india-budget-analysis

July 24, 2024

[1]:
```
# This Python 3 environment comes with many helpful analytics libraries␣
 ↪installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/
 ↪docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list␣
 ↪all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that␣
 ↪gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved␣
 ↪outside of the current session
```

```
/kaggle/input/india-budget-2021/India_budget_2021.csv
```

Budget of a country is a detailed report on the income and expenditure of the government for a financial year

0.0.1 Import libraries

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

0.0.2 Loading the dataset

[3]:
```
data = pd.read_csv("/kaggle/input/india-budget-2021/India_budget_2021.csv")
data.head()
```

[3]:
```
                    Department /Ministry  Fund allotted(in  crores)
0                 MINISTRY OF AGRICULTURE                 131531.19
1             DEPARTMENT OF ATOMIC ENERGY                  18264.89
2              MINISTRY OF AYURVEDA, YOGA                   2970.30
3  MINISTRY OF CHEMICALS AND FERTILISER                  80714.94
4              MINISTRY OF CIVIL AVIATION                   3224.67
```

[4]:
```
data.shape
```

[4]:
```
(57, 2)
```

[5]:
```
# finding NaN values in the dataset
data.isnull().sum()
```

[5]:
```
Department /Ministry          1
Fund allotted(in  crores)    1
dtype: int64
```

Since its a very small number we can simply these columns

[6]:
```
data=data.dropna()
```

[7]:
```
data.isnull().sum()
```

[7]:
```
Department /Ministry          0
Fund allotted(in  crores)    0
dtype: int64
```

Now we have no missing values Further in our dataset there are a number of department which are not mainstream department, it can be vogus to show them all in a single plot. we can simply consider to plot and visualize main departments and put other depatt in 'others' category

[8]:
```
data = data.iloc[[0,8,11,14,18,23,41,42,43],:]
row = {'Department /Ministry': 'OTHERS', 'Fund allotted(in  crores)': 592971.
 ↪0800000001}
data = data.append(row, ignore_index = True)
print(data)
```

```
                    Department /Ministry  Fund allotted(in  crores)
0                MINISTRY OF AGRICULTURE                  131531.19
1           MINISTRY OF CONSUMER AFFAIRS                  256948.40
2                    MINISTRY OF DEFENCE                  478195.62
3                  MINISTRY OF EDUCATION                   93224.31
```

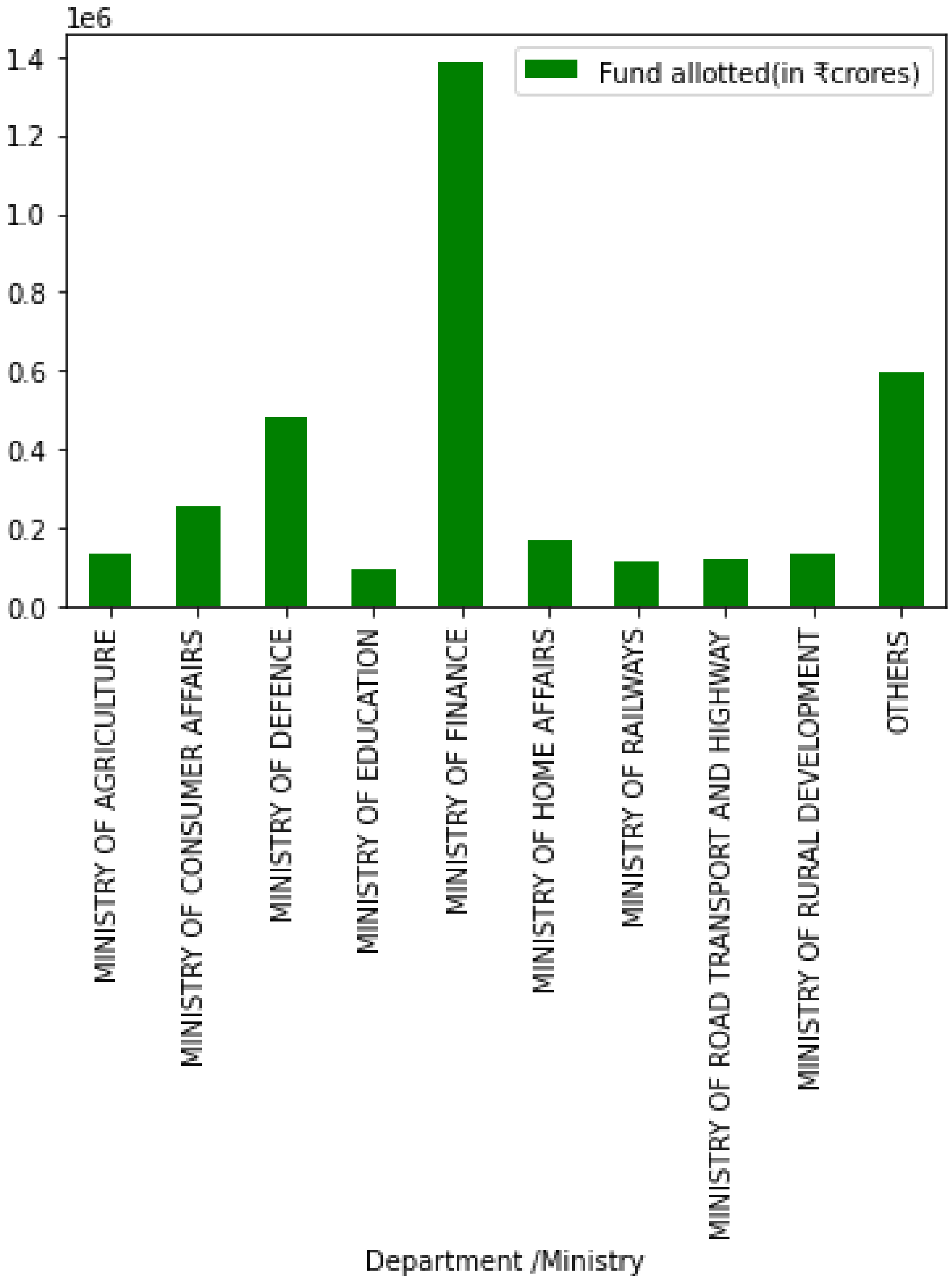
```
4                    MINISTRY OF FINANCE                  1386273.30
5                MINISTRY OF HOME AFFAIRS                  166546.94
6                    MINISTRY OF RAILWAYS                  110054.64
7  MINISTRY OF ROAD TRANSPORT AND HIGHWAY                  118101.00
8            MINISTRY OF RURAL DEVELOPMENT                  133689.50
9                                  OTHERS                  592971.08
```

[9]:

```
data.plot.bar(x='Department /Ministry', y='Fund allotted(in␣
 ↪crores)',color='green')
```

[9]:

```
<AxesSubplot:xlabel='Department /Ministry'>
```

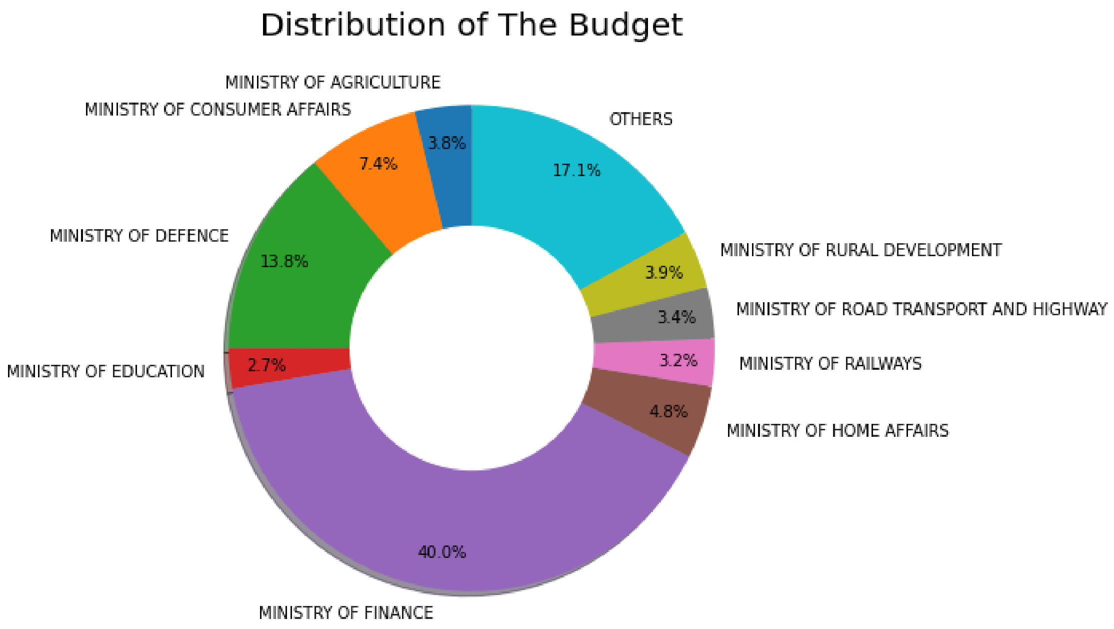


finance department is has the most value of budget from the total budget of the government

[10]:
```
# creating a donut plot for clear distribution of budgets in various department
df = data["Fund allotted(in  crores)"]
labels = data["Department /Ministry"]
```

[11]:
```
plt.figure(figsize=(7,7))
plt.pie(df, labels=labels, autopct='%1.1f%%', startangle=90, pctdistance=0.85,␣
  ↪shadow =True)
```

```
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Distribution of The Budget", fontsize=20)
plt.show()
```



```
[ ]:
```