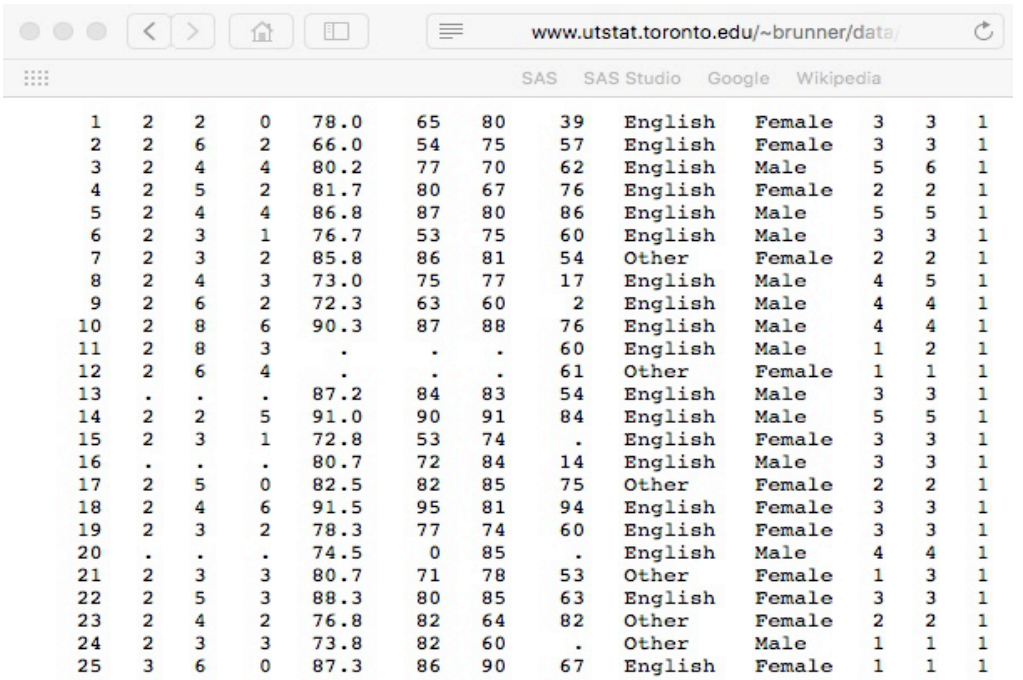
Regression and prediction on the math data

Before the beginning of the Fall term, students in a first-year Calculus class took a diagnostic test with two parts: Pre-calculus and Calculus. Thanks to Dr. Cleo Boyd for permission to use these data. The variables in the data file are:

Identification code
- Course: 1=Catch-up 2=Mainstream 3=Elite 4=NoResponse
- Score on pre-calculus part of diagnostic test
- Score on calculus part of diagnostic test
- High School GPA
- High School Calculus mark
- High School English mark
- University Calculus mark
- First language
- Sex
- National background according to rater one
- National background according to rater two
  - Asian
  - Eastern European
  - European not Eastern
  - Middle-Eastern and Pakistani
  - East Indian
  - Other and DK
- Sample: 1=Exploratory, 2=Replication



1	2	2	0	78.0	65	80	39	English	Female	3	3	1
2	2	6	2	66.0	54	75	57	English	Female	3	3	1
3	2	4	4	80.2	77	70	62	English	Male	5	6	1
4	2	5	2	81.7	80	67	76	English	Female	2	2	1
5	2	4	4	86.8	87	80	86	English	Male	5	5	1
6	2	3	1	76.7	53	75	60	English	Male	3	3	1
7	2	3	2	85.8	86	81	54	Other	Female	2	2	1
8	2	4	3	73.0	75	77	17	English	Male	4	5	1
9	2	6	2	72.3	63	60	2	English	Male	4	4	1
10	2	8	6	90.3	87	88	76	English	Male	4	4	1
11	2	8	3	.	.	.	60	English	Male	1	2	1
12	2	6	4	.	.	.	61	Other	Female	1	1	1
13	.	.	.	87.2	84	83	54	English	Male	3	3	1
14	2	2	5	91.0	90	91	84	English	Male	5	5	1
15	2	3	1	72.8	53	74	.	English	Female	3	3	1
16	.	.	.	80.7	72	84	14	English	Male	3	3	1
17	2	5	0	82.5	82	85	75	Other	Female	2	2	1
18	2	4	6	91.5	95	81	94	English	Female	3	3	1
19	2	3	2	78.3	77	74	60	English	Female	3	3	1
20	.	.	.	74.5	0	85	.	English	Male	4	4	1
21	2	3	3	80.7	71	78	53	Other	Female	1	3	1
22	2	5	3	88.3	80	85	63	English	Female	3	3	1
23	2	4	2	76.8	82	64	82	Other	Female	2	2	1
24	2	3	3	73.8	82	60	.	Other	Male	1	1	1
25	3	6	0	87.3	86	90	67	English	Female	1	1	1

```
> rm(list=ls()); options(scipen=999)
> # Read the exploratory sample (the "training" data)
> training =
read.table("http://www.utstat.toronto.edu/~brunner/data/legal/exploremath.data.txt")
> head(training)
  V1 V2 V3 V4   V5 V6 V7 V8      V9     V10 V11 V12 V13
1  1  2  2  0 78.0 65 80 39 English Female   3   3   1
2  2  2  6  2 66.0 54 75 57 English Female   3   3   1
3  3  2  4  4 80.2 77 70 62 English   Male   5   6   1
4  4  2  5  2 81.7 80 67 76 English Female   2   2   1
5  5  2  4  4 86.8 87 80 86 English   Male   5   5   1
6  6  2  3  1 76.7 53 75 60 English   Male   3   3   1
> dim(training)
[1] 579  13
> colnames(training) = c('id', 'course', 'precalc', 'calc', 'gpa', 'calculus', 'english',
'mark', 'lang', 'sex', 'nation1', 'nation2', 'sample')
> training[1:15,] # First 15 rows
   id course precalc calc  gpa calculus english mark    lang    sex nation1 nation2 sample
1   1      2       2    0 78.0       65      80   39 English Female       3       3      1
2   2      2       6    2 66.0       54      75   57 English Female       3       3      1
3   3      2       4    4 80.2       77      70   62 English   Male       5       6      1
4   4      2       5    2 81.7       80      67   76 English Female       2       2      1
5   5      2       4    4 86.8       87      80   86 English   Male       5       5      1
6   6      2       3    1 76.7       53      75   60 English   Male       3       3      1
7   7      2       3    2 85.8       86      81   54   Other Female       2       2      1
8   8      2       4    3 73.0       75      77   17 English   Male       4       5      1
9   9      2       6    2 72.3       63      60    2 English   Male       4       4      1
10 10      2       8    6 90.3       87      88   76 English   Male       4       4      1
11 11      2       8    3    .        .       .   60 English   Male       1       2      1
12 12      2       6    4    .        .       .   61   Other Female       1       1      1
13 13      .       .    . 87.2       84      83   54 English   Male       3       3      1
14 14      2       2    5 91.0       90      91   84 English   Male       5       5      1
15 15      2       3    1 72.8       53      74    . English Female       3       3      1
> mean(training$calc)
[1] NA
Warning message:
In mean.default(training$calc) :
  argument is not numeric or logical: returning NA
> # Variables 3 through 8 are numeric
> # The following will make . = NA
> for(j in 3:8) training[,j] = as.numeric(as.character(training[,j]))
Warning messages:
1: NAs introduced by coercion
2: NAs introduced by coercion
3: NAs introduced by coercion
4: NAs introduced by coercion
5: NAs introduced by coercion
6: NAs introduced by coercion
> training[1:15,] # First 15 rows
   id course precalc calc  gpa calculus english mark    lang    sex nation1 nation2 sample
1   1      2       2    0 78.0       65      80   39 English Female       3       3      1
2   2      2       6    2 66.0       54      75   57 English Female       3       3      1
3   3      2       4    4 80.2       77      70   62 English   Male       5       6      1
4   4      2       5    2 81.7       80      67   76 English Female       2       2      1
5   5      2       4    4 86.8       87      80   86 English   Male       5       5      1
6   6      2       3    1 76.7       53      75   60 English   Male       3       3      1
7   7      2       3    2 85.8       86      81   54   Other Female       2       2      1
8   8      2       4    3 73.0       75      77   17 English   Male       4       5      1
9   9      2       6    2 72.3       63      60    2 English   Male       4       4      1
10 10      2       8    6 90.3       87      88   76 English   Male       4       4      1
11 11      2       8    3   NA       NA      NA   60 English   Male       1       2      1
12 12      2       6    4   NA       NA      NA   61   Other Female       1       1      1
13 13      .      NA   NA 87.2       84      83   54 English   Male       3       3      1
14 14      2       2    5 91.0       90      91   84 English   Male       5       5      1
15 15      2       3    1 72.8       53      74   NA English Female       3       3      1
```

```
> summary(training)
       id         course     precalc          calc             gpa          calculus
 Min.   :  1.0   .: 99   Min.   :0.000   Min.   : 0.000   Min.   : 0.00   Min.   :  0.00
 1st Qu.:145.5   1: 59   1st Qu.:3.000   1st Qu.: 1.000   1st Qu.:74.50   1st Qu.: 65.00
 Median :290.0   2:373   Median :4.000   Median : 3.000   Median :78.00   Median : 75.00
 Mean   :290.0   3: 39   Mean   :4.402   Mean   : 3.319   Mean   :75.23   Mean   : 97.32
 3rd Qu.:434.5   4:  9   3rd Qu.:6.000   3rd Qu.: 5.000   3rd Qu.:82.25   3rd Qu.: 85.00
 Max.   :579.0           Max.   :9.000   Max.   :11.000   Max.   :97.30   Max.   :999.00
                         NA's   :99      NA's   :99       NA's   :88      NA's   :88
    english           mark           lang         sex         nation1         nation2
 Min.   : 0.00   Min.   :  0.0   .      : 28   .      : 28   Min.   :1.000   Min.   :1.000
 1st Qu.:70.00   1st Qu.: 50.0   English:402   Female:266   1st Qu.:2.000   1st Qu.:2.000
 Median :76.00   Median : 60.0   French :  5   Male  :285   Median :3.000   Median :3.000
 Mean   :74.44   Mean   :118.6   Other  :144                Mean   :3.081   Mean   :3.052
 3rd Qu.:82.00   3rd Qu.: 75.0                              3rd Qu.:4.000   3rd Qu.:4.000
 Max.   :96.00   Max.   :999.0                              Max.   :6.000   Max.   :6.000
 NA's   :90      NA's   :157
     sample
 Min.   :1
 1st Qu.:1
 Median :1
 Mean   :1
 3rd Qu.:1
 Max.   :1

>
> # Data Cleaning
> # Variables needing attention are:
>     # course
>     # gpa : Min is zero
>     # calculus: Min is zero, Max is 999
>     # english: Min is zero
>     # mark:  Min is zero, Max is 999
>     # lang
>     # sex
>
> # Fix up the variables outside the data frame to save typing.
> attach(training)
>
> # Fix up course
> is.factor(course)
[1] TRUE
> course = as.character(course) # I need to make it character to edit
> course[course=='.'] = NA
> course[course=='4'] = NA
> course[course=='1'] = 'Catch-up'
> course[course=='2'] = 'Mainstream'
> course[course=='3'] = 'Elite'
> table(course,useNA="ifany")
course
  Catch-up      Elite Mainstream       <NA>
        59         39        373        108
> course=factor(course)
> contrasts(course)
           Elite Mainstream
Catch-up       0          0
Elite          1          0
Mainstream     0          1
>
> # Fix up gpa
> bad = subset(id,gpa==0); bad
 [1]  39  40  60 106 127 225 235 244 256 263 274 276 284 296 319 342 390 392 408 416 436 504
[23] 518 546 555
> length(bad)
[1] 25
> gpa[bad] = NA
```

```
>
> # Fix up calculus
> bad = subset(id,calculus==0 | calculus == 999) # Logical or
> calculus[bad] = NA
>
> # Fix up english
> bad = subset(id,english==0); english[bad] = NA
>
> # Fix up mark
> bad = subset(id,mark==0 | mark == 999); mark[bad] = NA
>
> # Fix up lang
> lang = as.character(lang) # For editing
> dot = subset(id,lang=='.') ; lang[dot] = NA
> franc = subset(id,lang=='French'); lang[franc] = 'Other'
> lang = factor(lang)
> table(lang,useNA="ifany")
lang
English   Other    <NA>
    402     149      28
>
> # Fix up sex
> dot = subset(id,sex=='.') ; sex[dot] = NA
> sex = factor(sex)
>
> # Fix up nationality ratings
> is.numeric(nation1)
[1] TRUE
> nation1 = factor(nation1, labels = c('Asian', 'Eastern European',
+     'European not Eastern', 'Middle-Eastern and Pakistani', 'East Indian',
+     'Other and DK'))
> nation2 = factor(nation2, labels = c('Asian', 'Eastern European',
+     'European not Eastern', 'Middle-Eastern and Pakistani', 'East Indian',
+     'Other and DK'))
> table(nation1,nation2)
                           nation2
nation1                     Asian Eastern European European not Eastern
  Asian                       113                2                     2
  Eastern European              1               50                     4
  European not Eastern          9                8                   183
  Middle-Eastern and Pakistani  0                2                     1
  East Indian                   0                1                     2
  Other and DK                  8                2                     4
                           nation2
nation1                     Middle-Eastern and Pakistani East Indian Other and DK
  Asian                                                2           5            2
  Eastern European                                     1           3            2
  European not Eastern                                 1           1            2
  Middle-Eastern and Pakistani                        53           8            2
  East Indian                                          1          67            1
  Other and DK                                         1           2           33
```

```
>
> training2 = data.frame(id, course, precalc, calc, gpa, calculus, english, mark,
+     lang, sex, nation1, nation2, sample)
> summary(training2) # Look at mark!
       id               course       precalc          calc             gpa
 Min.   :  1.0   Catch-up  : 59   Min.   :0.000   Min.   : 0.000   Min.   :65.00
 1st Qu.:145.5   Elite     : 39   1st Qu.:3.000   1st Qu.: 1.000   1st Qu.:75.20
 Median :290.0   Mainstream:373   Median :4.000   Median : 3.000   Median :78.30
 Mean   :290.0   NA's      :108   Mean   :4.402   Mean   : 3.319   Mean   :79.27
 3rd Qu.:434.5                    3rd Qu.:6.000   3rd Qu.: 5.000   3rd Qu.:82.50
 Max.   :579.0                    Max.   :9.000   Max.   :11.000   Max.   :97.30
                                  NA's   :99      NA's   :99       NA's   :113
    calculus         english           mark            lang         sex
 Min.   : 50.00   Min.   :50.00   Min.   :  1.00   English:402   Female:266
 1st Qu.: 67.00   1st Qu.:71.00   1st Qu.: 50.00   Other  :149   Male  :285
 Median : 76.00   Median :76.00   Median : 60.00   NA's   : 28   NA's  : 28
 Mean   : 75.44   Mean   :75.84   Mean   : 91.05
 3rd Qu.: 85.00   3rd Qu.:82.00   3rd Qu.: 73.00
 Max.   :100.00   Max.   :96.00   Max.   :998.00
 NA's   :131      NA's   :99      NA's   :172
                        nation1                                nation2        sample
 Asian                      :126   Asian                      :131   Min.   :1
 Eastern European           : 61   Eastern European           : 65   1st Qu.:1
 European not Eastern       :204   European not Eastern       :196   Median :1
 Middle-Eastern and Pakistani: 66   Middle-Eastern and Pakistani: 59   Mean   :1
 East Indian                : 72   East Indian                : 86   3rd Qu.:1
 Other and DK               : 50   Other and DK               : 42   Max.   :1

>
> bad = subset(id,mark==998); mark[bad] = NA
> training2 = data.frame(id, course, precalc, calc, gpa, calculus, english, mark,
+     lang, sex, nation1, nation2, sample)
> summary(training2)
       id               course       precalc          calc             gpa
 Min.   :  1.0   Catch-up  : 59   Min.   :0.000   Min.   : 0.000   Min.   :65.00
 1st Qu.:145.5   Elite     : 39   1st Qu.:3.000   1st Qu.: 1.000   1st Qu.:75.20
 Median :290.0   Mainstream:373   Median :4.000   Median : 3.000   Median :78.30
 Mean   :290.0   NA's      :108   Mean   :4.402   Mean   : 3.319   Mean   :79.27
 3rd Qu.:434.5                    3rd Qu.:6.000   3rd Qu.: 5.000   3rd Qu.:82.50
 Max.   :579.0                    Max.   :9.000   Max.   :11.000   Max.   :97.30
                                  NA's   :99      NA's   :99       NA's   :113
    calculus         english           mark            lang         sex
 Min.   : 50.00   Min.   :50.00   Min.   : 1.00   English:402   Female:266
 1st Qu.: 67.00   1st Qu.:71.00   1st Qu.:50.00   Other  :149   Male  :285
 Median : 76.00   Median :76.00   Median :60.00   NA's   : 28   NA's  : 28
 Mean   : 75.44   Mean   :75.84   Mean   :58.74
 3rd Qu.: 85.00   3rd Qu.:82.00   3rd Qu.:72.00
 Max.   :100.00   Max.   :96.00   Max.   :99.00
 NA's   :131      NA's   :99      NA's   :186
                        nation1                                nation2        sample
 Asian                      :126   Asian                      :131   Min.   :1
 Eastern European           : 61   Eastern European           : 65   1st Qu.:1
 European not Eastern       :204   European not Eastern       :196   Median :1
 Middle-Eastern and Pakistani: 66   Middle-Eastern and Pakistani: 59   Mean   :1
 East Indian                : 72   East Indian                : 86   3rd Qu.:1
 Other and DK               : 50   Other and DK               : 42   Max.   :1

>
> sum(complete.cases(training2)) # Count TRUE = 1
[1] 287
```

```
> # Now try a regression model
>
> model0 = lm(mark ~ course + precalc + calc + gpa + calculus + english + lang +
+                    sex + nation1 + nation2)
> summary(model0)

Call:
lm(formula = mark ~ course + precalc + calc + gpa + calculus +
    english + lang + sex + nation1 + nation2)

Residuals:
    Min      1Q  Median      3Q     Max
-45.398  -6.332   1.113   9.019  30.629

Coefficients:
                                    Estimate Std. Error t value        Pr(>|t|)
(Intercept)                         -65.7349    12.2134  -5.382 0.0000001611610 ***
courseElite                          -8.0192     5.2316  -1.533         0.12650
courseMainstream                     -6.0256     4.4576  -1.352         0.17759
precalc                               1.8045     0.5714   3.158         0.00177 **
calc                                  0.6932     0.3893   1.781         0.07609 .
gpa                                   1.6421     0.2319   7.082 0.0000000000126 ***
calculus                              0.2159     0.1062   2.033         0.04306 *
english                              -0.3995     0.1331  -3.002         0.00294 **
langOther                             4.8891     2.2324   2.190         0.02939 *
sexMale                              -1.7960     1.7618  -1.019         0.30891
nation1Eastern European               4.7117     5.3038   0.888         0.37514
nation1European not Eastern           2.4248     4.1209   0.588         0.55675
nation1Middle-Eastern and Pakistani   9.8634     5.7775   1.707         0.08895 .
nation1East Indian                    7.6569     5.4812   1.397         0.16359
nation1Other and DK                   3.0353     4.9914   0.608         0.54363
nation2Eastern European              -3.9862     5.1345  -0.776         0.43823
nation2European not Eastern          -0.7450     4.0750  -0.183         0.85508
nation2Middle-Eastern and Pakistani  -9.4722     5.6745  -1.669         0.09624 .
nation2East Indian                   -2.1218     5.2468  -0.404         0.68624
nation2Other and DK                   0.4945     6.1543   0.080         0.93602
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.62 on 267 degrees of freedom
  (292 observations deleted due to missingness)
Multiple R-squared:  0.4883,     Adjusted R-squared:  0.4519
F-statistic: 13.41 on 19 and 267 DF,  p-value: < 0.00000000000000022

>
> # Combine the ratings of apparent national origin
> nation = nation2
> sub1 = subset(id,nation1=='Middle-Eastern and Pakistani')
> nation[sub1] = nation1[sub1] # Use rater one's opinion in this case
> # If Stuart says they are Middle-Eastern and Pakistani, I believe it.
```

```
>
> model1 = lm(mark ~ course + precalc + calc + gpa + calculus + english + lang
+                  + sex + nation)
> summary(model1)

Call:
lm(formula = mark ~ course + precalc + calc + gpa + calculus +
    english + lang + sex + nation)

Residuals:
    Min      1Q  Median      3Q     Max
-45.327  -6.181   0.919   8.859  30.619

Coefficients:
                                 Estimate Std. Error t value        Pr(>|t|)
(Intercept)                      -65.0793    11.8793  -5.478 0.0000000977655 ***
courseElite                       -7.3723     5.1914  -1.420         0.15673
courseMainstream                  -5.1113     4.3852  -1.166         0.24481
precalc                            1.7874     0.5684   3.145         0.00185 **
calc                               0.7071     0.3857   1.833         0.06787 .
gpa                                1.5636     0.2227   7.021 0.0000000000176 ***
calculus                           0.2445     0.1041   2.348         0.01958 *
english                           -0.3579     0.1288  -2.780         0.00582 **
langOther                          4.7348     2.1979   2.154         0.03210 *
sexMale                           -1.8113     1.7476  -1.036         0.30093
nationEastern European            -0.3363     2.9990  -0.112         0.91081
nationEuropean not Eastern         1.2190     2.3190   0.526         0.59954
nationMiddle-Eastern and Pakistani -1.0665    3.0529  -0.349         0.72710
nationEast Indian                  4.9021     2.8125   1.743         0.08247 .
nationOther and DK                 3.5497     5.5530   0.639         0.52320
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.57 on 272 degrees of freedom
  (292 observations deleted due to missingness)
Multiple R-squared:  0.483,      Adjusted R-squared:  0.4564
F-statistic: 18.15 on 14 and 272 DF,  p-value: < 0.00000000000000022

> red1 = update(model1, . ~ . - nation) # Take away nation
> anova(red1,model1) # Testing nation
Analysis of Variance Table

Model 1: mark ~ course + precalc + calc + gpa + calculus + english + lang +
    sex
Model 2: mark ~ course + precalc + calc + gpa + calculus + english + lang +
    sex + nation
  Res.Df   RSS Df Sum of Sq      F Pr(>F)
1    277 50965
2    272 50062  5    902.68 0.9809 0.4298
>
```

```
> model2 = red1; summary(model2)

Call:
lm(formula = mark ~ course + precalc + calc + gpa + calculus +
    english + lang + sex)

Residuals:
    Min      1Q  Median      3Q     Max
-45.637  -6.557   1.183   8.517  30.945

Coefficients:
                Estimate Std. Error t value         Pr(>|t|)
(Intercept)     -66.7948    11.5843  -5.766 0.00000002164014 ***
courseElite      -8.3597     5.0888  -1.643          0.10156
courseMainstream -5.6157     4.3396  -1.294          0.19672
precalc           1.7374     0.5657   3.071          0.00235 **
calc              0.7117     0.3789   1.878          0.06140 .
gpa               1.5717     0.2211   7.110 0.00000000000989 ***
calculus          0.2513     0.1033   2.433          0.01560 *
english          -0.3273     0.1264  -2.590          0.01011 *
langOther         4.4895     2.1334   2.104          0.03625 *
sexMale          -1.6280     1.7155  -0.949          0.34344
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.56 on 277 degrees of freedom
  (292 observations deleted due to missingness)
Multiple R-squared:  0.4737,     Adjusted R-squared:  0.4566
F-statistic:  27.7 on 9 and 277 DF,  p-value: < 0.00000000000000022

> # Test course
> red2 = update(model2, . ~ . - course); anova(red2,model2)
Analysis of Variance Table

Model 1: mark ~ precalc + calc + gpa + calculus + english + lang + sex
Model 2: mark ~ course + precalc + calc + gpa + calculus + english + lang +
    sex
  Res.Df   RSS Df Sum of Sq      F Pr(>F)
1    279 51464
2    277 50965  2    498.74 1.3553 0.2596
> summary(red2)

Call:
lm(formula = mark ~ precalc + calc + gpa + calculus + english +
    lang + sex)

Residuals:
    Min      1Q  Median      3Q     Max
-47.981  -6.620   0.892   8.775  30.625

Coefficients:
            Estimate Std. Error t value         Pr(>|t|)
(Intercept) -70.8000    11.2636  -6.286 0.00000000125232 ***
precalc       1.6828     0.5650   2.979          0.00315 **
calc          0.6579     0.3774   1.743          0.08245 .
gpa           1.5830     0.2212   7.155 0.00000000000738 ***
calculus      0.2251     0.1021   2.204          0.02837 *
english      -0.3271     0.1259  -2.598          0.00988 **
langOther     4.5356     2.1342   2.125          0.03445 *
sexMale      -1.7205     1.7022  -1.011          0.31302
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.58 on 279 degrees of freedom
  (292 observations deleted due to missingness)
Multiple R-squared:  0.4685,     Adjusted R-squared:  0.4552
F-statistic: 35.13 on 7 and 279 DF,  p-value: < 0.00000000000000022

>
> # I think I will drop sex, and combine the two sub-tests
>
```

```
> diagtest = precalc+calc # Diagnostic test score
>
> model3 = lm(mark ~ diagtest + gpa + calculus + english + lang)
> summary(model3)

Call:
lm(formula = mark ~ diagtest + gpa + calculus + english + lang)

Residuals:
    Min      1Q  Median      3Q     Max
-50.441  -6.719   1.546   9.036  32.288

Coefficients:
            Estimate Std. Error t value         Pr(>|t|)
(Intercept) -71.4517    11.1072  -6.433 0.00000000053717 ***
diagtest      0.9721     0.2518   3.860         0.000141 ***
gpa           1.5892     0.2209   7.195 0.00000000000569 ***
calculus      0.2176     0.1021   2.132         0.033908 *
english      -0.3002     0.1227  -2.447         0.015017 *
langOther     4.6966     2.1264   2.209         0.028003 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.6 on 281 degrees of freedom
  (292 observations deleted due to missingness)
Multiple R-squared:  0.4635,     Adjusted R-squared:  0.454
F-statistic: 48.56 on 5 and 281 DF,  p-value: < 0.00000000000000022


>
>
> # Multiple R-squared:  0.4635, Adjusted R-squared:  0.454
# F-statistic: 48.56 on 5 and 281 DF,  p-value: < 2.2e-16

# Compared to results for model0,

# Multiple R-squared:  0.4883,   Adjusted R-squared:  0.4519
# F-statistic: 13.41 on 19 and 267 DF,  p-value: < 2.2e-16

# So I like Model 3.
```

```
>
> # Stepwise search. Recall AIC = 2k - 2 ln(L-hat) = 2k + n ln(SSE) + c
>
> everything = lm(mark ~ course + precalc + calc + gpa + calculus + english +
+                        lang + sex + nation1 + nation2
+                        + nation + diagtest)
> nothing = lm(mark ~ 1) # Just an intercept
>
>
> backwards = step(everything)
Start:  AIC=1519.36
mark ~ course + precalc + calc + gpa + calculus + english + lang +
    sex + nation1 + nation2 + nation + diagtest


Step:  AIC=1519.36
mark ~ course + precalc + calc + gpa + calculus + english + lang +
    sex + nation1 + nation2 + nation

           Df Sum of Sq   RSS    AIC
- nation2   3     337.1 49022 1515.3
- nation1   5    1026.3 49711 1515.3
- course    2     510.0 49195 1518.3
- nation    3     859.0 49544 1518.4
- sex       1     265.7 48951 1518.9
<none>                  48685 1519.4
- calc      1     586.8 49272 1520.8
- calculus  1     636.3 49321 1521.1
- lang      1     834.3 49519 1522.2
- english   1    1888.7 50574 1528.3
- precalc   1    1985.7 50671 1528.8
- gpa       1    9730.8 58416 1569.7

Step:  AIC=1515.34
mark ~ course + precalc + calc + gpa + calculus + english + lang +
    sex + nation1 + nation

           Df Sum of Sq   RSS    AIC
- nation1   5    1040.3 50062 1511.4
- nation    5    1164.7 50187 1512.1
- course    2     510.7 49533 1514.3
- sex       1     264.4 49287 1514.9
<none>                  49022 1515.3
- calc      1     537.4 49559 1516.5
- calculus  1     775.3 49797 1517.8
- lang      1     838.0 49860 1518.2
- english   1    1810.0 50832 1523.7
- precalc   1    1976.9 50999 1524.7
- gpa       1    9536.6 58559 1564.3

Step:  AIC=1511.36
mark ~ course + precalc + calc + gpa + calculus + english + lang +
    sex + nation

           Df Sum of Sq   RSS    AIC
- nation    5     902.7 50965 1506.5
- course    2     371.2 50434 1509.5
- sex       1     197.7 50260 1510.5
<none>                  50062 1511.4
- calc      1     618.5 50681 1512.9
- lang      1     854.2 50917 1514.2
- calculus  1    1015.0 51077 1515.1
- english   1    1422.0 51484 1517.4
- precalc   1    1820.2 51883 1519.6
- gpa       1    9071.7 59134 1557.2
```

```
Step:  AIC=1506.49
mark ~ course + precalc + calc + gpa + calculus + english + lang +
    sex

           Df Sum of Sq   RSS    AIC
- course    2     498.7 51464 1505.3
- sex       1     165.7 51131 1505.4
<none>                  50965 1506.5
- calc      1     649.1 51614 1508.1
- lang      1     814.8 51780 1509.0
- calculus  1    1089.3 52054 1510.6
- english   1    1233.9 52199 1511.4
- precalc   1    1735.2 52700 1514.1
- gpa       1    9301.0 60266 1552.6

Step:  AIC=1505.29
mark ~ precalc + calc + gpa + calculus + english + lang + sex

           Df Sum of Sq   RSS    AIC
- sex       1     188.4 51652 1504.3
<none>                  51464 1505.3
- calc      1     560.3 52024 1506.4
- lang      1     833.1 52297 1507.9
- calculus  1     895.7 52360 1508.2
- english   1    1245.0 52709 1510.2
- precalc   1    1636.5 53100 1512.3
- gpa       1    9444.1 60908 1551.6

Step:  AIC=1504.34
mark ~ precalc + calc + gpa + calculus + english + lang

           Df Sum of Sq   RSS    AIC
<none>                  51652 1504.3
- calc      1     501.7 52154 1505.1
- calculus  1     894.9 52547 1507.3
- lang      1     919.2 52571 1507.4
- english   1    1090.1 52742 1508.3
- precalc   1    1516.6 53169 1510.6
- gpa       1    9341.7 60994 1550.0
> summary(backwards)

Call:
lm(formula = mark ~ precalc + calc + gpa + calculus + english +
    lang)

Residuals:
    Min      1Q  Median      3Q     Max
-48.714  -6.425   0.939   8.971  31.431

Coefficients:
            Estimate Std. Error t value         Pr(>|t|)
(Intercept) -72.5546    11.1295  -6.519 0.00000000032851 ***
precalc       1.6048     0.5597   2.867          0.00446 **
calc          0.6193     0.3755   1.649          0.10023
gpa           1.5728     0.2210   7.116 0.00000000000932 ***
calculus      0.2249     0.1021   2.202          0.02845 *
english      -0.2980     0.1226  -2.431          0.01569 *
langOther     4.7422     2.1244   2.232          0.02639 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.58 on 280 degrees of freedom
  (292 observations deleted due to missingness)
Multiple R-squared:  0.4666,     Adjusted R-squared:  0.4551
F-statistic: 40.82 on 6 and 280 DF,  p-value: < 0.00000000000000022

>
> # Recall model3 had diagtest, gpa, calculus, english, lang
> # R^2 = 0.4635
>
```

```
>
> forwards = step(nothing, scope=list(lower=formula(nothing),
+             upper=formula(everything)), direction="forward" ,trace=0)
Error in step(nothing, scope = list(lower = formula(nothing), upper = formula(everything)),
:
  number of rows in use has changed: remove missing values?
In addition: Warning message:
In add1.lm(fit, scope$add, scale = scale, trace = trace, k = k,  :
  using the 287/393 rows from a combined fit

>
> # To resolve error
> komplete = data.frame(mark, course, precalc, calc, gpa, calculus, english,
+                       lang, sex, nation1, nation2, nation, diagtest)
> dim(komplete)
[1] 579  13
> komplete = subset(komplete,complete.cases(komplete))
> dim(komplete)
[1] 287  13
>
> everything = lm(mark ~ course + precalc + calc + gpa + calculus + english +
+             lang + sex + nation1 + nation2 + nation + diagtest, data=komplete)
> nothing = lm(mark ~ 1, data=komplete) # Just an intercept
>
> forwards = step(nothing, scope=list(lower=formula(nothing),
+             upper=formula(everything)), direction="forward" ,trace=0)
> summary(forwards)

Call:
lm(formula = mark ~ gpa + diagtest + english + lang + calculus,
    data = komplete)

Residuals:
    Min      1Q  Median      3Q     Max
-50.441  -6.719   1.546   9.036  32.288

Coefficients:
            Estimate Std. Error t value          Pr(>|t|)
(Intercept) -71.4517    11.1072  -6.433 0.00000000053717 ***
gpa           1.5892     0.2209   7.195 0.00000000000569 ***
diagtest      0.9721     0.2518   3.860          0.000141 ***
english      -0.3002     0.1227  -2.447          0.015017 *
langOther     4.6966     2.1264   2.209          0.028003 *
calculus      0.2176     0.1021   2.132          0.033908 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.6 on 281 degrees of freedom
Multiple R-squared:  0.4635,     Adjusted R-squared:  0.454
F-statistic: 48.56 on 5 and 281 DF,  p-value: < 0.00000000000000022

>
> # Recall model3 had diagtest, gpa, calculus, english, lang
> # Forward selection yields exactly model3.
```

```
> # How good is the prediction? We will see, but try "predicting" the data we
> # already have. In this case the predict function uses only complete cases.
>
> dim(predict(model3, interval = 'prediction'))
[1] 287   3
Warning message:
In predict.lm(model3, interval = "prediction") :
  predictions on current data refer to _future_ responses

> dim(komplete)
[1] 287  13
> cbind(komplete$mark, predict(model3, interval = 'prediction') )
             fit       lwr       upr
1   39  44.57224 17.569969  71.57450
2   57  30.44263  3.212810  57.67244
3   62  59.51473 32.633582  86.39588
4   76  62.47987 35.498926  89.46081
5   86  69.17676 42.279227  96.07429
6   60  43.34073 16.222929  70.45853
7   54  68.84992 41.685103  96.01474
8   17  44.56362 17.577134  71.55011
9    2  46.91642 19.825418  74.00742
10  76  78.16973 51.159338 105.18012
14  84  72.22926 45.185332  99.27318
17  75  61.53430 34.330883  88.73771
18  94  80.03063 53.018299 107.04296
19  60  52.37792 25.488502  79.26733
21  53  59.35414 32.248976  86.45931
22  63  68.53619 41.607075  95.46530
23  82  59.75324 32.623607  86.88287
25  67  64.80705 37.806147  91.80795
26  55  48.83304 21.949716  75.71637
28  77  63.29094 36.323287  90.25858
29  80  59.81765 32.926385  86.70892
32  26  46.94482 19.969422  73.92023
33  81  58.87646 31.490655  86.26226
34  78  69.59901 42.629605  96.56842
36  17  50.48763 23.150395  77.82486
43  60  70.96918 43.650924  98.28744
45  53  65.49042 38.322411  92.65843
46  26  39.35656 12.395644  66.31748
49  65  55.08302 28.228580  81.93747
50  81 100.40727 72.468235 128.34630
51  51  62.07914 35.117398  89.04089

              . . .

560 57  47.21132 20.257529  74.16511
561 56  52.87046 26.027670  79.71325
562 87  84.22471 57.181869 111.26754
563 52  54.17758 27.290873  81.06429
564 86  69.13361 42.150925  96.11630
565 76  57.33906 30.433879  84.24423
566 90  78.23960 51.203744 105.27545
567 67  74.26427 47.061121 101.46743
568 57  65.14072 38.195430  92.08600
570 80  70.79105 43.775635  97.80647
571 56  52.62527 25.618039  79.63250
572 70  68.22469 41.285399  95.16397
574 60  56.68960 29.653398  83.72580
576 50  64.68653 37.638085  91.73497
577 50  50.49301 23.381642  77.60438
579 61  57.56981 30.671469  84.46816
Warning message:
In predict.lm(model3, interval = "prediction") :
  predictions on current data refer to _future_ responses
```

```
> ###################### A digression ######################
>
> # Bonferroni-corrected cross-validation
>
> # Recall the interesting results for HS English and first language.
> # In Bonferroni-corrected cross-validation, you use an exploratory sample
> # to develop a model and draw tentative conclusions, each supported by a
> # significant test statistic. Then using a replication sample, you test ONLY
> # those hypotheses, with a Bonferroni correction. If you are testing k
> # hypotheses, the Bonferroni correction holds down the probability of making
> # one or more Type I errors. Using a significance level of 0.05/k, the
> # probability of one or more Type I errors is less than or equal to 0.05.
> # Equivalently, compare k*p to 0.05.
>
> # Re-start R, read replication sample
> # Later, I will put this in a file called readmath2.txt
>
>
> rm(list=ls()); options(scipen=999)
> # Read the replication sample (the "test" data)
> test = read.table("http://www.utstat.toronto.edu/~brunner/data/legal/replicmath.data.txt")
> colnames(test) = c('id', 'course', 'precalc', 'calc', 'gpa', 'calculus',
+                'english', 'mark', 'lang', 'sex', 'nation1', 'nation2', 'sample')
> # Variables 3 through 8 are numeric
> # The following will make . = NA
> for(j in 3:8) test[,j] = as.numeric(as.character(test[,j]))
Warning messages:
1: NAs introduced by coercion
2: NAs introduced by coercion
3: NAs introduced by coercion
4: NAs introduced by coercion
5: NAs introduced by coercion
6: NAs introduced by coercion
> # Fix up the variables outside the data frame to save typing.
> attach(test)
> id = 1:dim(test)[1] # Because ids start at 580
>
> # Fix up course
> course = as.character(course) # I need to make it character to edit
> course[course=='.'] = NA
> course[course=='4'] = NA
> course[course=='1'] = 'Catch-up'
> course[course=='2'] = 'Mainstream'
> course[course=='3'] = 'Elite'
> course=factor(course)
> # Fix up gpa
> bad = subset(id,gpa==0)
> gpa[bad] = NA
> # Fix up calculus
> bad = subset(id,calculus==0 | calculus == 999) # Logical or
> calculus[bad] = NA
> # Fix up english
> bad = subset(id,english==0); english[bad] = NA
> # Fix up mark
> bad = subset(id,mark==0 | mark > 900); mark[bad] = NA
> # Fix up lang
> lang = as.character(lang) # For editing
> dot = subset(id,lang=='.') ; lang[dot] = NA
> franc = subset(id,lang=='French'); lang[franc] = 'Other'
> lang = factor(lang)
> # Fix up sex
> dot = subset(id,sex=='.') ; sex[dot] = NA
> sex = factor(sex)
> # Did NOT fix up nationality ratings -- still 18 or 19 categories
> test2 = data.frame(id, course, precalc, calc, gpa, calculus, english, mark,
+         lang, sex, nation1, nation2, sample)
> id = test$id # Put it back
> diagtest = precalc+calc
```

```
>
> model3 = lm(mark ~ diagtest + gpa + calculus + english + lang)
> summary(model3)

Call:
lm(formula = mark ~ diagtest + gpa + calculus + english + lang)

Residuals:
    Min      1Q  Median      3Q     Max
-61.484  -6.890   2.794   8.889  35.156

Coefficients:
             Estimate Std. Error t value     Pr(>|t|)
(Intercept) -58.93290   11.99704  -4.912 0.0000015277 ***
diagtest      1.58048    0.26400   5.987 0.0000000065 ***
gpa           1.15692    0.25532   4.531 0.0000086692 ***
calculus      0.17755    0.10404   1.707        0.089 .
english      -0.04691    0.13560  -0.346        0.730
langOther     1.34224    1.96918   0.682        0.496
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.09 on 282 degrees of freedom
  (291 observations deleted due to missingness)
Multiple R-squared:  0.3925,     Adjusted R-squared:  0.3818
F-statistic: 36.45 on 5 and 282 DF,  p-value: < 0.00000000000000022

> 0.05/5 # Compare p-value to 0.01
[1] 0.01
>
> ######################## End digression ########################
>
```

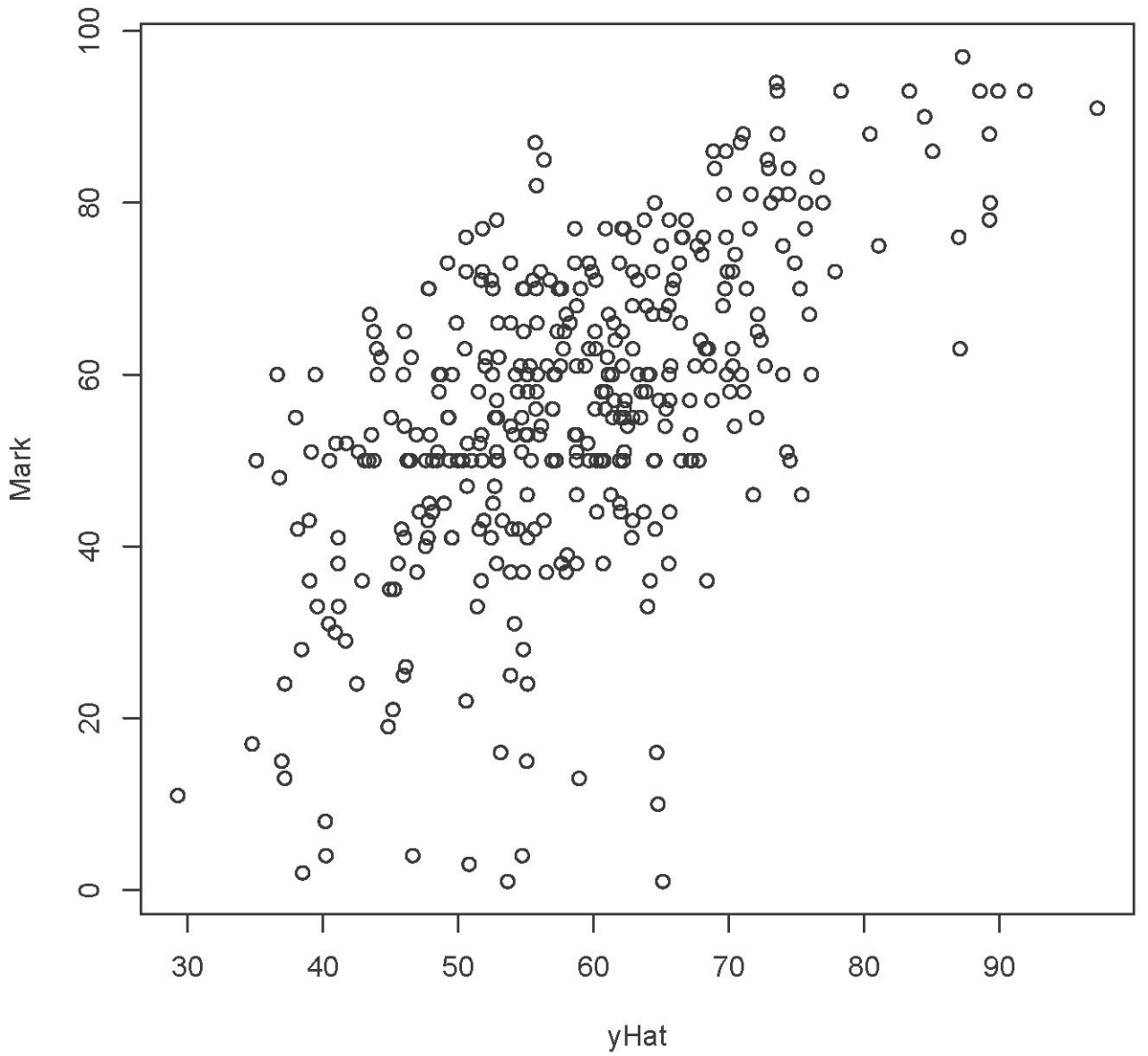
```
> ######################## Prediction ########################
>
> # Want to see how well model3 does at predicting mark for the replication
> # sample. For each student in the replication sample, the prediction will
> # be based only on explanatory variable values that are not missing. The
> # easy way is to fit a separate regression model for each case in the file.
> # It's not automatic. Here is a quick example to illustrate the code.
>
> source("readmath1.txt") # File must be in working directory
Warning messages:
1: NAs introduced by coercion
2: NAs introduced by coercion
3: NAs introduced by coercion
4: NAs introduced by coercion
5: NAs introduced by coercion
6: NAs introduced by coercion
>
> explanvars = colnames(datta)[-1]; explanvars # [-1] is to omit first element, mark
[1] "diagtest" "gpa"      "calculus" "english"  "lang"
>
> # Case 13 has one NA -- use for testing
> datta[13,]
   mark diagtest  gpa calculus english    lang
13   54       NA 87.2       84      83 English
>
> x = datta[13,][-1]; x # Omit mark, the first element
   diagtest  gpa calculus english    lang
13       NA 87.2       84      83 English
> good = explanvars[!is.na(x)]; good # Names of non-missing variables
[1] "gpa"      "calculus" "english"  "lang"
>
> ngood = length(good)
> call = 'mark ~ 1' # Start with just an intercept
> if(ngood>0) # If any explanatory vars are non-missing
+     { for(j in 1:ngood) call = paste(call,good[j],sep=" + ") }
>
> call
[1] "mark ~ 1 + gpa + calculus + english + lang"
>
> mod = lm(call, data=datta)
> predict.lm(mod,newdata = data.frame(x), interval='prediction')
       fit      lwr      upr
13 69.92754 41.75264 98.10244
>
```

```
> ######################################################
> # Prediction of the replication data
>
> # setwd("/Users/brunner/Dropbox/2018-19Teaching/2101f18/2101f18Lectures/24-
2101f18MathR")
>
> source("readmath1.txt") # Defines data frame datta
Warning messages:
1: NAs introduced by coercion
2: NAs introduced by coercion
3: NAs introduced by coercion
4: NAs introduced by coercion
5: NAs introduced by coercion
6: NAs introduced by coercion
> source("readmath2.txt") # Defines data frame replic
The following objects are masked from training:

    calc, calculus, course, english, gpa, id, lang, mark, nation1, nation2,
precalc,
    sample, sex

Warning messages:
1: NAs introduced by coercion
2: NAs introduced by coercion
3: NAs introduced by coercion
4: NAs introduced by coercion
5: NAs introduced by coercion
6: NAs introduced by coercion
>
>
> # Loop through the replication sample, fitting a model and generating a
> # prediction interval for each case.
> explanvars = colnames(datta)[-1] # Names of explanatory variables (omit mark, the
first one)
> n = dim(replic)[1]
> pred = data.frame(NULL)
> for(j in 1:n)
+     {
+     x = replic[j,][-1] # Explanatory variable values for case j
+     good = explanvars[!is.na(x)] # Names of non-missing variables
+     ngood = length(good)
+     call = 'mark ~ 1' # Start with just an intercept
+     if(ngood>0) # If any explanatory vars are non-missing
+         { for(j in 1:ngood) call = paste(call,good[j],sep=" + ") }
+     # cat(call,"\n") # For de-bugging
+     mod = lm(call, data=datta)
+     pinterval = predict.lm(mod,newdata= data.frame(x), interval='prediction')
+     pred = rbind(pred, pinterval)
+     } # Next case in the test data
> colnames(pred)[1] = "yHat" # I like this better than "fit"
>
> pred = cbind(replic$mark,pred)
> colnames(pred)[1] = "Mark"
> head(pred)
  Mark     yHat      lwr      upr
1   56 56.99021 29.74338  84.23703
2   97 87.26050 58.81423 115.70678
3   NA 58.29059 24.31047  92.27071
4   NA 45.83531 17.55756  74.11306
5   50 67.24679 40.36763  94.12595
6   67 57.96861 30.90664  85.03057
> attach(pred)
>
```

```
> # How good (or bad) is the fit?
> meanabserr = mean(abs(yHat-Mark),na.rm=TRUE); meanabserr
[1] 11.43447
> Rsq = cor(yHat,Mark, use="complete.obs")^2; Rsq
[1] 0.3392308
> plot(yHat,Mark)
```




```
>
> # How good is the prediction interval?
> inside = (lwr < Mark) & (Mark < upr) # A logical vector
> inside[1:10]
 [1]  TRUE  TRUE    NA    NA  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
> mean(inside,na.rm=TRUE)
[1] 0.95
> # Almost too good to be true.
>
```