consumer-price-index

March 29, 2025

[1]:

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

Load Dataset

[2]:

```
df2=pd.read_csv("/content/cpi Group data -modify.csv")
```

[3]:

```
df2.head()
```

[3]:

```
   BaseYear  Year    Month           State    Sector                Group  \
0      2012  2025  January  Andhra Pradesh  Combined              General
1      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages
2      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages
3      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages
4      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages

               SubGroup  Index Inflation (%)
0                     *  199.1          4.02
1  Cereals and Products  199.5          3.48
2         Meat and Fish  237.6          6.69
3                   Egg  238.4          1.02
4    Milk and Products  180.9          1.86
```

df=df2.drop(df2[df2[“SubGroup”]==”*”].index) l=[“Egg”,“Footwear”,“Personal Care and Effects”,”*”] for x in l: df2=df2.drop(df2[df2[“SubGroup”]==x].index) l=[“General”,“Miscellaneous”] for x in l: df2=df2.drop(df2[df2[“Group”]==x].index)

Data Cleaning and Preprocessing

[4]:

```
df2.shape
```

[4]:

```
(56745, 9)
```

[5]:

```
df2.rename(columns={"Inflation (%)":"Inflation"},inplace=True)
```

[6]:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 56745 entries, 0 to 56744
Data columns (total 9 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   BaseYear   56745 non-null  int64
 1   Year       56745 non-null  int64
 2   Month      56745 non-null  object
 3   State      56745 non-null  object
 4   Sector     56745 non-null  object
 5   Group      56745 non-null  object
 6   SubGroup   56745 non-null  object
 7   Index      56745 non-null  object
 8   Inflation  56745 non-null  object
dtypes: int64(2), object(7)
memory usage: 3.9+ MB
```

[7]:

```
for x in df2.columns:
  print(x,":",df2[x].unique())
```

```
BaseYear : [2012]
Year : [2025 2024 2023 2022 2021 2020 2019 2018 2017 2016 2015 2014 2013]
Month : ['January' 'December' 'November' 'October' 'September' 'August' 'July'
 'June' 'May' 'April' 'March' 'February']
State : ['Andhra Pradesh' 'Karnataka' 'Kerala' 'Tamil Nadu' 'Telangana']
Sector : ['Combined' 'Rural' 'Urban']
Group : ['General' 'Food and Beverages' 'Pan, Tobacco and Intoxicants'
 'Clothing and Footwear' 'Housing' 'Fuel and Light' 'Miscellaneous']
SubGroup : ['*' 'Cereals and Products' 'Meat and Fish' 'Egg' 'Milk and Products'
 'Oils and Fats' 'Fruits' 'Vegetables' 'Pulses and Products'
 'Sugar and Confectionery' 'Spices' 'Prepared Meals, Snacks, Sweets etc.'
 'Non-alcoholic Beverages' 'Clothing' 'Footwear'
 'Household Goods and Services' 'Health' 'Transport and Communication'
 'Recreation and Amusement' 'Education' 'Personal Care and Effects']
Index : ['199.1' '199.5' '237.6' … '99.3' '95.6' '94.6']
Inflation : ['4.02' '3.48' '6.69' … '41.43' '27.02' '17.22']
```

[8]:

```
for x in df2.columns:
  print(x,":",len(df2[df2[x]=="*"]))
```

```
BaseYear : 0
Year : 0
Month : 0
State : 0
Sector : 0
Group : 0
```

```
SubGroup : 14745
Index : 1388
Inflation : 7298
```

[9]:
```
df2.replace("*",np.nan,inplace=True)
```

[10]:
```
df2["SubGroup"].fillna("others",inplace=True)
```

[11]:
```
df2["Index"]=df2["Index"].astype("float64")
df2["Inflation"]=df2["Inflation"].astype("float64")
```

[12]:
```
df2["Inflation"].agg(["mean","max","min"])
```

[12]:
```
mean     5.543502
max     93.190000
min    -42.900000
Name: Inflation, dtype: float64
```

[13]:
```
df2["Index"].agg(["mean","max","min"])
```

[13]:
```
mean    149.64443
max     356.40000
min      81.90000
Name: Index, dtype: float64
```

[14]:
```
df2["Index"].fillna(df2["Index"].mean(),inplace=True)
df2["Inflation"].fillna(df2["Inflation"].mean(),inplace=True)
```

[15]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 56745 entries, 0 to 56744
Data columns (total 9 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   BaseYear   56745 non-null  int64
 1   Year       56745 non-null  int64
 2   Month      56745 non-null  object
 3   State      56745 non-null  object
 4   Sector     56745 non-null  object
 5   Group      56745 non-null  object
 6   SubGroup   56745 non-null  object
 7   Index      56745 non-null  float64
 8   Inflation  56745 non-null  float64
dtypes: float64(2), int64(2), object(5)
memory usage: 3.9+ MB
```

[16]:

```
df2.isnull().sum()
```

[16]:

```
BaseYear     0
Year         0
Month        0
State        0
Sector       0
Group        0
SubGroup     0
Index        0
Inflation    0
dtype: int64
```

[17]:

```
df2.describe()
```

[17]:

```
       BaseYear          Year         Index     Inflation
count   56745.0  56745.000000  56745.000000  56745.000000
mean     2012.0   2018.494052    149.644430      5.543502
std         0.0      3.531380     32.772797      7.285684
min      2012.0   2013.000000     81.900000    -42.900000
25%      2012.0   2015.000000    123.400000      2.890000
50%      2012.0   2018.000000    144.300000      5.543502
75%      2012.0   2022.000000    171.300000      7.420000
max      2012.0   2025.000000    356.400000     93.190000
```

Data Visualization

[18]:

```
# prompt: list year 2013 of months

# Assuming 'df2' is the DataFrame from the provided code.
#df_2013 = df2[df2["Year"] == 2013]
#months_2013 = df_2013["Month"].unique()
#months_2013
```

[19]:

```
df2.head()
```

[19]:

```
   BaseYear  Year    Month           State    Sector                Group  \
0      2012  2025  January  Andhra Pradesh  Combined              General
1      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages
2      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages
3      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages
4      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages

              SubGroup  Index  Inflation
0               others  199.1       4.02
1  Cereals and Products  199.5       3.48
2        Meat and Fish  237.6       6.69
3                  Egg  238.4       1.02
```

```
4     Milk and Products  180.9       1.86
```

[20]:

```
group=df2.groupby("Year")
d={}
for x,y in group:
  d[x]=y["Inflation"].mean()
```

[21]:

```
gropinf= pd.DataFrame(list(d.items()), columns=["Year", "Inflation"])
gropinf
```

[21]:

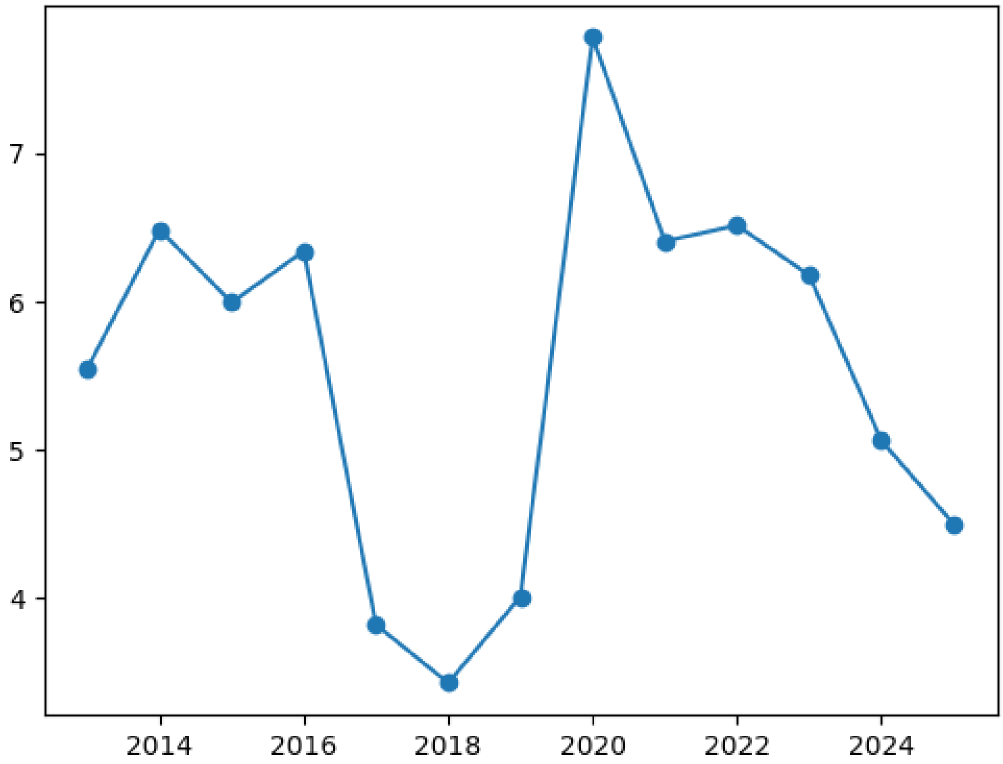
```
    Year  Inflation
0   2013   5.543502
1   2014   6.481564
2   2015   5.990237
3   2016   6.336910
4   2017   3.815702
5   2018   3.422925
6   2019   4.005196
7   2020   7.778755
8   2021   6.403802
9   2022   6.510058
10  2023   6.173776
11  2024   5.057570
12  2025   4.495790
```

[22]:

```
plt.scatter(gropinf["Year"],gropinf["Inflation"])
plt.plot(gropinf["Year"],gropinf["Inflation"])
```

[22]:

```
[<matplotlib.lines.Line2D at 0x79fbdb9de1d0>]
```



[23]:
```
groups=df2.groupby("State")
d1={}
for x,y in groups:
  d1[x]=y["Inflation"].mean()
```

[24]:
```
d1
```

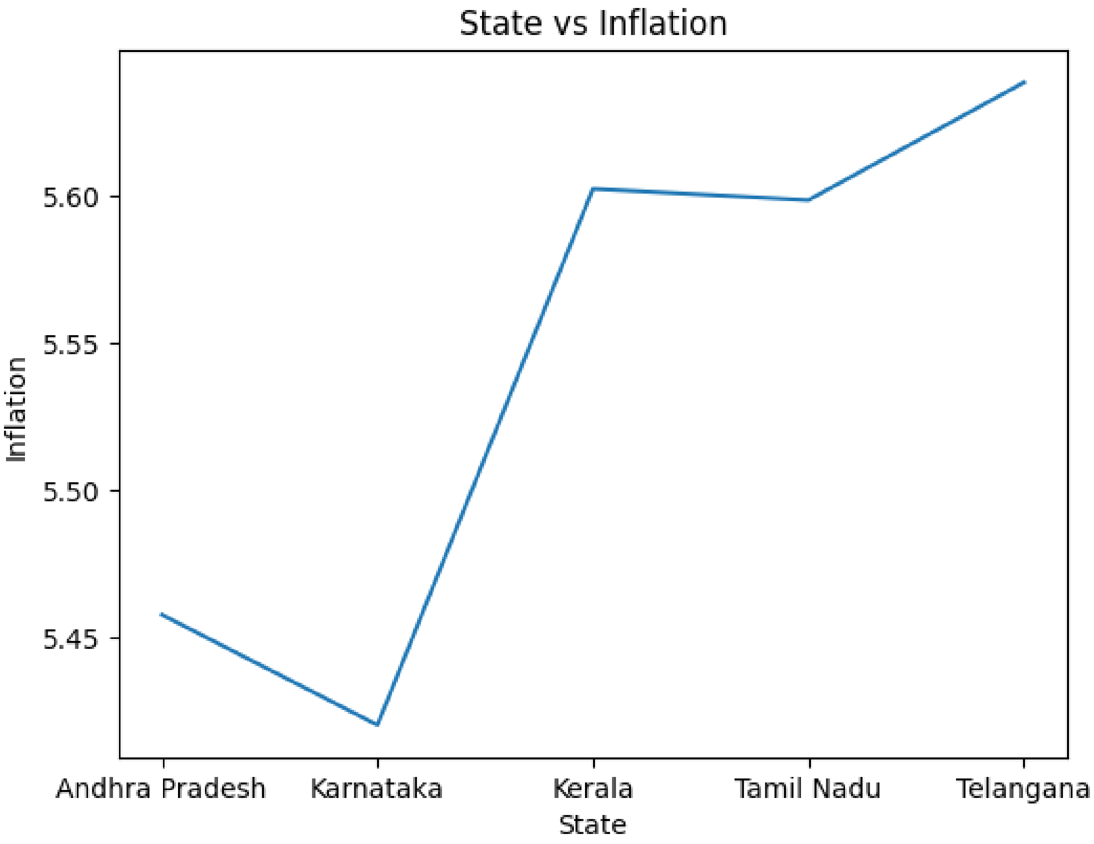
[24]:
```
{'Andhra Pradesh': np.float64(5.457833281168586),
 'Karnataka': np.float64(5.420392978058181),
 'Kerala': np.float64(5.602334162920943),
 'Tamil Nadu': np.float64(5.59852409093156),
 'Telangana': np.float64(5.6384271660923675)}
```

[25]:
```
gropinfs= pd.DataFrame(list(d1.items()), columns=["State", "Inflation"])
gropinfs
```

[25]:
```
            State  Inflation
0  Andhra Pradesh   5.457833
1       Karnataka   5.420393
2          Kerala   5.602334
3      Tamil Nadu   5.598524
4       Telangana   5.638427
```

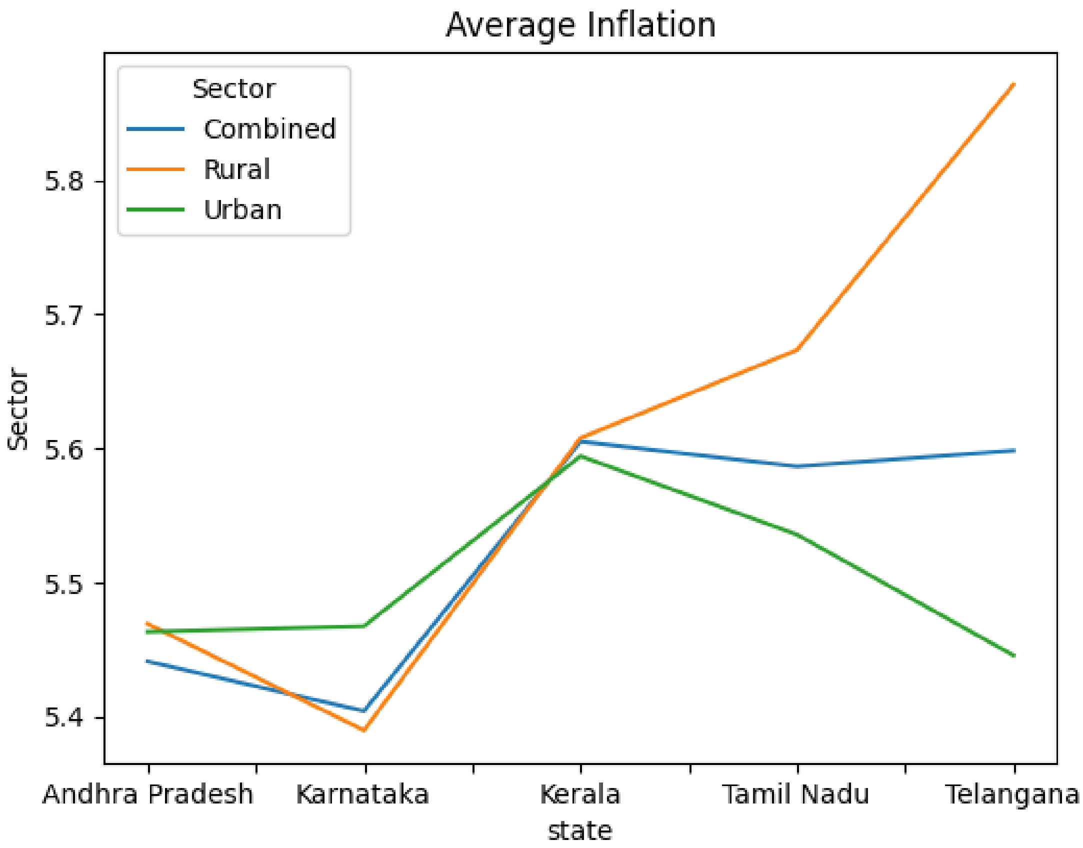
```
[26]: plt.plot(gropinfs["State"],gropinfs["Inflation"])
      plt.xlabel("State")
      plt.ylabel("Inflation")
      plt.title("State vs Inflation")
```

```
[26]: Text(0.5, 1.0, 'State vs Inflation')
```



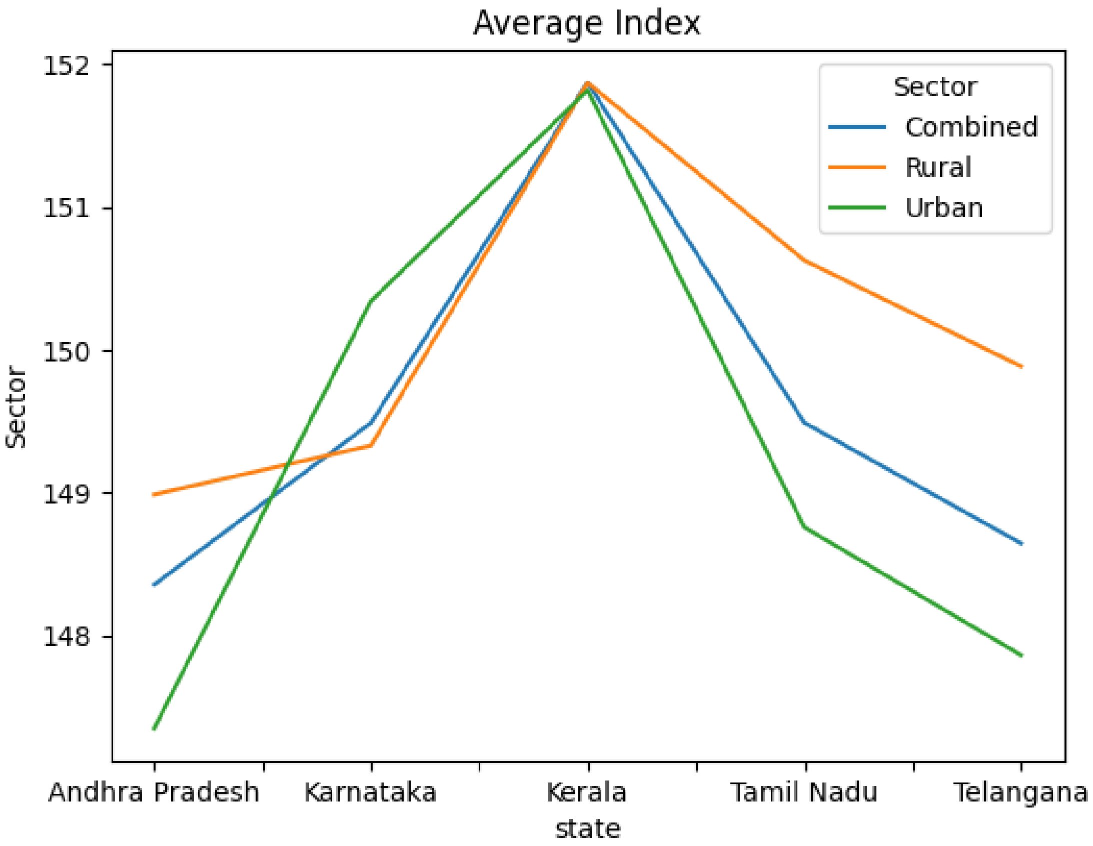
```
[27]: pd.crosstab(df2["State"],df2["Sector"],values=df2.Inflation,aggfunc=np.average).
        ↪plot()
      plt.xlabel("state")
      plt.ylabel("Sector")
      plt.title("Average Inflation")
```

```
[27]: Text(0.5, 1.0, 'Average Inflation')
```



```
[28]: pd.crosstab(df2["State"],df2["Sector"],values=df2.Index,aggfunc=np.average).
        ↪plot()
      plt.xlabel("state")
      plt.ylabel("Sector")
      plt.title("Average Index")
```

```
[28]: Text(0.5, 1.0, 'Average Index')
```

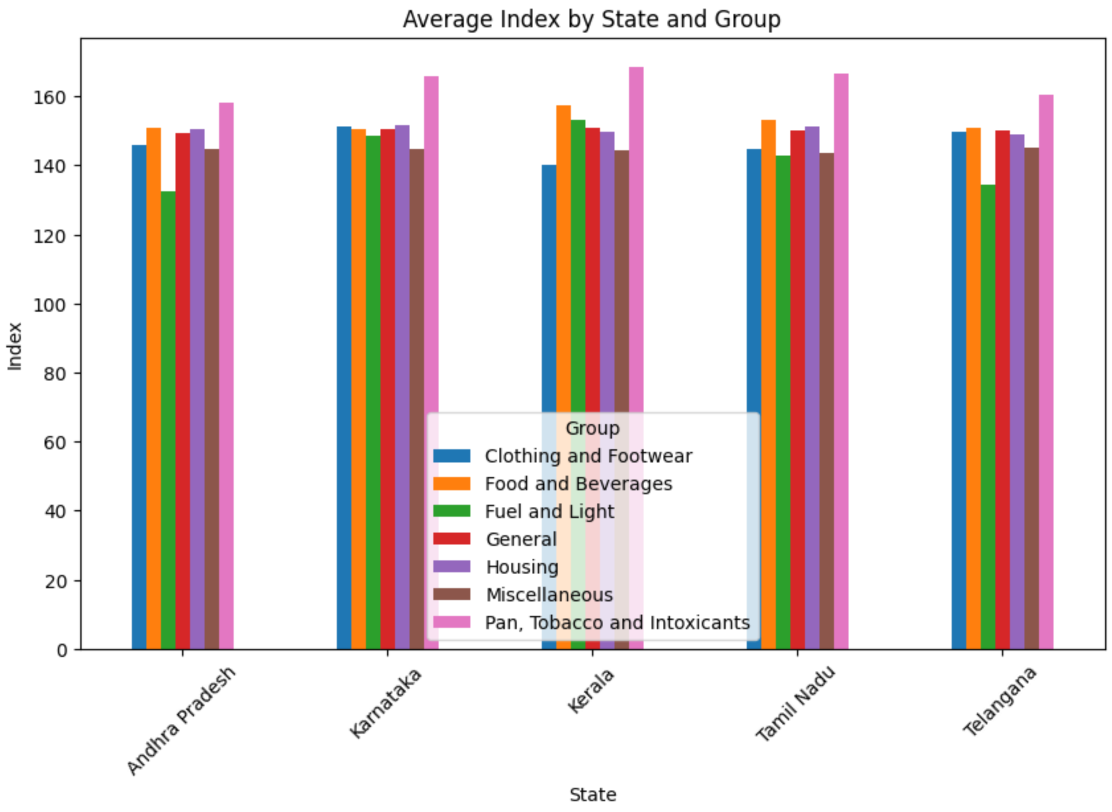


High Index = Higher costs or improved facilities. Low Index = Lower costs or potentially fewer amenities.

[29]:
```
df2.head()
```

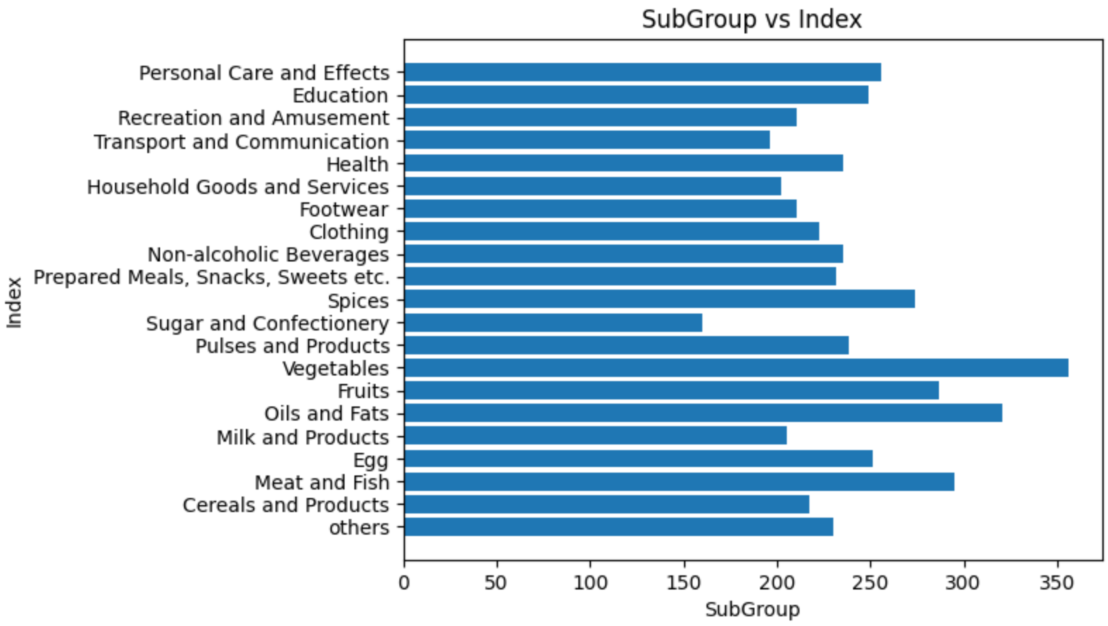
[29]:
```
   BaseYear  Year    Month           State    Sector               Group  \
0      2012  2025  January  Andhra Pradesh  Combined             General
1      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages
2      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages
3      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages
4      2012  2025  January  Andhra Pradesh  Combined  Food and Beverages

                SubGroup  Index  Inflation
0                 others  199.1       4.02
1  Cereals and Products  199.5       3.48
2          Meat and Fish  237.6       6.69
3                    Egg  238.4       1.02
4     Milk and Products  180.9       1.86
```

[30]:
```
pd.crosstab(df2["State"],df2["Group"],values=df2.Index,aggfunc=np.average).plot.
  ↪bar(figsize=(10, 6), legend=True)
plt.xlabel("State")
plt.ylabel("Index")
plt.title("Average Index by State and Group")
plt.xticks(rotation=45)
plt.show()
```



[31]:
```
plt.barh(df2["SubGroup"],df2["Index"])
plt.xlabel("SubGroup")
plt.ylabel("Index")
plt.title("SubGroup vs Index")
```

[31]:
```
Text(0.5, 1.0, 'SubGroup vs Index')
```



[32]:

```
group1=df2.groupby("Year")
d1={}
for x,y in group1:
  d1[x]=y["Inflation"].mean()
```

[33]:

```
group2=df2.groupby("Year")
d2={}
for x,y in group2:
  d2[x]=y["Index"].mean()
```

[34]:

```
inf1= pd.DataFrame(list(d1.items()), columns=["Year", "Inflation"])
inf2= pd.DataFrame(list(d2.items()), columns=["Year", "Index"])
```

[35]:

```
merge=pd.merge(inf1,inf2,on="Year")
merge.sort_values(by=["Year"],inplace=True)
```

[36]:

```
merge
```

[36]:

```
   Year  Inflation       Index
0  2013   5.543502  111.235644
1  2014   6.481564  118.078710
2  2015   5.990237  124.923690
3  2016   6.336910  132.264183
4  2017   3.815702  136.516550
5  2018   3.422925  140.779931
6  2019   4.005196  145.978957
```

```
7   2020   7.778755  157.909760
8   2021   6.403802  166.723450
9   2022   6.510058  176.969225
10  2023   6.173776  187.223031
11  2024   5.057570  196.271385
12  2025   4.495790  200.356406
```

[37]:

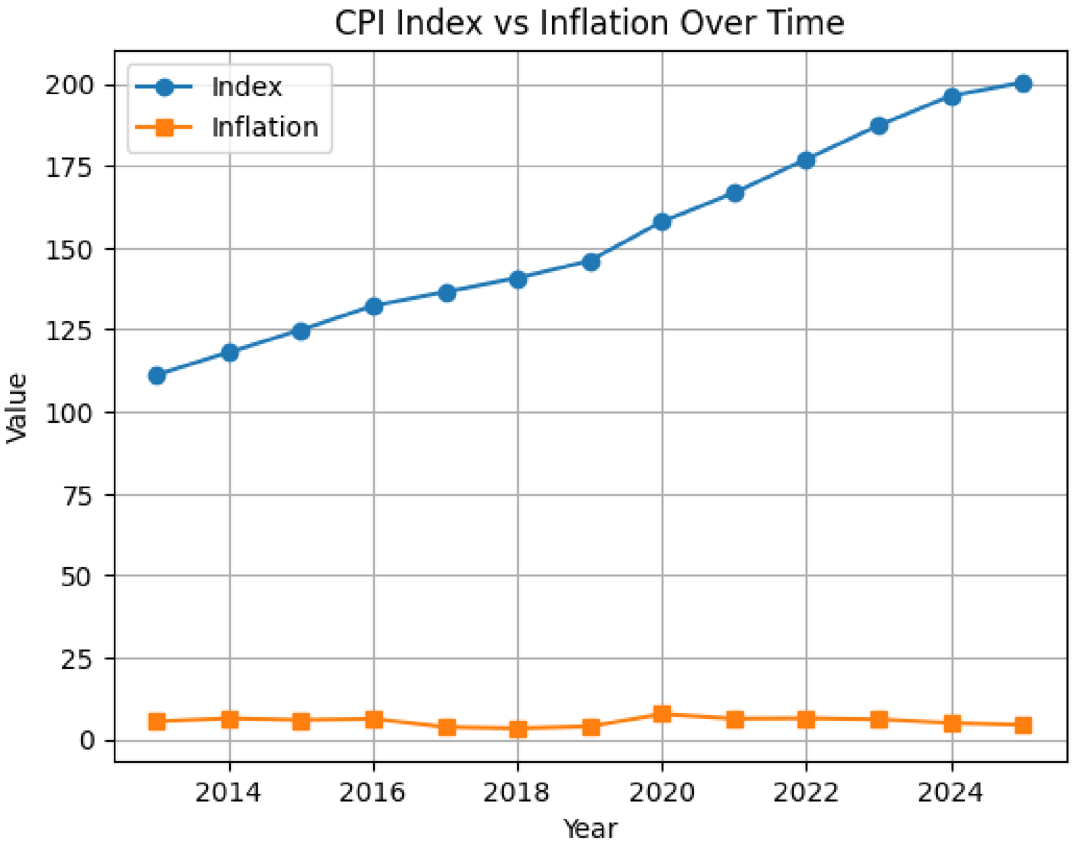
```
merge.corr()
```

[37]:

```
               Year  Inflation     Index
Year       1.000000  -0.043481  0.992541
Inflation -0.043481   1.000000  0.000736
Index      0.992541   0.000736  1.000000
```

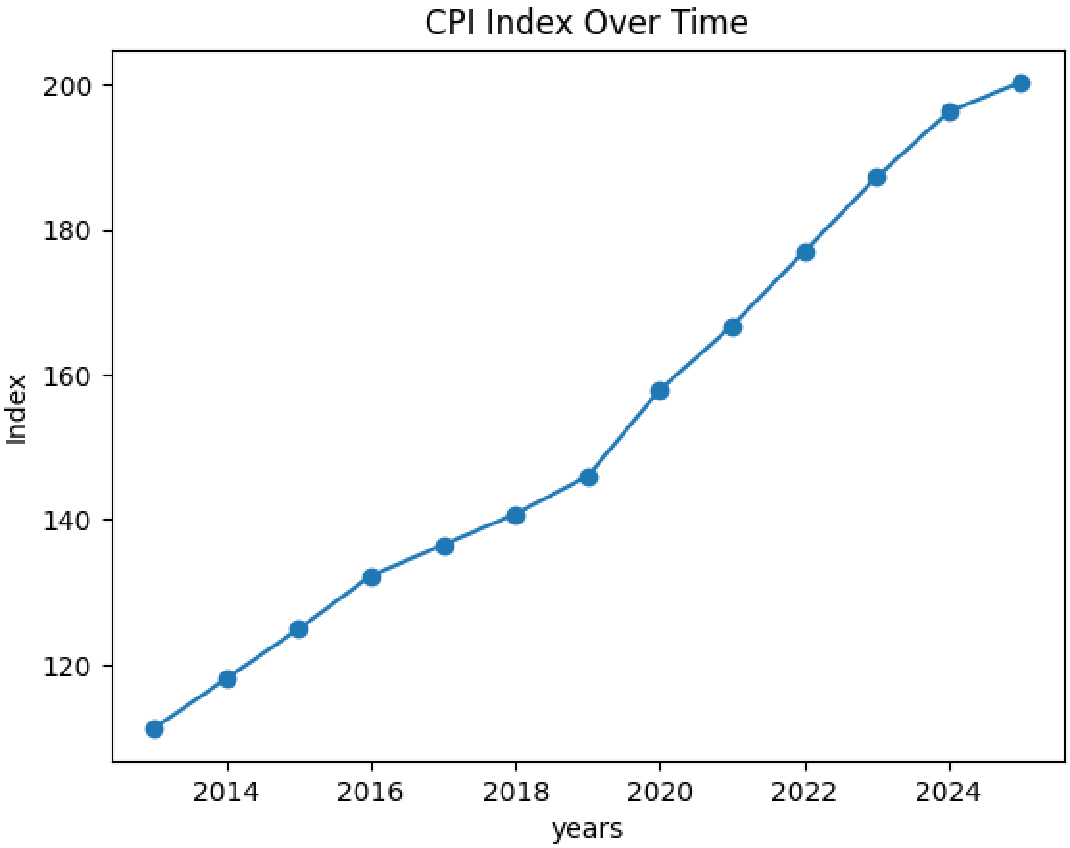
[38]:

```
plt.plot(merge["Year"], merge["Index"], label='Index', marker='o')
plt.plot(merge["Year"], merge["Inflation"], label='Inflation', marker='s')

plt.xlabel('Year')
plt.ylabel('Value')
plt.title('CPI Index vs Inflation Over Time')
plt.legend()
plt.grid(True)
plt.show()
```



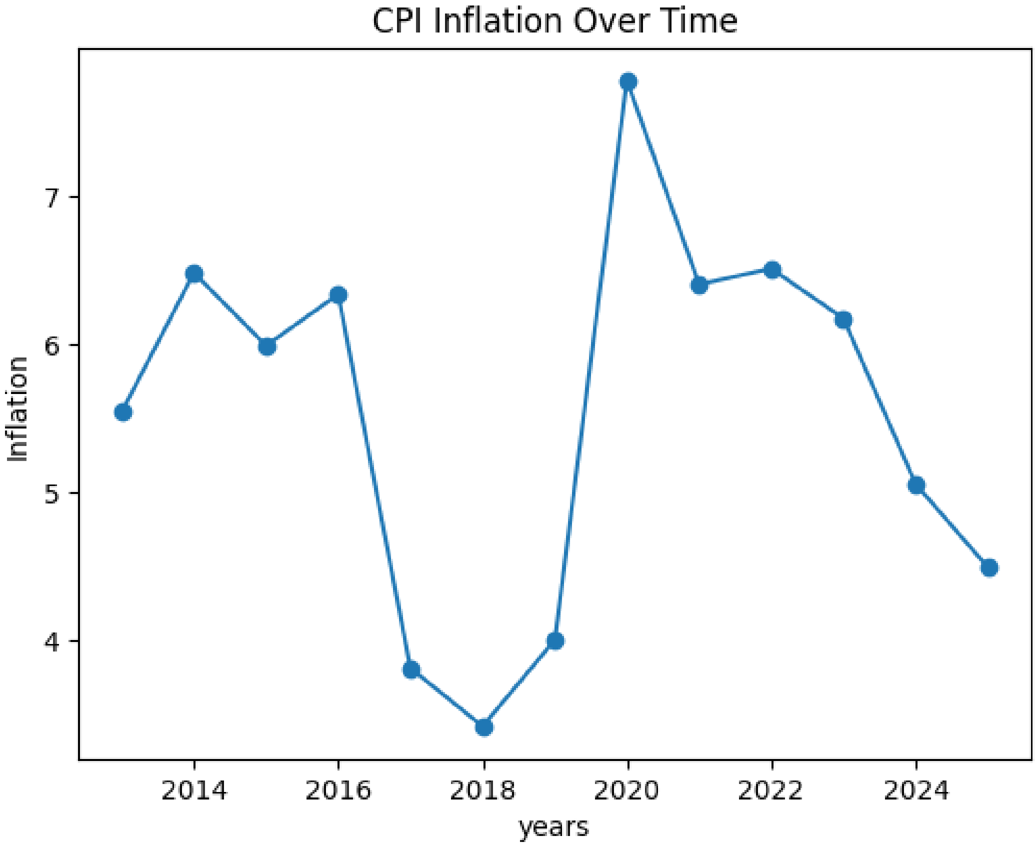
```
[39]: plt.plot(merge["Year"], merge["Index"], label='Index', marker='o')
      plt.xlabel("years")
      plt.ylabel("Index")
      plt.title("CPI Index Over Time")
```

```
[39]: Text(0.5, 1.0, 'CPI Index Over Time')
```



```
[42]: plt.plot(merge["Year"], merge["Inflation"], label='Index', marker='o')
      plt.xlabel("years")
      plt.ylabel("Inflation")
      plt.title("CPI Inflation Over Time")
```

```
[42]: Text(0.5, 1.0, 'CPI Inflation Over Time')
```



from derived insights ,I conclude that from above graphs the index values is increases year by year and inflation is flucation year by year