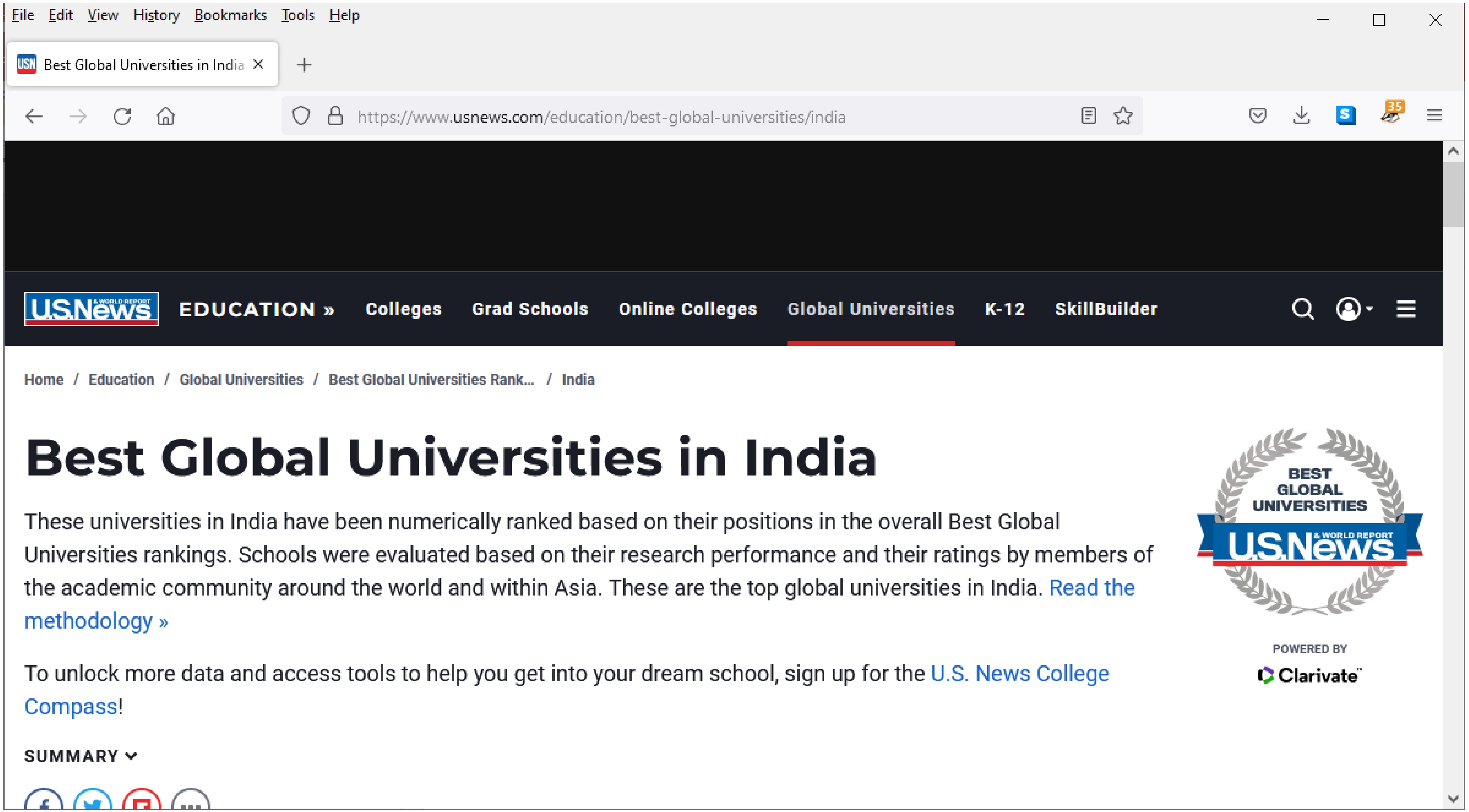
Ranking Analysis

CS5154/6054

Yizong Cheng

9/15/2022


File Edit View History Bookmarks Tools Help
Best Global Universities in India
https://www.usnews.com/education/best-global-universities/india
EDUCATION » Colleges Grad Schools Online Colleges Global Universities K-12 SkillBuilder
BEST GLOBAL UNIVERSITIES
POWERED BY Clarivate

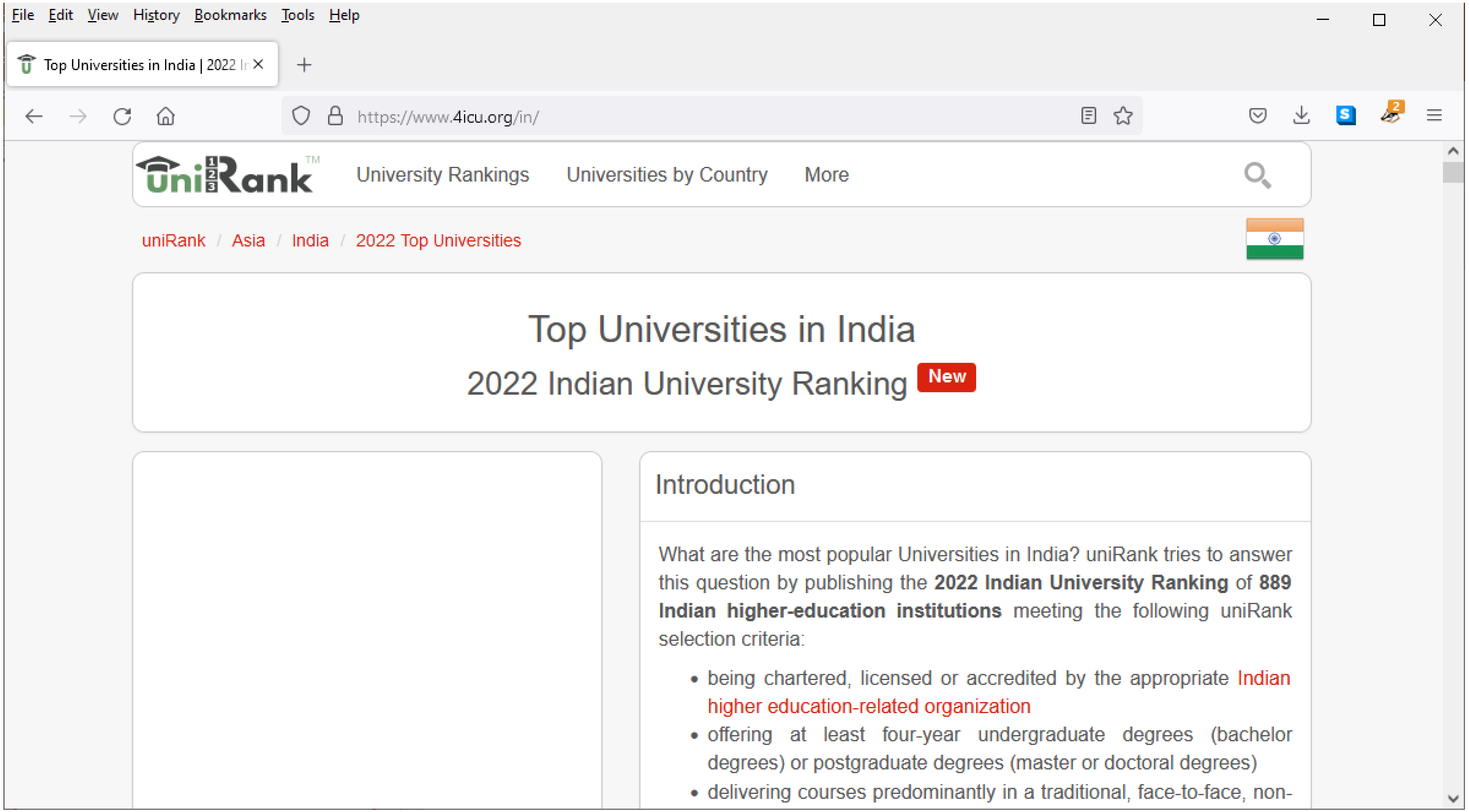
Home / Education / Global Universities / Best Global Universities Rank... / India

Best Global Universities in India

These universities in India have been numerically ranked based on their positions in the overall Best Global Universities rankings. Schools were evaluated based on their research performance and their ratings by members of the academic community around the world and within Asia. These are the top global universities in India. Read the methodology »

To unlock more data and access tools to help you get into your dream school, sign up for the U.S. News College Compass!

SUMMARY


File Edit View History Bookmarks Tools Help
Top Universities in India | 2022 I
https://www.4icu.org/in/
uniRank
University Rankings
Universities by Country
More

uniRank / Asia / India / 2022 Top Universities

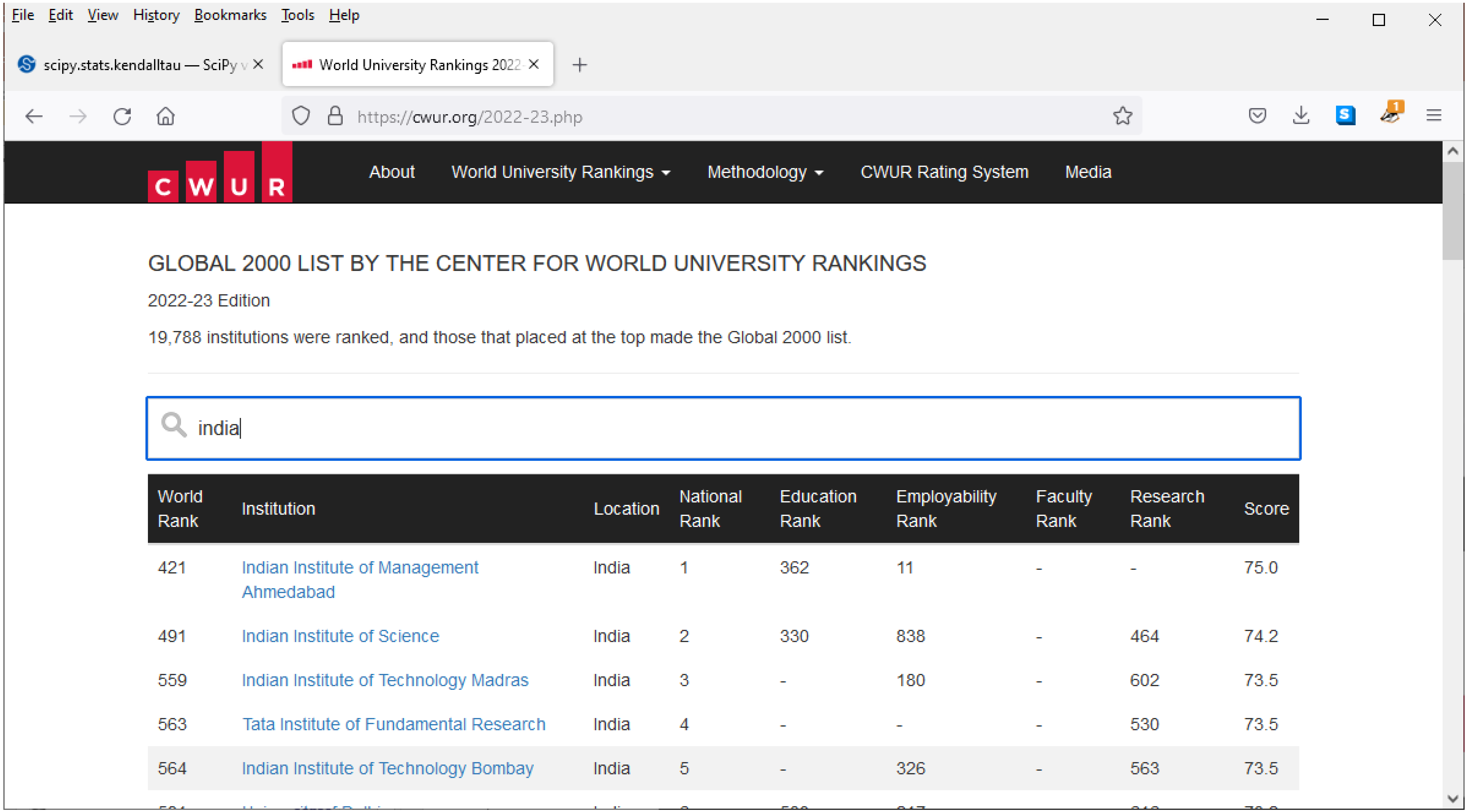
Top Universities in India

2022 Indian University Ranking New

Introduction

What are the most popular Universities in India? uniRank tries to answer this question by publishing the **2022 Indian University Ranking** of **889 Indian higher-education institutions** meeting the following uniRank selection criteria:

- being chartered, licensed or accredited by the appropriate Indian higher education-related organization
- offering at least four-year undergraduate degrees (bachelor degrees) or postgraduate degrees (master or doctoral degrees)
- delivering courses predominantly in a traditional, face-to-face, non-


File Edit View History Bookmarks Tools Help
scipy.stats.kendalltau — SciPy
World University Rankings 2022-
https://cwur.org/2022-23.php
CWUR
About
World University Rankings
Methodology
CWUR Rating System
Media

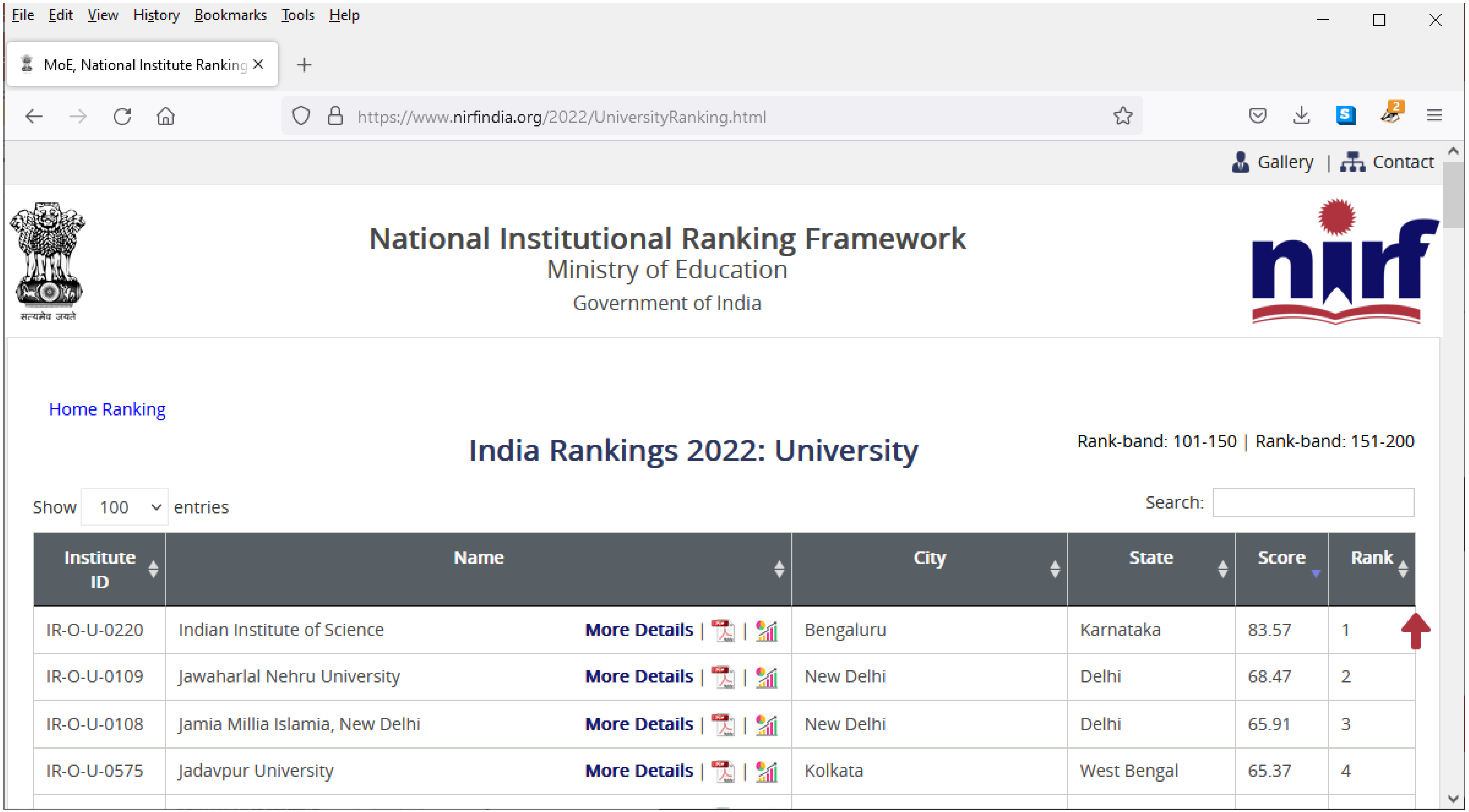
GLOBAL 2000 LIST BY THE CENTER FOR WORLD UNIVERSITY RANKINGS

2022-23 Edition

19,788 institutions were ranked, and those that placed at the top made the Global 2000 list.

india

World Rank	Institution	Location	National Rank	Education Rank	Employability Rank	Faculty Rank	Research Rank	Score
421	Indian Institute of Management Ahmedabad	India	1	362	11	-	-	75.0
491	Indian Institute of Science	India	2	330	838	-	464	74.2
559	Indian Institute of Technology Madras	India	3	-	180	-	602	73.5
563	Tata Institute of Fundamental Research	India	4	-	-	-	530	73.5
564	Indian Institute of Technology Bombay	India	5	-	326	-	563	73.5


File Edit View History Bookmarks Tools Help
MoE, National Institute Ranking
https://www.nirfindia.org/2022/UniversityRanking.html
Gallery | Contact
सत्यमेव जयते
nirf

National Institutional Ranking Framework
Ministry of Education
Government of India

Home Ranking

India Rankings 2022: University

Rank-band: 101-150 | Rank-band: 151-200

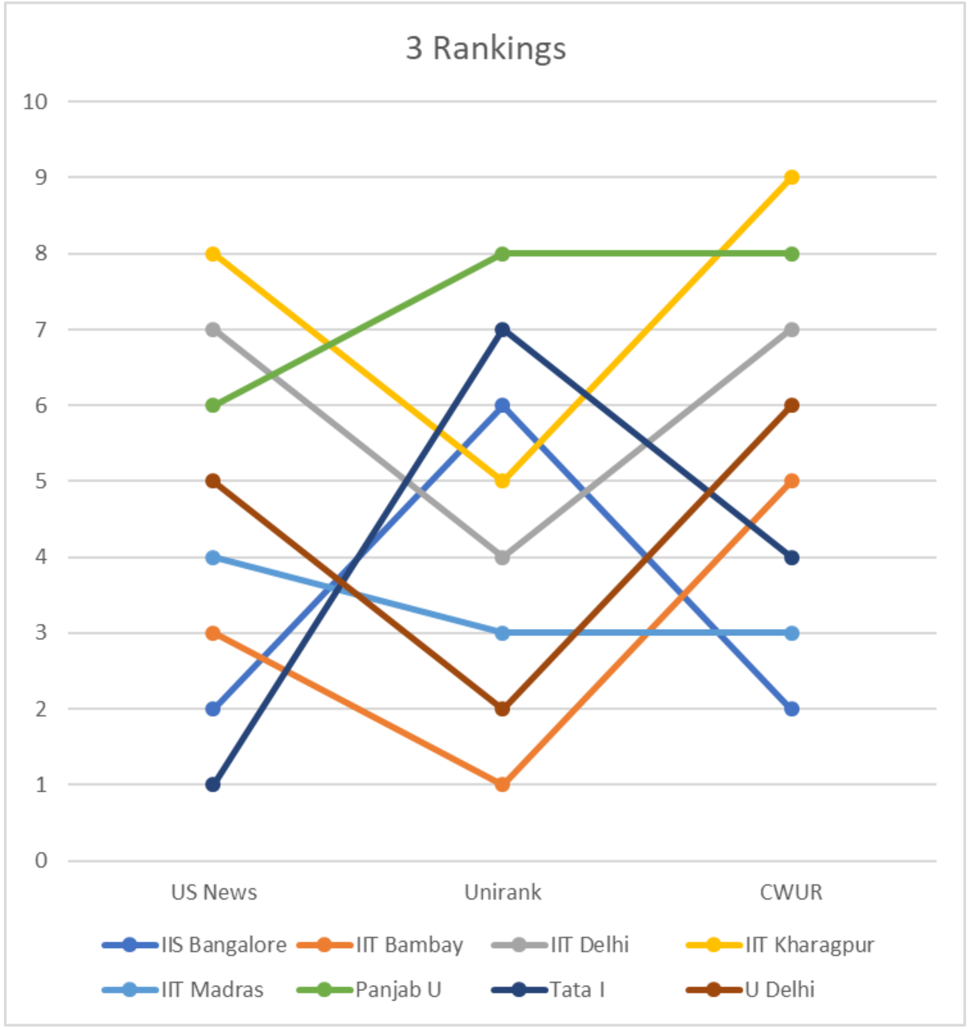
Show 100 entries

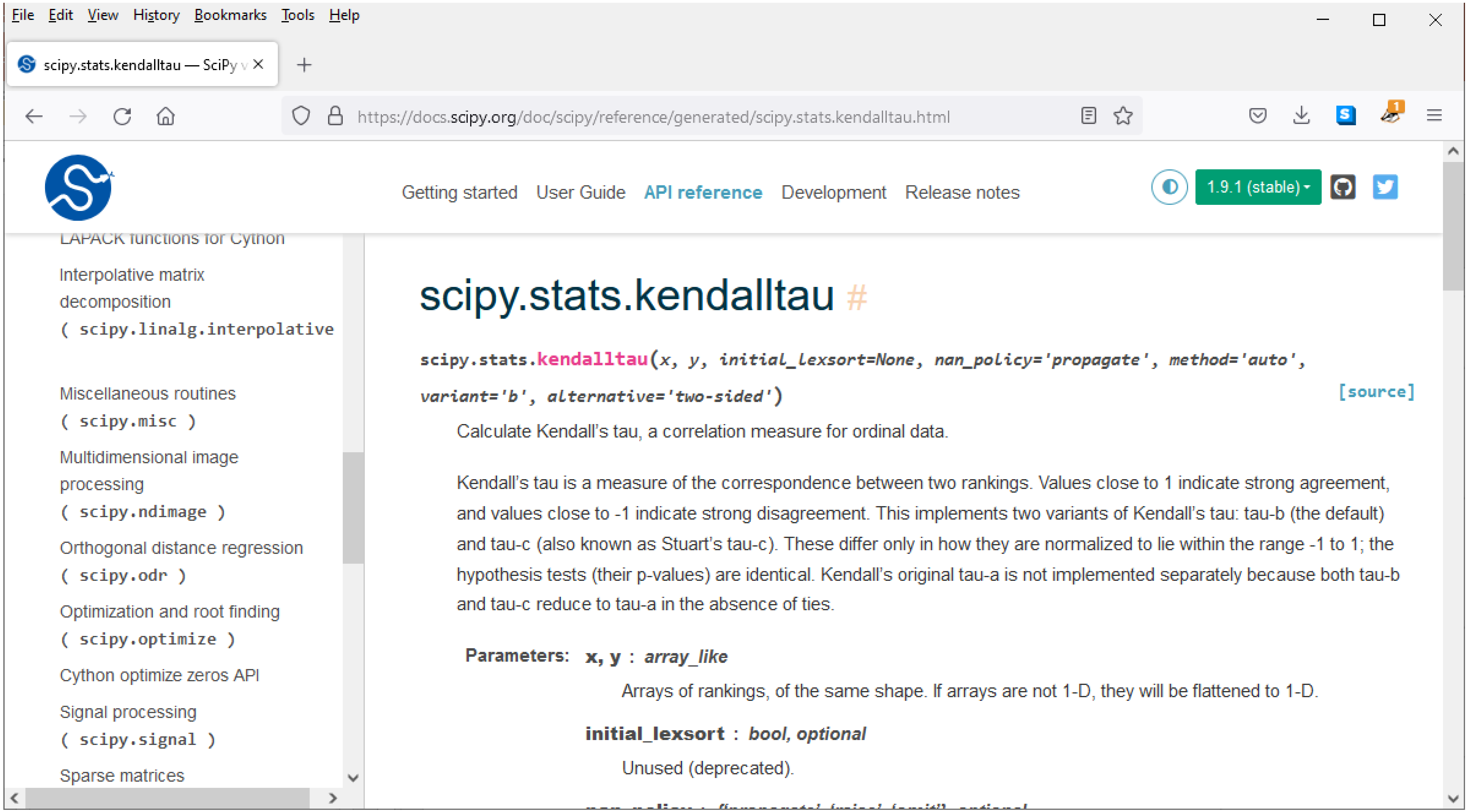
Search:

Institute ID	Name		City	State	Score	Rank
IR-O-U-0220	Indian Institute of Science	More Details | |	Bengaluru	Karnataka	83.57	1
IR-O-U-0109	Jawaharlal Nehru University	More Details | |	New Delhi	Delhi	68.47	2
IR-O-U-0108	Jamia Millia Islamia, New Delhi	More Details | |	New Delhi	Delhi	65.91	3
IR-O-U-0575	Jadavpur University	More Details | |	Kolkata	West Bengal	65.37	4

Excel Jump Chart

Name	US News	Unirank	CWUR
IIS Bangalore	2	6	2
IIT Bambay	3	1	5
IIT Delhi	7	4	7
IIT Kharagpur	8	5	9
IIT Madras	4	3	3
Panjab U	6	8	8
Tata I	1	7	4
U Delhi	5	2	6




File Edit View History Bookmarks Tools Help
scipy.stats.kendalltau — SciPy v
https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kendalltau.html
Getting started
User Guide
API reference
Development
Release notes
1.9.1 (stable)

scipy.stats.kendalltau

```
scipy.stats.kendalltau(x, y, initial_lexsort=None, nan_policy='propagate', method='auto', variant='b', alternative='two-sided')
```

[source]

Calculate Kendall's tau, a correlation measure for ordinal data.

Kendall's tau is a measure of the correspondence between two rankings. Values close to 1 indicate strong agreement, and values close to -1 indicate strong disagreement. This implements two variants of Kendall's tau: tau-b (the default) and tau-c (also known as Stuart's tau-c). These differ only in how they are normalized to lie within the range -1 to 1; the hypothesis tests (their p-values) are identical. Kendall's original tau-a is not implemented separately because both tau-b and tau-c reduce to tau-a in the absence of ties.

Parameters: **x, y** : *array_like*

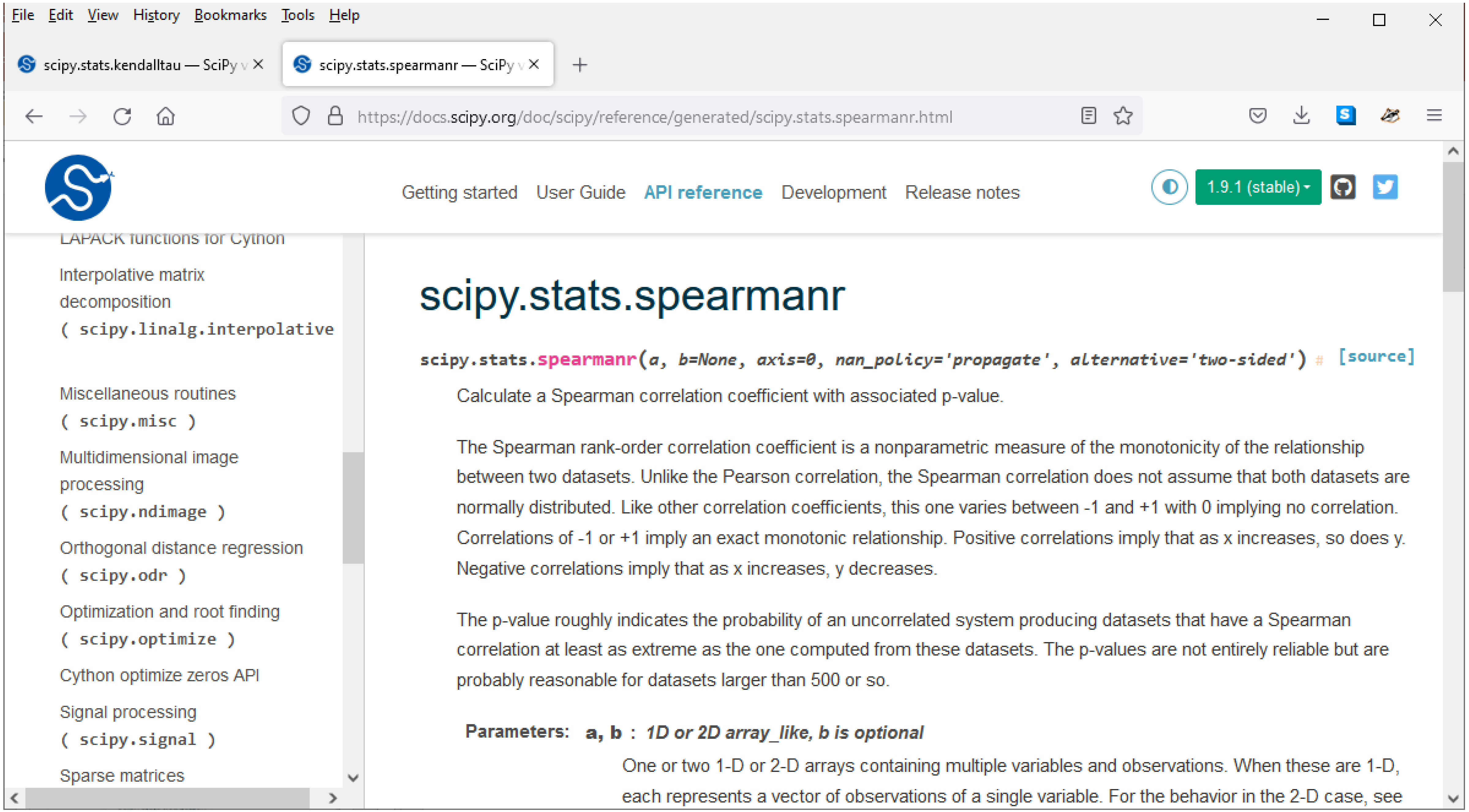
Arrays of rankings, of the same shape. If arrays are not 1-D, they will be flattened to 1-D.

initial_lexsort : *bool, optional*

Unused (deprecated).

```
from scipy.stats import kendalltau

usnews = [2, 3, 7, 8, 4, 6, 1, 5]
unirank = [6, 1, 4, 5, 3, 8, 7, 2]
cwur = [2, 5, 7, 9, 3, 8, 4, 6]

tau, p = kendalltau(usnews, unirank)
print('usnews:unirank', tau, p)
tau, p = kendalltau(usnews, cwur)
print('usnews:cwur', tau, p)
tau, p = kendalltau(cwur, unirank)
print('cwur:unirank', tau, p)
```

```
usnews:unirank 0.0 1.0
usnews:cwur 0.7142857142857142 0.014136904761904762
cwur:unirank 0.14285714285714285 0.7195436507936508
```


File Edit View History Bookmarks Tools Help
scipy.stats.kendalltau — SciPy v
scipy.stats.spearmanr — SciPy v
https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html
Getting started
User Guide
API reference
Development
Release notes
1.9.1 (stable)

LAPACK functions for Cython

Interpolative matrix decomposition (`scipy.linalg.interpolative`

Miscellaneous routines (`scipy.misc`)

Multidimensional image processing (`scipy.ndimage`)

Orthogonal distance regression (`scipy.odr`)

Optimization and root finding (`scipy.optimize`)

Cython optimize zeros API

Signal processing (`scipy.signal`)

Sparse matrices

scipy.stats.spearmanr

```
scipy.stats.spearmanr(a, b=None, axis=0, nan_policy='propagate', alternative='two-sided')
```
[source]

Calculate a Spearman correlation coefficient with associated p-value.

The Spearman rank-order correlation coefficient is a nonparametric measure of the monotonicity of the relationship between two datasets. Unlike the Pearson correlation, the Spearman correlation does not assume that both datasets are normally distributed. Like other correlation coefficients, this one varies between -1 and +1 with 0 implying no correlation. Correlations of -1 or +1 imply an exact monotonic relationship. Positive correlations imply that as x increases, so does y. Negative correlations imply that as x increases, y decreases.

The p-value roughly indicates the probability of an uncorrelated system producing datasets that have a Spearman correlation at least as extreme as the one computed from these datasets. The p-values are not entirely reliable but are probably reasonable for datasets larger than 500 or so.

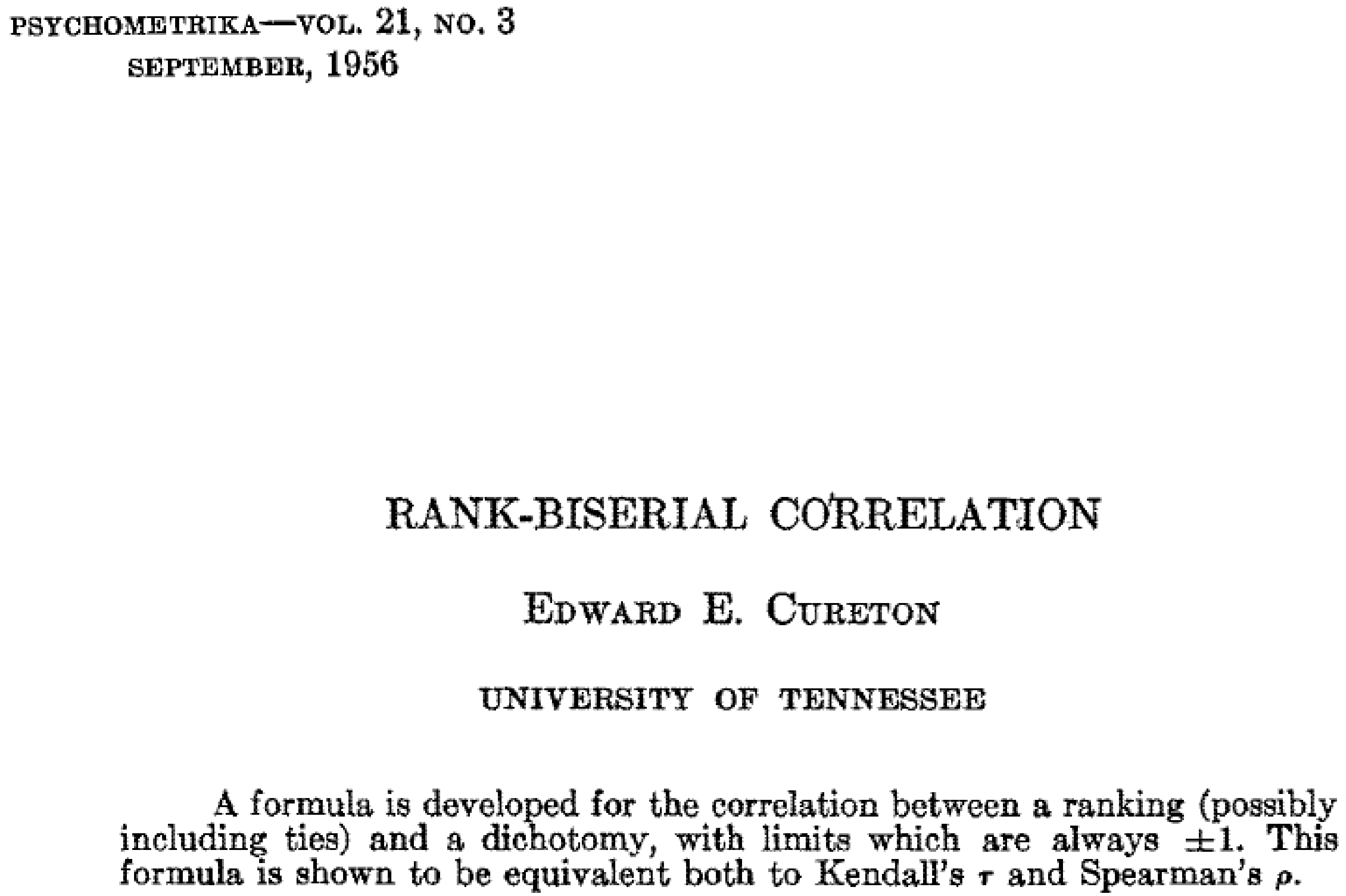
Parameters: **a, b** : *1D or 2D array_like, b is optional*

One or two 1-D or 2-D arrays containing multiple variables and observations. When these are 1-D, each represents a vector of observations of a single variable. For the behavior in the 2-D case, see

```
from scipy.stats import spearmanr

usnews = [2, 3, 7, 8, 4, 6, 1, 5]
unirank = [6, 1, 4, 5, 3, 8, 7, 2]
cwur = [2, 5, 7, 9, 3, 8, 4, 6]

rho, p = spearmanr(usnews, unirank)
print('usnews:unirank', rho, p)
rho, p = spearmanr(usnews, cwur)
print('usnews:cwur', rho, p)
rho, p = spearmanr(cwur, unirank)
print('cwur:unirank', rho, p)
```

```
usnews:unirank -0.04761904761904763 0.9108491685195836
usnews:cwur 0.8571428571428572 0.006530017254715292
cwur:unirank 0.11904761904761905 0.7788857260523797
```

Rank-Biserial Correlation of Cureton



PSYCHOMETRIKA—VOL. 21, NO. 3
SEPTEMBER, 1956

RANK-BISERIAL CORRELATION

EDWARD E. CURETON
UNIVERSITY OF TENNESSEE

A formula is developed for the correlation between a ranking (possibly including ties) and a dichotomy, with limits which are always ± 1. This formula is shown to be equivalent both to Kendall's τ and Spearman's ρ.

Rank-Biserial Correlation

- Used with ordinal vs dichotomous variables.
- rb = (2/n)(Y0 – Y1)
- where
- n is the number of ranked entries,
- Y1 is the mean rank of those scoring 1 on the dichotomy, and
- Y0 is the mean rank of those scoring 0 on the dichotomy.
- Justin Fister, correlation analysis of on-page attributes and search engine rankings, UC MS of CS theses 2007

Example for Rank-Biserial Correlation

- rank: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
- biserial: R R R R R R N N N N N N N N N N N N N N
- mean rank Y1 = (0 + 1 + ... + 5) / 6 = 5 * 6 / 2 / 6 = 15 / 6 = 2.5
- mean rank Y0 = (6 + ... + 19) / 14 = (190 – 15) / 14 = 175 / 14 = 12.5
- rb = (Y0 – Y1) / (n/2) = 10 / 10 = 1
- biserial: N N N N N N N N N N N N N N R R R R R R
- Y0 = (0 + 1 + ... + 13) / 14 = 13 * 14 / 2 / 14 = 91 / 14 = 13 / 2 = 6.5
- Y1 = (14 + ... + 19) / 6 = (190 – 91) / 6 = 99 / 6 = 33 / 3 = 16.5
- rb = (Y0 – Y1) / (n/2) = -10 / 10 = -1

8.4 Evaluation of ranked retrieval results

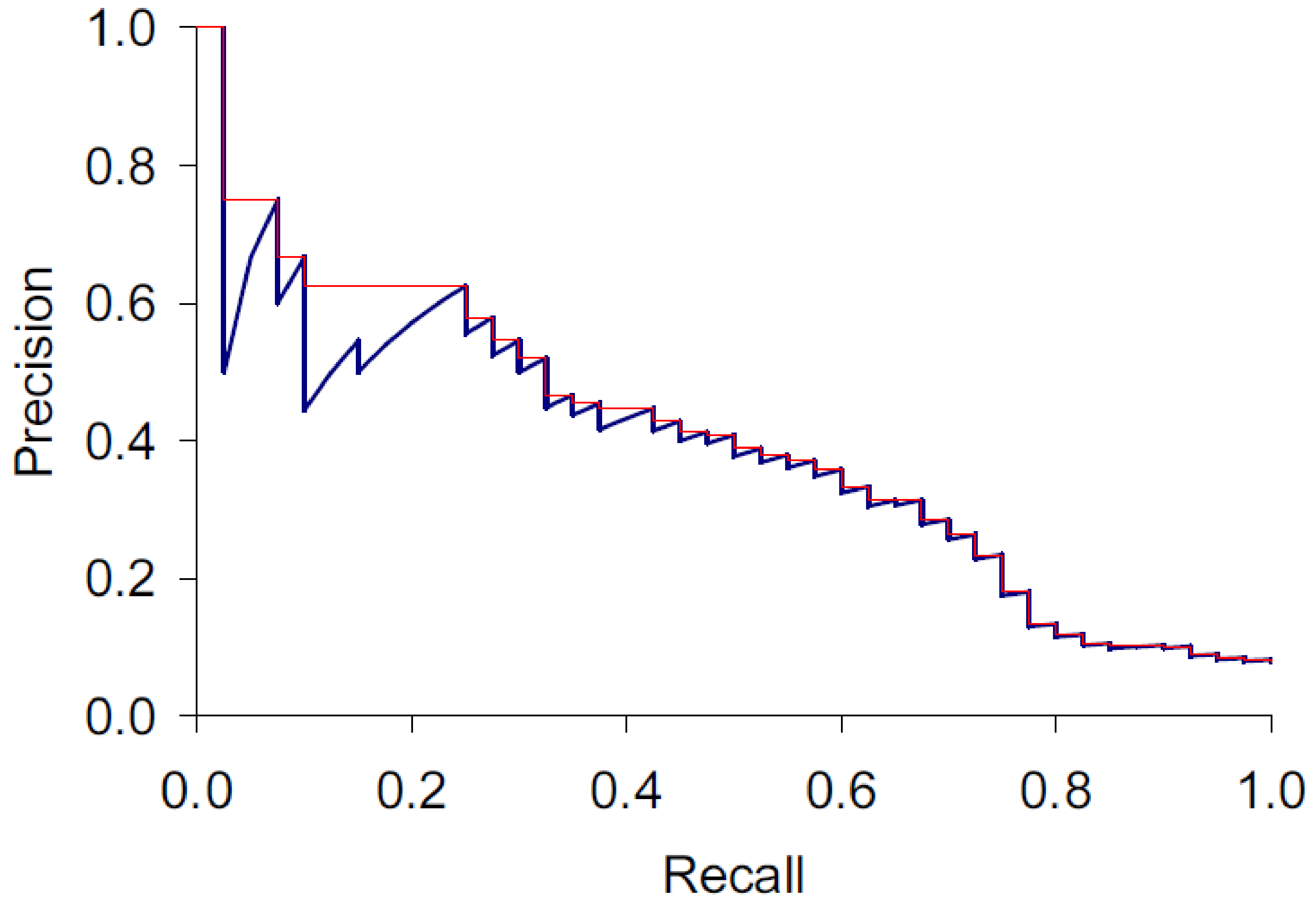
Precision, recall, and the F measure are set-based measures. They are computed using unordered sets of documents. We need to extend these measures (or to define new measures) if we are to evaluate the ranked retrieval results that are now standard with search engines. In a ranked retrieval context, appropriate sets of retrieved documents are naturally given by the top k retrieved documents. For each such set, precision and recall values can be plotted to give a *precision-recall curve*, such as the one shown in Figure 8.2. Precision-recall curves have a distinctive saw-tooth shape: if the $(k+1)^{\text{th}}$ document retrieved is nonrelevant then recall is the same as for the top k documents, but precision has dropped. If it is relevant, then both precision and recall increase, and the curve jags up and to the right. It is often useful to remove these jiggles and the standard way to do this is with an interpolated precision: the *interpolated precision* p_{interp} at a certain recall level r is defined as the highest precision found for any recall level $r' \geq r$:

PRECISION-RECALL CURVE

INTERPOLATED PRECISION

$$p_{interp}(r) = \max_{r' \geq r} p(r') \qquad (8.7)$$

The justification is that almost anyone would be prepared to look at a few more documents if it would increase the percentage of the viewed set that were relevant (that is, if the precision of the larger set is higher). Interpolated precision is shown by a thinner line in Figure 8.2. With this definition, the interpolated precision at a recall of 0 is well-defined (Exercise 8.4).

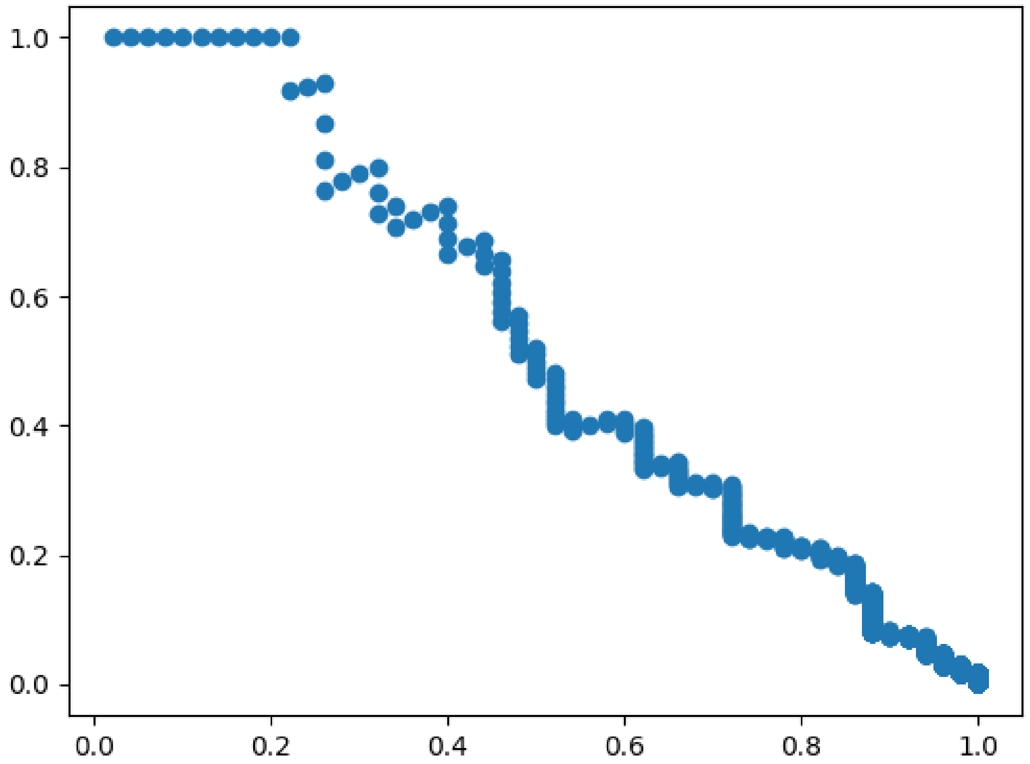


► **Figure 8.2** Precision/recall graph.

```
# IR8A.py CS5154/6054 cheng 2022
# TfidfVectorizer and CountVectorizer (binary=True) are used
# a random doc is the query and the top 50 cosine similarity
# in Tfidf are considered relevent
# CountVectors are ranked using cosine similarity
# precision and recall at each retrieval level are computed
# and the precision-recall graph (Fig 8.2 iir) is plotted
# Usage: python IR8A.py

import re
import numpy as np
import random
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from matplotlib import pyplot as plt
relevant = 50

f = open("bible.txt", "r")
docs = f.readlines()
f.close()
```

```
tfidf = TfidfVectorizer(max_df=0.4, min_df=2)
dt = tfidf.fit_transform(docs)
N = len(docs)
query = random.randint(0, N)
print(query, docs[query])

sim = cosine_similarity(dt[query], dt)
toptfidf = set()
for index in np.argsort(sim)[0][::-1][0:relevant]:
    toptfidf.add(index)

print(toptfidf)

cv = CountVectorizer(binary=True, max_df=0.4, min_df=2)
dt2 = cv.fit_transform(docs)
sim2 = cosine_similarity(dt2[query], dt2)
sorted = np.argsort(sim2)[0][::-1]
```

```
precision = np.zeros(N)
recall = np.zeros(N)
m = 0
for i in range(N):
    if sorted[i] in toptfidf:
        m = m + 1
        # tp = m, fn = relevant - m, fp = i + 1 - m, tn = N - tp - fn - fp
    precision[i] = ?
    recall[i] = ?

plt.scatter(recall, precision)
plt.show()
```



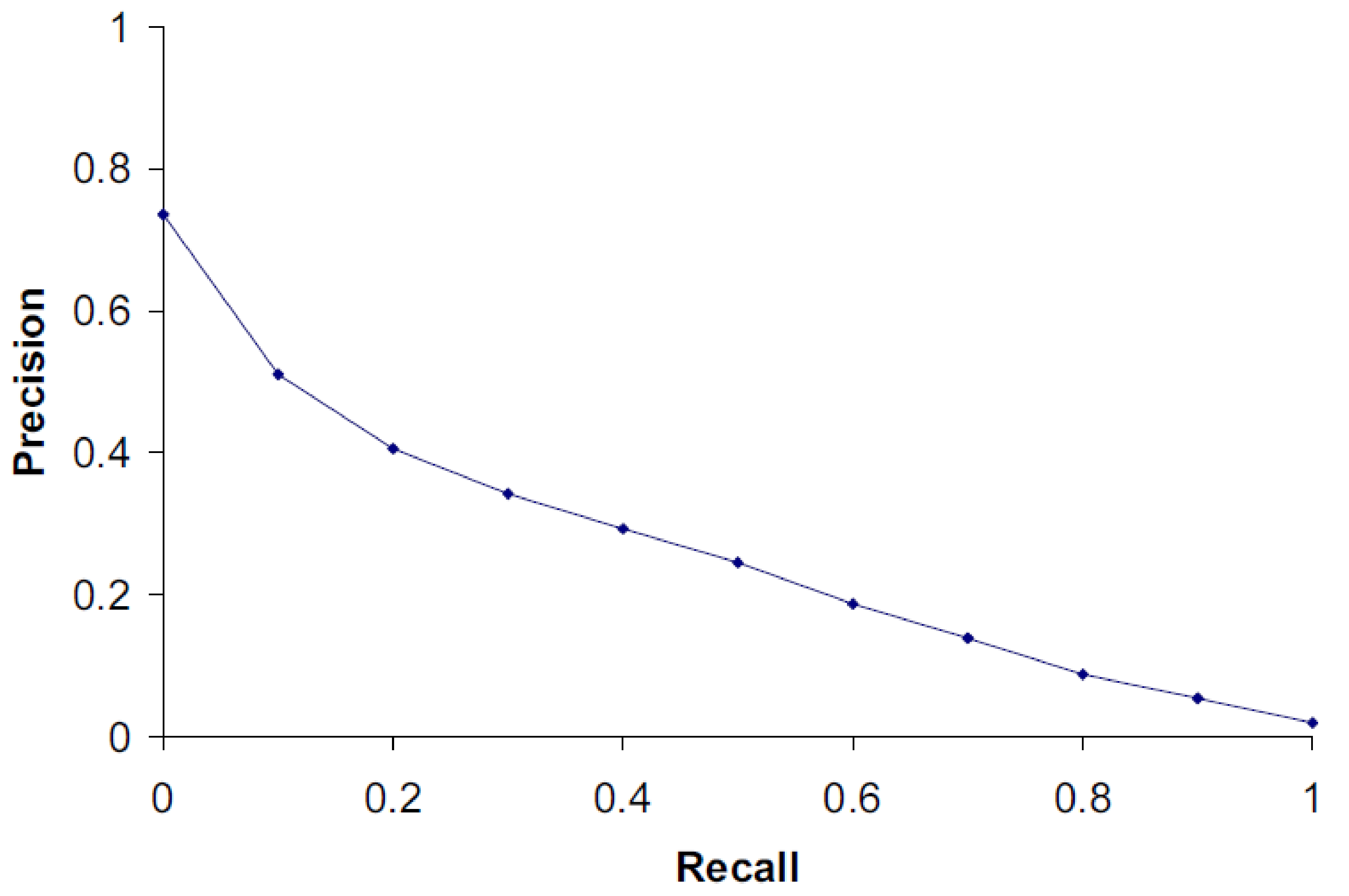
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

11-POINT INTERPOLATED AVERAGE PRECISION

Examining the entire precision-recall curve is very informative, but there is often a desire to boil this information down to a few numbers, or perhaps even a single number. The traditional way of doing this (used for instance in the first 8 TREC Ad Hoc evaluations) is the *11-point interpolated average precision*. For each information need, the interpolated precision is measured at the 11 recall levels of 0.0, 0.1, 0.2, . . . , 1.0. For the precision-recall curve in Figure 8.2, these 11 values are shown in Table 8.1. For each recall level, we then calculate the arithmetic mean of the interpolated precision at that recall level for each information need in the test collection. A composite precision-recall curve showing 11 points can then be graphed. Figure 8.3 shows an example graph of such results from a representative good system at TREC 8.

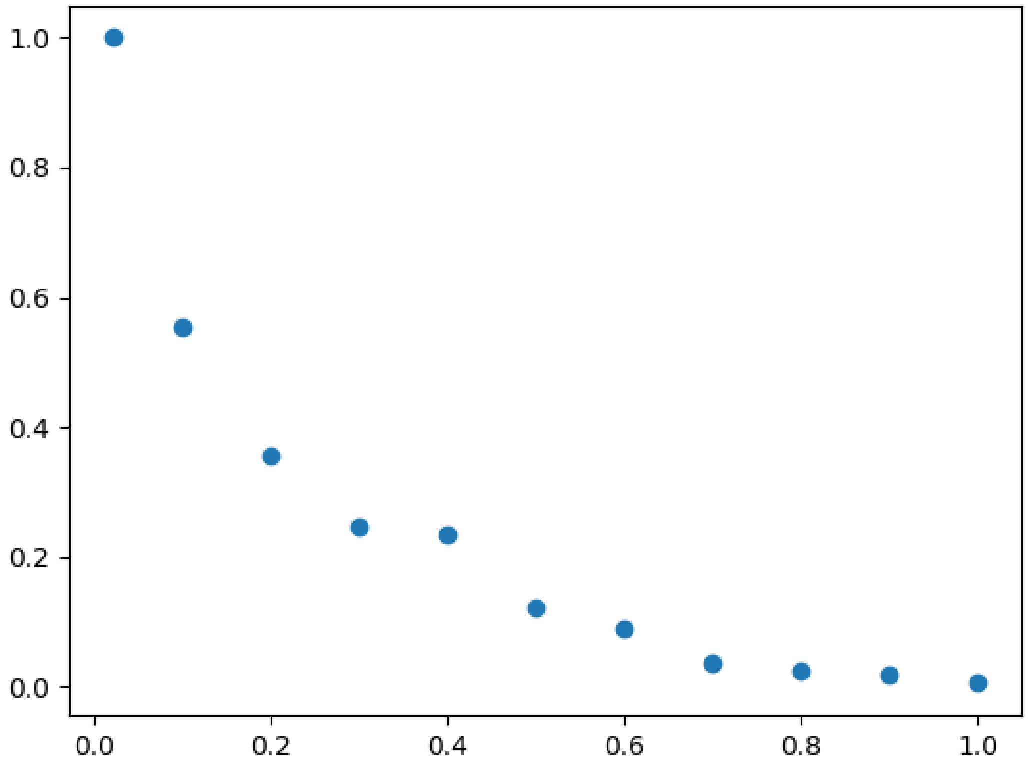
Recall	Interp. Precision
0.0	1.00
0.1	0.67
0.2	0.63
0.3	0.55
0.4	0.45
0.5	0.41
0.6	0.36
0.7	0.29
0.8	0.13
0.9	0.10
1.0	0.08

▶ **Table 8.1** Calculation of 11-point Interpolated Average Precision. This is for the precision-recall curve shown in Figure 8.2.



► **Figure 8.3** Averaged 11-point precision/recall graph across 50 queries for a representative TREC system. The Mean Average Precision for this system is 0.2553.

```
eleven_recalls = np.zeros(11)
interpolated = np.zeros(11)
n = 0
for i in range(N):
    if n <= 10 and recall[i] * 10 >= n:
        interpolated[n] = max(precision[i:])
        eleven_recalls[n] = recall[i]
        print(n, precision[i], interpolated[n])
        n = n + 1
    if n > 10:
        break

plt.scatter(eleven_recalls, interpolated)
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

MEAN AVERAGE PRECISION

In recent years, other measures have become more common. Most standard among the TREC community is *Mean Average Precision* (MAP), which provides a single-figure measure of quality across recall levels. Among evaluation measures, MAP has been shown to have especially good discrimination and stability. For a single information need, Average Precision is the average of the precision value obtained for the set of top k documents existing after each relevant document is retrieved, and this value is then averaged over information needs. That is, if the set of relevant documents for an information need $q_j \in Q$ is $\{d_1, \ldots d_{m_j}\}$ and R_{jk} is the set of ranked retrieval results from the top result until you get to document d_k, then

$$\mathrm{MAP}(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} \mathrm{Precision}(R_{jk}) \tag{8.8}$$

When a relevant document is not retrieved at all,[1] the precision value in the above equation is taken to be 0. For a single information need, the average precision approximates the area under the uninterpolated precision-recall curve, and so the MAP is roughly the average area under the precision-recall curve for a set of queries.

PRECISION AT *k*

The above measures factor in precision at all recall levels. For many prominent applications, particularly web search, this may not be germane to users. What matters is rather how many good results there are on the first page or the first three pages. This leads to measuring precision at fixed low levels of retrieved results, such as 10 or 30 documents. This is referred to as "Precision at *k*", for example "Precision at 10". It has the advantage of not requiring any estimate of the size of the set of relevant documents but the disadvantages that it is the least stable of the commonly used evaluation measures and that it does not average well, since the total number of relevant documents for a query has a strong influence on precision at *k*.

R-PRECISION

An alternative, which alleviates this problem, is *R-precision*. It requires having a set of known relevant documents *Rel*, from which we calculate the precision of the top *Rel* documents returned. (The set *Rel* may be incomplete, such as when *Rel* is formed by creating relevance judgments for the pooled top k results of particular systems in a set of experiments.) R-precision adjusts for the size of the set of relevant documents: A perfect system could score 1 on this metric for each query, whereas, even a perfect system could only achieve a precision at 20 of 0.4 if there were only 8 documents in the collection relevant to an information need. Averaging this measure across queries thus makes more sense. This measure is harder to explain to naive users than Precision at k but easier to explain than MAP. If there are $|Rel|$ relevant documents for a query, we examine the top $|Rel|$ results of a system, and find that r are relevant, then by definition, not only is the precision (and hence R-precision) $r/|Rel|$, but the recall of this result set is also $r/|Rel|$.

BREAK-EVEN POINT

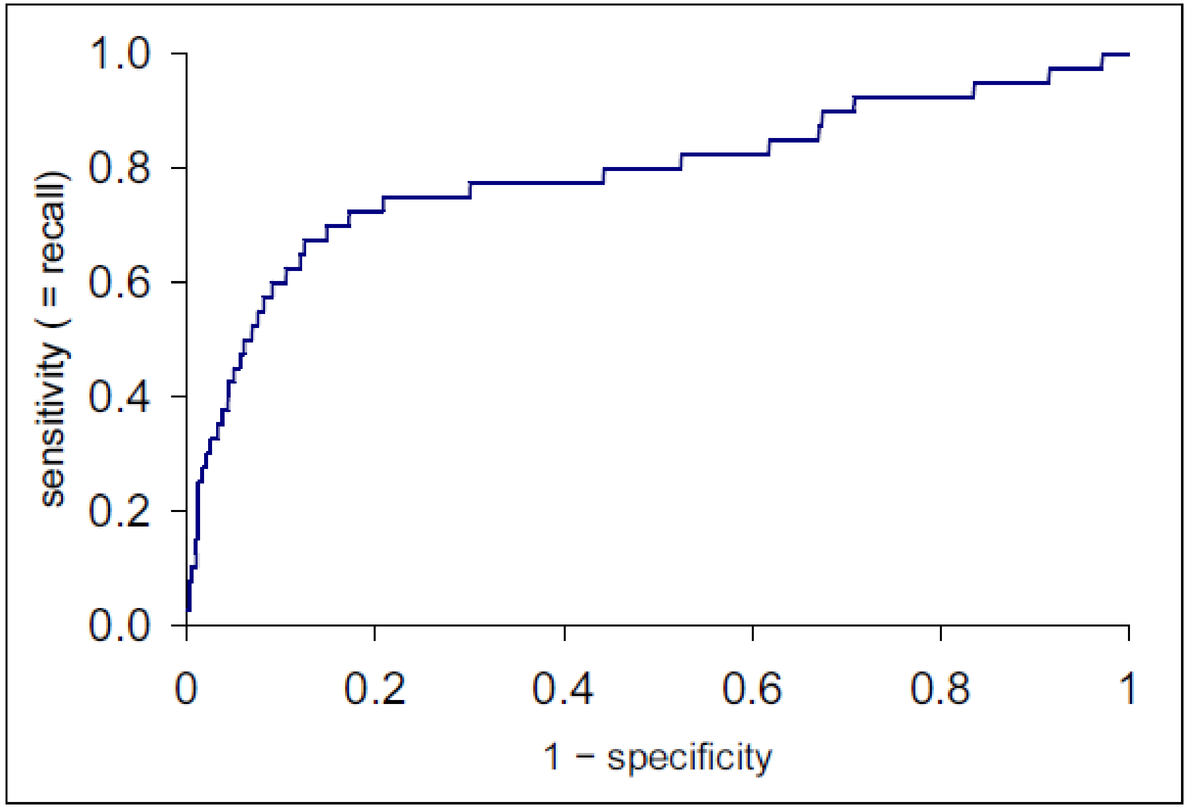
Thus, R-precision turns out to be identical to the *break-even point*, another measure which is sometimes used, defined in terms of this equality relationship holding. Like Precision at k, R-precision describes only one point on the precision-recall curve, rather than attempting to summarize effectiveness across the curve, and it is somewhat unclear why you should be interested in the break-even point rather than either the best point on the curve (the point with maximal F-measure) or a retrieval level of interest to a particular application (Precision at k). Nevertheless, R-precision turns out to be highly correlated with MAP empirically, despite measuring only a single point on the curve.

ROC CURVE

SENSITIVITY

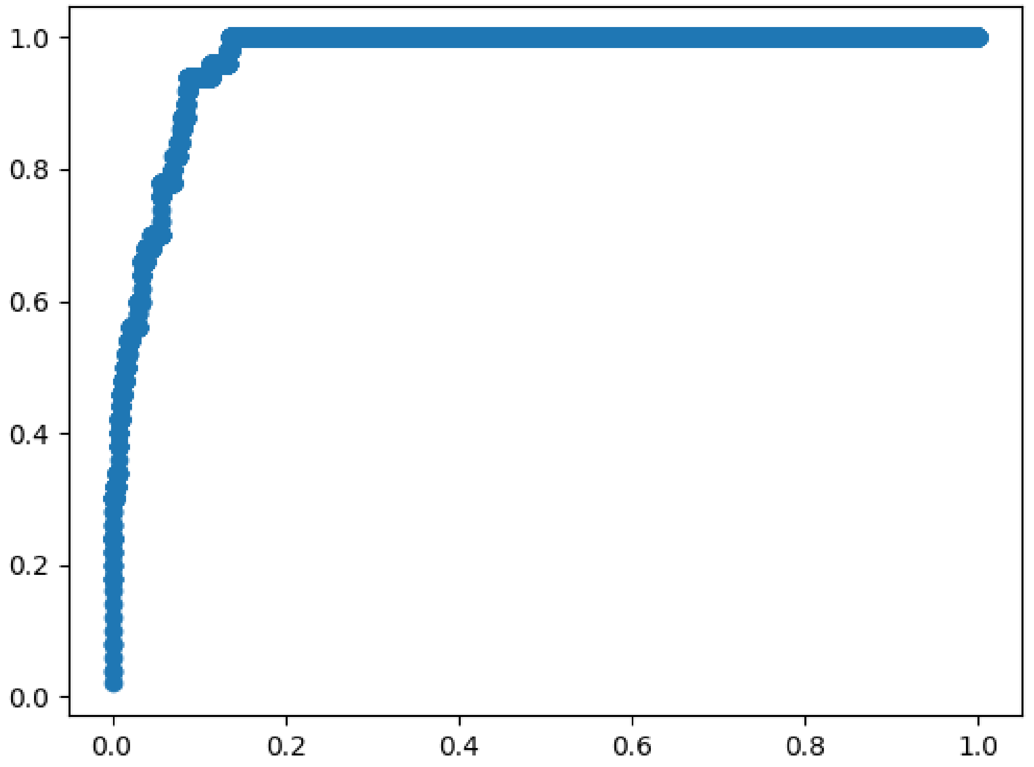
SPECIFICITY

Another concept sometimes used in evaluation is an *ROC curve*. ("ROC" stands for "Receiver Operating Characteristics", but knowing that doesn't help most people.) An ROC curve plots the true positive rate or sensitivity against the false positive rate or (1 − specificity). Here, *sensitivity* is just another term for recall. The false positive rate is given by $fp/(fp + tn)$. Figure 8.4 shows the ROC curve corresponding to the precision-recall curve in Figure 8.2. An ROC curve always goes from the bottom left to the top right of the graph. For a good system, the graph climbs steeply on the left side. For unranked result sets, *specificity*, given by $tn/(fp + tn)$, was not seen as a very useful notion. Because the set of true negatives is always so large, its value would be almost 1 for all information needs (and, correspondingly, the value of the false positive rate would be almost 0). That is, the "interesting" part of Figure 8.2 is $0 <$ recall < 0.4, a part which is compressed to a small corner of Figure 8.4. But an ROC curve could make sense when looking over the full retrieval spectrum, and it provides another way of looking at the data. In many fields, a common aggregate measure is to report the area under the ROC curve, which is the ROC analog of MAP. Precision-recall curves are sometimes loosely referred to as ROC curves. This is understandable, but not accurate.



► **Figure 8.4** The ROC curve corresponding to the precision-recall curve in Figure 8.2.

```
rocx = np.zeros(N)
recall = np.zeros(N)
m = 0
for i in range(N):
    if sorted[i] in toptfidf:
        m = m + 1
        # tp = m, fn = relevant - m, fp = i + 1 - m, tn = N - tp - fn - fp
    rocx[i] = ?
    recall[i] = ?

plt.scatter(rocx, recall)
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0