Customer Support Ticket Analytics & Agent Performance Dashboard

Analyzing Support Tickets for BeechTech to Improve Response Times and Agent Efficiency

Lava McCoy
November 17, 2025
Sr. Data Analyst

Table of Contents

Introduction

This is an analysis of customer support tickets for BeechTech, a small technology services company. The dataset includes ticket details such as creation and resolution dates, priority levels, issue types, customer regions, and assigned agents. The goal of the analysis is to evaluate agent performance, monitor ticket resolution times, and provide actionable insights to improve customer support efficiency. This report demonstrates skills in data cleaning, exploratory analysis, pivot tables, dashboards (Power BI), and programmatic analysis using Python.

2. EDA Summary

This exploratory data analysis (EDA) provides a high-level summary of ticket activity, resolution performance, and dataset completeness, serving as the foundation for deeper analysis in Excel, Power BI, and Python.

Exploratory Data Analysis Summary – Customer Support Analytics Project		
Metric	**Value**	**Notes**
Total Tickets	500	Total number of customer suppor tickets in the dataset
Tickets Resolved	490	Number of tickets marked as resolved
Pending Tickets	10	Tickets still pending resolution at the time of analysis
Average Resolution Hours	76.11	Mean resolution time across all resolved tickets; ignores pending/NaN values
Fastest Resolution	1	Minimum resolution time in hours (quickest ticket handled)
Longest Resolution	543	Maximum resoluition time in hours (longest ticket handled)
Distinct Agents	4	Only counting known agents; unknowns handled separately

3. Excel Pivot Tables

The following pivot tables summarize key aspects of the customer support dataset. They show ticket counts by agent, priority, issue type, and region, providing insight into workload distribution and trends.

a. Pivot Table 1 (Ticket Count by Priority per Agent):

This table highlights how many tickets each agent handled at each priority level, helping identify workload distribution and high-priority ticket assignments.

Ticket Count Priority	East	North	South	West	Grand Total
A	**26**	**33**	**27**	**26**	**112**
High	4	9	6	5	24
Low	12	9	9	8	38
Medium	10	10	10	11	41
Unknown		5	2	2	9
B	**19**	**41**	**18**	**32**	**110**
High	2	8	4	7	21
Low	10	16	7	15	48
Medium	7	14	7	10	38
Unknown		3			3
C	**31**	**40**	**38**	**23**	**132**
High	5	11	4	9	29
Low	15	16	15	6	52
Medium	11	11	18	7	47
Unknown		2	1	1	4
D	**37**	**31**	**25**	**33**	**126**
High	6	6	4	5	21
Low	14	10	15	16	55
Medium	15	13	6	10	44
Unknown	2	2		2	6
Unknown	**5**	**2**	**10**	**3**	**20**
High	2		2	1	5
Low			3	1	4
Medium	2	1	3	1	7
Unknown	1	1	2		4
Grand Total	**118**	**147**	**118**	**117**	**500**

b. **Pivot Table 2 (Tickets by Issue Type and Priority):**

This table breaks down tickets by issue type and priority, showing which types of issues occur most frequently and at what priority.

Ticket Types	# of Tickets
Account	**135**
High	32
Low	51
Medium	47
Unknown	5
Billing	**122**
High	26
Low	50
Medium	39
Unknown	7
General	**130**
High	25
Low	53
Medium	46
Unknown	6
Technical	**109**
High	17
Low	43
Medium	45
Unknown	4
Unknown	**4**
Unknown	4
Grand Total	**500**

c. Pivot Table 3 (Tickets by Region and Issue Type):

This table shows ticket distribution across customer regions and issue types, allowing us to identify regional trends and common problems.

Sum of Ticket Count Per Customer Region	Account	Billing	General	Technical	Unknown	Grand Total
East	35	25	29	28	1	118
North	41	41	37	27	1	147
South	25	28	38	25	2	118
West	34	28	26	29		117
Grand Total	**135**	**122**	**130**	**109**	**4**	**500**

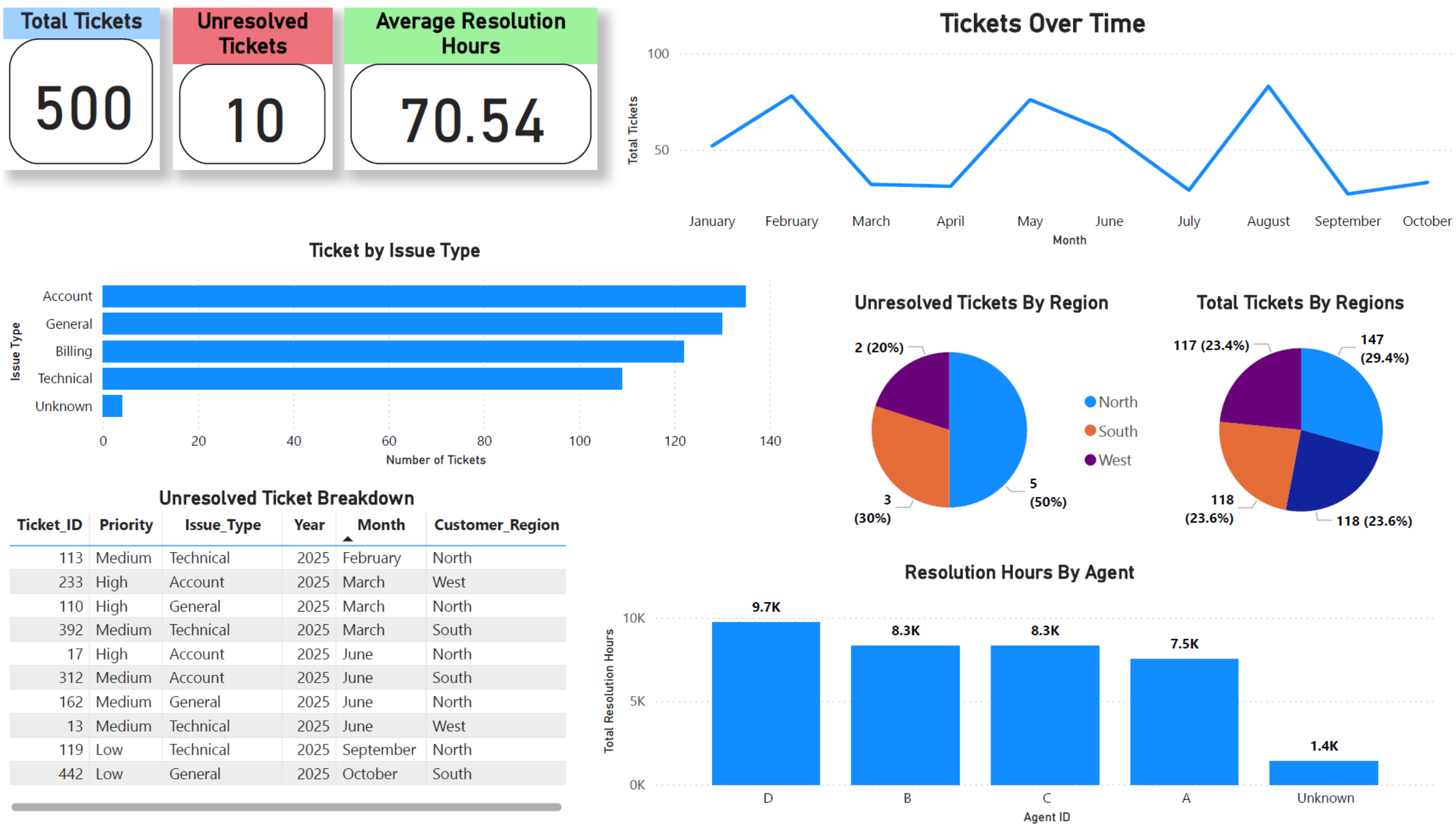
d. Table 4 (Agent Totals and Resolution Hours):

This table provides a performance overview for each agent, including total tickets handled, average resolution time, and total resolution hours, helping assess efficiency and workload balance.

Agent Name	Total Tickets	Average Resolution Hours	Total Resolution Hours
A	112	68.90825688	7,511
B	110	76.44036697	8,332
C	132	64.9765625	8,317
D	126	78.51612903	9,736

4. Power BI Visuals

Power BI was used to create interactive dashboards and visuals that summarize ticket and agent performance. While only static screenshots are included here for reporting purposes, the full interactive Power BI file is attached separately, allowing readers to explore the data dynamically.



Ticket_ID	Priority	Issue_Type	Year	Month	Customer_Region
113	Medium	Technical	2025	February	North
233	High	Account	2025	March	West
110	High	General	2025	March	North
392	Medium	Technical	2025	March	South
17	High	Account	2025	June	North
312	Medium	Account	2025	June	South
162	Medium	General	2025	June	North
13	Medium	Technical	2025	June	West
119	Low	Technical	2025	September	North
442	Low	General	2025	October	South

5. Audit Log

The audit log documents all data cleaning and preprocessing steps performed on the customer support ticket dataset. This ensures transparency, reproducibility, and accuracy for all subsequent analyses.

Step #	Issue / Observation	Action Taken	Reason / Justification	Date Completed
1	Duplicate Ticket_ID values (but different customers)	Used Random_ID or added new ID numbers like 501, 502, etc.	Prevented valid rows from being deleted; maintained one record per ticket event	11/11/2025
2	Missing values in Customer_Region	Filled missing values with "Unknown"	Ensured all tickets can be grouped in Tableau/Power BI	11/11/2025
3	Blank Date_Resolved for unresolved tickets	Marked Resolution_Hours as NaN	Keeps numeric field valid for calculations and dashboards	11/11/2025
4	Some Ticket_IDs had swapped dates (Date_Resolved earlier than Date_Created)	Manually corrected by swapping dates	Ensures accurate positive Resolution_Hours	11/11/2025
5	Negative values in Resolution_Hours due to date errors	Fixed after correcting swapped dates	Maintains correct resolution time calculations	11/11/2025
6	Inconsistent region names (e.g., "Nourth" → "North")	Corrected spelling	Standardized for grouping and filtering	11/11/2025
7	Unnecessary columns (e.g., Notes, extra IDs)	Dropped irrelevant columns	Simplified dataset for analysis and visualization	11/11/2025
8	Added dual Resolution columns	Created Resolution_Hours_Num (numeric) and Resolution_Hours_Text ("Pending")	Kept data analysis-ready and report-readable	11/11/2025
9	Verified data integrity using Python	Used Python (Pandas) to confirm: total rows = 500, no nulls in key columns, average resolution = 76.11 hrs	Ensured cleaned dataset matched expectations before visualization	11/11/2025
10	Required accurate time calculations	Converted Date_Created and Date_Resolved to datetime format and recalculated Resolution_Hours	Ensured consistency of time-based fields for analysis	11/12/2025
11	Wanted automated validation of results	Used groupby() to count tickets per Issue_Type and cross-checked with Power BI totals	Confirmed data consistency between Python and visualization tools	11/12/2025
12	Needed reproducibility for future updates	Saved cleaned dataset as new CSV (CustomerService_CLEAN.csv)	Allows future re-import into SQL/Power BI/Tableau without re-cleaning	11/12/2025

Each step records the issue observed, the action taken, the justification, and the date completed. This log demonstrates how the dataset was cleaned and prepared for analysis in Excel, Power BI, and Python.

Python Analysis

Objective:
To demonstrate repeatable, programmatic analysis of the cleaned customer support dataset using Python (pandas and matplotlib). Python was used to verify the data, create pivot tables, and generate visualizations for key metrics.

1. **Data Load and Verification**
   The cleaned CSV dataset was loaded into Python. The first 5 rows were displayed to verify that the data loaded correctly and to confirm column names and data types.
   Output:
   First 5 rows of the dataset showing ticket ID, dates, resolution hours, customer region, issue type, priority, and assigned agent.

```python
import pandas as pd

df = pd.read_csv("Customer_Support_Tickets_CLEANED SQL.csv")
df.head()
```

[9] ✓ 0.1s

	Ticket_ID	Date_Created	Date_Resolved	Resolution_Hours	Customer_Region	Issue_Type	Priority	Assigned_Agent	Random_ID	Ticket_Count
0	1	8/20/2025 0:00	8/21/2025 22:00	46.0	West	Technical	Low	B	78386	1
1	2	8/2/2025 0:00	8/3/2025 12:00	36.0	North	Technical	Low	B	40432	1
2	3	2/12/2025 0:00	2/13/2025 11:00	35.0	North	Billing	High	B	87193	1
3	4	5/18/2025 0:00	5/20/2025 18:00	66.0	East	Account	Medium	C	34651	1
4	5	7/15/2025 0:00	7/16/2025 10:00	34.0	East	Technical	High	C	31152	1

2. **Column and Data Overview**

A quick check of column names and data types ensured the dataset was ready for analysis.

Output:

- Full list of columns
- Data types (numeric, text, datetime)
- Non-null counts for each column

```python
# structure
df.info()

# stats
df.describe()
```

[10] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Ticket_ID         500 non-null    int64
 1   Date_Created      500 non-null    object
 2   Date_Resolved     500 non-null    object
 3   Resolution_Hours  500 non-null    float64
 4   Customer_Region   500 non-null    object
 5   Issue_Type        500 non-null    object
 6   Priority          500 non-null    object
 7   Assigned_Agent    500 non-null    object
 8   Random_ID         500 non-null    int64
 9   Ticket_Count      500 non-null    int64
dtypes: float64(1), int64(3), object(6)
memory usage: 39.2+ KB
```

	Ticket_ID	Resolution_Hours	Random_ID	Ticket_Count
count	500.000000	500.00000	500.000000	500.0
mean	254.010000	70.54200	55483.000000	1.0
std	146.854289	109.97001	26261.717812	0.0
min	1.000000	0.00000	10064.000000	1.0
25%	127.750000	34.00000	33251.250000	1.0
50%	252.500000	44.00000	54788.500000	1.0
75%	381.250000	65.00000	78814.500000	1.0
max	597.000000	543.00000	99700.000000	1.0

3. **Pivot Table: Ticket Count by Priority per Agent**
This pivot table replicates the Excel pivot table, counting the number of tickets each agent handled for each priority level.
Output:
- Table showing agents as rows, priorities as columns, and ticket counts as values.

```
pivot_priority = df.pivot_table(
    values='Ticket_ID',        # Count tickets
    index='Assigned_Agent',    # column name
    columns='Priority',        # column name
    aggfunc='count',           # Count the tickets
    fill_value=0
)
pivot_priority
```

✓ 0.0s

Priority	High	Low	Medium	Unknown
Assigned_Agent				
A	24	38	41	9
B	21	48	38	3
C	29	52	47	4
D	21	55	44	6
Unknown	5	4	7	4

4. **Pivot Table: Tickets by Region and Issue Type**
This pivot table summarizes tickets by customer region and issue type, providing insight into which regions experience certain types of issues more frequently.
Output:
    - Table with regions as rows, issue types as columns, and ticket counts as values.

```python
pivot_region = df.pivot_table(
    values='Ticket_ID',           # Total Ticket Count
    index='Customer_Region',      # Rows labeled/are Customer Region
    columns='Issue_Type',         # Columns labeled/are Issue Type
    aggfunc='count',              # Count the tickets
    fill_value=0                  # missing values to 0
)
pivot_region
```

✓ 0.0s

Issue_Type	Account	Billing	General	Technical	Unknown
Customer_Region					
East	35	25	29	28	1
North	41	41	37	27	1
South	25	28	38	25	2
West	34	28	26	29	0

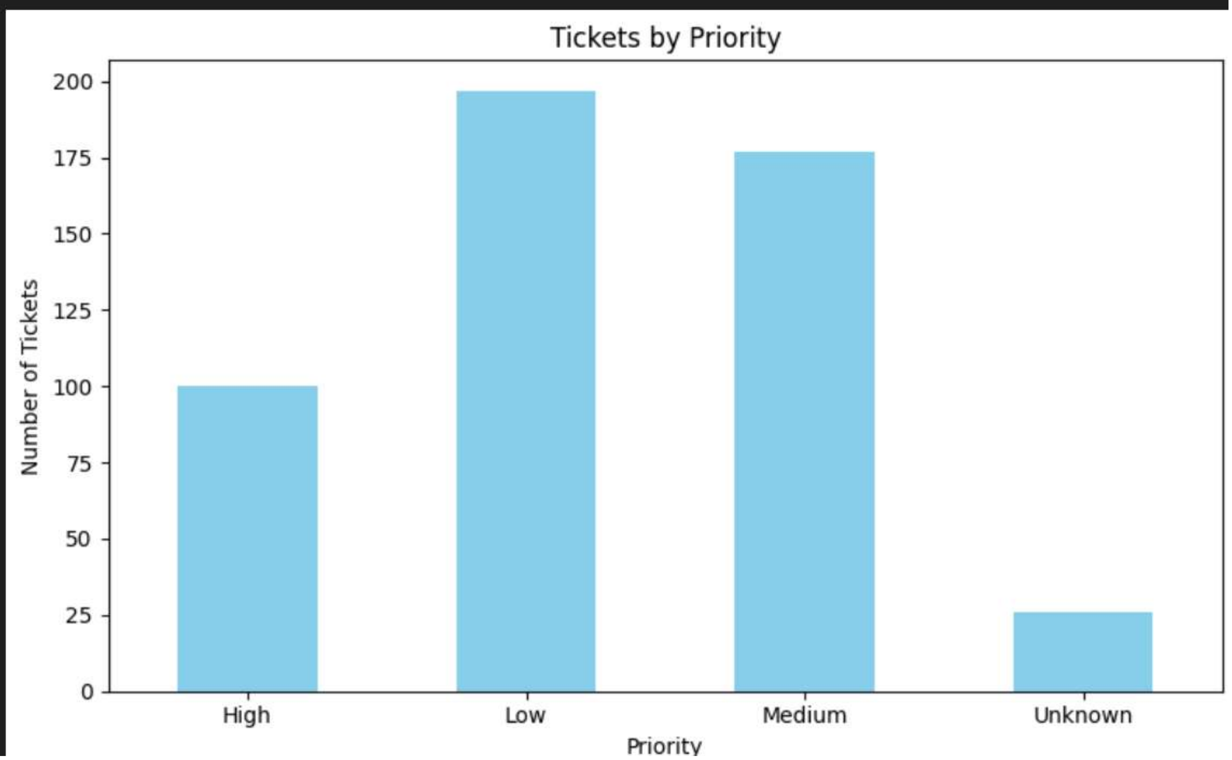
5. **Tickets by priority bar chart**

```python
import matplotlib.pyplot as plt

# Group together by priority and group tickets
priority_counts = df.groupby("Priority")["Ticket_ID"].count()

# Actual bar graph generation
plt.figure(figsize=(8,5))
priority_counts.plot(kind="bar", color="skyblue")
plt.title("Tickets by Priority")
plt.xlabel("Priority")
plt.ylabel("Number of Tickets")
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

✓ 2.1s

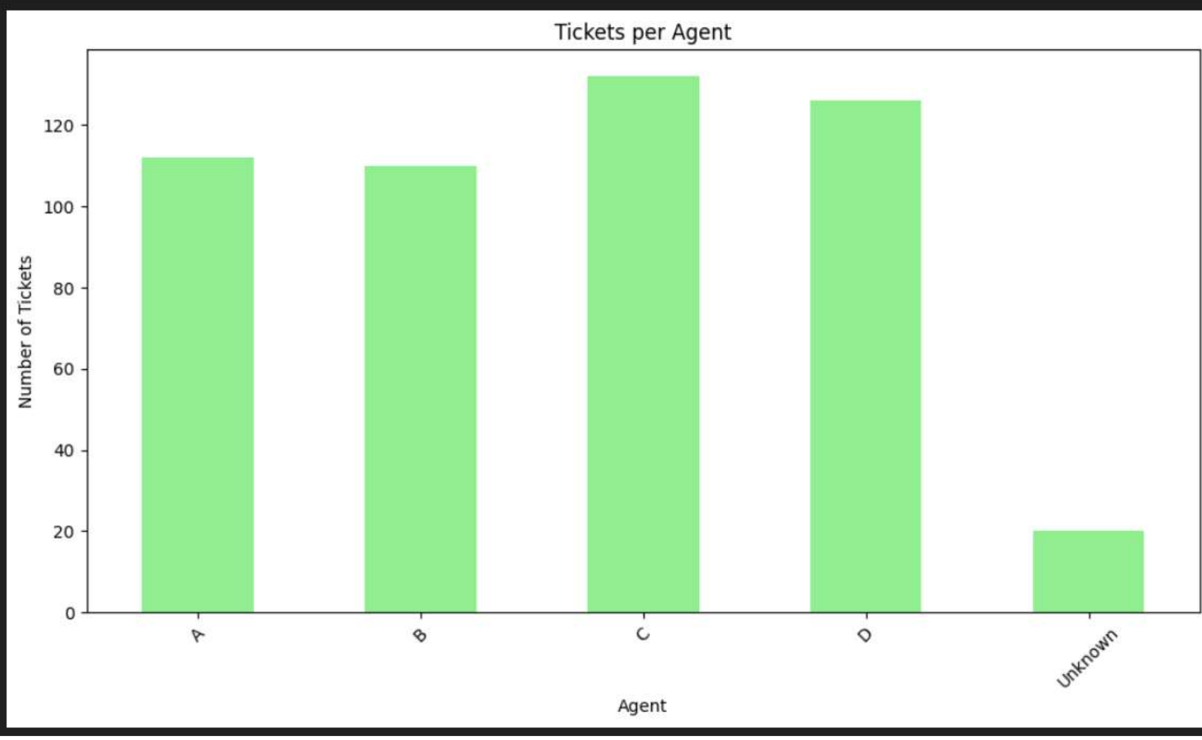


6. Tickets per Agent Bar Chart

```python
# Group together by Agent and count the tickets
agent_counts = df.groupby("Assigned_Agent")["Ticket_ID"].count()

# Bar
plt.figure(figsize=(10,6))
agent_counts.plot(kind="bar", color="lightgreen")
plt.title("Tickets per Agent")
plt.xlabel("Agent")
plt.ylabel("Number of Tickets")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

✓ 0.4s



6. Conclusion / Insights

- **Total Ticket Volume:** 500 tickets in the dataset, with medium priority accounting for the largest share. This indicates that the team should pay particular attention to mid-level tickets to avoid delays in resolution.
- **Agent Performance:** Pivot tables and Python analysis show that Agent C handled 132 tickets**,** while Agent B handled 110, highlighting uneven workloads that may require redistribution or support.
- **Regional Trends:** Certain regions, such as North, report more tickets related to account and billing issues, suggesting targeted training or regional support improvements.
- **Resolution Hours:** 10 tickets still remain unsolved, however, the completed tickets show an average resolution time of 71.98 hours, which can be used as a benchmark for future performance.
- **Overall Insight:** Combining Excel pivot tables, Python analysis, and Power BI dashboards provides actionable insights for BeechTech, helping optimize agent workload, target high-volume regions, and improve customer support efficiency.

Conclusion

This analysis highlights clear patterns in ticket behavior, agent performance, and priority distribution. However, 26 tickets were marked with an unknown priority, representing 5.2% of the total tickets. These misclassified tickets should have been placed into a defined priority level, as missing classifications weaken workload accuracy and SLA measurement. Additionally, 4 tickets had an unknown issue type, which limits insight into what customers were actually experiencing. These unknown values create blind spots that can distort trends and reduce the reliability of insights.

This report consolidates analyses across Excel, Power BI, and Python, providing actionable insights into customer support ticket trends and agent performance. All project files including raw datasets, Excel workbooks, Power BI dashboards, Python notebooks, and the audit log are available for review.