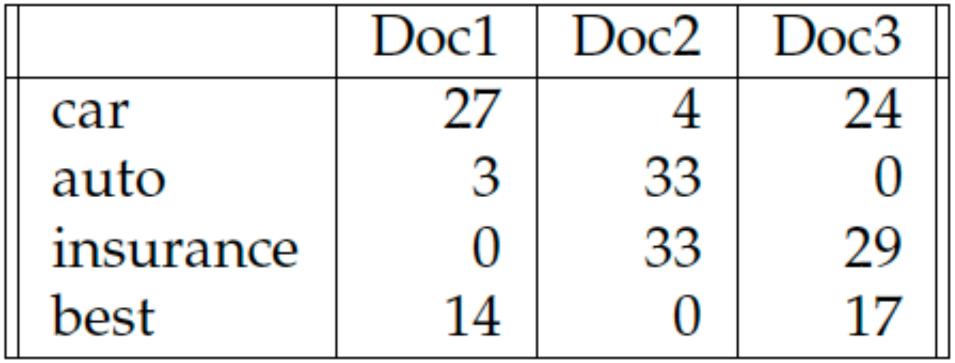
Exercise 6.8, 6.9, 6.10

- 6.8: Why is the idf of a term always finite?
- 6.9: What is the idf of a term that occurs in every document? Compare this with the use of stop word lists.
- 6.10: Consider the following table of term frequencies for 3 documents denoted Doc1, Doc2, Doc3. Compute the tf-idf weights (*logarithmic* variant) for the terms *car, auto, insurance, best*, for each document, using the idf values: $\text{idf}_{\text{car}}=1.65$; $\text{idf}_{\text{auto}}=2.08$, $\text{idf}_{\text{insurance}}=1.62$; $\text{idf}_{\text{best}}=1.5$



	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

Exercises 6.12, 6.13, 6.14

- 6.12: How does the base of the logarithm affect idf calculation? Convert the idf (using log base 10) to idf using log base 2.
- How does the change of the base of the logarithm affect the ratio of the scores of two documents on a given query? See slide 26 in part 5.
- 6.13: If the logarithm in the idf formula is computed base 2, suggest a simple approximation to the idf of a term.
- 6.14: If we stem jealous and jealousy to a common stem before setting up the vector space, detail how the definitions of tf and idf for the common stem must be computed using the term frequencies and document frequencies of the two original terms.

Exercises 6.15, 6.17

- 6.15: Recall the tf-idf weights computed in Exercise 6.10. Compute the Euclidean normalized document vectors for each of the documents. How many components have these vectors?
- 6.17: With term weights as computed in Exercise 6.15, rank the three documents by the computed score for the query "*car insurance*", for each of the following cases of term weighting in the query: *car, insurance*
  - 1. The weight of a term is 1 if present in the query, 0 otherwise.
  - 2. Euclidean normalized idf.

Exercise 6.18

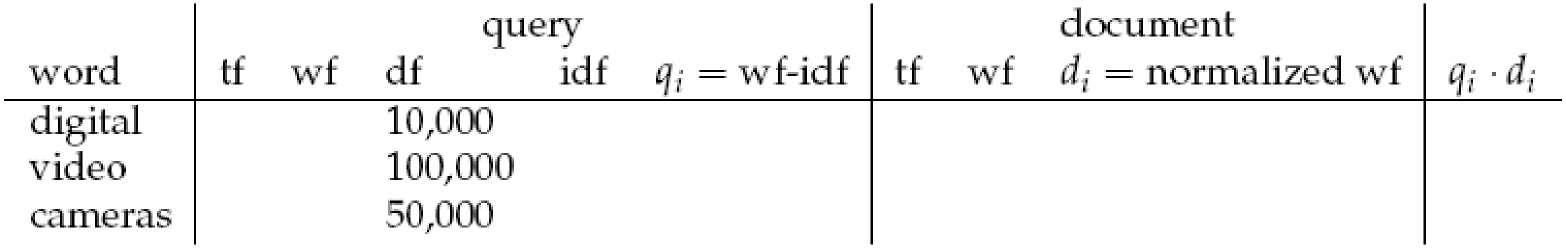
- One measure of the similarity of two vectors is the *Euclidean distance* (or L_2 distance) between them:

$$|\vec{x} - \vec{y}| = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2}$$

- Given a query *q* and documents d_1, d_2, . . ., we may rank the documents d_i in order of **increasing** Euclidean distance from *q*.
- Show that if *q* and the d_i are all normalized to unit vectors, then the rank ordering produced by Euclidean distance is identical to that produced by cosine similarities.

Exercise 6.19

- Compute the vector space similarity between the query “digital cameras” and the document “digital cameras and video cameras” by filling out the empty columns in Table below. Assume *N* = 10,000,000, logarithmic term weighting (wf columns) for query and document, idf weighting for the query only and cosine normalization for the document only. Treat and as a stop word. Enter term counts in the tf columns. What is the final similarity score?



			query				document		
word	tf	wf	df	idf	$q_i = \text{wf-idf}$	tf	wf	$d_i = \text{normalized wf}$	$q_i \cdot d_i$
digital			10,000						
video			100,000						
cameras			50,000						

Exercise

- Assume that you have a document collection C and you have computed for a particular document d and term t, $tf_{t,d}$ and idf_t . Imagine that the collection is updated and 2 new documents d’ and d’’ are added: one contains and the other does not contain the term t. Reply to the following questions explaining why:
  - a) After the addition of these 2 docs, does $tf_{t,d}$ increase, decrease or not change?
  - b) Explain if idf_t increases and when – find the threshold value v of df_t such that if df_t >=v, then idf_t increases.