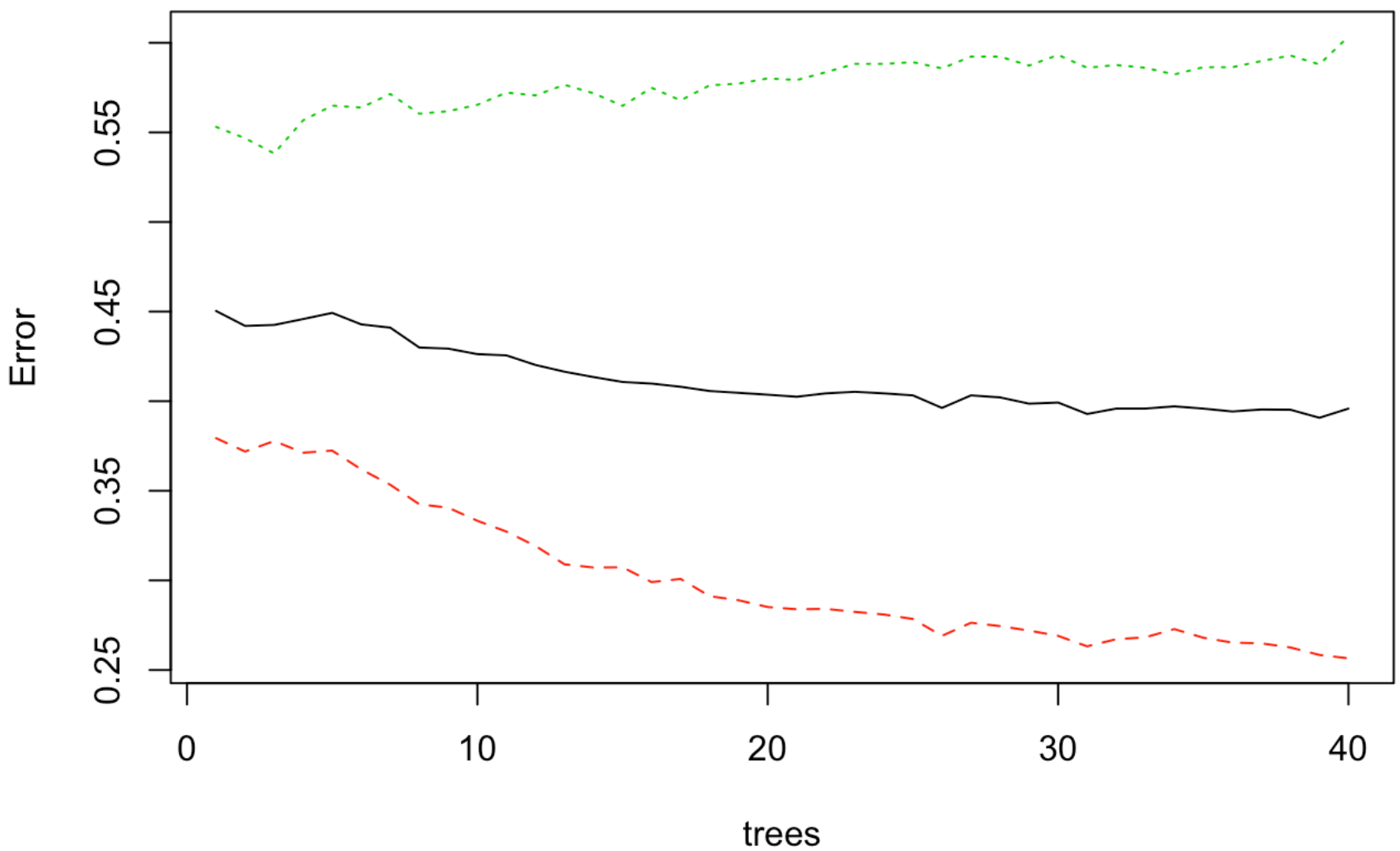
```
##
## Call:
##  randomForest(formula = readmitted ~ ., data = train_data, ntree = 50,      import
ance = T, proximity = T, na.action = na.exclude)
##                Type of random forest: classification
##                      Number of trees: 50
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 39.55%
## Confusion matrix:
##      0    1 class.error
## 0 3626 1161   0.2425319
## 1 2003 1210   0.6234049
```

```
model.rf <- randomForest(readmitted~., data=train_data, ntree=40, mtry = 8, na.action
=na.exclude, importance=T,proximity=T)
pred.rf <- predict(model.rf, test_data)
mean(pred.rf==test_data$readmitted) # Accuracy: 63.55%
```

```
## [1] 0.628
```

Including Plots

You can also embed plots, for example:




```
model.nn <- nnet(readmitted ~., data=train_data[,-1], size=5, maxit=1000)
```

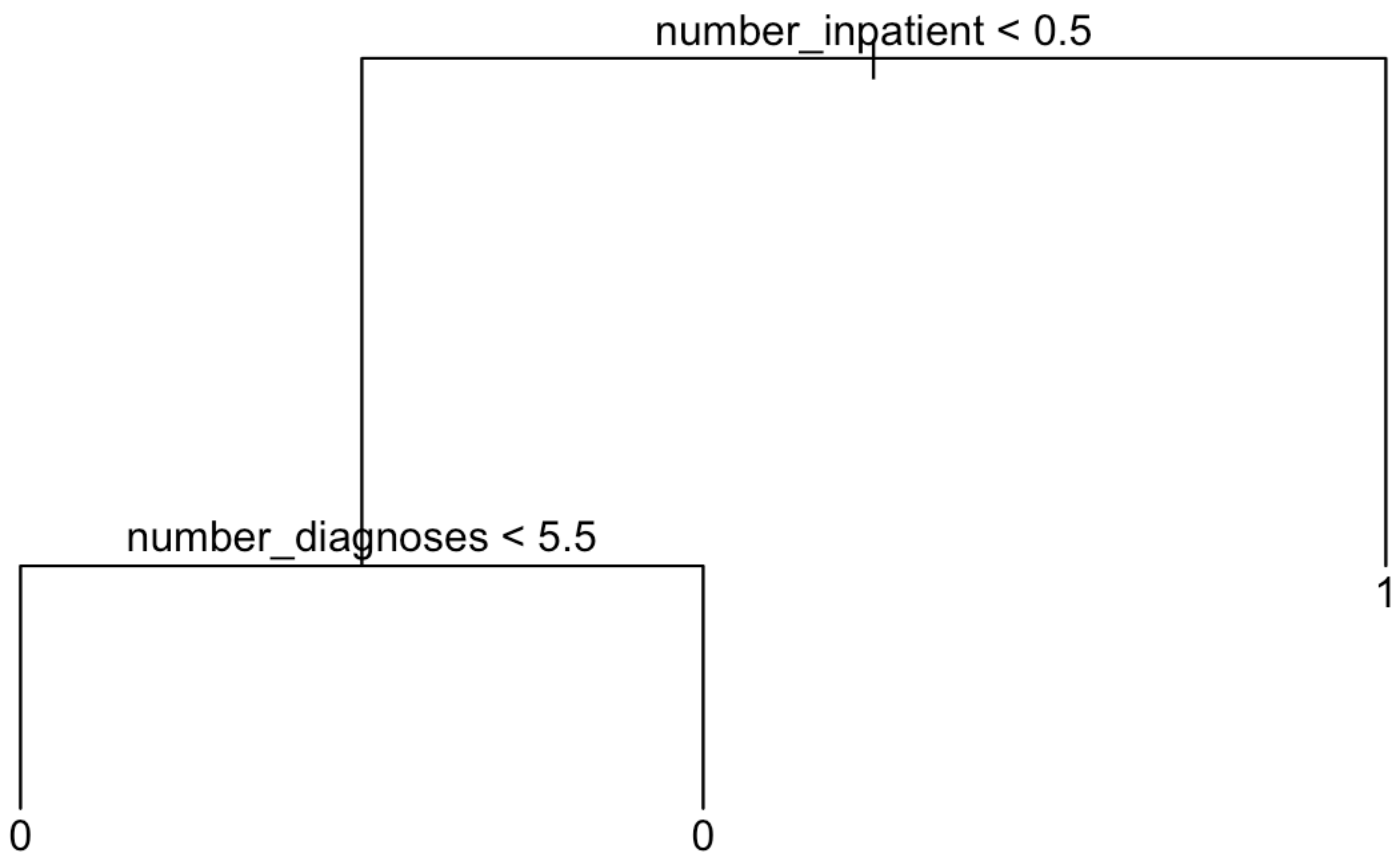
```
## # weights:  241
## initial  value 5691.154179
## iter  10 value 5384.183970
## iter  20 value 5350.282977
## iter  30 value 5278.626686
## iter  40 value 5253.595735
## iter  50 value 5196.914274
## iter  60 value 5105.847010
## iter  70 value 5053.733831
## iter  80 value 5031.836426
## iter  90 value 5012.502336
## iter 100 value 4998.684704
## iter 110 value 4984.222423
## iter 120 value 4963.893561
## iter 130 value 4955.588286
## iter 140 value 4953.016323
## iter 150 value 4952.943835
## final  value 4952.943285
## converged
```

```
pred.nn <- predict(model.nn, test_data,type= "raw")
pred.nn <- ifelse(pred.nn > 0.5, "TRUE", "FALSE")
mean(pred.nn==test_data$readmitted) # Accuracy: 63.35%
```

```
## [1] 0
```

```
#Model 4: Decision tree
df$readmitted <- as.factor(df$readmitted)
model.tree <- tree(readmitted~., data = train_data[,-1])
pred.tree <- predict(model.tree, test_data)
pred.response <- ifelse(pred.tree > 0.5, "FALSE", "TRUE")
mean(test_data$readmitted != pred.response) # Accuracy: 50%
```

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



```
# Model 5: Naive Bayes
train_data$readmitted <- as.factor(train_data$readmitted)
test_data$readmitted <- as.factor(test_data$readmitted)
model.nb <- naive_bayes(train_data[,-1], train_data$readmitted, laplace = 1, usekerne
l = T, prior = NULL)
pred.nb <- predict(model.nb, test_data, type = "prob", threshold = 0.01, eps = 0.1)
idx.nb <- apply(pred.nb, c(1), maxidx)
actual_result <- ifelse(df$readmitted == df$readmitted[1], 0,1)
mean(idx.nb-1 == actual_result) # Accuracy: 52.15%
```

```
## [1] 0.5221
```

```
# Model 6: SVM
model.svm <- svm(readmitted~., data = train_data, kernel = "linear",
                 type = "C-classification", cross = 10, cost = 0.01, gamma = 1000)
pred.svm <- predict(model.svm, test_data, decision.values = F)
mean(pred.svm == test_data$readmitted) # Accuracy: 63.95%
```

```
## [1] 0.6395
```

```
####### Task 1: Time in hospital #######
model.lm.steps <- step(lm(time_in_hospital~., data=train_data), direction = "both")
```

```
## Start:  AIC=14763.51
## time_in_hospital ~ rowID + race + gender + age + num_lab_procedures +
##     num_procedures + num_medications + number_outpatient + number_emergency +
##     number_inpatient + diag_1 + diag_2 + diag_3 + number_diagnoses +
##     max_glu_serum + A1Cresult + metformin + glimepiride + glipizide +
##     glyburide + pioglitazone + rosiglitazone + insulin + change +
##     diabetesMed + readmitted
##
##                      Df Sum of Sq   RSS   AIC
## - glimepiride         3       2.3 50045 14758
## - insulin             3      24.3 50067 14761
## - diabetesMed         1       0.0 50043 14762
## - change              1       1.9 50045 14762
## - rowID               1       2.8 50046 14762
## - readmitted          1       3.1 50046 14762
## <none>                            50043 14764
## - diag_3              1      21.1 50064 14765
## - number_emergency    1      22.8 50066 14765
## - gender              1      23.1 50066 14765
## - num_procedures      1      23.4 50066 14765
## - diag_2              1      41.0 50084 14768
## - metformin           3      89.4 50132 14772
## - pioglitazone        3      95.6 50138 14773
```

```
## - rosiglitazone       3     103.8 50147 14774
## - glipizide           3     113.0 50156 14776
## - race                4     153.2 50196 14780
## - glyburide           3     149.4 50192 14781
## - A1Cresult           3     160.9 50204 14783
## - number_inpatient    1     152.1 50195 14786
## - diag_1              1     170.8 50214 14789
## - number_outpatient   1     207.9 50251 14795
## - max_glu_serum       3     281.7 50325 14802
## - number_diagnoses    1     709.5 50752 14874
## - age                 1     791.9 50835 14887
## - num_lab_procedures  1    1636.5 51679 15019
## - num_medications     1    7851.1 57894 15927
##
## Step:  AIC=14757.88
## time_in_hospital ~ rowID + race + gender + age + num_lab_procedures +
##     num_procedures + num_medications + number_outpatient + number_emergency +
##     number_inpatient + diag_1 + diag_2 + diag_3 + number_diagnoses +
##     max_glu_serum + A1Cresult + metformin + glipizide + glyburide +
##     pioglitazone + rosiglitazone + insulin + change + diabetesMed +
##     readmitted
##
##                      Df Sum of Sq   RSS   AIC
## - insulin             3      22.6 50068 14756
## - diabetesMed         1       0.3 50045 14756
## - change              1       1.1 50046 14756
## - rowID               1       2.8 50048 14756
## - readmitted          1       3.2 50048 14756
## <none>                            50045 14758
## - diag_3              1      21.4 50067 14759
## - number_emergency    1      22.7 50068 14760
## - gender              1      23.2 50068 14760
## - num_procedures      1      23.5 50069 14760
## - diag_2              1      40.6 50086 14762
## + glimepiride         3       2.3 50043 14764
## - metformin           3      91.1 50136 14766
## - pioglitazone        3      94.5 50140 14767
## - rosiglitazone       3     102.5 50148 14768
## - glipizide           3     112.2 50157 14770
## - race                4     153.6 50199 14774
## - glyburide           3     147.9 50193 14776
## - A1Cresult           3     160.5 50206 14778
## - number_inpatient    1     152.2 50197 14780
## - diag_1              1     170.9 50216 14783
## - number_outpatient   1     207.9 50253 14789
## - max_glu_serum       3     283.1 50328 14797
## - number_diagnoses    1     710.9 50756 14869
## - age                 1     790.7 50836 14881
## - num_lab_procedures  1    1638.7 51684 15014
## - num_medications     1    7860.5 57906 15923
```

```
##
## Step:  AIC=14755.5
## time_in_hospital ~ rowID + race + gender + age + num_lab_procedures +
##     num_procedures + num_medications + number_outpatient + number_emergency +
##     number_inpatient + diag_1 + diag_2 + diag_3 + number_diagnoses +
##     max_glu_serum + A1Cresult + metformin + glipizide + glyburide +
##     pioglitazone + rosiglitazone + change + diabetesMed + readmitted
##
##                      Df Sum of Sq   RSS   AIC
## - change              1       0.5 50068 14754
## - rowID               1       2.7 50071 14754
## - readmitted          1       2.7 50071 14754
## <none>                            50068 14756
## - diabetesMed         1      15.2 50083 14756
## - diag_3              1      21.3 50089 14757
## - gender              1      22.6 50090 14757
## - number_emergency    1      22.7 50091 14757
## - num_procedures      1      24.3 50092 14757
## + insulin             3      22.6 50045 14758
## - diag_2              1      41.3 50109 14760
## + glimepiride         3       0.6 50067 14761
## - pioglitazone        3      84.6 50152 14763
## - rosiglitazone       3      94.2 50162 14764
## - metformin           3     105.7 50174 14766
## - glipizide           3     110.0 50178 14767
## - race                4     148.0 50216 14771
## - glyburide           3     141.9 50210 14772
## - A1Cresult           3     155.7 50224 14774
## - number_inpatient    1     150.1 50218 14777
## - diag_1              1     166.9 50235 14780
## - number_outpatient   1     206.3 50274 14786
## - max_glu_serum       3     276.5 50344 14794
## - number_diagnoses    1     709.5 50777 14866
## - age                 1     809.5 50877 14882
## - num_lab_procedures  1    1628.2 51696 15010
## - num_medications     1    7941.0 58009 15931
##
## Step:  AIC=14753.57
## time_in_hospital ~ rowID + race + gender + age + num_lab_procedures +
##     num_procedures + num_medications + number_outpatient + number_emergency +
##     number_inpatient + diag_1 + diag_2 + diag_3 + number_diagnoses +
##     max_glu_serum + A1Cresult + metformin + glipizide + glyburide +
##     pioglitazone + rosiglitazone + diabetesMed + readmitted
##
##                      Df Sum of Sq   RSS   AIC
## - rowID               1       2.7 50071 14752
## - readmitted          1       2.8 50071 14752
## <none>                            50068 14754
## - diabetesMed         1      18.7 50087 14755
## - diag_3              1      21.3 50090 14755
```

```
## - gender                1      22.6 50091 14755
## - number_emergency      1      22.7 50091 14755
## + change                1       0.5 50068 14756
## - num_procedures        1      24.9 50093 14756
## + insulin               3      21.9 50046 14756
## - diag_2                1      41.7 50110 14758
## + glimepiride           3       0.7 50068 14760
## - pioglitazone          3      88.2 50156 14762
## - rosiglitazone         3      98.4 50167 14763
## - metformin             3     106.3 50175 14764
## - glipizide             3     109.6 50178 14765
## - race                  4     147.9 50216 14769
## - glyburide             3     141.5 50210 14770
## - A1Cresult             3     155.2 50224 14772
## - number_inpatient      1     150.1 50218 14776
## - diag_1                1     166.5 50235 14778
## - number_outpatient     1     207.1 50275 14785
## - max_glu_serum         3     276.0 50344 14792
## - number_diagnoses      1     709.1 50777 14864
## - age                   1     812.0 50880 14880
## - num_lab_procedures    1    1629.1 51697 15008
## - num_medications       1    8156.2 58225 15959
##
## Step:  AIC=14752
## time_in_hospital ~ race + gender + age + num_lab_procedures +
##     num_procedures + num_medications + number_outpatient + number_emergency +
##     number_inpatient + diag_1 + diag_2 + diag_3 + number_diagnoses +
##     max_glu_serum + A1Cresult + metformin + glipizide + glyburide +
##     pioglitazone + rosiglitazone + diabetesMed + readmitted
##
##                       Df Sum of Sq   RSS   AIC
## - readmitted           1       2.9 50074 14750
## <none>                             50071 14752
## - diabetesMed          1      18.6 50090 14753
## - diag_3               1      21.4 50092 14753
## + rowID                1       2.7 50068 14754
## - gender               1      22.5 50094 14754
## - number_emergency     1      22.7 50094 14754
## + change               1       0.5 50071 14754
## - num_procedures       1      25.1 50096 14754
## + insulin              3      21.9 50049 14754
## - diag_2               1      41.7 50113 14757
## + glimepiride          3       0.7 50070 14758
## - pioglitazone         3      88.3 50159 14760
## - rosiglitazone        3      98.2 50169 14762
## - metformin            3     106.0 50177 14763
## - glipizide            3     110.0 50181 14764
## - race                 4     147.8 50219 14768
## - glyburide            3     141.5 50213 14769
## - A1Cresult            3     154.8 50226 14771
```

```
## - number_inpatient    1     150.0 50221 14774
## - diag_1              1     166.3 50237 14776
## - number_outpatient   1     207.0 50278 14783
## - max_glu_serum       3     276.2 50347 14790
## - number_diagnoses    1     708.3 50779 14862
## - age                 1     810.6 50882 14878
## - num_lab_procedures  1    1632.3 51703 15007
## - num_medications     1    8154.8 58226 15957
##
## Step:  AIC=14750.46
## time_in_hospital ~ race + gender + age + num_lab_procedures +
##     num_procedures + num_medications + number_outpatient + number_emergency +
##     number_inpatient + diag_1 + diag_2 + diag_3 + number_diagnoses +
##     max_glu_serum + A1Cresult + metformin + glipizide + glyburide +
##     pioglitazone + rosiglitazone + diabetesMed
##
##                      Df Sum of Sq   RSS   AIC
## <none>                             50074 14750
## - diabetesMed         1      19.1 50093 14752
## - diag_3              1      21.1 50095 14752
## + readmitted          1       2.9 50071 14752
## - gender              1      22.2 50096 14752
## + rowID               1       2.8 50071 14752
## - number_emergency    1      23.4 50097 14752
## + change              1       0.5 50073 14752
## - num_procedures      1      25.4 50099 14752
## + insulin             3      21.6 50052 14753
## - diag_2              1      42.1 50116 14755
## + glimepiride         3       0.7 50073 14756
## - pioglitazone        3      89.8 50164 14759
## - rosiglitazone       3      98.5 50172 14760
## - metformin           3     106.8 50181 14762
## - glipizide           3     110.8 50185 14762
## - race                4     149.6 50223 14766
## - glyburide           3     140.5 50214 14767
## - A1Cresult           3     155.4 50229 14769
## - number_inpatient    1     147.3 50221 14772
## - diag_1              1     166.5 50240 14775
## - number_outpatient   1     210.0 50284 14782
## - max_glu_serum       3     276.5 50350 14788
## - number_diagnoses    1     707.0 50781 14861
## - age                 1     808.4 50882 14877
## - num_lab_procedures  1    1629.6 51703 15005
## - num_medications     1    8156.5 58230 15956
```

```
model.task1.lm <- lm(time_in_hospital ~ race + gender + age + num_lab_procedures +
                               num_procedures + num_medications + number_outpatient + n
umber_inpatient +
                               diag_1 + number_diagnoses + max_glu_serum + A1Cresult +
                               metformin + glipizide + glyburide + pioglitazone + rosig
litazone +
                               diabetesMed + readmitted, data = train_data)
pred.task1.lm <- predict(model.task1.lm,test_data)
rmse(test_data$time_in_hospital,pred.task1.lm) #RMSE: 2.198489
```

```
## [1] 2.535836
```

```
rmsle(test_data$time_in_hospital,pred.task1.lm) #RMSLE: 0.4379768
```

```
## [1] 0.4711861
```

```
# Model 2: Neural networks
model.task1.nn <- nnet(time_in_hospital ~ race + gender + age + num_lab_procedures +
                                 num_procedures + num_medications + number_outpatient +
number_inpatient +
                                 diag_1 + number_diagnoses + max_glu_serum + A1Cresult
+
                                 metformin + glipizide + glyburide + pioglitazone + ros
iglitazone +
                                 diabetesMed + readmitted, data=train_data, size=5, max
it=1000)
```

```
## # weights:  191
## initial  value 205241.623349
## final  value 167227.000000
## converged
```

```
pred.task1.nn <- predict(model.task1.nn, test_data)
rmse(test_data$time_in_hospital, pred.task1.nn) #4.061443
```

```
## [1] 4.584539
```

```
rmsle(test_data$time_in_hospital, pred.task1.nn) #0.9621453
```

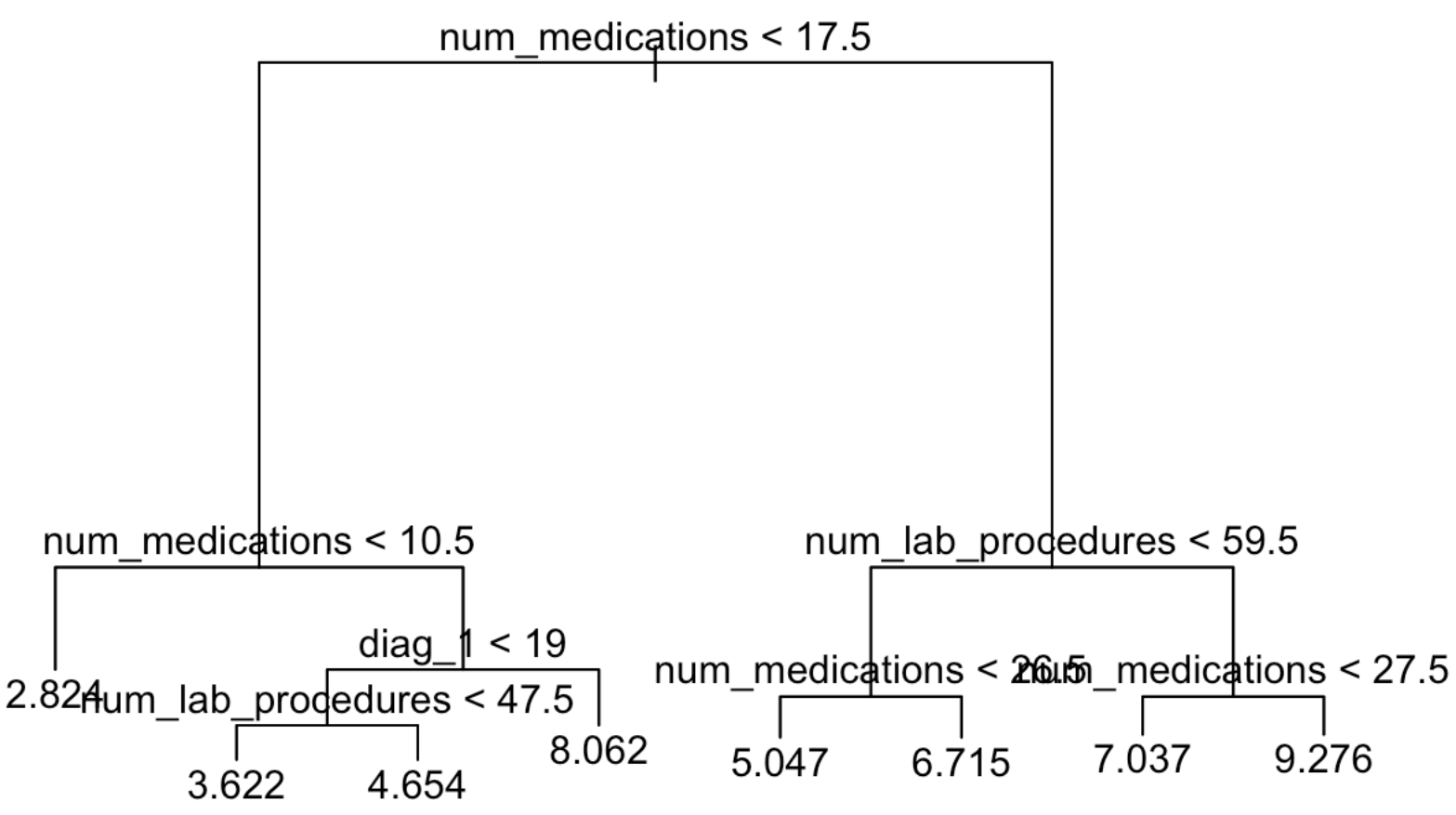
```
## [1] 1.013555
```

```
# Model 3: Decision tree
model.task1.tree <- tree(time_in_hospital ~ race + gender + age + num_lab_procedures
+
                                   num_procedures + num_medications + number_outpatient
+ number_inpatient +
                                   diag_1 + diag_2 + number_diagnoses + max_glu_serum +
A1Cresult +
                                   metformin + glipizide + glyburide + pioglitazone + r
osiglitazone +
                                   diabetesMed + readmitted, data = train_data)
pred.task1.tree <- predict(model.task1.tree, test_data)
rmse(test_data$time_in_hospital,pred.task1.tree) #2.276108
```

```
## [1] 2.651915
```

```
rmsle(test_data$time_in_hospital,pred.task1.tree) #0.4512615
```

```
## [1] 0.4803073
```



```
#Model 4: Random Forest
model.task1.rf1 <-randomForest(time_in_hospital ~ race + gender + age + num_lab_proce
dures +
                                    num_procedures + num_medications + number_outp
atient + number_inpatient +
                                    diag_1 + number_diagnoses + max_glu_serum + A1
Cresult +
                                    metformin + glipizide + glyburide + pioglitazo
ne + rosiglitazone +
                                    diabetesMed + readmitted,
                               data=train_data,ntree=10, na.action=na.exclude, import
ance=T,proximity=T)
print(model.task1.rf1) # % Var explained: 13.57%
```

```
##
## Call:
##  randomForest(formula = time_in_hospital ~ race + gender + age +      num_lab_proc
edures + num_procedures + num_medications + number_outpatient +      number_inpatient
+ diag_1 + number_diagnoses + max_glu_serum +      A1Cresult + metformin + glipizide
+ glyburide + pioglitazone +      rosiglitazone + diabetesMed + readmitted, data = tr
ain_data,      ntree = 10, importance = T, proximity = T, na.action = na.exclude)
##                Type of random forest: regression
##                      Number of trees: 10
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 7.459509
##                     % Var explained: 18.43
```

```
model.task1.rf2 <-randomForest(time_in_hospital ~ race + gender + age + num_lab_proce
dures +
                                    num_procedures + num_medications + number_outp
atient + number_inpatient +
                                    diag_1 + number_diagnoses + max_glu_serum + A1
Cresult +
                                    metformin + glipizide + glyburide + pioglitazo
ne + rosiglitazone +
                                    diabetesMed + readmitted,
                               data=train_data,ntree=20, na.action=na.exclude, import
ance=T,proximity=T)
print(model.task1.rf2) # % Var explained: 24.02%
```

```
##
## Call:
##  randomForest(formula = time_in_hospital ~ race + gender + age +      num_lab_proc
edures + num_procedures + num_medications + number_outpatient +      number_inpatient
+ diag_1 + number_diagnoses + max_glu_serum +      A1Cresult + metformin + glipizide
+ glyburide + pioglitazone +      rosiglitazone + diabetesMed + readmitted, data = tr
ain_data,      ntree = 20, importance = T, proximity = T, na.action = na.exclude)
##                Type of random forest: regression
##                      Number of trees: 20
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 6.552537
##                     % Var explained: 28.34
```

```
model.task1.rf3 <-randomForest(time_in_hospital ~ race + gender + age + num_lab_proce
dures +
                                    num_procedures + num_medications + number_outp
atient + number_inpatient +
                                    diag_1 + number_diagnoses + max_glu_serum + A1
Cresult +
                                    metformin + glipizide + glyburide + pioglitazo
ne + rosiglitazone +
                                    diabetesMed + readmitted,
                               data=train_data,ntree=30, na.action=na.exclude, import
ance=T,proximity=T)
print(model.task1.rf3) # % Var explained: 28.21%
```

```
##
## Call:
##  randomForest(formula = time_in_hospital ~ race + gender + age +      num_lab_proc
edures + num_procedures + num_medications + number_outpatient +      number_inpatient
+ diag_1 + number_diagnoses + max_glu_serum +      A1Cresult + metformin + glipizide
+ glyburide + pioglitazone +      rosiglitazone + diabetesMed + readmitted, data = tr
ain_data,      ntree = 30, importance = T, proximity = T, na.action = na.exclude)
##                Type of random forest: regression
##                      Number of trees: 30
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 6.188914
##                     % Var explained: 32.32
```

```
model.task1.rf4 <-randomForest(time_in_hospital ~ race + gender + age + num_lab_proce
dures +
                                    num_procedures + num_medications + number_outp
atient + number_inpatient +
                                    diag_1 + number_diagnoses + max_glu_serum + A1
Cresult +
                                    metformin + glipizide + glyburide + pioglitazo
ne + rosiglitazone +
                                    diabetesMed + readmitted,
                               data=train_data,ntree=40, na.action=na.exclude, import
ance=T,proximity=T)
print(model.task1.rf4) # % Var explained: 30.35%
```

```
##
## Call:
##  randomForest(formula = time_in_hospital ~ race + gender + age +      num_lab_proc
edures + num_procedures + num_medications + number_outpatient +      number_inpatient
+ diag_1 + number_diagnoses + max_glu_serum +      A1Cresult + metformin + glipizide
+ glyburide + pioglitazone +      rosiglitazone + diabetesMed + readmitted, data = tr
ain_data,      ntree = 40, importance = T, proximity = T, na.action = na.exclude)
##                Type of random forest: regression
##                      Number of trees: 40
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 6.209213
##                     % Var explained: 32.1
```

```
model.task1.rf5 <-randomForest(time_in_hospital ~ race + gender + age + num_lab_proce
dures +
                                    num_procedures + num_medications + number_outp
atient + number_inpatient +
                                    diag_1 + number_diagnoses + max_glu_serum + A1
Cresult +
                                    metformin + glipizide + glyburide + pioglitazo
ne + rosiglitazone +
                                    diabetesMed + readmitted,
                               data=train_data,ntree=50, na.action=na.exclude, import
ance=T,proximity=T)
print(model.task1.rf5) # % Var explained: 31.43%
```

```
##
## Call:
##  randomForest(formula = time_in_hospital ~ race + gender + age +      num_lab_proc
edures + num_procedures + num_medications + number_outpatient +      number_inpatient
+ diag_1 + number_diagnoses + max_glu_serum +      A1Cresult + metformin + glipizide
+ glyburide + pioglitazone +      rosiglitazone + diabetesMed + readmitted, data = tr
ain_data,      ntree = 50, importance = T, proximity = T, na.action = na.exclude)
##                Type of random forest: regression
##                      Number of trees: 50
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 6.05086
##                     % Var explained: 33.83
```

```
model.task1.rf <- randomForest(time_in_hospital~gender+readmitted+change+diabetesMed+
glimepiride,
                               data=train_data, ntree=50,mtry = 27, na.action=na.excl
ude, importance=T,
                               proximity=T)
```

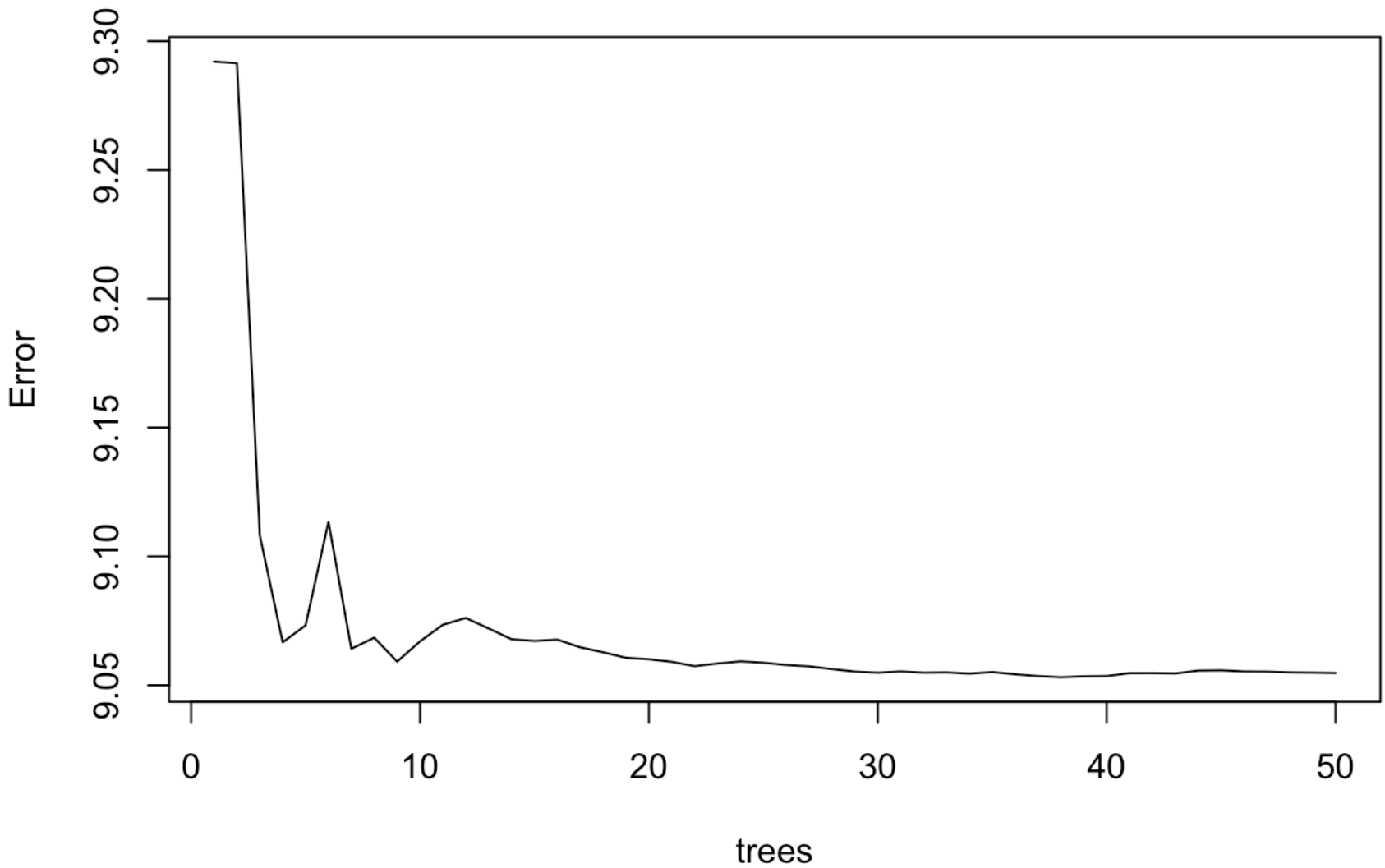
```
## Warning in randomForest.default(m, y, ...): invalid mtry: reset to within
## valid range
```

```
pred.task1.rf <- predict(model.task1.rf, test_data)
rmse(test_data$time_in_hospital,pred.task1.rf)#2.558702
```

```
## [1] 3.002373
```

```
rmsle(test_data$time_in_hospital,pred.task1.rf)#0.5073352
```

```
## [1] 0.5486051
```




```
####### Task 2: Diagnoses #######
# Model 1: SVM
test_data$diag_3 <- as.factor(test_data$diag_3)

model.task2.svm.1 <- svm(diag_2~., train_data[,-c(1,14)])
pred.task2.svm.1 <- predict(model.task2.svm.1, newdata = test_data)
mean(round(pred.task2.svm.1)==test_data$diag_2)
```

```
## [1] 0.216
```

```
model.task2.svm.2 <- svm(diag_3~., train_data[,-c(1)])
pred.task2.svm.2 <- predict(model.task2.svm.2, newdata = test_data)
mean(round(pred.task2.svm.2)==test_data$diag_3)
```

```
## [1] 0.146
```

```
# Model 4: Artificial Neural Networks
train_data$diag_2 <- as.factor(train_data$diag_2)
test_data$diag_2 <- as.factor(test_data$diag_2)
train_data$diag_3 <- as.factor(train_data$diag_3)
test_data$diag_3 <- as.factor(test_data$diag_3)
model.task2.nn.1 <- nnet(train_data$diag_2 ~ ., data=train_data[,-c(1,14)], size=5, m
axit=1000)
```

```
## # weights:  344
## initial  value 26125.254084
## iter  10 value 17907.001787
## iter  20 value 17515.309899
## iter  30 value 17407.144814
## final  value 17407.143040
## converged
```

```
pred.task2.nn.1 <- predict(model.task2.nn.1,newdata = test_data[,-1], type = "class")
mean(as.character(pred.task2.nn.1)==as.character(test_data$diag_2)) # 39.19%
```

```
## [1] 0.322
```

```
model.task2.nn.2 <- nnet(train_data$diag_3 ~ ., data=train_data[,-c(1)], size=5, maxi
t=1000)
```

```
## # weights:  434
## initial  value 25657.912758
## iter  10 value 17628.187188
## iter  20 value 17571.477933
## iter  30 value 17540.492200
## iter  40 value 17463.885054
## iter  50 value 17346.250838
## iter  60 value 17249.069796
## iter  70 value 17148.657992
## iter  80 value 16997.856199
## iter  90 value 16905.314547
## iter 100 value 16818.638886
## iter 110 value 16767.720273
## iter 120 value 16719.651600
## iter 130 value 16686.177590
## iter 140 value 16670.620863
## iter 150 value 16661.841484
## iter 160 value 16654.321058
## iter 170 value 16645.218726
## iter 180 value 16631.306330
## iter 190 value 16609.954512
## iter 200 value 16595.515265
```

```
## iter 210 value 16584.913669
## iter 220 value 16575.047082
## iter 230 value 16569.829832
## iter 240 value 16563.733597
## iter 250 value 16560.206923
## iter 260 value 16559.562481
## iter 270 value 16558.387904
## iter 280 value 16556.263641
## iter 290 value 16554.601510
## iter 300 value 16552.646924
## iter 310 value 16551.502135
## iter 320 value 16550.638708
## iter 330 value 16549.506735
## iter 340 value 16546.379256
## iter 350 value 16543.828600
## iter 360 value 16541.901460
## iter 370 value 16540.741912
## iter 380 value 16540.111271
## iter 390 value 16539.370882
## iter 400 value 16538.556998
## iter 410 value 16538.001208
## iter 420 value 16536.866869
## iter 430 value 16535.258696
## iter 440 value 16533.368688
## iter 450 value 16531.924404
## iter 460 value 16529.010544
## iter 470 value 16527.223516
## iter 480 value 16526.466365
## iter 490 value 16526.446555
## iter 500 value 16526.404873
## iter 510 value 16526.353525
## iter 520 value 16526.262702
## iter 530 value 16526.152769
## iter 540 value 16526.101126
## iter 550 value 16525.995069
## iter 560 value 16525.885057
## iter 570 value 16525.695497
## iter 580 value 16525.458677
## iter 590 value 16525.225026
## iter 600 value 16525.108983
## iter 610 value 16525.013047
## iter 620 value 16524.914043
## iter 630 value 16524.773338
## iter 640 value 16524.678680
## iter 650 value 16524.628309
## iter 660 value 16524.569866
## iter 670 value 16524.517467
## iter 680 value 16524.483213
## iter 690 value 16524.425608
## iter 700 value 16524.374427
```

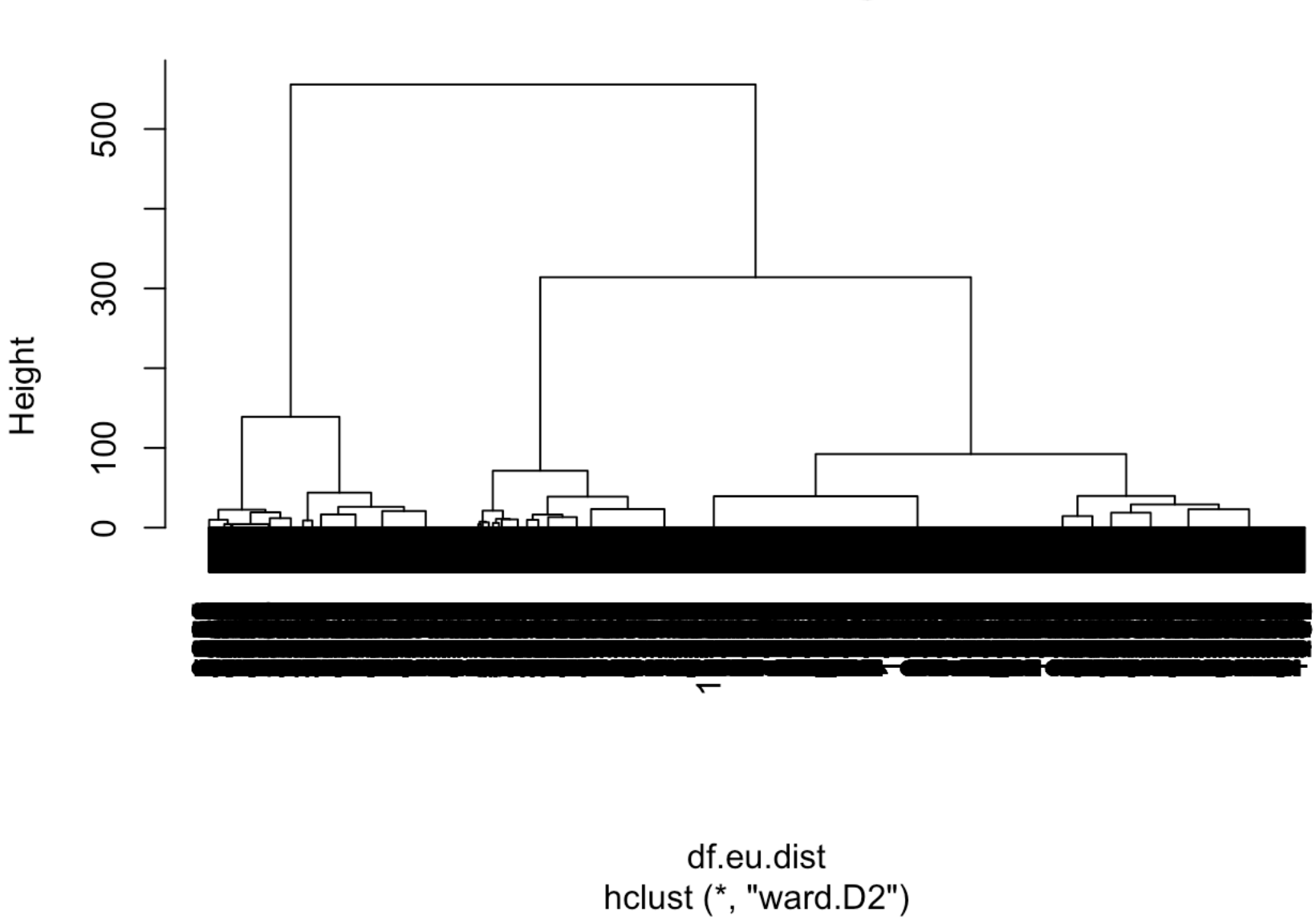
```
## iter 710 value 16524.288916
## iter 720 value 16524.171271
## iter 730 value 16524.012797
## iter 740 value 16523.838908
## iter 740 value 16523.838908
## final  value 16523.838908
## converged
```

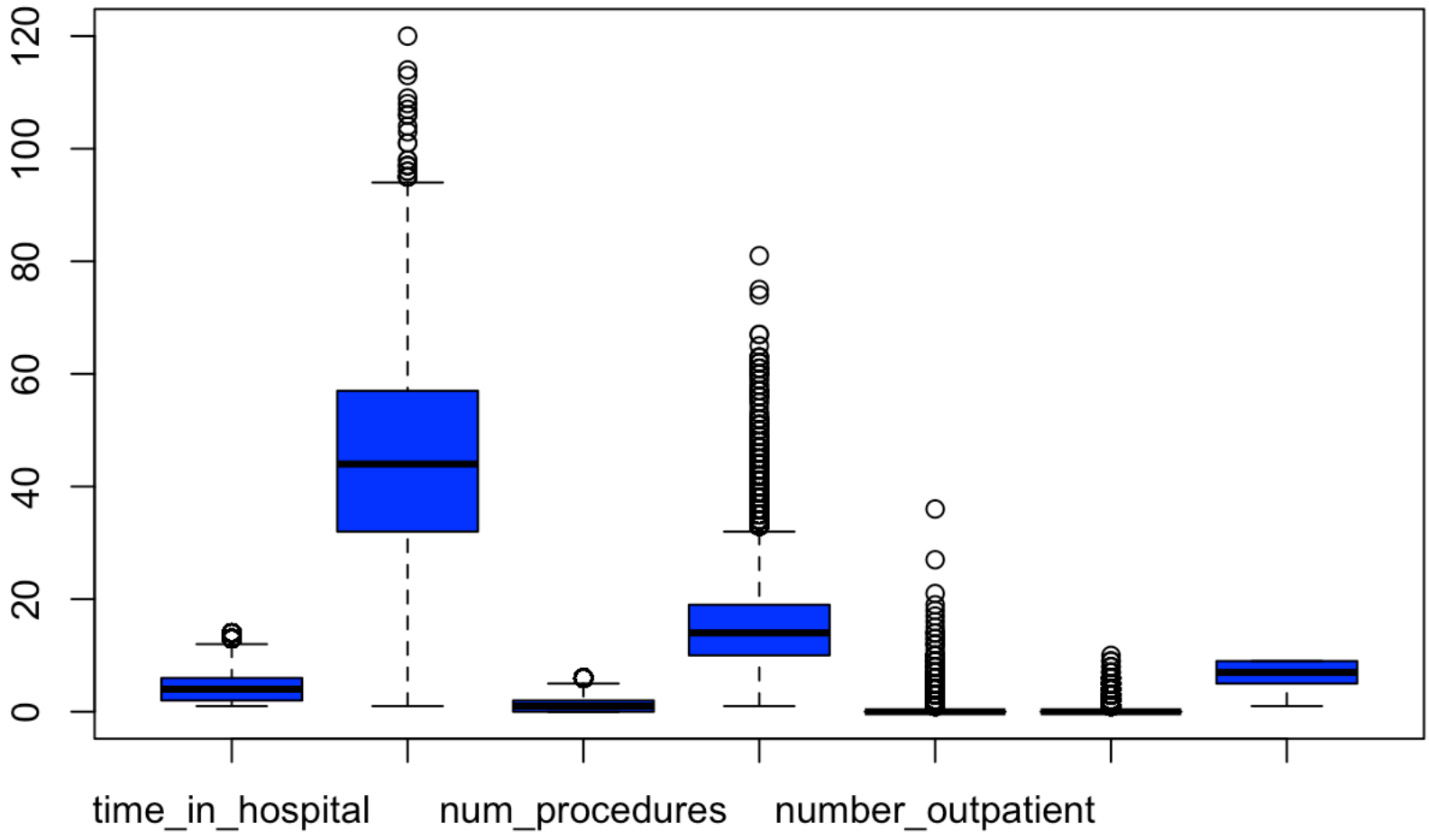
```
pred.task2.nn.2 <- as.factor(predict(model.task2.nn.2,newdata = test_data, type = "cl
ass"))
mean(as.character(pred.task2.nn.2)==as.character(test_data$diag_3)) # 37.43%
```

```
## [1] 0.352
```

```
df$readmitted <- ifelse(df$readmitted == "TRUE", 1, 0)
df.eu.dist <- dist(df[,c(12,27)], method = "euclidean")
hClust1 <- hclust(df.eu.dist, method = "ward.D2")
```

```
plot(hClust1)
```






```
####### Pattern mining #######
df.initial <- read.csv("10kDiabetes.csv", header = TRUE, strip.white = TRUE, na.strin
gs = c("NA", "?"," ","."))
mining <- df
mining$age <- df.initial$age
mining$diag_1 <- as.factor(mining$diag_1)
mining$diag_2 <- as.factor(mining$diag_2)
mining$diag_3 <- as.factor(mining$diag_3)
mining[,c(1, 5, 6, 7, 8, 9, 10, 11, 15)] <- NULL
mining$readmitted <- df.initial$readmitted
mining$readmitted <- as.factor(mining$readmitted)
rules1 <- apriori(mining, parameter=list(minlen=2,supp=0.005,conf=0.8),
                  appearance=list (rhs=c("readmitted=FALSE","readmitted=TRUE"), defau
lt="lhs"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.005      2
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 50
##
## set item appearances ...[2 item(s)] done [0.00s].
## set transactions ...[115 item(s), 10000 transaction(s)] done [0.01s].
## sorting and recoding items ... [100 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 7 8
```

```
## Warning in apriori(mining, parameter = list(minlen = 2, supp = 0.005, conf
## = 0.8), : Mining stopped (time limit reached). Only patterns up to a length
## of 8 returned!
```

```
##  done [8.57s].
## writing ... [2244 rule(s)] done [0.18s].
## creating S4 object  ... done [0.24s].
```

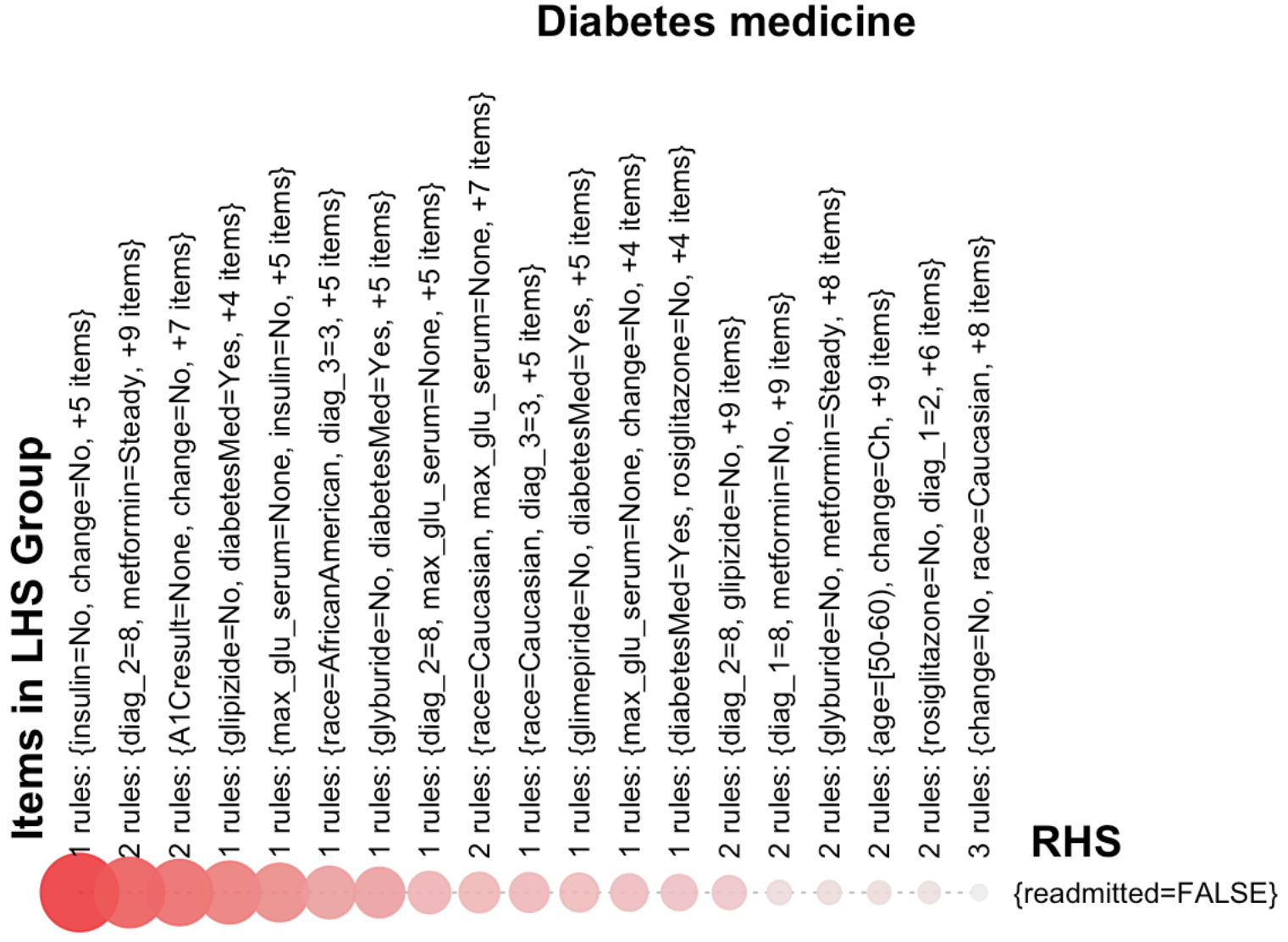
```
rules1.sort <- sort(rules1, by="lift")
subset.matrix<-is.subset(rules1.sort,rules1.sort)
subset.matrix[lower.tri(subset.matrix,diag=T)] <- 0
```

```
## Warning in `[<-`(`*tmp*`, as.vector(i), value = 0): x[.] <- val: x is
## "ngTMatrix", val not in {TRUE, FALSE} is coerced.
```

```
redudant<-colSums(subset.matrix) >= 1
rules1.pruned <- rules1.sort[!redudant]
rules.sub <- subset(rules1.pruned, subset = lhs %pin% "Male" & rhs %pin% "FALSE")
```

```
## Warning: Unknown control parameters: cex, itemLabels, arrowSize
```

```
## Available control parameters (with default values):
## main  =  Grouped Matrix for 29 Rules
## k     =  20
## rhs_max   =  10
## lhs_items     =  2
## aggr.fun  =  function (x, na.rm = FALSE)  UseMethod("median")
## col   =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F
0FFF", "#EE1212FF", "#EE1515FF", "#EE1818FF", "#EE1B1BFF", "#EE1E1EFF", "#EE2222FF",
"#EE2525FF", "#EE2828FF", "#EE2B2BFF", "#EE2E2EFF", "#EE3131FF", "#EE3434FF", "#EE373
7FF", "#EE3A3AFF", "#EE3D3DFF", "#EE4040FF", "#EE4444FF", "#EE4747FF", "#EE4A4AFF", "
#EE4D4DFF", "#EE5050FF", "#EE5353FF", "#EE5656FF", "#EE5959FF", "#EE5C5CFF", "#EE5F5F
FF", "#EE6262FF", "#EE6666FF", "#EE6969FF", "#EE6C6CFF", "#EE6F6FFF", "#EE7272FF", "#
EE7575FF",  "#EE7878FF", "#EE7B7BFF", "#EE7E7EFF", "#EE8181FF", "#EE8484FF", "#EE8888
FF", "#EE8B8BFF", "#EE8E8EFF", "#EE9191FF", "#EE9494FF", "#EE9797FF", "#EE9999FF", "#
EE9B9BFF", "#EE9D9DFF", "#EE9F9FFF", "#EEA0A0FF", "#EEA2A2FF", "#EEA4A4FF", "#EEA5A5F
F", "#EEA7A7FF", "#EEA9A9FF", "#EEABABFF", "#EEACACFF", "#EEAEAEFF", "#EEB0B0FF", "#E
EB1B1FF", "#EEB3B3FF", "#EEB5B5FF", "#EEB7B7FF", "#EEB8B8FF", "#EEBABAFF", "#EEBCBCFF
", "#EEBDBDFF", "#EEBFBFFF", "#EEC1C1FF", "#EEC3C3FF", "#EEC4C4FF", "#EEC6C6FF", "#EE
C8C8FF",  "#EEC9C9FF", "#EECBCBFF", "#EECDCDFF", "#EECFCFFF", "#EED0D0FF", "#EED2D2FF
", "#EED4D4FF", "#EED5D5FF", "#EED7D7FF", "#EED9D9FF", "#EEDBDBFF", "#EEDCDCFF", "#EE
DEDEFF", "#EEE0E0FF", "#EEE1E1FF", "#EEE3E3FF", "#EEE5E5FF", "#EEE7E7FF", "#EEE8E8FF"
, "#EEEAEAFF", "#EEECECFF", "#EEEEEEFF")
## reverse   =  TRUE
## xlab  =  NULL
## ylab  =  NULL
## legend    =  Size: lift  Color: lift
## spacing   =  -1
## panel.function    =  function (row, size, shading, spacing)  {     size[size == 0]
<- NA     shading[is.na(shading)] <- 1     grid.circle(x = c(1:length(size)), y = row
, r = size/2 * (1 - spacing), default.units = "native", gp = gpar(fill = shading, col
= shading, alpha = 0.9)) }
## gp_main   =  list(cex = 1.2, fontface = "bold", font = 2)
## gp_labels     =  list(cex = 0.8)
## gp_labs   =  list(cex = 1.2, fontface = "bold", font = 2)
## gp_lines  =  list(col = "gray", lty = 3)
## newpage   =  TRUE
## interactive   =  FALSE
## max.shading   =  NA
## verbose   =  FALSE
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.