## Section 2.1 A Familiar Problem

To show the |STAT style of interactive data analysis, I will work through a concrete example. The example is based on a familiar problem: grades in a course based on two midterm exams and a final exam. Scores on exams will be broken down by student gender (male or female) and by the lab section taught by one of two teaching assistants: John or Jane. Assume the following data are in the file **exam.dat**. Each line in the file includes a student identification number, the student's section's teaching assistant, the student's gender, and the scores (out of 100) on the two midterm exams and the final.

```
S-1     john    male    56      42      58
S-2     john    male    96      90      91
S-3     john    male    70      59      65
S-4     john    male    82      75      78
S-5     john    male    85      90      92
S-6     john    male    69      60      65
S-7     john    female  82      78      60
S-8     john    female  84      81      82
S-9     john    female  89      80      68
S-10    john    female  90      93      91
S-11    jane    male    42      46      65
S-12    jane    male    28      15      34
S-13    jane    male    49      68      75
S-14    jane    male    36      30      48
S-15    jane    male    58      58      62
S-16    jane    male    72      70      84
S-17    jane    female  65      61      70
S-18    jane    female  68      75      71
S-19    jane    female  62      50      55
S-20    jane    female  71      72      87
```

We are interested in computing final grades based on the exam scores, and comparing the performances of males versus females, and of the different teaching assistants. The following analyses can be tried by typing in the above file and running the commands in the examples. Minor variations on the example commands will help show how the programs work.

## Section 2.2 Computing Final Scores

Computing final scores is easy with the data manipulation program **dm**. Assume that the first midterm is worth 20 percent, the second 30 percent, and the final exam, 50 percent. The following command tells **dm** to repeat each input line with **INPUT**, and then print the weighted sum of columns 4, 5, and 6, treated as *numbers*. To print numbers, **dm** uses an **x** before the column number. The input to **dm** is read from the file **exam.dat** and the result is saved in the file **scores.dat**. Once all the original data and the final scores are in **scores.dat**, only that file will be used in following analyses.

```
dm INPUT ".2*x4 + .3*x5 + .5*x6" < exam.dat > scores.dat
```

The standard input is redirected from the file **exam.dat** with the **<** on the command line. Similarly, the standard output, which would ordinarily go to the screen, is redirected to the file **scores.dat** with the **>** on the command line. The second expression for **dm** is in quotes. This allows the insertion of spaces to make the expression more readable, and to make sure that any special characters (e.g., ***** is special to UNIX shells) are hidden from the command line interpreter. The output from the above command, saved in the file **scores.dat**, would begin with the following.

```
S-1     john    male    56      42      58      52.8
S-2     john    male    96      90      91      91.7
S-3     john    male    70      59      65      64.2
S-4     john    male    82      75      78      77.9
S-5     john    male    85      90      92      90
S-6     john    male    69      60      65      64.3
```
*etc.*

This could be sorted by final grade by reversing the columns and sending the output to the standard UNIX or MSDOS **sort** utility program using the ‘‘pipe’’ symbol |.

```
reverse -f < scores.dat | sort
```

The above command would produce the following output.

```
27.1    34      15      28      male    jane    S-12
40.2    48      30      36      male    jane    S-14
52.8    58      42      56      male    john    S-1
54.7    65      46      42      male    jane    S-11
54.9    55      50      62      female  jane    S-19
 ...
79.3    87      72      71      female  jane    S-20
82.1    82      81      84      female  john    S-8
90      92      90      85      male    john    S-5
91.4    91      93      90      female  john    S-10
91.7    91      90      96      male    john    S-2
```

To restore the order of the fields, **reverse** could be called again. Another way, more efficient, would be to use the **dsort** filter to sort based on column 7:

```
dsort 7 < scores.dat
```

## Section 2.3 Summary of Final Scores

**desc** prints summary statistics, histograms, and frequency tables. The following command takes the final scores (the weighted average from the previous section).

```
dm  s7  <  scores.dat
```

Summary order statistics are printed with the **-o** option and the distribution is tested against the passing grade of 75 with the **-t 75** option. **desc** makes a histogram (the **-h** option) with 10 point intervals (the **-i 10** option) starting at a minimum value of 0 (the **-m 0** option).

```
dm  s7  <  scores.dat | desc  -o  -t 75  -h  -i 10  -m 0

------------------------------------------------------------
 Under Range    In Range  Over Range     Missing         Sum
           0          20           0           0    1359.200
------------------------------------------------------------
        Mean      Median    Midpoint   Geometric    Harmonic
      67.960      68.750      59.400      65.564      62.529
------------------------------------------------------------
          SD   Quart Dev       Range     SE mean
      16.707      10.575      64.600       3.736
------------------------------------------------------------
     Minimum  Quartile 1  Quartile 2  Quartile 3     Maximum
      27.100      57.450      68.750      78.600      91.700
------------------------------------------------------------
        Skew     SD Skew    Kurtosis     SD Kurt
      -0.586       0.548       2.844       1.095
------------------------------------------------------------
   Null Mean           t    prob (t)           F    prob (F)
      75.000      -1.884       0.075       3.551       0.075
------------------------------------------------------------
       Midpt    Freq
       5.000       0
      15.000       0
      25.000       1 *
      35.000       0
      45.000       1 *
      55.000       4 ****
      65.000       5 *****
      75.000       5 *****
      85.000       2 **
      95.000       2 **
```

## Section 2.4 Predicting Final Exam Scores

The next analysis predicts final exam scores with those of the two midterm exams. The **regress** program assumes its input has the predicted variable in column 1 and the predictors in following columns. **dm** can extract the columns in the correct order from the file **scores.dat**. The command for **dm** looks like this.

```
dm x6 x4 x5 < scores.dat
```

The output from **dm** looks like this.

```
58    56    42
91    96    90
65    70    59
78    82    75
92    85    90
65    69    60
60    82    78
```

*etc.*

This is the correct format for input for **regress**, which is given mnemonic names for the columns. The **-e** option tells **regress** to save the regression equation in the file **regress.eqn** for a later analysis.

```
dm x6 x4 x5 < scores.dat | regress -e final midterm1 midterm2
```

The output from **regress** includes summary statistics for all the variables, a correlation matrix (e.g., the correlation of **midterm1** and **midterm2** is .9190), the regression equation relating the predicted variable, and the significance test of the multiple correlation coefficient. The squared multiple correlation coefficient of 0.7996 shows a strong relationship between midterm exams and the final.

```
Analysis for 20 cases of 3 variables:
Variable         final   midterm1   midterm2
Min            34.0000    28.0000    15.0000
Max            92.0000    96.0000    93.0000
Sum          1401.0000  1354.0000  1293.0000
Mean           70.0500    67.7000    64.6500
SD             15.3502    18.6720    20.4303

Correlation Matrix:
final           1.0000
midterm1        0.7586     1.0000
midterm2        0.8838     0.9190     1.0000
Variable         final   midterm1   midterm2

Regression Equation for final:
final  =  -0.2835 midterm1  +  0.9022 midterm2  +  30.9177

Significance test for prediction of final
    Mult-R  R-Squared      SEest    F(2,17)   prob (F)
    0.8942     0.7996     7.2640    33.9228     0.0000
```

## Predicted Plot

We can look at the predictions from the regression analysis. From the analysis above, the file **regress.eqn** contains a regression equation for **dm**.

```
s1
(x2 * -0.283512...) + (x3 * 0.902182...) + 30.9177...
```

Extra precision is used in **regress.eqn**, compared to the equation in the output from **regress** to allow more accurate calculations. These two expressions, one on each line, are the obtained and predicted final exam scores, respectively. To plot these against each other, we duplicate the input used to **`regress`**, and process **`regress`**'s output with **`dm`**, reading its expressions from the expression file **`regress.eqn`** that follows the letter **`E`**. The result is passed through a pipe to the paired data analysis program **`pair`** with the plotting option **`-p`**, options to control the height and width of the plot, the **`-h`** and **`-w`** options, and **`-x`** and **`-y`** options to label the plot.

```
dm x6 x4 x5 < scores.dat | dm Eregress.eqn |
      pair -p -h 10 -w 30 -x final -y predicted
```

```
|------------------------------|89.3045
|                             3|
|                   1    1     |
|             1   1   11  1 1  |
|                              |
|              1 2 1           |predicted
|          1     1             |
|            1                 |
|       1                      |
|                              |
|1                             |
|------------------------------|36.5121
34.000                    92.000
        final  r= 0.894
```

## Residual Plot

To plot the residuals (deviations) from prediction, you can run the data through another pass of **`dm`** to subtract the predicted scores from the obtained. Note that **`r`** must be zero.

```
dm x6 x4 x5 < scores.dat | dm Eregress.eqn | dm x2 x1-x2 |
      pair -p -h 10 -w 30 -x predicted -y residuals
```

```
|------------------------------|11.2546
|                     11       |
|                           1  |
|         1   1   1    1      1|
|      1        1        1    1|
|1               1      1      |residuals
|            1    1            |
|                        1     |
|                       1      |
|                              |
|                       1      |
|------------------------------|-18.0399
36.512                    89.304
      predicted  r= 0.000
```

# Section 2.5 Failures by Assistant and Gender

Now suppose the passing grade in the course is 75. To see how many people of each sex in the two sections passed, we can use the **contab** program to print contingency tables. First **dm** extracts the columns containing teaching assistant, gender, and the final grade (the weighted average computed earlier). Rather than include the final grade, a label indicating pass or fail is added, as appropriate.

```
dm  s2  s3  "if x7 >= 75 then 'pass' else 'fail'"  1  <  scores.dat
```

The huge third expression says ''if the final grade is greater than or equal to 75, then insert the string **pass**, else insert the string **fail**.'' Such expressions can be placed in files rather than be typed on the command line, and usually **dm** is used for simpler expressions. The fourth expression is the constant **1** used to tell **contab** that there was one replication for each combination of factor levels. Part of the output from **dm** follows.

```
john    male    fail    1
john    male    pass    1
john    male    fail    1
    ...
jane    female  fail    1
jane    female  fail    1
jane    female  pass    1
```

This is used as input to **contab**, which is given mnemonic factor names.

```
dm  s2  s3  "if x7 >= 75 then 'pass' else 'fail'"  1  <  scores.dat |
      contab assistant gender success count
```

Parts of the output from this command follow. First, there is a summary of the input, which contained three factors, each with 2 levels, and a sum of observation counts.

```
FACTOR:  assistant     gender    success      count
LEVELS:          2          2          2         20
```

The first contingency table does not provide new information. It shows that both Jane's section and John's section had 6 male and 4 female students.

```
SOURCE: assistant gender
             male  female  Totals
john            6       4      10
jane            6       4      10
Totals         12       8      20
```

The second contingency table tells us that 12 of 20 students failed the course--4 in John's section and 8 in Jane's. A significance test follows, and the warning about small expected frequencies suggests that the chi-square test for independence might be invalid. No matter, the Fisher exact test applies because we are dealing with a 2x2 table and total frequencies less than 100. It does not show a significant association of factors (ie. Jane's section did not do significantly better than John's).

```
SOURCE: assistant success
             fail    pass  Totals
john            4       6      10
jane            8       2      10
Totals         12       8      20
```

```
Analysis for assistant x success:
NOTE: Yates' correction for continuity applied
WARNING: 2 of 4 cells had expected frequencies < 5
chisq       1.875000    df   1    p  0.170904
Fisher Exact One-Tailed Probability  0.084901
Fisher Exact Two-Tailed Probability  0.169802
phi Coefficient == Cramer's V        0.306186
Contingency Coefficient              0.292770
```

The third contingency table shows that 8 male students and 4 female students failed the course.

```
SOURCE: gender success
            fail    pass  Totals
male           8       4      12
female         4       4       8
Totals        12       8      20
```

The final table, the three-way interaction, shows all the effects listed above, but no significance test is computed by **contab**. Some hints about the reason for the poorer performance of Jane's section follow from the next section's analysis of variance.

```
SOURCE: assistant gender success
assistan  gender success
    john    male    fail       3
    john    male    pass       3
    john  female    fail       1
    john  female    pass       3
    jane    male    fail       5
    jane    male    pass       1
    jane  female    fail       3
    jane  female    pass       1
```

## Section 2.6 Effects of Assistant and Gender

We now want to compare the performance of the two teaching assistants and of male versus female students. We are interested to see how an assistant's students progress through the term. **anova**, the analysis of variance program, is the program to analyze these data, but we have to get the data into the correct format for input to **anova**. **anova** assumes that there is only one datum per line, preceded by the levels of factors under which it was obtained. This is unlike the format of **scores.dat**, which has the three exam scores after the student number, teaching assistant name, and gender. Several transformations are needed to get the data in the correct format. As an example, the data for student 1:

```
S-1      john     male     56       42       58
```

must be transformed to:

```
S-1      john     male     m1       56
S-1      john     male     m2       42
S-1      john     male     final    58
```

This is made up of three replications of the labels with new labels, **m1**, **m2**, and **final**, for the exams inserted. First, **dm** extracts and inserts the desired information. The result is a 15 column output, one for each expression. Note that on UNIX, it is necessary to quote the quotes of the labels for the exam names. To insert the newlines, so that each datum is on one line, the program **maketrix** reformats the input to **anova** into 5 columns. Finally, mnemonic labels for factor names are given to **anova**.

```
dm  s1  s2  s3  "'m1'"     s4 ...
    s1  s2  s3  "'m2'"     s5 ...
    s1  s2  s3  "'final'"  s6 < scores.dat |
    maketrix 5 | anova student assistant gender exam score
```

Only parts of the output are shown below. First, John's students did better than Jane's students (F(1,16)=8.311, p=.011).

```
john       76.7000
jane       58.2333
```

Female students scored better than males, although the effect is not statistically significant (F(1,16)=3.102, p=.097).

```
male       62.8611
female     74.3750
```

There was no interaction between these two factors (F(1,16)=.289), but there were some interactions between section assistant and gender and the different exam grades. If we look at the interaction of section assistant and exam, we get a better picture of the performances of John and Jane.

```
SOURCE: assistant exam
assista exam        N       MEAN         SD         SE
john    m1         10    80.3000    11.9355     3.7743
john    m2         10    74.8000    16.3761     5.1786
john    final      10    75.0000    13.4247     4.2453
jane    m1         10    55.1000    15.5167     4.9068
jane    m2         10    54.5000    19.5973     6.1972
jane    final      10    65.1000    16.2101     5.1261
```

This is the first full cell-means table shown. It contains the names of factors and levels, cell counts, means, standard deviations, and standard errors. The results show that John's students started higher than Jane's (80.3 versus 55.1), and that over the term, Jane's students improved while John's got worse. The significance test for the interaction looks like this.

```
SOURCE           SS    df          MS        F       p
=====================================================
ae         610.4333     2    305.2167    9.502   0.001 ***
es/ag     1027.8889    32     32.1215
```

A Scheffe confidence interval around the difference between two means of this interaction can be found with the following formula.

```
sqrt (df1 * critf * MSerror * 2 / N)
```

**df1** is the degrees of freedom numerator, **critf** is the critical F-ratio given the degrees of freedom and confidence level desired, **MSerror** is the mean-square error for the overall F-test, and **N** is the number of scores going into each cell. The critical F ratio for a 95% confidence interval based on 2 and 32 degrees of freedom can be found with the **probdist** program.

```
probdist  crit  F  2  32  .05
3.294537
```

Then, the calculator program **calc** can be used interactively to substitute the values.

```
CALC: sqrt (2 * 3.294537 * 32.1215 * 2 / 10)
sqrt((((2 * 3.29454) * 32.1215) * 2) / 10)) = 6.50617
```

Any difference of two means in this interaction greater than 6.5 is significant at the .05 level.

There was a similar pattern of males versus females on the three exams. Males started out lower than females, and males improved slightly while females stayed about the same.

```
SOURCE: gender exam
gender  exam        N       MEAN         SD          SE
male    m1         12    61.9167    20.7822      5.9993
male    m2         12    58.5833    22.5931      6.5221
male    final      12    68.0833    17.1329      4.9459
female  m1          8    76.3750    11.1475      3.9413
female  m2          8    73.7500    13.1557      4.6512
female  final       8    73.0000    12.7167      4.4960
```

After the cell means in the output from **anova** is a summary of the design, followed by an F-table, parts of which were seen above.

```
FACTOR:     student  assistant      gender        exam       score
LEVELS:          20          2           2           3          60
TYPE  :      RANDOM    BETWEEN     BETWEEN      WITHIN        DATA
```

The results of the analysis show that John's section did better than Jane's. That must be qualified because it seems that Jane's students may not have been as good as John's. To Jane's credit, her students improved more than John's during the term.

# CHAPTER 2

# Annotated Example



A concrete example with several |STAT programs is worked in detail. The example shows the style of analysis in |STAT. New users of |STAT should not try to understand all the details in the examples. Details about all the programs can be found in on-line manual entries and more examples of program use appear in following chapters. Explanations about features common to all |STAT programs can be found in the next chapter.