NATIONAL RURAL HEALTH MISSION

IT 495 EXPLORATORY DATA ANALYSIS

SEMESTER II

INSTRUCTOR: DR. GOPINATH PANDA

GROUP NO: 2

NAKUL TOMAR
202218026

SHASHVA MACHCHHAR
202218028

AYUSH JAIN
202218039

HARSHIT SHAH
202218040

Declaration

We, the undersigned, declare that this project titled **"National Rural Health Mission"** is our original work and that all information contained herein has been properly cited and referenced. We affirm that we have not engaged in any form of academic dishonesty, including but not limited to plagiarism or falsification of data. This project has not been submitted, in part or in whole, for any other academic purpose and all sources of information used in this project have been identified in the references section. We understand that any breach of academic integrity may result in severe Penalties, including revocation of our degrees.

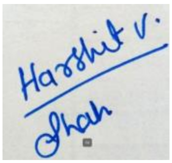
Additionally, we confirm that each member of the group contributed equally to the completion of the project. Each member participated in this project's planning, research, analysis, and writing. We also communicated and collaborated effectively throughout the project, ensuring that each member's contributions were properly integrated.

Date: 02-May-2023

NAKUL TOMAR
202218026

SHASHVA MACHCHHAR
202218028

AYUSH JAIN
202218039



HARSHIT SHAH
202218040

Certificate

This is to certify that the group project report titled **“National Rural Health Mission”** submitted by **Group II** members **NAKUL TOMAR, SHASHVA MACHCHHAR, AYUSH JAIN** and **HARSHIT SHAH** was original work and was completed under our supervision.

We confirm that this project report is the result of the original work of this group and that all sources of information users have been properly acknowledged with appropriate citations and references.

Every member of the team participated equally in the planning, research, analysis, and writing of this report and we recommend it for evaluation and examination with confidence as it meets the highest standards of academic excellence.

Signature of project supervisor

Date: 02-May-2023

Contents

List of Figures:

Introduction:

The National Rural Health Mission (NRHM) was launched by the Hon'ble Prime Minister on 12th April 2005, to provide accessible, affordable, and quality health care to the rural population, especially the vulnerable groups. The Union Cabinet vide its decision dated 1st May 2013, has approved the launch of National Urban Health Mission (NUHM) as a Sub-mission of an over-arching National Health Mission (NHM), with National Rural Health Mission (NRHM) being the other Sub-mission of National Health Mission.

NRHM seeks to provide equitable, affordable, and quality health care to the rural population, especially the vulnerable groups. Under the NRHM, the Empowered Action Group (EAG) States, as well as the North-eastern States, Jammu and Kashmir, and Himachal Pradesh, have been given special focus. The thrust of the mission is on establishing a fully functional, community-owned, decentralized health delivery system with inter-sectoral convergence at all levels, to ensure simultaneous action on a wide range of determinants of health such as water, sanitation, education, nutrition, social and gender equality. Institutional integration within the fragmented health sector was expected to provide a focus on outcomes, measured against Indian Public Health Standards for all health facilities.

In the case of the National Rural Health Mission (NRHM), EDA can help us to gain insights into the health infrastructure and services in rural areas and the impact of the mission on the population. It can also provide insights into the demographic and socio-economic characteristics of the rural population and their impact on health outcomes. In this project we have tried to perform EDA on NRHM for this we have taken from official government website data.gov.in and perform EDA on various sectors of health sector which are part of NHRM.

Tools and Technologies:

For our analysis we have used Python and following python libraries:

- Pandas
- Matplotlib
- Seaborn
- NumPy
- Plotly Express
- Ipywidgets
- Scikit Learn
- JSON

Methodology:

For our analysis we couldn't find a single dataset large enough to perform EDA therefore we decided to perform EDA on several small datasets. We collected datasets from data.gov.in website. Then we followed step by step procedure for each data set:

- Collect and load the data.
- Data Description.
- Data Cleaning.
- Data Pre-processing.
- Summarizing the Data.
- Visualization.
- Predictions (if possible).
- Interpretations

Exploratory Data Analysis (EDA):

1. National Health Mission Budget/Funds: 2015-16 to 2022-23 Dataset :

We start by loading the csv file.

```
NHM_df = pd.read_csv(r'Datasets\nhm---july-2022.csv')

NHM_df.head()
```

	State_UT	State_UT_Code	Fiscal Year	Budget Proposed by the States/UTs	Budget Approved for the States/UTs	Opening Balance with the States/UTs	Release of Government of India's Fund	Total Expenditure Reported (Including States' Share)	Extent of Budget Approved Against Budget Proposed	Extent of Funds Utilised Against Budget Approved	...
0	Andhra Pradesh	1.0	2015-16	2380.02	1336.13	NaN	659.04	1105.70	56.14	82.75	...
1	Arunachal Pradesh	2.0	2015-16	346.46	195.94	NaN	163.80	147.41	56.55	75.23	...
2	Assam	3.0	2015-16	2275.32	1853.40	NaN	997.59	1212.25	81.46	65.41	...
3	Bihar	4.0	2015-16	3874.98	2672.45	NaN	1269.67	1731.85	68.97	64.80	...
4	Chhattisgarh	5.0	2015-16	1528.15	1215.09	NaN	423.31	769.33	79.51	63.31	...

5 rows × 24 columns

Then, to get insights about the data we used .info() method which gives information about total number of rows and columns, name of each column, and datatype of each column.

```
NHM_df.info()
```

```
Output exceeds the size limit. Open the full output data in a text editor
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 962 entries, 0 to 961
Data columns (total 24 columns):
 #   Column                                              Non-Null Count  Dtype  
---  ------                                              --------------  -----  
 0   State_UT                                            296 non-null    object 
 1   State_UT_Code                                       296 non-null    float64
 2   Fiscal Year                                         296 non-null    object 
 3   Budget Proposed by the States/UTs                   282 non-null    float64
 4   Budget Approved for the States/UTs                  281 non-null    float64
 5   Opening Balance with the States/UTs                 180 non-null    float64
 6   Release of Government of India's Fund               289 non-null    float64
 7   Total Expenditure Reported (Including States' Share) 289 non-null    float64
 8   Extent of Budget Approved Against Budget Proposed   279 non-null    float64
 9   Extent of Funds Utilised Against Budget Approved    279 non-null    float64
 10  Extent of Funds Utilised Against Budget Proposed    280 non-null    float64
 11  Unnamed: 11                                         0 non-null      float64
 12  Unnamed: 12                                         0 non-null      float64
 13  Unnamed: 13                                         0 non-null      float64
 14  Unnamed: 14                                         0 non-null      float64
 15  Unnamed: 15                                         0 non-null      float64
 16  Unnamed: 16                                         0 non-null      float64
 17  Unnamed: 17                                         0 non-null      float64
 18  Unnamed: 18                                         0 non-null      float64
 19  Unnamed: 19                                         0 non-null      float64
...
 22  Unnamed: 22                                         0 non-null      float64
 23  Unnamed: 23                                         0 non-null      float64
dtypes: float64(22), object(2)
memory usage: 180.5+ KB
```

From the above output, we see that our dataset contains 962 rows and 24 columns, but we also see that there are many columns which have only null values. Therefore, we decided to drop these columns.

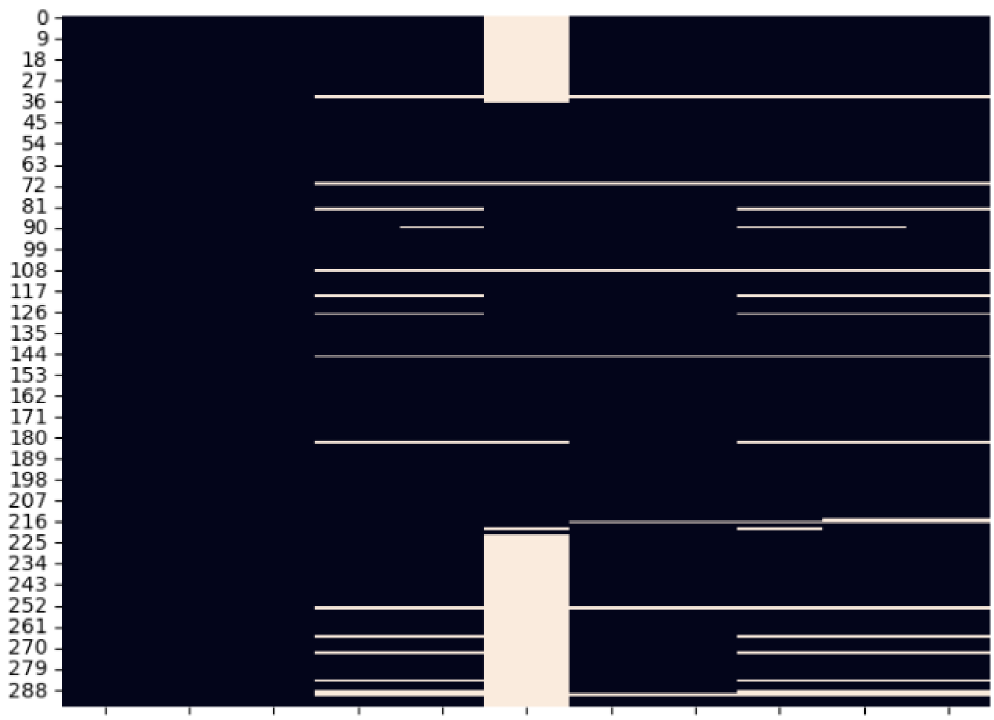
```
NHM_df = NHM_df.drop(['Unnamed: 11',
                      'Unnamed: 12', 'Unnamed: 13', 'Unnamed: 14', 'Unnamed: 15',
                      'Unnamed: 16', 'Unnamed: 17', 'Unnamed: 18', 'Unnamed: 19',
                      'Unnamed: 20', 'Unnamed: 21', 'Unnamed: 22', 'Unnamed: 23'],axis=1)
```

Now, to visualize how many null values are there in each row we plot a heatmap.

```
plt.figure(figsize=(8,6))
sns.heatmap(NHM_df.isnull(), cbar=False)
plt.show()
```

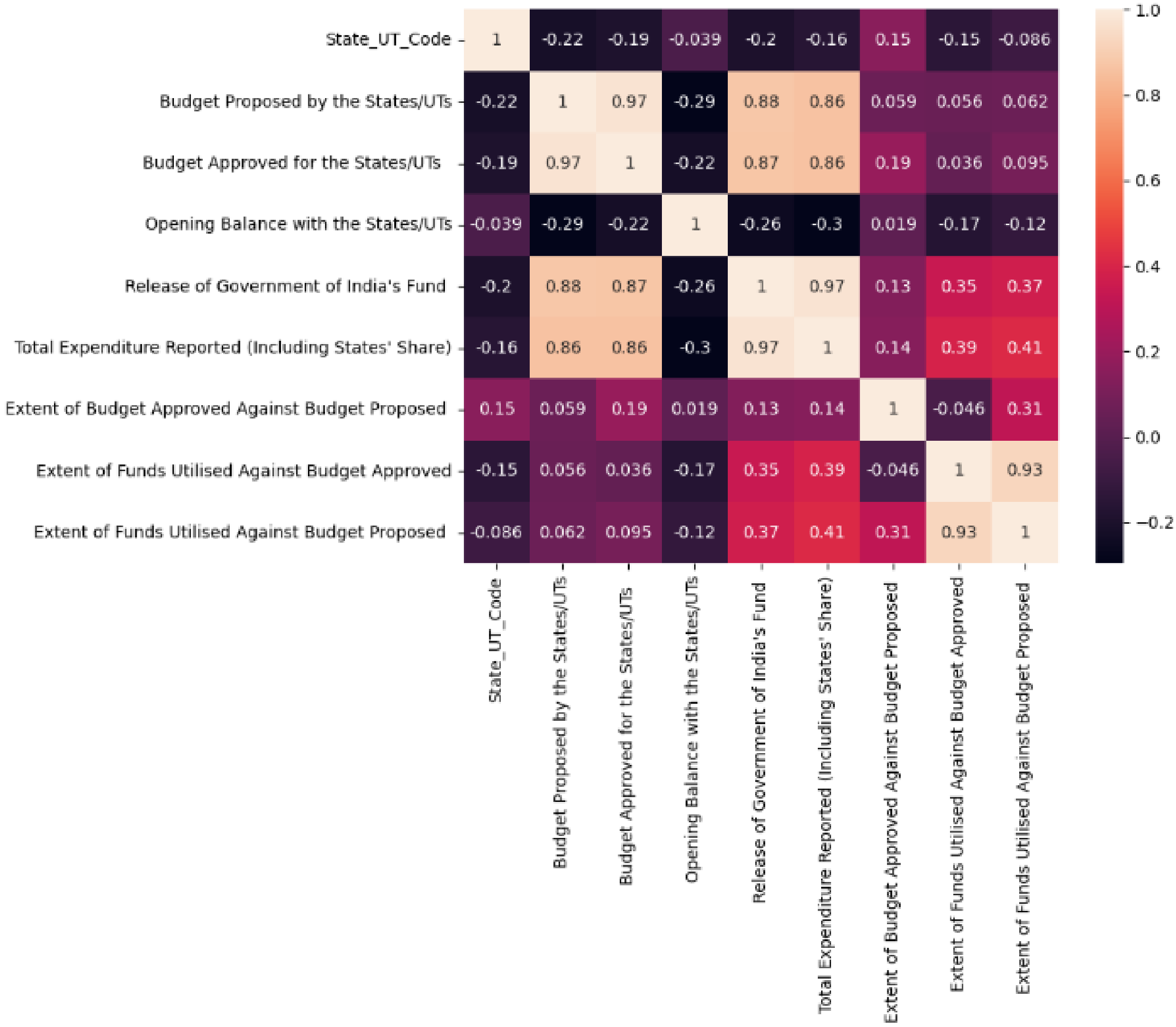
Since there are null values in all the columns, we can discard those values without any data loss. Opening Balance with the States/UTs is not available for the fiscal year 2015-16 , 2021-22 and 2022-23 but values for other years between these are available, thus we skip dropping this Null values.

```
# Dropping rows where all the column values are null
NHM_df = NHM_df.dropna(axis=0, how='all')

plt.figure(figsize=(8,6))
sns.heatmap(NHM_df.isna(), cbar=False)
plt.show()
```



After removing the null values from the rows and columns we try find the correlation between all the columns.

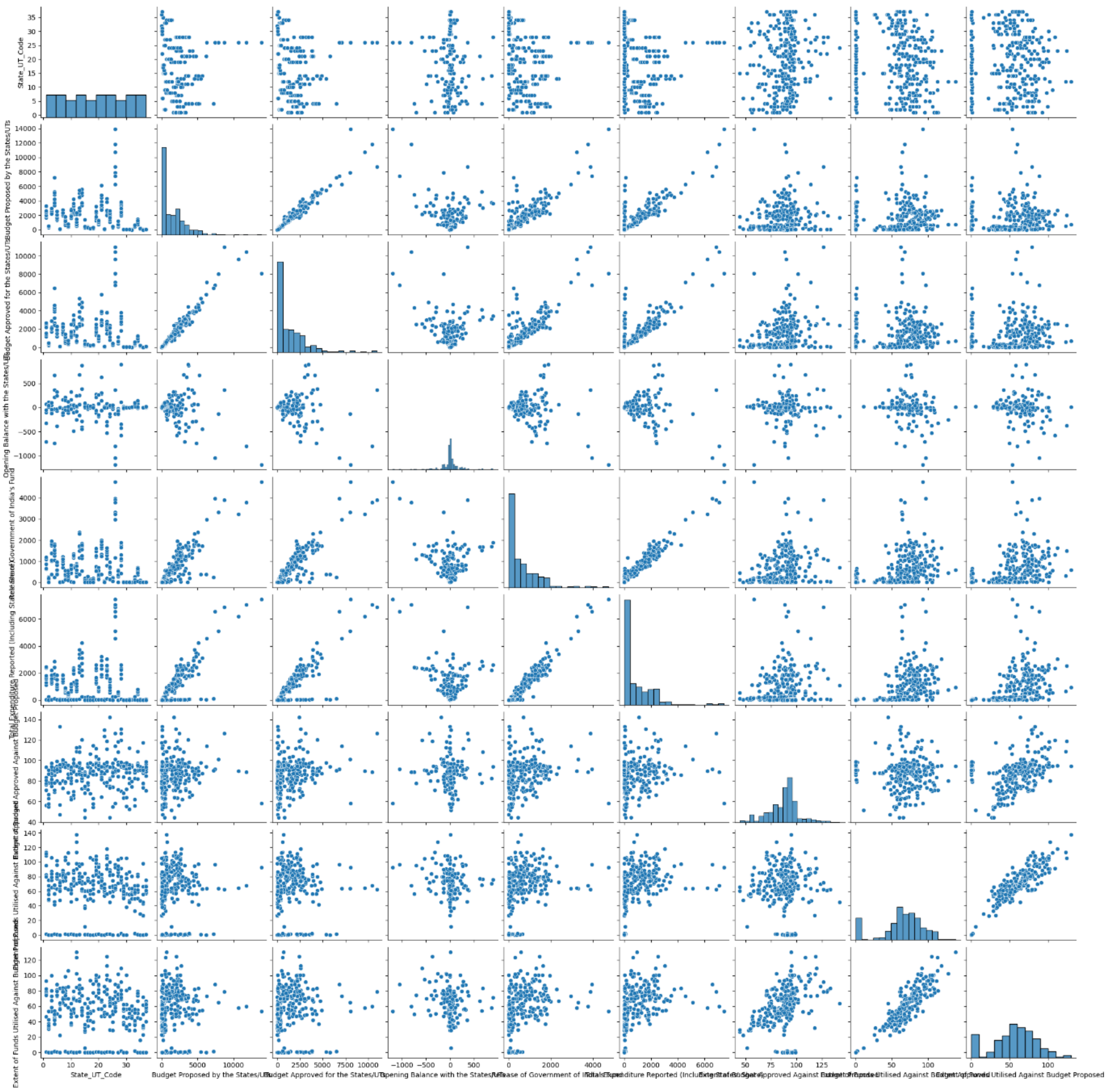
```
plt.figure(figsize=(8,6))
sns.heatmap(NHM_df[NHM_df.columns[NHM_df.dtypes == 'float64']].corr(), annot=True)
plt.show()
```



Correlation heatmap between the features show that how each column is related to each other. From the graph it is evident that features like Budget Proposed and Budget Allocated are related to each other. Similarly Release of government of India fund is highly correlated to the expenditure.

Also, Extent of funds utilised against Budget approved is highly related to Extent of funds utilised against Budget Proposed.

```
sns.pairplot(NHM_df)
plt.show()
```

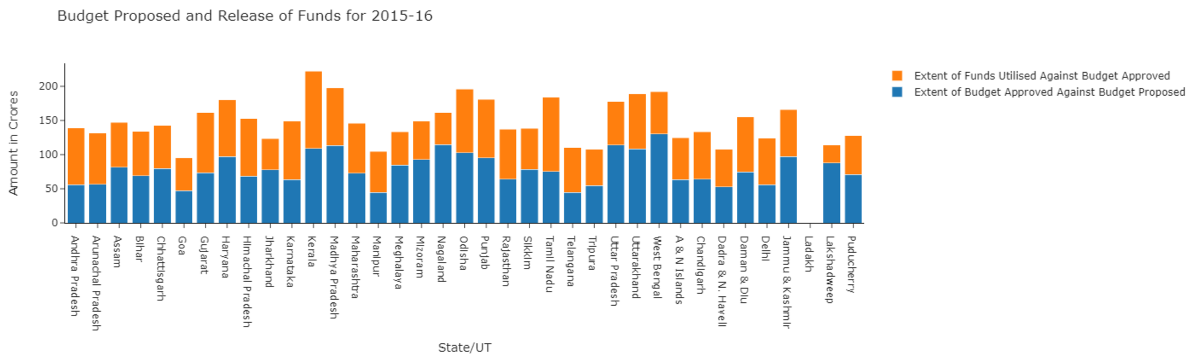


Scatter Pair plot gives an idea about each features relationship with other features. Trend that is followed by a feature by change in value other features are captured through it.

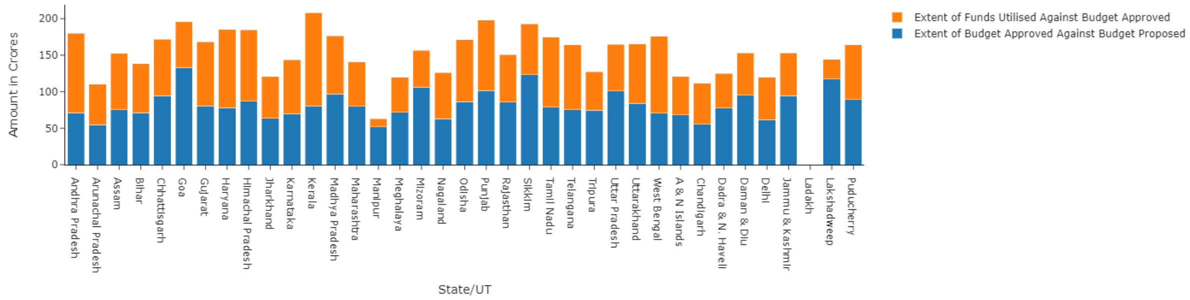
We have made this just to find relationship between features so that we can further apply Linear Regression Model to the ones that share linear relationship.

We plotted some stacked bar chart using Plotly library to visualize how much budget was approved and how much was utilized in different years. Below is the code for doing the same:

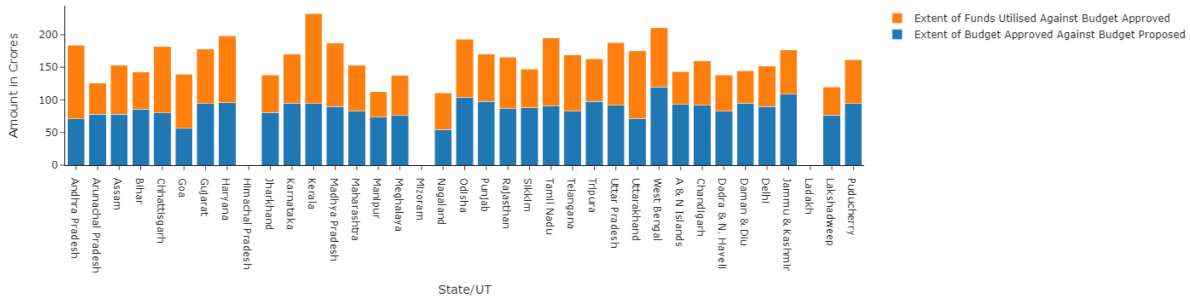
```
df = NHM_df.copy()

for year in df["Fiscal Year"].unique():
    df_year = df[df["Fiscal Year"] == year]

    trace1 = go.Bar(
        x=df_year["State_UT"],
        y=df_year["Extent of Budget Approved Against Budget Proposed "],
        name="Extent of Budget Approved Against Budget Proposed"
    )

    trace2 = go.Bar(
        x=df_year["State_UT"],
        y=df_year["Extent of Funds Utilised Against Budget Approved"],
        name="Extent of Funds Utilised Against Budget Approved"
    )

    data = [trace1, trace2]
    layout = go.Layout(
        barmode="stack",
        title=f"Budget Proposed and Release of Funds for {year}",
        xaxis=dict(title="State/UT"),
        yaxis=dict(title="Amount in Crores")
    )
    fig = go.Figure(data=data, layout=layout)

    fig.show()
```



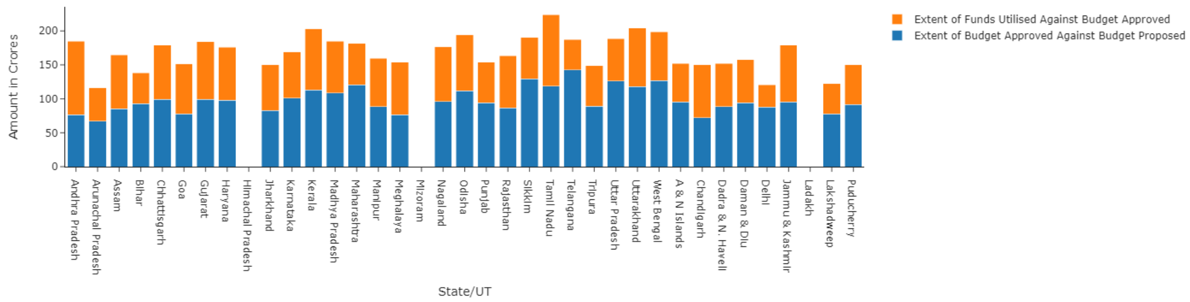
Budget Proposed and Release of Funds for 2016-17



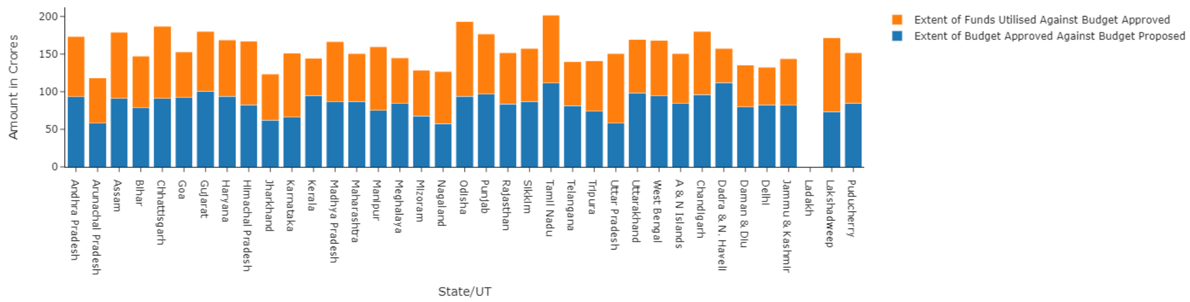
Budget Proposed and Release of Funds for 2017-18

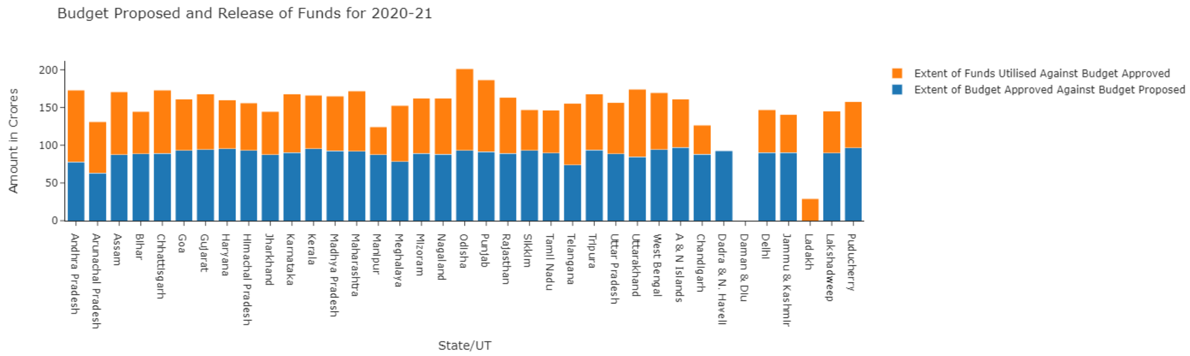


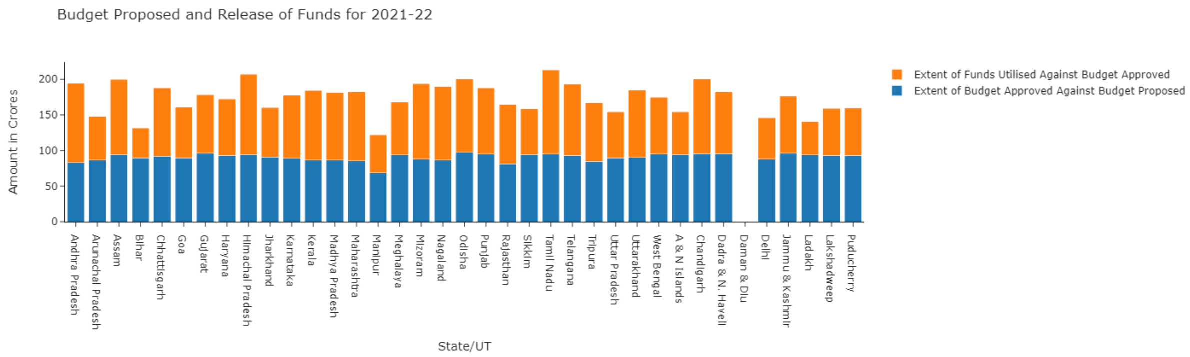
Budget Proposed and Release of Funds for 2018-19

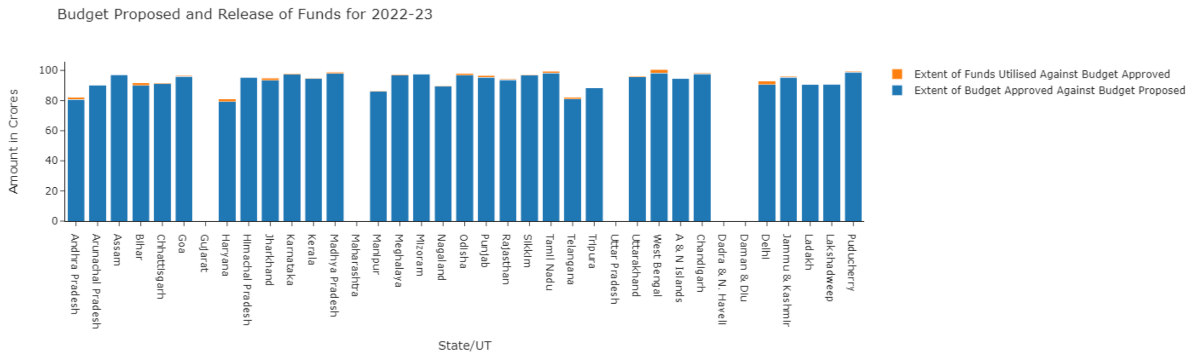


Budget Proposed and Release of Funds for 2019-20









From the chart we can see that the amount of funding that states have spent on NRHM has generally increased over the years. In some years, the amount of funds used by certain states was significantly higher, while in other years the increased funds were distributed more evenly among the different states. We can also observe that some states consistently use more funds for NRHM than others, such as Uttar Pradesh, which has the highest threshold in most years. In most states, the total budget proposed/allocated by the government increases year by year. This can be seen in the upward movement of the bars.

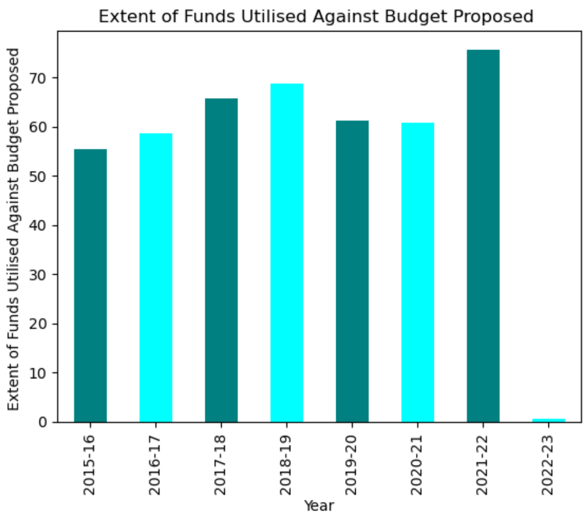
Over the years, Uttar Pradesh, Maharashtra, Karnataka, and Gujarat have had the highest combined budgets. Sikkim, Nagaland, and Mizoram have had the lowest combined budgets for many years. Many state governments have proposed substantial increases in total budgets for 2019-20. This may be due to various factors, such as the implementation of new policies or the allocation of funds for specific projects.

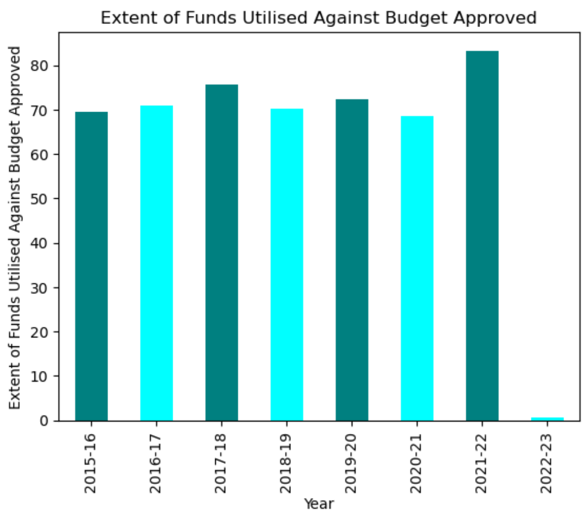
The total budget proposed by Andhra Pradesh from 2016-17 to 2018-19 has been drastically reduced. This could be due to factors such as a change of government or a change in policy.

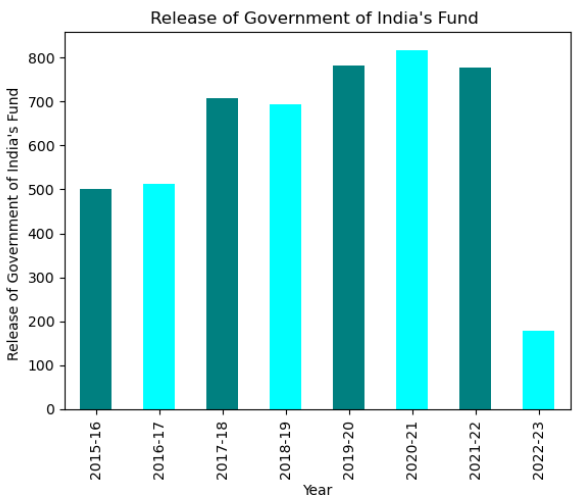
The bar graph shows that the total budget proposed by the government has generally increased over the years, although there has been some fluctuation and variation between states and years.

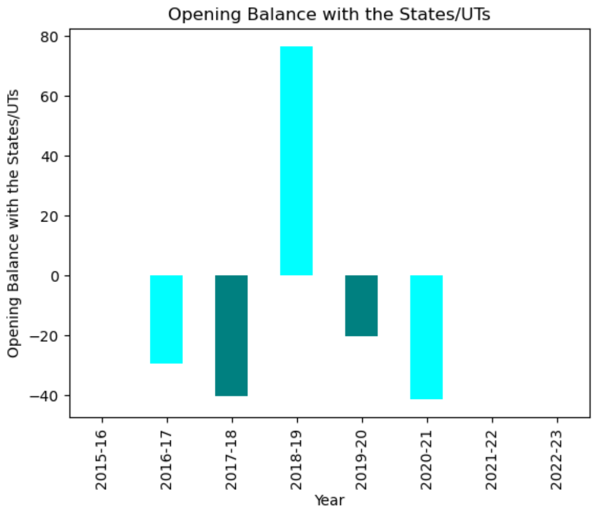
Now to find the change in different stages of Budget such as Proposal, approving etc. we will plot the moving average based on different years.

```
for i in NHM_df.columns[3:]:
    df_yearly = NHM_df.groupby('Fiscal Year')[i].mean().reset_index()

    # create a bar chart
    ax = df_yearly.plot(x='Fiscal Year', y=i, kind='bar',
                        title=i,
                        color=['teal', 'cyan'], legend=False)

    # update axis labels
    ax.set_xlabel('Year')
    ax.set_ylabel(i)

    # show the plot
    plt.show()
```



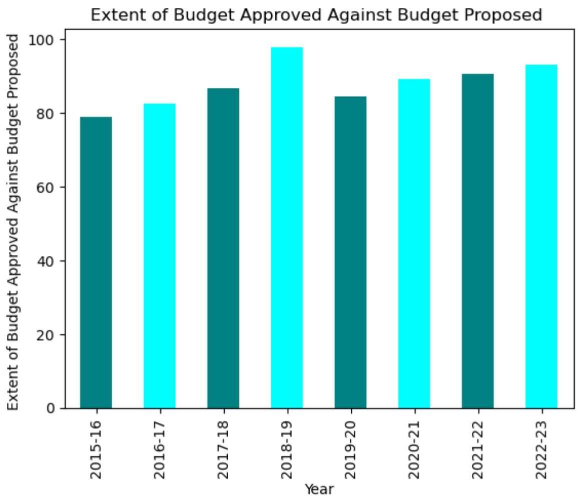


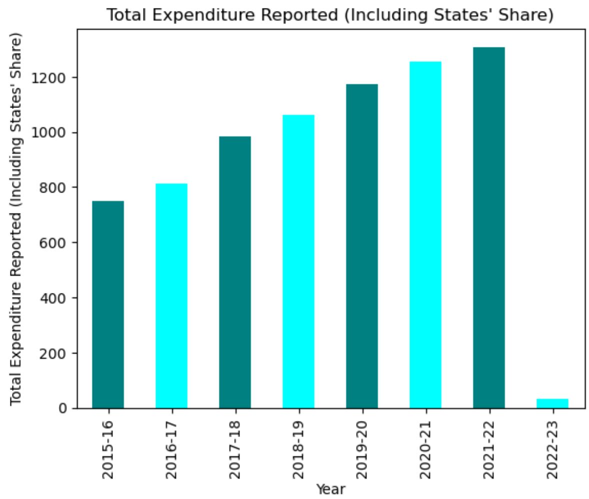




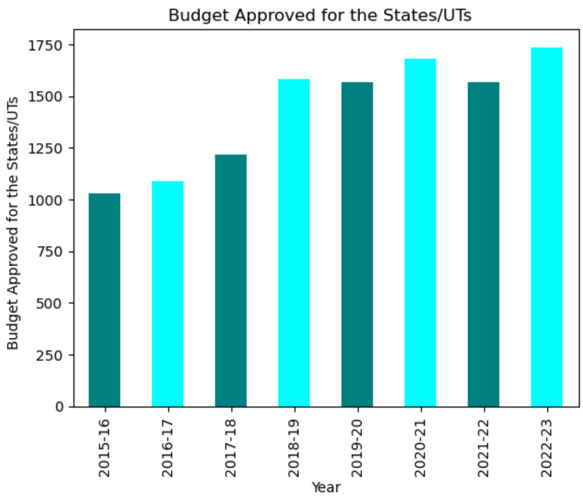
After the year 2016-17, there was a significant spike in funding which continued to increase in the following years as well. However, there seems to be a slight dip in funding after the year 2020-21. Overall, the trend seems to be positive and indicative of the government's increasing focus on improving rural health infrastructure.

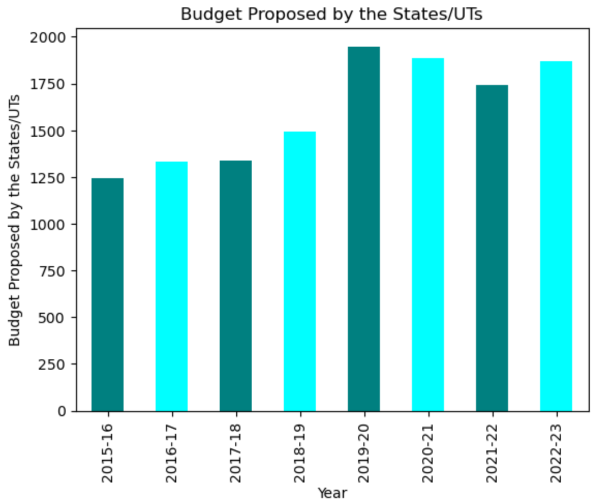
The state government often began the year with a budget deficit, as they received less funding than what they had proposed for in most years.





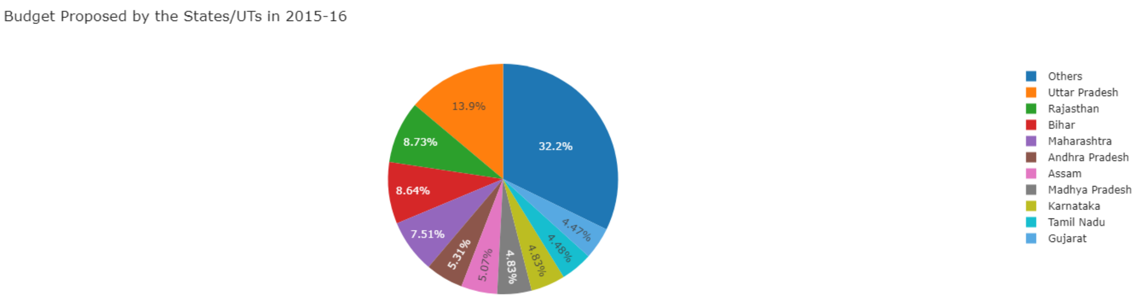
Total expenditure by the government is increasing on yearly basis as they are focusing more on health sector.





To find out which states/UTs have the most share in the Budget throughout the years we used pie chart for visualization.

```
years = NHM_df['Fiscal Year'].unique()

for year in years:
    currDf = NHM_df[NHM_df['Fiscal Year'] == year]
    currDf = currDf.dropna(subset=['Budget Proposed by the States/UTs'])
    p = (currDf.groupby(['State_UT'])['Budget Proposed by the States/UTs'].mean()).sort_values(ascending=False)[:10]
    u = pd.Series({'Others':sum((currDf.groupby(['State_UT'])['Budget Proposed by the States/UTs'].max()).sort_values(ascending=False)[10:-1])})
    p = pd.concat([p,u])

    fig = px.pie(NHM_df[NHM_df['Fiscal Year']==year], values=p.values, names=p.index,
                 title=f'Budget Proposed by the States/UTs in {year}')
    fig.show()
```



Budget Proposed by the States/UTs in 2016-17

Budget Proposed by the States/UTs in 2017-18

Budget Proposed by the States/UTs in 2018-19

Budget Proposed by the States/UTs in 2019-20

Budget Proposed by the States/UTs in 2021-22

Budget Proposed by the States/UTs in 2022-23

A review of the budgets put up by several states over the years reveals some intriguing trends. It appears that only a few states—Uttar Pradesh, Bihar, Rajasthan, Maharashtra, Karnataka, Assam, Gujarat, Madhya Pradesh, and West Bengal—propose over 70% of the nation's total budget. This implies that the states have more influence over how funds is distributed at the federal level.

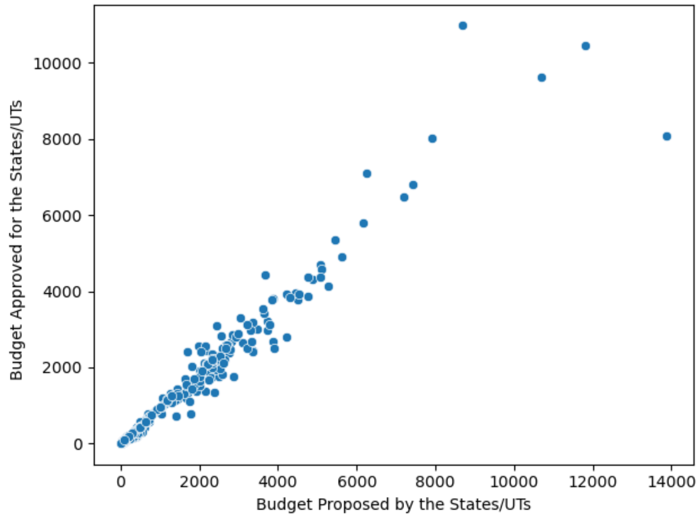
Additionally, it can be seen that Uttar Pradesh (UP) ranked among the top states for budget proposals across all years considered. This demonstrates that UP has fought for a bigger chunk of the state budget.

The remaining states account for another 30% of the total share and play a crucial role in the allocation of funds. The distribution of this budget proposal across states is critical because it allows us to understand each state's relative priorities, including their focus on the health sector. By analyzing trends in budget proposals, we can also gain insight into which states are investing more in the health sector and which states may need more attention in this area.

Applying regression model for data that seems to follow linear relationship:

Regression on Budget Proposed by the States/UTs and Budget Approved for the States/UTs

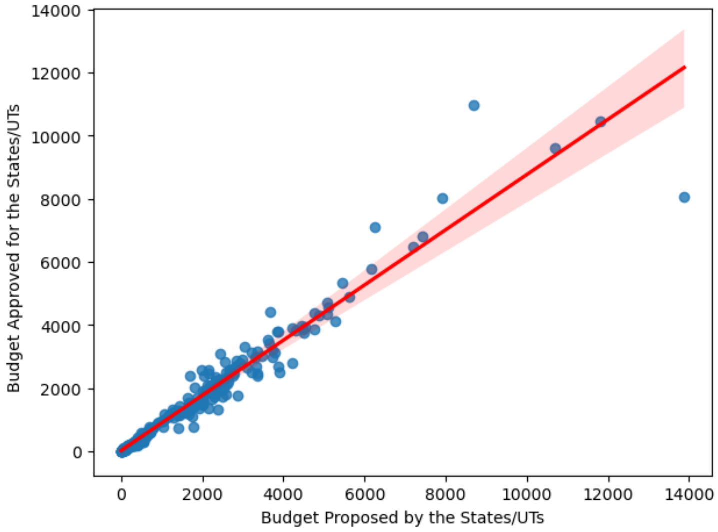
```
sns.scatterplot(x = 'Budget Proposed by the States/UTs',y= 'Budget Approved for the States/UTs  ',data=NHM_df)
plt.tight_layout()
plt.show()
```




```
data = NHM_df[['Budget Proposed by the States/UTs', 'Budget Approved for the States/UTs  ']].dropna(axis=0)
X = np.array(data['Budget Proposed by the States/UTs']).reshape(-1,1)
y = np.array(data['Budget Approved for the States/UTs  ']).reshape(-1,1)

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)
model = LinearRegression()
model.fit(X_train, y_train)
preds = model.predict(X_test)
score = model.score(X_test,y_test)

print(f'Score of Linear Regression Model: {score}')
```

Score of Linear Regression Model: 0.9853503449518848

```
sns.regplot(x = 'Budget Proposed by the States/UTs',y= 'Budget Approved for the States/UTs  ',data=NHM_df, line_kws={"color": "red"})
plt.tight_layout()
plt.show()
```

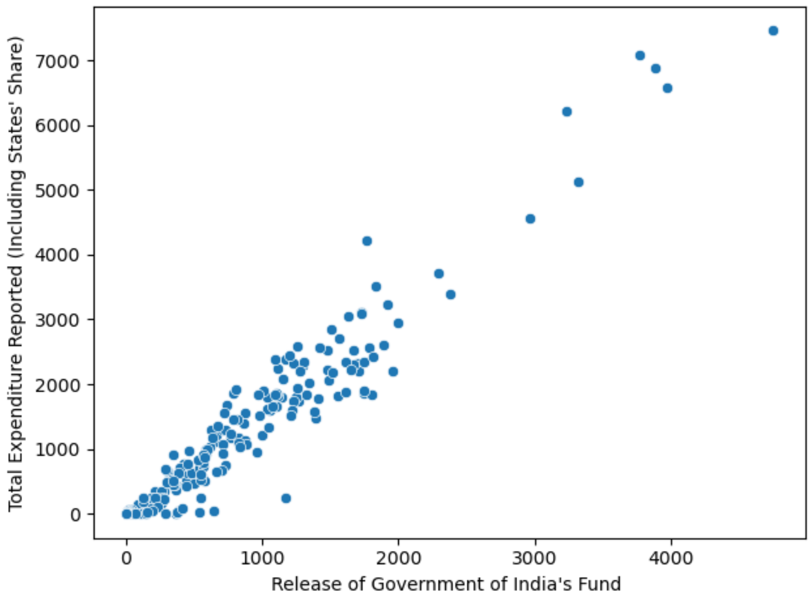


Looking at the scatterplots it could be seen that Budget Proposed by the States/UTs and Budget Approved for the States/UTs has some kind of linear relationship thus we modelled it with linear regression and found that we can predict the Expenditure based on Release of funds.

Our model had R2 score of 0.98 and it did great job predicting the budget approved.

Regression on Release of Government of India's Fund and Total Expenditure Reported

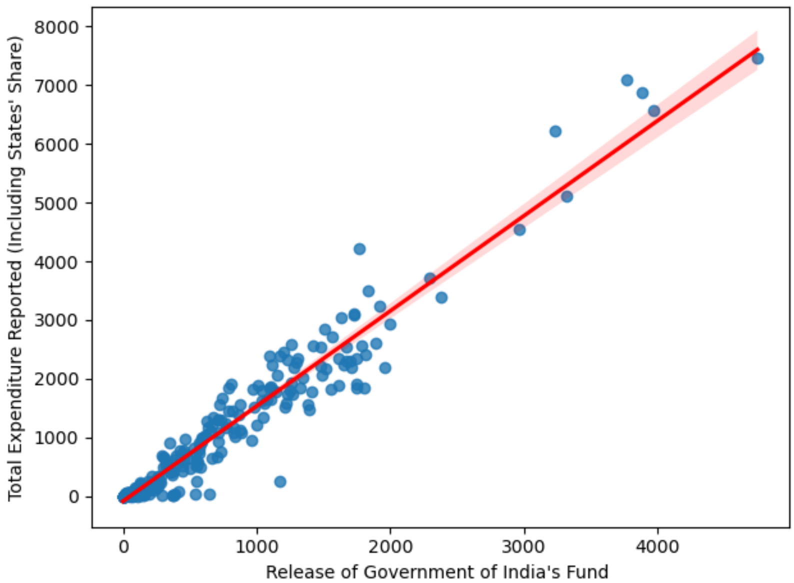
```
sns.scatterplot(x = "Release of Government of India's Fund ",y= "Total Expenditure Reported (Including States' Share)",data=NHM_df)
plt.tight_layout()
plt.show()
```




```
data = NHM_df[["Release of Government of India's Fund ", "Total Expenditure Reported (Including States' Share)"]].dropna(axis=0)
X = np.array(data["Release of Government of India's Fund "]).reshape(-1,1)
y = np.array(data["Total Expenditure Reported (Including States' Share)"]).reshape(-1,1)

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)
model = LinearRegression()
model.fit(X_train, y_train)
preds = model.predict(X_test)
score = model.score(X_test,y_test)

print(f'Score of Linear Regression Model: {score}')
```

```
Score of Linear Regression Model: 0.9615369512857515
```

```
sns.regplot(x = "Release of Government of India's Fund ",y= "Total Expenditure Reported (Including States' Share)",data=NHM_df, line_kws={"color": "red"})
plt.tight_layout()
plt.show()
```



Looking at the scatterplots it could be seen that Release of Government of India's Fund and Total Expenditure Reported has some kind of linear relationship thus we modelled it with linear regression and found that we can predict the Expenditure based on Release of funds.

Our model had R2 score of 0.96 and it did great job predicting the budget approved.

2. Mobile Medical Units (MMUs) data from September 2016:

This dataset tells us about the operational Mobile Medical Units (MMUs) in each state in year 2016. So, first we load our dataset and apply .info() method to initial insights.

```
# Import csv file which contains State/UT-wise Mobile Medical Units (MMUs) data from September 2016
MMU = pd.read_csv(r"Datasets\State UT-wise Status of Mobile Medical Units (MMUs) under National Rural Health Mission (NRHM) as on September, 2016.csv")

MMU.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 3 columns):
 #   Column                                   Non-Null Count  Dtype
---  ------                                   --------------  -----
 0   Sl. No.                                  37 non-null     object
 1   State/ UT Name                           37 non-null     object
 2   *Mobile Medical Units (MMUs) Operational 37 non-null     int64
dtypes: int64(1), object(2)
memory usage: 1016.0+ bytes
```

We observed that dataframe has 37 row and only 3 columns and there are no null values. Since there is no need for S1. No. column we will drop it.

```
# Drop unwanted columns: In our case "S1. No."
MMU.drop(columns=['Sl. No.'],inplace=True)

MMU.head()
```

	State/ UT Name	*Mobile Medical Units (MMUs) Operational
0	Bihar	7
1	Chhattisgarh	0
2	Himachal Pradesh	0
3	Jammu & Kashmir	11
4	Jharkhand	94

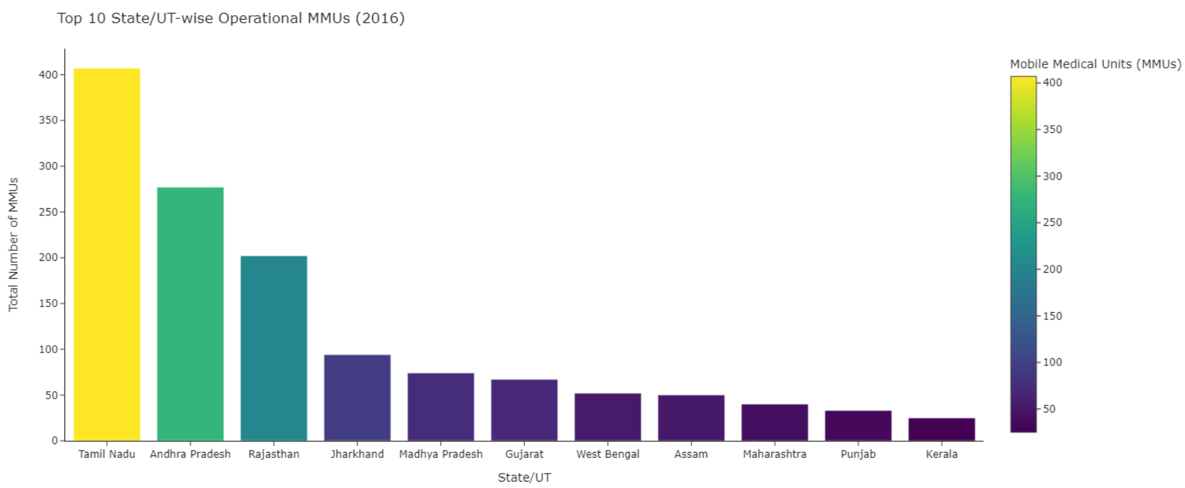
For ease of analysing further we renamed the columns and dropped Total row form the df. After that we ranked the states based on the number of operational MMUs and created a separate df for top 10 states.

```
# Renaming the column names for ease in handling.
MMU.rename(columns={'State/ UT Name': 'State/UT','*Mobile Medical Units (MMUs) Operational':'Mobile Medical Units (MMUs)'}, inplace = True)

# Droping the Total row so that we can perform visualization on rest rows.
MMU.drop((MMU[MMU['State/UT']=='Total']).index, inplace=True)

# Sort the values in descending order so we can rank the states.
MMU.sort_values(by=['Mobile Medical Units (MMUs)'], ascending=False, inplace=True)

# reset the indices.
MMU.reset_index(inplace=True, drop=True)

# Taking top 10 rows
MMU_Top10 = MMU[:11]

MMU_Top10
```

	State/UT	Mobile Medical Units (MMUs)
0	Tamil Nadu	407
1	Andhra Pradesh	277
2	Rajasthan	202
3	Jharkhand	94
4	Madhya Pradesh	74
5	Gujarat	67
6	West Bengal	52
7	Assam	50
8	Maharashtra	40
9	Punjab	33
10	Kerala	25

To plot the Top 10 States, we used simple bar chart.

```
# Using bar chart to compare MMUs count for top 10 States
fig = px.bar(MMU_Top10,
             x='State/UT',
             y='Mobile Medical Units (MMUs)',
             color='Mobile Medical Units (MMUs)',
             color_continuous_scale='Viridis')
fig.update_layout(height=600,
                  title='Top 10 State/UT-wise Operational MMUs (2016)',
                  xaxis_title='State/UT',
                  yaxis_title='Total Number of MMUs',)
fig.show()
```



From the above plot, we observed that Tamil Nadu has the maximum number of operational MMUs (407) according to the data which then followed by Andhra Pradesh and Rajasthan which have 277 and 202 operational units.

3. Operational Ambulances Under NRHM:

This dataset has the information about operational ambulances under NHM during 2018. Load the data and apply .info().

```
Ambulance = pd.read_csv(r"Datasets\StateUT-wise Ambulances Operational under National Health Mission (NHM) during 2018 .csv")

Ambulance.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 6 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   States/UTs                      37 non-null     object
 1   Dial 108                        37 non-null     int64
 2   Dial 102/104                    37 non-null     int64
 3   Other patients transport vehicle  37 non-null   int64
 4   Total Ambulance under NHM       37 non-null     int64
 5   State Ambulances                37 non-null     int64
dtypes: int64(5), object(1)
memory usage: 1.9+ KB
```

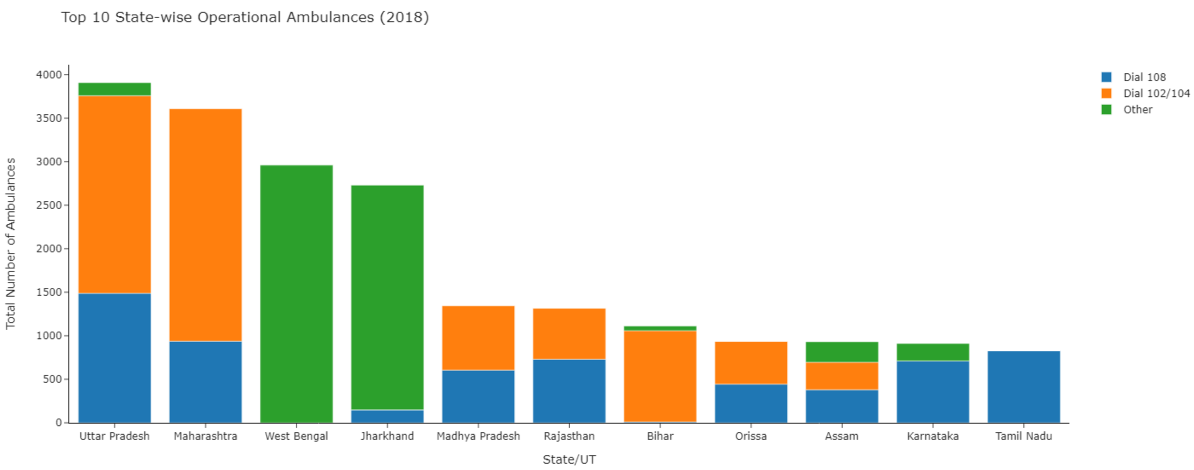
Dataframe contains 37 rows and 6 columns, drop unnecessary rows. And rank the states based on total number of ambulances.

```
# Drop "All-India" row so that it wont hamper our analysis.
Ambulance.drop((Ambulance[Ambulance['States/UTs']=='All-India']).index, inplace=True)

# Rank states based on Total number of operational Ambulances
Ambulance.sort_values(by=['Total Ambulance under NHM'], ascending=False, inplace=True)

# Reset indices
Ambulance.reset_index(inplace=True, drop=True)

# select Top 10 States with most number of Operational Ambulances.
Ambulance_Top10 = Ambulance[:11]

Ambulance_Top10
```

	States/UTs	Dial 108	Dial 102/104	Other patients transport vehicle	Total Ambulance under NHM	State Ambulances
0	Uttar Pradesh	1488	2270	150	3908	0
1	Maharashtra	937	2674	0	3611	3461
2	West Bengal	0	0	2960	2960	281
3	Jharkhand	149	0	2581	2730	271
4	Madhya Pradesh	606	740	0	1346	0
5	Rajasthan	730	587	0	1317	363
6	Bihar	10	1049	53	1112	164
7	Orissa	444	491	0	935	280
8	Assam	380	316	235	931	0
9	Karnataka	711	0	200	911	627
10	Tamil Nadu	829	0	0	829	950

We used stacked barplot to represent diversity in type of ambulances Top 10 states posess.

```
fig = go.Figure()
fig.add_trace(go.Bar(
    x=Ambulance_Top10['States/UTs'],
    y=Ambulance_Top10['Dial 108'],
    name='Dial 108',
))
fig.add_trace(go.Bar(
    x=Ambulance_Top10['States/UTs'],
    y=Ambulance_Top10['Dial 102/104'],
    name='Dial 102/104',
))
fig.add_trace(go.Bar(
    x=Ambulance_Top10['States/UTs'],
    y=Ambulance_Top10['Other patients transport vehicle'],
    name='Other',
))
fig.update_layout(height=600,
                title='Top 10 State-wise Operational Ambulances (2018)',
                xaxis_title='State/UT',
                yaxis_title='Total Number of Ambulances',
                barmode='relative')
fig.show()
```



From the above plot, we can deduce that Uttar Pradesh has the greatest number of Operational Ambulances.

Also UP has the greatest number of Dial 108 ambulances whereas Maharashtra and West Bengal has the greatest number of Dial 102/104 and other ambulances respectively.

4. Established Special New-born Care Units (SNCU) under NRHM:

SNCU dataset tells us about the established Special New-born Care Units in a State/UT up to January 2020. Load the data and apply .info() method.

```
SNCU = pd.read_csv(r"Datasets\StateUT-wise Established Special Newborn Care Units (SNCU) under National Health Mission (NHM) up to January, 2020.csv")

SNCU.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38 entries, 0 to 37
Data columns (total 3 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   S. No.                                      38 non-null     object
 1   States/UT                                   38 non-null     object
 2   Total Number of SNCU (up to January, 2020)  38 non-null     int64
dtypes: int64(1), object(2)
memory usage: 1.0+ KB
```

SNCU df has 38 rows and 3 columns, then we dropped unwanted columns and rows.

```
# Drop unwanted columns: In our case "S. No."
SNCU.drop(columns=['S. No.'],inplace=True)

SNCU.head()
```

	States/UT	Total Number of SNCU (up to January, 2020)
0	Andaman & Nicobar Islands	1
1	Andhra Pradesh	47
2	Arunachal Pradesh	5
3	Assam	28
4	Bihar	43

```
# Rename Column for ease
SNCU.rename(columns={'Total Number of SNCU (up to January, 2020)': 'Total Number of SNCU'}, inplace = True)

# Drop Total row
SNCU.drop((SNCU[(SNCU['States/UT']=='Total') | (SNCU['States/UT']=='Ladakh') ]).index, inplace=True)

# Combine Ladakh with Jammu & Kashmir
SNCU.loc[SNCU['States/UT']=='Jammu & Kashmir','Total Number of SNCU' ] += 2
```

For visualization part, we used Plotly library to plot Map of India and shade states based on the number of SNCUs.

So, for plotting the map we need a GeoJSON file which contains boundary of each state. After loading the JSON file we created a dictionary with state id and state name as key value pairs.

```python
# We need GeoJSON file for plotting boundary of State/UT
india_states = json.load(open("Datasets\states_india.geojson", "r"))

# Create a dictionary where State id is key and State name is value
state_id_map = {}
for feature in india_states["features"]:
    feature["id"] = feature["properties"]["state_code"]
    state_id_map[feature["properties"]["st_nm"]] = feature["id"]
```

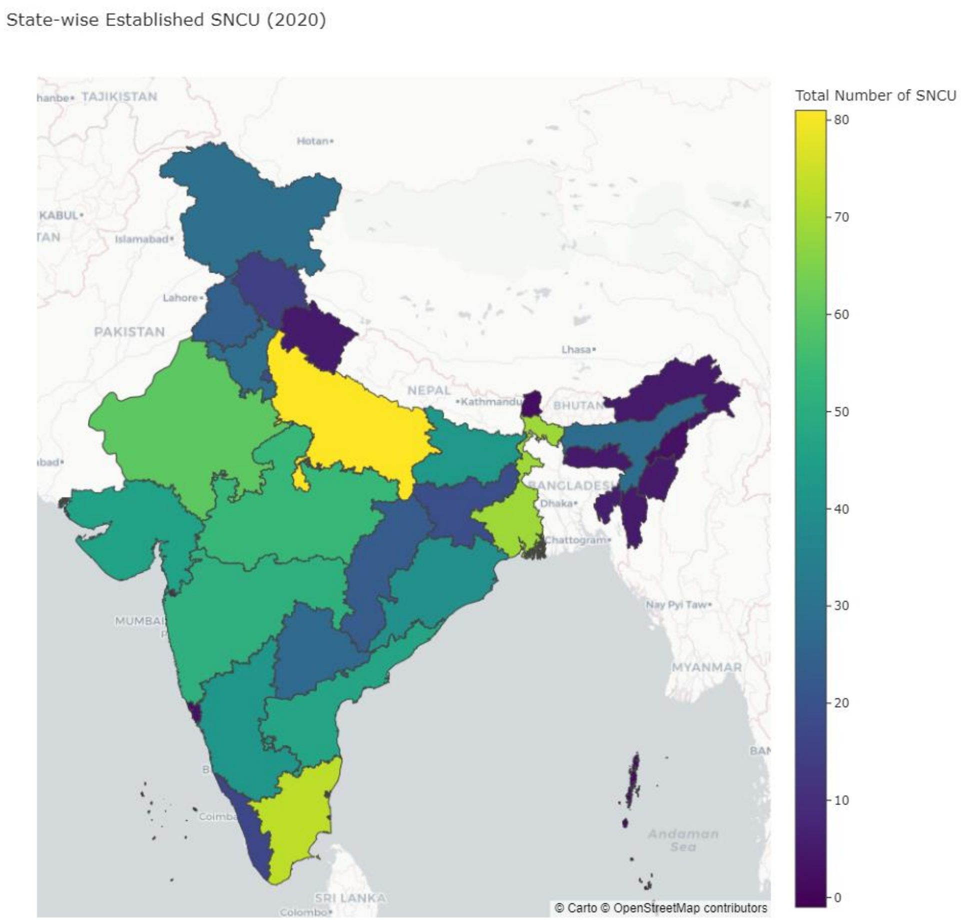
There was difference in spellings in the state name, so we altered the names for solving the problem. And added state id column in our SNCU df.

```python
# Change spelling so that it matches geojson file.
SNCU.loc[SNCU['States/UT']=='Andaman & Nicobar Islands','States/UT' ] = 'Andaman & Nicobar Island'

SNCU.loc[SNCU['States/UT']=='Dadar & Nagar Haveli','States/UT' ] = 'Dadara & Nagar Havelli'

SNCU.loc[SNCU['States/UT']=='Arunachal Pradesh','States/UT' ] = 'Arunanchal Pradesh'

SNCU.loc[SNCU['States/UT']=='Delhi','States/UT' ] = 'NCT of Delhi'

SNCU.loc[SNCU['States/UT']=='Pondicherry','States/UT' ] = 'Puducherry'

SNCU.loc[SNCU['States/UT']=='Tamilnadu','States/UT' ] = 'Tamil Nadu'

# add id column in the dataframe
SNCU["id"] = SNCU["States/UT"].apply(lambda x: state_id_map[x])

SNCU.head()
```

	States/UT	Total Number of SNCU	id
0	Andaman & Nicobar Island	1	35
1	Andhra Pradesh	47	28
2	Arunanchal Pradesh	5	12
3	Assam	28	18
4	Bihar	43	10

Now, to plot the map we used choropleth_mapbox() function from Plotly express, and changed the color based on the total number of SNCU.

```
# Plot the map of india
fig = px.choropleth_mapbox(
    SNCU,
    locations="id",
    geojson=india_states,
    color="Total Number of SNCU",
    hover_name="States/UT",
    title="State-wise Established SNCU (2020)",
    mapbox_style="carto-positron",
    center={"lat": 24, "lon": 83},
    zoom=4,
    color_continuous_scale="Viridis",
    color_continuous_midpoint=40)
fig.update_layout(width=1000, height=1000)
fig.show()
```



From the above map we can interpret that UP has the greatest number of established SNCU while north-eastern states have very a smaller number of SNCU. States like MP, Gujarat, Rajasthan, Maharashtra etc are in between.

5. Accredited Social Health Activists (ASHAs) Selected under NRHM:

This dataset tells us about the number of accredited Social health activists during 2008.

```
# import dataset
ASHAs = pd.read_csv(r"Datasets\StateUT-wise Accredited Social Health Activists (ASHAs) Selected under National Health Mission (NHM) during 2018.csv")

ASHAs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   State/UT     37 non-null     object
 1   Total ASHAs  37 non-null     int64
dtypes: int64(1), object(1)
memory usage: 720.0+ bytes
```

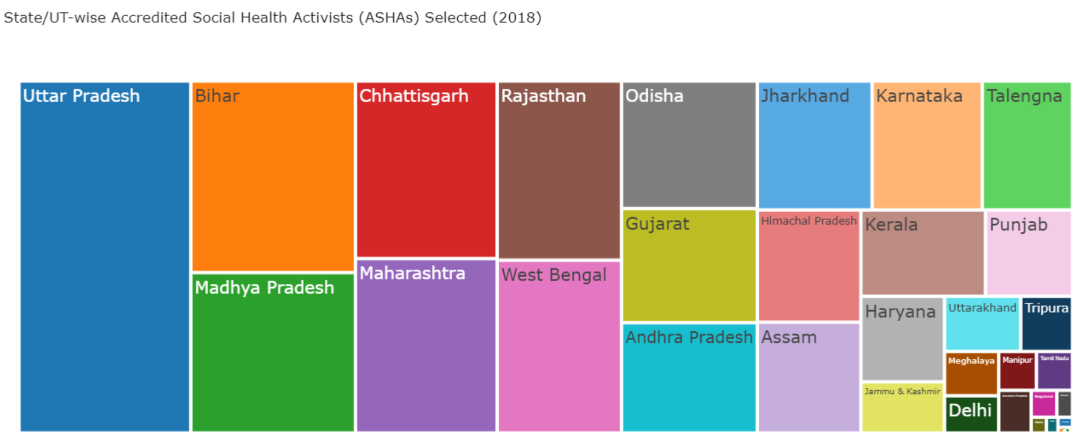
Dataframe has 37 rows and 2 columns. We removed the unwanted row which contains total of all the states.

```
# Remove row with total
ASHAs.drop((ASHAs[ASHAs['State/UT']=='Total']).index, inplace=True)

ASHAs.head()
```

	State/UT	Total ASHAs
0	Bihar	87788
1	Chhattisgarh	70008
2	Himachal Pradesh	32374
3	Jammu & Kashmir	11916
4	Jharkhand	41199

For visualization part, we used tree-map which easily shows which state/UT rank higher and which state/UT ranks lower.

```
# Plot a treeMap
fig = px.treemap(ASHAs, path=['State/UT'], values='Total ASHAs')
fig.update_layout(height=600,
                  title='State/UT-wise Accredited Social Health Activists (ASHAs) Selected (2018)')
fig.update_traces(textfont_size=20)
fig.show()
```



From the above Tree-map, we can see that again UP has the most number of ASHAs while Bihar and MP take second and third place.

6. Health Centres Under NRHM Dataset:

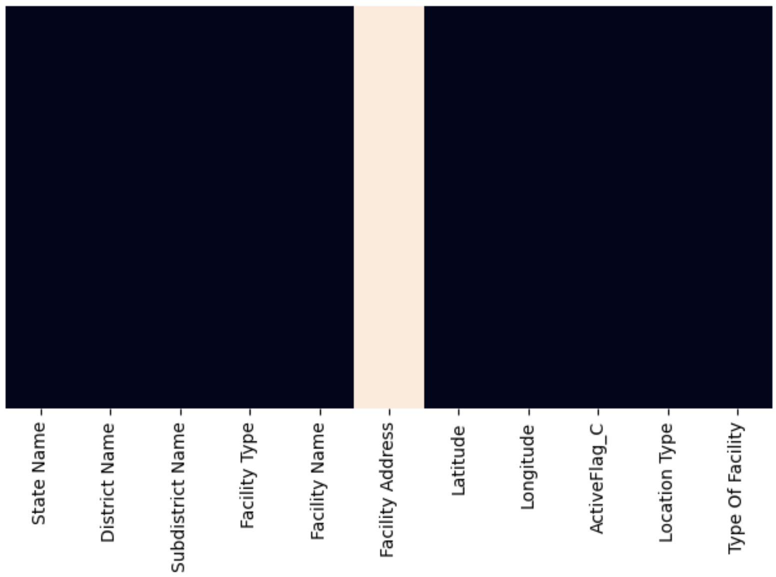
Health Centres dataset gives insight about several characteristics of health centres throughout the country. We tried to show these characteristic as follows:

```
# import dataset
Health_centers = pd.read_csv(r'Datasets\geocode_health_centre.csv',low_memory = False)

Health_centers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200438 entries, 0 to 200437
Data columns (total 11 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   State Name         200438 non-null  object
 1   District Name      200438 non-null  object
 2   Subdistrict Name   200438 non-null  object
 3   Facility Type      200438 non-null  object
 4   Facility Name      200438 non-null  object
 5   Facility Address   18 non-null      object
 6   Latitude           200438 non-null  object
 7   Longitude          200423 non-null  object
 8   ActiveFlag_C       200438 non-null  object
 9   Location Type      200438 non-null  object
 10  Type Of Facility   200437 non-null  object
dtypes: object(11)
memory usage: 16.8+ MB
```

Dataset has 200438 rows and 11 columns and datatype of all the columns is set to object. To visualize null values in each row we plot a heatmap.

```
# Plot Null values Heatmap
sns.heatmap(Health_centers.isnull(),yticklabels=False ,cbar=False)
plt.tight_layout()
plt.show()
```



From the heatmap we can saw that Facility Address column has a lot of null values, therefore we dropped that column.

```
# Drop column with lots of null values
Health_centers.drop(columns=['Facility Address'], axis=1, inplace=True)

Health_centers.head()
```

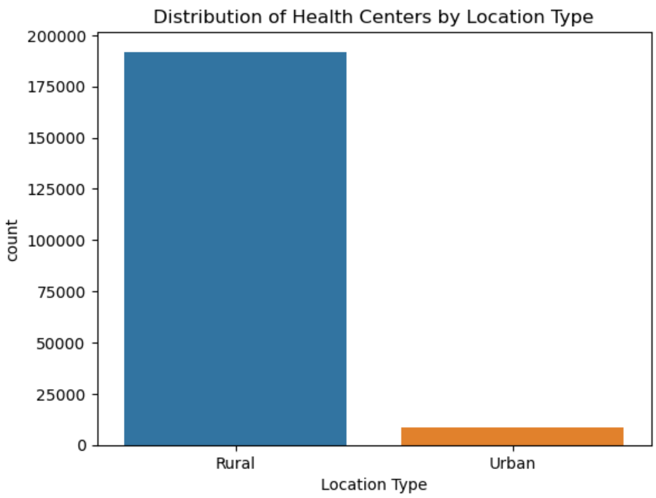
	State Name	District Name	Subdistrict Name	Facility Type	Facility Name	Latitude	Longitude	ActiveFlag_C	Location Type	Type Of Facility
0	A & N Islands	Nicobar	Nancowry	chc	CHC Nancowry	7.96109	93.5589	Y	Rural	Public
1	A & N Islands	South Andaman	Ferrargunj	chc	CHC Bambooflat	11.7303	92.65003	Y	Rural	Public
2	A & N Islands	North and Middle Andaman	Rangat	chc	CHC Rangat	12.71609	92.90579	Y	Rural	Public
3	A & N Islands	North and Middle Andaman	Diglipur	chc	CHC Diglipur	13.30682	92.9411	Y	Rural	Public
4	A & N Islands	Nicobar	Car Nicobar	dis_h	BJR Hospital	9.14893	92.75578	Y	Rural	Public

Then we typecasted latitude and longitude columns into numeric values from objects.

```
# Convert Latitude and Longitude column to numeric datatype
Health_centers['Latitude'] = pd.to_numeric(Health_centers['Latitude'], errors='coerce')
Health_centers['Longitude'] = pd.to_numeric(Health_centers['Longitude'], errors='coerce')

# Drop Null values
Health_centers = Health_centers.dropna()
```

After pre-processing our data, we started plotting the graphs. The first plot tells us about the number of Rural vs Urban healthcare centres using a countplot.

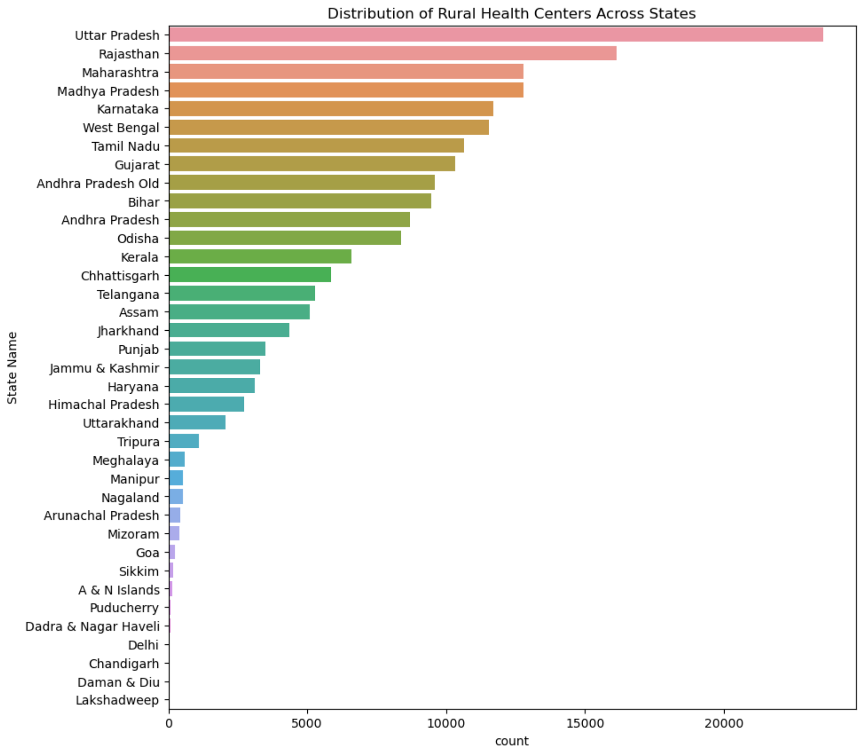
```
sns.countplot(x=Health_centers['Location Type'])
plt.title("Distribution of Health Centers by Location Type")
plt.show()
```



We can clearly see that Rural healthcare centres are much more in numbers than Urban which tells us government focused on Rural Healthcare more.

Then to find out which state has most number of rural HCs we used countplot.

```
plt.figure(figsize=(10,10))
sns.countplot(y=Health_centers[Health_centers['Location Type']=='Rural']['State Name'],
              order=Health_centers[Health_centers['Location Type']=='Rural']['State Name'].value_counts().index)
plt.title("Distribution of Rural Health Centers Across States")
plt.show()
```



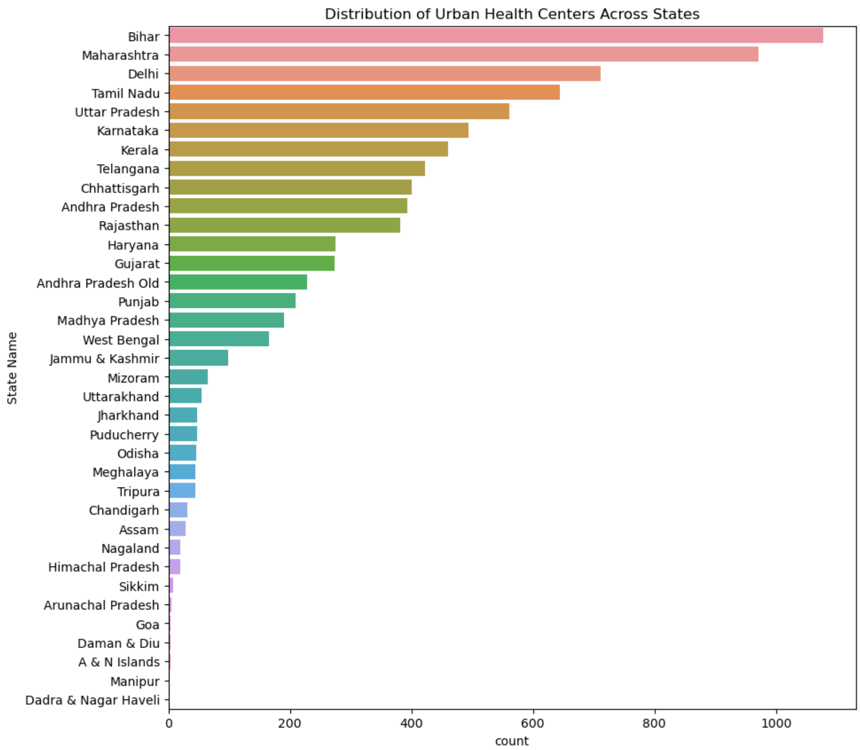
Uttar Pradesh, Bihar, Rajasthan, Maharashtra, Karnataka, Assam, Gujarat, Madhya Pradesh, and West Bengal are the leading proposers of the rural health sector budget in the country . This suggests that these states have a significant influence on the allocation of funds at the national level, which can play a crucial role in their rural health management. By having a greater say in the allocation of funds, these states can prioritize their specific health needs and ensure necessary resources for their rural populations. This can lead to better health outcomes, improved access to healthcare and ultimately a healthier population.

By having a greater say in the allocation of funds, these states can prioritize areas that need the most attention and address specific health challenges facing rural populations. This may include improving health care infrastructure, providing essential medicines, promoting preventive health care measures, and raising health care awareness among rural populations.

An increased focus on rural health management could significantly improve health outcomes in these states. It can also help reduce the burden of preventable disease, increase life expectancy, and improve the quality of life for millions of people in rural areas.

To find how many urban HCs are there in each State/UT we plotted countplot.

```
plt.figure(figsize=(10,10))
sns.countplot(y=Health_centers[Health_centers['Location Type']=='Urban']['State Name'],
              order=Health_centers[Health_centers['Location Type']=='Urban']['State Name'].value_counts().index)
plt.title("Distribution of Urban Health Centers Across States")
plt.show()
```



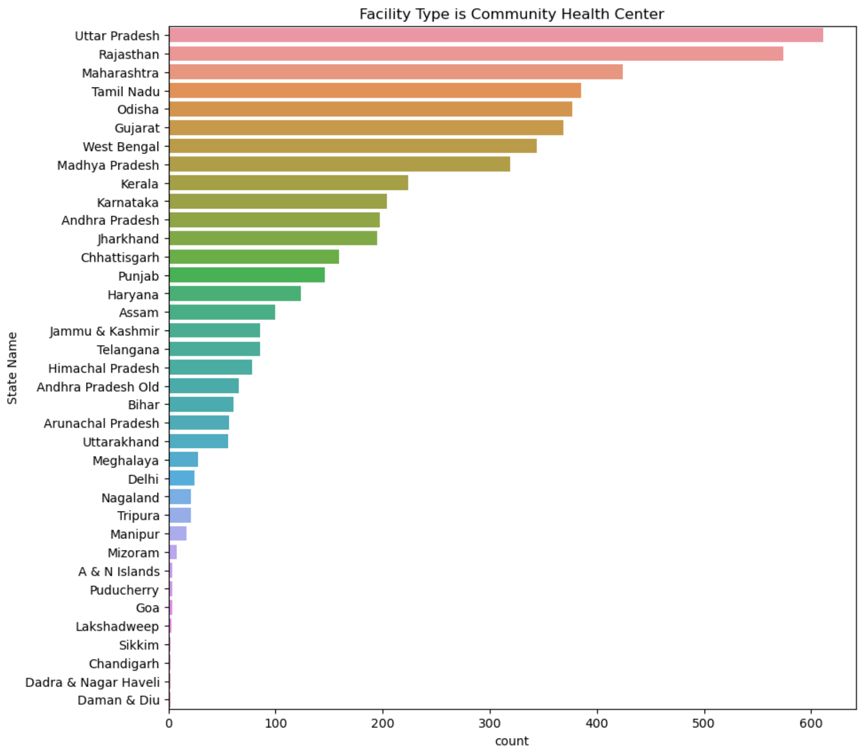
The top states that seem to care more about urban health than rural health are Bihar, Maharashtra, Tamil Nadu, Delhi, Uttar Pradesh, and Karnataka. Numerous factors, such as the greater concentration of infrastructure and resources in urban areas, as well as the higher population densities there, could account for this.

The fact that rural inhabitants frequently have poor access to healthcare and are more susceptible to a range of health issues makes it important to note that neglecting rural health can have major repercussions. A country's current health inequalities across its various areas and inhabitants can also be made worse by concentrating on urban health at the expense of rural health.

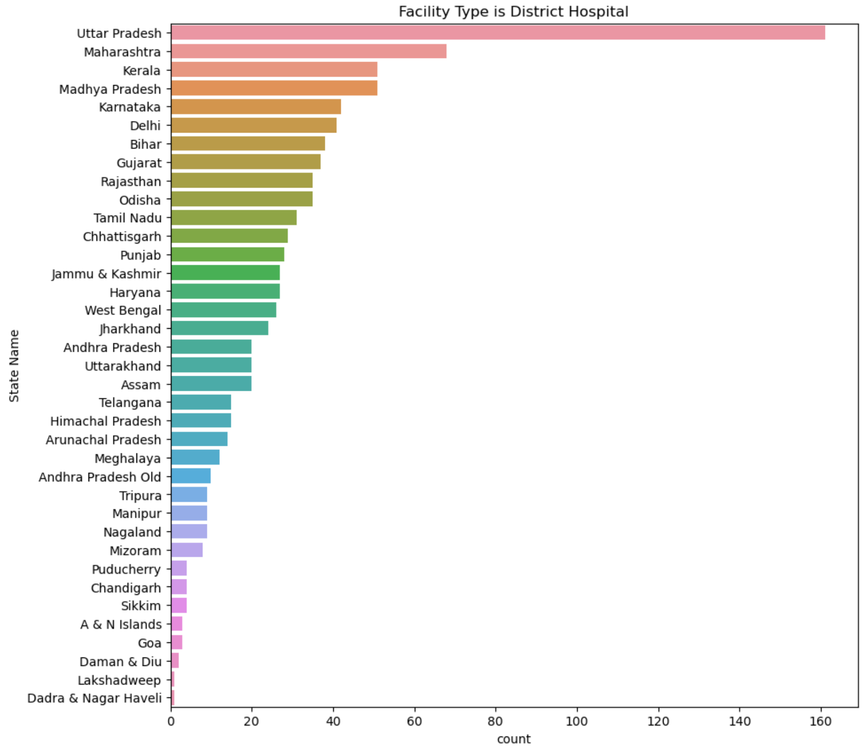
Therefore, states and governments must give rural and urban health care systems top priority and funding to guarantee that everyone has access to high-quality healthcare, regardless of where they live or their socioeconomic status. This can be accomplished in several ways, including by boosting funding for rural health programmes, enhancing the facilities and resources available there, and encouraging medical personnel to practise in rural regions.

Among rural and urban HCs there are different types of facilities dedicated for specific purposes. So, we plotted multiple count-plot to get State/UT-wise count of different type of facilities.

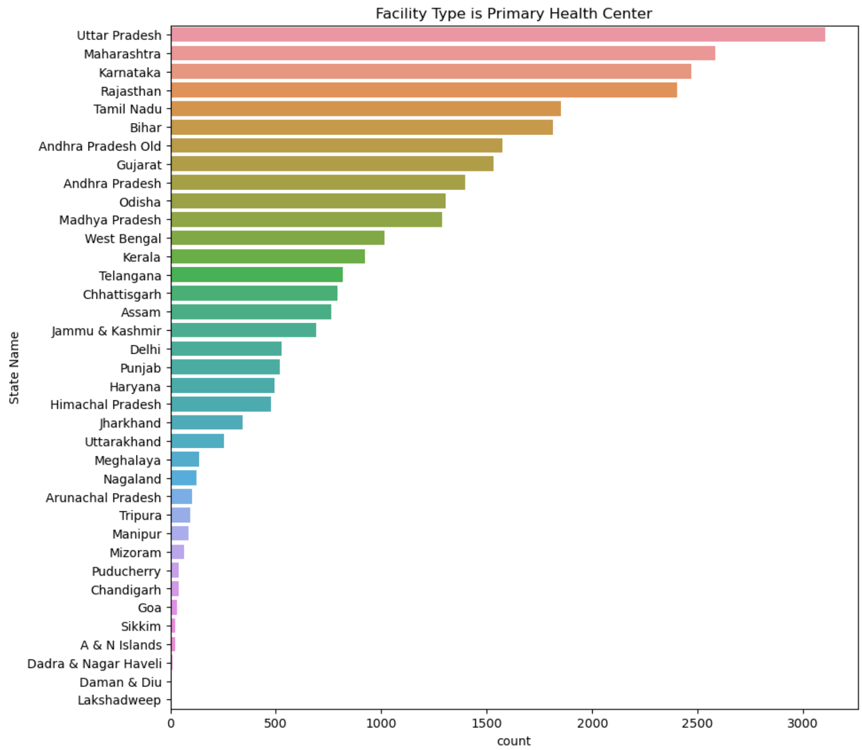
```
abbreviations = {'chc': 'Community Health Center', 'phc': 'Primary Health Center', 'sub_cen': 'Sub Center', 'dis_h': 'District Hospital', 's_t_h': 'State Hospital' }

for i in Health_centers['Facility Type'].unique():
    plt.figure(figsize=(10,10))
    sns.countplot(y=Health_centers[Health_centers['Facility Type'] == i]['State Name'],
                  order=Health_centers[Health_centers['Facility Type'] == i]['State Name'].value_counts().index)
    plt.title(f'Facility Type is {abbreviations[i]}')
    plt.show()
```



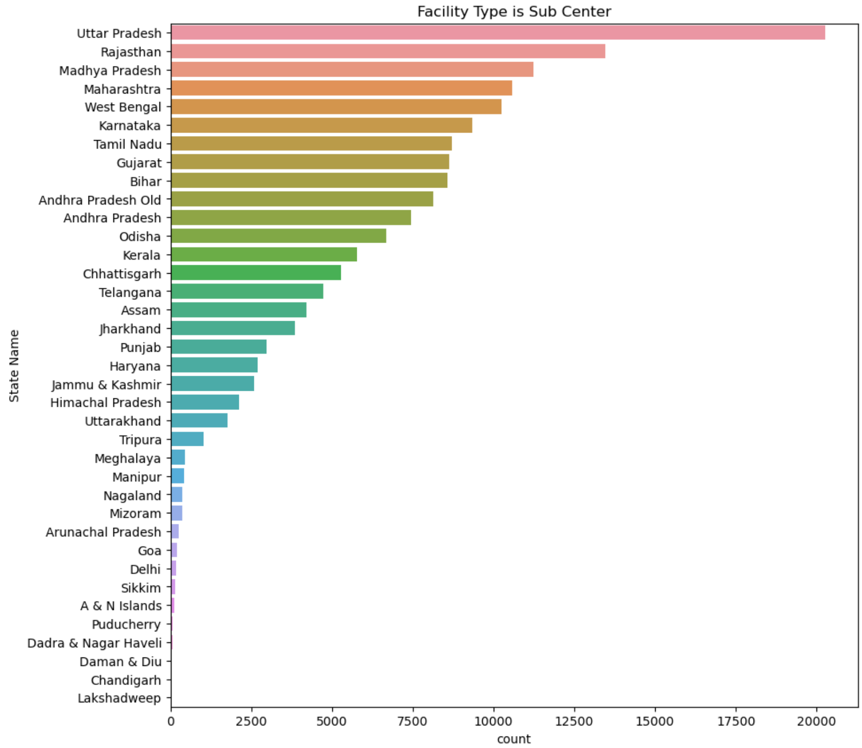
UP and Rajasthan has a greater number of Community Health Centres. Maybe their geographical size is the reason for this.

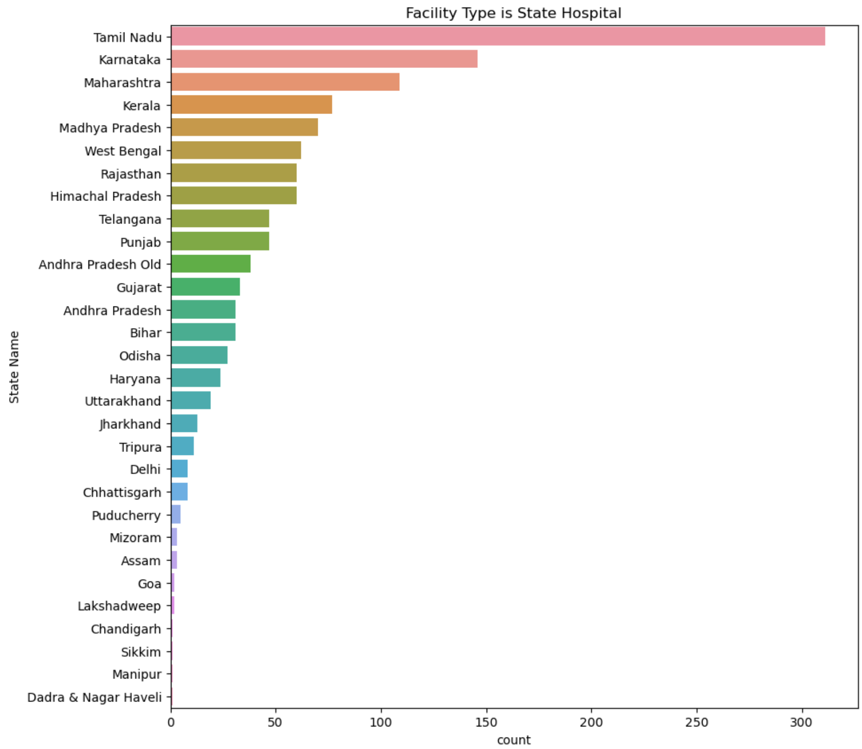


Most number of district hospitals are in UP, since UP has highest number of districts, but other states like MP, which has more districts than Maharashtra, has lower number of District hospitals.



UP once again has highest number of primary health centres as well.





From the above plots we can conclude that State of UP has highest number of Health Centres in each category. While North-Eastern States have lower number.

To summarize, the total number of State/UT Healthcare Centres we plotted a map. For that, we grouped rows based on the name of State/UT. Then we created a new df which contains pin-point location of each state using latitude and longitude.

```
groupedBy_States = Health_centers.groupby('State Name').count()

states = ['A & N Islands', 'Andhra Pradesh', 'Andhra Pradesh Old', 'Arunachal Pradesh', 'Assam', 'Bihar', 'Chandigarh', 'Chhattisgarh', 'Dadra & Nagar Haveli', 'Daman & Diu', 'Delhi', 'Goa', 'Gujarat', 'Haryana', 'Himachal Pradesh',

latitude = [11.66702557, 14.7504291, 15.91, 27.10039878, 26.7499809, 25.78541445, 30.71999697, 22.09042035, 20.26657819, 20.39737335, 28.6699929, 15.491997, 22.2587, 28.45000633, 31.10002545, 34.29995933, 23.

longitude = [92.73598262, 78.57002559, 79.74, 93.61660071, 94.21666744, 87.4799727, 76.78000565, 82.15998734, 73.0166178, 72.83973126, 77.23000403, 73.81800065, 71.1924, 77.01999101, 77.16659704, 74.4666584

longLatdata = pd.DataFrame({'State': states, 'latitude': latitude, 'longitude': longitude})

longLatdata.set_index(['State'],inplace=True)

longLatdata.head()
```

	latitude	longitude
State		
A & N Islands	11.667026	92.735983
Andhra Pradesh	14.750429	78.570026
Andhra Pradesh Old	15.910000	79.740000
Arunachal Pradesh	27.100399	93.616601
Assam	26.749981	94.216667

Join both the grouped df and new longLatdata into one dataframe.

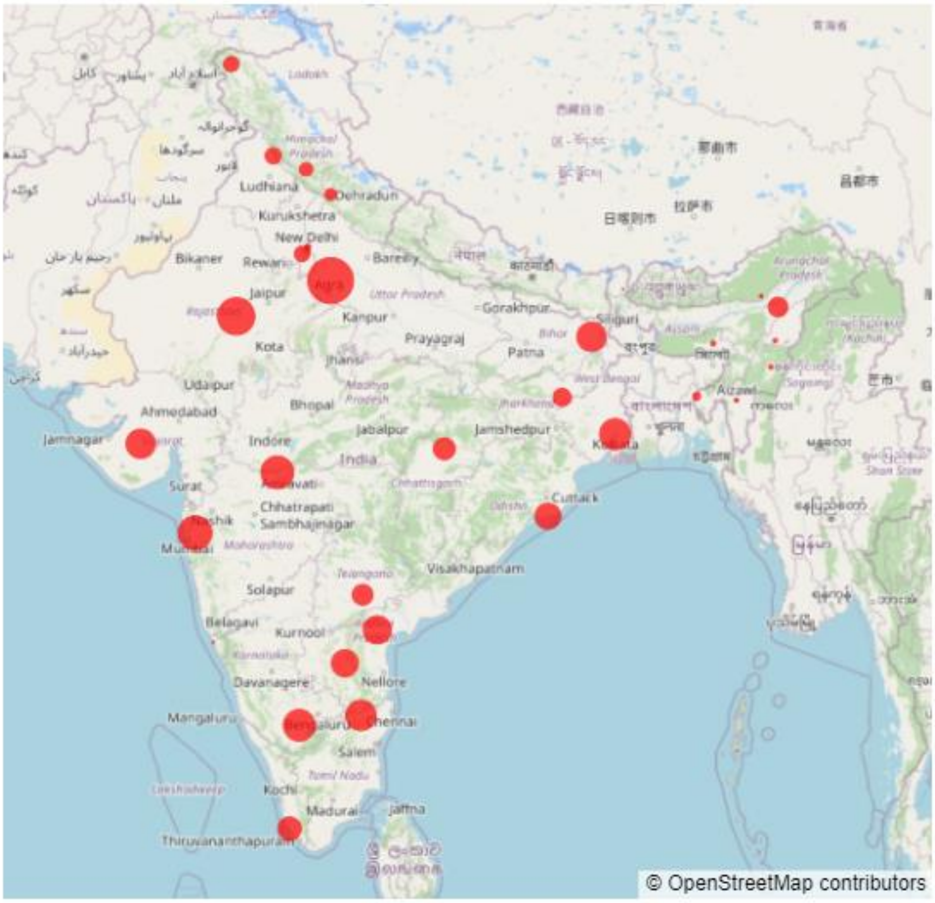
```
Joined_df = pd.concat([longLatdata,groupedBy_States],axis=1)

Joined_df.rename(columns={'District Name': 'Number of Facilities'}, inplace = True)

Joined_df.drop(columns=['Subdistrict Name','Facility Type','Facility Name', 'Latitude','Longitude','ActiveFlag_C','Location Type','Type Of Facility'],inplace=True)

Joined_df.head()
```

	latitude	longitude	Number of Facilities
A & N Islands	11.667026	92.735983	144
Andhra Pradesh	14.750429	78.570026	9103
Andhra Pradesh Old	15.910000	79.740000	9823
Arunachal Pradesh	27.100399	93.616601	432
Assam	26.749981	94.216667	5106

We used scatter_mapbox() function from plotly express library to plot a scatter plot on a map. In this plot we used the number of facilities column to vary the size of bubble and latitude and longitude columns to assign position to each bubble based on their state/UT name.

```python
fig = px.scatter_mapbox(Joined_df,
                        lat=Joined_df["latitude"],
                        lon=Joined_df["longitude"],
                        size='Number of Facilities',
                        zoom=3.5,
                        color_discrete_sequence=["red"],
                        mapbox_style="open-street-map",
                        center={"lat": 22, "lon": 83},
                        title='Scatter Map on Number of Facilities per State',
                        hover_name=Joined_df.index)

fig.update_layout(width=700, height=700)

# Show the map
fig.show()
```

Scatter Map on Number of Facilities per State



Conclusion:

In conclusion, this report has provided an overview on the National Rural Health Mission (NRHM) and National Health Mission (NHM) by highlighting different aspects in which government has worked. From our analysis we found out that significant amount of Funds has been allocated throughout the years for improving healthcare facilities in rural as well as urban areas. We also observed that, the amount of funds used by certain states was significantly higher, while in other years the increased funds were distributed more evenly among the different states. We can also observe that some states consistently use more funds for NRHM than others, such as Uttar Pradesh, which has the highest threshold in most years.

Other aspects such as number of Ambulances, MMUs, SNCUs and ASHAs were also explored in this report. We also analyzed different types of healthcare facilities and their distribution throughout the nation.

In all these categories UP has been consistently on the Top of the list while north-eastern states were bottom of the list. So, we can conclude that UP has been the prime beneficiary from the NRHM & NHM.

References:

For definitions:

https://nhm.gov.in/index1.php?lang=1&level=1&lid=49&sublinkid=969

For data collection:

https://data.gov.in/

https://openbudgetsindia.org/dataset/national-health-mission

For plots:

https://plotly.com/python/

https://seaborn.pydata.org/examples/index.html

https://matplotlib.org/stable/gallery/color/named_colors.html

Nakul Tomar

nakultomar28@gmail.com| (+91) 6352210752

EDUCATION

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY — Gandhinagar, India

Master of Science in Data Science ***(MS DS)*** CGPA - 7.29/10 — *07/22*

St. Xaviers College — Ahmedabad, India

Bachelor of science in statistics CGPA - 6.3/10 — 07/17 - 09/20

Kendriya Vidyalaya (CBSE) — Ahmedabad,India

Percentage: 70%

2016 – 2017 (HSC)

Percentage: 86%

2014 – 2015(SSC)

PROJECTS

Home Credit Default Risk

08/22 – 12/22

- Performed the initial part of the lifecycle of a data science project which included data cleaning, EDA, data manipulation, data filtering, and data visualization.
- Gained hands-on experience in using Pandas, NumPy, Plotly express, matplotlib, seaborn for data visualization.
- Feature engineering and feature selection
- Applied machine learning model and checked efficiency with variation of learning rate.

Sentiment Analysis

03/22 – 06/22

- Implementation of sentiment analysis and web scrapping to extract customer reviews for a product on Amazon and performing sentimental analysis to gain insights into the market value of the product.
- Tools and Technologies used : pandas,seaborn,selenium,matplotlib,Beautiful Soup , NLTK,TextBlob,Hugging fave transformers.
- Gained hands-on models like Random Forest Classifier,Naive Bayes classifier , Stochastic Gradient Descent (SVM) , logistic Regression and BERT Text classifier.

TECHNICAL SKILL SET

CODING LANGUAGES - Python (Advanced), R (Intermediate), SQL (Intermediate)

SOFTWARES/TOOLS - Pycharm, Jupyter Notebooks, RStudio, Postgres, MySQL, Oracle Live SQL, MS Excel
TECHNICAL KNOWLEDGE - Data Science, Machine Learning, Data Structures, DBMS
COMPETITIVE PLATFORM - HackerRank, Kaggle,Github

ADDITIONAL

CERTIFICATION - Advance MS Excel(Coursera) ,Python Programming(Udemy).
VOLUNTEERING - Event Manager at SolderX - Gujarat Technology University Events
HOBBIES - Basketball,Trekking, Football, Cricket, Management, Music, Reading

+91 7987015056
jainayushh@gamil.com

AYUSH JAIN

Education

DAIICT, Gandhinagar:
MSc. Data Science: SPI: 9.0

St. Xavier's College, Ahmedabad:
BSc. Physics: CGPA: 10

Central Board of Secondary Education:
HSC: 80%
SSC: 10 CGPA

Certifications

Programming Languages: **Python, JAVA, SQL, R**.
Technologies: **Hadoop, Android Studio, Excel, GitHub.**

Projects

Investment Portfolio Recommendation System:
April 2023 - Present

Indigenous Pipeline for Deployment of Machine learning models on RCD:
January 2023 - Present

Hobbies

Powerlifting/Bodybuilding
Sports

Harshit V. Shah

Ahmedabad, INDIA   hsharshitshah0307@gmail.com   9978645603   Harshit Shah

Profile

Data Science aspirant, adept in Python and its various libraries for Data visualization and Analytics, Data Scraping, Data Wrangling, and databases such as MongoDB and MySQL.
Skilled in Python, and C with Leadership, and Public Speaking.
Undergoing Master Of Science in Data Science - MSc. DS from Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat, India.

Education

2024 — **Master of Science - Data Science,**
Dhirubhai Ambani Institute of Information and Communication Technology

2022 — **Bachelor Of Science - Information Technology,** *Indus University*
CGPA: 9.2
Through this degree, I learned Software, Databases, and networking. I studied software development, software testing, software engineering, web design, databases, programming, computer networking, and computer systems.

Projects

2022 — **Sentiment Analysis**
Implementation of sentiment analysis and web scraping to extract customer reviews for a product from an e-commerce website and performing sentiment analysis to gain insights into the market value of the product. Followed by creating a dashboard visualization for the same using Tableau.

2022 — **Stocks Visualizing and Analysis**
Created a single-page web application using Dash (a python framework) which will show company (ticker and analysis) and stock plots based on the stock code given by the user. Also, user can see the combine analysis of the stocks and visualize it with the interactive graphs made using Plotly Express

2021 — **Live Face Mask Detection**
Developed a deep learning model for face mask detection using Python, Keras, and OpenCV. We developed the face mask detector model for detecting whether a person is wearing a mask or not. We have trained the model using Keras with network architecture. Training the model is the first part of this project and testing using a webcam using OpenCV is the second part.

Certificates

- Python Certificate
- Crash Course on Python (Google)

Skills

Python, R, Java, C++, HTML5, CSS, JavaScript

Tableau, MYSQL, Power BI, PostgreSQL

Advanced Microsoft Excel

Dhirubhai Ambani
Institute of Information and Communication Technology

SHASHVA MACHCHHAR

Msc Data Science

EDUCATION

Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT)
CPI: 9
August 2021 – Present | Gandhinagar, Gujarat

Gujarat University Bachelors in Data Science
CPI: 9.2
July 2019 – June 2021 | Ahmedabad, Gujarat

A.G High School (GHSEB)
Percentage: 84%
2018 – 2019 | Ahmedabad, Gujarat

St.Kabir School (GSEB)
Percentage: 74%
2016 – 2017 | Ahmedabad, Gujarat

SKILLS

Area(s) of Interest : Deep Learning, Machine Learning, Statistics.

Programming Languages : Python, R programming, SQL.

Tools and Technologies : Hadoop, Tableau, Data Visualization, Excel, Computer Vision.

Technical Electives : Exploratory Data Analysis

POSITIONS OF RESPONSIBILITY

Vice Head Boy of School

Student Council Played a major role in organising various events throughout the academic year and I served at one of the top most position of the student council.

August 2016 – September 2017

Core Member of Techno-Cultural Fest

Planned, executed and managed a triumphant Event

Sept 2019 – Feb 2020

Coordinated CONCOURS at DAIICT

Managed a Sports Event.

Sept 2022 – Dec 2022

PROJECTS

Image Caption Generation:

Jan 2021–June 2021

• Image Captioning is the process of generating textual description of an image. It uses both Natural Language Processing and Computer Vision to generate the captions. I made an encoder-decoder network such that it first extracted Features from images using pre-trained model VGG-16 and then generated relevant captions using LSTM from the extracted features. It is the task of Transfer learning.

• **Guide:** Dr. Ravi Gor

Predict car resale value using ML:

Oct 2020 - Dec 2020

• Developed a ML model using polynomial regression, polynomial with ridge regression, Decision tree and random forest after doing data cleaning and pre processing.

• Obtained the following results:

Accuracy on training data : 0.89

Accuracy on test data : 0.89

• **Guide:** Eric Shah (ML Prof at GU)

Deep Learning Guided Projects:

March 2021 - April 2021

• Car detection with YOLO, Image Segmentation with U-Net, Face Recognition, Art Generation with Neural Style Transfer, Speech recognition, Trigger Word Detection.

These projects were undertaken as a part of guided assignments in the course of deep learning offered by DeepLearning.ai.

• **Guide:** Andrew Ng.

CERTIFICATIONS

- Deep Learning specialization by Deeplearning.ai
- Python for everybody by University of Michigan
- Python and statistics for financial analysis by The Hong Kong University of science and Technology
- Introduction to MLops by Deeplearning.ai
- Intermediate Machine Learning by Kaggle

ACHIEVEMENTS

- Runners Up in State level Chess competition
- Achieved the highest academic performance in my bachelors program.

INTERESTS

- Sports
- Content lintening