Project_Pygmalion

May 7, 2021

[1]:
```python
import matplotlib.pyplot as plt
import requests
import pandas as pd
import json
import scipy.stats as st
import plotly.express as px
```

[2]:
```
!pip install -U plotly
```

```
Requirement already up-to-date: plotly in c:\users\robmir\anaconda3\lib\site-
packages (4.14.3)
Requirement already satisfied, skipping upgrade: retrying>=1.3.3 in
c:\users\robmir\anaconda3\lib\site-packages (from plotly) (1.3.3)
Requirement already satisfied, skipping upgrade: six in
c:\users\robmir\anaconda3\lib\site-packages (from plotly) (1.15.0)
```

1 Safe Travels

1.1 "Thriving against the odds"

1.1.1 Where should Mexico puts its very limited economic efforts to increase tourism (in 2021 and 2022)?

[3]:
```python
# Covid Data API import
url = "https://api.apify.com/v2/key-value-stores/vpfkeiYLXPIDIea2T/records/
 ↪LATEST?disableRedirect=true"
```

[4]:
```python
response = requests.get(url).json()
print(json.dumps(response, indent=4, sort_keys=True))
```

```
{
    "README": "https://apify.com/puorc/mexico-covid19?utm_source=app",
    "State": {
        "Aguascalientes": {
            "deceased": 676,
            "infected": 7753
        },
        "Baja California": {
            "deceased": 3680,
        "infected": 22137
    },
    "Baja California Sur": {
        "deceased": 537,
        "infected": 10925
    },
    "Campeche": {
        "deceased": 840,
        "infected": 6235
    },
    "Chiapas": {
        "deceased": 1323,
        "infected": 8079
    },
    "Chihuahua": {
        "deceased": 1483,
        "infected": 12753
    },
    "Ciudad de Mexico": {
        "deceased": 10730,
        "infected": 138329
    },
    "Coahuila": {
        "deceased": 1996,
        "infected": 28317
    },
    "Colima": {
        "deceased": 579,
        "infected": 5671
    },
    "Durango": {
        "deceased": 689,
        "infected": 9844
    },
    "Estado de Mexico": {
        "deceased": 13007,
        "infected": 88619
    },
    "Guanajuato": {
        "deceased": 3099,
        "infected": 43054
    },
    "Guerrero": {
        "deceased": 2057,
        "infected": 20295
    },
    "Hidalgo": {
        "deceased": 2061,
            "infected": 13844
        },
        "Jalisco": {
            "deceased": 3509,
            "infected": 29252
        },
        "Michoacan": {
            "deceased": 1827,
            "infected": 21927
        },
        "Morelos": {
            "deceased": 1125,
            "infected": 6283
        },
        "Nayarit": {
            "deceased": 782,
            "infected": 6247
        },
        "Nuevo Leon": {
            "deceased": 3306,
            "infected": 43667
        },
        "Oaxaca": {
            "deceased": 1554,
            "infected": 18694
        },
        "Puebla": {
            "deceased": 4372,
            "infected": 32922
        },
        "Queretaro": {
            "deceased": 997,
            "infected": 10086
        },
        "Quintana Roo": {
            "deceased": 1743,
            "infected": 12590
        },
        "San Luis Potosi": {
            "deceased": 1782,
            "infected": 24279
        },
        "Sinaloa": {
            "deceased": 3366,
            "infected": 19791
        },
        "Sonora": {
            "deceased": 2971,
            "infected": 35177
        },
        "Tabasco": {
            "deceased": 2893,
            "infected": 32868
        },
        "Tamaulipas": {
            "deceased": 2381,
            "infected": 30066
        },
        "Tlaxcala": {
            "deceased": 1133,
            "infected": 7820
        },
        "Veracruz": {
            "deceased": 4591,
            "infected": 34679
        },
        "Yucatan": {
            "deceased": 1635,
            "infected": 19426
        },
        "Zacatecas": {
            "deceased": 783,
            "infected": 8122
        }
    },
    "country": "Mexico",
    "deceased": 83507,
    "historyData":
"https://api.apify.com/v2/datasets/4efvuMEdxdQPCreW7/items?format=json&clean=1",
    "infected": 809751,
    "lastUpdatedAtApify": "2020-10-12T20:00:13.734Z",
    "lastUpdatedAtSource": "2020-10-09T00:00:00.000Z",
    "negative": 956251,
    "recovered": "N/A",
    "sourceUrl": "https://coronavirus.gob.mx/datos/",
    "suspected": 302645,
    "tested": "N/A"
}
```

[5]:
```
Covid_data_dic = response['State']
Covid_data_dic
```

[5]:
```
{'Ciudad de Mexico': {'infected': 138329, 'deceased': 10730},
 'Baja California Sur': {'infected': 10925, 'deceased': 537},
 'Tabasco': {'infected': 32868, 'deceased': 2893},
 'Sonora': {'infected': 35177, 'deceased': 2971},
 'Coahuila': {'infected': 28317, 'deceased': 1996},
 'Yucatan': {'infected': 19426, 'deceased': 1635},
 'San Luis Potosi': {'infected': 24279, 'deceased': 1782},
 'Tamaulipas': {'infected': 30066, 'deceased': 2381},
 'Nuevo Leon': {'infected': 43667, 'deceased': 3306},
 'Quintana Roo': {'infected': 12590, 'deceased': 1743},
 'Colima': {'infected': 5671, 'deceased': 579},
 'Guanajuato': {'infected': 43054, 'deceased': 3099},
 'Sinaloa': {'infected': 19791, 'deceased': 3366},
 'Campeche': {'infected': 6235, 'deceased': 840},
 'Baja California': {'infected': 22137, 'deceased': 3680},
 'Tlaxcala': {'infected': 7820, 'deceased': 1133},
 'Guerrero': {'infected': 20295, 'deceased': 2057},
 'Aguascalientes': {'infected': 7753, 'deceased': 676},
 'Durango': {'infected': 9844, 'deceased': 689},
 'Estado de Mexico': {'infected': 88619, 'deceased': 13007},
 'Puebla': {'infected': 32922, 'deceased': 4372},
 'Zacatecas': {'infected': 8122, 'deceased': 783},
 'Nayarit': {'infected': 6247, 'deceased': 782},
 'Michoacan': {'infected': 21927, 'deceased': 1827},
 'Oaxaca': {'infected': 18694, 'deceased': 1554},
 'Hidalgo': {'infected': 13844, 'deceased': 2061},
 'Queretaro': {'infected': 10086, 'deceased': 997},
 'Veracruz': {'infected': 34679, 'deceased': 4591},
 'Jalisco': {'infected': 29252, 'deceased': 3509},
 'Chihuahua': {'infected': 12753, 'deceased': 1483},
 'Morelos': {'infected': 6283, 'deceased': 1125},
 'Chiapas': {'infected': 8079, 'deceased': 1323}}
```

```
[6]: # Build a standard list of states for Mexico
```

```
[7]: states = ['Aguascalientes'        ,
      'Baja California'        ,
      'Baja California Sur'        ,
      'Campeche'        ,
      'Chiapas'        ,
      'Chihuahua'        ,
      'Ciudad de Mexico'        ,
      'Coahuila'        ,
      'Colima'        ,
      'Durango'        ,
      'Estado de Mexico'        ,
      'Guanajuato'        ,
      'Guerrero'        ,
      'Hidalgo'        ,
      'Jalisco'        ,
      'Michoacan'        ,
      'Morelos'        ,
      'Nayarit'        ,
      'Nuevo Leon'        ,
      'Oaxaca'        ,
      'Puebla'        ,
      'Queretaro'        ,
      'Quintana Roo'        ,
      'San Luis Potosi'        ,
      'Sinaloa'        ,
      'Sonora'        ,
      'Tabasco'        ,
      'Tamaulipas'        ,
      'Tlaxcala'        ,
      'Veracruz'        ,
      'Yucatan'        ,
      'Zacatecas'        ]
```

[8]:

```
infected = []
```

[9]:

```
for x in states:
    infected.append(Covid_data_dic[x]['infected'])
```

[10]:

```
infected
```

[10]:

```
[7753,
 22137,
 10925,
 6235,
 8079,
 12753,
 138329,
 28317,
 5671,
 9844,
 88619,
 43054,
 20295,
 13844,
 29252,
 21927,
 6283,
 6247,
 43667,
 18694,
 32922,
 10086,
 12590,
 24279,
 19791,
 35177,
 32868,
 30066,
 7820,
 34679,
 19426,
 8122]
```

[11]:

```
data = {"State": states,"Covid Cases":infected}


covid_df = pd.DataFrame(data,columns=['State',  'Covid Cases'])
covid_df.head()

# Covid confirmed cases (Oct 2020) per state
```

[11]:

```
                 State  Covid Cases
0       Aguascalientes         7753
1      Baja California        22137
2  Baja California Sur        10925
3             Campeche         6235
4              Chiapas         8079
```

[12]:

```
covid_df.to_csv("clean_data/covid.csv", index=False)
```

[13]:

```
print(f"COVID19 infected data has been updated succesfully")
```

```
COVID19 infected data has been updated succesfully
```

2 Who wants to take vacations on a place filled with Covid and Crime? Nobody!

[16]:

```
# Import other dataframes and clean them: Population (to normalize data and␣
 ↪make it comparable), Number of crimes
# and Number of tourists.
# 1. Population per state data
# https://www.inegi.org.mx/app/tabulados/interactivos/?
 ↪pxq=Poblacion_Poblacion_01_e60cd8cf-927f-4b94-823e-972457a12d4b
```

[17]:

```
inegi = "raw_data/INEGI_Censo_Población_Vivienda_2020.csv"
census = pd.read_csv(inegi)
```

[18]:
```
census.head(15)
#Remove Estados Unidos Mexicanos, keep only 2020 data and stay only with Total␣
 ↪data (from age group)
```

[18]:
```
          Entidad federativa Grupo quinqenal de edad      1990      1995  \
0   Estados Unidos Mexicanos                   Total  81249645  91158290
1   Estados Unidos Mexicanos              0 a 4 años  10195178  10724100
2   Estados Unidos Mexicanos              5 a 9 años  10562234  10867563
3   Estados Unidos Mexicanos            10 a 14 años  10389092  10670048
4   Estados Unidos Mexicanos            15 a 19 años   9664403  10142071
5   Estados Unidos Mexicanos            20 a 24 años   7829163   9397424
6   Estados Unidos Mexicanos            25 a 29 años   6404512   7613090
7   Estados Unidos Mexicanos            30 a 34 años   5387619   6564605
8   Estados Unidos Mexicanos            35 a 39 años   4579116   5820178
9   Estados Unidos Mexicanos            40 a 44 años   3497770   4434317
10  Estados Unidos Mexicanos            45 a 49 años   2971860   3612452
11  Estados Unidos Mexicanos            50 a 54 años   2393791   2896049
12  Estados Unidos Mexicanos            55 a 59 años   1894484   2231897
13  Estados Unidos Mexicanos            60 a 64 años   1611317   1941953
14  Estados Unidos Mexicanos            65 a 69 años   1183651   1425809

        2000       2005       2010       2020
0   97483412  103263388  112336538  126014024
1   10635157   10186243   10528322   10047365
2   11215323   10511738   11047537   10764379
3   10736493   10952123   10939937   10943540
4    9992135   10109021   11026112   10806690
5    9071134    8964629    9892271   10422095
6    8157743    8103358    8788177    9993001
7    7136523    7933951    8470798    9420827
8    6352538    7112526    8292987    9020276
9    5194833    6017268    7009226    8503586
10   4072091    5015255    5928730    7942413
11   3357953    4090650    5064291    7037532
12   2559231    3117071    3895365    5695958
13   2198146    2622476    3116466    4821062
14   1660785    1958069    2317265    3645077
```

[19]:
```
census.count()
```

[19]:
```
Entidad federativa         759
Grupo quinquenal de edad   759
1990                       759
1995                       759
2000                       759
2005                       759
2010                       759
```

```
2020                            759
dtype: int64
```

[20]:
```
census.dtypes
```

[20]:
```
Entidad federativa          object
Grupo quinquenal de edad    object
1990                         int64
1995                         int64
2000                         int64
2005                         int64
2010                         int64
2020                         int64
dtype: object
```

[21]:
```
del census['1990']
del census['1995']
del census['2000']
del census['2005']
del census['2010']
```

[22]:
```
census.describe()
```

[22]:
```
               2020
count  7.590000e+02
mean   6.641055e+05
std    4.808992e+06
min    5.500000e+01
25%    2.965800e+04
50%    1.257660e+05
75%    2.778645e+05
max    1.260140e+08
```

[23]:
```
census.head()
```

[23]:
```
         Entidad federativa Grupo quinquenal de edad       2020
0  Estados Unidos Mexicanos                    Total  126014024
1  Estados Unidos Mexicanos               0 a 4 años   10047365
2  Estados Unidos Mexicanos               5 a 9 años   10764379
3  Estados Unidos Mexicanos             10 a 14 años   10943540
4  Estados Unidos Mexicanos             15 a 19 años   10806690
```

[24]:
```
census_new = census.loc[census["Grupo quinquenal de edad"] == "Total"]
census_new.head(25)
```

[24]:
```
           Entidad federativa Grupo quinquenal de edad       2020
0    Estados Unidos Mexicanos                    Total  126014024
23             Aguascalientes                    Total    1425607
```

```
46            Baja California                    Total    3769020
69        Baja California Sur                    Total     798447
92                   Campeche                    Total     928363
115                  Coahuila                    Total    3146771
138                    Colima                    Total     731391
161                   Chiapas                    Total    5543828
184                 Chihuahua                    Total    3741869
207          Ciudad de Mexico                    Total    9209944
230                   Durango                    Total    1832650
253                Guanajuato                    Total    6166934
276                  Guerrero                    Total    3540685
299                   Hidalgo                    Total    3082841
322                   Jalisco                    Total    8348151
345          Estado de Mexico                    Total   16992418
368                 Michoacan                    Total    4748846
391                   Morelos                    Total    1971520
414                   Nayarit                    Total    1235456
437                Nuevo Leon                    Total    5784442
460                    Oaxaca                    Total    4132148
483                    Puebla                    Total    6583278
506                 Queretaro                    Total    2368467
529              Quintana Roo                    Total    1857985
552           San Luis Potosi                    Total    2822255
```

[25]:

```
census_final = census_new.loc[census_new["Entidad federativa"] != "Estados␣
 ↪Unidos Mexicanos", ["Entidad federativa", "2020"] ]
census_final.head()
```

[25]:

```
      Entidad federativa     2020
23        Aguascalientes  1425607
46       Baja California  3769020
69   Baja California Sur   798447
92              Campeche   928363
115             Coahuila  3146771
```

[26]:

```
census_export = census_final.rename(columns={"Entidad federativa": "State",
                                                "2020": "Total"})
```

[27]:

```
census_sorted = census_export.sort_values(["State"], ascending=True )
census_sorted['Total'] = pd.to_numeric(census_sorted['Total'])
census_sorted.to_csv("clean_data/poblacion.csv", index=False)
```

[28]:

```
# 2. Crime per state data
# https://www.gob.mx/sesnsp/acciones-y-programas/
 ↪datos-abiertos-de-incidencia-delictiva
```

```
[29]: gobfed = "raw_data/Gobierno_Federal_Incidencia_Delictiva.csv"
      crime_df = pd.read_csv(gobfed)
```

```
[30]: crime_df.head()
```

```
[30]:     Año  Clave_Ent         Entidad            Bien jurídico afectado  \
      0  2019          1  Aguascalientes  La vida y la Integridad corporal
      1  2019          1  Aguascalientes  La vida y la Integridad corporal
      2  2019          1  Aguascalientes  La vida y la Integridad corporal
      3  2019          1  Aguascalientes  La vida y la Integridad corporal
      4  2019          1  Aguascalientes  La vida y la Integridad corporal

        Tipo de delito  Subtipo de delito          Modalidad  Enero  Febrero  Marzo  \
      0      Homicidio   Homicidio doloso  Con arma de fuego      7        4      6
      1      Homicidio   Homicidio doloso    Con arma blanca      1        1      1
      2      Homicidio   Homicidio doloso  Con otro elemento      1        2      2
      3      Homicidio   Homicidio doloso    No especificado      0        0      0
      4      Homicidio  Homicidio culposo  Con arma de fuego      0        0      0

         Abril  Mayo  Junio  Julio  Agosto  Septiembre  Octubre  Noviembre  \
      0      2     2      5      3       1          11       10          3
      1      2     4      0      2       0           0        1          3
      2      2     2      1      0       1           0        0          1
      3      0     2      0      0       0           0        0          0
      4      0     0      1      0       1           0        0          0

         Diciembre  Total
      0          2     15
      1          4      8
      2          2      3
      3          0      0
      4          0      0
```

```
[31]: crime_df.count()
```

```
[31]: Año                         3520
      Clave_Ent                   3520
      Entidad                     3520
      Bien jurídico afectado      3520
      Tipo de delito              3520
      Subtipo de delito           3520
      Modalidad                   3520
      Enero                       3520
      Febrero                     3520
      Marzo                       3520
      Abril                       3520
      Mayo                        3520
```

```
Junio                         3520
Julio                         3520
Agosto                        3520
Septiembre                    3520
Octubre                       3520
Noviembre                     3520
Diciembre                     3520
Total                         3520
dtype: int64
```

[32]:

```
del crime_df['Total']
del crime_df['Modalidad']
del crime_df['Subtipo de delito']
del crime_df['Tipo de delito']
del crime_df['Bien jurídico afectado']
del crime_df['Clave_Ent']
```

[33]:

```
crime_df.dtypes
```

[33]:

```
Año              int64
Entidad         object
Enero            int64
Febrero          int64
Marzo            int64
Abril            int64
Mayo             int64
Junio            int64
Julio            int64
Agosto           int64
Septiembre       int64
Octubre          int64
Noviembre        int64
Diciembre        int64
dtype: object
```

[34]:

```
# COVID latest data is from OCT 2020 so we want to use crime data corresponding␣
 ↪to Jan-Sep 2020 and Oct-Dec 2019 (Rolling year)
```

[35]:

```
df2019 = crime_df.loc[crime_df["Año"] == 2019 ]
```

[36]:

```
df2019.head()
```

[36]:

```
    Año         Entidad  Enero  Febrero  Marzo  Abril  Mayo  Junio  Julio  \
0  2019  Aguascalientes      7        4      6      2     2      5      3
1  2019  Aguascalientes      1        1      1      2     4      0      2
2  2019  Aguascalientes      1        2      2      2     2      1      0
3  2019  Aguascalientes      0        0      0      0     2      0      0
```

```
4  2019  Aguascalientes      0        0      0      0     0      1      0

   Agosto  Septiembre  Octubre  Noviembre  Diciembre
0       1          11       10          3          2
1       0           0        1          3          4
2       1           0        0          1          2
3       0           0        0          0          0
4       1           0        0          0          0
```

[37]:
```
del df2019['Enero']
del df2019['Febrero']
del df2019['Marzo']
del df2019['Abril']
del df2019['Mayo']
del df2019['Junio']
del df2019['Julio']
del df2019['Agosto']
del df2019['Septiembre']
del df2019['Año']
```

[38]:
```
df2019.head()
```

[38]:
```
          Entidad  Octubre  Noviembre  Diciembre
0  Aguascalientes       10          3          2
1  Aguascalientes        1          3          4
2  Aguascalientes        0          1          2
3  Aguascalientes        0          0          0
4  Aguascalientes        0          0          0
```

[39]:
```
df_2019 = df2019.groupby(["Entidad"])
crimes_2019 = df_2019.sum()
new2019 = crimes_2019.reset_index()
```

[40]:
```
new2019.head()
```

[40]:
```
               Entidad  Octubre  Noviembre  Diciembre
0       Aguascalientes     1117        958        967
1      Baja California     3890       3485       3441
2  Baja California Sur      757        679        616
3             Campeche       73         78         73
4              Chiapas      591        567        572
```

[41]:
```
df2020 = crime_df.loc[crime_df["Año"] == 2020 ]
del df2020['Octubre']
del df2020['Noviembre']
del df2020['Diciembre']
del df2020['Año']
```

```
df_2020 = df2020.groupby(["Entidad"])
crimes_2020 = df_2020.sum()
new2020 = crimes_2020.reset_index()
```

[42]:
```
new2020.head()
```

[42]:
```
               Entidad  Enero  Febrero  Marzo  Abril  Mayo  Junio  Julio  \
0       Aguascalientes   1207     1209   1304    848   945   1064   1048
1      Baja California   3363     3417   3525   2482  2621   2954   3253
2  Baja California Sur    708      699    718    353   412    594    642
3             Campeche     77       86     89     41    54     52     55
4              Chiapas    634      614    664    403   374    368    580

   Agosto  Septiembre
0     866         867
1    3279        3091
2     564         614
3      73          76
4     537         578
```

[43]:
```
crime_df_final = pd.merge(new2020, new2019, on="Entidad")
```

[44]:
```
crime_df_final.head()
```

[44]:
```
               Entidad  Enero  Febrero  Marzo  Abril  Mayo  Junio  Julio  \
0       Aguascalientes   1207     1209   1304    848   945   1064   1048
1      Baja California   3363     3417   3525   2482  2621   2954   3253
2  Baja California Sur    708      699    718    353   412    594    642
3             Campeche     77       86     89     41    54     52     55
4              Chiapas    634      614    664    403   374    368    580

   Agosto  Septiembre  Octubre  Noviembre  Diciembre
0     866         867     1117        958        967
1    3279        3091     3890       3485       3441
2     564         614      757        679        616
3      73          76       73         78         73
4     537         578      591        567        572
```

[45]:
```
crime_df_final["Total"] = crime_df_final.sum(axis=1)
crime_df_export = crime_df_final[["Entidad", "Total"]]

crime_renamed_df = crime_df_export.rename(columns={"Entidad": "State",
                                                   "Total": "Crimes"})
crime_renamed_df.head()
```

[45]:
```
                 State  Crimes
0       Aguascalientes   12400
```

```
1      Baja California   38801
2  Baja California Sur    7356
3             Campeche     827
4              Chiapas    6482
```

[46]:

```
crime_sorted = crime_renamed_df.sort_values(["State"], ascending=True )
crime_sorted.to_csv("clean_data/crimenes.csv", index=False)
```

[47]:

```
# 3. Tourist per state data
# Source INEGI - Webchat
# http://www.datatur.sectur.gob.mx/SitePages/InfTurxEdo.aspx
```

[48]:

```
tourist = "raw_data/INEGI_Tourist_data.csv"
tourist_df = pd.read_csv(tourist)
```

[49]:

```
tourist_df.head()
```

[49]:

```
                 State  Number of Tourists
0       Aguascalientes            856960.0
1      Baja California           3958843.0
2  Baja California Sur           3445908.0
3             Campeche           1578131.0
4              Chiapas           4376440.0
```

[50]:

```
tourist_df.dtypes
```

[50]:

```
State                  object
Number of Tourists    float64
dtype: object
```

[51]:

```
tourist_sorted = tourist_df.sort_values(["State"], ascending=True )
tourist_sorted['Number of Tourists'] = pd.to_numeric(tourist_sorted['Number of␣
 ↪Tourists'])

tourist_sorted.to_csv("clean_data/Tourist.csv", index=False)
```

[52]:

```
## Getting all data in the same dataframe
```

[53]:

```
file1 = "clean_data/poblacion.csv"
file2 = "clean_data/tourist.csv"
file3 = "clean_data/crimenes.csv"
```

[54]:

```
poblacion_df = pd.read_csv(file1)
tourist_df = pd.read_csv(file2)
crimes_df = pd.read_csv(file3)
```

[55]:

```
poblacion_df.head()
```

```
[55]:                 State    Total
      0       Aguascalientes  1425607
      1      Baja California  3769020
      2  Baja California Sur   798447
      3             Campeche   928363
      4              Chiapas  5543828
```

```
[56]: tourist_df.head()
```

```
[56]:                 State  Number of Tourists
      0       Aguascalientes            856960.0
      1      Baja California           3958843.0
      2  Baja California Sur           3445908.0
      3             Campeche           1578131.0
      4              Chiapas           4376440.0
```

```
[57]: crimes_df.head()
```

```
[57]:                 State  Crimes
      0       Aguascalientes   12400
      1      Baja California   38801
      2  Baja California Sur    7356
      3             Campeche     827
      4              Chiapas    6482
```

```
[58]: covid_df.head()
```

```
[58]:                 State  Covid Cases
      0       Aguascalientes         7753
      1      Baja California        22137
      2  Baja California Sur        10925
      3             Campeche         6235
      4              Chiapas         8079
```

```
[59]: merge_df = pd.merge(covid_df, poblacion_df, on="State")
```

```
[60]: merge_df.head()
```

```
[60]:                 State  Covid Cases    Total
      0       Aguascalientes         7753  1425607
      1      Baja California        22137  3769020
      2  Baja California Sur        10925   798447
      3             Campeche         6235   928363
      4              Chiapas         8079  5543828
```

```
[61]: final_df = pd.merge(merge_df, tourist_df, on="State")
      final_df.head()
```

```
[61]:                  State  Covid Cases    Total  Number of Tourists
      0       Aguascalientes         7753  1425607            856960.0
      1      Baja California        22137  3769020           3958843.0
      2  Baja California Sur        10925   798447           3445908.0
      3             Campeche         6235   928363           1578131.0
      4              Chiapas         8079  5543828           4376440.0
```

```
[62]: final_final_df = pd.merge(final_df, crimes_df, on="State")
      final_final_df.head()
```

```
[62]:                  State  Covid Cases    Total  Number of Tourists  Crimes
      0       Aguascalientes         7753  1425607            856960.0   12400
      1      Baja California        22137  3769020           3958843.0   38801
      2  Baja California Sur        10925   798447           3445908.0    7356
      3             Campeche         6235   928363           1578131.0     827
      4              Chiapas         8079  5543828           4376440.0    6482
```

```
[63]: renamed_df = final_final_df.rename(columns={"Covid Cases": "Covid Cases",
                                                  "Total": "Population",
                                                  "Number of Tourists":␣
       ↪"Tourists",

                                                  "Crimes": "Crimes"
                                                  })
      renamed_df
```

```
[63]:                  State  Covid Cases  Population     Tourists  Crimes
      0       Aguascalientes         7753     1425607     856960.0   12400
      1      Baja California        22137     3769020    3958843.0   38801
      2  Baja California Sur        10925      798447    3445908.0    7356
      3             Campeche         6235      928363    1578131.0     827
      4              Chiapas         8079     5543828    4376440.0    6482
      5            Chihuahua        12753     3741869    5228183.0   25089
      6     Ciudad de Mexico       138329     9209944   11331505.0   72500
      7             Coahuila        28317     3146771    1956640.0   19368
      8               Colima         5671      731391    1450627.0    7881
      9              Durango         9844     1832650     829529.0    9696
      10    Estado de Mexico        88619    16992418    3127227.0  147717
      11          Guanajuato        43054     6166934    5026515.0   54610
      12            Guerrero        20295     3540685    9065181.0    9692
      13             Hidalgo        13844     3082841    2925426.0   15484
      14             Jalisco        29252     8348151    9499223.0   44860
      15           Michoacan        21927     4748846    3005225.0   20195
      16             Morelos         6283     1971520    1659199.0   14566
      17             Nayarit         6247     1235456    3073656.0    1220
      18          Nuevo Leon        43667     5784442    3222964.0   25930
      19              Oaxaca        18694     4132148    3666038.0   15474
      20              Puebla        32922     6583278    6608202.0   20145
```

```
21           Queretaro        10086     2368467    2520716.0    24457
22         Quintana Roo       12590     1857985   16675407.0    15369
23      San Luis Potosi       24279     2822255    2132770.0    15504
24              Sinaloa       19791     3026943    5271130.0     9194
25               Sonora       35177     2944840    2671758.0    12693
26              Tabasco       32868     2402598    1408949.0    15546
27           Tamaulipas       30066     3527735    3743766.0    11018
28             Tlaxcala        7820     1342977     458161.0     1055
29             Veracruz       34679     8062579    5332441.0    26453
30              Yucatan       19426     2320898    2617911.0     1207
31            Zacatecas        8122     1622138    1325235.0     9806
```

[64]:

```python
Covid_rate = renamed_df["Covid Cases"] / renamed_df["Population"]
States = renamed_df["State"]
Tourist_rate = renamed_df["Tourists"] / renamed_df["Population"]
Crime_rate = renamed_df["Crimes"] / renamed_df["Population"]

ratio_df = pd.DataFrame({"State": States,
                              "Covid Rate":Covid_rate,
                              "Tourist Rate": Tourist_rate,
                              "Crime Rate": Crime_rate})
ratio_df.head()
```

[64]:

```
                 State  Covid Rate  Tourist Rate  Crime Rate
0       Aguascalientes    0.005438      0.601119    0.008698
1      Baja California    0.005873      1.050364    0.010295
2  Baja California Sur    0.013683      4.315763    0.009213
3             Campeche    0.006716      1.699907    0.000891
4              Chiapas    0.001457      0.789426    0.001169
```

[65]:

```python
# Using data for Covid, tourists and Crime divided by the population in each state allows us to have
# comparable data to avoid arriving at obvious/wrong conclusions. i.e. Mexico City and Mexico State will have
# the most number of crimes and Covid cases just beacuse they have the largest populations.
```

[66]:

```python
renamed_df["Tourists"]= renamed_df["Tourists"].astype(int)
renamed_df.dtypes
```

[66]:

```
State           object
Covid Cases      int64
Population       int64
Tourists         int32
Crimes           int64
dtype: object
```

3 Summary statistics

[67]:
```
#We calculated the mean,median,variance, standard_dv and sem for each Covid␣
 ↪Case in the Mexico.
#We then created a dataframe with all the information.

mean_covid = renamed_df['Covid Cases'].mean()
median = renamed_df['Covid Cases'].median()
variance = renamed_df['Covid Cases'].var()
standard_dv = renamed_df['Covid Cases'].std()
sem = renamed_df['Covid Cases'].sem()

summary_stats_covid = pd.DataFrame({"Mean": mean_covid, "Median": median,␣
 ↪"Variance": variance, "Standard Deviation": standard_dv, "SEM":␣
 ↪sem},index=[0])
summary_stats_covid
```

[67]:
```
          Mean   Median      Variance  Standard Deviation          SEM
0  25304.71875  19608.5  7.015299e+08        26486.410258  4682.180076
```

[68]:
```
#Verifying the results with another method
summary_stats2 = renamed_df.agg(['mean','median','var','std','sem'])["Covid␣
 ↪Cases"]
summary_stats2
```

[68]:
```
mean      2.530472e+04
median    1.960850e+04
var       7.015299e+08
std       2.648641e+04
sem       4.682180e+03
Name: Covid Cases, dtype: float64
```

[69]:
```
#We calculated the mean,median,variance, standard_dv and sem for Population in␣
 ↪the Mexico.
#We then created a dataframe with all the information.

mean = renamed_df['Population'].mean()
median = renamed_df['Population'].median()
variance = renamed_df['Population'].var()
standard_dv = renamed_df['Population'].std()
sem = renamed_df['Population'].sem()

summary_stats_covid = pd.DataFrame({"Mean": mean, "Median": median, "Variance":␣
 ↪variance, "Standard Deviation": standard_dv, "SEM": sem},index=[0])
summary_stats_covid
```

```
[69]:          Mean     Median      Variance  Standard Deviation            SEM
      0  3937938.25  3054892.0  1.074534e+13        3.278009e+06  579475.614521
```

```
[70]: #Verifying the results with another method
      summary_stats2 = renamed_df.
       ↪agg(['mean','median','var','std','sem'])["Population"]
      summary_stats2
```

```
[70]: mean      3.937938e+06
      median    3.054892e+06
      var       1.074534e+13
      std       3.278009e+06
      sem       5.794756e+05
      Name: Population, dtype: float64
```

```
[71]: #We calculated the mean,median,variance, standard_dv and sem for Tourists in␣
       ↪the Mexico.
      #We then created a dataframe with all the information.
      mean_tourists = renamed_df['Tourists'].mean()
      median = renamed_df['Tourists'].median()
      variance = renamed_df['Tourists'].var()
      standard_dv = renamed_df['Tourists'].std()
      sem = renamed_df['Tourists'].sem()

      summary_stats_covid = pd.DataFrame({"Mean": mean, "Median": median, "Variance":␣
       ↪variance, "Standard Deviation": standard_dv, "SEM": sem},index=[0])
      summary_stats_covid
```

```
[71]:          Mean     Median      Variance  Standard Deviation           SEM
      0  3937938.25  3100441.5  1.179505e+13        3.434392e+06  607120.43243
```

```
[72]: #Verifying the results with another method
      summary_stats2 = renamed_df.agg(['mean','median','var','std','sem'])["Tourists"]
      summary_stats2
```

```
[72]: mean      4.064058e+06
      median    3.100442e+06
      var       1.179505e+13
      std       3.434392e+06
      sem       6.071204e+05
      Name: Tourists, dtype: float64
```

```
[73]: #We calculated the mean,median,variance, standard_dv and sem for Crimes in the␣
       ↪Mexico.
      #We then created a dataframe with all the information.
      mean = renamed_df['Crimes'].mean()
      median = renamed_df['Crimes'].median()
```

```
variance = renamed_df['Crimes'].var()
standard_dv = renamed_df['Crimes'].std()
sem = renamed_df['Crimes'].sem()

summary_stats_covid = pd.DataFrame({"Mean": mean, "Median": median, "Variance":␣
 ↪variance, "Standard Deviation": standard_dv, "SEM": sem},index=[0])
summary_stats_covid
```

[73]:

```
          Mean   Median      Variance  Standard Deviation        SEM
0  22268.59375  15421.5  7.701380e+08        27751.360507  4905.7938
```

[74]:

```
summary_stats2 = renamed_df.agg(['mean','median','var','std','sem'])["Crimes"]
summary_stats2
```

[74]:

```
mean      2.226859e+04
median    1.542150e+04
var       7.701380e+08
std       2.775136e+04
sem       4.905794e+03
Name: Crimes, dtype: float64
```

[75]:

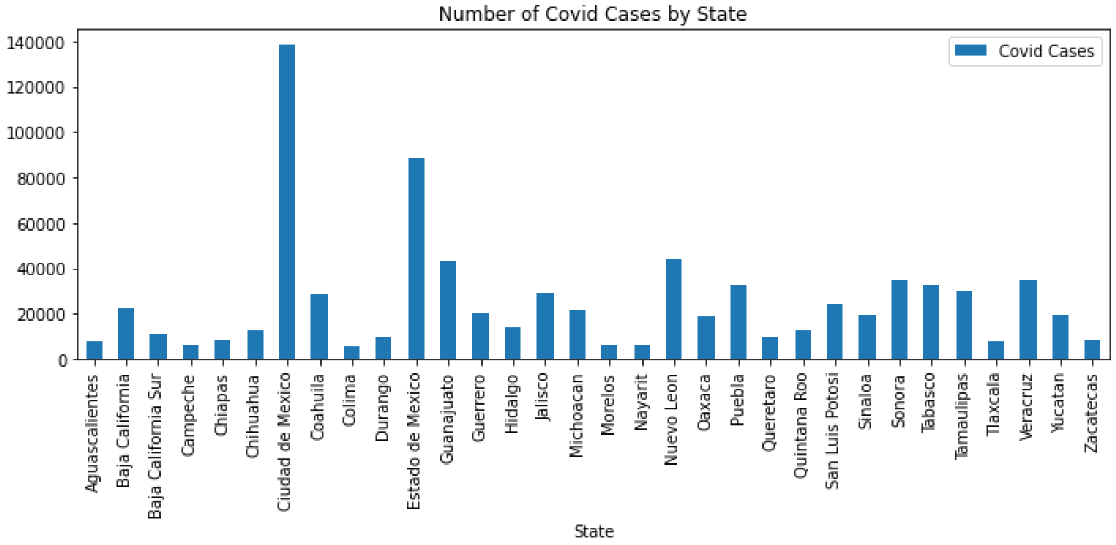
```
#Error en merge
unique_items=len(renamed_df["Tourists"].unique())
unique_items
```

[75]:

```
32
```

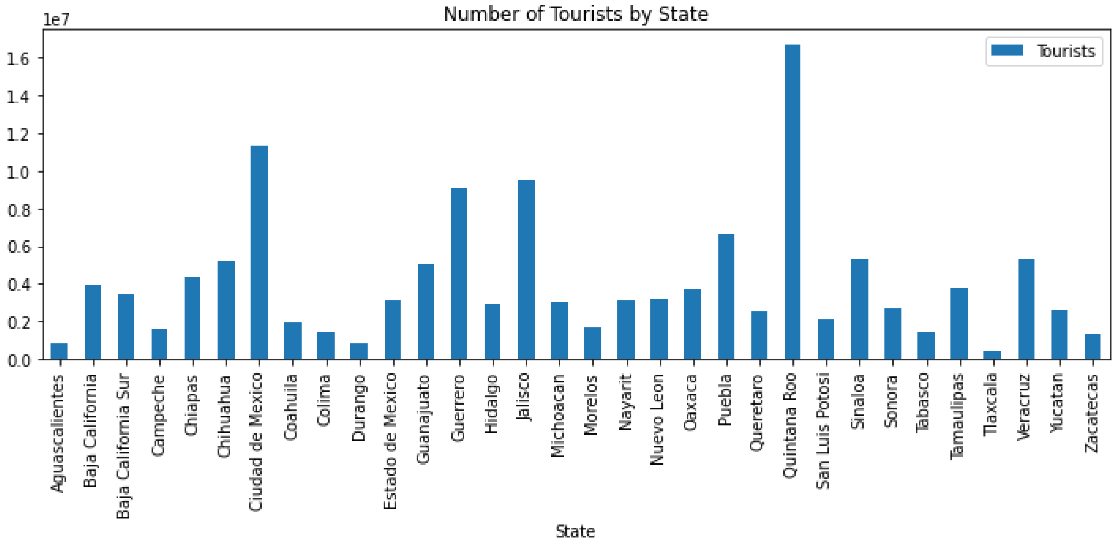
[76]:

```
# Generate a bar plot showing the total number of Covid Cases by State
bar_data = pd.DataFrame(renamed_df.groupby(["State"]).sum()).reset_index()
bar_data
# #Barframe into two columns
bar_data = bar_data [["State", "Covid Cases"]]
bar_data = bar_data .set_index("State")

#Creating the bar chart
bar_data.plot(kind="bar", figsize=(10,5))

plt.title("Number of Covid Cases by State")
plt.tight_layout()
plt.savefig("images/Number of Covid Cases by State.png")
plt.show()
```



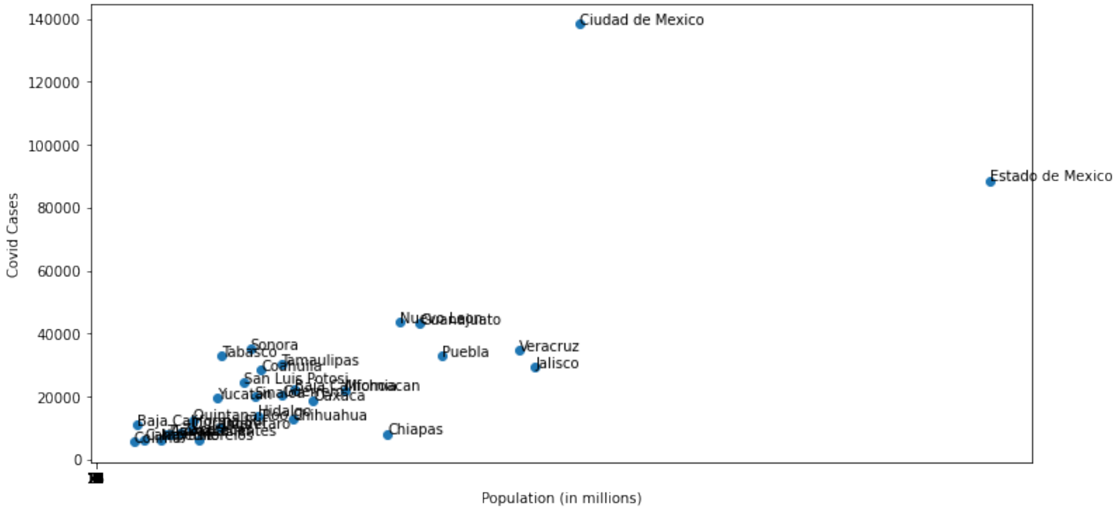
[77]:

```
# Generate a bar plot showing the total number of tourists by State
bar_data = pd.DataFrame(renamed_df.groupby(["State"]).sum()).reset_index()
bar_data
# #Barframe into two columns
bar_data = bar_data [["State", "Tourists"]]
bar_data = bar_data .set_index("State")

#Creating the bar chart
bar_data.plot(kind="bar", figsize=(10,5))

plt.title("Number of Tourists by State")
plt.tight_layout()
plt.savefig("images/Number of Tourists by State.png")
plt.show()
```

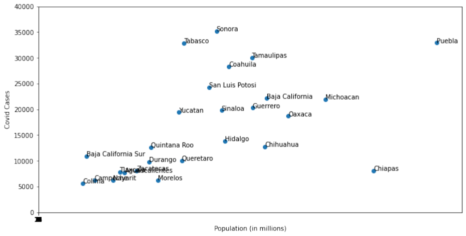


[78]:

```
# Pull values for x and y values
fig, ax = plt.subplots(figsize=(12,6))
covid_cases = renamed_df["Covid Cases"]
population = renamed_df["Population"]
n = renamed_df["State"]
# Create Scatter Plot with values calculated above
ax.scatter(population,covid_cases)
for i,txt in enumerate(n):
    ax.annotate(txt,(population[i],covid_cases[i]))
ax.set_xticks(range(len(population)))
ax.set_xlabel("Population (in millions)")
ax.set_ylabel("Covid Cases")
# Zooming on the image
# plt.xlim(0,7000000)
# plt.ylim(0, 40000)
plt.savefig("images/ScatterPlot of Covid and Population.png")
plt.show()
```

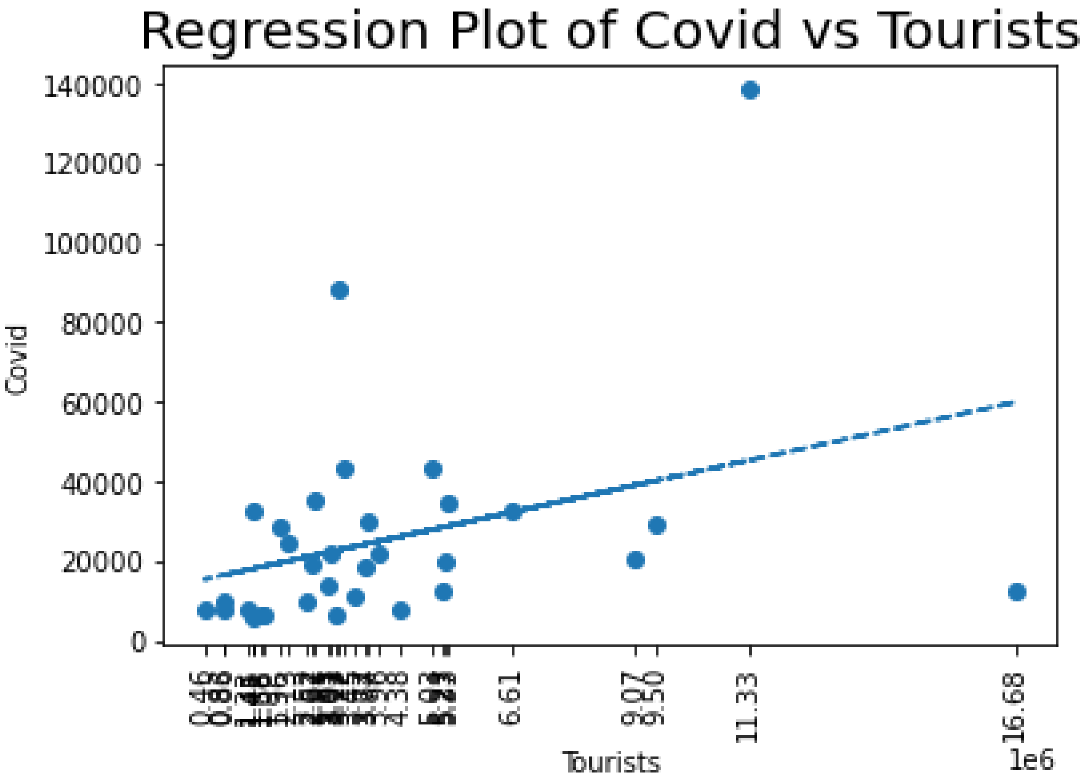


[94]:
```
## Zooming in on the graph
# Pull values for x and y values
fig, ax = plt.subplots(figsize=(12,6))
covid_cases = renamed_df["Covid Cases"]
population = renamed_df["Population"]
n = renamed_df["State"]
# Create Scatter Plot with values calculated above
ax.scatter(population,covid_cases)
for i,txt in enumerate(n):
    ax.annotate(txt,(population[i],covid_cases[i]))
ax.set_xticks(range(len(population)))
ax.set_xlabel("Population (in millions)")
ax.set_ylabel("Covid Cases")
# Zooming on the image
plt.xlim(0,7000000)
plt.ylim(0, 40000)
plt.savefig("images/ScatterPlot of Covid and Population_zoom.png")
plt.show()
```



4 Correlation and Regression

[79]:
```
# Calculate the correlation coefficient and linear regression model
#Getting our x and y values
mean_covid = renamed_df.groupby(renamed_df["State"])["Covid Cases"].mean()
mean_tourists= renamed_df.groupby(renamed_df["State"])["Tourists"].mean()
mean_covidtrim= mean_covid.loc[mean_covid.index!="Quintana Roo"]
#Independent variable is number of Tourists
#Covid cases is the dependent variable

#Performing the linear regression
slope, intercept, r, p, std_err = st.linregress(mean_tourists, mean_covid)
# Create equation of line to calculate our regression
fit = slope *mean_tourists + intercept
equation = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2))
# Plot the linear model on top of scatter plot
plt.scatter(mean_tourists,mean_covid)
plt.title('Regression Plot of Covid vs Tourists',fontsize =20)
plt.xlabel("Tourists")
plt.ylabel("Covid")
plt.plot(mean_tourists,fit,"--")
plt.xticks(mean_tourists, rotation=90)
plt.savefig("images/Regression Plot of Covid vs Tourists.png")
plt.show()

# Caculate correlation coefficient
corr = round(st.pearsonr(mean_covid,mean_tourists)[0],2)
print(f'The correlation between Covid and Tourists {corr}')

#calculate the R squared
print(f"The r-squared is: {corr**2}")

#Calculate the regression formula
print(equation)
```

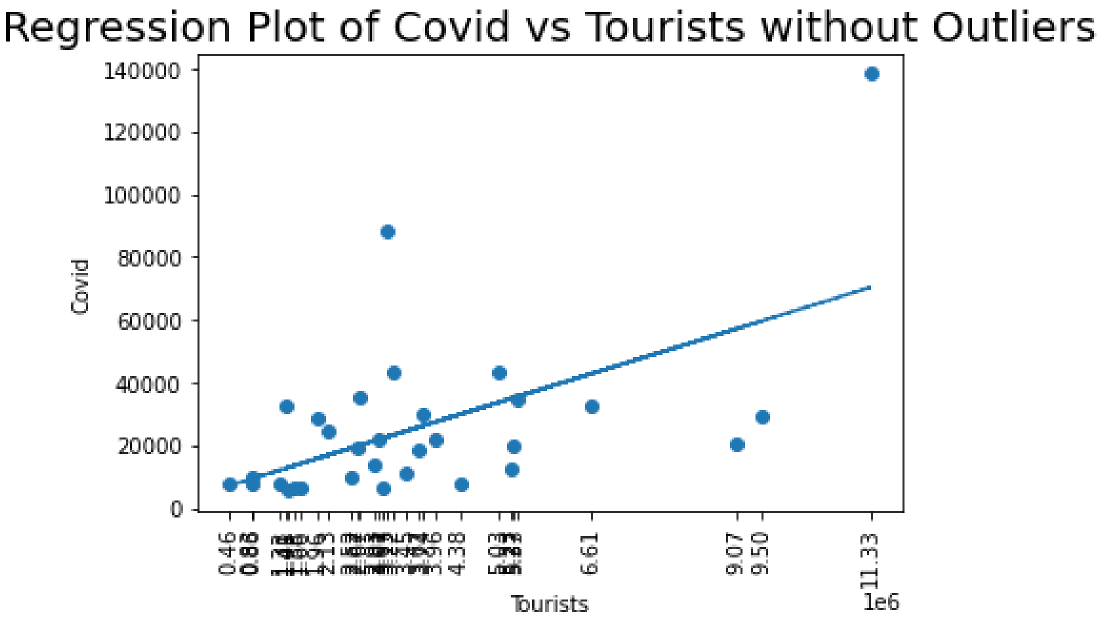


```
The correlation between Covid and Tourists 0.36
The r-squared is: 0.1296
y = 0.0x + 14135.69
```

Quintana Roo state appears to be an outlier. Removing it to see how the model changes

[80]:
```
# Calculate the correlation coefficient and linear regression model
#Getting our x and y values
mean_covid = renamed_df.groupby(renamed_df["State"])["Covid Cases"].mean()
```

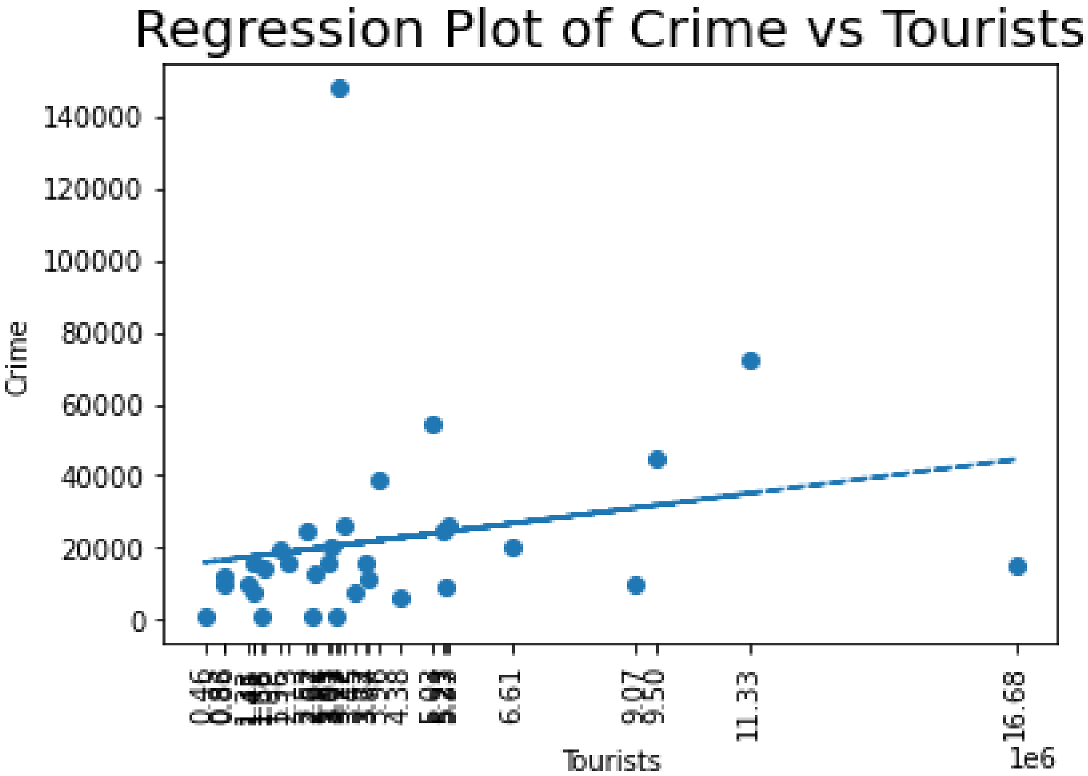
```
mean_tourists= renamed_df.groupby(renamed_df["State"])["Tourists"].mean()
mean_covidtrim= mean_covid.loc[mean_covid.index!="Quintana Roo"]
mean_touristtrim= mean_tourists.loc[mean_tourists.index!="Quintana Roo"]

#Performing the linear regression
slope, intercept, r, p, std_err = st.linregress(mean_touristtrim,␣
 ↪mean_covidtrim)
# Create equation of line to calculate our regression
fit = slope *mean_touristtrim + intercept
equation = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2))
# Plot the linear model on top of scatter plot
plt.scatter(mean_touristtrim,mean_covidtrim)
plt.title('Regression Plot of Covid vs Tourists without Outliers',fontsize =20)
plt.xlabel("Tourists")
plt.ylabel("Covid")
plt.plot(mean_touristtrim,fit,"--")
plt.xticks(mean_touristtrim, rotation=90)
plt.savefig("images/Regression Plot of Coviv vs Tourists without Outliers.png")
plt.show()

# Caculate correlation coefficient
corr = round(st.pearsonr(mean_covidtrim,mean_touristtrim)[0],2)
print(f'The correlation between Covid and Tourists {corr}')

#calculate the R squared
print(f"The r-squared is: {corr**2}")

#Calculate the regression formula
print(equation)
```



```
The correlation between Covid and Tourists 0.56
The r-squared is: 0.31360000000000005
y = 0.01x + 4468.68
```

[81]:

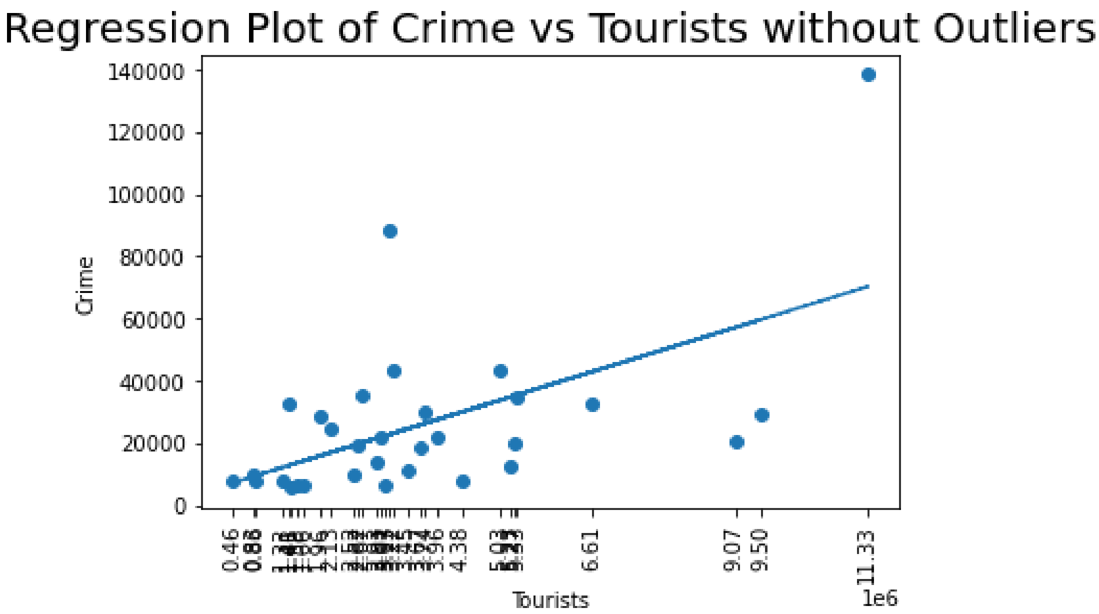
```python
# Calculate the correlation coefficient and linear regression model
#Getting our x and y values
mean_crime= renamed_df.groupby(renamed_df["State"])["Crimes"].mean()
mean_tourists= renamed_df.groupby(renamed_df["State"])["Tourists"].mean()
mean_crimestrim= mean_covid.loc[mean_covid.index!="Quintana Roo"]

# How does Crime impact Tourism?

#Performing the linear regression
slope, intercept, r, p, std_err = st.linregress(mean_tourists, mean_crime)
# Create equation of line to calculate our regression
fit = slope *mean_tourists + intercept
equation = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2))
# Plot the linear model on top of scatter plot
plt.scatter(mean_tourists,mean_crime)
plt.title('Regression Plot of Crime vs Tourists',fontsize =20)
plt.xlabel("Tourists")
plt.ylabel("Crime")
plt.plot(mean_tourists,fit,"--")
plt.xticks(mean_tourists, rotation=90)
plt.savefig("images/Regression Plot of Crime vs Tourists.png")
plt.show()

# Calculate correlation coefficient
corr = round(st.pearsonr(mean_crime,mean_tourists)[0],2)
print(f'The correlation between Crime and Tourists {corr}')

#calculate the R squared
print(f"The r-squared is: {corr**2}")

#Calculate the regression formula
print(equation)
```



```
The correlation between Crime and Tourists 0.22
The r-squared is: 0.0484
y = 0.0x + 15096.34
```

[82]:

```
# Calculate the correlation coefficient and linear regression model
#Getting our x and y values
mean_crime = renamed_df.groupby(renamed_df["State"])["Crimes"].mean()
mean_tourists= renamed_df.groupby(renamed_df["State"])["Tourists"].mean()
mean_crimetrim= mean_covid.loc[mean_crime.index!="Quintana Roo"]
mean_touristtrim= mean_tourists.loc[mean_tourists.index!="Quintana Roo"]

#Same question without Quintana Roo (Outlier)

#Performing the linear regression
slope, intercept, r, p, std_err = st.linregress(mean_touristtrim,␣
 ↪mean_crimetrim)
# Create equation of line to calculate our regression
fit = slope *mean_touristtrim + intercept
equation = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2))
# Plot the linear model on top of scatter plot
plt.scatter(mean_touristtrim,mean_crimetrim)
plt.title('Regression Plot of Crime vs Tourists without Outliers',fontsize =20)
plt.xlabel("Tourists")
plt.ylabel("Crime")
plt.plot(mean_touristtrim,fit,"--")
plt.xticks(mean_touristtrim, rotation=90)
plt.savefig("images/Regression Plot of Crime vs Tourists without Outliers.png")
plt.show()

# Caculate correlation coefficient
corr = round(st.pearsonr(mean_crimetrim,mean_touristtrim)[0],2)
print(f'The correlation between Crime and Tourists {corr}')

#calculate the R squared
print(f"The r-squared is: {corr**2}")

#Calculate the regression formula
print(equation)
```

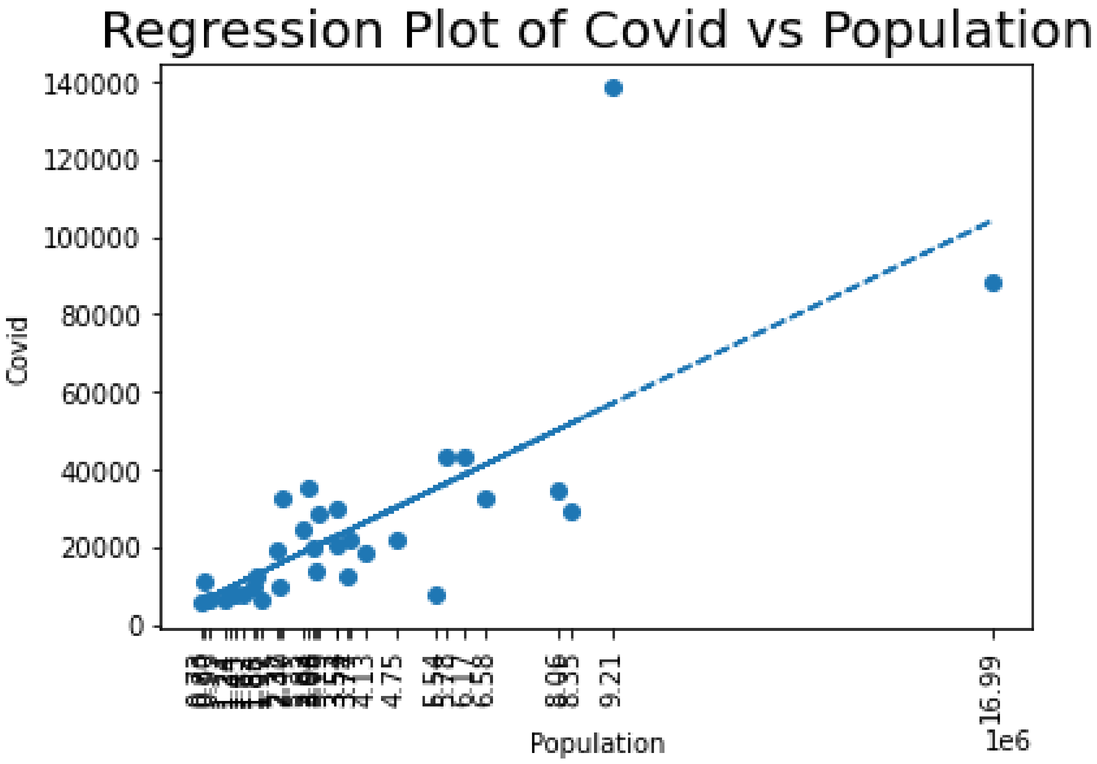


```
The correlation between Crime and Tourists 0.56
The r-squared is: 0.31360000000000005
y = 0.01x + 4468.68
```

[83]:

```
# Calculate the correlation coefficient and linear regression model
#Getting our x and y values
mean_covid = renamed_df.groupby(renamed_df["State"])["Covid Cases"].mean()
mean_population= renamed_df.groupby(renamed_df["State"])["Population"].mean()
mean_covidtrim= mean_covid.loc[mean_covid.index!="Quintana Roo"]
```

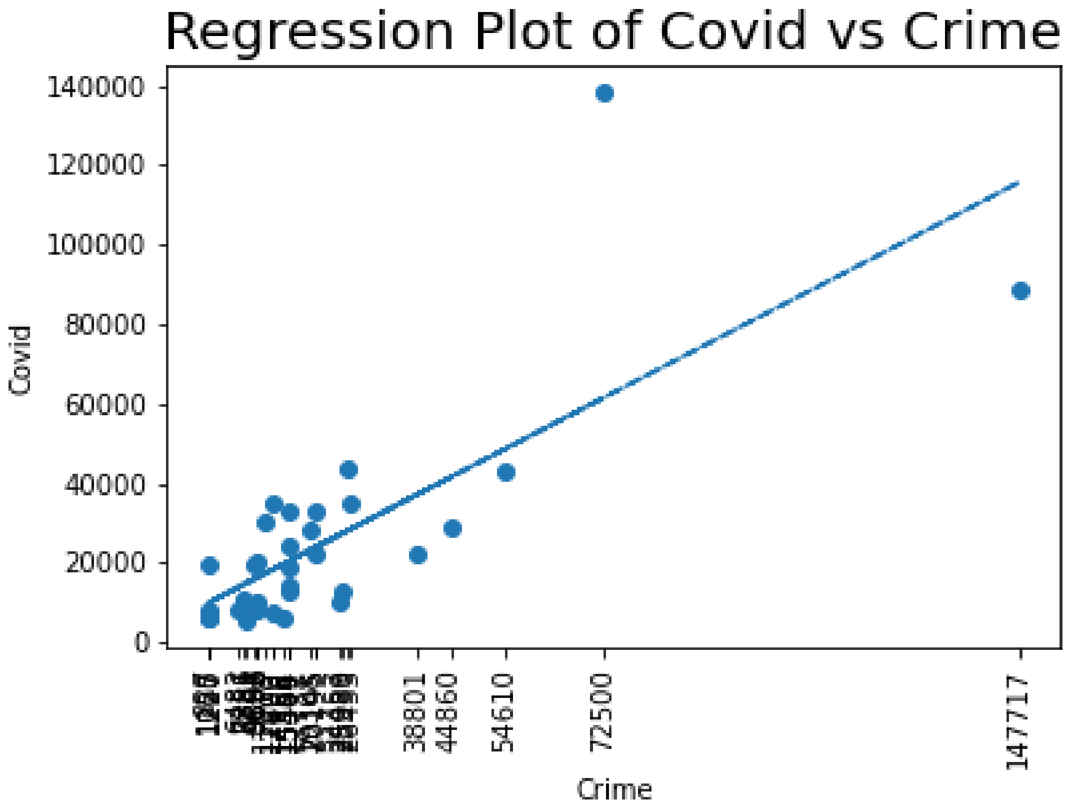
```
#Performing the linear regression
slope, intercept, r, p, std_err = st.linregress(mean_population, mean_covid)
# Create equation of line to calculate our regression
fit = slope *mean_population + intercept
equation = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2))
# Plot the linear model on top of scatter plot
plt.scatter(mean_population,mean_covid)
plt.title('Regression Plot of Covid vs Population',fontsize =20)
plt.xlabel("Population")
plt.ylabel("Covid")
plt.plot(mean_population,fit,"--")
plt.xticks(mean_population, rotation=90)
plt.savefig("images/Regression Plot of Covid vs Population.png")
plt.show()

# Caculate correlation coefficient
corr = round(st.pearsonr(mean_covid,mean_population)[0],2)
print(f'The correlation between Covid and Population {corr}')

#calculate the R squared
print(f"The r-squared is: {corr**2}")

#Calculate the regression formula
print(equation)
```



```
The correlation between Covid and Population 0.75
The r-squared is: 0.5625
y = 0.01x + 1523.15
```

[84]:

```python
# Calculate the correlation coefficient and linear regression model
#Getting our x and y values
mean_covid = renamed_df.groupby(renamed_df["State"])["Covid Cases"].mean()
mean_crime= renamed_df.groupby(renamed_df["State"])["Crimes"].mean()

#Performing the linear regression
slope, intercept, r, p, std_err = st.linregress(mean_crime, mean_covid)
# Create equation of line to calculate our regression
fit = slope *mean_crime + intercept
equation = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2))
# Plot the linear model on top of scatter plot
plt.scatter(mean_crime,mean_covid)
plt.title('Regression Plot of Covid vs Crime',fontsize =20)
plt.xlabel("Crime")
plt.ylabel("Covid")
plt.plot(mean_crime,fit,"--")
plt.xticks(mean_crime, rotation=90)
plt.savefig("images/Regression Plot of Covid vs Crime.png")
plt.show()

# Caculate correlation coefficient
corr = round(st.pearsonr(mean_covid,mean_crime)[0],2)
print(f'The correlation between Covid and Crime {corr}')

#calculate the R squared
print(f"The r-squared is: {corr**2}")

#Calculate the regression formula
print(equation)
```



```
The correlation between Covid and Crime 0.76
The r-squared is: 0.5776
y = 0.72x + 9255.85
```

4.1 Covid Rate Heat Map

4.2 Which states have the biggest ratios of Covid cases?

```
[85]: df=pd.read_csv('clean_data/covid_a.csv', encoding = "ISO-8859-1")
      repo_url = 'https://raw.githubusercontent.com/angelnmara/geojson/master/
       ↪mexicoHigh.json' #Archivo GeoJSON
      mx_regions_geo = requests.get(repo_url).json()
      fig = px.choropleth(data_frame=df,
                          geojson=mx_regions_geo,
                          locations='State', # nombre de la columna del Dataframe
                          featureidkey='properties.name',  # ruta al campo del␣
       ↪archivo GeoJSON con el que se hará la relación (nombre de los estados)
                          color="Rate" , #El color depende de las cantidades
                          color_continuous_scale="burg", #greens
                          #scope="north america"
                         )
```

```
fig.update_geos(showcountries=True, showcoastlines=True, showland=True,␣
 ↪fitbounds="locations")
fig.update_layout(
    title_text = 'Covid confirmed cases rate (%) in Mexico',
    font=dict(
        #family="Courier New, monospace",
        family="Ubuntu",
        size=18,
        color="#7F7F7F"
    ),
)
# plt.savefig("images/Covid confirmed cases rate in Mexico.png")
fig.show()
```

4.3 Crime Rate Heat Map

4.4 Which states have the highest Crime rates?

[86]:

```
df=pd.read_csv('clean_data/crimenes_a.csv', encoding = "ISO-8859-1")
repo_url = 'https://raw.githubusercontent.com/angelnmara/geojson/master/
 ↪mexicoHigh.json' #Archivo GeoJSON
mx_regions_geo = requests.get(repo_url).json()
fig = px.choropleth(data_frame=df,
                    geojson=mx_regions_geo,
                    locations='State', # nombre de la columna del Dataframe
                    featureidkey='properties.name',  # ruta al campo del␣
 ↪archivo GeoJSON con el que se hará la relación (nombre de los estados)
                    color="Rate" , #El color depende de las cantidades
                    color_continuous_scale='Blues', #blue
                    #scope="north america"
                   )
fig.update_geos(showcountries=True, showcoastlines=True, showland=True,␣
 ↪fitbounds="locations")
fig.update_layout(
    title_text = 'Crime rate (%) in Mexico',
    font=dict(
        #family="Courier New, monospace",
        family="Ubuntu",
        size=18,
        color="#7F7F7F"
    ),
)
#plt.savefig("images/Crime rate in Mexico.png")
fig.show()
```

[87]:

```
# Safe Travel Locations Mexico Heatmap
## Which states have the lowest Crime rates and Covid rates combined?
```

```
### We want to direct Toruism investment to the safest places for tourists.
```

[88]:
```
df=pd.read_csv('clean_data/Combined_rates.csv', encoding = "ISO-8859-1")
repo_url = 'https://raw.githubusercontent.com/angelnmara/geojson/master/
↪mexicoHigh.json' #Archivo GeoJSON
mx_regions_geo = requests.get(repo_url).json()
fig = px.choropleth(data_frame=df,
                    geojson=mx_regions_geo,
                    locations='State', # nombre de la columna del Dataframe
                    featureidkey='properties.name',  # ruta al campo del␣
↪archivo GeoJSON con el que se hará la relación (nombre de los estados)
                    color="Rate" , #El color depende de las cantidades
                    color_continuous_scale='twilight', #blue
                    #scope="north america"
                   )
fig.update_geos(showcountries=True, showcoastlines=True, showland=True,␣
↪fitbounds="locations")
fig.update_layout(
    title_text = 'Safe Travel Locations Mexico Heatmap',
    font=dict(
        #family="Courier New, monospace",
        family="Ubuntu",
        size=18,
        color="#7F7F7F"
    ),
)
# plt.savefig("images/Safe Travel Locations Mexico Heatmap.png")
fig.show()
```

5 Best places to invest in Safe Tourism are Baja California Sur, Campeche, Yucatán and Chiapas.

[92]:
```
## If you cant see the maps please refer to the images folder
```

[ ]:

[ ]: