Curricular Complexity

Jonathan Chang, Dan Jin, Situ Ma, Ki Hyun Park, Jeremy Tran

Goals

- Utilize course prerequisite data to compute a set of course complexity scores
- Employ prerequisite and major requirement data to compute a set of curricular complexity scores for 15 selected majors, so we can easily assess and compare complexity and schools
- Uncover role that the structure of a curriculum plays in student academic success, if any, and realize the amount of variance these complexity metrics are explaining
- Provide suggestions on how to should structure major requirement data into tables so that it can be of use for our future students, administrators, and needs

Methodologies

- Obtain relevant data for 15 majors
  - Conduct EDA and comparative analysis
- Gain insights on major related data and major complexity
- Web scrape from MyUCLA to extract major requirements and course requirements
- Create curriculum mappings to visualize major complexity

Data Table - Result

Major	Hours	Out Class	In Class	Longest F	Bottlenec	Avg TTD	Stem	Curriculum Complexity Score
Aerospace Engineering	124	13	4	5	3	13.047197	Yes	16.56
Nursing	140	13	12	13	2	12.482598	Yes	17.93
Ling & Philosphy	100	4	6	6	3	11.777777	No	12.09
Statistics	81	14	12	12	11	12.495088	Yes	10.66
Political Science	65	1	1	1	0	12.247796	No	7.96
Art	97	4	3	3	1	12.426592	No	12.25
Atmosphere and Oceanic Sciences	96	4	6	7	4	12.698924	Yes	12.55
Computer Science	123	6	6	7	10	12.581143	Yes	15.90
World Arts and Culture	70	2	4	5	2	12.620689	No	9.07
Economics	66	8	9	8	3	12.115099	Yes	8.39
Biology	88	4	6	5	3	12.611016	Yes	11.42
DESMA	89	5	5	5	3	12.404639	No	11.36
Mathematics	93	20	8	11	8	12.617529	Yes	12.26
Music	102	6	6	6	8	12.803125	No	13.47
Russian Studies	91	9	9	9	7	13.074074	No	12.38

Data Table - Variables

```
((Hours+log(Out Class)+log(In Class)+log(Longest path)+log(Bottlenecks))*Average TTD)/100
```

For each major field of study:

- **Hours**: number of required units to obtain the degree
- **Out class**: highest number of pre-requirements that the lowest vyebaclass fulfills
- **In class**: highest number of pre-reqs coming from a upper division class
- **Longest path**: number of classes need to take consecutively
- **Bottlenecks**: number of classes that each is a prerequisite for more than 3 classes
- **Avg TTD**: average total terms needed to graduate
- **Curriculum Complexity Score**: overall complexity of a major

Calculating Curriculum Complexity

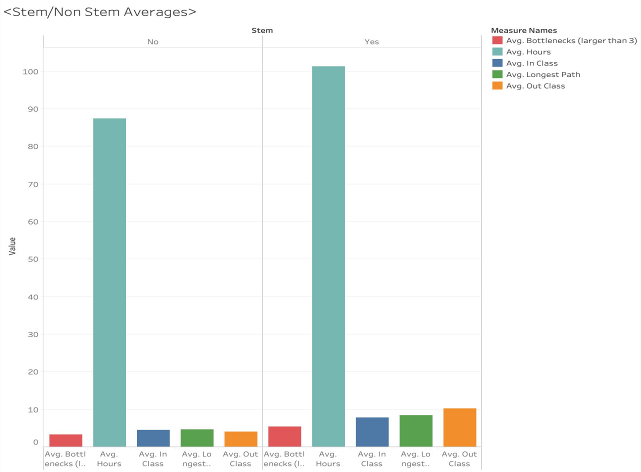
- Curriculum Complexity was calculated using hours, out class, in class, longest path, bottlenecks and average time to degree

Curriculum Complexity = ((Hours+log(Out Class)+log(In Class)+log(Longest Path)+log(Bottlenecks))*Average TTD)/100

- Take the log of several variable sto normalize the data and avoid skewness

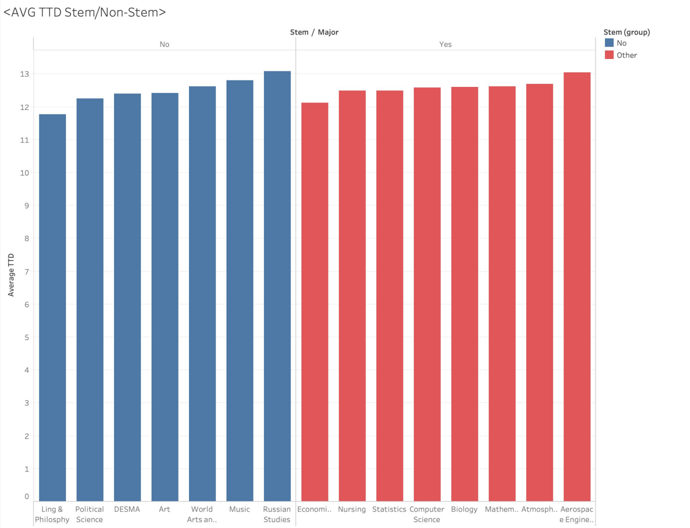
Plots

The average bottlenecks, hours in/out class and longest paths for stem majors versus non stem



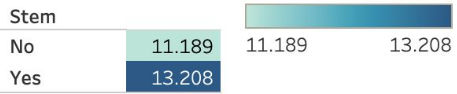
Avg. Bottlenecks (larger than 3), Avg. Hours, Avg. In Class, Avg. Longest Path and Avg. Out Class for each Stem. Color shows details about Avg. Bottlenecks (larger than 3), Avg. Hours, Avg. In Class, Avg. Longest Path and Avg. Out Class.

The average time to degree for stem versus non stem majors



Sum of Average TTD for each Major broken down by Stem. Color shows details about Stem (group).

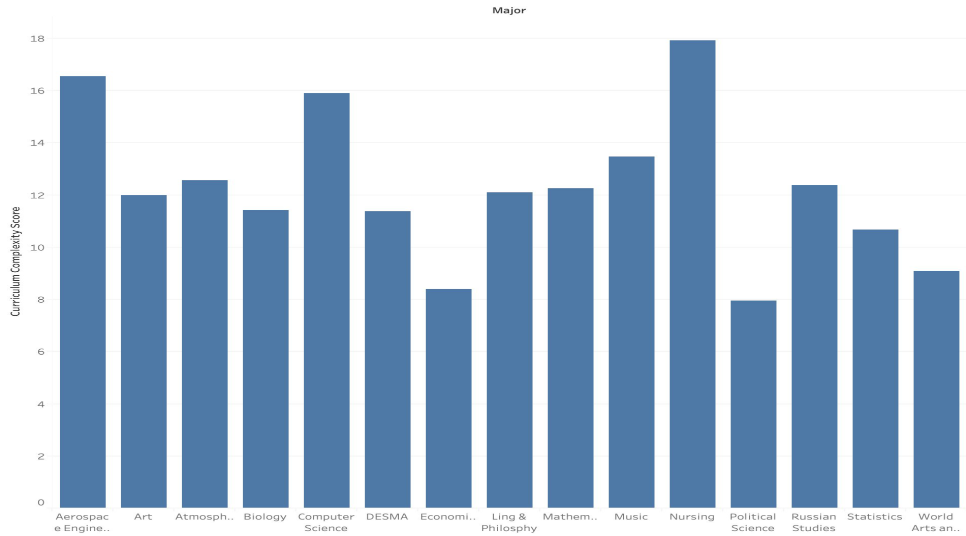
Plots


Average Curriculum Complexity Score Stem vs Non Stem

Stem	
No	11.189
Yes	13.208

Average of Curriculum Complexity Score broken down by Stem. Color shows average of Curriculum Complexity Score. The marks are labeled by average of Curriculum Complexity Score.

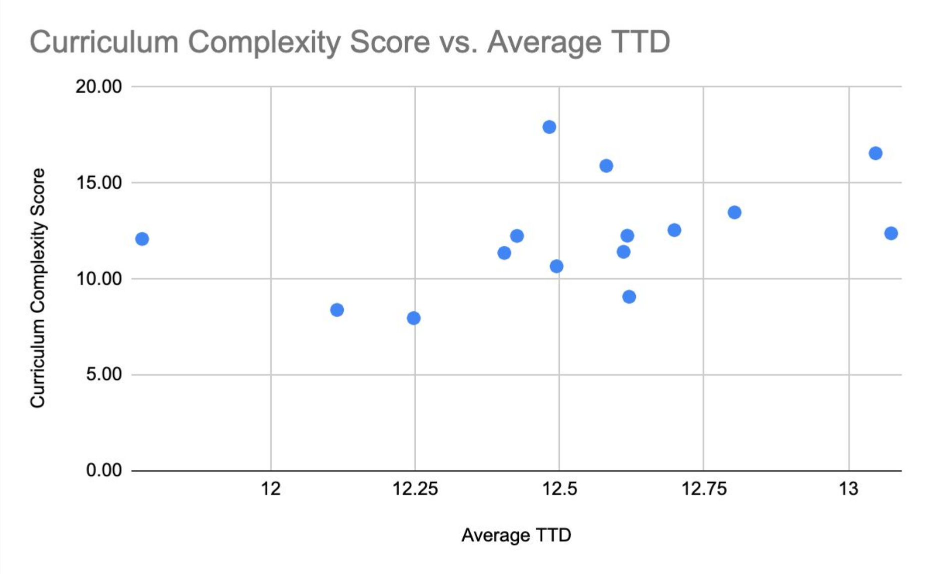
Stem majors on average have a higher curriculum complexity score than non stem majors



Sum of Curriculum Complexity Score for each Major.

Plots

Average TTD is not a statistically insignificant contributor to Curriculum Complexity Score



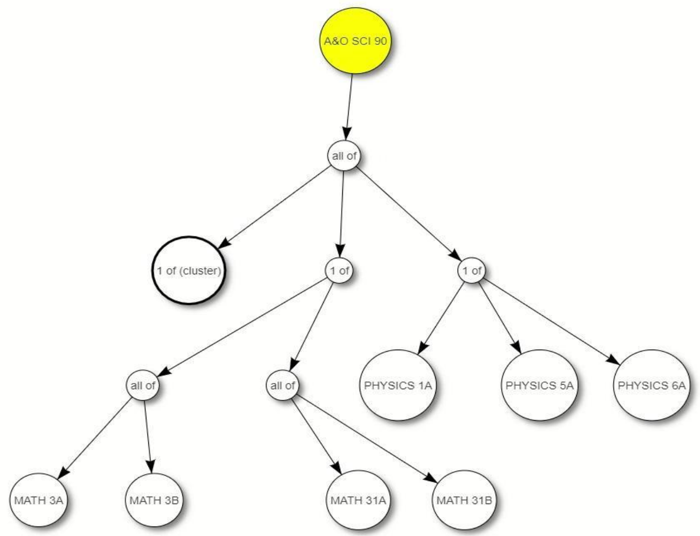

```
Call:
lm(formula = Curriculum.Complexity.Score ~ Avg.TTD)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.5159 -1.6350 -0.3149  1.3846  5.8236 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -31.240     27.267  -1.146    0.273
Avg.TTD        3.473      2.175   1.597    0.134

Residual standard error: 2.698 on 13 degrees of freedom
  (3 observations deleted due to missingness)
Multiple R-squared:  0.164,	Adjusted R-squared:  0.09964 
F-statistic: 2.549 on 1 and 13 DF,  p-value: 0.1344
```

A&O SCI 90 Requisite Expression Tree

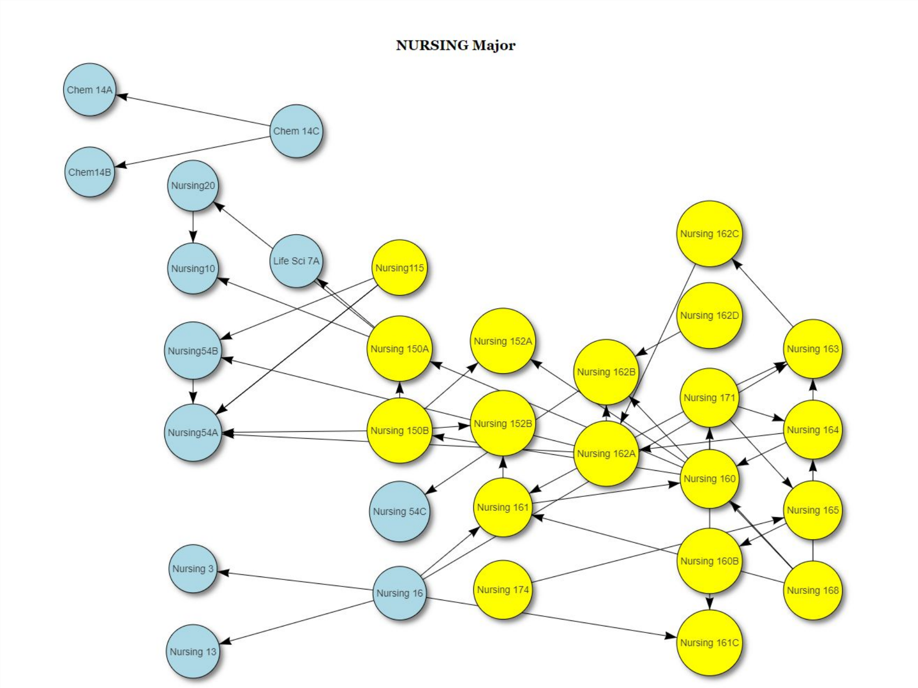


Interpreting Curriculum Tree

- **Yellow** means a class is a upper division class
- **Blue** means the class is an lower division class
- **Arrows** mean the previous class is a prerequisite for the next class

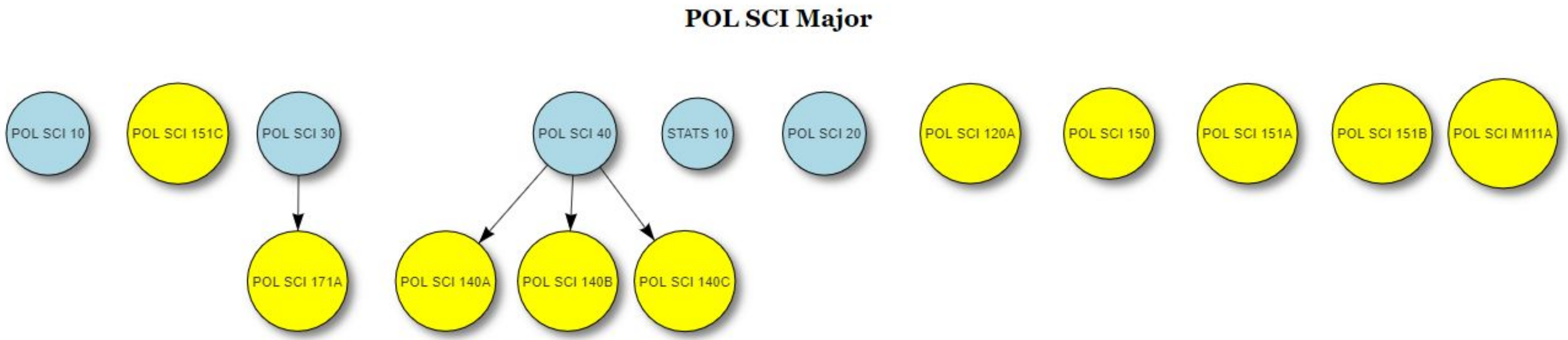
Nursing Major

Curriculum Complexity:
17.93



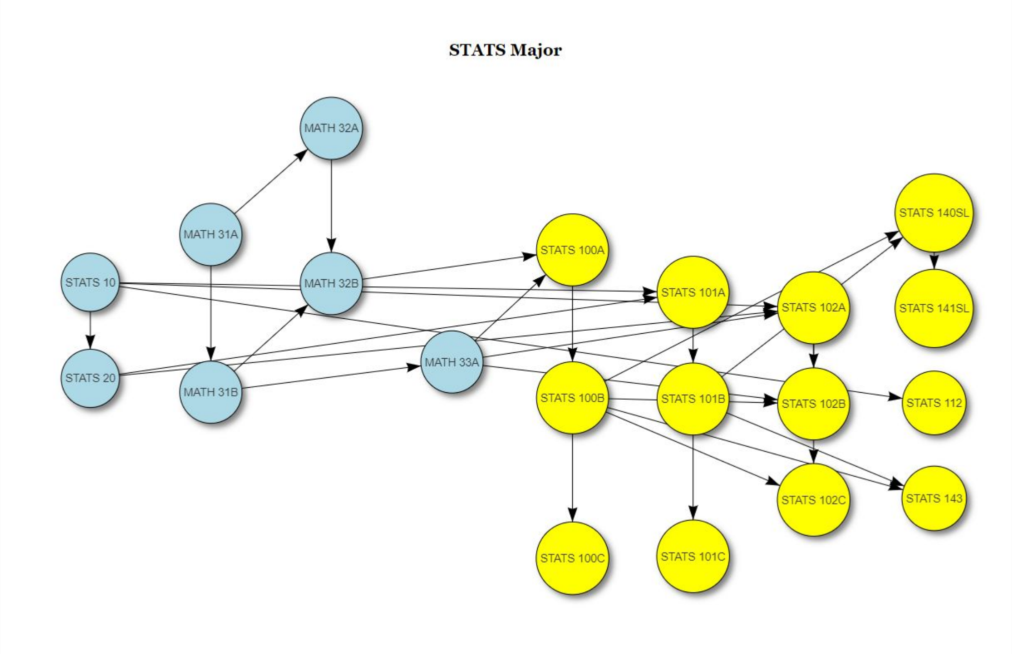
Political Science Major

Curriculum Complexity: 7.96



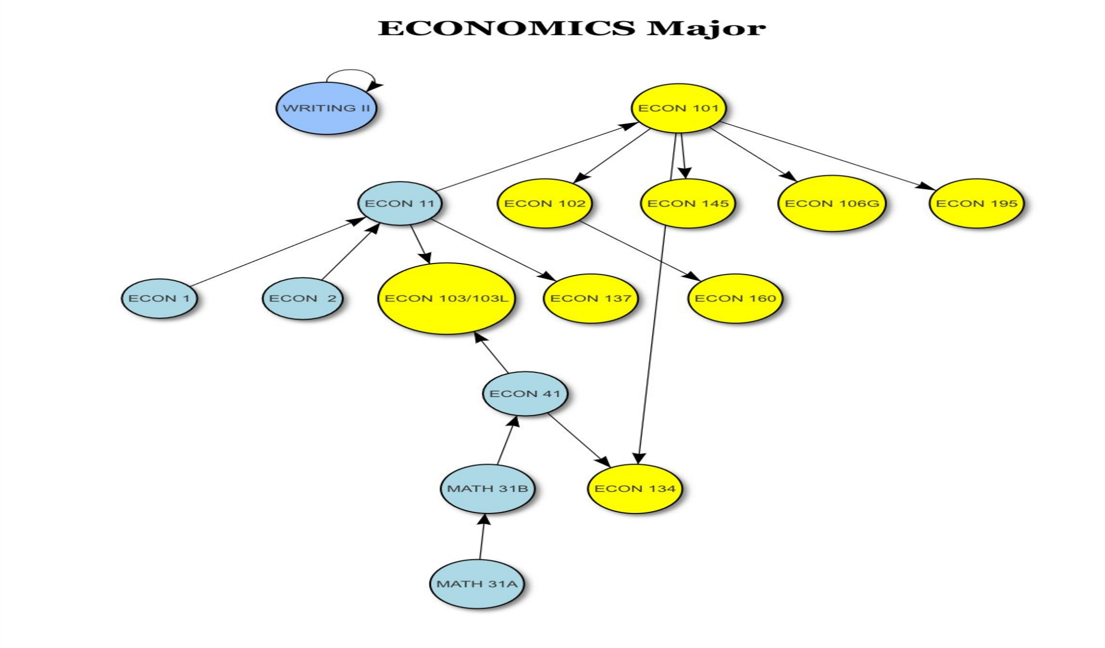
Statistics Major

Curriculum Complexity: 10.66



Economics Major

Curriculum Complexity: 8.39



Web Scraping (Pure Math)

```
[[1]]
 [1] "MATH 31A or MATH 31AL"          "MATH 31B"                       "MATH 32A"
 [4] "MATH 32B"                       "MATH 33A"                       "MATH 33B"
 [7] "PHYSICS 1A"                     "COMPTNG 10A"                    "and two courses from  CHEM 20A"
[10] "CHEM 20B"                       "ECON 11"                        "LIFESCI 7A"
[13] "PHILOS 31"                      "PHILOS 132"                     "PHYSICS 1B"
[16] "PHYSICS 1C"                     "PHYSICS 5B"                     "PHYSICS 5C"

[[2]]
[1] "MATH 110A"
[2] "MATH 110B"
[3] "MATH 115A"
[4] "MATH 120A"
[5] "MATH 131A"
[6] "MATH 131B"
[7] "MATH 132"
[8] "and at least five elective courses from MATH 106 through MATH 199 and  STATS 100A through STATS 102C"
```

Web Scraping (Applied Math)

```
[[1]]
 [1] "MATH 31A or MATH 31AL"        "MATH 31B"                   "MATH 32A"
 [4] "MATH 32B"                     "MATH 33A"                   "MATH 33B"
 [7] "PHYSICS 1A"                   "PHYSICS 1B"                 "COMPTNG 10A"
[10] "and one course from  CHEM 20A" "CHEM 20B"                  "PHYSICS 1C"

[[2]]
[1] "MATH 115A"


[2] "MATH 131A"


[3] "either MATH 131B or MATH 132"


[4] "MATH 142; two two-term sequences from two of the following categories: numerical analysis - courses MATH
 151A and MATH 151B"

[5] "probability and statistics - courses MATH 170A and MATH 170B"


[6] "or  STATS 100A and STATS 100B"


[7] "differential equations - courses STATS 134 and STATS 135; four courses from STATS 106 through STATS 199 a
nd  STATS 100A through STATS 102C (appropriate courses from other departments may be substituted for some of t
he additional courses provided departmental consent is given before such courses are taken)"
```

Unfortunately, human inconsistencies make machine-readable text very difficult.

Looking Ahead

Naive Recursion

```
> in_class("A&O SCI 90")
 [1] "C&EE M20"     "COMPTNG 10A" "EPS SCI 71"  "LIFESCI 30A" "LIFESCI 30B" "MATH 1"      "MATH 31A"
 [8] "MATH 31B"     "MATH 32A"    "MATH 33A"    "MATH 3A"     "MATH 3B"     "MATH 3C"     "PHYSICS 1A"
[15] "PHYSICS 5A"   "PHYSICS 6A"
```

```
> in_class_print("MATH 33A")
MATH 31B
MATH 31A
MATH 1
MATH 32A
MATH 31A
MATH 1
MATH 3B
MATH 3A
MATH 1
> in_class("MATH 33A")
[1] "MATH 1"   "MATH 31A" "MATH 31B" "MATH 32A" "MATH 3A"  "MATH 3B"
```

We need to handle OR statements more effectively. Currently, we can only list all possible requisites.

Looking Ahead

Proper recursion will allow us to best gauge course complexity:

1. Find the longest/shortest paths in any given major
2. Apply potential transformations at each expression node (e.g., treating “OR groups” as less complex by using an inverse log-choose multiplier)
3. Connect to other course-level information, such as number of units, pass rate, class size, etc.