In [1]:
```
import numpy as np
import pandas as pd
import requests
import json
import dotenv
import os
from bs4 import BeautifulSoup
```

```
/home/digifort/anaconda3/lib/python3.11/site-packages/pandas/core/arrays/ma
sked.py:60: UserWarning: Pandas requires version '1.3.6' or newer of 'bottl
eneck' (version '1.3.5' currently installed).
  from pandas.core import (
```

In [2]:
```
botname = 'ds6600'
version = '0.0'
useragent = f'{botname}/{version} python-requests/{requests.__version__}'
headers = {'User-Agent':useragent}
headers
```

Out[2]:
```
{'User-Agent': 'ds6600/0.0 python-requests/2.32.4'}
```

QS 1

Part a

In [13]:
```
path = '/home/digifort/Documents/Data_Management_F25/lab_2/cands22.txt'

cand = pd.read_csv(path, sep=',', quotechar='|',
                   engine='python', header=None,
                   on_bad_lines='error')
```

In [14]:
```
cand.shape
```

Out[14]:
```
(8928, 12)
```

In [15]:
```
obj_cols = cand.select_dtypes(include='object').columns
cand[obj_cols] = cand[obj_cols].apply(lambda s: s.str.strip('|'))

cand.head(3)
```

Out[15]:

	0	1	2	3	4	5	6	7	8	9	10	11
0	2022	H0AK00105	N00039029	Thomas Lamb (3)	3	AK01					3N	
1	2022	H0AL01055	N00044245	Jerry Carl (R)	R	AL01	AL01	Y	Y	I	RW	
2	2022	H0AL01063	N00044288	Wes Lambert (R)	R	AL01					RN	

In [16]:
```
cand.columns = ['Cycle','FECCandID','CID','FirstLastP','Party','DistIDRunFo
                'DistIDCurr','CurrCand','CycleCand','CRPICO','recip_code','r
cand.head(2)
```

Out[16]:

	Cycle	FECCandID	CID	FirstLastP	Party	DistIDRunFor	DistIDCurr	CurrCand	CycleCand
0	2022	H0AK00105	N00039029	Thomas Lamb (3)	3	AK01			
1	2022	H0AL01055	N00044245	Jerry Carl (R)	R	AL01	AL01	Y	Y

Part b

In [19]:

```
cand_skip = pd.read_csv('cands22.txt', on_bad_lines='skip')
cand_skip
```

Out[19]:

	|2022|	|H0AK00105|	|N00039029|	|Thomas Lamb (3)|	|3|	|AK01|	| |	| |	| |.1	| |.2	|3N|	| |.3
0	|2022|	|H0AL01055|	|N00044245|	|Jerry Carl (R)|	|R|	|AL01|	|AL01|	|Y|	|Y|	|I|	|RW|	| |
1	|2022|	|H0AL01063|	|N00044288|	|Wes Lambert (R)|	|R|	|AL01|	| |	| |	| |	| |	|RN|	| |
2	|2022|	|H0AL01097|	|N00044750|	|James Averhart (D)|	|D|	|AL01|	| |	| |	| |	| |	|DN|	| |
3	|2022|	|H0AL02087|	|N00030768|	|Martha Roby (R)|	|R|	|AL02|	| |	| |	| |	| |	|RN|	| |
4	|2022|	|H0AL02137|	|N00045120|	|Will Dismukes (R)|	|R|	|AL02|	| |	| |	| |	| |	|RN|	| |
...	...	...	...	...	...	...	...	...	...	...	...	...
8909	|2022|	|S8WI00224|	|N00041929|	|Leah Vukmir (R)|	|R|	|WIS1|	| |	| |	| |	| |	|RN|	| |
8910	|2022|	|S8WV00119|	|N00041293|	|Paula Jean Swearengin (D)|	|D|	|WVS2|	| |	| |	| |	| |	|DN|	| |
8911	|2022|	|S8WV00127|	|N00035531|	|Evan Jenkins (R)|	|R|	|WVS1|	| |	| |	| |	| |	|RN|	| |
8912	|2022|	|S8WV00135|	|N00041474|	|Bo Copley (R)|	|R|	|WVS1|	| |	| |	| |	| |	|RN|	| |
8913	|2022|	|S8WV00143|	|N00012642|	|Patrick Morrisey (R)|	|R|	|WVS1|	| |	| |	| |	| |	|RN|	| |

8914 rows × 12 columns

'on_bad_lines='skip'' seem to drops any rows that the parser can't tokenize.

This seem to fix the error by throwing away problematic lines, which also risks data loss and row/column misalignment.

For this dataset, problems arise because text fields are wrapped in '|' and may contain commas. When pandas is not told about quotechar='|', it mis-splits fields.

Hence, skipping lines is not appropriate here. The correct solution would be to parse properly with quotechar='|' (as in part a), not to discard data.

Qs 2

Part a

In [ ]:

```
url = "https://stats.nba.com/js/data/sportvu/2015/shootingTeamData.json"
resp = requests.get(url, headers=headers, timeout=30)
resp.raise_for_status()
nba_json = resp.json()
type(nba_json), list(nba_json.keys())
```

Out[ ]:

```
(dict, ['parameters', 'resource', 'resultSets'])
```

In [25]:

```
#nba_json
nba_json['resultSets'][0]
```

Out[25]:

```
{'name': 'TeamTrackingShootingEfficiencyStats',
 'headers': ['TEAM_ID',
  'TEAM_CITY',
  'TEAM_NAME',
  'TEAM_ABBREVIATION',
  'TEAM_CODE',
  'GP',
  'MIN',
  'PTS',
  'PTS_DRIVE',
  'FGP_DRIVE',
  'PTS_CLOSE',
  'FGP_CLOSE',
  'PTS_CATCH_SHOOT',
  'FGP_CATCH_SHOOT',
  'PTS_PULL_UP',
  'FGP_PULL_UP',
  'FGA_DRIVE',
  'FGA_CLOSE',
  'FGA_CATCH_SHOOT',
  'FGA_PULL_UP',
  'EFG_PCT',
  'CFGM',
  'CFGA',
  'CFGP',
  'UFGM',
  'UFGA',
  'UFGP',
  'CFG3M',
  'CFG3A',
  'CFG3P',
  'UFG3M',
  'UFG3A',
  'UFG3P'],
 'rowSet': [['1610612744',
   'Golden State',
   'Warriors',
   'GSW',
   '',
   82,
   48.7,
   114.9,
   14.9,
   0.498,
   16.7,
   0.645,
   33.7,
   0.428,
   21.5,
   0.418,
   11.0,
   11.1,
   28.3,
   21.5,
   0.563,
   21.4,
   44.8,
   0.478,
   21.2,
   42.5,
   0.497,
   2.3,
   6.3,
   0.363,
  10.8,
  25.3,
  0.429],
 ['1610612759',
  'San Antonio',
  'Spurs',
  'SAS',
  '',
  82,
  48.3,
  103.5,
  14.8,
  0.481,
  17.8,
  0.611,
  27.1,
  0.419,
  18.0,
  0.398,
  11.8,
  12.2,
  25.3,
  21.4,
  0.526,
  21.8,
  43.1,
  0.506,
  18.3,
  39.8,
  0.46,
  0.9,
  2.6,
  0.341,
  6.1,
  15.9,
  0.381],
 ['1610612739',
  'Cleveland',
  'Cavaliers',
  'CLE',
  '',
  82,
  48.7,
  104.3,
  16.9,
  0.481,
  14.3,
  0.622,
  28.2,
  0.394,
  18.6,
  0.36,
  13.4,
  9.7,
  25.2,
  22.7,
  0.524,
  20.5,
  43.3,
  0.473,
  18.2,
  40.7,
  0.447,
  1.7,
  5.7,
  0.299,
  9.0,
  23.9,
  0.378],
 ['1610612746',
  'Los Angeles',
  'Clippers',
  'LAC',
  '',
  82,
  48.6,
  104.5,
  15.0,
  0.497,
  12.7,
  0.712,
  26.5,
  0.404,
  23.9,
  0.385,
  10.9,
  7.8,
  24.3,
  27.7,
  0.524,
  19.4,
  40.5,
  0.48,
  18.9,
  42.0,
  0.45,
  2.0,
  6.0,
  0.334,
  7.7,
  20.8,
  0.373],
 ['1610612760',
  'Oklahoma City',
  'Thunder',
  'OKC',
  '',
  82,
  48.6,
  110.2,
  16.1,
  0.48,
  15.3,
  0.677,
  24.7,
  0.402,
  20.6,
  0.39,
  11.9,
  9.5,
  23.7,
  23.3,
  0.524,
  23.7,
  47.7,
  0.497,
  17.5,
  38.7,
  0.451,
  1.6,
  5.1,
  0.321,
  6.6,
  18.6,
  0.356],
 ['1610612737',
  'Atlanta',
  'Hawks',
  'ATL',
  '',
  82,
  48.6,
  102.8,
  19.0,
  0.463,
  14.4,
  0.622,
  31.9,
  0.399,
  12.9,
  0.355,
  15.3,
  10.1,
  29.7,
  15.8,
  0.516,
  19.2,
  39.9,
  0.483,
  19.4,
  44.6,
  0.435,
  1.0,
  3.1,
  0.311,
  9.0,
  25.3,
  0.355],
 ['1610612745',
  'Houston',
  'Rockets',
  'HOU',
  '',
  82,
  48.6,
  106.5,
  17.2,
  0.433,
  15.2,
  0.629,
  24.5,
  0.37,
  16.5,
  0.349,
  13.8,
  10.2,
  22.8,
  19.6,
  0.516,
  22.2,
  47.1,
  0.472,
  15.5,
  36.4,
  0.426,
  2.3,
  7.4,
  0.318,
  8.4,
  23.5,
  0.355],
 ['1610612757',
  'Portland',
  'Trail Blazers',
  'POR',
  '',
  82,
  48.5,
  105.1,
  17.5,
  0.441,
  10.4,
  0.646,
  26.9,
  0.395,
  23.1,
  0.398,
  14.5,
  6.8,
  24.5,
  25.7,
  0.511,
  20.6,
  46.0,
  0.447,
  18.0,
  39.8,
  0.453,
  1.7,
  5.9,
  0.295,
  8.8,
  22.6,
  0.389],
 ['1610612758',
  'Sacramento',
  'Kings',
  'SAC',
  '',
  81,
  48.4,
  106.7,
  18.7,
  0.452,
  16.3,
  0.599,
  25.1,
  0.383,
  13.8,
  0.368,
  15.3,
  11.4,
  24.4,
  16.7,
  0.511,
  22.1,
  46.8,
  0.473,
  18.1,
  39.7,
  0.454,
  0.9,
  3.1,
  0.276,
  7.2,
  19.4,
  0.372],
 ['1610612764',
  'Washington',
  'Wizards',
  'WAS',
  '',
  82,
  48.5,
  104.1,
  15.4,
  0.48,
  15.1,
  0.629,
  28.7,
  0.394,
  17.8,
  0.371,
  11.3,
  10.1,
  27.4,
  22.1,
  0.511,
  20.0,
  41.5,
  0.483,
  19.5,
  44.3,
  0.439,
  0.7,
  2.7,
  0.254,
  8.0,
  21.5,
  0.371],
 ['1610612748',
  'Miami',
  'Heat',
  'MIA',
  '',
  82,
  48.6,
  100.0,
  17.9,
  0.488,
  15.7,
  0.623,
  22.5,
  0.386,
  15.2,
  0.371,
  13.9,
  10.4,
  22.3,
  19.5,
  0.508,
  22.7,
  46.4,
  0.49,
  15.7,
  35.2,
  0.445,
  0.8,
  2.9,
  0.282,
  5.3,
  15.1,
  0.347],
 ['1610612761',
  'Toronto',
  'Raptors',
  'TOR',
  '',
  81,
  48.5,
  102.7,
  23.0,
  0.462,
  11.6,
  0.636,
  24.6,
  0.394,
  17.1,
  0.369,
  18.2,
  7.6,
  22.3,
  21.1,
  0.504,
  22.5,
  48.8,
  0.461,
  14.1,
  32.4,
  0.436,
  1.8,
  5.6,
  0.327,
  6.8,
  17.7,
  0.384],
 ['1610612742',
  'Dallas',
  'Mavericks',
  'DAL',
  '',
  82,
  49.0,
  102.3,
  18.2,
  0.473,
  13.7,
  0.645,
  29.7,
  0.384,
  17.4,
  0.365,
  14.6,
  8.7,
  28.8,
  21.1,
  0.502,
  19.8,
  42.7,
  0.464,
  17.5,
  41.4,
  0.423,
  1.4,
  5.3,
  0.273,
  8.4,
  23.3,
  0.36],
 ['1610612766',
  'Charlotte',
  'Hornets',
  'CHA',
  '',
  82,
  48.6,
  103.4,
  16.8,
  0.459,
  11.4,
  0.565,
  30.3,
  0.389,
  17.9,
  0.353,
  12.8,
  8.0,
  28.1,
  22.3,
  0.502,
  20.0,
  44.6,
  0.449,
  17.0,
  39.8,
  0.427,
  1.8,
  6.0,
  0.297,
  8.9,
  23.4,
  0.379],
 ['1610612762',
  'Utah',
  'Jazz',
  'UTA',
  '',
  82,
  49.0,
  97.7,
  18.1,
  0.445,
  11.9,
  0.609,
  24.2,
  0.383,
  17.6,
  0.371,
  14.8,
  8.2,
  23.1,
  21.1,
  0.501,
  20.2,
  43.2,
  0.468,
  15.9,
  37.2,
  0.426,
  1.4,
  4.3,
  0.318,
  7.1,
  19.5,
  0.363],
 ['1610612753',
  'Orlando',
  'Magic',
  'ORL',
  '',
  81,
  48.7,
  102.0,
  18.0,
  0.456,
  13.3,
  0.599,
  29.5,
  0.398,
  14.4,
  0.363,
  15.0,
  9.6,
  29.0,
  18.3,
  0.5,
  21.0,
  44.1,
  0.475,
  18.5,
  42.6,
  0.435,
  0.7,
  2.7,
  0.249,
  7.1,
  19.5,
  0.363],
 ['1610612749',
  'Milwaukee',
  'Bucks',
  'MIL',
  '',
  82,
  48.7,
  99.0,
  17.4,
  0.463,
  18.0,
  0.621,
  18.3,
  0.37,
  13.3,
  0.362,
  14.3,
  12.3,
  18.7,
  17.2,
  0.499,
  25.2,
  52.8,
  0.477,
  13.2,
  29.4,
  0.448,
  1.1,
  4.0,
  0.27,
  4.3,
  11.6,
  0.37],
 ['1610612740',
  'New Orleans',
  'Pelicans',
  'NOP',
  '',
  82,
  48.5,
  102.7,
  19.9,
  0.458,
  11.8,
  0.641,
  23.3,
  0.387,
  21.0,
  0.382,
  16.3,
  7.5,
  22.4,
  24.3,
  0.498,
  20.6,
  44.7,
  0.46,
  17.9,
  41.1,
  0.434,
  0.6,
  2.6,
  0.247,
  7.9,
  21.2,
  0.374],
 ['1610612750',
  'Minnesota',
  'Timberwolves',
  'MIN',
  '',
  82,
  48.6,
  102.4,
  15.1,
  0.464,
  15.6,
  0.579,
  21.5,
  0.4,
  15.5,
  0.346,
  10.9,
  10.7,
  21.6,
  20.5,
  0.498,
  21.6,
  45.9,
  0.471,
  16.1,
  35.4,
  0.455,
  0.7,
  2.6,
  0.272,
  4.8,
  13.8,
  0.35],
 ['1610612754',
  'Indiana',
  'Pacers',
  'IND',
  '',
  82,
  48.8,
  102.2,
  13.7,
  0.453,
  15.0,
  0.603,
  25.3,
  0.375,
  19.0,
  0.361,
  10.5,
  10.6,
  25.7,
  23.7,
  0.497,
  21.9,
  47.1,
  0.465,
  16.4,
  38.1,
  0.431,
  1.7,
  5.7,
  0.299,
  6.4,
  17.4,
  0.368],
 ['1610612751',
  'Brooklyn',
  'Nets',
  'BKN',
  '',
  82,
  48.4,
  98.6,
  14.4,
  0.457,
  13.8,
  0.595,
  22.4,
  0.387,
  16.0,
  0.362,
  12.3,
  9.8,
  22.5,
  20.2,
  0.492,
  22.4,
  48.3,
  0.464,
  15.8,
  36.1,
  0.438,
  1.0,
  3.3,
  0.303,
  5.5,
  15.1,
  0.363],
 ['1610612765',
  'Detroit',
  'Pistons',
  'DET',
  '',
  82,
  48.7,
  102.0,
  17.5,
  0.464,
  14.1,
  0.597,
  23.7,
  0.378,
  19.2,
  0.367,
  14.2,
  10.0,
  22.3,
  23.4,
  0.491,
  22.3,
  49.3,
  0.452,
  15.7,
  37.2,
  0.422,
  0.9,
  4.0,
  0.227,
  8.1,
  22.2,
  0.366],
 ['1610612743',
  'Denver',
  'Nuggets',
  'DEN',
  '',
  82,
  48.6,
  101.9,
  15.9,
  0.406,
  16.6,
  0.598,
  23.5,
  0.376,
  15.2,
  0.349,
  13.4,
  11.7,
  23.3,
  19.2,
  0.489,
  21.3,
  47.6,
  0.448,
  16.4,
  37.8,
  0.434,
  1.1,
  4.3,
  0.264,
  6.9,
  19.5,
  0.354],
 ['1610612738',
  'Boston',
  'Celtics',
  'BOS',
  '',
  81,
  48.5,
  105.6,
  18.9,
  0.453,
  13.5,
  0.589,
  26.4,
  0.359,
  16.4,
  0.35,
  15.0,
  9.5,
  27.5,
  20.7,
  0.488,
  22.3,
  49.5,
  0.451,
  16.9,
  39.9,
  0.424,
  1.6,
  5.7,
  0.274,
  7.1,
  20.3,
  0.35],
 ['1610612741',
  'Chicago',
  'Bulls',
  'CHI',
  '',
  82,
  48.9,
  101.6,
  18.1,
  0.458,
  13.6,
  0.587,
  28.3,
  0.416,
  16.7,
  0.358,
  14.9,
  9.8,
  26.2,
  21.5,
  0.487,
  21.6,
  48.9,
  0.442,
  17.0,
  38.5,
  0.441,
  1.3,
  3.9,
  0.332,
  6.6,
  17.5,
  0.38],
 ['1610612755',
  'Philadelphia',
  '76ers',
  'PHI',
  '',
  82,
  48.6,
  97.4,
  19.7,
  0.445,
  13.4,
  0.621,
  26.5,
  0.361,
  12.8,
  0.322,
  16.6,
  9.0,
  26.2,
  17.3,
  0.487,
  21.0,
  46.6,
  0.449,
  15.3,
  37.4,
  0.409,
  1.6,
  5.7,
  0.281,
  7.7,
  21.8,
  0.354],
 ['1610612756',
  'Phoenix',
  'Suns',
  'PHX',
  '',
  82,
  48.4,
  100.9,
  15.6,
  0.44,
  14.0,
  0.565,
  25.6,
  0.376,
  17.3,
  0.354,
  13.1,
  10.2,
  24.9,
  21.5,
  0.487,
  20.6,
  46.1,
  0.447,
  16.6,
  39.5,
  0.421,
  1.4,
  5.0,
  0.288,
  7.6,
  20.8,
  0.363],
 ['1610612752',
  'New York',
  'Knicks',
  'NYK',
  '',
  82,
  48.5,
  98.4,
  10.4,
  0.447,
  14.4,
  0.568,
  26.6,
  0.387,
  18.9,
  0.38,
  7.9,
  10.1,
  26.7,
  23.1,
  0.483,
  20.9,
  47.6,
  0.439,
  15.9,
  36.4,
  0.438,
  1.5,
  4.9,
  0.305,
  5.9,
  16.6,
  0.358],
 ['1610612763',
  'Memphis',
                 'Grizzlies',
                 'MEM',
                 '',
                 82,
                 48.6,
                 99.1,
                 16.4,
                 0.44,
                 14.4,
                 0.546,
                 22.5,
                 0.379,
                 13.2,
                 0.364,
                 13.3,
                 10.6,
                 23.2,
                 16.6,
                 0.477,
                 20.7,
                 45.2,
                 0.459,
                 16.1,
                 38.5,
                 0.418,
                 0.7,
                 2.5,
                 0.278,
                 5.4,
                 16.0,
                 0.34],
                ['1610612747',
                 'Los Angeles',
                 'Lakers',
                 'LAL',
                 '',
                 82,
                 48.3,
                 97.3,
                 15.6,
                 0.441,
                 11.3,
                 0.584,
                 19.8,
                 0.355,
                 22.3,
                 0.347,
                 13.1,
                 7.9,
                 21.2,
                 28.1,
                 0.46,
                 21.1,
                 50.3,
                 0.42,
                 14.0,
                 34.5,
                 0.406,
                 2.2,
                 7.9,
                 0.278,
                 5.6,
                 16.7,
                 0.335]]}
```

Part b

The team rows live at:

- nba_json['resultSets'][0]['rowSet'] → this is a list of 30 rows (one per team).

The column names live at:

- nba_json['resultSets'][0]['headers'] → this is a list of column names aligned to each element in a row.

Part c

In [26]:

```
base = nba_json['resultSets'][0]
df_raw = pd.json_normalize(base, record_path=['rowSet'])
df_raw.shape
```

Out[26]:

```
(30, 33)
```

Part d

In [27]:

```
headers_list = nba_json['resultSets'][0]['headers']
len(headers_list), headers_list[:5]

df_nba = df_raw.copy()
df_nba.columns = headers_list
df_nba.head(3)
```

Out[27]:

	TEAM_ID	TEAM_CITY	TEAM_NAME	TEAM_ABBREVIATION	TEAM_CODE	GP	MIN	PTS
0	1610612744	Golden State	Warriors	GSW		82	48.7	114.9
1	1610612759	San Antonio	Spurs	SAS		82	48.3	103.5
2	1610612739	Cleveland	Cavaliers	CLE		82	48.7	104.3

3 rows × 33 columns

Part e

In [30]:

```
# produces a dict with keys: "columns", "index", "data"
out_path = '/home/digifort/Documents/Data_Management_F25/lab_2/nba_shooting_
df_nba.to_json(out_path, orient='split')
out_path
```

Out[30]:

```
'/home/digifort/Documents/Data_Management_F25/lab_2/nba_shooting_2015_spli
t.json'
```

In [33]:

```
import json

with open("nba_shooting_2015_split.json", "r") as f:
    data = json.load(f)

print(data.keys())
print('\n\n')
print(data["columns"])
print('\n\n')
print(data["data"][0])
```

```
dict_keys(['columns', 'index', 'data'])



['TEAM_ID', 'TEAM_CITY', 'TEAM_NAME', 'TEAM_ABBREVIATION', 'TEAM_CODE', 'G
P', 'MIN', 'PTS', 'PTS_DRIVE', 'FGP_DRIVE', 'PTS_CLOSE', 'FGP_CLOSE', 'PTS_
CATCH_SHOOT', 'FGP_CATCH_SHOOT', 'PTS_PULL_UP', 'FGP_PULL_UP', 'FGA_DRIVE',
'FGA_CLOSE', 'FGA_CATCH_SHOOT', 'FGA_PULL_UP', 'EFG_PCT', 'CFGM', 'CFGA',
'CFGP', 'UFGM', 'UFGA', 'UFGP', 'CFG3M', 'CFG3A', 'CFG3P', 'UFG3M', 'UFG3
A', 'UFG3P']



['1610612744', 'Golden State', 'Warriors', 'GSW', '', 82, 48.7, 114.9, 14.
9, 0.498, 16.7, 0.645, 33.7, 0.428, 21.5, 0.418, 11.0, 11.1, 28.3, 21.5, 0.
563, 21.4, 44.8, 0.478, 21.2, 42.5, 0.497, 2.3, 6.3, 0.363, 10.8, 25.3, 0.4
29]
```

Qs 3

In [34]:

```
reddit_url = "https://www.reddit.com/r/popular/top.json?limit=25"
r = requests.get(reddit_url, headers=headers, timeout=30)
r.raise_for_status()
rj = r.json()


rows = []
for child in rj['data']['children']:
    d = child['data']
    rows.append([
        d.get('subreddit'),
        d.get('title'),
        d.get('ups'),
        d.get('created_utc')
    ])

# dataFrame from list-of-lists
df_reddit = pd.DataFrame(rows, columns=['subreddit','title','ups','created_u
df_reddit.head(5), df_reddit.shape
```

Out[34]:

```
(            subreddit                                                title
\
 0   mildlyinfuriating    everybody apologizing for cheating with chatgpt
 1  NatureIsFuckingLit   🔥 NOMINEES FOR COMEDY WILDLIFE PHOTO OF THE YEAR
 2         interesting      Baby gator just started its first death roll.
 3                 law  Trump: I don't think we're gonna necessarily a...
 4   interestingasfuck  Dress wore by queen Victoria. Her height was 4'11

       ups   created_utc
 0  120176  1.761260e+09
 1   85093  1.761256e+09
 2   78324  1.761244e+09
 3   74355  1.761252e+09
 4   52867  1.761247e+09  ,
 (25, 4))
```

Qs 4

root: https://maps.spirithalloween.com

endpoint: /api/getAsyncLocations

params: template=domain, level=domain,

lat=40.380028, lng=-97.910156, radius=1800, limit=2000

Part a

In [37]:

```
root = "https://maps.spirithalloween.com"
endpoint = "/api/getAsyncLocations"

params = {
    "template": "domain",
    "level": "domain",
    # "options" overridden
    "lat": 40.380028,
    "lng": -97.910156,
    "radius": 1800,    # miles
    "limit": 2000      # more than enough
}

resp = requests.get(root + endpoint, params=params, headers=headers, timeou
resp.raise_for_status()
spirit = resp.json()
list(spirit.keys())
```

Out[37]:

```
['lat', 'lng', 'originLat', 'originLng', 'markers', 'maplist', 'options']
```

Part b

In [38]:

```
stores = spirit.get('markers', [])
len(stores), type(stores[:1])
```

Out[38]:

```
(1535, list)
```

Part c

In [39]:

```
info_list = []
for s in stores:
    info_list.append(s.get('info', ''))

len(info_list), info_list[0][:120]
```

Out[39]:

```
(1535,
 '<div class="tlsmap_popup">{    "fid":"61508",    "lid":"4364",    "ur
l":"https://stores.spirithalloween.com/ne/grandisla')
```

Part d

In [40]:

```
def extract_address2_from_info(info_html: str) -> str | None:
    """
    Given a single element from info_list, parse out the address_2 field.
    Returns a string (e.g., 'Former Big Lots') or None if not found/parseab
    """
    if not info_html:
        return None

    soup = BeautifulSoup(info_html, 'html.parser')
    div = soup.find('div', class_='tlsmap_popup')
    if div is None or div.string is None:
        return None

    raw = div.string.strip()
    if not raw.endswith(','):
        # capability to also handle cases which might not have the trailing
        jsonish = raw
    else:
        jsonish = raw[:-1]  # drop the trailing comma

    try:
        data = json.loads(jsonish)
        return data.get('address_2')
    except Exception:
        return None

# smoke test on first non-empty info
sample = next((x for x in info_list if x), '')
extract_address2_from_info(sample)
```

Out[40]:

```
'Former Goodwill'
```

Part e

In [ ]:

```
address2_all = [extract_address2_from_info(x) for x in info_list]
address2_all[:10], len(address2_all)
```

Out[ ]:

```
(['Former Goodwill',
  'Former Hobby Lobby',
  'Near AMC 13',
  'Former JoAnn Fabrics',
  'Former CVS',
  'Next to Old Navy',
  'Next to At Home',
  'Former Sears Hometown',
  'Former Old Navy',
  'Next to Your Wireless'],
 1535)
```

Part f

In [42]:

```
former = [x for x in address2_all if isinstance(x, str) and 'Former' in x]
pd.Series(former).value_counts()
```

Out[42]:

```
Former Party City              166
Former Big Lots                113
Former Rite Aid                 51
Former Joann Fabrics            46
Former Forever 21               42
                              ...
Former Half Price Books          1
Former Burkes                    1
Former Gander Mountain           1
Former Bowlskis Bowling Alley    1
Former Save A Lot                1
Name: count, Length: 342, dtype: int64
```

```
In [ ]:
```