Australian Coronavirus (COVID-19) Analyst

Introduction

This report is to analyze Australian COVID19 data for general public
Findings of the report is as below

- Most of cases are from NSW and VIC state
- Population, and distance to top affected areas are the major cause contributing to cases
- Lockdown and vaccination is helpful to stop Delta

Step1: Download data from Elephant DB

postgres://mheodhlf:m6FmQMj_66D6CO7BPmOZAfcUG2-La9Tv@rosie.db.elephantsql.com/mheodhlf

In [2]:
```
# %run hide_button.py

import pandas as pd
from statsmodels.formula.api import ols
from matplotlib import pyplot as plt
from sqlalchemy import create_engine
from pandasql import sqldf
engine = create_engine("postgresql://mheodhlf:m6FmQMj_66D6CO7BPmOZAfcUG2-La9Tv@rosie
```

In [3]:
```
# download data into pandas
df_covid_by_state = pd.read_sql("SELECT * FROM covid_by_state", engine)
df_covid_by_state_cum = pd.read_sql("SELECT * FROM covid_by_state_cum", engine)
df_nsw_cases_by_postcode = pd.read_sql("SELECT * FROM nsw_cases_by_postcode", engine
df_vic_cases_by_postcode = pd.read_sql("SELECT * FROM vic_cases_by_postcode", engine
df_australian_postcode = pd.read_sql("SELECT * FROM australian_postcode", engine)
df_australian_suburbs = pd.read_sql("SELECT * FROM australian_suburbs", engine)
# format data
df_covid_by_state['date'] = pd.to_datetime(df_covid_by_state['date'], format="%Y/%m/
df_covid_by_state_cum['date'] = pd.to_datetime(df_covid_by_state_cum['date'], format
```

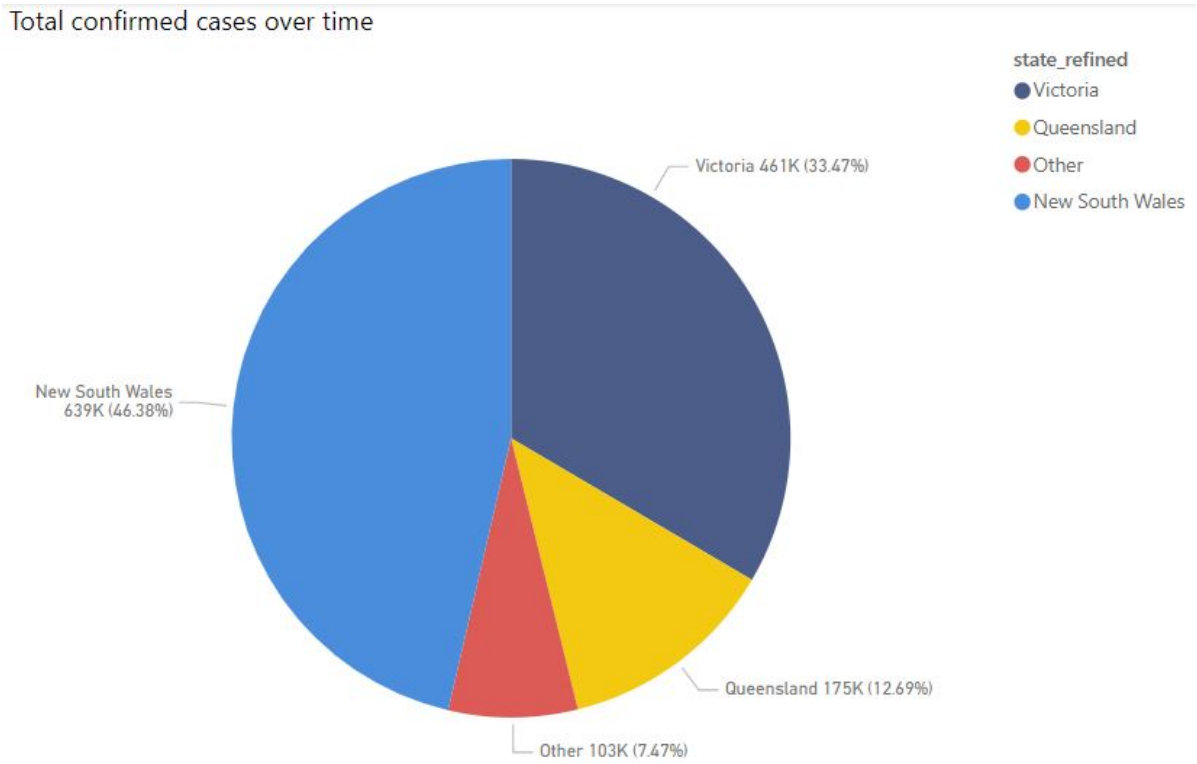
Step2: Visualize the data, and conduct data analysis

Data in states: it finds that NSW and VIC has most of cases in Australia

- NSW: 46.3%, VIC: 33.4%
- QLD and the other states are about 20% in total

In [3]:
```
# Total Confirmed Cases by State
df_sql_1 = sqldf('''
SELECT state
     , CASE
            WHEN state in ('New South Wales', 'Victoria', 'Queensland') THEN state
            ELSE 'Other'
            END AS state_refined
     , population
     , MAX(confirmed) as confirmed_cum
  FROM df_covid_by_state_cum
 WHERE state <> 'None'
```

```
 GROUP BY 1,2
''')
```

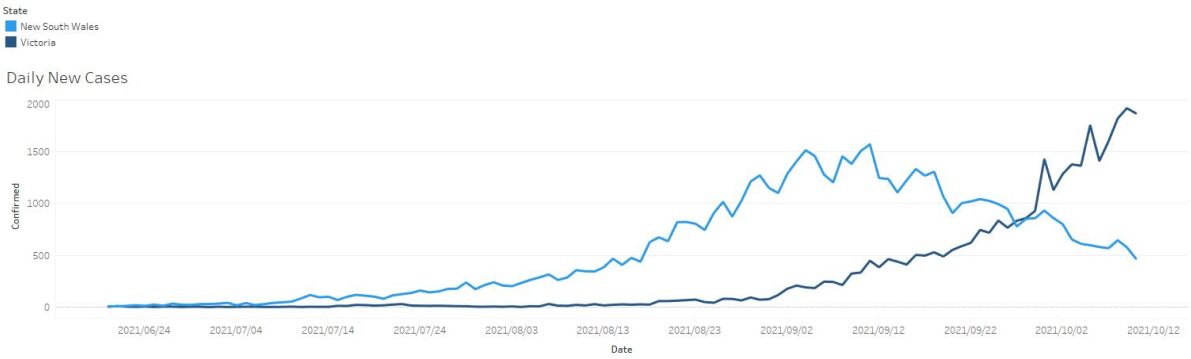


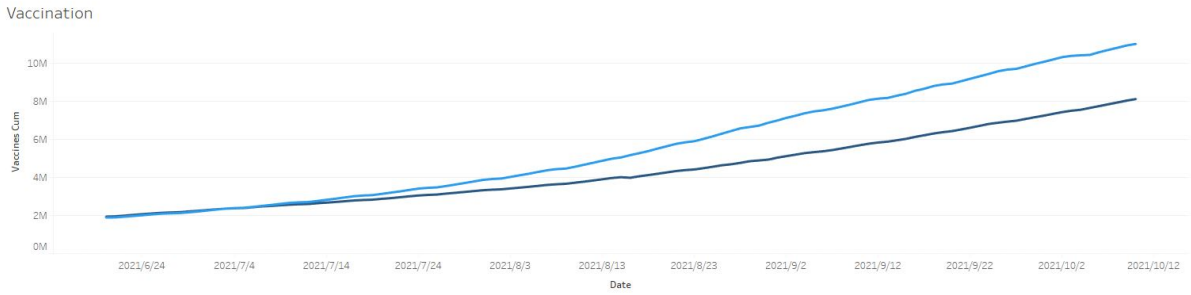
Hyphothesis 1: vaccination is related with covid cases during lockdown

- Check data for NSW and VIC during the Lockdown for Delta (2021-06-20 to 2021-10-10)

In [4]:

```
df_sql_nsw = sqldf('''
SELECT *
  FROM df_covid_by_state
 WHERE state ="New South Wales"
''')

df_sql_vic = sqldf('''
SELECT *
  FROM df_covid_by_state
 WHERE state ="Victoria"
''')
df_sql_nsw['date'] = pd.to_datetime(df_sql_nsw['date'], format="%Y/%m/%d")
df_sql_vic['date'] = pd.to_datetime(df_sql_vic['date'], format="%Y/%m/%d")
df_sql_nsw = df_sql_nsw.where(df_sql_nsw['date'] <='2021-10-10')
df_sql_nsw = df_sql_nsw.where(df_sql_nsw['date'] >='2021-06-20')
df_sql_vic = df_sql_vic.where(df_sql_vic['date'] <='2021-10-10')
df_sql_vic = df_sql_vic.where(df_sql_vic['date'] >='2021-06-20')
```





OLS Analysis for NSW vaccines and cases

R-squared is 0.6, this indicates strong correlation between vaccinations and cases in NSW

In [5]:
```
model_name = 'confirmed ~ vaccines_cum'
model = ols(model_name, data = df_sql_nsw).fit()
model.summary()
```

Out[5]:

OLS Regression Results

Dep. Variable:	confirmed	**R-squared:**	0.609
Model:	OLS	**Adj. R-squared:**	0.606
Method:	Least Squares	**F-statistic:**	173.2
Date:	Sun, 23 Jan 2022	**Prob (F-statistic):**	2.09e-24
Time:	23:41:15	**Log-Likelihood:**	-805.29
No. Observations:	113	**AIC:**	1615.
Df Residuals:	111	**BIC:**	1620.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-188.7203	63.539	-2.970	0.004	-314.627	-62.814
vaccines_cum	0.0001	1e-05	13.162	0.000	0.000	0.000

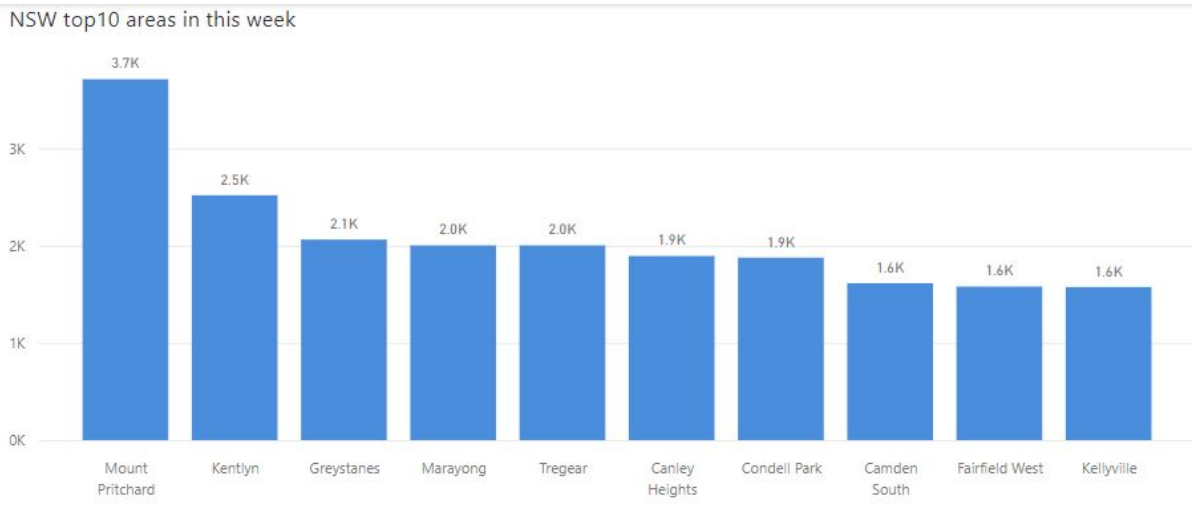
Omnibus:	3.208	**Durbin-Watson:**	0.077
Prob(Omnibus):	0.201	**Jarque-Bera (JB):**	2.664
Skew:	0.269	**Prob(JB):**	0.264
Kurtosis:	3.526	**Cond. No.**	1.41e+07

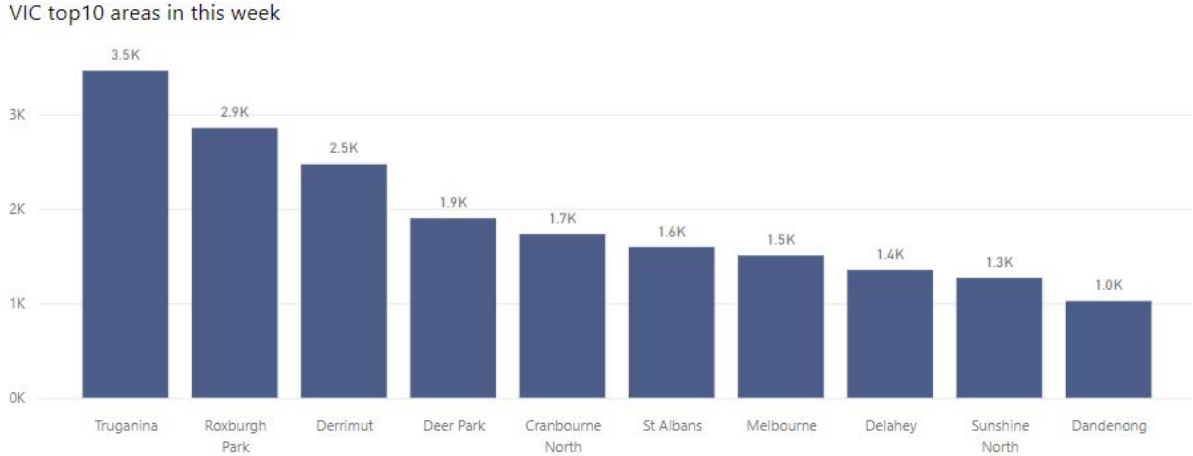
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.41e+07. This might indicate that there are strong multicollinearity or other numerical problems.

Hyphothesis 2: distance to CBD or top1 areas is related with covid cases

- Check data for NSW and VIC, which areas were most of cases

In [6]:
```
#check Top10 suburbs that has the most cases
df_sql_nsw_cases = sqldf('''
SELECT postcode
     , suburb
     , population
     , this_week AS cases
  FROM df_nsw_cases_by_postcode
 ORDER BY this_week DESC
 LIMIT 10
''')

df_sql_vic_cases = sqldf('''
SELECT postcode
     , suburb
     , population
     , active AS cases
  FROM df_vic_cases_by_postcode
 ORDER BY active DESC
 LIMIT 10
''')
```





- Calculate distance of each suburb to top_one suburb, and to CBD -- NSW CBD: 2000, VIC CBD: 3000 -- NSW top1: 2170, VIC top1: 3029

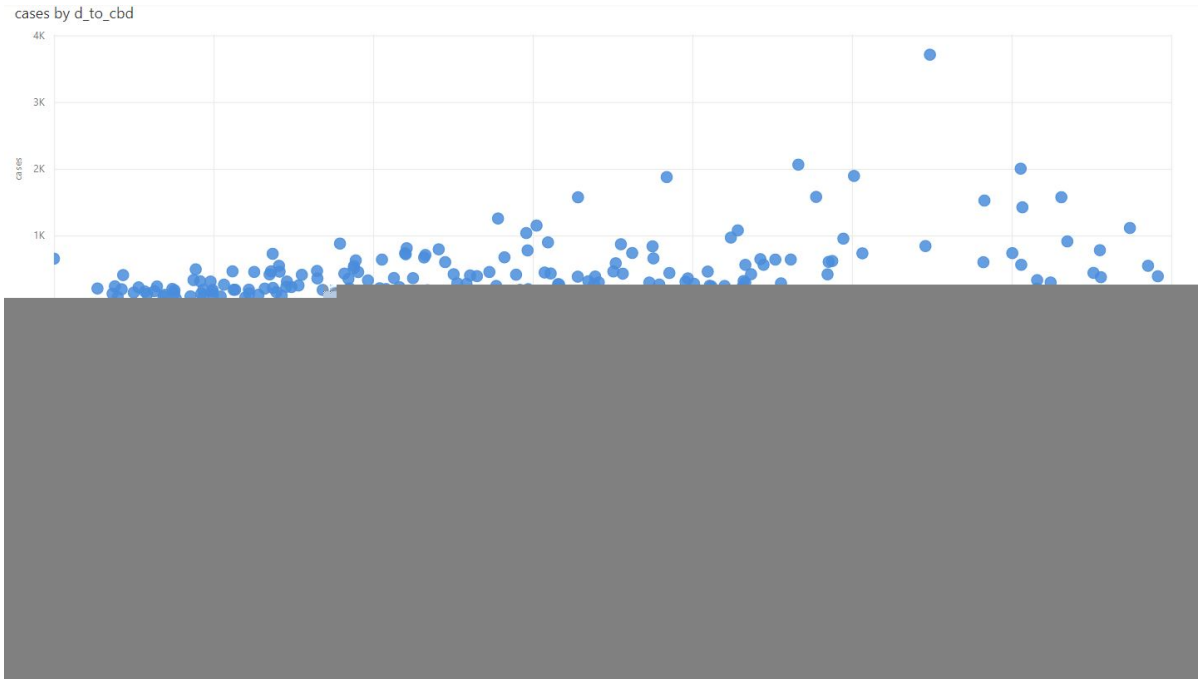
In [7]:

```
from typing import Tuple
from my_haversine import *
import csv


def clean_suburbs_data(raw_suburb_data: dict) -> dict:
    clean_suburb_data = raw_suburb_data.copy()
    lat = clean_suburb_data.get('lat')
    lng = clean_suburb_data.get('lng')
    if lat:
        clean_suburb_data['lat'] = float(lat)
    if lng:
        clean_suburb_data['lng'] = float(lng)
    return clean_suburb_data


def suburb_distance(cleaned_suburb_1: dict, cleaned_suburb_2: dict) -> float:
    s1: Tuple[float, float] = (cleaned_suburb_1['lat'], cleaned_suburb_1['lng'])
    s2: Tuple[float, float] = (cleaned_suburb_2['lat'], cleaned_suburb_2['lng'])
    return haversine(s1, s2)


# def my_read_csv(csv_file_name: str) -> list[dict]:
#     with open(csv_file_name, "r") as f:
#         reader = csv.DictReader(f)
#         return [row for row in reader]


df_australian_suburbs_refined = sqldf('''
SELECT postcode
     , state
     , state_name
     , AVG(lat) AS lat
     , AVG(lng) as lng
  FROM df_australian_suburbs
 GROUP BY 1,2,3
''')

df_australian_suburbs_refined.to_csv('df_australian_suburbs_refined.csv', index = Fa
# cleaned_suburbs = list(map(clean_suburbs_data, my_read_csv('df_australian_suburbs_
# nsw_top1 = next(item for item in cleaned_suburbs if item['postcode'] == '2170')
# vic_top1 = next(item for item in cleaned_suburbs if item['postcode'] == '3029')
# nsw_cbd = next(item for item in cleaned_suburbs if item['postcode'] == '2000')
# vic_cbd = next(item for item in cleaned_suburbs if item['postcode'] == '3000')

# with open("to_top_one_distance.csv", "w", newline = "") as f:
#     field_names = ['postcode', 'distance_to_top_one', 'distance_to_cbd']
#     writer = csv.DictWriter(f, fieldnames = field_names)
#     writer.writeheader()
#     for s in cleaned_suburbs:
#       if s['state'] == 'NSW':
#          postcode = s['postcode']
#          distance = suburb_distance(s, nsw_top1)
#          distance2 = suburb_distance(s, nsw_cbd)
#          writer.writerow(dict(zip(field_names, [postcode, distance, distance2])))
#       elif s['state'] == 'VIC':
#          postcode = s['postcode']
#          distance = suburb_distance(s, vic_top1)
#          distance2 = suburb_distance(s, vic_cbd)
#          writer.writerow(dict(zip(field_names, [postcode, distance, distance2])))
```

```
# pd.read_csv("to_top_one_distance.csv").to_sql('to_top_one_distance', engine, index
```

In [8]:

```
#check relationship of distance to top1 and total cases
df_to_top_one_distance = pd.read_sql("SELECT * FROM to_top_one_distance", engine)
df_sql_nsw_combined = sqldf('''
SELECT a.postcode
     , population
     , this_week AS cases
     , total_cases
     , ROUND(((this_week*1.0)/population)*100,2) as case_percent
     , distance_to_top_one AS d_to_top1
     , distance_to_cbd AS d_to_cbd
  FROM df_nsw_cases_by_postcode AS a
  LEFT JOIN df_to_top_one_distance AS b
    ON a.postcode = b.postcode
 WHERE distance_to_top_one between 0 AND 35
   AND distance_to_cbd between 0 AND 35
''')

df_sql_vic_combined = sqldf('''
SELECT a.postcode
     , population
     , active AS cases
     , total_cases
     , ROUND(((active*1.0)/population)*100,2) as case_percent
     , distance_to_top_one AS d_to_top1
     , distance_to_cbd AS d_to_cbd
  FROM df_vic_cases_by_postcode AS a
  LEFT JOIN df_to_top_one_distance AS b
    ON a.postcode = b.postcode
 WHERE distance_to_top_one between 0 AND 35
   AND distance_to_cbd between 0 AND 35
   AND active < population
   AND case_percent < 30
''')
```



OLS Analysis for NSW distance and cases

R-squared is 0.45, this indicates a correlation between distance and cases

In [9]:
```
model_name = 'case_percent ~ d_to_top1'
model = ols(model_name, data = df_sql_nsw_combined).fit()
model.summary()
```

Out[9]:

OLS Regression Results

Dep. Variable:	case_percent	**R-squared:**	0.451
Model:	OLS	**Adj. R-squared:**	0.448
Method:	Least Squares	**F-statistic:**	142.9
Date:	Sun, 23 Jan 2022	**Prob (F-statistic):**	1.96e-24
Time:	23:41:16	**Log-Likelihood:**	-160.30
No. Observations:	176	**AIC:**	324.6
Df Residuals:	174	**BIC:**	330.9
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3.4339	0.130	26.468	0.000	3.178	3.690
d_to_top1	-0.0691	0.006	-11.956	0.000	-0.081	-0.058

Omnibus:	122.269	**Durbin-Watson:**	1.450
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1446.254
Skew:	2.391	**Prob(JB):**	0.00
Kurtosis:	16.204	**Cond. No.**	64.0

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Hyphothesis 3: Population is related with covid cases

- Check data for NSW and VIC, population and cases of each suburb

OLS Analysis for NSW distance and cases

R-squared is 0.82, this indicates strong correlation between population and cases

In [10]:
```
model_name = 'cases ~ population'
model = ols(model_name, data = df_sql_nsw_combined).fit()
model.summary()
```

Out[10]:

OLS Regression Results

Dep. Variable:	cases	**R-squared:**	0.829
Model:	OLS	**Adj. R-squared:**	0.828
Method:	Least Squares	**F-statistic:**	843.7
Date:	Sun, 23 Jan 2022	**Prob (F-statistic):**	1.22e-68

Time:	23:41:16	**Log-Likelihood:**	-1173.7
No. Observations:	176	**AIC:**	2351.
Df Residuals:	174	**BIC:**	2358.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-140.3022	25.697	-5.460	0.000	-191.021	-89.584
population	0.0275	0.001	29.047	0.000	0.026	0.029

Omnibus:	16.303	**Durbin-Watson:**	0.995
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	32.526
Skew:	0.416	**Prob(JB):**	8.65e-08
Kurtosis:	4.935	**Cond. No.**	4.82e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.82e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Hyphothesis 4: Lockdown is correlated with cases in NSW

AS Covid-19 is becoming a 'New Normal' of our life, this visualisation aims to demonstrate how the currently active cases are located in NSW. Do people who live close to CBD with more people living around have a higher risk of catching covid-19 today? To answer this question, we will compare the overall performance of each suburb during and after the lockdown. The answer is no during the lockdown and yes when all the restrictions have been eased.

In [12]:
```
import pandas as pd
from numpy import nan
import sqlite3
from pandasql import sqldf
from typing import List
import psycopg2
import pandas.io.sql as psql
from haversine import haversine, Unit
from matplotlib import pyplot as plt
plt.rcParams['figure.figsize'] = [10, 5]
from sqlalchemy import create_engine
from statsmodels.formula.api import ols
engine = create_engine('postgresql://aqlxfqja:F6bE-fv-jhA_VaaLV284XVgxXOLNAp_2@rosie
from numpy import cos, sin, arcsin, sqrt
from math import radians
```

Load data from elephant db to local.

In [7]:
```
con = psycopg2.connect('postgresql://aqlxfqja:F6bE-fv-jhA_VaaLV284XVgxXOLNAp_2@rosie
df = psql.read_sql('SELECT * FROM nsw_covidv2', con)
df2 = psql.read_sql('SELECT * FROM suburbs', con)
```

Get the data for the covid pandemic in NSW during the lockdown in 2021 from June 24 to December 15. Then rank the covid cases per 100 population in each postcode from 0 to 1 by summing up each of their ranks. We can estimate their overall ranking during the lockdown.

In [8]:

```
df_lockdown = sqldf('''
WITH enriched AS(

    SELECT strftime('%m', date) AS month
         , postcode
         , suburb
         , active_cases
         , cases_per_100
      FROM df
     WHERE date > '2021-06-24' AND date < '2021-12-15'

), avg_month AS(

   SELECT *
      , AVG(cases_per_100) AS monthly_cases_100
      FROM enriched
   GROUP BY suburb, month
   ORDER BY suburb, month ASC

), all_postcode AS(

   SELECT postcode
        , lat
        , lng
        , suburb
        , SUM(population) AS population
     FROM df2
    WHERE state = 'NSW' AND postcode < '3000'
    GROUP BY postcode

), joined_table AS(

   SELECT a.postcode
      , a.suburb AS suburb
      , a.lat
      , a.lng
      , a.population
      , b.month
      , b.monthly_cases_100
   FROM all_postcode AS a
   LEFT JOIN avg_month AS b
      ON a.postcode = b.postcode

), ranked AS(

   SELECT *
      , PERCENT_RANK() OVER (PARTITION BY month ORDER BY monthly_cases_100 ASC) AS r
   FROM joined_table

)

SELECT postcode
     , lat
     , lng
     , suburb
     , population
     , SUM(rank) AS total_rank
  FROM ranked
```

```
 GROUP BY postcode
 ORDER BY total_rank

''')
print(df_lockdown.head(1))
```

```
   postcode       lat        lng   suburb  population  total_rank
0      2338 -31.72628  150.79265  Ardglen        1428         0.0
```

Calculate the distance from each suburb to Sydney CBD.

In [13]:
```
def distance_to_cbd(row):
    loc1 = [row['lat'], row['lng']]
    syd = [-33.86794, 151.20998]
    return haversine(loc1, syd, unit=Unit.KILOMETERS)

df_lockdown['distance_cbd'] = df_lockdown.apply(lambda row: distance_to_cbd(row), ax
df_lockdown = df_lockdown.drop(columns=['lat', 'lng'])
```

Summing up the monthly ranking during the lockdown, each suburb received a mark to rank their overall performance during the lockdown, and lower is better. Below are the 20 worst performed suburbs.

In [14]:
```
print(df_lockdown.tail(5).sort_values('total_rank', ascending=False))
```

```
     postcode          suburb  population  total_rank  distance_cbd
612      2168           Busby       43449    6.158283     30.755024
611      2191        Belfield        6322    5.987127     12.120299
610      2190     Mount Lewis       25568    5.984774     15.672347
609      2192         Belmore       12718    5.922951     12.651825
608      2174  Edmondson Park        2271    5.855576     34.803631
```

5 best performanced suburbs within 50km from CBD.

In [15]:
```
print(df_lockdown.loc[df_lockdown['distance_cbd'] <= 50].head(5))
```

```
     postcode          suburb  population  total_rank  distance_cbd
40       2555  Badgerys Creek         225    0.000000     43.033619
143      2083       Bar Point        1524    0.358779     41.236992
149      2071         Killara       13552    0.454194     12.005354
167      2082  Berowra Waters        5402    0.617326     30.533327
171      2072          Gordon        7668    0.630901     13.270306
```

R-squared represents how good distance to CBD and population can explain the performance of epidemic prevention during the lockdown. In this case, the correlation is not strong enough. This means, during NSW 2021 lockdown, the distance from a suburb to Sydney CBD and its population doesn't strongly affect the level of risk of catching covid-19.

In [17]:
```
model_name = 'total_rank ~ distance_cbd + population'
model = ols(model_name, data = df_lockdown).fit()
model.summary()
```

Out[17]:

OLS Regression Results

Dep. Variable:	total_rank	**R-squared:**	0.365
Model:	OLS	**Adj. R-squared:**	0.363
Method:	Least Squares	**F-statistic:**	175.3
Date:	Sun, 23 Jan 2022	**Prob (F-statistic):**	6.97e-61

Time:	23:41:23	**Log-Likelihood:**	-992.04
No. Observations:	613	**AIC:**	1990.
Df Residuals:	610	**BIC:**	2003.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.0383	0.097	20.914	0.000	1.847	2.230
distance_cbd	-0.0032	0.000	-12.696	0.000	-0.004	-0.003
population	3.052e-05	3.95e-06	7.733	0.000	2.28e-05	3.83e-05

Omnibus:	50.711	**Durbin-Watson:**	0.647
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	61.482
Skew:	0.769	**Prob(JB):**	4.46e-14
Kurtosis:	3.209	**Cond. No.**	3.60e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.6e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Get the data for the covid pandemic in NSW after eased restrictions in 2021 from DEC 15 to current. We can estimate their overall ranking during the lockdown by summing up their relative position each day.

In [18]:

```
df_eased = sqldf('''
WITH enriched AS(

    SELECT date
         , postcode
         , suburb
         , active_cases
         , cases_per_100
      FROM df
     WHERE date >= '2021-12-15'

), current AS(

   SELECT postcode
        , active_cases
        , new_cases
        , cases_per_100
     FROM df
    WHERE date = '2022-01-17'

), all_postcode AS(

   SELECT postcode
        , lat
        , lng
```

```
        , suburb
        , SUM(population) AS population
     FROM df2
    WHERE state = 'NSW' AND postcode < '3000'
    GROUP BY postcode

), joined_table AS(

   SELECT a.postcode
      , a.suburb AS suburb
      , a.lat
      , a.lng
      , a.population
      , b.date
      , b.cases_per_100
   FROM all_postcode AS a
   LEFT JOIN enriched AS b
      ON a.postcode = b.postcode

), ranked AS(

   SELECT *
      , PERCENT_RANK() OVER (PARTITION BY date ORDER BY cases_per_100 ASC) AS rank
   FROM joined_table

)

SELECT postcode
     , lat
     , lng
     , suburb
     , population
     , SUM(rank) AS total_rank
  FROM ranked
 GROUP BY postcode
 ORDER BY total_rank

''')
print(df_eased.head(1))
```

```
   postcode       lat        lng    suburb  population  total_rank
0      2611 -35.66563  148.70878  Cooleman          52         0.0
```

Add distance to CBD into the table

In [19]:

```
df_eased['distance_cbd'] = df_eased.apply(lambda row: distance_to_cbd(row), axis=1)
df_eased = df_eased.drop(columns=['lat', 'lng'])
```

Like the calculation above, we summed up the daily ranking for each suburb after the lockdown.

Below are the 20 worst performed suburbs in NSW.

In [20]:

```
print(df_eased.tail(5))
```

```
     postcode          suburb  population  total_rank  distance_cbd
608      2026     North Bondi       32488   32.383562      6.507199
609      2762      Schofields        4983   32.433877     35.942849
610      2020          Mascot       14772   32.692533      9.076371
611      2481     Broken Head       11772   33.570675    614.497119
612      2174  Edmondson Park        2271   33.754071     34.803631
```

Below are the 5 best performed suburbs in NSW after restriction eased.

In [21]:
```
print(df_eased.head(5))
```

```
   postcode          suburb  population  total_rank  distance_cbd
0      2611        Cooleman          52    0.000000    303.551512
1      2649  Nurenmerenmong          31    0.000000    342.085900
2      2668       Barmedman         459    0.000000    349.789948
3      2356        Gwabegar         162    0.004098    421.904178
4      2735       Koraleigh         451    0.007590    725.843478
```

5 best performed suburbs in NSW within 50km from Sydney CBD.

In [22]:
```
print(df_eased.loc[df_eased['distance_cbd'] <= 50].head(5))
```

```
     postcode            suburb  population  total_rank  distance_cbd
223      2563     Menangle Park         257    5.675193     49.109234
228      2083         Bar Point        1524    5.971365     41.236992
242      2082    Berowra Waters        5402    6.651336     30.533327
263      2080  Mount Kuring-Gai        1708    7.804422     25.830430
267      2105        Lovett Bay        1854    7.933197     26.921494
```

5 worst performed suburbs in NSW within 50km from Sydney CBD.

In [23]:
```
print(df_eased.loc[df_eased['distance_cbd'] <= 50].tail(5))
```

```
     postcode          suburb  population  total_rank  distance_cbd
607      2179      Leppington        6522   32.001325     39.792150
608      2026     North Bondi       32488   32.383562      6.507199
609      2762      Schofields        4983   32.433877     35.942849
610      2020          Mascot       14772   32.692533      9.076371
612      2174  Edmondson Park        2271   33.754071     34.803631
```

The correlation between suburb performance and CBD distance and suburb population is stronger after restriction eased. After NSW 2021 lockdown, the distance from a suburb to Sydney CBD and the number of people to an extent affects the likelihood of catching covid-19.

In [24]:
```
model_name = 'total_rank ~ distance_cbd + population'
model = ols(model_name, data = df_eased).fit()
model.summary()
```

Out[24]:

OLS Regression Results

Dep. Variable:	total_rank	**R-squared:**	0.525
Model:	OLS	**Adj. R-squared:**	0.523
Method:	Least Squares	**F-statistic:**	337.1
Date:	Sun, 23 Jan 2022	**Prob (F-statistic):**	2.48e-99
Time:	23:41:24	**Log-Likelihood:**	-2059.7
No. Observations:	613	**AIC:**	4125.
Df Residuals:	610	**BIC:**	4139.
Df Model:	2		
Covariance Type:	nonrobust		

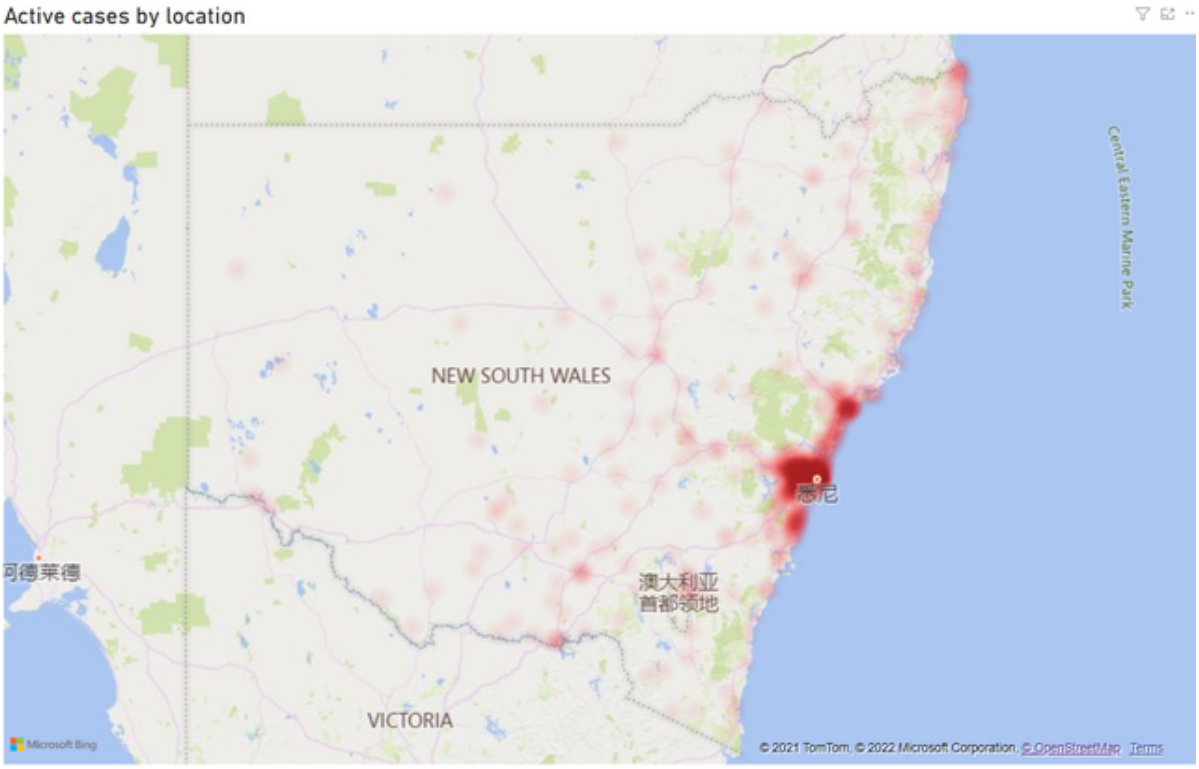
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	16.1465	0.556	29.029	0.000	15.054	17.239
distance_cbd	-0.0281	0.001	-19.331	0.000	-0.031	-0.025

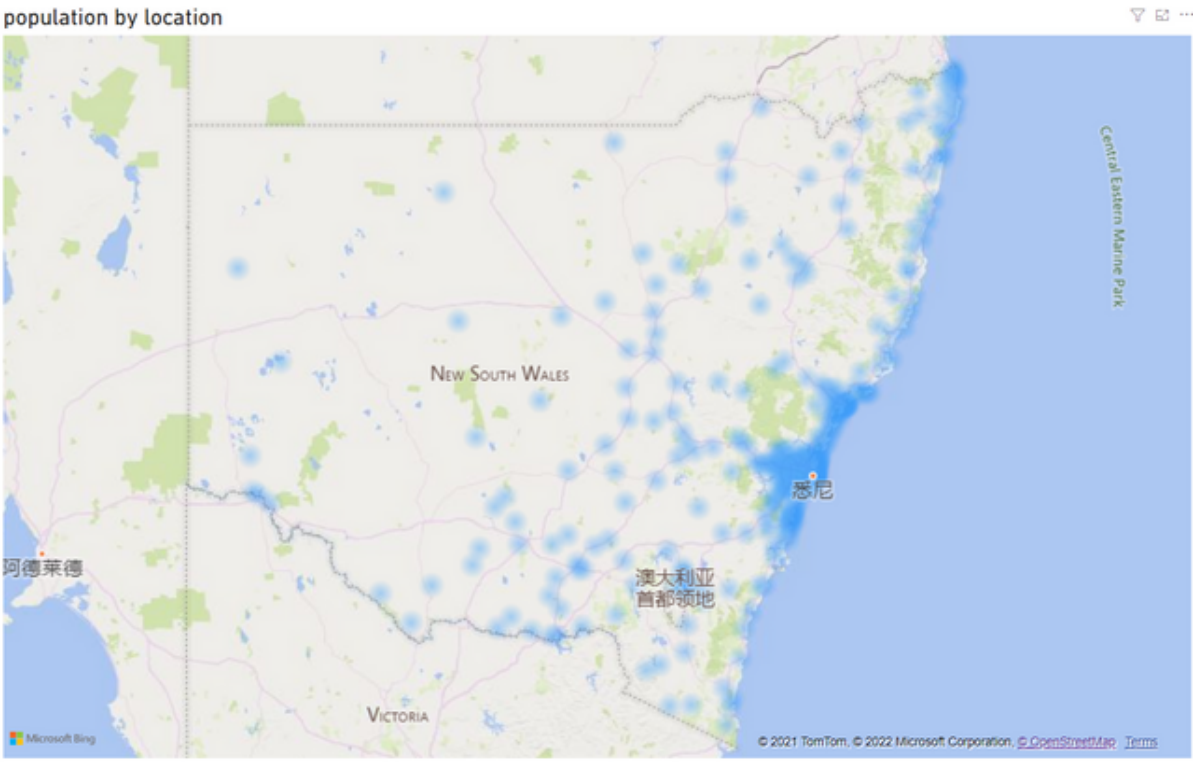
population	0.0002	2.25e-05	8.436	0.000	0.000	0.000

Omnibus:	44.868	**Durbin-Watson:**	0.857
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	54.062
Skew:	0.654	**Prob(JB):**	1.82e-12
Kurtosis:	3.636	**Cond. No.**	3.60e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.6e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

Converting the active case and population into a heat map in PowerBI allows us to visualise this correlation.





Conclusion: Total cases shows a strong correlation with population, and little correlation with distance to Mount Partchard, Roxbargh Park or CBD

- If your living area has a population, and close to above areas, better to stay at home to keep safe
- New Cases in NSW and VIC continues and doesnt show decreasing trend.
- After comparing OLS model between Lockdown and Eased, the effectiveness of lockdown is proved.