Data Analysis of Covid-19 in India

Anurag Sengupta

Golden Gate University

GGU ID: 0597276

MSBA 320, Summer 2020

Contents

Introduction to Covid-19

Coronaviruses are a large family of viruses which may cause illness in animals or humans. In humans, several coronaviruses are known to cause respiratory infections ranging from the common cold to more severe diseases such as Middle East Respiratory Syndrome (MERS) and Severe Acute Respiratory Syndrome (SARS). The most recently discovered coronavirus causes coronavirus disease COVID-19.

COVID-19 is the infectious disease caused by the most recently discovered coronavirus. This new virus and disease were unknown before the outbreak began in Wuhan, China, in December 2019. COVID-19 is now a pandemic affecting many countries globally. People can catch COVID-19 from others who have the virus. The disease spreads primarily from person to person through small droplets from the nose or mouth, which are expelled when a person with COVID-19 coughs, sneezes, or speaks. These droplets are relatively heavy, do not travel far and quickly sink to the ground. People can catch COVID-19 if they breathe in these droplets from a person infected with the virus.

The most common symptoms of COVID-19 are fever, dry cough, and tiredness. Other symptoms that are less common and may affect some patients include aches and pains, nasal congestion, headache, conjunctivitis, sore throat, diarrhea, loss of taste or smell or a rash on skin or discoloration of fingers or toes. These symptoms are usually mild and begin gradually. Some people become infected but only have very mild symptoms. Most people (about 80%) recover from the disease without needing hospital treatment. Around 1 out of every 5 people who gets COVID-19 becomes seriously ill and develops difficulty breathing. Older people, and those with underlying medical problems like high blood pressure, heart and lung problems, diabetes, or cancer, are at higher risk of developing serious illness. However, anyone can catch COVID-19 and become seriously ill.

Covid-19 in India

Owing to the huge population of India which can be approximated close to 1.2 billion, it has been one of the worst affected countries in terms of cases as well as death toll. As of 6th August 2020, the number of active cases stands at 595501 and the death toll stands at 40,699 as per the information from Ministry of Health and Family Welfare, Government of India. In Spite of enforcing the largest lockdown in the world, India hasn't been able to curb the rampant number of cases successfully. With a surge in the number of cases everyday Covid-19 has the potential to inflict a major catastrophe in India. This project has been undertaken to analyze the impact of Covid-19 in India across the states and evaluate the Indian healthcare infrastructure to combat the pandemic with availability of beds and testing facility. After analyzing the procured data, we can comment on the efficacy of the government authorities in flattening the curve and the necessary measures that can be taken to mitigate the potential crisis.

Gathering data information on Covid-19 in India

The datasets for the proposed analysis have been gathered from the Ministry of Health and Family Welfare, Government of India. The area of focus remains of three datasets namely:

a) **Covid-19 India,** which contains data on the number cases since 30th January 2020 until 4th August 2020 across the states of India.

b) Hospital Beds, which focuses on the available hospital beds across the country segregated under different types of health centers.

c) **India Population,** which comprises of the population of India segregated across different states and areas.

d) **Testing,** which focuses on the testing across the states of India accounting for the total number of samples and positive cases found.

Descriptive Statistics

Descriptive statistics does not let anyone jump into conclusions beyond the data that has been analysed or make conclusions with regards to any hypotheses one might have made. Descriptive statistics are very essential because it helps to present the concise and yet transparent synopsis of the data in a more meaningful way. It enables the easy interpretation of the data. Here in the project which has been undertaken, each sheets of the datasets contains data under a segregated column and row head that will be explained independently.

1) **Covid-19 India Dataset has the following column heads:**

   Serial Number, Date of data collected, Time of data collected, State/Union Territory of the data to be reported, Positive Cases of Confirmed Indian National, Positive cases of Confirmed Foreign National, Total Cured cases until the respective date, Death toll until the respective date, conformed cases until that date.

   We closely analysed the column heads Cured, Deaths and Confirmed Cases and could identify the following: Maharashtra is the worst affected State with highest number of cumulative Confirmed cases: 4,50,196 and deaths : 15,842 until 4th August 2020. In terms of cured cases, Maharashtra comes on top of the list with 2,87,030 cases.

   Among the states with least numbers of positive cases, Mizoram tops the chart with 501 cases until 4th August 2020. It also features as the only state with zero deaths. The mean for the above column heads i.e. Cured Cases, Death Tolls and Confirmed Cases stands at 6503.380, 382.182 and 10887.63 respectively, it has factored in every value in the data sheet as part of the calculation. It has a significant higher value for each and every entity owing to the outliers, as the number of cured cases, deaths and confirmed cases increased every subsequent day and they were significantly higher by the start of August. We can also attribute the high mean count for the entities to the numbers reported by certain states like Maharashtra, Delhi, Andhra Pradesh, Karnataka and

Tamil Nadu. All these states have an extremely high number of confirmed cases, deaths as well cured cases. In the same datasheet of Covid-19 India we can also witness an extremely high standard deviation across the heads: Cured Cases, Death Tolls and Confirmed Cases which stands at 21762.762, 1167.32 and 35093.89, respectively. This is owing to the spread of data from 30th January 2020 until 4th August 2020 which involved the gradual increase in number of cases across the different states of India with every passing day and the presence of outliers,

	Sno	Cured	Deaths	Confirmed
count	4846.000000	4846.000000	4846.000000	4.847000e+03
mean	2423.500000	6503.380726	282.182212	2.177078e+04
std	1399.064032	21762.767123	1167.320531	7.585008e+05
min	1.000000	0.000000	0.000000	0.000000e+00
25%	1212.250000	8.000000	0.000000	4.100000e+01
50%	2423.500000	188.000000	4.000000	6.590000e+02
75%	3634.750000	2681.000000	65.000000	5.496000e+03
max	4846.000000	287030.000000	15842.000000	5.276148e+07

Fig: Descriptive Statistics Summary for Covid-19 India datasheet

2) In the second datasheet the **Hospital Beds,** comprising of the available hospital beds across the country segregated under different types of health centers we have the data segregated along the following column heads:
   a) Number of Primary Health Care Centers HMIS
   b) Number of Community Health Care Centers HMIS
   c) Number of Sub District Hospitals HMIS
   d) Number of District Hospitals HMIS
   e) Total Public Health Facilities HMIS
   f) Number of Public Beds HMIS
   g) Number of Rural Hospitals NHP18

h) Number of Rural Beds NHP18

i) Number of Urban Hospitals NHP18

j) Number of Urban Beds NHP18

With an all India count of 29899 Primary Health Care centers HMIS, the number of Primary Health care centers are well spread out across the states of India. With the highest being in Uttar Pradesh accounting for 3277 and Daman and Diu accounting for the least at 4. The meant count and standard deviation for the above column head stands at 830.52 and 906.91 respectively.

The number of Community Health Care Centers HMIS throughout India stands at 5568 with Uttar Pradesh yet again accounting for the highest count at 671 and the lowest count is attributed to Daman & Diu, Dadra & Nagar Haveli and Sikkim each having 2 centers each.

The meant count and standard deviation for the above column head stands at 154.66 and 178.37 respectively.

The number of district and sub district hospitals across India stands at 1003 and 1255, respectively. The count for the total public health care facilities is the highest across the country which stands at 37725, with yet again Uttar Pradesh accounting for highest number at 4122 public health care facilities. The total number of public beds available across the country stands at 739024 with mean and standard deviation at 1047.91 and 1122.10 respectively. Again, in this instance, it can be observed that the mean and standard deviation is remarkably high, and this can be attributed to certain states having higher number of public beds available in states like Uttar Pradesh and Maharashtra. This figure can be identified as the outliers amidst the other states with a moderate or minimal count of public beds. We further

delve into Number of Rural Hospitals NHP18 and Number of Rural Beds NHP18. We again observe a similar trend like the public health care infrastructure here.

We again observe an extremely high mean and standard deviation for Number of rural beds at 7766.33 and 9955.55. This can again be attributed to the high number of beds available in states like Uttar Pradesh. Tamil Nadu, Karnataka, and Maharashtra.

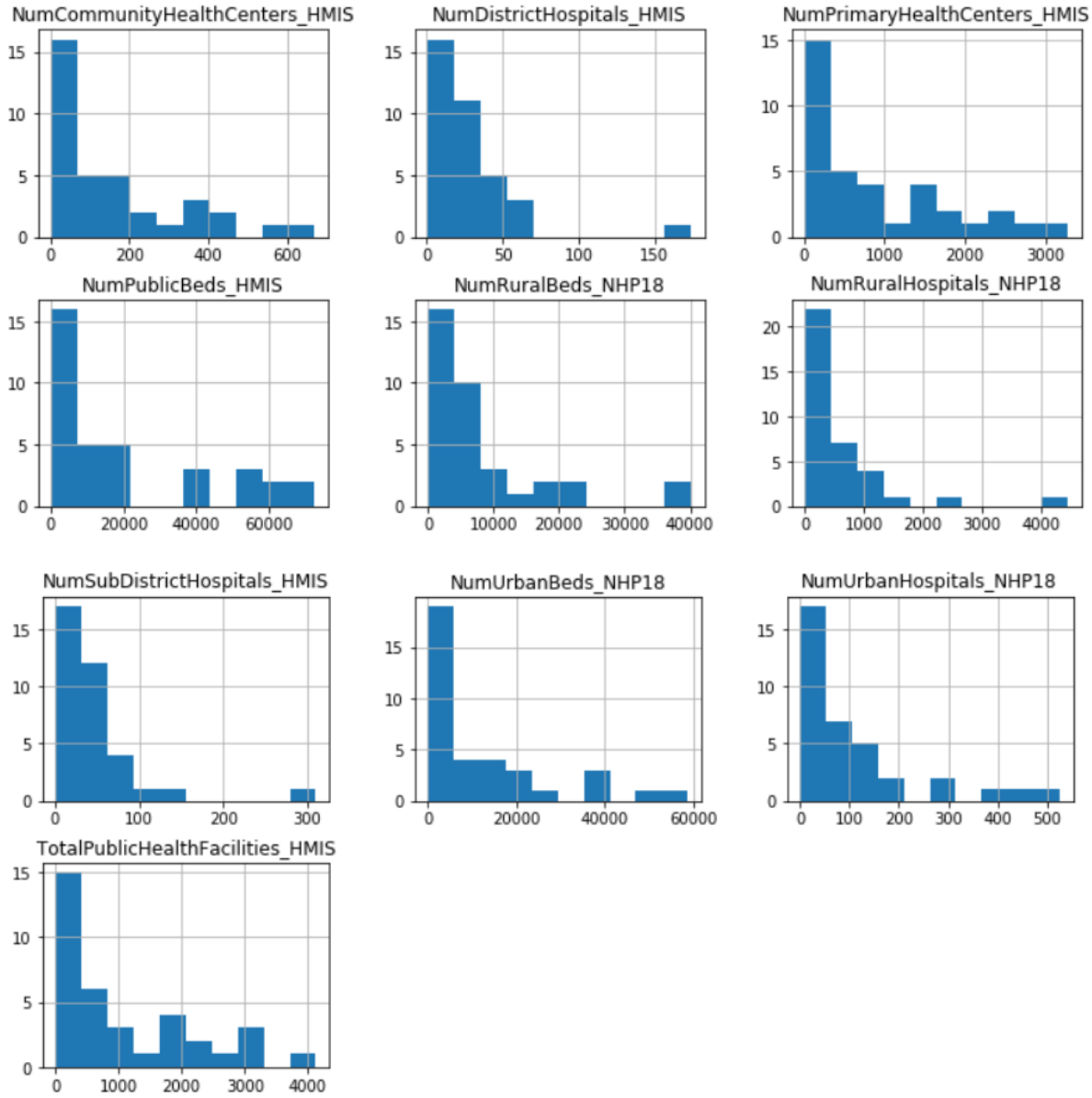
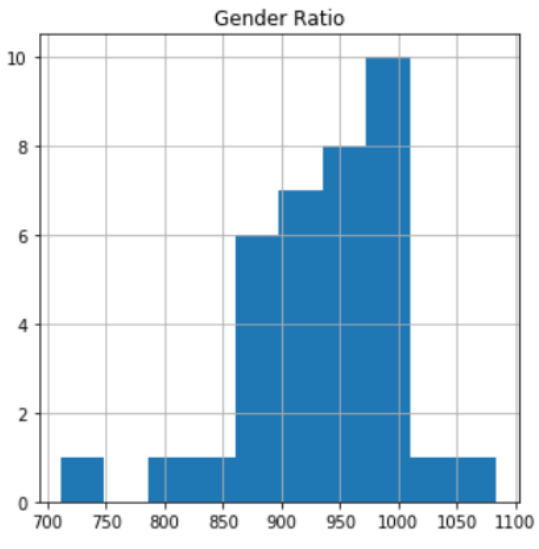


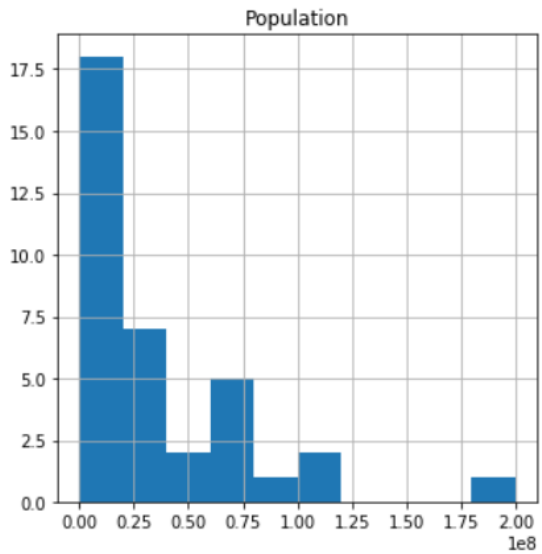
Fig: Histogram on the distribution of beds across the country

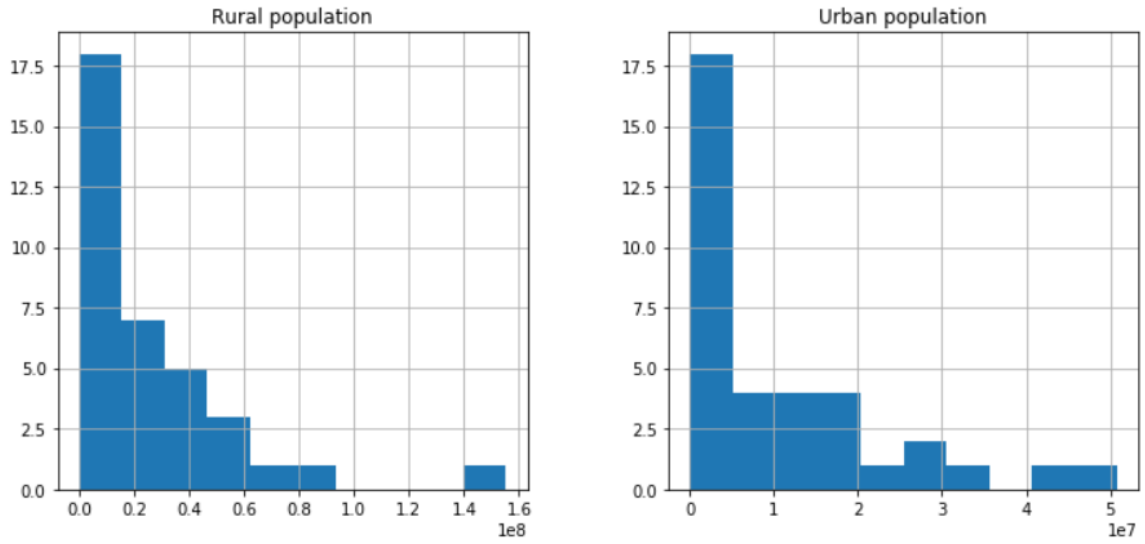
3) In the third datasheet **India Population**, we observe the population is again heavily concentrated across a selected few state in India. Uttar Pradesh and Maharashtra are the most populous states of the country with a population of 199812341 and 112374333 respectively whereas Ladakh and Lakshadweep are the least populous states of the country with a population of 274000 and 64473, respectively.

	Sno	Population	Rural population	Urban population	Gender Ratio
count	36.000000	3.600000e+01	3.600000e+01	3.600000e+01	36.000000
mean	18.500000	3.362689e+07	2.315336e+07	1.047353e+07	937.583333
std	10.535654	4.305758e+07	3.212429e+07	1.312631e+07	65.544478
min	1.000000	6.447300e+04	1.414100e+04	5.033200e+04	711.000000
25%	9.750000	1.439840e+06	5.451570e+05	6.652765e+05	907.750000
50%	18.500000	2.106970e+07	1.278679e+07	5.167890e+06	947.000000
75%	27.250000	5.229275e+07	3.496766e+07	1.604342e+07	976.750000
max	36.000000	1.998123e+08	1.553173e+08	5.081826e+07	1084.000000

Fig: Descriptive Statistics Summary for Population







Fig(above): Histogram on the Gender Ratio and Population of the country

4) The final datasheet i.e. Testing focuses on the vigorous testing that is being carried out across the country in different states with respect to total number of samples, positive cases as well as the negative cases. From this datasheet we can extrapolate the efficacy of the states in terms of executing the testing with the sheer numbers in the form of total samples considered.

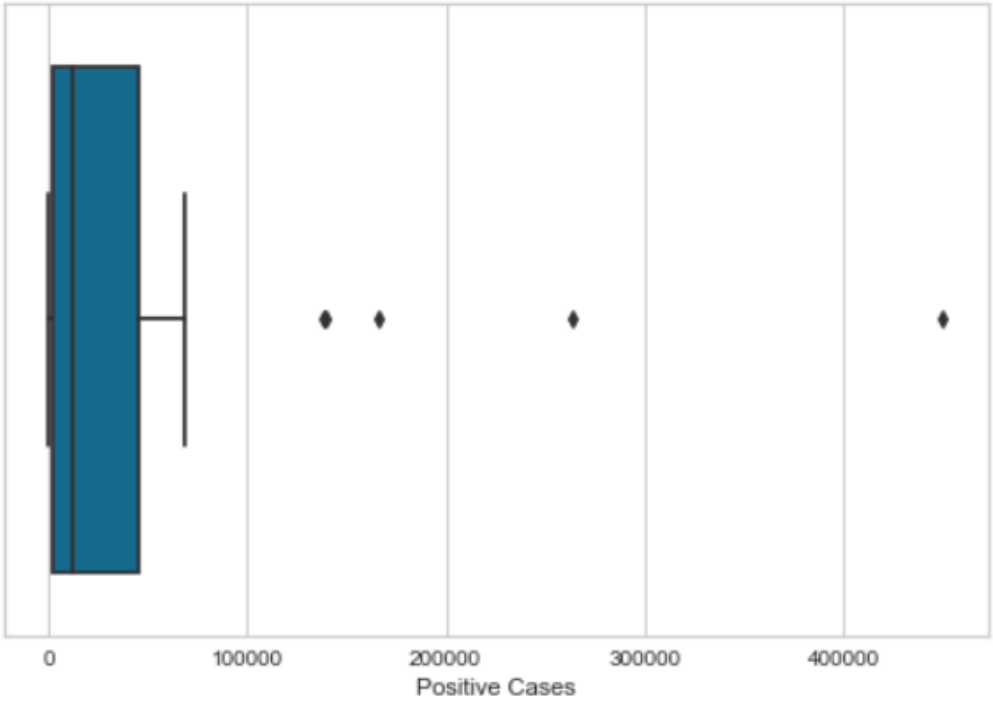
   On running the descriptive statistics on the above datasheet, we found that the states with the greatest number of positive cases until August 3rd, 2020 was Maharashtra. The number stood at a staggering 457476. In terms of Total Samples considered Tamil Nadu leads the list with 2623260 samples considered until August 3rd, 2020.

   For the positive cases, the mean and standard deviation stands at 13741.47 and 39462.57, respectively. The high standard deviation is owing to the widespread of data. Augmented testing facilities across the states led to discovery of higher positive cases every subsequent day until the last day of the collection of data. Therefore, such greater numbers can be observed.

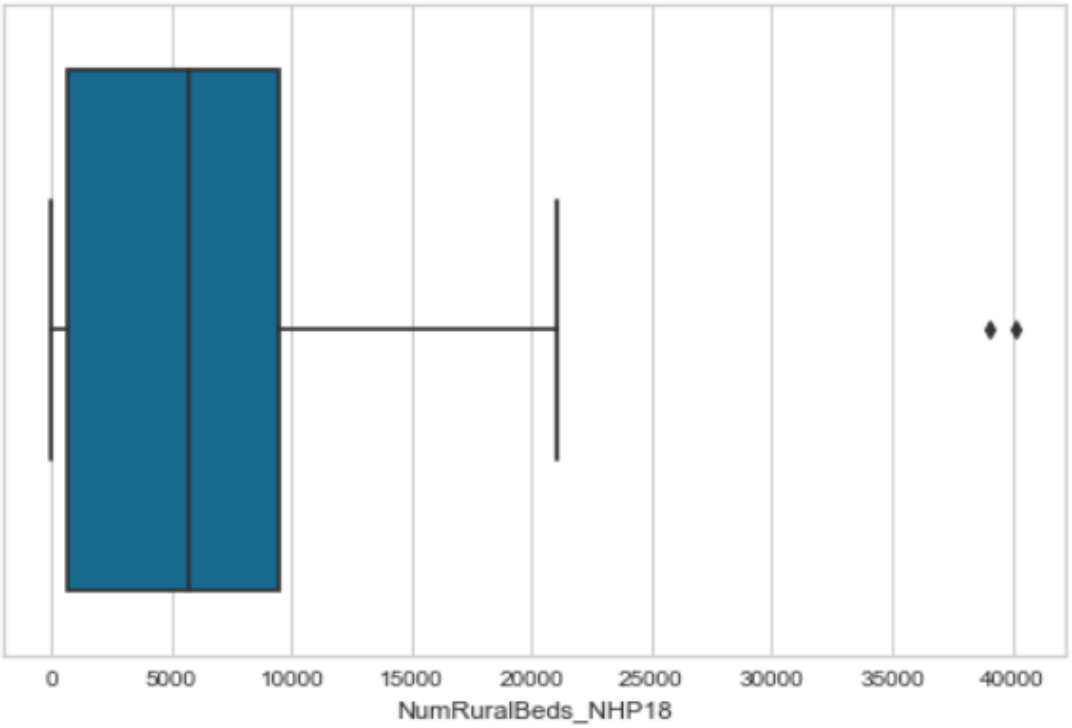
	Total Samples	Positive	Negative
count	3.780000e+03	3780.000000	3.780000e+03
mean	2.199296e+05	13741.470260	2.062894e+05
std	3.707076e+05	39462.573668	3.408445e+05
min	5.800000e+01	0.000000	5.700000e+01
25%	1.210975e+04	173.000000	1.188350e+04
50%	5.781500e+04	1396.500000	5.630950e+04
75%	2.709585e+05	8937.000000	2.570562e+05
max	2.837273e+06	457476.000000	2.574051e+06

Fig(above): Descriptive Statistics Summary for the Testing datasheet

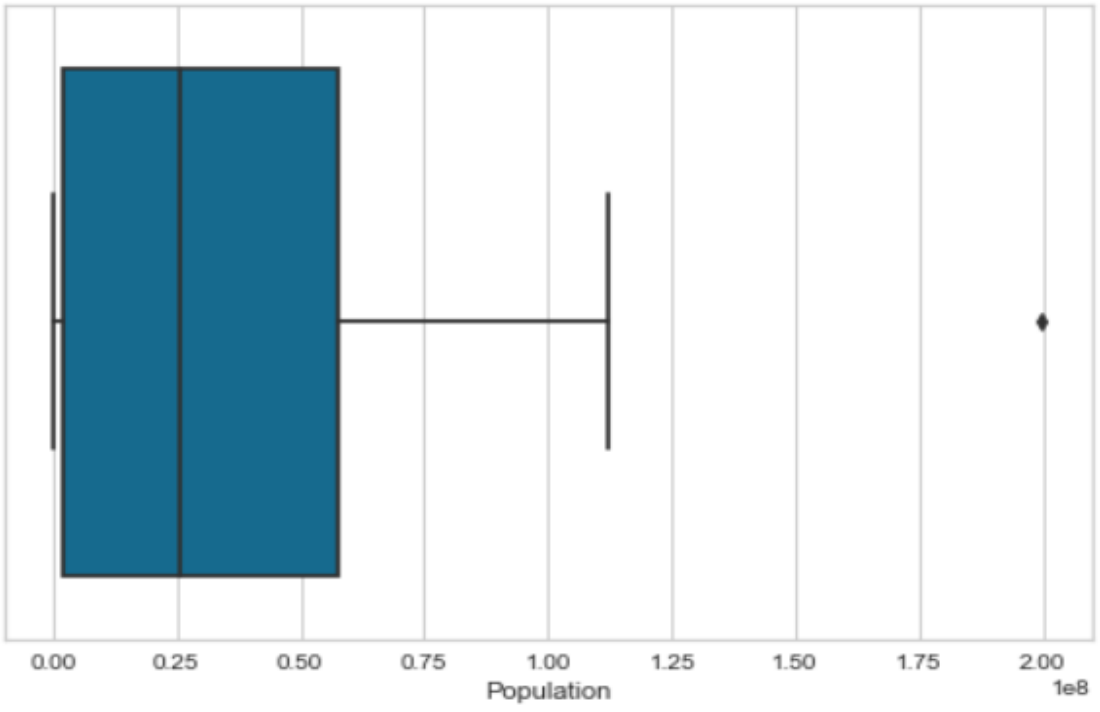
Box Plots



In the above boxplot for Positive case, the median number is around 1000, median (middle quartile) marks the mid-point of the data and is shown by the line that divides the box into two parts. Half the scores are greater than or equal to this value and half are less.



In the above boxplot for the number of rural beds, the median number is 5500.



In the above box plot for population the median population is 0.25 X 1e8 = 25000000.

Correlation Analysis

To test the relationships between the quantitative variables or categorical variables present in this dataset Correlation will be used. To study the relationship between the variables and make predictions about the future behavior correlation analysis constitutes as a very vital test. A correlation coefficient is a way to put a value to the relationship. Correlation coefficients have a value of between -1 and 1. A "0" means there is **no relationship** between the variables at all, while -1 or 1 means that there is a **perfect negative or positive correlation** (negative or positive correlation here refers to the type of graph the relationship will produce).

For the correlation analysis in this data set, Positive cases has been considered as the Independent Variable and the dependent variables considered are:

Total Samples, Negative Cases, Number of Primary Health Care Centers HMIS Number of Community Health Care Centers HMIS, Number of Sub District Hospitals HMIS, Number of District Hospitals HMIS, Total Public Health Facilities HMIS, Number of Public Beds HMIS, Number of Rural Hospitals NHP18, Number of Rural Beds NHP18, Number of Urban Hospitals NHP18, Number of Urban Beds NHP18, Population (Total), Rural Population, Urban Population and Gender Ratio.

Fig (below): Correlation Analysis Table between the dependent and independent Variables

	Total Samples	Positive Cases	Negative Cases	NumPrimaryHealthCenters_HMIS	NumCommunityHealthCenters_HMIS
Total Samples	1.000000	0.699724	0.996003	0.824740	0.760059
Positive Cases	0.699724	1.000000	0.633119	0.569170	0.417832
Negative Cases	0.996003	0.633119	1.000000	0.822434	0.771273
NumPrimaryHealthCenters_HMIS	0.824740	0.569170	0.822434	1.000000	0.873355
NumCommunityHealthCenters_HMIS	0.760059	0.417832	0.771273	0.873355	1.000000
NumSubDistrictHospitals_HMIS	0.587149	0.489045	0.575026	0.436193	0.393200
NumDistrictHospitals_HMIS	0.091742	-0.083542	0.109845	-0.040053	-0.036086
TotalPublicHealthFacilities_HMIS	0.145007	-0.031102	0.161001	0.024957	0.017916
NumPublicBeds_HMIS	0.901597	0.738513	0.884536	0.883494	0.813248
NumRuralHospitals_NHP18	0.162833	-0.089924	0.187669	0.072373	0.061496
NumRuralBeds_NHP18	0.820257	0.415990	0.836727	0.806661	0.819242
NumUrbanHospitals_NHP18	0.050652	-0.091211	0.066284	-0.089019	-0.086575
NumUrbanBeds_NHP18	0.038939	-0.102135	0.054959	-0.100452	-0.096647
Population	0.704731	0.370730	0.717215	0.781153	0.725872
Rural population	0.643107	0.252805	0.665190	0.774949	0.708020
Urban population	0.718692	0.590465	0.704870	0.654658	0.637574
Gender Ratio	-0.070797	0.001637	-0.076912	-0.108158	-0.154781

NumSubDistrictHospitals_HMIS	NumDistrictHospitals_HMIS	TotalPublicHealthFacilities_HMIS	NumPublicBeds_HMIS	NumRuralHospitals_NHP18
0.587149	0.091742	0.145007	0.901597	0.162833
0.489045	-0.083542	-0.031102	0.738513	-0.089924
0.575026	0.109845	0.161001	0.884536	0.187669
0.436193	-0.040053	0.024957	0.883494	0.072373
0.393200	-0.036086	0.017916	0.813248	0.061496
1.000000	0.006293	0.051361	0.580458	0.052721
0.006293	1.000000	0.993889	-0.117671	0.981237
0.051361	0.993889	1.000000	-0.051640	0.981323
0.580458	-0.117671	-0.051640	1.000000	-0.044034
0.052721	0.981237	0.981323	-0.044034	1.000000
0.694664	-0.085069	-0.036437	0.824122	0.037557
0.004473	0.991072	0.992615	-0.151211	0.966833
-0.006640	0.992683	0.991885	-0.163983	0.966689
0.277807	0.165490	0.200699	0.604491	0.261486
0.185774	0.086655	0.116708	0.549597	0.193097
0.449126	0.271531	0.313048	0.631763	0.325944
0.145187	-0.033400	-0.032950	-0.084314	-0.031515

NumRuralBeds_NHP18	NumUrbanHospitals_NHP18	NumUrbanBeds_NHP18	Population	Rural population	Urban population	Gender Ratio
0.820257	0.050652	0.038939	0.704731	0.643107	0.718692	-0.070797
0.415990	-0.091211	-0.102135	0.370730	0.252805	0.590465	0.001637
0.836727	0.066284	0.054959	0.717215	0.665190	0.704870	-0.076912
0.806661	-0.089019	-0.100452	0.781153	0.774949	0.654658	-0.108158
0.819242	-0.086575	-0.096647	0.725872	0.708020	0.637574	-0.154781
0.694664	0.004473	-0.006640	0.277807	0.185774	0.449126	0.145187
-0.085069	0.991072	0.992683	0.165490	0.086655	0.271531	-0.033400
-0.036437	0.992615	0.991885	0.200699	0.116708	0.313048	-0.032950
0.824122	-0.151211	-0.163983	0.604491	0.549597	0.631763	-0.084314
0.037557	0.966833	0.966689	0.261486	0.193097	0.325944	-0.031515
1.000000	-0.130037	-0.141715	0.662804	0.637360	0.607130	-0.082271
-0.130037	1.000000	0.999662	0.109385	0.025844	0.237663	-0.018648
-0.141715	0.999662	1.000000	0.101440	0.019236	0.227826	-0.017742
0.662804	0.109385	0.101440	1.000000	0.976565	0.863963	-0.079695
0.637360	0.025844	0.019236	0.976565	1.000000	0.735885	-0.141134
0.607130	0.237663	0.227826	0.863963	0.735885	1.000000	0.091116
-0.082271	-0.018648	-0.017742	-0.079695	-0.141134	0.091116	1.000000

In the correlation analysis table, it can be observed:

a) The total samples have a moderate positive relationship with the positive cases which stands at +0.69.

b) The negative cases have a moderate positive relationship with the positive cases which stands at +0.63.

c) The Number of Primary Health Care Centers HMIS, Number of Community Health Care Centers HMIS and the Number of Sub District Hospitals HMIS all share a moderate positive relationship with the positive cases which stands at +0.56, +0.41, +0.48 with the positive cases.

d) Both, Number of District Hospitals HMIS and Total Public Health Facilities HMIS share a weak negative linear relationship with the Positive cases which stands at -0.08 and -0.03.

e) The number of Public beds hmis and the number of rural beds nhp 18, both share a moderate positive relationship with the positive cases which stands at +0.73 and +0.41.

f) The number of rural hospitals nhp 18, number of urban hospitals nhp 18 and number of urban beds nhp 18 all share a weak negative linear relationship with the Positive cases which stands -0.08, -0.09 and -0.10 respectively.

g) The Population has a weak positive linear relationship with the positive cases which stands at +0.37

Correlation Heat Map

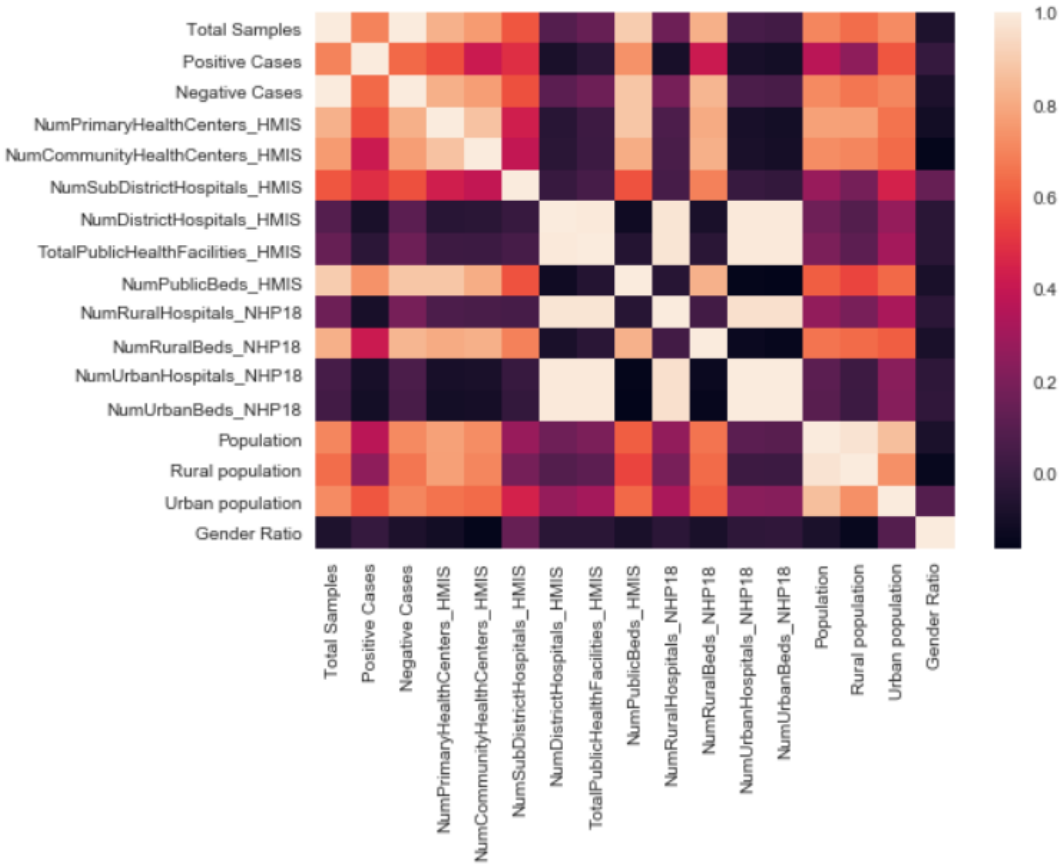


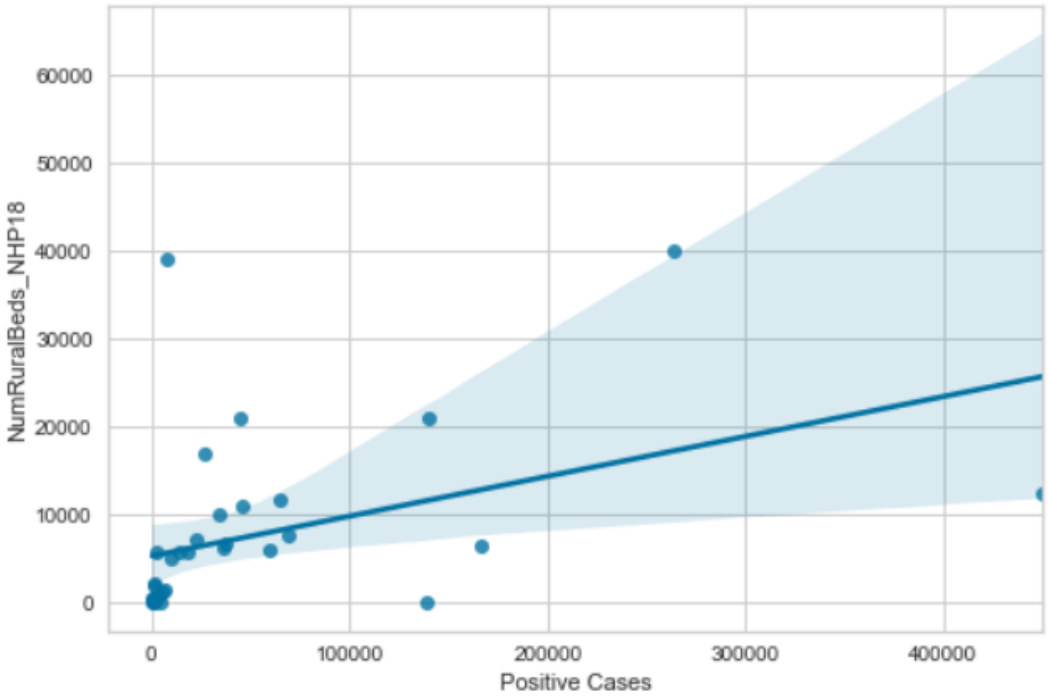
Fig: Correlation Heat map between the dependent and Independent variables

The colour value of the cells is proportional to the number of measurements that matches the dimensional values. This enables us to quickly identify incidence patterns, and to recognize anomalies in a correlational heat map. In the above figure we can observe that the cells with the purple share represent negative correlation whereas the cells with the salmon shade

represent moderately positive correlation and as they move towards the off white shades it represents a strong positive linear relationship.

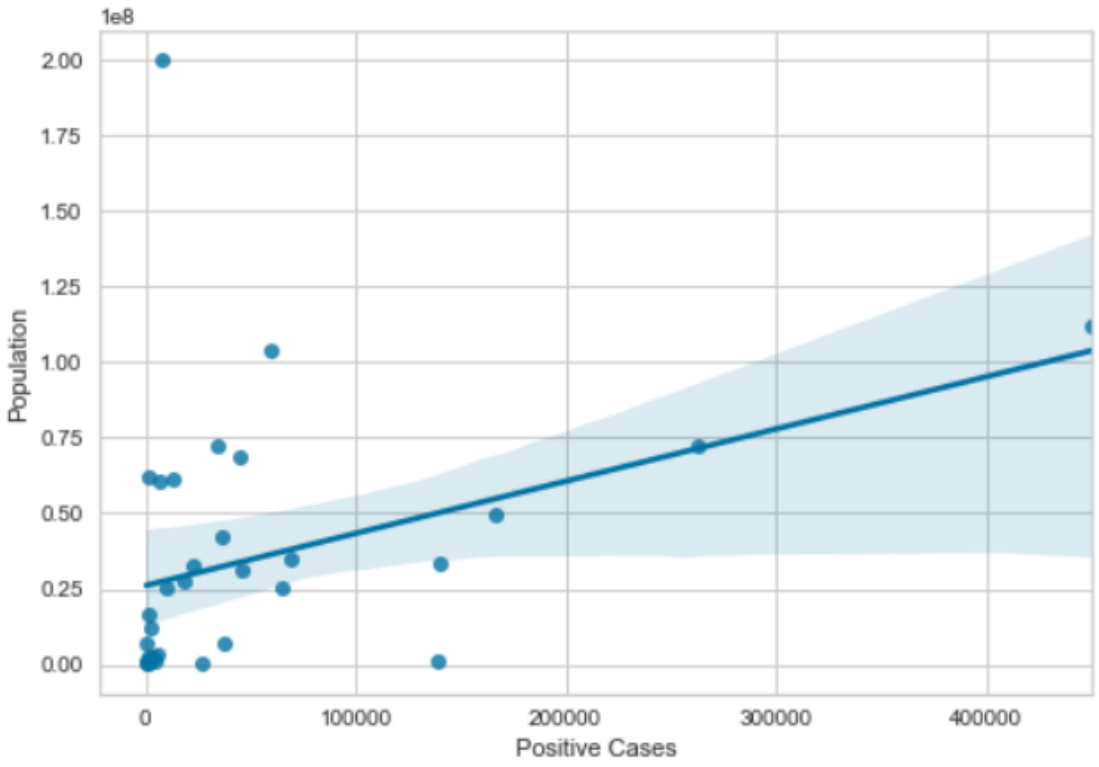
Regression Analysis

Linear regression is an approach to model the relationship between a single **dependent variable** (target variable) and one (simple regression) or more (multiple regression) **independent variables**. The linear regression model assumes a linear relationship between the population and number of Public beds. If this relationship is present, we can estimate the coefficients required by the model to make predictions on new data.



Fig(above): Scatter plot for Positive Cases and Number of Rural Beds_NHP18

From to above scatter plot we can witness an uphill pattern as we move from left to right, this indicates a positive relationship between Positive Cases and Number of Rural Beds. As the X axis -values increase (move right), the Y axis-values tend to increase (move up). We can also observe a few outliers on the scatter plot.



Fig(above): Regression Plot for Positive Cases and Population

In the above case we have undertaken a linear regression between the dependent variable i.e. the positive case and the independent variable i.e. the Population. Here also we can witness an uphill pattern as we move from left to right, this indicates a positive relationship between Positive cases and Population. As the X axis -values increase (move right), the Y axis-values tend to increase (move up).

Anova (Analysis of Variance) Testing

Here, we are trying to find out through an experiment if the results are significant, in other words it helps us to figure out if there is a need to reject the null hypothesis.

The Anova test has two means: One way and Two way.

The one way mean has one independent variable and two way means has two independent variables. Here we are doing a one-way Anova test, comparing between two groups which is used to test two groups to see if there is a difference between them. In this test,

The Null hypothesis: We have enough hospital beds available across different regions and categories of Hospitals in India,

Alternate hypothesis: We do not have enough hospital beds available.

```
# stats f_oneway functions takes the groups as input and returns F and
P-value
fvalue, pvalue = stats.f_oneway(covid_BedVsPopVsCases['Total Samples'],
covid_BedVsPopVsCases['Positive Cases'], covid_BedVsPopVsCases['Urban p
opulation'], covid_BedVsPopVsCases['TotalPublicHealthFacilities_HMIS'])
print("The value from the Stats is given below:")
print("The F value is:", fvalue," The Pvalue is:", pvalue)
```

The value from the Stats is given below:

The F value is: 22.647575259851166. The P value is: 6.832450353949321e-12

In [232]:

```
# reshape the d dataframe suitable for statsmodels package
d_melt = pd.melt(covid_BedVsPopVsCases.reset_index(), id_vars=['index']
, value_vars=['Total Samples', 'Positive Cases', 'Urban population',  '
TotalPublicHealthFacilities_HMIS'])
# replace column names
d_melt.columns = ['index', 'treatments', 'value']
# Ordinary Least Squares (OLS) model
model = ols('value ~ C(treatments)', data=d_melt).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
print("\n\t\tAnova table:")
anova_table
```

Anova table:

	sum_sq	df	F	PR(>F)
C(treatments)	3.001662e+15	3.0	22.647575	6.832450e-12
Residual	5.831668e+15	132.0	NaN	NaN

Interpretation: The P-value obtained from ANOVA analysis is significant (P<0.05), and therefore, we conclude that there are significant differences among treatments and hence we reject the Null Hypothesis.

Ridge Regression

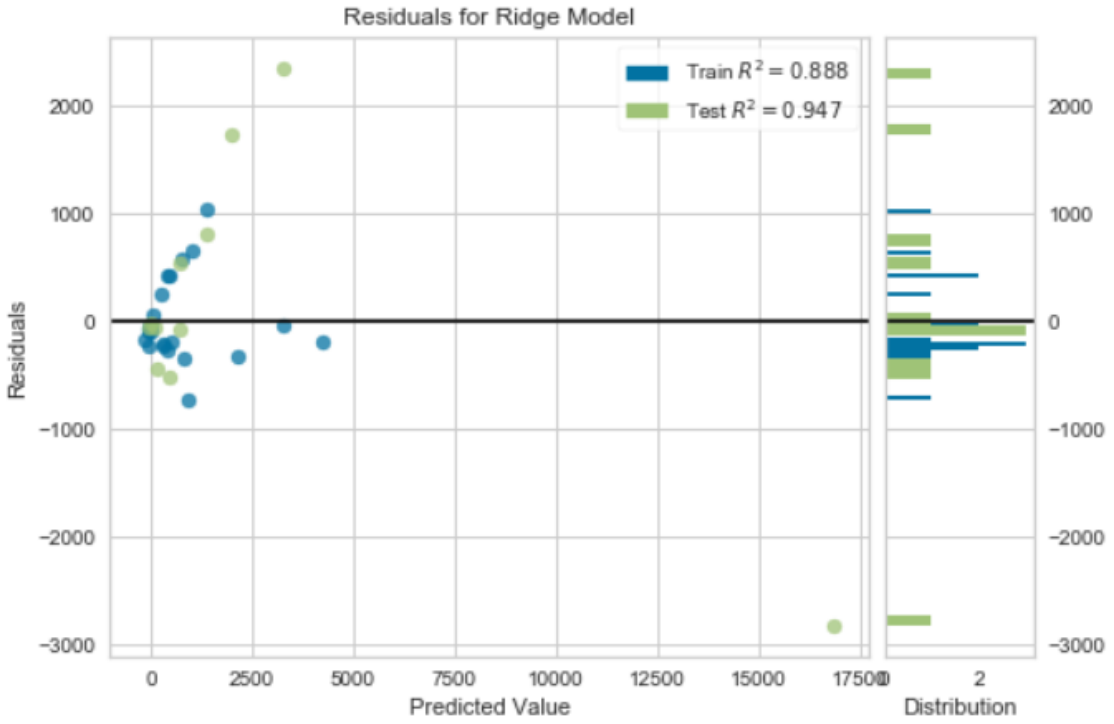
This type of regression is known as ridge, which come from scikitlearn package of Machine Learning. This follows the mathematical theorem of regression. The model gives us a training R^2 of 89% and testing R^2 of 94%.

We have taken multiple variables for the ridge regression with alpha=0, we have taken 75% of the data into training set and 25% into the testing set. Assigning the random state as 890 to adjust the R^2 value and fit the model.

Our target variable is number of rural beds NHP18 as it is significant with 'Positive Cases', 'Negative Cases', 'Number of Community Health Centers HMIS', 'Number of Primary Health Centers _HMIS', 'Rural population', 'Population', 'Number of District Hospitals HMIS'. From the correlation matrix we can observe that the above-mentioned variables share the following degree of correlation with the number of rural beds available:

'Positive Cases : 0.41', 'Negative Cases : 0.83', 'Number of Community Health Centers HMIS : 0.81', 'Number of Primary Health Centers _HMIS : 0.80', 'Rural population : 0.63', 'Population : 0.66', 'Number of District Hospitals HMIS : -0.08 '

The variance of the residuals is constant across observations heteroscedasticity. The standard deviation of the predicted variable is predicted over different values of rural beds.



There are a few points on the residual plot which are on the negative quadrant of the plot.

Results

After running the descriptive statistics across different datasheets, we could identify the summaries about the sample and the measures across the independent variable i.e. the Positive Cases and every other dependent variable considered. Together with simple graphics analysis, they form the basis of virtually every quantitative analysis of data. We could identify that the number of cases has been increasing at an alarming rate across all the states of India with certain states like Maharashtra and Uttar Pradesh facing a crisis. In terms of testing certain states have been prolific whereas certain north eastern states like Meghalaya, Mizoram and Nagaland haven't been carrying out the tests extensively.

Further the correlation analysis helped us quantify the degree to which the dependent variables are related to the independent variable. From our analysis we found that with the rising number of positive cases certain verticals of healthcare infrastructure in India are not well equipped to accommodate the patients in a dire situation as the numbers don't seem convincing considering the rate at which the number of positive cases are increasing. We found that the public healthcare infrastructure is not satisfactory. In the Urban areas the presence of Private Health care Infrastructure resolves the crisis to certain extent but in the rural areas there lies a genuine crisis. Both the number of district hospitals as well as the number of rural hospitals share a weak negative correlation with the positive cases. This further has a direct impact on the availability of rural beds as well and hence we do not see a strong positive correlation between rural beds and Positive cases.

Further, The P-value obtained from ANOVA analysis is significant ($P<0.05$), and therefore, we conclude that there are significant differences among treatments and hence we reject the

Null Hypothesis i.e. we have enough hospital beds available across different regions and categories of Hospitals in India. After running different statistical analysis tests across the datasets, we can conclude that the dataset has a high statistical significance to our target variable i.e. the number of rural beds, with a statistical significance of 88%.

Recommendation

As per the analysis of data and considering the rampant transmission of Covid-19 these are the few recommendations:

a) **Escalated Dynamic Covid-19 Testing in the Rural areas and smaller states of India:** After the analysis of the dataset we can comprehend that the rural areas and smaller states of India probably lacks the infrastructure to carry out mass testing for Covid-19. The testing across these regions of India should be made easily accessible along with the diligent efforts on creating awareness among people.

b) **Create Safe havens and containment zones in Urban areas and areas with high population density:** In certain states and area India which has high population density, the concerned authorities should utilize the government real estate assets to accommodate the sick and elderly. With most of families cramped up in smaller homes in India, there lingers a higher probability of the virus transmission.

c) **Installation of Public Sanitizing Booths at places with high footfalls**: Places like railway stations and Bus Terminals across the urban and rural areas must have multiple sanitizing booths for commuters to sanitize themselves and curb the spread of the virus.

d) **Mandatory face covering ordinance where a violation may lead to fine:** Rigid regulations on face covering should be laid out on areas with higher transmission rates like Maharashtra, Uttar Pradesh, Karnataka, Tamil Nadu and Andhra Pradesh. For individuals to take this seriously, a fine on violation must also be levied.

e) **Collaboration of Government Health Care System with Private Health Care bodies:** For ideal execution and access to greater resource pool the government should look forward to collaborating with Private bodies in the Health Care industry. This may include reserving certain number of beds in the Private Health Care centres for the Covid-19 patients if required and medical staff making themselves available of philanthropic grounds.

f) **Invest astutely on uplifting the Rural Health Care Infrastructure:** The Covid-19 in India possess a huge threat to the rural population as the number of hospitals as well as beds available are not well proportionate to the population of the country, with 0.55 beds per 1000 population. The elderly population (aged 60 and above) is especially vulnerable, given more complications which are reported for patients in this age group. The availability of beds for elderly population in India is 5.18 beds per 1000 population. It is time the government embraces its vulnerability and works on laying a strong health care infrastructure with more hospitals and beds in rural areas.

References

World Health Organization. (2020, April 20). Q&A on coronaviruses. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/q-a-coronaviruses

Arcadia Instant 4.0.0.(2017). Correlation Heatmap Visuals.

http://docs.arcadiadata.com/4.0.0/pages/topics/visualcorrelation.html#:~:text=The%20color%20value%20of%20the,both%20compare%20exactly%20two%20dimensions.

Rumsey, D. How to Interpret a Correlation Coefficient. https://www.dummies.com/education/math/statistics/how-to-interpret-a-correlation-coefficient-r/

Moreno, A. (2019, July 27) Simple and multiple linear regression with Python. (https://towardsdatascience.com/simple-and-multiple-linear-regression-with-python-c9ab422ec29c

NCSS, LLC. Ridge Regression https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Ridge_Regression.pdf

Singh, P & Ravi, S. (2020, March 24). COVID-19 | Is India's health infrastructure equipped to handle an epidemic? https://www.brookings.edu/blog/up-front/2020/03/24/is-indias-health-infrastructure-equipped-to-handle-an-epidemic/

Coding:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler,PolynomialFeatures
from sklearn.linear_model import LinearRegression
%matplotlib inline
```

```
#df = pd.read_excel(io=file_name, sheet_name=sheet)
filename = "Covid-19 India data set (version 1).xlsx"
sheet1 = 1
sheet2 = 2
sheet3 = 3
sheet4 = 4
#Covid 19 Cases in India
covid_19=pd.read_excel(io=filename,  sheet_name=sheet1)

#Hopital Beds availablity data in India
beds = pd.read_excel(io=filename,  sheet_name=sheet2)

#India's Population
pop = pd.read_excel(io=filename,  sheet_name=sheet3)

#Statewise testing data
testing = pd.read_excel(io=filename,  sheet_name=sheet4)
```

```
covid_19.head()
```

```
beds.head(36)
```

```
pop.head(36).sort_values(by = ['State / Union Territory'], axis=0, ascending=True)
```

```
covid_19.isnull().sum()
```

```
Sno                         1
Date                        1
Time                        1
State/UnionTerritory        1
ConfirmedIndianNational     1
ConfirmedForeignNational    1
Cured                       1
Deaths                      1
Confirmed                   0
dtype: int64
```

```
beds.isnull().sum()
```

```
Sno                                 0
State/UT                            0
NumPrimaryHealthCenters_HMIS        0
NumCommunityHealthCenters_HMIS      0
NumSubDistrictHospitals_HMIS        7
NumDistrictHospitals_HMIS           0
TotalPublicHealthFacilities_HMIS    0
NumPublicBeds_HMIS                  0
NumRuralHospitals_NHP18             0
NumRuralBeds_NHP18                  0
NumUrbanHospitals_NHP18             0
NumUrbanBeds_NHP18                  0
dtype: int64
```

```
pop.isnull().sum()
```

```
Sno                         0
State / Union Territory     0
Population                  0
Rural population            0
Urban population            0
Area                        0
Density                     0
Gender Ratio                0
dtype: int64
```

```
testing.isnull().sum()
```

```
Date            0
State           0
TotalSamples    0
Positive       14
Negative        1
dtype: int64
```

```
print("number of NaN values for the column:", beds['NumSubDistrictHospitals_HMIS'].isnull().sum())
```

```
number of NaN values for the column: 7
```

```
mean1 = beds['NumSubDistrictHospitals_HMIS'].mean()
beds['NumSubDistrictHospitals_HMIS'].replace(np.nan, mean1, inplace = True)
```

```
print("number of NaN values for the column:", beds['NumSubDistrictHospitals_HMIS'].isnull().sum())
```

```
number of NaN values for the column: 0
```

```
mean2 = testing['Positive'].mean()
testing['Positive'].replace(np.nan, mean2, inplace = True)
```

```
print("number of NaN values for the column:", testing['Positive'].isnull().sum())
```

```
number of NaN values for the column: 0
```

```
mean3 = testing['Negative'].mean()
testing['Negative'].replace(np.nan, mean3, inplace = True)
```

```
print("number of NaN values for the column:", testing['Negative'].isnull().sum())
```

```
number of NaN values for the column: 0
```

```
testing['State'].value_counts()
```

```
testing['State'].value_counts()
```

```
covid_19.describe()
```

```
beds.describe()
```

```
pop.describe()
```

testing.describe()

```
covid_19.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4847 entries, 0 to 4846
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Sno                       4846 non-null   float64
 1   Date                      4846 non-null   datetime64[ns]
 2   Time                      4846 non-null   object
 3   State/UnionTerritory      4846 non-null   object
 4   ConfirmedIndianNational   4846 non-null   object
 5   ConfirmedForeignNational  4846 non-null   object
 6   Cured                     4846 non-null   float64
 7   Deaths                    4846 non-null   float64
 8   Confirmed                 4847 non-null   int64
dtypes: datetime64[ns](1), float64(3), int64(1), object(4)
memory usage: 340.9+ KB
```

```
beds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 12 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Sno                               36 non-null     int64
 1   State/UT                          36 non-null     object
 2   NumPrimaryHealthCenters_HMIS      36 non-null     int64
 3   NumCommunityHealthCenters_HMIS    36 non-null     int64
 4   NumSubDistrictHospitals_HMIS      36 non-null     float64
 5   NumDistrictHospitals_HMIS         36 non-null     int64
 6   TotalPublicHealthFacilities_HMIS  36 non-null     int64
 7   NumPublicBeds_HMIS                36 non-null     int64
 8   NumRuralHospitals_NHP18           36 non-null     int64
 9   NumRuralBeds_NHP18                36 non-null     int64
 10  NumUrbanHospitals_NHP18           36 non-null     int64
 11  NumUrbanBeds_NHP18                36 non-null     int64
dtypes: float64(1), int64(10), object(1)
memory usage: 3.5+ KB
```

```
pop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 8 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Sno                     36 non-null     int64
 1   State / Union Territory  36 non-null     object
 2   Population              36 non-null     int64
 3   Rural population        36 non-null     int64
 4   Urban population        36 non-null     int64
 5   Area                    36 non-null     object
 6   Density                 36 non-null     object
 7   Gender Ratio            36 non-null     int64
dtypes: int64(5), object(3)
memory usage: 2.4+ KB
```

```
testing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3780 entries, 0 to 3779
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Date          3780 non-null   datetime64[ns]
 1   State         3780 non-null   object
 2   TotalSamples  3780 non-null   int64
 3   Positive      3780 non-null   float64
 4   Negative      3780 non-null   float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(1)
memory usage: 147.8+ KB
```

```
covid_19_r = covid_19.drop(['Sno'], axis=1)

#Hopital Beds availablity data in India
bed_r = beds.drop(['Sno'], axis=1).sort_values(by = ['State/UT'])

#India's Population
pop_r = pop.drop(['Sno'], axis=1).sort_values(by = ['State / Union Territory'])

#Statewise testing data
```

```
covid_19_r.hist(figsize=(12,12))
plt.show()
```

```
bed_r.hist(figsize=(12,12))
plt.show()
```

```
pop_r.hist(figsize=(12,12))
plt.show()
```

```
values1 = bed_r[['State/UT', 'NumPrimaryHealthCenters_HMIS', 'NumCommunityHealthCenters_HMIS', 'NumSubDistrictHospitals_HMIS',
'NumDistrictHospitals_HMIS', 'TotalPublicHealthFacilities_HMIS', 'NumPublicBeds_HMIS' ,'NumRuralHospitals_NHP18', 'NumRuralBeds_
NHP18', 'NumUrbanHospitals_NHP18','NumUrbanBeds_NHP18' ]]
values2 = pop_r[['State / Union Territory', 'Population', 'Rural population', 'Urban population','Area', 'Density', 'Gender Rati
o']]
```

```
df_covid = pd.concat([values1, values2])
```

```
#df_covid.to_excel('covid_19_consolidated.xlsx')
```

```
covid_BedVsPopVsCases = pd.read_excel('covid_19_consolidated.xlsx')
```

```
covid_BedVsPopVsCases=covid_BedVsPopVsCases.drop(['Unnamed: 0'], axis=1)
```

```
covid_BedVsPopVsCases.head()
```

```
covid_BedVsPopVsCases.corr()
```

```
sns.heatmap(covid_BedVsPopVsCases.corr())
```

```
sns.regplot("Population", "Positive Cases", data=covid_BedVsPopVsCases)
```

```
# Scatter plot of Positive Cases and Number of Public Beds

ax1 = covid_BedVsPopVsCases.plot(kind='scatter', x='Population', y='NumPublicBeds_HMIS', color='blue', alpha=0.5, figsize=(10, 7)

covid_fit = np.polyfit(covid_BedVsPopVsCases.Population, covid_BedVsPopVsCases.NumPublicBeds_HMIS, 1)

#plt.plot(x, y, 'o')
plt.plot(covid_BedVsPopVsCases.Population, covid_fit[0] * covid_BedVsPopVsCases.Population + covid_fit[1], color='darkblue', line
#create scatter plot


m, b = np.polyfit(covid_BedVsPopVsCases.Population, covid_BedVsPopVsCases.NumPublicBeds_HMIS, 1)
#m = slope, b=intercept
```

```
sns.regplot("Positive Cases", "Population", data=covid_BedVsPopVsCases)
```

```
sns.regplot("Positive Cases", "NumRuralBeds_NHP18", data=covid_BedVsPopVsCases)
```

```
for val in covid_BedVsPopVsCases:
    print(val)
```

```
State/UT
Total Samples
Positive Cases
Negative Cases
NumPrimaryHealthCenters_HMIS
NumCommunityHealthCenters_HMIS
NumSubDistrictHospitals_HMIS
NumDistrictHospitals_HMIS
TotalPublicHealthFacilities_HMIS
NumPublicBeds_HMIS
NumRuralHospitals_NHP18
NumRuralBeds_NHP18
NumUrbanHospitals_NHP18
NumUrbanBeds_NHP18
Population
Rural population
Urban population
Gender Ratio
```

```
for val in covid_BedVsPopVsCases:
    print(f"covid_BedVsPopVsCases['{val}'],")
```

```
covid_BedVsPopVsCases['State/UT'],
covid_BedVsPopVsCases['Total Samples'],
covid_BedVsPopVsCases['Positive Cases'],
covid_BedVsPopVsCases['Negative Cases'],
covid_BedVsPopVsCases['NumPrimaryHealthCenters_HMIS'],
covid_BedVsPopVsCases['NumCommunityHealthCenters_HMIS'],
covid_BedVsPopVsCases['NumSubDistrictHospitals_HMIS'],
covid_BedVsPopVsCases['NumDistrictHospitals_HMIS'],
covid_BedVsPopVsCases['TotalPublicHealthFacilities_HMIS'],
covid_BedVsPopVsCases['NumPublicBeds_HMIS'],
covid_BedVsPopVsCases['NumRuralHospitals_NHP18'],
covid_BedVsPopVsCases['NumRuralBeds_NHP18'],
covid_BedVsPopVsCases['NumUrbanHospitals_NHP18'],
covid_BedVsPopVsCases['NumUrbanBeds_NHP18'],
covid_BedVsPopVsCases['Population'],
covid_BedVsPopVsCases['Rural population'],
covid_BedVsPopVsCases['Urban population'],
covid_BedVsPopVsCases['Gender Ratio'],
```

```
import scipy.stats as stats
# stats f_oneway functions takes the groups as input and returns F and P-value
fvalue, pvalue = stats.f_oneway(covid_BedVsPopVsCases['Total Samples'], covid_BedVsPopVsCases['Positive Cases'], covid_BedVsPopVs
print("The value from the Stats is given below:")
print("The F value is:", fvalue," The Pvalue is:", pvalue)
```

```
The value from the Stats is given below:
The F value is: 22.647575259851166  The Pvalue is: 6.832450353949321e-12
```

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
# reshape the d dataframe suitable for statsmodels package
d_melt = pd.melt(covid_BedVsPopVsCases.reset_index(), id_vars=['index'], value_vars=['Total Samples', 'Positive Cases', 'Urban
# replace column names
d_melt.columns = ['index', 'treatments', 'value']
# Ordinary Least Squares (OLS) model
model = ols('value ~ C(treatments)', data=d_melt).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
print("\n\t\tAnova table:")
anova_table
```

Anova table:

	sum_sq	df	F	PR(>F)
C(treatments)	3.001662e+15	3.0	22.647575	6.832450e-12
Residual	5.831668e+15	132.0	NaN	NaN

Interpretation: The P-value obtained from ANOVA analysis is significant (P<0.05), and therefore, we conclude that there are significant differences among treatments.

```
sns.boxplot('Positive Cases', data = covid_BedVsPopVsCases)
```

```
sns.boxplot('NumRuralBeds_NHP18', data = covid_BedVsPopVsCases)
```

```
sns.boxplot('', data = covid_BedVsPopVsCases)
```

```
covid_BedVsPopVsCases.columns
```

```
Index(['State/UT', 'Total Samples', 'Positive Cases', 'Negative Cases',
       'NumPrimaryHealthCenters_HMIS', 'NumCommunityHealthCenters_HMIS',
       'NumSubDistrictHospitals_HMIS', 'NumDistrictHospitals_HMIS',
       'TotalPublicHealthFacilities_HMIS', 'NumPublicBeds_HMIS',
       'NumRuralHospitals_NHP18', 'NumRuralBeds_NHP18',
       'NumUrbanHospitals_NHP18', 'NumUrbanBeds_NHP18', 'Population',
       'Rural population', 'Urban population', 'Gender Ratio'],
      dtype='object')
```

```
significant_dict = {

 # significant variables only
 'significant_columns' : ['Positive Cases','Negative Cases', 'NumCommunityHealthCenters_HMIS', 'Num
        'NumDistrictHospitals_HMIS']

}
```

```
covid_data   =  covid_BedVsPopVsCases.loc[ : , significant_dict['significant_columns']]
covid_target =  covid_BedVsPopVsCases.loc[ : , 'NumRuralHospitals_NHP18']
```

```
#Using Scikit Learn Model
from sklearn.linear_model import Ridge
from yellowbrick.regressor import ResidualsPlot
from sklearn.model_selection import train_test_split
# train/test split
X_train, X_test, y_train, y_test = train_test_split(
            covid_data,
            covid_target,
            random_state =890,
            test_size    = 0.30)
# Instantiate the linear model and visualizer
model = Ridge()
visualizer = ResidualsPlot(model)

visualizer.fit(X_train, y_train)  # Fit the training data to the visualizer
visualizer.score(X_test, y_test)  # Evaluate the model on the test data
visualizer.show()                 # Finalize and render the figure
```