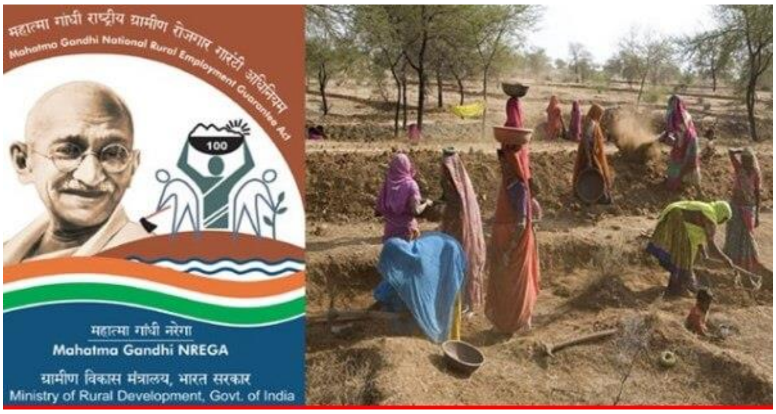
ODISHA MNREGA Report

The *Employment Generation Physical Progress Report* provides government stakeholders with the most granular and actionable policy insights due to its comprehensive tracking of employment demand, allocation, and utilization at the district level.

At the finest level of aggregation, this report allows policymakers to identify trends and disparities in employment distribution across different social groups—Scheduled Castes (SCs), Scheduled Tribes (STs), and Others—ensuring targeted policy interventions. By analysing job card issuance, deletion, and inclusion, the government can assess the effectiveness of employment outreach efforts and adjust strategies to improve accessibility.


महात्मा गांधी राष्ट्रीय ग्रामीण रोजगार गारंटी अधिनियम
Mahatma Gandhi National Rural Employment Guarantee Act
100
महात्मा गांधी नरेगा
Mahatma Gandhi NREGA
ग्रामीण विकास मंत्रालय, भारत सरकार
Ministry of Rural Development, Govt. of India

Moreover, the distinction between employment *demanded*, *offered*, and *availed* highlights potential gaps in job allocation and labour absorption, helping authorities address implementation bottlenecks. The breakdown of *person days* under central and state liability allows financial planning and monitoring of fund utilization, ensuring efficient allocation of resources.

Tracking *families completing 100 days of work* helps evaluate program reach and effectiveness, while the inclusion of land reform and disability beneficiaries enables social welfare analysis. These insights aid in refining poverty alleviation strategies and inclusive employment policies.

At a micro-level, district-wise data supports evidence-based decision-making, enabling targeted interventions in underperforming areas. The dataset empowers stakeholders to track employment trends, measure policy impact, and optimize resource allocation, ensuring equitable employment distribution under schemes like MGNREGA. Thus, the *Employment Generation Physical Progress Report* serves as a crucial tool for policy refinement and social equity enhancement.

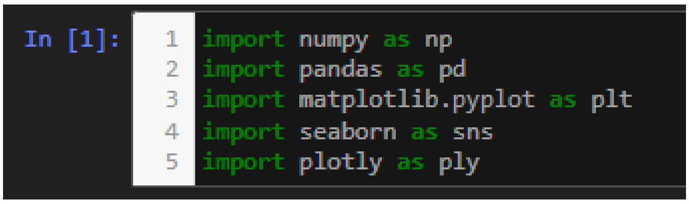
Variables used in the Dataset

- **District** – Name of the district where the data is collected.
- **No. of Registered Household** – Total number of households registered under the scheme.
- **No. of Registered Persons** – Total number of individuals registered under the scheme.
- **No. of Jobcard deleted in current YR Household** – Number of households whose job cards were canceled in the current financial year.
- **No. of Jobcard deleted in current YR Persons** – Number of individuals whose job cards were canceled in the current financial year.
- **No. of Jobcard included in current YR Household** – Number of households newly issued job cards in the current financial year.
- **No. of Jobcard included in current YR Persons** – Number of individuals newly issued job cards in the current financial year.
- **Cumulative No. of HH issued jobcards SCs** – Total number of job cards issued to Scheduled Caste (SC) households.
- **Cumulative No. of HH issued jobcards STs** – Total number of job cards issued to Scheduled Tribe (ST) households.
- **Cumulative No. of HH issued jobcards Others** – Total number of job cards issued to households from other categories (non-SC/ST).
- **Cumulative No. of HH issued jobcards Total** – Total number of job cards issued to all categories of households.
- **Employment demanded Household** – Number of households that have requested work under the scheme.
- **Employment demanded Persons** – Number of individuals who have requested work under the scheme.
- **Employment offered Household** – Number of households that have been provided employment.
- **Employment offered Persons** – Number of individuals who have been provided employment.
- **Employment Availed Household** – Number of households that have actually availed employment.
- **Employment Availed Persons** – Number of individuals who have actually availed employment.
- **Employment Availed Total Persondays** – Total number of workdays generated for individuals under the scheme.
- **Employment Availed Persondays of Central Liability** – Number of workdays paid for by the central government.
- **Employment Availed Persondays of States Liability** – Number of workdays paid for by the state government.
- **No. of Families Completed 100 days** – Number of families that have completed 100 days of employment in a year.
- **No. of HH which are beneficiary of land reform/IAY** – Number of households benefiting from land reforms or the Indira Awaas Yojana (IAY).
- **No. of Disabled beneficiary individuals** – Number of disabled individuals benefiting from the scheme.

All the above mention variable are of integer type except the District variable which is object type and unit of measurement for all the variables is whole number.

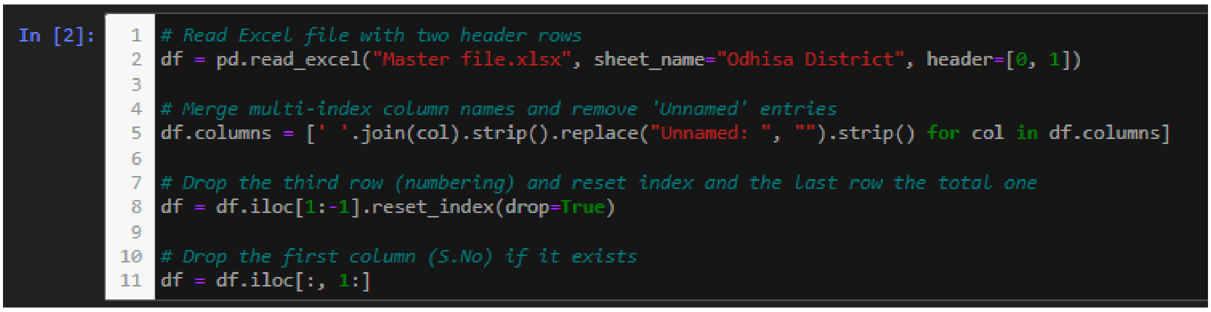
Data Analysis of Employment Generation Physical Progress Report of Odisha

Data Loading and Cleaning



```
In [1]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly as ply
```

Here basically the libraries were imported so that we can use the function which are there in those libraries on it. For example read_excel() function is used when we import the pandas library.



```
In [2]:
# Read Excel file with two header rows
df = pd.read_excel("Master file.xlsx", sheet_name="Odhisa District", header=[0, 1])

# Merge multi-index column names and remove 'Unnamed' entries
df.columns = [' '.join(col).strip().replace("Unnamed: ", "").strip() for col in df.columns]

# Drop the third row (numbering) and reset index and the last row the total one
df = df.iloc[1:-1].reset_index(drop=True)

# Drop the first column (S.No) if it exists
df = df.iloc[:, 1:]
```

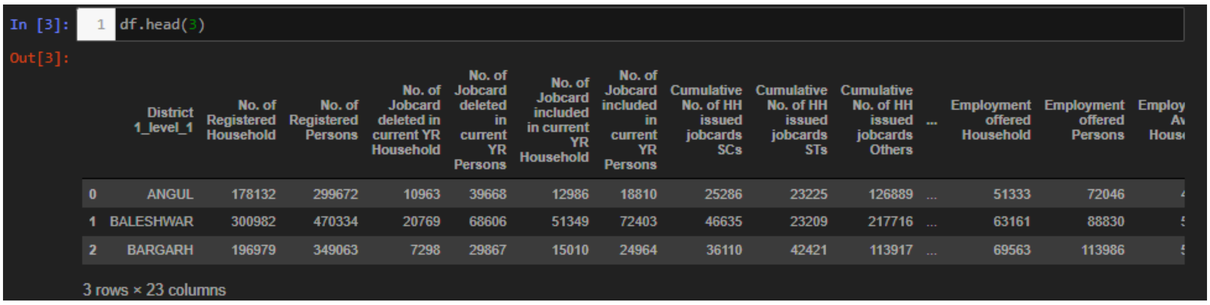
Here we are doing Data Preprocessing (or Data Cleaning). We are reading one of the sheets of the Master file with is in xlsx format. The 'header=[0,1]' indicates that we have a multi-index or hierarchical header. It helps to interpret the header rows of the data.

' '.join(col): This joins the elements of each column tuple col into a single string, separated by a space. So, ('Year', '') becomes "Year " and ('Population', 'Male') becomes "Population Male"

strip(): This removes any leading or trailing whitespace from the joined string. So "Year " becomes "Year"

replace("Unnamed: ",""): This replaces the string "Unnamed: " with an empty string. This is used to remove default column names that Pandas assigns when it encounters blank cells in the header rows of the Excel file.

df.iloc[1:-1]: This selects rows from the DataFrame using integer-based indexing. 1 is the starting row (inclusive), and -1 is the ending row (exclusive). So, this effectively *drops* the first row (index 0) and the *last* row of the DataFrame.

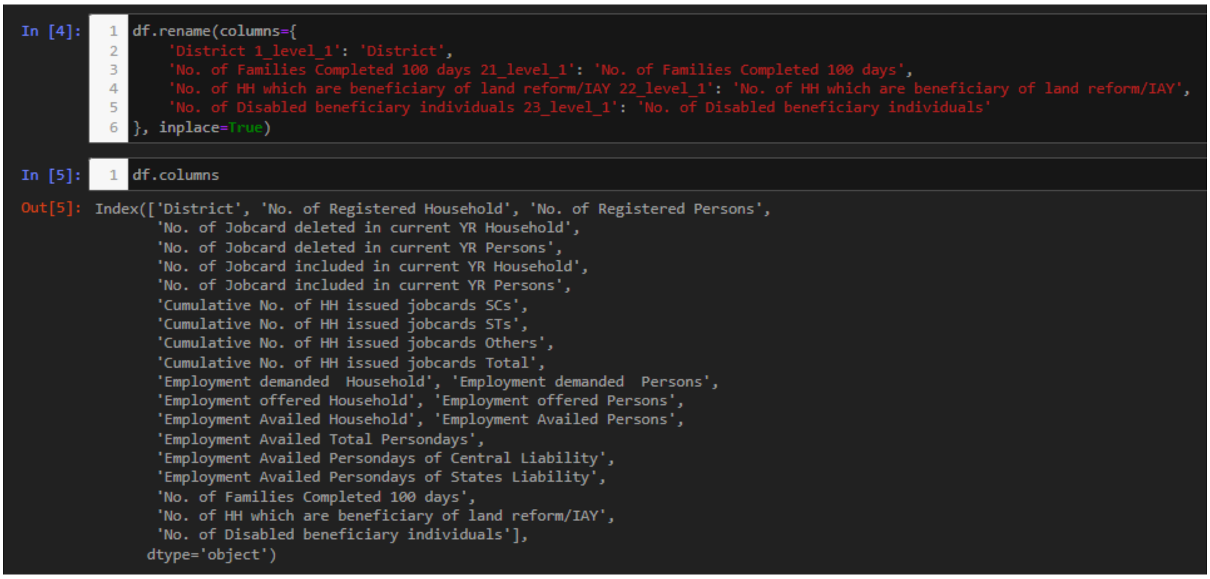


```
In [3]: df.head(3)
Out[3]:
```

	District 1_level_1	No. of Registered Household	No. of Registered Persons	No. of Jobcard deleted in current YR Household	No. of Jobcard deleted in current YR Persons	No. of Jobcard included in current YR Household	No. of Jobcard included in current YR Persons	Cumulative No. of HH issued jobcards SCs	Cumulative No. of HH issued jobcards STs	Cumulative No. of HH issued jobcards Others	...	Employment offered Household	Employment offered Persons	Employ Av Hous
0	ANGUL	178132	299672	10963	39668	12986	18810	25286	23225	126889	...	51333	72046	4
1	BALESHWAR	300982	470334	20769	68606	51349	72403	46635	23209	217716	...	63161	88830	5
2	BARGARH	196979	349063	7298	29867	15010	24964	36110	42421	113917	...	69563	113986	5

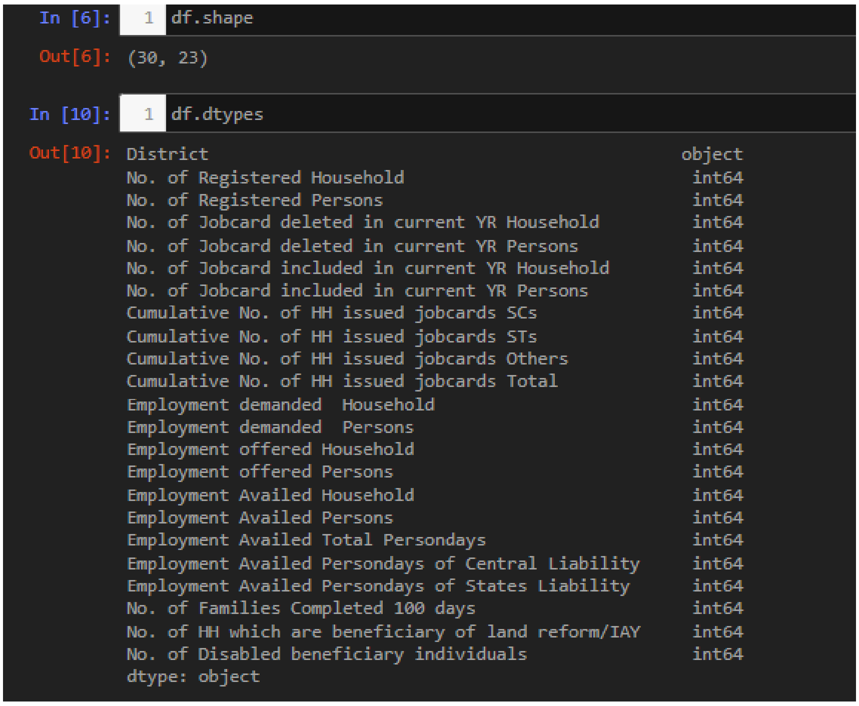
3 rows × 23 columns

Here we can just see the top 3 rows along with the header. It basically done to have a quick view about the dataset.



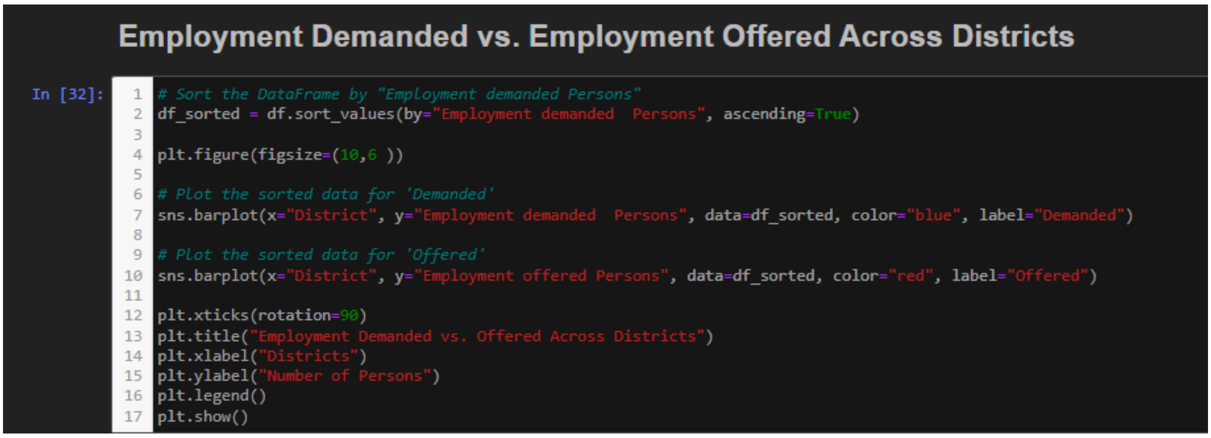
```
In [4]: df.rename(columns={
            'District 1_level_1': 'District',
            'No. of Families Completed 100 days 21_level_1': 'No. of Families Completed 100 days',
            'No. of HH which are beneficiary of land reform/IAY 22_level_1': 'No. of HH which are beneficiary of land reform/IAY',
            'No. of Disabled beneficiary individuals 23_level_1': 'No. of Disabled beneficiary individuals'
        }, inplace=True)

In [5]: df.columns

Out[5]: Index(['District', 'No. of Registered Household', 'No. of Registered Persons',
               'No. of Jobcard deleted in current YR Household',
               'No. of Jobcard deleted in current YR Persons',
               'No. of Jobcard included in current YR Household',
               'No. of Jobcard included in current YR Persons',
               'Cumulative No. of HH issued jobcards SCs',
               'Cumulative No. of HH issued jobcards STs',
               'Cumulative No. of HH issued jobcards Others',
               'Cumulative No. of HH issued jobcards Total',
               'Employment demanded  Household', 'Employment demanded  Persons',
               'Employment offered Household', 'Employment offered Persons',
               'Employment Availed Household', 'Employment Availed Persons',
               'Employment Availed Total Persondays',
               'Employment Availed Persondays of Central Liability',
               'Employment Availed Persondays of States Liability',
               'No. of Families Completed 100 days',
               'No. of HH which are beneficiary of land reform/IAY',
               'No. of Disabled beneficiary individuals'],
              dtype='object')
```

Here we have renamed the columns. For renaming we have give the existing name along with the new name. Then we checked whether it is it renamed properly or not.



```
In [6]: df.shape

Out[6]: (30, 23)

In [10]: df.dtypes

Out[10]: District                                              object
         No. of Registered Household                            int64
         No. of Registered Persons                              int64
         No. of Jobcard deleted in current YR Household         int64
         No. of Jobcard deleted in current YR Persons           int64
         No. of Jobcard included in current YR Household        int64
         No. of Jobcard included in current YR Persons          int64
         Cumulative No. of HH issued jobcards SCs               int64
         Cumulative No. of HH issued jobcards STs               int64
         Cumulative No. of HH issued jobcards Others            int64
         Cumulative No. of HH issued jobcards Total             int64
         Employment demanded  Household                         int64
         Employment demanded  Persons                           int64
         Employment offered Household                           int64
         Employment offered Persons                             int64
         Employment Availed Household                           int64
         Employment Availed Persons                             int64
         Employment Availed Total Persondays                    int64
         Employment Availed Persondays of Central Liability     int64
         Employment Availed Persondays of States Liability      int64
         No. of Families Completed 100 days                     int64
         No. of HH which are beneficiary of land reform/IAY     int64
         No. of Disabled beneficiary individuals                int64
         dtype: object
```

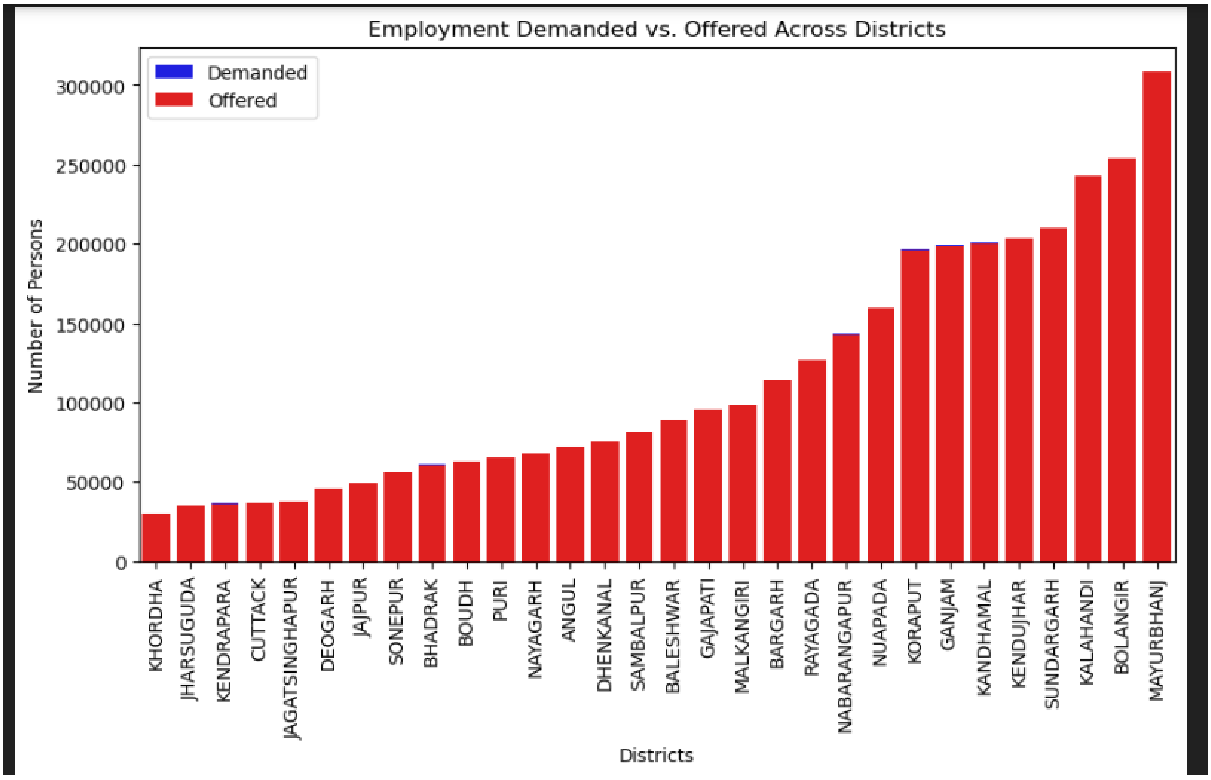
Here we check how many rows and columns are there in the dataset. Then we check what are the data types of all the columns.



Employment Demanded vs. Employment Offered Across Districts

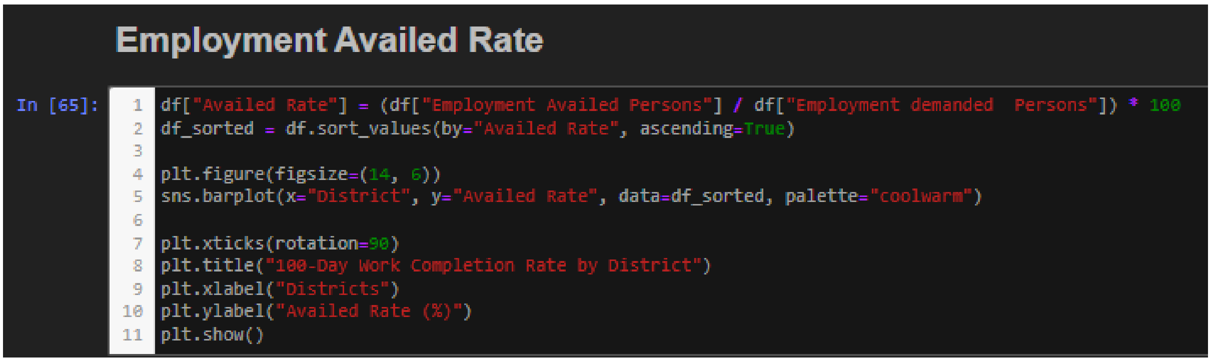
```
In [32]:
# Sort the DataFrame by "Employment demanded Persons"
df_sorted = df.sort_values(by="Employment demanded  Persons", ascending=True)

plt.figure(figsize=(10,6 ))

# Plot the sorted data for 'Demanded'
sns.barplot(x="District", y="Employment demanded  Persons", data=df_sorted, color="blue", label="Demanded")

# Plot the sorted data for 'Offered'
sns.barplot(x="District", y="Employment offered Persons", data=df_sorted, color="red", label="Offered")

plt.xticks(rotation=90)
plt.title("Employment Demanded vs. Offered Across Districts")
plt.xlabel("Districts")
plt.ylabel("Number of Persons")
plt.legend()
plt.show()
```

Employment Demanded vs. Offered: This code visualizes the employment demand and offer across districts using a grouped bar chart. It sorts the data by employment demand and then plots demanded and offered employment side-by-side for each district. sns.barplot() is used to create the bar plots, and plt.xticks() rotates the x-axis labels for better readability. This chart helps in understanding the gap between demand and offer in each district.



Here we can see number of jobs demanded which is highlight with blue and number of jobs offer which is highlighted in red. Here district Khordha has been offered the least number of jobs whereas district Mayurbhanj has been offered the most number of jobs.

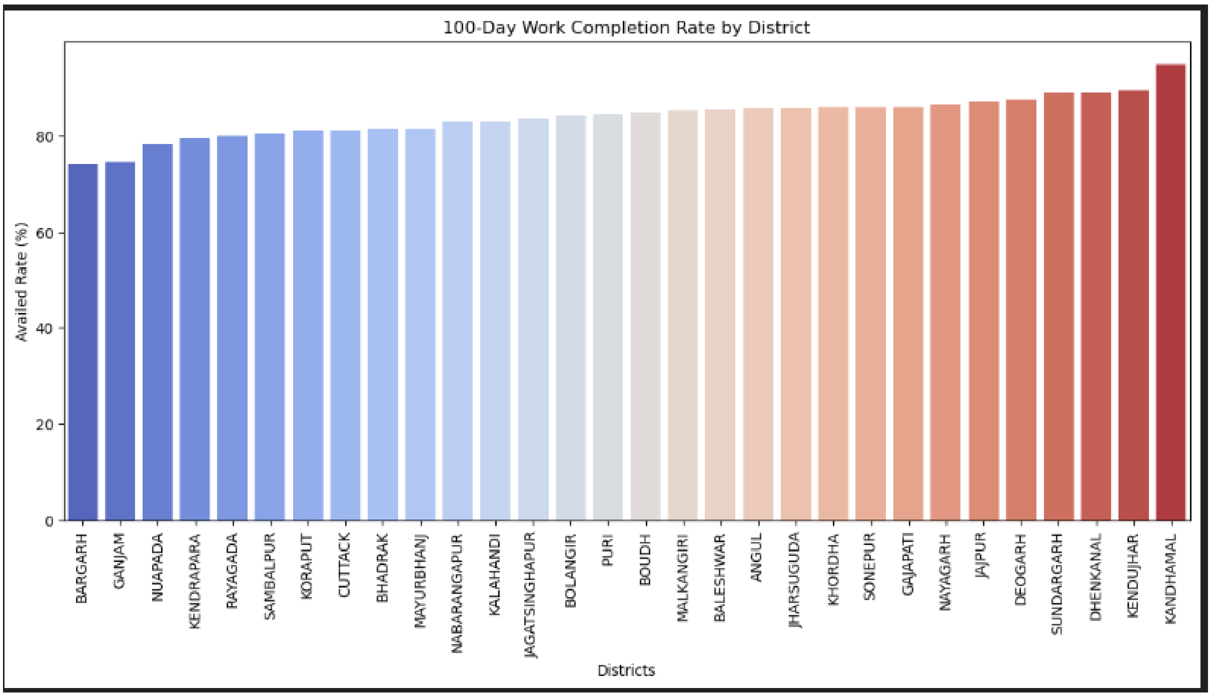
District such as Kendrapara, Bhadrak, Nabarangapur, Koraput ,Ganjam, Kandhamal here the job demanded was more than the job offered.



Employment Availed Rate

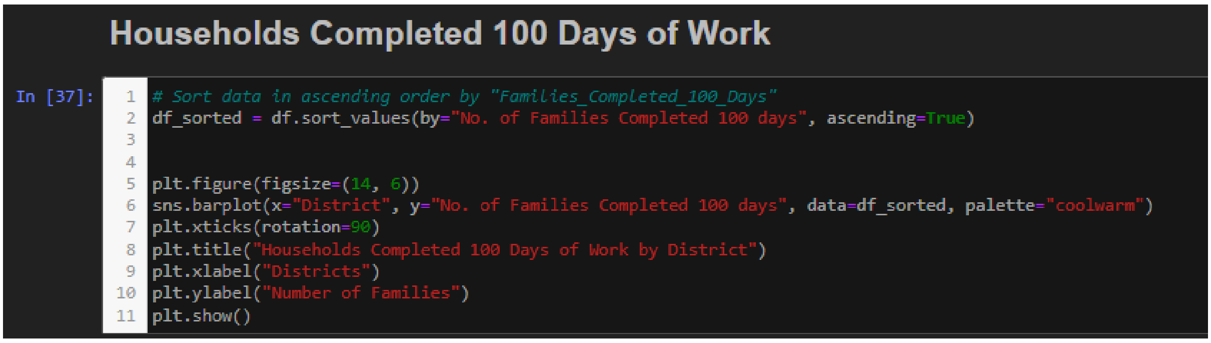
```
In [65]:
df["Availed Rate"] = (df["Employment Availed Persons"] / df["Employment demanded  Persons"]) * 100
df_sorted = df.sort_values(by="Availed Rate", ascending=True)

plt.figure(figsize=(14, 6))
sns.barplot(x="District", y="Availed Rate", data=df_sorted, palette="coolwarm")

plt.xticks(rotation=90)
plt.title("100-Day Work Completion Rate by District")
plt.xlabel("Districts")
plt.ylabel("Availed Rate (%)")
plt.show()
```

Employment Availed Rate: This snippet calculates and displays the employment availed rate for each district. It calculates the rate as a percentage and stores it in a new column. The data is then sorted by this rate, and a bar plot is generated to show the availed rate for each district. sns.barplot() visualizes the availed rate, providing insights into how effectively the offered employment is being utilized.



Here we can see the Employment Availed Rate by person. District Kandhamal has the highest availed rate where as Bargarh has the lowest avail rate. The formula for the calculating Availed rate is

Availed Rate = (Employment Availed Persons/ Employment demanded Persons) * 100

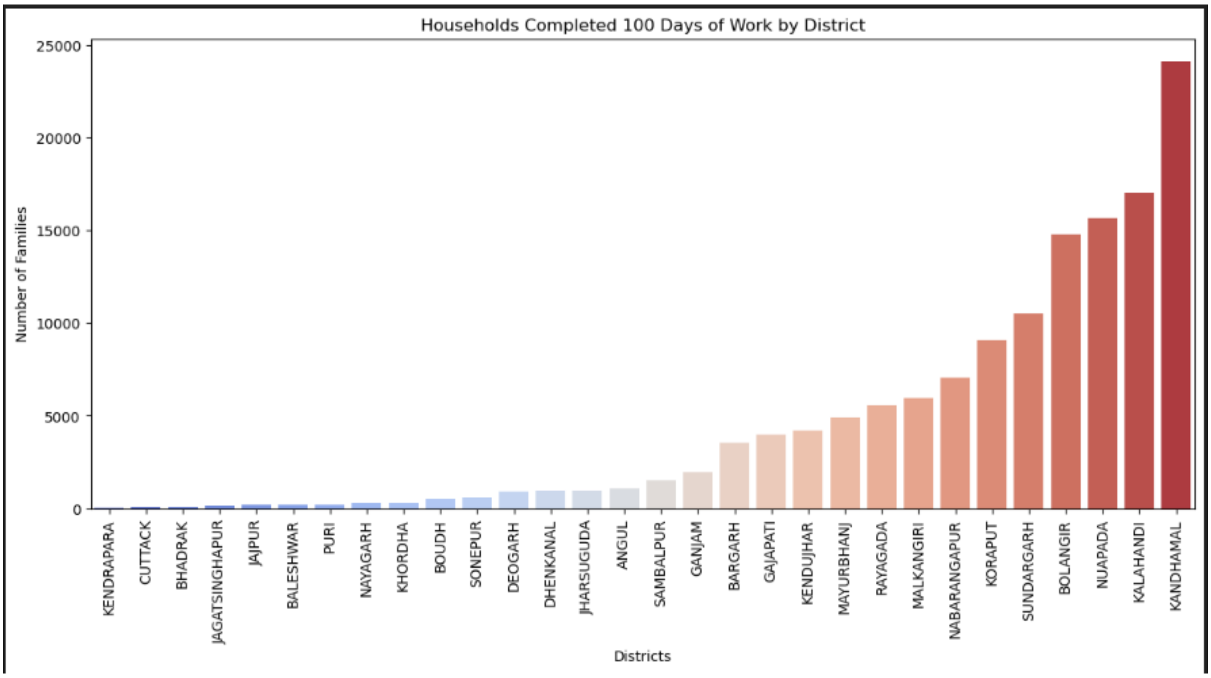


Households Completed 100 Days of Work

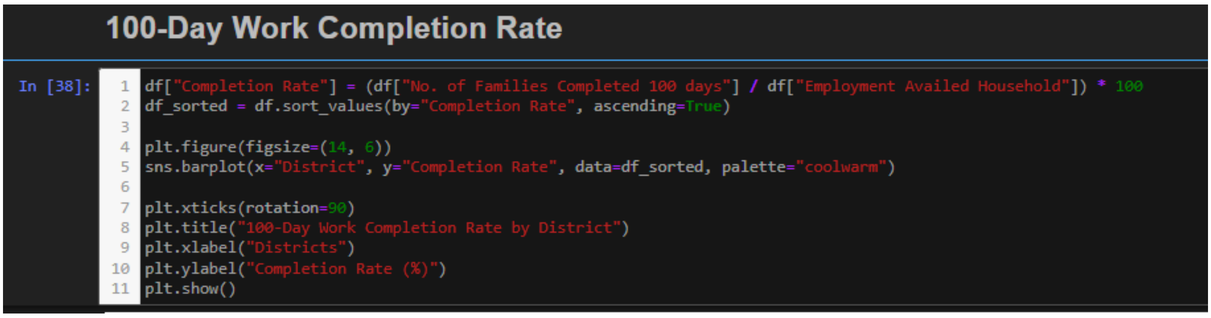
In [37]:

```
# Sort data in ascending order by "Families_Completed_100_Days"
df_sorted = df.sort_values(by="No. of Families Completed 100 days", ascending=True)


plt.figure(figsize=(14, 6))
sns.barplot(x="District", y="No. of Families Completed 100 days", data=df_sorted, palette="coolwarm")
plt.xticks(rotation=90)
plt.title("Households Completed 100 Days of Work by District")
plt.xlabel("Districts")
plt.ylabel("Number of Families")
plt.show()
```

Households Completed 100 Days of Work: This section visualizes the number of households that completed 100 days of work in each district. The data is sorted by the number of households completing 100 days, and a bar plot is generated to display this information for each district. sns.barplot() is used to create the chart, highlighting the districts with the highest and lowest completion rates.



Here the number of household by district that have successfully completed the 100 days of work. District KANDHAMAL has the highest number of household that have successfully completed the 100 days of work where as KENDRAPARA is the district which has the least number households which have completed 100 days of work.

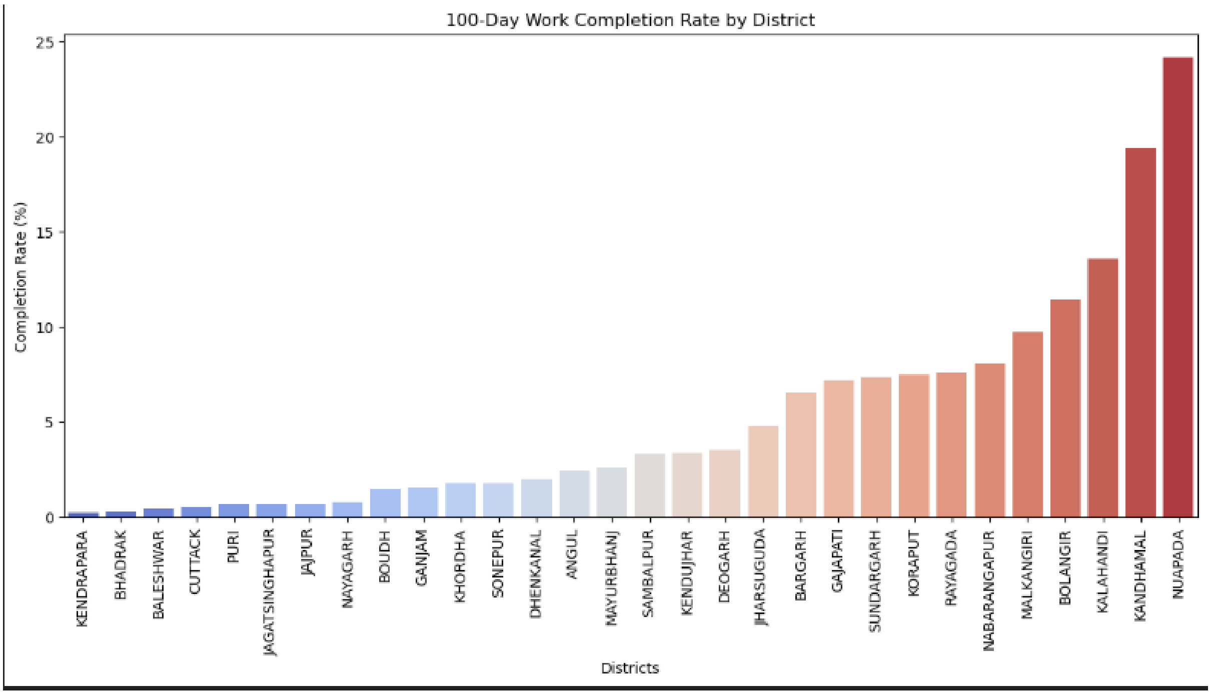


100-Day Work Completion Rate

In [38]:

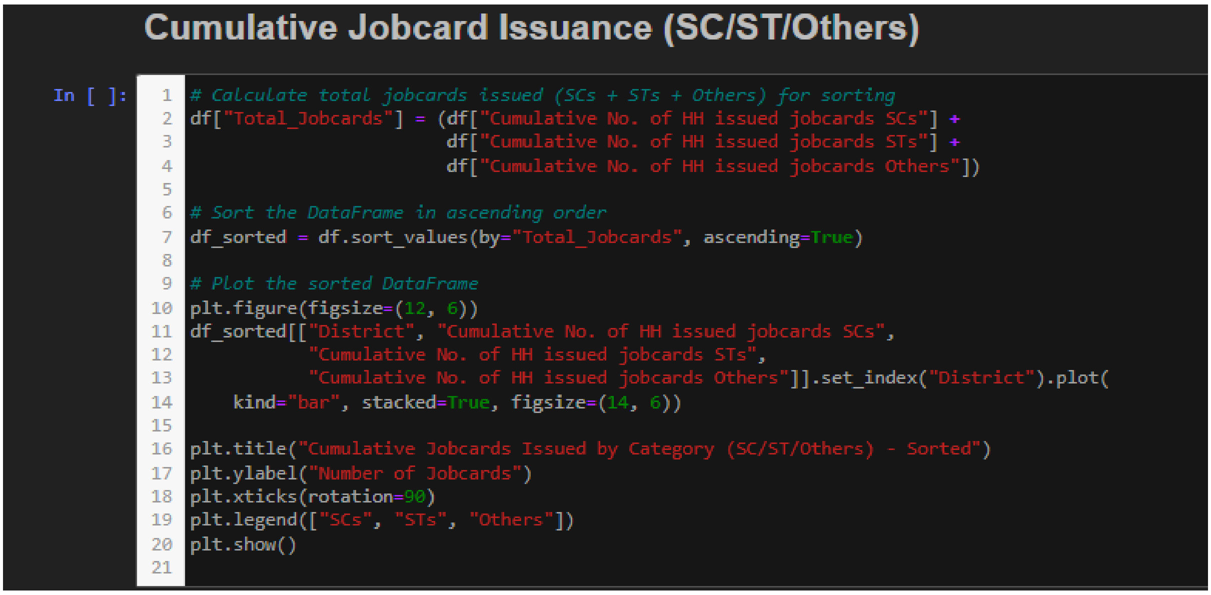
```
df["Completion Rate"] = (df["No. of Families Completed 100 days"] / df["Employment Availed Household"]) * 100
df_sorted = df.sort_values(by="Completion Rate", ascending=True)

plt.figure(figsize=(14, 6))
sns.barplot(x="District", y="Completion Rate", data=df_sorted, palette="coolwarm")

plt.xticks(rotation=90)
plt.title("100-Day Work Completion Rate by District")
plt.xlabel("Districts")
plt.ylabel("Completion Rate (%)")
plt.show()
```

100-Day Work Completion Rate: This code calculates and visualizes the 100-day work completion rate for each district. The rate is calculated as a percentage, and the data is sorted by this rate. A bar plot is then created to show the completion rate for each district. sns.barplot() visualizes the completion rate, indicating the districts where the 100-day work target is being achieved most effectively.



Here we can see the 100 Days Completion Rate. District Nuapada has the highest Completion Rate where as Kendrapara has the lowest Complete Rate. The formula for the calculating completion rate is

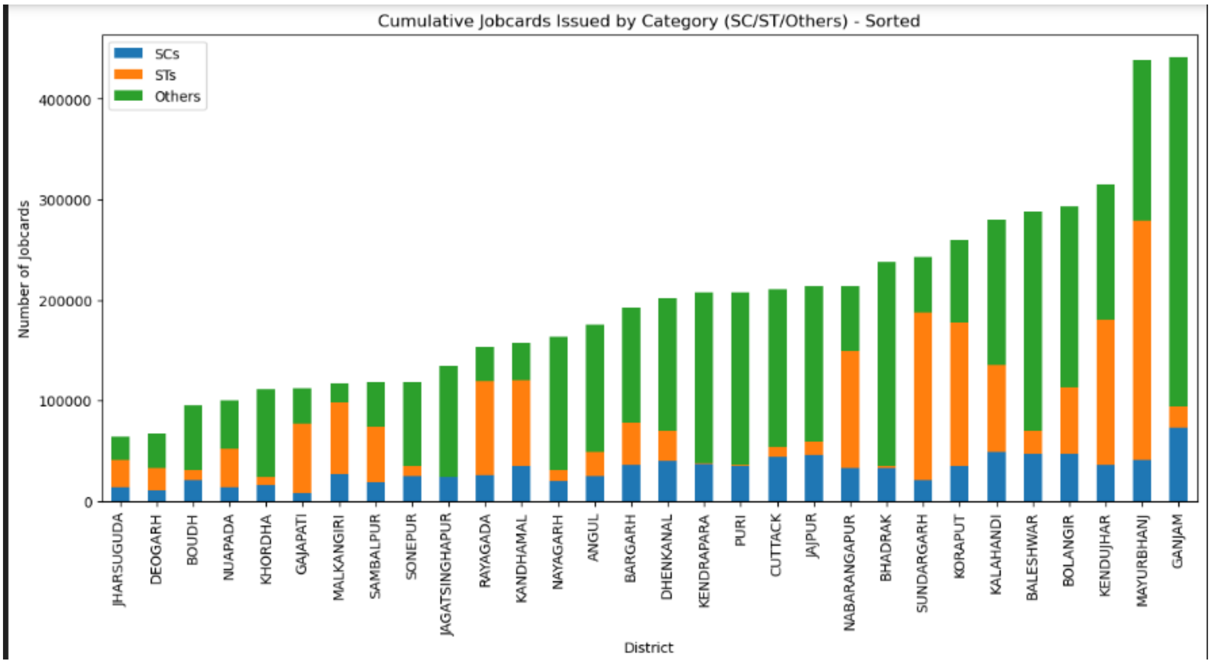
100 Days Completion Rate = (No. of Families Completed 100 days/ Employment Availed Household) * 100



Cumulative Jobcard Issuance (SC/ST/Others)

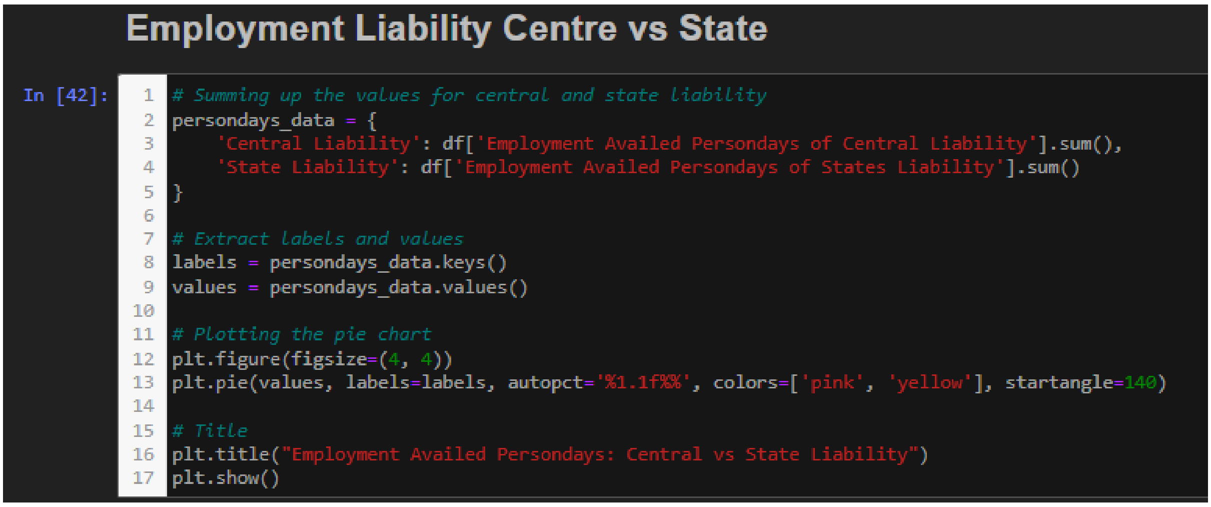
```python
# Calculate total jobcards issued (SCs + STs + Others) for sorting
df["Total_Jobcards"] = (df["Cumulative No. of HH issued jobcards SCs"] +
                        df["Cumulative No. of HH issued jobcards STs"] +
                        df["Cumulative No. of HH issued jobcards Others"])

# Sort the DataFrame in ascending order
df_sorted = df.sort_values(by="Total_Jobcards", ascending=True)

# Plot the sorted DataFrame
plt.figure(figsize=(12, 6))
df_sorted[["District", "Cumulative No. of HH issued jobcards SCs",
           "Cumulative No. of HH issued jobcards STs",
           "Cumulative No. of HH issued jobcards Others"]].set_index("District").plot(
    kind="bar", stacked=True, figsize=(14, 6))

plt.title("Cumulative Jobcards Issued by Category (SC/ST/Others) - Sorted")
plt.ylabel("Number of Jobcards")
plt.xticks(rotation=90)
plt.legend(["SCs", "STs", "Others"])
plt.show()

```

Cumulative Jobcard Issuance (SC/ST/Others): This code visualizes the cumulative job card issuance for SC, ST, and Other households across districts. It calculates the total job cards issued and sorts the data accordingly. A stacked bar chart is created to show the breakdown of job card issuance by category (SC, ST, Others) for each district. df.plot(kind="bar", stacked=True) creates the stacked bar chart, illustrating the distribution of job cards among different social groups.



Here in this visualization we can see the Cumulative Jobcards Issued for the 3 categories. The 3 categories are SCs which is highlighted in blue , STs which is highlighted in yellow and Others which is highlighted in green.

Here we can see that GANJAM has most number of job cards issue and in which the Others category is the highest number of job seekers. For STs category highest number of card issue in MAYURBHANJ district. And for SCs category also highest number of job seekers are from GANJAM district only.

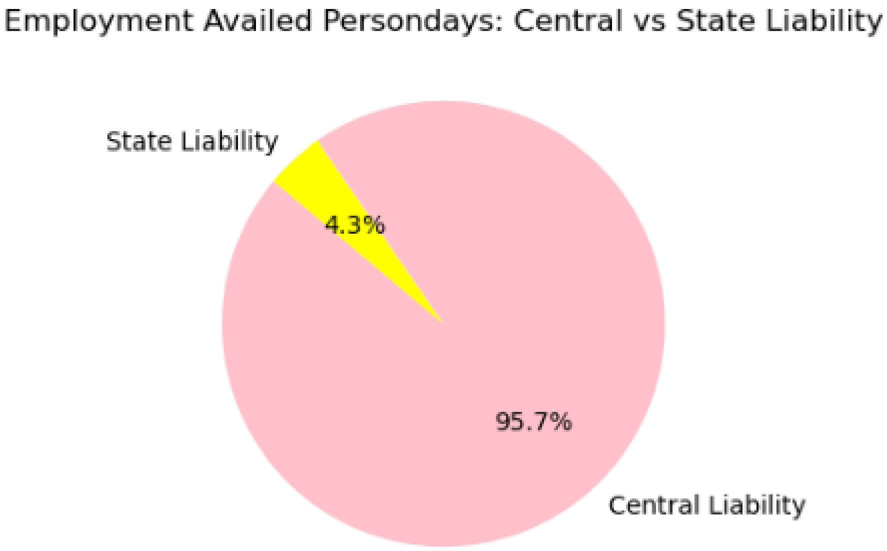


Employment Liability Centre vs State

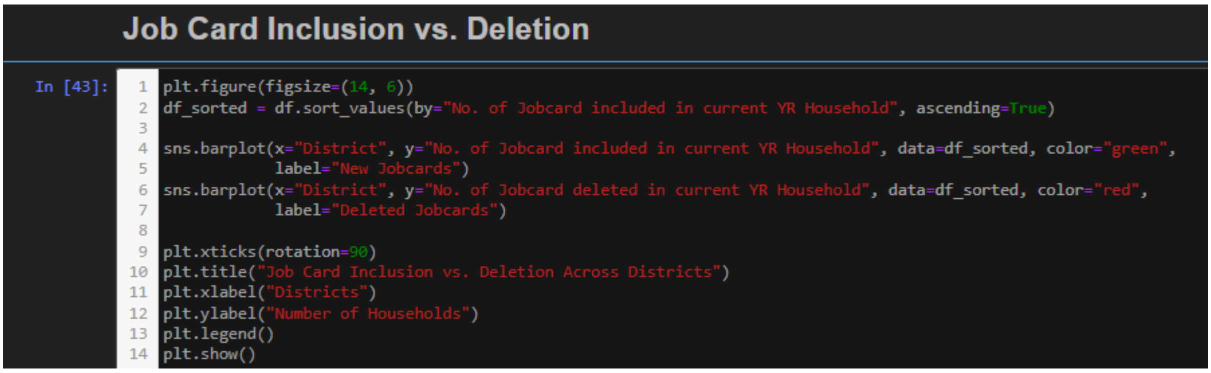
In [42]:

```
# Summing up the values for central and state liability
persondays_data = {
    'Central Liability': df['Employment Availed Persondays of Central Liability'].sum(),
    'State Liability': df['Employment Availed Persondays of States Liability'].sum()
}

# Extract labels and values
labels = persondays_data.keys()
values = persondays_data.values()

# Plotting the pie chart
plt.figure(figsize=(4, 4))
plt.pie(values, labels=labels, autopct='%1.1f%%', colors=['pink', 'yellow'], startangle=140)

# Title
plt.title("Employment Availed Persondays: Central vs State Liability")
plt.show()
```

Employment Liability Centre vs State: This snippet creates a pie chart comparing the total employment availed person-days under Central and State liability. It sums the person-days for each category and then generates a pie chart to show the proportion of each. plt.pie() creates the pie chart, providing a clear view of the contribution of central and state governments to employment generation.

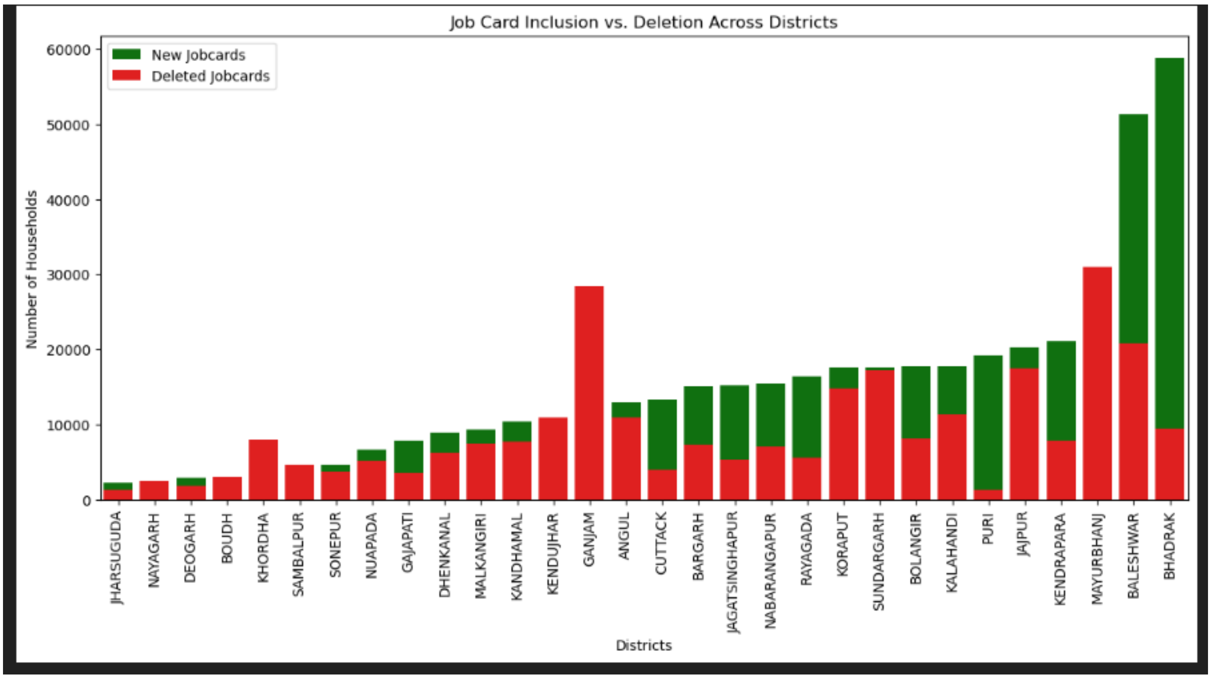


Here we can see the share of money that is allotted for the MNREGA scheme. Here Central government supports majorly with 95.7% of liability then the state government with 4.3% of liability.

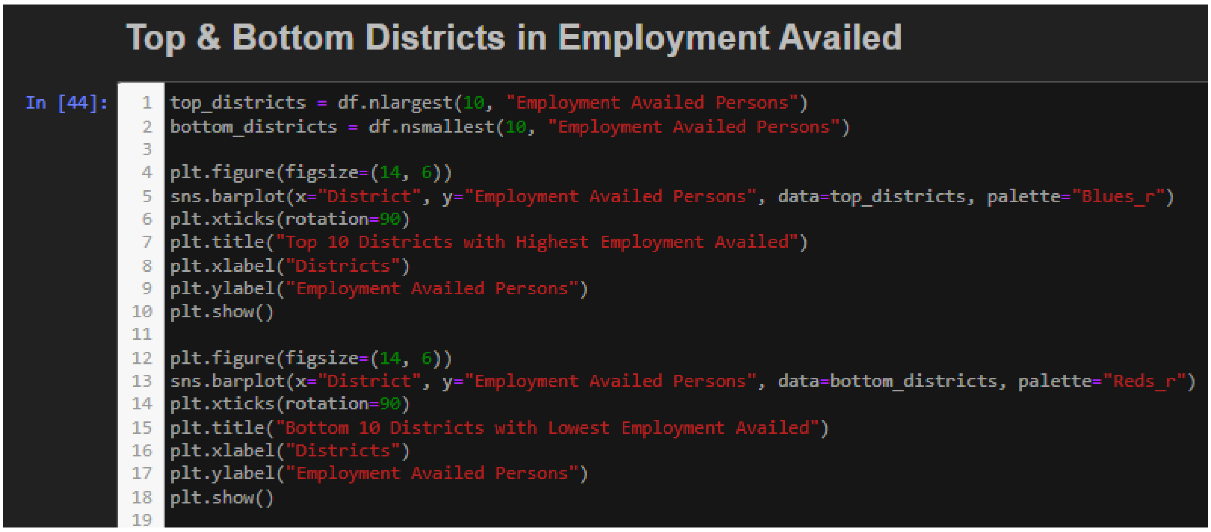

Job Card Inclusion vs. Deletion

In [43]:
```
plt.figure(figsize=(14, 6))
df_sorted = df.sort_values(by="No. of Jobcard included in current YR Household", ascending=True)

sns.barplot(x="District", y="No. of Jobcard included in current YR Household", data=df_sorted, color="green",
            label="New Jobcards")
sns.barplot(x="District", y="No. of Jobcard deleted in current YR Household", data=df_sorted, color="red",
            label="Deleted Jobcards")

plt.xticks(rotation=90)
plt.title("Job Card Inclusion vs. Deletion Across Districts")
plt.xlabel("Districts")
plt.ylabel("Number of Households")
plt.legend()
plt.show()
```

Job Card Inclusion vs. Deletion: This code visualizes the number of job cards included and deleted in the current year for each district. The data is sorted by the number of job cards included, and a grouped bar chart is created to show both inclusions and deletions for each district. sns.barplot() is used to create the bar plots, allowing for a direct comparison of new job cards versus those that were removed.



Here in this stacked bar plot green denotes the new job cards issued and red denotes the deleted job cards. BHADRAK has the most number of new green card issue. Where as GANJAM and MAYURBHANJ has most number of job card deletion. Plus negligible new card has been issued in GANJAM , MAYURBHANJ, NAYAGARH, KHORDHA, BOUDH, KENDUJHAR.

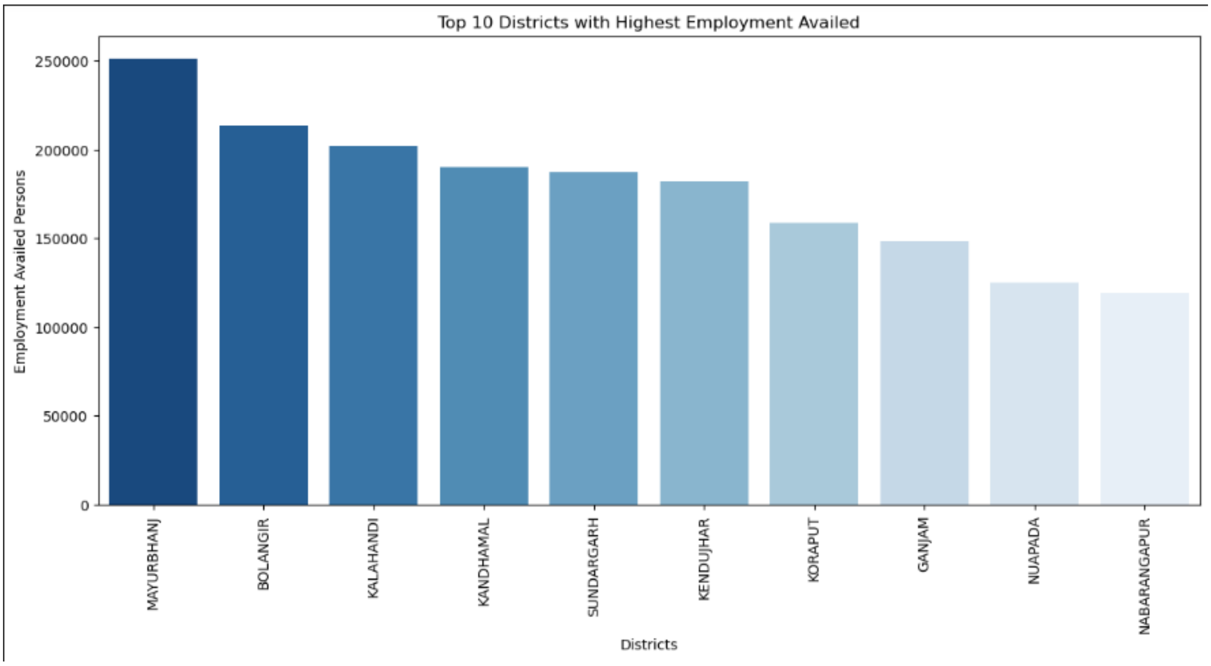


Top & Bottom Districts in Employment Availed

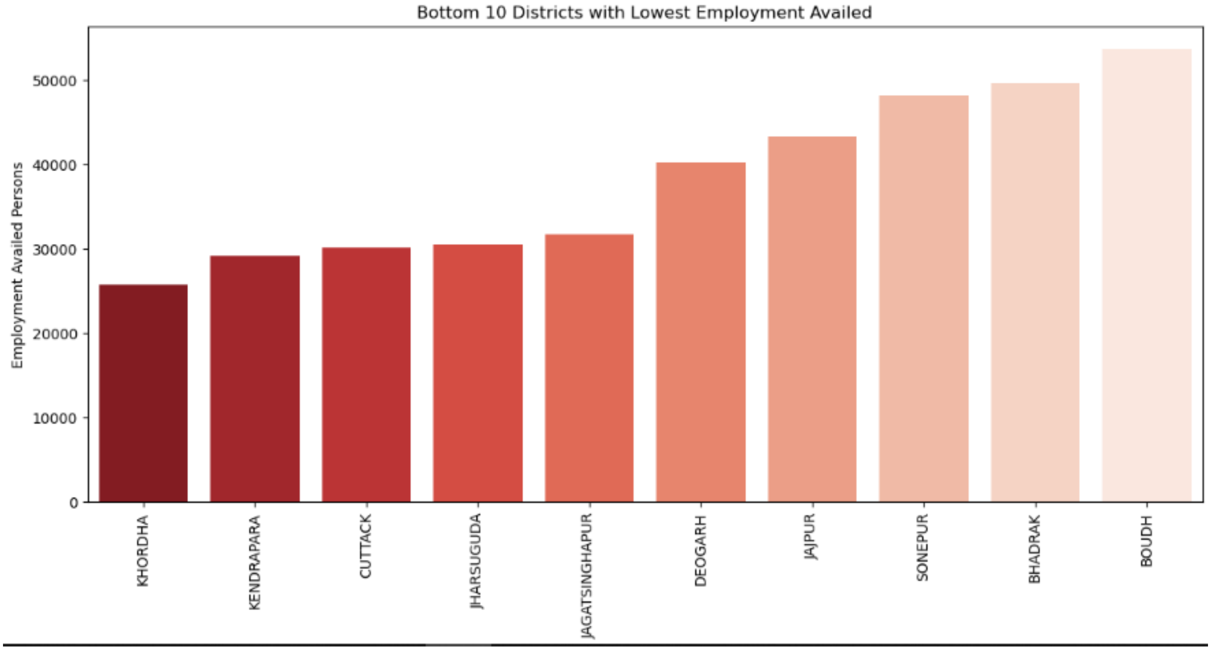
In [44]:

```
top_districts = df.nlargest(10, "Employment Availed Persons")
bottom_districts = df.nsmallest(10, "Employment Availed Persons")

plt.figure(figsize=(14, 6))
sns.barplot(x="District", y="Employment Availed Persons", data=top_districts, palette="Blues_r")
plt.xticks(rotation=90)
plt.title("Top 10 Districts with Highest Employment Availed")
plt.xlabel("Districts")
plt.ylabel("Employment Availed Persons")
plt.show()

plt.figure(figsize=(14, 6))
sns.barplot(x="District", y="Employment Availed Persons", data=bottom_districts, palette="Reds_r")
plt.xticks(rotation=90)
plt.title("Bottom 10 Districts with Lowest Employment Availed")
plt.xlabel("Districts")
plt.ylabel("Employment Availed Persons")
plt.show()

```

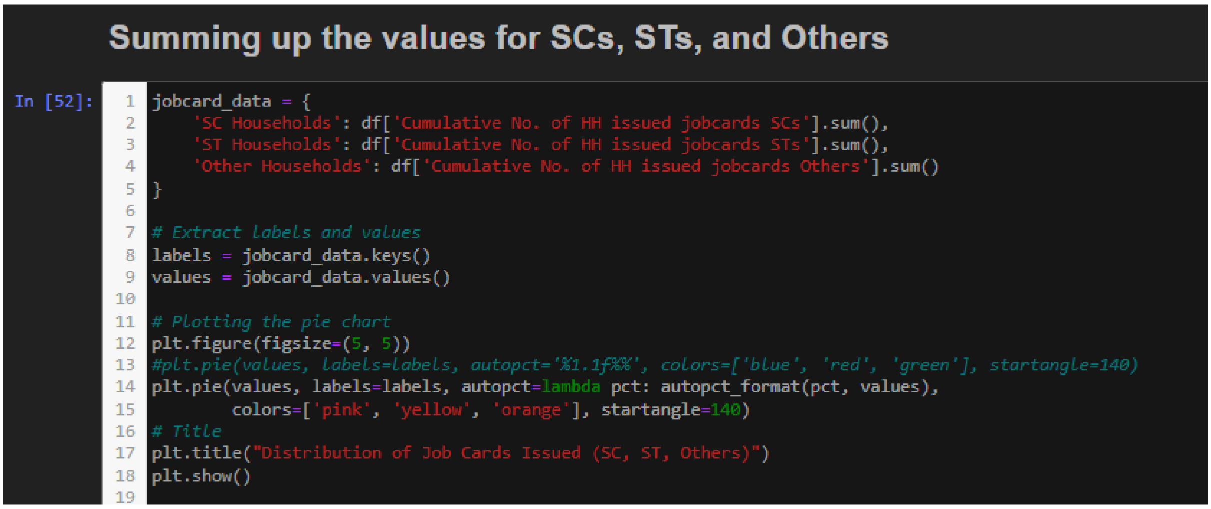
Top & Bottom Districts in Employment Availed: This section identifies and visualizes the top 10 and bottom 10 districts based on the number of persons who availed employment. It uses nlargest() and nsmallest() to get these districts and then generates separate bar plots for each. sns.barplot() is used to create the charts, highlighting the best and worst performing districts in terms of employment availed.



Here we can see the top 10 districts which has highest Employment Availed. MAYURBHANJ is the highest employment availed ,then followed by BOLANGIR with second highest employment availed , and then KALAHANDI.



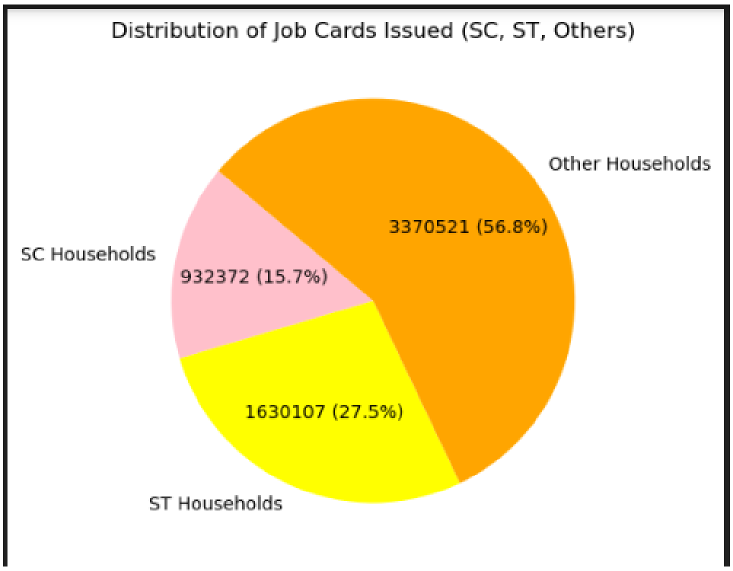
Here we can see the bottom 10 districts which has lowest Employment Availed. KHORDHA is the lowest employment availed ,then followed by KENDRAPARA with second lowest employment availed , and then CUTTACK.



Summing up the values for SCs, STs, and Others

```
In [52]:
jobcard_data = {
    'SC Households': df['Cumulative No. of HH issued jobcards SCs'].sum(),
    'ST Households': df['Cumulative No. of HH issued jobcards STs'].sum(),
    'Other Households': df['Cumulative No. of HH issued jobcards Others'].sum()
}

# Extract labels and values
labels = jobcard_data.keys()
values = jobcard_data.values()

# Plotting the pie chart
plt.figure(figsize=(5, 5))
#plt.pie(values, labels=labels, autopct='%1.1f%%', colors=['blue', 'red', 'green'], startangle=140)
plt.pie(values, labels=labels, autopct=lambda pct: autopct_format(pct, values),
        colors=['pink', 'yellow', 'orange'], startangle=140)
# Title
plt.title("Distribution of Job Cards Issued (SC, ST, Others)")
plt.show()

```

Distribution of Job Cards Issued (SC, ST, Others): This snippet creates a pie chart showing the overall distribution of job cards issued to SC, ST, and Other households. It calculates the total job cards issued for each category and generates a pie chart to show the proportion of each. plt.pie() creates the pie chart. A custom function, autopct_format, is used to display both the percentage and the absolute number of households on each slice of the pie chart, providing more detailed information.

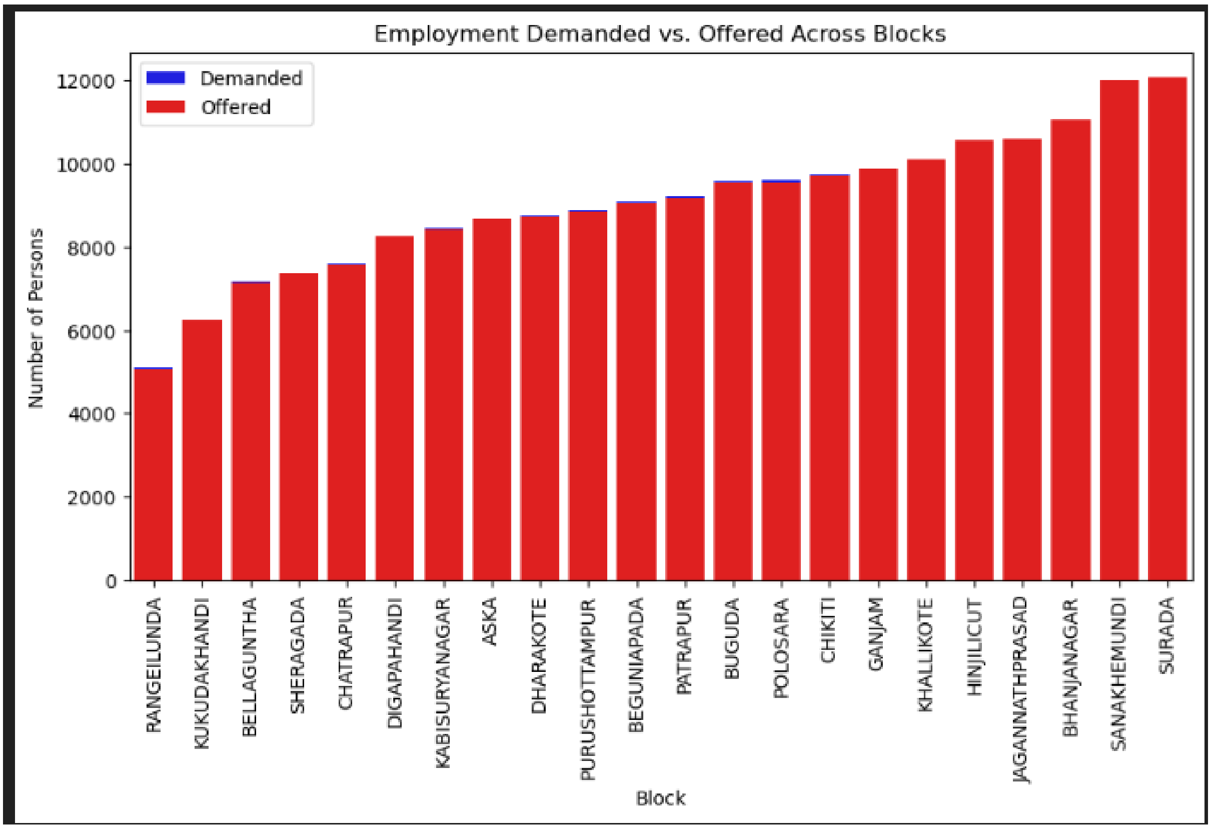


In this Pie chart we can see this job cards issued category wise. The most number of job cards has been issued for the Others category which is 56.8%, then followed by STs category with27.5% and then SCs category with 15.7%.

Conclusion & Government Actions

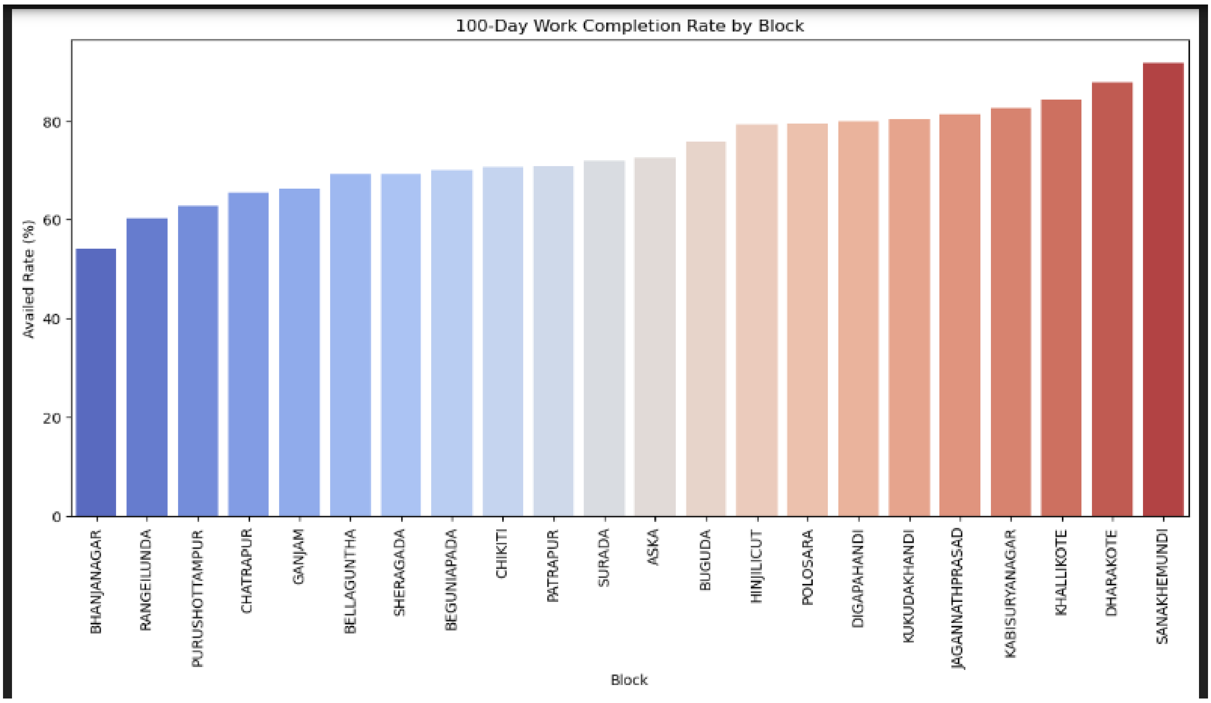
1. **Increase job allocations** in districts where employment demand exceeds supply.
2. **Investigate job card deletions** to ensure genuine workers are not being removed unfairly.
3. **Ensure fair job distribution** among SC/ST groups by improving awareness and accessibility.
4. **Strengthen work availability** in districts with low 100-day work completion rates.
5. **Reallocate MNREGA funding** if the state government is bearing a disproportionate share.
6. **Enhance employment awareness programs** in districts where availed employment is low.

Data Analysis of Employment Generation Physical Progress Report of Odisha’s Ganjam District



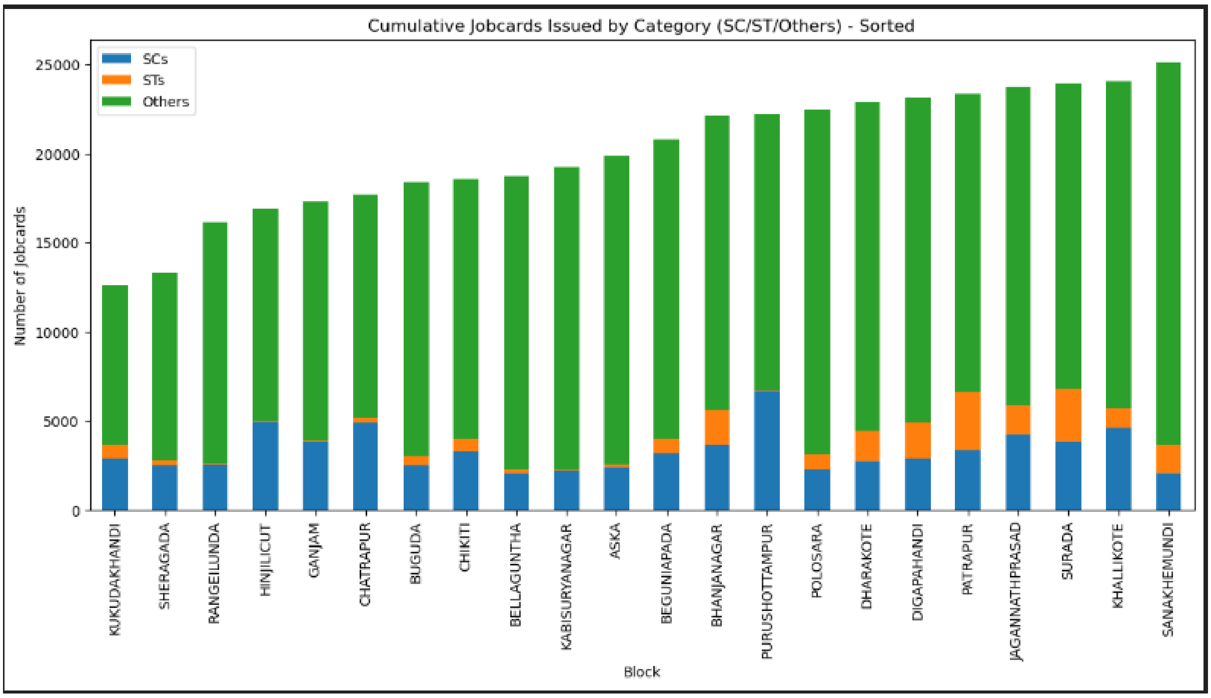
Here we can see number of jobs demanded which is highlight with blue and number of jobs offer which is highlighted in red. Here block RANGEILUNDA has been offered the least number of jobs whereas block SANAKHEMUNDI , SURADA has been offered the most number of jobs.

Blocks such as BUGUDA,POLOSARA,PURUSHOTTAMPUR,DHARAKOTE,RANGEILUNDA,BELLAGUNTHA here the job demanded was more than the job offered.



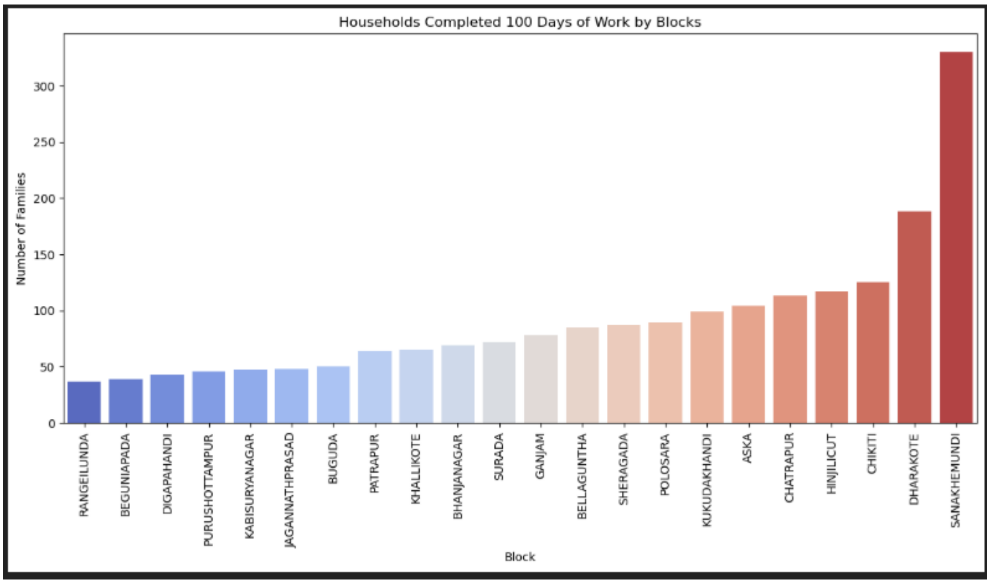
Here we can see the Employment Availed Rate by person. Block SANAKHEMUNDI has the highest availed rate where as BHANJANAGAR has the lowest avail rate. The formula for the calculating Availed rate is

Availed Rate = (Employment Availed Persons/ Employment demanded Persons) * 100

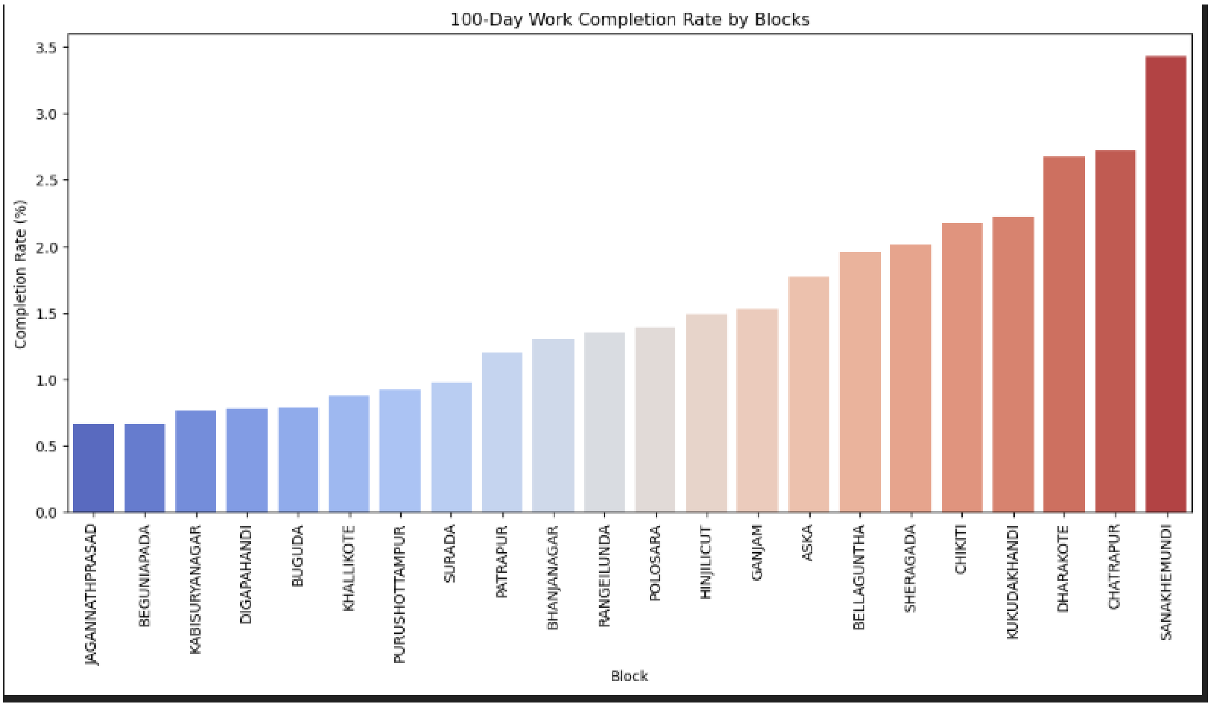


Here in this visualization we can see the Cumulative Jobcards Issued for the 3 categories. The 3 categories are SCs which is highlighted in blue , STs which is highlighted in yellow and Others which is highlighted in green.

Here we can see that SANAKHEMUNDI has most number of job cards issue and in which the Others category is the highest number of job seekers. For STs category highest number of card issue in PATRAPUR block. And for SCs category also highest number of job seekers are from PURUSHOTTAMPUR block.

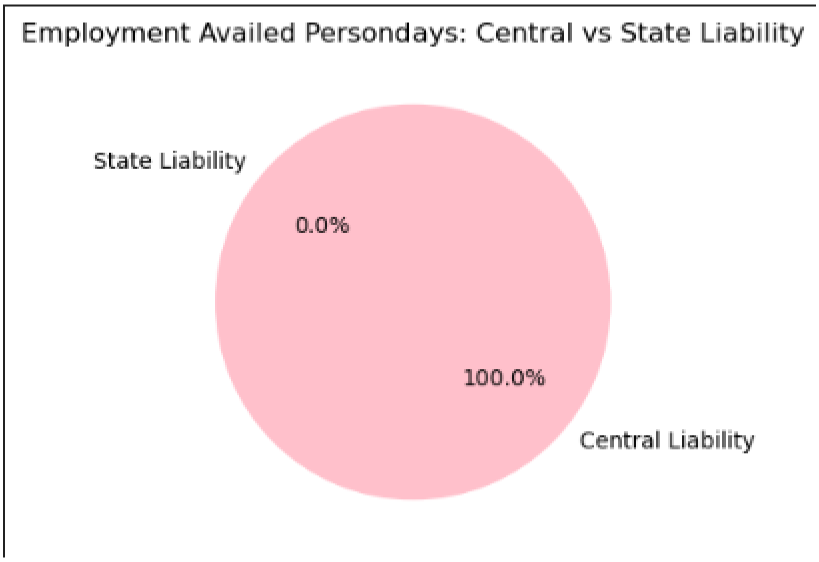


Here the number of household by district that have successfully completed the 100 days of work. Block SANAKHEMUNDI has the highest number of household that have successfully completed the 100 days of work where as RANGEILUNDA is the block which has the least number households which have completed 100 days of work.

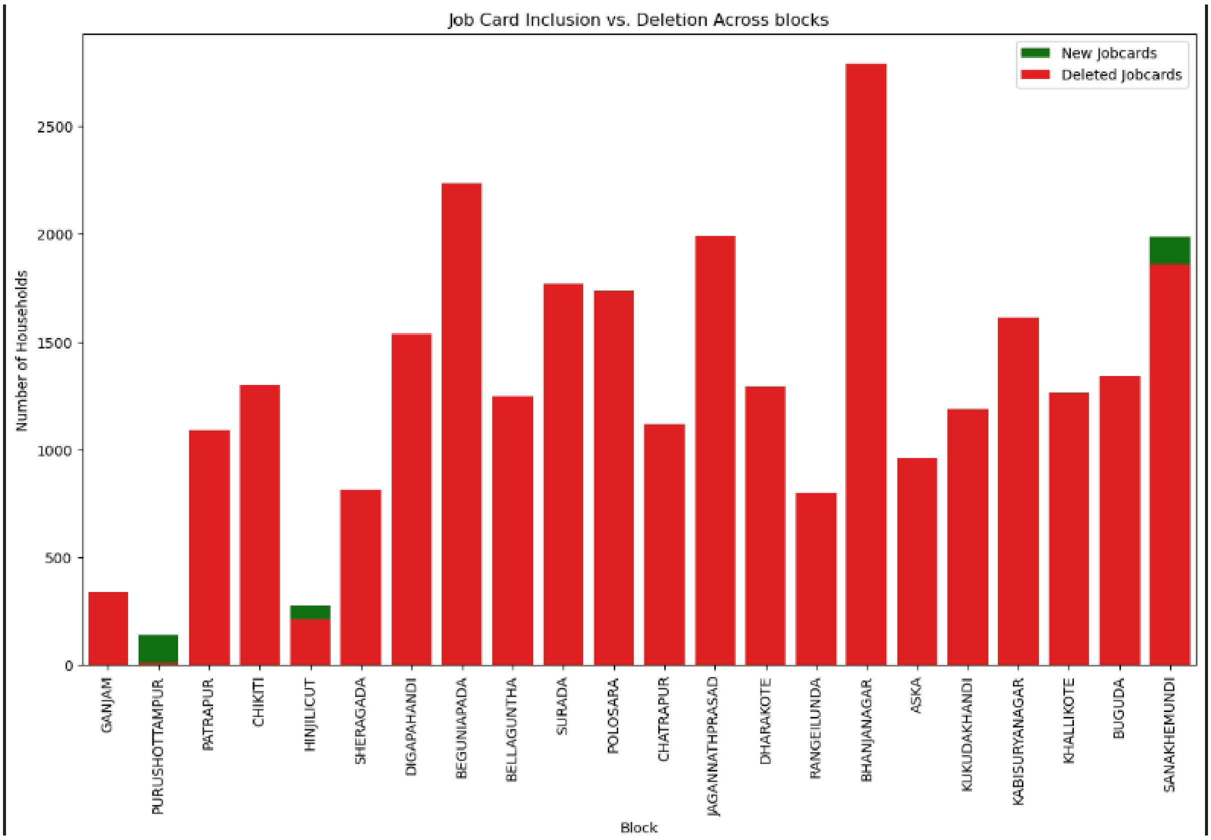


Here we can see the 100 Days Completion Rate.Block SANAKHEMUNDI has the highest Completion Rate where as JAGANNATHPRASAD has the lowest Complete Rate. The formula for the calculating completion rate is

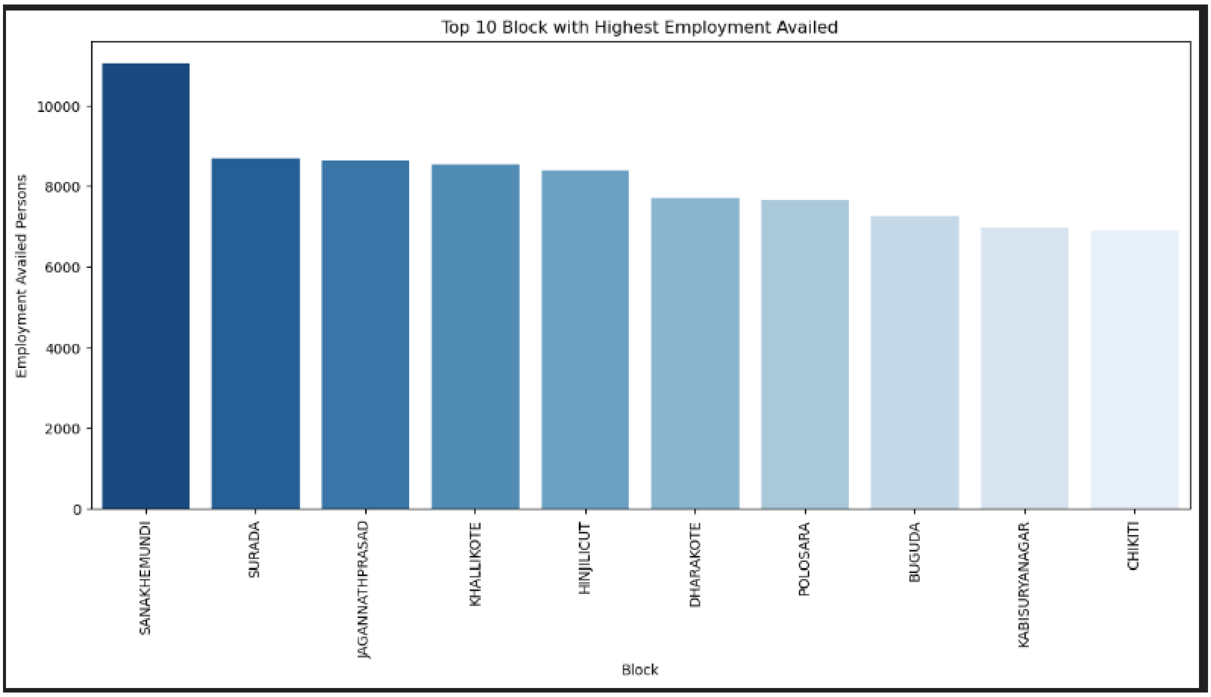
100 Days Completion Rate = (No. of Families Completed 100 days/ Employment Availed Household) * 100



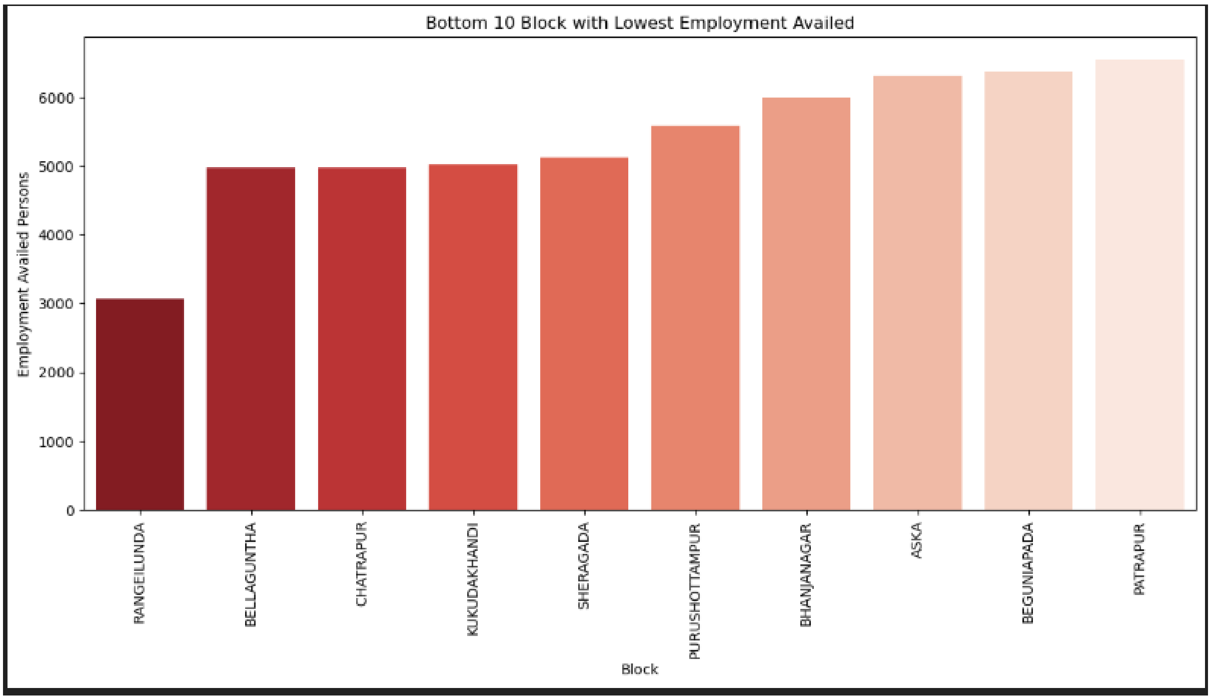
Here we can see the share of money that is allotted for the MNREGA scheme. Here Central government supports majorly with 100% of liability then the state government with no liability.



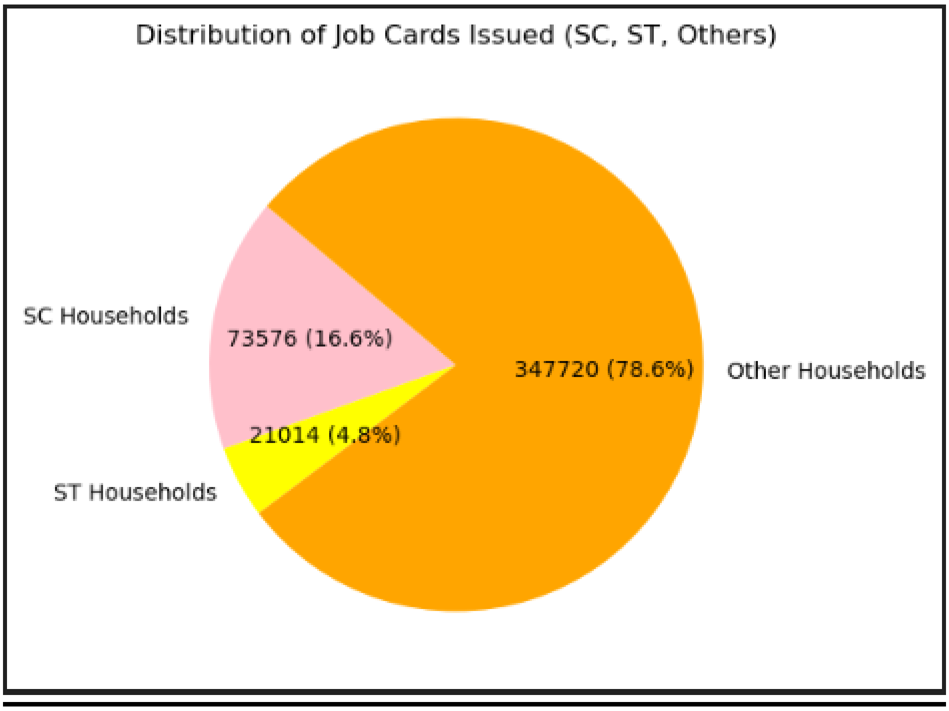
Here in this stacked bar plot green denotes the new job cards issued and red denotes the deleted job cards. PURUSHOTTAMPUR, SANAKHEMUNDI has the most number of new green card issue. Where as BHANJANAGAR, BEGUNIAPADA has most number of job card deletion. Plus most of the blocks have negligible new card issued.



Here we can see the top 10 Blocks which has highest Employment Availed. SANAKHEMUNDI is the highest employment availed ,then followed by SURADA with second highest employment availed , and then JAGANNATHPRASAD.

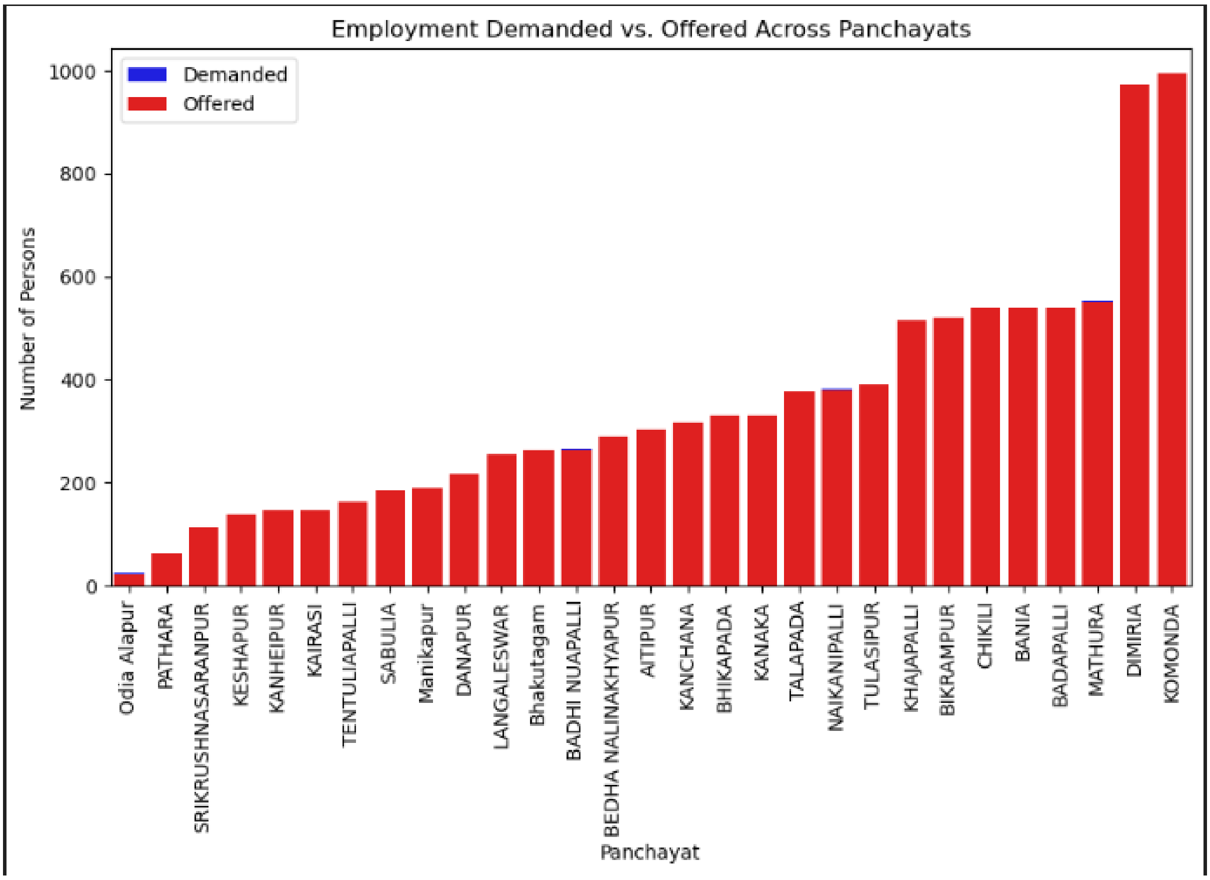


Here we can see the bottom 10 Blocks which has lowest Employment Availed. RANGEILUNDA is the lowest employment availed ,then followed by BELLAGUNTHA with second lowest employment availed , and then CHATRAPUR.



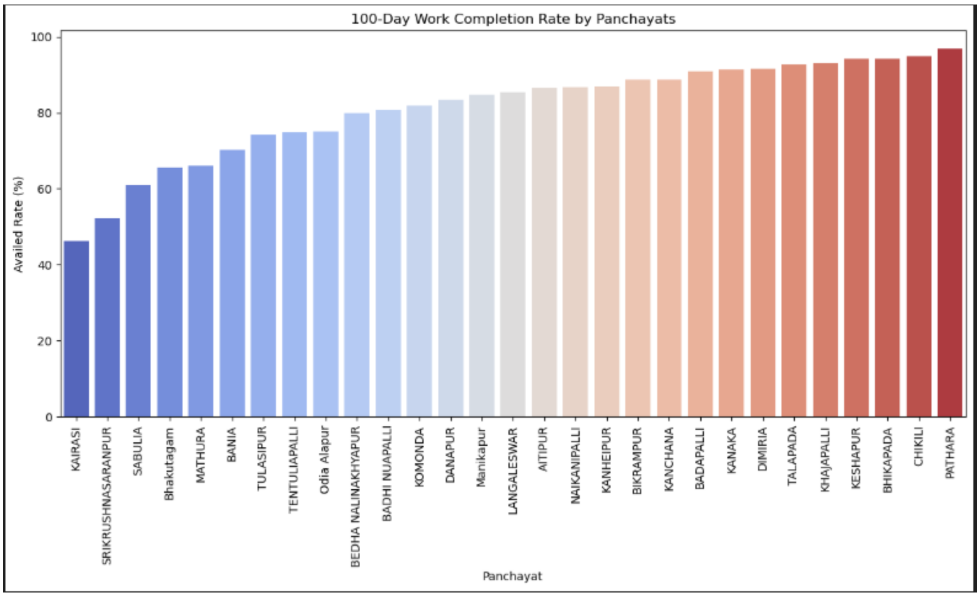
In this Pie chart we can see this job cards issued category wise. The most number of job cards has been issued for the Others category which is 78.6%, then followed by SCs category with 16.6% and then STs category with 4.8%.

Data Analysis of Employment Generation Physical Progress Report of Odisha's Ganjam District's KHALLIKOTE Block



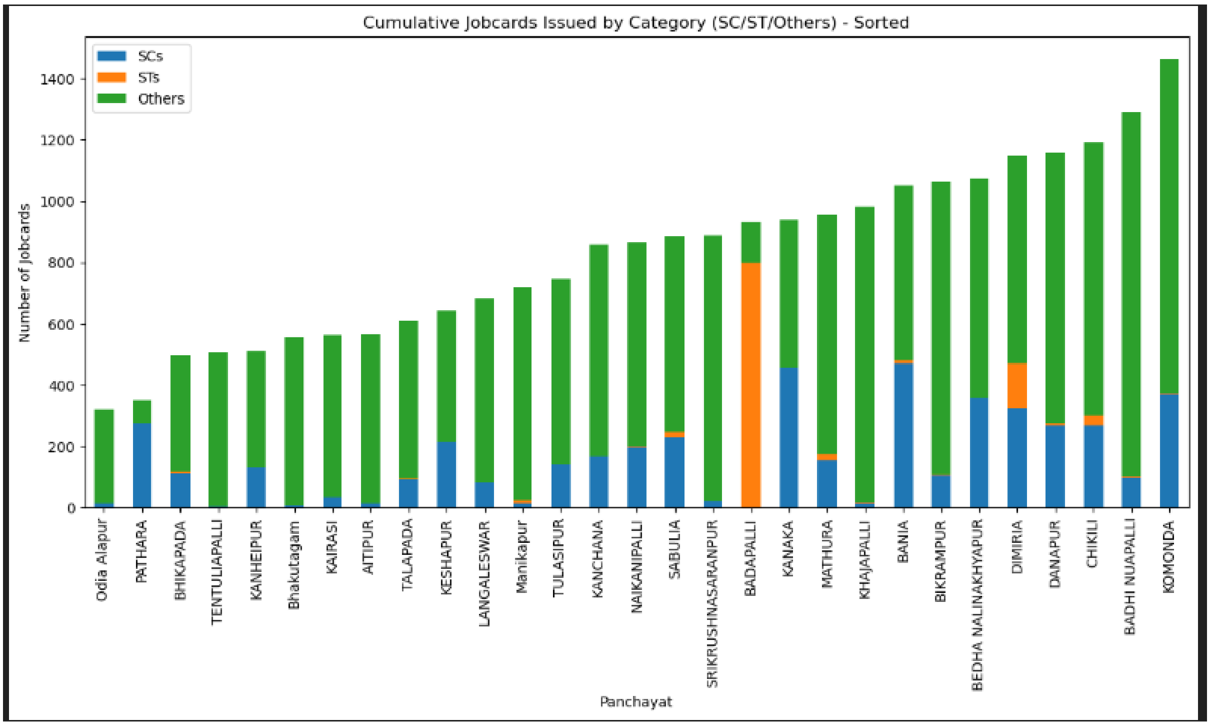
Here we can see number of jobs demanded which is highlight with blue and number of jobs offer which is highlighted in red. Here Odia Alapur has been offered the least number of jobs whereas panchayat KOMONDA has been offered the most number of jobs.

Panchayat such as NAIKANIPALLI, MATHURA, BADHI NUAPALLI here the job demanded was more than the job offered.



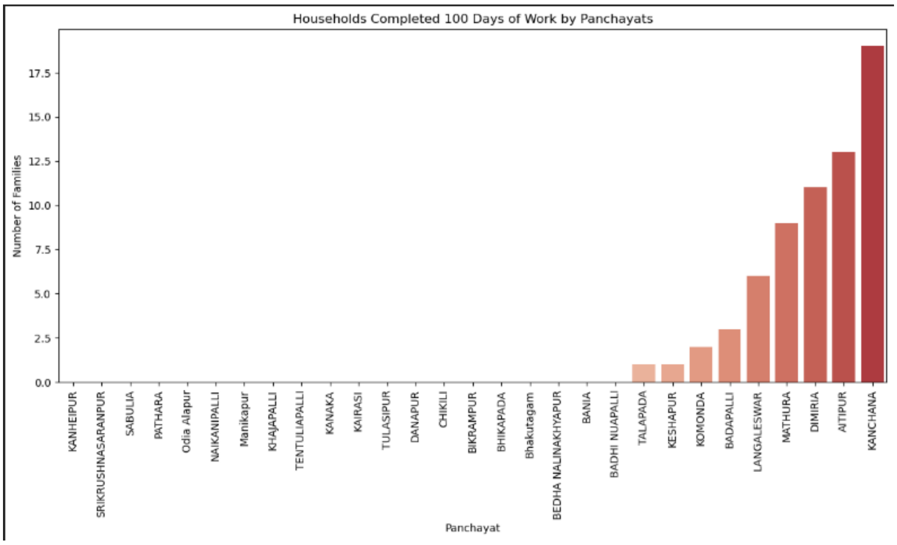
Here we can see the Employment Availed Rate by person. Panchayat PATHARA has the highest availed rate where as KAIRASI has the lowest avail rate. The formula for the calculating Availed rate is

Availed Rate = (Employment Availed Persons/ Employment demanded Persons) * 100

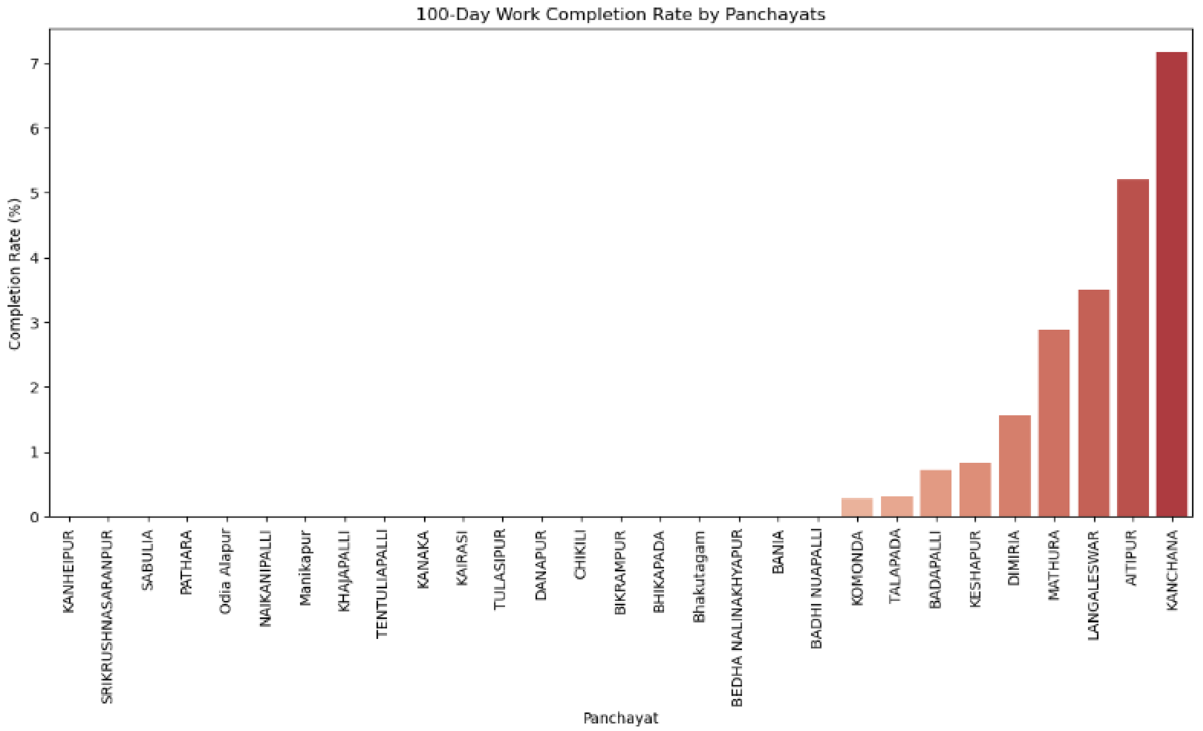


Here in this visualization we can see the Cumulative Jobcards Issued for the 3 categories. The 3 categories are SCs which is highlighted in blue , STs which is highlighted in yellow and Others which is highlighted in green.

Here we can see that KOMONDA has most number of job cards issue and in which the Others category is the highest number of job seekers. For STs category highest number of card issue in BADAPALLI panchayat. And for SCs category also highest number of job seekers are from BANIA, KANAKA panchayat.

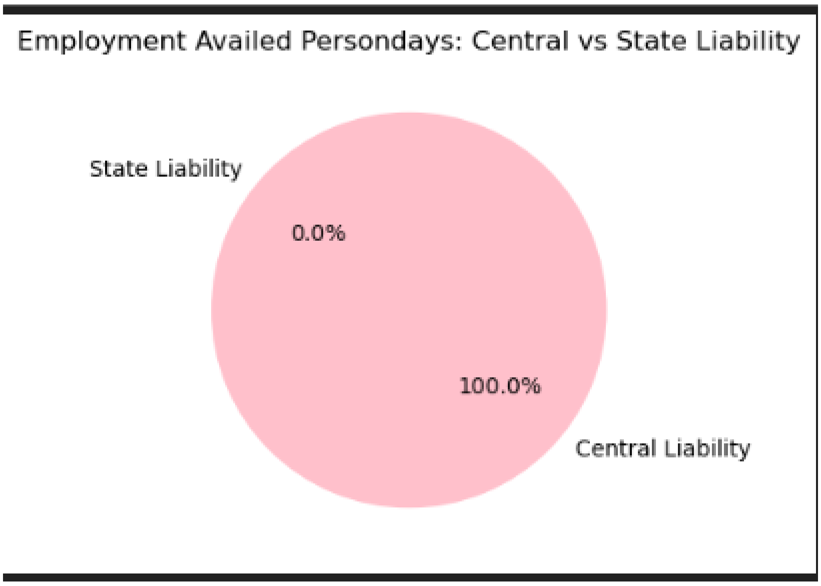


Here the number of household by panchayat that have successfully completed the 100 days of work. Panchayat KANCHANA has the highest number of household that have successfully completed the 100 days of work where as KANHEIPUR is the panchayat which has the least number households which have completed 100 days of work.

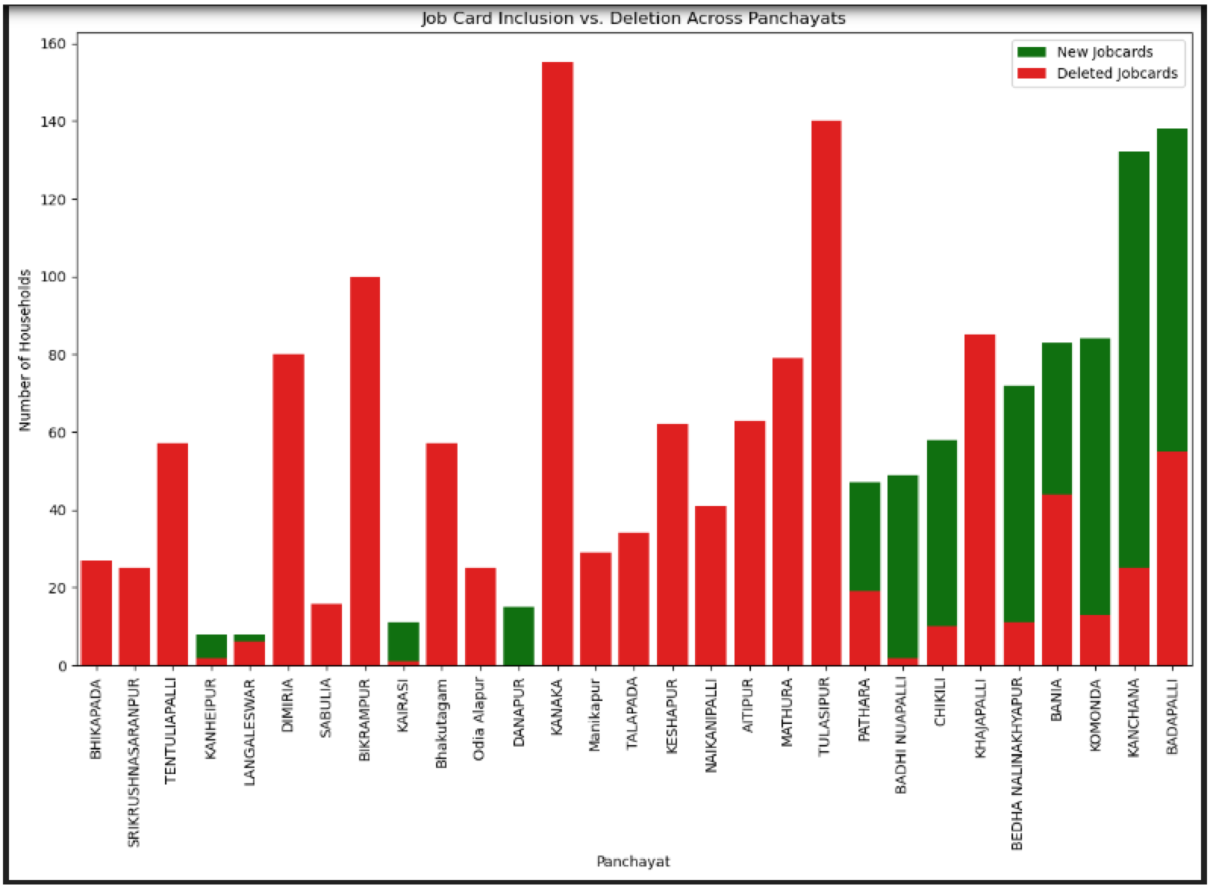


Here we can see the 100 Days Completion Rate. Panchayat KANCHANA has the highest Completion Rate where as KANHEIPUR has the lowest Complete Rate. The formula for the calculating completion rate is

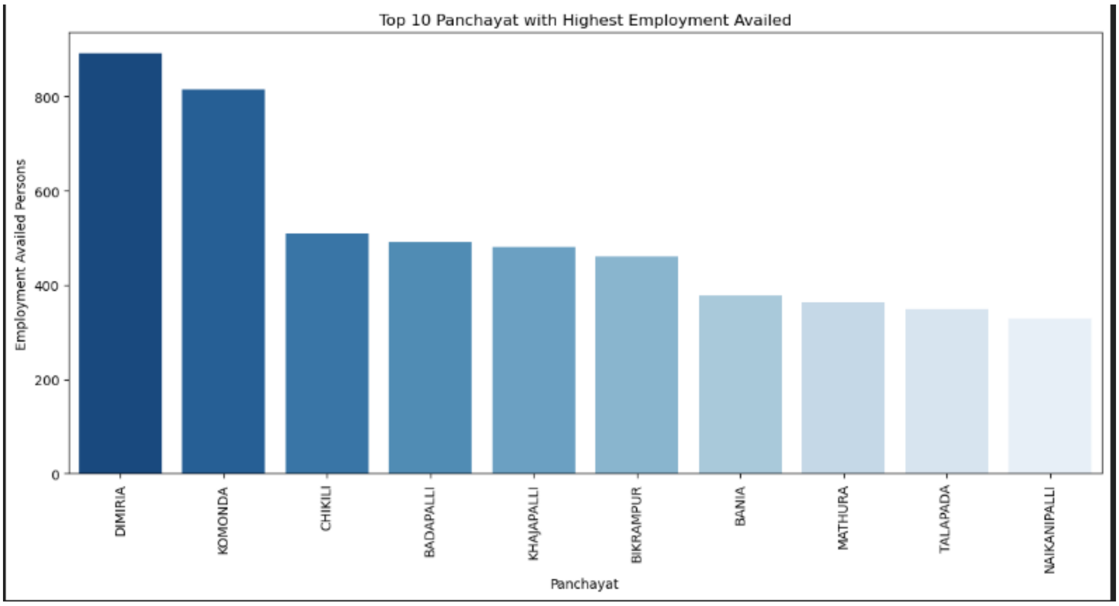
100 Days Completion Rate = (No. of Families Completed 100 days/ Employment Availed Household) * 100



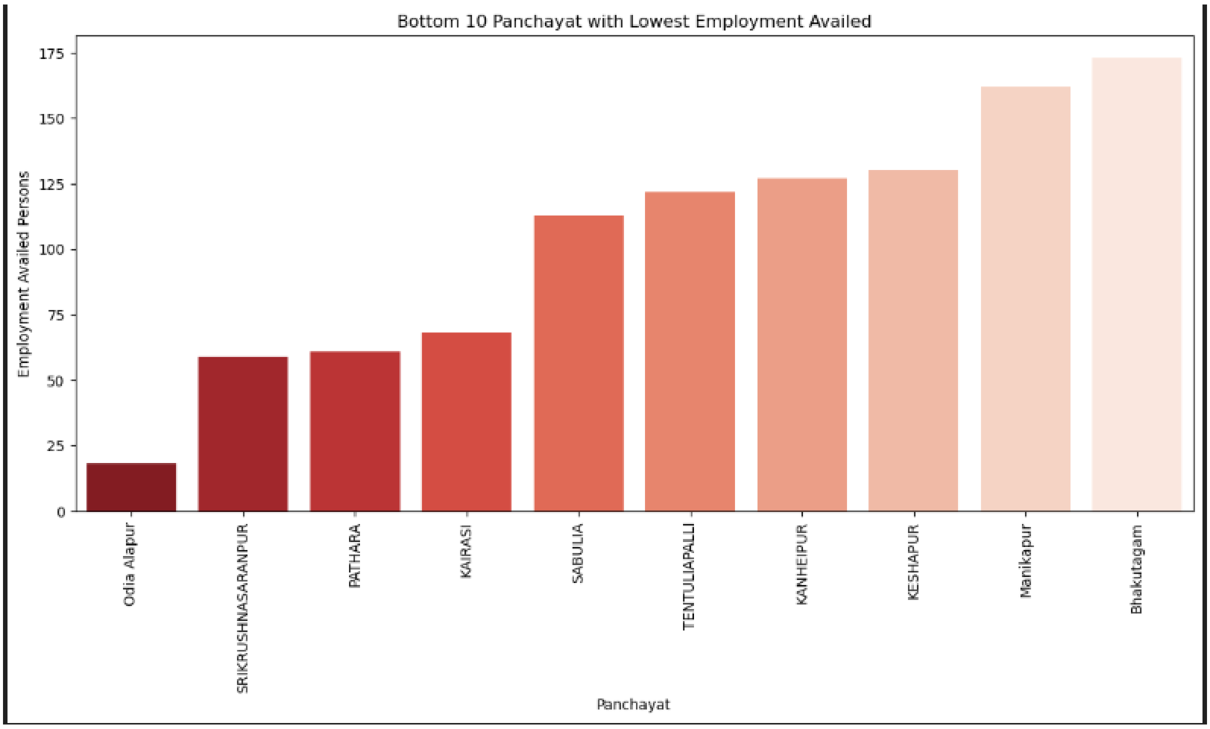
Here we can see the share of money that is allotted for the MNREGA scheme. Here Central government supports majorly with 100% of liability then the state government with no liability.



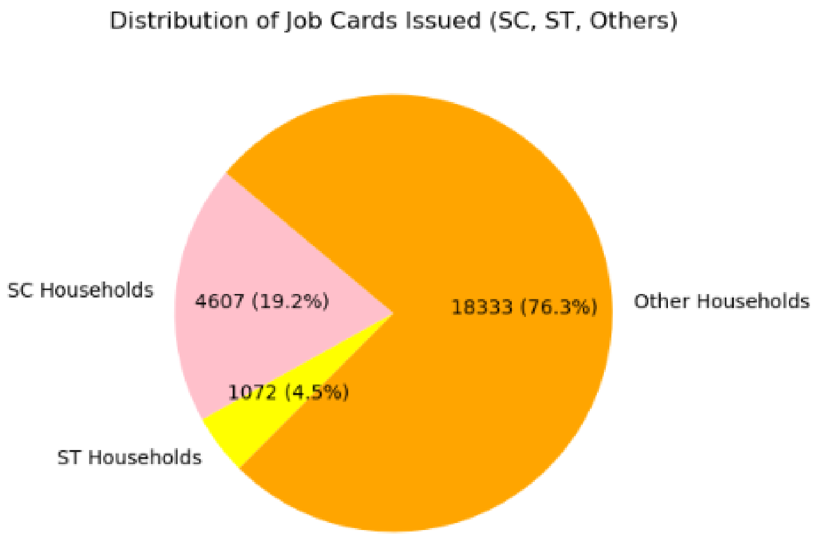
Here in this stacked bar plot green denotes the new job cards issued and red denotes the deleted job cards. KANCHANA has the most number of new green card issue. Where as KANAKA and TULASIPUR has most number of job card deletion.



Here we can see the top 10 panchayat which has highest Employment Availed. DIMIRIA is the highest employment availed ,then followed by KOMONDA with second highest employment availed , and then CHIKILI.



Here we can see the bottom 10 panchayat which has lowest Employment Availed. Odia Alapur is the lowest employment availed ,then followed by SRIKRUSHNASARANPUR with second lowest employment availed , and then PATHARA.



In this Pie chart we can see this job cards issued category wise. The most number of job cards has been issued for the Others category which is 76.3%, then followed by SCs category with 19.2% and then STs category with 4.5%.