Expenditure Data Analysis Project

Index Table:

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.
- Values : Gives the description about expenditure spending in millions.

The Dataset as listed on NITI Aayog Website from 1980_81 to 2015_16. That is collected by using web scraping.

##2.1. Introduction:

An Expenditure Data Analysis the project releted to Exploratory data analysis(EDA) and Data Visualization of 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 Goverment of India, and the nodal agency tasked with catalyzing economic development, and fostering cooperative federalism through the involvement of State Goverments of India in the economic policy-making process using a bottom-up apporach

2.2 Data source and data Set:

The dataset as listed on NITI Aayog website from 1980_81 to 2015_16. That is collected 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 from of data mining.

3. Load the Package and Data

1. Import Libraries

In [3]:

```
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 [8]:

```
expenditure = pd.read_excel("Downloads/final_expenditure.xlsx")
```

In [9]:

```
expenditure.head()
```

Out[9]:

	Exp Category	State	Year	value
0	Aggregate	Andhra Pradesh	1980-81	1610
1	Aggregate	Andhra Pradesh	1981-82	1611
2	Aggregate	Andhra Pradesh	1982-83	1612
3	Aggregate	Andhra Pradesh	1983-84	1613
4	Aggregate	Andhra Pradesh	1984-85	1614

4.Data Profiling:

4.1. Understanding the Dataset

```
In [10]: expenditure.shape # To know shape of dataset
```

```
Out[10]: (8753, 4)
```

- Their are 8753 rows and 4 columns in dataset after combining.

```
In [11]: expenditure.size  # to show the total no. of volume(elements)
```

```
Out[11]: 35012
```

```
In [12]: expenditure.columns # to show each columns name in dataset
```

```
Out[12]: Index(['Exp Category', 'State', 'Year', 'value'], dtype='object')
```

```
In [13]: expenditure.dtypes  # to shows data types of each column name
```

```
Out[13]: Exp Category    object
         State           object
         Year            object
         value           object
         dtype: object
```

```
In [14]: expenditure.describe() # To show Statistic information of dataset
```

Out[14]:

	Exp Category	State	Year	value
count	8753	8753	8736	8536
unique	10	32	47	6258
top	Aggregate	Andhra Pradesh	2013-14	–
freq	1116	289	248	741

```
In [15]: expenditure.describe(include = 'all') # To show Statistics information of all
```

Out[15]:

	Exp Category	State	Year	value
count	8753	8753	8736	8536
unique	10	32	47	6258
top	Aggregate	Andhra Pradesh	2013-14	–
freq	1116	289	248	741

In [16]:
```
expenditure.info()  # to show indexes , data types each columns name
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8753 entries, 0 to 8752
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Exp Category  8753 non-null   object
 1   State         8753 non-null   object
 2   Year          8736 non-null   object
 3   value         8536 non-null   object
dtypes: object(4)
memory usage: 273.7+ KB
```

In [17]:
```
#Finfing how many unique values are in the dataset
expenditure.nunique()
```

Out[17]:
```
Exp Category      10
State             32
Year              47
value           6258
dtype: int64
```

In [18]:
```
expenditure['Year'].unique() # unique values in year columns
```

Out[18]:
```
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 (RE)',
       '2015-16 (BE)', 'Year', '1980-82', '1980-83', '1980-84', '1980-85',
       '1980-86', 'Attribute', '2014-15', '2015-16', nan, '2015-16 (RE)',
       'year'], dtype=object)
```

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

In [19]:
```
expenditure['Exp Category'].unique() # categories of expenditure
```

Out[19]:
```
array(['Aggregate', 'Exp Category', 'Capital', 'Exp Category\t',
       'Gross_Fiscal_Deficits', 'Nominal_GSDP', 'Own_Tax_Revenues',
       'Revenue_Deficits', 'Revenue', 'Social_Sector_Expenditure'],
      dtype=object)
```

```
In [20]: expenditure['State'].unique() # unique Values in State columns
```

```
Out[20]: 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', 'State'],
               dtype=object)
```

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

4.2 Preprofiling:

By pandas profiling, an interctive 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 [21]: import pandas_profiling as prf
```

To generate the standard profiling report,merely run:

```
In [22]: expenditure_profile = prf.ProfileReport(expenditure)
         expenditure_profile
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

```
Out[22]:
```

```
In [23]: # save profile
         expenditure_profile.to_file(output_file="expenditure_before_preprocessing.htm
```

```
Export report to file:   0%|          | 0/1 [00:00<?, ?it/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. Filling null values
3. Checking and removing Duplicates rows

1. Checking null Values

```
In [24]: m = expenditure.isnull().sum()
```

```
In [25]: m
```

```
Out[25]: Exp Category      0
         State             0
         Year             17
         value           217
         dtype: int64
```

```
In [26]: #missing Values in percentage
         m1 = m/len(expenditure)*100
```

```
In [27]: m1
```

```
Out[27]: Exp Category    0.000000
         State           0.000000
         Year            0.194219
         value           2.479150
         dtype: float64
```

```
In [28]: #missing values with %
         pd.concat([m,m1],axis = 1,keys =['Total','Missing %'])
```

Out[28]:

	Total	Missing %
Exp Category	0	0.000000
State	0	0.000000
Year	17	0.194219
value	217	2.479150

- Year having 0.19% and value having 2.4% missing values contains in the dataset

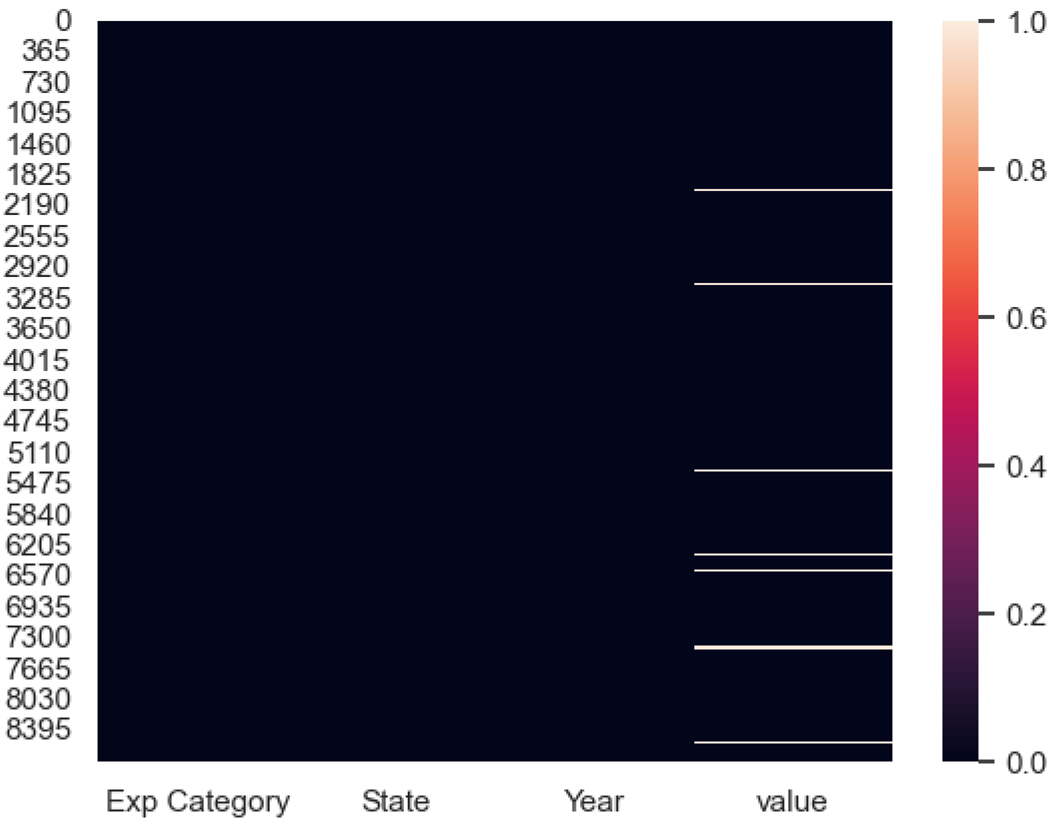
Null values shown by heatmap

In [29]:

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

Out[29]:

```
<AxesSubplot:>
```



2. Filling Null values

- filling null values with 0.

In [30]:

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

Out[30]:

	Exp Category	State	Year	value
0	Aggregate	Andhra Pradesh	1980-81	1610
1	Aggregate	Andhra Pradesh	1981-82	1611
2	Aggregate	Andhra Pradesh	1982-83	1612
3	Aggregate	Andhra Pradesh	1983-84	1613
4	Aggregate	Andhra Pradesh	1984-85	1614

In [31]:
```
exp_data.fillna(0,inplace = True)
```

In [32]:
```
#checking missing values again
exp_data.isnull().sum()
```

Out[32]:
```
Exp Category    0
State           0
Year            0
value           0
dtype: int64
```

3. Checking and removing Duplicates rows

In [33]:
```
exp_data[exp_data.duplicated()]  # duplicates rows
```

Out[33]:

	Exp Category	State	Year	value
3331	Exp Category	State	Year	Value
4376	Exp Category	State	Year	Value
5421	Exp Category	State	Year	Value
6530	Exp Category	State	Year	Value

In [34]:
```
expenditure.duplicated().sum() #number of duplicates rows
```

Out[34]:
```
4
```

- only 4 rows are duplicates.
- so lets drop them for better analysis.

In [35]:
```
exp_data.drop_duplicates(inplace=True)
```

In [36]:
```
#again checking for duplicates
exp_data.duplicated().sum()
```

Out[36]:
```
0
```

In [37]:
```
#checking size after cleaning
exp_data.shape
```

Out[37]:
```
(8749, 4)
```

4.4 Post Profiling

In [38]:

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

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Out[38]:

In [39]:

```
# save clean profile file

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

```
Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]
```

In [40]:

```
#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 [41]:

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

Out[41]: __

- There is no feature for correlation.

2. All unique categories of expenditure.

In [42]:

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

Out[42]:

	Exp Category	State	Year	value
0	Aggregate	Andhra Pradesh	1980-81	1610
1	Aggregate	Andhra Pradesh	1981-82	1611

```
In [43]: exp_data['Exp Category'].nunique()
```

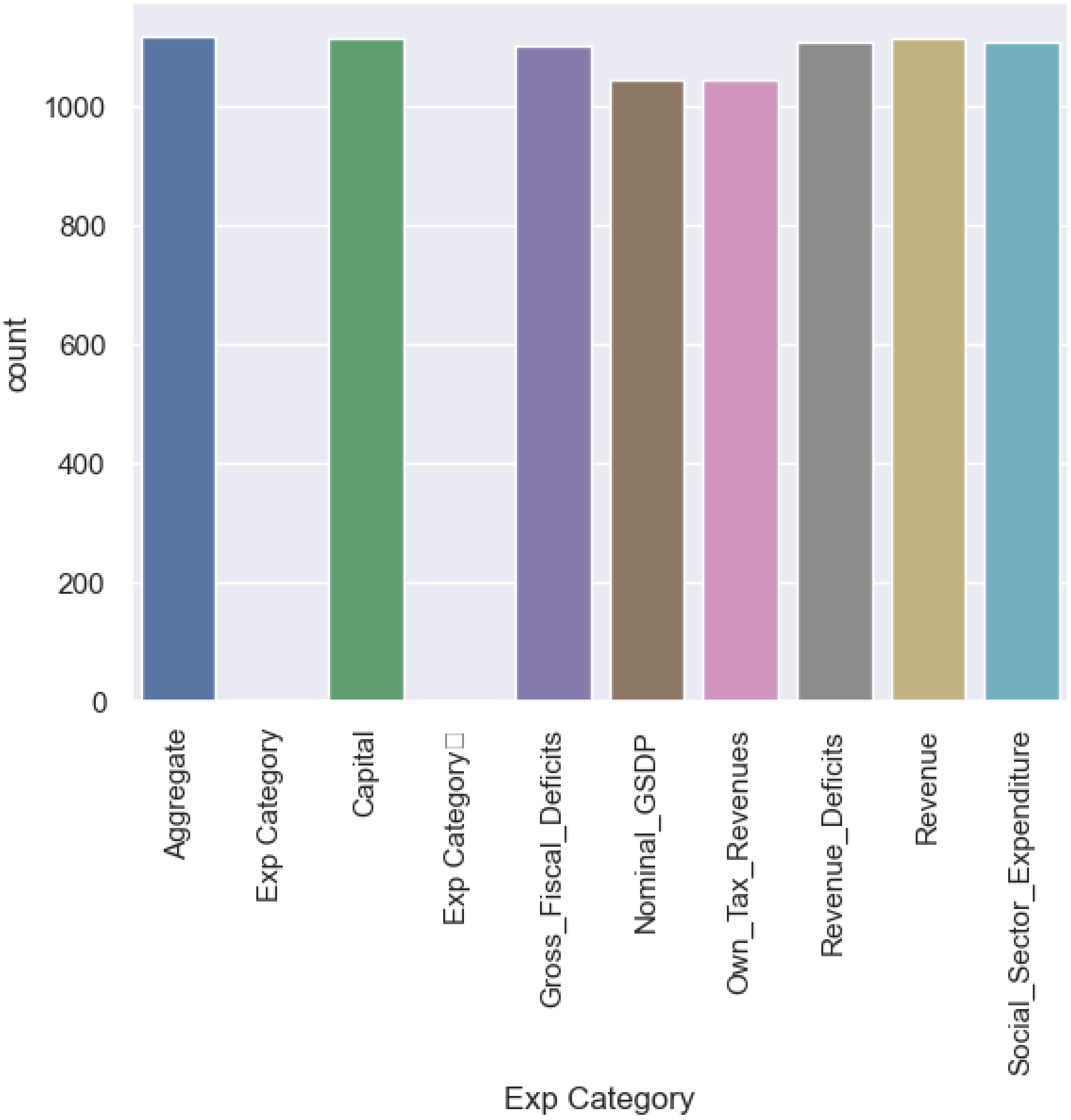
```
Out[43]: 10
```

In [44]:

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

Out[44]:

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'Aggregate'),
  Text(1, 0, 'Exp Category'),
  Text(2, 0, 'Capital'),
  Text(3, 0, 'Exp Category\t'),
  Text(4, 0, 'Gross_Fiscal_Deficits'),
  Text(5, 0, 'Nominal_GSDP'),
  Text(6, 0, 'Own_Tax_Revenues'),
  Text(7, 0, 'Revenue_Deficits'),
  Text(8, 0, 'Revenue'),
  Text(9, 0, 'Social_Sector_Expenditure')])
```



Insights : If we ignore Exp Category, Its clearly shown there are 8 expenditure categories in this NITI Aayog dataset.

3. Names of all States in india.

```
In [45]: exp_data['State'].nunique()
```

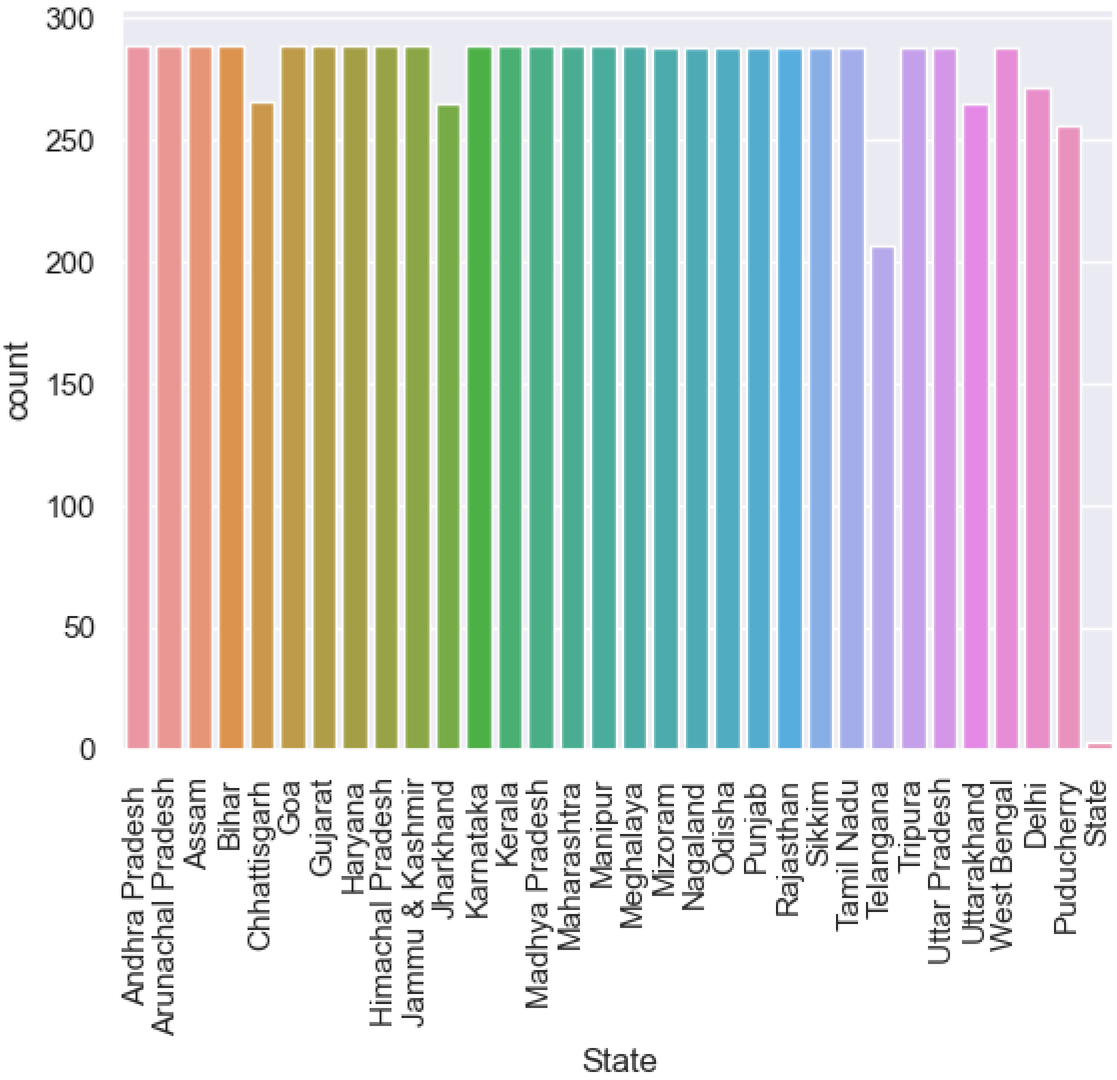
```
Out[45]: 32
```

In [49]:

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

Out[49]:

```
(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, 31]),
 [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'),
  Text(31, 0, 'State')])
```



Insights: If we ignore 31 text, 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 [50]: exp_data['Exp Category'].describe(include=all)
```

```
Out[50]: count          8749
         unique           10
         top       Aggregate
         freq           1116
         Name: Exp Category, dtype: object
```

```
In [51]: exp_data.groupby("Exp Category")["State"].agg(pd.Series.mode)
```

```
Out[51]: Exp Category
         Aggregate                    [Andhra Pradesh , Arunachal Pradesh, Assam, B
         i...
         Capital                      [Andhra Pradesh , Arunachal Pradesh, Assam, B
         i...
         Exp Category                                                             Sta
         te
         Exp Category\t                                                           Sta
         te
         Gross_Fiscal_Deficits        [Andhra Pradesh , Arunachal Pradesh, Assam, B
         i...
         Nominal_GSDP                 [Andhra Pradesh , Arunachal Pradesh, Assam, B
         i...
         Own_Tax_Revenues             [Andhra Pradesh , Arunachal Pradesh, Assam, B
         i...
         Revenue                      [Andhra Pradesh , Arunachal Pradesh, Assam, B
         i...
         Revenue_Deficits             [Andhra Pradesh , Arunachal Pradesh, Assam, B
         i...
         Social_Sector_Expenditure    [Andhra Pradesh , Arunachal Pradesh, Assam, B
         i...
         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 [52]: exp_data.groupby(['Exp Category','State']).count()["value"]
```

```
Out[52]: Exp Category                State
         Aggregate                   Andhra Pradesh       36
                                     Arunachal Pradesh    36
                                     Assam                36
                                     Bihar                36
                                     Chhattisgarh         36
                                                          ..
         Social_Sector_Expenditure   Telangana            28
                                     Tripura              36
                                     Uttar Pradesh        36
                                     Uttarakhand          36
                                     West Bengal          36
         Name: value, Length: 250, 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

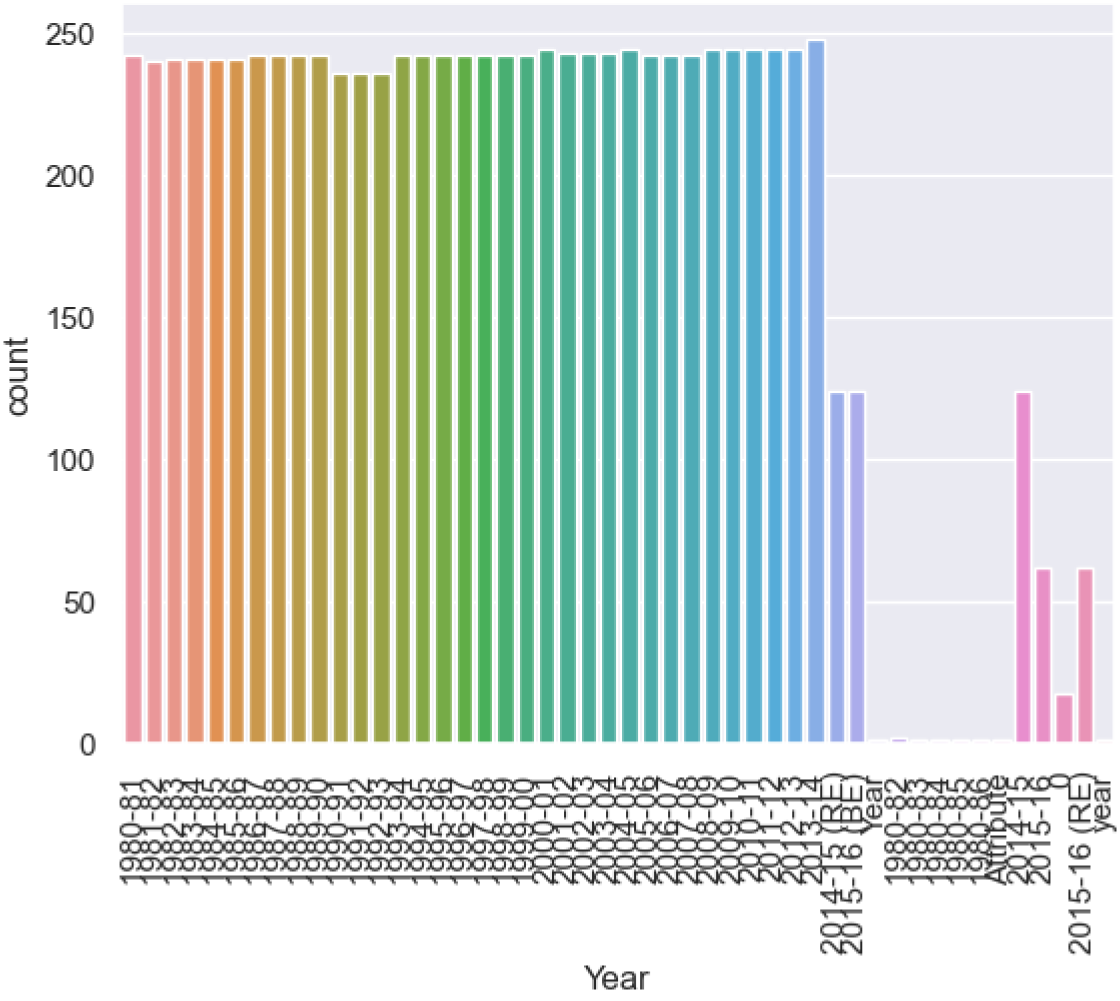
```
In [53]: exp_data.Year.value_counts().to_frame('value')
```

Out[53]:

	value
2013-14	248
2004-05	244
2012-13	244
2011-12	244
2010-11	244
2009-10	244
2008-09	244
2000-01	244
2002-03	243
2001-02	243
2003-04	243
1999-00	242
2007-08	242
2005-06	242
2006-07	242
1997-98	242
1998-99	242
1980-81	242
1996-97	242
1986-87	242
1994-95	242
1993-94	242
1995-96	242
1989-90	242
1988-89	242
1987-88	242
1983-84	241
1984-85	241
1985-86	241
1982-83	241
1981-82	240
1990-91	236
1991-92	236
1992-93	236
2014-15 (RE)	124
2015-16 (BE)	124

	value
2014-15	124
2015-16 (RE)	62
2015-16	62
0	17
1980-82	2
Attribute	1
1980-84	1
1980-86	1
1980-85	1
1980-83	1
Year	1
year	1

In [55]:
```
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.