In [1]:
```
'''
Author: Qiyan Zhong
Date: 01/09/2022
Tool: anaconda jupyter notebook
Purpose: Data wrangling for bird open data
'''
```

Out[1]:
```
'\nAuthor: Qiyan Zhong\nDate: 01/09/2022\nTool: anaconda jupyter n
otebook\nPurpose: Data wrangling for bird open data\n'
```

In [2]:
```
#import packages
import shapefile
from shapely.geometry import Point
from shapely.geometry import shape
from shapely.geometry.polygon import Polygon
import pandas as pd
```

In [3]:
```
#read shape file
sf = shapefile.Reader("VIC_LOCALITY_POLYGON_shp.shp")
```

In [4]:
```
shapes = sf.shapes()
records = sf.records()
```

In [5]:
```
#have a look at one record
#we can see that the 7th index have name of suburb
records[20]
```

Out[5]:
```
Record #20: ['10418', datetime.date(2016, 12, 16), None, 'VIC2908'
, datetime.date(2017, 2, 2), None, 'YAPEEN', '', '', 'G', None, '2
']
```

In [6]:
```
#create a list to store suburb name
suburb_list = []
for i in range (len(records)):
    suburb = records[i][6]
    suburb_list.append(suburb)
suburb_list
```

Out[6]:
```
['UNDERBOOL',
 'NURRAN',
 'WOORNDOO',
 'DEPTFORD',
 'YANAC',
 'MINIMAY',
 'GLEN FORBES',
 'ADAMS ESTATE',
 'DIMBOOLA',
 'CANNUM',
 'WALLUP',
 'MURRA WARRA',
 'KALKEE',
 'WAIL',
 'PIMPINIO',
 'DOOEN',
 'VECTIS',
 'QUANTONG',
 'CARWARP',
```

In [7]:
```
#the points included in a shape
s = sf.shape(1)
s.points
```

Out[7]:
```
[(148.668767, -37.39571245),
 (148.66876202, -37.39571345),
 (148.66848331, -37.39576293),
 (148.66821178, -37.39581231),
 (148.66789227, -37.3959711),
 (148.66766529, -37.39609431),
 (148.66754021, -37.3962906),
 (148.66745957, -37.396555),
 (148.66732943, -37.39685439),
 (148.66719625, -37.39700499),
 (148.6671564, -37.39716009),
 (148.66724878, -37.397468),
 (148.66744026, -37.39771163),
 (148.6675745, -37.39796173),
 (148.66755685, -37.39815088),
 (148.6673804, -37.3982906),
 (148.66717597, -37.39846504),
 (148.66710756, -37.39862623),
 (148.66697508, -37.39881117),
```

In [8]:
```
#create a dictionary
#key is suburb, values are the points included in that suburb
suburb_dict = {}
for i in range(len(sf)):
    #list contains all point for that shape
    point_list = sf.shape(i).points
    #update dictionary
    key = suburb_list[i]
    suburb_dict[key]= point_list
suburb_dict
```

```
  (148.66693184, -37.40439895),
  (148.66643551, -37.40433109),
  (148.66621132, -37.40424245),
  (148.66584643, -37.40393814),
  (148.66556912, -37.40370708),
  (148.6653882, -37.40362933),
  (148.66516789, -37.40372955),
  (148.66514566, -37.4040447),
  (148.66521617, -37.4043357199999),
  (148.66536487, -37.40459136),
  (148.665663, -37.4047878),
  (148.66599382, -37.40482923),
  (148.66640409, -37.40489824),
  (148.66659313, -37.40502168),
  (148.66665485, -37.40523268),
  (148.66677533, -37.4055116),
  (148.66719091, -37.40583814),
  (148.66764775, -37.40607826),
  (148.66811095, -37.40627822),
  (148.66849702, -37.40621588),
```

In [9]:
```
# read open data store in data frame
df=pd.read_csv('records-2022-08-28.csv')
```

```
/opt/anaconda3/lib/python3.8/site-packages/IPython/core/interactiv
eshell.py:3146: DtypeWarning: Columns (4,8,10,11,12,16,19,20,22,23
,34,50,62,68,70,71,72,73,74,78,79,80,81,92,106,110,111,112,113,157
,165,171,174,189,195,203) have mixed types.Specify dtype option on
import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [10]:
```
#filter the dataframe (needed columns)
df1 = df[['vernacularName',"year", "month",'day','verbatimLatitude'
df1
```

Out[10]:

	vernacularName	year	month	day	verbatimLatitude	verbatimLongitude	
0	White-throated Treecreeper	2022	3.0	17.0	-37.589600	145.691600	https://bic
1	Fan-tailed Cuckoo	2022	3.0	9.0	-37.820400	145.705200	https://bio
2	Grey Shrike-thrush	2022	3.0	18.0	-37.407500	145.940300	https://bic
3	White-fronted Scrubwren	2022	3.0	17.0	-37.589600	145.691600	https://bio
4	Grey Shrike-thrush	2022	3.0	18.0	-37.406600	145.940400	https://bic
...	...	...	...	...	...	...	
24453	Galah	2022	2.0	23.0	-37.848535	144.974635	https://bio
24454	Musk Duck	2022	2.0	15.0	-35.801771	143.872977	https://bio
24455	Black Swan	2022	2.0	28.0	-36.820469	144.222365	https://bic
24456	Noisy Miner	2022	2.0	24.0	-37.844180	144.971030	https://bic
24457	Little Black Cormorant	2022	2.0	23.0	-37.839087	144.970021	https://bic

24458 rows × 11 columns

In [11]:
```
#create a new column for suburb
#set default value to 'not available'
df1['suburb'] = 'Not available'
df1
```

```
<ipython-input-11-2f3dadda061a>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df1['suburb'] = 'Not available'
```

Out[11]:

	vernacularName	year	month	day	verbatimLatitude	verbatimLongitude	
0	White-throated Treecreeper	2022	3.0	17.0	-37.589600	145.691600	https://bic
1	Fan-tailed Cuckoo	2022	3.0	9.0	-37.820400	145.705200	https://bio
2	Grey Shrike-thrush	2022	3.0	18.0	-37.407500	145.940300	https://bic
3	White-fronted Scrubwren	2022	3.0	17.0	-37.589600	145.691600	https://bio
4	Grey Shrike-thrush	2022	3.0	18.0	-37.406600	145.940400	https://bic
...	...	...	...	...	...	...	
24453	Galah	2022	2.0	23.0	-37.848535	144.974635	https://bio
24454	Musk Duck	2022	2.0	15.0	-35.801771	143.872977	https://bio
24455	Black Swan	2022	2.0	28.0	-36.820469	144.222365	https://bic
24456	Noisy Miner	2022	2.0	24.0	-37.844180	144.971030	https://bic
24457	Little Black Cormorant	2022	2.0	23.0	-37.839087	144.970021	https://bic

24458 rows × 12 columns

In [12]:
```
#update the value for suburb column
for i in range(len(df1)):
    # create the required point
    lat = df1.loc[i,'verbatimLatitude']
    lng = df1.loc[i,'verbatimLongitude']
    point = Point(lng,lat)
    for key,values in suburb_dict.items():
        # create shape using points of suburb
```

```
        polygon = Polygon(values)

        # if the shape contain the point
        if polygon.contains(point):
            df1.loc[i,'suburb'] = key
        else:
            pass
df1
```

```
/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py
:1765: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy)
  isetter(loc, value)
```

Out[12]:

	vernacularName	year	month	day	verbatimLatitude	verbatimLongitude	
0	White-throated Treecreeper	2022	3.0	17.0	-37.589600	145.691600	https://bic
1	Fan-tailed Cuckoo	2022	3.0	9.0	-37.820400	145.705200	https://bio
2	Grey Shrike-thrush	2022	3.0	18.0	-37.407500	145.940300	https://bic
3	White-fronted Scrubwren	2022	3.0	17.0	-37.589600	145.691600	https://bio
4	Grey Shrike-thrush	2022	3.0	18.0	-37.406600	145.940400	https://bic
...	...	...	...	...	...	...	
24453	Galah	2022	2.0	23.0	-37.848535	144.974635	https://bio
24454	Musk Duck	2022	2.0	15.0	-35.801771	143.872977	https://bio
24455	Black Swan	2022	2.0	28.0	-36.820469	144.222365	https://bic
24456	Noisy Miner	2022	2.0	24.0	-37.844180	144.971030	https://bic
24457	Little Black Cormorant	2022	2.0	23.0	-37.839087	144.970021	https://bic

24458 rows × 12 columns

In [13]:
```
#remove null value for name column
df2 = df1[df1['vernacularName'].notna()]
#remove null value for suburb column
df3 = df2[df2['suburb'].notna()]
```

In [14]:
```
#output datagframe to csv file
df3.to_csv('bird_data_cleaned.csv',index = False)
```