```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

file_path=(r"/content/Sales and Marketing Call Center.csv")
df=pd.read_csv(file_path)

df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 54732,\n  \"fields\":
[\n    {\n      \"column\": \"Call_ID\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 13,\n
\"samples\": [\n          \"na-Wi\",\n          \"el-Sm\",\n
\"el-Pa\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Date\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 105,\n        \"min\": 45292,\n        \"max\": 45657,\n
\"num_unique_values\": 366,\n        \"samples\": [\n          45431,\
n          45520,\n          45628\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Agent_First_Name\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 13,\n        \"samples\": [\n
\"Katrina\",\n          \"Samuel\",\n          \"Michael\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Agent_Last_Name\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 13,\n
\"samples\": [\n          \"Williams\",\n          \"Smith\",\n
\"Page\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Agent_Rating\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.0396959091303157,\n        \"min\":
0.0,\n        \"max\": 5.0,\n        \"num_unique_values\": 42,\n
\"samples\": [\n          3.1,\n          1.8,\n          4.6\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Product_Discussed\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"Internet Package\",\n          \"Electronics\",\n          \"Travel
Package\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Call_Duration_Minutes\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 12.449894966933968,\n
\"min\": 2.0,\n        \"max\": 45.0,\n        \"num_unique_values\":
4290,\n        \"samples\": [\n          41.66,\n          40.44,\n
27.74\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

```
\"Call_Outcome\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"Failure\",\n          \"Ab\",\n          \"Success\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Customer_Age\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
10.512408631522554,\n        \"min\": 14.0,\n        \"max\": 69.0,\n
\"num_unique_values\": 50,\n        \"samples\": [\n          30.0,\n
34.0,\n          36.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Callers_Name\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 41797,\n
\"samples\": [\n          \"Nicholas Barber MD\",\n          \"Rebecca
Nunez\",\n          \"Rachael Hardy\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Customer_Gender\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n
\"Female\",\n          \"Male\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"State\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\": 20,\
n        \"samples\": [\n          \"New York\",\n
\"Illinois\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Customer_Income_Bracket\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"High\",\n          \"Low\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Time_of_Day\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n
\"Afternoon\",\n          \"Morning\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Follow_Up_Call_Required\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          \"No\",\n
\"Yes\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Repeat_Customer\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Yes\",\n          \"No\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Reason_Call_Abandoned\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n          \"long
Wait Time\",\n          \"Long Wait Time\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54732 entries, 0 to 54731
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Call_ID                  54732 non-null  object 
 1   Date                     54732 non-null  int64  
 2   Agent_First_Name         54732 non-null  object 
 3   Agent_Last_Name          54732 non-null  object 
 4   Agent_Rating             42457 non-null  float64
 5   Product_Discussed        42457 non-null  object 
 6   Call_Duration_Minutes    42457 non-null  float64
 7   Call_Outcome             54732 non-null  object 
 8   Customer_Age             54731 non-null  float64
 9   Callers_Name             54731 non-null  object 
 10  Customer_Gender          54731 non-null  object 
 11  State                    54731 non-null  object 
 12  Customer_Income_Bracket  54731 non-null  object 
 13  Time_of_Day              54731 non-null  object 
 14  Follow_Up_Call_Required  54731 non-null  object 
 15  Repeat_Customer          54731 non-null  object 
 16  Reason_Call_Abandoned    31810 non-null  object 
dtypes: float64(3), int64(1), object(13)
memory usage: 7.1+ MB
```

```
df.describe()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n{\n      \"column\": \"Date\",\n      \"properties\": {\n\"dtype\": \"number\",\n        \"std\": 59096.93737475224,\n\"min\": 105.66797170061025,\n        \"max\": 200000.0,\n\"num_unique_values\": 8,\n        \"samples\": [\n45474.40702,\n          45475.0,\n          200000.0\n        ],\n\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Agent_Rating\",\n\"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 54925.44972455816,\n        \"min\": 0.0,\n        \"max\": 155356.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n4.254642884729266,\n          4.6,\n          155356.0\n        ],\n\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Call_Duration_Minutes\",\n\"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 54920.50117818238,\n        \"min\": 2.0,\n        \"max\": 155356.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n16.910974729009503,\n          12.24,\n          155356.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n}\n    },\n    {\n      \"column\": \"Customer_Age\",\n\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
```

70700.79195435153,\n   \"min\": 10.467033233601963,\n
\"max\": 200000.0,\n   \"num_unique_values\": 8,\n
\"samples\": [\n   26.316825,\n   25.0,\n
200000.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
df.isnull().sum()
```

```
Call_ID                        0
Date                           0
Agent_First_Name               0
Agent_Last_Name                0
Agent_Rating               44644
Product_Discussed          44644
Call_Duration_Minutes      44644
Call_Outcome                   0
Customer_Age                   0
Callers_Name                   0
Customer_Gender                0
State                          0
Customer_Income_Bracket        0
Time_of_Day                    0
Follow_Up_Call_Required        0
Repeat_Customer                0
Reason_Call_Abandoned      83519
dtype: int64
```

Data Cleaning

1. Who are the top-performing agents based on Agent Ratings?

```
df.head(2)
```

{"type":"dataframe","variable_name":"df"}

```
# Concating the first name and last name of agents

df['Agent Full Name']=df['Agent_First_Name']+' '+df['Agent_Last_Name']
df.head()
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 54732,\n \"fields\":
[\n   {\n   \"column\": \"Call_ID\",\n   \"properties\": {\n
\"dtype\": \"category\",\n   \"num_unique_values\": 13,\n
\"samples\": [\n   \"na-Wi\",\n   \"el-Sm\",\n
\"el-Pa\"\n   ],\n   \"semantic_type\": \"\",\n

```
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Date\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 105,\n        \"min\": 45292,\n        \"max\": 45657,\n
\"num_unique_values\": 366,\n        \"samples\": [\n          45431,\
n          45520,\n          45628\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Agent_First_Name\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 13,\n        \"samples\": [\n
\"Katrina\",\n          \"Samuel\",\n          \"Michael\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Agent_Last_Name\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 13,\n
\"samples\": [\n          \"Williams\",\n          \"Smith\",\n
\"Page\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Agent_Rating\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.0396959091303157,\n        \"min\":
0.0,\n        \"max\": 5.0,\n        \"num_unique_values\": 42,\n
\"samples\": [\n          3.1,\n          1.8,\n          4.6\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Product_Discussed\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"Internet Package\",\n          \"Electronics\",\n          \"Travel
Package\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Call_Duration_Minutes\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 12.449894966933968,\n
\"min\": 2.0,\n        \"max\": 45.0,\n        \"num_unique_values\":
4290,\n        \"samples\": [\n          41.66,\n          40.44,\n
27.74\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Call_Outcome\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"Failure\",\n          \"Ab\",\n          \"Success\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Customer_Age\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
10.512408631522554,\n        \"min\": 14.0,\n        \"max\": 69.0,\n
\"num_unique_values\": 50,\n        \"samples\": [\n          30.0,\n
34.0,\n          36.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Callers_Name\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 41797,\n
\"samples\": [\n          \"Nicholas Barber MD\",\n          \"Rebecca
Nunez\",\n          \"Rachael Hardy\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

```
n    },\n    {\n      \"column\": \"Customer_Gender\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n
\"Female\",\n          \"Male\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"State\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\": 20,\
n        \"samples\": [\n          \"New York\",\n
\"Illinois\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Customer_Income_Bracket\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"High\",\n          \"Low\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Time_of_Day\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n
\"Afternoon\",\n          \"Morning\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Follow_Up_Call_Required\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          \"No\",\n
\"Yes\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Repeat_Customer\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Yes\",\n          \"No\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Reason_Call_Abandoned\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n          \"long
Wait Time\",\n          \"Long Wait Time\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Agent Full Name\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 13,\n        \"samples\": [\n
\"Katrina Williams\",\n          \"Samuel Smith\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```


```
# Changing Agent_Rating values as numperic

df['Agent_Rating']=pd.to_numeric(df['Agent_Rating'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54732 entries, 0 to 54731
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
```

```
 0   Call_ID                   54732 non-null  object
 1   Date                      54732 non-null  int64
 2   Agent_First_Name          54732 non-null  object
 3   Agent_Last_Name           54732 non-null  object
 4   Agent_Rating              42457 non-null  float64
 5   Product_Discussed         42457 non-null  object
 6   Call_Duration_Minutes     42457 non-null  float64
 7   Call_Outcome              54732 non-null  object
 8   Customer_Age              54731 non-null  float64
 9   Callers_Name              54731 non-null  object
 10  Customer_Gender           54731 non-null  object
 11  State                     54731 non-null  object
 12  Customer_Income_Bracket   54731 non-null  object
 13  Time_of_Day               54731 non-null  object
 14  Follow_Up_Call_Required   54731 non-null  object
 15  Repeat_Customer           54731 non-null  object
 16  Reason_Call_Abandoned     31810 non-null  object
 17  Agent Full Name           54732 non-null  object
dtypes: float64(3), int64(1), object(14)
memory usage: 7.5+ MB
```

```
df[df['Agent_Rating'].isnull()]
```

{"repr_error":"0","type":"dataframe"}

```
# fill Nan value with zero

df['Agent_Rating']=df['Agent_Rating'].fillna(0)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54732 entries, 0 to 54731
Data columns (total 18 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Call_ID                   54732 non-null  object
 1   Date                      54732 non-null  int64
 2   Agent_First_Name          54732 non-null  object
 3   Agent_Last_Name           54732 non-null  object
 4   Agent_Rating              54732 non-null  float64
 5   Product_Discussed         42457 non-null  object
 6   Call_Duration_Minutes     42457 non-null  float64
 7   Call_Outcome              54732 non-null  object
 8   Customer_Age              54731 non-null  float64
 9   Callers_Name              54731 non-null  object
 10  Customer_Gender           54731 non-null  object
 11  State                     54731 non-null  object
 12  Customer_Income_Bracket   54731 non-null  object
 13  Time_of_Day               54731 non-null  object
 14  Follow_Up_Call_Required   54731 non-null  object
```

```
 15  Repeat_Customer          54731 non-null  object
 16  Reason_Call_Abandoned    31810 non-null  object
 17  Agent Full Name          54732 non-null  object
dtypes: float64(3), int64(1), object(14)
memory usage: 7.5+ MB
```

```
df['Agent_Rating'].isnull().sum()
```

```
np.int64(12275)
```

```
# Now finding after filtering Highest rating 5 agents and droping
duplicates
```

```
highest_rating=df[df['Agent_Rating']==4.0][['Agent Full
Name','Agent_Rating']].drop_duplicates()
highest_rating
```


```
{"summary":"{\n  \"name\": \"highest_rating\",\n  \"rows\": 13,\n
\"fields\": [\n    {\n      \"column\": \"Agent Full Name\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 13,\n        \"samples\": [\n
\"Michael Page\",\n          \"Olivia Lyons\",\n          \"Drew
Clay\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Agent_Rating\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.0,\n        \"min\": 4.0,\n
\"max\": 4.0,\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          4.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"highest_rating"}
```


```
# Create time series chart basis on above data
```

```
fig=px.bar(highest_rating,x='Agent Full
Name',y='Agent_Rating',title='Top-performing agents based on Agent
Ratings')
fig.show()
```

```
# based on above data create graph of agents who has recieved maximum
count of rating greater than 4.0
```

```
# Filter for ratings greater than 5.0
high_ratings_df = df[df['Agent_Rating'] == 5.0]

# Count the number of high ratings for each agent
agent_rating_counts = high_ratings_df['Agent Full
Name'].value_counts().reset_index()
agent_rating_counts.columns = ['Agent Full Name', 'Rating Count']

# Create the bar chart
fig = px.bar(agent_rating_counts, x='Agent Full Name', y='Rating
```

```
Count', title='Agents with Maximum Ratings (== 5.0)', text_auto=True,
color='Agent Full Name')
fig.show()
```

Inside and Observations from the highest count of rating 5 agents

From the above graph, we can say that Agent name Ava Sandoval has maximum count of highest rating received of 5 while minimum rating count of agent name Samuel Smith

2.What is the Average Call Duration by each Agent?

```
df.head(2)
```

```
{"type":"dataframe","variable_name":"df"}
```

```
# Converting Call_Duration_Minutes into numerical values

df['Call_Duration_Minutes']=pd.to_numeric(df['Call_Duration_Minutes'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54732 entries, 0 to 54731
Data columns (total 18 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Call_ID                  54732 non-null  object
 1   Date                     54732 non-null  int64
 2   Agent_First_Name         54732 non-null  object
 3   Agent_Last_Name          54732 non-null  object
 4   Agent_Rating             54732 non-null  float64
 5   Product_Discussed        42457 non-null  object
 6   Call_Duration_Minutes    42457 non-null  float64
 7   Call_Outcome             54732 non-null  object
 8   Customer_Age             54731 non-null  float64
 9   Callers_Name             54731 non-null  object
 10  Customer_Gender          54731 non-null  object
 11  State                    54731 non-null  object
 12  Customer_Income_Bracket  54731 non-null  object
 13  Time_of_Day              54731 non-null  object
 14  Follow_Up_Call_Required  54731 non-null  object
 15  Repeat_Customer          54731 non-null  object
 16  Reason_Call_Abandoned    31810 non-null  object
```

```
 17  Agent Full Name          54732 non-null  object
dtypes: float64(3), int64(1), object(14)
memory usage: 7.5+ MB
```

```
# filling Nan value with mean

df['Call_Duration_Minutes']=df['Call_Duration_Minutes'].fillna(df['Cal
l_Duration_Minutes'].mean())
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54732 entries, 0 to 54731
Data columns (total 18 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Call_ID                  54732 non-null  object
 1   Date                     54732 non-null  int64
 2   Agent_First_Name         54732 non-null  object
 3   Agent_Last_Name          54732 non-null  object
 4   Agent_Rating             54732 non-null  float64
 5   Product_Discussed        42457 non-null  object
 6   Call_Duration_Minutes    54732 non-null  float64
 7   Call_Outcome             54732 non-null  object
 8   Customer_Age             54731 non-null  float64
 9   Callers_Name             54731 non-null  object
 10  Customer_Gender          54731 non-null  object
 11  State                    54731 non-null  object
 12  Customer_Income_Bracket  54731 non-null  object
 13  Time_of_Day              54731 non-null  object
 14  Follow_Up_Call_Required  54731 non-null  object
 15  Repeat_Customer          54731 non-null  object
 16  Reason_Call_Abandoned    31810 non-null  object
 17  Agent Full Name          54732 non-null  object
dtypes: float64(3), int64(1), object(14)
memory usage: 7.5+ MB
```

```
df['Call_Duration_Minutes'].isnull().sum()
```

```
np.int64(0)
```

```
# Calculating the average call duration

avg_call_duration=df.groupby('Agent Full Name')
['Call_Duration_Minutes'].mean().reset_index()
avg_call_duration
```


```
{"summary":"{\n  \"name\": \"avg_call_duration\",\n  \"rows\": 13,\n
\"fields\": [\n    {\n      \"column\": \"Agent Full Name\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 13,\n        \"samples\": [\n          \"Sophia
Delacruz\",\n          \"Samuel Smith\",\n          \"Ava Sandoval\"\n
```

```
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Call_Duration_Minutes\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.4483276512515495,\n        \"min\": 15.070694639156086,\n
\"max\": 19.247327834541974,\n        \"num_unique_values\": 13,\n
\"samples\": [\n          17.67081912666338,\n
15.721060001264435,\n          17.706335918107534\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\
n}","type":"dataframe","variable_name":"avg_call_duration"}

# Sorting basis on Call Duration Minutes index

sorting_avg_call_duration=avg_call_duration.sort_values(by='Call_Durat
ion_Minutes',ascending=True)
sorting_avg_call_duration

{"summary":"{\n  \"name\": \"sorting_avg_call_duration\",\n  \"rows\":
13,\n  \"fields\": [\n    {\n      \"column\": \"Agent Full Name\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 13,\n        \"samples\": [\n
\"Katrina Williams\",\n          \"Sophia Delacruz\",\n
\"Elijah Hawkins\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Call_Duration_Minutes\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1.4483276512515495,\n
\"min\": 15.070694639156086,\n        \"max\": 19.247327834541974,\n
\"num_unique_values\": 13,\n        \"samples\": [\n
19.064235262381384,\n          17.67081912666338,\n
15.070694639156086\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"sorting_avg_call_duration"}

# Create Graph basis on this

fig=px.bar(sorting_avg_call_duration,x='Agent Full
Name',y='Call_Duration_Minutes',title='Average Call Duration by each
Agent',color='Agent Full Name', text_auto=True)
fig.update_layout(xaxis_title='Agent Full Name',yaxis_title='Average
Call Duration in Minutes')
fig.update_traces(texttemplate='%{y:.2f}')
fig.show()
```

Insights and Observations from Average Call Duration

From the above charts we can see that Agent Name Zoe Newman has minimum call duration avg 15.98 while Agent Name Michael Page has maximumm call duration avg 19.56

3.Which agents have the highest success rates in closing calls?

```
df.head(2)
```

```
{"type":"dataframe","variable_name":"df"}
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 18 columns):
 #   Column                   Non-Null Count   Dtype  
---  ------                   --------------   -----  
 0   Call_ID                  200000 non-null  object 
 1   Date                     200000 non-null  int64  
 2   Agent_First_Name         200000 non-null  object 
 3   Agent_Last_Name          200000 non-null  object 
 4   Agent_Rating             200000 non-null  float64
 5   Product_Discussed        155356 non-null  object 
 6   Call_Duration_Minutes    200000 non-null  float64
 7   Call_Outcome             200000 non-null  object 
 8   Customer_Age             200000 non-null  int64  
 9   Callers_Name             200000 non-null  object 
 10  Customer_Gender          200000 non-null  object 
 11  State                    200000 non-null  object 
 12  Customer_Income_Bracket  200000 non-null  object 
 13  Time_of_Day              200000 non-null  object 
 14  Follow_Up_Call_Required  200000 non-null  object 
 15  Repeat_Customer          200000 non-null  object 
 16  Reason_Call_Abandoned    116481 non-null  object 
 17  Agent Full Name          200000 non-null  object 
dtypes: float64(2), int64(2), object(14)
memory usage: 27.5+ MB
```

```
df['Call_Outcome'].unique()
```

```
array(['Success', 'Failure', 'Abandoned', 'Ab'], dtype=object)
```

```
successful_calls=df[df['Call_Outcome']=='Success']
```

```
# need to filter value basis on success only

successful_calls=df[df['Call_Outcome']=='Success']

# Calculating the success rate per Agent

success_rates=(successful_calls.groupby('Agent Full
Name').size()/df.groupby('Agent Full Name').size())
success_rates

# Converting decimal number to percentage strings

success_rates_percentage=success_rates.apply(lambda x: '{:.2f}
%'.format(x*100))

# reset index and rename column

success_rates=success_rates_percentage.reset_index()
success_rates.columns=['Agent Full Name','Success Rates']
success_rates

# Sorting the order of values success rates in dscending order

success_rates=success_rates.sort_values(by='Success
Rates',ascending=True)
success_rates
```


```
{"summary":"{\n  \"name\": \"success_rates\",\n  \"rows\": 13,\n
\"fields\": [\n    {\n      \"column\": \"Agent Full Name\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 13,\n        \"samples\": [\n
\"Monique Lawrence\",\n          \"Olivia Lyons\",\n          \"Elijah
Hawkins\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Success Rates\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 12,\n        \"samples\":
[\n          \"91.20%\",\n          \"90.15%\",\n          \"19.92%\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"success_rates"}
```


```
# Create Graph basis on above data

fig=px.bar(success_rates,x='Agent Full Name',y='Success
Rates',title='Agents with Highest Success Rates',color='Agent Full
Name',text_auto=True)
fig.update_layout(xaxis_title='Agent Full Name',yaxis_title='Success
Rates')
fig.update_traces(texttemplate='%{y}')
fig.show()
```

Insights and Observations from Agent Highest Sucess Rates

From the above graph we can observe that Samuel Smith has the highest sucess rates of 92.28% while Drew Clay has minimum success rates of 20.15%.

4. Are agent ratings correlated with call outcomes (e.g, higher ratings lead to more successful calls)?

```
df.head(2)
```

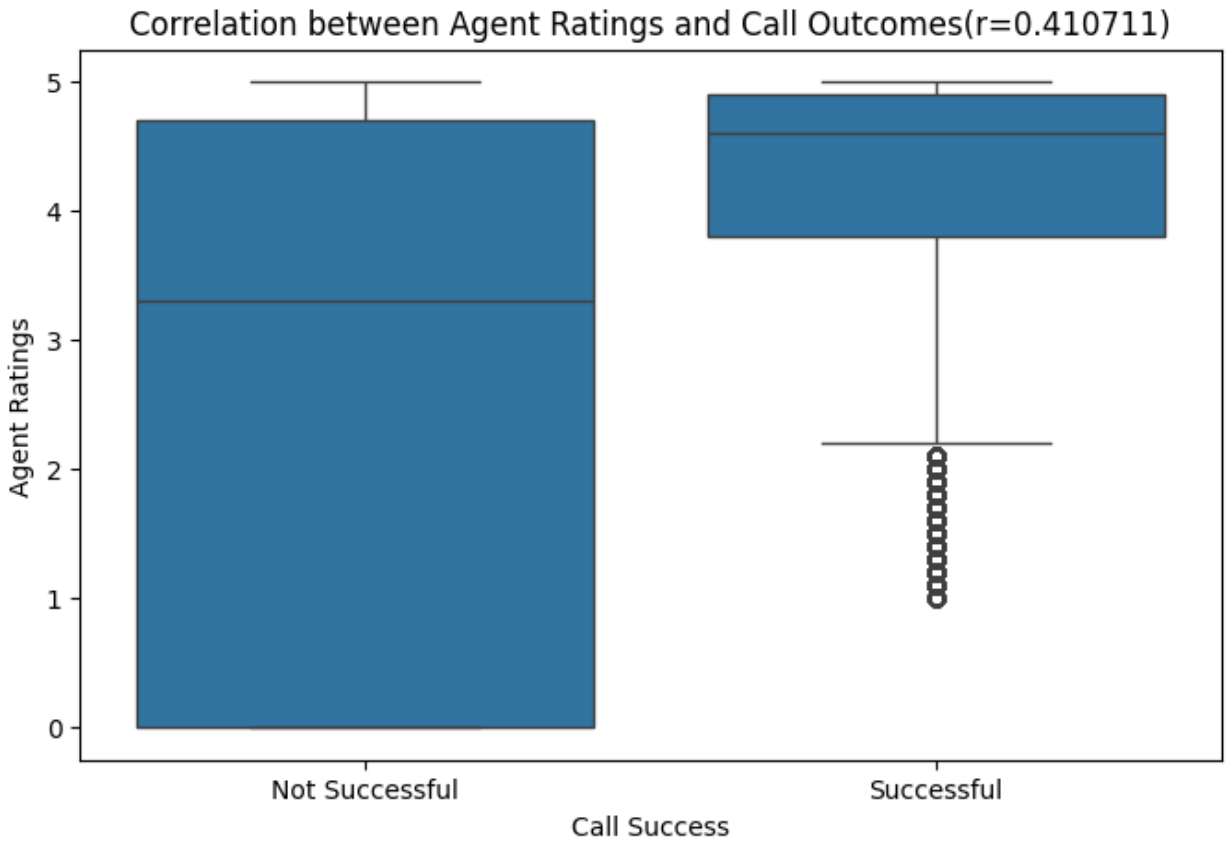

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 54732,\n  \"fields\":
[\n    {\n      \"column\": \"Call_ID\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 13,\n
\"samples\": [\n          \"na-Wi\",\n          \"el-Sm\",\n
\"el-Pa\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Date\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 105,\n        \"min\": 45292,\n        \"max\": 45657,\n
\"num_unique_values\": 366,\n        \"samples\": [\n          45431,\
n          45520,\n          45628\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Agent_First_Name\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 13,\n        \"samples\": [\n
\"Katrina\",\n          \"Samuel\",\n          \"Michael\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Agent_Last_Name\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 13,\n
\"samples\": [\n          \"Williams\",\n          \"Smith\",\n
\"Page\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Agent_Rating\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.9631938840447705,\n        \"min\":
0.0,\n        \"max\": 5.0,\n        \"num_unique_values\": 42,\n
\"samples\": [\n          3.6,\n          2.8,\n          4.8\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Product_Discussed\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"Internet Package\",\n          \"Electronics\",\n          \"Travel
Package\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

```
\"Call_Duration_Minutes\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 10.965249885607943,\n
\"min\": 2.0,\n        \"max\": 45.0,\n        \"num_unique_values\":
4291,\n        \"samples\": [\n          43.41,\n          8.73,\n
9.09\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Call_Outcome\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"Failure\",\n          \"Ab\",\n          \"Success\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Customer_Age\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
10.512408631522554,\n        \"min\": 14.0,\n        \"max\": 69.0,\n
\"num_unique_values\": 50,\n        \"samples\": [\n          30.0,\n
34.0,\n          36.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Callers_Name\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 41797,\n
\"samples\": [\n          \"Nicholas Barber MD\",\n          \"Rebecca
Nunez\",\n          \"Rachael Hardy\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Customer_Gender\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n
\"Female\",\n          \"Male\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"State\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\": 20,\
n        \"samples\": [\n          \"New York\",\n
\"Illinois\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Customer_Income_Bracket\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"High\",\n          \"Low\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Time_of_Day\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n
\"Afternoon\",\n          \"Morning\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Follow_Up_Call_Required\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          \"No\",\n
\"Yes\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Repeat_Customer\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Yes\",\n          \"No\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

```
n    },\n    {\n      \"column\": \"Reason_Call_Abandoned\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n          \"long
Wait Time\",\n          \"Long Wait Time\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Agent Full Name\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 13,\n        \"samples\": [\n
\"Katrina Williams\",\n          \"Samuel Smith\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Call_Success\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}

# Checking if higher ratings lead to more successful calls

# Creating a binary variable for call outcome from 1 to 0

df['Call_Success']=np.where(df['Call_Outcome']=='Success',1,0)
df.head()

# Calculating the correlation between Agent rating and Call success

correlation=df[['Agent_Rating','Call_Success']].corr().iloc[0,1]
correlation
```

```
np.float64(0.4107114824114599)
```

```
# Creating Boxplot basis on this correlation

plt.figure(figsize=(8,5))
sns.boxplot(x='Call_Success',y='Agent_Rating',data=df)
plt.title(f"Correlation between Agent Ratings and Call
Outcomes(r={correlation:2f})")
plt.xticks(ticks=[0,1],labels=['Not Successful','Successful'])
plt.xlabel('Call Success')
plt.ylabel('Agent Ratings')
plt.show()
```



Insights and Observation from Agent Rating and Highest call Success

The boxplot visually confirms the positive correlation between agent ratings and call success. The median rating for successful calls is significantly higher than for unsuccessful calls, and the entire distribution of ratings for successful calls is shifted upwards. This indicates that higher-rated agents are more likely to achieve successful call outcomes.

Summarization of the Analysis

This analysis of the call center data provides several key insights into agent performance and call outcomes.

Top Performing Agents:

- **Agent Ratings:** Ava Sandoval received the highest number of 5-star ratings, indicating strong customer satisfaction. However, Samuel Smith has the highest success rate.
- **Call Duration:** Michael Page has the longest average call duration, while Elijah Hawkins has the shortest. This could indicate that Michael is taking more time to resolve customer issues, or that he is less efficient.
- **Success Rates:** Samuel Smith has the highest success rate at 92.28%, while Drew Clay has the lowest at 20.15%. This is a significant difference and warrants further investigation.

Correlation between Agent Ratings and Call Outcomes:

- There is a positive correlation between agent ratings and call success. This suggests that agents who provide better customer service are more likely to achieve successful outcomes.

Actionable Recommendations:

- **Investigate Low-Performing Agents:** Further investigation is needed to understand why some agents, like Drew Clay, have such low success rates. This could be due to a lack of training, a difficult customer base, or other factors.
- **Share Best Practices:** High-performing agents, like Samuel Smith, should be encouraged to share their best practices with the rest of the team. This could include things like call scripts, objection handling techniques, and product knowledge.
- **Provide Additional Training:** All agents could benefit from additional training on topics like customer service, product knowledge, and sales techniques. This would help to improve the overall quality of service and increase the number of successful calls.
- **Incentivize High Performance:** The company should consider implementing an incentive program to reward agents for high performance. This could include bonuses, prizes, or other recognition. This would help to motivate agents and encourage them to provide the best possible service.