

ANUSHKA GHARAGE

PROJECT 8: ABC Call Volume Trend Analysis

Project Description:

The project focused on analyzing the call volume trends of the ABC Insurance Company's Customer Experience (CX) team over a 23-day period. The primary objectives were to understand the distribution of calls throughout the day, identify patterns in call durations, analyze agent performance, and propose recommendations for optimizing manpower allocation. The dataset included various details such as agent information, queue time, call duration, call status, and more.

Approach:

- Starting with data exploration and understanding.
- After exploring the dataset, I conducted descriptive statistics, time series analysis, agent-level analysis, and call status breakdown.
- Techniques such as sorting, filtering, and pivot tables were used to perform the analyses.
- Visualization tools were employed to create charts and graphs for better presentation of insights.

Tech-Stack Used:

- **Microsoft Excel 2016:** Excel was chosen for its versatility in handling tabular data, performing calculations, and creating visualizations.
- **Python:** Used for data analysis and understanding.

Insights:

- The analysis revealed several insights into the ABC Insurance Company's call volume trends.
- Key findings included identifying peak call times, understanding the distribution of call durations, and evaluating agent performance.
- The impact of time buckets, call status, and individual agent efficiency were crucial in understanding the overall performance of the CX team.

Result:

- Through the project, a comprehensive understanding of the ABC Call Volume Trend Analysis was achieved.
- The insights obtained provided actionable recommendations for improving manpower allocation, optimizing agent efficiency, and enhancing overall customer experience.
- The project contributed to a better understanding of the dynamics of the inbound calling team and paved the way for data-driven decisions to improve the company's customer service operations.

TASK 1: What is the average duration of calls for each time bucket?

Time_Bucket	Average Call Duration(sec)
9_10	199.0057078
10_11	209.146173
11_12	203.3522837
12_13	190.7582262
13_14	193.5996208
14_15	193.196702
15_16	195.4858855
16_17	196.542932
17_18	198.7127846
18_19	201.0931383
19_20	203.9389068
20_21	202.5537849
TOTAL AVG	198.9488455

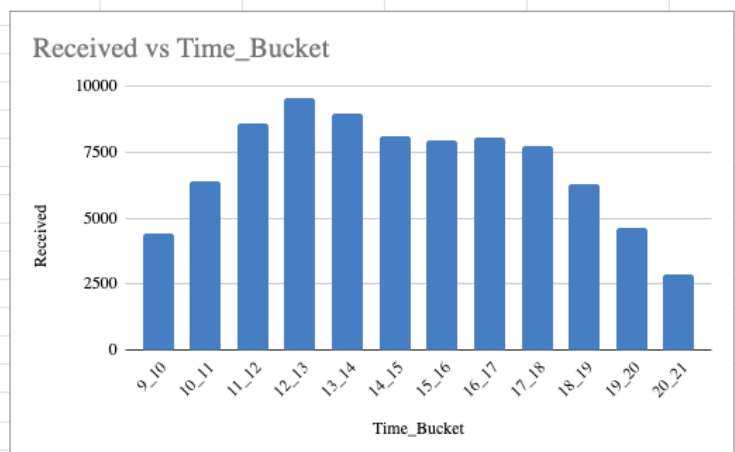
TASK 1

➤ The average call duration is around 198-199 seconds, that is, 3 minutes 20 secs.

TASK 2: Create a chart or graph that shows the number of calls received in each time bucket.

Time_Bucket	Calls Received	Incoming Calls
9_10	4433	9588
10_11	6391	13313
11_12	8589	14626
12_13	9565	12652
13_14	8937	11561
14_15	8076	10561
15_16	7937	9159
16_17	8027	8788
17_18	7742	8534
18_19	6298	7238
19_20	4608	6463
20_21	2873	5505
		117988

TASK 2



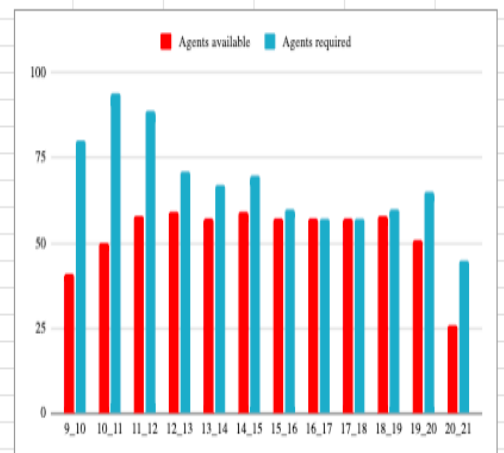
➤ The number calls received from the incoming calls are much less during 9-11am and 7-9pm.

TASK 3: What is the minimum number of agents required in each time bucket to reduce the abandon rate to 10%?

✓ <code>dff = df[df['Time_Bucket']=='9_10'] ...</code>	✓ <code>df13 = df[df['Time_Bucket']=='13_14'] ...</code>	✓ <code>df17 = df[df['Time_Bucket']=='17_18'] ...</code>
✓ <code>dff['Agent_Name'].nunique() ...</code>	✓ <code>df13['Agent_Name'].nunique() ...</code>	✓ <code>df17['Agent_Name'].nunique() ...</code>
41	57	57
✓ <code>df10 = df[df['Time_Bucket']=='10_11'] ...</code>	✓ <code>df14 = df[df['Time_Bucket']=='14_15'] ...</code>	✓ <code>df18 = df[df['Time_Bucket']=='18_19'] ...</code>
✓ <code>df10['Agent_Name'].nunique() ...</code>	✓ <code>df14['Agent_Name'].nunique() ...</code>	✓ <code>df18['Agent_Name'].nunique() ...</code>
50	59	58
✓ <code>df11 = df[df['Time_Bucket']=='11_12'] ...</code>	✓ <code>df15 = df[df['Time_Bucket']=='15_16'] ...</code>	✓ <code>df19 = df[df['Time_Bucket']=='19_20'] ...</code>
✓ <code>df11['Agent_Name'].nunique() ...</code>	✓ <code>df15['Agent_Name'].nunique() ...</code>	✓ <code>df19['Agent_Name'].nunique() ...</code>
58	57	51
✓ <code>df12 = df[df['Time_Bucket']=='12_13'] ...</code>	✓ <code>df16 = df[df['Time_Bucket']=='16_17'] ...</code>	✓ <code>df20 = df[df['Time_Bucket']=='20_21'] ...</code>
✓ <code>df12['Agent_Name'].nunique() ...</code>	✓ <code>df16['Agent_Name'].nunique() ...</code>	✓ <code>df20['Agent_Name'].nunique() ...</code>
59	57	26

Received	Incoming	Call Answer Rate	Desired Call Received	DesiredReceived - RECEIVED	Agents available	Agents required	Time_Bucket
4433	9588	0.4623487693	8630	4197	41	80	9_10
6391	13313	0.4800570871	11982	5591	50	94	10_11
8589	14626	0.587241898	13164	4575	58	89	11_12
9565	12652	0.7560069554	11387	1822	59	71	12_13
8937	11561	0.7730300147	10405	1468	57	67	13_14
8076	10561	0.7647003125	9505	1429	59	70	14_15
7937	9159	0.8665793209	8244	307	57	60	15_16
8027	8788	0.9134046427	7910	-117	57	57	16_17
7742	8534	0.9071947504	7681	-61	57	57	17_18
6298	7238	0.8701298701	6515	217	58	60	18_19
4608	6463	0.7129815875	5817	1209	51	65	19_20
2873	5505	0.5218891916	4955	2082	26	45	20_21
83476	117988	0.7179637					

TASK 3



- ★ Call Answer Rate = ~70%
- ★ Call Abandon Rate = ~30%
- ★ Desired Call Abandon Rate = 10%

➤ Therefore agents required will increase in order to decrease abandon rate.

TASK 4: Propose a manpower plan for each time bucket throughout the day, keeping the maximum abandon rate at 10%.

[illegible]

- Assuming that for every 100 calls that customers make between 9 am and 9 pm, they also make 30 calls at night between 9 pm and 9 am.
- Therefore, for 117988 incoming calls between 9am-9pm, there are 3540 calls between 9pm-9am.
- Agents required throughout the day depends on the time bucket. More agents are required during the and comparatively less during night.