

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
In [2]: df = pd.read_csv('EDA_Practice_Dataset_2_Telemarketingtopic_added1627027202s_p1515805800.csv')
```

```
In [3]: df.head()
```

```
Out[3]:
```

	Call_ID	Sale	Agent_ID	Age	Product_ID	Timezone	Phone_code	First_Name	Last_Name	Area_Code	Gender	Call_Count
0	9545434	False	5265	42	147	2	37.0	Jk	Jk	2302	Male	1
1	9211206	False	5226	74	146	2	37.0	Em	Sh	1501	Male	10
2	8873010	False	4452	35	144	2	37.0	BI	MI	1550	Male	9
3	9852034	False	5461	40	149	2	37.0	WT	LI	1401	Male	6
4	9416548	False	5298	26	147	2	37.0	LA	LA	125	Female	12

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Call_ID     100000 non-null  int64
1   Sale        99968 non-null   object
2   Agent_ID    97685 non-null   object
3   Age         100000 non-null  int64
4   Product_ID  100000 non-null  int64
5   Timezone    100000 non-null  int64
6   Phone_code  99972 non-null   float64
7   First_Name  99318 non-null   object
8   Last_Name   99319 non-null   object
9   Area_Code   99540 non-null   object
10  Gender      100000 non-null  object
11  Call_Count  100000 non-null  int64
dtypes: float64(1), int64(5), object(6)
memory usage: 9.2+ MB
```

```
In [5]: df.describe()
```

```
Out[5]:
```

	Call_ID	Age	Product_ID	Timezone	Phone_code	Call_Count
count	1.000000e+05	100000.000000	100000.000000	100000.000000	99972.0	100000.000000
mean	9.562981e+06	52.558530	147.216690	1.999420	37.0	3.26866
std	4.618927e+05	16.144553	2.182815	0.034054	0.0	2.96882
min	8.432514e+06	25.000000	142.000000	0.000000	37.0	1.00000
25%	9.230592e+06	39.000000	146.000000	2.000000	37.0	1.00000
50%	9.527831e+06	53.000000	147.000000	2.000000	37.0	2.00000
75%	9.919197e+06	66.000000	149.000000	2.000000	37.0	4.00000
max	1.047664e+07	80.000000	151.000000	2.000000	37.0	55.00000

```
In [6]: df.shape
```

```
Out[6]: (100000, 12)
```

```
In [7]: print(df['Sale'].unique())
print(df['Agent_ID'].unique())
print(df['Phone_code'].unique())
print(df['First_Name'].unique())
print(df['Last_Name'].unique())
```

```
print(df['Area_Code'].unique())
```

```
[False True nan]
['5265' '5226' '4452' '5461' '5298' '5139' '4828' '5292' '5044' '4912'
 '5230' '5002' '5053' '5312' '5214' '4926' '5425' '5077' '5357' '4722'
 '5076' '4994' '5051' '5472' '5296' '5207' '5386' '5209' '5208' '4866' nan
 '5297' '4995' '5313' '4958' '5073' '5028' '4804' '5341' '4876' '5382'
 '5405' '5146' '5181' '4702' '5272' '4894' '5065' '5403' '5459' '5424'
 '4971' '4677' '4748' '4899' '5266' '5035' '5460' '4968' '4865' '5252'
 '5399' '5061' '4706' '4955' '5220' '4956' '5216' '5032' '5406' '5224'
 '5462' '4121' '5271' '5407' '5206' '5001' '5005' '5287' '5396' '5464'
 '5294' '5052' '5423' '5421' '4827' '4483' '4709' '5384' '5416' '5089'
 '5429' '5428' '5245' '5401' '4970' '5366' '5381' '5367' '5302' '5368'
 '5106' '5495' '5393' '5062' '5143' '5055' '5268' '5042' '5210' '4867'
 '5286' '5470' '5310' '4888' '5218' '5467' '5404' '5455' '5471' '4734'
 '5342' '5205' '5267' '5221' '5327' '5398' '4539' '5417' '5420' '5466'
 '5374' '5441' '4893' '5400' '5299' '5150' '5114' '5246' '5088' '5397'
 '5295' '5426' '5219' 'QUEUE5065' '5457' '5358' '4959' '5356' '5270'
 '5264' '5515' '5369' '5275' '5365' '5064' '5273' '5254' '5293' '4966'
 'QUEUE5404' '5349' '5319' '5383' '5153' '5456' '5303' '5306' '5458'
 '4829' '5427' '5500' 'QUEUE5457' '5364' '5473' '5377' '5503' 'QUEUE5028'
 '5350' '4717' 'QUEUE5341' '4425' '5347' 'QUEUE5002' '5152' '5419' '5415'
 '5380' '5274' '5348' '5155' '5430' '5215' '5203' '5320' '5204' '5225'
 '5351' '5502' '5354' 'QUEUE5456' '5217' '5370' '5373' '5340' '5494'
 '5385' '5179' '5463' '5234' '5322' '5518' '5211' 'QUEUE4702' 'QUEUE4748'
 '5256' '5344' '4992' '5516' '5321' 'QUEUE5181' 'QUEUE4968' '5355' '5376'
 'QUEUE4912' '5519' '5517' 'QUEUE5459' 'QUEUE5381' '4219' '5394' '5343'
 'QUEUE4804' '5269' '5501' '5123' 'QUEUE4971' '4304' '5063' 'QUEUE5405'
 '5390' 'QUEUE5472' '4033' 'QUEUE5386']
[37. nan]
['Jk' 'Em' 'BI' ... 'Qw' 'Yz' 'XC']
['Jk' 'Sh' 'MI' ... 'Yz' 'h' 'mR']
['2302' '1501' '1550' ... '5421' '4013' '3486']
```

```
In [8]: print(df['Call_ID'].unique())
print(df['Age'].unique())
print(df['Product_ID'].unique())
print(df['Timezone'].unique())
print(df['Gender'].unique())
print(df['Call_Count'].unique())
```

```
[9545434 9211206 8873010 ... 9299394 9549720 9702206]
[42 74 35 40 26 33 70 41 72 34 73 30 53 67 79 63 39 25 49 69 55 56 59 58
 43 68 45 62 29 47 46 65 38 54 27 64 71 57 50 48 78 60 32 31 66 36 76 77
 28 75 61 51 80 37 44 52]
[147 146 144 149 150 151 145 143 148 142]
[2 0]
['Male' 'Female' 'Others']
[ 1 10  9  6 12  3  7  8 11  2  5  4 15 14 13 16 17 29 38 21 19 22 55]
```

```
In [9]: df.isnull().sum()
```

```
Out[9]: Call_ID      0
Sale      32
Agent_ID   2315
Age        0
Product_ID  0
Timezone   0
Phone_code 28
First_Name 682
Last_Name  681
Area_Code  460
Gender     0
Call_Count 0
dtype: int64
```

```
In [10]: df['Sale'].mode()
```

```
Out[10]: 0    False
dtype: object
```

```
In [11]: df['Sale'].fillna(df['Sale'].mode()[0], inplace=True)
```

```
In [12]: df.isnull().sum()
```

```
Out[12]: Call_ID      0
Sale      0
Agent_ID   2315
Age        0
Product_ID 0
Timezone   0
Phone_code 28
First_Name 682
Last_Name  681
Area_Code  460
Gender      0
Call_Count 0
dtype: int64
```

```
In [13]: df['Phone_code'].fillna(df['Phone_code'].mean(), inplace=True)
```

```
In [14]: df.isnull().sum()
```

```
Out[14]: Call_ID      0
Sale      0
Agent_ID   2315
Age        0
Product_ID 0
Timezone   0
Phone_code 0
First_Name 682
Last_Name  681
Area_Code  460
Gender      0
Call_Count 0
dtype: int64
```

```
In [15]: df['Agent_ID'] = df['Agent_ID'].replace('QUEUE', '', regex=True)
print(df['Agent_ID'].unique())
```

```
['5265' '5226' '4452' '5461' '5298' '5139' '4828' '5292' '5044' '4912'
'5230' '5002' '5053' '5312' '5214' '4926' '5425' '5077' '5357' '4722'
'5076' '4994' '5051' '5472' '5296' '5207' '5386' '5209' '5208' '4866' nan
'5297' '4995' '5313' '4958' '5073' '5028' '4804' '5341' '4876' '5382'
'5405' '5146' '5181' '4702' '5272' '4894' '5065' '5403' '5459' '5424'
'4971' '4677' '4748' '4899' '5266' '5035' '5460' '4968' '4865' '5252'
'5399' '5061' '4706' '4955' '5220' '4956' '5216' '5032' '5406' '5224'
'5462' '4121' '5271' '5407' '5206' '5001' '5005' '5287' '5396' '5464'
'5294' '5052' '5423' '5421' '4827' '4483' '4709' '5384' '5416' '5089'
'5429' '5428' '5245' '5401' '4970' '5366' '5381' '5367' '5302' '5368'
'5106' '5495' '5393' '5062' '5143' '5055' '5268' '5042' '5210' '4867'
'5286' '5470' '5310' '4888' '5218' '5467' '5404' '5455' '5471' '4734'
'5342' '5205' '5267' '5221' '5327' '5398' '4539' '5417' '5420' '5466'
'5374' '5441' '4893' '5400' '5299' '5150' '5114' '5246' '5088' '5397'
'5295' '5426' '5219' '5457' '5358' '4959' '5356' '5270' '5264' '5515'
'5369' '5275' '5365' '5064' '5273' '5254' '5293' '4966' '5349' '5319'
'5383' '5153' '5456' '5303' '5306' '5458' '4829' '5427' '5500' '5364'
'5473' '5377' '5503' '5350' '4717' '4425' '5347' '5152' '5419' '5415'
'5380' '5274' '5348' '5155' '5430' '5215' '5203' '5320' '5204' '5225'
'5351' '5502' '5354' '5217' '5370' '5373' '5340' '5494' '5385' '5179'
'5463' '5234' '5322' '5518' '5211' '5256' '5344' '4992' '5516' '5321'
'5355' '5376' '5519' '5517' '4219' '5394' '5343' '5269' '5501' '5123'
'4304' '5063' '5390' '4033']
```

```
In [16]: df.isnull().sum()
```

```
Out[16]: Call_ID      0
Sale      0
Agent_ID   2315
Age        0
Product_ID 0
Timezone   0
Phone_code 0
```

```
First_Name    682
Last_Name     681
Area_Code     460
Gender        0
Call_Count    0
dtype: int64
```

```
In [17]: df = df.dropna()
df
```

```
Out[17]:
```

	Call_ID	Sale	Agent_ID	Age	Product_ID	Timezone	Phone_code	First_Name	Last_Name	Area_Code	Gender	Call_Count
0	9545434	False	5265	42	147	2	37.0	Jk	Jk	2302	Male	1
1	9211206	False	5226	74	146	2	37.0	Em	Sh	1501	Male	10
2	8873010	False	4452	35	144	2	37.0	Bl	MI	1550	Male	9
3	9852034	False	5461	40	149	2	37.0	WT	LI	1401	Male	6
4	9416548	False	5298	26	147	2	37.0	LA	LA	125	Female	12
...
99995	10249682	False	5404	65	150	2	37.0	ly	Rn	4068	Female	1
99996	8955506	False	5181	37	144	2	37.0	ZA	OE	4450	Female	6
99997	9299394	False	5077	69	146	2	37.0	Wm	Vr	1759	Male	1
99998	9549720	False	5028	67	147	2	37.0	Ze	Ze	1475	Male	1
99999	9702206	False	5464	63	148	2	37.0	EH	MA	1864	Female	15

96118 rows × 12 columns

```
In [18]: df.isnull().sum()
```

```
Out[18]: Call_ID      0
Sale              0
Agent_ID         0
Age              0
Product_ID       0
Timezone         0
Phone_code       0
First_Name       0
Last_Name        0
Area_Code        0
Gender           0
Call_Count       0
dtype: int64
```

```
In [19]: df.shape
```

```
Out[19]: (96118, 12)
```

```
In [20]: df.dtypes
```

```
Out[20]: Call_ID      int64
Sale              bool
Agent_ID         object
Age              int64
Product_ID       int64
Timezone         int64
Phone_code       float64
First_Name       object
Last_Name        object
Area_Code        object
Gender           object
Call_Count       int64
dtype: object
```

```
In [21]: df['Agent_ID'] = df['Agent_ID'].astype(int)
```

```
In [22]: df.dtypes
```

```
Out[22]: Call_ID      int64
Sale              bool
Agent_ID         int32
Age              int64
Product_ID       int64
Timezone         int64
Phone_code       float64
First_Name       object
Last_Name        object
Area_Code        object
Gender           object
Call_Count       int64
dtype: object
```

```
In [23]: df['Area_Code'] = pd.to_numeric(df['Area_Code'], errors='coerce')
```

```
In [24]: df = df.dropna()
df
```

Out[24]:

	Call_ID	Sale	Agent_ID	Age	Product_ID	Timezone	Phone_code	First_Name	Last_Name	Area_Code	Gender	Call_Count
0	9545434	False	5265	42	147	2	37.0	Jk	Jk	2302.0	Male	1
1	9211206	False	5226	74	146	2	37.0	Em	Sh	1501.0	Male	10
2	8873010	False	4452	35	144	2	37.0	BI	MI	1550.0	Male	9
3	9852034	False	5461	40	149	2	37.0	WT	LI	1401.0	Male	6
4	9416548	False	5298	26	147	2	37.0	LA	LA	125.0	Female	12
...
99995	10249682	False	5404	65	150	2	37.0	ly	Rn	4068.0	Female	1
99996	8955506	False	5181	37	144	2	37.0	ZA	OE	4450.0	Female	6
99997	9299394	False	5077	69	146	2	37.0	Wm	Vr	1759.0	Male	1
99998	9549720	False	5028	67	147	2	37.0	Ze	Ze	1475.0	Male	1
99999	9702206	False	5464	63	148	2	37.0	EH	MA	1864.0	Female	15

96114 rows × 12 columns

```
In [25]: df['Area_Code'].astype(int)
```

```
Out[25]: 0      2302
1      1501
2      1550
3      1401
4       125
...
99995   4068
99996   4450
99997   1759
99998   1475
99999   1864
Name: Area_Code, Length: 96114, dtype: int32
```

```
In [26]: df.dtypes
```

```
Out[26]: Call_ID      int64
Sale              bool
Agent_ID         int32
Age              int64
Product_ID       int64
Timezone         int64
Phone_code       float64
First_Name       object
Last_Name        object
Area_Code        float64
Gender           object
```

Call_Count int64
dtype: object

```
In [27]: df.shape
```

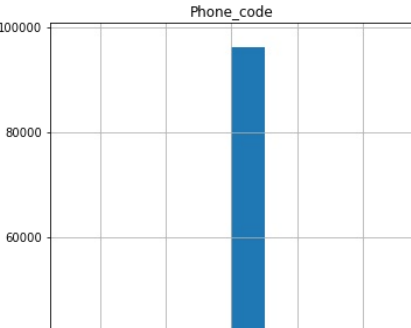
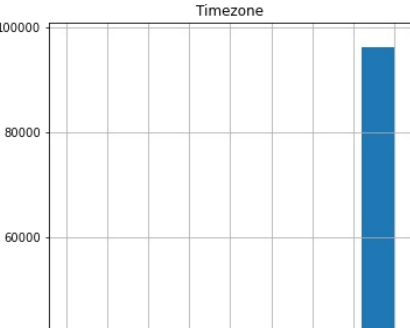
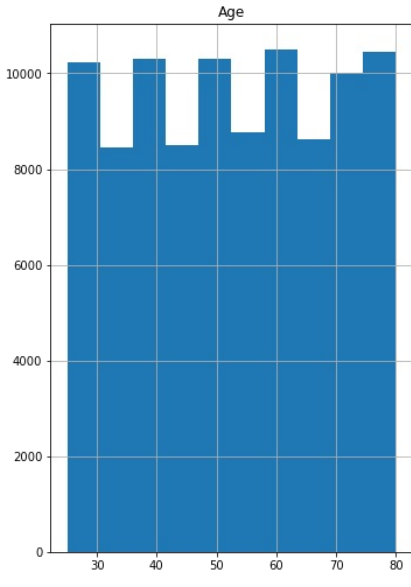
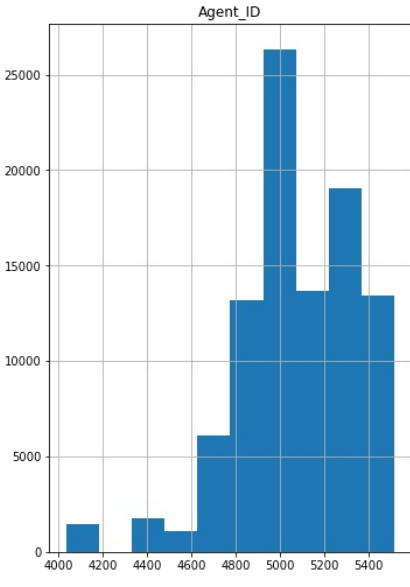
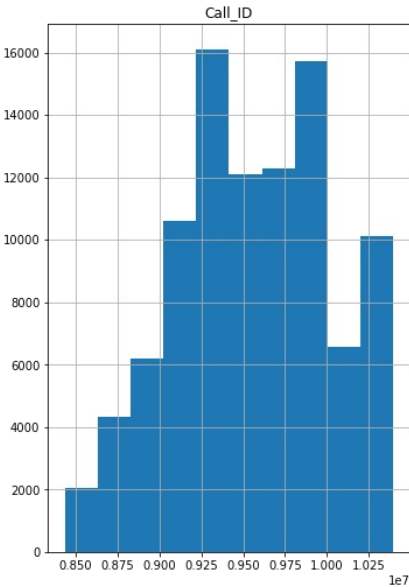
Out[27]: (96114, 12)

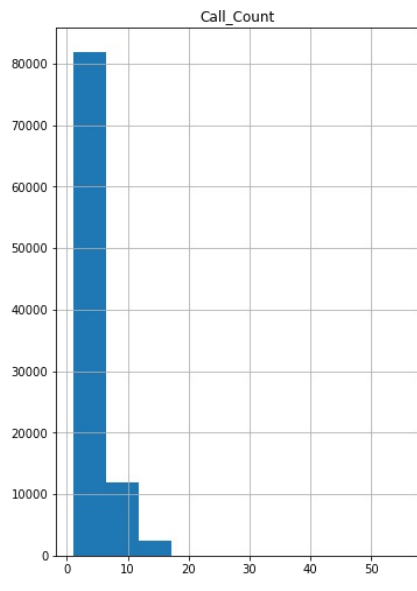
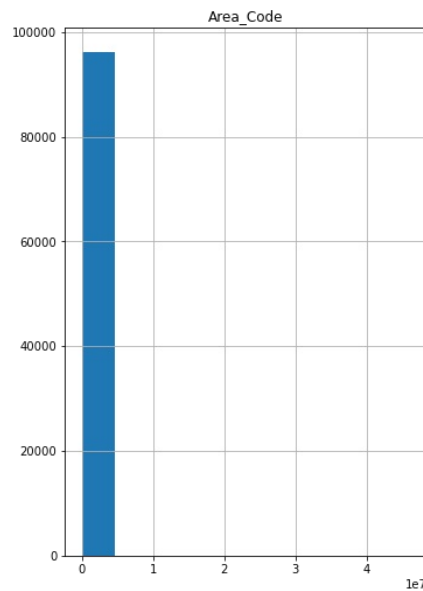
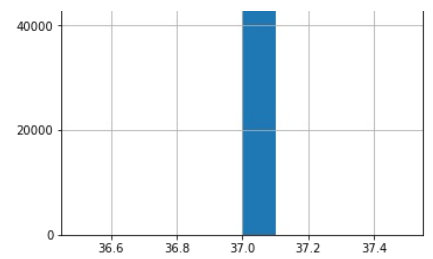
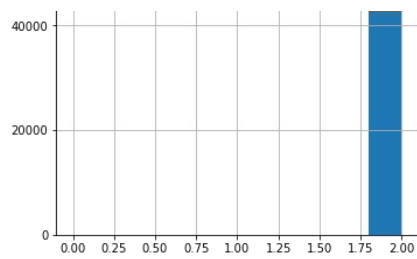
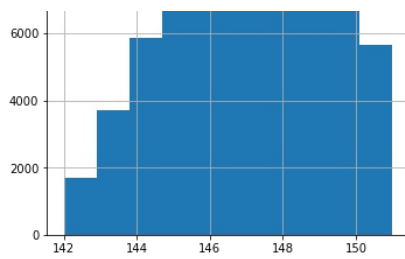
```
In [28]: df.describe()
```

	Call_ID	Agent_ID	Age	Product_ID	Timezone	Phone_code	Area_Code	Call_Count
count	9.611400e+04	96114.000000	96114.000000	96114.000000	96114.000000	96114.0	9.611400e+04	96114.000000
mean	9.557849e+06	5071.442204	52.556006	147.193375	1.999750	37.0	4.160567e+03	3.271074
std	4.611897e+05	260.300428	16.136000	2.180240	0.022346	0.0	1.487234e+05	2.974950
min	8.432514e+06	4033.000000	25.000000	142.000000	0.000000	37.0	0.000000e+00	1.000000
25%	9.228832e+06	4926.000000	39.000000	146.000000	2.000000	37.0	1.619000e+03	1.000000
50%	9.521530e+06	5065.000000	53.000000	147.000000	2.000000	37.0	3.201000e+03	2.000000
75%	9.913814e+06	5271.000000	66.000000	149.000000	2.000000	37.0	6.045000e+03	4.000000
max	1.039830e+07	5519.000000	80.000000	151.000000	2.000000	37.0	4.610361e+07	55.000000

```
In [29]: df.hist(figsize=(20,30))
```

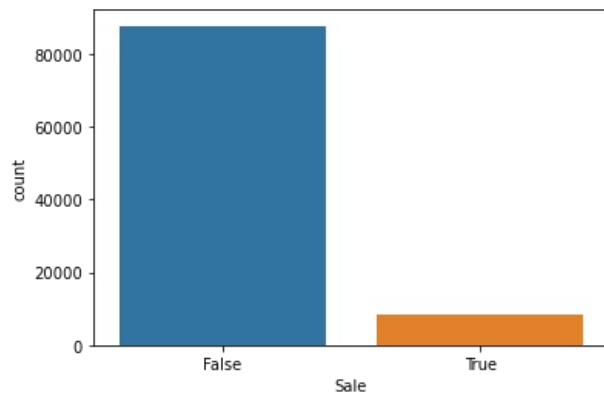
Out[29]: array([[<AxesSubplot:title={'center':'Call_ID'}>,
 <AxesSubplot:title={'center':'Agent_ID'}>,
 <AxesSubplot:title={'center':'Age'}>],
 [<AxesSubplot:title={'center':'Product_ID'}>,
 <AxesSubplot:title={'center':'Timezone'}>,
 <AxesSubplot:title={'center':'Phone_code'}>],
 [<AxesSubplot:title={'center':'Area_Code'}>,
 <AxesSubplot:title={'center':'Call_Count'}>],
 dtype=object)





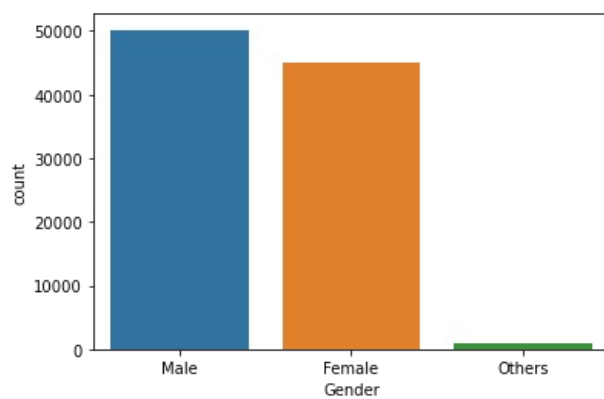
```
In [30]: sns.countplot(x='Sale',data=df)
```

```
Out[30]: <AxesSubplot:xlabel='Sale', ylabel='count'>
```



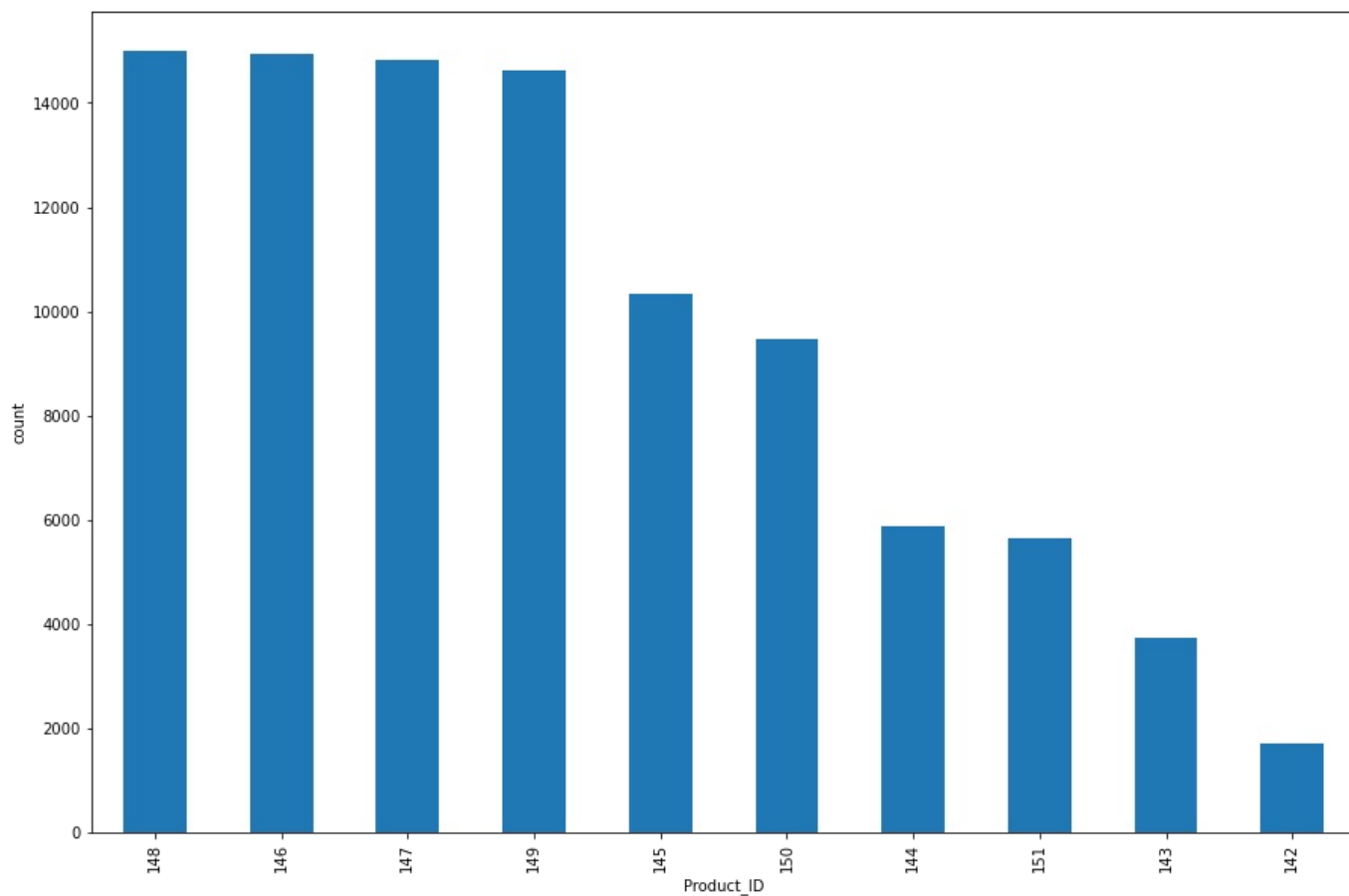
```
In [31]: sns.countplot(x='Gender',data=df)
```

```
Out[31]: <AxesSubplot:xlabel='Gender', ylabel='count'>
```



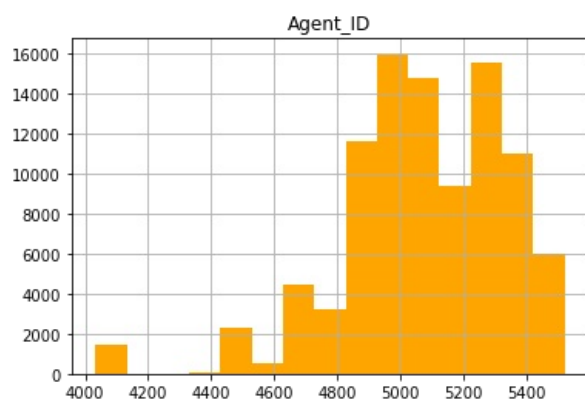
```
In [32]: df.Product_ID.value_counts().plot(kind='bar', figsize=(15,10))
plt.xlabel('Product_ID')
plt.ylabel('count')
```

```
Out[32]: Text(0, 0.5, 'count')
```



```
In [33]: df.hist(column='Agent_ID', bins=15, color='orange')
```

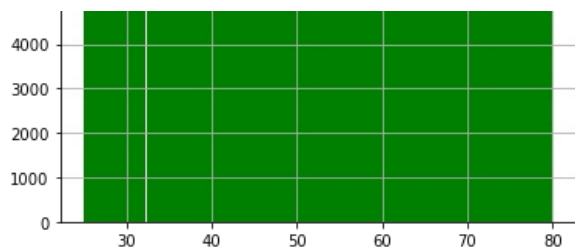
```
Out[33]: array([[<AxesSubplot:title={'center': 'Agent_ID'}>]], dtype=object)
```



```
In [34]: df.hist(column='Age', bins=15, color='green')
```

```
Out[34]: array([[<AxesSubplot:title={'center': 'Age'}>]], dtype=object)
```





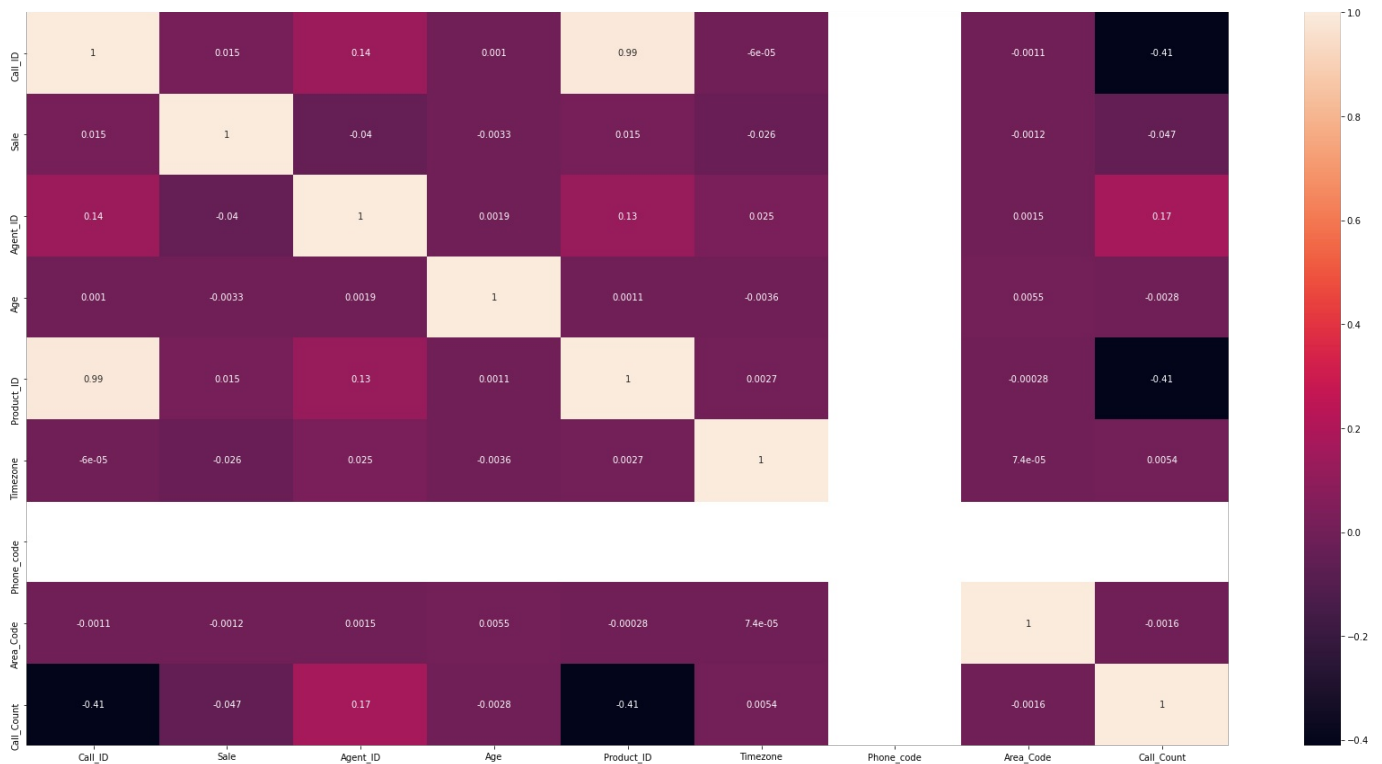
```
In [35]: corr = df.corr()
corr
```

```
Out[35]:
```

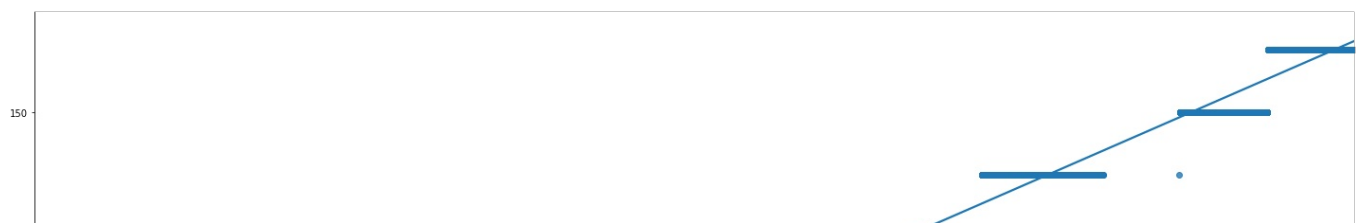
	Call_ID	Sale	Agent_ID	Age	Product_ID	Timezone	Phone_code	Area_Code	Call_Count
Call_ID	1.000000	0.015440	0.138804	0.001010	0.991769	-0.000060	NaN	-0.001095	-0.410631
Sale	0.015440	1.000000	-0.040038	-0.003272	0.015214	-0.026223	NaN	-0.001244	-0.047179
Agent_ID	0.138804	-0.040038	1.000000	0.001888	0.129293	0.025122	NaN	0.001522	0.173343
Age	0.001010	-0.003272	0.001888	1.000000	0.001070	-0.003597	NaN	0.005505	-0.002786
Product_ID	0.991769	0.015214	0.129293	0.001070	1.000000	0.002700	NaN	-0.000280	-0.405538
Timezone	-0.000060	-0.026223	0.025122	-0.003597	0.002700	1.000000	NaN	0.000074	0.005400
Phone_code	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Area_Code	-0.001095	-0.001244	0.001522	0.005505	-0.000280	0.000074	NaN	1.000000	-0.001611
Call_Count	-0.410631	-0.047179	0.173343	-0.002786	-0.405538	0.005400	NaN	-0.001611	1.000000

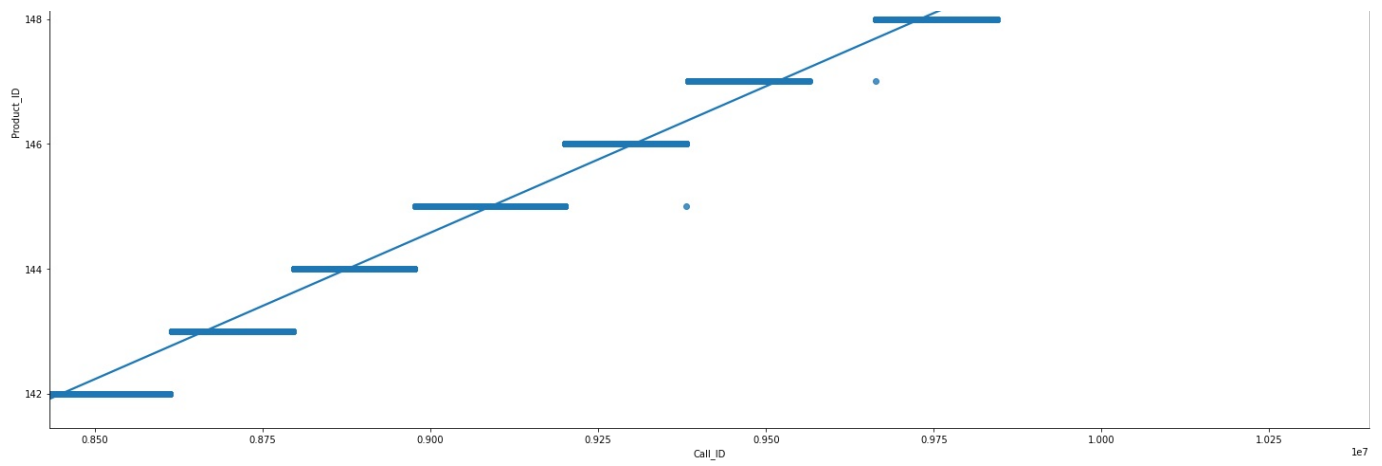
```
In [36]: plt.rcParams['figure.figsize']=(30,15)
sns.heatmap(corr,annot=True,)
```

```
Out[36]: <AxesSubplot:>
```

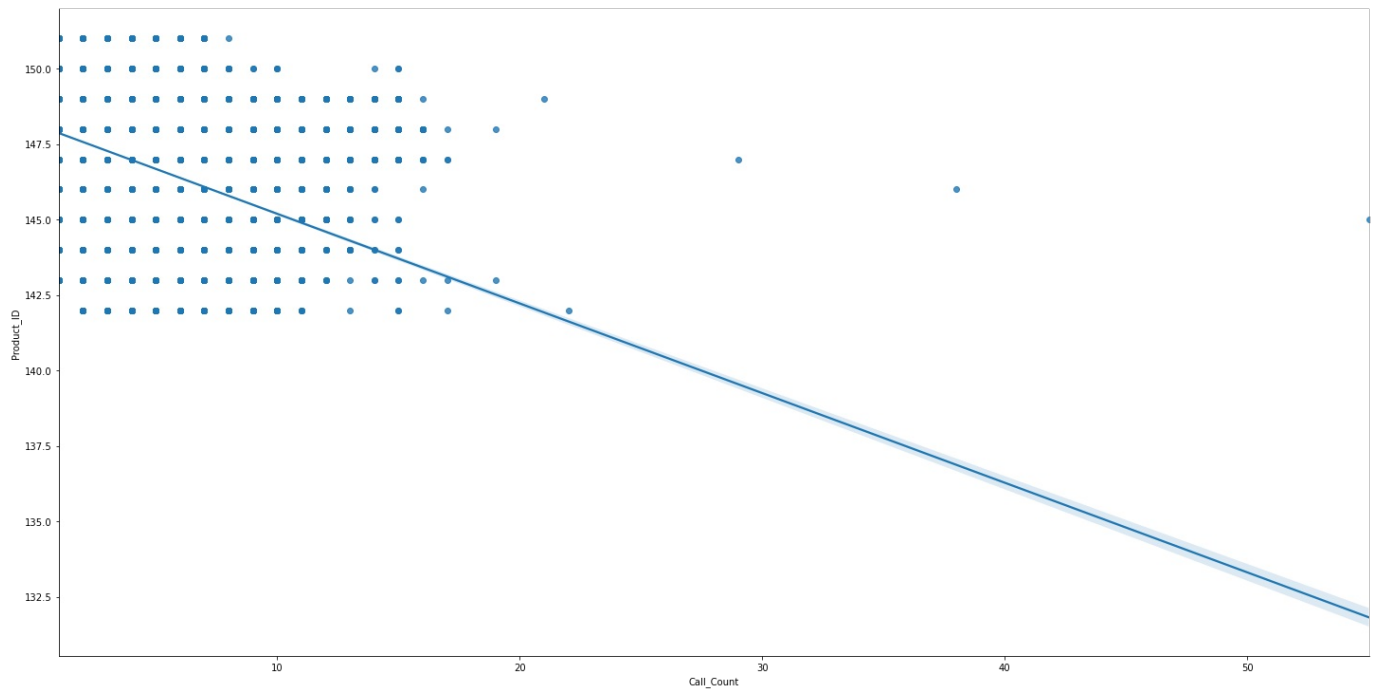


```
In [77]: sns.lmplot(x='Call_ID',y='Product_ID',data=df, height=10, aspect=2);
```

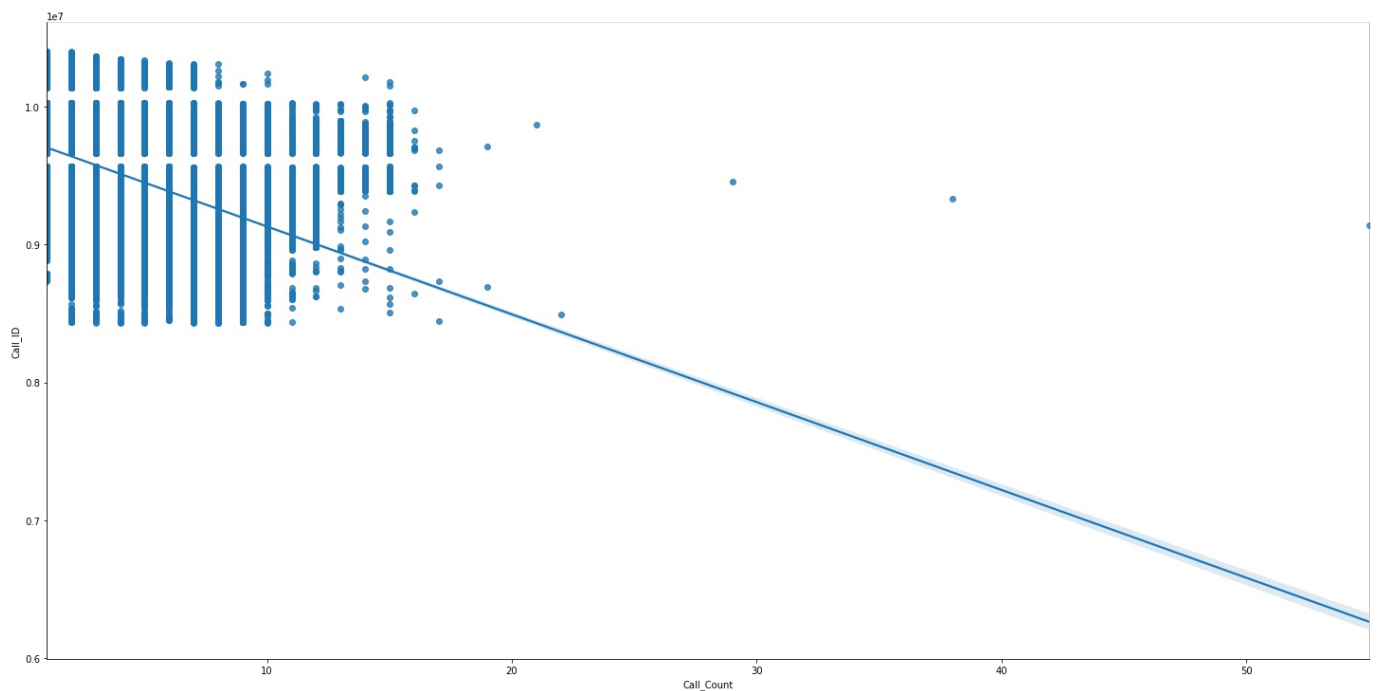




In [78]: `sns.lmplot(x='Call_Count',y='Product_ID',data=df, height=10, aspect=2);`



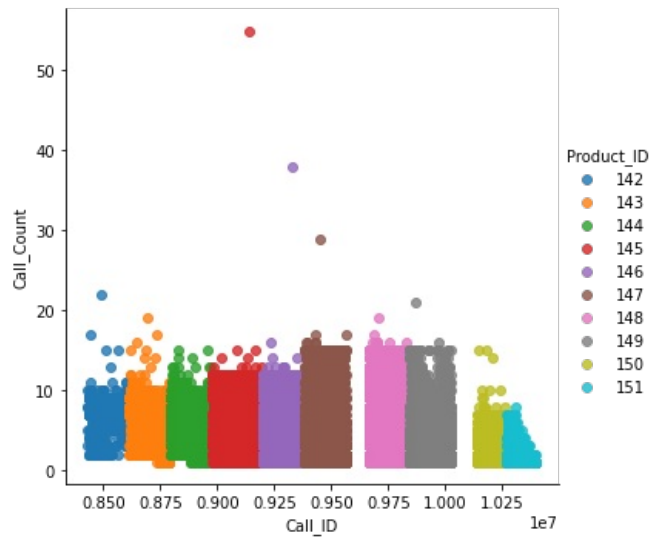
In [76]: `sns.lmplot(x='Call_Count',y='Call_ID',data=df, height=10, aspect=2);`



In [76]:

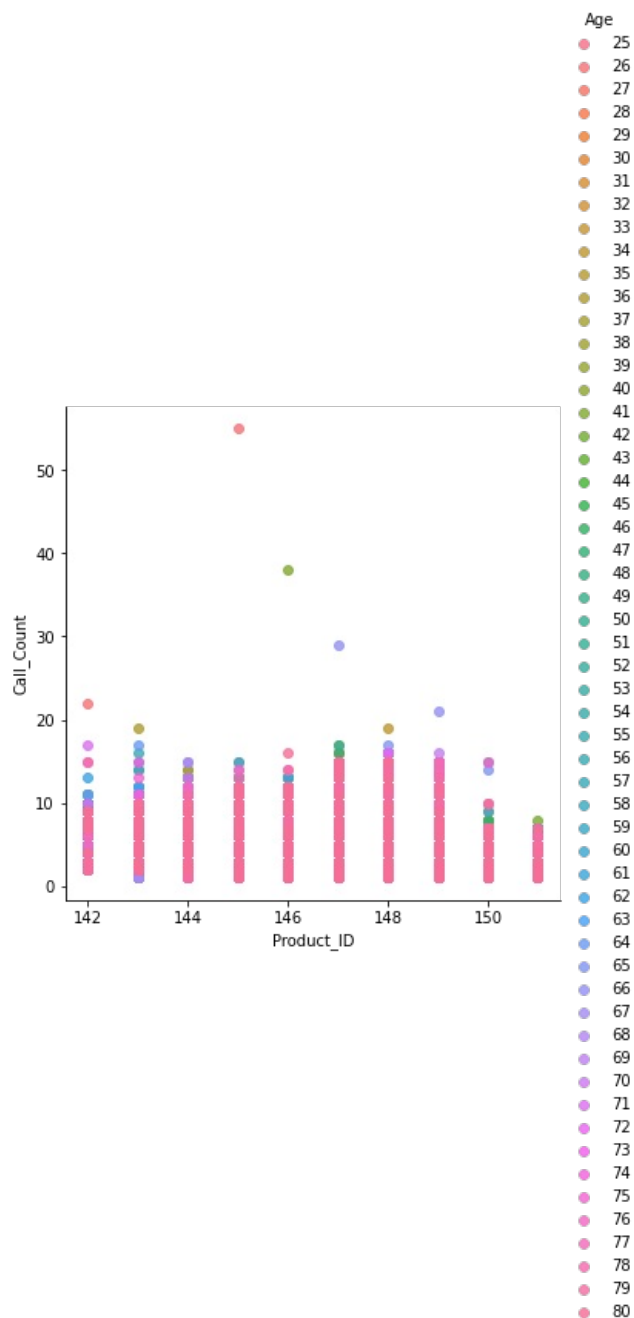
```
In [79]: sns.lmplot(x='Call_ID', y="Call_Count", data=df, hue="Product_ID",fit_reg=False)
```

Out[79]: <seaborn.axisgrid.FacetGrid at 0x16a8a980cd0>



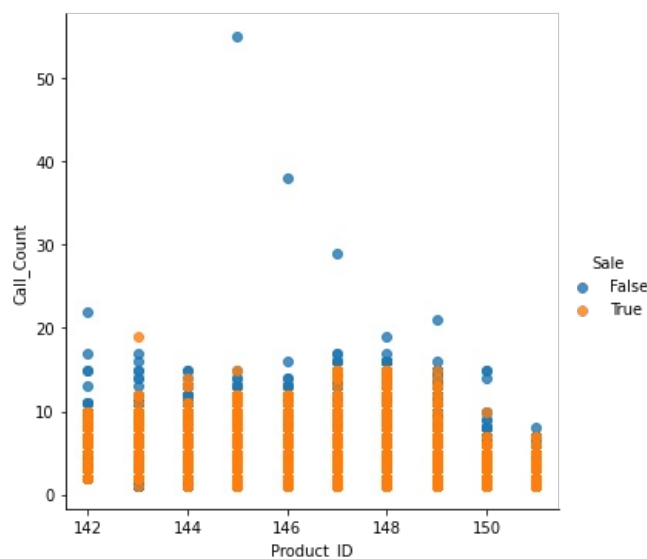
```
In [41]: sns.lmplot(x='Product_ID', y="Call_Count", data=df, hue="Age",fit_reg=False)
```

Out[41]: <seaborn.axisgrid.FacetGrid at 0x16a859ce0a0>



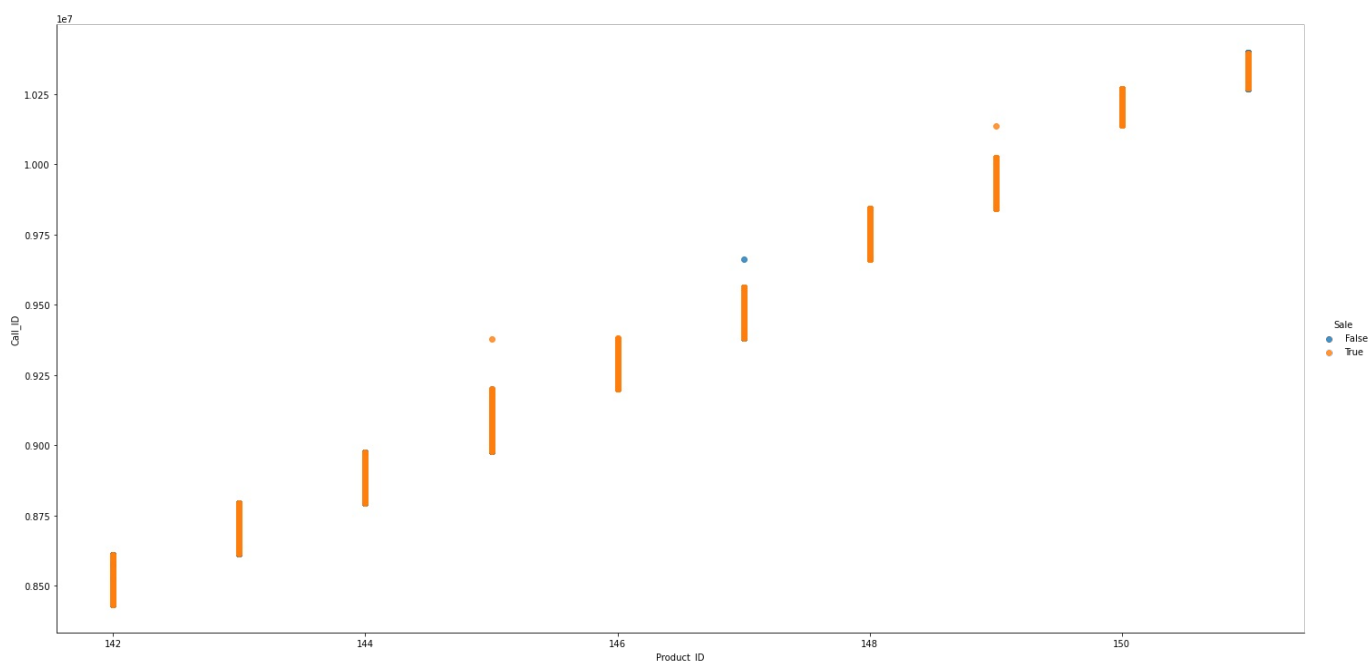
```
In [42]: sns.lmplot(x='Product_ID', y="Call_Count", data=df, hue="Sale", fit_reg=False)
```

```
Out[42]: <seaborn.axisgrid.FacetGrid at 0x16a85851310>
```

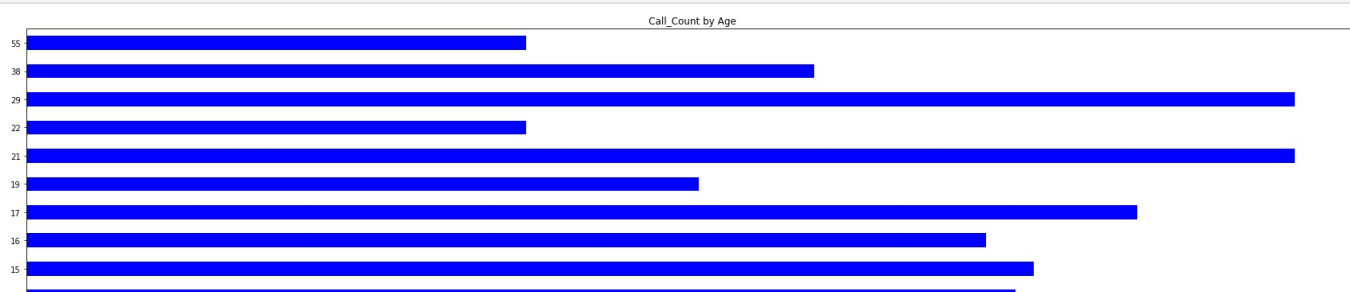


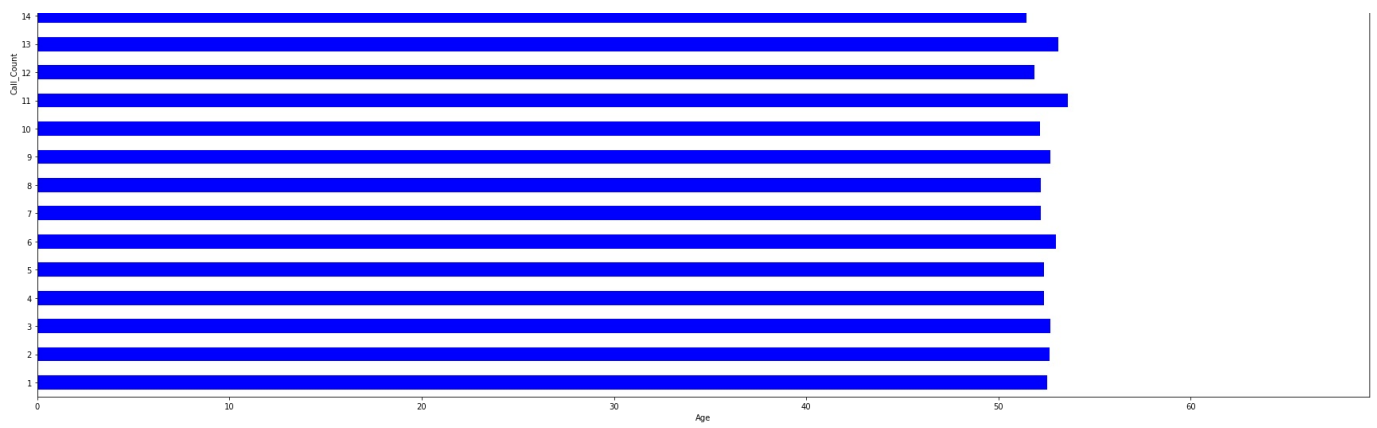
```
In [92]: sns.lmplot(x='Product_ID', y="Call_ID", data=df, hue="Sale", fit_reg=False, height=10, aspect=2)
```

```
Out[92]: <seaborn.axisgrid.FacetGrid at 0x16a902c3af0>
```



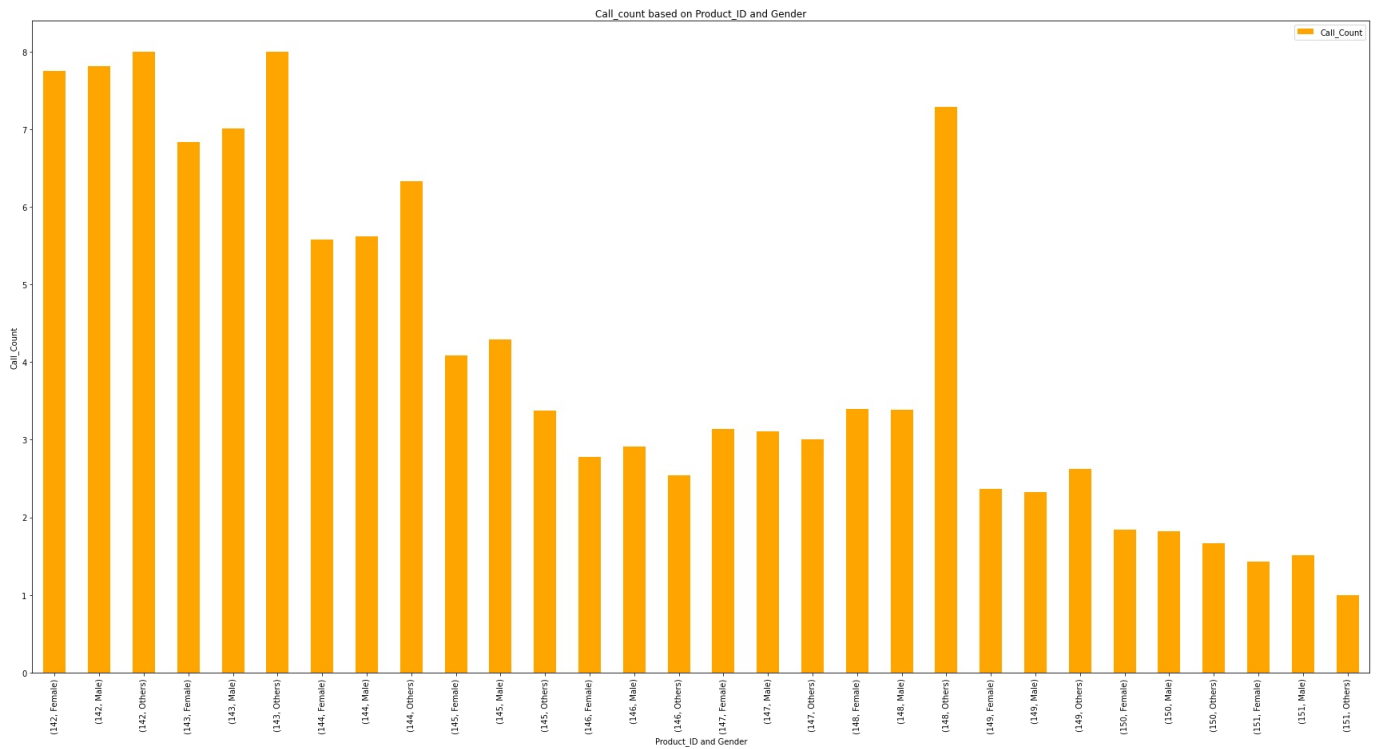
```
In [90]: df.groupby('Call_Count')['Age'].mean().plot(kind='barh', color = 'blue')
plt.title("Call_Count by Age")
plt.ylabel('Call_Count')
plt.xlabel('Age');
```





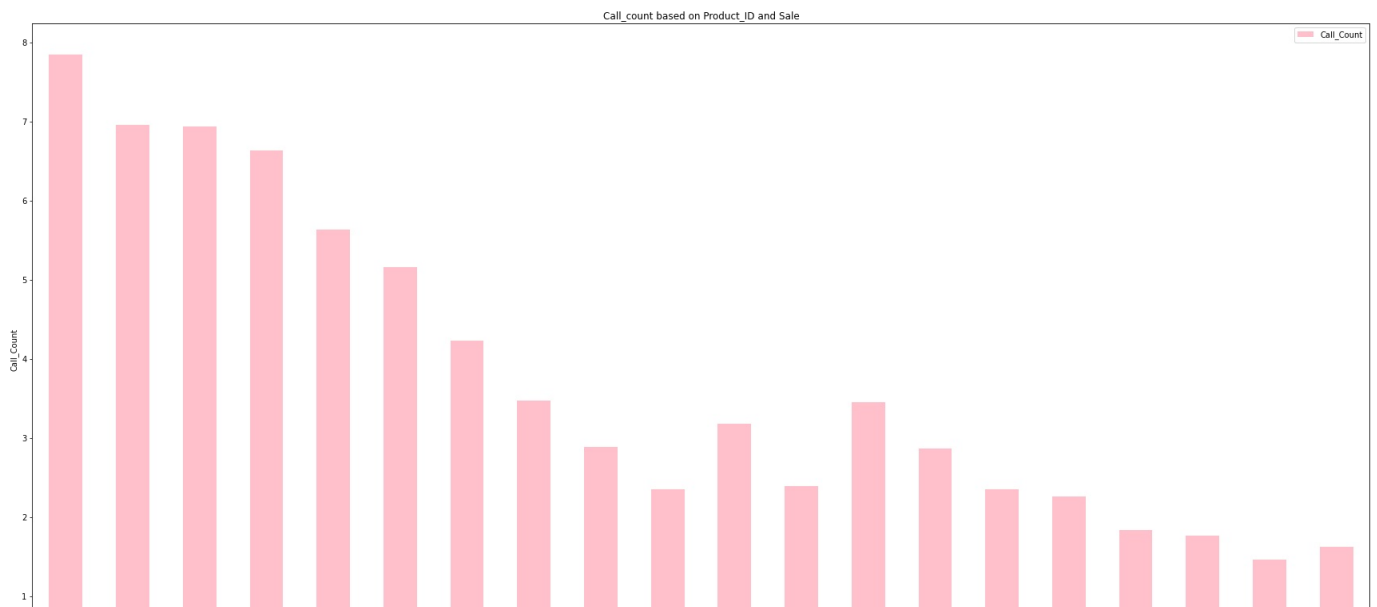
In [44]:

```
pd.pivot_table(df,index=['Product_ID','Gender'], values='Call_Count').plot(kind='bar',color='orange')
plt.title("Call_count based on Product_ID and Gender")
plt.ylabel('Call_Count')
plt.xlabel('Product_ID and Gender');
```



In [45]:

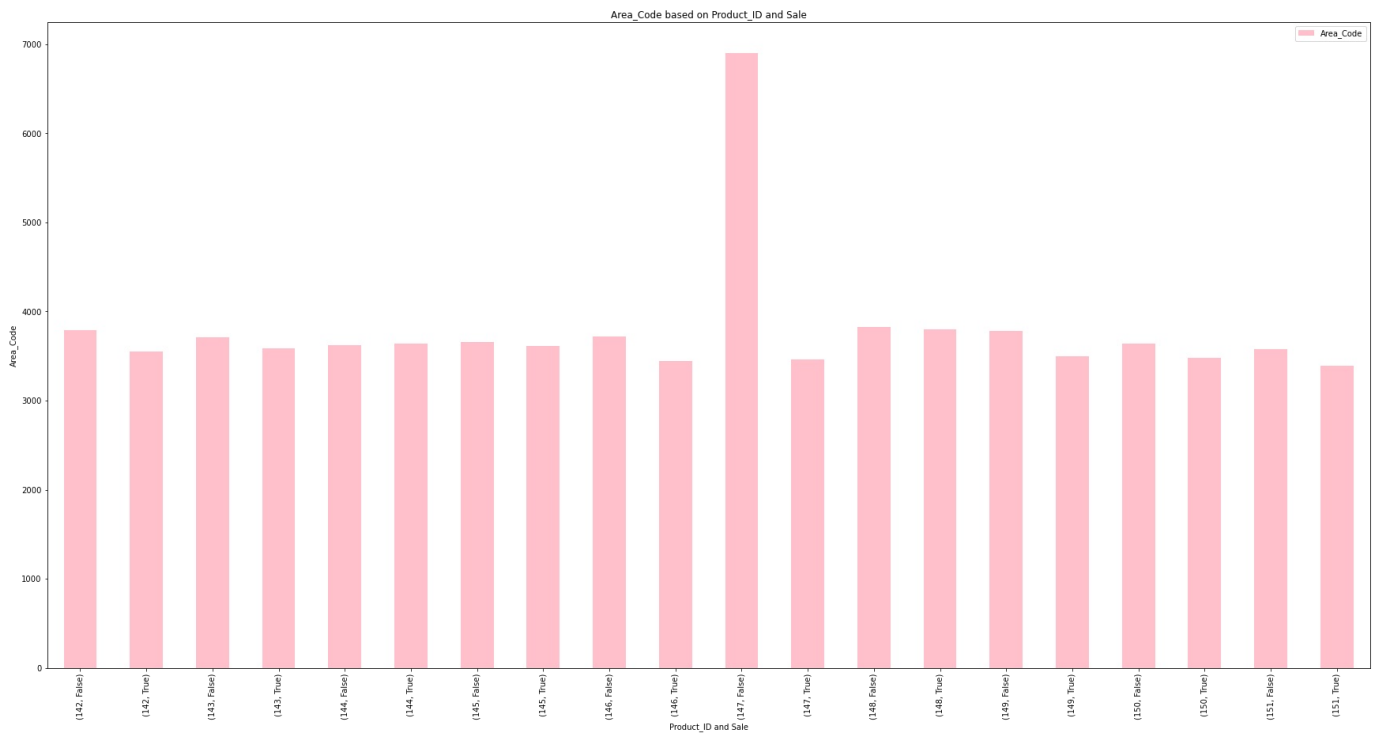
```
pd.pivot_table(df,index=['Product_ID','Sale'], values='Call_Count').plot(kind='bar',color='pink')
plt.title("Call_count based on Product_ID and Sale")
plt.ylabel('Call_Count')
plt.xlabel('Product_ID and Sale');
```





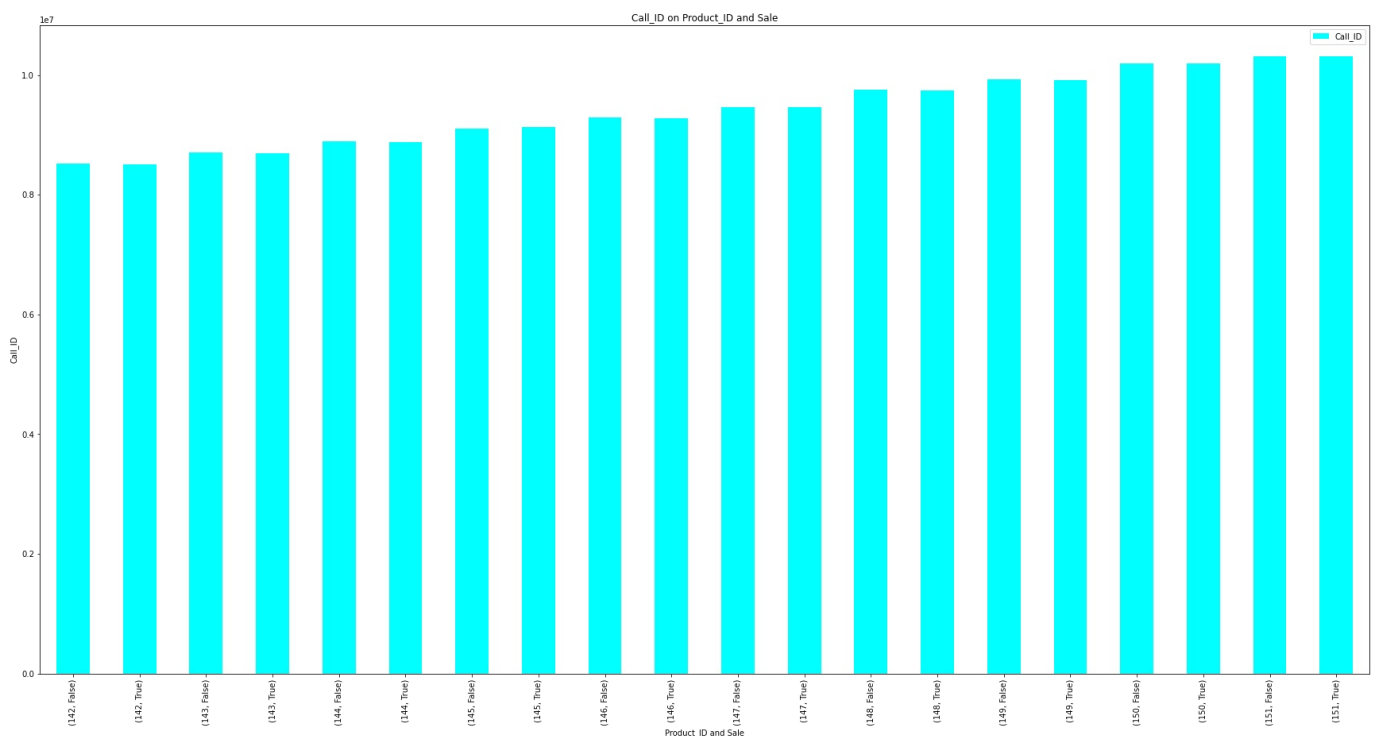
In [46]:

```
pd.pivot_table(df,index=['Product_ID','Sale'], values='Area_Code').plot(kind='bar',color='pink')
plt.title("Area_Code based on Product_ID and Sale")
plt.ylabel('Area_Code')
plt.xlabel('Product_ID and Sale');
```



In [47]:

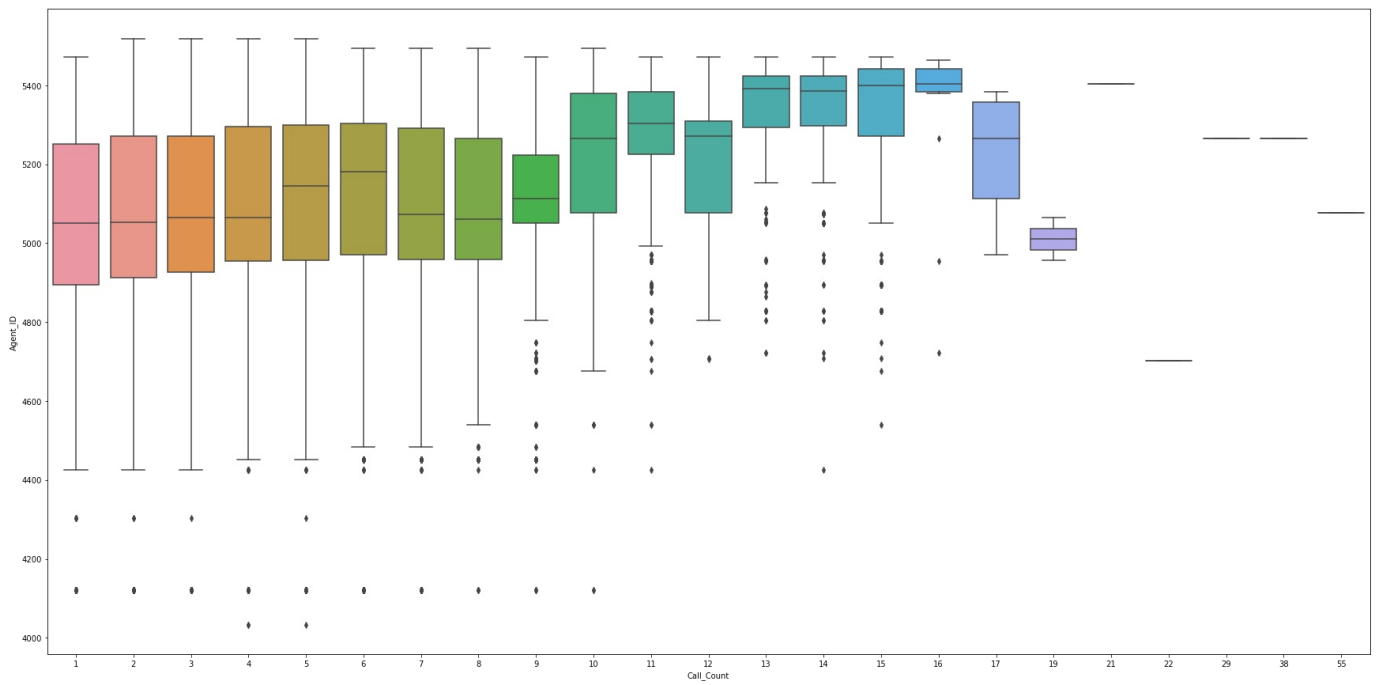
```
pd.pivot_table(df,index=['Product_ID','Sale'], values='Call_ID').plot(kind='bar',color='cyan')
plt.title("Call_ID on Product_ID and Sale")
plt.ylabel('Call_ID')
plt.xlabel('Product_ID and Sale');
```



In [48]:

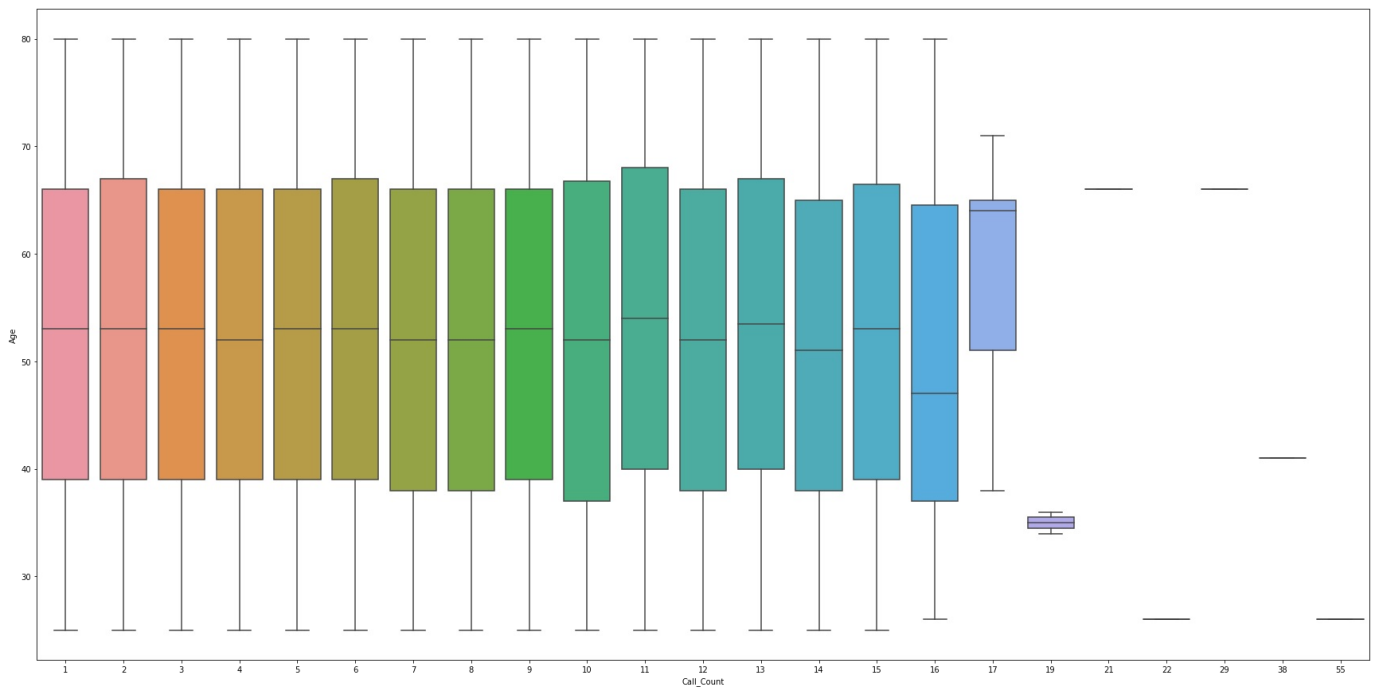
```
sns.boxplot(x='Call_Count', y='Agent_ID', data=df)
```

Out[48]: <AxesSubplot:xlabel='Call Count', ylabel='Agent ID'>



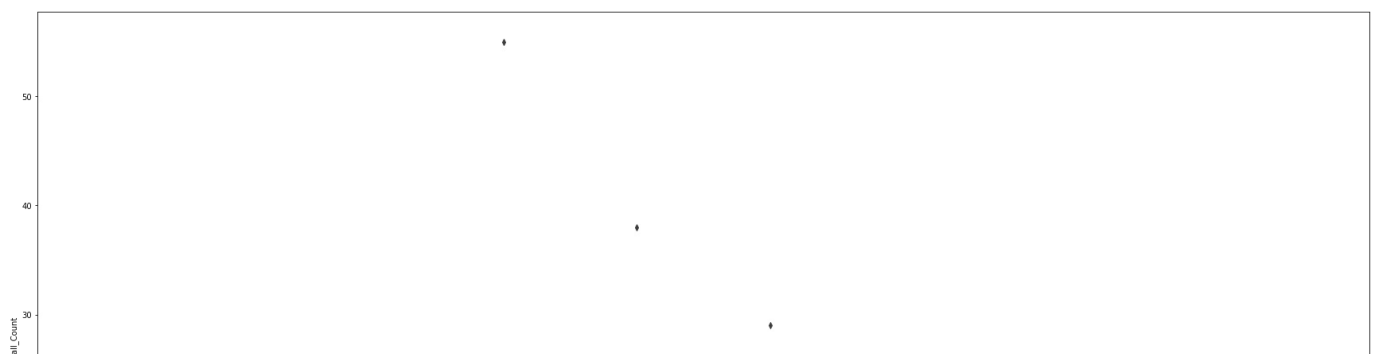
```
In [49]: sns.boxplot(x='Call_Count', y='Age', data=df)
```

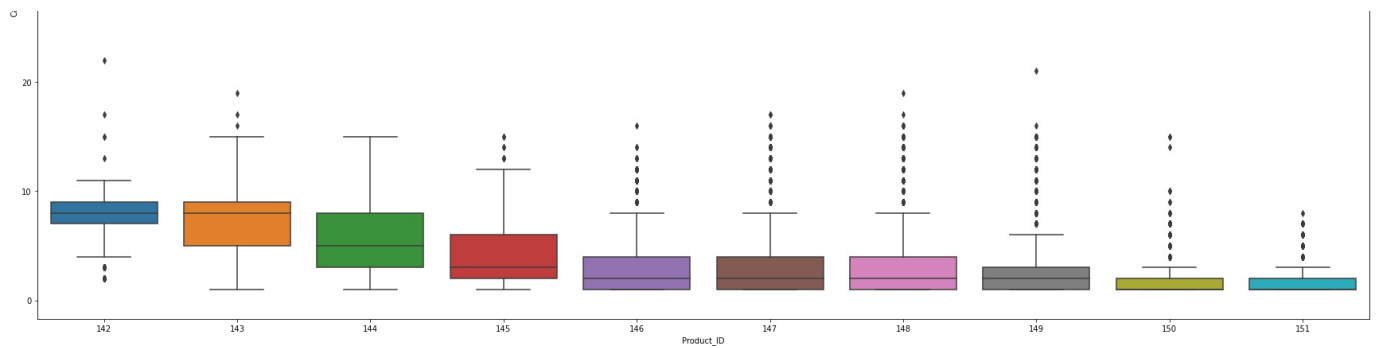
```
Out[49]: <AxesSubplot:xlabel='Call_Count', ylabel='Age'>
```



```
In [50]: sns.boxplot(x='Product_ID', y='Call_Count', data=df)
```

```
Out[50]: <AxesSubplot:xlabel='Product_ID', ylabel='Call_Count'>
```

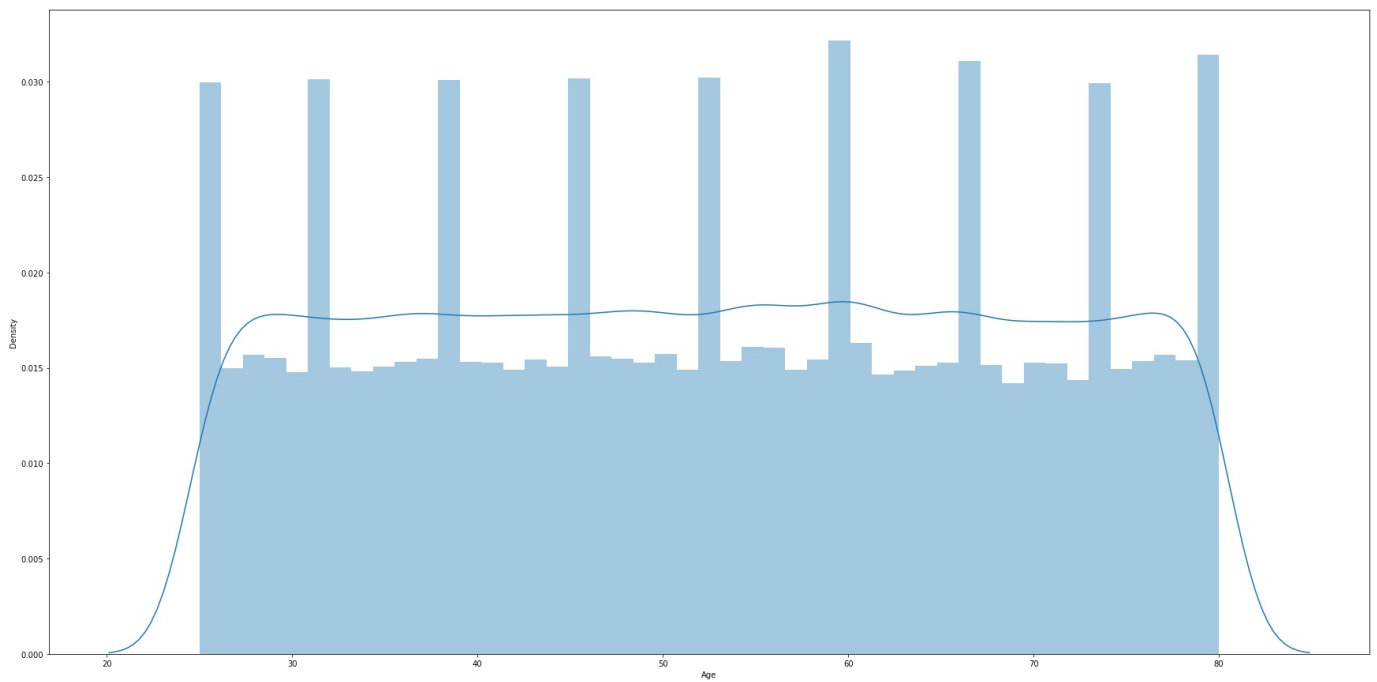




```
In [51]: import warnings
warnings.filterwarnings("ignore")

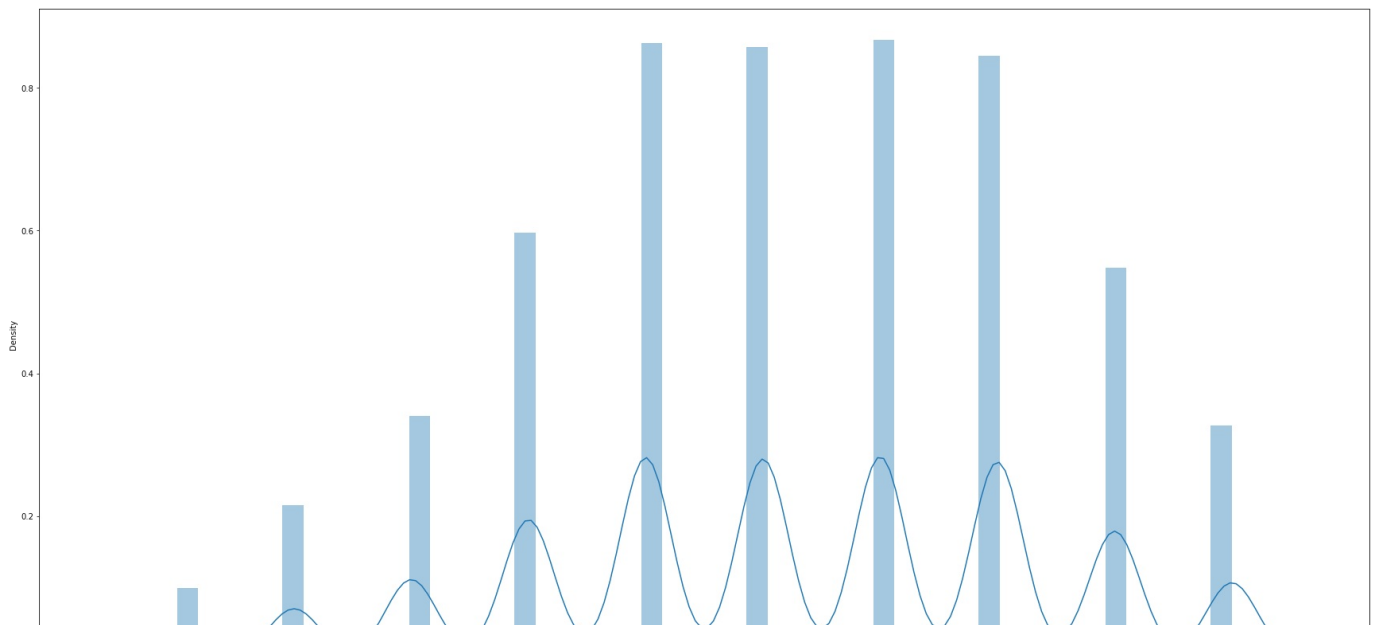
sns.distplot(df['Age'])
print('This distribution has skew' ,df['Age'].skew())
print('This distribution has kurtosis' ,df['Age'].kurt())
```

This distribution has skew -0.0027345736206543223
This distribution has kurtosis -1.1906311325526457



```
In [52]: sns.distplot(df['Product_ID'])
```

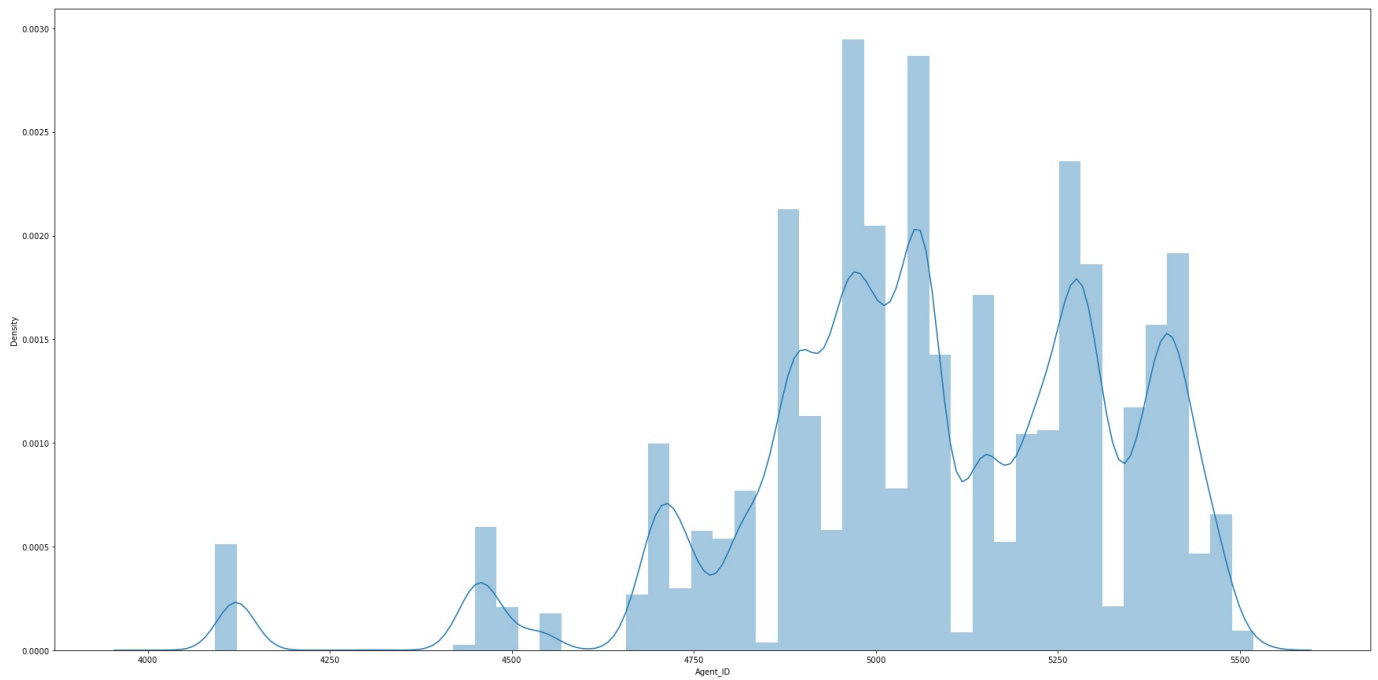
Out[52]: <AxesSubplot:xlabel='Product_ID', ylabel='Density'>





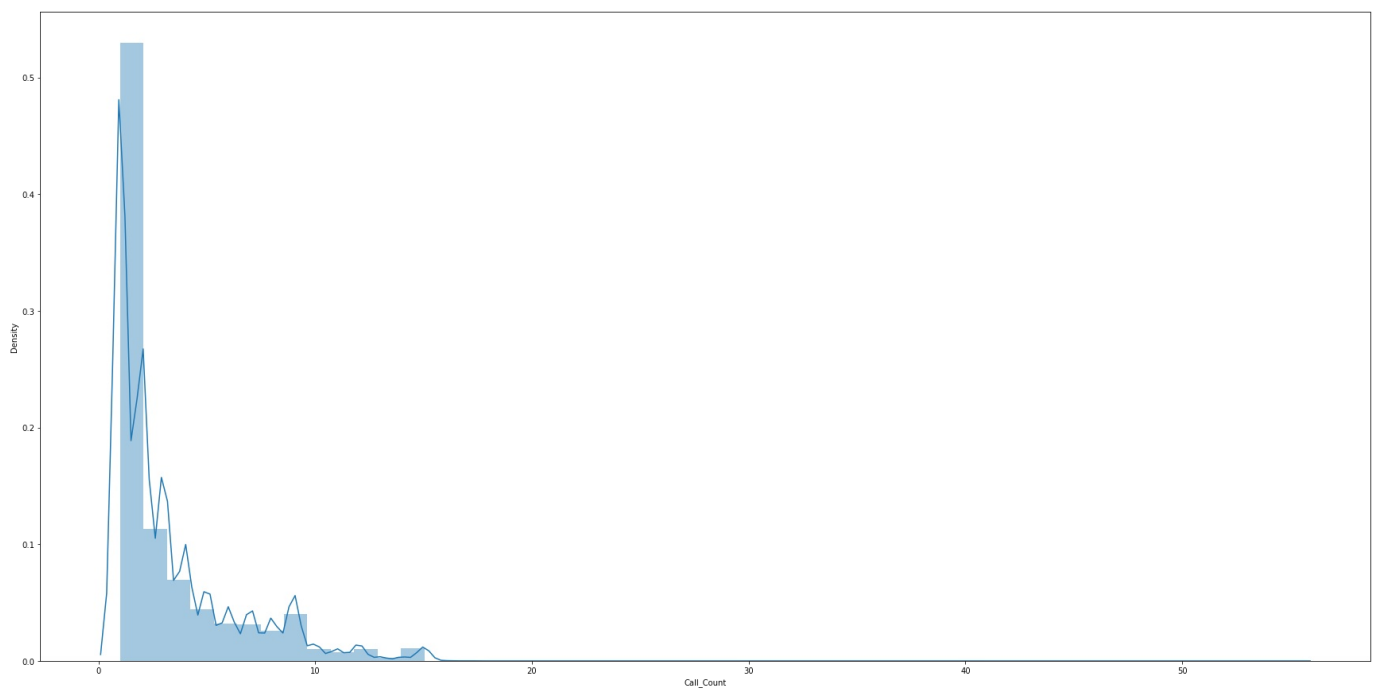
```
In [53]: sns.distplot(df['Agent_ID'])
```

```
Out[53]: <AxesSubplot:xlabel='Agent_ID', ylabel='Density'>
```



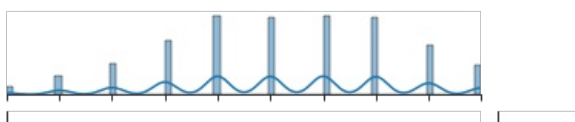
```
In [54]: sns.distplot(df['Call_Count'])
```

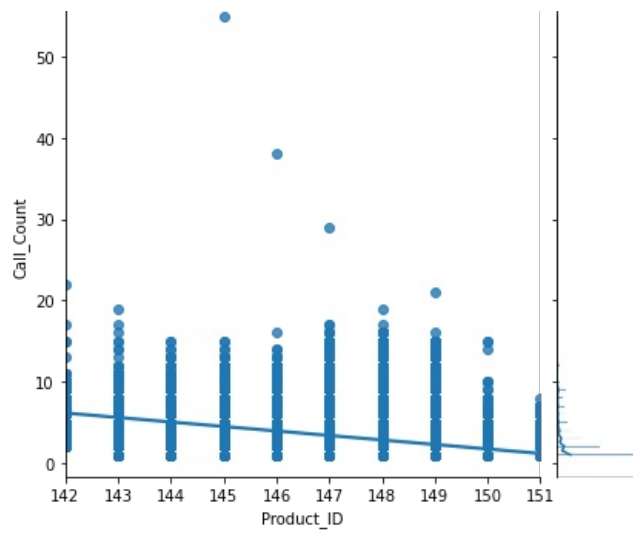
```
Out[54]: <AxesSubplot:xlabel='Call_Count', ylabel='Density'>
```



```
In [56]: sns.jointplot(x='Product_ID',y='Call_Count',data=df, kind='reg')
```

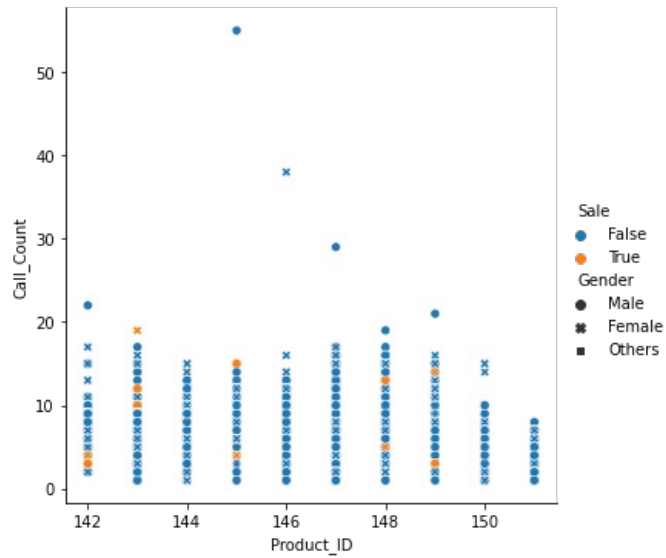
```
Out[56]: <seaborn.axisgrid.JointGrid at 0x16a88ea5e20>
```





```
In [57]: sns.relplot(x='Product_ID', y="Call_Count", data=df, hue="Sale", style='Gender')
```

```
Out[57]: <seaborn.axisgrid.FacetGrid at 0x16a88f3c220>
```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js