Project Report: ABC Call Volume Trend Analysis

Project Description:

This project involves analyzing a dataset of a Customer Experience (CX) inbound calling team for 23 days. The dataset includes information such as agent names, queue times, call durations, and call statuses. The objective of this project is to calculate the average call time duration for all incoming calls received by agents in each time bucket, show the total volume/number of calls coming in via charts/graphs, and propose a manpower plan required during each time bucket to reduce the abandon rate. Additionally, the project aims to propose a manpower plan required during each time bucket in a day to handle customer calls during the night. The analysis is performed using Jupyter Notebook, and the insights and findings are presented in a report to the leadership team.

Approach:

Steps Performed:

step1: Importing Necessary Libaries

step2: Loading csv file to DataFrame

step3:Data Cleaning

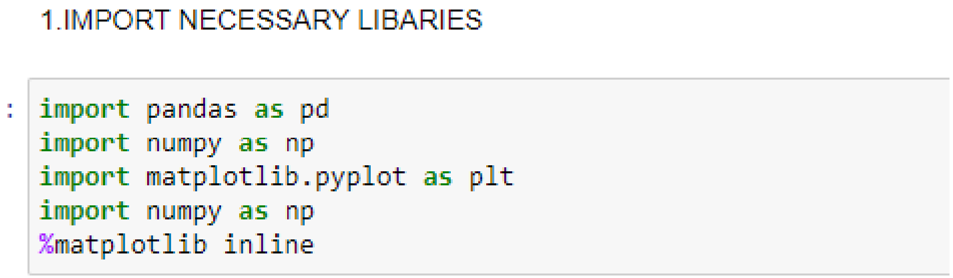
1)check dtypes

2)drop Duplicates

3)check null %

step4) EDA and Question's Answer

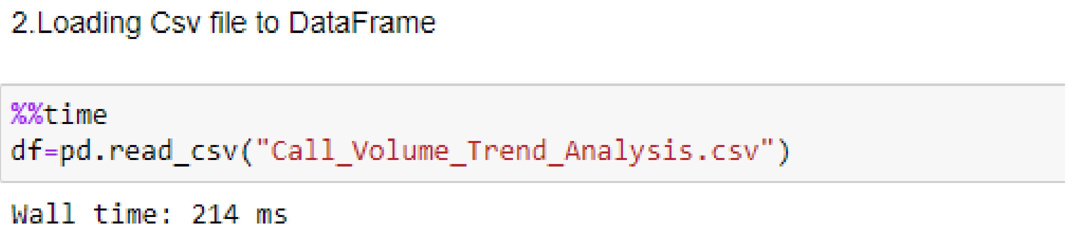
Step 1:



1.IMPORT NECESSARY LIBARIES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

Step 2:



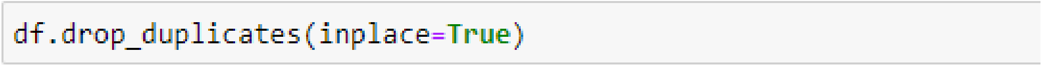
2.Loading Csv file to DataFrame

```
%%time
df=pd.read_csv("Call_Volume_Trend_Analysis.csv")
```

```
Wall time: 214 ms
```

Step 3:

Drop duplicates:

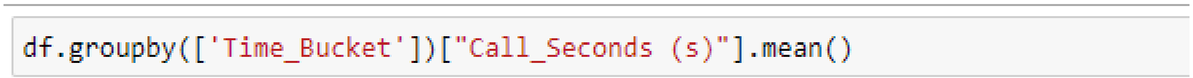


```
df.drop_duplicates(inplace=True)
```

We have also checked the data types of all columns and identified the presence of null values. The data types are correct and the null values are present only where calls are abandoned. Therefore, there is no need to fill in the null values for the project.

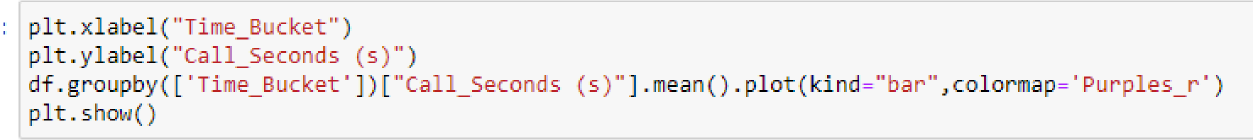
Step 4: The project has been approached by analyzing the data in Jupyter Notebook. The analysis is performed by answering the three questions asked in the project description. The first question is about calculating the average call time duration for all incoming calls received by agents (in each Time_Bucket). The second question is about showing the total volume/number of calls coming in via charts/graphs [Number of calls v/s Time]. The third question is about proposing a manpower plan required during each time bucket [between 9 am to 9 pm] to reduce the abandon rate to 10%. Additionally, a manpower plan is proposed to handle the calls that customers make during the night time between 9 pm to 9 am.

a)



```
df.groupby(['Time_Bucket'])["Call_Seconds (s)"].mean()
```

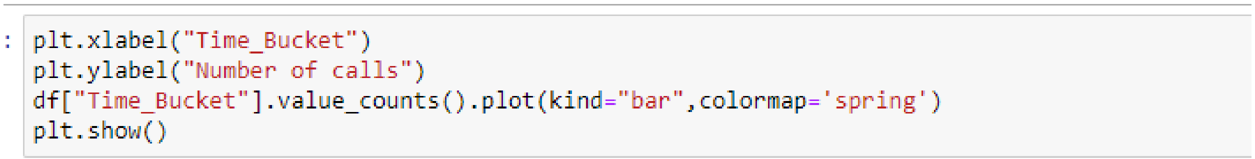
Bar Chart:-



```
: plt.xlabel("Time_Bucket")
  plt.ylabel("Call_Seconds (s)")
  df.groupby(['Time_Bucket'])["Call_Seconds (s)"].mean().plot(kind="bar",colormap='Purples_r')
  plt.show()
```

b)

Bar Chart:-



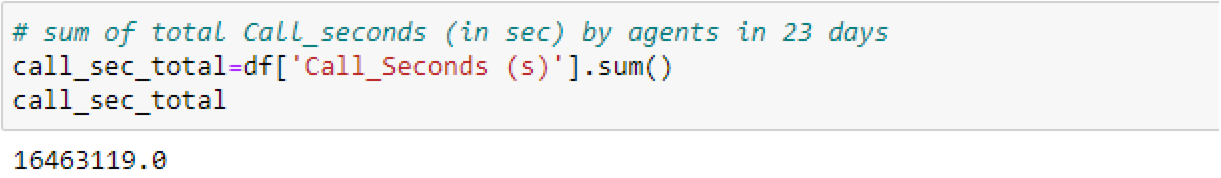
```
: plt.xlabel("Time_Bucket")
  plt.ylabel("Number of calls")
  df["Time_Bucket"].value_counts().plot(kind="bar",colormap='spring')
  plt.show()
```

c)

Based on assumption:-

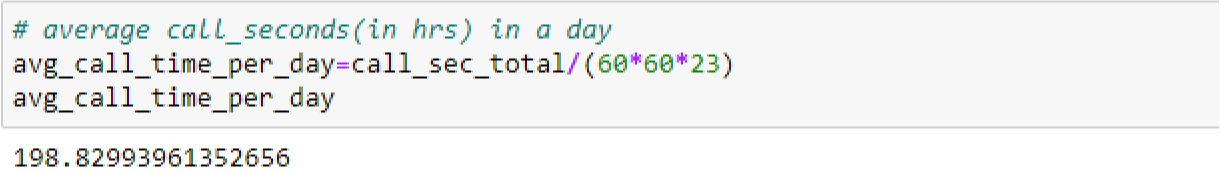
Agent Work Hour : 9 Hrs
Lunch Time : 1.5 Hrs
Actual Working Hrs : 7.5 Hrs
Agent on Call with Customers: 60% of 7.5Hrs=4.5Hrs

Calculation of Number of Agents required:



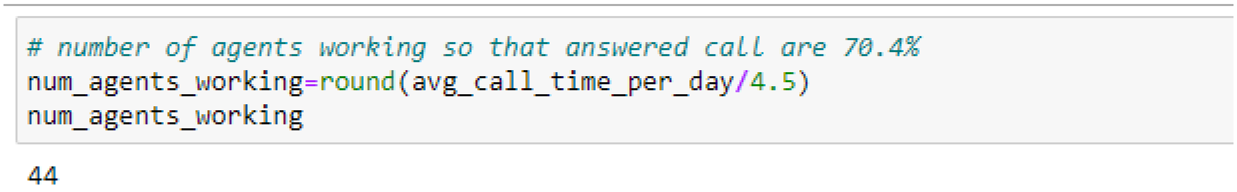
```
# sum of total Call_seconds (in sec) by agents in 23 days
call_sec_total=df['Call_Seconds (s)'].sum()
call_sec_total
```

```
16463119.0
```



```
# average call_seconds(in hrs) in a day
avg_call_time_per_day=call_sec_total/(60*60*23)
avg_call_time_per_day
```

```
198.82993961352656
```

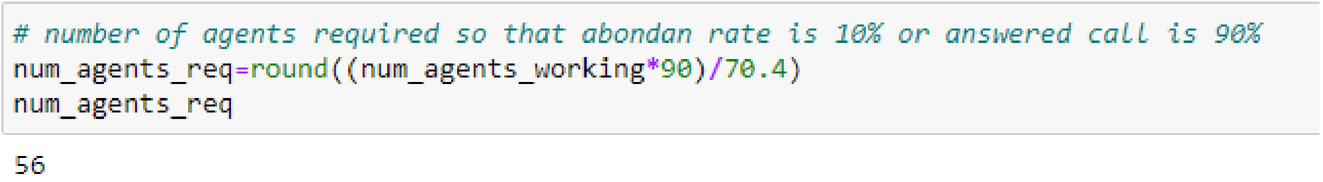


```
# number of agents working so that answered call are 70.4%
num_agents_working=round(avg_call_time_per_day/4.5)
num_agents_working
```

```
44
```

First, we calculate the total sum of Call_Seconds by agents. Then, we calculate the average daily Call_Seconds in hours. After that, we find the total number of agents working by dividing the average Call_Seconds (hours) by 4.5 (agents on call with customers: 60% of 7.5 hours = 4.5 hours).

-



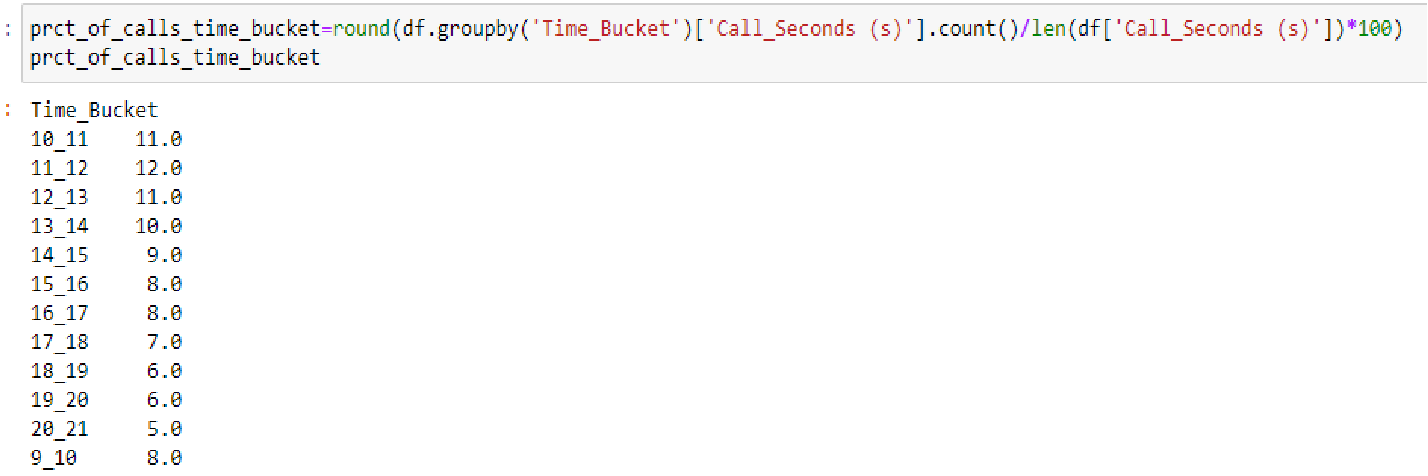
```
# number of agents required so that abondan rate is 10% or answered call is 90%
num_agents_req=round((num_agents_working*90)/70.4)
num_agents_req
```

```
56
```

If the number of agents working is 44 and they answer 70.4% of the calls, then the number of required agents to answer 90% of the calls should be approximately ~56.

Number of agents in each Time_Bucket(9am-9pm):

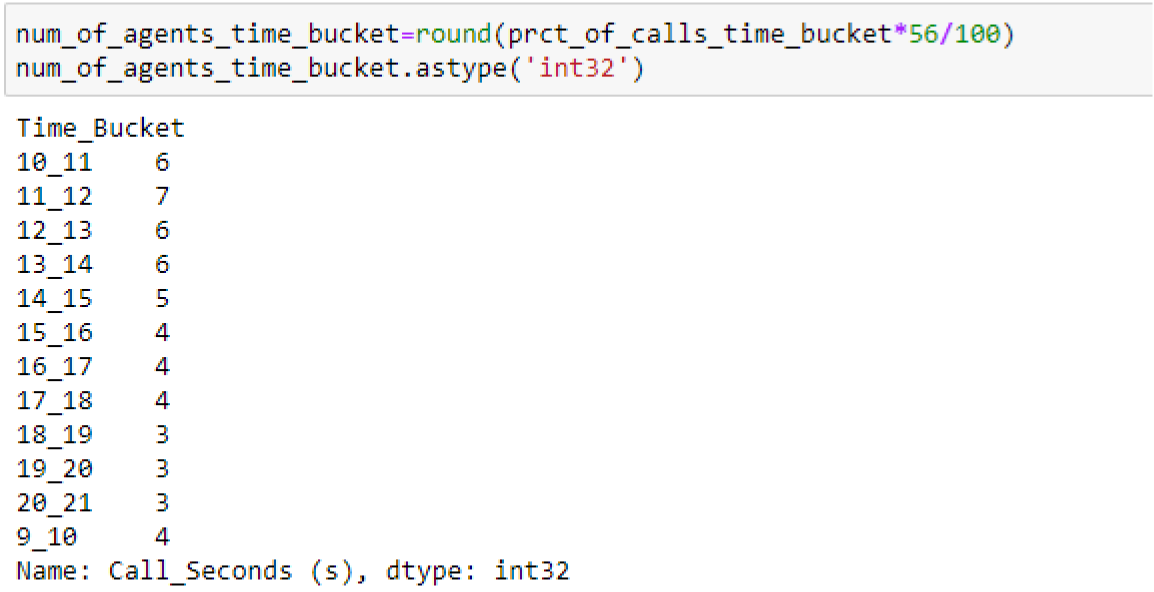
Below are the rounded percentages of calls in each time bucket. Based on these percentages, the number of agents required in each time bucket is calculated to achieve an abandonment rate of 10%.



```
: prct_of_calls_time_bucket=round(df.groupby('Time_Bucket')['Call_Seconds (s)'].count()/len(df['Call_Seconds (s)'])*100)
  prct_of_calls_time_bucket
```

```
: Time_Bucket
  10_11    11.0
  11_12    12.0
  12_13    11.0
  13_14    10.0
  14_15     9.0
  15_16     8.0
  16_17     8.0
  17_18     7.0
  18_19     6.0
  19_20     6.0
  20_21     5.0
  9_10      8.0
```

Number of agents based on % of calls in each time bucket.



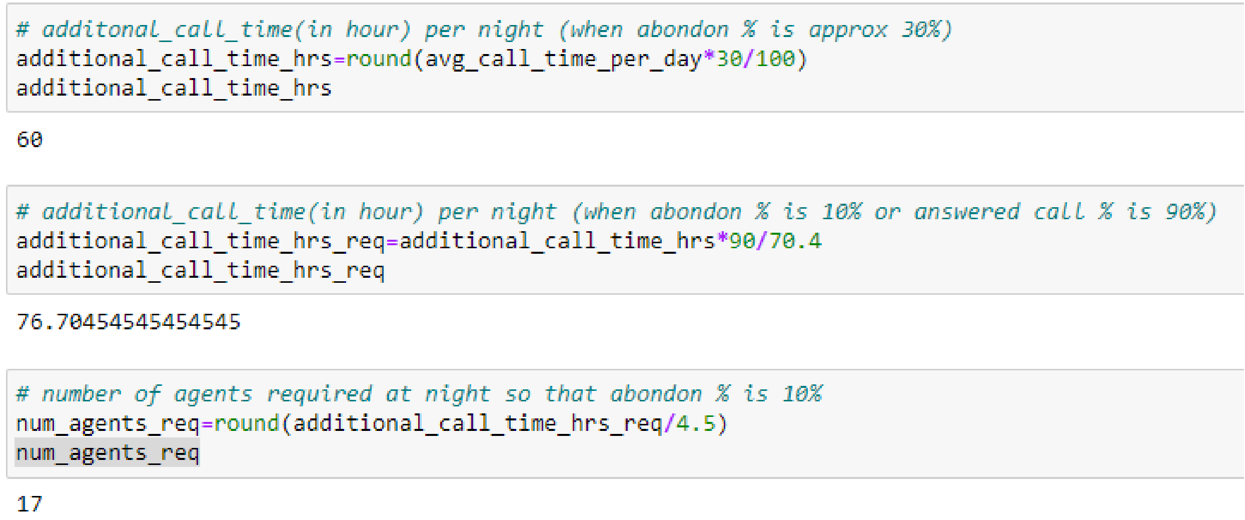
```
num_of_agents_time_bucket=round(prct_of_calls_time_bucket*56/100)
num_of_agents_time_bucket.astype('int32')
```

```
Time_Bucket
10_11    6
11_12    7
12_13    6
13_14    6
14_15    5
15_16    4
16_17    4
17_18    4
18_19    3
19_20    3
20_21    3
9_10     4
Name: Call_Seconds (s), dtype: int32
```

d)

Total numbers of calls daily(9am-9pm)= 5059 calls
Total numbers of calls per night(9pm-9am)= 1527(30% of daily day calls)

Agent on Call with Customers: 60% of 7.5Hrs=4.5Hrs(Based on assumption)



```
# additonal_call_time(in hour) per night (when abondon % is approx 30%)
additional_call_time_hrs=round(avg_call_time_per_day*30/100)
additional_call_time_hrs
```

```
60
```

```
# additional_call_time(in hour) per night (when abondon % is 10% or answered call % is 90%)
additional_call_time_hrs_req=additional_call_time_hrs*90/70.4
additional_call_time_hrs_req
```

```
76.70454545454545
```

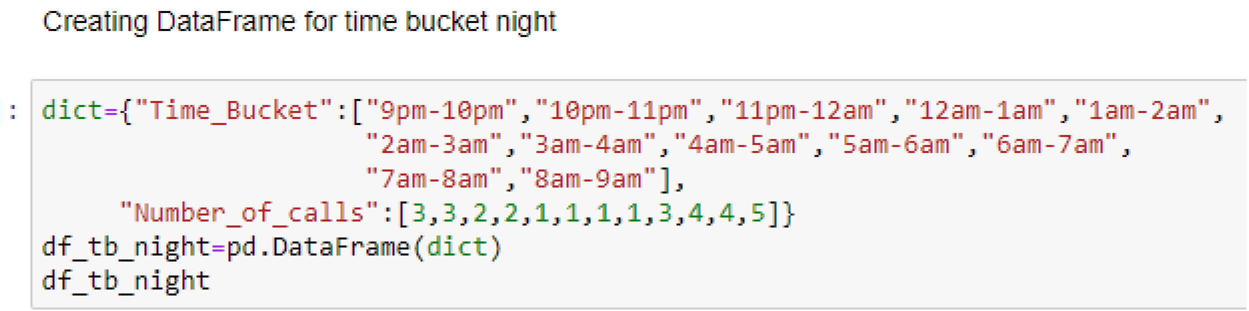
```
# number of agents required at night so that abondon % is 10%
num_agents_req=round(additional_call_time_hrs_req/4.5)
num_agents_req
```

```
17
```

First, we calculate the additional call time in hours per night, which should be 30% of the average call time during the day (as the number of calls is 30% of calls during the day). Then, using unitary method, we calculate that if the additional call time at night for 70.4% answered calls is 60 hours, then for 90% calls answered, it is 76.70 hours.

Number of agents required at night are calculated by dividing additional call time(Hrs) by 4.5.

Creating a DataFrame for Time_Bucket (9pm-9am):

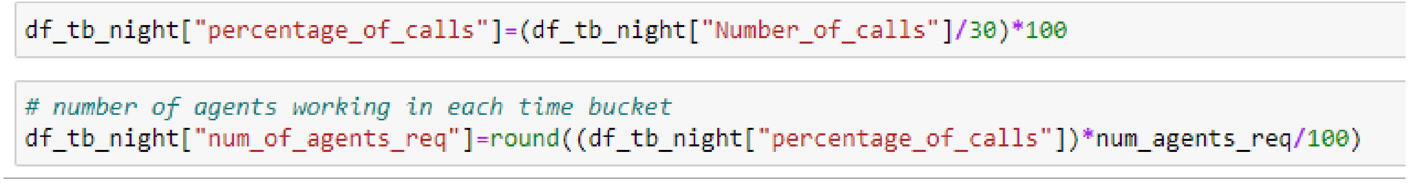


Creating DataFrame for time bucket night

```
dict={"Time_Bucket":["9pm-10pm","10pm-11pm","11pm-12am","12am-1am","1am-2am",
                     "2am-3am","3am-4am","4am-5am","5am-6am","6am-7am",
                     "7am-8am","8am-9am"],
     "Number_of_calls":[3,3,2,2,1,1,1,1,3,4,4,5]}
df_tb_night=pd.DataFrame(dict)
df_tb_night
```

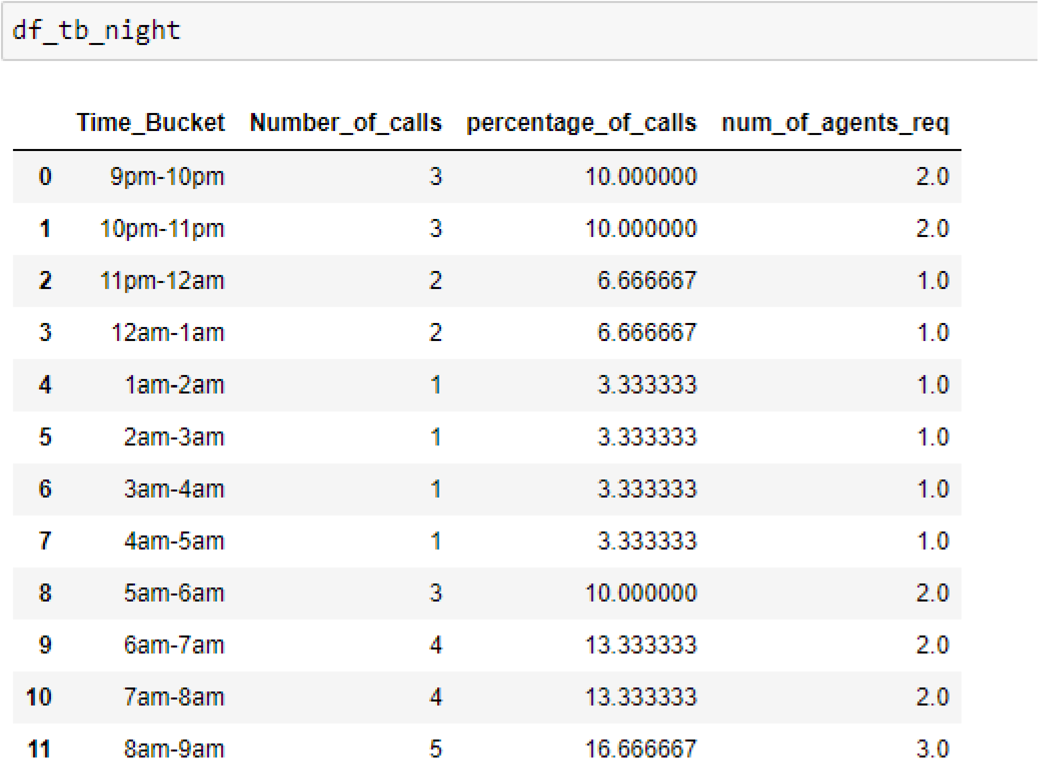
Number of agents in each Time_Bucket(9pm-9am):

First percentage of calls in each time bucket is calculated . Based on these percentages, the number of agents required in each time bucket is calculated to achieve an abandonment rate of 10%.



```
df_tb_night["percentage_of_calls"]=(df_tb_night["Number_of_calls"]/30)*100
```

```
# number of agents working in each time bucket
df_tb_night["num_of_agents_req"]=round((df_tb_night["percentage_of_calls"])*num_agents_req/100)
```



```
df_tb_night
```

	Time_Bucket	Number_of_calls	percentage_of_calls	num_of_agents_req
0	9pm-10pm	3	10.000000	2.0
1	10pm-11pm	3	10.000000	2.0
2	11pm-12am	2	6.666667	1.0
3	12am-1am	2	6.666667	1.0
4	1am-2am	1	3.333333	1.0
5	2am-3am	1	3.333333	1.0
6	3am-4am	1	3.333333	1.0
7	4am-5am	1	3.333333	1.0
8	5am-6am	3	10.000000	2.0
9	6am-7am	4	13.333333	2.0
10	7am-8am	4	13.333333	2.0
11	8am-9am	5	16.666667	3.0

Tech-Stack Used:

Jupyter Notebook:

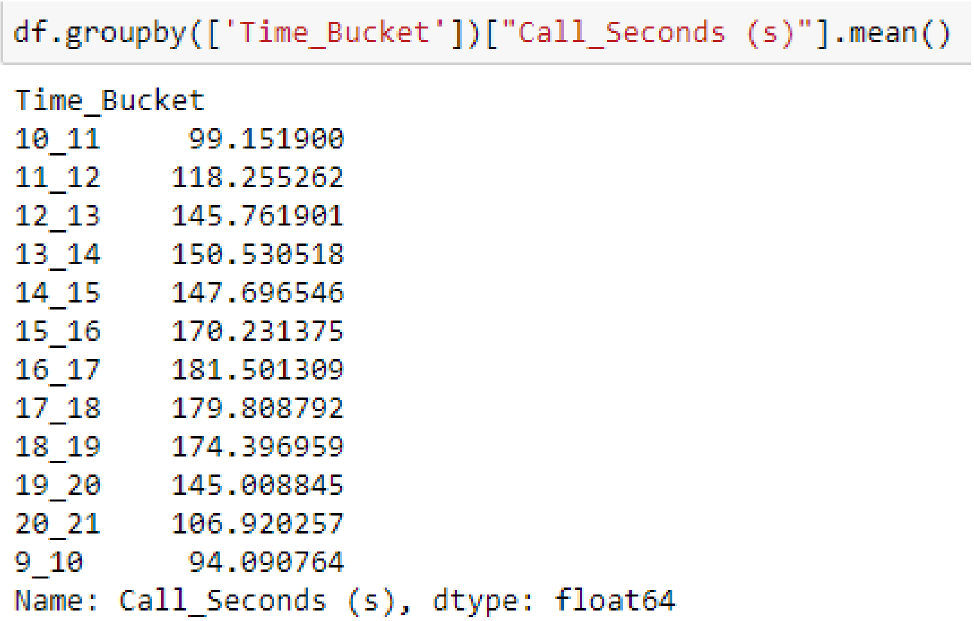
The analysis was performed using Jupyter Notebook, with Python and its libraries, such as Pandas, NumPy, Matplotlib, and Seaborn, utilized to perform the analysis.

MS Word 2010:

It is used to make a report (PDF) to be presented to the leadership team.

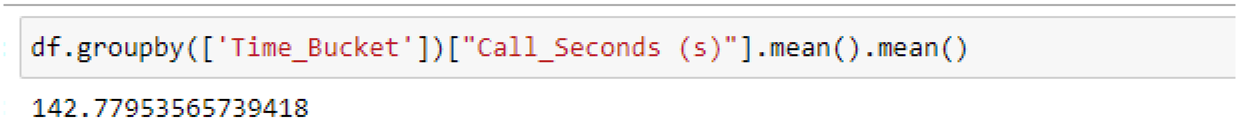
Insights:

a. Calculate the average call time duration for all incoming calls received by agents (in each Time_Bucket).



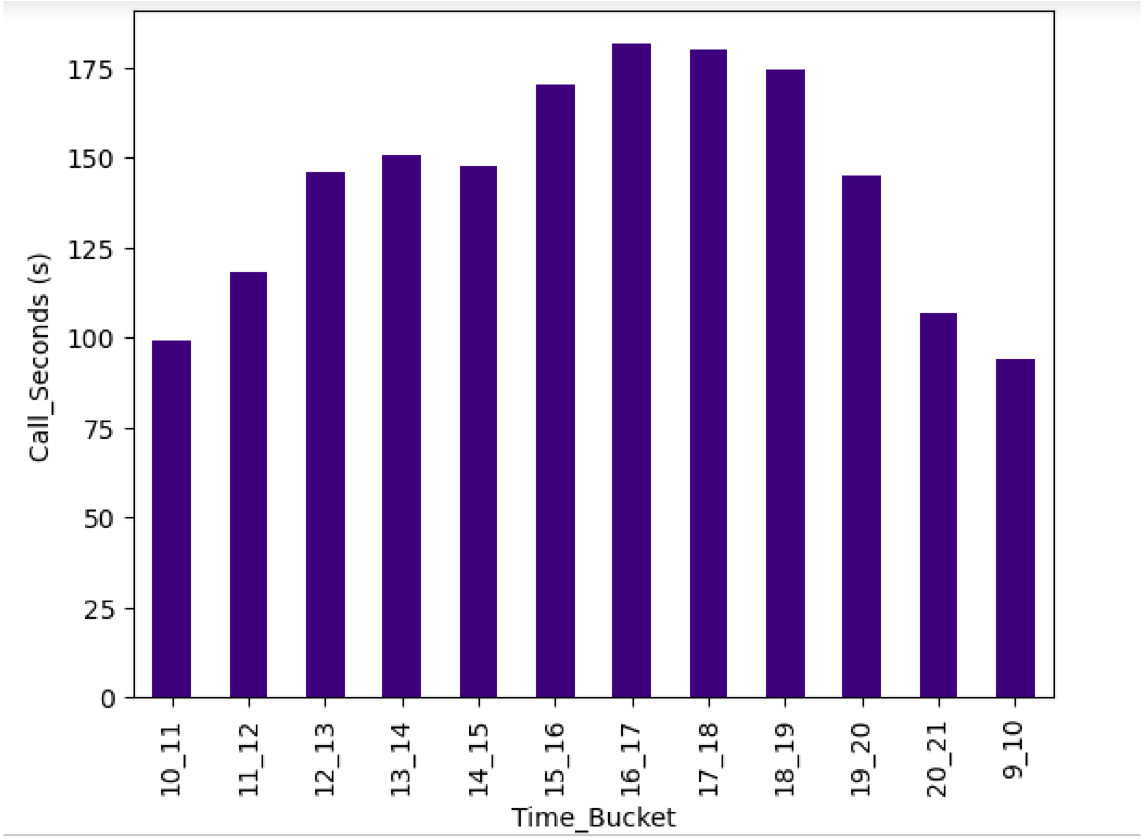
```
df.groupby(['Time_Bucket'])["Call_Seconds (s)"].mean()
```

```
Time_Bucket
10_11     99.151900
11_12    118.255262
12_13    145.761901
13_14    150.530518
14_15    147.696546
15_16    170.231375
16_17    181.501309
17_18    179.808792
18_19    174.396959
19_20    145.008845
20_21    106.920257
9_10      94.090764
Name: Call_Seconds (s), dtype: float64
```



```
df.groupby(['Time_Bucket'])["Call_Seconds (s)"].mean().mean()
```

```
142.77953565739418
```



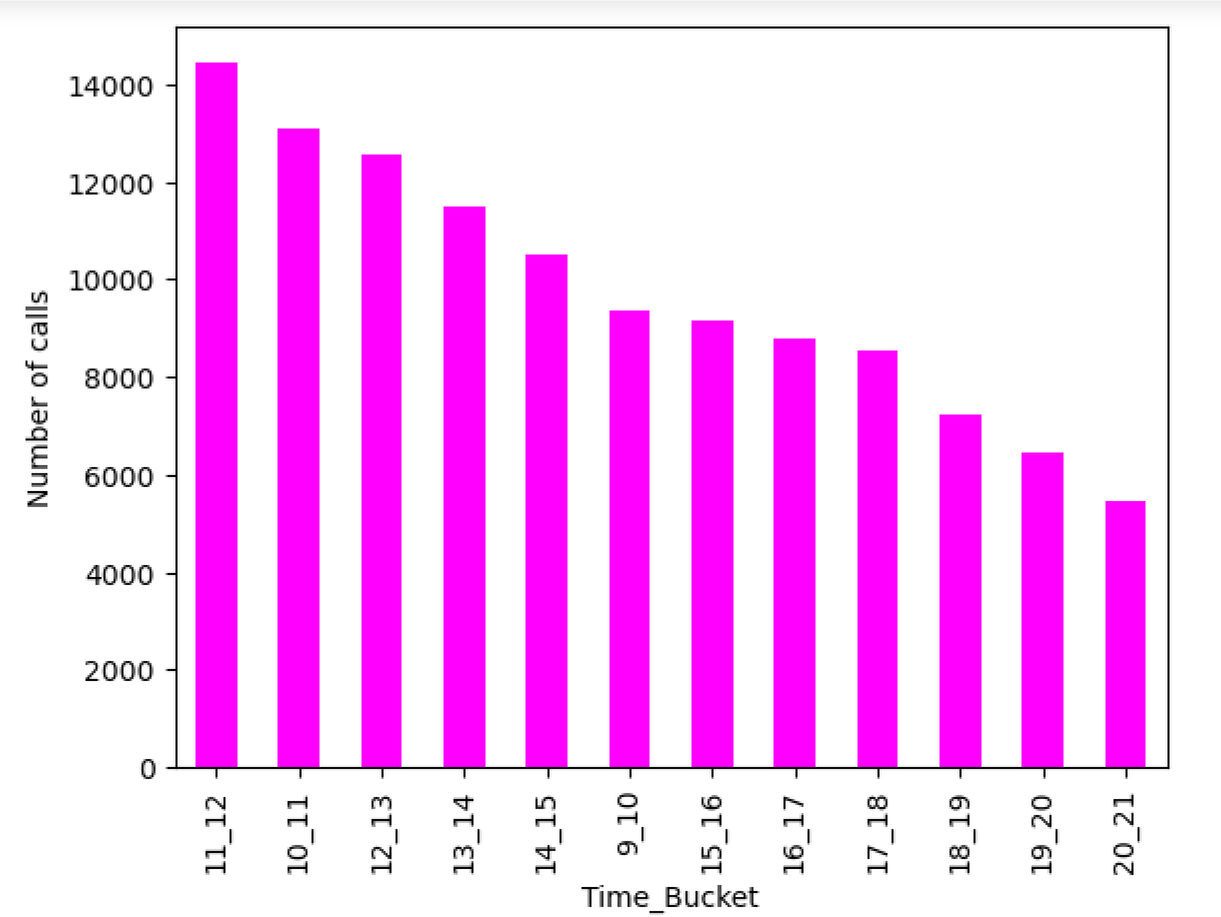
:-

The total average of call time duration which are answered by the agents is 142.77 seconds.

The average call time duration for all incoming calls received by agents is the highest in between 4 pm to 5 pm and from 5 pm to 6 pm

The average call time duration for all incoming calls received by agents is the least in between 9 am to 10 pm.

b. Show the total volume/ number of calls coming in via charts/ graphs [Number of calls v/s Time]. You can select time in a bucket form (i.e. 1-2, 2-3, …..)



:-

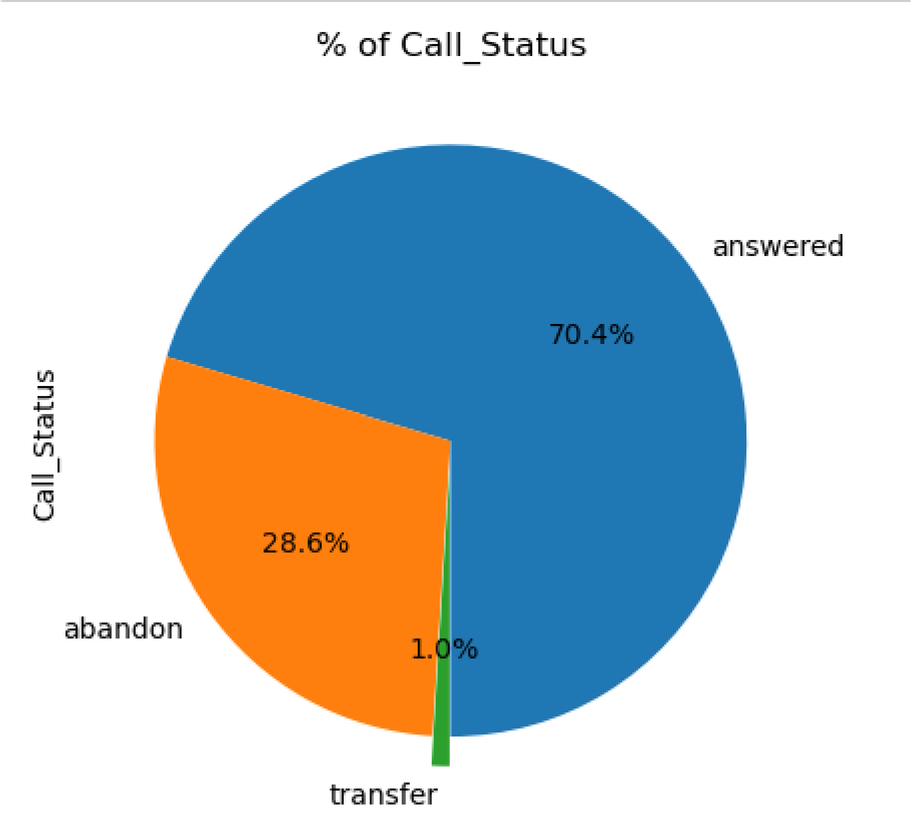
The customers call the most in between 11 am to 12 noon.

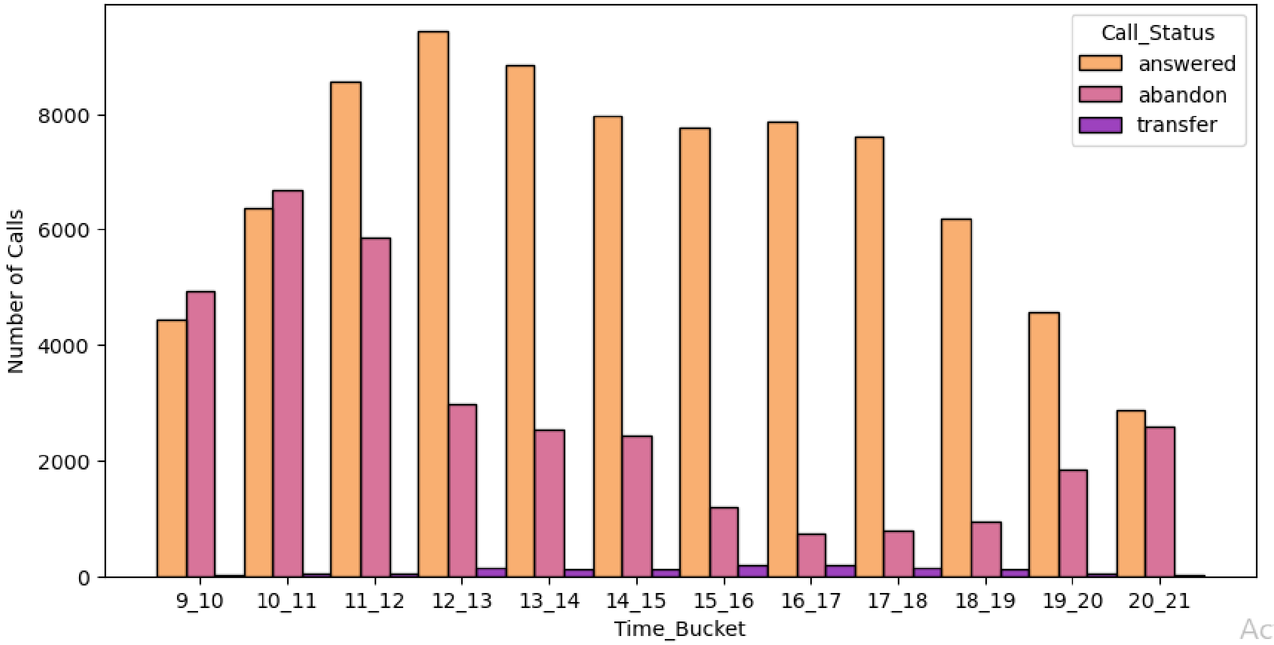
The customers call the least in between 8 pm to 9 pm.

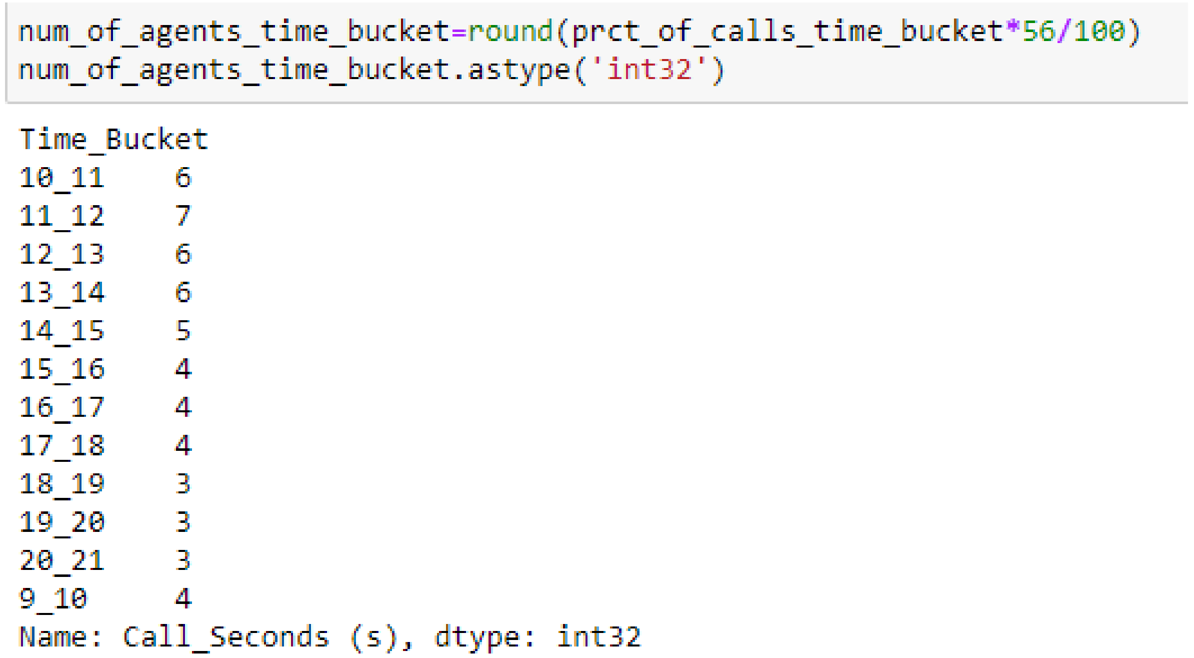
c. As you can see current abandon rate is approximately 30%. Propose a manpower plan required during each time bucket [between 9am to 9pm] to reduce the abandon rate to 10%. (i.e. You have to calculate minimum number of agents required in

each time bucket so that at least 90 calls should be answered out of 100.)

Assumption: An agent work for 6 days a week; On an average total unplanned leaves per agent is 4 days a month; An agent total working hrs is 9 Hrs out of which 1.5 Hrs goes into lunch and snacks in the office. On average an agent occupied for 60% of his total actual working Hrs (i.e 60% of 7.5 Hrs) on call with customers/ users. Total days in a month is 30 days.







```
num_of_agents_time_bucket=round(prct_of_calls_time_bucket*56/100)
num_of_agents_time_bucket.astype('int32')
```

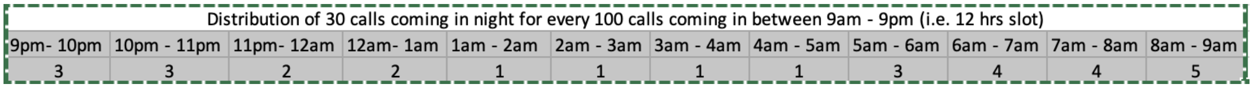
```
Time_Bucket
10_11    6
11_12    7
12_13    6
13_14    6
14_15    5
15_16    4
16_17    4
17_18    4
18_19    3
19_20    3
20_21    3
9_10     4
Name: Call_Seconds (s), dtype: int32
```

:-

Approx. 29% of the calls are abandoned, 1% is transferred, while 70% of the calls are answered from all incoming calls.

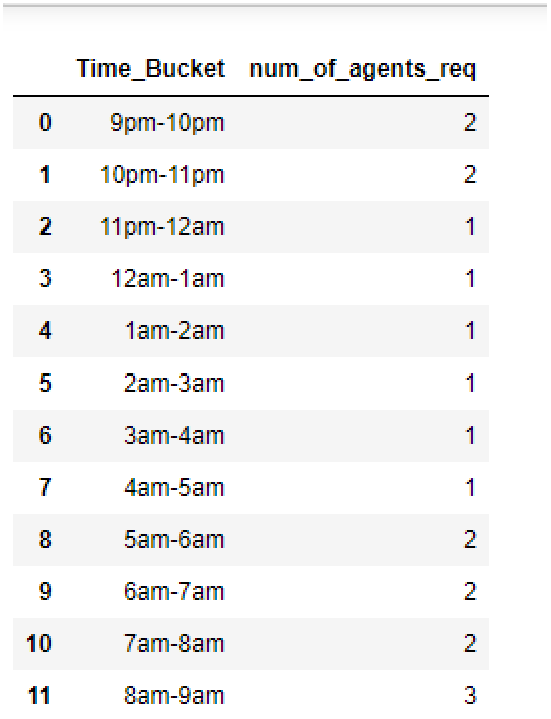
Total agents required to answer the 90% of the calls per day is 56.

d. **Let’s say customers also call this ABC insurance company in night but didn’t get answer as there are no agents to answer, this creates a bad customer experience for this Insurance company. Suppose every 100 calls that customer made during 9 Am to 9 Pm, customer also made 30 calls in night between interval [9 Pm to 9 Am] and distribution of those 30 calls are as follows:**



Distribution of 30 calls coming in night for every 100 calls coming in between 9am - 9pm (i.e. 12 hrs slot)

9pm- 10pm	10pm - 11pm	11pm- 12am	12am- 1am	1am - 2am	2am - 3am	3am - 4am	4am - 5am	5am - 6am	6am - 7am	7am - 8am	8am - 9am
3	3	2	2	1	1	1	1	3	4	4	5



	Time_Bucket	num_of_agents_req
0	9pm-10pm	2
1	10pm-11pm	2
2	11pm-12am	1
3	12am-1am	1
4	1am-2am	1
5	2am-3am	1
6	3am-4am	1
7	4am-5am	1
8	5am-6am	2
9	6am-7am	2
10	7am-8am	2
11	8am-9am	3

:-

There are least number of calls in the night so , the number of agents required to answer calls are also less.

The number of agents needed for night shift work are 17.

Number of calls in between 12am to 5am are least.

The company can shift some of the day workers to the night shift to ensure that there are enough agents available during peak call times. They can also propose a plan to make agents available during the busiest hours of customer calls so that maximum calls can be answered and agents are available 24/7.

Result:

In conclusion, the proposed manpower plan can help ABC Insurance Company to reduce the abandon rate and improve customer experience. However, the actual implementation of the plan would require further analysis and consideration of additional factors such as agent availability, training, and performance metrics.

Overall, the results of an ABC call volume trend analysis can provide valuable insights for call center managers and help them to optimize their operations for improved customer satisfaction and efficiency.

In this project, I learned how analysts can make an impact in the customer service department and how companies deal with customers to provide them with the utmost satisfaction. I gained knowledge about IVR Duration, an AI tool that answers calls and identifies the customer's exact question and then transfers it to the right agent to answer their queries. Additionally, I learned about behavioral analytics.