Homework 6

Shun-Lung Chang, Dilip Hiremath

```
# load packages
library(Hmisc) # impute()
library(corrplot) # corrplot()
library(rpart) # rpart()
library(rpart.plot) # rpart.plot()
```

```
# load dataset
library(foreign)
timss <- read.spss('data/timss.sav',
                   use.value.labels = FALSE,
                   max.value.labels = Inf,
                   to.data.frame = TRUE)
```

1. First of all you take the eleven indicators (bsbgday1-6, bsbmday7, bsbsday8, bsbgday9, bsbgclub, bsbgpaid) that summarize outside school activities and you aim at identifying their factor structure. These variables comprise the answers of the students to the following questions: On a normal school day, how much time do you spend before or after school doing each of these things?

```
bsbgday1 I watch TV or videos
bsbgday2 I play computer games
bsbgday3 I playing or talk with friends
bsbgday4 I do jobs at home
bsbgday5 I play sports
bsbgday6 I read a book for enjoyment
bsbmday7 I study math
bsbsday8 I study science
bsgbday9 I study other subjects
bsbgclub I participate in clubs
bsbgpaid I work at a paid job
```

The given answering categories are coded as: 1 = no time, 2= less than one hour, 3 = one to two hours, 4= more than two but less than 4 hours, 5 = more than five hours.

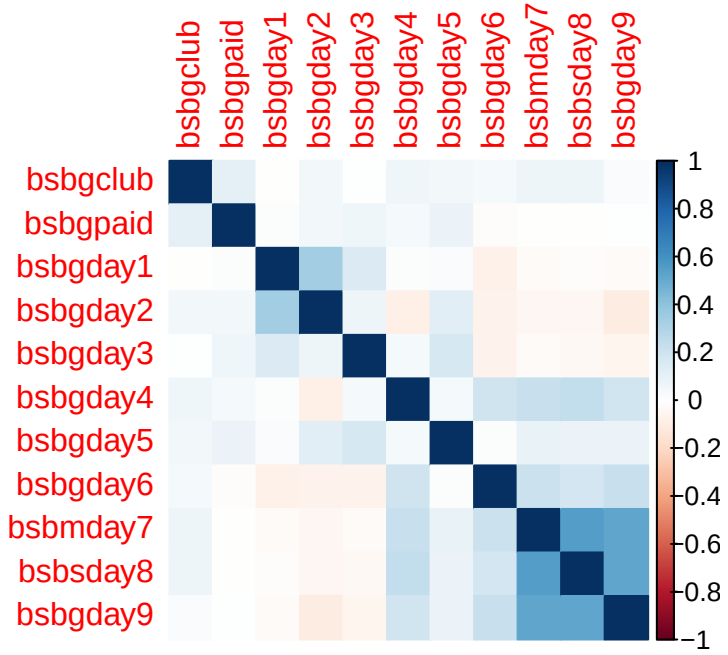
(a) (1 point) Check for missing values. Report how many missings are there in these eleven variables.

```
colSums(is.na(timss[, 7:17]))
```

```
bsbgclub bsbgpaid bsbgday1 bsbgday2 bsbgday3 bsbgday4 bsbgday5 bsbgday6
     320      286      182      272      215      204      198      207
bsbmday7 bsbsday8 bsbgday9
     169      171      166
```

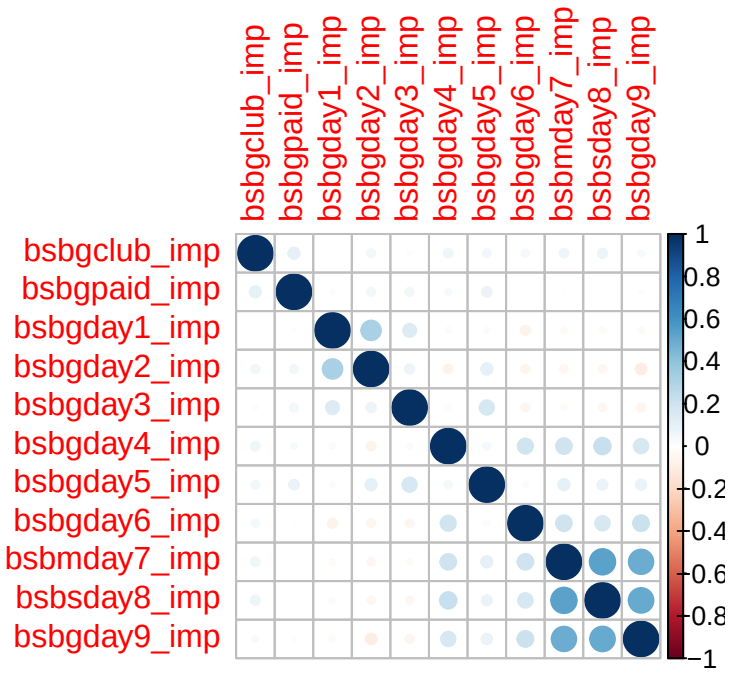
(b) (1 point) Calculate the correlation matrix for these eleven variables. Which variables have the highest correlation coefficient?

```
cor_m <- cor(timss[, c(7:17)], use = "complete")
corrplot(cor_m, method = 'color')
```



2. (2 points) Using the command impute in the library Hmisc create new variables in which the missing values are imputed randomly for these eleven variables. In order to ensure reproducability of the results use the command set.seed(26112017) prior to the imputation. Re-calculate the correlation matrix and compute how much it differs from the correlation matrix that was calculated for the original data including the missings. (Hint: Just calculate the difference of the two correlation matrices and round the result to two digits.)

```
set.seed(26112017)
var_list <- names(timss[, 7:17])
timss[paste0(var_list, '_imp')] <- lapply(timss[var_list],
                                  function(x) impute(x, 'random'))

cor_m_imp <- cor(timss[, 79:89], use = 'complete')
corrplot(cor_m_imp)
```




```
cor_m_diff <- cor_m - cor_m_imp
cor_m_diff
```

```
              bsbgclub     bsbgpaid      bsbgday1      bsbgday2
bsbgclub  0.0000000000  0.014213874 -0.0003050354 -0.0060847909
bsbgpaid  0.0142138740  0.000000000  0.0013920430  0.0030877685
bsbgday1 -0.0003050354  0.001392043  0.0000000000  0.0265523709
bsbgday2 -0.0060847909  0.003087769  0.0265523709  0.0000000000
bsbgday3 -0.0061844229  0.004958559  0.0136321852 -0.0004037687
bsbgday4 -0.0010304940  0.009323293 -0.0094646761 -0.0187211213
bsbgday5  0.0032075667  0.005190273  0.0036297463  0.0101634808
bsbgday6  0.0017255483 -0.004614904 -0.0012370708 -0.0102678869
bsbmday7  0.0088108825 -0.007193894 -0.0052282552  0.0001291941
bsbsday8  0.0013393126 -0.004968186  0.0091127519 -0.0036650458
bsbgday9 -0.0009565373 -0.009275288 -0.0009400591 -0.0053056764
              bsbgday3       bsbgday4       bsbgday5     bsbgday6
bsbgclub -0.0061844229 -0.00103049401  0.00320756670  0.001725548
bsbgpaid  0.0049585594  0.00932329309  0.00519027287 -0.004614904
bsbgday1  0.0136321852 -0.00946467609  0.00362974635 -0.001237071
bsbgday2 -0.0004037687 -0.01872112126  0.01016348079 -0.010267887
bsbgday3  0.0000000000  0.01239189141  0.00856511469 -0.011794218
bsbgday4  0.0123918914  0.00000000000 -0.00009007003  0.008509967
bsbgday5  0.0085651147 -0.00009007003  0.00000000000 -0.009020473
bsbgday6 -0.0117942181  0.00850996695 -0.00902047319  0.000000000
bsbmday7 -0.0021918310  0.02068077366 -0.00810717102  0.008222207
bsbsday8  0.0060489215  0.02154940538 -0.00277340878  0.009050093
bsbgday9  0.0029023690  0.02016246832 -0.00641708371  0.010566555
              bsbmday7     bsbsday8      bsbgday9
bsbgclub  0.0088108825  0.001339313 -0.0009565373
bsbgpaid -0.0071938939 -0.004968186 -0.0092752877
bsbgday1 -0.0052282552  0.009112752 -0.0009400591
bsbgday2  0.0001291941 -0.003665046 -0.0053056764
bsbgday3 -0.0021918310  0.006048921  0.0029023690
```

```
bsbgday4  0.0206807737  0.021549405  0.0201624683
bsbgday5 -0.0081071710 -0.002773409 -0.0064170837
bsbgday6  0.0082222072  0.009050093  0.0105665546
bsbmday7  0.0000000000  0.020419510  0.0231612463
bsbsday8  0.0204195102  0.000000000  0.0144576891
bsbgday9  0.0231612463  0.014457689  0.0000000000
```

3. (2 points) Next, you split the data into training and test data by randomly selecting 75% of your data for training, the remainder is for testing. Use set.seed(26112017) as seed for the random number generator to ensure replicability of your analysis. Report mean, median, and standard deviation for the international science score (bisciscr) for each of the two data sets (training and test).

```
set.seed(26112017)
index <- sample(1:nrow(timss), nrow(timss) * 0.75)
tim_train <- timss[index, ]
tim_test <- timss[-index, ]

mean(tim_train$bisciscr)
```

```
[1] 515.0306
```

```
median(tim_train$bisciscr)
```

```
[1] 515.19
```

```
sd(tim_train$bisciscr)
```

```
[1] 99.92839
```

```
mean(tim_test$bisciscr)
```

```
[1] 513.5114
```

```
median(tim_test$bisciscr)
```

```
[1] 515.31
```

```
sd(tim_test$bisciscr)
```

```
[1] 100.1583
```

4. Perform a principal component analysis on the training data.

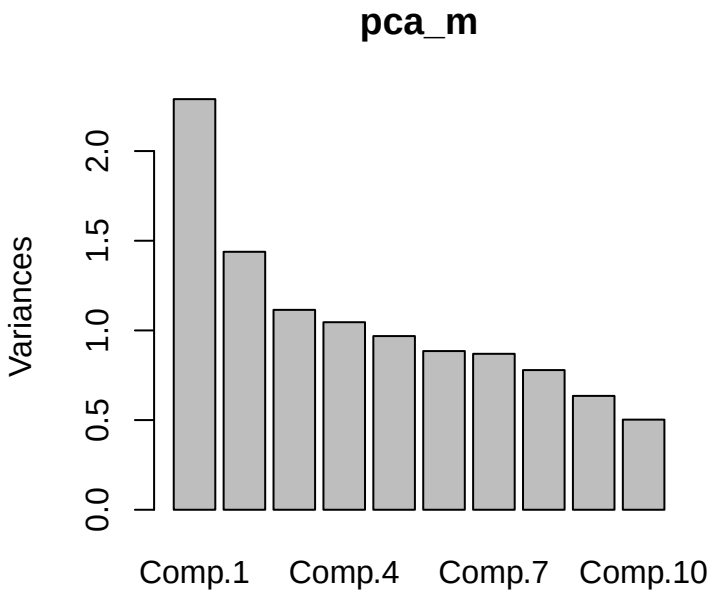
```
pca_m <- princomp(tim_train[, 79:89], cor = TRUE)
```

(a) (1 point) How many factors do you extract according to the Kaiser criterion, how many according to the scree plot?

```
# Kaiser criterion: eigenvalue larger than 1
(pca_m$sdev ^ 2) > 1
```

```
 Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6  Comp.7  Comp.8  Comp.9
   TRUE    TRUE    TRUE    TRUE   FALSE   FALSE   FALSE   FALSE   FALSE
Comp.10 Comp.11
  FALSE   FALSE
```

```
screeplot(pca_m)
```



(b) (1 point) Which percentage of variability of the original items is retained in the factor structure?

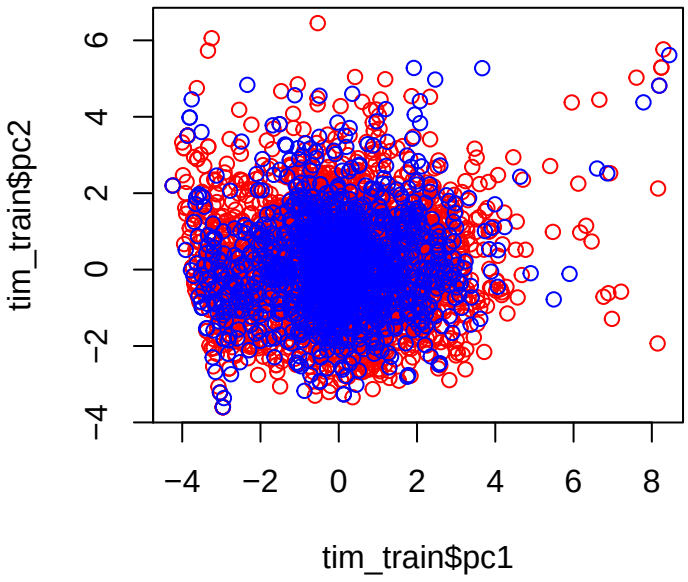
```
sum(pca_m$sdev[1:4]^2) / sum(pca_m$sdev^2)
```

```
[1] 0.5352912
```

5. (2 points) Use the PCA model to predict PCA scores for the test data (use the function predict). Extract the first four principal components for the training and the test data and store them for later use. Plot the first two principal components for training and test data in one graphic using different colours for the two data sets.

```
pca_m_pred <- predict(pca_m, tim_test)
tim_train <- cbind(tim_train, pca_m$scores[, 1:4])
colnames(tim_train)[90:93] <- paste0('pc', 1:4)

tim_test <- cbind(tim_test, pca_m_pred[, 1:4])
colnames(tim_test)[90:93] <- paste0('pc', 1:4)
```

```
plot(tim_train$pc1, tim_train$pc2, col = 'red')
points(tim_test$pc1, tim_test$pc2, col = 'blue')
```



6. (2 points) Perform a factor analysis using the Kaiser criterion and a varimax rotation. Save regression scores for later use. Label the rotated factors.

```
fact_m_1 <- factanal(tim_train[, 79:89],
                     factors = 4,
                     scores = "Bartlett",
                     rotation = "varimax")
fact_m_1
```

```
Call:
factanal(x = tim_train[, 79:89], factors = 4, scores = "Bartlett",     rotation = "varimax")

Uniquenesses:
bsbgclub_imp bsbgpaid_imp bsbgday1_imp bsbgday2_imp bsbgday3_imp
       0.896        0.909        0.005        0.837        0.005
bsbgday4_imp bsbgday5_imp bsbgday6_imp bsbmday7_imp bsbsday8_imp
       0.913        0.909        0.917        0.488        0.476
bsbgday9_imp
       0.517

Loadings:
             Factor1 Factor2 Factor3 Factor4
bsbgclub_imp                          0.318
bsbgpaid_imp                          0.300
bsbgday1_imp          0.993
bsbgday2_imp -0.102   0.330           0.207
bsbgday3_imp                  0.987
bsbgday4_imp  0.290
bsbgday5_imp  0.105           0.147   0.242
bsbgday6_imp  0.276
bsbmday7_imp  0.709
bsbsday8_imp  0.719
bsbgday9_imp  0.693
```

```
               Factor1 Factor2 Factor3 Factor4
SS loadings      1.689   1.113   1.005   0.320
Proportion Var   0.154   0.101   0.091   0.029
Cumulative Var   0.154   0.255   0.346   0.375

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 141.2 on 17 degrees of freedom.
The p-value is 1.27e-21
```

```
tim_train <- cbind(tim_train, fact_m_1$scores)
colnames(tim_train)[94:97] <- paste0('fv', 1:4)
```

7. (2 points) Perform a factor analysis using the Kaiser criterion and promax rotation. Is there a substantial difference between the two models? Which of the two models do you consider to be better?

```
fact_m_2 <- factanal(tim_train[, 79:89],
                     factors = 4,
                     scores = "Bartlett",
                     rotation = "promax")
fact_m_2
```

```

Call:
factanal(x = tim_train[, 79:89], factors = 4, scores = "Bartlett",     rotation = "promax")

Uniquenesses:
bsbgclub_imp bsbgpaid_imp bsbgday1_imp bsbgday2_imp bsbgday3_imp
       0.896        0.909        0.005        0.837        0.005
bsbgday4_imp bsbgday5_imp bsbgday6_imp bsbmday7_imp bsbsday8_imp
       0.913        0.909        0.917        0.488        0.476
bsbgday9_imp
       0.517

Loadings:
             Factor1 Factor2 Factor3 Factor4
bsbgclub_imp                          0.338
bsbgpaid_imp                          0.317
bsbgday1_imp           1.012
bsbgday2_imp -0.105    0.256          0.234
bsbgday3_imp                   0.997
bsbgday4_imp  0.294
bsbgday5_imp                   0.126  0.233
bsbgday6_imp  0.272
bsbmday7_imp  0.711
bsbsday8_imp  0.722
bsbgday9_imp  0.701

               Factor1 Factor2 Factor3 Factor4
SS loadings      1.702   1.108   1.018   0.333
Proportion Var   0.155   0.101   0.093   0.030
Cumulative Var   0.155   0.255   0.348   0.378
```

```
Factor Correlations:
        Factor1 Factor2 Factor3 Factor4
Factor1   1.000 -0.1369  0.1243 -0.2791
Factor2  -0.137  1.0000 -0.0559  0.2287
Factor3   0.124 -0.0559  1.0000  0.0889
Factor4  -0.279  0.2287  0.0889  1.0000

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 141.2 on 17 degrees of freedom.
The p-value is 1.27e-21
```

8. (2 points) Next, you run a linear regression model in order to predict the international science score (bisciscr) using bsbghome, bsbgedum, bsbgeduf, bsbgedus, bsbgsex, bsbgbrn1, bsbglang and the factors derived in question 6 as predictors. Provide a brief ver- bal summary of the model.

```r
lm_mod_1 <- lm(bisciscr ~ bsbghome + bsbgedum + bsbgeduf +
                bsbgedus + bsbgsex + bsbgbrn1 + bsbglang +
                fv1 + fv2 + fv3 + fv4, data = tim_train)
summary(lm_mod_1)
```

```
Call:
lm(formula = bisciscr ~ bsbghome + bsbgedum + bsbgeduf + bsbgedus +
    bsbgsex + bsbgbrn1 + bsbglang + fv1 + fv2 + fv3 + fv4, data = tim_train)

Residuals:
    Min      1Q  Median      3Q     Max
-306.50  -62.93   -2.46   61.92  334.48

Coefficients:
            Estimate Std. Error t value    Pr(>|t|)
(Intercept) 541.5409    15.5124  34.910     < 2e-16
bsbghome     -0.8069     1.4617  -0.552      0.5810
bsbgedum     -0.1212     2.7282  -0.044      0.9646
bsbgeduf      3.6195     2.3119   1.566      0.1176
bsbgedus     18.8876     1.9777   9.550     < 2e-16
bsbgsex       9.4377     4.8269   1.955      0.0507
bsbgbrn1    -45.3392     8.7137  -5.203 0.000000222
bsbglang    -32.8598     6.9520  -4.727 0.000002486
fv1          -2.1006     2.1354  -0.984      0.3254
fv2          -1.5657     2.3667  -0.662      0.5084
fv3         -10.2925     2.3830  -4.319 0.000016642
fv4          -1.3478     1.2166  -1.108      0.2681

Residual standard error: 90.95 on 1574 degrees of freedom
  (2736 observations deleted due to missingness)
Multiple R-squared:  0.1397,    Adjusted R-squared:  0.1337
F-statistic: 23.24 on 11 and 1574 DF,  p-value: < 2.2e-16
```

9. (2 points) Next, you run a linear regression model in order to predict the international science score (bisciscr) using bsbghome, bsbgedum, bsbgeduf, bsbgedus, bsbgsex, bsbgbrn1, bsbglang and the first four principal components derived in question 5 as predictors. Compare the results of the two models and provide a brief summary.

```
lm_mod_2 <- lm(bisciscr ~ bsbghome + bsbgedum + bsbgeduf +
                 bsbgedus + bsbgsex + bsbgbrn1 + bsbglang +
                 pc1 + pc2 + pc3 + pc4, data = tim_train)
summary(lm_mod_2)
```

```
Call:
lm(formula = bisciscr ~ bsbghome + bsbgedum + bsbgeduf + bsbgedus +
    bsbgsex + bsbgbrn1 + bsbglang + pc1 + pc2 + pc3 + pc4, data = tim_train)

Residuals:
    Min      1Q  Median      3Q     Max
-309.22  -60.77   -2.22   61.78  335.71

Coefficients:
             Estimate Std. Error t value    Pr(>|t|)
(Intercept) 537.309844  15.323316  35.065     < 2e-16
bsbghome     -0.785746   1.461180  -0.538    0.590828
bsbgedum     -0.002018   2.727054  -0.001    0.999410
bsbgeduf      3.553929   2.309548   1.539    0.124055
bsbgedus     18.875345   1.977281   9.546     < 2e-16
bsbgsex      12.632184   4.846566   2.606    0.009236
bsbgbrn1    -46.229167   8.721210  -5.301 0.000000132
bsbglang    -32.461460   6.948753  -4.672 0.000003244
pc1          -0.267594   1.633942  -0.164    0.869932
pc2          -5.718484   2.049973  -2.790    0.005342
pc3          -1.203830   2.238076  -0.538    0.590732
pc4           8.521600   2.300336   3.705    0.000219

Residual standard error: 90.92 on 1574 degrees of freedom
  (2736 observations deleted due to missingness)
Multiple R-squared:  0.1404,    Adjusted R-squared:  0.1344
F-statistic: 23.36 on 11 and 1574 DF,  p-value: < 2.2e-16
```

```
anova(lm_mod_1, lm_mod_2)
```

```
Analysis of Variance Table

Model 1: bisciscr ~ bsbghome + bsbgedum + bsbgeduf + bsbgedus + bsbgsex +
    bsbgbrn1 + bsbglang + fv1 + fv2 + fv3 + fv4
Model 2: bisciscr ~ bsbghome + bsbgedum + bsbgeduf + bsbgedus + bsbgsex +
    bsbgbrn1 + bsbglang + pc1 + pc2 + pc3 + pc4
  Res.Df      RSS Df Sum of Sq F Pr(>F)
1   1574 13020009
2   1574 13010116  0    9892.9
```

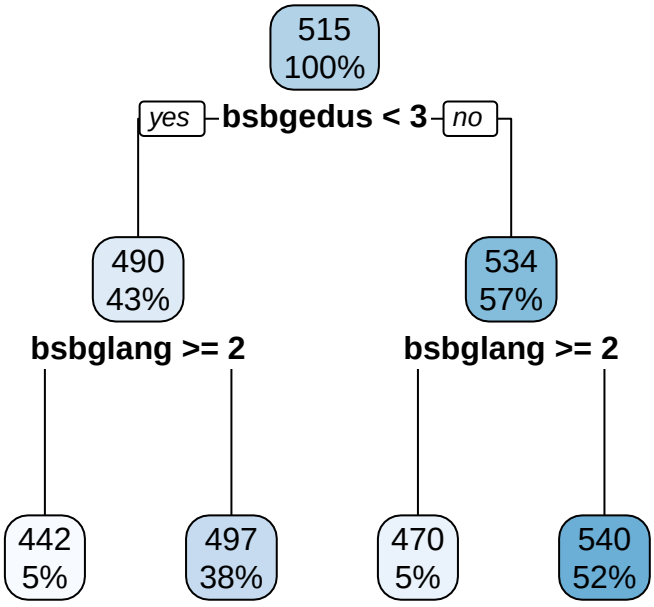
10. (2 points) Now, you run a tree model in order to predict the international science score (bisciscr) using bsbghome, bsbgedum, bsbgeduf, bsbgedus, bsbgsex, bsbgbrn1, bsbglang and the first four principal components derived in question 5 as predictors. Compare the results of the tree model to the two regression models and provide a brief summary.

```
library(rpart)
tree <- rpart(bisciscr ~ bsbghome + bsbgedum + bsbgeduf +
                  bsbgedus + bsbgsex + bsbgbrn1 + bsbglang +
                  pc1 + pc2 + pc3 + pc4, data = tim_train)
summary(tree)
```

```
Call:
rpart(formula = bisciscr ~ bsbghome + bsbgedum + bsbgeduf + bsbgedus +
    bsbgsex + bsbgbrn1 + bsbglang + pc1 + pc2 + pc3 + pc4, data = tim_train)
  n= 4322

          CP nsplit rel error    xerror       xstd
1 0.04631710      0 1.0000000 1.0004139 0.02216862
2 0.02353121      1 0.9536829 0.9546339 0.02147883
3 0.01429070      2 0.9301517 0.9315196 0.02121542
4 0.01000000      3 0.9158610 0.9177322 0.02093907

Variable importance
bsbgedus bsbglang      pc1      pc2 bsbghome      pc4 bsbgbrn1      pc3
      54       32        6        2        2        2        2        1

Node number 1: 4322 observations,    complexity param=0.0463171
  mean=515.0306, MSE=9983.372
  left son=2 (1872 obs) right son=3 (2450 obs)
  Primary splits:
      bsbgedus < 2.5        to the left,  improve=0.066497070, (1246 missing)
      bsbglang < 1.5        to the right, improve=0.041754160, (803 missing)
      bsbgbrn1 < 1.5        to the right, improve=0.027186430, (35 missing)
      pc2      < 1.256483   to the right, improve=0.011116340, (0 missing)
      bsbgeduf < 3.5        to the left,  improve=0.008922876, (1587 missing)
  Surrogate splits:
      pc1      < -0.5351077 to the left,  agree=0.582, adj=0.103, (1246 split)
      pc2      < 0.2008802  to the right, agree=0.553, adj=0.040, (0 split)
      bsbghome < 5.5        to the right, agree=0.552, adj=0.039, (0 split)
      pc4      < -1.142866  to the left,  agree=0.545, adj=0.024, (0 split)
      pc3      < -1.196195  to the left,  agree=0.540, adj=0.013, (0 split)

Node number 2: 1872 observations,    complexity param=0.0142907
  mean=490.4303, MSE=8525.575
  left son=4 (227 obs) right son=5 (1645 obs)
  Primary splits:
      bsbglang < 1.5        to the right, improve=0.038622680, (339 missing)
      bsbgbrn1 < 1.5        to the right, improve=0.026662140, (15 missing)
      bsbgsex  < 1.5        to the left,  improve=0.017250570, (28 missing)
      pc2      < 1.276434   to the right, improve=0.007870083, (0 missing)
      pc4      < -0.8557661 to the left,  improve=0.007131988, (0 missing)
  Surrogate splits:
```

```
      bsbgbrn1 < 1.5        to the right, agree=0.877, adj=0.129, (325 split)
      pc4      < 3.792724   to the right, agree=0.861, adj=0.018, (14 split)
      pc3      < 4.293767   to the right, agree=0.860, adj=0.009, (0 split)
      pc2      < -3.449686  to the left,  agree=0.859, adj=0.005, (0 split)

Node number 3: 2450 observations,    complexity param=0.02353121
  mean=533.8272, MSE=10281.53
  left son=6 (224 obs) right son=7 (2226 obs)
  Primary splits:
      bsbglang < 1.5        to the right, improve=0.041842150, (464 missing)
      bsbgbrn1 < 1.5        to the right, improve=0.025530380, (20 missing)
      pc1      < 2.058473   to the right, improve=0.013298520, (0 missing)
      pc2      < 1.734756   to the right, improve=0.011736270, (0 missing)
      bsbgsex  < 1.5        to the left,  improve=0.009071197, (26 missing)
  Surrogate splits:
      pc3 < 4.863373   to the right, agree=0.890, adj=0.018, (464 split)
      pc4 < 4.596317   to the right, agree=0.888, adj=0.004, (0 split)

Node number 4: 227 observations
  mean=441.5735, MSE=9244.005

Node number 5: 1645 observations
  mean=497.1722, MSE=8051.593

Node number 6: 224 observations
  mean=469.6532, MSE=11014.82

Node number 7: 2226 observations
  mean=540.2849, MSE=9751.623
```

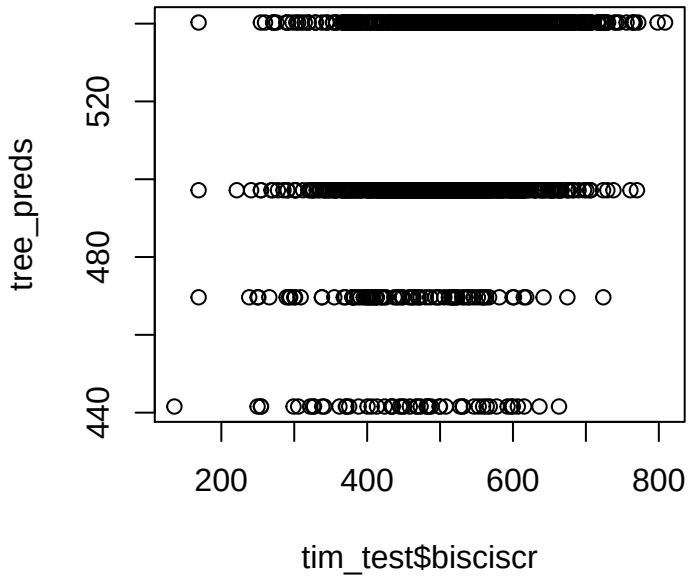
```
rpart.plot(tree)
```



11. (2 points) Predict the international science score using the tree model and using the model derived in Question 9 for the test data. Plot the predicted scores against the observed international science score in the test data and compute the correlation coefficients. Are you satisfied with the predictions of the two models?

```
tree_preds <- predict(tree, tim_test)
lm_preds <- predict(lm_mod_2, tim_test)
```

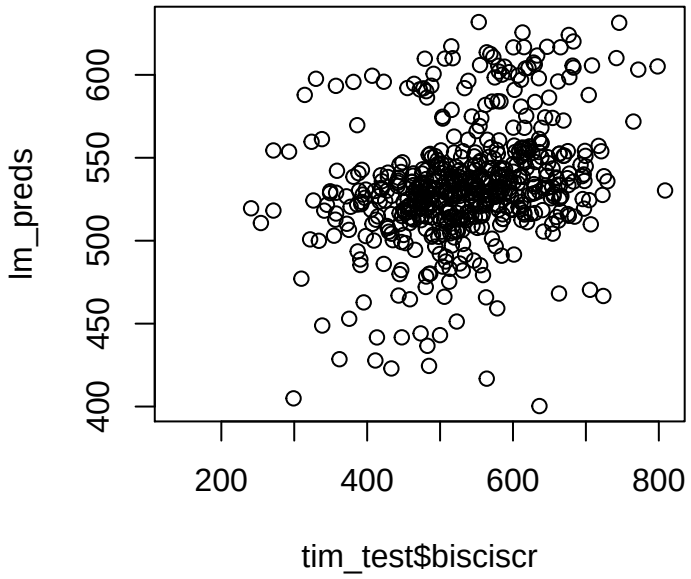
```
plot(tim_test$bisciscr, tree_preds)
```




```
cor(tim_test$bisciscr, tree_preds, use = 'complete')
```

```
[1] 0.2876012
```

```
plot(tim_test$bisciscr, lm_preds)
```



```r
cor(tim_test$bisciscr, lm_preds, use = 'complete')
```

```
[1] 0.3034878
```