Microtask Instructions

Download the **dataset** described here:
https://lists.wikimedia.org/pipermail/wiki-research-l/2016-April/005129.html
It consists of a list of all sections from all pages on the English Wikipedia (255MB packed / 1.4GB unpacked).

The task: Generate the list of the 100 most frequent section titles, each with the total number of how often they occurred in the dataset. Disregard leading and trailing spaces (" ") in a section title, e.g. treat "Early life" and "Early life " as identical.

Bonus task: Generate the list of the 100 section titles that are used in the largest number of articles, each with the percentage of articles that contain such a section. Caveat: Not all pages are articles. For example, there are also talk pages and help pages (see https://en.wikipedia.org/wiki/Wikipedia:Namespace). For the purposes of this bonus task, let's make the simplifying assumption that all entries in the dataset with no "/" and no "." in the page title correspond to articles, and vice versa that all page titles that contain a "/" or "." do not correspond to an article.

Background: While this is mostly an opportunity to examine Wikipedia content in general, it came up as a real life question during the Reading team's product development. We have been developing a feature showing an automated selection of "related articles" below a Wikipedia article, and wanted to know how many Wikipedia articles already contain a "See also" section, because it fulfills a similar function.

My analysis was done using:

- Jupyter 4.2.0
- Python 2.7.12
- Pandas 0.18.0

Task

Output Cell 18 in the Jupyter notebook has the mostFrequentTitles_DF dataframe with my answer of the most frequent section titles, sorted in descending order by their frequency.

Additional Notes:

- Input Error with line 23763217. I need to investigate why this line is problematic. When I look at in in Sublime Text, it seems to have only 4 tab separated fields and the lines above and below it

seem similar, so not sure why they didn't also cause errors

```
23763213   46304936    2014–15 FK Vardar season    5   Results
23763214   46304936    2014–15 FK Vardar season    5   Table
23763215   46304936    2014–15 FK Vardar season    4    Second phase (Championship group)
23763216   46304936    2014–15 FK Vardar season    5   Results
23763217   46304936    2014–15 FK Vardar season    5    Table
23763218   46304936    2014–15 FK Vardar season    3   Macedonian Cup
```

- "Licensing" and "Licensing:" both show up in the top 15 most frequent section titles
- Row 72 + 73 of mostFrequentTitles_DF are strange.

72	21068	Clerk, CheckUser,...
73	21059	Comments by other...

100 most frequent header titles:

	frequency	**section_title**
0	3933510	References
1	2249554	External links
2	1114379	See also
3	732295	Summary
4	695887	Licensing
5	509498	History
6	276974	Notes
7	162768	Career
8	144400	Biography
9	141217	Track listing
10	137631	Further reading
11	137035	Sources
12	111723	Bibliography
13	105465	Licensing:
14	103177	Geography
15	102048	Cast
16	101171	Early life
17	96112	Links
18	95009	Users

19	90384	Background
20	89379	Plot
21	89003	Personal life
22	80724	Reception
23	78673	Description
24	76854	Discography
25	76182	Demographics
26	75327	Awards
27	72011	Personnel
28	69550	Assessed
29	67931	Additions
30	61171	Results
31	60067	Education
32	54906	Reassessed
33	52637	Filmography
34	48354	Life
35	48223	Honours
36	45873	Production
37	45286	Works
38	42110	Gallery
39	38216	Overview
40	37731	Singles
41	36731	Discussion
42	36433	Footnotes
43	33370	Death
44	31548	Charts
45	31409	Distribution
46	29230	Population
47	28422	Publications
48	27651	Family

49	27532	Legacy
50	26709	Entry
51	26542	Selected additions
52	26297	Final
53	26037	Events
54	25631	Music
55	25329	Species
56	25034	Usage
57	25006	Albums
58	24846	Books
59	24603	Television
60	24093	Club career
61	24084	Source
62	23870	Economy
63	23587	Early years
64	23384	Removed
65	23123	Early life and education
66	22888	Critical reception
67	22465	International career
68	22446	Schedule
69	22347	Location
70	21785	Professional career
71	21327	Notes and references
72	21068	Clerk, CheckUser,...
73	21059	Comments by other...
74	20745	Career statistics
75	20677	Notable people
76	20606	Synopsis
77	20173	Chart performance
78	19891	Other

79	19545	Etymology
80	19536	Playing career
81	18636	Climate
82	18441	Development
83	18001	Politics
84	17915	Political career
85	17830	Deaths
86	17817	Transportation
87	17434	Fair use rationale
88	17309	People
89	17117	Aftermath
90	17002	Life and career
91	16894	Film
92	16354	Births
93	16343	Release
94	16318	Early career
95	16254	Soundtrack
96	16223	Music video
97	16211	Members
98	15938	Club
99	15923	Regular season

Bonus Task

API Ettiquette: https://www.mediawiki.org/wiki/API:Etiquette

Idea:
I could simply filter out any page titles which contain "/" or "." to keep articles, but if I do that, it might leave non-articles in the analysis or incorrectly drop some articles. I came up with a more sophisticated approach and it would be great to get your input. I can make a Python list of the 6.3 million unique page_ids in this dataset, write a loop to make an API call to the Wikipedia API using urllib3, and get back

the "ns" field. If ns = 0, I can keep that row, but if I get anything else, I'll drop it. Let me know your thoughts.

Reply:
That sounds like a cool idea, but you will want to be aware of the API rate limits and other policies (like on user agents). If you go ahead with that path, please note as part of the solution (e.g. as comments in the code) how this issue was taken into account.

A more efficient and natural way would be to simply regenerate the dataset (from a more up to date dump too) as in Aaron's original PAWS notebook, while restricting it to articles or including the namespace as an additional column. I was going to include this as part of the microtask, but unfortunately it seems the mwparserfromhell package (as used there) is no more available on PAWS currently. Still, it might be worth trying to get that fixed instead.

Update:
I wrote a loop to make an API call to the first 200 unique page_ids and tested how long each API call would take. The time varied from .09 seconds - .46 seconds for each call. If we assume, on average each API call will take .25 seconds, then it would take 18.22 days to make 6.3 million individual API calls. It's possible to make multiple API calls at the same time, but this would not follow the API Etiquette, which suggests making requests in series rather than in parallel to avoid adding extra load.

I added my own User-Agent header while making these requests.

```
user_agent = {'user-agent': 'ZareenFarooquiOutreachyTask/1.0 (https://zareenfarooqui.com; zareenf@gmail.com)'}
http = urllib3.PoolManager(num_pools=1, maxsize=20, headers=user_agent, cert_reqs='CERT_REQUIRED',
    ca_certs=certifi.where())
r1 = http.urlopen('GET', 'http://httpbin.org/headers')
print(r1.data)
```

```
{
  "headers": {
    "Accept-Encoding": "identity",
    "Host": "httpbin.org",
    "User-Agent": "ZareenFarooquiOutreachyTask/1.0 (https://zareenfarooqui.com; zareenf@gmail.com)"
  }
}
```

100 most frequent header titles, along with their frequency and percentage (filtered out by only articles):

	header_frequency	header_title	header_percent
0	3792066	References	82.002300
1	2168509	External links	46.893363
2	1060642	See also	22.936068
3	491924	History	10.637710

4	263258	Notes	5.692876
5	152605	Career	3.300038
6	135144	Track listing	2.922449
7	132480	Further reading	2.864841
8	130716	Biography	2.826695
9	106534	Bibliography	2.303766
10	102130	Geography	2.208531
11	100152	Cast	2.165757
12	93913	Sources	2.030841
13	93566	Early life	2.023337
14	87526	Plot	1.892724
15	84240	Background	1.821665
16	83965	Personal life	1.815718
17	78534	Reception	1.698274
18	75219	Demographics	1.626589
19	75189	Discography	1.625940
20	74200	Description	1.604553
21	70968	Awards	1.534662
22	69402	Personnel	1.500798
23	57946	Results	1.253065
24	55912	Education	1.209080
25	51004	Filmography	1.102946
26	45485	Life	0.983600
27	44881	Honours	0.970538
28	44628	Production	0.965067
29	42711	Works	0.923613
30	41026	Gallery	0.887175
31	36871	Singles	0.797324
32	36359	Overview	0.786253
33	34791	Footnotes	0.752345

34	31317	Distribution	0.677221
35	30425	Charts	0.657932
36	30265	Death	0.654472
37	29181	Population	0.631030
38	26104	Family	0.564491
39	25960	Final	0.561377
40	25718	Publications	0.556144
41	25500	Legacy	0.551430
42	25306	Events	0.547235
43	25282	Species	0.546716
44	25007	Music	0.540769
45	24512	Albums	0.530065
46	24082	Television	0.520766
47	24065	Club career	0.520398
48	23698	Economy	0.512462
49	22421	International career	0.484847
50	22405	Books	0.484501
51	22298	Critical reception	0.482188
52	22134	Schedule	0.478641
53	22020	Early years	0.476176
54	21861	Location	0.472738
55	20965	Professional career	0.453362
56	20645	Career statistics	0.446442
57	20524	Notable people	0.443825
58	20511	Notes and references	0.443544
59	20195	Synopsis	0.436711
60	20070	Early life and education	0.434008
61	19699	Chart performance	0.425985
62	19474	Etymology	0.421119
63	19311	Playing career	0.417595

64	18539	Climate	0.400900
65	18538	Other	0.400879
66	17688	Development	0.382498
67	17543	Transportation	0.379362
68	17333	Politics	0.374821
69	17016	Deaths	0.367966
70	16899	People	0.365436
71	16582	Aftermath	0.358581
72	16538	Film	0.357629
73	15993	Soundtrack	0.345844
74	15945	Life and career	0.344806
75	15935	Release	0.344590
76	15852	Music video	0.342795
77	15824	Transport	0.342189
78	15816	Political career	0.342016
79	15638	Members	0.338167
80	15570	Club	0.336697
81	15557	Births	0.336416
82	15441	Seeds	0.333907
83	15426	Early career	0.333583
84	15280	Regular season	0.330425
85	14789	Plot summary	0.319808
86	14674	Characters	0.317321
87	14652	Awards and nominations	0.316845
88	14437	Gameplay	0.312196
89	14253	Literature	0.308217
90	14084	In popular culture	0.304562
91	13987	Citations	0.302465
92	13893	Selected filmography	0.300432
93	13633	Episodes	0.294810

94	13540	Sports	0.292798
95	13523	Places	0.292431
96	13464	Draw	0.291155
97	12829	Statistics	0.277423
98	12629	Notable alumni	0.273098
99	12627	Roster	0.273055

Note: I divided frequency by number of unique page_titles (4624341) to get percentage. The number of unique page_ids was different (4631002).