Expenditure Data Analysis Project

Table of Contents:

1. Problem statements:
2. Data Description:
    - 2.1 Introduction
    - 2.2 Data source and data set
3. Load the Packages and Data
4. Data Profiling:
    - 4.1 Understanding the Dataset
    - 4.2 Pre Profiling
    - 4.3 Preprocessing
    - 4.4 Post Profiling
5. Data Visualization:
6. Conclusions:

1. Problem statements:

No business can survive in this competitive market without managing their cost. It does not matter if revenues are high but if cost is higher it is a red flag. So you are tasked to help management in creating and establishing new structure and models to reduce cost.

2. Data Description:

- **Exp Category**:Gives the description about expenditure Category.
- **State**: Gives the description about States and UTs of India.
- **Year**: Gives the description about Year.
- **Value**: Gives the description about expenditure spending in millions.

The dataset as listed on NITI Aayog website from 1980_81 to 2015_16.That is colllected by using web scraping.

2.1. Introduction:

An Expenditure Data Analysisis the project related to Exploratory data analysis(EDA)and Data visualization of expenditure dataset from **NITI Aayog** of India I have used python libraries to get expenditure information, visualize different aspects of it, and finally I worked at a few ways of analyzing the spending of expenditure based on its previous performance history statewise in India. The **NITI Aayog**(National Institution for Transforming India) serves as the apex public policy think tank of the Government of India, and the nodal agency tasked with catalyzing economic development, and fostering cooperative federalism through the involvement of State Governments of India in the economic policy-making process using a bottom-up approach

2.2. Data source and data set:

The dataset as listed on NITI Aayog website from 1980_81 to 2015_16.That is colllected by using web scraping.

You can find the dataset on the given link. https://www.niti.gov.in/ (https://www.niti.gov.in/)

Approach

The main goal of the project is to find key metrics and factors and show the meaningful relationships between attributes based on different features available in the dataset.

- **Do ETL** : Extract-Transform-Load the dataset and find for some information from this large data. This is form of data mining.

3. Load the Packages and Data

1. Import libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
sns.set()
```

2. Loading data

In [2]:

```
expenditure = pd.read_csv("Final_expenditure.csv")
expenditure.head()
```

Out[2]:

	Exp Category	State	Year	Value
0	Aggregate_Expenditure	Andhra Pradesh	1980-81	1610.0
1	Aggregate_Expenditure	Andhra Pradesh	1981-82	1831.0
2	Aggregate_Expenditure	Andhra Pradesh	1982-83	1933.0
3	Aggregate_Expenditure	Andhra Pradesh	1983-84	2588.0
4	Aggregate_Expenditure	Andhra Pradesh	1984-85	3119.0

4. Data Profiling:

4.1. Understanding the Dataset

In [3]:

```
expenditure.shape # To know shape of dataset
```

Out[3]:

```
(8760, 4)
```

- Their are 8760 rows and 4 column in dataset after combining.

In [4]:

```
expenditure.size       # to show the total no. of volume(elements)
```

Out[4]:

```
35040
```

In [5]:

```
expenditure.columns # to show eachh columns name in dataset
```

Out[5]:

```
Index(['Exp Category', 'State', 'Year', 'Value'], dtype='object')
```

In [6]:

```
expenditure.dtypes  # to shows data types of each column name
```

Out[6]:

```
Exp Category     object
State            object
Year             object
Value           float64
dtype: object
```

In [7]:

```
expenditure.describe() # To show Statistic information of dataset
```

Out[7]:

	Value
count	8.529000e+03
mean	1.855860e+04
std	7.518330e+04
min	0.000000e+00
25%	2.630000e+02
50%	1.780000e+03
75%	8.884390e+03
max	1.792122e+06

In [8]:

```
expenditure.describe(include='all') # To show Statistic information of all dataset
```

Out[8]:

	Exp Category	State	Year	Value
count	8760	8760	6521	8.529000e+03
unique	8	31	36	NaN
top	Aggregate_Expenditure	Andhra Pradesh	2015-16	NaN
freq	1116	289	186	NaN
mean	NaN	NaN	NaN	1.855860e+04
std	NaN	NaN	NaN	7.518330e+04
min	NaN	NaN	NaN	0.000000e+00
25%	NaN	NaN	NaN	2.630000e+02
50%	NaN	NaN	NaN	1.780000e+03
75%	NaN	NaN	NaN	8.884390e+03
max	NaN	NaN	NaN	1.792122e+06

In [9]:

```
expenditure.info() # to shows indexes,data types each columns name
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8760 entries, 0 to 8759
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Exp Category  8760 non-null   object 
 1   State         8760 non-null   object 
 2   Year          6521 non-null   object 
 3   Value         8529 non-null   float64
dtypes: float64(1), object(3)
memory usage: 273.9+ KB
```

In [10]:

```
#Finding how many unique values are in the dataset
expenditure.nunique()
```

Out[10]:

```
Exp Category       8
State             31
Year              36
Value           5695
dtype: int64
```

In [11]:

```
expenditure['Year'].unique() #  unique values in Year columns
```

Out[11]:

```
array(['1980-81', '1981-82', '1982-83', '1983-84', '1984-85', '1985-86',
       '1986-87', '1987-88', '1988-89', '1989-90', '1990-91', '1991-92',
       '1992-93', '1993-94', '1994-95', '1995-96', '1996-97', '1997-98',
       '1998-99', '1999-00', '2000-01', '2001-02', '2002-03', '2003-04',
       '2004-05', '2005-06', '2006-07', '2007-08', '2008-09', '2009-10',
       '2010-11', '2011-12', '2012-13', '2013-14', '2014-15', '2015-16',
       nan], dtype=object)
```

- This Dataset contains from year 1980-81 to 2015-16.

In [12]:

```
expenditure['Exp Category'].unique() #  categories of expenditure
```

Out[12]:

```
array(['Aggregate_Expenditure', 'Capital_Expenditure',
       'Gross_Fiscal_Deficits', 'Nominal_GSDP_Series', 'Own_Tax_Revenues',
       'Revenue_Deficits', 'Revenue_Expenditure',
       'Social_Sector_Expenditure'], dtype=object)
```

In [13]:

```
expenditure['State'].unique() #  unique values in state columns
```

Out[13]:

```
array(['Andhra Pradesh ', 'Arunachal Pradesh', 'Assam', 'Bihar',
       'Chhattisgarh', 'Goa', 'Gujarat', 'Haryana', 'Himachal Pradesh',
       'Jammu & Kashmir', 'Jharkhand', 'Karnataka', 'Kerala',
       'Madhya Pradesh', 'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram',
       'Nagaland', 'Odisha', 'Punjab', 'Rajasthan', 'Sikkim',
       'Tamil Nadu', 'Telangana', 'Tripura', 'Uttar Pradesh',
       'Uttarakhand', 'West Bengal', 'Delhi', 'Puducherry'], dtype=object)
```

- These are the names of States and UTs of India.

4.2 Preprofilling:

By pandas profiling, an interactive **HTML report** gets generated which contains all the information about the columns of the dataset, like the counts and type of each column.

1. Detailed information about each column, coorelation between different columns and a sample of dataset
2. It gives us visual interpretation of each column in the data
3. Spread of the data can be better understood by the distribution plot
4. Grannular level analysis of each column.

Now performing pandas profiling to understand data better.

In [14]:

```
import pandas_profiling as prf
```

To generate the standard profiling report, merely run:

In [15]:

```
expenditure_profile = prf.ProfileReport(expenditure)
expenditure_profile
```

Summarize dataset: 100% 15/15 [00:09<00:00, 1.54it/s, Completed]

Generate report structure: 100% 1/1 [00:03<00:00, 3.76s/it]

Render HTML: 100% 1/1 [00:05<00:00, 5.96s/it]

Overview

Dataset statistics

Number of variables	4
Number of observations	8760
Missing cells	2470
Missing cells (%)	7.0%
Duplicate rows	35
Duplicate rows (%)	0.4%
Total size in memory	273.9 KiB
Average record size in memory	32.0 B

Variable types

Categorical	3
Numeric	1

Alerts

Dataset has 35 (0.4%) duplicate rows	Duplicates
`Year` has 2239 (25.6%) missing values	Missing
`Value` has 231 (2.6%) missing values	Missing
`Year` is uniformly distributed	Uniform
`Value` has 768 (8.8%) zeros	Zeros

Reproduction

Out[15]:

In [16]:

```
# save profile
expenditure_profile.to_file(output_file="expenditure_before_preprocessing.html")
```

Export report to file: 100% 1/1 [00:00<00:00, 7.14it/s]

4.3 Preprocessing

Modified the structure of data in order to make it more understandable and suitable and convenient for statistical analysis.

1. Checking null values
2. Feeling null values
3. Checking and removing Duplicates rows

1. Checking null values

In [17]:

```
m = expenditure.isnull().sum()
m
```

Out[17]:

```
Exp Category       0
State              0
Year            2239
Value            231
dtype: int64
```

In [18]:

```
#missing values in percentage
m1 = m/len(expenditure)*100
m1
```

Out[18]:

```
Exp Category     0.000000
State            0.000000
Year            25.559361
Value            2.636986
dtype: float64
```

In [19]:

```
#missing values with %
pd.concat([m,m1],axis=1,keys = ['Total','Missing %'])
```

Out[19]:

	Total	Missing %
Exp Category	0	0.000000
State	0	0.000000
Year	2239	25.559361
Value	231	2.636986

- Year having 25% and Value having 2 % missing values contains in the dataset.

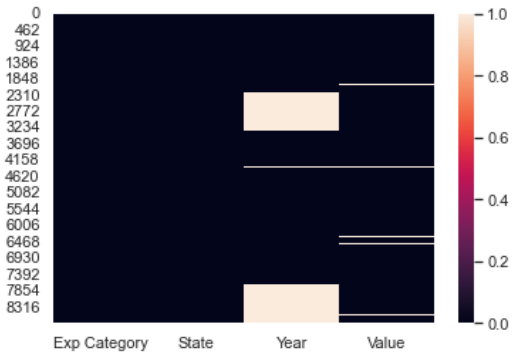
Null values shown by heatmap

In [20]:

```
sns.heatmap(expenditure.isnull())
```

Out[20]:

```
<AxesSubplot:>
```



2.Feeling nulll values

- feeling null values with 0 .

In [21]:

```
# make copy of dataset before changes
exp_data = expenditure.copy()
exp_data.head()
```

Out[21]:

	Exp Category	State	Year	Value
0	Aggregate_Expenditure	Andhra Pradesh	1980-81	1610.0
1	Aggregate_Expenditure	Andhra Pradesh	1981-82	1831.0
2	Aggregate_Expenditure	Andhra Pradesh	1982-83	1933.0
3	Aggregate_Expenditure	Andhra Pradesh	1983-84	2588.0
4	Aggregate_Expenditure	Andhra Pradesh	1984-85	3119.0

In [22]:

```
exp_data.fillna(0,inplace=True)
```

In [23]:

```
#checking missing values again
exp_data.isnull().sum()
```

Out[23]:

```
Exp Category    0
State           0
Year            0
Value           0
dtype: int64
```

3. Checking and removing Duplicates rows

In [24]:

```
exp_data[exp_data.duplicated()] #duplicates rows
```

Out[24]:

	Exp Category	State	Year	Value
2266	Gross_Fiscal_Deficits	Arunachal Pradesh	0	0.0
2267	Gross_Fiscal_Deficits	Arunachal Pradesh	0	0.0
2268	Gross_Fiscal_Deficits	Arunachal Pradesh	0	0.0
2269	Gross_Fiscal_Deficits	Arunachal Pradesh	0	0.0
2270	Gross_Fiscal_Deficits	Arunachal Pradesh	0	0.0
...	...	...	...	...
8744	Social_Sector_Expenditure	Puducherry	0	0.0
8745	Social_Sector_Expenditure	Puducherry	0	0.0
8746	Social_Sector_Expenditure	Puducherry	0	0.0
8747	Social_Sector_Expenditure	Puducherry	0	0.0
8748	Social_Sector_Expenditure	Puducherry	0	0.0

275 rows × 4 columns

In [25]:

```
expenditure.duplicated().sum() #number of duplicates rows
```

Out[25]:

```
273
```

- 273 rows are duplicates.
- so lets drop them for better analysis.

In [26]:

```
exp_data.drop_duplicates(inplace=True)
```

In [27]:

```
#again checking for duplicates
exp_data.duplicated().sum()
```

Out[27]:

```
0
```

In [28]:

```
#checking size after cleaning
exp_data.shape
```

Out[28]:

```
(8485, 4)
```

- All 273 duplicated rows are removed.

4.4 Post Profiling

- Post profilling after cleaning dataset.

In [29]:

```
exp_clean_profile = prf.ProfileReport(exp_data)
exp_clean_profile
```

Summarize dataset: 100% 14/14 [00:02<00:00, 3.49it/s, Completed]

Generate report structure: 100% 1/1 [00:03<00:00, 3.40s/it]

Render HTML: 100% 1/1 [00:00<00:00, 1.08it/s]

Overview

Dataset statistics

Number of variables	4
Number of observations	8485
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	109
Duplicate rows (%)	1.3%
Total size in memory	589.5 KiB
Average record size in memory	71.1 B

Variable types

Categorical	2
Unsupported	1
Numeric	1

Alerts

Dataset has 109 (1.3%) duplicate rows	Duplicates
`Year` is an unsupported type, check if it needs cleaning or further analysis	Unsupported
`Value` has 741 (8.7%) zeros	Zeros

Reproduction

Analysis started	2023-04-29 13:04:00.513773

Out[29]:

In [30]:

```
#save clean profile file

exp_clean_profile.to_file(output_file="expenditure_after_preprocessing.html")
```

Export report to file: 100% 1/1 [00:00<00:00, 5.29it/s]

In [31]:

```
#save clean dataset into csv
exp_data.to_csv('expenditure1.csv')
```

5. Data Visualization:

Data visualization is concerned with visually presenting sets of primarily quantitative raw data in a schematic form. The visual formats used in data visualization include tables, charts and graphs.

- In this project we use matplotlib and seaborn python libraries.

1. Correlation between features

In [32]:

```
corr = exp_data.corr()
corr
```

Out[32]:

	Value
Value	1.0

- Only one feature for correlation.

2. All unique categories of expenditure.

In [33]:

```
exp_data.head(2)
```

Out[33]:

	Exp Category	State	Year	Value
0	Aggregate_Expenditure	Andhra Pradesh	1980-81	1610.0
1	Aggregate_Expenditure	Andhra Pradesh	1981-82	1831.0

In [34]:

```
exp_data['Exp Category'].nunique()
```

Out[34]:

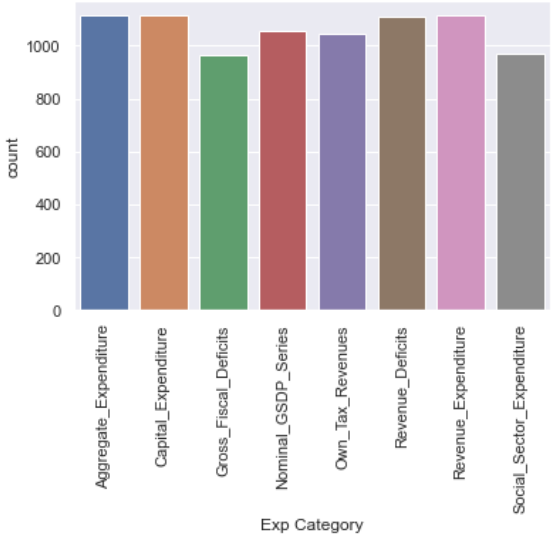
8

In [35]:

```
sns.countplot(exp_data['Exp Category'],orient='v')
#sns.set_theme(style="darkgrid")
plt.xticks(rotation=90)
```

Out[35]:

```
(array([0, 1, 2, 3, 4, 5, 6, 7]),
 [Text(0, 0, 'Aggregate_Expenditure'),
  Text(1, 0, 'Capital_Expenditure'),
  Text(2, 0, 'Gross_Fiscal_Deficits'),
  Text(3, 0, 'Nominal_GSDP_Series'),
  Text(4, 0, 'Own_Tax_Revenues'),
  Text(5, 0, 'Revenue_Deficits'),
  Text(6, 0, 'Revenue_Expenditure'),
  Text(7, 0, 'Social_Sector_Expenditure')])
```



Insights: Its clearly shown there are 8 expenditure categories in this NITI Aayog dataset.

3. Names of all states in india.

In [36]:

```
exp_data['State'].nunique()
```

Out[36]:

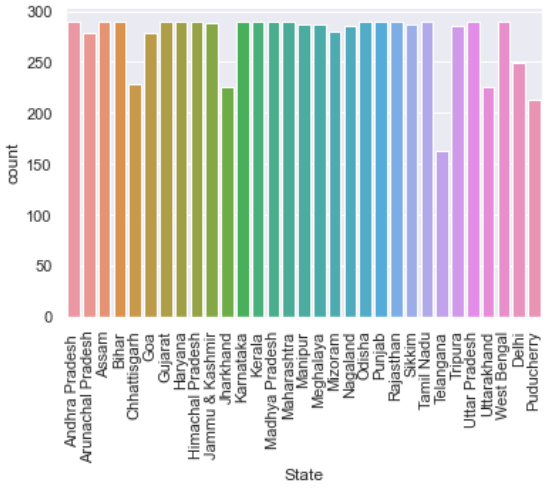
```
31
```

In [37]:

```
#shows in countplot
sns.countplot(exp_data['State'])
sns.set_theme(style="darkgrid")
plt.xticks(rotation=90)
```

Out[37]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]),
 [Text(0, 0, 'Andhra Pradesh '),
  Text(1, 0, 'Arunachal Pradesh'),
  Text(2, 0, 'Assam'),
  Text(3, 0, 'Bihar'),
  Text(4, 0, 'Chhattisgarh'),
  Text(5, 0, 'Goa'),
  Text(6, 0, 'Gujarat'),
  Text(7, 0, 'Haryana'),
  Text(8, 0, 'Himachal Pradesh'),
  Text(9, 0, 'Jammu & Kashmir'),
  Text(10, 0, 'Jharkhand'),
  Text(11, 0, 'Karnataka'),
  Text(12, 0, 'Kerala'),
  Text(13, 0, 'Madhya Pradesh'),
  Text(14, 0, 'Maharashtra'),
  Text(15, 0, 'Manipur'),
  Text(16, 0, 'Meghalaya'),
  Text(17, 0, 'Mizoram'),
  Text(18, 0, 'Nagaland'),
  Text(19, 0, 'Odisha'),
  Text(20, 0, 'Punjab'),
  Text(21, 0, 'Rajasthan'),
  Text(22, 0, 'Sikkim'),
  Text(23, 0, 'Tamil Nadu'),
  Text(24, 0, 'Telangana'),
  Text(25, 0, 'Tripura'),
  Text(26, 0, 'Uttar Pradesh'),
  Text(27, 0, 'Uttarakhand'),
  Text(28, 0, 'West Bengal'),
  Text(29, 0, 'Delhi'),
  Text(30, 0, 'Puducherry')])
```



Insights: Its clearly shown there are 31 counts of states and Union territories in India.

4. Which is the Highest invested category of expenditure on which state?

In [38]:

```
exp_data['Exp Category'].describe(include=all)
```

Out[38]:

```
count                     8485
unique                       8
top       Aggregate_Expenditure
freq                      1116
Name: Exp Category, dtype: object
```

In [39]:

```
exp_data.groupby("Exp Category")["State"].agg(pd.Series.mode)
```

Out[39]:

```
Exp Category
Aggregate_Expenditure        [Andhra Pradesh , Arunachal Pradesh, Assam, Bi...
Capital_Expenditure          [Andhra Pradesh , Arunachal Pradesh, Assam, Bi...
Gross_Fiscal_Deficits        [Andhra Pradesh , Assam, Bihar, Gujarat, Harya...
Nominal_GSDP_Series          [Andhra Pradesh , Arunachal Pradesh, Assam, Bi...
Own_Tax_Revenues             [Andhra Pradesh , Arunachal Pradesh, Assam, Bi...
Revenue_Deficits             [Andhra Pradesh , Arunachal Pradesh, Assam, Bi...
Revenue_Expenditure          [Andhra Pradesh , Arunachal Pradesh, Assam, Bi...
Social_Sector_Expenditure    [Andhra Pradesh , Assam, Bihar, Gujarat, Harya...
Name: State, dtype: object
```

- Aggregate_Expenditure is Highest invested category of expenditure on Andhra Pradesh.

Insights: Aggregate_Expenditure is Highest invested category of expenditure on Andhra Pradesh .

5. Top 5 state having aggregate expenditure spending?

In [40]:

```
exp_data.groupby(['Exp Category', 'State']).count()["Value"]
```

Out[40]:

```
Exp Category               State
Aggregate_Expenditure      Andhra Pradesh       36
                           Arunachal Pradesh    36
                           Assam                36
                           Bihar                36
                           Chhattisgarh         36
                                                ..
Social_Sector_Expenditure  Telangana             3
                           Tripura              35
                           Uttar Pradesh        36
                           Uttarakhand          17
                           West Bengal          36
Name: Value, Length: 248, dtype: int64
```

Insights: The Aggregate expenditure spending on these top 5 states are Andhra Pradesh, Arunachal Pradesh ,Assam ,Bihar & Chhattisgarh .

6. Expenditure spending over the years

In [41]:

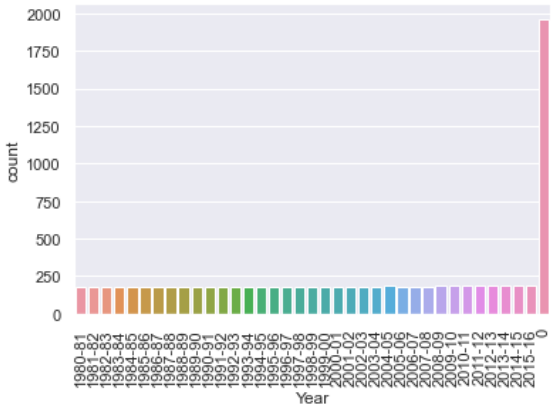
```
exp_data.Year.value_counts().to_frame('Value')
```

Out[41]:

	Value
0	1964
2015-16	186
2014-15	186
2013-14	186
2010-11	184
2004-05	184
2009-10	184
2008-09	184
2011-12	184
2012-13	184
2003-04	183
2002-03	183
2001-02	183
2007-08	182
2006-07	182
2005-06	182
2000-01	182
1999-00	180
1980-81	180
1981-82	180
1997-98	180
1982-83	180
1983-84	180
1984-85	180
1985-86	180
1986-87	180
1987-88	180
1988-89	180
1989-90	180
1993-94	180
1994-95	180
1995-96	180
1996-97	180
1998-99	180
1990-91	174
1991-92	174
1992-93	174

In [42]:

```
sns.countplot(x=exp_data['Year'],orient='v')
plt.xticks(rotation=90)
sns.set(rc={'figure.figsize':(30,30)})
```



- Anual progress of expenditure.

6. Conclusion:

In this way, I collect expenditure dataset fron **Niti Aayog** website.Load,clean and perform data analysis by using Exploratory data analysis in Python.I using python libraries such as pandas ,numpy,matplotlib,seaborn and pandas_profiling.For visualization using heatmap, counplot and graphs. In this EDA We extracted clean dataset as expenditure1 in csv for using for Data visualization.

In [ ]:

```

```