Project Title: Analysis of Foreign Direct Investment (FDI) in India (2000-01 to 2016-17)

Introduction

Understanding the trends and dynamics of Foreign Direct Investment (FDI) is crucial for evaluating economic growth and sectoral performance. This project focuses on analyzing the FDI inflows in India over the past 17 years, from the fiscal year 2000-01 to 2016-17. The goal is to provide actionable insights by comparing annual FDI inflows, sectoral distribution, and identifying key trends and patterns.

Problem Statement

Investment strategies require a deep understanding of historical FDI data and trends. This analysis aims to examine the inflows of FDI into India, identifying periods of significant growth, decline, and sectoral preferences. The dataset includes annual FDI inflow figures and sector-wise distribution to help management and stakeholders make informed decisions.

About the Project

Objective: To analyze FDI inflows into India over the last 17 years by evaluating annual inflows and sectoral distribution to uncover trends, fluctuations, and key insights.

Technologies and Libraries Used:

- **Python**: For data processing, cleaning, and analysis.
- **Pandas**: For data manipulation and analysis.
- **NumPy**: For numerical operations.
- **Matplotlib**: For data visualization.
- **Plotly**: For interactive charts and visualizations.

Key Questions

1. What are the annual trends in FDI inflows from 2000-01 to 2016-17?
2. Which sectors received the highest and lowest FDI inflows?
3. How did FDI inflows fluctuate across different financial years?
4. What are the key periods of significant growth or decline in FDI inflows?

In [ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
import plotly.io as pt
import plotly.express as px
import plotly.graph_objects as go
from sklearn.impute import SimpleImputer
from fancyimpute import IterativeImputer
from sklearn.preprocessing import StandardScaler
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import plotly.io as pio
import random
from IPython.display import Image
import plotly.figure_factory as ff
import plotly.subplots as sp
import plotly.graph_objects as go
pd.set_option("display.precision",3)
np.set_printoptions(precision=5, suppress=True)
pd.options.display.float_format = '{:.4f}'.format

pio.renderers.default = "svg"
%matplotlib inline
```

In [ ]:
```
df = pd.read_csv("FDI data.csv")
```

In [ ]:
```
df.head(10)
```

Out[ ]:

	Sector	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	20
0	METALLURGICAL INDUSTRIES	22.6900	14.1400	36.6100	8.1100	200.3800	149.1300	169.9400	1175.7500	959.9400	419.8800	1098.1400	1786.1400	1466.2300	567.6300	359.3400	456.3100	144(
1	MINING	1.3200	6.5200	10.0600	23.4800	9.9200	7.4000	6.6200	444.3600	34.1600	174.4000	79.5100	142.6500	57.8900	12.7300	684.3900	520.6700	5!
2	POWER	89.4200	757.4400	59.1100	27.0900	43.3700	72.6900	157.1500	988.6800	907.6600	1271.7900	1271.7700	1652.3800	535.6800	1066.0800	707.0400	868.8000	111
3	NON-CONVENTIONAL ENERGY	0.0000	0.0000	1.7000	4.1400	1.2700	1.3500	2.4400	58.8200	125.8800	622.5200	214.4000	452.1700	1106.5200	414.2500	615.9500	776.5100	78:
4	COAL PRODUCTION	0.0000	0.0000	0.0000	0.0400	0.0000	9.1400	1.3000	14.0800	0.2200	0.0000	0.0000	0.0000	0.0000	2.9600	0.0000	0.0000	(
5	PETROLEUM & NATURAL GAS	9.3500	211.0700	56.7800	80.6400	102.7800	12.0900	87.7100	1405.0400	349.2900	265.5300	556.4300	2029.9800	214.8000	112.2300	1079.0200	103.0200	18(
6	BOILERS AND STEAM GENERATING PLANTS	0.0000	0.0000	0.0000	0.0400	0.5400	0.0000	3.3100	1.5100	0.0000	3.9600	0.6300	31.7900	20.0500	0.1700	1.3300	77.9100	5:
7	PRIME MOVER (OTHER THAN ELECTRICAL GENERATORS)	0.0000	0.0000	0.0000	0.0000	2.6600	0.7400	25.5700	40.5300	74.8800	39.5000	166.4400	313.7500	184.6000	212.7800	230.7000	159.1300	28(
8	ELECTRICAL EQUIPMENTS	79.7600	65.7600	34.7100	73.2000	97.4000	39.5000	76.8500	653.7400	417.3500	728.2700	153.9000	566.3900	195.8700	134.3100	574.8300	444.8800	223(
9	COMPUTER SOFTWARE & HARDWARE	228.3900	419.3900	314.2400	368.3200	527.9000	1359.9700	2613.3300	1382.2500	1543.3400	871.8600	779.8100	796.3500	485.9600	1126.2700	2296.0400	5904.3600	365

In [ ]:
```
df.tail(10)
```

Out[ ]:

	Sector	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	
53	TRADING	11.4900	43.2700	38.1300	31.1200	14.2200	28.9300	114.6500	345.0200	643.6400	737.9500	498.0400	731.5500	717.8000	1343.3900	2727.9600	3845.3200	2
54	RETAIL TRADING	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.2700	0.0900	13.7300	26.2700	31.7000	22.3100	11.3000	168.7200	262.2400	
55	AGRICULTURE SERVICES	17.5200	14.0600	11.0100	0.5900	3.8300	9.0800	12.5300	58.1300	5.3500	1222.2200	43.9000	49.0200	161.4700	91.0100	59.9500	84.6500	
56	DIAMOND,GOLD ORNAMENTS	18.8300	0.3600	1.3000	1.9600	8.5800	15.5200	61.9700	59.1500	83.5000	31.0800	19.5900	36.3000	52.6100	42.5600	280.1800	58.5400	
57	TEA AND COFFEE (PROCESSING & WAREHOUSING COFFE...	20.2300	0.1400	0.0000	0.3200	0.0100	1.4300	6.2000	18.9400	37.0800	8.1500	3.1200	5.3200	0.2700	5.8600	1.4300	1.1200	
58	PRINTING OF BOOKS (INCLUDING LITHO PRINTING IN...	0.0000	0.0000	6.3000	0.0000	0.0600	9.9000	20.0400	35.5400	31.6100	70.5100	36.6300	47.3900	14.3400	113.7800	72.5800	122.8100	
59	COIR	0.0000	0.0000	0.0000	0.0000	0.4700	0.5900	0.0400	0.0100	0.0000	0.2500	0.1000	0.5500	0.1500	0.5400	1.3600	0.0000	
60	CONSTRUCTION (INFRASTRUCTURE) ACTIVITIES	0.0000	0.0000	0.0000	0.0000	0.0000	0.9300	64.0600	182.9200	172.7000	324.5600	675.0700	386.2800	283.8900	485.3700	870.2500	4510.7100	1
61	CONSTRUCTION DEVELOPMENT: Townships, housing, ...	24.3300	51.7500	36.1000	47.0400	152.0600	228.7100	1392.9500	3887.3300	4657.5100	5466.1300	1663.0300	3140.7800	1332.4900	1226.0500	769.1400	112.5500	
62	MISCELLANEOUS INDUSTRIES	832.0700	221.3700	218.7600	235.4800	121.8300	164.7600	304.8700	528.4200	1549.7000	1147.5600	1475.9700	813.3800	229.4900	468.7400	765.8800	668.7700	

In [ ]:
```
df.sample(10)
```

Out[ ]:

	Sector	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
15	SEA TRANSPORT	2.4100	19.8100	29.3200	21.9500	36.9500	53.6300	72.5200	128.3600	50.2100	284.8500	300.5100	129.3600	64.6200	20.4900	333.2200	429.3
2	POWER	89.4200	757.4400	59.1100	27.0900	43.3700	72.6900	157.1500	988.6800	907.6600	1271.7900	1271.7700	1652.3800	535.6800	1066.0800	707.0400	868.8
57	TEA AND COFFEE (PROCESSING & WAREHOUSING COFFE...	20.2300	0.1400	0.0000	0.3200	0.0100	1.4300	6.2000	18.9400	37.0800	8.1500	3.1200	5.3200	0.2700	5.8600	1.4300	1.1
24	MEDICAL AND SURGICAL APPLIANCES	5.4200	42.3500	21.6300	1.9700	5.3500	1.5200	13.4300	13.1700	75.4200	167.3500	32.2200	141.6100	83.0200	173.4800	145.9300	173.2
31	DYE-STUFFS	1.0500	0.1800	0.0000	0.4300	1.1800	0.0000	0.0000	5.5100	1.1700	4.0200	5.3700	0.5800	0.0000	0.0000	54.8900	3.3
43	GLASS	33.8700	8.3700	44.9800	5.2400	8.3600	0.8100	1.4300	11.0400	23.1600	2.8300	7.6000	32.2200	209.1600	43.0900	41.8200	25.7
27	MATHEMATICAL,SURVEYING AND DRAWING INSTRUMENTS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.2700	0.0000	0.0000	0.0000	0.0000	6.7100	0.0000	0.0000	0.0
17	RAILWAY RELATED COMPONENTS	0.0000	0.0000	0.5600	2.9500	10.7500	22.6200	25.8200	12.4100	18.0100	34.4300	70.6600	42.2700	29.8500	236.9300	129.7300	73.9
13	AUTOMOBILE INDUSTRY	195.3300	235.7600	419.9600	119.0900	121.9700	139.9300	260.7200	656.1000	1150.0300	1236.2900	1299.4100	922.9900	1537.2800	1517.2800	2725.6400	2526.8
19	MACHINE TOOLS	1.4200	4.3100	14.1700	54.5100	11.0400	23.0000	37.2800	56.8700	45.6600	133.8300	11.6300	127.8700	101.3900	64.5200	24.0600	126.3

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63 entries, 0 to 62
Data columns (total 18 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   Sector   63 non-null     object 
 1   2000-01  63 non-null     float64
 2   2001-02  63 non-null     float64
 3   2002-03  63 non-null     float64
 4   2003-04  63 non-null     float64
 5   2004-05  63 non-null     float64
 6   2005-06  63 non-null     float64
 7   2006-07  63 non-null     float64
 8   2007-08  63 non-null     float64
 9   2008-09  63 non-null     float64
 10  2009-10  63 non-null     float64
 11  2010-11  63 non-null     float64
 12  2011-12  63 non-null     float64
 13  2012-13  63 non-null     float64
 14  2013-14  63 non-null     float64
 15  2014-15  63 non-null     float64
 16  2015-16  63 non-null     float64
 17  2016-17  63 non-null     float64
dtypes: float64(17), object(1)
memory usage: 9.0+ KB
```

In [ ]:
```
df.dtypes
```

Out[ ]:
```
Sector      object
2000-01    float64
2001-02    float64
2002-03    float64
2003-04    float64
2004-05    float64
2005-06    float64
2006-07    float64
2007-08    float64
2008-09    float64
2009-10    float64
2010-11    float64
2011-12    float64
2012-13    float64
2013-14    float64
2014-15    float64
2015-16    float64
2016-17    float64
dtype: object
```

In [ ]:
```
df.describe()
```

Out[ ]:

	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	2016-17
count	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000	63.0000
mean	37.7573	63.9316	42.9257	34.7278	51.0903	87.9325	198.2819	390.0857	498.3486	410.0695	339.4138	557.4727	355.9300	385.7035	490.9598	634.9363	690.1311
std	112.2279	157.8787	86.6064	67.6537	101.9349	206.4370	686.7831	1026.2499	1134.6490	926.8146	627.1411	1031.4741	778.0914	658.4299	837.7871	1335.3077	1411.9654
min	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25%	0.0000	0.0000	0.2000	0.2150	0.7150	1.2300	4.1600	9.9500	11.9500	7.8800	8.4300	22.7200	15.1150	16.6100	33.8000	30.0000	19.9050
50%	4.0300	5.0700	11.0100	6.3700	9.0900	22.6200	25.8200	58.8200	84.8800	69.7400	58.0700	129.3600	95.4100	113.7800	177.2200	159.1300	110.8600
75%	23.5100	44.8300	36.5550	38.6600	43.2050	63.8550	108.3250	279.2700	383.3200	341.5950	304.2800	593.5250	288.0250	473.0600	595.3900	519.0700	741.2200
max	832.0700	873.2300	419.9600	368.3200	527.9000	1359.9700	4713.7800	6986.1700	6183.4900	5466.1300	3296.0900	5215.9800	4832.9800	3982.8900	4443.2600	6889.4600	8684.0700

In [ ]:
```
df.isnull().sum()
```

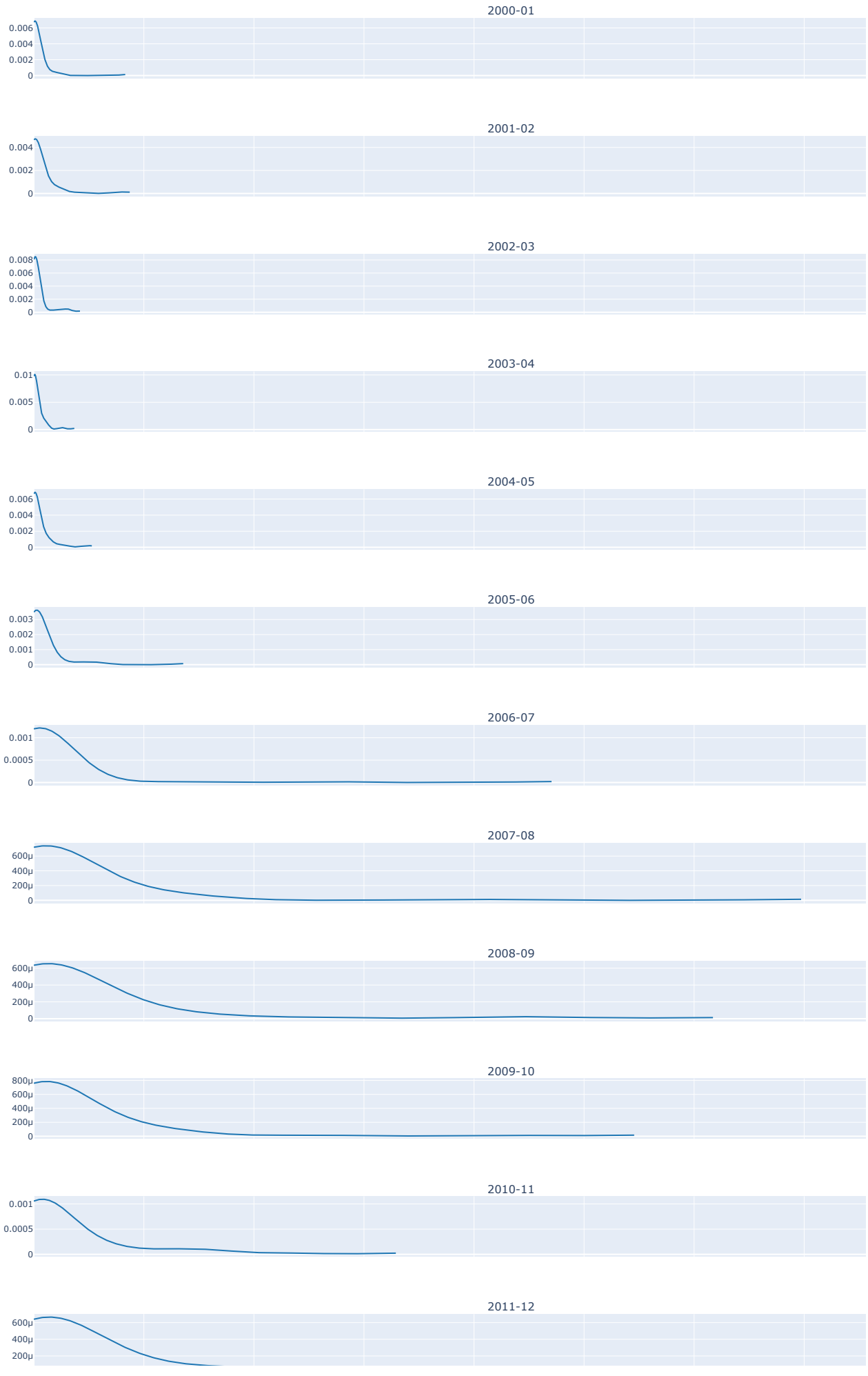
Out[ ]:
```
Sector     0
2000-01    0
2001-02    0
2002-03    0
2003-04    0
2004-05    0
2005-06    0
2006-07    0
2007-08    0
2008-09    0
2009-10    0
2010-11    0
2011-12    0
2012-13    0
2013-14    0
2014-15    0
2015-16    0
2016-17    0
dtype: int64
```

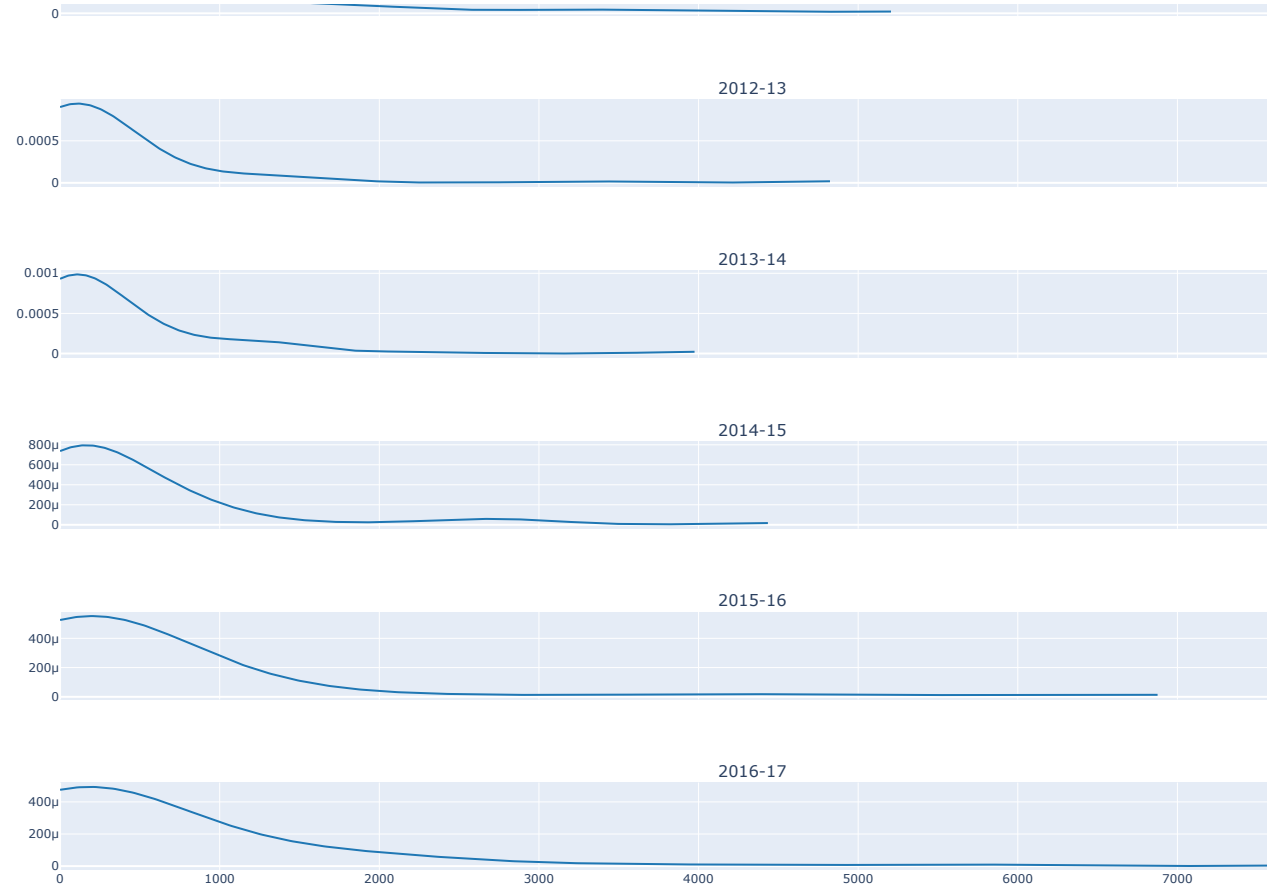
In [ ]:
```
df.duplicated().sum()
```

Out[ ]:
```
0
```

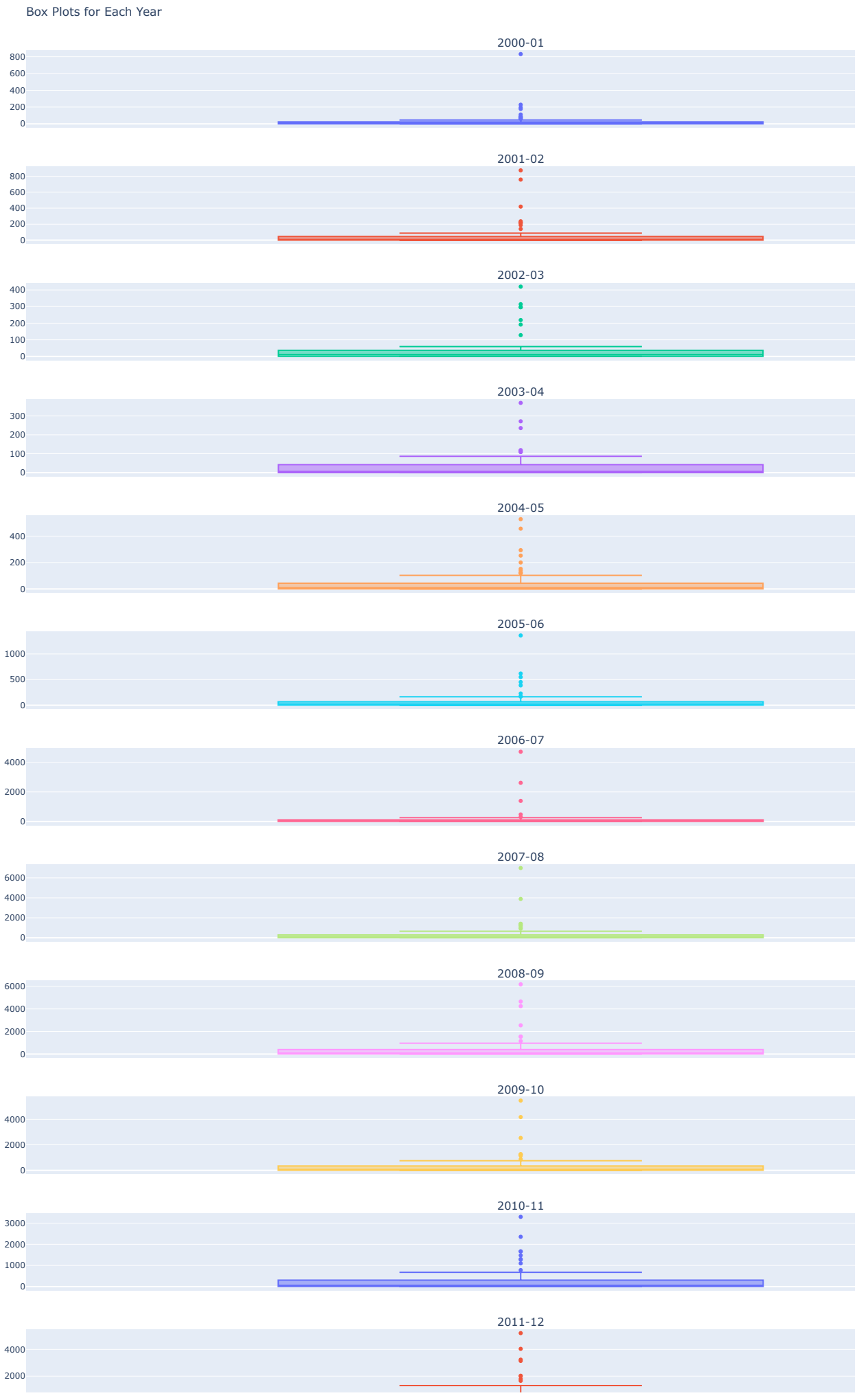
In [ ]:
```
# Create subplots
fig = sp.make_subplots(rows=17, cols=1, shared_xaxes=True, subplot_titles=df.columns[1:])

# Iterate over each float column to add KDE plots
for i, column in enumerate(df.columns[1:], start=1):
    # Extract the data for the current column
    data = df[column].dropna()

    # Create KDE plot
    kde = ff.create_distplot([data], [column], show_hist=False, show_rug=False)

    # Add KDE plot to subplot
    for trace in kde.data:
        fig.add_trace(trace, row=i, col=1)

# Update layout
fig.update_layout(height=3000, width=1600, title_text="KDE Plots for Each Year")

# Show the plot
fig.show()
```

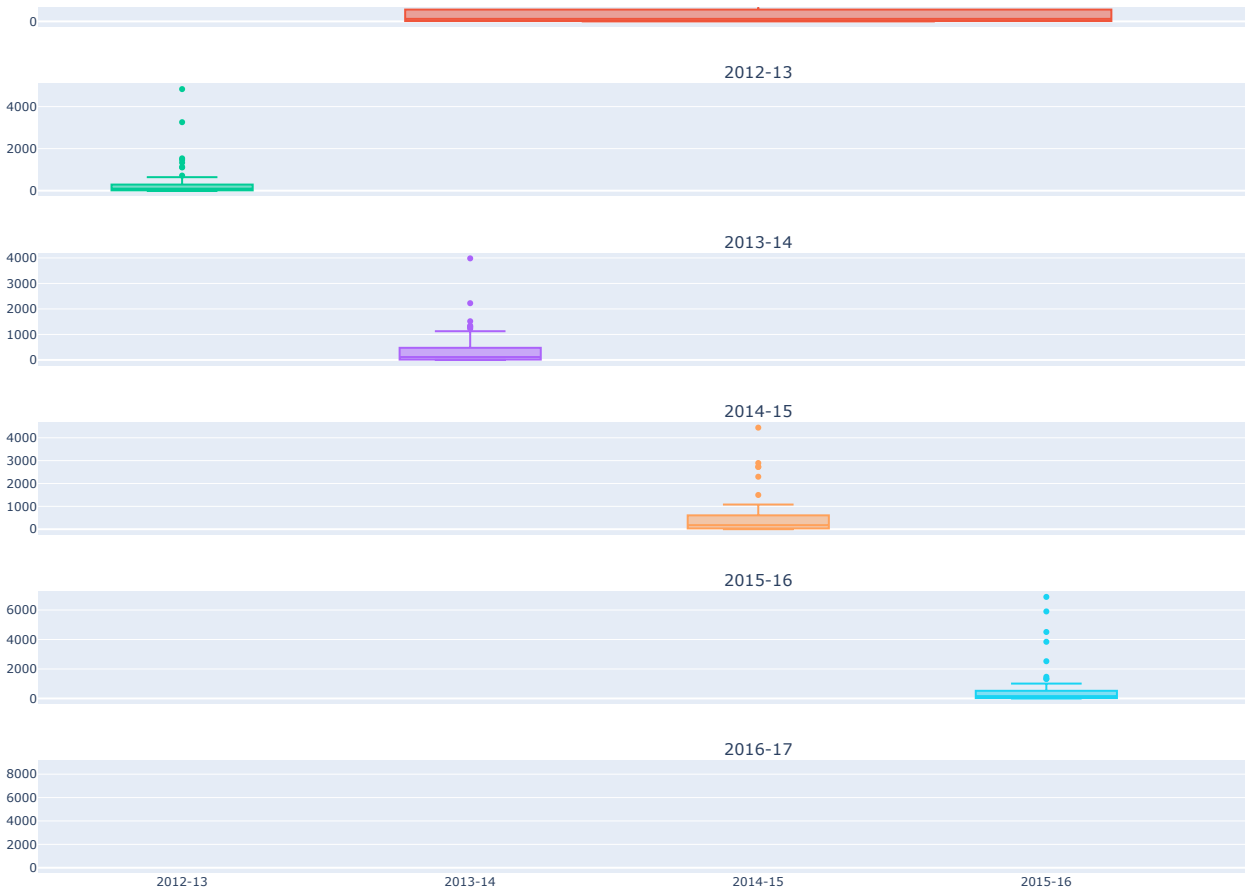
KDE Plots for Each Year





In [ ]:
```
fig = sp.make_subplots(rows=17, cols=1, shared_xaxes=True, subplot_titles=df.columns[1:], vertical_spacing=0.02)

# Iterate over each float column to add box plots
for i, column in enumerate(df.columns[1:], start=1):
    # Extract the data for the current column
    data = df[column].dropna()

    # Add box plot to subplot
    fig.add_trace(go.Box(y=data, name=column), row=i, col=1)

# Update layout
fig.update_layout(height=3000, width=1600, title_text="Box Plots for Each Year", showlegend=False)

# Show the plot
fig.show()
```


Box Plots for Each Year
2000-01
800
600
400
200
0
2001-02
800
600
400
200
0
2002-03
400
300
200
100
0
2003-04
300
200
100
0
2004-05
400
200
0
2005-06
1000
500
0
2006-07
4000
2000
0
2007-08
6000
4000
2000
0
2008-09
6000
4000
2000
0
2009-10
4000
2000
0
2010-11
3000
2000
1000
0
2011-12
4000
2000



In [ ]:
```
#Creating a function to Convert FDI's value from USD to INR
def multiply_columns(df, col_list, num):
    for col in col_list:
        df[col] = df[col] * Rates[col_list.index(col)]/10
    return df
```

In [ ]:
```
Year = ['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', '2016-17']
Sectors = ['Sector']
```

In [ ]:
```
Rates = [45.68,47.69,48.4,45.95,44.93,44.27,45.24,40.26,46,47.44,45.56,48,54.41,60.50,61.14,65.46,67.1]
```

In [ ]:
```
store_df=df.copy()
df2 = multiply_columns(df, Year, Rates)
```

In [ ]:
```
melt = pd.melt(store_df, id_vars = Sectors, value_vars = Year, var_name='Year',
    value_name='FDI(US$ Million)',ignore_index=True)
melt
```

Out[ ]:

	Sector	Year	FDI(US$ Million)
0	METALLURGICAL INDUSTRIES	2000-01	22.6900
1	MINING	2000-01	1.3200
2	POWER	2000-01	89.4200
3	NON-CONVENTIONAL ENERGY	2000-01	0.0000
4	COAL PRODUCTION	2000-01	0.0000
...	...	...	...
1066	PRINTING OF BOOKS (INCLUDING LITHO PRINTING IN...	2016-17	53.1700
1067	COIR	2016-17	0.0000
1068	CONSTRUCTION (INFRASTRUCTURE) ACTIVITIES	2016-17	1860.7300
1069	CONSTRUCTION DEVELOPMENT: Townships, housing, ...	2016-17	105.1400
1070	MISCELLANEOUS INDUSTRIES	2016-17	296.4000

1071 rows × 3 columns

In [ ]:
```
melt01 = pd.melt(df2, id_vars = Sectors, value_vars = Year, var_name='Year',
    value_name='FDI(₹ Crores)',ignore_index=True)
melt01=round(melt01,2)
melt01
```

Out[ ]:

	Sector	Year	FDI(₹ Crores)
0	METALLURGICAL INDUSTRIES	2000-01	103.6500
1	MINING	2000-01	6.0300
2	POWER	2000-01	408.4700
3	NON-CONVENTIONAL ENERGY	2000-01	0.0000
4	COAL PRODUCTION	2000-01	0.0000
...	...	...	...
1066	PRINTING OF BOOKS (INCLUDING LITHO PRINTING IN...	2016-17	356.7700
1067	COIR	2016-17	0.0000
1068	CONSTRUCTION (INFRASTRUCTURE) ACTIVITIES	2016-17	12485.5000
1069	CONSTRUCTION DEVELOPMENT: Townships, housing, ...	2016-17	705.4900
1070	MISCELLANEOUS INDUSTRIES	2016-17	1988.8400

1071 rows × 3 columns

In [ ]:
```
# Merging the FDI(USD Million) column of melt Dataframe into melto1 Dataframe
Merged=melt01.merge(melt,how='left')
Merged
```

Out[ ]:

	Sector	Year	FDI(₹ Crores)	FDI(US$ Million)
0	METALLURGICAL INDUSTRIES	2000-01	103.6500	22.6900
1	MINING	2000-01	6.0300	1.3200
2	POWER	2000-01	408.4700	89.4200
3	NON-CONVENTIONAL ENERGY	2000-01	0.0000	0.0000
4	COAL PRODUCTION	2000-01	0.0000	0.0000
...	...	...	...	...
1066	PRINTING OF BOOKS (INCLUDING LITHO PRINTING IN...	2016-17	356.7700	53.1700
1067	COIR	2016-17	0.0000	0.0000
1068	CONSTRUCTION (INFRASTRUCTURE) ACTIVITIES	2016-17	12485.5000	1860.7300
1069	CONSTRUCTION DEVELOPMENT: Townships, housing, ...	2016-17	705.4900	105.1400
1070	MISCELLANEOUS INDUSTRIES	2016-17	1988.8400	296.4000

1071 rows × 4 columns

In [ ]:
```
Sorted = Merged.sort_values(['Sector','Year'], ignore_index=True)
Sorted
```

Out[ ]:

	Sector	Year	FDI(₹ Crores)	FDI(US$ Million)
0	AGRICULTURAL MACHINERY	2000-01	16.6300	3.6400
1	AGRICULTURAL MACHINERY	2001-02	4.9600	1.0400
2	AGRICULTURAL MACHINERY	2002-03	65.2400	13.4800
3	AGRICULTURAL MACHINERY	2003-04	218.4500	47.5400
4	AGRICULTURAL MACHINERY	2004-05	0.0000	0.0000
...	...	...	...	...
1066	VEGETABLE OILS AND VANASPATI	2012-13	589.7500	108.3900
1067	VEGETABLE OILS AND VANASPATI	2013-14	130.3800	21.5500
1068	VEGETABLE OILS AND VANASPATI	2014-15	906.9500	148.3400
1069	VEGETABLE OILS AND VANASPATI	2015-16	224.0000	34.2200
1070	VEGETABLE OILS AND VANASPATI	2016-17	727.7000	108.4500

1071 rows × 4 columns

In [ ]:
```
#Repalcing some Long values of Sector Column to Short form
Sorted = Sorted[['Sector','FDI(₹ Crores)', 'FDI(US$ Million)'
                 ,]].replace(["CONSTRUCTION DEVELOPMENT: Townships, housing, built-up infrastructure and construction-development projects"
                              ,"SERVICES SECTOR (Fin.,Banking,Insurance,Non Fin/Business,Outsourcing,R&D,Courier,Tech. Testing and Analysis, Other)"
                              ,'TEA AND COFFEE (PROCESSING & WAREHOUSING COFFEE & RUBBER)']
                             ,["CONSTRUCTION DEVELOPMENT","SERVICES SECTOR",'TEA AND COFFEE'])
```

In [ ]:
```
#Grouping by Sector column to find Total FDI Inflow per Sector from FY2000-01 to FY2016-17
Sectorwise_fdi = Sorted.groupby('Sector').sum()
Sectorwise_fdi.sort_values(by='FDI(US$ Million)',ascending=False)
```

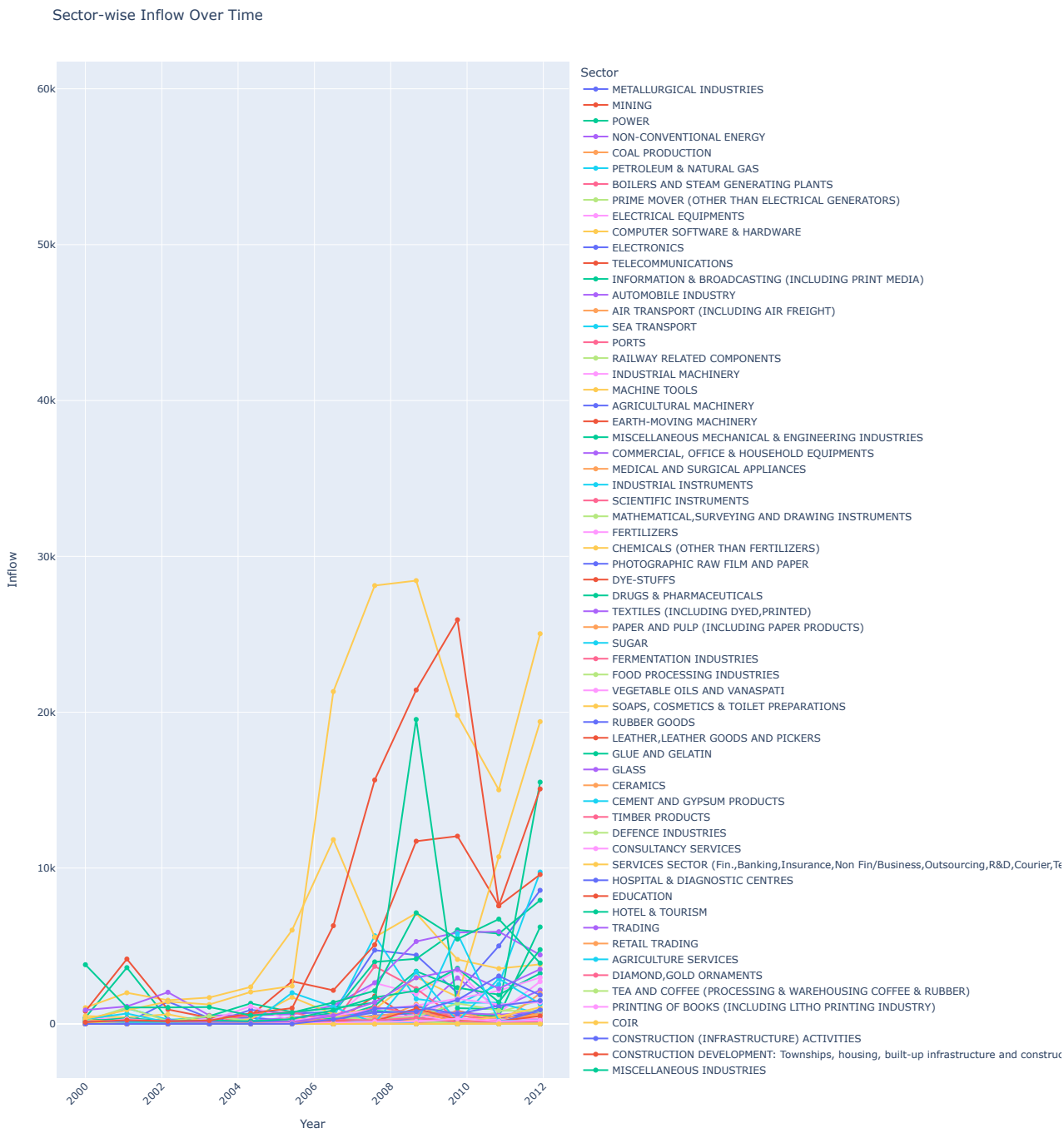
Out[ ]:

Sector	FDI(₹ Crores)	FDI(US$ Million)
SERVICES SECTOR	316426.6700	59476.4900
COMPUTER SOFTWARE & HARDWARE	137296.4800	24669.4900
CONSTRUCTION DEVELOPMENT	115217.4300	24293.0900
TELECOMMUNICATIONS	131001.7600	23946.0100
AUTOMOBILE INDUSTRY	92637.9800	16673.9200
...	...	...
PHOTOGRAPHIC RAW FILM AND PAPER	278.3700	67.2800
COAL PRODUCTION	122.1300	27.7400
MATHEMATICAL,SURVEYING AND DRAWING INSTRUMENTS	41.6200	7.9800
DEFENCE INDUSTRIES	26.1200	5.1200
COIR	21.6400	4.0600

63 rows × 2 columns

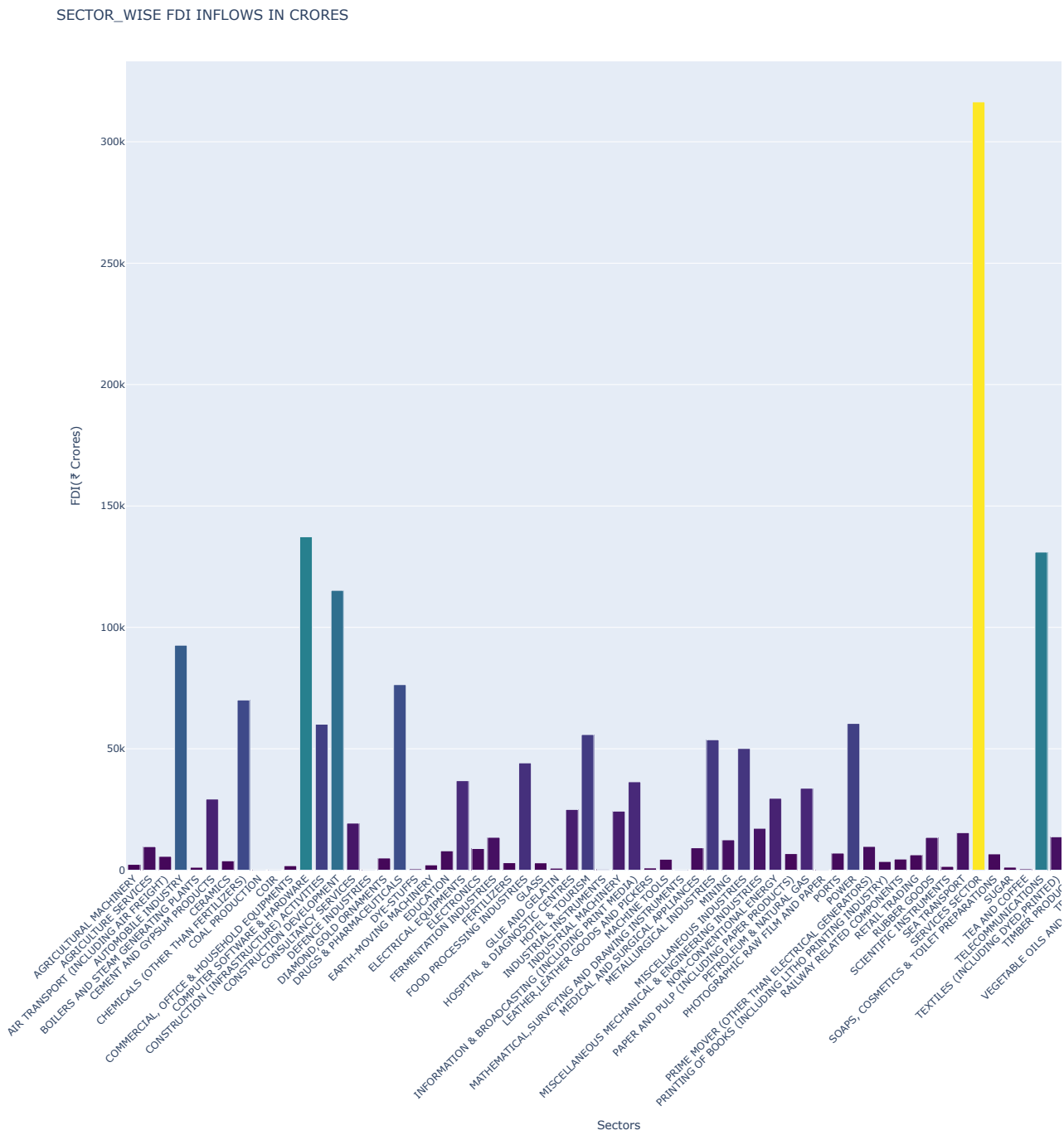
In [ ]:
```
df_long = df.melt(id_vars='Sector', var_name='Year', value_name='Inflow')
```

In [ ]:
```
fig = go.Figure()

# Add a trace for each sector
for sector in df['Sector'].unique():
    sector_data = df_long[df_long['Sector'] == sector]
    fig.add_trace(go.Scatter(
        x=sector_data['Year'],
        y=sector_data['Inflow'],
        mode='lines+markers',
        name=sector
    ))

# Update layout
fig.update_layout(
    title='Sector-wise Inflow Over Time',
    xaxis_title='Year',
    yaxis_title='Inflow',
    legend_title='Sector',
    xaxis=dict(tickangle=-45),
    width=1500,  # Width of the figure
    height=1400,
)

# Show the figure
fig.show()
```



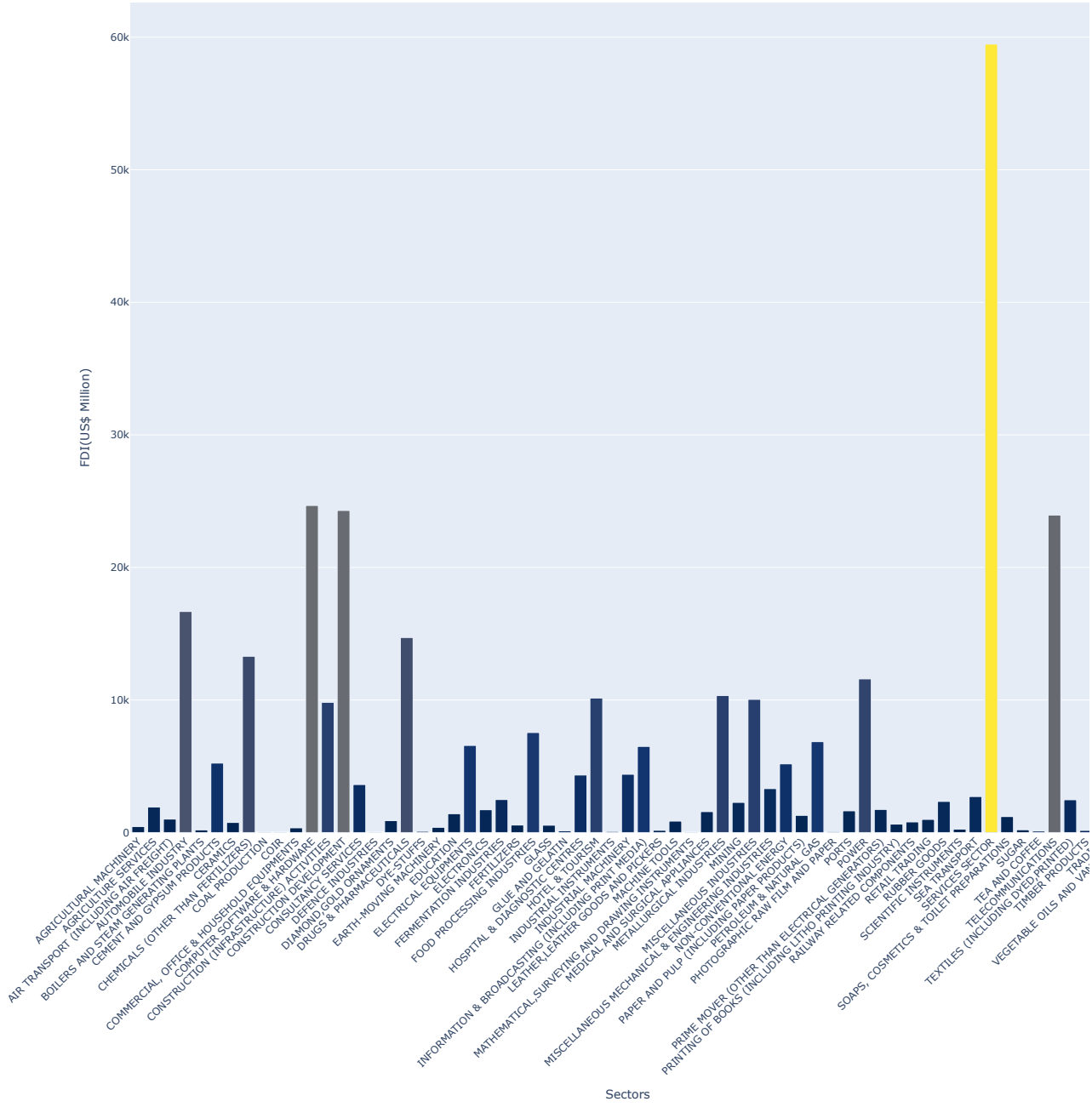
In [ ]:
```
fig = px.bar(Sectorwise_fdi,
             x=Sectorwise_fdi.index,
             y='FDI(₹ Crores)',
             title='SECTOR_WISE FDI INFLOWS IN CRORES',
             labels={'FDI(₹ Crores)': 'FDI(₹ Crores)'},
             color='FDI(₹ Crores)',
             color_continuous_scale='Viridis')  # Change the color scale here
```

```
fig.update_layout(xaxis_title='Sectors',
                  yaxis_title='FDI(₹ Crores)',
                  width=1500,  # Width of the figure
    height=1400,
                  xaxis_tickangle=-45)  # Adjust tick angle for better readability

fig.show()
```



In [ ]:
```
fig = px.bar(Sectorwise_fdi,
             x=Sectorwise_fdi.index,
             y='FDI(US$ Million)',
             title='SECTOR_WISE FDI INFLOWS IN MILLIONS',
             labels={'FDI(US$ Million)': 'FDI(US$ Million)'},
             color='FDI(US$ Million)',
             color_continuous_scale='Cividis')  # You can choose a color scale you prefer

fig.update_layout(xaxis_title='Sectors',
                  yaxis_title='FDI(US$ Million)',
                  xaxis_tickangle=-45,
                  width=1500,  # Width of the figure
    height=1400,)  # Adjust height if needed
fig.show()
```

SECTOR_WISE FDI INFLOWS IN MILLIONS



In [ ]:
```
#Top 10 and bottom 10 sectors
Top_10_Sectors = Sectorwise_fdi.nlargest(10,['FDI(₹ Crores)'])
```

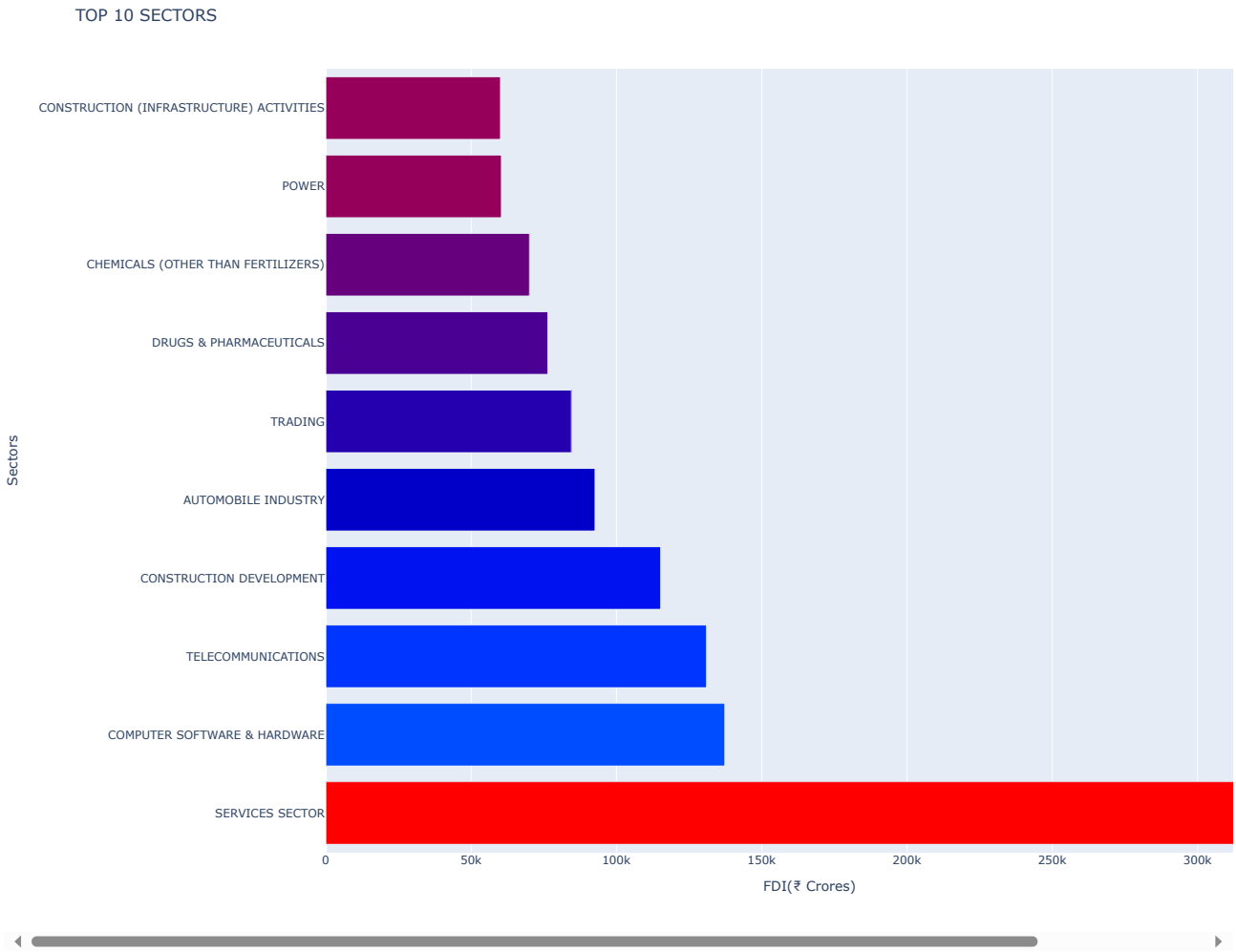
In [ ]:
```
#Calculating percentage-wise FDI share among top 10 sectors and among all sectors
Total_fdi = round(melt01['FDI(₹ Crores)'].sum(),2)
Sum = Top_10_Sectors['FDI(₹ Crores)'].sum()
Top_10_Sectors['In %age'] = round(Top_10_Sectors['FDI(₹ Crores)']/Sum*100,2)
Top_10_Sectors['%age to Total Inflows'] = round((Top_10_Sectors['FDI(₹ Crores)']/Total_fdi)*100,2)
Top_10_Sectors
```

Out[ ]:

Sector	FDI(₹ Crores)	FDI(US$ Million)	In %age	%age to Total Inflows
SERVICES SECTOR	316426.6700	59476.4900	27.6600	17.6500
COMPUTER SOFTWARE & HARDWARE	137296.4800	24669.4900	12.0000	7.6600
TELECOMMUNICATIONS	131001.7600	23946.0100	11.4500	7.3100
CONSTRUCTION DEVELOPMENT	115217.4300	24293.0900	10.0700	6.4300
AUTOMOBILE INDUSTRY	92637.9800	16673.9200	8.1000	5.1700
TRADING	84502.0900	14210.8800	7.3900	4.7100
DRUGS & PHARMACEUTICALS	76411.4900	14706.9000	6.6800	4.2600
CHEMICALS (OTHER THAN FERTILIZERS)	70068.7600	13293.0900	6.1200	3.9100
POWER	60416.0300	11589.1300	5.2800	3.3700
CONSTRUCTION (INFRASTRUCTURE) ACTIVITIES	60108.7200	9817.4700	5.2500	3.3500

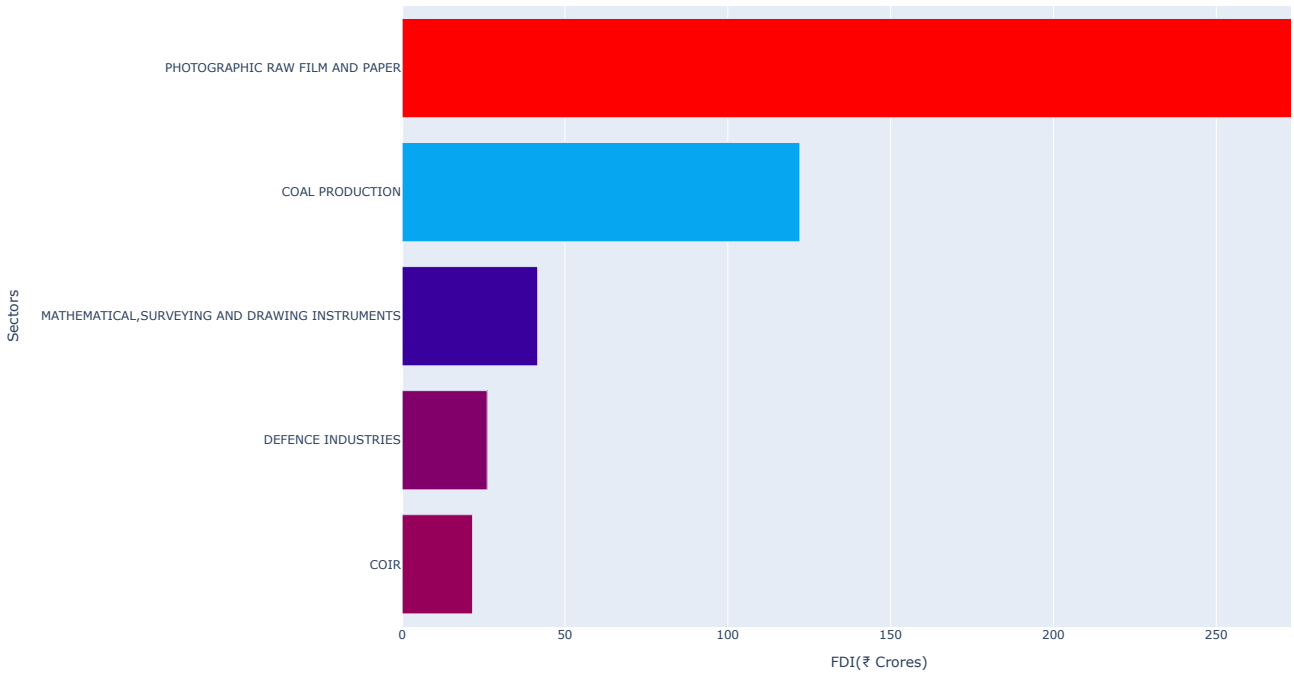
In [ ]:
```
fig = px.bar(Top_10_Sectors,
             x='FDI(₹ Crores)',
             y=Top_10_Sectors.index,
```

```
            orientation='h',  # Horizontal bar chart
            title='TOP 10 SECTORS',
            labels={'FDI(₹ Crores)': 'FDI(₹ Crores)'},
            color='FDI(₹ Crores)',
            color_continuous_scale='Rainbow')  # Choose a colorful scale

fig.update_layout(xaxis_title='FDI(₹ Crores)',
                  yaxis_title='Sectors',
                  yaxis=dict(tickmode='linear'),
                  width=1500,  # Width of the figure
    height=1000,)  # Ensure all y-axis labels are visible
fig.show()
```



In [ ]:
```
fig = px.bar(Bottom_5_Sectors,
            x='FDI(₹ Crores)',
            y=Bottom_5_Sectors.index,
            orientation='h',  # Horizontal bar chart
            title='BOTTOM 5 SECTORS',
            labels={'FDI(₹ Crores)': 'FDI(₹ Crores)'},
            color='FDI(₹ Crores)',
            color_continuous_scale='Rainbow')  # Choose a colorful scale

fig.update_layout(xaxis_title='FDI(₹ Crores)',
                  yaxis_title='Sectors',
                  yaxis=dict(tickmode='linear'),
                  width=1500,  # Width of the figure
    height=800,)  # Ensure all y-axis labels are visible
fig.show()
```

BOTTOM 5 SECTORS



In [ ]:
```
#Creating Dataframe
melt02 = melt01[['Year', 'FDI(₹ Crores)']]
melt02=round(melt02.groupby('Year').sum(),2)
```

In [ ]:
```
#reating new column of % growth over previous year
melt02['% growth over previous year'] = round(melt02.pct_change()*100,2)
```

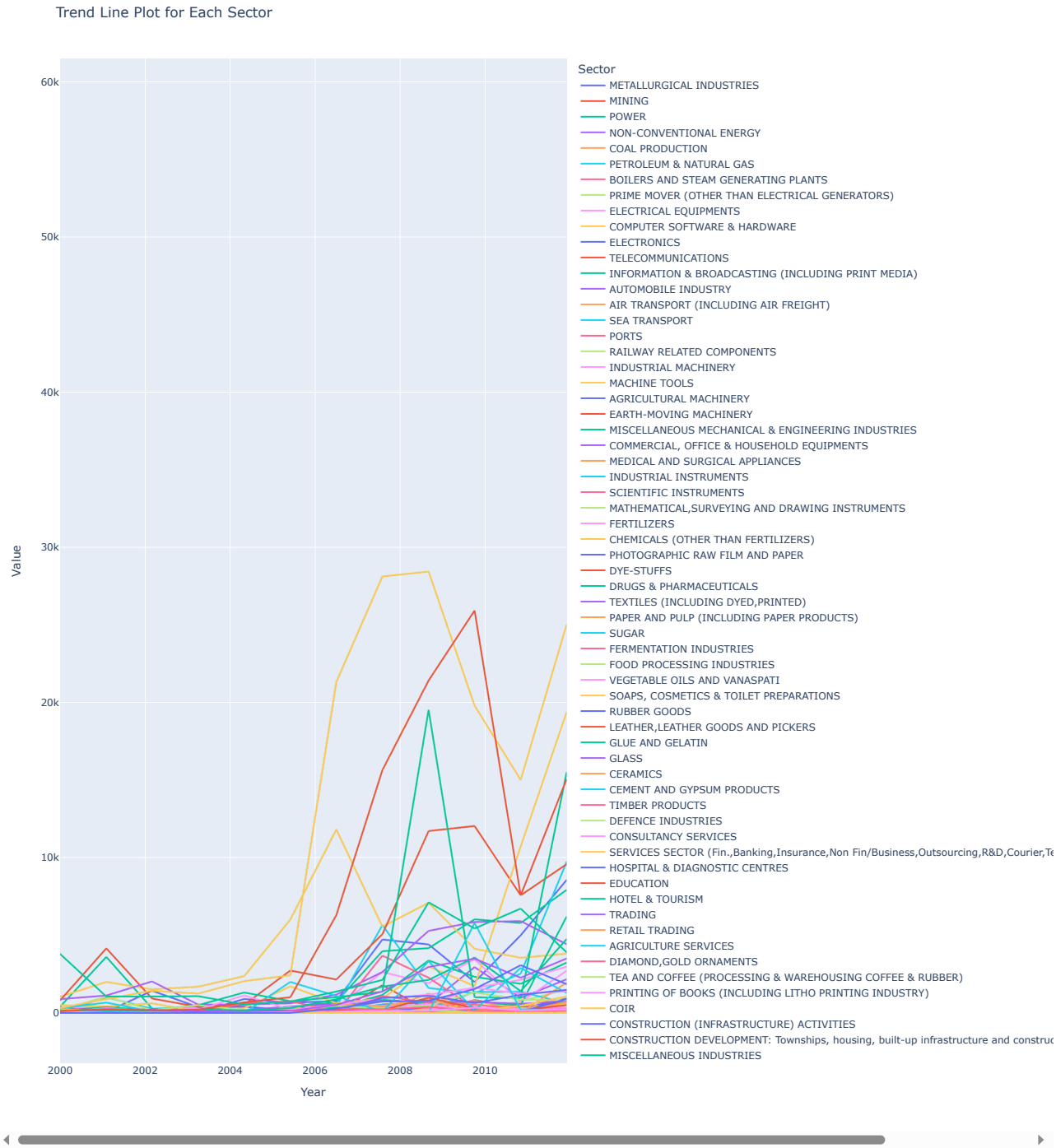
In [ ]:
```
print('\n'+"*"*8+"Details on Variation of FDI INFLOW Year-wise"+'*'*8) #Year-wise FDI Inflow
melt02.fillna('-')
```

********Details on Variation of FDI INFLOW Year-wise********

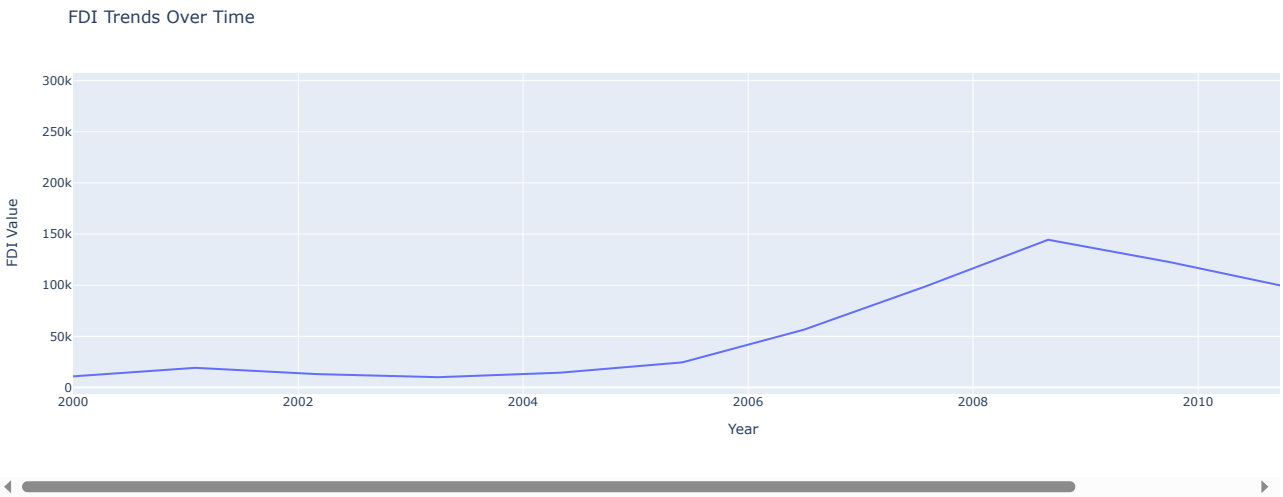
Out[ ]:

Year	FDI(₹ Crores)	% growth over previous year
2000-01	10865.9700	-
2001-02	19208.0200	76.7700
2002-03	13088.9100	-31.8600
2003-04	10053.1500	-23.1900
2004-05	14461.5900	43.8500
2005-06	24524.5100	69.5800
2006-07	56512.7400	130.4300
2007-08	98940.5700	75.0800
2008-09	144421.4400	45.9700
2009-10	122558.2700	-15.1400
2010-11	97421.2900	-20.5100
2011-12	168579.7400	73.0400
2012-13	122006.7700	-27.6300
2013-14	147010.9000	20.4900
2014-15	189108.8800	28.6400
2015-16	261846.4500	38.4600
2016-17	291739.0900	11.4200

In [ ]:
```
df_melted = df.melt(id_vars=['Sector'], var_name='Year', value_name='Value')

# Create the line plot using Plotly Express
fig = px.line(df_melted, x='Year', y='Value', color='Sector', title='Trend Line Plot for Each Sector',width=1500,  # Width of the figure
    height=1400,)

# Show the plot
fig.show()
```



In [ ]:
```
df_melted = df.melt(id_vars=['Sector'], var_name='Year', value_name='Value')

# Aggregate the data by year
df_aggregated = df_melted.groupby('Year').sum().reset_index()

# Create a line plot using Plotly Express
fig = px.line(df_aggregated, x='Year', y='Value', title='FDI Trends Over Time', labels={'Value': 'FDI Value'},width=1500,  # Width of the figure
    height=500,)

# Show the plot
fig.show()
```



Findings and Insights

Annual Trends:

- **Highest Annual FDI Inflow:** ₹291,608.67 Cr (2016-17)
- **Lowest Annual FDI Inflow:** ₹10,865.97 Cr (2000-01)

Sectoral Distribution:

- **Largest Sector by FDI:** Service Sector (17% of total FDI)
- **Least Attractive Sectors:** Coir, Defence Industries, Coal Production, and others with minimal investment.

Fluctuation Insights:

- Significant increase observed in 2007-08 with ₹98,940.57 Cr.
- Sharp decline in 2009-10 and 2010-11, with figures of ₹1,22,558.27 Cr and ₹97,421.29 Cr respectively.

Growth and Decline Periods:

- Major growth observed from 2005-06 to 2008-09.
- Noticeable decline and stabilization in the subsequent years before a final peak in 2016-17.

Conclusion

The analysis highlights a robust growth trajectory for FDI inflows in India, with significant peaks and troughs over the 17-year period. The service sector remains the most attractive for foreign investors, reflecting its high profit potential. Despite fluctuations, the overall trend shows increasing confidence and investment in India. This project provides valuable insights into the FDI landscape and can aid in future economic and investment planning.

Future Work

Further research could involve:

- **Sector-Specific Analysis:** Deep dive into underperforming sectors to identify potential growth areas.
- **Impact Assessment:** Evaluating the economic impact of FDI on various sectors and the broader economy.
- **Predictive Modeling:** Using historical data to forecast future FDI trends and potential economic impacts.
