

EV

June 14, 2025

```
[1]: import math
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
import sklearn as sl
import warnings
warnings.filterwarnings('ignore')
from collections import Counter
```

```
[2]: data = pd.read_csv(r'C:\Users\banga\OneDrive\Desktop\Internship\DATA SETS\EV Sales.csv')
```

```
[3]: data.head()
```

```
[3]:      Year Month_Name     Date       State Vehicle_Class \
0  2014.0        jan  1/1/2014  Andhra Pradesh    ADAPTED VEHICLE
1  2014.0        jan  1/1/2014  Andhra Pradesh  AGRICULTURAL TRACTOR
2  2014.0        jan  1/1/2014  Andhra Pradesh        AMBULANCE
3  2014.0        jan  1/1/2014  Andhra Pradesh ARTICULATED VEHICLE
4  2014.0        jan  1/1/2014  Andhra Pradesh          BUS

   Vehicle_Category Vehicle_Type  EV_Sales_Quantity
0            Others      Others             0.0
1            Others      Others             0.0
2            Others      Others             0.0
3            Others      Others             0.0
4             Bus        Bus              0.0
```

```
[4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96845 entries, 0 to 96844
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   Year              96845 non-null  float64 
 1   Month_Name        96845 non-null  object 

```

```
2   Date          96845 non-null  object
3   State         96845 non-null  object
4   Vehicle_Class 96845 non-null  object
5   Vehicle_Category 96845 non-null  object
6   Vehicle_Type   96845 non-null  object
7   EV_Sales_Quantity 96845 non-null  float64
dtypes: float64(2), object(6)
memory usage: 5.9+ MB
```

```
[5]: data.describe()
```

```
[5]:      Year  EV_Sales_Quantity
count  96845.000000    96845.000000
mean    2018.622768     37.108896
std      2.895581     431.566675
min     2014.000000     0.000000
25%    2016.000000     0.000000
50%    2019.000000     0.000000
75%    2021.000000     0.000000
max    2024.000000    20584.000000
```

```
[6]: data['EV_Sales_Quantity'].describe()
```

```
[6]: count    96845.000000
mean      37.108896
std       431.566675
min      0.000000
25%     0.000000
50%     0.000000
75%     0.000000
max     20584.000000
Name: EV_Sales_Quantity, dtype: float64
```

```
[7]: data['Year'] = data['Year'].astype(int)
```

```
[8]: data['Year'].value_counts().sort_index()
```

```
[8]: Year
2014    9022
2015    9052
2016    9348
2017    9799
2018   10225
2019   10315
2020    8675
2021    9249
2022   10021
2023   10279
```

```
2024      860  
Name: count, dtype: int64
```

```
[9]: data['State'].value_counts()
```

```
[9]: State  
Maharashtra          4912  
Karnataka            4830  
Uttar Pradesh         4557  
Rajasthan             4552  
Gujarat               4517  
West Bengal            4196  
Tamil Nadu             4063  
Odisha                 4027  
Haryana                3842  
Kerala                  3666  
Chhattisgarh           3590  
Madhya Pradesh          3587  
Andhra Pradesh          3457  
Assam                   3114  
Uttarakhand             3045  
Himachal Pradesh        2980  
Punjab                  2950  
Jharkhand                2773  
Bihar                     2544  
Jammu and Kashmir       2292  
Arunachal Pradesh        2285  
Goa                        2139  
DNH and DD                1927  
Delhi                      1871  
Meghalaya                  1867  
Puducherry                1832  
Manipur                     1632  
Nagaland                     1588  
Tripura                      1564  
Mizoram                     1557  
Chandigarh                  1554  
Sikkim                      1246  
Andaman & Nicobar Island     1226  
Ladakh                       1063  
Name: count, dtype: int64
```

```
[10]: data['State'].describe()
```

```
[10]: count          96845  
unique            34  
top      Maharashtra
```

```
freq          4912  
Name: State, dtype: object
```

```
[11]: data['Vehicle_Class'].value_counts()
```

```
[11]: Vehicle_Class  
MOTOR CAR                  4111  
M-CYCLE/SCOOTER            4101  
GOODS CARRIER              4096  
MOTOR CAB                  3985  
BUS                         3813  
...  
SEMI-TRAILER (COMMERCIAL)  18  
X-RAY VAN                  12  
MOTOR CYCLE/SCOOTER-WITH TRAILER 9  
MODULAR HYDRAULIC TRAILER    3  
MOTOR CARAVAN              3  
Name: count, Length: 73, dtype: int64
```

```
[12]: data['Vehicle_Class'].describe()
```

```
[12]: count      96845  
unique       73  
top         MOTOR CAR  
freq        4111  
Name: Vehicle_Class, dtype: object
```

```
[13]: data['Vehicle_Category'].value_counts()
```

```
[13]: Vehicle_Category  
Others        54423  
2-Wheelers   13121  
3-Wheelers   11491  
Bus           9119  
4-Wheelers   8691  
Name: count, dtype: int64
```

```
[14]: data['Vehicle_Category'].describe()
```

```
[14]: count      96845  
unique       5  
top         Others  
freq        54423  
Name: Vehicle_Category, dtype: object
```

```
[15]: data['Vehicle_Type'].value_counts()
```

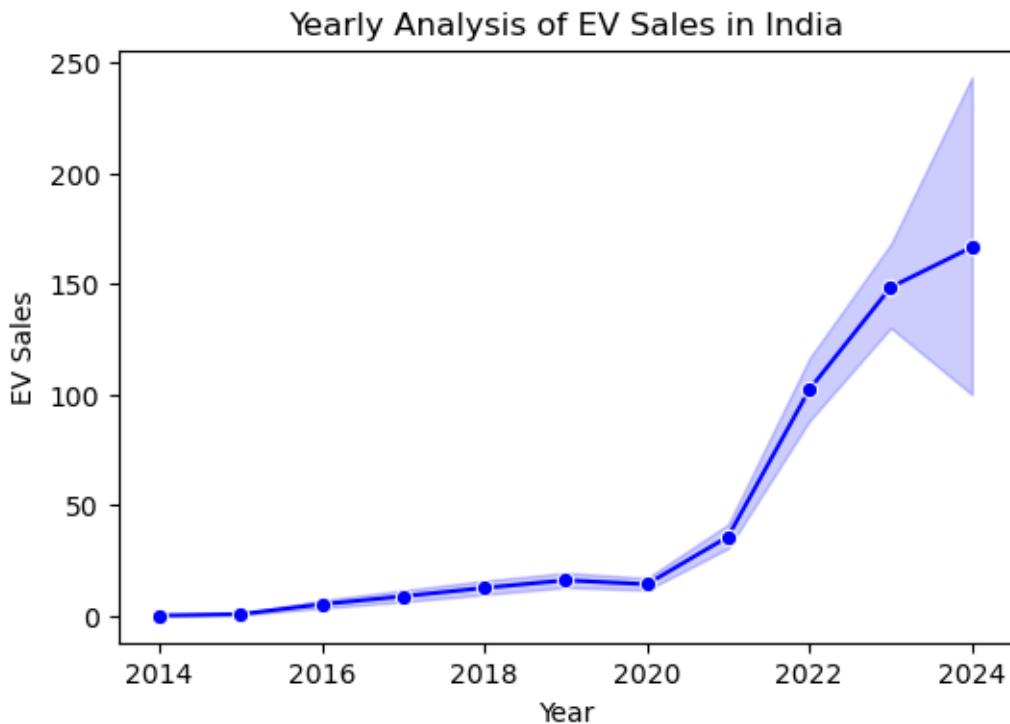
```
[15]: Vehicle_Type  
Others           54423  
2W_Personal     11700  
Bus              7026  
4W_Shared       4580  
4W_Personal     4111  
3W_Shared       3786  
3W_Goods        3208  
Institution Bus 2093  
3W_Shared_LowSpeed 1951  
3W_Goods_LowSpeed 1517  
2W_Shared       1421  
3W_Personal     1029  
Name: count, dtype: int64
```

```
[16]: data['Vehicle_Type'].describe()
```

```
[16]: count      96845  
unique       12  
top          Others  
freq         54423  
Name: Vehicle_Type, dtype: object
```

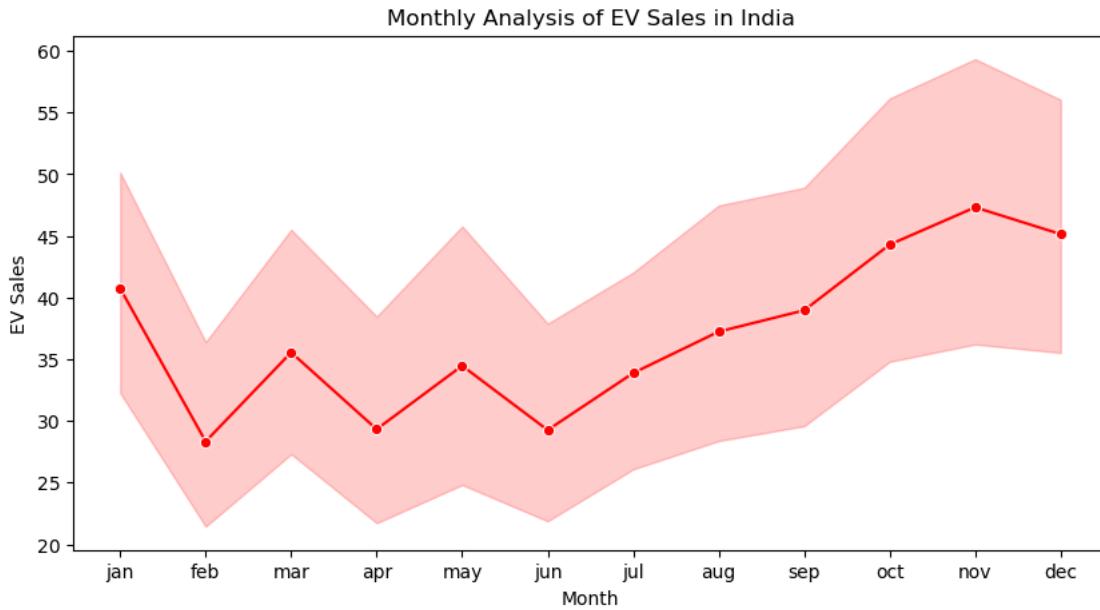
```
[17]: plt.figure(figsize=(6,4))  
plt.title('Yearly Analysis of EV Sales in India')  
sb.lineplot(x='Year', y='EV_Sales_Quantity', data=data, marker='o', color='b')  
plt.xlabel('Year')  
plt.ylabel('EV Sales')
```

```
[17]: Text(0, 0.5, 'EV\x00Sales')
```



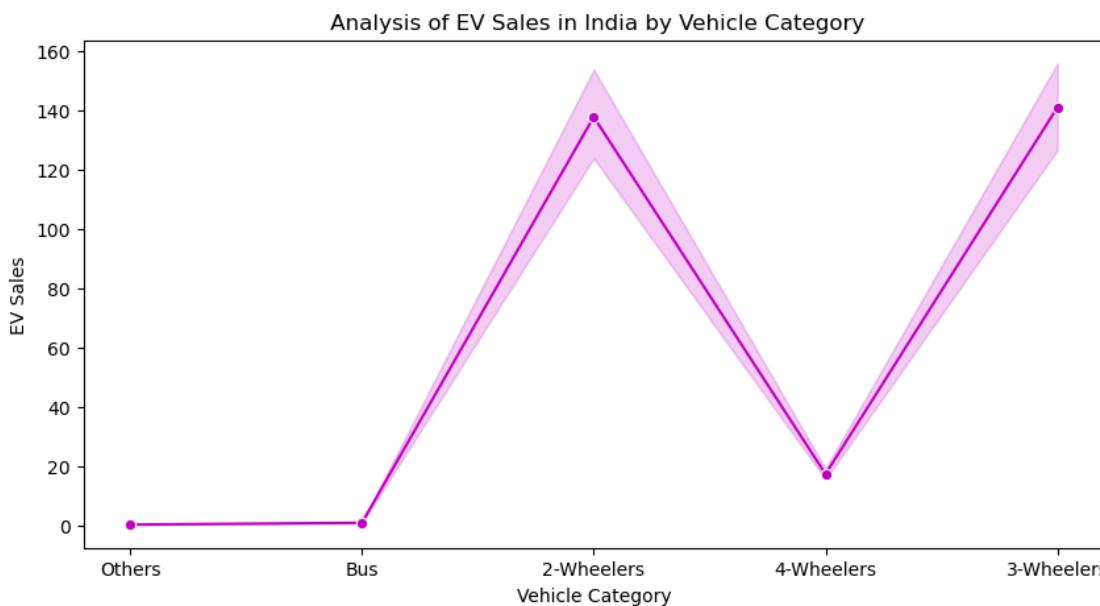
```
[18]: plt.figure(figsize=(10,5))
plt.title('Monthly Analysis of EV Sales in India')
sb.lineplot(x='Month_Name', y='EV_Sales_Quantity', data=data,marker='o', color='r')
plt.xlabel('Month')
plt.ylabel('EV Sales')
```

```
[18]: Text(0, 0.5, 'EV\x00Sales')
```



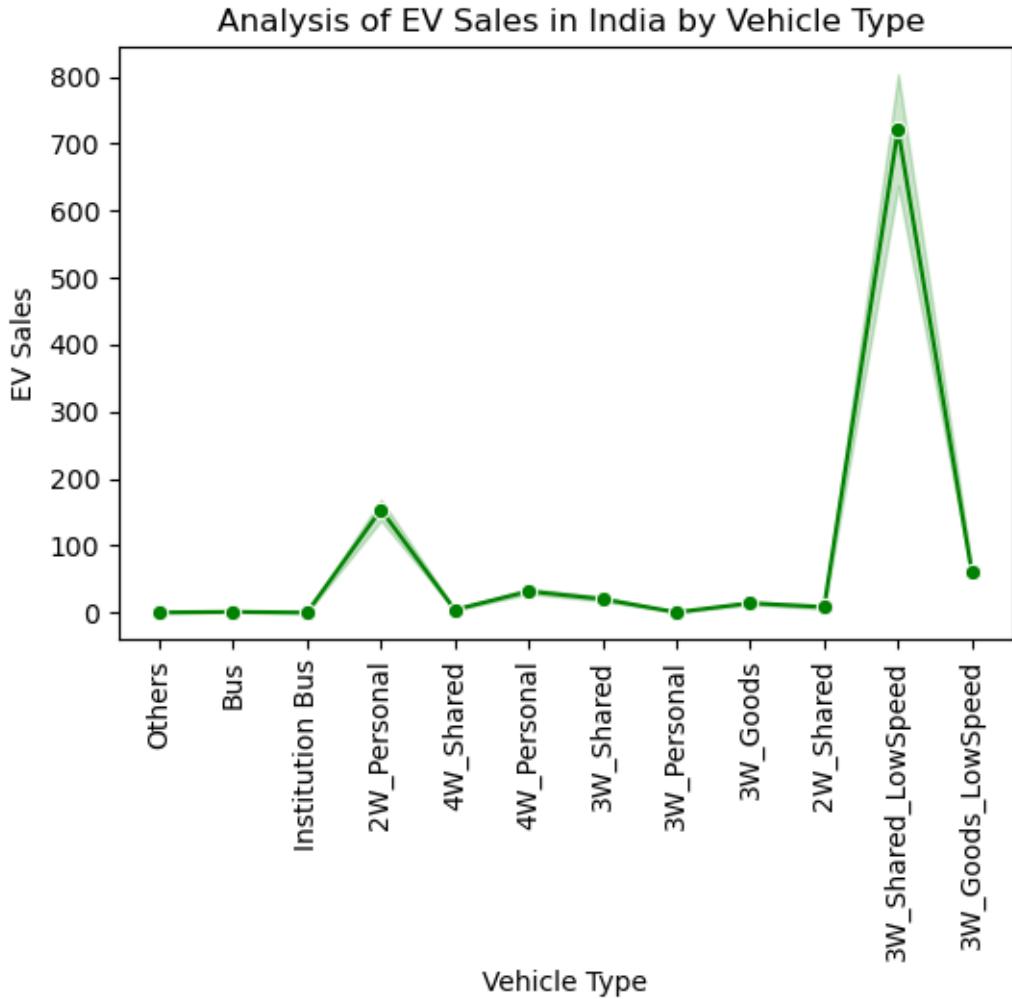
```
[19]: plt.figure(figsize=(10,5))
plt.title(' Analysis of EV Sales in India by Vehicle Category')
sb.lineplot(x='Vehicle_Category', y='EV_Sales_Quantity', data=data,marker='o', color='m')
plt.xlabel('Vehicle Category')
plt.ylabel('EV Sales')
```

[19]: Text(0, 0.5, 'EV\x00Sales')



```
[20]: plt.figure(figsize=(6,4))
plt.title(' Analysis of EV Sales in India by Vehicle Type')
sb.lineplot(x='Vehicle_Type',y='EV_Sales_Quantity',data=data,marker='o',color='g')
plt.xlabel('Vehicle Type')
plt.ylabel('EV Sales')
plt.xticks(rotation=90)
```

```
[20]: ([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11],  
[Text(0, 0, 'Others'),  
Text(1, 0, 'Bus'),  
Text(2, 0, 'Institution Bus'),  
Text(3, 0, '2W_Personal'),  
Text(4, 0, '4W_Shared'),  
Text(5, 0, '4W_Personal'),  
Text(6, 0, '3W_Shared'),  
Text(7, 0, '3W_Personal'),  
Text(8, 0, '3W_Goods'),  
Text(9, 0, '2W_Shared'),  
Text(10, 0, '3W_Shared_LowSpeed'),  
Text(11, 0, '3W_Goods_LowSpeed')])
```



```
[21]: plt.figure(figsize=(20,4))
plt.title(' Analysis of EV Sales in India by Vehicle Class')
sb.barplot(x='Vehicle_Class',  

           y='EV_Sales_Quantity',data=data,hue='Vehicle_Class')
plt.xlabel('Vehicle Class')
plt.ylabel('EV Sales')
plt.xticks(rotation=90)
```

```
[21]: ([0,  

        1,  

        2,  

        3,  

        4,  

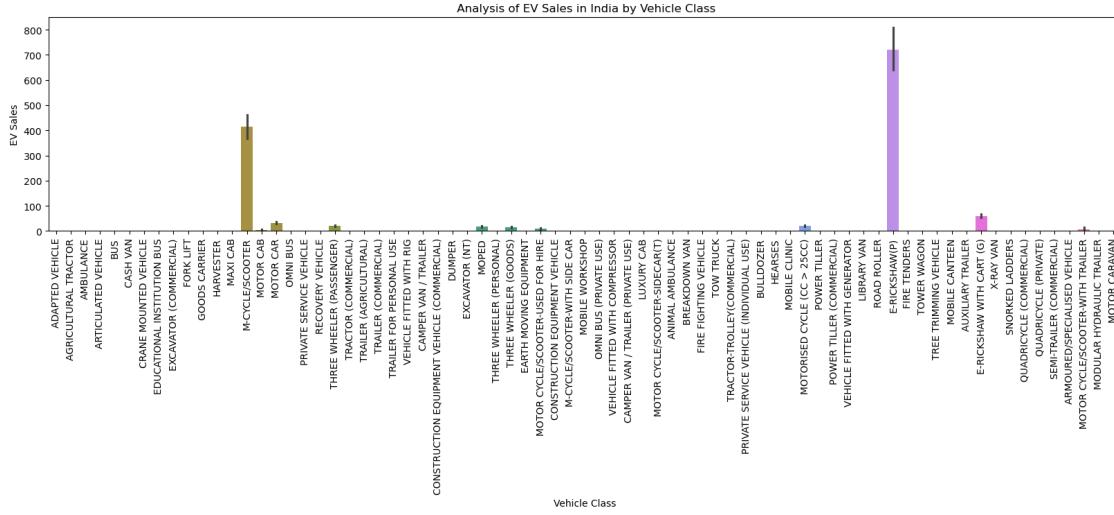
        5,  

        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,
32,
33,
34,
35,
36,
37,
38,
39,
40,
41,
42,
43,
44,
45,
46,
47,
48,
49,
50,
51,
52,
53,

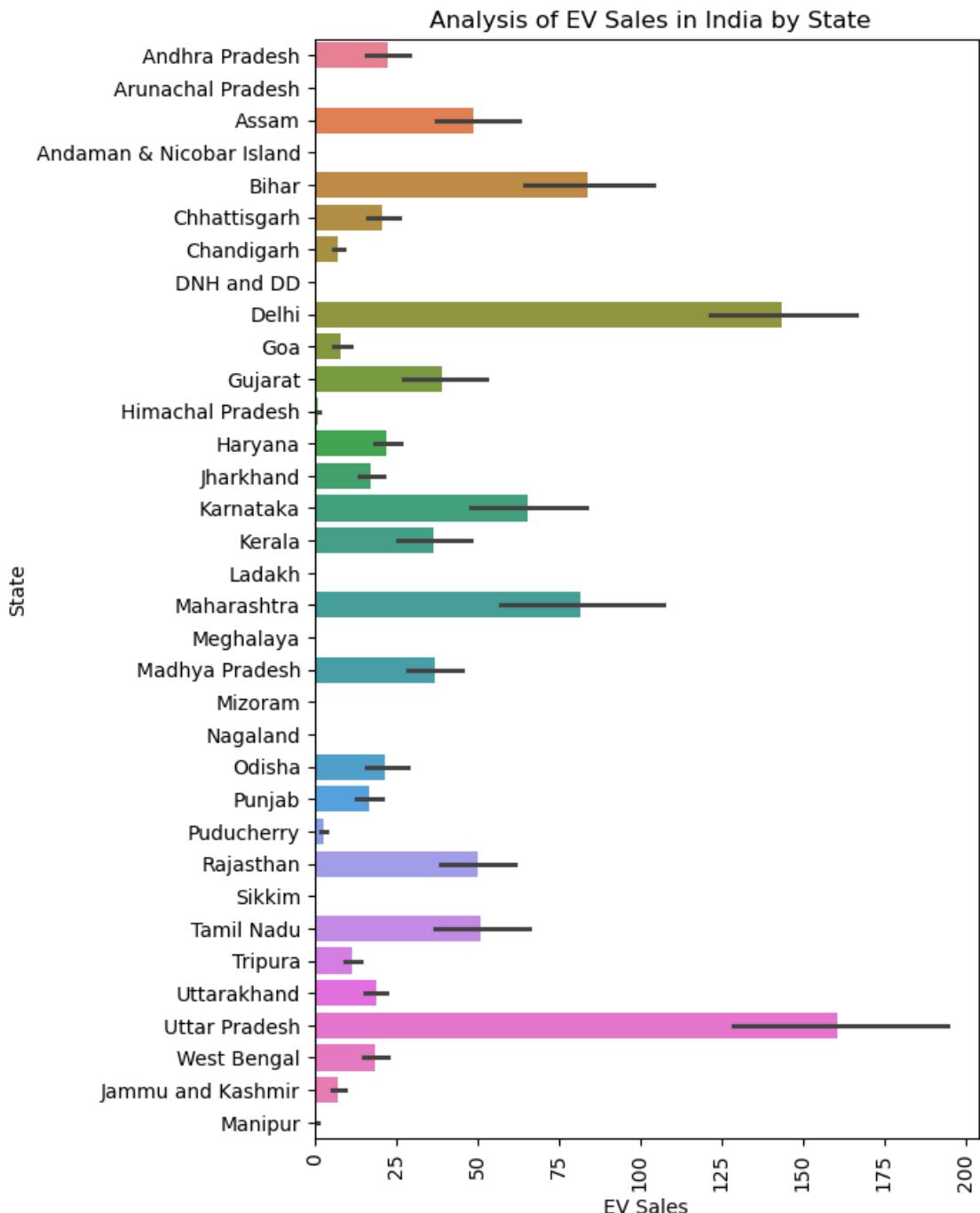
54,
55,
56,
57,
58,
59,
60,
61,
62,
63,
64,
65,
66,
67,
68,
69,
70,
71,
72],
[Text(0, 0, 'ADAPTED VEHICLE'),
Text(1, 0, 'AGRICULTURAL TRACTOR'),
Text(2, 0, 'AMBULANCE'),
Text(3, 0, 'ARTICULATED VEHICLE'),
Text(4, 0, 'BUS'),
Text(5, 0, 'CASH VAN'),
Text(6, 0, 'CRANE MOUNTED VEHICLE'),
Text(7, 0, 'EDUCATIONAL INSTITUTION BUS'),
Text(8, 0, 'EXCAVATOR (COMMERCIAL)'),
Text(9, 0, 'FORK LIFT'),
Text(10, 0, 'GOODS CARRIER'),
Text(11, 0, 'HARVESTER'),
Text(12, 0, 'MAXI CAB'),
Text(13, 0, 'M-CYCLE/SCOOTER'),
Text(14, 0, 'MOTOR CAB'),
Text(15, 0, 'MOTOR CAR'),
Text(16, 0, 'OMNI BUS'),
Text(17, 0, 'PRIVATE SERVICE VEHICLE'),
Text(18, 0, 'RECOVERY VEHICLE'),
Text(19, 0, 'THREE WHEELER (PASSENGER)'),
Text(20, 0, 'TRACTOR (COMMERCIAL)'),
Text(21, 0, 'TRAILER (AGRICULTURAL)'),
Text(22, 0, 'TRAILER (COMMERCIAL)'),
Text(23, 0, 'TRAILER FOR PERSONAL USE'),
Text(24, 0, 'VEHICLE FITTED WITH RIG'),
Text(25, 0, 'CAMPER VAN / TRAILER'),
Text(26, 0, 'CONSTRUCTION EQUIPMENT VEHICLE (COMMERCIAL)'),
Text(27, 0, 'DUMPER'),

Text(28, 0, 'EXCAVATOR (NT)'),
Text(29, 0, 'MOPED'),
Text(30, 0, 'THREE WHEELER (PERSONAL)'),
Text(31, 0, 'THREE WHEELER (GOODS)'),
Text(32, 0, 'EARTH MOVING EQUIPMENT'),
Text(33, 0, 'MOTOR CYCLE/SCOOTER-USED FOR HIRE'),
Text(34, 0, 'CONSTRUCTION EQUIPMENT VEHICLE'),
Text(35, 0, 'M-CYCLE/SCOOTER-WITH SIDE CAR'),
Text(36, 0, 'MOBILE WORKSHOP'),
Text(37, 0, 'OMNI BUS (PRIVATE USE)'),
Text(38, 0, 'VEHICLE FITTED WITH COMPRESSOR'),
Text(39, 0, 'CAMPER VAN / TRAILER (PRIVATE USE)'),
Text(40, 0, 'LUXURY CAB'),
Text(41, 0, 'MOTOR CYCLE/SCOOTER-SIDECAR(T)'),
Text(42, 0, 'ANIMAL AMBULANCE'),
Text(43, 0, 'BREAKDOWN VAN'),
Text(44, 0, 'FIRE FIGHTING VEHICLE'),
Text(45, 0, 'TOW TRUCK'),
Text(46, 0, 'TRACTOR-TROLLEY(COMMERCIAL)'),
Text(47, 0, 'PRIVATE SERVICE VEHICLE (INDIVIDUAL USE)'),
Text(48, 0, 'BULLDOZER'),
Text(49, 0, 'HEARSE'),
Text(50, 0, 'MOBILE CLINIC'),
Text(51, 0, 'MOTORIZED CYCLE (CC > 25CC)'),
Text(52, 0, 'POWER TILLER'),
Text(53, 0, 'POWER TILLER (COMMERCIAL)'),
Text(54, 0, 'VEHICLE FITTED WITH GENERATOR'),
Text(55, 0, 'LIBRARY VAN'),
Text(56, 0, 'ROAD ROLLER'),
Text(57, 0, 'E-RICKSHAW(P)'),
Text(58, 0, 'FIRE TENDERS'),
Text(59, 0, 'TOWER WAGON'),
Text(60, 0, 'TREE TRIMMING VEHICLE'),
Text(61, 0, 'MOBILE CANTEEN'),
Text(62, 0, 'AUXILIARY TRAILER'),
Text(63, 0, 'E-RICKSHAW WITH CART (G)'),
Text(64, 0, 'X-RAY VAN'),
Text(65, 0, 'SNORKED LADDERS'),
Text(66, 0, 'QUADRICYCLE (COMMERCIAL)'),
Text(67, 0, 'QUADRICYCLE (PRIVATE)'),
Text(68, 0, 'SEMI-TRAILER (COMMERCIAL)'),
Text(69, 0, 'ARMOURED/SPECIALISED VEHICLE'),
Text(70, 0, 'MOTOR CYCLE/SCOOTER-WITH TRAILER'),
Text(71, 0, 'MODULAR HYDRAULIC TRAILER'),
Text(72, 0, 'MOTOR CARAVAN'))



```
[22]: plt.figure(figsize=(6,10))
plt.title(' Analysis of EV Sales in India by State')
sb.barplot(x='EV_Sales_Quantity',y='State',data=data,hue='State')
plt.xlabel('EV Sales')
plt.ylabel('State')
plt.xticks(rotation=90)
```

```
[22]: (array([ 0.,  25.,  50.,  75., 100., 125., 150., 175., 200., 225.]),
[Text(0.0, 0, '0'),
Text(25.0, 0, '25'),
Text(50.0, 0, '50'),
Text(75.0, 0, '75'),
Text(100.0, 0, '100'),
Text(125.0, 0, '125'),
Text(150.0, 0, '150'),
Text(175.0, 0, '175'),
Text(200.0, 0, '200'),
Text(225.0, 0, '225')])
```

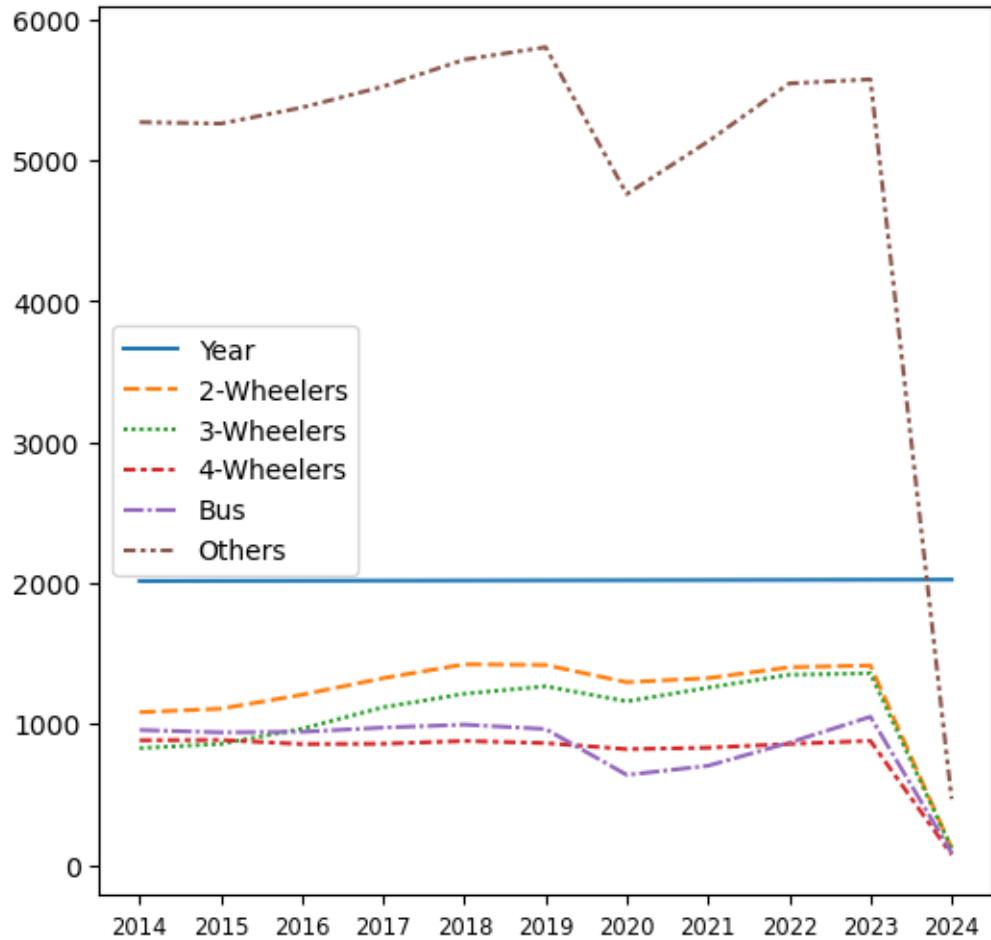


```
[23]: Category_Sales_Year = data.groupby(['Vehicle_Category', 'Year']).count()['Date'].
    ↪reset_index().rename(columns={'Date':'count'}).
    ↪pivot(index='Year', columns='Vehicle_Category', values='count').reset_index()
Category_Sales_Year
```

```
[23]: Vehicle_Category  Year  2-Wheelers  3-Wheelers  4-Wheelers  Bus  Others
      0                2014    1083        828        884    957    5270
      1                2015    1108        858        886    940    5260
      2                2016    1207        966        857    944    5374
      3                2017    1325       1118        859    974    5523
      4                2018    1423       1214        879    995    5714
      5                2019    1418       1267        864    964    5802
      6                2020    1296       1160        821    638    4760
      7                2021    1325       1257        831    703    5133
      8                2022    1402       1349        858    867    5545
      9                2023    1415       1359        880   1051    5574
     10               2024     119        115         72     86    468
```

```
[24]: plt.figure(figsize=(6,6))
sb.lineplot(data=Category_Sales_Year)
plt.legend(loc='upper left')
plt.legend(fontsize=10)
plt.
    xticks(range(len(Category_Sales_Year['Year'])),Category_Sales_Year['Year'],size='small')
```

```
[24]: ([<matplotlib.axis.XTick at 0x1e520883350>,
 <matplotlib.axis.XTick at 0x1e5209a10a0>,
 <matplotlib.axis.XTick at 0x1e51d3117c0>,
 <matplotlib.axis.XTick at 0x1e5208196a0>,
 <matplotlib.axis.XTick at 0x1e52081a060>,
 <matplotlib.axis.XTick at 0x1e52081a960>,
 <matplotlib.axis.XTick at 0x1e52081b2c0>,
 <matplotlib.axis.XTick at 0x1e52081a3f0>,
 <matplotlib.axis.XTick at 0x1e520882480>,
 <matplotlib.axis.XTick at 0x1e520f94530>,
 <matplotlib.axis.XTick at 0x1e520f94ec0>],
[Text(0, 0, '2014'),
 Text(1, 0, '2015'),
 Text(2, 0, '2016'),
 Text(3, 0, '2017'),
 Text(4, 0, '2018'),
 Text(5, 0, '2019'),
 Text(6, 0, '2020'),
 Text(7, 0, '2021'),
 Text(8, 0, '2022'),
 Text(9, 0, '2023'),
 Text(10, 0, '2024')])
```



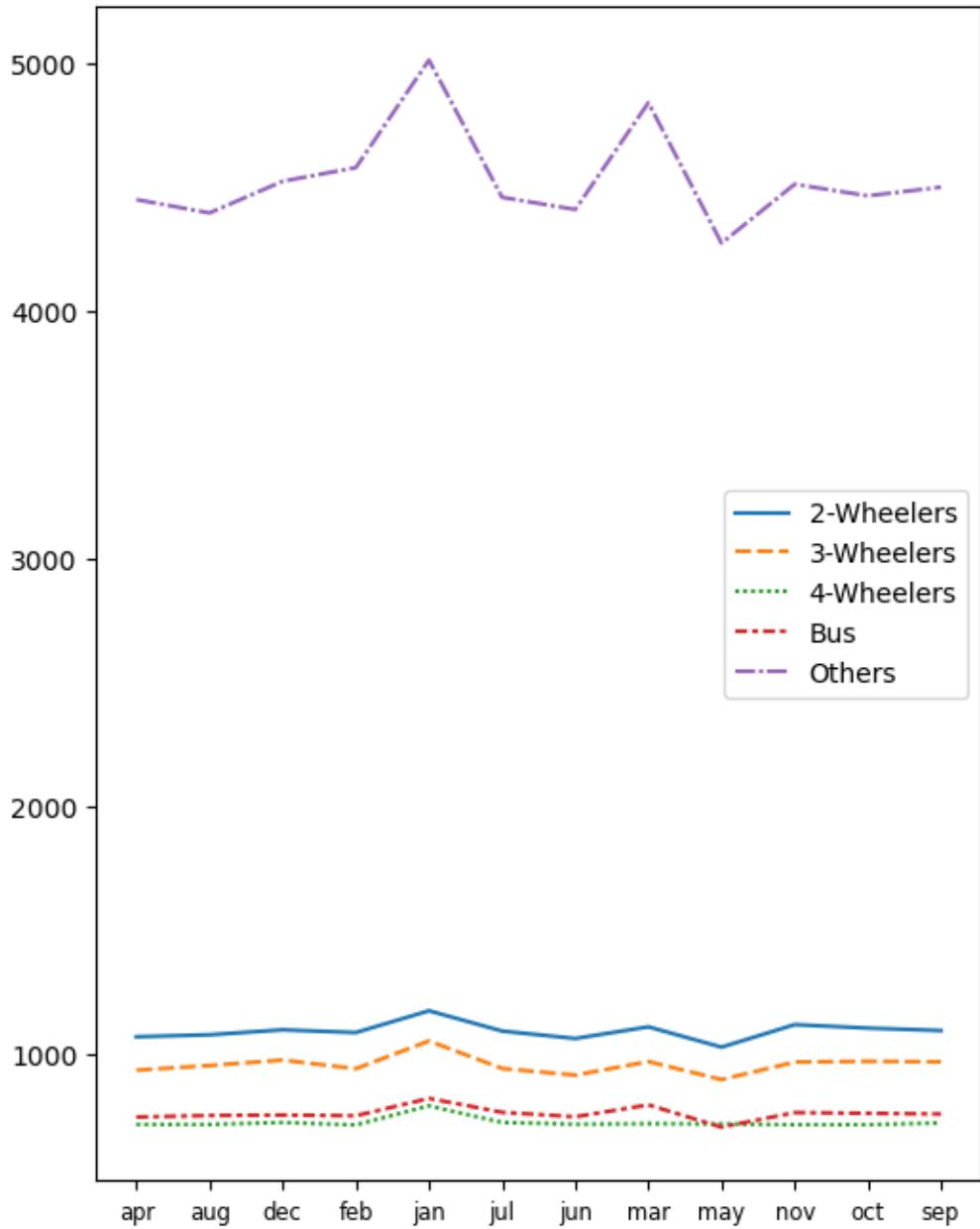
```
[25]: Category_Sales_Month = data.groupby(['Vehicle_Category','Month_Name']).  
      ↪count()['Date'].reset_index().rename(columns={'Date':'count'}).  
      ↪pivot(index='Month_Name',columns='Vehicle_Category',values='count').  
      ↪reset_index()  
Category_Sales_Month
```

	Vehicle_Category	Month_Name	2-Wheelers	3-Wheelers	4-Wheelers	Bus	Others
0		apr	1070	935	716	746	4450
1		aug	1078	954	716	752	4396
2		dec	1098	976	724	754	4524
3		feb	1087	941	714	751	4579
4		jan	1175	1053	791	821	5013
5		jul	1093	942	724	765	4459
6		jun	1063	915	717	747	4410
7		mar	1110	970	719	795	4841
8		may	1028	897	718	705	4274
9		nov	1119	968	715	764	4512

10	oct	1105	971	715	761	4465
11	sep	1095	969	722	758	4500

```
[26]: plt.figure(figsize=(6,8))
sb.lineplot(data=Category_Sales_Month)
plt.legend(loc='upper left')
plt.legend(fontsize=10)
plt.
    ↪xticks(range(len(Category_Sales_Month['Month_Name'])),Category_Sales_Month['Month_Name'],si
```

[26]: ([<matplotlib.axis.XTick at 0x1e520fc2660>,<matplotlib.axis.XTick at 0x1e520fc2b10>,<matplotlib.axis.XTick at 0x1e520f4a420>,<matplotlib.axis.XTick at 0x1e520ff3680>,<matplotlib.axis.XTick at 0x1e520ff3c50>,<matplotlib.axis.XTick at 0x1e520f964e0>,<matplotlib.axis.XTick at 0x1e521024d40>,<matplotlib.axis.XTick at 0x1e5210256d0>,<matplotlib.axis.XTick at 0x1e520fac1d0>,<matplotlib.axis.XTick at 0x1e521401760>,<matplotlib.axis.XTick at 0x1e5214020c0>,<matplotlib.axis.XTick at 0x1e520f70650>], [Text(0, 0, 'apr'),Text(1, 0, 'aug'),Text(2, 0, 'dec'),Text(3, 0, 'feb'),Text(4, 0, 'jan'),Text(5, 0, 'jul'),Text(6, 0, 'jun'),Text(7, 0, 'mar'),Text(8, 0, 'may'),Text(9, 0, 'nov'),Text(10, 0, 'oct'),Text(11, 0, 'sep')])



```
[27]: Type_Sales_Year = data.groupby(['Vehicle_Type', 'Year']).count()['Date'].
    ↪reset_index().rename(columns={'Date': 'count'}).
    ↪pivot(index='Year', columns='Vehicle_Type', values='count').reset_index()
Type_Sales_Year
```

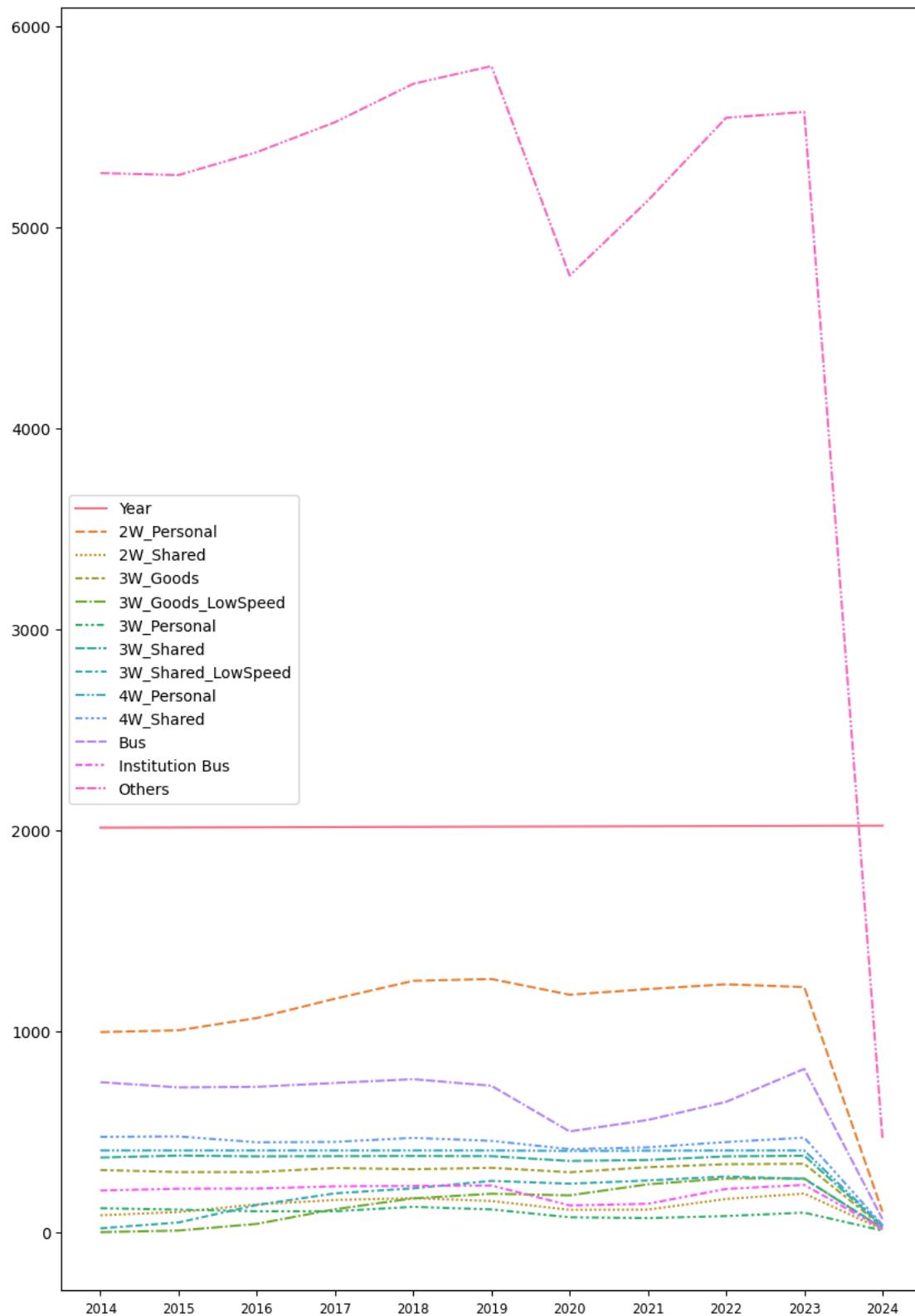
	Vehicle_Type	Year	2W_Personal	2W_Shared	3W_Goods	3W_Goods_LowSpeed	\
0		2014	997	86	311	2	
1		2015	1006	102	301	10	
2		2016	1067	140	301	43	

3	2017	1163	162	321	116
4	2018	1252	171	315	170
5	2019	1261	157	322	193
6	2020	1183	113	300	185
7	2021	1211	114	325	240
8	2022	1235	167	340	269
9	2023	1221	194	342	269
10	2024	104	15	30	20
Vehicle_Type	3W_Personal	3W_Shared	3W_Shared_LowSpeed	4W_Personal	\
0	121	373	21	408	
1	114	383	50	408	
2	106	379	137	408	
3	106	380	195	408	
4	128	381	220	408	
5	115	380	257	408	
6	76	356	243	406	
7	72	361	259	407	
8	82	379	279	408	
9	99	382	267	408	
10	10	32	23	34	
Vehicle_Type	4W_Shared	Bus	Institution	Bus	Others
0	476	748		209	5270
1	478	722		218	5260
2	449	725		219	5374
3	451	744		230	5523
4	471	763		232	5714
5	456	730		234	5802
6	415	503		135	4760
7	424	560		143	5133
8	450	650		217	5545
9	472	814		237	5574
10	38	67		19	468

```
[28]: plt.figure(figsize=(10,15))
sb.lineplot(data=Type_Sales_Year)
plt.legend(loc='upper left')
plt.legend(fontsize=10)
plt.
    xticks(range(len(Type_Sales_Year['Year'])), Type_Sales_Year['Year'], size='small')

[28]: ([<matplotlib.axis.XTick at 0x1e5214010a0>,
        <matplotlib.axis.XTick at 0x1e521422b10>,
        <matplotlib.axis.XTick at 0x1e520865ac0>,
        <matplotlib.axis.XTick at 0x1e5214aa9c0>,
        <matplotlib.axis.XTick at 0x1e5214ab290>,
```

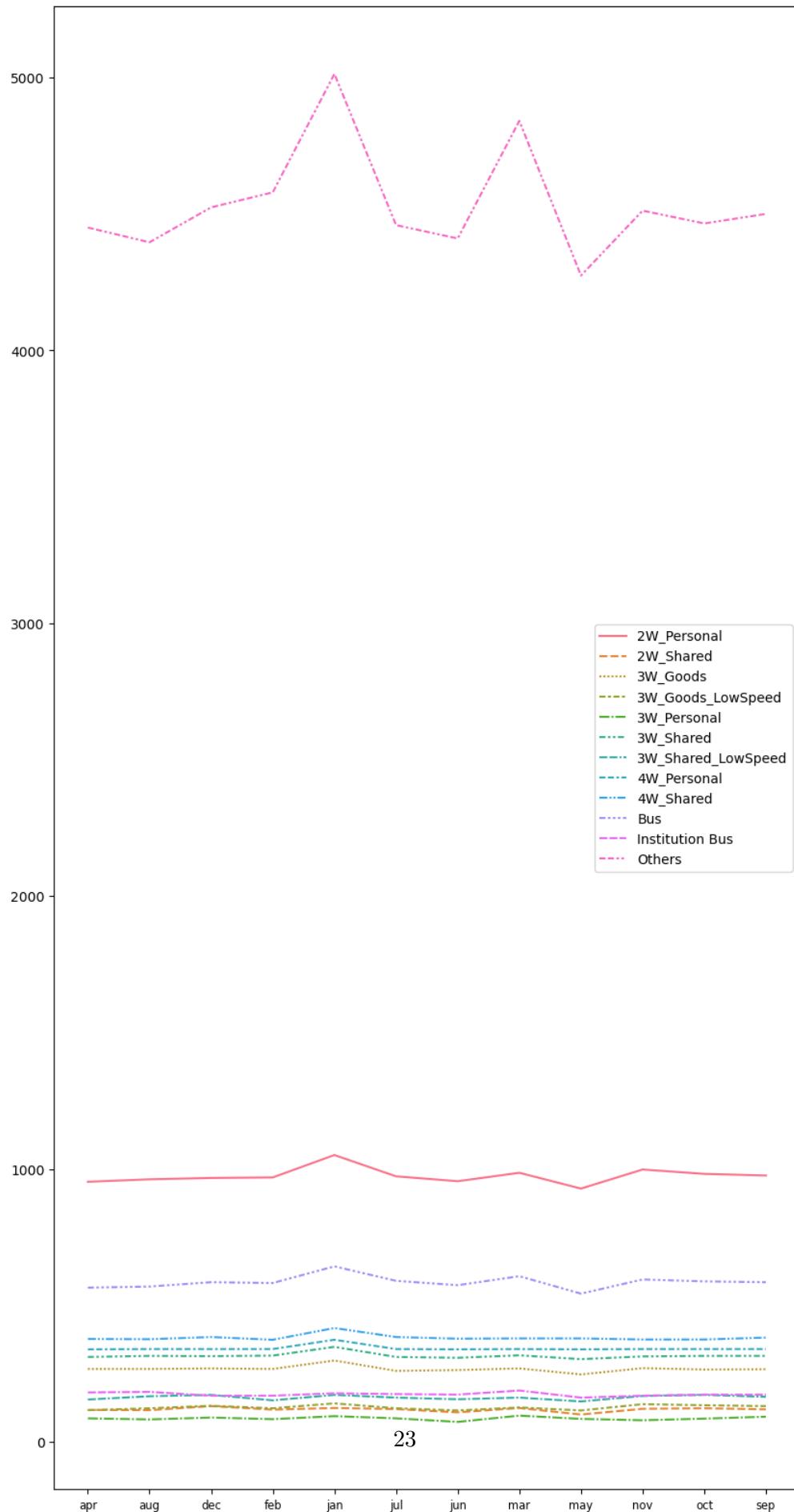
```
<matplotlib.axis.XTick at 0x1e5214abb90>,
<matplotlib.axis.XTick at 0x1e5214d04d0>,
<matplotlib.axis.XTick at 0x1e521420c20>,
<matplotlib.axis.XTick at 0x1e521439490>,
<matplotlib.axis.XTick at 0x1e521541dc0>,
<matplotlib.axis.XTick at 0x1e521541eb0>] ,
[Text(0, 0, '2014'),
Text(1, 0, '2015'),
Text(2, 0, '2016'),
Text(3, 0, '2017'),
Text(4, 0, '2018'),
Text(5, 0, '2019'),
Text(6, 0, '2020'),
Text(7, 0, '2021'),
Text(8, 0, '2022'),
Text(9, 0, '2023'),
Text(10, 0, '2024')])
```



```
[29]: Type_sales_Month = data.groupby(['Vehicle_Type','Month_Name']).count()['Date'].reset_index().rename(columns={'Date':'count'}).pivot(index='Month_Name',columns='Vehicle_Type',values='count').reset_index()
```

```
[30]: plt.figure(figsize=(10,20))
sb.lineplot(data=Type_sales_Month)
plt.legend(loc='upper left')
plt.legend(fontsize=10)
plt.
    xticks(range(len(Type_sales_Month['Month_Name'])),Type_sales_Month['Month_Name'],size='small')
```

```
[30]: ([<matplotlib.axis.XTick at 0x1e52151d0d0>,
        <matplotlib.axis.XTick at 0x1e52151d1f0>,
        <matplotlib.axis.XTick at 0x1e521403ef0>,
        <matplotlib.axis.XTick at 0x1e5215769c0>,
        <matplotlib.axis.XTick at 0x1e5215772f0>,
        <matplotlib.axis.XTick at 0x1e521577800>,
        <matplotlib.axis.XTick at 0x1e521577020>,
        <matplotlib.axis.XTick at 0x1e5215cc860>,
        <matplotlib.axis.XTick at 0x1e51d2f6090>,
        <matplotlib.axis.XTick at 0x1e5215f63f0>,
        <matplotlib.axis.XTick at 0x1e52186ee10>,
        <matplotlib.axis.XTick at 0x1e521438e60>],
 [Text(0, 0, 'apr'),
  Text(1, 0, 'aug'),
  Text(2, 0, 'dec'),
  Text(3, 0, 'feb'),
  Text(4, 0, 'jan'),
  Text(5, 0, 'jul'),
  Text(6, 0, 'jun'),
  Text(7, 0, 'mar'),
  Text(8, 0, 'may'),
  Text(9, 0, 'nov'),
  Text(10, 0, 'oct'),
  Text(11, 0, 'sep')])
```



```
[31]: Class_Sales_Year = data.groupby(['Vehicle_Class','Year']).count()['Date'].
      ↪reset_index().rename(columns={'Date':'count'}).
      ↪pivot(index='Year',columns='Vehicle_Class',values='count').reset_index()
Class_Sales_Year
```

Vehicle_Class	Year	ADAPTED VEHICLE	AGRICULTURAL TRACTOR	AMBULANCE	\
0	2014	218.0	335.0	337.0	
1	2015	227.0	344.0	331.0	
2	2016	238.0	334.0	335.0	
3	2017	230.0	314.0	329.0	
4	2018	245.0	319.0	329.0	
5	2019	232.0	312.0	332.0	
6	2020	203.0	285.0	283.0	
7	2021	223.0	302.0	354.0	
8	2022	228.0	310.0	371.0	
9	2023	245.0	304.0	363.0	
10	2024	21.0	27.0	29.0	

Vehicle_Class	ANIMAL AMBULANCE	ARMOURED/SPECIALISED VEHICLE	\
0	15.0	NaN	
1	20.0	NaN	
2	27.0	NaN	
3	18.0	NaN	
4	21.0	2.0	
5	33.0	1.0	
6	27.0	1.0	
7	29.0	7.0	
8	40.0	25.0	
9	46.0	32.0	
10	3.0	4.0	

Vehicle_Class	ARTICULATED VEHICLE	AUXILIARY TRAILER	BREAKDOWN VAN	\
0	152.0	6.0	18.0	
1	165.0	4.0	19.0	
2	167.0	3.0	22.0	
3	192.0	12.0	27.0	
4	211.0	22.0	33.0	
5	189.0	11.0	32.0	
6	125.0	5.0	10.0	
7	178.0	4.0	11.0	
8	193.0	6.0	14.0	
9	217.0	8.0	20.0	
10	17.0	NaN	1.0	

Vehicle_Class	BULLDOZER	TRACTOR (COMMERCIAL)	\
0	6.0	...	248.0
1	11.0	...	240.0
2	4.0	...	235.0
3	2.0	...	238.0
4	7.0	...	222.0
5	2.0	...	225.0
6	NaN	...	217.0
7	NaN	...	230.0
8	1.0	...	219.0
9	1.0	...	232.0
10	NaN	...	19.0

Vehicle_Class	TRACTOR-TROLLEY(COMMERCIAL)	TRAILER (AGRICULTURAL)	\
0	50.0	141.0	
1	46.0	147.0	
2	56.0	147.0	
3	64.0	153.0	
4	80.0	119.0	
5	76.0	113.0	
6	66.0	111.0	
7	64.0	102.0	
8	68.0	97.0	
9	60.0	94.0	
10	6.0	7.0	

Vehicle_Class	TRAILER (COMMERCIAL)	TRAILER FOR PERSONAL USE	\
0	208.0	86.0	
1	209.0	63.0	
2	214.0	68.0	
3	197.0	56.0	
4	199.0	21.0	
5	200.0	6.0	
6	175.0	12.0	
7	170.0	10.0	
8	157.0	7.0	
9	171.0	4.0	
10	16.0	NaN	

Vehicle_Class	TREE TRIMMING VEHICLE	VEHICLE FITTED WITH COMPRESSOR	\
0	6.0	78.0	
1	8.0	85.0	
2	4.0	96.0	
3	7.0	67.0	
4	3.0	83.0	
5	4.0	83.0	
6	1.0	55.0	

```

7                      NaN                  53.0
8                      1.0                  64.0
9                      NaN                  53.0
10                     NaN                 3.0

Vehicle_Class  VEHICLE FITTED WITH GENERATOR  VEHICLE FITTED WITH RIG \
0                      54.0                111.0
1                      54.0                117.0
2                      47.0                123.0
3                      46.0                118.0
4                      44.0                103.0
5                      48.0                123.0
6                      24.0                 75.0
7                      27.0                 70.0
8                      31.0                 64.0
9                      45.0                 59.0
10                     7.0                  8.0

Vehicle_Class  X-RAY VAN
0                      2.0
1                      NaN
2                      1.0
3                      2.0
4                      4.0
5                      NaN
6                      2.0
7                      1.0
8                      NaN
9                      NaN
10                     NaN

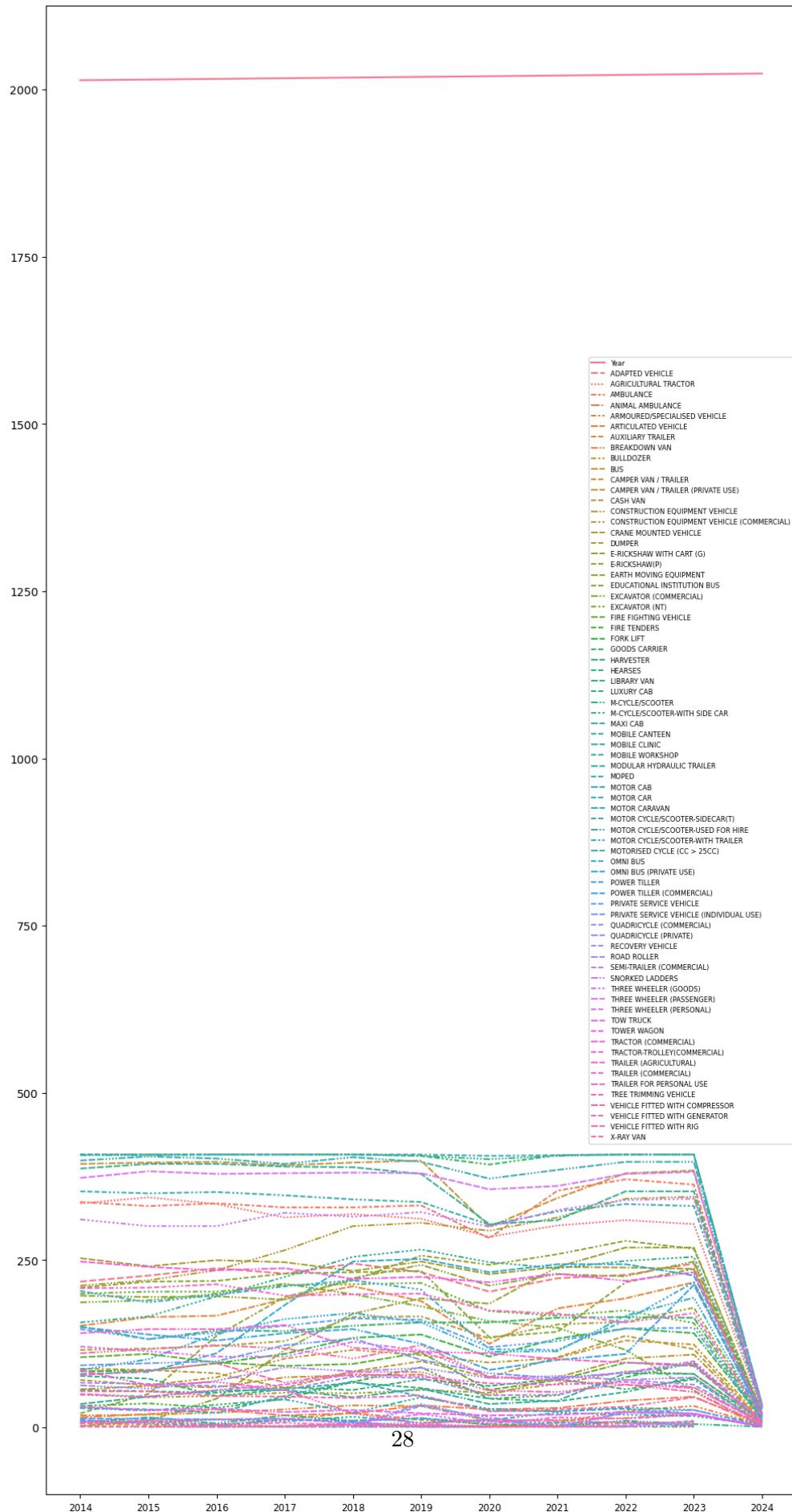
[11 rows x 74 columns]

```

```
[32]: plt.figure(figsize=(12,24))
sb.lineplot(data=Class_Sales_Year)
plt.legend(loc='upper left')
plt.legend(fontsize=6)
plt.
    ↪xticks(range(len(Class_Sales_Year['Year'])), Class_Sales_Year['Year'], size='small')
```

```
[32]: ([<matplotlib.axis.XTick at 0x1e5218aeff0>,
    <matplotlib.axis.XTick at 0x1e5218ade80>,
    <matplotlib.axis.XTick at 0x1e521952e40>,
    <matplotlib.axis.XTick at 0x1e5214aa7e0>,
    <matplotlib.axis.XTick at 0x1e5208930b0>,
    <matplotlib.axis.XTick at 0x1e521ab8290>,
    <matplotlib.axis.XTick at 0x1e521ab8a10>,
```

```
<matplotlib.axis.XTick at 0x1e521a83e90>,
<matplotlib.axis.XTick at 0x1e520866a50>,
<matplotlib.axis.XTick at 0x1e52050f260>,
<matplotlib.axis.XTick at 0x1e52050fc20>],
[Text(0, 0, '2014'),
 Text(1, 0, '2015'),
 Text(2, 0, '2016'),
 Text(3, 0, '2017'),
 Text(4, 0, '2018'),
 Text(5, 0, '2019'),
 Text(6, 0, '2020'),
 Text(7, 0, '2021'),
 Text(8, 0, '2022'),
 Text(9, 0, '2023'),
 Text(10, 0, '2024')])
```



```
[33]: Class_Sales_Month = data.groupby(['Vehicle_Class','Month_Name']).  
    ↪count()['Date'].reset_index().rename(columns={'Date':'count'}).  
    ↪pivot(index='Month_Name',columns='Vehicle_Class',values='count').  
    ↪reset_index()  
Class_Sales_Month
```

Vehicle_Class	Month_Name	ADAPTED VEHICLE	AGRICULTURAL TRACTOR	AMBULANCE	\
0	apr	182.0	254.0	264.0	
1	aug	189.0	262.0	285.0	
2	dec	199.0	257.0	279.0	
3	feb	194.0	268.0	288.0	
4	jan	214.0	293.0	304.0	
5	jul	198.0	266.0	286.0	
6	jun	179.0	259.0	271.0	
7	mar	200.0	277.0	293.0	
8	may	175.0	258.0	274.0	
9	nov	190.0	259.0	297.0	
10	oct	195.0	269.0	282.0	
11	sep	195.0	264.0	270.0	

Vehicle_Class	ANIMAL AMBULANCE	ARMOURED/SPECIALISED VEHICLE	\
0	30.0	3.0	
1	22.0	6.0	
2	25.0	6.0	
3	24.0	3.0	
4	23.0	10.0	
5	24.0	2.0	
6	21.0	9.0	
7	26.0	8.0	
8	22.0	5.0	
9	24.0	4.0	
10	17.0	12.0	
11	21.0	4.0	

Vehicle_Class	ARTICULATED VEHICLE	AUXILIARY TRAILER	BREAKDOWN VAN	\
0	149.0	6.0	15.0	
1	145.0	7.0	12.0	
2	153.0	4.0	14.0	
3	157.0	7.0	11.0	
4	167.0	7.0	20.0	
5	149.0	6.0	20.0	
6	137.0	9.0	19.0	
7	159.0	10.0	18.0	
8	137.0	6.0	22.0	

9		148.0	7.0	16.0
10		157.0	5.0	19.0
11		148.0	7.0	21.0

Vehicle_Class	BULLDOZER	TRACTOR (COMMERCIAL)	\
0	2.0	...	189.0
1	3.0	...	186.0
2	2.0	...	193.0
3	4.0	...	195.0
4	7.0	...	215.0
5	3.0	...	190.0
6	2.0	...	198.0
7	1.0	...	199.0
8	NaN	...	184.0
9	5.0	...	194.0
10	1.0	...	184.0
11	4.0	...	198.0

Vehicle_Class	TRACTOR-TROLLEY(COMMERCIAL)	TRAILER (AGRICULTURAL)	\
0		50.0	98.0
1		52.0	101.0
2		56.0	102.0
3		48.0	111.0
4		57.0	115.0
5		55.0	106.0
6		59.0	102.0
7		55.0	100.0
8		46.0	95.0
9		50.0	103.0
10		55.0	100.0
11		53.0	98.0

Vehicle_Class	TRAILER (COMMERCIAL)	TRAILER FOR PERSONAL USE	\
0	161.0		33.0
1	151.0		28.0
2	157.0		23.0
3	166.0		30.0
4	171.0		26.0
5	154.0		31.0
6	158.0		30.0
7	167.0		35.0
8	157.0		23.0
9	156.0		28.0
10	157.0		21.0
11	161.0		25.0

Vehicle_Class	TREE TRIMMING VEHICLE	VEHICLE FITTED WITH COMPRESSOR	\
---------------	-----------------------	--------------------------------	---

```
0           6.0          65.0
1            NaN         45.0
2            1.0         57.0
3            3.0         59.0
4            3.0         73.0
5            NaN         54.0
6            3.0         53.0
7            4.0         73.0
8            7.0         56.0
9            3.0         68.0
10           2.0         60.0
11           2.0         57.0
```

```
Vehicle_Class  VEHICLE FITTED WITH GENERATOR  VEHICLE FITTED WITH RIG \
0                  25.0          80.0
1                  35.0          68.0
2                  44.0          82.0
3                  39.0          79.0
4                  45.0         103.0
5                  31.0          73.0
6                  34.0          73.0
7                  44.0          95.0
8                  35.0          74.0
9                  34.0          84.0
10                 31.0          84.0
11                 30.0          76.0
```

```
Vehicle_Class  X-RAY VAN
0            NaN
1            1.0
2            NaN
3            2.0
4            2.0
5            NaN
6            1.0
7            4.0
8            1.0
9            NaN
10           1.0
11           NaN
```

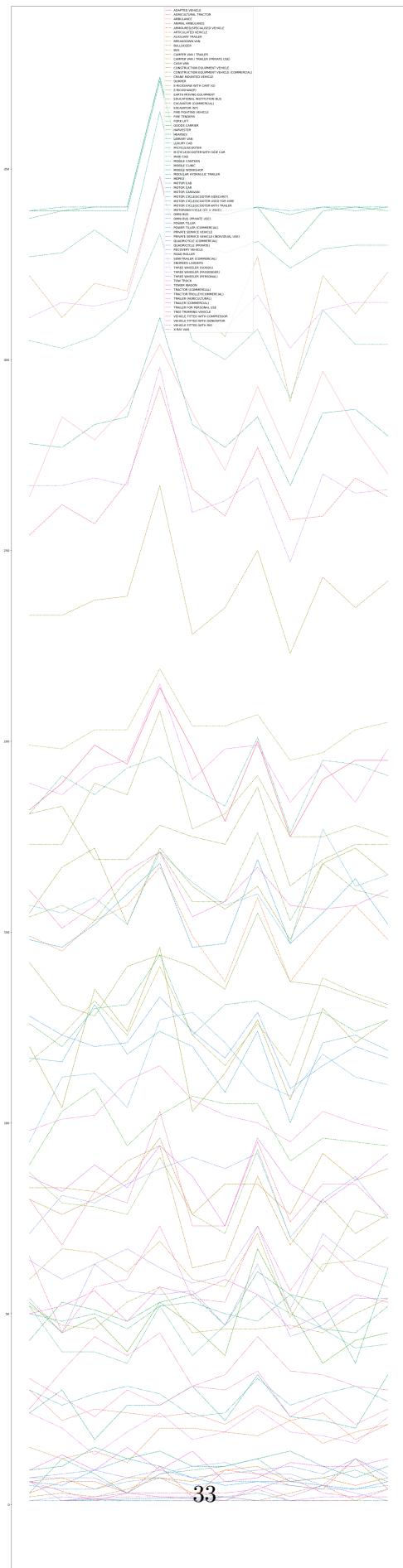
[12 rows x 74 columns]

```
[34]: plt.figure(figsize=(24,96))
sb.lineplot(data=Class_Sales_Month)
plt.legend(loc='upper right')
plt.legend(fontsize=10)
```

```
plt.
```

```
    ↪xticks(range(len(Class_Sales_Month['Month_Name'])),Class_Sales_Month['Month_Name'],size='sm')
```

```
[34]: ([<matplotlib.axis.XTick at 0x1e5206a8f80>,
<matplotlib.axis.XTick at 0x1e5206aa450>,
<matplotlib.axis.XTick at 0x1e52052b020>,
<matplotlib.axis.XTick at 0x1e52069d6d0>,
<matplotlib.axis.XTick at 0x1e5206efad0>,
<matplotlib.axis.XTick at 0x1e521cb1340>,
<matplotlib.axis.XTick at 0x1e521cb1fd0>,
<matplotlib.axis.XTick at 0x1e521cb2720>,
<matplotlib.axis.XTick at 0x1e52076e840>,
<matplotlib.axis.XTick at 0x1e522f0d400>,
<matplotlib.axis.XTick at 0x1e5231315e0>,
<matplotlib.axis.XTick at 0x1e523131d90>],
[Text(0, 0, 'apr'),
Text(1, 0, 'aug'),
Text(2, 0, 'dec'),
Text(3, 0, 'feb'),
Text(4, 0, 'jan'),
Text(5, 0, 'jul'),
Text(6, 0, 'jun'),
Text(7, 0, 'mar'),
Text(8, 0, 'may'),
Text(9, 0, 'nov'),
Text(10, 0, 'oct'),
Text(11, 0, 'sep')])
```



```
[35]: State_Sales_Year = data.groupby(['State', 'Year']).count()['Date'].reset_index()
      .rename(columns={'Date':'count'}).
      .pivot(index='Year',columns='State',values='count').reset_index()
State_Sales_Year
```

	State	Year	Andaman & Nicobar Island	Andhra Pradesh	Arunachal Pradesh	\
0		2014	110	307	219	
1		2015	122	315	219	
2		2016	133	323	232	
3		2017	134	337	233	
4		2018	130	350	253	
5		2019	123	364	272	
6		2020	93	336	192	
7		2021	111	362	203	
8		2022	117	379	217	
9		2023	143	357	226	
10		2024	10	27	19	

	State	Assam	Bihar	Chandigarh	Chhattisgarh	DNH and DD	Delhi	...	Odisha	\
0		275	202	143	324	166	188	...	346	
1		280	216	144	326	168	209	...	350	
2		285	232	135	334	181	192	...	351	
3		297	264	137	365	172	183	...	385	
4		322	282	156	406	215	189	...	452	
5		314	264	159	380	204	196	...	455	
6		293	235	149	346	162	163	...	384	
7		332	255	151	348	185	164	...	413	
8		338	277	175	365	209	179	...	423	
9		347	292	191	365	244	191	...	431	
10		31	25	14	31	21	17	...	37	

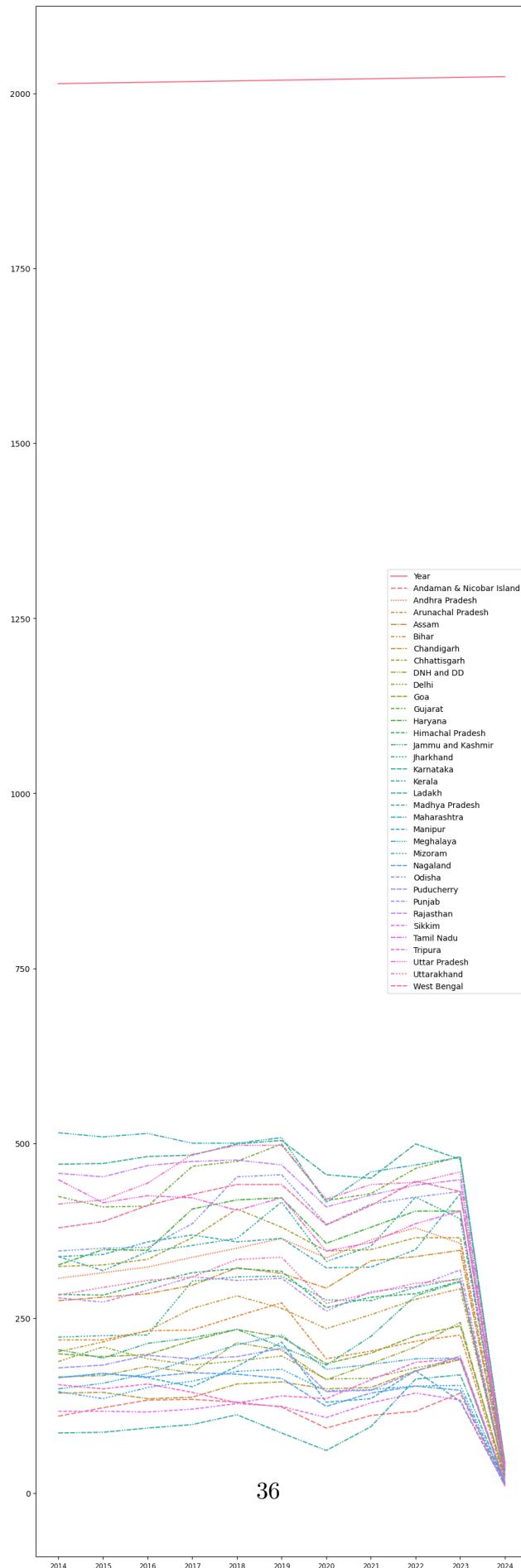
	State	Puducherry	Punjab	Rajasthan	Sikkim	Tamil Nadu	Tripura	\
0		179	279	457	117	448	155	
1		183	273	452	117	415	149	
2		197	290	468	116	425	156	
3		192	309	474	120	422	144	
4		195	304	476	128	404	129	
5		207	307	469	124	422	139	
6		145	260	409	108	346	135	
7		147	288	426	129	358	162	
8		174	295	440	143	385	187	
9		196	319	448	133	403	192	
10		17	26	33	11	35	16	

State	Uttar Pradesh	Uttarakhand	West Bengal
0	413	283	379
1	419	294	388
2	443	304	411
3	484	309	427
4	497	334	441
5	497	337	441
6	420	271	383
7	441	286	411
8	444	300	446
9	459	305	431
10	40	22	38

[11 rows x 35 columns]

```
[36]: plt.figure(figsize=(11,35))
sb.lineplot(data=State_Sales_Year)
plt.legend(loc='upper left')
plt.legend(fontsize=10)
plt.
    ↪xticks(range(len(State_Sales_Year['Year'])), State_Sales_Year['Year'], size='small')
```

```
[36]: ([<matplotlib.axis.XTick at 0x1e517517d10>,
<matplotlib.axis.XTick at 0x1e5175170b0>,
<matplotlib.axis.XTick at 0x1e523159520>,
<matplotlib.axis.XTick at 0x1e52305d7f0>,
<matplotlib.axis.XTick at 0x1e52305e150>,
<matplotlib.axis.XTick at 0x1e52305cad0>,
<matplotlib.axis.XTick at 0x1e52305e480>,
<matplotlib.axis.XTick at 0x1e52305e7b0>,
<matplotlib.axis.XTick at 0x1e517d91dc0>,
<matplotlib.axis.XTick at 0x1e520f97bf0>,
<matplotlib.axis.XTick at 0x1e520f73170>],
[Text(0, 0, '2014'),
Text(1, 0, '2015'),
Text(2, 0, '2016'),
Text(3, 0, '2017'),
Text(4, 0, '2018'),
Text(5, 0, '2019'),
Text(6, 0, '2020'),
Text(7, 0, '2021'),
Text(8, 0, '2022'),
Text(9, 0, '2023'),
Text(10, 0, '2024')])
```



```
[37]: State_Sales_Month = data.groupby(['State','Month_Name']).count()['Date'].
      ↪reset_index().rename(columns={'Date':'count'}).
      ↪pivot(index='Month_Name',columns='State',values='count').reset_index()
State_Sales_Month
```

	State Month_Name	Andaman & Nicobar Island	Andhra Pradesh	Arunachal Pradesh	\
0	apr	102	290		176
1	aug	96	285		188
2	dec	101	294		192
3	feb	107	287		192
4	jan	107	313		201
5	jul	106	275		198
6	jun	95	273		186
7	mar	103	305		212
8	may	96	277		179
9	nov	103	284		183
10	oct	101	287		191
11	sep	109	287		187

State	Assam	Bihar	Chandigarh	Chhattisgarh	DNH and DD	Delhi	...	Odisha	\
0	245	198	132	291	158	160	...	338	
1	259	213	129	291	157	156	...	327	
2	255	221	135	304	166	149	...	341	
3	260	209	123	293	160	153	...	334	
4	288	228	140	337	183	167	...	371	
5	259	217	128	307	158	153	...	325	
6	261	203	123	294	153	161	...	325	
7	267	218	137	310	162	158	...	343	
8	249	195	117	274	150	147	...	312	
9	266	215	136	300	166	158	...	345	
10	256	217	125	294	167	152	...	334	
11	249	210	129	295	147	157	...	332	

State	Puducherry	Punjab	Rajasthan	Sikkim	Tamil Nadu	Tripura	\	
0		142	239	382	104	315	116	
1		150	240	376	107	333	131	
2		145	251	379	104	329	133	
3		159	246	371	109	349	139	
4		164	264	404	105	380	149	
5		151	239	375	107	337	130	
6		154	242	367	105	335	128	
7		170	253	390	113	359	135	
8		141	227	362	102	320	127	
9		152	244	378	102	332	130	

```
10          146    255      381     94     329    123
11          158    250      387     94     345    123
```

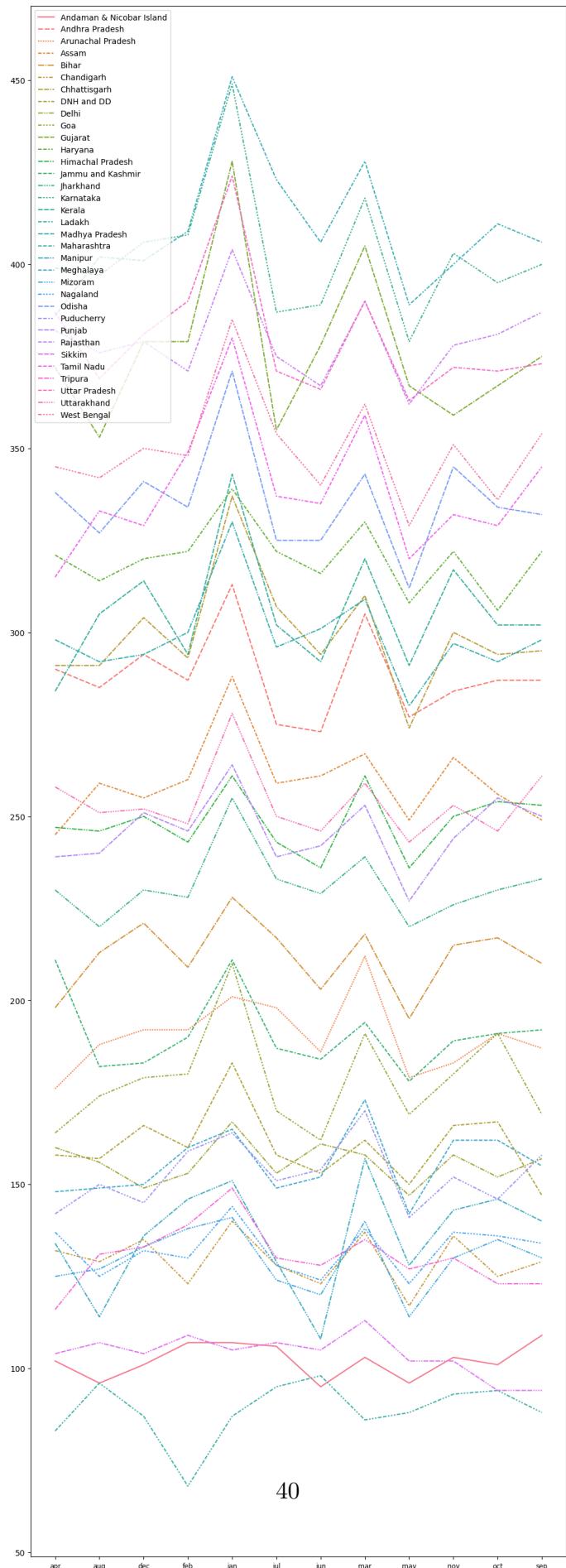
	State	Uttar Pradesh	Uttarakhand	West Bengal
0		387	258	345
1		369	251	342
2		381	252	350
3		390	248	348
4		424	278	385
5		371	250	354
6		366	246	340
7		390	259	362
8		363	243	329
9		372	253	351
10		371	246	336
11		373	261	354

[12 rows x 35 columns]

```
[38]: plt.figure(figsize=(12,35))
sb.lineplot(data=State_Sales_Month)
plt.legend(loc='upper left')
plt.legend(fontsize=10)
plt.
    ↪xticks(range(len(State_Sales_Month['Month_Name'])), State_Sales_Month['Month_Name'], size='sm')
```

```
[38]: ([<matplotlib.axis.XTick at 0x1e521421880>,
        <matplotlib.axis.XTick at 0x1e5214036b0>,
        <matplotlib.axis.XTick at 0x1e52068d2b0>,
        <matplotlib.axis.XTick at 0x1e5232ad6d0>,
        <matplotlib.axis.XTick at 0x1e5232d3d10>,
        <matplotlib.axis.XTick at 0x1e5203dc9e0>,
        <matplotlib.axis.XTick at 0x1e5203dd280>,
        <matplotlib.axis.XTick at 0x1e5203ddb80>,
        <matplotlib.axis.XTick at 0x1e520472660>,
        <matplotlib.axis.XTick at 0x1e52338b020>,
        <matplotlib.axis.XTick at 0x1e52338b6b0>,
        <matplotlib.axis.XTick at 0x1e52338bda0>],
 [Text(0, 0, 'apr'),
  Text(1, 0, 'aug'),
  Text(2, 0, 'dec'),
  Text(3, 0, 'feb'),
  Text(4, 0, 'jan'),
  Text(5, 0, 'jul'),
  Text(6, 0, 'jun'),
  Text(7, 0, 'mar'),
  Text(8, 0, 'may'),
```

```
Text(9, 0, 'nov'),  
Text(10, 0, 'oct'),  
Text(11, 0, 'sep'))])
```



[]: