ABSTRACT

Personnel in the registrar's office at Syracuse University needed a method of informing the schools and colleges within the University as to the demand for courses during the pre-registration process. This paper will show how the SAS® System was used to project final course enrollments so that academic units on campus could make informed decisions regarding the allocation of resources for additional sections of courses in high demand.

This session will highlight: 1) efficient methods of accessing variable logical record length data files, 2) the powers and flexibilities of the SAS DATA step, and 3) the creative use of SAS procedures to aid in calculations. The paper will also touch upon how the results led to the administration's realization of the need for a more complex Demand Tracking System (DTS).

THE NEED

During the pre-registration period of each semester, academic units at Syracuse University receive reports containing enrollment totals for each section of each course offered by that unit. The reports tend to be very long for the larger schools and colleges at the university. Because of this, academic unit administrators have not been able to make concise determinations of the demand for courses. The Registrar's office requested a system be developed that would project final enrollment for courses based upon the relationship between the percentage of students who had participated in the pre-registration process, and the total course enrollment on any given day during pre-registration.

THE FILE STRUCTURE

The data required to make enrollment projections was stored in two systems, the Future Semester Registration system (FSR) and the Course Maintenance Scheduling system (CMS). The FSR consists of data records, one for each student, which are comprised of a biographical information segment, and up to twenty three course segments (figure 1).

Future Semester Registration Record (FSR)

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Academic Class</th>
<th>College</th>
<th># Credits Earned</th>
<th># Sections Registered</th>
</tr>
</thead>
</table>

Course Segments (Up to 23 Occurrences)

| Reference Number | Section Credits | Course Title | Reference Number | Section Credits | Course Title | Reference Number | Section Credits | Course Title |

The CMS system tracks enrollment in courses. Each section of every course offered at Syracuse University has a record in CMS. The data elements needed from this system were: the college offering the course, the department within the college, the course and section numbers, the maximum allowed enrollment, and the total number of students registered for the particular section of the course (figure 2).

Both files are composed of variable logical record length data elements. In the case of the FSR, the record length is dependent on the number of courses for which the student is registered. By first reading the variable containing the number of courses the student is currently enrolled, and maintaining a pointer variable, it is possible to span the record.
length without reading past the end-of-line (figure 3). Likewise, the data records of the CMS allow space for up to 900 student Social Security Numbers.

**figure 3**

```
DATA FSR (KEEP=SSN NAME ACCLASS ...);
INFILE IN1;

------------------------------;
* SEGSIZE=THE LENGTH OF AN INDIVIDUAL ;
* COURSE SEGMENT;
------------------------------;
SEGSIZE=400;
------------------------------;
* READ THE BIOGRAPHICAL SEGMENT OF ;
* THE RECORD;
------------------------------;

INPUT @1  SSN  $9.
@10 NAME  $25.
@35 ACCLASS  $1.
@35 COLLEGE  $2.
... ...
@253 CREDCAR  3.1
@256 SECTIONS  2.8;

------------------------------;
* PTR=RECORD COLUMN POINTER;
------------------------------;
PTR=1;
------------------------------;
* READ ALL VALID COURSE SEGMENTS;
------------------------------;

IF SECTIONS > 1 THEN DO;
  INPUT @PTR REPNUM1  $5.
  SECCR1  3.1
  COURSE1  $8.
  ... ...
  0;
END;

IF SECTIONS > 2 THEN DO;
  PTR=PTR+SEGSIZE;
  INPUT @PTR REPNUM2  $5.
  SECCR2  3.1
  COURSE2  $8.
  ... ...
  0;
END;
RUN;
```

By defining the 900 variables with a LENGTH statement and creating an explicitly subscripted array, it is possible, once the enrollment variable is read, to fill the elements of the array by reading the data within a DO loop. Obviously, there is little need to maintain all of these variables so the data step was made as efficient as possible by using techniques such as subsetting IF statements, the (KEEP= ...) option, and column oriented input as described by Howard (1988).

**USING THE SAS SYSTEM**

SAS Institute products are not frequently used as productional software tools in administrative computing at Syracuse. However, given the time constraints, and the complexity of the enrollment projection problem, a decision was made to use the SAS System. The Department of Testing Services has been able to use the experience gained in academic computing, to benefit administrators needing fast turn-around decision support processing.

**INPUT**

As discussed earlier, the input consisted of two data files, FSR and CMS. From the FSR system, we needed to keep all undergraduates students eligible to register. From the CMS system, we wanted only undergraduate level courses that carried at least 1 credit (enrollments in 0 credit labs and recitation sections were not projected).

**PROCESSING**

The Registrar's main concern was having enough seats available for underclassmen when they pre-registered. The program separated students into six different categories.

- underclassmen (including freshmen, sophomores, and juniors) registered,
- underclassmen not registered,
- seniors registered,
- seniors not registered,
- other students which were eligible to register and did so,
- others that did not register.
"Other" students included non-main campus students taking a main campus course, graduate students taking undergraduate courses, and any student who did not fall into the underclassmen or senior categories. Once these groupings were determined, the number of registered and non-registered underclassmen were calculated with PROC SUMMARY. These values were then attached to each observation of the initial data set (containing the FSR information) (figure 4). This was required for formulating the projection ratio later in the process.

**figure 4**

```plaintext
PROC SUMMARY DATA=FSR NWAY:
  VAR UNCLSR UNCLSNR;
  OUTPUT OUT=RTOTAL SUM=ISREGTOT NOREGTOT;
RUN;
```

The next step was to expand the "Student" data set into a "Course" data set. For example, a student registered for five courses, had five course segments in the FSR system. The resulting expanded data set would contain five observations for that student, each containing information about one course (figure 5).

**figure 5**

```plaintext
DATA EXPAND (KEEP= ... );
  SET NEWFSR;
  ARRAY CRSE[23] COURSE1 - COURSE23;
  I=1;
  DO WHILE (I <= SECTIONS):
    COURSE=CRSE[I];
    OUTPUT;
    I=I+1;
  END;
RUN;
```

It was convenient during this data step to compute the percentage of students who had gone through the pre-registration process, storing it in a macro variable for use in calculations and titles. Once we had a "Course" data set, it was easy to use PROC SUMMARY, again, to cipher total enrollments of each course for the three categories of registered students. Computing maximum allowed course enrollments followed in the process. The CMS system was input and put through PROC SUMMARY. The resulting data set was then merged with summarized FSR data (by course). It was then possible to calculate the enrollment projection.

The formula used to project the final enrollments was:

\[
\frac{\text{# of Underclassmen Registered For Course}}{.01 \times \text{(Percentage Of Underclassmen Registered)}} + \text{Seniors} + \text{Others}
\]

As can be seen by the ratio, as the denominator tends toward one, the projection becomes the actual underclassmen enrollment. The seniors, who all pre-register before the underclassmen, are then added, as are the few odd cases in the "other" category. Following this, the difference between projected and maximum enrollment was calculated. It was this difference the administrators examined and based resource allocation decisions.

**OUTPUT**

The newly created data set containing the course enrollments was sorted by academic unit. PROC PRINT was then used to produce a report displaying, the courses offered by the unit, the enrollment totals of underclassmen, seniors, and "others", the projected final enrollment, maximum allowed enrollment and the difference between projected and maximum enrollments. Other reports were also produced to show enrollment patterns. One contained listings of closed courses by academic unit (this report emphasized were enrollment potential was not met, for it listed courses which closed and the number of days into pre-registration when they closed). The enrollment projection at 29% registration, and the maximum allowed enrollment were also printed on this report.

**CONCLUSIONS**

The effectiveness of the projection model was exemplified by the results. Seven hundred twenty-nine seats were added to eleven different courses. Four hundred ninety-five of those seats were filled during the remaining days of pre-registration. That represents sixty-eight percent of those added. The average difference between the projected enrollment on
the second day of pre-registration and the actual enrollment at the end of the last day of the process, was -6. Meaning that, on the average, the model is accurate to six seats less than the actual final enrollment very early in the pre-registration process. The Demand Tracking System (DTS) at Syracuse University was also an outcome of the project.

The Demand Tracking System is a "rule" based registration simulation. Currently, only implemented for incoming freshmen registration, the DTS, batch simulates registration based upon departmental curriculum requirements, students' course requests, and course availability. The object being, run the system, produce the "simulated" registration and distribute reports to the academic units on campus so that they can make resource allocation decisions. Once any changes to the number of seats in course sections have been entered into system, the DTS run again, this time actually scheduling the students for the course and section they requested. My office was not involved in the planning or implementation of this system, so further information should be requested from Syracuse University's, Office of the Registrar. It has been presented here to emphasize the usefulness of developing relatively simple systems to demonstrate the need for complex systems.

REFERENCES


SAS is a Registered Trademark of SAS Institute Inc., Cary, NC