JSON examples and exercise

- get familiar with packages for dealing with JSON
- study examples with JSON strings and files
- work on exercise to be completed and submitted

- reference: http://pandas.pydata.org/pandas-docs/stable/io.html#io-json-reader
- data source: http://jsonstudio.com/resources/

In [4]:
```
import pandas as pd
```

imports for Python, Pandas

In [5]:
```
import json
from pandas.io.json import json_normalize
```

JSON example, with string

- demonstrates creation of normalized dataframes (tables) from nested json string
- source: http://pandas.pydata.org/pandas-docs/stable/io.html#normalization

In [6]:
```
# define json string
data = [{'state': 'Florida',
         'shortname': 'FL',
         'info': {'governor': 'Rick Scott'},
         'counties': [{'name': 'Dade', 'population': 12345},
                      {'name': 'Broward', 'population': 40000},
                      {'name': 'Palm Beach', 'population': 60000}]},
        {'state': 'Ohio',
         'shortname': 'OH',
         'info': {'governor': 'John Kasich'},
         'counties': [{'name': 'Summit', 'population': 1234},
                      {'name': 'Cuyahoga', 'population': 1337}]}]
```

In [7]:
```
# use normalization to create tables from nested element
json_normalize(data, 'counties')
```

Out[7]:

	name	population
0	Dade	12345
1	Broward	40000
2	Palm Beach	60000
3	Summit	1234
4	Cuyahoga	1337

In [8]:
```
# further populate tables created from nested element
json_normalize(data, 'counties', ['state', 'shortname', ['info', 'governor']])
```

Out[8]:

	name	population	state	shortname	info.governor
0	Dade	12345	Florida	FL	Rick Scott
1	Broward	40000	Florida	FL	Rick Scott
2	Palm Beach	60000	Florida	FL	Rick Scott
3	Summit	1234	Ohio	OH	John Kasich
4	Cuyahoga	1337	Ohio	OH	John Kasich

JSON example, with file

- demonstrates reading in a json file as a string and as a table
- uses small sample file containing data about projects funded by the World Bank
- data source: http://jsonstudio.com/resources/

In [9]:
```
# load json as string
json.load((open('data/world_bank_projects_less.json')))
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-9-721b6769f6f5> in <module>
      1 # load json as string
----> 2 json.load((open('data/world_bank_projects_less.json')))

FileNotFoundError: [Errno 2] No such file or directory: 'data/world_bank_projects_less.json'
```

In [ ]:
```
# load as Pandas dataframe
sample_json_df = pd.read_json('data/world_bank_projects_less.json')
sample_json_df
```

JSON exercise

Using data in file 'data/world_bank_projects.json' and the techniques demonstrated above,

1. Find the 10 countries with most projects
2. Find the top 10 major project themes (using column 'mjtheme_namecode')
3. In 2. above you will notice that some entries have only the code and the name is missing. Create a dataframe with the missing names filled in.

In [20]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [21]:
```
df = pd.read_json('world_bank_projects.json')
df.head()
```

Out[21]:

	_id	approvalfy	board_approval_month	boardapprovaldate	borrower	closingdate	country
0	{'$oid': '52b213b38594d8a2be17c780'}	1999	November	2013-11-12T00:00:00Z	FEDERAL DEMOCRATIC REPUBLIC OF ETHIOPIA	2018-07-07T00:00:00Z	Federa E
1	{'$oid': '52b213b38594d8a2be17c781'}	2015	November	2013-11-04T00:00:00Z	GOVERNMENT OF TUNISIA	NaN	
2	{'$oid': '52b213b38594d8a2be17c782'}	2014	November	2013-11-01T00:00:00Z	MINISTRY OF FINANCE AND ECONOMIC DEVEL	NaN	
3	{'$oid': '52b213b38594d8a2be17c783'}	2014	October	2013-10-31T00:00:00Z	MIN. OF PLANNING AND INT'L COOPERATION	NaN	
4	{'$oid': '52b213b38594d8a2be17c784'}	2014	October	2013-10-31T00:00:00Z	MINISTRY OF FINANCE	2019-04-30T00:00:00Z	L

5 rows × 50 columns

In [22]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 50 columns):
_id                         500 non-null object
approvalfy                  500 non-null int64
board_approval_month        500 non-null object
boardapprovaldate           500 non-null object
borrower                    485 non-null object
closingdate                 370 non-null object
country_namecode            500 non-null object
countrycode                 500 non-null object
countryname                 500 non-null object
countryshortname            500 non-null object
docty                       446 non-null object
envassesmentcategorycode    430 non-null object
grantamt                    500 non-null int64
ibrdcommamt                 500 non-null int64
id                          500 non-null object
idacommamt                  500 non-null int64
impagency                   472 non-null object
lendinginstr                495 non-null object
lendinginstrtype            495 non-null object
lendprojectcost             500 non-null int64
majorsector_percent         500 non-null object
mjsector_namecode           500 non-null object
mjtheme                     491 non-null object
mjtheme_namecode            500 non-null object
mjthemecode                 500 non-null object
prodline                    500 non-null object
prodlinetext                500 non-null object
productlinetype             500 non-null object
project_abstract            362 non-null object
project_name                500 non-null object
projectdocs                 446 non-null object
projectfinancialtype        500 non-null object
projectstatusdisplay        500 non-null object
regionname                  500 non-null object
sector                      500 non-null object
sector1                     500 non-null object
sector2                     380 non-null object
sector3                     265 non-null object
sector4                     174 non-null object
sector_namecode             500 non-null object
sectorcode                  500 non-null object
source                      500 non-null object
status                      500 non-null object
supplementprojectflg        498 non-null object
theme1                      500 non-null object
theme_namecode              491 non-null object
themecode                   491 non-null object
totalamt                    500 non-null int64
totalcommamt                500 non-null int64
url                         500 non-null object
dtypes: int64(7), object(43)
memory usage: 195.4+ KB
```

In [23]:
```
df.dtypes
```

Out[23]:
```
_id                         object
approvalfy                   int64
board_approval_month        object
boardapprovaldate           object
borrower                    object
closingdate                 object
country_namecode            object
countrycode                 object
countryname                 object
countryshortname            object
docty                       object
envassesmentcategorycode    object
grantamt                     int64
ibrdcommamt                  int64
id                          object
idacommamt                   int64
impagency                   object
lendinginstr                object
lendinginstrtype            object
lendprojectcost              int64
majorsector_percent         object
mjsector_namecode           object
mjtheme                     object
mjtheme_namecode            object
mjthemecode                 object
prodline                    object
prodlinetext                object
productlinetype             object
project_abstract            object
project_name                object
projectdocs                 object
projectfinancialtype        object
projectstatusdisplay        object
regionname                  object
sector                      object
sector1                     object
sector2                     object
sector3                     object
sector4                     object
sector_namecode             object
sectorcode                  object
source                      object
status                      object
supplementprojectflg        object
theme1                      object
theme_namecode              object
themecode                   object
totalamt                     int64
totalcommamt                 int64
url                         object
dtype: object
```

In [24]:
```
df.describe()
```

Out[24]:

	approvalfy	grantamt	ibrdcommamt	idacommamt	lendprojectcost	totalamt	totalcommamt
count	500.000000	5.000000e+02	5.000000e+02	5.000000e+02	5.000000e+02	5.000000e+02	5.000000e+02
mean	2013.108000	4.432400e+06	3.286010e+07	3.542136e+07	1.547241e+08	6.828146e+07	7.271386e+07
std	0.722066	2.023307e+07	1.089197e+08	7.681431e+07	4.764211e+08	1.242662e+08	1.234705e+08
min	1999.000000	0.000000e+00	0.000000e+00	0.000000e+00	3.000000e+04	0.000000e+00	3.000000e+04
25%	2013.000000	0.000000e+00	0.000000e+00	0.000000e+00	6.472500e+06	0.000000e+00	5.000000e+06
50%	2013.000000	0.000000e+00	0.000000e+00	0.000000e+00	3.500000e+07	2.000000e+07	2.500000e+07
75%	2013.000000	1.695000e+06	0.000000e+00	3.700000e+07	1.021250e+08	8.625000e+07	9.045000e+07
max	2015.000000	3.650000e+08	1.307800e+09	6.000000e+08	5.170000e+09	1.307800e+09	1.307800e+09

In [25]:
```
df.dtypes
```

Out[25]:
```
_id                         object
approvalfy                   int64
board_approval_month        object
boardapprovaldate           object
borrower                    object
closingdate                 object
country_namecode            object
countrycode                 object
countryname                 object
countryshortname            object
docty                       object
envassesmentcategorycode    object
grantamt                     int64
ibrdcommamt                  int64
id                          object
idacommamt                   int64
impagency                   object
lendinginstr                object
lendinginstrtype            object
lendprojectcost              int64
majorsector_percent         object
mjsector_namecode           object
mjtheme                     object
mjtheme_namecode            object
mjthemecode                 object
prodline                    object
prodlinetext                object
productlinetype             object
project_abstract            object
project_name                object
projectdocs                 object
projectfinancialtype        object
projectstatusdisplay        object
regionname                  object
sector                      object
sector1                     object
sector2                     object
sector3                     object
sector4                     object
sector_namecode             object
sectorcode                  object
source                      object
status                      object
supplementprojectflg        object
theme1                      object
theme_namecode              object
themecode                   object
totalamt                     int64
totalcommamt                 int64
url                         object
dtype: object
```

10 countries with most projects

In [26]:
```
df['countryname'].value_counts().head(10)
```

Out[26]:
```
People's Republic of China         19
Republic of Indonesia              19
Socialist Republic of Vietnam      17
Republic of India                  16
Republic of Yemen                  13
Kingdom of Morocco                 12
Nepal                              12
People's Republic of Bangladesh    12
Republic of Mozambique             11
Africa                             11
Name: countryname, dtype: int64
```

Top 10 major project themes

In [27]:
```
df['mjtheme_namecode'].value_counts().head(10)
```

Out[27]:
```
[{'code': '11', 'name': 'Environment and natural resources management'}, {'code': '11', 'name':
'Environment and natural resources management'}]
12
[{'code': '8', 'name': 'Human development'}, {'code': '11', 'name': ''}]
11
[{'code': '8', 'name': 'Human development'}, {'code': '8', 'name': 'Human development'}]
8
[{'code': '4', 'name': 'Financial and private sector development'}, {'code': '4', 'name': 'Finan
cial and private sector development'}]
6
[{'code': '2', 'name': 'Public sector governance'}, {'code': '2', 'name': 'Public sector governa
nce'}, {'code': '2', 'name': 'Public sector governance'}]
6
[{'code': '8', 'name': 'Human development'}, {'code': '7', 'name': 'Social dev/gender/inclusio
n'}]
5
[{'code': '8', 'name': 'Human development'}, {'code': '8', 'name': 'Human development'}, {'cod
e': '8', 'name': 'Human development'}, {'code': '8', 'name': 'Human development'}, {'code': '8',
'name': 'Human development'}]      5
[{'code': '11', 'name': 'Environment and natural resources management'}, {'code': '11', 'name':
''}]
5
[{'code': '11', 'name': 'Environment and natural resources management'}, {'code': '4', 'name':
''}]
5
[{'code': '4', 'name': 'Financial and private sector development'}, {'code': '5', 'name': 'Trade
and integration'}]
5
Name: mjtheme_namecode, dtype: int64
```

In [28]:
```
import pandas as pd
import json
from pandas.io.json import json_normalize
```

In [29]:
```
with open(r'C:\Users\user\Desktop\spring board\data_wrangling_json\data\world_bank_projects.json') a
s f:
    raw = json.load(f)

df_themes = json_normalize(raw, 'mjtheme_namecode', ['id'])
```

Created DataFrame with missing values

In [30]:
```
df_themes.head()
```

Out[30]:

	code	name	id
0	8	Human development	P129828
1	11		P129828
2	1	Economic management	P144674
3	6	Social protection and risk management	P144674
4	5	Trade and integration	P145310

In [31]:
```
df_themes.sort_values('code',inplace = True)
df_themes
```

Out[31]:

	code	name	id
458	1	Economic management	P130925
1235	1	Economic management	P131440
1230	1	Economic management	P129465
1229	1	Economic management	P129465
1218	1	Economic management	P130824
900	1	Economic management	P143819
648	1	Economic management	P129825
647	1	Economic management	P129825
1078	1	Economic management	P131234
1206	1	Economic management	P133255
1437	1		P130412
357	1	Economic management	P130459
363	1		P144030
1010	1	Economic management	P133706
784	1	Economic management	P110836
1024	1		P124114
88	1	Economic management	P132425
1045	1	Economic management	P133791
497	1	Economic management	P133663
1056	1	Economic management	P144633
1057	1	Economic management	P144633
1114	1		P131094
454	1	Economic management	P130925
453	1	Economic management	P128573
1257	1	Economic management	P133230
1260	1	Economic management	P133339
1212	1	Economic management	P121019
249	1	Economic management	P132948
841	1	Economic management	P122793
220	1	Economic management	P144917
...	...	...	...
669	9	Urban development	P127303
1303	9	Urban development	P130528
1347	9	Urban development	P131394
460	9		P117394
1342	9	Urban development	P130174
183	9	Urban development	P121917
200	9	Urban development	P125120
1318	9	Urban development	P130461
1290	9	Urban development	P118152
585	9	Urban development	P123729
737	9	Urban development	P117876
822	9	Urban development	P125022
369	9	Urban development	P126749
597	9	Urban development	P129713
53	9	Urban development	P145359
598	9	Urban development	P129713
356	9	Urban development	P127543
354	9	Urban development	P127543
962	9	Urban development	P132268
738	9	Urban development	P117876
512	9	Urban development	P127163
742	9	Urban development	P127079
1428	9	Urban development	P118916
333	9		P132154
1426	9	Urban development	P112901
1425	9	Urban development	P112901
447	9	Urban development	P126899
739	9	Urban development	P127079
341	9	Urban development	P144357
1102	9	Urban development	P130972

1499 rows × 3 columns

Filled missing values with their respective values

In [35]:
```
df1 = df_themes.replace({'':np.nan})
df1
df1.fillna(method = 'ffill', inplace = True)
df1
```

Out[35]:

	code	name	id
458	1	Economic management	P130925
1235	1	Economic management	P131440
1230	1	Economic management	P129465
1229	1	Economic management	P129465
1218	1	Economic management	P130824
900	1	Economic management	P143819
648	1	Economic management	P129825
647	1	Economic management	P129825
1078	1	Economic management	P131234
1206	1	Economic management	P133255
1437	1	Economic management	P130412
357	1	Economic management	P130459
363	1	Economic management	P144030
1010	1	Economic management	P133706
784	1	Economic management	P110836
1024	1	Economic management	P124114
88	1	Economic management	P132425
1045	1	Economic management	P133791
497	1	Economic management	P133663
1056	1	Economic management	P144633
1057	1	Economic management	P144633
1114	1	Economic management	P131094
454	1	Economic management	P130925
453	1	Economic management	P128573
1257	1	Economic management	P133230
1260	1	Economic management	P133339
1212	1	Economic management	P121019
249	1	Economic management	P132948
841	1	Economic management	P122793
220	1	Economic management	P144917
...	...	...	...
669	9	Urban development	P127303
1303	9	Urban development	P130528
1347	9	Urban development	P131394
460	9	Urban development	P117394
1342	9	Urban development	P130174
183	9	Urban development	P121917
200	9	Urban development	P125120
1318	9	Urban development	P130461
1290	9	Urban development	P118152
585	9	Urban development	P123729
737	9	Urban development	P117876
822	9	Urban development	P125022
369	9	Urban development	P126749
597	9	Urban development	P129713
53	9	Urban development	P145359
598	9	Urban development	P129713
356	9	Urban development	P127543
354	9	Urban development	P127543
962	9	Urban development	P132268
738	9	Urban development	P117876
512	9	Urban development	P127163
742	9	Urban development	P127079
1428	9	Urban development	P118916
333	9	Urban development	P132154
1426	9	Urban development	P112901
1425	9	Urban development	P112901
447	9	Urban development	P126899
739	9	Urban development	P127079
341	9	Urban development	P144357
1102	9	Urban development	P130972

1499 rows × 3 columns

In [ ]: