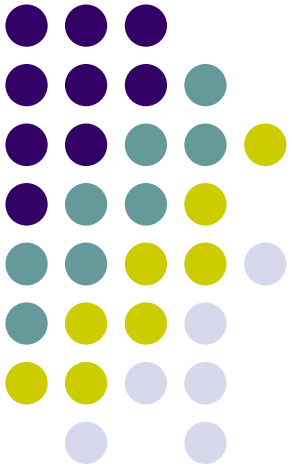
Applied Scaling & Classification Techniques in Political Science

A compressed sparse matrix



Sparsity problem

In text mining, huge matrices are created based on word frequencies with many cells having zero values

This problem is called **sparsity**

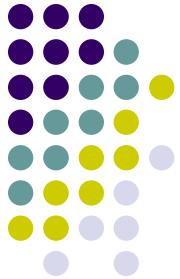
Dense Matrix

1	2	31	2	9	7	34	22	11	5
11	92	4	3	2	2	3	3	2	1
3	9	13	8	21	17	4	2	1	4
8	32	1	2	34	18	7	78	10	7
9	22	3	9	8	71	12	22	17	3
13	21	21	9	2	47	1	81	21	9
21	12	53	12	91	24	81	8	91	2
61	8	33	82	19	87	16	3	1	55
54	4	78	24	18	11	4	2	99	5
13	22	32	42	9	15	9	22	1	21

Sparse Matrix

1	.	3	.	9	.	3	.	.	.
11	.	4	.	.	.	.	.	2	1
.	.	1	.	.	.	4	.	1	.
8	.	.	.	3	1	.	.	.	.
.	.	.	9	.	.	1	.	17	.
13	21	.	9	2	47	1	81	21	9
.	.	.	.	.	.	.	.	.	.
.	.	.	.	19	8	16	.	.	55
54	4	.	.	.	11	.	.	.	.
.	.	2	.	.	.	.	22	.	21

Sparsity problem



The challenge in this sense is how to store a sparse matrix in a clever way (w/o all those 0s!) so that it occupies less space

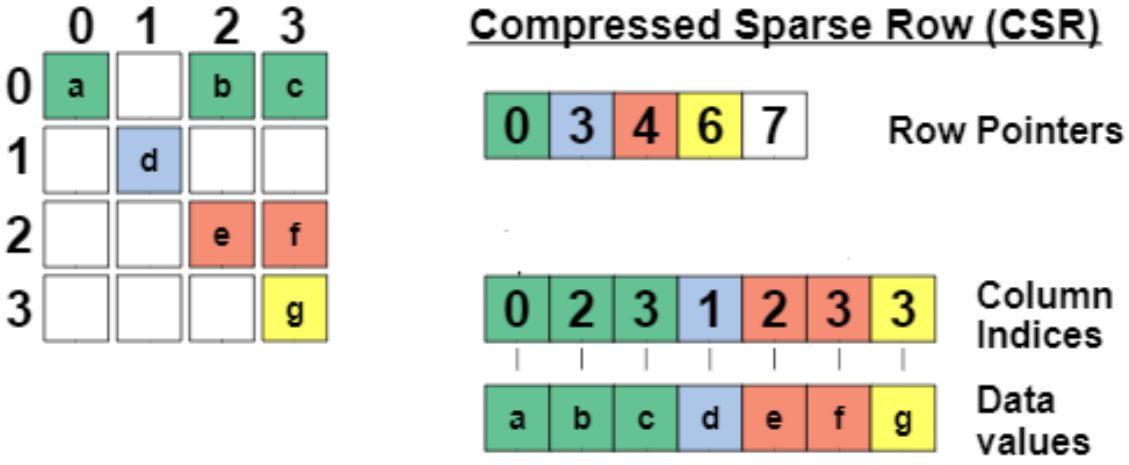
Being able of doing it can substantially (and I really mean that!) reduce the time spent by your computer in running any algorithm

Sparsity problem

There are several way of doing it

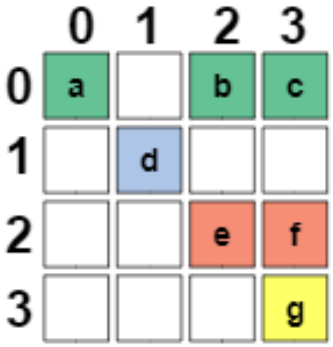
`Quanteda` stores the sparse matrix in a clever (and compressed) way! How?

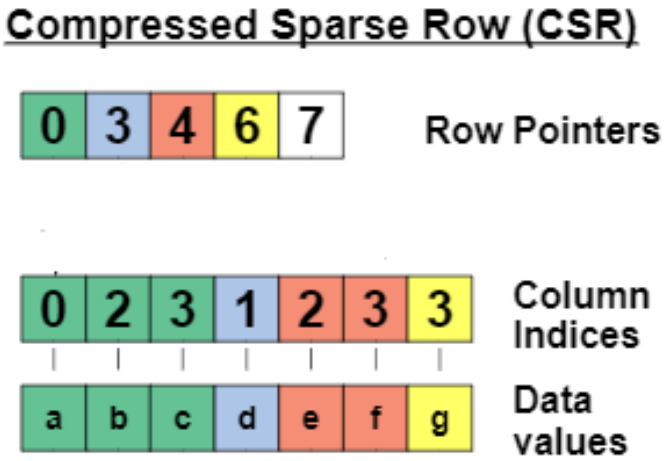
One possibility (just as an illustration): the CSR (the compressed sparse row)



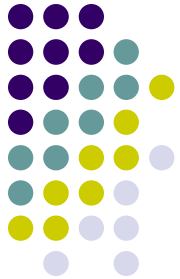
Sparsity problem

For the row pointers: always add a “0” at the beginning, and then for each row count the number of non-0 values and keep adding





Sparsity problem



However note that not all the packages in R allows you to work with compressed sparse matrices (and this could be a problem) – for example `randomForest`