```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Analyze the structure and broad properties of EID online archive

@author: chadheilig

Sections of this script, based on levels of EID archive:
0. EID  home https://www.cdc.gov/eid/ (home)
1. List of volumes (and some articles)
2. List and contents of volumes
3. List and contents of issues (tables of contents)
4. List of articles
3. Complete list of EID files

Main product: eid_dframe
"""

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

os.chdir('/Users/cmheilig/cdc-corpora/_test')

#%% 0. Start with EID home https://wwwnc.cdc.gov/eid/

base_url = 'https://wwwnc.cdc.gov/eid/'
pd.DataFrame([process_aTag(aTag, base_url)
    for aTag in BeautifulSoup(get_html_from_url(base_url), 'lxml').\
    find_all('a', href=True)]).\
    to_excel('eid-base-anchors.xlsx', engine='openpyxl')
# [342 rows x 7 columns]

home_a = BeautifulSoup(get_html_from_url(base_url), 'lxml').\
   find('a', href=re.compile('eid'),
            string=re.compile('EID Journal'))
# process_aTag(home_a, base_url)
# {'base': 'https://wwwnc.cdc.gov/eid/',
#  'href': '/eid/',
#  'url': 'https://wwwnc.cdc.gov/eid/',
#  'path': '/eid/',
#  'filename': '',
#  'mirror_path': '/eid/index.html',
#  'string': 'EID Journal'}

home_dframe = pd.DataFrame(process_aTag(home_a, base_url), index = [0])
# home_dframe.loc[:, ['path', 'string']]
#     path       string
# 0  /eid/  EID Journal
home_html = get_html_from_url(home_dframe.url[0]) # len(home_html) # 747171
home_soup = BeautifulSoup(home_html, 'lxml')

# review all anchor-hrefs from home URL
# len(home_soup.find_all('a', href=True)) # 397
# pd.DataFrame([process_aTag(aTag, home_dframe.url[0])
#     for aTag in home_soup.find_all('a', href=True)]).\
#     to_excel('eid-home-anchors.xlsx', engine='openpyxl')
```

```
# same as eid-base-anchors.xlsx

#%% 1. List of volumes (and some articles)

# Review of anchor elements in home page, eid-home-anchors.xlsx
# https://www.cdc.gov/eid/current  # current issue
#    all articles in current issue (April 2020)
# https://www.cdc.gov/eid/past-issues/volume-26 # past volumes
#    all previous issues in vol 26 (Jan-Mar 2020), previous volumes (1995-2019)

series_a = home_soup.find_all('a', string=re.compile('Past Issues'))
# [<a aria-expanded="true" href="/eid/past-issues/volume-27">Past Issues</a>]

series_dframe = pd.DataFrame(
   [process_aTag(aTag, home_dframe.url[0]) for aTag in series_a], index=[0])
# series_dframe.loc[:, ['path', 'string']]
#                          path       string
# 0  /eid/past-issues/volume-27  Past Issues

series_html = get_html_from_url(series_dframe.url[0]) # len(series_html) # 322719
series_soup = BeautifulSoup(series_html, 'lxml')

# review all anchor-hrefs from series URL
# len(series_soup.find_all('a', href=True)) # 173
# pd.DataFrame([process_aTag(aTag, series_dframe.url[0])
#     for aTag in series_soup.find_all('a', href=True)]).\
#     to_excel('eid-series-anchors.xlsx', engine='openpyxl')

#%% 2. List and contents of volumes

# Review of anchor elements in series page, eid-series-anchors.xlsx
# eid/past-issues/volume{1-26}
# href contains 'volume-\d{1,2}' and string contains 'Volume'
# volume-27 doesn't contain 'Volume 27–2021'
# obtain volumes 2-present from volume-1 and Volume 1 from volume-2
eid_vol_re0 = re.compile(r'volume-\d{1,2}')
eid_vol_re1 = re.compile(r'Volume.+?\d{4}') # 1995-2021
eid_vol_re2 = re.compile(r'Volume.+?1995')  # 1995
volumes_a = BeautifulSoup(get_html_from_url(
      'https://wwwnc.cdc.gov/eid/past-issues/volume-1'), 'lxml').\
      find_all('a', href=eid_vol_re0, string=eid_vol_re1) + \
   BeautifulSoup(get_html_from_url(
      'https://wwwnc.cdc.gov/eid/past-issues/volume-2'), 'lxml').\
      find_all('a', href=eid_vol_re0, string=eid_vol_re2)
# len(volumes_a) # 27

volumes_dframe = pd.DataFrame(
   [process_aTag(aTag, series_dframe.url[0]) for aTag in volumes_a])
# volumes_dframe.loc[:, ['path', 'string']]
#                            path          string
# 0   /eid/past-issues/volume-27  Volume 27–2021
# 1   /eid/past-issues/volume-26  Volume 26–2020
# 2   /eid/past-issues/volume-25  Volume 25–2019
# ...
# 24   /eid/past-issues/volume-3   Volume 3–1997
# 25   /eid/past-issues/volume-2   Volume 2–1996
```

```
# 26   /eid/past-issues/volume-1   Volume 1–1995

volumes_html = [get_html_from_url_(url) for url in volumes_dframe.url]
# [len(x) for x in volumes_html]
# [322719, 334178, 334177, 334168, 335223, 334170, 334246, 333992, 333992, .../
volumes_soup = [BeautifulSoup(html, 'lxml') for html in volumes_html]

# review all anchor-refs from volumes URLs
# pd.DataFrame([process_aTag(aTag, url)
#     for soup, url in zip(volumes_soup, volumes_dframe.url)
#     for aTag in soup.find_all('a', href=True)]).\
#     to_excel('eid-volumes-anchors.xlsx', engine='openpyxl')
# [5623 rows x 7 columns]

#%% 3. List and contents of issues (tables of contents)

# Review of anchor elements in volumes page, eid-volumes-anchors.xlsx
# All 255 issue paths have the form /eid/articles/issue/#0/#0/table-of-contents,
#    or href containing regex '\d{1,2}/\d{1,2}/table-of-contents'
# They also all have string 'Table of Contents'

eid_iss_re = re.compile(r'Table of Contents')
issues_a = [soup.find_all('a', string=eid_iss_re) for soup in volumes_soup]
issues_a_n = [len(x) for x in issues_a] # sum(issues_a_n) # 265
# [ 1, 12, 12, 12, 13, 12, 12, 12, 12, 12, 12, 12, 12,
#  12, 12, 12, 12, 12, 12, 12,  7,  6,  6,  4,  4,  4,  4]

issues_dframe = pd.DataFrame([process_aTag(aTag, url)
   for a_list, url in zip(issues_a, volumes_dframe.url)
   for aTag in a_list])
# (265, 7)
# issues_dframe.loc[:, ['path', 'string']]
#                                                path             string
# 0     /eid/articles/issue/27/1/table-of-contents  Table of Contents
# 1    /eid/articles/issue/26/12/table-of-contents  Table of Contents
# 2    /eid/articles/issue/26/11/table-of-contents  Table of Contents
# 3    /eid/articles/issue/26/10/table-of-contents  Table of Contents
# 4     /eid/articles/issue/26/9/table-of-contents  Table of Contents
# ..                                           ...              ...
# 260    /eid/articles/issue/2/1/table-of-contents  Table of Contents
# 261    /eid/articles/issue/1/4/table-of-contents  Table of Contents
# 262    /eid/articles/issue/1/3/table-of-contents  Table of Contents
# 263    /eid/articles/issue/1/2/table-of-contents  Table of Contents
# 264    /eid/articles/issue/1/1/table-of-contents  Table of Contents

issues_repeated = {
   label: content.loc[content.duplicated(keep = False)].index.to_list()
      for label, content
      in issues_dframe.loc[:, ['href', 'url', 'path', 'filename']].items() }
# { k: len(v) for k, v in issues_repeated.items() }
# {'href': 0, 'url': 0, 'path': 0, 'filename': 265}

# pickle.dump(issues_dframe, open("issues_dframe.pkl", "wb"))

# issues_dframe.to_excel('eid-issues_dframe.xlsx', engine='openpyxl')
start_time = time.time()
```

```
169 issues_html = [get_html_from_url(url, print_url=True, timeout=1) for url in
169 tqdm(issues_dframe.url)]
170 print(f"\nTime elapsed: {int((time.time() - start_time) // 60)} min {round((time.time() -
170 start_time) % 60, 1)} sec")
171 # sum([len(x)==0 for x in issues_html]) # 224
172
173 # check for failed requests -- those with length 0; repeat until there are none
174 start_time = time.time()
175 for iss in range(265):
176    if issues_html[iss] == '':
177       issues_html[iss] = get_html_from_url(issues_dframe.url[iss], print_url=True,
177       timeout=5)
178 print(f"\nTime elapsed: {int((time.time() - start_time) // 60)} min {round((time.time() -
178 start_time) % 60, 1)} sec")
179 # sum([len(x)==0 for x in issues_html]) # 0
180
181 # [len(x) for x in issues_html]
182 # [542509, 555040, 508360, 536572, 583448, 562393, 531330, 503779, 482331, ...]
183 issues_soup = [BeautifulSoup(html, 'lxml') for html in tqdm(issues_html, total=265)]
184
185 # review all anchor-refs from issue URLs
186 # pd.DataFrame([process_aTag(aTag, url)
187 #     for soup, url in zip(issues_soup, issues_dframe.url)
188 #     for aTag in soup.find_all('a', href=True)]).\
189 #     to_excel('eid-issues-anchors.xlsx', engine='openpyxl')
190 # [63159 rows x 7 columns]
191
192 #%% 4. List of articles
193
194 # Review of anchor elements in volumes page, eid-issues-anchors.xlsx
195 # All 11222 article paths have form /eid/article/#0/#0/
196 #    For nearly all articles (11211), the path ends in '_article'
197 #    The exception is 11 photo quizzes, which we omit
198 # Most paths (11211) follow pattern '/\d{1,2}/\d{1,2}/\d{2}-\d{4}_article'
199
200 eid_art_re = re.compile(r'_article$')
201 articles_a = [soup.find_all('a', href=eid_art_re) for soup in issues_soup]
202 articles_a_n = [len(x) for x in articles_a] # sum(articles_a_n) # 11211
203
204 articles_dframe = pd.DataFrame([process_aTag(aTag, url)
205    for a_list, url in zip(articles_a, issues_dframe.url)
206    for aTag in a_list])
207 # (11211, 7)
208 # articles_dframe.loc[:, ['path', 'string']]
209 #                                      path                                    string
210 # 0      /eid/article/27/1/19-1364_article  Impact of Human Papillomavirus V...
211 # 1      /eid/article/27/1/20-2656_article  Nosocomial Coronavirus Disease O...
212 # 2      /eid/article/27/1/20-2896_article  Aspergillosis Complicating Sever...
213 # 3      /eid/article/27/1/19-0782_article  Invasive Fusariosis in Nonneutro...
214 # 4      /eid/article/27/1/19-1220_article  Differential Yellow Fever Suscep...
215 # ...                                  ...                                 ...
216 # 11206   /eid/article/1/1/95-0108_article  Electronic Communication and the...
217 # 11207   /eid/article/1/1/ac-0101_article                    Volume 1, Issue 1
218 # 11208   /eid/article/1/1/95-0109_article   Communicable Diseases Intelligence
219 # 11209   /eid/article/1/1/95-0110_article  DxMONITOR: Compiling Veterinary ...
220 # 11210   /eid/article/1/1/95-0111_article  WHO Scientific Working Group on ...
```

```
221
222 #%% 5. Complete list of EID files
223 eid_dframe = pd.concat([
224    home_dframe.assign(level='home'),
225 #  series_dframe.assign(level='series'), # omit as redundant with volumes
226    volumes_dframe.assign(level='volume'),
227    issues_dframe.assign(level='issue'),
228    articles_dframe.assign(level='article')],
229    axis = 0, ignore_index = True) # eid_dframe.index = list(range(10922))
230 # (11504, 8)
231
232 # pickle
233 pickle.dump(eid_dframe, open("eid_dframe.pkl", "wb"))
234 # eid_dframe_ = pickle.load(open("eid_dframe.pkl", "rb"))
235 # eid_dframe.equals(eid_dframe_)
236
237 # Excel; coulad also use engine=
238 eid_dframe.to_excel('eid_dframe.xlsx', engine='openpyxl')
239 # Excelternatives
240 # eid_dframe.to_excel('eid_dframe.xlsx', engine='xlsxwriter') # pd default
241 # eid_dframe.to_excel('eid_dframe.xls', engine='xlwt')
```