Project by – Alokk Joshi

ABC Call Volume Trend Analysis

Link to Python File

Link to Project Excel File

Ctrl+Click the above links

Project by – Alokk Joshi

Project Description:
ABC Call Volume Trend Analysis

This project focuses on **Customer Experience (CX) analytics**, specifically analyzing **inbound call volume trends** in an insurance company's call center. The dataset provided spans **23 days** and contains details like **agent ID, queue time, call time, call duration, and call status** (answered, abandoned, or transferred).

The objective is to analyze call trends, agent workload, and manpower planning to optimize customer service and reduce abandoned calls.

Project by – Alokk Joshi

Tasks Required in the Project:

1. **Average Call Duration Analysis**
   Calculate the **average call duration for each time bucket**.
2. **Call Volume Analysis**
   Create **charts/graphs** to visualize the **total number of calls received per time bucket**.
3. **Manpower Planning (Day Shift: 9 AM - 9 PM)**
   Determine the **minimum number of agents required per time bucket** to reduce the **abandoned call rate from 30% to 10%**.
4. **Night Shift Manpower Planning (9 PM - 9 AM)**
   Since **30% of the day's call volume occurs at night**, propose a **manpower plan** to ensure a **maximum 10% abandon rate** at night.

Tools applied for analytics:

1. Python
2. Excel 2021

Project by – Alokk Joshi

Task 1

1. **Average Call Duration:** Determine the average duration of all incoming calls received by agents. This should be calculated for each time bucket.

Project 8 – ABC Call Volume Trend Analysis

Project by – Alokk Joshi

```
avg_call_duration = df.groupby('time_bucket')['call_seconds_s'].mean()
avg_call_duration = avg_call_duration.astype(int).sort_values(ascending = True)
avg_call_duration.sort_values(ascending = True)
```

time_bucket	call_seconds_s
9_10	92
10_11	97
20_21	105
11_12	116
12_13	144
19_20	144
14_15	146
13_14	149
15_16	169
18_19	174
17_18	179
16_17	181

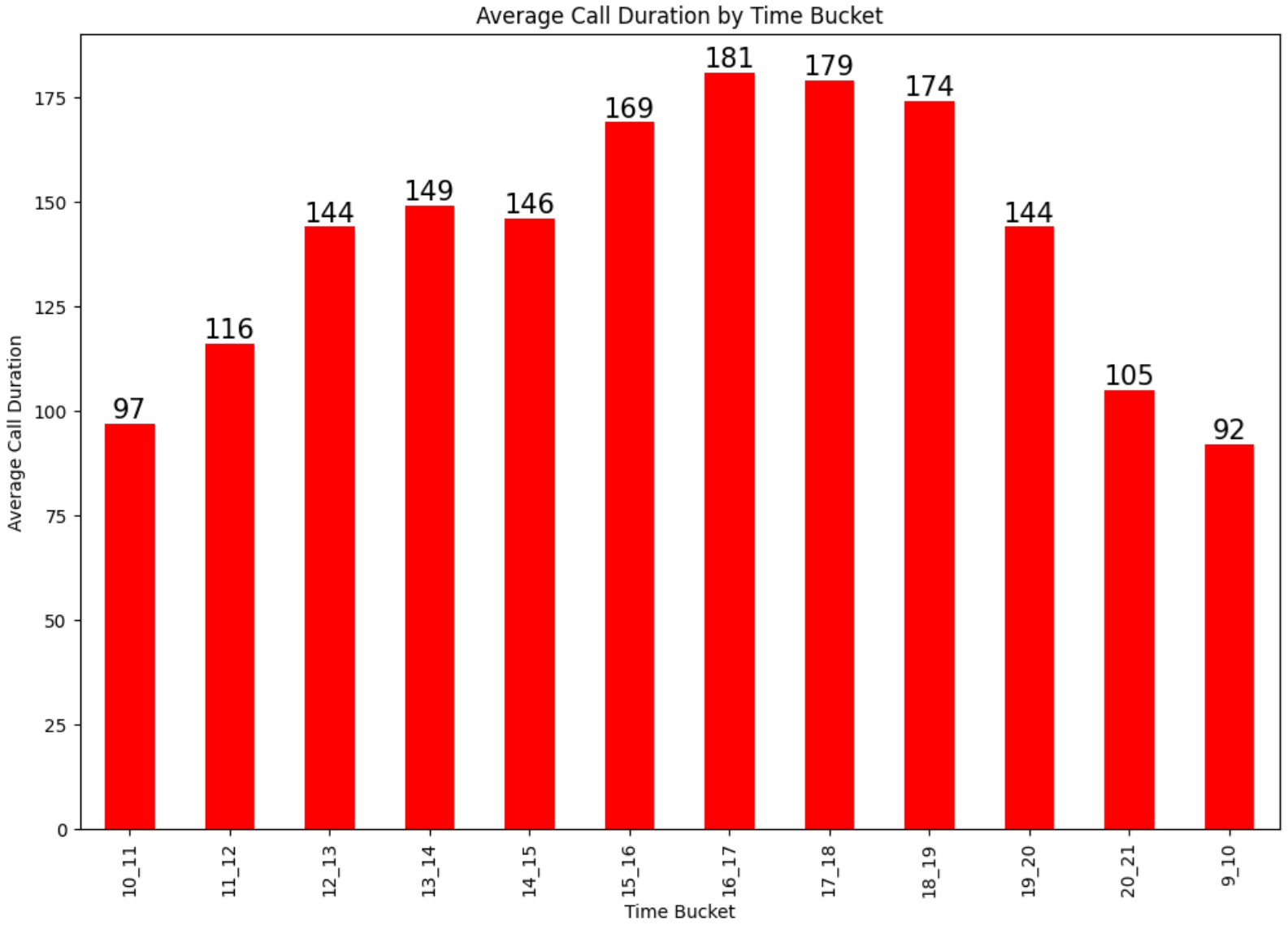
Creating the bar chart for the above data

0s

```
plt.figure(figsize = (12,8))
x = avg_call_duration.plot(kind='bar', color='Red')
for container in x.containers:
  x.bar_label(container, fontsize = 10)
plt.xlabel('Time Bucket')
plt.ylabel('Average Call Duration')
plt.title('Average Call Duration by Time Bucket')
plt.show()
```

Project 8 – ABC Call Volume Trend Analysis

Project by – Alokk Joshi



Insights from Task 1 (Call Duration Analysis)

1. **Peak Call Duration Timing**: The highest average call duration is observed between **4:00 PM and 5:00 PM**, indicating that callers prefer detailed discussions during this slot.

2. **Afternoon & Evening Preference**: Time slots **3-4 PM, 4-5 PM, 5-6 PM, and 6-7 PM** show maximum average call duration, suggesting that more staff is required during these hours.

3. **Lower Call Duration in the Morning**: The **9-10 AM** time slot has the lowest average call duration, meaning calls during this period might be for quick queries rather than detailed discussions.

Task 2

1. **Call Volume Analysis:** Visualize the total number of calls received. This should be represented as a graph or chart showing the number of calls against time. Time should be represented in buckets (e.g., 1-2, 2-3, etc.).

2. **Task:** Can you create a chart or graph that shows the number of calls received in each time bucket?

Project 8 – ABC Call Volume Trend Analysis

Project by – Alokk Joshi

```
no_of_calls = df.groupby('time_bucket')['call_seconds_s'].count().sort_values(ascending = True)
no_of_calls
```

time_bucket	call_seconds_s
20_21	5505
19_20	6463
18_19	7238
17_18	8534
16_17	8788
15_16	9159
9_10	9588
14_15	10561
13_14	11561
12_13	12652
10_11	13313
11_12	14626

dtype: int64

Project 8 – ABC Call Volume Trend Analysis

Project by – Alokk Joshi

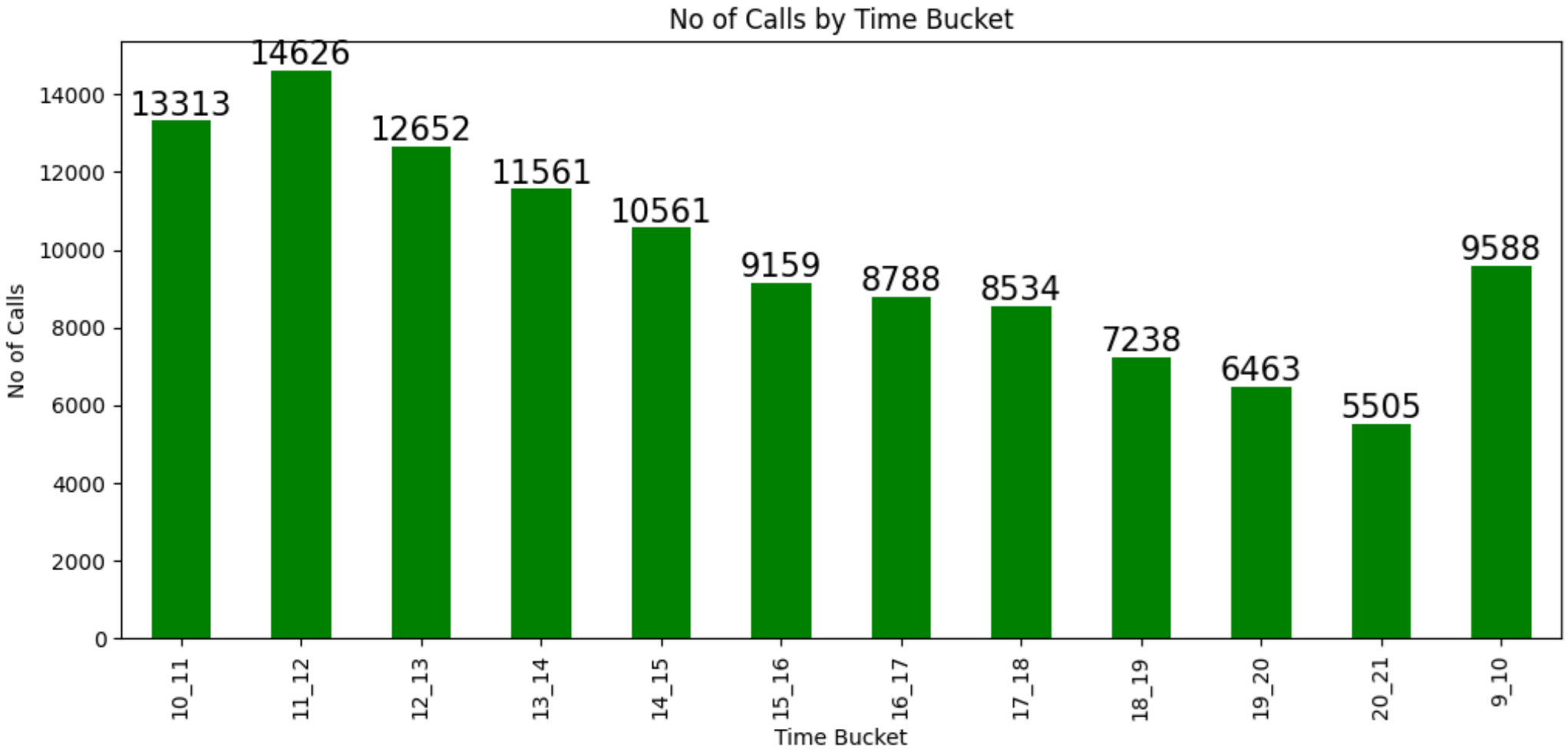
Creating the chart for the no_of_calls

0s

```
plt.figure(figsize = (12,5))
calls = no_of_calls.plot(kind = 'bar', color = 'Green')
for container in calls.containers:
  calls.bar_label(container, fontsize = 10)
plt.xlabel('Time Bucket')
plt.ylabel('No of Calls')
plt.title('No of Calls by Time Bucket')
plt.show()
```

Project 8 – ABC Call Volume Trend Analysis

Project by – Alokk Joshi



Insights from Task 2 (Call Volume Analysis)

1. **Highest Call Volume Timing**: The **11 AM to 12 PM** slot receives the most calls, but the average call duration is relatively lower, indicating more quick queries.

2. **Early Morning Call Trends**: The **9-10 AM** slot has significant call volume but with a lower duration, suggesting it's a secondary peak for quick customer inquiries.

3. **Lowest Call Volume**: The **8-9 PM** slot receives the least number of calls, possibly due to customer preference for resolving issues earlier in the day.

Task 3

1. **Manpower Planning:** The current rate of abandoned calls is approximately 30%. Propose a plan for manpower allocation during each time bucket (from 9 am to 9 pm) to reduce the abandon rate to 10%. In other words, you need to calculate the minimum number of agents required in each time bucket to ensure that at least 90 out of 100 calls are answered.

2. **Your Task:** What is the minimum number of agents required in each time bucket to reduce the abandon rate to 10%?

Assumptions: An agent works for 6 days a week; On average, each agent takes 4 unplanned leaves per month; An agent's total working hours are 9 hours, out of which 1.5 hours are spent on lunch and snacks in the office. On average, an agent spends 60% of their total actual working hours (i.e., 60% of 7.5 hours) on calls with customers/users. The total number of days in a month is 30.

Project 8 – ABC Call Volume Trend Analysis

Project by – Alokk Joshi

```
[120] manpower = round(df.groupby('time_bucket')['call_seconds_s'].sum()/3600/4.5, 0)
      print(manpower)
```

```
time_bucket
10_11     80.0
11_12    105.0
12_13    113.0
13_14    107.0
14_15     96.0
15_16     96.0
16_17     98.0
17_18     95.0
18_19     78.0
19_20     58.0
20_21     36.0
9_10      54.0
Name: call_seconds_s, dtype: float64
1016.0
```

Project 8 – ABC Call Volume Trend Analysis

Project by – Alokk Joshi

time bucket wise manpower for 23 days and futher calculations

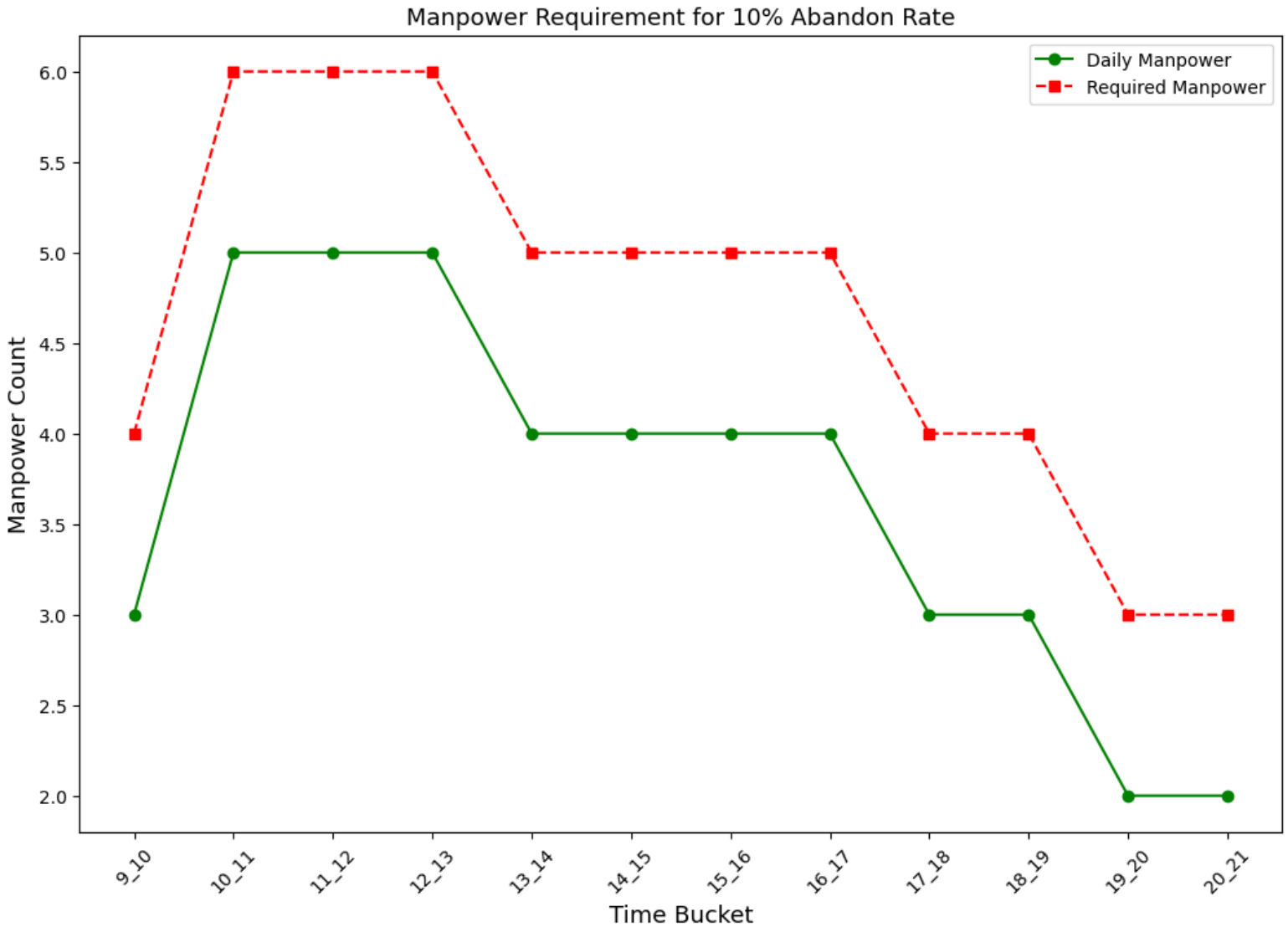
```
[121] daily_manpower = round(manpower / 23, 0) # for a single day at 30% call abandon rate
      print(daily_manpower)
      req_manpower = round((daily_manpower * 90)/70, 0) # requirement of manpower for 10% call abandon rate
      print(req_manpower)
      total_req_manpower = req_manpower.sum()
      print(total_req_manpower)
```

```
time_bucket
10_11    3.0
11_12    5.0
12_13    5.0
13_14    5.0
14_15    4.0
15_16    4.0
16_17    4.0
17_18    4.0
18_19    3.0
19_20    3.0
20_21    2.0
9_10     2.0
Name: call_seconds_s, dtype: float64
44.0
```

```
time_bucket
10_11    4.0
11_12    6.0
12_13    6.0
13_14    6.0
14_15    5.0
15_16    5.0
16_17    5.0
17_18    5.0
18_19    4.0
19_20    4.0
20_21    3.0
9_10     3.0
Name: call_seconds_s, dtype: float64
56.0
```

Project 8 – ABC Call Volume Trend Analysis

Project by – Alokk Joshi



Project Files link in slide no. 4

Project 8 – ABC Call Volume Trend Analysis

Project by – Alokk Joshi

Current manpower with 30% call abandon rate = 44

Manpower required to reduce it to 10% = 56

Insights from Task 3 (Manpower Planning)

1. **Current vs. Required Manpower**: The existing workforce of **44 agents** leads to a **30% call abandonment rate**, while **56 agents** are needed to reduce the abandonment rate to **10%**.

2. **Agent Efficiency Considerations**: Each agent effectively works **4.5 hours daily** on calls, accounting for breaks and other activities.

3. **Staffing Adjustments Needed**: Increasing the workforce by **12 agents** would ensure that at least **90 out of every 100 calls** are answered, improving customer satisfaction.

Task 4

1. **Night Shift Manpower Planning:** Customers also call ABC Insurance Company at night but don't get an answer because there are no agents available. This creates a poor customer experience. Assume 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. The distribution of these 30 calls is as follows:

2. **Your Task:** Propose a manpower plan for each time bucket throughout the day, keeping the maximum abandon rate at 10%.

Step 1

Creating a dataframe with nightshift time_bucket and call distributions ratio

Creating a dataframe for calculation of night shift

```
nightshift = { 'shift' : ['21_22', '22_23', '23_24', '0_1', '1_2', '2_3', '3_4', '4_5', '5_6', '6_7', '7_8', '8_9'],
               'call_distribution' : [3, 3, 2, 2, 1, 1, 1, 1, 3, 4, 4, 5]
               }
nshiftdf = pd.DataFrame(nightshift)
nshiftdf
```

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

Code output

	shift	call_distribution
0	21_22	3
1	22_23	3
2	23_24	2
3	0_1	2
4	1_2	1
5	2_3	1
6	3_4	1
7	4_5	1
8	5_6	3
9	6_7	4
10	7_8	4
11	8_9	5

Step 2

Calculating the night calls as per the 30% of day calls quantum

```
[86] dailycallseconds = (df['call_seconds_s'].sum() / 23)
     nightcallseconds = (dailycallseconds * 0.30).astype(int)
     nightcallseconds
```

```
np.int64(214736)
```

Calculating the night call_seconds, creating a new columnn and calculating the regualar manpower

```
[87] nshiftdf['nightcallseconds'] = (nshiftdf['call_distribution'] * nightcallseconds) / 30
```

```
[137] nshiftdf['nightcallseconds'] = nshiftdf['nightcallseconds'].astype(int)
```

Calculating the manpower for the calculated callseconds

```
nshiftdf['nmanpower'] = round(((nshiftdf['nightcallseconds'] / 3600) / 4.5), 0)
nshiftdf
```

Since we cannot keep the Executive count 0 if there are calls, we adjusted 0 to 1 wherever applicable

Replacing less than 1 with 1 as there are calls in the time_buckets and they cannot unresponded

```
nshiftdf['nmanpower'] = nshiftdf['nmanpower'].transform(lambda x: 1 if x < 1 else x)
nshiftdf
```

	shift	call_distribution	nightcallseconds	nmanpower
0	21_22	3	21473	1.0
1	22_23	3	21473	1.0
2	23_24	2	14315	1.0
3	0_1	2	14315	1.0
4	1_2	1	7157	0.0
5	2_3	1	7157	0.0
6	3_4	1	7157	0.0
7	4_5	1	7157	0.0
8	5_6	3	21473	1.0
9	6_7	4	28631	2.0
10	7_8	4	28631	2.0

Before = 9

After = 13

	shift	call_distribution	nightcallseconds	nmanpower
0	21_22	3	21473	1.0
1	22_23	3	21473	1.0
2	23_24	2	14315	1.0
3	0_1	2	14315	1.0
4	1_2	1	7157	1.0
5	2_3	1	7157	1.0
6	3_4	1	7157	1.0
7	4_5	1	7157	1.0
8	5_6	3	21473	1.0
9	6_7	4	28631	2.0
10	7_8	4	28631	2.0

Calculating the required night shift manpower for 10% abandon rate

```
rmanpower = round((nshiftdf['nmanpower'] * 90 / 70), 0)
rmanpower.sum()
```

```
np.float64(18.0)
```

Inserting the required manpower column to the dataframe

```
nshiftdf['req_nightshift_manpower'] = rmanpower
nshiftdf
```

Final Output

Inserting the required manpower column to the dataframe

```
nshiftdf['req_nightshift_manpower'] = rmanpower
nshiftdf['req_nightshift_manpower'] = nshiftdf['req_nightshift_manpower'].astype(int)
nshiftdf['nmanpower'] = nshiftdf['nmanpower'].astype(int)
print(nshiftdf.to_string())
```

```
    shift  call_distribution  nightcallseconds  nmanpower  nightcallseconds90  req_nightshift_manpower
0   21_22                  3             21473          1               27608                        1
1   22_23                  3             21473          1               27608                        1
2   23_24                  2             14315          1               18405                        1
3     0_1                  2             14315          1               18405                        1
4     1_2                  1              7157          1                9201                        1
5     2_3                  1              7157          1                9201                        1
6     3_4                  1              7157          1                9201                        1
7     4_5                  1              7157          1                9201                        1
8     5_6                  3             21473          1               27608                        1
9     6_7                  4             28631          2               36811                        3
10    7_8                  4             28631          2               36811                        3
11    8_9                  5             35789          2               46014                        3
```

Project 8 – ABC Call Volume Trend Analysis

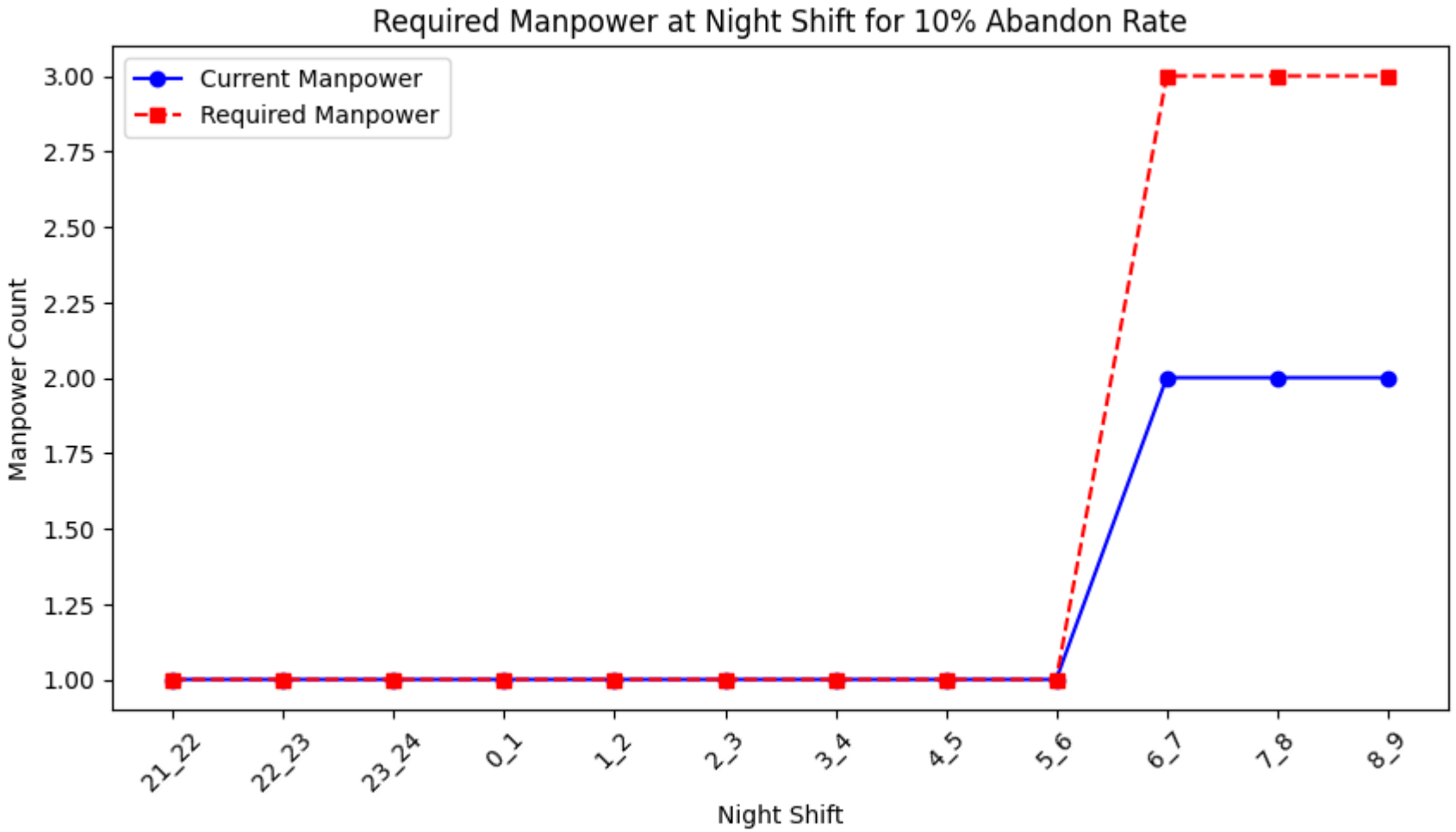
Project by – Alokk Joshi

Creating the chart with the above table

```
[216] plt.figure(figsize=(10, 5))
      # linechart
      plt.plot(nshiftdf['shift'], nshiftdf['nmanpower'], marker='o', linestyle='-', label='Current Manpower', color='b')
      plt.plot(nshiftdf['shift'], nshiftdf['req_nightshift_manpower'], marker='s', linestyle='--', label='Required Manpower', color='r')
      # Titles
      plt.xlabel('Night Shift')
      plt.ylabel('Manpower Count')
      plt.title('Required Manpower at Night Shift for 10% Abandon Rate')
      plt.xticks(rotation=45)
      # Adding legend
      plt.legend()
      plt.show()
```

Project 8 – ABC Call Volume Trend Analysis

Project by – Alokk Joshi



Insights from Task 4 (Night Shift Manpower Planning)

1. **Night Call Volume Distribution**: For every **100 calls made during the day (9 AM - 9 PM)**, an additional **30 calls are made at night (9 PM - 9 AM)**, requiring staffing adjustments.

2. **Manpower Increase Required**: Initially, **9 agents** were planned for night shifts, but after adjustments, **13 agents** were assigned to meet demand while keeping the call abandonment rate under **10%**.

3. **Staffing Redistribution for Coverage**: Executives were assigned even in time slots where call volume was low to ensure there were no periods of **zero availability**.

5 Major Takeaways

1. **Peak Call Handling Strategy**: Most calls occur between **11 AM - 12 PM**, while the longest call durations happen **after 3 PM**, requiring **different staffing strategies** for handling volume vs. duration.

2. **Manpower Efficiency is Critical**: The current workforce is **insufficient**, and increasing it from **44 to 56 agents** can significantly reduce call abandonment.

3. **Night Call Handling Needs Attention**: **30% of daytime calls** also happen at night, requiring a dedicated **night shift** to improve customer experience.

4. **Staffing Optimization Reduces Customer Frustration**: Without proper staffing, **high abandonment rates (30%)** lead to poor customer service. Proper workforce planning can cut it down to **10%**.

5. **Data-Driven Decision-Making Works**: Analysis of **call volume, duration, and efficiency metrics** led to a structured manpower planning approach, demonstrating how **data insights optimize operations**.

5 Major Learnings

1. **Data Visualization Enhances Understanding**: Charts and time-bucket graphs make trends in **call volume and duration** easier to interpret.

2. **Correlation Matters**: Understanding the relationship between **call volume, call duration, and staffing** helps in making **data-backed staffing decisions**.

3. **Assumption-Based Modeling is Useful**: Using workforce assumptions like **working hours, break times, and efficiency levels** helps in predicting manpower requirements accurately.

4. **Problem-Solving via Data-Driven Approaches**: The project demonstrated how analyzing data in steps (e.g., peak hours, call patterns, staffing needs) leads to **effective manpower planning**.

5. **Data Cleaning & Adjustments Are Essential**: Simple errors like setting the **executive count to zero** in some slots had to be adjusted, reinforcing the importance of **data integrity and logical adjustments**.