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

UNC GREENSBORO
Institutional Research

AIR Forum DENVER
May 28-31, 2019

Office of Institutional Research
The 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

Enrollment Data

- 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

Markov Chain 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*	x	Probabilities of moving amongst each state	=	Estimated number of students in each state next semester

$$\begin{bmatrix} F_t & P_t & J_t & S_t \end{bmatrix} \times \begin{bmatrix} P_{FF} & P_{FP} & P_{FJ} & P_{FS} \\ P_{PF} & P_{PP} & P_{PJ} & P_{PS} \\ P_{JF} & P_{JP} & P_{JJ} & P_{JS} \\ P_{SF} & P_{SP} & P_{SJ} & P_{SS} \end{bmatrix} = \begin{bmatrix} F_{t+1} & P_{t+1} & J_{t+1} & S_{t+1} \end{bmatrix}$$

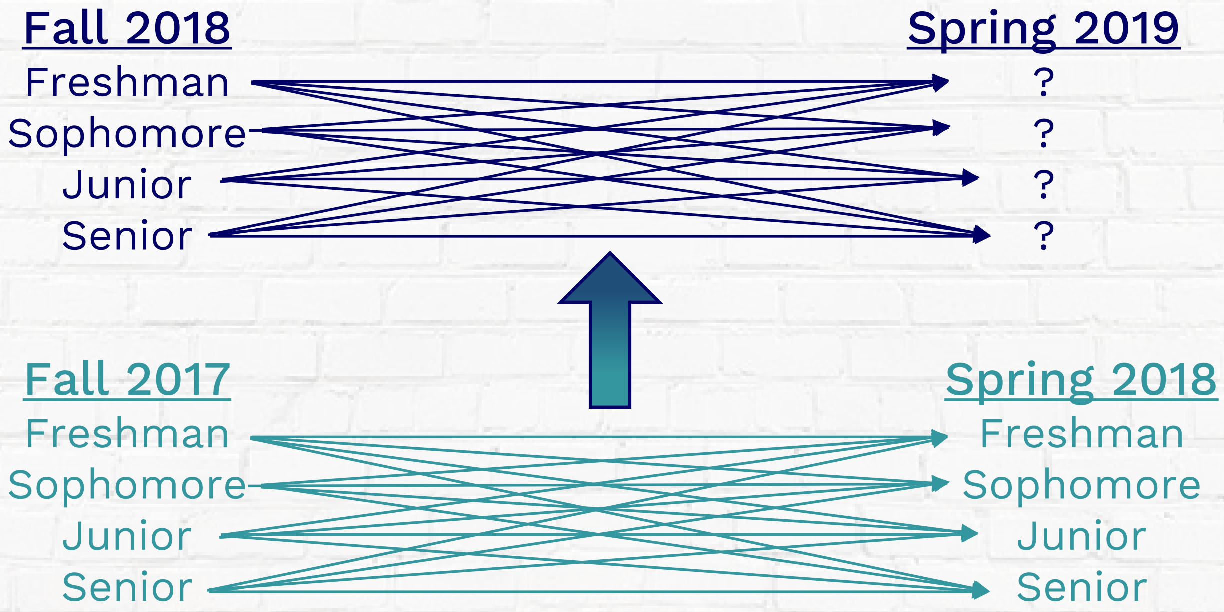
UNC GREENSBORO
Institutional Research

AIR Forum DENVER May 28-31, 2019

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



Building the Transition Probability Matrix

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:

Start with student-level enrollment data

Fall 2017	Spring 2018
F	F
F	F
F	F
F	P
P	P
P	P
P	P
P	P
P	J
J	J
J	J
J	J
J	J
J	S
J	S
S	S
S	S
S	S
S	S
S	S

Cross-tabulate Fall 2017 by Spring 2018

Counts

Fall 2017 \ Spring 2018	F	P	J	S
F	3	1	0	0
P	0	4	1	0
J	0	0	4	2
S	0	0	0	5

Percentages

Fall 2017 \ Spring 2018	F	P	J	S
F	.75	.25	.00	.00
P	.00	.80	.20	.00
J	.00	.00	.66	.33
S	.00	.00	.00	1.0

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* × Probabilities of moving amongst each state = Estimated number of students in each state next semester

$$\begin{bmatrix} F_t & P_t & J_t & S_t \end{bmatrix} \times \begin{bmatrix} P_{FF} & P_{FP} & P_{FJ} & P_{FS} \\ P_{PF} & P_{PP} & P_{PJ} & P_{PS} \\ P_{JF} & P_{JP} & P_{JJ} & P_{JS} \\ P_{SF} & P_{SP} & P_{SJ} & P_{SS} \end{bmatrix} = \begin{bmatrix} F_{t+1} & P_{t+1} & J_{t+1} & S_{t+1} \end{bmatrix}$$

$$\begin{bmatrix} 5 & 5 & 8 & 6 \end{bmatrix} \times \begin{bmatrix} 0.75 & 0.25 & 0 & 0 \\ 0 & 0.8 & 0.2 & 0 \\ 0 & 0 & 0.66 & 0.33 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 5 & 6 & 8 \end{bmatrix}$$

Fall 2018 headcounts per state

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

Predicted Spring 2019 headcounts

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

Fall 2017 → Spring 2018

Fall 2016 → Spring 2017

Fall 2015 → Spring 2016

Fall 2014 → Spring 2015

Fall 2013 → Spring 2014

$$\begin{bmatrix} P_{FF} & P_{FP} & P_{FJ} & P_{FS} \\ P_{PF} & P_{PP} & P_{PJ} & P_{PS} \\ P_{JF} & P_{JP} & P_{JJ} & P_{JS} \\ P_{SF} & P_{SP} & P_{SJ} & P_{SS} \end{bmatrix}$$

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

DEGREE	
0	Post Baccalaureate Certificate
3	Bachelor's
4	Master's
5	Post Master's Certificate
8	Unclassified
P	Doctoral Professional
R	Doctorate

ENROLL	
1	New Student
2	New Transfer Student
3	Continuing Student
4	Returning Student
6	Unclassified

CLASS	
1	Freshman
2	Sophomore
3	Junior
4	Senior
6	Unclassified Undergraduate
7	Graduate

TIME	
F	Full-time
P	Part-time

Example: 3_2_3_P is a new transferring junior seeking a Bachelor's degree part-time

New Entries

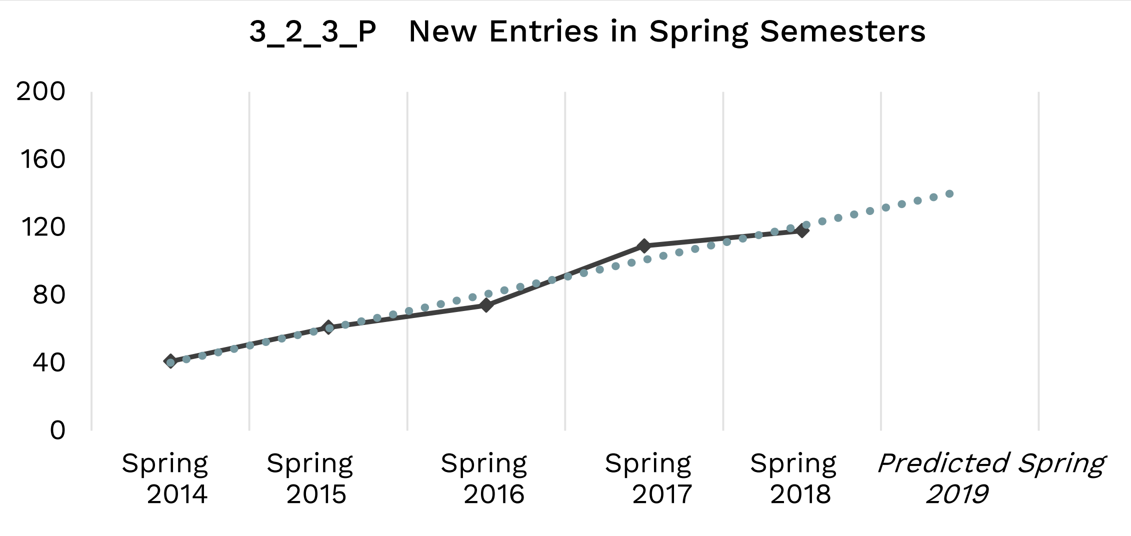
There are new students entering and exiting the university every semester

- Exits 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

3_2_3_P	New Transferring Junior seeking a Bachelor's degree Part-time
Semester	Number of New Entries
Spring 2019 estimate	*141*
Spring 2018	118
Spring 2017	109
Spring 2016	74
Spring 2015	61
Spring 2014	41



Enhanced Markov Chain Model

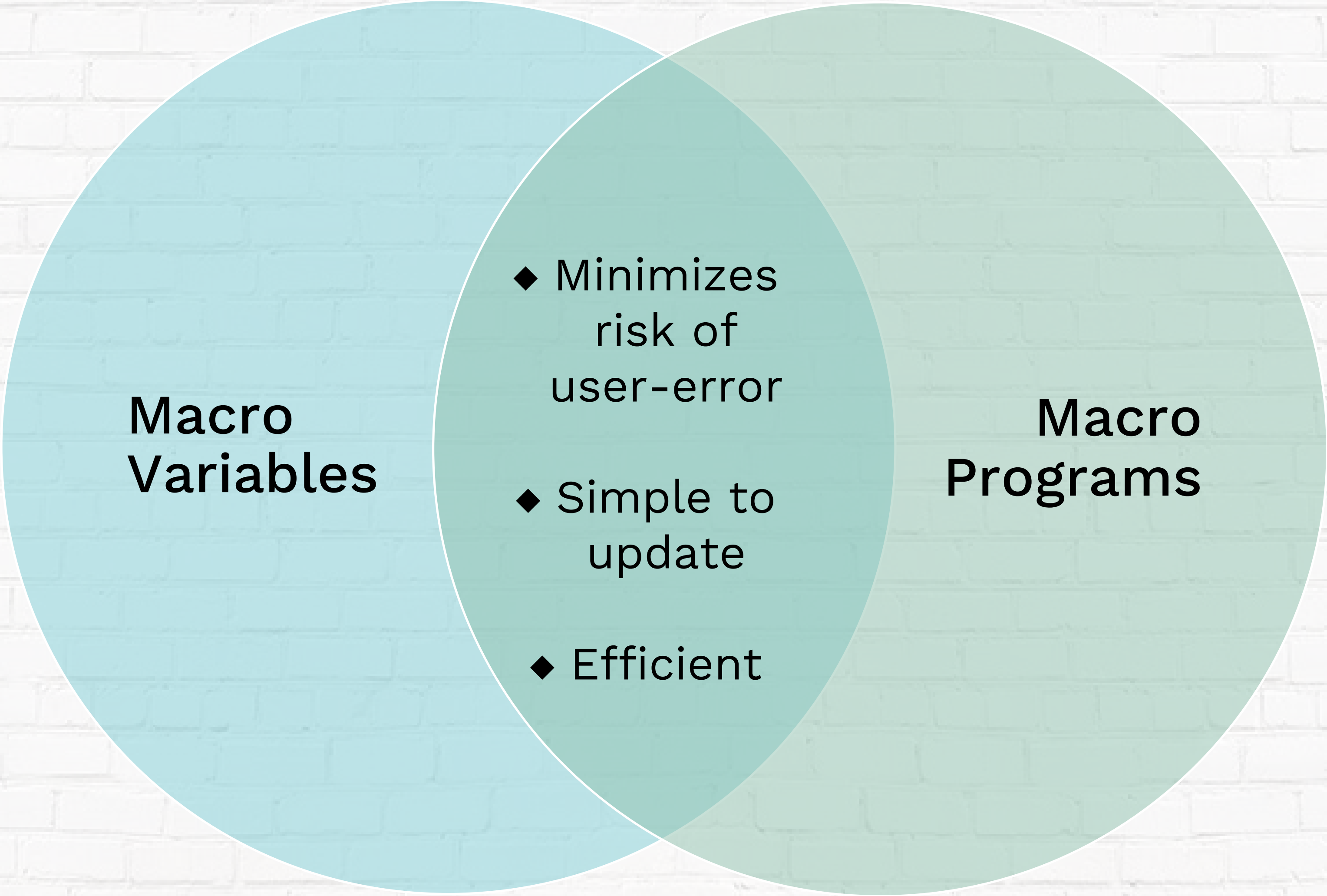
[Number of students we have this semester in each state at time *t* x Probabilities of moving amongst each state, averaged across past 5 years] + Predicted new entries into each state = Estimated number of students in each state next semester

$$
\left[
\begin{bmatrix} 3\_1\_1\_F_{t} & 3\_1\_1\_P_{t} & 3\_2\_3\_F_{t} & \cdots \end{bmatrix}
\times
\begin{bmatrix}
P_{\substack{3\_1\_1\_F \\ 3\_1\_1\_F}} & P_{\substack{3\_1\_1\_F \\ 3\_1\_1\_P}} & P_{\substack{3\_1\_1\_F \\ 3\_2\_3\_F}} & \cdots \\
P_{\substack{3\_1\_1\_P \\ 3\_1\_1\_F}} & P_{\substack{3\_1\_1\_P \\ 3\_1\_1\_P}} & P_{\substack{3\_1\_1\_P \\ 3\_2\_3\_F}} & \cdots \\
P_{\substack{3\_2\_3\_F \\ 3\_1\_1\_F}} & P_{\substack{3\_2\_3\_F \\ 3\_1\_1\_P}} & P_{\substack{3\_2\_3\_F \\ 3\_2\_3\_F}} & \cdots \\
\cdots & \cdots & \cdots & \cdots
\end{bmatrix}
\right]
+
\begin{bmatrix} 3\_1\_1\_F_{new} & 3\_1\_1\_P_{new} & 3\_2\_3\_F_{new} & \cdots \end{bmatrix}
=
\begin{bmatrix} 3\_1\_1\_F_{t+1} & 3\_1\_1\_P_{t+1} & 3\_2\_3\_F_{t+1} & \cdots \end{bmatrix}
$$

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



only element the user changes

```
/* Enter the projection term */
%let projection=201801;
```

SAS processes simple arithmetic to create variables for past semesters.

Given a projection term of '201801', code resolves:

```
semester0  = 201801
semester1  = 201708
semester2  = 201701
semester3  = 201608
semester4  = 201601
semester5  = 201508
semester6  = 201501
semester7  = 201408
semester8  = 201401
semester9  = 201308
semester10 = 201301
semester11 = 201208
```

```
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-7,6.);
    semester2=PUT(semester1-93,6.);
    semester3=PUT(semester2-7,6.);
    semester4=PUT(semester3-93,6.);
    semester5=PUT(semester4-7,6.);
    semester6=PUT(semester5-93,6.);
    semester7=PUT(semester6-7,6.);
    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;
```

```
CALL SYMPUT('semester0',semester0);
CALL SYMPUT('semester1',semester1);
CALL SYMPUT('semester2',semester2);
CALL SYMPUT('semester3',semester3);
CALL SYMPUT('semester4',semester4);
CALL SYMPUT('semester5',semester5);
CALL SYMPUT('semester6',semester6);
CALL SYMPUT('semester7',semester7);
CALL SYMPUT('semester8',semester8);
CALL SYMPUT('semester9',semester9);
CALL SYMPUT('semester10',semester10);
CALL SYMPUT('semester11',semester11);
```

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-degr";
SELECT TRIM(LEFT(NAME))
    INTO :grnd SEPARATED BY ','
    FROM vars
    WHERE student_cat="gr non-degr";
SELECT TRIM(LEFT(NAME))
    INTO :dctr SEPARATED BY ','
    FROM vars
    WHERE student_cat="doctorate";
QUIT;
```

creating macro variables for each student category within a PROC SQL step

```
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;
```

call the macro variables anywhere throughout the program macro program that compares semester pairs to identify new entries between first and second semester

```
%MACRO entry(i,semestera,semesterb);
/* start at the earliest term and work up */
DATA academicyear;
SET one;
WHERE termcode in("&semestera","&semesterb");
RUN;
PROC SORT DATA=academicyear;
BY CAMPUS_ID TERMCODE;
RUN;
DATA entry&i;
SET academicyear;
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 enrl campus_id flow years_past;
RUN;
%MEND entry;

%entry(1,&semester11,&semester10);
%entry(2,&semester9,&semester8);
%entry(3,&semester7,&semester6);
%entry(4,&semester5,&semester4);
%entry(5,&semester3,&semester2);
```

uses macro variables to determine semester pairs

macro program that loops through every distinct flow state and conducts a linear regression to predict new entries into each flow state

```
%MACRO reg;
%DO i=1 %TO &cnt;
PROC REG DATA=new_flows_reg NOPRINT;
MODEL COUNT=years_past;
WHERE flow="&&Var&i";
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 in SAS

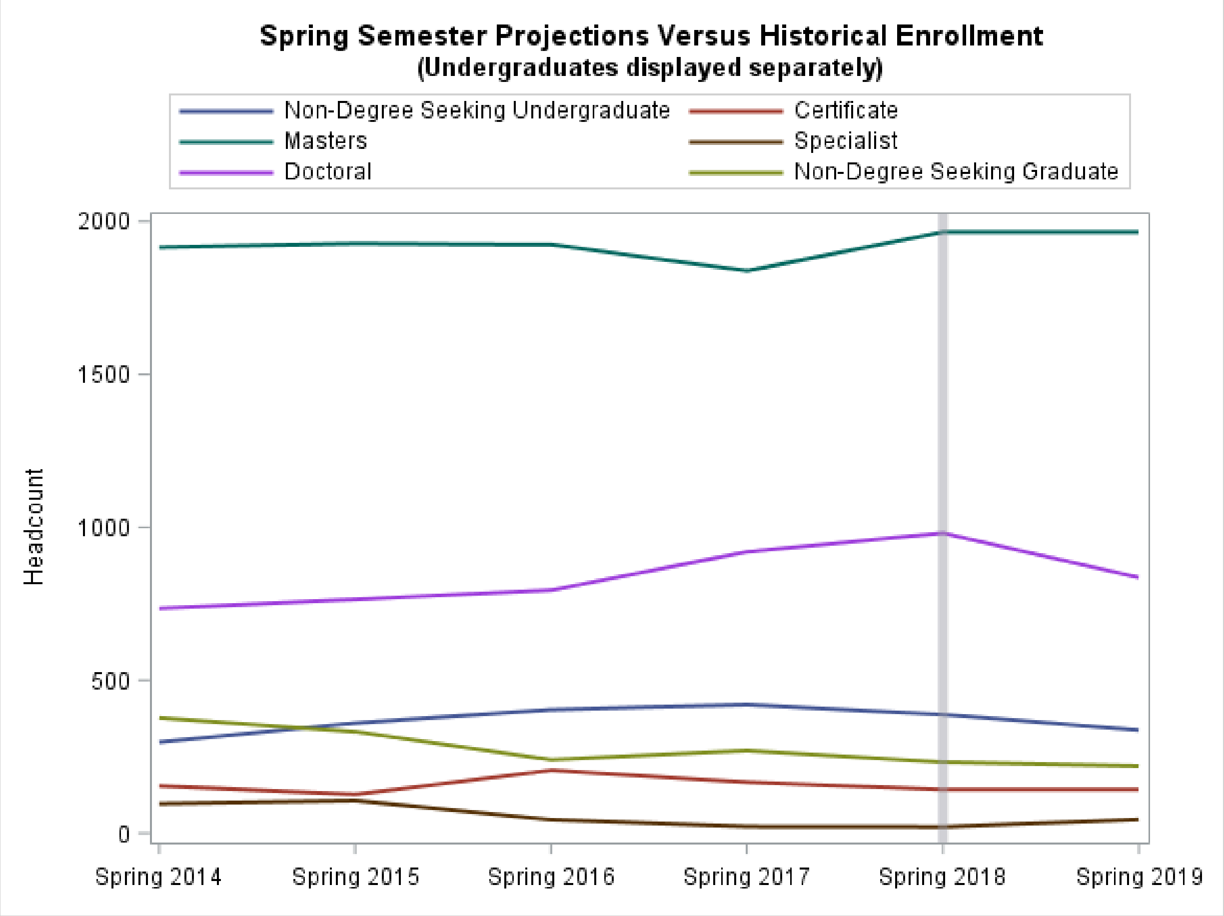
```
PROC IML;
vars={&c_list};
USE trans_matrix;   READ ALL INTO trans_matrix;
USE base_pop;       READ ALL INTO base_pop;
USE new_entries;    READ ALL INTO new_entries;
base_pop=base_pop[1, 2:(&cnt+1)];
new_entries=new_entries[,1:&cnt];
iml_projection=(base_pop*trans_matrix)+new_entries ;
CREATE iml_projection FROM iml_projection [COLNAME=vars];
APPEND FROM iml_projection;
QUIT;
```

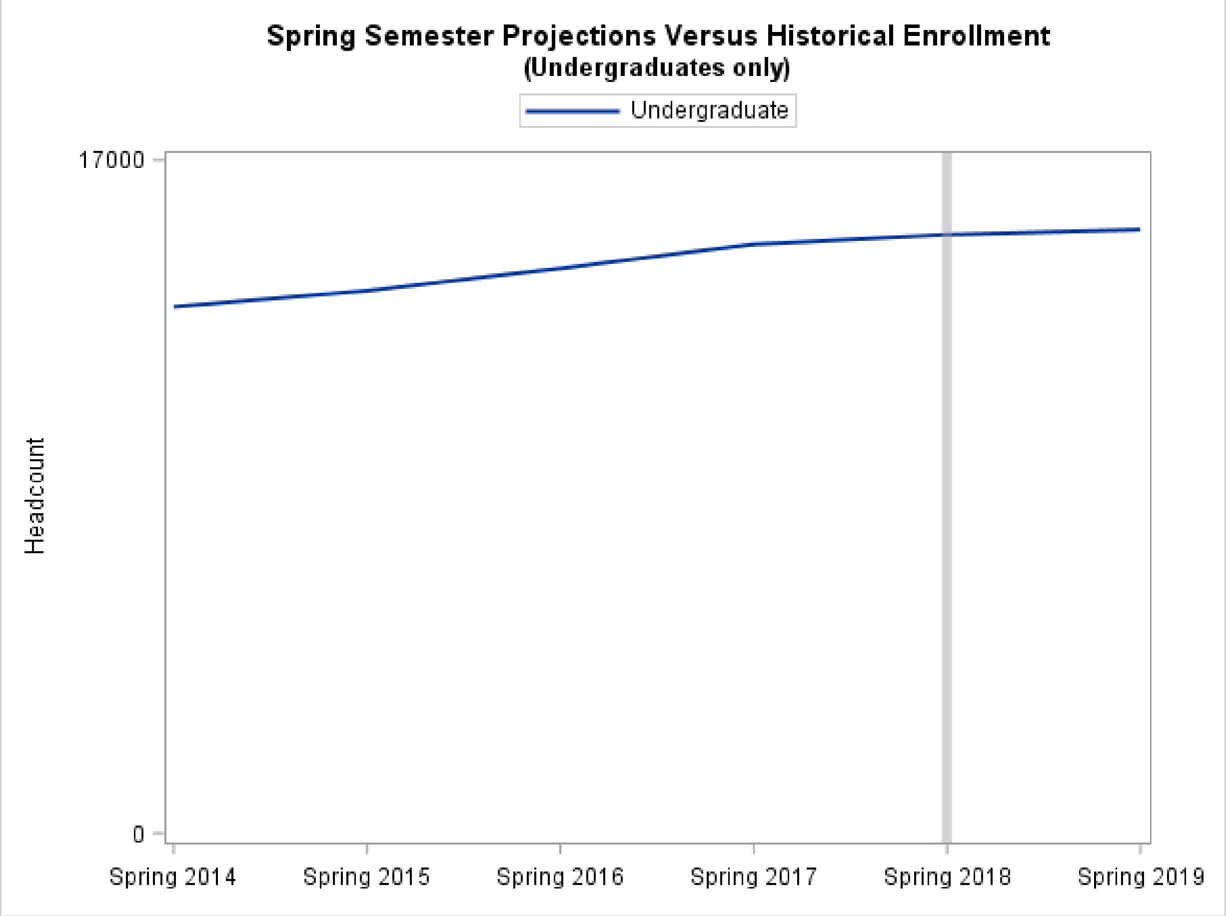
[Number of students we have this semester in each state at time *t* × Probabilities of moving amongst each state, averaged across past 5 years] + Predicted new entries into each state = Estimated number of students in each state next semester

Results

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

Results





UNC GREENSBORO
Institutional Research

AIR Forum DENVER May 28-31, 2019

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.