student_performance_code

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

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

```
#reading csv
stud_perf <- read.csv("C:/Users/cmanz/OneDrive/Documents/Ryerson
stuff/cind820/student dataset/student-perf.csv", header = T, stringsAsFactors
= F,                          na.strings = c("", "NA"), sep = ";")
head(stud_perf)

##   school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob
reason
## 1     GP   F  18       U     GT3       A    4    4  at_home  teacher
course
## 2     GP   F  17       U     GT3       T    1    1  at_home    other
course
## 3     GP   F  15       U     LE3       T    1    1  at_home    other
other
## 4     GP   F  15       U     GT3       T    4    2   health services
home
## 5     GP   F  16       U     GT3       T    3    3    other    other
home
## 6     GP   M  16       U     LE3       T    4    3 services    other
reputation
##   guardian traveltime studytime failures schoolsup famsup paid activities
## 1   mother          2         2        0       yes     no   no         no
## 2   father          1         2        0        no    yes   no         no
## 3   mother          1         2        3       yes     no  yes         no
## 4   mother          1         3        0        no    yes  yes        yes
## 5   father          1         2        0        no    yes  yes         no
## 6   mother          1         2        0        no    yes  yes        yes
##   nursery higher internet romantic famrel freetime goout Dalc Walc health
## 1     yes    yes       no       no      4        3     4    1    1      3
## 2      no    yes      yes       no      5        3     3    1    1      3
## 3     yes    yes      yes       no      4        3     2    2    3      3
## 4     yes    yes      yes      yes      3        2     2    1    1      5
## 5     yes    yes       no       no      4        3     2    1    2      5
```

```
## 6     yes    yes      yes       no      5        4     2    1    2      5
##   absences G1 G2 G3
## 1        6  5  6  6
## 2        4  5  5  6
## 3       10  7  8 10
## 4        2 15 14 15
## 5        4  6 10 10
## 6       10 15 15 15
```

```
#checking the datatypes of the attributes
str(stud_perf)
```

```
## 'data.frame':    1044 obs. of  33 variables:
##  $ school    : chr  "GP" "GP" "GP" "GP" ...
##  $ sex       : chr  "F" "F" "F" "F" ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : chr  "U" "U" "U" "U" ...
##  $ famsize   : chr  "GT3" "GT3" "LE3" "GT3" ...
##  $ Pstatus   : chr  "A" "T" "T" "T" ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ Mjob      : chr  "at_home" "at_home" "at_home" "health" ...
##  $ Fjob      : chr  "teacher" "other" "other" "services" ...
##  $ reason    : chr  "course" "course" "other" "home" ...
##  $ guardian  : chr  "mother" "father" "mother" "mother" ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : chr  "yes" "no" "yes" "no" ...
##  $ famsup    : chr  "no" "yes" "no" "yes" ...
##  $ paid      : chr  "no" "no" "yes" "yes" ...
##  $ activities: chr  "no" "no" "no" "yes" ...
##  $ nursery   : chr  "yes" "no" "yes" "yes" ...
##  $ higher    : chr  "yes" "yes" "yes" "yes" ...
##  $ internet  : chr  "no" "yes" "yes" "yes" ...
##  $ romantic  : chr  "no" "no" "no" "yes" ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G1        : int  5 5 7 15 6 15 12 6 16 14 ...
##  $ G2        : int  6 5 8 14 10 15 12 5 18 15 ...
##  $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...
```

```
#checking for missing values
sum(is.na(stud_perf))
```

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

```r
#looking for correlation between numeric attributes except final grade(G3)
cor(stud_perf[, c('age', 'Medu', 'Fedu', 'traveltime', 'studytime',
'failures', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'health',
'absences',                      'G1', 'G2')])
```

```
##                    age         Medu          Fedu   traveltime
studytime
## age         1.000000000 -0.130196115 -0.1385207614  0.049215707 -
0.007870098
## Medu       -0.130196115  1.000000000  0.6420631457 -0.238180728
0.090616377
## Fedu       -0.138520761  0.642063146  1.0000000000 -0.196328161
0.033457874
## traveltime  0.049215707 -0.238180728 -0.1963281605  1.000000000 -
0.081328016
## studytime  -0.007870098  0.090616377  0.0334578745 -0.081328016
1.000000000
## failures    0.282363566 -0.187769404 -0.1913904210  0.087177495 -
0.152023523
## famrel      0.007161921  0.015003618  0.0130659150 -0.012577522
0.012324093
## freetime    0.002645147  0.001054219  0.0021417298 -0.007402578 -
0.094429345
## goout       0.118510124  0.025614278  0.0300748764  0.049739783 -
0.072940739
## Dalc        0.133452990  0.001515097 -0.0001648393  0.109423016 -
0.159664641
## Walc        0.098291406 -0.029330541  0.0195239342  0.084292404 -
0.229073148
## health     -0.029129265 -0.013254090  0.0342882377 -0.029001978 -
0.063044459
## absences    0.153195647  0.059707676  0.0408288855 -0.022668699 -
0.075593669
## G1         -0.124121249  0.226100602  0.1958980209 -0.121053301
0.211313915
## G2         -0.119474744  0.224661748  0.1826339619 -0.140162973
0.183166702
##               failures       famrel     freetime       goout          Dalc
## age         0.28236357  0.007161921  0.002645147  0.11851012  0.1334529897
## Medu       -0.18776940  0.015003618  0.001054219  0.02561428  0.0015150967
## Fedu       -0.19139042  0.013065915  0.002141730  0.03007488 -0.0001648393
## traveltime  0.08717749 -0.012577522 -0.007402578  0.04973978  0.1094230162
## studytime  -0.15202352  0.012324093 -0.094429345 -0.07294074 -0.1596646413
## failures    1.00000000 -0.053676457  0.102678757  0.07468331  0.1163357901
## famrel     -0.05367646  1.000000000  0.136900650  0.08061921 -0.0764826572
## freetime    0.10267876  0.136900650  1.000000000  0.32355575  0.1449791279
## goout       0.07468331  0.080619212  0.323555753  1.00000000  0.2531348291
## Dalc        0.11633579 -0.076482657  0.144979128  0.25313483  1.0000000000
## Walc        0.10743159 -0.100663375  0.130377028  0.39979373  0.6278138380
## health      0.04831102  0.104100776  0.081517225 -0.01373623  0.0655153422
```

```
## absences    0.09999785 -0.062170662 -0.032078736  0.05614214  0.1328671345
## G1         -0.37417487  0.036947274 -0.051984712 -0.10116347 -0.1509425374
## G2         -0.37717218  0.042053621 -0.068951886 -0.10841089 -0.1315764840
##                   Walc      health    absences          G1          G2
## age         0.09829141 -0.02912927  0.15319565 -0.12412125 -0.11947474
## Medu       -0.02933054 -0.01325409  0.05970768  0.22610060  0.22466175
## Fedu        0.01952393  0.03428824  0.04082889  0.19589802  0.18263396
## traveltime  0.08429240 -0.02900198 -0.02266870 -0.12105330 -0.14016297
## studytime  -0.22907315 -0.06304446 -0.07559367  0.21131391  0.18316670
## failures    0.10743159  0.04831102  0.09999785 -0.37417487 -0.37717218
## famrel     -0.10066338  0.10410078 -0.06217066  0.03694727  0.04205362
## freetime    0.13037703  0.08151722 -0.03207874 -0.05198471 -0.06895189
## goout       0.39979373 -0.01373623  0.05614214 -0.10116347 -0.10841089
## Dalc        0.62781384  0.06551534  0.13286713 -0.15094254 -0.13157648
## Walc        1.00000000  0.10666944  0.13970313 -0.14240140 -0.12811435
## health      0.10666944  1.00000000 -0.02747860 -0.06047794 -0.08800109
## absences    0.13970313 -0.02747860  1.00000000 -0.09242463 -0.08933169
## G1         -0.14240140 -0.06047794 -0.09242463  1.00000000  0.85873875
## G2         -0.12811435 -0.08800109 -0.08933169  0.85873875  1.00000000
```

```
#graphing frequency distribution of final grade(G3)
library(epiDisplay)
```

```
## Warning: package 'epiDisplay' was built under R version 4.1.3
```

```
## Loading required package: foreign
```

```
## Loading required package: survival
```

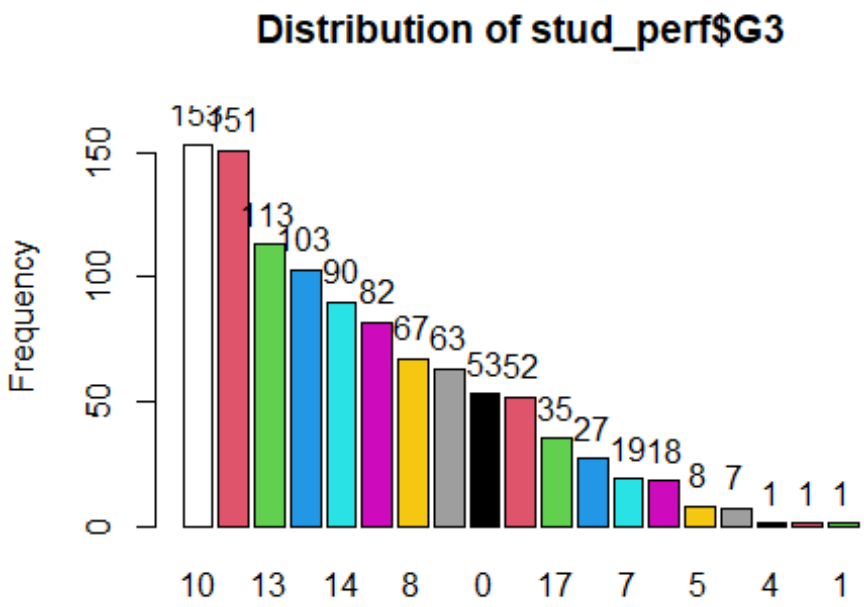
```
## Loading required package: MASS
```

```
## Warning: package 'MASS' was built under R version 4.1.2
```

```
## Loading required package: nnet
```

```
## Warning: package 'nnet' was built under R version 4.1.3
```

```
tab1(stud_perf$G3, sort.group = "decreasing", cum.percent = T)
```




```
## stud_perf$G3 :
##          Frequency Percent Cum. percent
## 10             153    14.7         14.7
## 11             151    14.5         29.1
## 13             113    10.8         39.9
## 12             103     9.9         49.8
## 14              90     8.6         58.4
## 15              82     7.9         66.3
## 8               67     6.4         72.7
## 9               63     6.0         78.7
## 0               53     5.1         83.8
## 16              52     5.0         88.8
## 17              35     3.4         92.1
## 18              27     2.6         94.7
## 7               19     1.8         96.6
## 6               18     1.7         98.3
## 5                8     0.8         99.0
## 19               7     0.7         99.7
## 4                1     0.1         99.8
## 20               1     0.1         99.9
## 1                1     0.1        100.0
##   Total       1044   100.0        100.0

#assigning numeric values to Mjob(mother's job) and Fjob(father's job).
# 1 - at_home
# 2 - services
# 3 - other
```

```
# 4 - teacher
# 5 - health
stud_perf$Mjob[stud_perf$Mjob == 'at_home'] = 1
stud_perf$Mjob[stud_perf$Mjob == 'services'] = 2
stud_perf$Mjob[stud_perf$Mjob == 'other'] = 3
stud_perf$Mjob[stud_perf$Mjob == 'teacher'] = 4
stud_perf$Mjob[stud_perf$Mjob == 'health'] = 5

stud_perf$Fjob[stud_perf$Fjob == 'at_home'] = 1
stud_perf$Fjob[stud_perf$Fjob == 'services'] = 2
stud_perf$Fjob[stud_perf$Fjob == 'other'] = 3
stud_perf$Fjob[stud_perf$Fjob == 'teacher'] = 4
stud_perf$Fjob[stud_perf$Fjob == 'health'] = 5

head(stud_perf)
```

```
##   school sex age address famsize Pstatus Medu Fedu Mjob Fjob     reason
## 1     GP   F  18       U     GT3       A    4    4    1    4     course
## 2     GP   F  17       U     GT3       T    1    1    1    3     course
## 3     GP   F  15       U     LE3       T    1    1    1    3      other
## 4     GP   F  15       U     GT3       T    4    2    5    2       home
## 5     GP   F  16       U     GT3       T    3    3    3    3       home
## 6     GP   M  16       U     LE3       T    4    3    2    3 reputation
##   guardian traveltime studytime failures schoolsup famsup paid activities
## 1   mother          2         2        0       yes     no   no         no
## 2   father          1         2        0        no    yes   no         no
## 3   mother          1         2        3       yes     no  yes         no
## 4   mother          1         3        0        no    yes  yes        yes
## 5   father          1         2        0        no    yes  yes         no
## 6   mother          1         2        0        no    yes  yes        yes
##   nursery higher internet romantic famrel freetime goout Dalc Walc health
## 1     yes    yes       no       no      4        3     4    1    1      3
## 2      no    yes      yes       no      5        3     3    1    1      3
## 3     yes    yes      yes       no      4        3     2    2    3      3
## 4     yes    yes      yes      yes      3        2     2    1    1      5
## 5     yes    yes       no       no      4        3     2    1    2      5
## 6     yes    yes      yes       no      5        4     2    1    2      5
##   absences G1 G2 G3
## 1        6  5  6  6
## 2        4  5  5  6
## 3       10  7  8 10
## 4        2 15 14 15
## 5        4  6 10 10
## 6       10 15 15 15
```

```
#assigning binary values to yes or no attributes (schoolsup, famsup, paid,
activities, nursery, higher, internet, romantic)
stud_perf$schoolsup[stud_perf$schoolsup == 'yes'] = 1
stud_perf$schoolsup[stud_perf$schoolsup == 'no'] = 0
```

```
stud_perf$famsup[stud_perf$famsup == 'yes'] = 1
stud_perf$famsup[stud_perf$famsup == 'no'] = 0

stud_perf$paid[stud_perf$paid == 'yes'] = 1
stud_perf$paid[stud_perf$paid == 'no'] = 0

stud_perf$activities[stud_perf$activities == 'yes'] = 1
stud_perf$activities[stud_perf$activities == 'no'] = 0

stud_perf$nursery[stud_perf$nursery == 'yes'] = 1
stud_perf$nursery[stud_perf$nursery == 'no'] = 0

stud_perf$higher[stud_perf$higher == 'yes'] = 1
stud_perf$higher[stud_perf$higher == 'no'] = 0

stud_perf$internet[stud_perf$internet == 'yes'] = 1
stud_perf$internet[stud_perf$internet == 'no'] = 0

stud_perf$romantic[stud_perf$romantic == 'yes'] = 1
stud_perf$romantic[stud_perf$romantic == 'no'] = 0

head(stud_perf)
```

```
##   school sex age address famsize Pstatus Medu Fedu Mjob Fjob     reason
## 1     GP   F  18       U     GT3       A    4    4    1    4     course
## 2     GP   F  17       U     GT3       T    1    1    1    3     course
## 3     GP   F  15       U     LE3       T    1    1    1    3      other
## 4     GP   F  15       U     GT3       T    4    2    5    2       home
## 5     GP   F  16       U     GT3       T    3    3    3    3       home
## 6     GP   M  16       U     LE3       T    4    3    2    3 reputation
##   guardian traveltime studytime failures schoolsup famsup paid activities
## 1   mother          2         2        0         1      0    0          0
## 2   father          1         2        0         0      1    0          0
## 3   mother          1         2        3         1      0    1          0
## 4   mother          1         3        0         0      1    1          1
## 5   father          1         2        0         0      1    1          0
## 6   mother          1         2        0         0      1    1          1
##   nursery higher internet romantic famrel freetime goout Dalc Walc health
## 1       1      1        0        0      4        3     4    1    1      3
## 2       0      1        1        0      5        3     3    1    1      3
## 3       1      1        1        0      4        3     2    2    3      3
## 4       1      1        1        1      3        2     2    1    1      5
## 5       1      1        0        0      4        3     2    1    2      5
## 6       1      1        1        0      5        4     2    1    2      5
##   absences G1 G2 G3
## 1        6  5  6  6
## 2        4  5  5  6
## 3       10  7  8 10
## 4        2 15 14 15
```

```
## 5        4  6 10 10
## 6       10 15 15 15
```

```
#assigning pass or fail to the grades columns
#for G1
stud_perf$G1[stud_perf$G1 == 0] = 'Fail'
stud_perf$G1[stud_perf$G1 == 1] = 'Fail'
stud_perf$G1[stud_perf$G1 == 2] = 'Fail'
stud_perf$G1[stud_perf$G1 == 3] = 'Fail'
stud_perf$G1[stud_perf$G1 == 4] = 'Fail'
stud_perf$G1[stud_perf$G1 == 5] = 'Fail'
stud_perf$G1[stud_perf$G1 == 6] = 'Fail'
stud_perf$G1[stud_perf$G1 == 7] = 'Fail'
stud_perf$G1[stud_perf$G1 == 8] = 'Fail'
stud_perf$G1[stud_perf$G1 == 9] = 'Fail'
stud_perf$G1[stud_perf$G1 == 10] = 'Pass'
stud_perf$G1[stud_perf$G1 == 11] = 'Pass'
stud_perf$G1[stud_perf$G1 == 12] = 'Pass'
stud_perf$G1[stud_perf$G1 == 13] = 'Pass'
stud_perf$G1[stud_perf$G1 == 14] = 'Pass'
stud_perf$G1[stud_perf$G1 == 15] = 'Pass'
stud_perf$G1[stud_perf$G1 == 16] = 'Pass'
stud_perf$G1[stud_perf$G1 == 17] = 'Pass'
stud_perf$G1[stud_perf$G1 == 18] = 'Pass'
stud_perf$G1[stud_perf$G1 == 19] = 'Pass'
stud_perf$G1[stud_perf$G1 == 20] = 'Pass'

#for G2
stud_perf$G2[stud_perf$G2 == 0] = 'Fail'
stud_perf$G2[stud_perf$G2 == 1] = 'Fail'
stud_perf$G2[stud_perf$G2 == 2] = 'Fail'
stud_perf$G2[stud_perf$G2 == 3] = 'Fail'
stud_perf$G2[stud_perf$G2 == 4] = 'Fail'
stud_perf$G2[stud_perf$G2 == 5] = 'Fail'
stud_perf$G2[stud_perf$G2 == 6] = 'Fail'
stud_perf$G2[stud_perf$G2 == 7] = 'Fail'
stud_perf$G2[stud_perf$G2 == 8] = 'Fail'
stud_perf$G2[stud_perf$G2 == 9] = 'Fail'
stud_perf$G2[stud_perf$G2 == 10] = 'Pass'
stud_perf$G2[stud_perf$G2 == 11] = 'Pass'
stud_perf$G2[stud_perf$G2 == 12] = 'Pass'
stud_perf$G2[stud_perf$G2 == 13] = 'Pass'
stud_perf$G2[stud_perf$G2 == 14] = 'Pass'
stud_perf$G2[stud_perf$G2 == 15] = 'Pass'
stud_perf$G2[stud_perf$G2 == 16] = 'Pass'
stud_perf$G2[stud_perf$G2 == 17] = 'Pass'
stud_perf$G2[stud_perf$G2 == 18] = 'Pass'
stud_perf$G2[stud_perf$G2 == 19] = 'Pass'
stud_perf$G2[stud_perf$G2 == 20] = 'Pass'
```

```r
#for G3
stud_perf$G3[stud_perf$G3 == 0] = 'Fail'
stud_perf$G3[stud_perf$G3 == 1] = 'Fail'
stud_perf$G3[stud_perf$G3 == 2] = 'Fail'
stud_perf$G3[stud_perf$G3 == 3] = 'Fail'
stud_perf$G3[stud_perf$G3 == 4] = 'Fail'
stud_perf$G3[stud_perf$G3 == 5] = 'Fail'
stud_perf$G3[stud_perf$G3 == 6] = 'Fail'
stud_perf$G3[stud_perf$G3 == 7] = 'Fail'
stud_perf$G3[stud_perf$G3 == 8] = 'Fail'
stud_perf$G3[stud_perf$G3 == 9] = 'Fail'
stud_perf$G3[stud_perf$G3 == 10] = 'Pass'
stud_perf$G3[stud_perf$G3 == 11] = 'Pass'
stud_perf$G3[stud_perf$G3 == 12] = 'Pass'
stud_perf$G3[stud_perf$G3 == 13] = 'Pass'
stud_perf$G3[stud_perf$G3 == 14] = 'Pass'
stud_perf$G3[stud_perf$G3 == 15] = 'Pass'
stud_perf$G3[stud_perf$G3 == 16] = 'Pass'
stud_perf$G3[stud_perf$G3 == 17] = 'Pass'
stud_perf$G3[stud_perf$G3 == 18] = 'Pass'
stud_perf$G3[stud_perf$G3 == 19] = 'Pass'
stud_perf$G3[stud_perf$G3 == 20] = 'Pass'

head(stud_perf)
```

```
##   school sex age address famsize Pstatus Medu Fedu Mjob Fjob     reason
## 1     GP   F  18       U     GT3       A    4    4    1    4     course
## 2     GP   F  17       U     GT3       T    1    1    1    3     course
## 3     GP   F  15       U     LE3       T    1    1    1    3      other
## 4     GP   F  15       U     GT3       T    4    2    5    2       home
## 5     GP   F  16       U     GT3       T    3    3    3    3       home
## 6     GP   M  16       U     LE3       T    4    3    2    3 reputation
##   guardian traveltime studytime failures schoolsup famsup paid activities
## 1   mother          2         2        0         1      0    0          0
## 2   father          1         2        0         0      1    0          0
## 3   mother          1         2        3         1      0    1          0
## 4   mother          1         3        0         0      1    1          1
## 5   father          1         2        0         0      1    1          0
## 6   mother          1         2        0         0      1    1          1
##   nursery higher internet romantic famrel freetime goout Dalc Walc health
## 1       1      1        0        0      4        3     4    1    1      3
## 2       0      1        1        0      5        3     3    1    1      3
## 3       1      1        1        0      4        3     2    2    3      3
## 4       1      1        1        1      3        2     2    1    1      5
## 5       1      1        0        0      4        3     2    1    2      5
## 6       1      1        1        0      5        4     2    1    2      5
##   absences   G1   G2   G3
## 1        6 Fail Fail Fail
## 2        4 Fail Fail Fail
## 3       10 Fail Fail Pass
```

```
## 4        2 Pass Pass Pass
## 5        4 Fail Pass Pass
## 6       10 Pass Pass Pass
```

```
#changing specific columns to numeric
stud_perf$Mjob <- as.numeric(as.character(stud_perf$Mjob))
stud_perf$Fjob <- as.numeric(as.character(stud_perf$Fjob))
stud_perf$schoolsup <- as.numeric(as.character(stud_perf$schoolsup))
stud_perf$famsup <- as.numeric(as.character(stud_perf$famsup))
stud_perf$paid <- as.numeric(as.character(stud_perf$paid))
stud_perf$activities <- as.numeric(as.character(stud_perf$activities))
stud_perf$nursery <- as.numeric(as.character(stud_perf$nursery))
stud_perf$higher <- as.numeric(as.character(stud_perf$higher))
stud_perf$internet <- as.numeric(as.character(stud_perf$internet))
stud_perf$romantic <- as.numeric(as.character(stud_perf$romantic))
```

```
str(stud_perf)
```

```
## 'data.frame':    1044 obs. of  33 variables:
##  $ school    : chr  "GP" "GP" "GP" "GP" ...
##  $ sex       : chr  "F" "F" "F" "F" ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : chr  "U" "U" "U" "U" ...
##  $ famsize   : chr  "GT3" "GT3" "LE3" "GT3" ...
##  $ Pstatus   : chr  "A" "T" "T" "T" ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ Mjob      : num  1 1 1 5 3 2 3 3 2 3 ...
##  $ Fjob      : num  4 3 3 2 3 3 3 4 3 3 ...
##  $ reason    : chr  "course" "course" "other" "home" ...
##  $ guardian  : chr  "mother" "father" "mother" "mother" ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : num  1 0 1 0 0 0 0 1 0 0 ...
##  $ famsup    : num  0 1 0 1 1 1 0 1 1 1 ...
##  $ paid      : num  0 0 1 1 1 1 0 0 1 1 ...
##  $ activities: num  0 0 0 1 0 1 0 0 0 1 ...
##  $ nursery   : num  1 0 1 1 1 1 1 1 1 1 ...
##  $ higher    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ internet  : num  0 1 1 1 0 1 1 0 1 1 ...
##  $ romantic  : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G1        : chr  "Fail" "Fail" "Fail" "Pass" ...
```

```
##  $ G2        : chr  "Fail" "Fail" "Fail" "Pass" ...
##  $ G3        : chr  "Fail" "Fail" "Pass" "Pass" ...

#normalizing the numeric attributes
minmaxNorm <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}

studperf_norm1 <- as.data.frame(lapply(stud_perf[7:10], minmaxNorm))
head(studperf_norm1)

##   Medu Fedu Mjob Fjob
## 1 1.00 1.00 0.00 0.75
## 2 0.25 0.25 0.00 0.50
## 3 0.25 0.25 0.00 0.50
## 4 1.00 0.50 1.00 0.25
## 5 0.75 0.75 0.50 0.50
## 6 1.00 0.75 0.25 0.50

studperf_norm2 <- as.data.frame(lapply(stud_perf[13:30], minmaxNorm))
head(studperf_norm2)

##   traveltime studytime failures schoolsup famsup paid activities nursery
higher
## 1  0.3333333 0.3333333        0         1      0    0          0       1
1
## 2  0.0000000 0.3333333        0         0      1    0          0       0
1
## 3  0.0000000 0.3333333        1         1      0    1          0       1
1
## 4  0.0000000 0.6666667        0         0      1    1          1       1
1
## 5  0.0000000 0.3333333        0         0      1    1          0       1
1
## 6  0.0000000 0.3333333        0         0      1    1          1       1
1
##   internet romantic famrel freetime goout Dalc Walc health   absences
## 1        0        0   0.75     0.50  0.75 0.00 0.00    0.5 0.08000000
## 2        1        0   1.00     0.50  0.50 0.00 0.00    0.5 0.05333333
## 3        1        0   0.75     0.50  0.25 0.25 0.50    0.5 0.13333333
## 4        1        1   0.50     0.25  0.25 0.00 0.00    1.0 0.02666667
## 5        0        0   0.75     0.50  0.25 0.00 0.25    1.0 0.05333333
## 6        1        0   1.00     0.75  0.25 0.00 0.25    1.0 0.13333333

#merging the normalized data frames side by side
studperf_norm <- cbind(studperf_norm1, studperf_norm2)
head(studperf_norm)

##   Medu Fedu Mjob Fjob traveltime studytime failures schoolsup famsup paid
## 1 1.00 1.00 0.00 0.75  0.3333333 0.3333333        0         1      0    0
## 2 0.25 0.25 0.00 0.50  0.0000000 0.3333333        0         0      1    0
```

```
## 3 0.25 0.25 0.00 0.50  0.0000000 0.3333333        1         1      0    1
## 4 1.00 0.50 1.00 0.25  0.0000000 0.6666667        0         0      1    1
## 5 0.75 0.75 0.50 0.50  0.0000000 0.3333333        0         0      1    1
## 6 1.00 0.75 0.25 0.50  0.0000000 0.3333333        0         0      1    1
##   activities nursery higher internet romantic famrel freetime goout Dalc
Walc
## 1          0       1      1        0        0   0.75     0.50  0.75 0.00
0.00
## 2          0       0      1        1        0   1.00     0.50  0.50 0.00
0.00
## 3          0       1      1        1        0   0.75     0.50  0.25 0.25
0.50
## 4          1       1      1        1        1   0.50     0.25  0.25 0.00
0.00
## 5          0       1      1        0        0   0.75     0.50  0.25 0.00
0.25
## 6          1       1      1        1        0   1.00     0.75  0.25 0.00
0.25
##   health   absences
## 1    0.5 0.08000000
## 2    0.5 0.05333333
## 3    0.5 0.13333333
## 4    1.0 0.02666667
## 5    1.0 0.05333333
## 6    1.0 0.13333333
```

```
#creating the train and test sets
train_ind <- sample(1:nrow(stud_perf), 0.7 * nrow(stud_perf))
train.perf <- stud_perf[train_ind, ]
test.perf <- stud_perf[-train_ind, ]

head(train.perf)
```

```
##      school sex age address famsize Pstatus Medu Fedu Mjob Fjob     reason
## 1021     MS   F  18       U     GT3       T    2    3    1    2     course
## 768      GP   F  17       U     GT3       T    2    2    3    3     course
## 525      GP   M  16       U     GT3       T    2    3    3    3     course
## 411      GP   F  16       U     GT3       T    4    4    5    3       home
## 764      GP   M  18       U     LE3       T    4    4    3    3 reputation
## 266      GP   M  18       R     LE3       A    3    4    3    3 reputation
##      guardian traveltime studytime failures schoolsup famsup paid
activities
## 1021   father          2         1        0         0      1    0
0
## 768    mother          1         2        0         0      1    0
0
## 525    mother          2         3        0         0      1    0
0
## 411    mother          1         1        0         0      1    0
0
```

```
## 764    father          1         1        0         0      1    0
0
## 266    mother          2         2        0         0      1    1
1
##      nursery higher internet romantic famrel freetime goout Dalc Walc
health
## 1021       1      1        1        1      5        2     3    1    2
4
## 768        1      1        0        1      4        2     2    1    1
3
## 525        0      1        1        1      3        2     3    2    2
1
## 411        1      1        1        0      4        4     4    1    2
2
## 764        1      1        1        0      4        2     5    3    4
5
## 266        1      1        1        0      4        2     5    3    4
1
##      absences   G1   G2   G3
## 1021        0 Pass Pass Pass
## 768         4 Pass Pass Pass
## 525         4 Pass Pass Pass
## 411         6 Pass Pass Pass
## 764         2 Fail Fail Pass
## 266        13 Pass Pass Pass
```

```
head(test.perf)
```

```
##    school sex age address famsize Pstatus Medu Fedu Mjob Fjob     reason
## 17     GP   F  16       U     GT3       T    4    4    2    2 reputation
## 18     GP   F  16       U     GT3       T    3    3    3    3 reputation
## 21     GP   M  15       U     GT3       T    4    3    4    3 reputation
## 24     GP   M  16       U     LE3       T    2    2    3    3 reputation
## 33     GP   M  15       R     GT3       T    4    3    4    1     course
## 42     GP   M  15       U     LE3       T    4    4    4    3       home
##    guardian traveltime studytime failures schoolsup famsup paid activities
## 17   mother          1         3        0         0      1    1          1
## 18   mother          3         2        0         1      1    0          1
## 21   mother          1         2        0         0      0    0          0
## 24   mother          2         2        0         0      1    0          1
## 33   mother          1         2        0         0      1    0          1
## 42    other          1         1        0         0      1    0          0
##    nursery higher internet romantic famrel freetime goout Dalc Walc health
## 17       1      1        1        0      3        2     3    1    2      2
## 18       1      1        0        0      5        3     2    1    1      4
## 21       1      1        1        0      4        4     1    1    1      1
## 24       1      1        1        0      5        4     4    2    4      5
## 33       1      1        1        1      4        5     2    1    1      5
## 42       0      1        1        1      5        4     3    2    4      5
##    absences   G1   G2   G3
```

```
## 17        6 Pass Pass Pass
## 18        4 Fail Pass Pass
## 21        0 Pass Pass Pass
## 24        0 Pass Pass Pass
## 33        0 Pass Pass Pass
## 42        8 Pass Pass Pass
```

```
#creating the regression model
glm_model <-
glm(as.factor(G3)~Medu+Fedu+Mjob+Fjob+traveltime+studytime+failures+schoolsup
+famsup+paid+activities+nursery+higher+internet+romantic+famrel+freetime+goou
t+Dalc+Walc+health+absences, family = "binomial", data = train.perf)

summary(glm_model)
```

```
##
## Call:
## glm(formula = as.factor(G3) ~ Medu + Fedu + Mjob + Fjob + traveltime +
##     studytime + failures + schoolsup + famsup + paid + activities +
##     nursery + higher + internet + romantic + famrel + freetime +
##     goout + Dalc + Walc + health + absences, family = "binomial",
##     data = train.perf)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4244   0.3562   0.5043   0.6392   2.0756
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.85976    0.82178   1.046   0.2955
## Medu         0.13946    0.12809   1.089   0.2763
## Fedu         0.04582    0.12171   0.376   0.7066
## Mjob        -0.06038    0.10049  -0.601   0.5479
## Fjob         0.08971    0.12756   0.703   0.4819
## traveltime   0.05078    0.14016   0.362   0.7171
## studytime    0.19020    0.13350   1.425   0.1542
## failures    -0.83660    0.14755  -5.670 1.43e-08 ***
## schoolsup   -0.61323    0.28983  -2.116   0.0344 *
## famsup      -0.10839    0.21153  -0.512   0.6084
## paid        -0.53062    0.24108  -2.201   0.0277 *
## activities   0.20392    0.20176   1.011   0.3122
## nursery     -0.31263    0.25880  -1.208   0.2271
## higher       0.95932    0.31731   3.023   0.0025 **
## internet     0.26547    0.24279   1.093   0.2742
## romantic    -0.35416    0.20831  -1.700   0.0891 .
## famrel       0.13180    0.10421   1.265   0.2060
## freetime    -0.11642    0.10449  -1.114   0.2652
## goout       -0.15425    0.09834  -1.569   0.1168
## Dalc        -0.04834    0.13656  -0.354   0.7234
## Walc         0.02126    0.10821   0.196   0.8442
```

```
## health      -0.07063    0.07245  -0.975   0.3296
## absences    -0.03608    0.01668  -2.164   0.0305 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 770.30  on 729  degrees of freedom
## Residual deviance: 661.44  on 707  degrees of freedom
## AIC: 707.44
##
## Number of Fisher Scoring iterations: 4
```

```
#confusion matrix for the regression model
predicted <- predict(glm_model, test.perf, type = "response")

predicted_class <- ifelse(predicted >= 0.5, 1, 0)
ConfusionMatrix <- table(actual = test.perf$G3, predicted = predicted_class)
ConfusionMatrix
```

```
##       predicted
## actual   0   1
##   Fail  23  46
##   Pass  10 235
```

```
#finding accuracy, precision, recall, sensitivity and specificity using the
confusion matrix
#accuracy
acc <- sum(diag(ConfusionMatrix))/nrow(test.perf)

#precision
prec <- ConfusionMatrix[2,2]/sum(ConfusionMatrix[2,2]+ConfusionMatrix[2,1])

#recall
recall <- ConfusionMatrix[2,2]/sum(ConfusionMatrix[2,2]+ConfusionMatrix[1,2])

#sensitivity
sens <- ConfusionMatrix[1,1]/sum(ConfusionMatrix[1,1]+ConfusionMatrix[2,1])

#specificity
spec <- ConfusionMatrix[2,2]/sum(ConfusionMatrix[1,2]+ConfusionMatrix[2,2])

acc
```

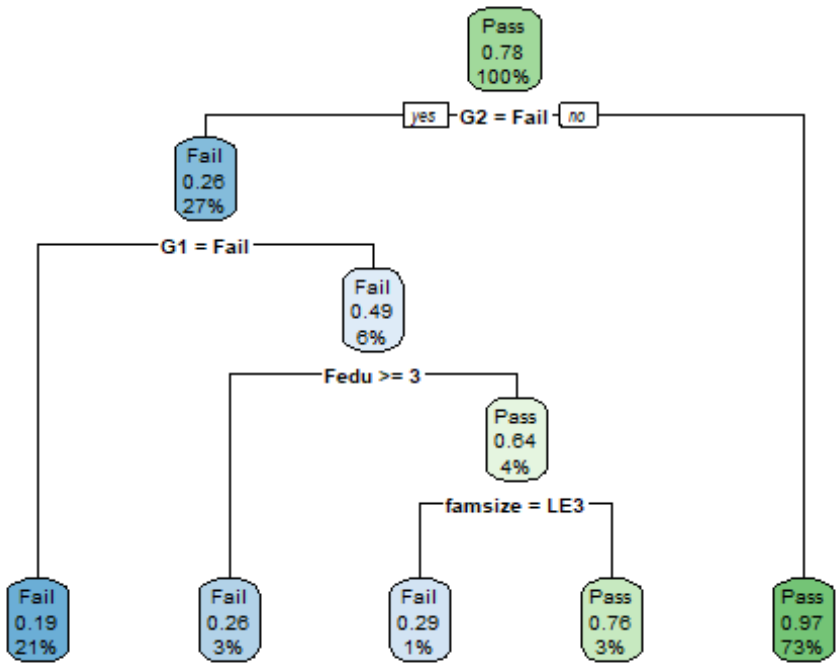
```
## [1] 0.8216561
```

```
prec
```

```
## [1] 0.9591837
```

```
recall
```

```
## [1] 0.8362989

sens

## [1] 0.6969697

spec

## [1] 0.8362989

#finding f score
fscore <- (2*prec*recall)/(prec+recall)
fscore

## [1] 0.8935361

#creating a decision tree to predict G3
#install.packages("rpart.plot")
library(rpart)
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.1.3

tree <- rpart(G3~., data = train.perf, method = 'class')
rpart.plot(tree)
```




```
#matrix for the decision tree
matrix_tree <-predict(tree, test.perf, type = 'class')
```

```
table_mat <- table(test.perf$G3, matrix_tree)
table_mat
```

```
##       matrix_tree
##        Fail Pass
##   Fail   61    8
##   Pass   22  223
```

```
#accuracy of the decision tree matrix
acc_tree <- sum(diag(table_mat))/sum(table_mat)

#precision
prec_tree <- table_mat[2,2]/sum(table_mat[2,2]+table_mat[2,1])

#recall
recall_tree <- table_mat[2,2]/sum(table_mat[2,2]+table_mat[1,2])

#sensitivity
sens_tree <- table_mat[1,1]/sum(table_mat[1,1]+table_mat[2,1])

#specificity
spec_tree <- table_mat[2,2]/sum(table_mat[1,2]+table_mat[2,2])

acc_tree
```

```
## [1] 0.9044586
```

```
prec_tree
```

```
## [1] 0.9102041
```

```
recall_tree
```

```
## [1] 0.965368
```

```
sens_tree
```

```
## [1] 0.7349398
```

```
spec_tree
```

```
## [1] 0.965368
```

```
#changing all columns to factors for the randdom forest model
cols <- c("school", "sex", "age", "address", "famsize", "Pstatus", "Medu",
"Fedu", "Mjob", "Fjob", "reason", "guardian", "traveltime", "studytime",
"failures", "schoolsup", "famsup", "paid", "activities", "nursery", "higher",
"internet", "romantic", "famrel", "freetime", "goout", "Dalc", "Walc",
"health", "absences", "G1", "G2", "G3")
train.perf[cols] <- lapply(train.perf[cols], factor)

cols <- c("school", "sex", "age", "address", "famsize", "Pstatus", "Medu",
```

```
"Fedu", "Mjob", "Fjob", "reason", "guardian", "traveltime", "studytime",
"failures", "schoolsup", "famsup", "paid", "activities", "nursery", "higher",
"internet", "romantic", "famrel", "freetime", "goout", "Dalc", "Walc",
"health", "absences", "G1", "G2", "G3")
test.perf[cols] <- lapply(test.perf[cols], factor)

str(train.perf)

## 'data.frame':    730 obs. of  33 variables:
##  $ school    : Factor w/ 2 levels "GP","MS": 2 1 1 1 1 1 1 1 1 2 ...
##  $ sex       : Factor w/ 2 levels "F","M": 1 1 2 1 2 2 2 1 2 2 ...
##  $ age       : Factor w/ 8 levels "15","16","17",..: 4 3 2 2 4 4 3 5 5 3
...
##  $ address   : Factor w/ 2 levels "R","U": 2 2 2 2 2 1 1 2 2 2 ...
##  $ famsize   : Factor w/ 2 levels "GT3","LE3": 1 1 1 1 2 2 1 2 1 1 ...
##  $ Pstatus   : Factor w/ 2 levels "A","T": 2 2 2 2 2 1 2 1 2 2 ...
##  $ Medu      : Factor w/ 5 levels "0","1","2","3",..: 3 3 3 5 5 4 2 3 3 2
...
##  $ Fedu      : Factor w/ 5 levels "0","1","2","3",..: 4 3 4 5 5 5 3 4 2 2
...
##  $ Mjob      : Factor w/ 5 levels "1","2","3","4",..: 1 3 3 5 3 3 1 1 3 3
...
##  $ Fjob      : Factor w/ 5 levels "1","2","3","4",..: 2 3 3 3 3 3 3 3 3 3
...
##  $ reason    : Factor w/ 4 levels "course","home",..: 1 1 1 2 4 4 2 2 4 2
...
##  $ guardian  : Factor w/ 3 levels "father","mother",..: 1 2 2 2 1 2 2 3 2
2 ...
##  $ traveltime: Factor w/ 4 levels "1","2","3","4": 2 1 2 1 1 2 1 2 1 1 ...
##  $ studytime : Factor w/ 4 levels "1","2","3","4": 1 2 3 1 1 2 2 1 1 2 ...
##  $ failures  : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 2 1 1 ...
##  $ schoolsup : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ famsup    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 1 1 1 1 ...
##  $ paid      : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 2 ...
##  $ activities: Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ nursery   : Factor w/ 2 levels "0","1": 2 2 1 2 2 2 2 2 2 1 ...
##  $ higher    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 2 2 ...
##  $ internet  : Factor w/ 2 levels "0","1": 2 1 2 2 2 2 1 2 2 2 ...
##  $ romantic  : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 1 1 1 ...
##  $ famrel    : Factor w/ 5 levels "1","2","3","4",..: 5 4 3 4 4 4 3 2 5 4
...
##  $ freetime  : Factor w/ 5 levels "1","2","3","4",..: 2 2 2 4 2 2 1 2 3 4
...
##  $ goout     : Factor w/ 5 levels "1","2","3","4",..: 3 2 3 4 5 5 3 3 4 3
...
##  $ Dalc      : Factor w/ 5 levels "1","2","3","4",..: 1 1 2 1 3 3 1 3 1 2
...
##  $ Walc      : Factor w/ 5 levels "1","2","3","4",..: 2 1 2 2 4 4 5 4 4 4
...
##  $ health    : Factor w/ 5 levels "1","2","3","4",..: 4 3 1 2 5 1 3 5 4 5
```

```
...
##  $ absences  : Factor w/ 29 levels "0","1","2","3",..: 1 5 5 7 3 14 7 17
11 5 ...
##  $ G1        : Factor w/ 2 levels "Fail","Pass": 2 2 2 2 1 2 1 2 1 1 ...
##  $ G2        : Factor w/ 2 levels "Fail","Pass": 2 2 2 2 1 2 1 2 2 1 ...
##  $ G3        : Factor w/ 2 levels "Fail","Pass": 2 2 2 2 2 2 2 2 2 1 ...

str(test.perf)

## 'data.frame':    314 obs. of  33 variables:
##  $ school    : Factor w/ 2 levels "GP","MS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex       : Factor w/ 2 levels "F","M": 1 1 2 2 2 2 1 2 2 1 ...
##  $ age       : Factor w/ 8 levels "15","16","17",..: 2 2 1 2 1 1 2 2 1 2
...
##  $ address   : Factor w/ 2 levels "R","U": 2 2 2 2 1 2 2 2 2 2 ...
##  $ famsize   : Factor w/ 2 levels "GT3","LE3": 1 1 1 2 1 2 2 1 1 2 ...
##  $ Pstatus   : Factor w/ 2 levels "A","T": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Medu      : Factor w/ 5 levels "0","1","2","3",..: 5 4 5 3 5 5 3 5 5 3
...
##  $ Fedu      : Factor w/ 5 levels "0","1","2","3",..: 5 4 4 3 4 5 3 4 3 3
...
##  $ Mjob      : Factor w/ 5 levels "1","2","3","4",..: 2 3 4 3 4 4 3 5 4 2
...
##  $ Fjob      : Factor w/ 5 levels "1","2","3","4",..: 2 3 3 3 1 3 1 2 3 2
...
##  $ reason    : Factor w/ 4 levels "course","home",..: 4 4 4 4 1 2 1 4 2 1
...
##  $ guardian  : Factor w/ 3 levels "father","mother",..: 2 2 2 2 2 3 1 2 2
2 ...
##  $ traveltime: Factor w/ 4 levels "1","2","3","4": 1 3 1 2 1 1 2 1 1 3 ...
##  $ studytime : Factor w/ 4 levels "1","2","3","4": 3 2 2 2 2 1 2 4 2 2 ...
##  $ failures  : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 2 1 1 1 ...
##  $ schoolsup : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 2 1 1 1 ...
##  $ famsup    : Factor w/ 2 levels "0","1": 2 2 1 2 2 2 1 1 2 2 ...
##  $ paid      : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 2 2 ...
##  $ activities: Factor w/ 2 levels "0","1": 2 2 1 2 2 1 2 2 1 1 ...
##  $ nursery   : Factor w/ 2 levels "0","1": 2 2 2 2 2 1 2 2 2 2 ...
##  $ higher    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ internet  : Factor w/ 2 levels "0","1": 2 1 2 2 2 2 2 2 1 2 ...
##  $ romantic  : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 1 1 1 1 ...
##  $ famrel    : Factor w/ 5 levels "1","2","3","4",..: 3 5 4 5 4 5 4 4 4 4
...
##  $ freetime  : Factor w/ 5 levels "1","2","3","4",..: 2 3 4 4 5 4 3 2 3 3
...
##  $ goout     : Factor w/ 5 levels "1","2","3","4",..: 3 2 1 4 2 3 3 2 3 3
...
##  $ Dalc      : Factor w/ 5 levels "1","2","3","4",..: 1 1 1 2 1 2 2 1 2 2
...
##  $ Walc      : Factor w/ 5 levels "1","2","3","4",..: 2 1 1 4 1 4 2 1 2 3
...
```

```
##  $ health    : Factor w/ 5 levels "1","2","3","4",..: 2 4 1 5 5 5 5 2 5 4
...
##  $ absences  : Factor w/ 26 levels "0","1","2","3",..: 7 5 1 1 1 9 14 5 3
3 ...
##  $ G1        : Factor w/ 2 levels "Fail","Pass": 2 1 2 2 2 2 2 2 2 2 ...
##  $ G2        : Factor w/ 2 levels "Fail","Pass": 2 2 2 2 2 2 2 2 2 2 ...
##  $ G3        : Factor w/ 2 levels "Fail","Pass": 2 2 2 2 2 2 1 2 2 2 ...
```

```
#random forest option 1
library(datasets)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

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

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:epiDisplay':
##
##     alpha
```

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

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:epiDisplay':
##
##     dotplot
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(nnet)
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.3
```

```
## randomForest 4.7-1.1
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin

rf <- randomForest(G3~., data = train.perf, proximity = TRUE)
p1 <- predict(rf, train.perf)
confusionMatrix(p1, train.perf$G3)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction Fail Pass
##       Fail  161    0
##       Pass    0  569
##
##                Accuracy : 1
##                  95% CI : (0.995, 1)
##     No Information Rate : 0.7795
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.2205
##          Detection Rate : 0.2205
##    Detection Prevalence : 0.2205
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : Fail
##

#random forest option 2 step 1

#install.packages("caret")
#install.packages("e1071")
#install.packages("randomForest")
library(caret)
library(e1071)

## Warning: package 'e1071' was built under R version 4.1.3

library(randomForest)

#training the random forest model
trControl <- trainControl(method = "cv", number = 10, search = "grid")
```

```r
rf_default <- train(G3~., data = train.perf, method = "rf", metric =
"Accuracy", trControl = trControl)

print(rf_default)
```

```
## Random Forest
##
## 730 samples
##  32 predictor
##   2 classes: 'Fail', 'Pass'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 657, 657, 657, 657, 658, 657, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##     2   0.7835719  0.02828784
##    53   0.9124563  0.75266560
##   104   0.9015159  0.71964647
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 53.
```

```r
#step 2
#constructs a vector from 1:10 (mtry is the vector)
set.seed(1234)
tuneGrid <- expand.grid(.mtry = c(1: 10))
rf_mtry <- train(G3~.,
    data = train.perf,
    method = "rf",
    metric = "Accuracy",
    tuneGrid = tuneGrid,
    trControl = trControl,
    importance = TRUE,
    nodesize = 14,
    ntree = 300)
print(rf_mtry)
```

```
## Random Forest
##
## 730 samples
##  32 predictor
##   2 classes: 'Fail', 'Pass'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 658, 657, 656, 657, 657, 657, ...
## Resampling results across tuning parameters:
##
```

```
##   mtry  Accuracy   Kappa
##    1    0.7794623  0.0000000
##    2    0.7794623  0.0000000
##    3    0.8068611  0.1885804
##    4    0.8396833  0.3981113
##    5    0.8671007  0.5489908
##    6    0.8876111  0.6446153
##    7    0.9054568  0.7114393
##    8    0.9109178  0.7304093
##    9    0.9054003  0.7177557
##   10    0.9095099  0.7322898
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 8.
```

```
#best value of mtry is stored in:
rf_mtry$bestTune$mtry
```

```
## [1] 8
```

```
#stored so it can be used later
max(rf_mtry$results$Accuracy)
```

```
## [1] 0.9109178
```

```
best_mtry <- rf_mtry$bestTune$mtry
best_mtry
```

```
## [1] 8
```

```
#step 3
#creates a variable with the best value of the mtry parameter
#creates a loop then stores the current value of the max node
store_maxnode <- list()
tuneGrid <- expand.grid(.mtry = best_mtry)
for (maxnodes in c(5: 15)) {
    set.seed(1234)
    rf_maxnode <- train(G3~.,
        data = train.perf,
        method = "rf",
        metric = "Accuracy",
        tuneGrid = tuneGrid,
        trControl = trControl,
        importance = TRUE,
        nodesize = 14,
        maxnodes = maxnodes,
        ntree = 300)
    current_iteration <- toString(maxnodes)
    store_maxnode[[current_iteration]] <- rf_maxnode
}
```

```
results_mtry <- resamples(store_maxnode)
summary(results_mtry)

##
## Call:
## summary.resamples(object = results_mtry)
##
## Models: 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
## Number of resamples: 10
##
## Accuracy
##         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## 5  0.7671233 0.7785388 0.7808219 0.7808322 0.7808219 0.7945205    0
## 6  0.7671233 0.7808219 0.7808219 0.7849233 0.7830433 0.8219178    0
## 7  0.7671233 0.7849408 0.8082192 0.8082685 0.8304795 0.8493151    0
## 8  0.7671233 0.7842466 0.8138318 0.8136364 0.8321918 0.8648649    0
## 9  0.7671233 0.8007515 0.8287671 0.8273536 0.8493151 0.9041096    0
## 10 0.7945205 0.8253425 0.8424658 0.8492734 0.8720034 0.9054054    0
## 11 0.8082192 0.8356164 0.8493151 0.8493480 0.8609774 0.9041096    0
## 12 0.7945205 0.8390411 0.8552131 0.8533825 0.8630137 0.9041096    0
## 13 0.8082192 0.8493151 0.8690068 0.8588995 0.8767123 0.8904110    0
## 14 0.8219178 0.8615868 0.8904110 0.8834825 0.9006849 0.9452055    0
## 15 0.8356164 0.8493151 0.8621575 0.8739310 0.8982321 0.9315068    0
##
## Kappa
##           Min.   1st Qu.    Median       Mean    3rd Qu.      Max. NA's
## 5  -0.02646816 0.0000000 0.0000000 0.02196449 0.00000000 0.1518203    0
## 6   0.00000000 0.0000000 0.0328000 0.06087710 0.09267646 0.2649109    0
## 7  -0.02646816 0.0918009 0.2236555 0.22067068 0.34212261 0.4481100    0
## 8   0.06560000 0.1177049 0.2333136 0.25682506 0.36256388 0.5188557    0
## 9   0.09613984 0.1799030 0.3446469 0.33119302 0.43986980 0.6675342    0
## 10  0.26491092 0.3810015 0.4464884 0.45652245 0.53862255 0.6875754    0
## 11  0.31684492 0.3889703 0.4316296 0.46492091 0.51901142 0.6675342    0
## 12  0.28757319 0.3989501 0.4659220 0.48408718 0.56342348 0.6843731    0
## 13  0.30881282 0.5060191 0.5496767 0.51525181 0.59013215 0.6096257    0
## 14  0.41514931 0.4900770 0.6473322 0.61143039 0.70500942 0.8240964    0
## 15  0.44810997 0.4736053 0.5485607 0.57652203 0.67943961 0.7745522    0

#step 4
#with the value of the max node and mtry, number of trees can be tuned
store_maxtrees <- list()
for (ntree in c(250, 300, 350, 400, 450, 500, 550, 600, 800, 1000, 2000)) {
    set.seed(5678)
    rf_maxtrees <- train(G3~.,
        data = train.perf,
        method = "rf",
        metric = "Accuracy",
        tuneGrid = tuneGrid,
        trControl = trControl,
        importance = TRUE,
```

```
        nodesize = 14,
        maxnodes = 24,
        ntree = ntree)
    key <- toString(ntree)
    store_maxtrees[[key]] <- rf_maxtrees
}
results_tree <- resamples(store_maxtrees)
summary(results_tree)

##
## Call:
## summary.resamples(object = results_tree)
##
## Models: 250, 300, 350, 400, 450, 500, 550, 600, 800, 1000, 2000
## Number of resamples: 10
##
## Accuracy
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## 250  0.8493151 0.8664384 0.8904110 0.8931656 0.9000563 0.9589041    0
## 300  0.8630137 0.8767123 0.8843947 0.8945540 0.8996861 0.9589041    0
## 350  0.8630137 0.8767123 0.8843947 0.8945540 0.8996861 0.9589041    0
## 400  0.8630137 0.8767123 0.8843947 0.8959238 0.8996861 0.9726027    0
## 450  0.8630137 0.8771288 0.8904110 0.8986826 0.9101027 0.9726027    0
## 500  0.8630137 0.8767123 0.8843947 0.8959238 0.8996861 0.9726027    0
## 550  0.8630137 0.8767123 0.8904110 0.8972752 0.9000563 0.9726027    0
## 600  0.8630137 0.8767123 0.8775454 0.8945540 0.8996861 0.9726027    0
## 800  0.8630137 0.8767123 0.8904110 0.8972752 0.9000563 0.9726027    0
## 1000 0.8630137 0.8767123 0.8911514 0.8972752 0.9037766 0.9589041    0
## 2000 0.8630137 0.8801370 0.8904110 0.8986641 0.9104730 0.9589041    0
##
## Kappa
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## 250  0.5496354 0.5809414 0.6406329 0.6674577 0.6973232 0.8771733    0
## 300  0.5725439 0.5899063 0.6307156 0.6721908 0.6973232 0.8771733    0
## 350  0.5809414 0.5998457 0.6307156 0.6743558 0.6973232 0.8771733    0
## 400  0.5725439 0.5899063 0.6398563 0.6782973 0.6973232 0.9199561    0
## 450  0.5809414 0.6200785 0.6398563 0.6889439 0.7341475 0.9199561    0
## 500  0.5809414 0.6200785 0.6396205 0.6794462 0.6859716 0.9199561    0
## 550  0.5809414 0.6303134 0.6479570 0.6829016 0.6867621 0.9199561    0
## 600  0.5809414 0.5998457 0.6396205 0.6758745 0.6859716 0.9199561    0
## 800  0.5809414 0.6355702 0.6481928 0.6847297 0.6867621 0.9199561    0
## 1000 0.5809414 0.6355702 0.6497737 0.6861865 0.6993325 0.8771733    0
## 2000 0.5809414 0.6223657 0.6655895 0.6881989 0.7341475 0.8771733    0

#step 4.1
#with the final model the random forest can be trained
#800 trees will be trained
#24 is the max number of leaves
fit_rf <- train(G3~.,
    train.perf,
```

```
    method = "rf",
    metric = "Accuracy",
    tuneGrid = tuneGrid,
    trControl = trControl,
    importance = TRUE,
    nodesize = 14,
    ntree = 800,
    maxnodes = 24)

print(fit_rf)
```

```
## Random Forest
##
## 730 samples
##  32 predictor
##   2 classes: 'Fail', 'Pass'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 657, 657, 657, 657, 657, 658, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8944223  0.6713754
##
## Tuning parameter 'mtry' was held constant at a value of 8
```

```
#step 5
#confusion matrix and accuracy score for the final model
prediction <- predict(fit_rf, train.perf)
confusionMatrix(prediction, train.perf$G3)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Fail Pass
##       Fail  126   17
##       Pass   35  552
##
##                Accuracy : 0.9288
##                  95% CI : (0.9076, 0.9463)
##     No Information Rate : 0.7795
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7842
##
##  Mcnemar's Test P-Value : 0.0184
##
##             Sensitivity : 0.7826
##             Specificity : 0.9701
```

```
##          Pos Pred Value : 0.8811
##          Neg Pred Value : 0.9404
##              Prevalence : 0.2205
##          Detection Rate : 0.1726
##    Detection Prevalence : 0.1959
##       Balanced Accuracy : 0.8764
##
##        'Positive' Class : Fail
##
```

```
#step 6
#shows the variables with the greatest importance
library(randomForest)
varImp(fit_rf)
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 104)
##
##            Importance
## G2Pass         100.00
## G1Pass          74.47
## failures3       37.98
## failures1       29.01
## failures2       28.64
## higher1         27.39
## absences26      19.65
## Dalc3           17.35
## Dalc4           16.83
## schoolMS        15.59
## absences4       13.52
## age19           13.50
## Medu1           13.29
## Dalc2           12.59
## Fedu1           12.45
## romantic1       12.39
## goout5          11.93
## absences6       11.83
## famrel4         11.71
## studytime3      11.61
```