02B – INFORMATION RETRIEVAL 2

CS 1656 Introduction to Data Science

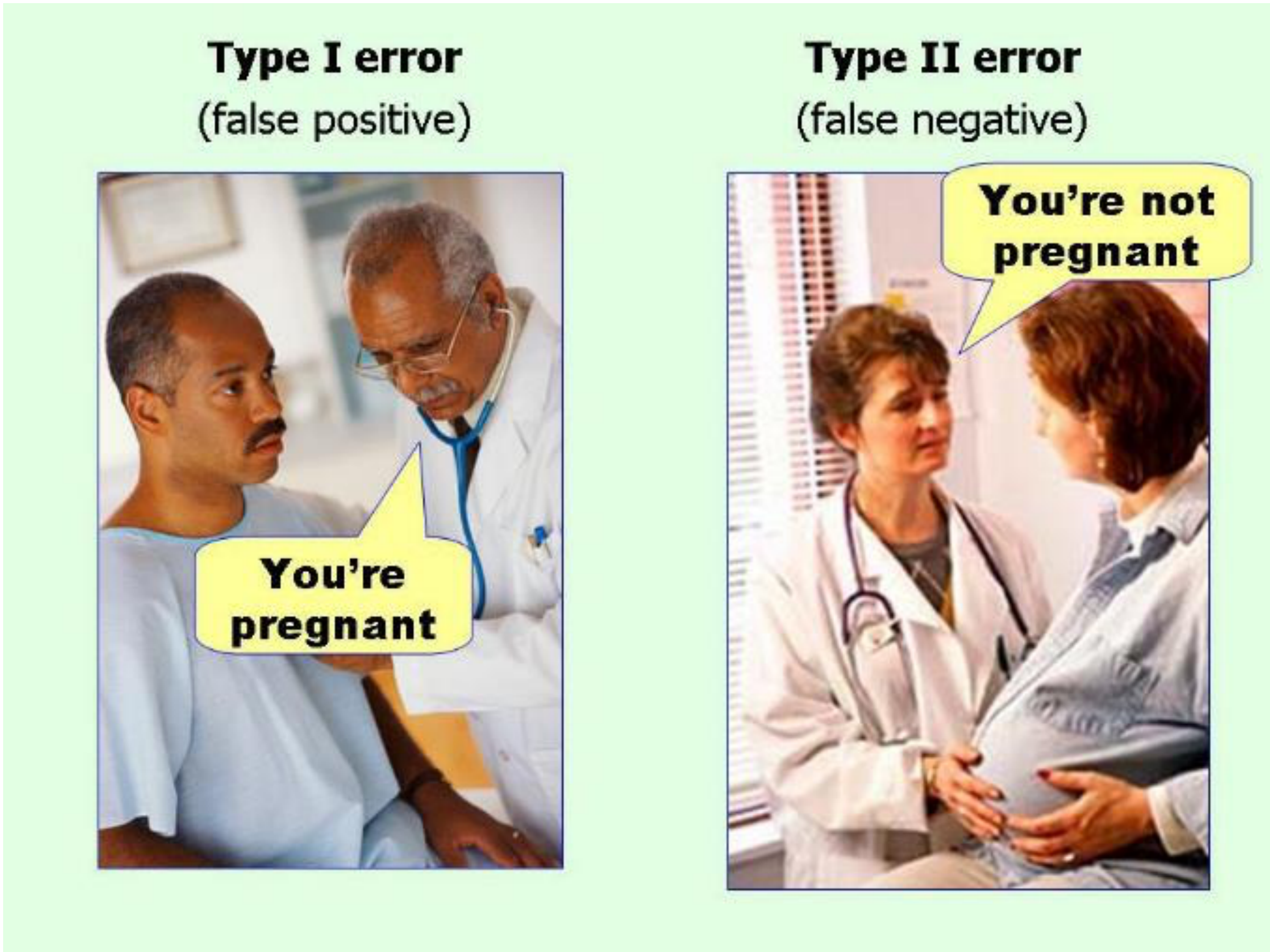
Alexandros Labrinidis – http://labrinidis.cs.pitt.edu
University of Pittsburgh

METRICS

How to measure effectiveness

- Approximate, incomplete results are usual
  - Especially if using an index to expedite computation
- **False negative (FN)**:
  - A relevant document was not returned
- **False positive (FP)**:
  - An irrelevant document was returned
- **True negative (TN)**:
  - An irrelevant document was not returned
- **True positive (TP)**:
  - A relevant document was returned

An illustration

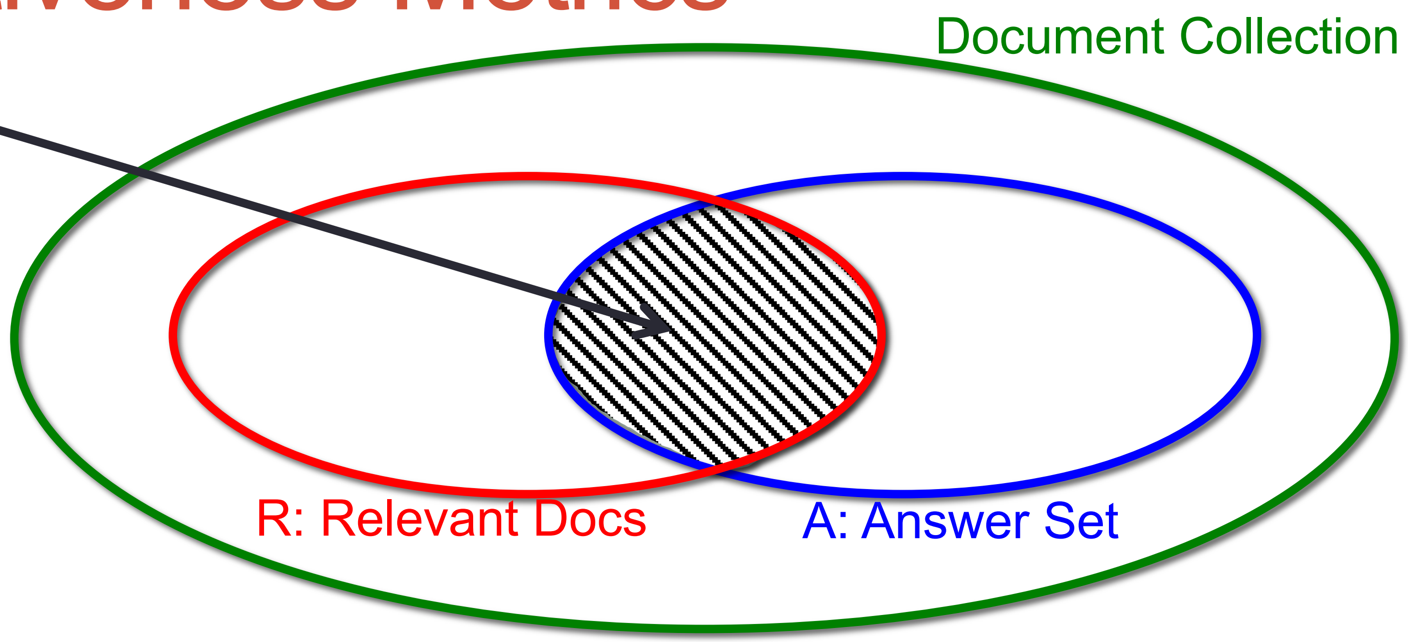


{True, False} {Positive, Negative}

		Returned by IR technique?	
		Yes	**No**
Relevant Document?	**Yes**	True Positive (TP)	False Negative (FN)
	No	False Positive (FP)	True Negative (TN)

Effectiveness Metrics

R ∩ A: Relevant docs in the answer set



- **Precision,** $p = |R \cap A| / |A| = TP / (TP + FP)$
  - What percentage of retrieved documents are relevant to the query
- **Recall,** $r = |R \cap A| / |R| = TP / (TP + FN)$
  - What percentage of the documents that are relevant to the query has been retrieved

Handout Example

Algorithm #1		Algorithm #2	
1	104.txt	1	106.txt
2	106.txt	2	120.txt
3	108.txt	3	104.txt
4	120.txt	4	102.txt
5	122.txt	5	108.txt
6	124.txt		

Assume a collection of 100 documents, named 101.txt, 102.txt, 103.txt, …, 200.txt.

Assume that we run two different relevance ranking algorithms and get the above ordered lists of documents that are relevant to the user's query.

Assume that the correct relevant documents in the collection are 101.txt, 102.txt, 103.txt, 104.txt, 105.txt, 106.txt, 107.txt, 108.txt, 109.txt, and 110.txt.

Understanding Question

- **Question**:
  - Given the data from the handout, what is the precision of algorithm #1?

- **Possible Answers**:
  - 30%
  - 40%
  - 50%
  - 60%
  - 80%

Handout Example - Solutions

Algorithm #1		Algorithm #2	
1	104.txt	1	106.txt
2	106.txt	2	120.txt
3	108.txt	3	104.txt
4	120.txt	4	102.txt
5	122.txt	5	108.txt
6	124.txt		

Assume that the correct relevant documents in the collection are
101.txt, 102.txt, 103.txt, 104.txt, 105.txt, 106.txt, 107.txt, 108.txt, 109.txt, and 110.txt.

Precision of Algorithm #1: 3 / 6 = 50%

Recall of Algorithm #1 3 / 10 = 30%

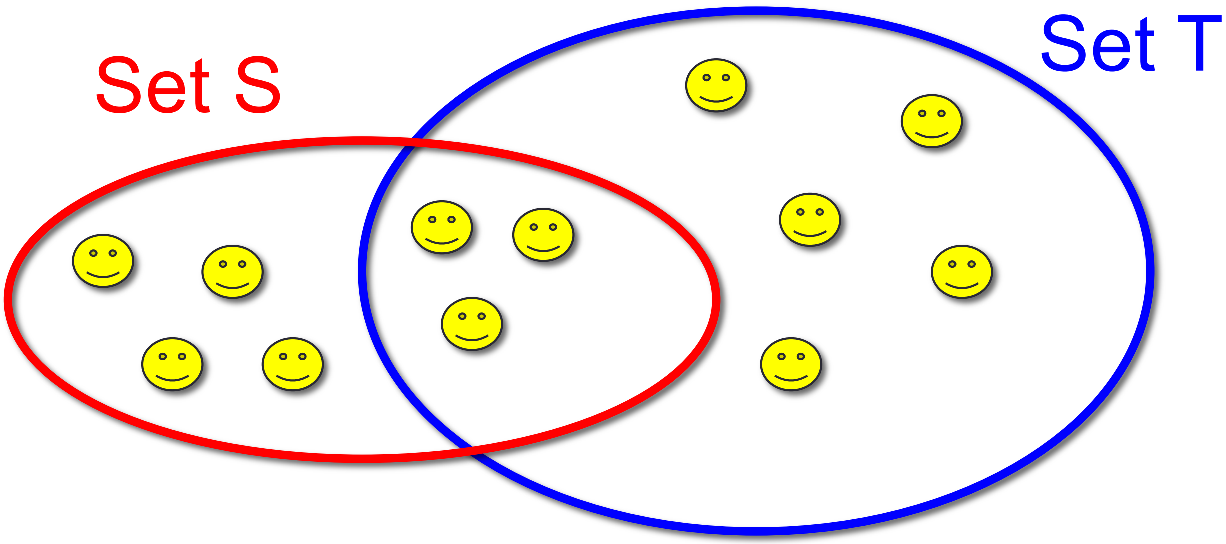
Precision of Algorithm #2: 4 / 5 = 80%

Recall of Algorithm #2 4 / 10 = 40%

COMPARING SETS

Jaccard Similarity of Sets

- Jaccard similarity of sets S and T is defined as:
  - (size of intersection of S, T) / (size of union of S, T)



Jaccard similarity of above sets = 3 / (4+3+5) = 3/12 = 25%

COMPARING RANKINGS

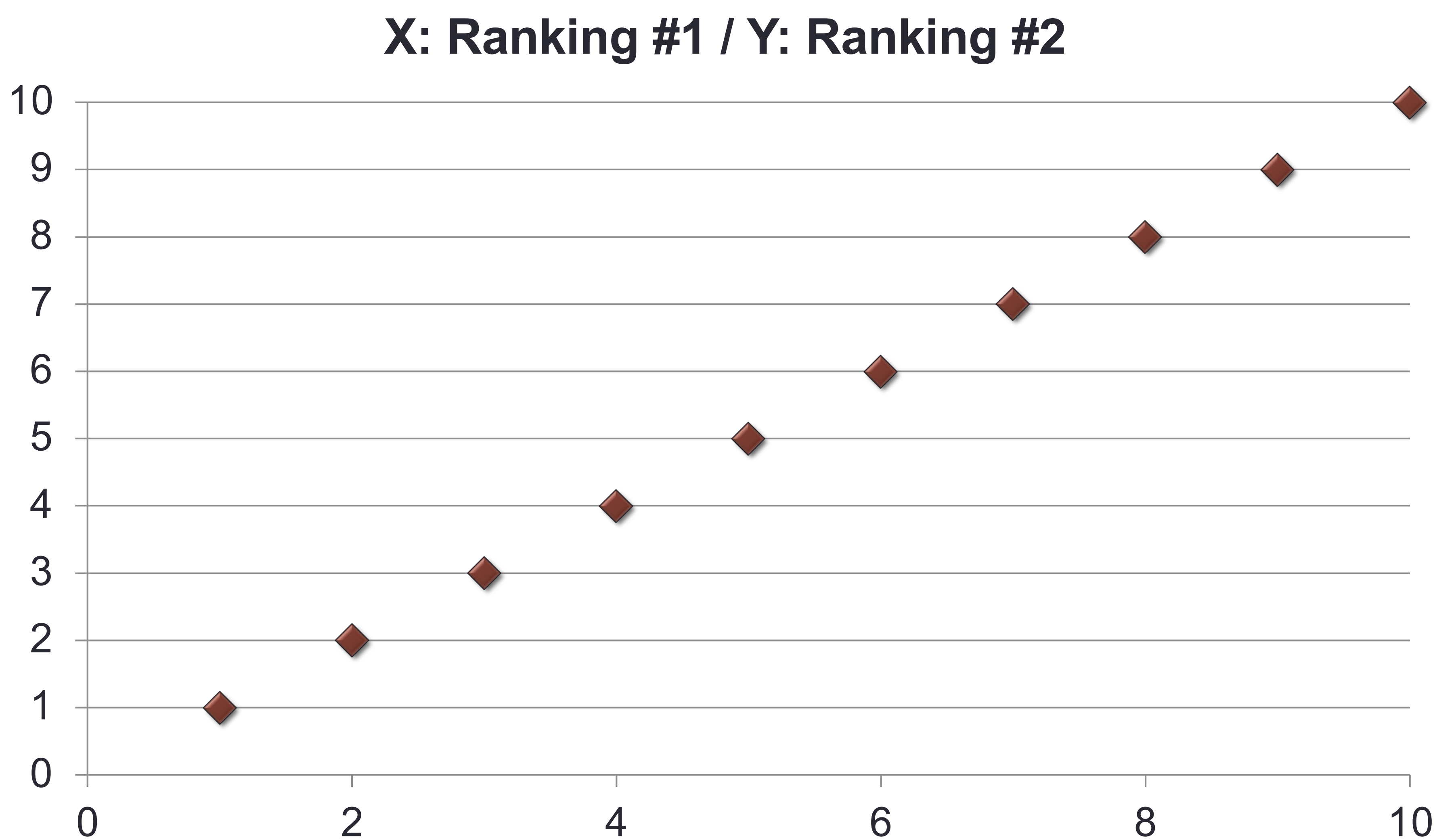
How to compare rankings?

- Would like a similarity metric that allows us to compare two rankings and:
  - Has a value that ranges between -1 and 1
  - **1** means perfect **agreement** between the rankings
  - **-1** means perfect **disagreement** between the rankings
  - **0** means the two rankings are completely **independent**

Two Rankings – Perfect Agreement

Document	Ranking #1	Ranking #2
Doc #123	1	1
Doc #84	2	2
Doc #56	3	3
Doc #6	4	4
Doc #8	5	5
Doc #9	6	6
Doc #511	7	7
Doc #129	8	8
Doc #187	9	9
Doc #25	10	10

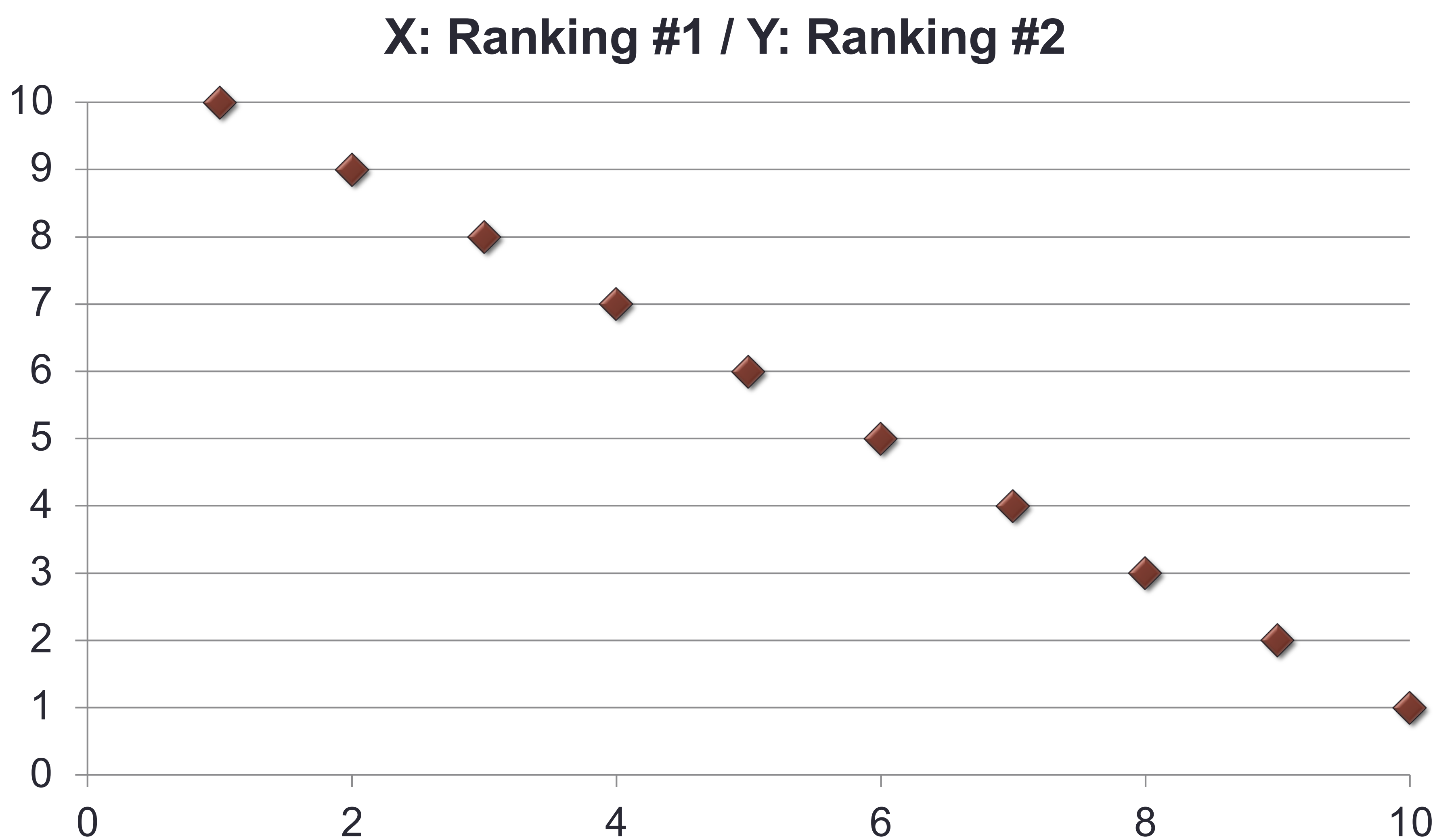
Two Rankings – Perfect Agreement



Two Rankings – Perfect Disagreement

Document	Ranking #1	Ranking #2
Doc #123	1	10
Doc #84	2	9
Doc #56	3	8
Doc #6	4	7
Doc #8	5	6
Doc #9	6	5
Doc #511	7	4
Doc #129	8	3
Doc #187	9	2
Doc #25	10	1

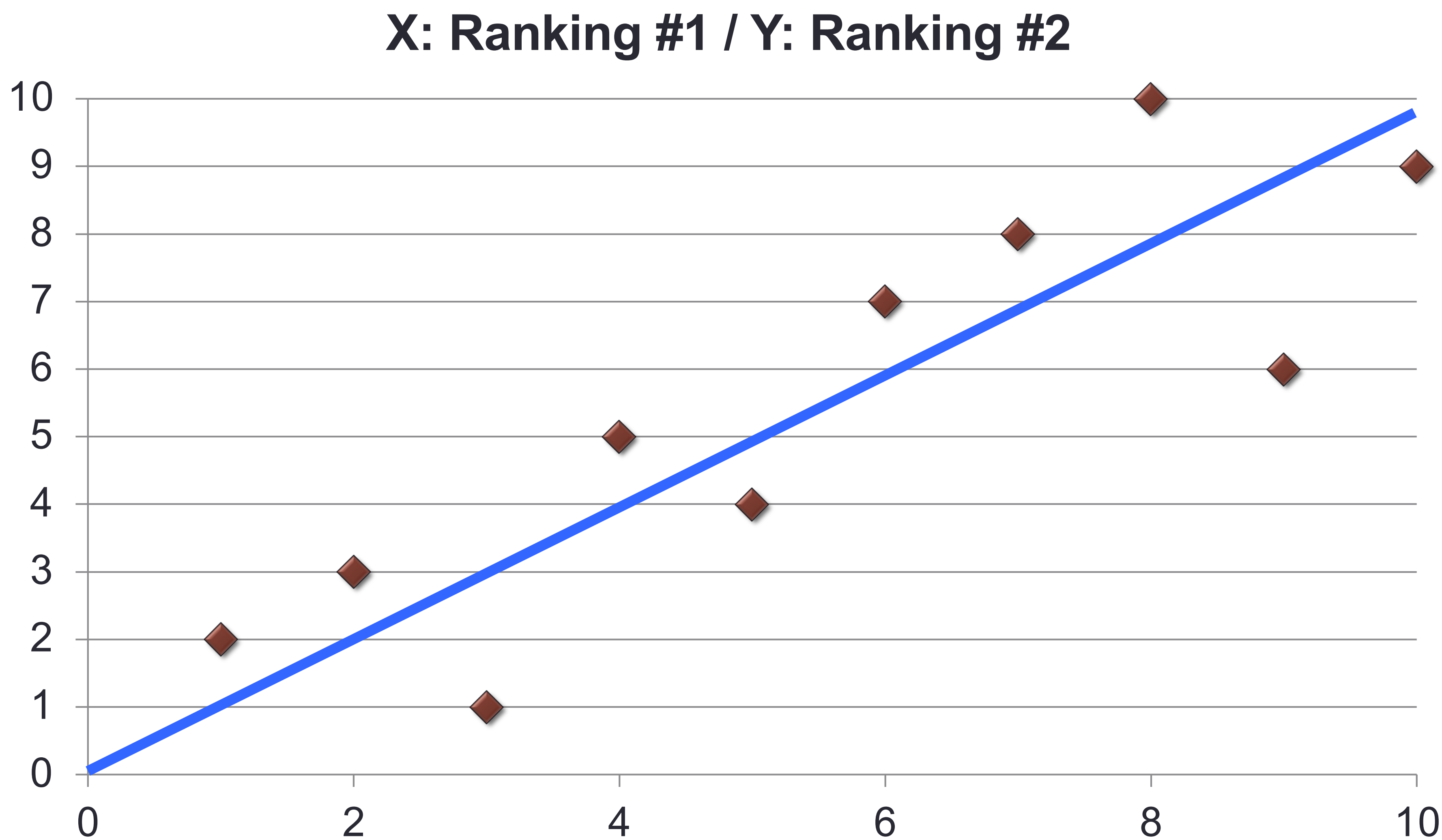
Two Rankings – Perfect Disagreement



Two Rankings – How to measure?

Document	Ranking #1 S1, j	Ranking #2 S2, j	S1,j – S2, j	(S1,j – S2, j)^2
Doc #123	1	2	-1	1
Doc #84	2	3	-1	1
Doc #56	3	1	+2	4
Doc #6	4	5	-1	1
Doc #8	5	4	+1	1
Doc #9	6	7	-1	1
Doc #511	7	8	-1	1
Doc #129	8	10	-2	4
Doc #187	9	6	+3	9
Doc #25	10	9	+1	1

Two Rankings – Plot



Spearman Coefficient

- If $s_{1,j}$ and $s_{2,j}$ are the positions of a document *j* in two rankings *R1* and *R2*, then the Spearman Coefficient is:

$$S(R1, R2) = 1 - \frac{6 \times \sum_{j=1}^{K} (s_{1,j} - s_{2,j})^2}{K \times (K^2 - 1)}$$

where *K* is the total number of documents.

For previous example: $S = 1 - (6x24) / (10x(10^2 - 1)) = 0.854$

What if rankings vary in similar ways?

- Assume documents d_j and d_k and the differences in their positions in rankings R1 and R2 respectively:
  - S1, k – S1, j (i.e., difference in ranking 1 between d_j and d_k)
  - S2, k – S2, j (i.e., difference in ranking 2 between d_j and d_k)

- If these differences have same sign, then document pair is d_j and d_k concordant
- If these differences have different sign, then document pair is d_j and d_k discordant

- A simple way to measure strength of correlation between rankings is to measure #concordant - #discordant

Kendall Tau Coefficient

- A simple way to measure strength of correlation between rankings is to measure #concordant - #discordant

$$\tau(R_1, R_2) = P(R_1 = R_2) - P(R_1 \neq R_2)$$

- Measures probability that rankings are concordant – probability that rankings are discordant

Kendall Tau Coefficient Computation

- Trick: need to consider all possible pairs
  - For each pair determine if C or D
- All possible pairs means for **each** of the two rankings
- Once we have Cs and Ds fully determined, count and report percentages.

Kendall Tau Coefficient Example

Doc	Rank X	Rank Y
A	1	2
B	2	3
C	3	1
D	4	4

Example:

- Sx (A) – Sx (B) = -
- Sy (A) – Sy (B) = -
- Concordant

- All possible pairs under Rank X:
  - [A, B] C
  - [A, C] D
  - [A, D] C
  - [B, C] D
  - [B, D] C
  - [C, D] C

- All possible pairs under Rank Y:
  - [C, A] D
  - [C, B] D
  - [C, D] C
  - [A, B] C
  - [A, D] C
  - [B, D] C

T = #C/total - #D/total = 8/12 – 4/12 = 4/12 = 0.33