Predicting Student Enrollment Using Markov Chain Modeling in SAS

Concurrent Session    Thursday, May 30th    3:15pm – 4:00pm

Samantha Bradley, M.A. Applied Economics
Office of Institutional Research
University of North Carolina at Greensboro
Public, coeducational state university founded in 1891

20,106 students enrolled in Fall 2018

IR aggregates, analyzes, and disseminates data in support of:

- Institutional Planning
- Policy formulation
- Decision-making for internal/external constituents
Why Enrollment Projections?

◆ IR prepares Enrollment Projections every year
  ◆ Headcounts by student level
  ◆ Student credit hours by cost category

◆ Used by UNC System Office during decision-making about university funding
◆ Helps the university plan resource allocation
◆ Identify areas with growth potential
IR maintains SAS datasets of enrollment going back to Fall 2004

- 150+ variables:
  - Demographics
  - Areas of study
  - Degree programs
  - Credit hours

How can we leverage all this data to create the most accurate Enrollment Projections?
Markov Chain Model

- Lets us estimate the movements of a population over time

- The population must be categorized into exhaustive, mutually exclusive groups or ‘states’
  - ex.) Freshman, Sophomore, Junior, Senior

- Estimates the probability of moving from one state to another, or remaining in the same state
  - Probabilities are arranged to create a NxN Transition Probability Matrix
  - N is the number of unique states in the model
To predict enrollment for next semester, a simple Markov Chain Model looks like this:

Number of students we have this semester in each state at time $t$  

<table>
<thead>
<tr>
<th>$F_t$</th>
<th>$P_t$</th>
<th>$J_t$</th>
<th>$S_t$</th>
<th>$x$</th>
<th>Probabilities of moving amongst each state</th>
<th>$=$</th>
<th>Estimated number of students in each state next semester</th>
</tr>
</thead>
</table>

\[
\begin{pmatrix}
    F_t \\
    P_t \\
    J_t \\
    S_t
\end{pmatrix}
\times
\begin{pmatrix}
    P_{FF} & P_{FP} & P_{FJ} & P_{FS} \\
    P_{PF} & P_{PP} & P_{PJ} & P_{PS} \\
    P_{Pj} & P_{JP} & P_{JJ} & P_{JS} \\
    P_{PS} & P_{SP} & P_{SJ} & P_{SS}
\end{pmatrix}
= 
\begin{pmatrix}
    F_{t+1} \\
    P_{t+1} \\
    J_{t+1} \\
    S_{t+1}
\end{pmatrix}
\]
Building the Transition Probability Matrix

Let’s say we want to predict enrollment for next Spring.

- We know how many students we have in each state this Fall.
- We can think about this as predicting how students will move between states from this Fall to next Spring.
- We can use last year’s enrollment data to track movements from last Fall to last Spring.

<table>
<thead>
<tr>
<th>Fall 2018</th>
<th>Spring 2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freshman</td>
<td>?</td>
</tr>
<tr>
<td>Sophomore</td>
<td>?</td>
</tr>
<tr>
<td>Junior</td>
<td>?</td>
</tr>
<tr>
<td>Senior</td>
<td>?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fall 2017</th>
<th>Spring 2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freshman</td>
<td>Freshman</td>
</tr>
<tr>
<td>Sophomore</td>
<td>Sophomore</td>
</tr>
<tr>
<td>Junior</td>
<td>Junior</td>
</tr>
<tr>
<td>Senior</td>
<td>Senior</td>
</tr>
</tbody>
</table>
We can compare our Fall 2017 headcounts in each state to our Spring 2018 headcounts in each state.

- Cross-tabulate Fall 2017 by Spring 2018 and calculate the row percentages:

<table>
<thead>
<tr>
<th>Fall 2017</th>
<th>Spring 2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>P</td>
</tr>
<tr>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
</tr>
</tbody>
</table>

Counts

<table>
<thead>
<tr>
<th>Fall 2017</th>
<th>Spring 2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>3 1 0 0</td>
</tr>
<tr>
<td>P</td>
<td>0 4 1 0</td>
</tr>
<tr>
<td>J</td>
<td>0 0 4 2</td>
</tr>
<tr>
<td>S</td>
<td>0 0 0 5</td>
</tr>
</tbody>
</table>

Percentages

<table>
<thead>
<tr>
<th>Fall 2017</th>
<th>Spring 2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>.75 .25 .00 .00</td>
</tr>
<tr>
<td>P</td>
<td>.00 .80 .20 .00</td>
</tr>
<tr>
<td>J</td>
<td>.00 .00 .66 .33</td>
</tr>
<tr>
<td>S</td>
<td>.00 .00 .00 1.0</td>
</tr>
</tbody>
</table>

We can see that from Fall 2017 to Spring 2018, 75% of Freshmen remained Freshmen, while 25% of Freshmen became Sophomores.

In other words, the probability of becoming a Sophomore in the Spring if you were a Freshman in the Fall is 25%.
Simple Markov Chain Model

Number of students we have this semester in each state at time $t$\[ \begin{array}{c} F_t \quad P_t \quad J_t \quad S_t \end{array} \]

Probabilities of moving amongst each state:

\[
\begin{pmatrix}
P_{FF} & P_{FP} & P_{FJ} & P_{FS} \\
P_{PF} & P_{PP} & P_{PJ} & P_{PS} \\
P_{PJ} & P_{JP} & P_{JJ} & P_{JS} \\
P_{SF} & P_{SP} & P_{SJ} & P_{SS} \\
\end{pmatrix}
\]

Estimated number of students in each state next semester:

\[ \begin{array}{c} F_{t+1} \quad P_{t+1} \quad J_{t+1} \quad S_{t+1} \end{array} \]

Fall 2018 headcounts per state:

\[ \begin{array}{c} 5 \quad 5 \quad 8 \quad 6 \end{array} \]

Transition Probability Matrix based on state flows from Fall 2017 to Spring 2018:

\[
\begin{pmatrix}
0.75 & 0.25 & 0 & 0 \\
0 & 0.8 & 0.2 & 0 \\
0 & 0 & 0.66 & 0.33 \\
0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

Predicted Spring 2019 headcounts:

\[ \begin{array}{c} 4 \quad 5 \quad 6 \quad 8 \end{array} \]
Enhancing the Model

We have so much data, we should be using it!

- Incorporate 5 years of historical data
- Build five Transition Probability Matrices for each set of historical Fall to Spring terms
- Average them to create a master Transition Probability Matrix

\[ P_{FF} \quad P_{FP} \quad P_{FJ} \quad P_{FS} \]
\[ P_{PF} \quad P_{PP} \quad P_{PJ} \quad P_{PS} \]
\[ P_{JF} \quad P_{JP} \quad P_{JJ} \quad P_{JS} \]
\[ P_{SF} \quad P_{SP} \quad P_{SJ} \quad P_{SS} \]
Enhancing the Model

Create detailed states to track granular flows of students

- Concatenate multiple variables to create detailed states that are exhaustive and mutually exclusive

<table>
<thead>
<tr>
<th>DEGREE</th>
<th>ENROLL</th>
<th>CLASS</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Post Baccalaureate Certificate</td>
<td>1 New Student</td>
<td>1 Freshman</td>
<td>F Full-time</td>
</tr>
<tr>
<td>3 Bachelor's</td>
<td>2 New Transfer</td>
<td>2 Sophomore</td>
<td>P Part-time</td>
</tr>
<tr>
<td>4 Master's</td>
<td>3 Student</td>
<td>3 Junior</td>
<td></td>
</tr>
<tr>
<td>5 Post Master's Certificate</td>
<td>4 Continuing Student</td>
<td>4 Senior</td>
<td></td>
</tr>
<tr>
<td>8 Unclassified</td>
<td>5 Returning Student</td>
<td>6 Unclassified Undergraduate</td>
<td></td>
</tr>
<tr>
<td>P Doctoral Professional</td>
<td>6 Unclassified</td>
<td>7 Graduate</td>
<td></td>
</tr>
<tr>
<td>R Doctorate</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example: **3_2_3_P** is a new transferring junior seeking a Bachelor’s degree part-time
New Entries

There are new students entering and exiting the university every semester

- Exit are already accounted for by using the Transition Probability Matrix
- New entries must be modeled separately
  - Use our semester pairings to identify how many new students enter in each Spring
    - Flag students who were not here in Fall, but were here in Spring
  - Our data shows that new entries are very consistent across semesters, so we can estimate future new entries using linear regression

<table>
<thead>
<tr>
<th>3_2_3_P</th>
<th>New Transferring Junior seeking a Bachelor's degree Part-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester</td>
<td>Number of New Entries</td>
</tr>
<tr>
<td>Spring 2019 estimate</td>
<td>141</td>
</tr>
<tr>
<td>Spring 2018</td>
<td>118</td>
</tr>
<tr>
<td>Spring 2017</td>
<td>109</td>
</tr>
<tr>
<td>Spring 2016</td>
<td>74</td>
</tr>
<tr>
<td>Spring 2015</td>
<td>61</td>
</tr>
<tr>
<td>Spring 2014</td>
<td>41</td>
</tr>
</tbody>
</table>
Enhanced Markov Chain Model

Number of students we have this semester in each state at time $t$ x Probabilities of moving amongst each state, averaged across past 5 years + Predicted new entries into each state = Estimated number of students in each state next semester

\[
\begin{array}{ccc}
3_{1,1,F_t} & 3_{1,1,P_t} & 3_{2,3,F_t} \\
3_{1,1,F} & 3_{1,1,F} & 3_{1,1,F} \\
3_{1,1,P} & 3_{1,1,P} & 3_{1,1,P} \\
3_{2,3,F} & 3_{1,1,P} & 3_{1,1,F} \\
\vdots & \vdots & \vdots \\
\end{array}
\times
\begin{array}{ccc}
P_{3_{1,1,F}} & P_{3_{1,1,F}} & P_{3_{1,1,F}} \\
P_{3_{1,1,F}} & P_{3_{1,1,F}} & P_{3_{1,1,F}} \\
P_{3_{1,1,F}} & P_{3_{1,1,F}} & P_{3_{1,1,F}} \\
P_{3_{2,3,F}} & P_{3_{1,1,F}} & P_{3_{1,1,F}} \\
\vdots & \vdots & \vdots \\
\end{array}
+ \begin{array}{ccc}
3_{1,1,F_{\text{new}}} & 3_{1,1,F_{\text{new}}} & 3_{2,3,F_{\text{new}}} \\
3_{1,1,F} & 3_{1,1,F} & 3_{1,1,F} \\
3_{1,1,F} & 3_{1,1,F} & 3_{1,1,F} \\
3_{2,3,F} & 3_{1,1,F} & 3_{1,1,F} \\
\vdots & \vdots & \vdots \\
\end{array}
= \begin{array}{ccc}
3_{1,1,F_{t+1}} & 3_{1,1,F_{t+1}} & 3_{2,3,F_{t+1}} \\
3_{1,1,F_{t+1}} & 3_{1,1,F_{t+1}} & 3_{1,1,F_{t+1}} \\
3_{1,1,F_{t+1}} & 3_{1,1,F_{t+1}} & 3_{1,1,F_{t+1}} \\
3_{2,3,F_{t+1}} & 3_{1,1,F_{t+1}} & 3_{1,1,F_{t+1}} \\
\vdots & \vdots & \vdots \\
\end{array}
\]
Markov Chain Modeling in SAS

- Efficiently process large data
  - Combine multiple historical datasets

- Dynamic model
  - Enter term predicted, SAS does the rest

- Concatenate multiple variables to create detailed flow states
  - Very large Transition Probability Matrices

- Easily conduct multiple kinds of analyses
  - Regressions, cross-tabulations, matrix algebra, etc.
Dynamic SAS Programming

- Minimizes risk of user-error
- Simple to update
- Efficient

Macro Variables

Macro Programs
SAS processes simple arithmetic to create variables for past semesters. Given a projection term of '201801', code resolves:

```
DATA null;
IF substr("&projection", 5, 2)="01" THEN DO;
  semester0=PUT(&projection, 6.);
  semester1=PUT(&projection-93, 6.);
  semester2=PUT(semester1-7, 6.);
  semester3=PUT(semester2-93, 6.);
  semester4=PUT(semester3-7, 6.);
  semester5=PUT(semester4-93, 6.);
  semester6=PUT(semester5-7, 6.);
  semester7=PUT(semester6-93, 6.);
  semester8=PUT(semester7-7, 6.);
  semester9=PUT(semester8-93, 6.);
  semester10=PUT(semester9-7, 6.);
  semester11=PUT(semester10-93, 6.);
  predict_term=substr("&projection", 5, 2);
END;
ELSE IF substr("&projection", 5, 2)="08" THEN DO;
  semester0=PUT(&projection, 6.);
  semester1=PUT(&projection-93, 6.);
  semester2=PUT(semester1-7, 6.);
  semester3=PUT(semester2-93, 6.);
  semester4=PUT(semester3-7, 6.);
  semester5=PUT(semester4-93, 6.);
  semester6=PUT(semester5-7, 6.);
  semester7=PUT(semester6-93, 6.);
  semester8=PUT(semester7-93, 6.);
  semester9=PUT(semester8-7, 6.);
  semester10=PUT(semester9-93, 6.);
  semester11=PUT(semester10-7, 6.);
  predict_term=substr("&projection", 5, 2);
END;
```

The CALL SYMPUT routine creates macro variables for each semester that assign the calculated semester values.
PROC SQL NOPRINT;
SELECT TRIM(LEFT(NAME)) INTO :cert SEPARATED BY ',' FROM vars
WHERE student_cat="certificate";
SELECT TRIM(LEFT(NAME)) INTO :ugrd SEPARATED BY ',' FROM vars
WHERE student_cat="undergrad";
SELECT TRIM(LEFT(NAME)) INTO :mstr SEPARATED BY ',' FROM vars
WHERE student_cat="masters";
SELECT TRIM(LEFT(NAME)) INTO :spcl SEPARATED BY ',' FROM vars
WHERE student_cat="specialist";
SELECT TRIM(LEFT(NAME)) INTO :ugnd SEPARATED BY ',' FROM vars
WHERE student_cat="ug non-deg";
SELECT TRIM(LEFT(NAME)) INTO :grnd SEPARATED BY ',' FROM vars
WHERE student_cat="gr non-deg";
SELECT TRIM(LEFT(NAME)) INTO :dctr SEPARATED BY ',' FROM vars
WHERE student_cat="doctorate";
QUIT;

DATA projections;
SET iml_projection;
Certificate=ROUND(sum(&cert),1);
Undergraduate=ROUND(sum(&ugrd),1);
Masters=ROUND(sum(&mstr),1);
Specialist=ROUND(sum(&spcl),1);
UG_Nondegree=ROUND(sum(&ugnd),1);
GR_Nondegree=ROUND(sum(&grnd),1);
Doctoral=ROUND(sum(&dctr),1);
Total=sum(Certificate,
   Undergraduate,
   Masters,
   Specialist,
   UG_nondegree,
   GR_nondegree,
   Doctoral);
TERM="&semester0";
KEEP TERM
   Certificate
   Undergraduate
   Masters
   Specialist
   UG_Nondegree
   GR_Nondegree
   Doctoral
   Total;
RUN;
PROC PRINT DATA=projections noobs;
TITLE "&semester0 Enrollment Projections";
RUN;
macro program that loops through every distinct flow state and conducts a linear regression to predict new entries into each flow state

```latex
%MACRO entry(i, semestera, semesterb);
/* start at the earliest term and work up */
DATA academicyear;
SET one;
WHERE termcode in("&semestera", "&semesterb");
PROC SORT DATA=academicyear;
RUN;
PROC SORT DATA=academicyear;
RUN;
DATA entry\&i;
SET campus_id;
BY campus_id;
IF TERMCODE IN("&semester10", "&semester11") THEN years_past=0;
ELSE IF TERMCODE IN("&semester8", "&semester9") THEN years_past=1;
ELSE IF TERMCODE IN("&semester6", "&semester7") THEN years_past=2;
ELSE IF TERMCODE IN("&semester4", "&semester5") THEN years_past=3;
ELSE IF TERMCODE IN("&semester2", "&semester3") THEN years_past=4;
ELSE IF TERMCODE IN("&semester0", "&semester1") THEN years_past=5;
IF FIRST.campus_id and termcode="&semesterb" THEN entry=1;
IF entry NE 1 THEN DELETE;
KEEP termcode enr1 campus_id flow years_past;
RUN;
%MEND entry;
%entry(1, &semestera1, &semestera10);
%entry(2, &semestera9, &semestera8);
%entry(3, &semestera7, &semestera6);
%entry(4, &semestera5, &semestera4);
%entry(5, &semestera3, &semestera2);
```

uses macro variables to determine semester pairs

macro program that compares semester pairs to identify new entries between first and second semester

```latex
%MACRO reg;
%DO i=1 %TO &cnt;
PROC REG DATA=new_flows_reg NOPRINT;
MODEL COUNT=yrs_past;
WHERE flow="&Numi";
OUTPUT OUT=new\&i
predicted=predict_cnt
residual=resid;
QUIT;
%END;
%MEND reg;
%reg;
```

uses macro variables for each flow state
SAS Methodology

Step 1
- Read in the data – student level, most recent term and past 5 years
  - Concatenate Degree, Enrollment Status, Class, and Full-time/Part-time

Step 2
- Create five semester pairings of Springs > Falls (or Falls > Springs)

Step 3
- Create five transition probability matrices for each semester pairing
  - Compare semester pairings to see what percentage of students in each flow state retained, dropped out, or moved to another flow state

Step 4
- Average across the five transition probability matrices to create an overall Transition Probability Matrix

Step 5
- Pull in last semester’s enrollment values as our baseline population

Step 6
- Use linear regression to model new entries

Step 7
- Use PROC IML to forecast enrollment for next semester!
PROC IML;
vars={&c_list};
USE trans_matrix; READ ALL INTO trans_matrix;
USE base_pop; READ ALL INTO base_pop;
USE new_entries; READ ALL INTO new_entries;
base_pop=base_pop[1, 2:(&cnt+1)];
new_entries=new_entries[,1:&cnt];
iml_projection=(base_pop*trans_matrix)+new_entries;
CREATE iml_projection FROM iml_projection [COLNAME=vars];
APPEND FROM iml_projection;
QUIT;

\[
\text{Number of students we have this semester in each state at time } t 
\times \text{ Probabilities of moving amongst each state, averaged across past 5 years} 
+ \text{ Predicted new entries into each state} 
= \text{Estimated number of students in each state next semester}
\]
### Results

<table>
<thead>
<tr>
<th>Semester</th>
<th>Undergraduate</th>
<th>Non-Degree Seeking Undergraduate</th>
<th>Certificate</th>
<th>Masters</th>
<th>Specialist</th>
<th>Doctoral</th>
<th>Non-Degree Seeking Graduate</th>
<th>Total Enrollment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring 2014</td>
<td>13,294</td>
<td>298</td>
<td>155</td>
<td>1,915</td>
<td>97</td>
<td>735</td>
<td>377</td>
<td>16,871</td>
</tr>
<tr>
<td>Spring 2015</td>
<td>13,702</td>
<td>360</td>
<td>127</td>
<td>1,927</td>
<td>107</td>
<td>764</td>
<td>332</td>
<td>17,319</td>
</tr>
<tr>
<td>Spring 2016</td>
<td>14,265</td>
<td>404</td>
<td>206</td>
<td>1,924</td>
<td>44</td>
<td>794</td>
<td>240</td>
<td>17,877</td>
</tr>
<tr>
<td>Spring 2017</td>
<td>14,874</td>
<td>420</td>
<td>167</td>
<td>1,838</td>
<td>22</td>
<td>920</td>
<td>270</td>
<td>18,511</td>
</tr>
<tr>
<td>Spring 2018</td>
<td>15,116</td>
<td>388</td>
<td>143</td>
<td>1,965</td>
<td>21</td>
<td>981</td>
<td>232</td>
<td>18,846</td>
</tr>
<tr>
<td><strong>Projected</strong></td>
<td><strong>15,242</strong></td>
<td><strong>337</strong></td>
<td><strong>145</strong></td>
<td><strong>1,966</strong></td>
<td><strong>45</strong></td>
<td><strong>836</strong></td>
<td><strong>220</strong></td>
<td><strong>18,791</strong></td>
</tr>
<tr>
<td><strong>Spring 2019</strong></td>
<td><strong>15,081</strong></td>
<td><strong>391</strong></td>
<td><strong>137</strong></td>
<td><strong>1,967</strong></td>
<td><strong>52</strong></td>
<td><strong>994</strong></td>
<td><strong>235</strong></td>
<td><strong>18,857</strong></td>
</tr>
<tr>
<td><strong>Actual</strong></td>
<td><strong>15,081</strong></td>
<td><strong>391</strong></td>
<td><strong>137</strong></td>
<td><strong>1,967</strong></td>
<td><strong>52</strong></td>
<td><strong>994</strong></td>
<td><strong>235</strong></td>
<td><strong>18,857</strong></td>
</tr>
</tbody>
</table>
Results

Spring Semester Projections Versus Historical Enrollment
(Undergraduates displayed separately)

- Non-Degree Seeking Undergraduate
- Certificate
- Masters
- Specialist
- Doctoral
- Non-Degree Seeking Graduate

Spring Semester Projections Versus Historical Enrollment
(Undergraduates only)

Headcount

Spring 2014  Spring 2015  Spring 2016  Spring 2017  Spring 2018  Spring 2019

Headcount

Spring 2014  Spring 2015  Spring 2016  Spring 2017  Spring 2018  Spring 2019

UNC GREENSBORO
Institutional Research
Questions?

You can download this presentation at:
https://ire.uncg.edu/research/PredictEnrollment/SRB-AIR-2019/

Contact info:
Samantha Bradley
srbradle@uncg.edu
(336) 256-0399

Please remember to submit your evaluation for this session.