On branch **default**

[1] `LOAD DATASET pdm AS com.crealytics.spark.excel FROM pdm.xlsx @ artifact file 21`

Console ▾ Timing Datasets ▾ Charts ▾

pdm (221 rows)

Views

Park Address (string)	Location (string)	Boundaries - ZIP Codes (short)	Commun
count	count	count	
513 W. 72nd St.	513 W. 72nd St. Chicago, IL (41.763495, -87.637669)	11	66
5010 W. 50th St.	5010 W. 50th St. Chicago, IL (41.802024, -87.748205)	7	53
8930 S. Muskegon Ave.	8930 S. Muskegon Ave. Chicago, IL (41.732921, -87.555452)	25	42
in Chicago 5800 N. Lake Shore Dr.	5800 N. Lake Shore Dr. Chicago, IL		
515 S. Washtenaw Ave.	515 S. Washtenaw Ave. Chicago, IL (41.874341, -87.6936999)	28	28
401 W. 123rd St.	401 W. 123rd St. Chicago, IL (41.670629, -87.6324)	19	50
3309 S. Shields Ave.	3309 S. Shields Ave. Chicago, IL (41.834212, -87.635226)	40	35
1001 W. Wrightwood Ave.	1001 W. Wrightwood Ave. Chicago, IL (41.9290029, -87.6538839)	16	68
5800 N. Lake Shore Dr.	5800 N. Lake Shore Dr. Chicago, IL		
2021 N. Burling St.	2021 N. Burling St. Chicago, IL (41.919044, -87.647303)	16	68

[2]

```
ds = vizierdb.get_dataset('pdm')
'''
issue-1: drop useless columns according to description that only have 1@ columns.
https: //data. cityofchicago.org/Events/Chicago-Park-District-Movies-in-the-Parks-2018/e2v8-k3us
columns_to_drop = ["Community Areas", "Boundaries - ZIP Codes", "Zip Codes", "Census Tracts", "Wards"]
'''
ds.delete_column( 'Community Areas')
ds.delete_column('Boundaries - ZIP Codes')
ds.delete_column( 'Zip Codes')
ds.delete_column( 'Census Tracts')
ds.delete_column( 'Wards')
vizierdb.update_dataset('pdm', ds)
```

Console ▾ Timing Datasets ▾ Charts ▾

pdm (221 rows)

Views

Underwriter (string)	Park Address (string)	Location (string)
count	count	count
	513 W. 72nd St.	513 W. 72nd St. Chicago, IL (41.763495, -87.637669)
	5010 W. 50th St.	5010 W. 50th St. Chicago, IL (41.802024, -87.748205)
	8930 S. Muskegon Ave.	8930 S. Muskegon Ave. Chicago, IL (41.732921, -87.555452)
The Cultural Services at the Consulate General of France in Chicago	5800 N. Lake Shore Dr.	5800 N. Lake Shore Dr. Chicago, IL
	515 S. Washtenaw Ave.	515 S. Washtenaw Ave. Chicago, IL (41.874341, -87.6936999)
	401 W. 123rd St.	401 W. 123rd St. Chicago, IL (41.670629, -87.6324)
	3309 S. Shields Ave.	3309 S. Shields Ave. Chicago, IL (41.834212, -87.635226)
The Wrightwood Neighbors Association	1001 W. Wrightwood Ave.	1001 W. Wrightwood Ave. Chicago, IL (41.9290029, -87.6538839)
	5800 N. Lake Shore Dr.	5800 N. Lake Shore Dr. Chicago, IL
	2021 N. Burling St.	2021 N. Burling St. Chicago, IL (41.919044, -87.647303)

[3]

```
import re
ds = vizierdb.get_dataset('pdm')
'''
issue-2: Remove the duplicate information in the Location column and retain the latitude and longitude.
'''
def extract_coordinates(location_string):
    pattern = r'\(([-\d.]+), [-\d.]+\)'
    match = re.search(pattern, location_string)
    if match:
        return match.group(0)
    else:
        return None

for row in ds.rows:
    loc = extract_coordinates(row.get_value('Location'))
    row.set_value('Location', loc)

vizierdb.update_dataset('pdm', ds)
```

Console ▾ Timing Datasets ▾ Charts ▾

pdm (221 rows)

Views

Park (string)	Park Phone (string)	Title (string)	CC (boolean)	
count	count	count	count	
amilton Park	(312) 747-6174	Black Panther	true	P
ltum Park	(773) 284-6022	Justice League	true	P
essemer Park	(312) 747-6023	Star Wars: The Last Jedi	true	P
sterman Beach	(312) 742-1154	Donkey Skin (Peau d'ane)	true	N
lgaid Park	(312) 746-5001	Justice League	true	P
iiet Pullman Park	(312) 747-7090	The Fate of the Furious	true	P
rmour Square Park	(312) 747-6012	Cars 3	true	G
znquil Playlot Park	(312) 742-7816	Paddington 2	true	P
sterman Beach	(312) 742-1154	9 to 5	true	P
o Park	(312) 742-7898	The Wizard of Oz	true	G

[4]

```
import requests

ds = vizierdb.get_dataset('pdm')

"""
issue-3: Find the rows with missing latitude and longitude and
    use the Google API query address to populate the latitude and longitude
"""
api_key = 'AIzaSyDhSfxYmNWFksZlWDFUVFJKrAApLSO7ZQ'

def get_coordinates_from_api(park_address):
    api_url = f'https://maps.googleapis.com/maps/api/geocode/json?address={park_address}&key={api_key}'
    response = requests.get(api_url)
    data = response.json()
    if data['status'] == 'OK':
        location = data['results'][0]['geometry']['location']
        return f"({location['lat']}, {location['lng']})"
    else:
        # If the initial request fails, try again with "Chicago, IL" appended
        api_url = f'https://maps.googleapis.com/maps/api/geocode/json?address={park_address}, Chicago, IL&key={api_key}'
        response = requests.get(api_url)
        data = response.json()
        if data['status'] == 'OK':
            location = data['results'][0]['geometry']['location']
            return f"({location['lat']}, {location['lng']})"
        else:
            return None

for row in ds.rows:
    if row.get_value('Location') is None:
        loc = get_coordinates_from_api(row.get_value('Park Address'))
        row.set_value('Location', loc)


vizierdb.update_dataset('pdm', ds)
```

Console ▾ Timing Datasets ▾ Charts ▾

pdm (221 rows)

Views

(string)	Title (string)	CC (boolean)	Rating (string)	Underwriter (string)	Park Address (string)	Location (string)
74	Black Panther	true	PG-13		513 W. 72nd St.	(41.763495, -87.637669)
102	Justice League	true	PG-13		5010 W. 50th St.	(41.802024, -87.748205)
103	Star Wars: The Last Jedi	true	PG-13		8930 S. Muskegon Ave.	(41.732921, -87.555452)
34	Donkey Skin (Peau d'ane)	true	NR	The Cultural Services at the Consulate General of France in Chicago	5800 N. Lake Shore Dr.	(41.9861926, -87.6529006)
101	Justice League	true	PG-13		515 S. Washtenaw Ave.	(41.874341, -87.6936999)
190	The Fate of the Furious	true	PG-13		401 W. 123rd St.	(41.670629, -87.6324)
112	Cars 3	true	G		3309 S. Shields Ave.	(41.834212, -87.635226)
116	Paddington 2	true	PG	The Wrightwood Neighbors Association	1001 W. Wrightwood Ave.	(41.9290029, -87.6538839)
34	9 to 5	true	PG		5800 N. Lake Shore Dr.	(41.9861926, -87.6529006)
198	The Wizard of Oz	true	G		2021 N. Burling St.	(41.919044, -87.647303)

[5]

```
from datetime import datetime

"""
issue-4: verify time and day of the week column
"""
ds = vizierdb.get_dataset('pdm')

match_all = True

for row in ds.rows:
    date = datetime.strptime(row.get_value('Date'), '%m/%d/%Y').strftime('%A')
    day_of_week = date[:3]
    if day_of_week != row.get_value('Day'):
        match_all = False
        print(f"Day and Date mismatch in row {row.get_value('Day')} vs {date}")

if match_all:
    print("All rows match")

"""
issue-5:  verify Location column
"""

all_locations_not_none = True

for row in ds.rows:
    location = row.get_value('Location')
    if location is None:
        all_locations_not_none = False
        print(f"None value found in Location column in row {row}")
    '''

if all_locations_not_none:
    print("All rows have non-None values in the Location column")
```

Console ▾ Timing Datasets ▾ Charts ▾

```
All rows match
All rows have non-None values in the Location column
```

[6]

```
import re
ds = vizierdb.get_dataset('pdm')
"""
issue-6: Do some data mining.
Map location and Movie Rating in the OpenStreetMap
"""

def parse_location(location):
    if location is not None:
        match = re.match(r'\(([-\d.]+), ([-\d.]+)\)', location)
        if match:
            x = match.group(1)
            y = match.group(2)
            return x, y
    return None

# Create a new column with a given name. The column
# position is optional.
ds.insert_column('latitude', position=11)
ds.insert_column('longitude', position=12)

for row in ds.rows:
    location = row.get_value('Location')
    x, y = parse_location(location)
    row.set_value('latitude', x)
    row.set_value('longitude',y)


# Render an interactive OpenStreetMap with points marked
# (Expects a dataset named 'ds')
ds.show_map(
  lat_col = 'latitude',
  lon_col = 'longitude',
  label_col = 'Rating' #
)
```

Console ▾ Timing Datasets ▾ Charts ▾