Dimensions Documentation

MEIZHU WANG, EDGAR A. BERNAL

All data comes from Dimension API (https://docs.dimensions.ai/dsl/api.html). The project is aimed at understanding current research patterns and trends, strengths and weaknesses of different research institutions in the USA and worldwide, as well as predicting future trends.

IDE: *Python 3.7.2*
Data set: *Dimensions (https://app.dimensions.ai/discover/publication)*

1. Field of Research (FoR) counts

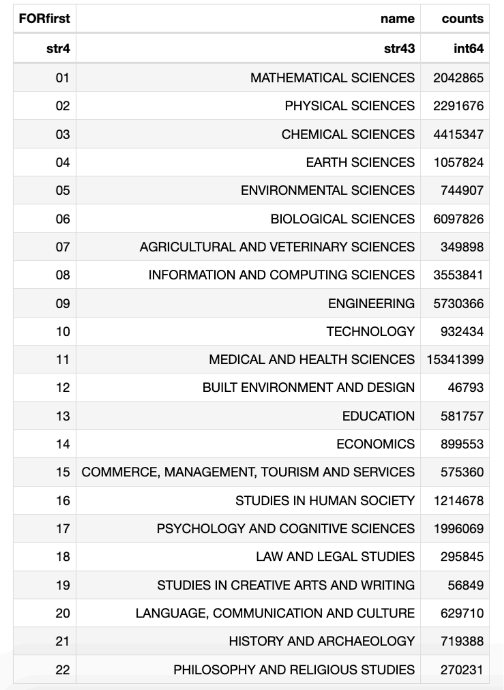
To analyze trends in different research fields, we adopted FoR codes (Australian and New Zealand Standard Research Classification (ANZSRC), 2008) that break down research fields into categories. According to this scheme, there are 22 two-digit FoRs ranging from 01 to 22. Each of the 22 FoRs has several sub categories coded with two additional digits starting with 01, and appended to the two-digit code. Different FoRs may have different numbers of sub FoRs. The last sub-FoR, which usually includes sub-fields not covered by previous sub-FoRs, is coded by adding 99. The code in the following Python notebook summarizes the number of papers in each FoR and sub-FoR:

— FoRcounts.ipynb

Table 1 shows counts of 22 FoRs directly from the API. Table 2 (of length 201), apart from copying counts of FoRs from first table, which are labeled as direct query, there are sub-FoR counts and summation of sub-FoR counts as counts of two digit FoRs labeled by FoR text names. These two tables are saved in the following files.

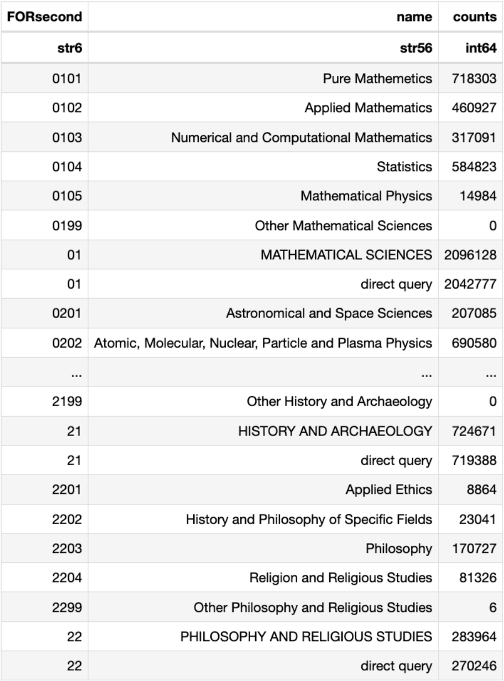
— FoR_first_counts.csv
— FoRcounts.csv

Table 1



FORfirst	name	counts
str4	str43	int64
01	MATHEMATICAL SCIENCES	2042865
02	PHYSICAL SCIENCES	2291676
03	CHEMICAL SCIENCES	4415347
04	EARTH SCIENCES	1057824
05	ENVIRONMENTAL SCIENCES	744907
06	BIOLOGICAL SCIENCES	6097826
07	AGRICULTURAL AND VETERINARY SCIENCES	349898
08	INFORMATION AND COMPUTING SCIENCES	3553841
09	ENGINEERING	5730366
10	TECHNOLOGY	932434
11	MEDICAL AND HEALTH SCIENCES	15341399
12	BUILT ENVIRONMENT AND DESIGN	46793
13	EDUCATION	581757
14	ECONOMICS	899553
15	COMMERCE, MANAGEMENT, TOURISM AND SERVICES	575360
16	STUDIES IN HUMAN SOCIETY	1214678
17	PSYCHOLOGY AND COGNITIVE SCIENCES	1996069
18	LAW AND LEGAL STUDIES	295845
19	STUDIES IN CREATIVE ARTS AND WRITING	56849
20	LANGUAGE, COMMUNICATION AND CULTURE	629710
21	HISTORY AND ARCHAEOLOGY	719388
22	PHILOSOPHY AND RELIGIOUS STUDIES	270231

Table 2



FORsecond	name	counts
str6	str56	int64
0101	Pure Mathemetics	718303
0102	Applied Mathematics	460927
0103	Numerical and Computational Mathematics	317091
0104	Statistics	584823
0105	Mathematical Physics	14984
0199	Other Mathematical Sciences	0
01	MATHEMATICAL SCIENCES	2096128
01	direct query	2042777
0201	Astronomical and Space Sciences	207085
0202	Atomic, Molecular, Nuclear, Particle and Plasma Physics	690580
...	...	...
2199	Other History and Archaeology	0
21	HISTORY AND ARCHAEOLOGY	724671
21	direct query	719388
2201	Applied Ethics	8864
2202	History and Philosophy of Specific Fields	23041
2203	Philosophy	170727
2204	Religion and Religious Studies	81326
2299	Other Philosophy and Religious Studies	6
22	PHILOSOPHY AND RELIGIOUS STUDIES	283964
22	direct query	270246

The difference between the counts of FoRs from different sources is due to some papers falling into multiple sub-FoRs within the same FoR: the number resulting from the summation is usually larger than that obtained via direct query.

Additionally, the following files are used or created in this notebook:

— *FoRtext.rtf*
— *Forbig*
— *Forsmall*
— *Menu*

FoRtext.rtf is the raw data downloaded from the website for performing the mapping between FoR codes and their related text names. *Forbig* contains all FoR codes; *Forsmall* contains a dictionary with FoR codes as keys and sub FoR codes as values. *Menu* is a final mapping between all two-/four-digit codes and text names.

2. Publication trends for top 60 universities in the US

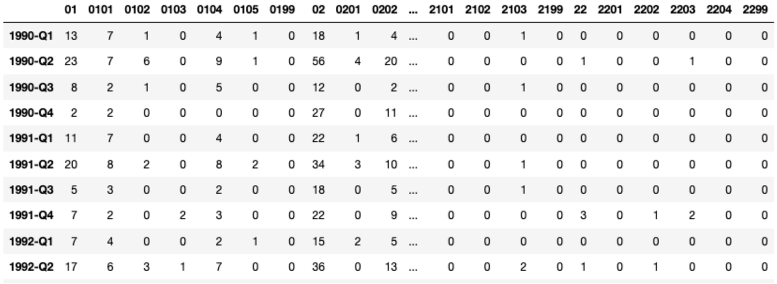
To analyze trends and patterns of all research fields listed in the above section, we first acquire time series data for each school representing their research productivity. To this end, we use the number of publications under FoR/sub FoRs in every quarter from 1990 to 2019. Code for this section can be found in the following notebook:

— *Publication trends for top 60 universities in the US.ipynb*

The schools chosen for analysis are those in the Top 60 list published by USNews. (University of Virginia is left out due to difficulty in querying it.) Schools are stored in *Schools* to be easily transferred to other notebooks for reuse.

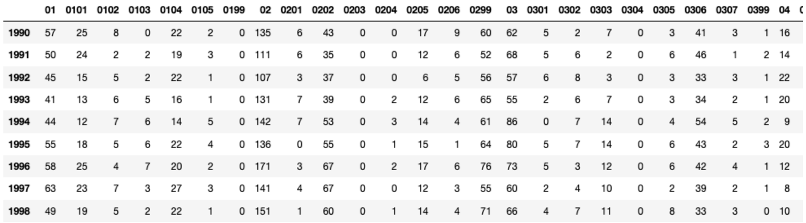
— *Schools*
— *paper dates for top60 schools*

This script queries dates and FoR/sub-FoRs of all publications from 1990 to 2019, then stores the raw data in *paper dates for top60 schools*. The number of papers per quarter/year under each FoR/sub-FoR code is counted locally, which results in four different Pandas Dataframes for each school. Take “University of Rochester” as an example,



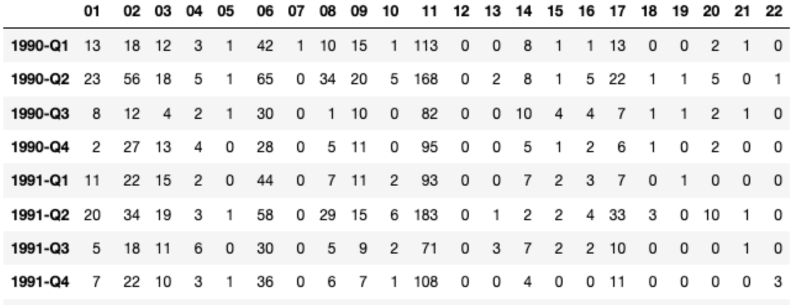
	01	0101	0102	0103	0104	0105	0199	02	0201	0202	...	2101	2102	2103	2199	22	2201	2202	2203	2204	2299
1990-Q1	13	7	1	0	4	1	0	18	1	4	...	0	0	1	0	0	0	0	0	0	0
1990-Q2	23	7	6	0	9	1	0	56	4	20	...	0	0	0	0	1	0	0	1	0	0
1990-Q3	8	2	1	0	5	0	0	12	0	2	...	0	0	1	0	0	0	0	0	0	0
1990-Q4	2	2	0	0	0	0	0	27	0	11	...	0	0	0	0	0	0	0	0	0	0
1991-Q1	11	7	0	0	4	0	0	22	1	6	...	0	0	0	0	0	0	0	0	0	0
1991-Q2	20	8	2	0	8	2	0	34	3	10	...	0	0	1	0	0	0	0	0	0	0
1991-Q3	5	3	0	0	2	0	0	18	0	5	...	0	0	1	0	0	0	0	0	0	0
1991-Q4	7	2	0	2	3	0	0	22	0	9	...	0	0	0	0	3	0	1	2	0	0
1992-Q1	7	4	0	0	2	1	0	15	2	5	...	0	0	0	0	0	0	0	0	0	0
1992-Q2	17	6	3	1	7	0	0	36	0	13	...	0	0	2	0	1	0	1	0	0	0

schooldict:
*120 quarters * 179 FoRs*



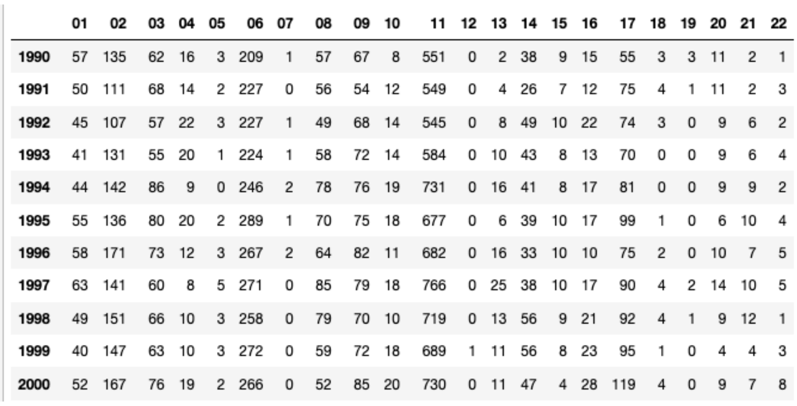
	01	0101	0102	0103	0104	0105	0199	02	0201	0202	0203	0204	0205	0206	0299	03	0301	0302	0303	0304	0305	0306	0307	0399	04
1990	57	25	8	0	22	2	0	135	6	43	0	0	17	9	60	62	5	2	7	0	3	41	3	1	16
1991	50	24	2	2	19	3	0	111	6	35	0	0	12	6	52	68	5	6	2	0	6	46	1	2	14
1992	45	15	5	2	22	1	0	107	3	37	0	0	6	5	56	57	6	8	3	0	3	33	3	1	22
1993	41	13	6	5	16	1	0	131	7	39	0	2	12	6	65	55	2	6	7	0	3	34	2	1	20
1994	44	12	7	6	14	5	0	142	7	53	0	3	14	4	61	86	0	7	14	0	4	54	5	2	9
1995	55	18	5	6	22	4	0	136	0	55	0	1	15	1	64	80	5	7	14	0	6	43	2	3	20
1996	58	25	4	7	20	2	0	171	3	67	0	2	17	6	76	73	5	3	12	0	6	42	4	1	12
1997	63	23	7	3	27	3	0	141	4	67	0	0	12	3	55	60	2	4	10	0	2	39	2	1	8
1998	49	19	5	2	22	1	0	151	1	60	0	1	14	4	71	66	4	7	11	0	8	33	3	0	10

schooldict2:
*30 years * 179 FoRs*



	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
1990-Q1	13	18	12	3	1	42	1	10	15	1	113	0	0	8	1	1	13	0	0	2	1	0
1990-Q2	23	56	18	5	1	65	0	34	20	5	168	0	2	8	1	5	22	1	1	5	0	1
1990-Q3	8	12	4	2	1	30	0	1	10	0	82	0	0	10	4	4	7	1	1	2	1	0
1990-Q4	2	27	13	4	0	28	0	5	11	0	95	0	0	5	1	2	6	1	0	2	0	0
1991-Q1	11	22	15	2	0	44	0	7	11	2	93	0	0	7	2	3	7	0	1	0	0	0
1991-Q2	20	34	19	3	1	58	0	29	15	6	183	0	1	2	2	4	33	3	0	10	1	0
1991-Q3	5	18	11	6	0	30	0	5	9	2	71	0	3	7	2	2	10	0	0	0	1	0
1991-Q4	7	22	10	3	1	36	0	6	7	1	108	0	0	4	0	0	11	0	0	0	0	3

schooldict3:
*120 quarters * 22 FoR_first*



	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
1990	57	135	62	16	3	209	1	57	67	8	551	0	2	38	9	15	55	3	3	11	2	1
1991	50	111	68	14	2	227	0	56	54	12	549	0	4	26	7	12	75	4	1	11	2	3
1992	45	107	57	22	3	227	1	49	68	14	545	0	8	49	10	22	74	3	0	9	6	2
1993	41	131	55	20	1	224	1	58	72	14	584	0	10	43	8	13	70	0	0	9	6	4
1994	44	142	86	9	0	246	2	78	76	19	731	0	16	41	8	17	81	0	0	9	9	2
1995	55	136	80	20	2	289	1	70	75	18	677	0	6	39	10	17	99	1	0	6	10	4
1996	58	171	73	12	3	267	2	64	82	11	682	0	16	33	10	10	75	2	0	10	7	5
1997	63	141	60	8	5	271	0	85	79	18	766	0	25	38	10	17	90	4	2	14	10	5
1998	49	151	66	10	3	258	0	79	70	10	719	0	13	56	9	21	92	4	1	9	12	1
1999	40	147	63	10	3	272	0	59	72	18	689	1	11	56	8	23	95	1	0	4	4	3
2000	52	167	76	19	2	266	0	52	85	20	730	0	11	47	4	28	119	4	0	9	7	8

schooldict4:
*30 years * 22 FoR_first*

These Dataframes are stored in the following files with schools as keys and one of the Dataframes from above as values,

— *schooldict*
— *schooldict2*
— *schooldict3*
— *schooldict4*

With any of the above Dataframesd=, we can quantitatively compare which schools are ahead or lagging behind other schools in a specific field by measuring the Spearman correlation between the time series resulting from concatenating the respective publication numbers across time. The reason that we adopted this metric is that it doesn't take the raw publication numbers for comparison (which, for example, may favor institutions with larger faculty bodies), and instead it pays attention to growth and trends within each time series by converting the paper counts to ranks inside of each series. This addresses the bias of comparing trends between institutes of different sizes, because large-sized institutes may tend to have larger publication numbers.

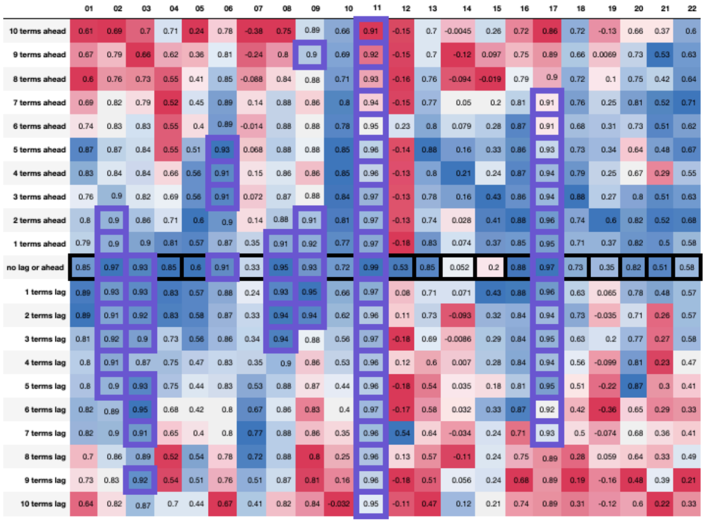
In one example involving *schooldict3* Dataframes for two institutions, we choose two time series in a given FoR from two schools of interest, then calculate correlation between the two vectors. The null assumption here is that if the two schools are at the same level in this field, the two unmodified vectors should have high correlation. If not, the correlation is not significant. In contrast, we conclude one school is ahead of the other when the correlation between the past history vector (e.g., obtained by shifting the history vector forward *n* time steps) of the school and the unmodified history vector of the second is higher than the first result. We assume the lag between the two schools corresponds to the value of *n* for which the highest correlation results; for visualization purposes, we display resulting correlation coefficients between the two schools for varying values of *n*. In addition to showing the correlation value, the cells corresponding to different lags are color-coded according to the degree of correlation: high correlation cells are indicated in dark blue, with the color code transitioning to light blue, then light and dark red as the correlation decreases, with dark red indicating lowest correlation cells.

The row with cells surrounded by black boundaries corresponds to the *n* = 0 lag computation (i.e., the 0-lag row), while the rows above/below correspond to negative/positive lag values. More specifically, in this case the correlation coefficient values in cells within black boundaries are calculated by using unmodified (i.e., with zero lag, or *n* = 0) time series for the 1990-2019 date range for both institutions, namely “University of Rochester” and the competing institute. The row, with row index 1 (i.e., with *n* = 1), contains the correlation coefficients between the 1991-2019 time series for “University of Rochester” and the 1990-2018 time series for the competing institute. Similarly, the row above uses the 1990-2018 time series for “University of Rochester” and the 1991-2019 for the competing institution (i.e., with *n* = -1).

The function *correlation* takes parameters *school, lag, advance, Pvalue, dictofdf, coefficient* and *p*, and produces the figures below. This function computes how well “University of Rochester” performs compared to another school of choice. The column indices are FoR/sub-FoRs (depending on which *Dataframe* is used) and the row indices indicate lag in years/quarters (again, depending on which *Dataframe* is used).

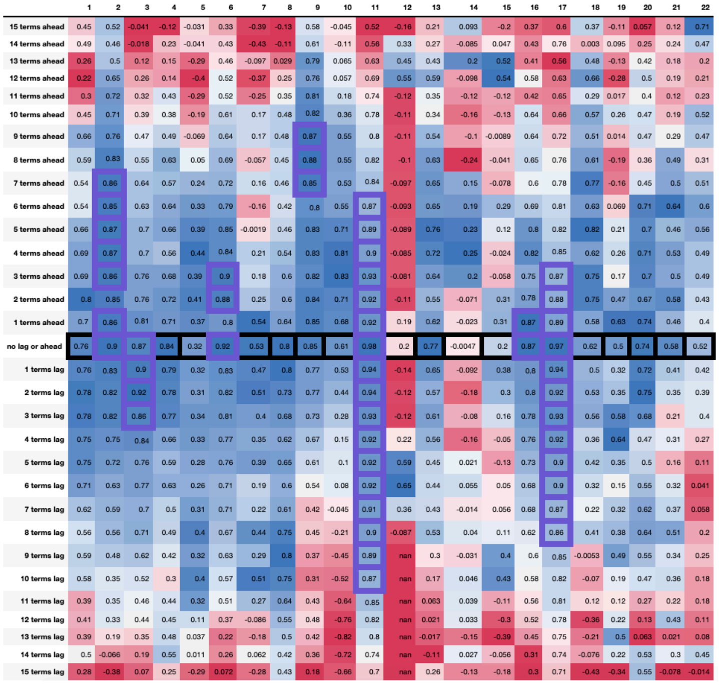
In the first table, we compare “University of Rochester” with “Duke University”, and in the second, we compare “University of Rochester” with “Brown University”.

U of R vs Duke University



	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
10 terms ahead	0.61	0.69	0.7	0.71	0.24	0.78	-0.38	0.75	0.89	0.66	0.91	-0.15	0.7	-0.0045	0.26	0.72	0.86	0.72	-0.13	0.66	0.37	0.6
9 terms ahead	0.67	0.79	0.66	0.62	0.36	0.81	-0.24	0.8	0.9	0.69	0.92	-0.15	0.7	-0.12	0.097	0.75	0.89	0.66	0.0069	0.73	0.53	0.63
8 terms ahead	0.6	0.76	0.73	0.55	0.41	0.85	-0.088	0.84	0.88	0.71	0.93	-0.16	0.76	-0.094	-0.019	0.79	0.9	0.72	0.1	0.75	0.42	0.64
7 terms ahead	0.69	0.82	0.79	0.52	0.45	0.89	0.14	0.88	0.86	0.8	0.94	-0.15	0.77	0.05	0.2	0.81	0.91	0.76	0.25	0.81	0.52	0.71
6 terms ahead	0.74	0.83	0.83	0.55	0.4	0.89	-0.014	0.88	0.88	0.78	0.95	0.23	0.8	0.079	0.28	0.87	0.91	0.68	0.31	0.73	0.51	0.62
5 terms ahead	0.87	0.87	0.84	0.55	0.51	0.93	0.068	0.88	0.88	0.85	0.96	-0.14	0.88	0.16	0.33	0.86	0.93	0.73	0.34	0.64	0.48	0.67
4 terms ahead	0.83	0.84	0.84	0.66	0.56	0.91	0.15	0.86	0.86	0.85	0.96	-0.13	0.8	0.21	0.24	0.87	0.94	0.79	0.25	0.67	0.29	0.55
3 terms ahead	0.76	0.9	0.82	0.69	0.56	0.91	0.072	0.87	0.88	0.84	0.97	-0.13	0.78	0.16	0.43	0.86	0.94	0.88	0.27	0.8	0.51	0.63
2 terms ahead	0.8	0.9	0.86	0.71	0.6	0.9	0.14	0.88	0.91	0.81	0.97	-0.13	0.74	0.028	0.41	0.88	0.96	0.74	0.6	0.82	0.52	0.68
1 terms ahead	0.79	0.9	0.9	0.81	0.57	0.87	0.35	0.91	0.92	0.77	0.97	-0.18	0.83	0.074	0.37	0.85	0.95	0.71	0.37	0.82	0.5	0.58
no lag or ahead	0.85	0.97	0.93	0.85	0.6	0.91	0.33	0.95	0.93	0.72	0.99	0.53	0.85	0.052	0.2	0.88	0.97	0.73	0.35	0.82	0.51	0.58
1 terms lag	0.89	0.93	0.93	0.83	0.57	0.88	0.24	0.93	0.95	0.66	0.97	0.08	0.71	0.071	0.43	0.88	0.96	0.63	0.065	0.78	0.48	0.57
2 terms lag	0.89	0.91	0.92	0.83	0.58	0.87	0.33	0.94	0.94	0.62	0.96	0.11	0.73	-0.093	0.32	0.84	0.94	0.73	-0.035	0.71	0.26	0.57
3 terms lag	0.81	0.92	0.9	0.73	0.56	0.86	0.34	0.94	0.88	0.56	0.97	-0.18	0.69	-0.0086	0.29	0.84	0.95	0.63	0.2	0.77	0.27	0.58
4 terms lag	0.8	0.91	0.87	0.75	0.47	0.83	0.35	0.9	0.86	0.53	0.96	0.12	0.6	0.007	0.28	0.84	0.94	0.56	-0.099	0.81	0.23	0.47
5 terms lag	0.8	0.9	0.93	0.75	0.44	0.83	0.53	0.88	0.87	0.44	0.96	-0.18	0.54	0.035	0.18	0.81	0.95	0.51	-0.22	0.87	0.3	0.41
6 terms lag	0.82	0.89	0.95	0.68	0.42	0.8	0.67	0.86	0.83	0.4	0.97	-0.17	0.58	0.032	0.33	0.87	0.92	0.42	-0.36	0.65	0.29	0.33
7 terms lag	0.82	0.9	0.91	0.65	0.4	0.8	0.77	0.88	0.86	0.35	0.96	0.54	0.64	-0.034	0.24	0.71	0.93	0.5	-0.074	0.68	0.36	0.41
8 terms lag	0.7	0.86	0.89	0.52	0.54	0.78	0.72	0.88	0.8	0.25	0.96	0.13	0.57	-0.11	0.24	0.75	0.89	0.28	0.059	0.64	0.33	0.49
9 terms lag	0.73	0.83	0.92	0.54	0.51	0.76	0.51	0.87	0.81	0.16	0.96	-0.18	0.51	0.056	0.24	0.68	0.89	0.19	-0.16	0.48	0.39	0.21
10 terms lag	0.64	0.82	0.87	0.7	0.44	0.67	0.41	0.82	0.84	-0.032	0.95	-0.11	0.47	0.12	0.21	0.74	0.89	0.31	-0.12	0.6	0.22	0.33

U of R vs Brown University



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
15 terms ahead	0.45	0.52	-0.041	-0.12	-0.031	0.33	-0.39	-0.13	0.58	-0.045	0.52	-0.16	0.21	0.093	-0.2	0.37	0.6	0.37	-0.11	0.057	0.12	0.71
14 terms ahead	0.49	0.46	-0.018	0.23	-0.041	0.43	-0.43	-0.11	0.61	-0.11	0.56	0.33	0.27	-0.085	0.047	0.43	0.76	0.003	0.095	0.25	0.24	0.47
13 terms ahead	0.26	0.5	0.12	0.15	-0.29	0.46	-0.097	0.029	0.79	0.065	0.63	0.45	0.43	0.2	0.52	0.41	0.56	0.48	-0.13	0.42	0.18	0.2
12 terms ahead	0.22	0.65	0.26	0.14	-0.4	0.52	-0.37	0.25	0.76	0.057	0.69	0.55	0.59	-0.098	0.54	0.58	0.63	0.66	-0.28	0.5	0.19	0.21
11 terms ahead	0.3	0.72	0.32	0.43	-0.29	0.52	-0.25	0.35	0.81	0.18	0.74	-0.12	0.35	-0.12	-0.12	0.42	0.65	0.29	0.017	0.4	0.12	0.23
10 terms ahead	0.45	0.71	0.39	0.38	-0.19	0.61	0.17	0.48	0.82	0.36	0.78	-0.11	0.34	-0.16	-0.13	0.64	0.66	0.57	0.26	0.47	0.19	0.52
9 terms ahead	0.66	0.76	0.47	0.49	-0.069	0.64	0.17	0.48	0.87	0.55	0.8	-0.11	0.54	-0.1	-0.0089	0.64	0.72	0.51	0.014	0.47	0.29	0.47
8 terms ahead	0.59	0.83	0.55	0.63	0.05	0.69	-0.057	0.45	0.88	0.55	0.82	-0.1	0.63	-0.24	-0.041	0.65	0.76	0.61	-0.19	0.36	0.49	0.31
7 terms ahead	0.54	0.86	0.64	0.57	0.24	0.72	0.16	0.46	0.85	0.53	0.84	-0.097	0.65	0.15	-0.078	0.6	0.78	0.77	-0.16	0.45	0.5	0.51
6 terms ahead	0.54	0.85	0.63	0.64	0.33	0.79	-0.16	0.42	0.8	0.55	0.87	-0.093	0.65	0.19	0.29	0.69	0.81	0.63	0.069	0.71	0.64	0.6
5 terms ahead	0.66	0.87	0.7	0.66	0.39	0.85	-0.0019	0.46	0.83	0.71	0.89	-0.089	0.76	0.23	0.12	0.8	0.82	0.82	0.21	0.7	0.46	0.56
4 terms ahead	0.69	0.87	0.7	0.56	0.44	0.84	0.21	0.54	0.83	0.81	0.9	-0.085	0.72	0.25	-0.024	0.82	0.85	0.62	0.26	0.71	0.53	0.49
3 terms ahead	0.69	0.86	0.76	0.68	0.39	0.9	0.18	0.6	0.82	0.83	0.93	-0.081	0.64	0.2	-0.058	0.75	0.87	0.75	0.17	0.7	0.5	0.49
2 terms ahead	0.8	0.85	0.76	0.72	0.41	0.88	0.25	0.6	0.84	0.71	0.92	-0.11	0.55	-0.071	0.31	0.78	0.88	0.75	0.47	0.67	0.58	0.43
1 terms ahead	0.7	0.86	0.81	0.71	0.37	0.8	0.54	0.64	0.85	0.68	0.92	0.19	0.62	-0.023	0.31	0.87	0.89	0.58	0.63	0.74	0.46	0.4
no lag or ahead	0.76	0.9	0.87	0.84	0.32	0.92	0.53	0.8	0.85	0.61	0.98	0.2	0.77	-0.0047	0.2	0.87	0.97	0.62	0.5	0.74	0.58	0.52
1 terms lag	0.76	0.83	0.9	0.79	0.32	0.83	0.47	0.8	0.77	0.53	0.94	-0.14	0.65	-0.092	0.38	0.8	0.94	0.5	0.32	0.72	0.41	0.42
2 terms lag	0.78	0.82	0.92	0.78	0.31	0.82	0.51	0.73	0.77	0.44	0.94	-0.12	0.57	-0.18	0.3	0.8	0.92	0.53	0.35	0.75	0.35	0.39
3 terms lag	0.78	0.82	0.86	0.77	0.34	0.81	0.4	0.68	0.73	0.28	0.93	-0.12	0.61	-0.08	0.16	0.78	0.93	0.56	0.58	0.68	0.21	0.4
4 terms lag	0.75	0.75	0.84	0.66	0.33	0.77	0.35	0.62	0.67	0.15	0.92	0.22	0.56	-0.16	-0.05	0.76	0.92	0.36	0.64	0.47	0.31	0.27
5 terms lag	0.75	0.72	0.76	0.59	0.28	0.76	0.39	0.65	0.61	0.1	0.92	0.59	0.45	0.021	-0.13	0.73	0.9	0.42	0.35	0.5	0.16	0.11
6 terms lag	0.71	0.63	0.77	0.63	0.26	0.71	0.19	0.6	0.54	0.08	0.92	0.65	0.44	0.055	0.05	0.68	0.9	0.32	0.15	0.55	0.32	0.041
7 terms lag	0.62	0.59	0.7	0.5	0.31	0.71	0.22	0.61	0.42	-0.045	0.91	0.36	0.43	-0.014	0.056	0.68	0.87	0.22	0.32	0.62	0.37	0.058
8 terms lag	0.56	0.56	0.71	0.49	0.4	0.67	0.44	0.75	0.45	-0.21	0.9	-0.087	0.53	0.04	0.11	0.62	0.86	0.41	0.38	0.64	0.51	0.2
9 terms lag	0.59	0.48	0.62	0.42	0.32	0.63	0.29	0.8	0.37	-0.45	0.89	nan	0.3	-0.031	0.4	0.6	0.85	-0.0053	0.49	0.55	0.34	0.25
10 terms lag	0.58	0.35	0.52	0.3	0.4	0.57	0.51	0.75	0.31	-0.52	0.87	nan	0.17	0.046	0.43	0.58	0.82	-0.07	0.19	0.47	0.36	0.18
11 terms lag	0.39	0.35	0.46	0.44	0.32	0.51	0.27	0.64	0.43	-0.64	0.85	nan	0.063	0.039	-0.11	0.56	0.81	0.12	0.12	0.27	0.22	0.18
12 terms lag	0.41	0.33	0.44	0.45	0.11	0.37	-0.086	0.55	0.48	-0.76	0.82	nan	0.021	0.033	-0.3	0.52	0.78	-0.36	0.22	0.13	0.43	0.11
13 terms lag	0.39	0.19	0.35	0.48	0.037	0.22	-0.18	0.5	0.42	-0.82	0.8	nan	-0.017	-0.15	-0.39	0.45	0.75	-0.21	0.5	0.063	0.021	0.08
14 terms lag	0.5	-0.066	0.19	0.55	0.011	0.26	0.062	0.42	0.36	-0.72	0.74	nan	-0.11	0.027	-0.056	0.31	0.74	-0.076	0.22	0.53	0.3	0.45
15 terms lag	0.28	-0.38	0.07	0.25	-0.29	0.072	-0.28	0.43	0.18	-0.66	0.7	nan	0.26	-0.13	-0.18	0.3	0.71	-0.43	-0.34	0.55	-0.078	-0.014

As mentioned, the darker the blue, the higher the correlation for that particular cell; similarly, correlation in blue cells is higher than correlation in red cells. In the second column of the first table, most blue cells are concentrated below the row with the black cells, which indicates that “University of Rochester” is several years behind “Duke University” in *FoR=2* (*Physical Sciences*); in contrast, in the second column of the second table, blue cells are more highly concentrated above the row with the black cells, indicating that “University of Rochester” is several years ahead of “Brown University” in the same FoR. Results from other FoR fields can be interpreted similarly. Generally, since there are more blue cells above the 0-lag row, we can conclude that the research level at “University of Rochester” is higher than that of “Brown University”, but lower than that of “Duke University”.

Parameters for Correlation Function:

— *School:* input one school from top 60 school list to be compared to *"University of Rochester"*.

— *Lag* and *Advance*: this parameter determines the range of lags to consider. Note that this term corresponds to the height of the table below and above the 0-lag row.

— *Dictofdf*: there are four choices, namely *schooldict, schooldict2, schooldict3, schooldict4*. The examples in this document are produced with the last choice, which corresponds to a configuration of 30 years by 22 FoRs. That is, only two-digit FoR trends at one year steps are considered. Different levels of granularity can be obtained with different choices for this parameter.

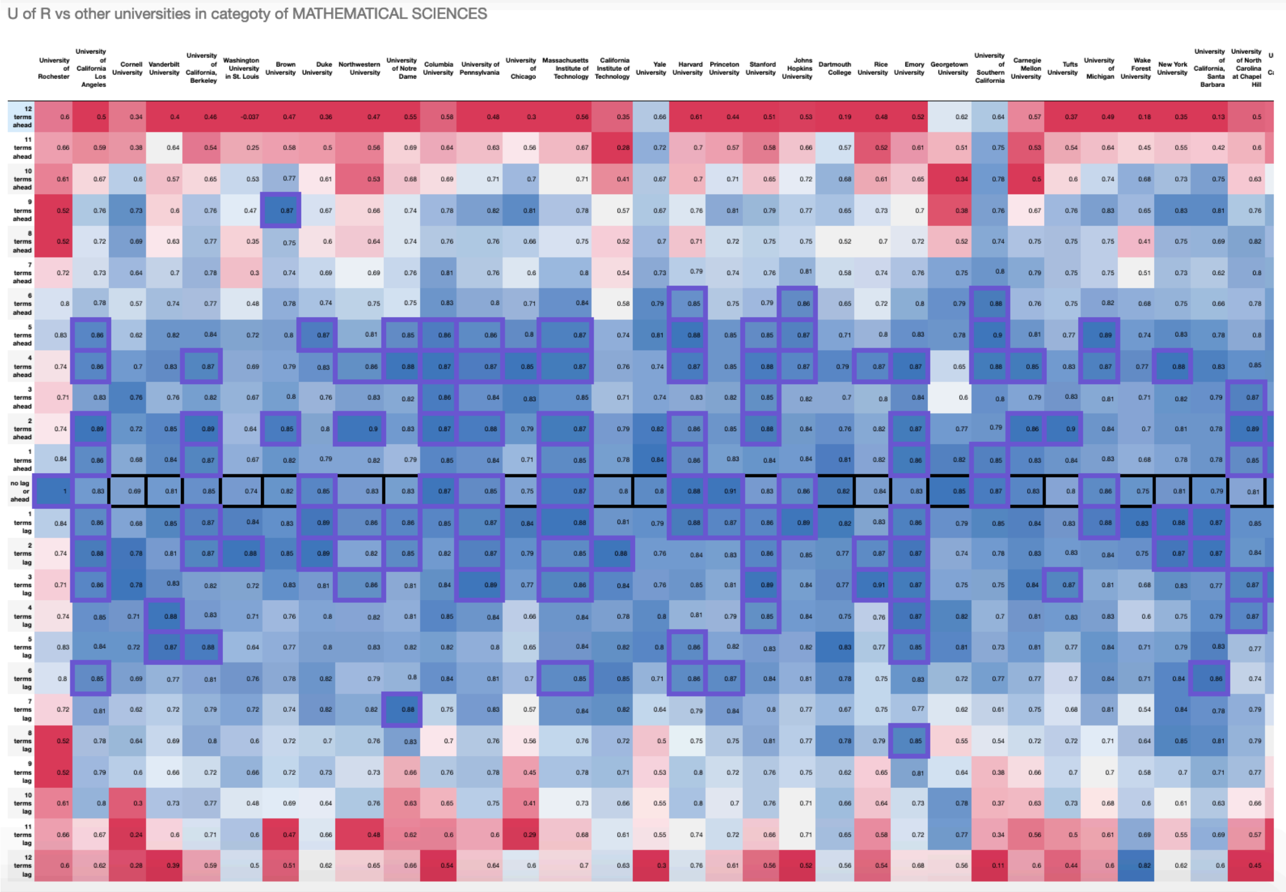
— *Pvalue*: this is a boolean variable. If *Pvalue=True*, the entries in the *Dataframes* will be *p* values. Otherwise, the values in the *Dataframes* will be coefficients. The higher the coefficient, the higher the correlation, and the lower the *p* value.

— *Coefficient* and *p*: these two parameters can be used to manually set the threshold that constitutes a high enough correlation to make a determination. When the cell entry exceeds this threshold, a purple border is added to the cells. The tables in this document are produced by using coefficients (not *p* value), and the selected threshold is *0.9*.

We note that the color scheme is normalized independently on a column-basis. In other words, for any given column, the cell colors will cover the full range from dark blue to dark red regardless of the absolute coefficient value. On the other hand, the thresholding operation is absolute, and equivalent across columns.

In summary, when comparing two schools, the higher the density of cells with purple borders or blue patterns, the higher the correlation computed for given values of lag. Presence of purple or blue cells above the 0-lag row indicates that "University of Rochester" is ahead of the competing school. The converse is also true in that presence of purple or blue cells below the 0-lag row indicates that "University of Rochester" is behind of the competing school.

Correlation2 compares*,* for a given FoR, "University of Rochester" to other schools. The row index and color codes are as before, while the column index indicates the particular competing school (this is because we are comparing across schools instead of FoRs). Both functions share all of the parameters but one: *school* becomes *FoR number*. As before, FoR and text names can be checked by using pickle file *menu*.


U of R vs other universities in categoty of MATHEMATICAL SCIENCES

3. Future Trend Prediction with Deep Learning

In order to evaluate how research patterns will evolve into the future, we predict future institutional behavior based on past observed behavior. To that end, we adopt a Long-short term memory (LSTM) based prediction model since LSTMs have proven effective at memorizing long- and short-term past, and quantifying temporal dependencies. The techniques from this section are implemented in the following notebook:

— *Time Series Prediction with deep learning.ipynb*

In order to train the model, we first separate the data into training and test data; only the training data is available during learning, while the test data is used to measure how good the model is at predicting new, previously unseen data. In this document, we use *schooldict3* for modeling, although use of other Dataframes is possible. We train a separate model for each school, the underlying assumption being that school behavior is independent across institutions. All implemented models comprise two-layer LSTM networks. Modeling details are as follows:

1. Format time series data into LSTM input (18 steps) and output (9 steps) sequence pairs, where each pair is a data point (either in the training or the test set). Time series are of length 120. The last four entries are from the four quarters of 2019. Since this portion of the data may not be reliable, it is eliminated from either training or testing for higher accuracy. For the remaining 116 time steps (a matrix of 116×22, one row per FoR field), the end of the input of the first data point occurs at time step 18, and the end of the input of the last pair occurs at time step 107 (116-9), which yields a total of 90 data points.

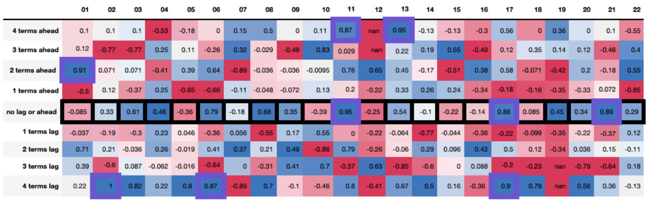
2. Split training and test data. The first 85 points are used as training data and last 5 as test data. After every iteration of training, we calculate MSE (mean square error) between predicted values of test data and real test data as a measure of how well the model has achieved so far. Training stops when the MSE doesn't improve significantly.

3. The output of this training process is stored in two files: *oldfuture* contains all predicted values for each school between 2016-Q4 and 2018-Q4 (i.e., the output of the last test data point). *Predgraph* contains the sequence of MSE values throughout the training process.

— *Oldfuture*
— *Predgraph*

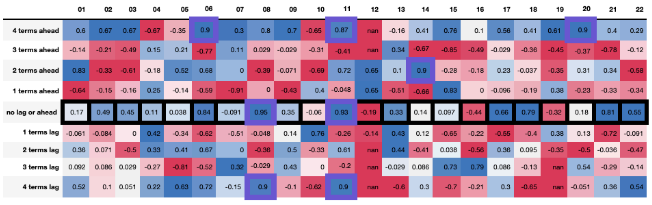
By applying *correlation function* to *oldftuture,* we are estimating future patterns of relative standing, that is, which school will be ahead or lagging in the future.

U of R vs Yale University



	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
4 terms ahead	0.1	0.1	0.1	-0.53	-0.18	0	0.15	0.5	0	0.11	0.87	nan	0.95	-0.13	-0.13	-0.3	0.56	0	0.36	0	0.1	-0.55
3 terms ahead	0.12	-0.77	-0.77	0.25	0.11	-0.26	0.32	-0.029	-0.49	0.83	0.029	nan	0.22	0.19	0.55	-0.49	0.12	0.35	0.14	0.12	-0.46	0.4
2 terms ahead	0.91	0.071	0.071	-0.41	0.39	0.64	-0.89	-0.036	-0.036	-0.0095	0.76	0.65	0.45	-0.17	-0.51	0.38	0.58	-0.071	-0.42	0.2	-0.18	0.55
1 terms ahead	-0.5	0.12	-0.37	0.25	-0.65	-0.66	-0.11	-0.048	-0.072	0.13	0.2	-0.22	0.33	0.26	0.24	-0.34	-0.18	-0.16	-0.35	-0.33	0.072	-0.85
no lag or ahead	-0.085	0.33	0.61	0.46	-0.36	0.79	-0.18	0.68	0.35	-0.39	0.95	-0.25	0.54	-0.1	-0.22	-0.14	0.86	0.085	0.45	0.34	0.89	0.29
1 terms lag	-0.037	-0.19	-0.3	0.23	0.046	-0.36	0.056	-0.55	0.17	0.55	0	-0.22	-0.064	-0.77	-0.044	-0.36	-0.22	-0.099	-0.35	-0.22	-0.37	0.12
2 terms lag	0.71	0.21	-0.036	0.26	-0.019	0.41	0.37	0.21	0.49	-0.86	0.79	-0.26	-0.06	0.29	0.096	0.42	0.5	0.12	-0.34	0.036	0.15	-0.11
3 terms lag	0.39	-0.6	0.087	-0.062	-0.016	-0.64	0	-0.31	0.41	0.7	-0.37	0.63	-0.85	-0.6	0	0.088	-0.2	-0.23	nan	-0.79	-0.64	0.18
4 terms lag	0.22	1	0.82	0.22	0.8	0.87	-0.89	0.7	-0.1	-0.46	0.8	-0.41	0.67	0.5	0.16	-0.36	0.9	0.79	nan	0.56	0.36	-0.13

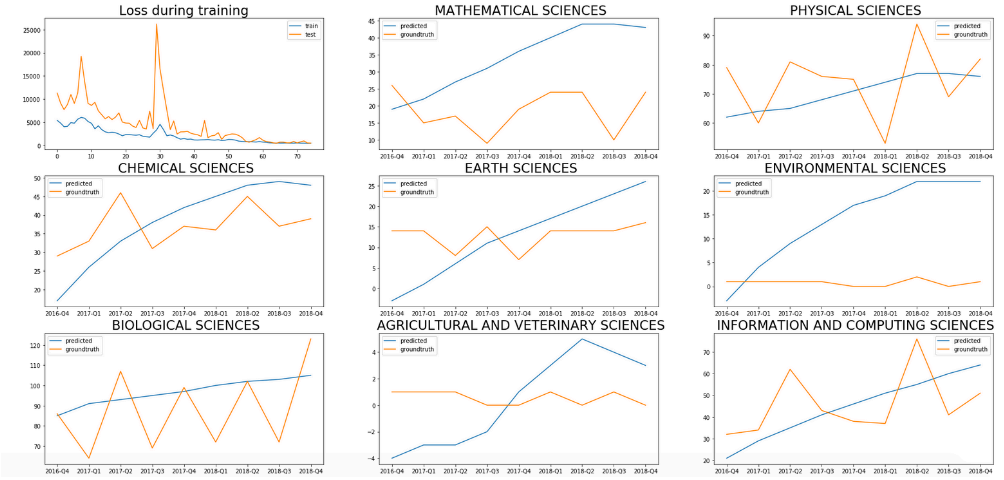
U of R vs Brown University



	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
4 terms ahead	0.6	0.67	0.67	-0.67	-0.35	0.9	0.3	0.8	0.7	-0.65	0.87	nan	-0.16	0.41	0.76	0.1	0.56	0.41	0.61	0.9	0.4	0.29
3 terms ahead	-0.14	-0.21	-0.49	0.15	0.21	-0.77	0.11	0.029	-0.029	-0.31	-0.41	nan	0.34	-0.67	-0.85	-0.49	-0.029	-0.36	-0.45	-0.37	-0.78	-0.12
2 terms ahead	0.83	-0.33	-0.61	-0.18	0.52	0.68	0	-0.39	-0.071	-0.69	0.72	0.65	0.1	0.9	-0.28	-0.18	0.23	-0.037	-0.35	0.31	0.34	-0.58
1 terms ahead	-0.64	-0.15	-0.16	0.25	-0.14	-0.59	-0.91	0	-0.43	0.4	-0.048	0.65	-0.51	-0.66	0.83	0	-0.096	-0.19	0.38	-0.23	-0.33	-0.34
no lag or ahead	0.17	0.49	0.45	0.11	0.038	0.84	-0.091	0.95	0.35	-0.06	0.93	-0.19	0.33	0.14	0.097	-0.44	0.66	0.79	-0.32	0.18	0.81	0.55
1 terms lag	-0.061	-0.084	0	0.42	-0.34	-0.62	-0.51	-0.048	0.14	0.76	-0.26	-0.14	0.43	0.12	-0.65	-0.22	-0.55	-0.4	0.38	0.13	-0.72	-0.091
2 terms lag	0.36	0.071	-0.5	0.33	0.41	0.67	0	-0.36	0.5	-0.33	0.61	nan	0.44	-0.41	0.038	-0.56	0.36	0.095	-0.35	-0.5	-0.036	-0.47
3 terms lag	0.092	0.086	0.029	-0.27	-0.81	-0.52	0.32	-0.029	0.43	0	-0.2	nan	-0.029	0.086	0.73	0.79	0.086	-0.13	nan	0.54	-0.29	-0.14
4 terms lag	0.52	0.1	0.051	0.22	0.63	0.72	-0.15	0.9	-0.1	-0.62	0.9	nan	-0.6	0.3	-0.7	-0.21	0.3	-0.65	nan	-0.051	0.36	0.54

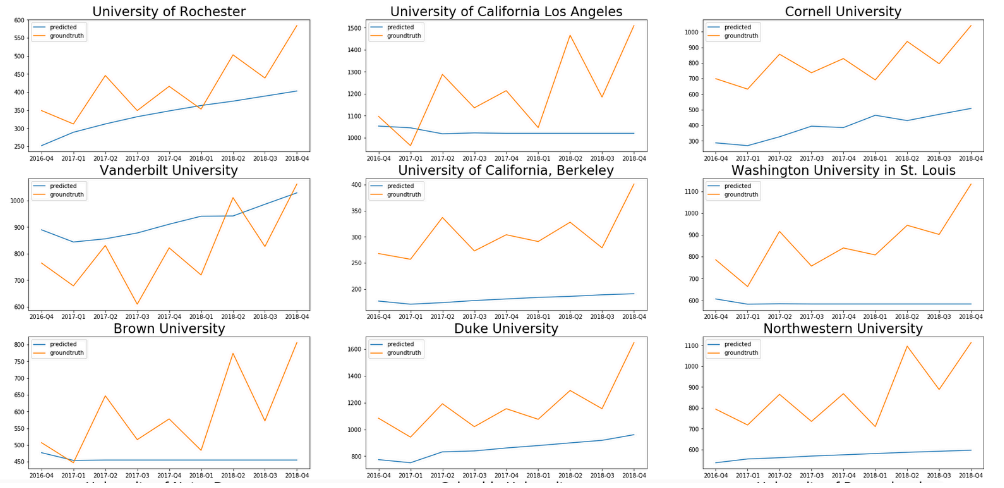
Futurepred and *futurepred2* are designed to see prediction processes and results*.* The former takes a school as input, while the latter takes an FoR. The figures below illustrate the output of *futurepred* after passing *"University of Rochester"* as parameter. The first plot is the MSE throughout the training process. The MSE keeps decreasing until around 70 iterations and stays level from that point on. The remaining plots show the outcome of *futurepred* for all 22 fields (14 are omitted due to space constraints), with the ground truth plotted in yellow and the predicted values in blue for the period 2016-Q4 to 2018-Q4.

University of Rochester



The figures below illustrate the output of *futurepred2* when passing *11* as its parameter. The plots show predicted performance compared to real observed values for different schools between 2016 Q4 and 2018_Q4 for *FoR=11 Medical and Health Sciences*.

MEDICAL AND HEALTH SCIENCES



4. Poster Presented at University Technology Showcase 2019

RDSC_Dimensions_poster.pptx

The poster summarized midterm results.

5. Grants to Publications Ratio

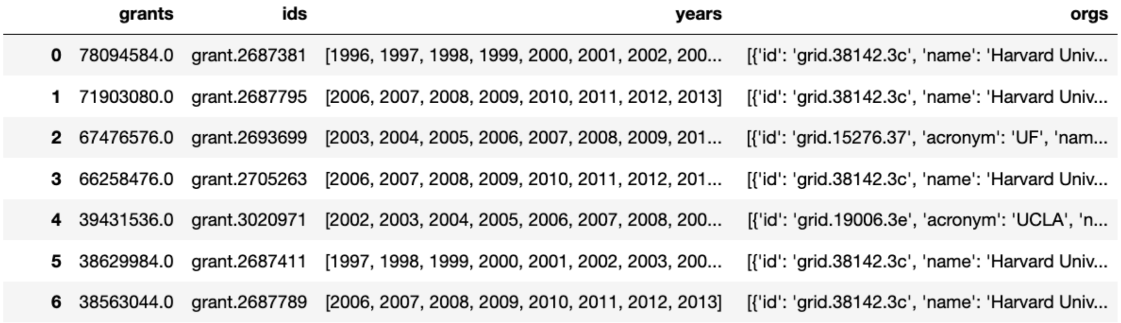
In this section we compute a measure of an institution's efficiency as estimated by the ratio between the grant amount and the publication number. The first half of the analysis focuses on how efficient each school is, and whether the size of the grant has an impact on efficiency. The second half of the analysis computes the efficiency across time, from 1998 to today. All results are included in the following notebook.

— *GrantsOverPublicationsRatio.ipynb*

In preparation for the analysis, data from grants relevant to target schools including grant amounts, receiving schools, grand IDs and active years from all fields are collected and stored in *Grants*. Then resulting papers are collected and stored in *Papers*. Both files are dictionaries of Pandas *Dataframes* with FoRs as keys.

— *Grants*
— *Papers*

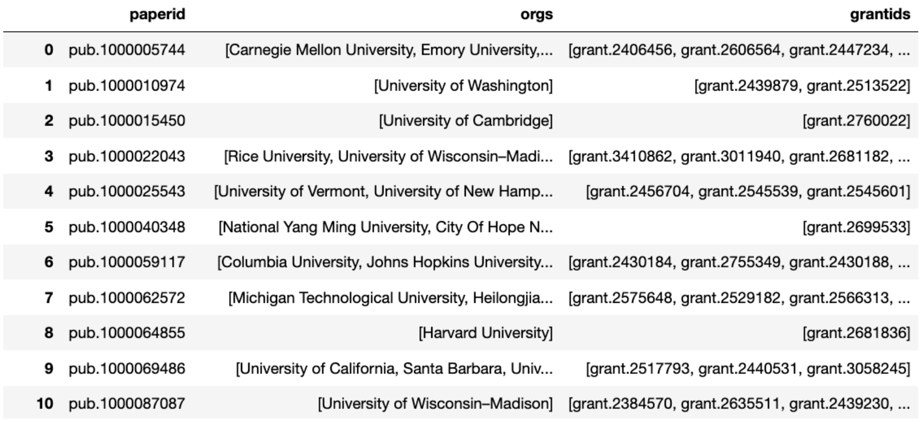
The raw data for grants (*FoR=1*) looks like the following:



	grants	ids	years	orgs
0	78094584.0	grant.2687381	[1996, 1997, 1998, 1999, 2000, 2001, 2002, 200...	[{'id': 'grid.38142.3c', 'name': 'Harvard Univ...
1	71903080.0	grant.2687795	[2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013]	[{'id': 'grid.38142.3c', 'name': 'Harvard Univ...
2	67476576.0	grant.2693699	[2003, 2004, 2005, 2006, 2007, 2008, 2009, 201...	[{'id': 'grid.15276.37', 'acronym': 'UF', 'nam...
3	66258476.0	grant.2705263	[2006, 2007, 2008, 2009, 2010, 2011, 2012, 201...	[{'id': 'grid.38142.3c', 'name': 'Harvard Univ...
4	39431536.0	grant.3020971	[2002, 2003, 2004, 2005, 2006, 2007, 2008, 200...	[{'id': 'grid.19006.3e', 'acronym': 'UCLA', 'n...
5	38629984.0	grant.2687411	[1997, 1998, 1999, 2000, 2001, 2002, 2003, 200...	[{'id': 'grid.38142.3c', 'name': 'Harvard Univ...
6	38563044.0	grant.2687789	[2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013]	[{'id': 'grid.38142.3c', 'name': 'Harvard Univ...

The first column (**grants**) shows grant money amount. In this case, the *Dataframe* is sorted by decreasing grant value, which means the 7 largest grants in *FoR=1 Mathematical Sciences* are being shown. The second column (**ids**) contains grant unique ID. It is the bridge between grants and papers. The third column (**years**) indicates the period when the grant is active. The last column (**orgs**) contains grant receiving research institutions.

The raw data for Papers (*FoR=1*) is illustrated below:



	paperid	orgs	grantids
0	pub.1000005744	[Carnegie Mellon University, Emory University,...	[grant.2406456, grant.2606564, grant.2447234, ...
1	pub.1000010974	[University of Washington]	[grant.2439879, grant.2513522]
2	pub.1000015450	[University of Cambridge]	[grant.2760022]
3	pub.1000022043	[Rice University, University of Wisconsin–Madi...	[grant.3410862, grant.3011940, grant.2681182, ...
4	pub.1000025543	[University of Vermont, University of New Hamp...	[grant.2456704, grant.2545539, grant.2545601]
5	pub.1000040348	[National Yang Ming University, City Of Hope N...	[grant.2699533]
6	pub.1000059117	[Columbia University, Johns Hopkins University...	[grant.2430184, grant.2755349, grant.2430188, ...
7	pub.1000062572	[Michigan Technological University, Heilongjia...	[grant.2575648, grant.2529182, grant.2566313, ...
8	pub.1000064855	[Harvard University]	[grant.2681836]
9	pub.1000069486	[University of California, Santa Barbara, Univ...	[grant.2517793, grant.2440531, grant.3058245]
10	pub.1000087087	[University of Wisconsin–Madison]	[grant.2384570, grant.2635511, grant.2439230, ...

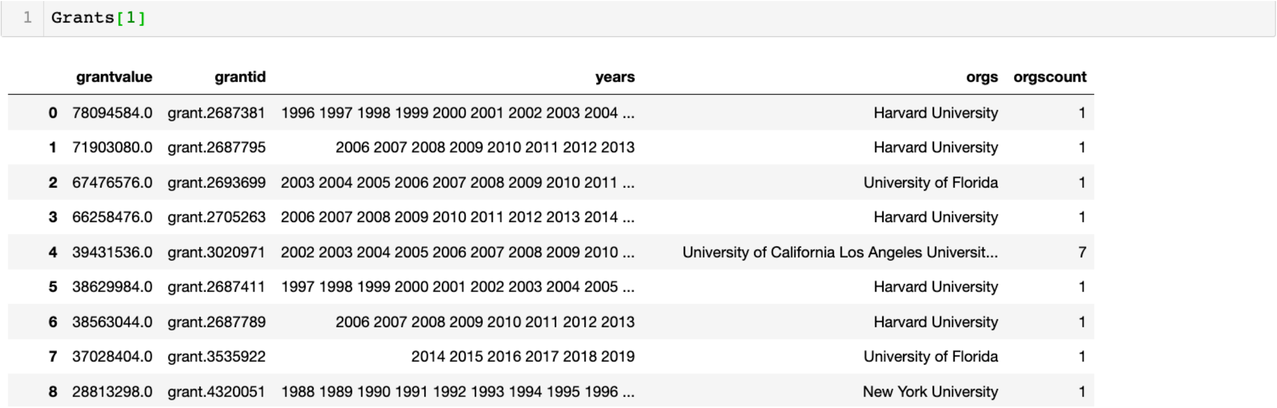
The first column (**paperid**) contains unique paper IDs. The second column (**orgs**) contains the co-authorizing research organizations. The third column (**grantids**) contains supporting grant IDs, which can be used to connect to *Grants*.

There may be several grants and several schools associated with a given paper. The exact connection between which grant(s) supported which school and what portion of each grant is distributed to each school are not fully known due to confidentiality reasons. Grants may support more than one paper as well, in which case the portions assigned to different papers are also unknown. To calculate money efficiency for each school under a given FoR, the first assumption is that each grant is distributed equally among schools. Thus, we can use *Grants* to calculate how much money each school received. The second assumption is that each grant contributes equally to each paper, and that each of the supported schools in the grant had equal contribution. So we can use *Grants* and *Papers* together to calculate how many papers are output by schools. Then we can use ratios between input and output as an efficiency measure. Needless to say, there is room for improvement with regards to the granularity of the available data.

For convenience of *Dataframe* vectorization and ratio calculation, we convert the above two *Dataframes*, which are stored in *papers2* and *grants2*:

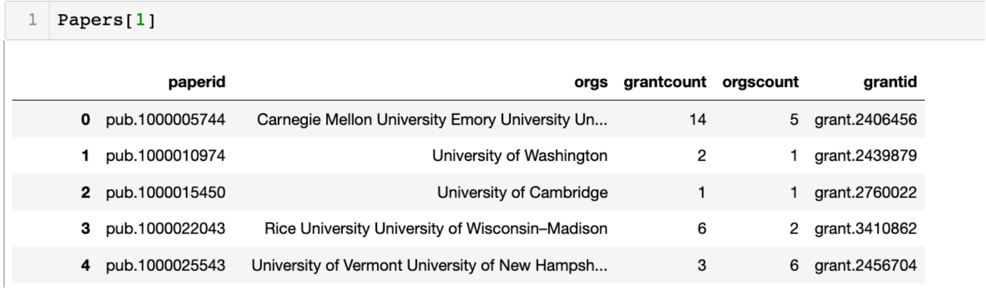
— *Grants2*
— *Papers2*

The **years** and **orgs** fields in *Grants* and *Papers* are converted to strings with spaces to facilitate string interpretation. In *Papers*, multiple supporting grants are separated and listed in individual rows repeating same paperid, orgs, grantcount and orgscount. Each paper is expanded to several rows depending on how many supporting grants it has. This enables fast processing by vectorization.



```
Grants[1]
```

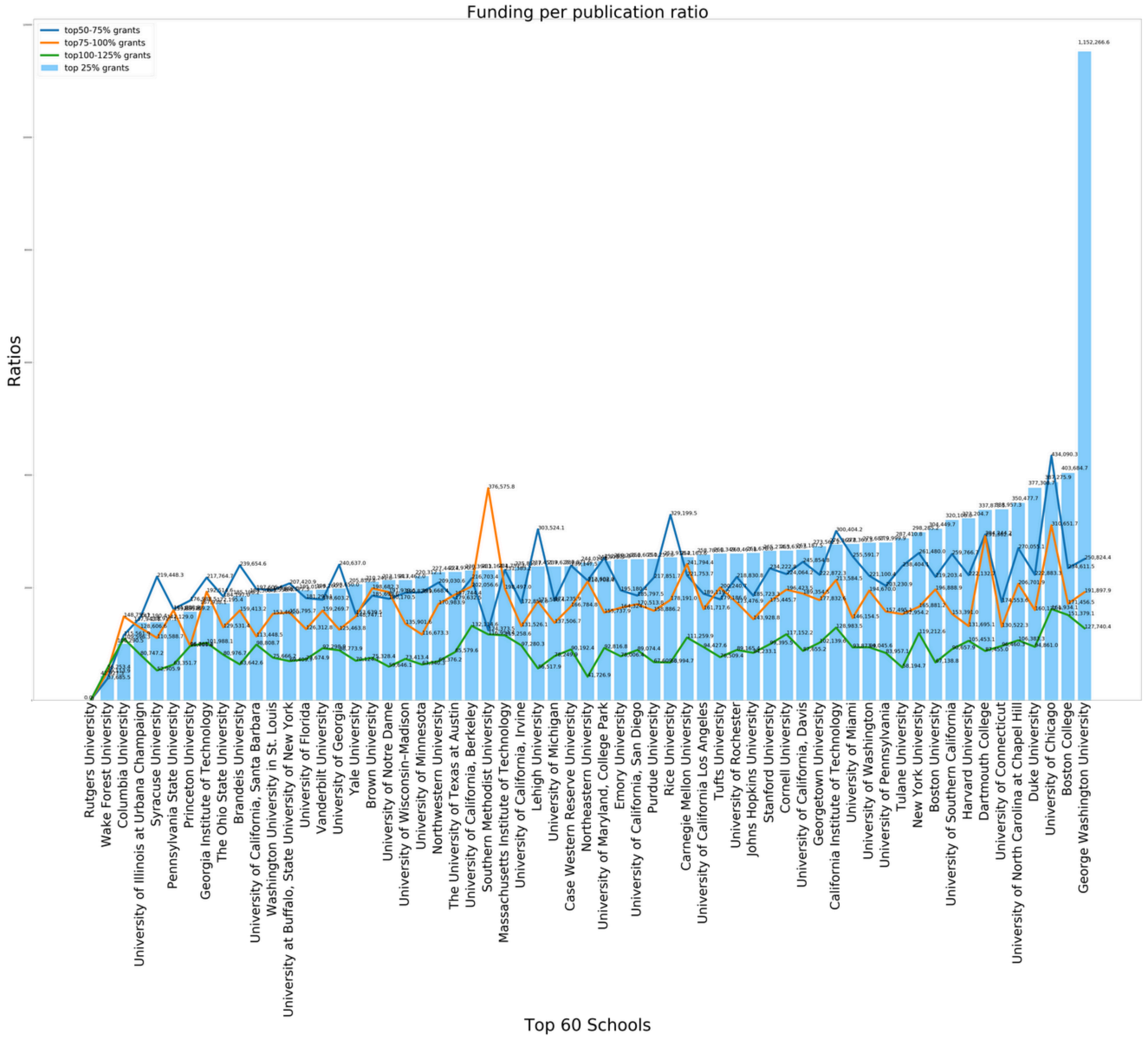
	grantvalue	grantid	years	orgs	orgscount
0	78094584.0	grant.2687381	1996 1997 1998 1999 2000 2001 2002 2003 2004 ...	Harvard University	1
1	71903080.0	grant.2687795	2006 2007 2008 2009 2010 2011 2012 2013	Harvard University	1
2	67476576.0	grant.2693699	2003 2004 2005 2006 2007 2008 2009 2010 2011 ...	University of Florida	1
3	66258476.0	grant.2705263	2006 2007 2008 2009 2010 2011 2012 2013 2014 ...	Harvard University	1
4	39431536.0	grant.3020971	2002 2003 2004 2005 2006 2007 2008 2009 2010 ...	University of California Los Angeles Universit...	7
5	38629984.0	grant.2687411	1997 1998 1999 2000 2001 2002 2003 2004 2005 ...	Harvard University	1
6	38563044.0	grant.2687789	2006 2007 2008 2009 2010 2011 2012 2013	Harvard University	1
7	37028404.0	grant.3535922	2014 2015 2016 2017 2018 2019	University of Florida	1
8	28813298.0	grant.4320051	1988 1989 1990 1991 1992 1993 1994 1995 1996 ...	New York University	1



```
Papers[1]
```

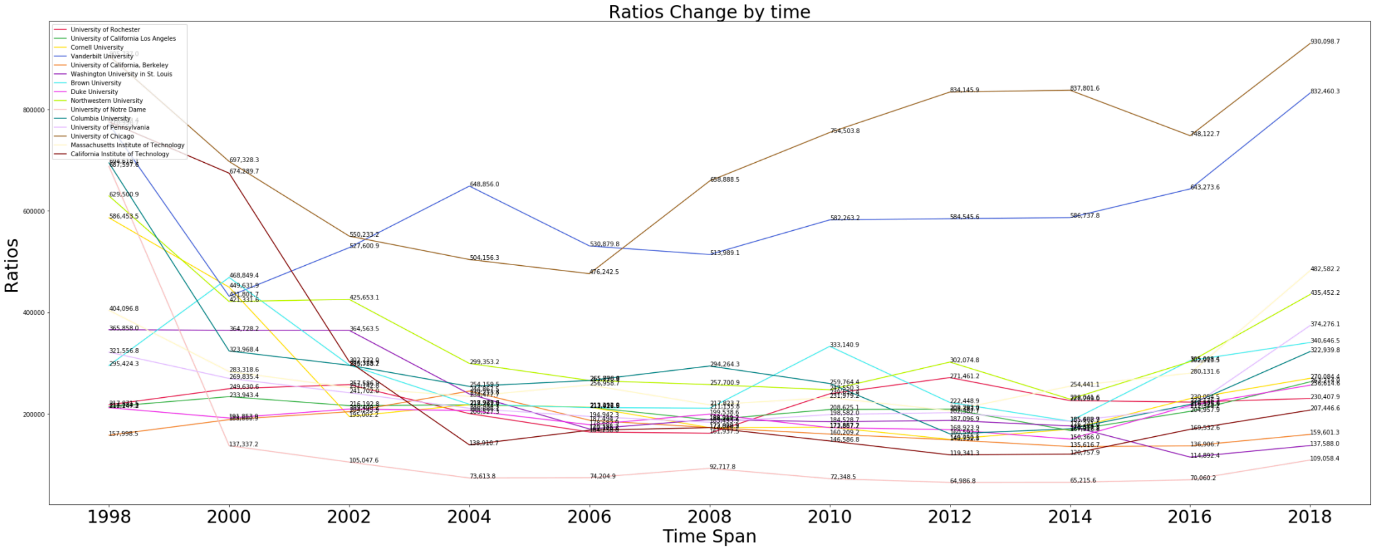
	paperid	orgs	grantcount	orgscount	grantid
0	pub.1000005744	Carnegie Mellon University Emory University Un...	14	5	grant.2406456
1	pub.1000010974	University of Washington	2	1	grant.2439879
2	pub.1000015450	University of Cambridge	1	1	grant.2760022
3	pub.1000022043	Rice University University of Wisconsin–Madison	6	2	grant.3410862
4	pub.1000025543	University of Vermont University of New Hampsh...	3	6	grant.2456704

Function *ratios_bygrant* takes a FoR number, and calculates the money/publication number ratio for each school in this given FoR. Grants are categorized by quartile: the top 25% (i.e., largest), the 25%-50% (i.e., the upper middle range), the 50%-75% (i.e., the lower middle range), and the 75%-100% (i.e., the smallest grants). The results for *FoR=11 Medical and Health Sciences* are shown below:



The ratios are labeled on each curve/column. Blue columns represent ratios for top 25% grants; while the blue/yellow/green curves represent ratios of upper middle/lower middle and smallest grants.

The function *ratios_byyear* also takes a FoR number as input, and calculates the evolution of ratios by year. Only selected (top 15) schools are plotted for clarity. The ratio for each year is calculated using data from grants that were active in the year and resulting papers. Each curve represents the performance of one school. Note that plotting and calculations for *FoR=12 /19* are missing due to data incompleteness. Results for *FoR=1* are shown below:



References

AUSTRALIAN AND NEW ZEALAND STANDARD RESEARCH CLASSIFICATION (ANZSRC), 2008:
HTTPS://WWW.ABS.GOV.AU/AUSSTATS/ABS@.NSF/LATESTPRODUCTS/6BB427AB9696C225CA2574180004463E?OPENDOCUMENT
Top 60 University in the US:
https://www.usnews.com/best-colleges/rankings/national-universities
Spearman correlation:
https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient
LSTM model:
https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/
Colors:
https://sashat.me/2017/01/11/list-of-20-simple-distinct-colors/

Index of Files/Functions/Dataframes/Parameters

File Name	Page Number
FoRcounts.ipynb	1
FoR_first_counts.csv	1
FoRcounts.csv	2
FoRtext.rtf	2
Forbig	2
Forsmall	2
Menu	2
Publication trends for top 60 universities in the US.ipynb	3
Schools	3
paper dates for top60 schools	3
schooldict	3
schooldict2	3
schooldict3	3
schooldict4	3
correlation	4

Notes on Reading Files without Extension

If a file has no extension, it is a Pickle file which can be read with the following set of commands:

```
import pickle
with open('filename', 'rb') as f:
   filename=pickle.load(f)
```