```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Extract and organize metadata and text of MMWR

@author: chadheilig

Revive mmwr_dframe from pickle file (14571 x 8)
Filter to article subset (9981 x 3)
Extract and parse metadata from <div class="dateline">; correct errors (2711 x 8)
Extract and parse metadata from <title>, <meta>, <link> (9981 x 33)
Obtain and construct Almetric scores; correct DOI errors (1242 x 4)
Construct 41 groupings; merge with Altmetric (1241 x 10)
Append previous with boolean for articles on which consulted (51); (1241 x 10)
Merge source metadata, Altmetrics, consultation, and groupings (1241 x 11)
Construct candidate list: consulted or top-2 ranks in each group (1241 x 11)
Construct selected list of 45 articles; output to Excel

"""

#%% Import modules and set up environment
# import from 0_cdc-corpora-header.py

import time
from dateutil.parser import parse
import copy
from bs4 import SoupStrainer
import numpy as np

os.chdir('/Users/cmheilig/cdc-corpora/_test')
MMWR_BASE_PATH_u3 = normpath(expanduser('~/cdc-corpora/mmwr_u3/'))

# MMWR DataFrame, reduced to 2 columns for articles only
mmwr_dframe = pickle.load(open('pickle-files/mmwr_dframe.pkl', 'rb'))
# mmwr_dframe.filename.str.match('mm|rr|ss|su')
mmwr_art_frame = mmwr_dframe.loc[\
    (mmwr_dframe.level == 'article') & mmwr_dframe.filename.str.match('mm|rr|ss|su'),
    'filename':'string']
mmwr_art_frame['cat'] = mmwr_art_frame.filename.str[:2].astype('category')
mmwr_art_frame.index = mmwr_art_frame.filename.str.split(".").str[0]
mmwr_art_frame.drop(columns='filename', inplace=True)
mmwr_art_frame.sort_index(inplace=True)
# [9981 rows x 3 columns]
# pickle.dump(mmwr_art_frame, open('mmwr_art_frame.pkl', 'wb'))

#%% Read HTML from mirror into list of strings

# mmwr_art_html = [read_uni_html(MMWR_BASE_PATH_u3 + path)
#                         for path in tqdm(mmwr_art_frame.mirror_path)]
# 14620/14620 [00:08<00:00, 1662.37it/s]
# pickle.dump(mmwr_art_html, open('mmwr_art_html.pkl', 'wb'))
# mmwr_art_html = pickle.load(open('mmwr_art_html.pkl', 'rb'))

mmwr_art_html = [html_reduce_space_u(read_uni_html(MMWR_BASE_PATH_u3 + path))
                      for path in tqdm(mmwr_art_frame.mirror_path)]
# 9981/9981 [01:14<00:00, 133.14it/s]
```

```
# pickle.dump(mmwr_art_html, open('mmwr_art_html.pkl', 'wb'))
# mmwr_art_html = pickle.load(open('mmwr_art_html.pkl', 'rb'))

#%% Parse pubilcation category, date, and volume(issue);pages

only_dateline = SoupStrainer(name='div', class_='dateline')
mmwr_dl_soup = [
    BeautifulSoup(html, 'lxml', parse_only=only_dateline).find('div', class_='dateline')
                for html in tqdm(mmwr_art_html)]
# 9981/9981 [05:18<00:00, 31.37it/s]
# mmwr_dl_soup[6605]

# mmwr_dl_soup = [
#     BeautifulSoup(html, 'lxml').find('body').find('div',
class_='dateline')#.get_text(strip=True)
#                 for html in tqdm(mmwr_art_html)]
# # 9981/9981 [15:59<00:00, 10.40it/s]
# mmwr_dl_soup[:5]

mmwr_dl_text = [{ file.split('.')[0].lower(): soup.get_text(strip=True) }
                    for file, soup in zip(mmwr_art_frame.index, mmwr_dl_soup)
                    if soup is not None]
# mmwr_dl_text[:3]

re_dateline = re.compile(r'''
                         (?P<dl_string>             # delimit whole string
                         (?P<dl_category>[\w\s]+\b) # category
                         \s*?/\s*?                  # forward slash delimiter
                         (?P<dl_date>\w[,\s\w]+\w)  # date
                         \s*/\s*                    # forward slash delimiter
                         (?P<dl_volume>[-\d]+)      # volume
                         \(                         # paren delimiter
                         (?P<dl_issue>[-\d]+)       # issue
                         \)                         # paren delimiter
                         ;?\s?                      # semicolon delimiter
                         (?P<dl_page0>\d*)          # first page number
                         (?P<dl_delim>[\D]*)        # page range delimiter
                         (?P<dl_page1>\d*)          # last page number
                         )''', re.VERBOSE | re.ASCII)
mmwr_dl_list = [
    dict(dl_item_id=dl_item_id, **re.match(re_dateline, text).groupdict())
    # (file, re.match(re_dateline, text))
        for dl in mmwr_dl_text
        for dl_item_id, text in dl.items()]
# len([i for i, j in enumerate(mmwr_dl_list) if j[1] is None])
# mmwr_dl_text[_]

# mmwr_dl_text[2548]
# mmwr_dl_list[2548][1].groupdict()
mmwr_dl_df = pd.DataFrame(mmwr_dl_list) # 2711 x 9
# mmwr_dl_df.to_excel('mmwr_dl_df.xlsx')

# create categorical cat with values mm, rr, ss, su
# convert category to categorical, date to ISO date
# convert volume, issue, page0, page1 to integer
mmwr_dl_df.drop(columns='dl_delim', inplace=True)
```

```
112 mmwr_dl_df.set_index('dl_item_id', inplace=True)
113 mmwr_dl_df['dl_category'] = mmwr_dl_df['dl_category'].astype('category')
114 mmwr_dl_df['dl_cat'] = mmwr_dl_df.index.str[:2].astype('category')
115 mmwr_dl_df['dl_date'] = pd.to_datetime(mmwr_dl_df['dl_date'])
116 for _col in ['dl_volume', 'dl_issue', 'dl_page0', 'dl_page1']:
117     mmwr_dl_df[_col] = \
118         pd.to_numeric(mmwr_dl_df[_col], downcast='integer', errors='coerce').\
119             astype('Int64')
120
121 # ad hoc corrections to date, issue, page0, page1
122 dl_corrections = [\
123  {'dl_item_id': 'mm6518e1', 'dl_page0': 474},
124  {'dl_item_id': 'mm6518e2', 'dl_page0': 475, 'dl_page1': 478},
125  {'dl_item_id': 'mm6518e3', 'dl_page0': 479, 'dl_page1': 480},
126  {'dl_item_id': 'mm6520e1', 'dl_page0': 514, 'dl_page1': 519},
127  {'dl_item_id': 'mm6521e1', 'dl_page0': 543, 'dl_page1': 546},
128  {'dl_item_id': 'mm6524e2', 'dl_page0': 627, 'dl_page1': 628},
129  {'dl_item_id': 'mm6524e3', 'dl_page0': 629, 'dl_page1': 635},
130  {'dl_item_id': 'mm6525e1', 'dl_page0': 650, 'dl_page1': 654},
131  {'dl_item_id': 'mm6526e1', 'dl_page0': 672, 'dl_page1': 677},
132  {'dl_item_id': 'mm655051e1', 'dl_issue': 5051},
133  {'dl_item_id': 'mm6645a2', 'dl_page0': 1248, 'dl_page1': 1251},
134  {'dl_item_id': 'mm6946e1', 'dl_issue': 46},
135  {'dl_item_id': 'mm695152a1', 'dl_page0': 1933, 'dl_page1': 1937},
136  {'dl_item_id': 'mm695152a2', 'dl_page0': 1938, 'dl_page1': 1941},
137  {'dl_item_id': 'mm695152a3', 'dl_page0': 1942, 'dl_page1': 1947},
138  {'dl_item_id': 'mm695152a4', 'dl_page0': 1948, 'dl_page1': 1952},
139  {'dl_item_id': 'mm695152e1', 'dl_page0': 1953, 'dl_page1': 1956},
140  {'dl_item_id': 'mm695152e2', 'dl_page0': 1957, 'dl_page1': 1960},
141  {'dl_item_id': 'mm695152a5', 'dl_page0': 1961, 'dl_page1': 1962},
142  {'dl_item_id': 'mm695152a6', 'dl_page0': 1963},
143  {'dl_item_id': 'mm695152a7', 'dl_page0': 1963},
144  {'dl_item_id': 'mm695152a8', 'dl_page0': 1964},
145  {'dl_item_id': 'mm695152a9', 'dl_page0': 1965},
146  {'dl_item_id': 'mm7017e3', 'dl_issue': 17},
147  {'dl_item_id': 'mm6945a7', 'dl_date': np.datetime64('2020-11-13')},
148  {'dl_item_id': 'rr6804a1', 'dl_date': np.datetime64('2019-12-13')}]
149 # rows with values to correct
150 mmwr_dl_df.\
151     loc[mmwr_dl_df.index.isin([x.get('dl_item_id') for x in dl_corrections])].\
152     iloc[:,[0,2,3,4,5,6]]
153
154 # the sort by cat, vol, iss, page0, page1, date, file
155 # there must be a more elegant way to do this
156 z_df = mmwr_dl_df.copy()
157 dl_copy = copy.deepcopy(dl_corrections)
158 for _dict in dl_copy:
159     item_id = _dict.pop('dl_item_id')
160     mmwr_dl_df.loc[mmwr_dl_df.index == item_id, list(_dict)] = list(_dict.values())
161
162 mmwr_dl_df.sort_values(['dl_cat', 'dl_volume', 'dl_issue', 'dl_page0',
163                         'dl_page1', 'dl_date', 'dl_item_id'], inplace=True)
164 mmwr_dl_df.to_pickle('mmwr_dl_df.pkl')
165 # mmwr_dl_df.to_excel('mmwr_dl_df.xlsx')
166
167 # del dl_copy, dl_corrections, item_id, only_dateline, re_dateline, z_df
```

```
168
169 #%% Determine metadata elements to extract from HTML head elements
170 only_head = SoupStrainer(name='head')
171 mmwr_head_soup = [BeautifulSoup(html, 'lxml', parse_only=only_head)
172                 for html in tqdm(mmwr_art_html)]
173 # 9981/9981 [04:55<00:00, 33.73it/s]
174
175 # mmwr_art_soup[200]
176 # y = [x.attrs for x in mmwr_art_soup[200].find_all(name=True)]
177
178 mmwr_head_meta = [
179     dict(head_item_id=item_id, tagname=tag.name, **tag.attrs)
180         for item_id, soup in zip(mmwr_art_frame.index, mmwr_head_soup)
181         for tag in soup.find_all(name=True)] # list with 340,263 dicts
182 mmwr_head_meta_df = pd.DataFrame(mmwr_head_meta) # 340263 x 19
183 # mmwr_head_meta_df.to_excel('all-head-tags.xlsx')
184 # values for meta name tags and meta property tags
185 mmwr_head_meta_df.loc[(mmwr_head_meta_df.tagname == 'meta'), 'name'].value_counts().index
186 mmwr_head_meta_df.loc[(mmwr_head_meta_df.tagname == 'meta'),
186 'property'].value_counts().index
187
188 # parse again, harvesting a narrower set of tags
189 # title
190 # link href when rel='canonical' -> l_href
191
192 # labels for content of meta name <meta name=. content=.>
193 names_capture = ['Volume', 'Issue', 'Issue_Num', 'Page', 'Date',
194     'Year', 'Month', 'Day', 'MMWR_Type', 'Keywords',
195     'keywords', 'Description', 'description', 'citation_categories',
196     'citation_title', 'citation_author', 'citation_publication_date',
197     'citation_volume', 'citation_doi', 'DC.date',
198     'cdc:last_published', 'twitter:description', 'twitter:domain']
199 names_rename =  ['hm_Volume', 'hm_Issue', 'hm_Issue_Num', 'hm_Page', 'hm_Date',
200     'hm_Year', 'hm_Month', 'hm_Day', 'hm_MMWR_Type', 'hm_Keywords',
201     'hm_keywords', 'hm_Description', 'hm_description', 'hm_citation_categories',
202     'hm_citation_title', 'hm_citation_author', 'hm_citation_publication_date',
203     'hm_citation_volume', 'hm_citation_doi', 'hm_DC_date',
204     'hm_cdc_last_published', 'hm_twitter_description', 'hm_twitter_domain']
205 names_remap = dict(zip(names_capture, names_rename))
206
207 # labels for content of meta property <meta property=. content=.>
208 props_capture = ['cdc:first_published', 'cdc:last_updated',
209     'cdc:last_reviewed', 'cdc:content_id', 'article:published_time',
210     'og:title', 'og:description', 'og:url']
211 props_rename =  ['hm_cdc_first_published', 'hm_cdc_last_updated',
212     'hm_cdc_last_reviewed', 'hm_cdc_content_id', 'hm_article_published_time',
213     'hm_og_title', 'hm_og_description', 'hm_og_url']
214 props_remap = dict(zip(props_capture, props_rename))
215
216 def mmwr_head_meta_fn(soup):
217     result = { k: None for k in ['h_title', 'hl_href_canonical'] +
218                                  names_rename + props_rename }
219     h_title = soup.find('title')
220     result['h_title'] = '' if h_title is None else h_title.get_text(strip=True)
221     hl_href_canonical = soup.find('link', rel='canonical')
222     result['hl_href_canonical'] = '' if hl_href_canonical is None else \
```

```
223         hl_href_canonical.get('href')
224     meta_tags = soup.find_all(name='meta')
225     for meta_tag in meta_tags:
226         meta_attrs = meta_tag.attrs
227         if meta_attrs.get('name') in names_capture:
228             # print(meta_attrs)
229             result[ names_remap[meta_attrs.get('name')] ] = meta_attrs.get('content')
230         elif meta_attrs.get('property') in props_capture:
231             result[ props_remap[meta_attrs.get('property')] ] = meta_attrs.get('content')
232     return result
233
234 # mmwr_head_meta_fn(mmwr_head_soup[2166])
235
236 mmwr_head_meta = [
237     dict(h_item_id=item_id, **mmwr_head_meta_fn(soup))
238         for item_id, soup in zip(mmwr_art_frame.index, mmwr_head_soup)]
239 # list with 9,981 dicts
240 mmwr_head_meta_df = pd.DataFrame(mmwr_head_meta) # 9981 x 34
241 mmwr_head_meta_df.set_index('h_item_id', inplace=True)
242 mmwr_head_meta_df.sort_index(inplace=True)       # 9981 x 33
243 # mmwr_head_meta_df.to_excel('selected-head-tags.xlsx')
244
245 # number of unique values in each column
246 {col: mmwr_head_meta_df[col].value_counts().size for col in mmwr_head_meta_df.columns}
247
248 # del names_capture, names_remap, names_rename, only_head, props_capture, props_remap,
248 props_rename
249
250 #%% Altmetric
251 import json
252 # import datetime as dt
253
254 # /Users/cmheilig/cdc-corpora/_test/mmwr-altmetric_20220529/14.json
255 # with open('mmwr-altmetric_20220529/14.json', 'r') as jfile:
256 #     x = json.load((jfile)
257 # x = json.load(open('mmwr-altmetric_20220529/14.json', 'r'))
258
259 # first load all the data
260 mmwr_altm_dict = [j for i in range(1, 15)
261     for j in json.load(open(f'mmwr-altmetric_20220529/{i:02d}.json', 'r'))['results']]
262 # 1400 entries
263
264 # then process it to yield a list of dicts with keys
265 #    doi, cited_by_*, scirem last_updated, details_url
266 mmwr_altm_sub = [{k: v for k, v in madict.items()
267     if k in {'doi', 'score', 'last_updated', 'details_url'} or k.startswith('cited_by_')}
268     for madict in mmwr_altm_dict]
269 # pd.DataFrame(mmwr_altm_sub).to_excel('mmwr_altmetric.xlsx')
270
271 mmwr_altm_df = pd.DataFrame([
272     {k: v for k, v in madict.items() if k in {'doi', 'score'}}
273         for madict in mmwr_altm_dict])
274 # 1400 x 2
275 mmwr_altm_df['am_score'] = mmwr_altm_df['score'].round(decimals=3)
276 mmwr_altm_df.drop(columns='score', inplace=True)
277
```

```
278 # find and correct improper DOIs
279 re_item_id = re.compile(r'10\.15585/mmwr\.(mm|rr|ss|su)(\d{4}|\d{6})(a|e)\d{1,2}')
280 mmwr_altm_df.doi.loc[~mmwr_altm_df.doi.str.fullmatch(re_item_id)].to_dict()
281 altm_doi_corrections = {'doi':
282  {'10.15585/mmwr.7009a4':      '10.15585/mmwr.mm7009a4',
283   '10.15585/mm7007a6':         '10.15585/mmwr.mm7007a6',
284   '10.15585/mmwr':             '10.15585/mmwr.mm6944e1',
285   '10.15585/mmwr.ss.6809a1':   '10.15585/mmwr.ss6809a1',
286   '10.15585/mmwr,mm6743a1':    '10.15585/mmwr.mm6743a1',
287   '10.15585/mmwr.mm6751521e1': '10.15585/mmwr.mm675152e1'}}
288 mmwr_altm_df['doi'].replace(altm_doi_corrections['doi'], inplace=True)
289 mmwr_altm_df['am_item_id'] = mmwr_altm_df.doi.str.split('.', expand=True)[2]
290 mmwr_altm_df.drop(columns='doi', inplace=True)
291
292 mmwr_altm_df['am_cat'] = mmwr_altm_df.am_item_id.str[:2]
293 mmwr_altm_df['am_volume'] = pd.to_numeric(mmwr_altm_df.am_item_id.str[2:4]) # 1..51
294 mmwr_altm_df['am_issue'] = pd.to_numeric(mmwr_altm_df.am_item_id.str[4:6]) # 65..71
295 # limit to mm68 - mm71
296 mmwr_altm_df = mmwr_altm_df.loc[\
297     (mmwr_altm_df.am_cat == 'mm') &
298      mmwr_altm_df.am_volume.isin([68, 69, 70, 71])]
299 mmwr_altm_df.sort_values(['am_volume', 'am_issue', 'am_item_id'], inplace=True)
300 # 1242 x 4
301 mmwr_altm_df.set_index('am_item_id', inplace=True)
302 # mmwr_altm_df.sort_index(inplace=True)
303
304 # mmwr_altm_df.to_excel('mmwr_altmetric.xlsx')
305
306 #%% Groups of 41 sets of contiguous issues with about 21 full reports per group
307 mmwr_gp_df = pd.DataFrame({
308  'gp_cat': ['mm'] * 175,
309  'gp_volume': [
310      68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68,
311      68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68,
312      68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 69, 69, 69, 69, 69,
313      69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69,
314      69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69,
315      69, 69, 69, 69, 69, 69, 69, 69, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70,
316      70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70,
317      70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70,
318      70, 70, 71, 71, 71, 71, 71, 71, 71, 71, 71, 71, 71, 71, 71, 71, 71, 71, 71,
319      71, 71, 71, 71],
320   'gp_issue': [
321      1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
322      22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
323      41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 53, 1, 2, 3, 4, 5, 6, 7, 8, 9,
324      10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28,
325      29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
326      48, 49, 50, 51, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
327      18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
328      37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 1, 2, 3, 4, 5,
329      6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21],
330   'gp_group': [
331      1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4,
332      4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 8, 8, 8,
333      8, 8, 8, 9, 9, 9, 9, 10, 10, 10, 10, 10, 11, 11, 11, 11, 12, 12, 12, 12,
```

```
     13, 13, 13, 13, 14, 14, 14, 15, 15, 15, 16, 16, 16, 16, 17, 17, 17, 18, 18,
     18, 19, 19, 19, 20, 20, 21, 21, 21, 22, 22, 22, 23, 23, 23, 24, 24, 24, 24,
     25, 25, 25, 26, 26, 26, 27, 27, 27, 28, 28, 28, 28, 29, 29, 29, 30, 30, 30,
     30, 31, 31, 31, 31, 31, 31, 32, 32, 32, 33, 33, 33, 33, 34, 34, 34, 35, 35,
     35, 35, 35, 36, 36, 36, 36, 37, 37, 37, 37, 38, 38, 38, 38, 39, 39, 39, 40,
     40, 40, 40, 40, 41, 41, 41, 41, 41, 41]})
mmwr_gp_df['gp_cat'] = mmwr_gp_df['gp_cat'].astype('category')

# Merge Altmetric with groups and calculate within-group ranks
# merge with how='outer', indicator=True shows that
# Altmetric set does not include 68(53); groups do not include 71(22)
mmwr_am_gp_df = pd.merge(mmwr_altm_df.reset_index(), mmwr_gp_df, how='inner',
         left_on=['am_volume', 'am_issue'], right_on=['gp_volume', 'gp_issue'])
# 1241 x 9

#%% Published reports, volumes 68-71, on which we were consulted late in production
consulted = [\
    'mm6802a1', 'mm6827a2', 'mm6834a3', 'mm6844a1', 'mm6846a2', 'mm6906a3',
    'mm6911a5', 'mm6913e2', 'mm6923e4', 'mm6924e1', 'mm6924e2', 'mm6925a1',
    'mm6926e1', 'mm6927a4', 'mm6928e3', 'mm6929e1', 'mm6932a1', 'mm6932e5',
    'mm6935a2', 'mm6935e2', 'mm6937a2', 'mm6938a1', 'mm6943e3', 'mm6944e3',
    'mm6947e2', 'mm6949a2', 'mm7001a4', 'mm7005a4', 'mm7006e2', 'mm7007a4',
    'mm7010e3', 'mm7011e3', 'mm7022e2', 'mm7023e2', 'mm7032e1', 'mm7032e3',
    'mm7039e3', 'mm7041a2', 'mm7044e1', 'mm705152a2', 'mm705152a3',
    'mm705152e2', 'mm7102a2', 'mm7106e1', 'mm7107e1', 'mm7109a1', 'mm7112a1',
    'mm7118a4', 'mm7120a1', 'mm7121a1', 'mm7121a2']

mmwr_am_gp_df['consulted'] = mmwr_am_gp_df.am_item_id.isin(consulted)
# mmwr_am_gp_df['consulted_'] = mmwr_am_gp_df.am_item_id.isin(consulted).\
#     map(lambda x: 'x' if x else '')
mmwr_am_gp_df.set_index('am_item_id', inplace=True) # 1241 x 9

#%% Merge MMWR metadata (dateline, <head>), Altmetric, and consulted

mmwr_review_df = pd.merge(how='inner', right=mmwr_dl_df, left=mmwr_am_gp_df,
    left_index=True, right_index=True) # 1241 x 17
mmwr_review_df = pd.merge(how='inner', right=mmwr_review_df,
    left=mmwr_head_meta_df.loc[mmwr_head_meta_df.hm_citation_categories == 'Full Report'],
    left_index=True, right_index=True) # 846 x 50

# Compute ranks of full reports within groups
mmwr_review_df['am_rank'] = \
    mmwr_review_df.groupby('gp_group')['am_score'].\
        rank(method='min', ascending=False).astype('Int64')
# 846 x 51
mmwr_review_df['candidate'] = (\
    mmwr_review_df['consulted'] | mmwr_review_df.am_rank.isin([1,2]))
# 846 x 52

selected = [\
    'mm6802a1', 'mm6806a2', 'mm6817a3', 'mm6827a2', 'mm6834a3', 'mm6841e3',
    'mm6844a1', 'mm6848a1', 'mm6903a1', 'mm6906a3', 'mm6911a5', 'mm6916e1',
    'mm6920e2', 'mm6924e1', 'mm6927a4', 'mm6932e5', 'mm6935a2', 'mm6936a5',
    'mm6939e2', 'mm6943e3', 'mm6944e3', 'mm6947e2', 'mm6949a2', 'mm7001a4',
    'mm7004e3', 'mm7006e2', 'mm7010e3', 'mm7010e4', 'mm7013e3', 'mm7018e1',
    'mm7021e1', 'mm7023e2', 'mm7031e1', 'mm7032e3', 'mm7034e5', 'mm7037e1',
```

```
390     'mm7039e3', 'mm7043e2', 'mm7047e1', 'mm705152a3', 'mm7104e1', 'mm7110e1',
391     'mm7114e1', 'mm7121a2', 'mm7121e1']
392 mmwr_review_df['selected'] = mmwr_review_df.index.isin(selected)
393 # 846 x 53
394 
395 # subset and rerder columns
396 mmwr_review_df = mmwr_review_df[[\
397     'dl_date', 'dl_volume', 'dl_issue', 'dl_page0', 'dl_page1', 'gp_group',
398     'am_score', 'am_rank', 'consulted', 'candidate', 'selected',
399     'h_title', 'hm_keywords', 'hm_description', 'hm_citation_author',
400     'hl_href_canonical', 'hm_citation_doi', 'dl_string']] # 846 x 18
401 # trim ' | MMWR' from right-hand side of title
402 mmwr_review_df['h_title'] = mmwr_review_df['h_title'].str[:-7]
403 
404 mmwr_review_df = mmwr_review_df.\
405     reset_index().\
406     rename(columns={'index': 'item_id'}).\
407     sort_values(['dl_volume', 'dl_issue', 'dl_page0', 'dl_page1', 'dl_date', 'item_id']).\
408     set_index('item_id') # 846 x 18
409 
410 mmwr_review_df.to_excel('mmwr_review_df.xlsx')
```