2.1

June 16, 2021

[1]:
```
import pandas as pd
import s3fs
```

[5]:
```
s3 = s3fs.S3FileSystem(
    anon=True,
    client_kwargs={
        'endpoint_url': 'https://storage.budsc.midwest-datascience.com'
      }
   )

df = pd.read_csv(
    s3.open('data/external/tidynomicon/site.csv', mode='rb')
 )

df2 = pd.read_csv(
    s3.open('data/external/tidynomicon/measurements.csv', mode='rb')
 )

df3 = pd.read_csv(
    s3.open('data/external/tidynomicon/person.csv', mode='rb')
 )

df4 = pd.read_csv(
    s3.open('data/external/tidynomicon/visited.csv', mode='rb')
 )
```

[4]:
```
df.head()
```

[4]:
```
  site_id  latitude  longitude
0    DR-1    -49.85    -128.57
1    DR-3    -47.15    -126.72
2   MSK-4    -48.87    -123.40
```

[6]:
```
df2.head()
```

[6]:
```
   visit_id person_id quantity  reading
0       619      dyer      rad     9.82
```

```
1       619      dyer      sal     0.13
2       622      dyer      rad     7.80
3       622      dyer      sal     0.09
4       734        pb      rad     8.41
```

```
[8]: df3.head()
```

```
[8]:   person_id personal_name family_name
     0      dyer       William        Dyer
     1        pb         Frank     Pabodie
     2      lake      Anderson        Lake
     3       roe     Valentina     Roerich
     4  danforth         Frank    Danforth
```

```
[9]: df4.head()
```

```
[9]:    visit_id site_id  visit_date
     0       619    DR-1  1927-02-08
     1       622    DR-1  1927-02-10
     2       734    DR-3  1930-01-07
     3       735    DR-3  1930-01-12
     4       751    DR-3  1930-02-26
```

```
[10]: import json
      from pathlib import Path
      import os

      import pandas as pd
      import s3fs


      def read_cluster_csv(file_path, endpoint_url='https://storage.budsc.
       ↪midwest-datascience.com'):
          s3 = s3fs.S3FileSystem(
              anon=True,
              client_kwargs={
                  'endpoint_url': endpoint_url
              }
          )
          return pd.read_csv(s3.open(file_path, mode='rb'))

      current_dir = Path(os.getcwd()).absolute()
      results_dir = current_dir.joinpath('results')
      kv_data_dir = results_dir.joinpath('kvdb')
      kv_data_dir.mkdir(parents=True, exist_ok=True)

      people_json = kv_data_dir.joinpath('people.json')
```

```
visited_json = kv_data_dir.joinpath('visited.json')
sites_json = kv_data_dir.joinpath('sites.json')
measurements_json = kv_data_dir.joinpath('measurements.json')
```

[12]:

```
class KVDB(object):
    def __init__(self, db_path):
        self._db_path = Path(db_path)
        self._db = {}
        self._load_db()

    def _load_db(self):
        if self._db_path.exists():
            with open(self._db_path) as f:
                self._db = json.load(f)

    def get_value(self, key):
        return self._db.get(key)

    def set_value(self, key, value):
        self._db[key] = value

    def save(self):
        with open(self._db_path, 'w') as f:
            json.dump(self._db, f, indent=2)
```

[14]:

```
def create_sites_kvdb():
    db = KVDB(sites_json)
    df = read_cluster_csv('data/external/tidynomicon/site.csv')
    for site_id, group_df in df.groupby('site_id'):
        db.set_value(site_id, group_df.to_dict(orient='records')[0])
    db.save()


def create_people_kvdb():
    db = KVDB(people_json)
    ## TODO: Implement code
    df = read_cluster_csv('data/external/tidynomicon/person.csv')
    for person_id, group_df in df.groupby('person_id'):
        db.set_value(person_id, group_df.to_dict(orient='records')[0])
    db.save()


def create_visits_kvdb():
    db = KVDB(visited_json)
    ## TODO: Implement code
    df = read_cluster_csv('data/external/tidynomicon/visited.csv')
    for key, group_df in df.groupby(['visit_id', 'site_id']):
        db.set_value(str(key), group_df.to_dict(orient='records')[0])
    db.save()


def create_measurements_kvdb():
    db = KVDB(measurements_json)
    ## TODO: Implement code
    df = read_cluster_csv('data/external/tidynomicon/measurements.csv')
    for key, group_df in df.groupby(['person_id', 'visit_id', 'quantity']):
        db.set_value(str(key), group_df.to_dict(orient='records')[0])
    db.save()
```

[15]:

```
create_sites_kvdb()
create_people_kvdb()
create_visits_kvdb()
create_measurements_kvdb()
```

[ ]:

```

```