COVID-19 cases in Japan by prefecture (47 prefectures)

In this project, I'm going to analyze the correlationship between population (& population density) and the number of positive COVID-19 cases in Japan, by prefecture. (Japan has 47 prefectures.Tokyo is one of them.)

I will only focus on the total COVID-19 patients by prefecture, not on changes in the trend at daily or weekly basis.

Data source

Source: ./covid-19-japan-2022-01-04.csv

The csv file can be downloaded from https://www.stopcovid19.jp/tableview.html.
This is a portal website for COVID-19 run by the Japanese governmental organization.
There is no potential for personally identifiable distinctions as it only contains data 'by prefecture'.

Data modification

By passing "thousands=','" parameters, I will treat "npopulation" & "area" columns as int64 (the original not string object) so that they can be easily sorted for analysis. Also, even though the csv file has columns for # deaths or # current patients (who are not recovered yet), I will not focus on those data.

In [204…

```
# import modules
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# read the csv file
df = pd.read_csv(
    './covid-19-japan-2022-01-04.csv',
    thousands=',',
    usecols=['name', 'ntotalpatients', 'npopulation', 'areainkm^2'])
```

table sorted by # covid-19 patients

This table shows that top 10 prefectures for COVID-19 patients (positive) cases are:
Tokyo, Osaka, Kanagawa, Saitama, Aichi, Chiba, Hyogo, Fukuoka, Hokkaido and Okinawa.

In [242…

```
df_num_patients = df.sort_values(by=['ntotalpatients'], ascending=False)
# sort by # total patients
df_num_patients.reset_index(drop=True, inplace=True)
df_num_patients.index = np.arange(1, len(df)+1)
df_num_patients.index.name = '#_patients_rank'
# indexing from 1 & changing the index name for better readability

df_num_patients['#_patients_rank'] = df_num_patients.index
# creating another column for when this df is merged with the another one
# df_num_patients["#_patients_rank"] = df_num_patients["#_patients_rank"].astype(str)
df_num_patients
```

Out[242…

	name	ntotalpatients	npopulation	areainkm^2	populationdensity	#_patients_rank
#_patients_rank						

	name	ntotalpatients	npopulation	areainkm^2	populationdensity	#_patients_rank
#_patients_rank						
1	Tokyo	383495	13159000	2104	6254.277567	1
2	Osaka	204120	8865000	1901	4663.335087	2
3	Kanagawa	169960	9048000	2416	3745.033113	3
4	Saitama	116175	7195000	3768	1909.501062	4
5	Aichi	106840	7411000	5116	1448.592651	5
6	Chiba	100779	6216000	5082	1223.140496	6
7	Hyogo	78927	5588000	8396	665.555026	7
8	Fukuoka	74842	5072000	4847	1046.420466	8
9	Hokkaido	61576	5506000	83457	65.974094	9
10	Okinawa	51120	1393000	2277	611.769873	10
11	Kyoto	36278	2636000	4613	571.428571	11
12	Shizuoka	26912	3765000	7255	518.952447	12
13	Ibaraki	24504	2970000	6096	487.204724	13
14	Hiroshima	22339	2861000	8480	337.382075	14
15	Gifu	19303	2081000	9768	213.042588	15
16	Gunma	17432	2008000	6362	315.624018	16
17	Miyagi	16300	2348000	6862	342.174293	17
18	Nara	15997	1401000	3691	379.571932	18
19	Tochigi	15688	2008000	6408	313.358302	19
20	Okayama	15568	1945000	7010	277.460770	20
21	Mie	14825	1855000	5762	321.936827	21
22	Kumamoto	14398	1817000	7268	250.000000	22
23	Shiga	12576	1411000	3767	374.568622	23
24	Fukushima	9532	2029000	13783	147.210332	24
25	Kagoshima	9133	1706000	9045	188.612493	25
26	Nagano	8998	2152000	13105	164.212133	26
27	Niigata	8264	2374000	10364	229.062138	27
28	Oita	8189	1197000	5100	234.705882	28
29	Ishikawa	8059	1170000	4186	279.503106	29
30	Miyazaki	6153	1135000	6795	167.034584	30
31	Nagasaki	6133	1427000	4106	347.540185	31
32	Yamaguchi	6020	1451000	6114	237.324174	32
33	Aomori	5912	1373000	9645	142.353551	33
34	Saga	5881	850000	2440	348.360656	34
35	Ehime	5415	1431000	5679	251.980983	35
36	Wakayama	5306	1002000	4726	212.018620	36
37	Yamanashi	5166	863000	4201	205.427279	37
38	Toyama	4870	1093000	2046	534.213099	38

#_patients_rank	name	ntotalpatients	npopulation	areainkm^2	populationdensity	#_patients_rank
39	Kagawa	4712	996000	1862	534.908700	39
40	Kochi	4172	764000	7105	107.529909	40
41	Yamagata	3608	1169000	6652	175.736621	41
42	Iwate	3496	1330000	15279	87.047582	42
43	Tokushima	3292	785000	4147	189.293465	43
44	Fukui	3119	806000	4190	192.362768	44
45	Akita	1934	1086000	11636	93.331042	45
46	Shimane	1774	717000	6708	106.887299	46
47	Tottori	1669	589000	3507	167.949815	47

table sorted by # population

This table shows that top 10 prefectures for population are:
Tokyo, Osaka, Kanagawa, Saitama, Aichi, Chiba, Hyogo, Fukuoka, Hokkaido and Okinawa.

In [250…

```
df_num_population = df.sort_values(by=['npopulation'], ascending=False)
# sort by # population
df_num_population.reset_index(drop=True, inplace=True)
df_num_population.index = np.arange(1, len(df)+1)
df_num_population.index.name = '#_population'
# indexing from 1 & changing the index name for better readability
df_num_population['#_population_rank'] = df_num_population.index
df_num_population
```

Out[250…

#_population	name	ntotalpatients	npopulation	areainkm^2	populationdensity	#_population_rank
1	Tokyo	383495	13159000	2104	6254.277567	1
2	Kanagawa	169960	9048000	2416	3745.033113	2
3	Osaka	204120	8865000	1901	4663.335087	3
4	Aichi	106840	7411000	5116	1448.592651	4
5	Saitama	116175	7195000	3768	1909.501062	5
6	Chiba	100779	6216000	5082	1223.140496	6
7	Hyogo	78927	5588000	8396	665.555026	7
8	Hokkaido	61576	5506000	83457	65.974094	8
9	Fukuoka	74842	5072000	4847	1046.420466	9
10	Shizuoka	26912	3765000	7255	518.952447	10
11	Ibaraki	24504	2970000	6096	487.204724	11
12	Hiroshima	22339	2861000	8480	337.382075	12
13	Kyoto	36278	2636000	4613	571.428571	13
14	Niigata	8264	2374000	10364	229.062138	14
15	Miyagi	16300	2348000	6862	342.174293	15
16	Nagano	8998	2152000	13105	164.212133	16

	name	ntotalpatients	npopulation	areainkm^2	populationdensity	#_population_rank
#_population						
17	Gifu	19303	2081000	9768	213.042588	17
18	Fukushima	9532	2029000	13783	147.210332	18
19	Tochigi	15688	2008000	6408	313.358302	19
20	Gunma	17432	2008000	6362	315.624018	20
21	Okayama	15568	1945000	7010	277.460770	21
22	Mie	14825	1855000	5762	321.936827	22
23	Kumamoto	14398	1817000	7268	250.000000	23
24	Kagoshima	9133	1706000	9045	188.612493	24
25	Yamaguchi	6020	1451000	6114	237.324174	25
26	Ehime	5415	1431000	5679	251.980983	26
27	Nagasaki	6133	1427000	4106	347.540185	27
28	Shiga	12576	1411000	3767	374.568622	28
29	Nara	15997	1401000	3691	379.571932	29
30	Okinawa	51120	1393000	2277	611.769873	30
31	Aomori	5912	1373000	9645	142.353551	31
32	Iwate	3496	1330000	15279	87.047582	32
33	Oita	8189	1197000	5100	234.705882	33
34	Ishikawa	8059	1170000	4186	279.503106	34
35	Yamagata	3608	1169000	6652	175.736621	35
36	Miyazaki	6153	1135000	6795	167.034584	36
37	Toyama	4870	1093000	2046	534.213099	37
38	Akita	1934	1086000	11636	93.331042	38
39	Wakayama	5306	1002000	4726	212.018620	39
40	Kagawa	4712	996000	1862	534.908700	40
41	Yamanashi	5166	863000	4201	205.427279	41
42	Saga	5881	850000	2440	348.360656	42
43	Fukui	3119	806000	4190	192.362768	43
44	Tokushima	3292	785000	4147	189.293465	44
45	Kochi	4172	764000	7105	107.529909	45
46	Shimane	1774	717000	6708	106.887299	46
47	Tottori	1669	589000	3507	167.949815	47

Merged table: Correlationship between # patients & # population

Here's a top 10 prefectures for # COVID-19 patients (positive cases).
We can see that prefectures that are in from the 1st place to 9th place for population are also ranked similarly for COVID-19 patients.

The only exception is **Okinawa** (10th place for # patients but 30th place for population).
There are multiple reasons that have caused this.

One of them is the fact that Okinawa is one of the most popular destination for domestic tourism. According to [JTB] (https://www.jtb.or.jp/wp-content/uploads/2020/10/nenpo2020_1-2.pdf), Okinawa was 4th most popular destination for domestic tourists after Tokyo, Osaka and Hokkaido in 2019. Therefore, there may be possibility that domestic tourists travelled to Okinawa more than to the other prefectures and it caused Okinawa is ranked in the 10th place.

I will have a look at another perspective later in this analysis in **the popluation density** section.

In [251…

```
merged_df_npatients_npop = pd.merge(df_num_patients, df_num_population)
# merge two dfs
merged_df_npatients_npop.reset_index(drop=True, inplace=True)
merged_df_npatients_npop.index = np.arange(1, len(df)+1)
# indexing from 1 for better readability
merged_df_npatients_npop.head(10)
# showing top 10 prefectures
```

Out[251…

	name	ntotalpatients	npopulation	areainkm^2	populationdensity	#_patients_rank	#_population_rank
1	Tokyo	383495	13159000	2104	6254.277567	1	1
2	Osaka	204120	8865000	1901	4663.335087	2	3
3	Kanagawa	169960	9048000	2416	3745.033113	3	2
4	Saitama	116175	7195000	3768	1909.501062	4	5
5	Aichi	106840	7411000	5116	1448.592651	5	4
6	Chiba	100779	6216000	5082	1223.140496	6	6
7	Hyogo	78927	5588000	8396	665.555026	7	7
8	Fukuoka	74842	5072000	4847	1046.420466	8	9
9	Hokkaido	61576	5506000	83457	65.974094	9	8
10	Okinawa	51120	1393000	2277	611.769873	10	30

Scatter: correlationship between # patients and # population in Japan by prefecture

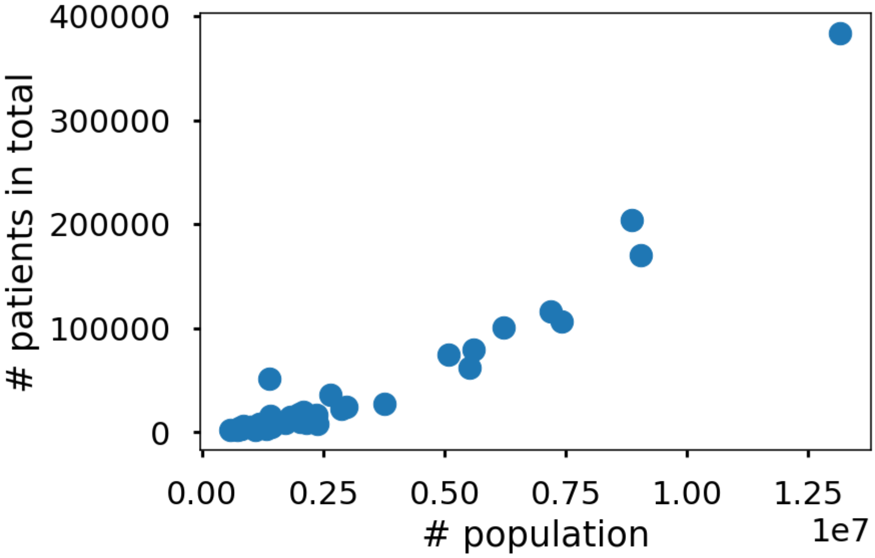
We can see a correlationship between # patients and # population.
The more population a prefecture has (x axis), the more # COVID-19 patients it is likely to have.

In [206…

```
df = df.sort_values(by=['ntotalpatients'], ascending=False)
plt.ylabel("# patients in total")
plt.xlabel("# population")
plt.scatter(df['npopulation'],df['ntotalpatients'])
```

Out[206…

```
<matplotlib.collections.PathCollection at 0x157c07b6dc0>
```



correlationship between # patients and # population density in Japan by prefecture

Another perspective which is worth having look at will be **population density** (not just # population). This may have a clearer correlationship than # population as we don't see an exception like Okinawa as we did in the other table.

In [252...

```
df_pop_density = df
df_pop_density['populationdensity'] = (df_pop_density['npopulation']/df_pop_density['areainkm^2'
df_pop_density = df_pop_density.sort_values(by=['populationdensity'], ascending=False)
df_pop_density.reset_index(drop=True, inplace=True)
df_pop_density.index = np.arange(1, len(df)+1)
df_pop_density['population_density_rank'] = df_pop_density.index

merged_df_popdensity_num_patients = pd.merge(df_pop_density, df_num_patients)
merged_df_popdensity_num_patients.reset_index(drop=True, inplace=True)
merged_df_popdensity_num_patients.index = np.arange(1, len(df)+1)
merged_df_popdensity_num_patients.head(10)
```

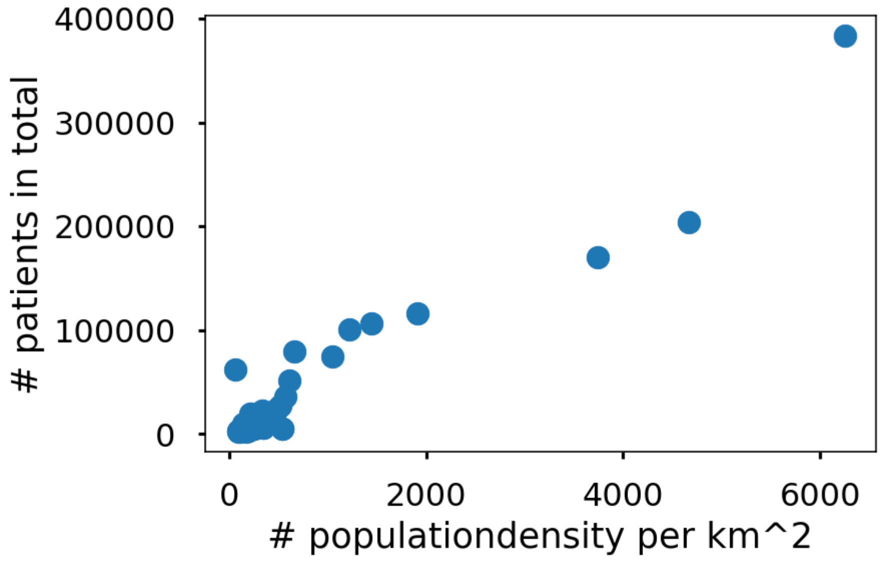
Out[252...

	name	ntotalpatients	npopulation	areainkm^2	populationdensity	population_density_rank	#_patients_r
1	Tokyo	383495	13159000	2104	6254.277567	1	
2	Osaka	204120	8865000	1901	4663.335087	2	
3	Kanagawa	169960	9048000	2416	3745.033113	3	
4	Saitama	116175	7195000	3768	1909.501062	4	
5	Aichi	106840	7411000	5116	1448.592651	5	
6	Chiba	100779	6216000	5082	1223.140496	6	
7	Fukuoka	74842	5072000	4847	1046.420466	7	
8	Hyogo	78927	5588000	8396	665.555026	8	
9	Okinawa	51120	1393000	2277	611.769873	9	
10	Kyoto	36278	2636000	4613	571.428571	10	

In [262...

```
plt.ylabel("# patients in total")
plt.xlabel("# populationdensity per km^2")
plt.scatter(df_pop_density['populationdensity'],df_pop_density['ntotalpatients'])
```

Out[262… <matplotlib.collections.PathCollection at 0x157c0ba9e80>



Exception: Hokkaido population density - 47th place, # patients 9th

However, there's still an exception in the scatter map, which is Hokkaido. Even though its population density is in the 47th place (best/worst), it is in the 9th place for # patients. This may be becuase its unique charactaristics: **Hokkaido has 22% of whole the area of the country with only 4% of the national poplulation.**

In addition, its capital city Sapporo has almost 2 million population, which is more than 40% of the prefectual population. These charactaristic caused the exception in the table. It'd be interesting to see compare COVID-19 patients data between cities focusing on population density.

In [261…

```
merged_df_popdensity_num_patients.tail(1)
```

Out[261…

	name	ntotalpatients	npopulation	areainkm^2	populationdensity	population_density_rank	#_patients_ra
47	Hokkaido	61576	5506000	83457	65.974094	47	