Shipwreck student-mat

davidabraham

2/14/2017

1. Investigation into G3

Goal:-Predict students ability to pass and their grades based on certain variables and find which variable(s) is the best predictor.Dataset used is student-mat.csv

Student dataset has G3 variable which is used for classifying Pass/Fail and for actegorising student's grades into Fail,Sufficient,Satisfactory,Good and Excellent.These classifications will be predicted based on some independent variables.

Predictors are:- 1. ParentStatus(living together or not) 2. Mother??? s education(factors:- none, upto 4th grade, upto 9th grade, secondary education and higher education) 3. Travel time to school 4. Romantic status of the student 5. G1 - score from test1 6. G2 - score from test2

Different methods used for prediction :- 1. Linear regression 2. Decision Tree 3. Naive Bayes Method

1. Linear regression is used on variables G1 and G2 individually to predict G3
2. Decision tree is used on variables G1 and G2 together to predict pass-ability and Grades
3. Naive Bayes method is used on categorical variables - ParentStatus, MotherEducation, TravelTime and Romantic Status to predict pass-ability and grades.

1. Load students data

```
students<-read.table('~/Desktop/Shipwreck/student-mat.csv',header = TRUE,sep = ",")
```

2. Extract necessary columns for analysis

```
students<- students[,c(6,7,13,23,31,32,33)]
```

3. Calculate pass or fail variable and store it in variable Pass

```
Pass <- ifelse(students$G3>9,'PASS','FAIL')
students <- data.frame(students,Pass)
```

4. Calculate Grade variable

```
Grade <- ifelse(students$G3<=9,'FAIL','PASS')
Grade <- ifelse(students$G3>=10 & students$G3<=11,'Sufficient',Grade)
Grade <- ifelse(students$G3>=12 & students$G3<=13,'Satisfactory',Grade)
Grade <- ifelse(students$G3>=14 & students$G3<=15,'Good',Grade)
Grade <- ifelse(students$G3>=16 & students$G3<=20,'Excellent',Grade)
students <- data.frame(students,Grade)
```

5. Exploration of data

Dimensions

```
dim(students)
```

```
## [1] 395   9
```

Number of rows in data

```
nrow(students)
```

```
## [1] 395
```

Number of columns

```
ncol(students)
```

```
## [1] 9
```

Structure

```
str(students)
```

```
## 'data.frame':    395 obs. of  9 variables:
##  $ Pstatus   : Factor w/ 2 levels "A","T": 1 2 2 2 2 2 2 1 1 2 ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ romantic  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ 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 ...
##  $ Pass      : Factor w/ 2 levels "FAIL","PASS": 1 1 2 2 2 2 2 1 2 2 ...
##  $ Grade     : Factor w/ 5 levels "Excellent","FAIL",..: 2 2 5 3 5 3 5 2 1 3 ...
```

Variable or column names

```
names(students)
```

```
## [1] "Pstatus"    "Medu"       "traveltime" "romantic"   "G1"
## [6] "G2"         "G3"         "Pass"       "Grade"
```

Attributes

```
attributes(students)
```

```
## $names
## [1] "Pstatus"    "Medu"       "traveltime" "romantic"   "G1"
## [6] "G2"         "G3"         "Pass"       "Grade"
```

```
##
## $row.names
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
##  [18]  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
##  [35]  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
##  [52]  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68
##  [69]  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
##  [86]  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102
## [103] 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
## [120] 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136
## [137] 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
## [154] 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170
## [171] 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
## [188] 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204
## [205] 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221
## [222] 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238
## [239] 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255
## [256] 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272
## [273] 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289
## [290] 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306
## [307] 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323
## [324] 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340
## [341] 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357
## [358] 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374
## [375] 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391
## [392] 392 393 394 395
##
## $class
## [1] "data.frame"
```

Top 10 rows

```
head(students,n=10)
```

```
##    Pstatus Medu traveltime romantic G1 G2 G3 Pass      Grade
## 1        A    4          2       no  5  6  6 FAIL       FAIL
## 2        T    1          1       no  5  5  6 FAIL       FAIL
## 3        T    1          1       no  7  8 10 PASS Sufficient
## 4        T    4          1      yes 15 14 15 PASS       Good
## 5        T    3          1       no  6 10 10 PASS Sufficient
## 6        T    4          1       no 15 15 15 PASS       Good
## 7        T    2          1       no 12 12 11 PASS Sufficient
## 8        A    4          2       no  6  5  6 FAIL       FAIL
## 9        A    3          1       no 16 18 19 PASS  Excellent
## 10       T    3          1       no 14 15 15 PASS       Good
```

Variable distribution before factorisation

```
summary(students)
```

```
##  Pstatus      Medu          traveltime     romantic        G1
##  A: 41   Min.   :0.000   Min.   :1.000   no :263   Min.   : 3.00
##  T:354   1st Qu.:2.000   1st Qu.:1.000   yes:132   1st Qu.: 8.00
##          Median :3.000   Median :1.000             Median :11.00
##          Mean   :2.749   Mean   :1.448             Mean   :10.91
##          3rd Qu.:4.000   3rd Qu.:2.000             3rd Qu.:13.00
##          Max.   :4.000   Max.   :4.000             Max.   :19.00
##        G2              G3           Pass              Grade
##  Min.   : 0.00   Min.   : 0.00   FAIL:130   Excellent   : 40
##  1st Qu.: 9.00   1st Qu.: 8.00   PASS:265   FAIL        :130
##  Median :11.00   Median :11.00              Good        : 60
##  Mean   :10.71   Mean   :10.42              Satisfactory: 62
##  3rd Qu.:13.00   3rd Qu.:14.00              Sufficient  :103
##  Max.   :19.00   Max.   :20.00
```

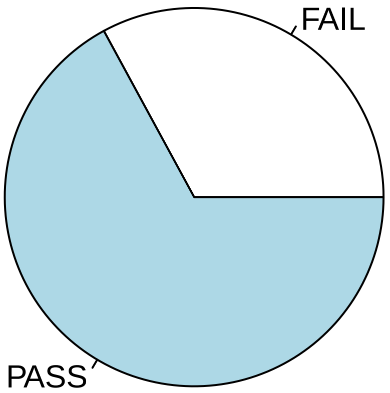
Factorize continuous predictor variables

Variable distribution after necessary factorisation

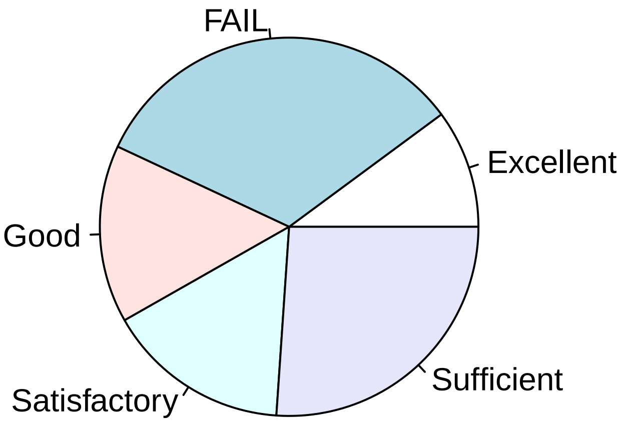
```
students$Medu <- factor(students$Medu)
students$traveltime <- factor(students$traveltime)
summary(students)
```

```
##  Pstatus Medu    traveltime romantic        G1              G2
##  A: 41   0:  3   1:257      no :263   Min.   : 3.00   Min.   : 0.00
##  T:354   1: 59   2:107      yes:132   1st Qu.: 8.00   1st Qu.: 9.00
##          2:103   3: 23                Median :11.00   Median :11.00
##          3: 99   4:  8                Mean   :10.91   Mean   :10.71
##          4:131                        3rd Qu.:13.00   3rd Qu.:13.00
##                                       Max.   :19.00   Max.   :19.00
##        G3          Pass              Grade
##  Min.   : 0.00   FAIL:130   Excellent   : 40
##  1st Qu.: 8.00   PASS:265   FAIL        :130
##  Median :11.00              Good        : 60
##  Mean   :10.42              Satisfactory: 62
##  3rd Qu.:14.00              Sufficient  :103
##  Max.   :20.00
```

Pie chart for pass

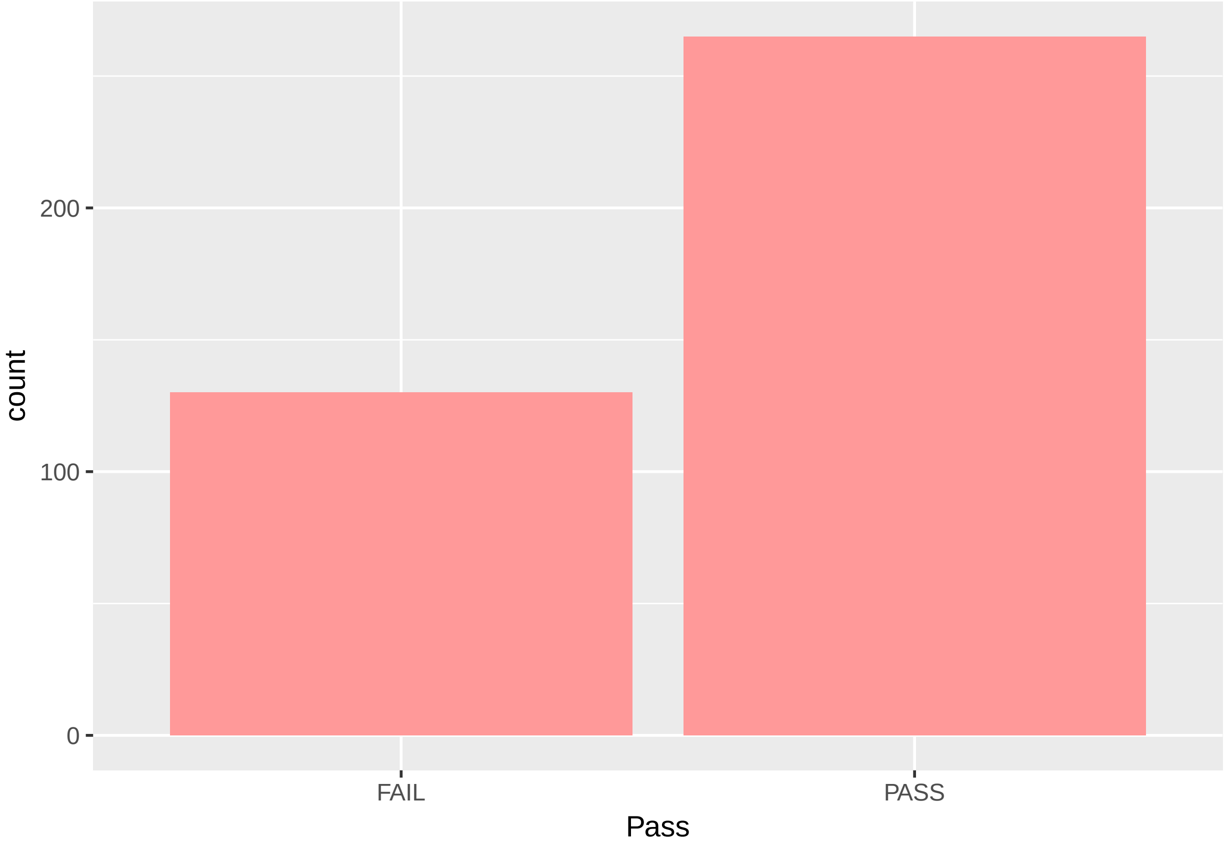


Pie chart for grade

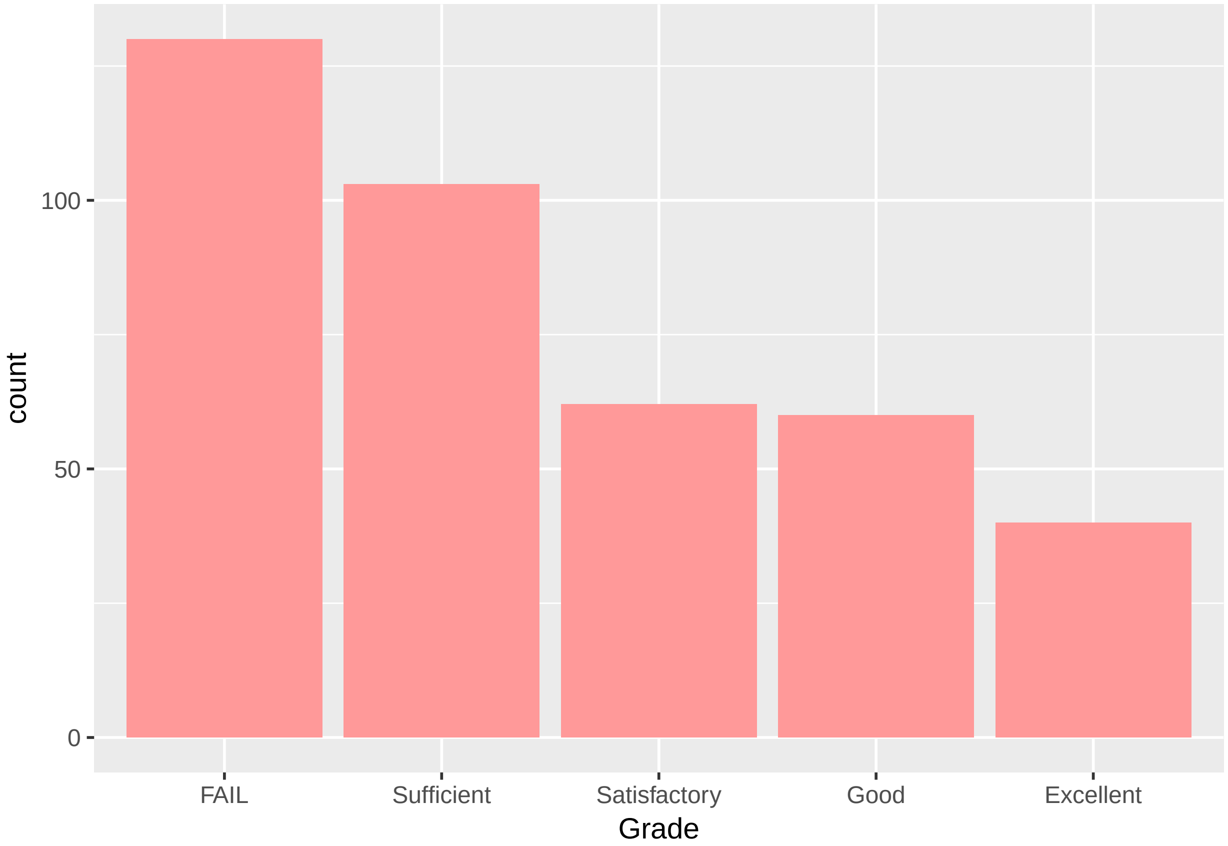


Bar graph for pass

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```



Bar graph for grade



Statistical data of G3

```
summary(with(students,G3))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    8.00   11.00   10.42   14.00   20.00
```

```
sprintf('variance is %f',var(with(students,G3)))
```

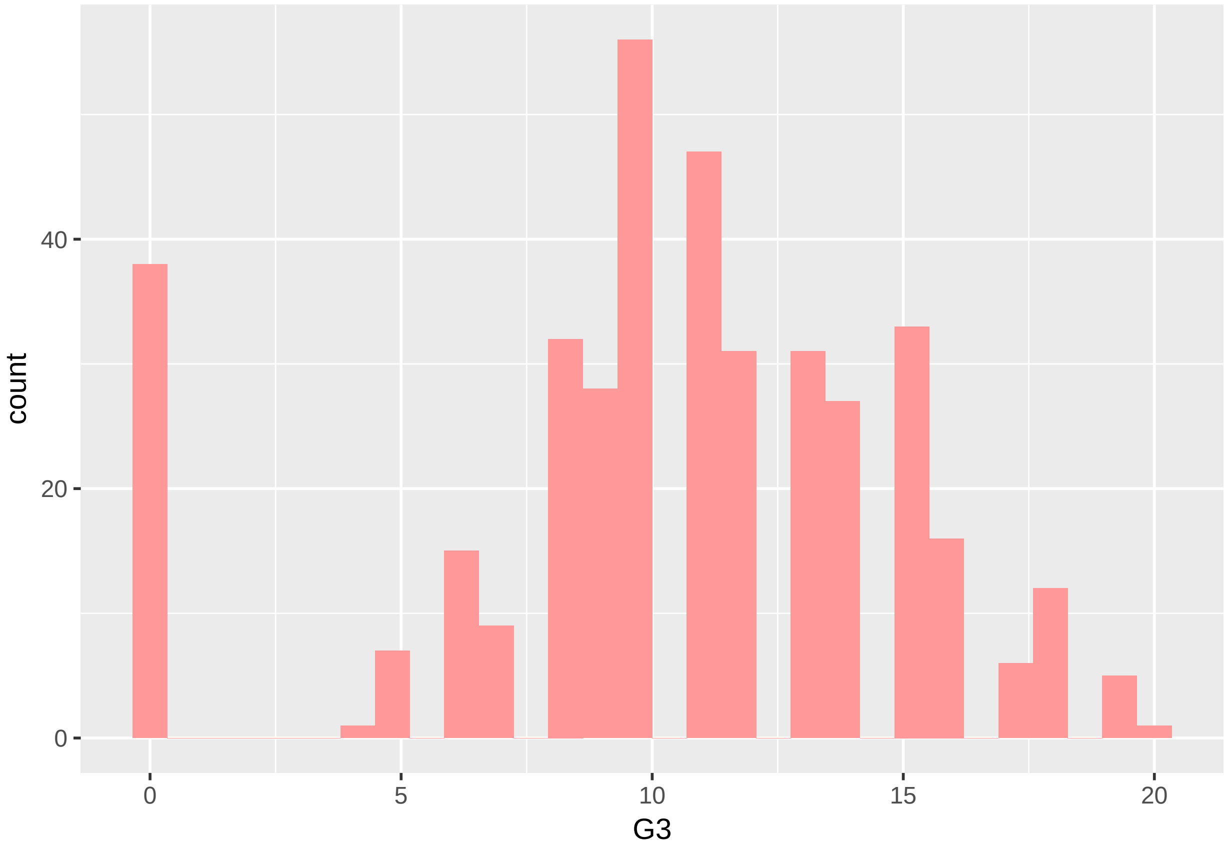
```
## [1] "variance is 20.989616"
```

```
sprintf('standard deviation is %f',sd(with(students,G3)))
```

```
## [1] "standard deviation is 4.581443"
```

Histogram for G3

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



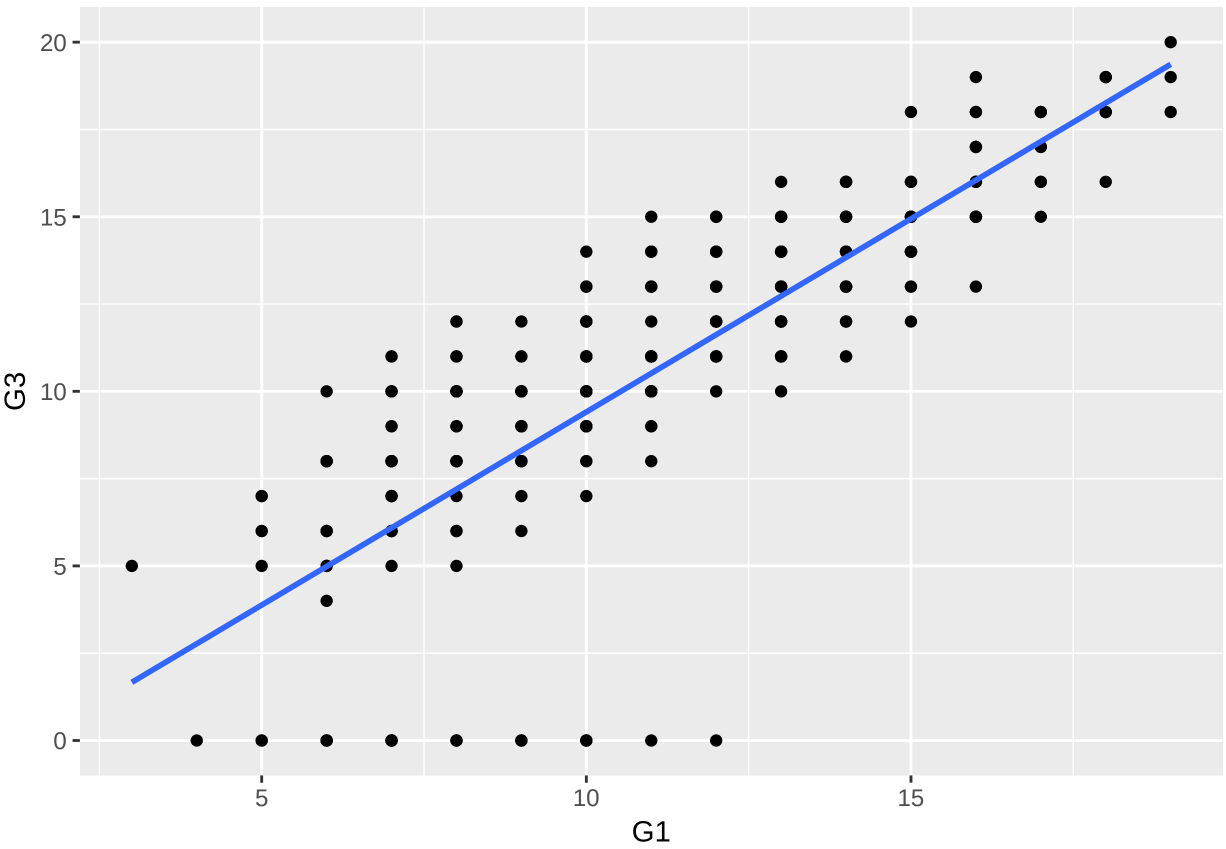
Predicting G3 using G1

Correlation between G1 and G3

```
r <- cor(with(students,G1), with(students,G3))
sprintf("G3 shows a positive correllation with G1")
```

```
## [1] "G3 shows a positive correllation with G1"
```

Scatterplot of G1, G3



### Fit linear regression using G1 as predictor to predict G3

```
fit <- with(students,lm(G3 ~ G1))
fit
```

```
##
## Call:
## lm(formula = G3 ~ G1)
##
## Coefficients:
## (Intercept)           G1
##      -1.653        1.106
```

```
attributes(fit)
```

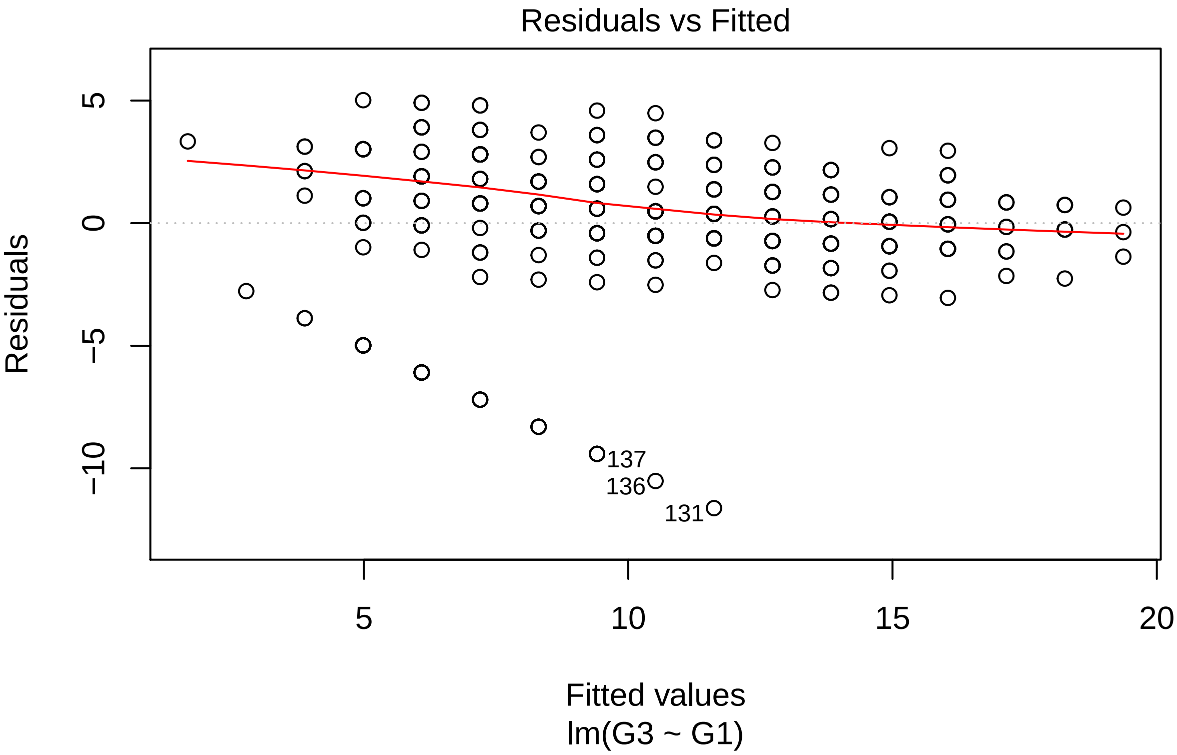
```
## $names
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
##
## $class
## [1] "lm"
```

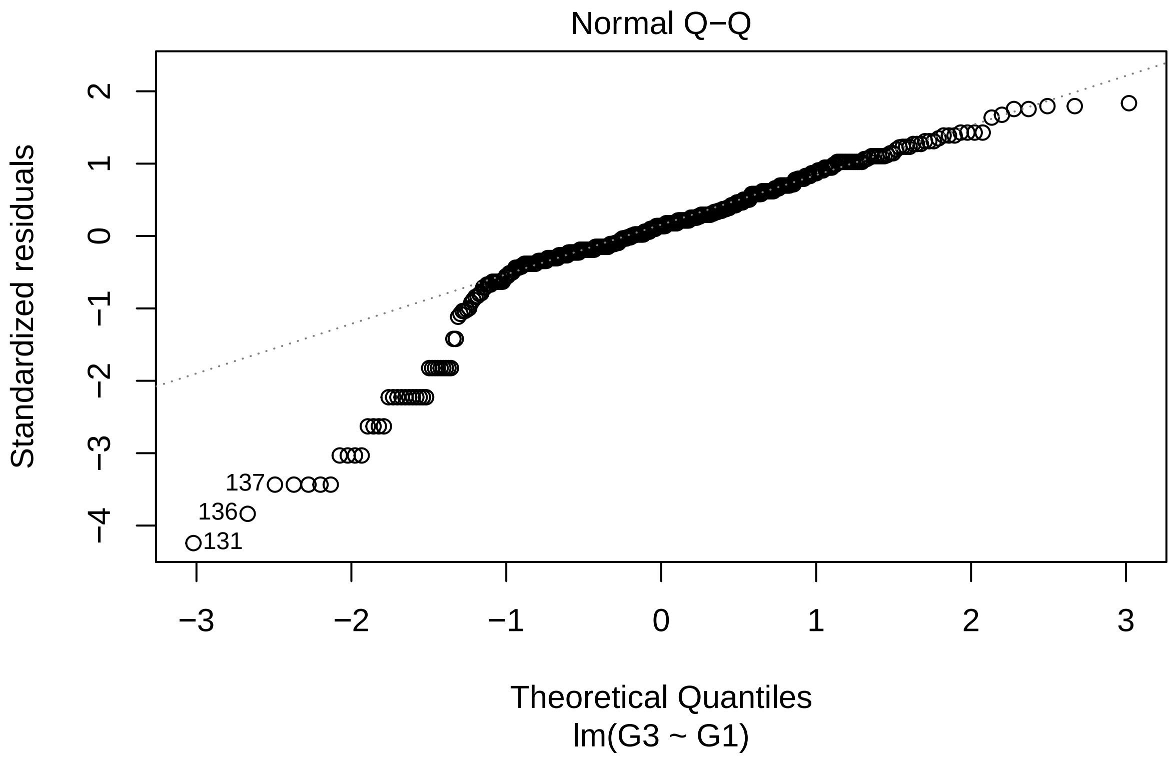
```
summary(fit)
```

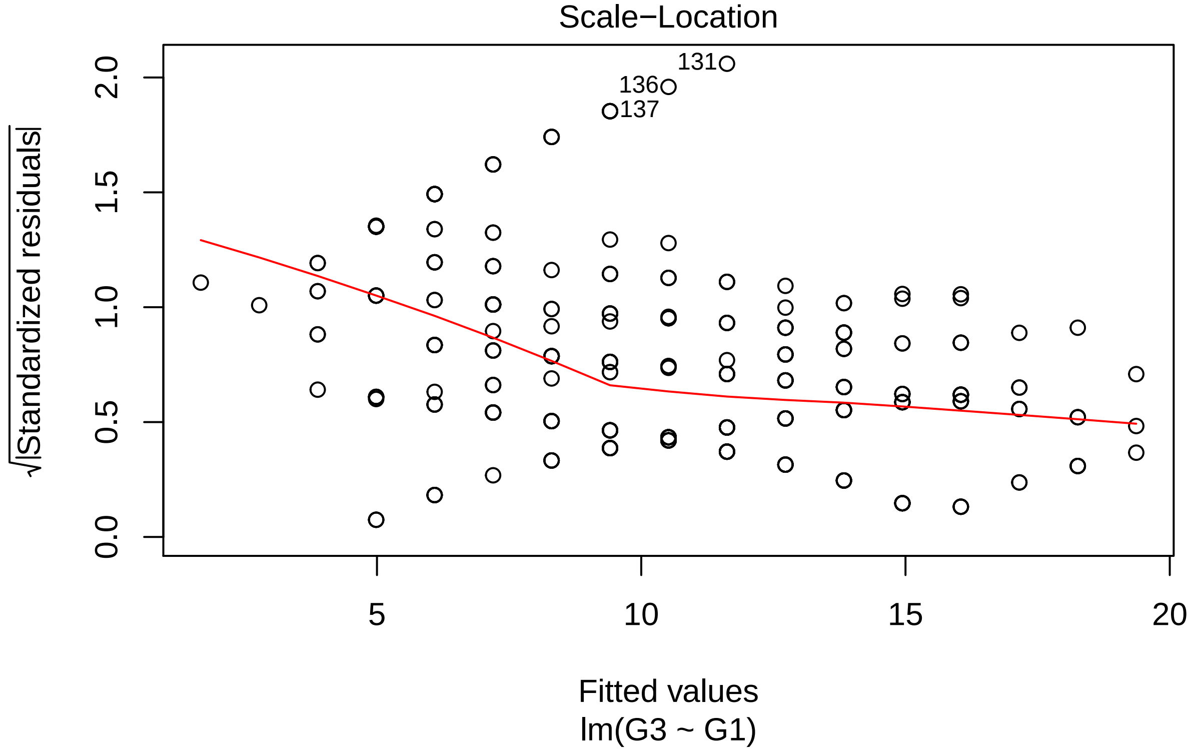
```
##
## Call:
## lm(formula = G3 ~ G1)
```

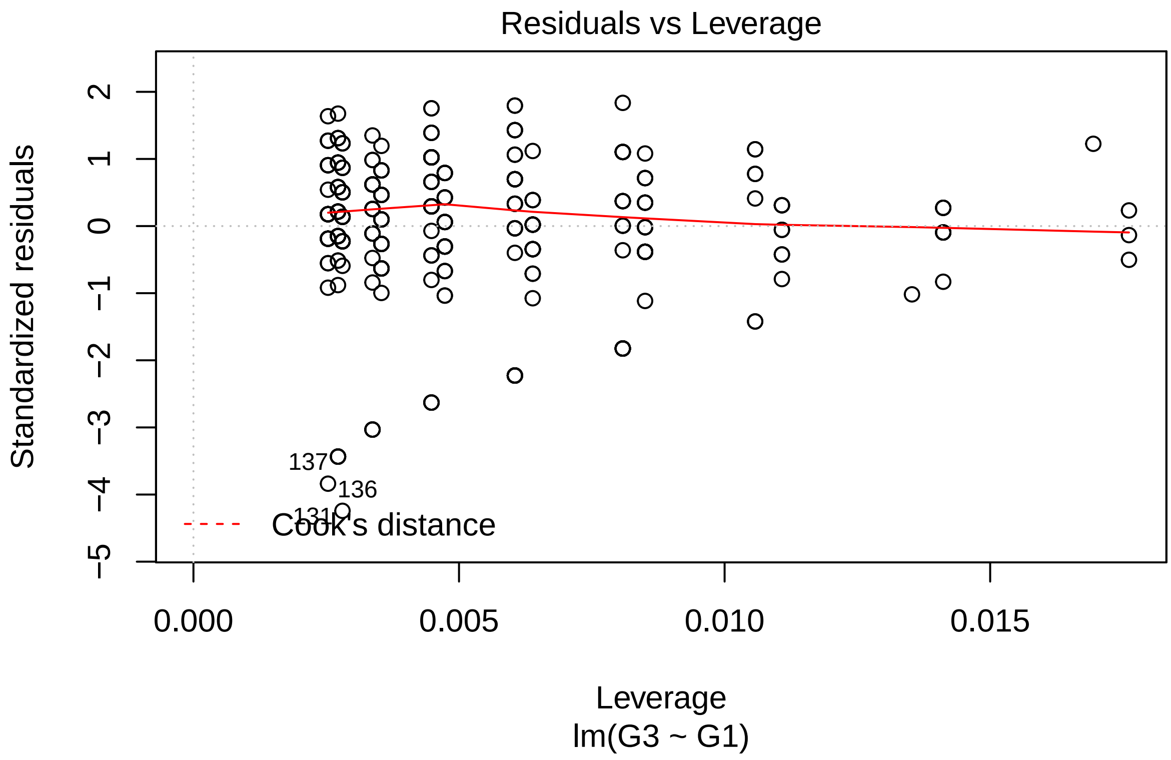
```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6223  -0.8348   0.3777   1.6965   5.0153
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.65280    0.47475  -3.481 0.000555 ***
## G1           1.10626    0.04164  26.568  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.743 on 393 degrees of freedom
## Multiple R-squared:  0.6424, Adjusted R-squared:  0.6414
## F-statistic: 705.8 on 1 and 393 DF,  p-value: < 2.2e-16
```

Plotting line of best fit




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(G3 ~ G1)
−4
−3
−2
−1
0
1
2
−3
−2
−1
0
1
2
3
137
136
131


Scale−Location
√|Standardized residuals|
Fitted values
lm(G3 ~ G1)
0.0
0.5
1.0
1.5
2.0
5
10
15
20
131
136
137



```
## [1] "Residual graph is random in nature suggesting linear regression is not a bad choice for this da
```

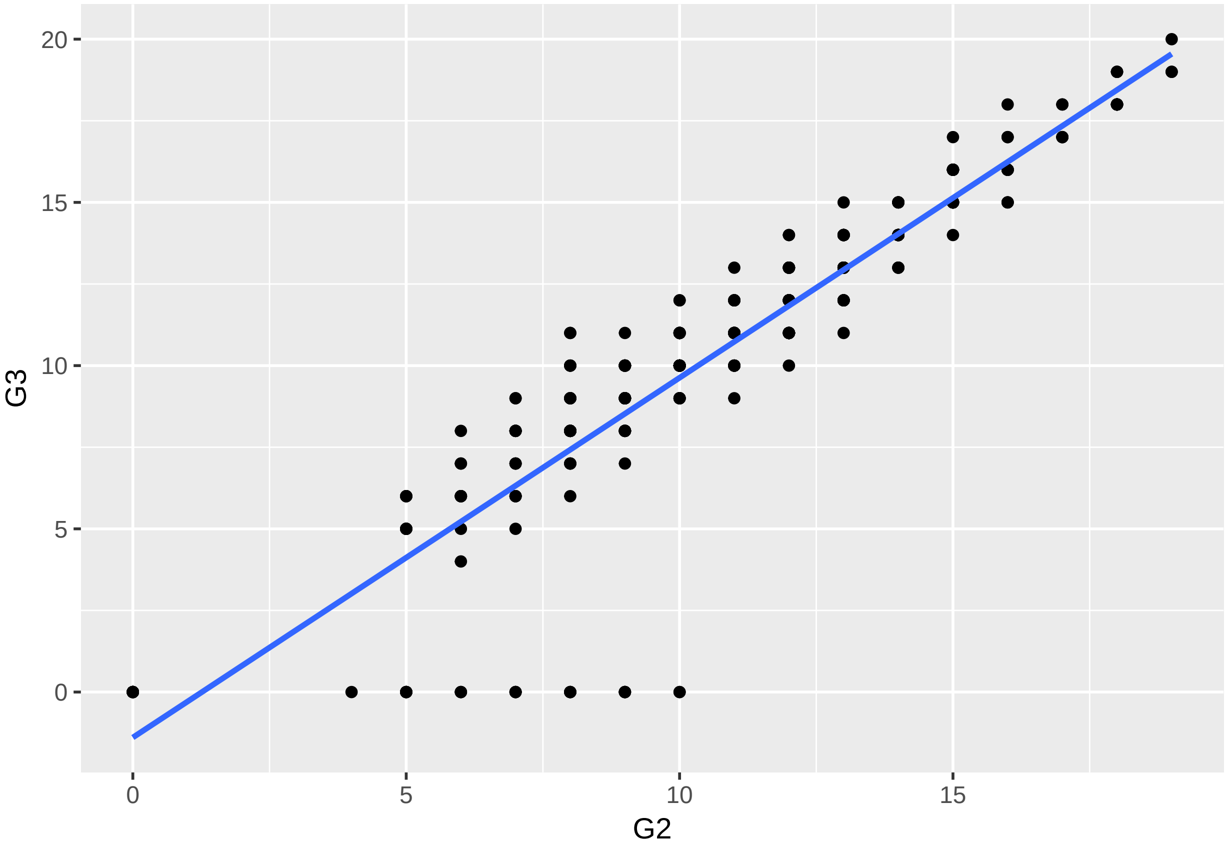
Predicting G3 using G2

Correlation between G2 and G3

```
r <- cor(with(students,G2), with(students,G3))
sprintf('Correlation between G2 and G3 is %f and the coefficient of determination is %f',r, r^2)
```

```
## [1] "Correlation between G2 and G3 is 0.904868 and the coefficient of determination is 0.818786"
```

Scatterplot of G2, G3



### Fit linear regression using G1 as predictor to predict G3

```
fit <- with(students,lm(G3 ~ G2))
fit
```

```
##
## Call:
## lm(formula = G3 ~ G2)
##
## Coefficients:
## (Intercept)           G2
##      -1.393        1.102
```

```
attributes(fit)
```

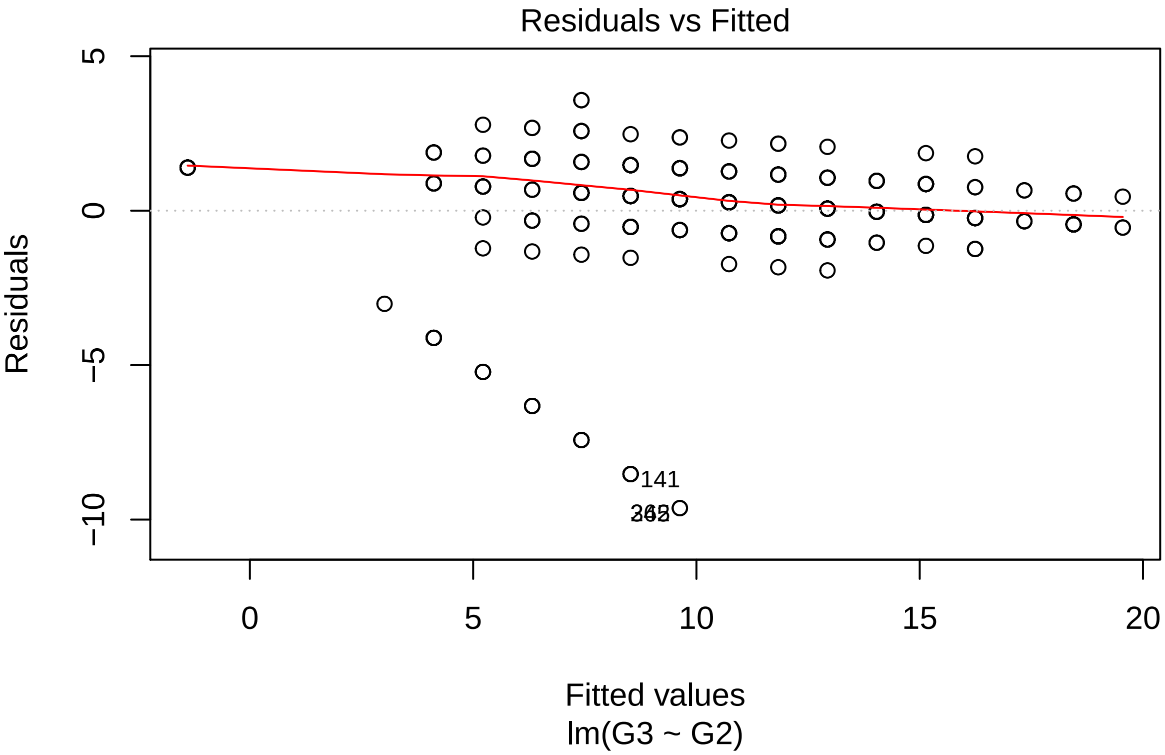
```
## $names
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
##
## $class
## [1] "lm"
```

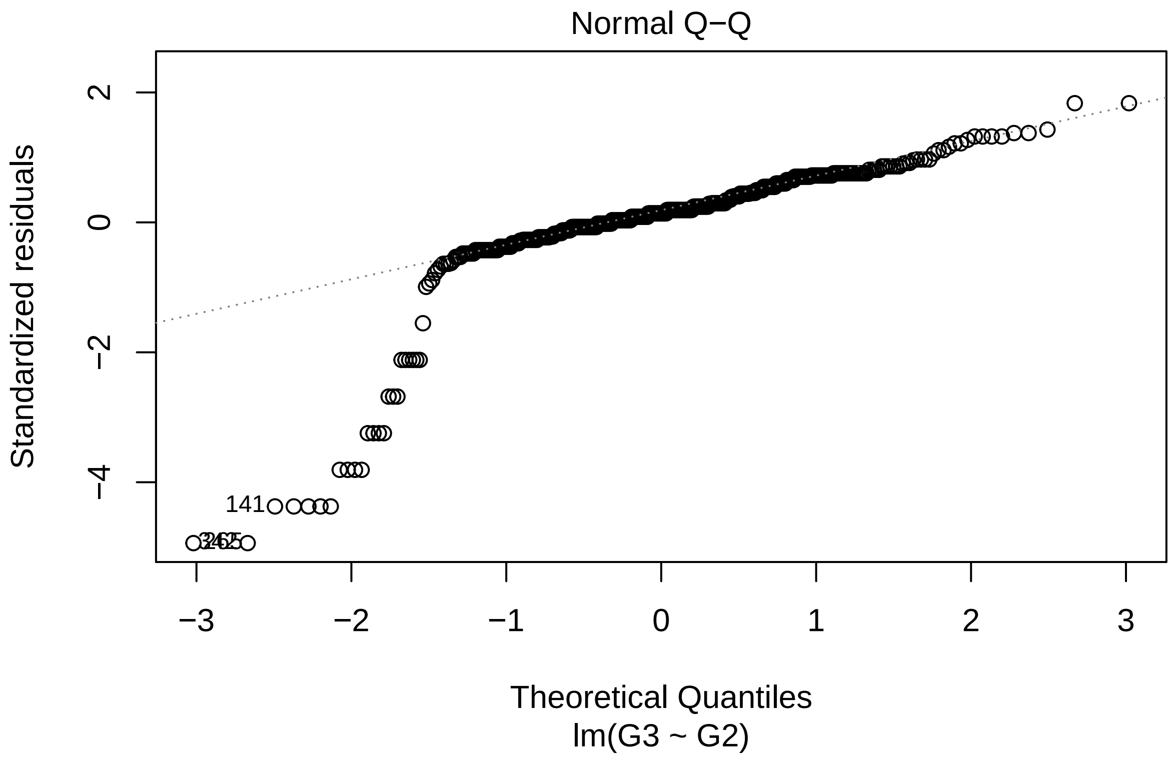
```
summary(fit)
```

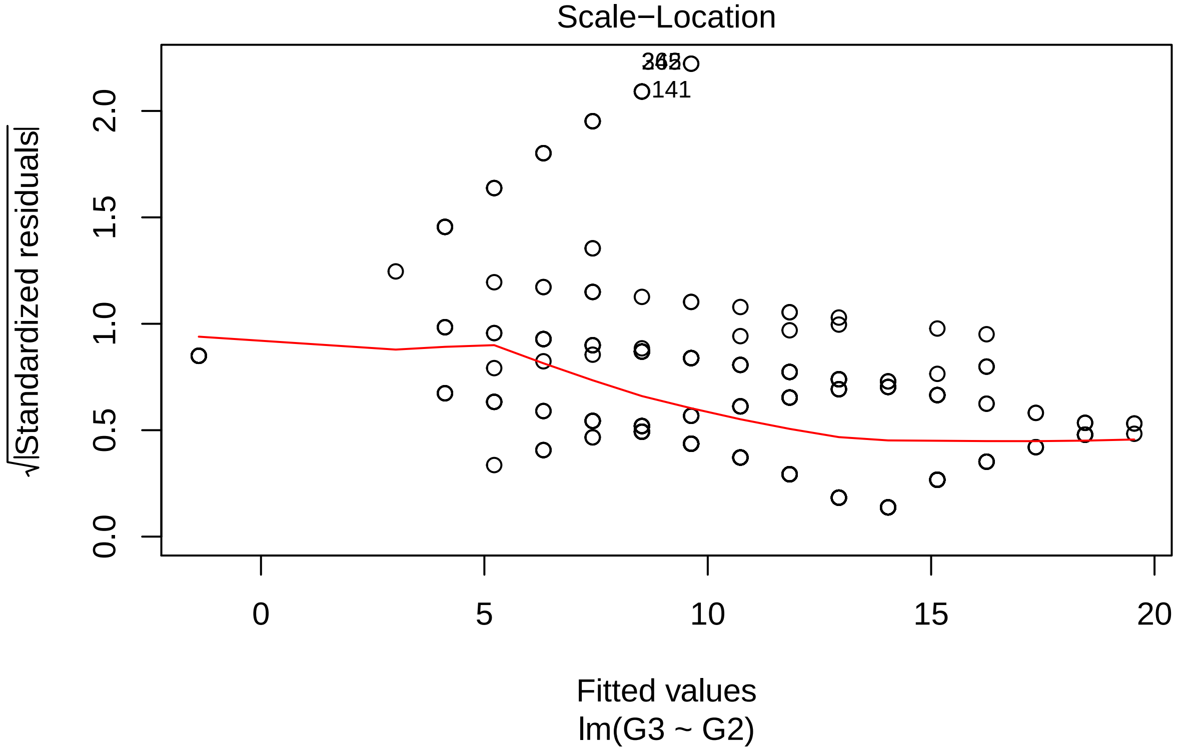
```
##
## Call:
## lm(formula = G3 ~ G2)
```

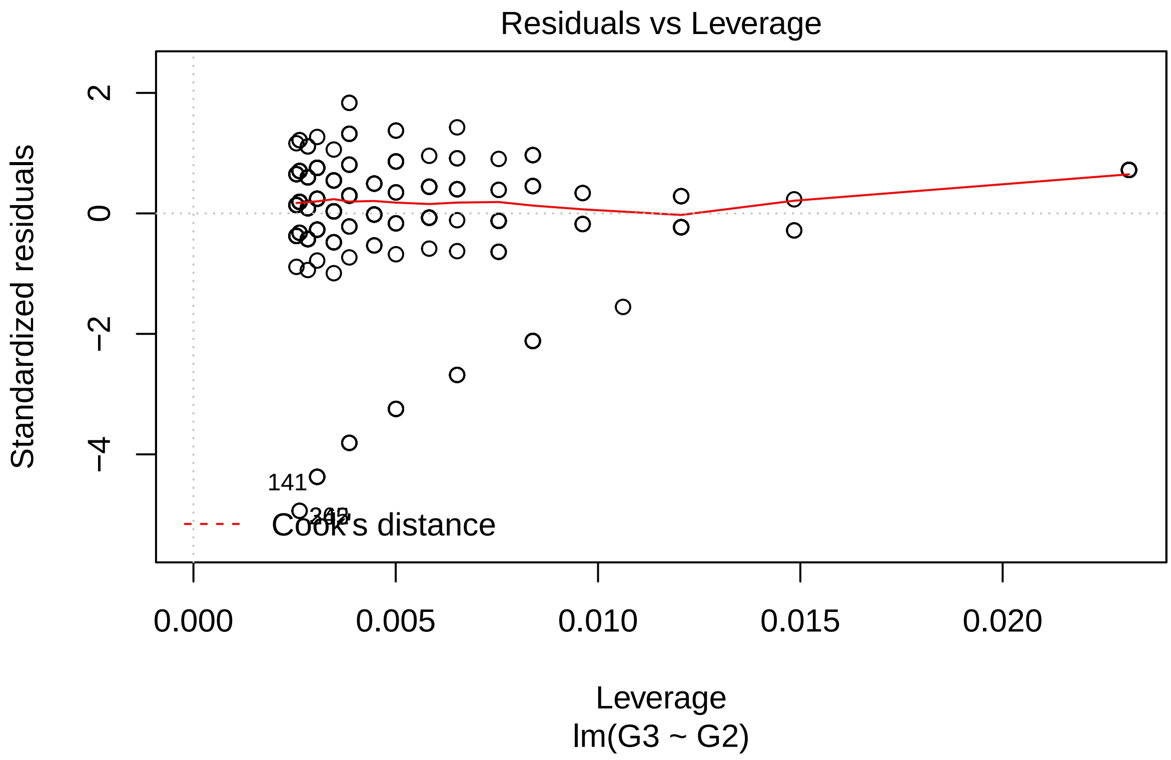
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6284 -0.3326  0.2695  1.0653  3.5759
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.39276    0.29694   -4.69 3.77e-06 ***
## G2           1.10211    0.02615   42.14  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.953 on 393 degrees of freedom
## Multiple R-squared:  0.8188, Adjusted R-squared:  0.8183
## F-statistic:  1776 on 1 and 393 DF,  p-value: < 2.2e-16
```

Plotting line of best fit




Normal Q−Q
Standardized residuals
2
0
−2
−4
141
34
267
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(G3 ~ G2)


Scale−Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
265
342
141
0
5
10
15
20
Fitted values
lm(G3 ~ G2)



```
## [1] "Residual graph is random in nature suggesting linear regression is not a bad choice for this da
```

To predict Pass and Fail using G1 + G2

Using Decision Tree

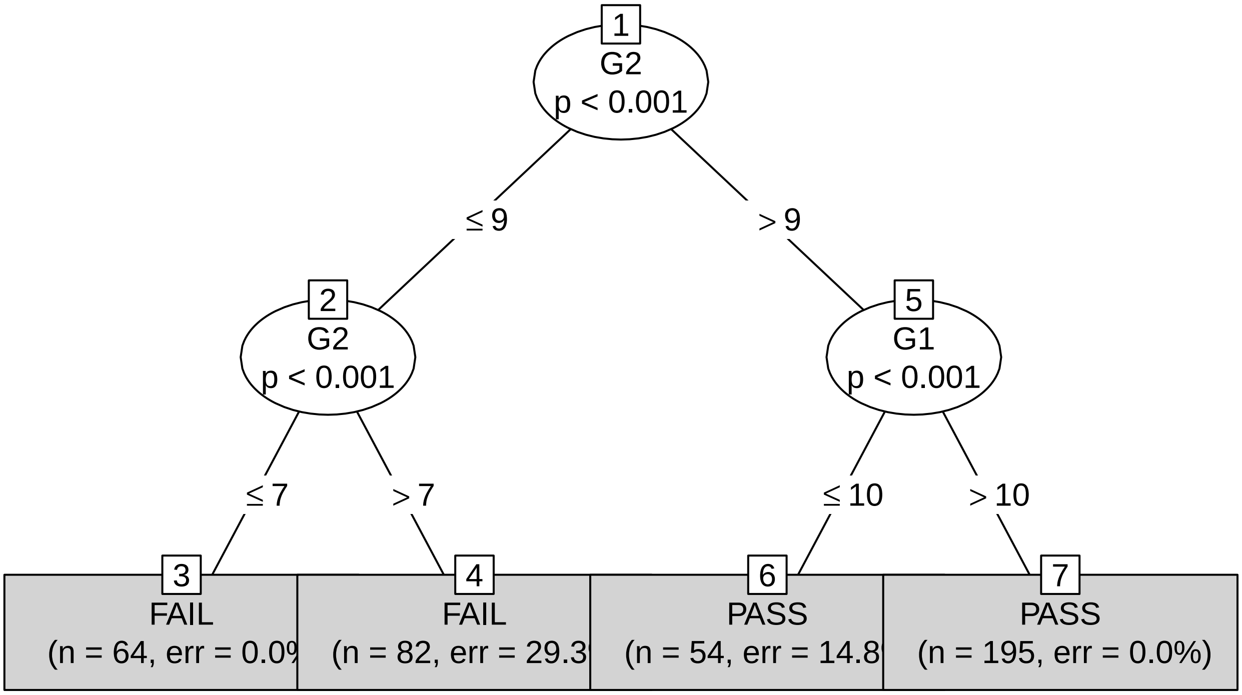
```
library(partykit)
```

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

```
formula <- Pass ~ G1 + G2
tree <- ctree(formula, data=students)
print(tree)
```

```
##
## Model formula:
## Pass ~ G1 + G2
##
## Fitted party:
## [1] root
## |   [2] G2 <= 9
## |   |   [3] G2 <= 7: FAIL (n = 64, err = 0.0%)
## |   |   [4] G2 > 7: FAIL (n = 82, err = 29.3%)
## |   [5] G2 > 9
## |   |   [6] G1 <= 10: PASS (n = 54, err = 14.8%)
## |   |   [7] G1 > 10: PASS (n = 195, err = 0.0%)
##
## Number of inner nodes:    3
## Number of terminal nodes: 4
```

```
plot(tree,type = "simple")
```



```
sprintf('Errors-on-predictions Matrix')
```

```
## [1] "Errors-on-predictions Matrix"
```

```
table(predict(tree, newdata=students), students$Pass,dnn=c('Predicted','Actual'))
```

```
##           Actual
## Predicted FAIL PASS
##      FAIL  122   24
##      PASS    8  241
```

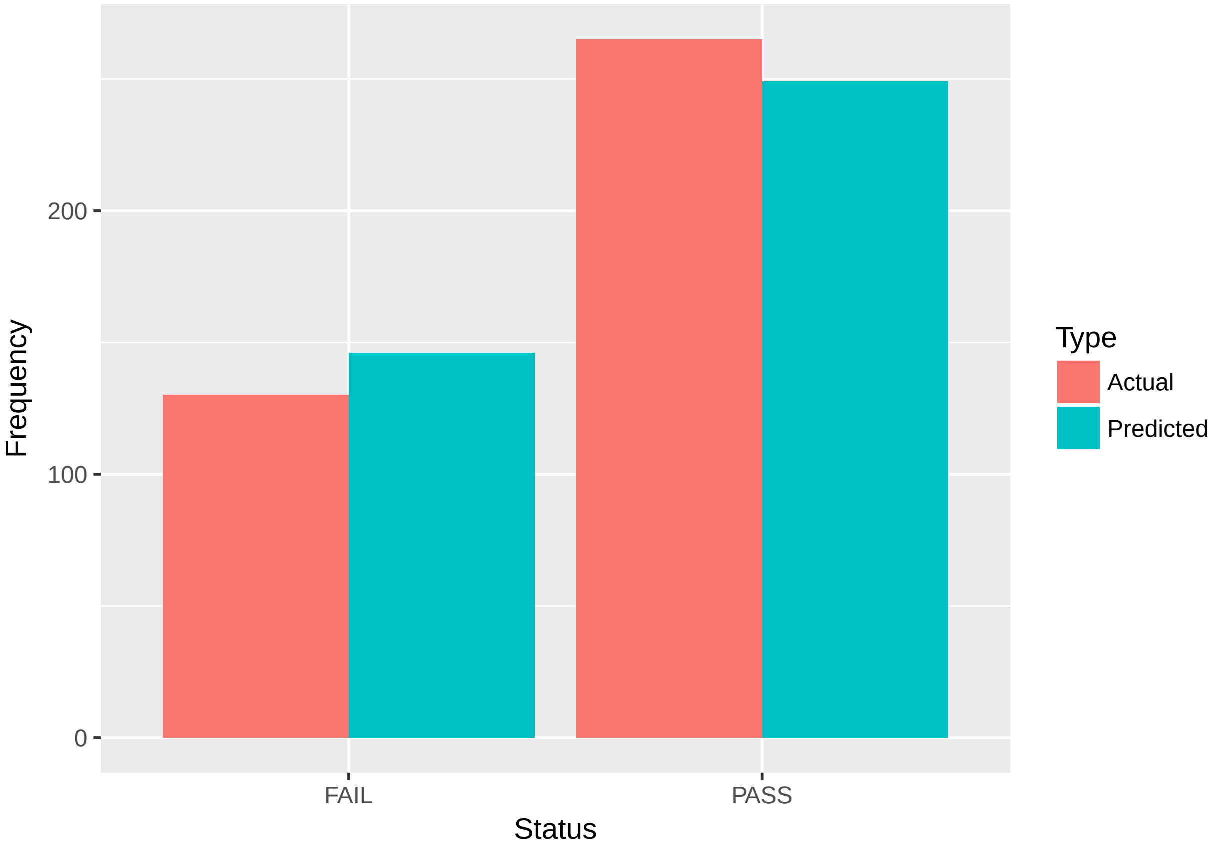
```
df.confmatrix <- data.frame(table(predict(tree, newdata=students), students$Pass,dnn=c('Predicted','Actu
library(tidyr)
data_long <- gather(df.confmatrix, Type, Status, Predicted:Actual)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data_long <- data_long %>% group_by(Status,Type) %>% summarise(Frequency=sum(Freq))
ggplot(data_long, aes(x=Status,y=Frequency,fill=Type)) + geom_bar(stat='identity', position='dodge')
```



Using Naive-Bayes Prediction

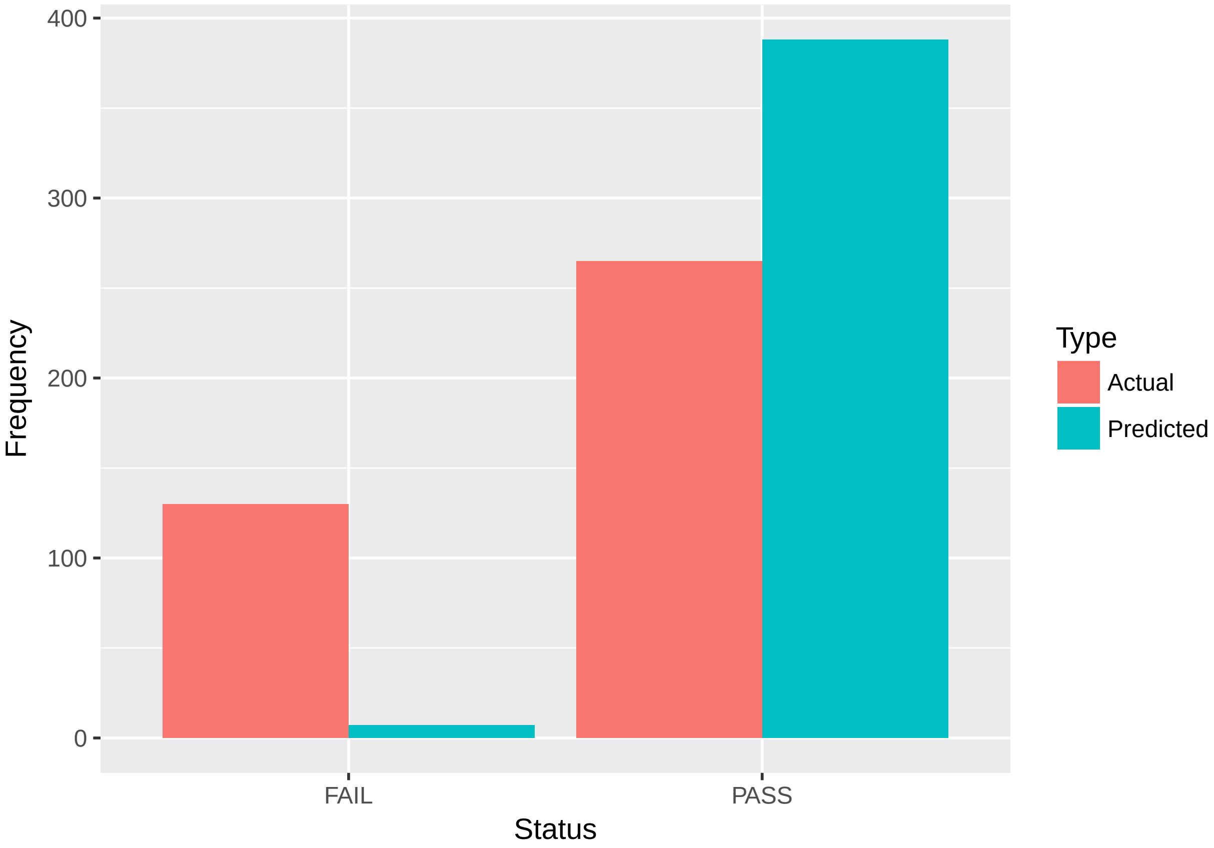
```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.3.2
```

```
classifier<-naiveBayes(students[,1:4], students[,8])
table(predict(classifier, students[,1:4]), students[,8], dnn = c('Predicted','Actual'))
```

```
##          Actual
## Predicted FAIL PASS
##      FAIL    3    4
##      PASS  127  261
```

```
df.confmatrix <- data.frame(table(predict(classifier, students[,1:4]), students[,8], dnn = c('Predicted
data_long <- gather(df.confmatrix, Type, Status, Predicted:Actual)
data_long <- data_long %>% group_by(Status,Type) %>% summarise(Frequency=sum(Freq))
ggplot(data_long, aes(x=Status,y=Frequency,fill=Type)) + geom_bar(stat='identity', position='dodge')
```



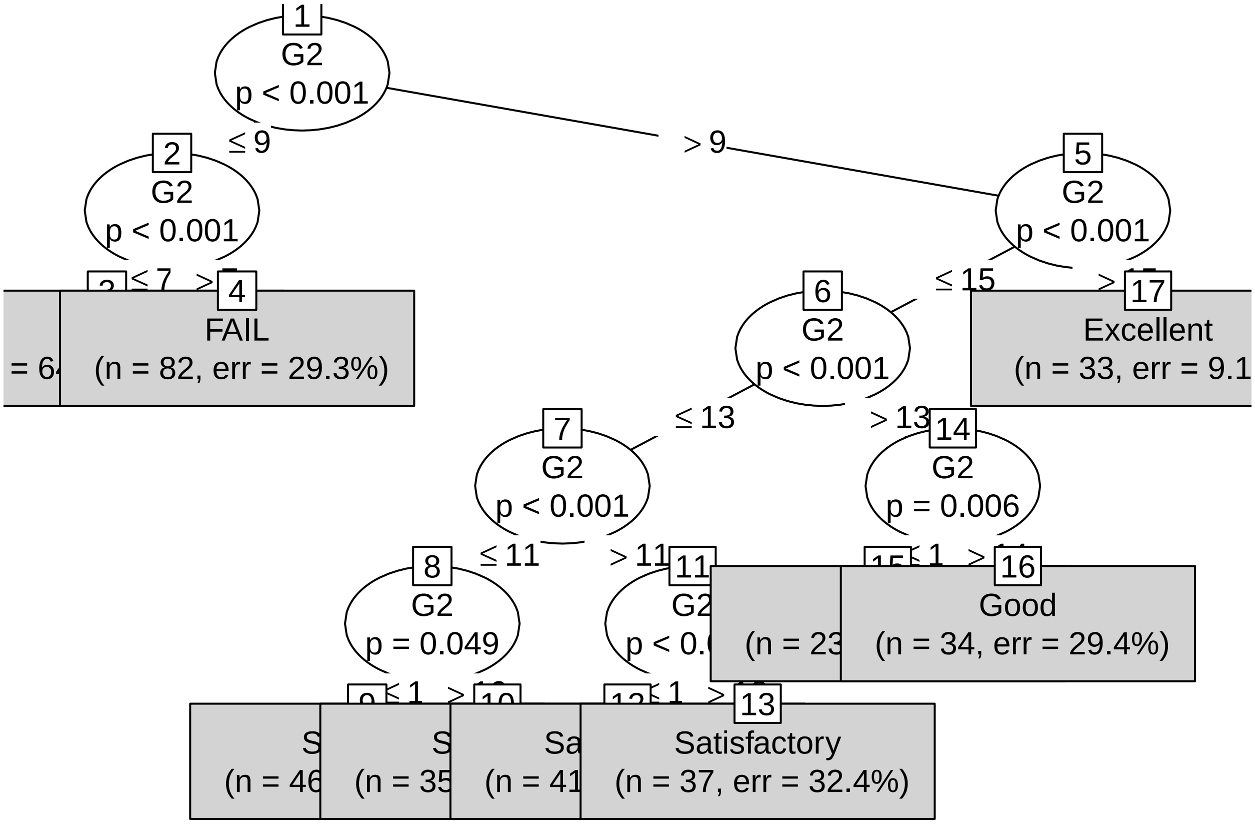
To predict Grades using G1+G3

Using Decision Tree

```
formula <- Grade ~ G1 + G2
tree <- ctree(formula, data=students)
print(tree)
```

```
##
## Model formula:
## Grade ~ G1 + G2
##
## Fitted party:
## [1] root
## |   [2] G2 <= 9
## |   |   [3] G2 <= 7: FAIL (n = 64, err = 0.0%)
## |   |   [4] G2 > 7: FAIL (n = 82, err = 29.3%)
## |   [5] G2 > 9
## |   |   [6] G2 <= 15
## |   |   |   [7] G2 <= 13
## |   |   |   |   [8] G2 <= 11
## |   |   |   |   |   [9] G2 <= 10: Sufficient (n = 46, err = 19.6%)
## |   |   |   |   |   [10] G2 > 10: Sufficient (n = 35, err = 22.9%)
## |   |   |   |   [11] G2 > 11
## |   |   |   |   |   [12] G2 <= 12: Satisfactory (n = 41, err = 39.0%)
## |   |   |   |   |   [13] G2 > 12: Satisfactory (n = 37, err = 32.4%)
## |   |   |   [14] G2 > 13
## |   |   |   |   [15] G2 <= 14: Good (n = 23, err = 13.0%)
## |   |   |   |   [16] G2 > 14: Good (n = 34, err = 29.4%)
## |   |   [17] G2 > 15: Excellent (n = 33, err = 9.1%)
##
## Number of inner nodes:    8
## Number of terminal nodes: 9
```

```
plot(tree,type = "simple")
```



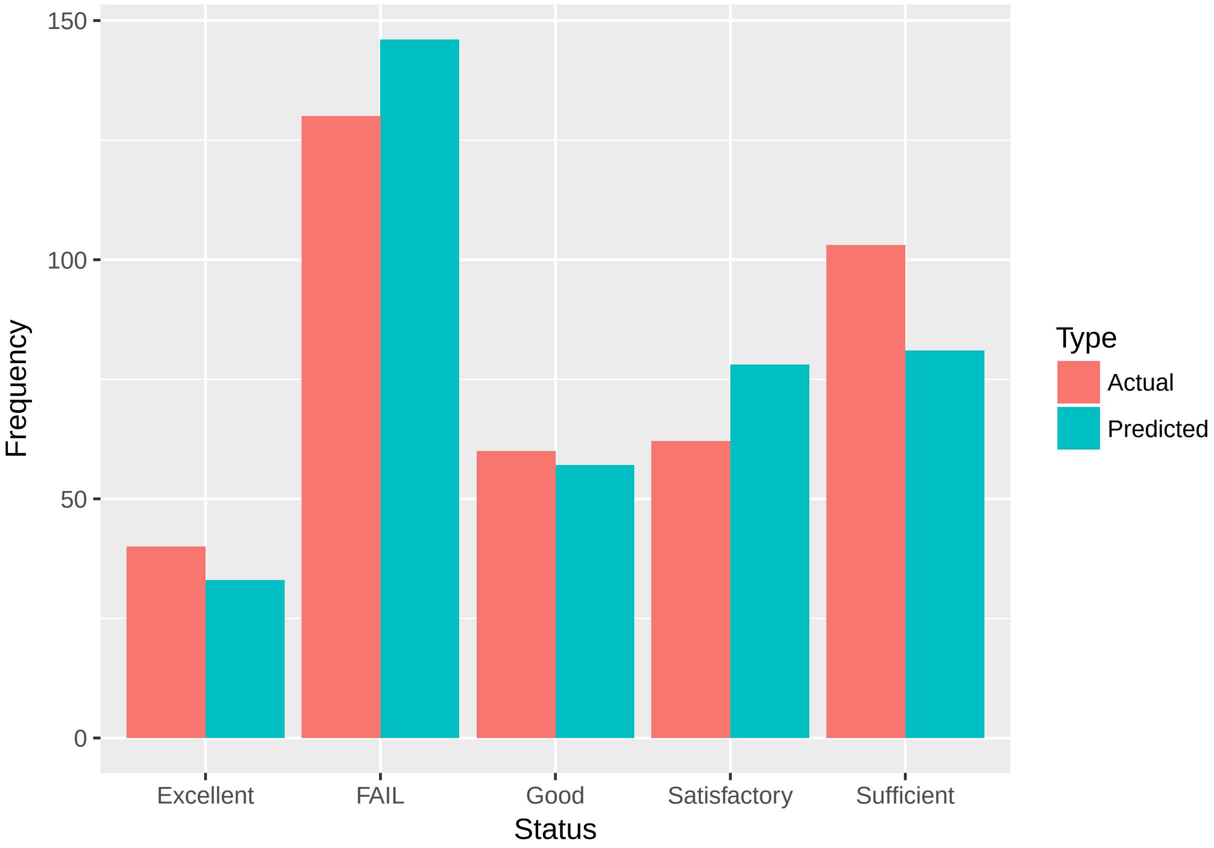
```
sprintf('Errors-on-predictions Matrix')
```

```
## [1] "Errors-on-predictions Matrix"
```

```
table(predict(tree, newdata=students), students$Grade,dnn=c('Predicted','Actual'))
```

```
##                 Actual
## Predicted        FAIL Sufficient Satisfactory Good Excellent
##   FAIL            122         24            0    0         0
##   Sufficient        8         64            9    0         0
##   Satisfactory      0         15           50   13         0
##   Good              0          0            3   44        10
##   Excellent         0          0            0    3        30
```

```
df.confmatrix <- data.frame(table(predict(tree, newdata=students), students$Grade,dnn=c('Predicted','Ac
data_long <- gather(df.confmatrix, Type, Status, Predicted:Actual)
data_long <- data_long %>% group_by(Status,Type) %>% summarise(Frequency=sum(Freq))
ggplot(data_long, aes(x=Status,y=Frequency,fill=Type)) + geom_bar(stat='identity', position='dodge')
```



Using Naive-Bayes Prediction

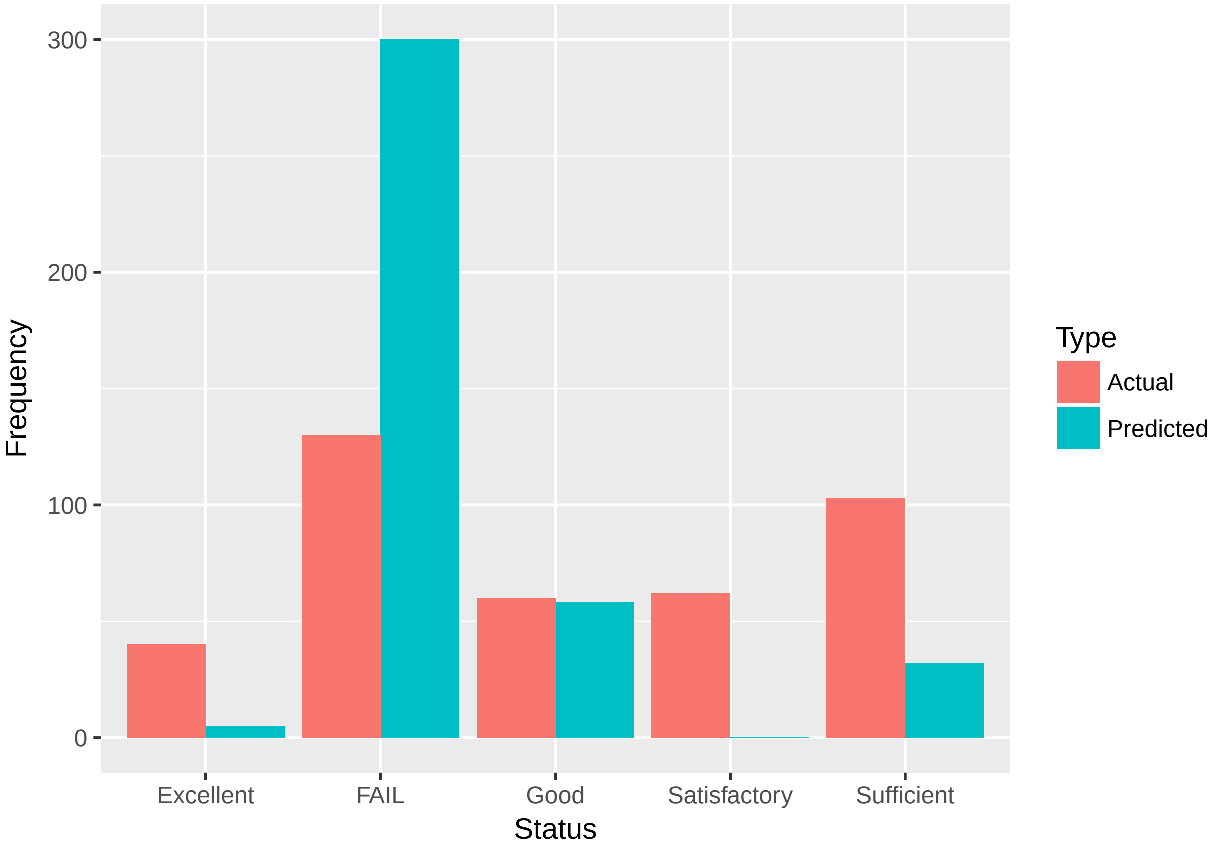
```
classifier<-naiveBayes(students[,1:4], students[,9])
sprintf('Errors-on-predictions Matrix')
```

```
## [1] "Errors-on-predictions Matrix"
```

```
table(predict(classifier, students[,1:4]), students[,9], dnn = c('Predicted','Actual'))
```

```
##               Actual
## Predicted      FAIL Sufficient Satisfactory Good Excellent
##   FAIL          111         78           51   37        23
##   Sufficient      7         12            5    3         5
##   Satisfactory    0          0            0    0         0
##   Good           11         12            6   18        11
##   Excellent       1          1            0    2         1
```

```
df.confmatrix <- data.frame(table(predict(classifier, students[,1:4]), students[,9], dnn = c('Predicted
data_long <- gather(df.confmatrix, Type, Status, Predicted:Actual)
data_long <- data_long %>% group_by(Status,Type) %>% summarise(Frequency=sum(Freq))
ggplot(data_long, aes(x=Status,y=Frequency,fill=Type)) + geom_bar(stat='identity', position='dodge')
```



Conclusion

Linear regression showed strong relationship between G3 and G2. G1 also showed positive relationship but not as strong as G2. Decision tree prediction on the same dataset showed very less errors on predictions making G1 and G2 suitable for predicting Grades and Pass-ability of students Naive Bayes method showed large errors on predictions. So, either the those four variables are not good predictors or Naive Bayes method is not a good predicting model for this dataset. Based on all the analysis, G2 is the strongest predictor for G3, which in turn, for pass-ability and grades.