Building a Service Ticket Management Dashboard

Jenn Bushey and Mehreen Akmal

Ticket Generator Program

First, we need to generate a sample event log to populate a database.

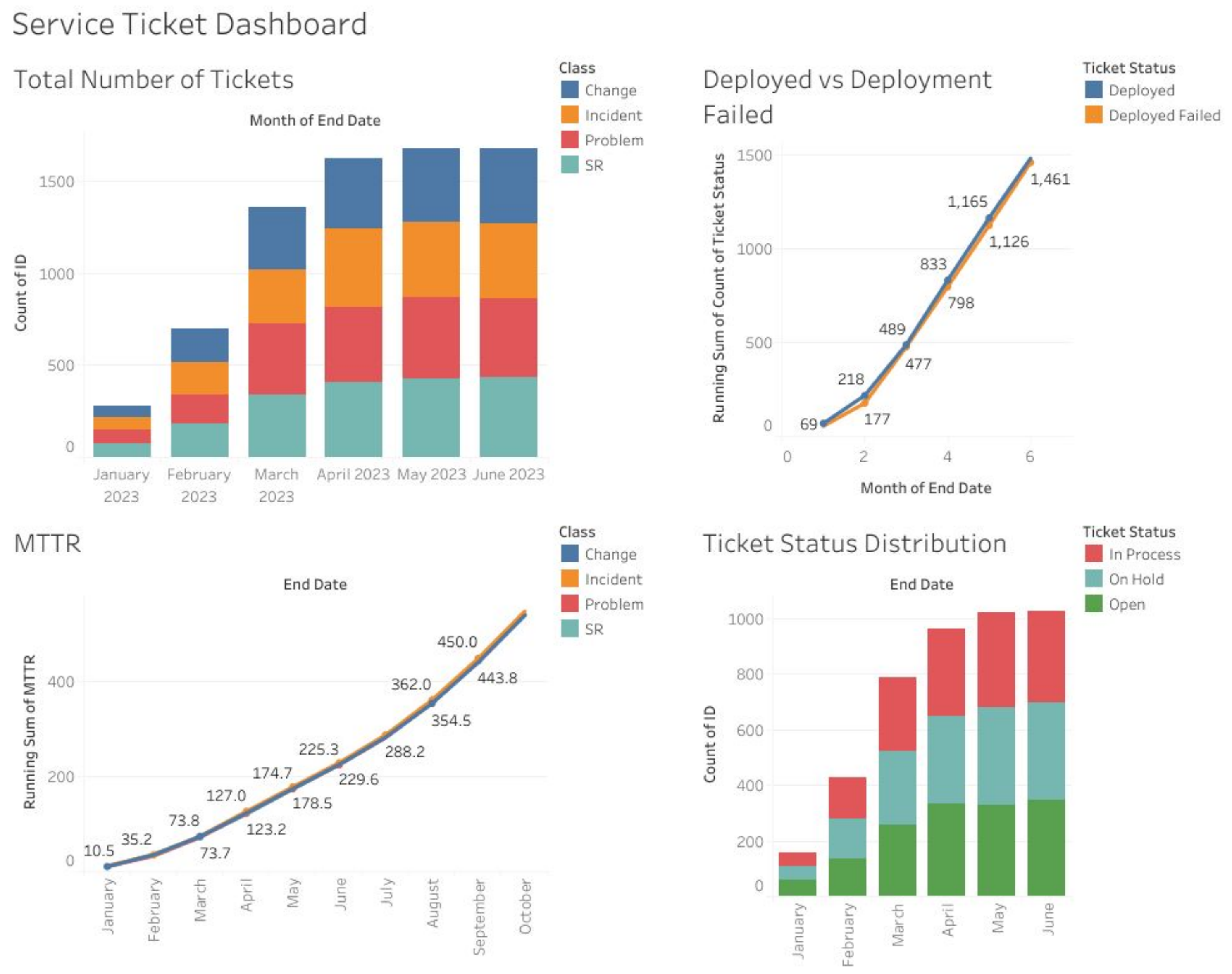
- Start date randomized between user-selected dates.
- Generates end dates no more than 100 days after the randomized start date.
- Duration days must be greater than 0.
- All values in the ticket are randomly determined based on the respective values in the database table.

Programming Assumptions

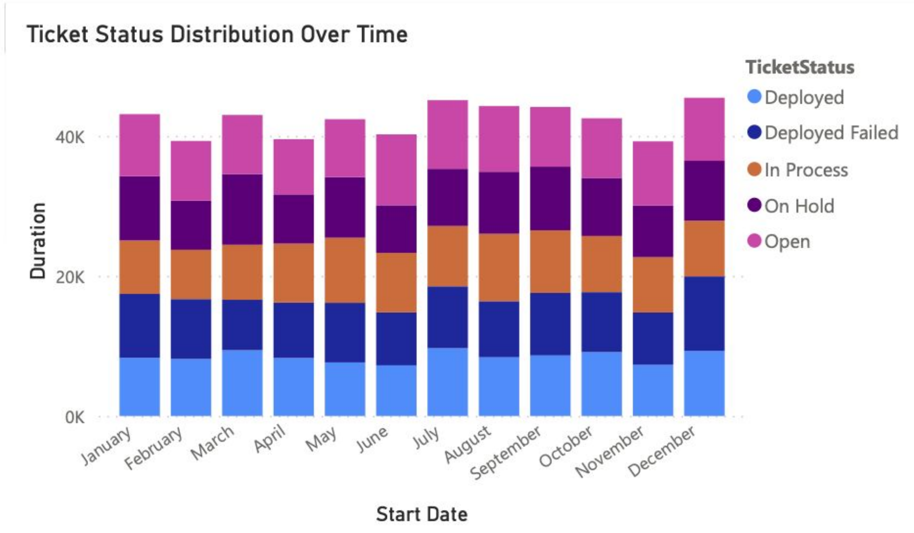
- User input:
  - 10,000 tickets
  - Start date: 2023-01-01
  - End date: 2023-06-30
- Assuming the user selected start and end dates are the dates for which they want to see the data. We have truncated the view of our data to this timeline (January – June).
- All data was randomly selected with even probability of occurrence.

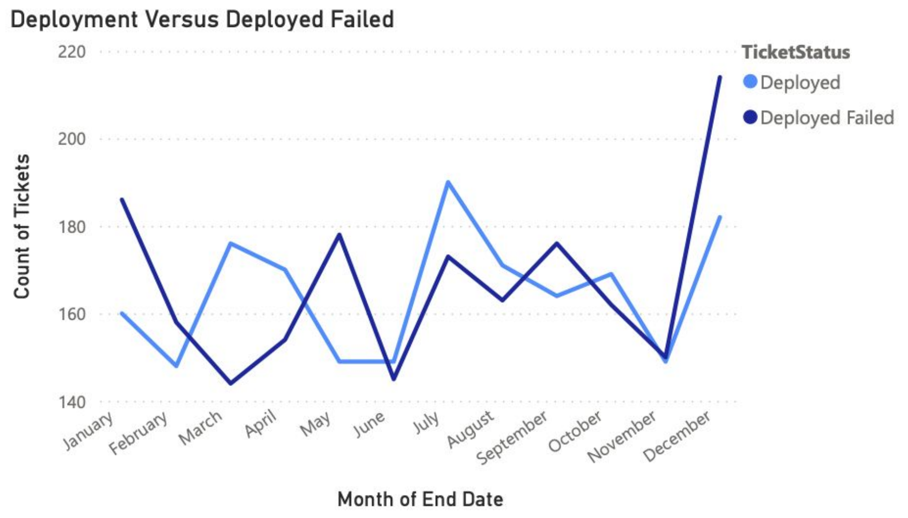
Visualize Ticket Data

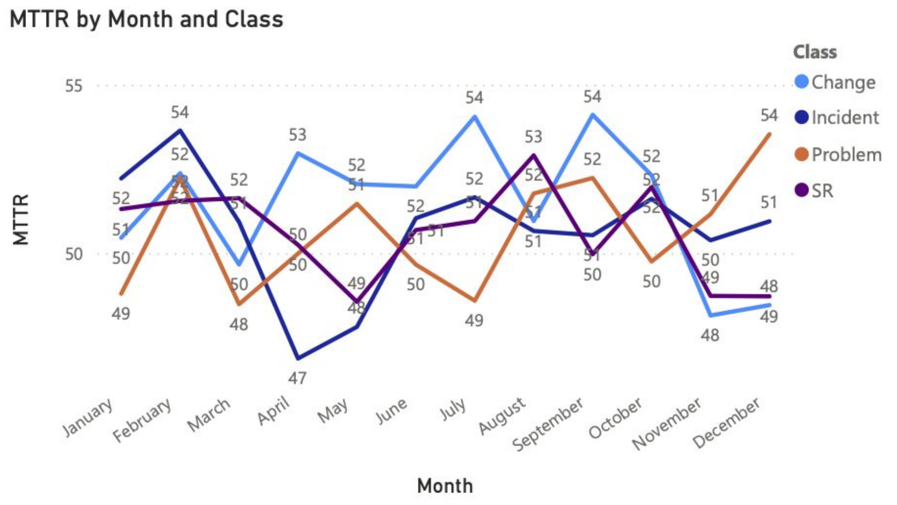
Tableau

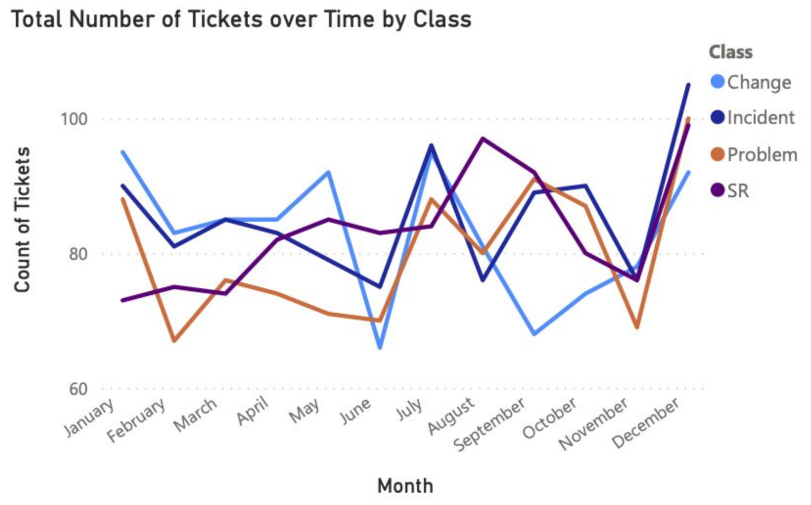

Service Ticket Dashboard
Total Number of Tickets
Month of End Date
Count of ID
1500
1000
500
0
January 2023
February 2023
March 2023
April 2023
May 2023
June 2023
Class
Change
Incident
Problem
SR
Deployed vs Deployment Failed
Ticket Status
Deployed
Deployed Failed
Running Sum of Count of Ticket Status
Month of End Date
69
218
177
489
477
833
798
1,165
1,126
1,461
0
2
4
6
MTTR
End Date
Running Sum of MTTR
400
200
0
10.5
35.2
73.8
73.7
127.0
123.2
174.7
178.5
225.3
229.6
288.2
362.0
354.5
450.0
443.8
January
February
March
April
May
June
July
August
September
October
Class
Change
Incident
Problem
SR
Ticket Status Distribution
End Date
Count of ID
1000
800
600
400
200
0
January
February
March
April
May
June
Ticket Status
In Process
On Hold
Open

PowerBI


Ticket Status Distribution Over Time
TicketStatus
Deployed
Deployed Failed
In Process
On Hold
Open
Duration
40K
20K
0K
January
February
March
April
May
June
July
August
September
October
November
December
Start Date


Deployment Versus Deployed Failed
TicketStatus
Deployed
Deployed Failed
Count of Tickets
220
200
180
160
140
January
February
March
April
May
June
July
August
September
October
November
December
Month of End Date

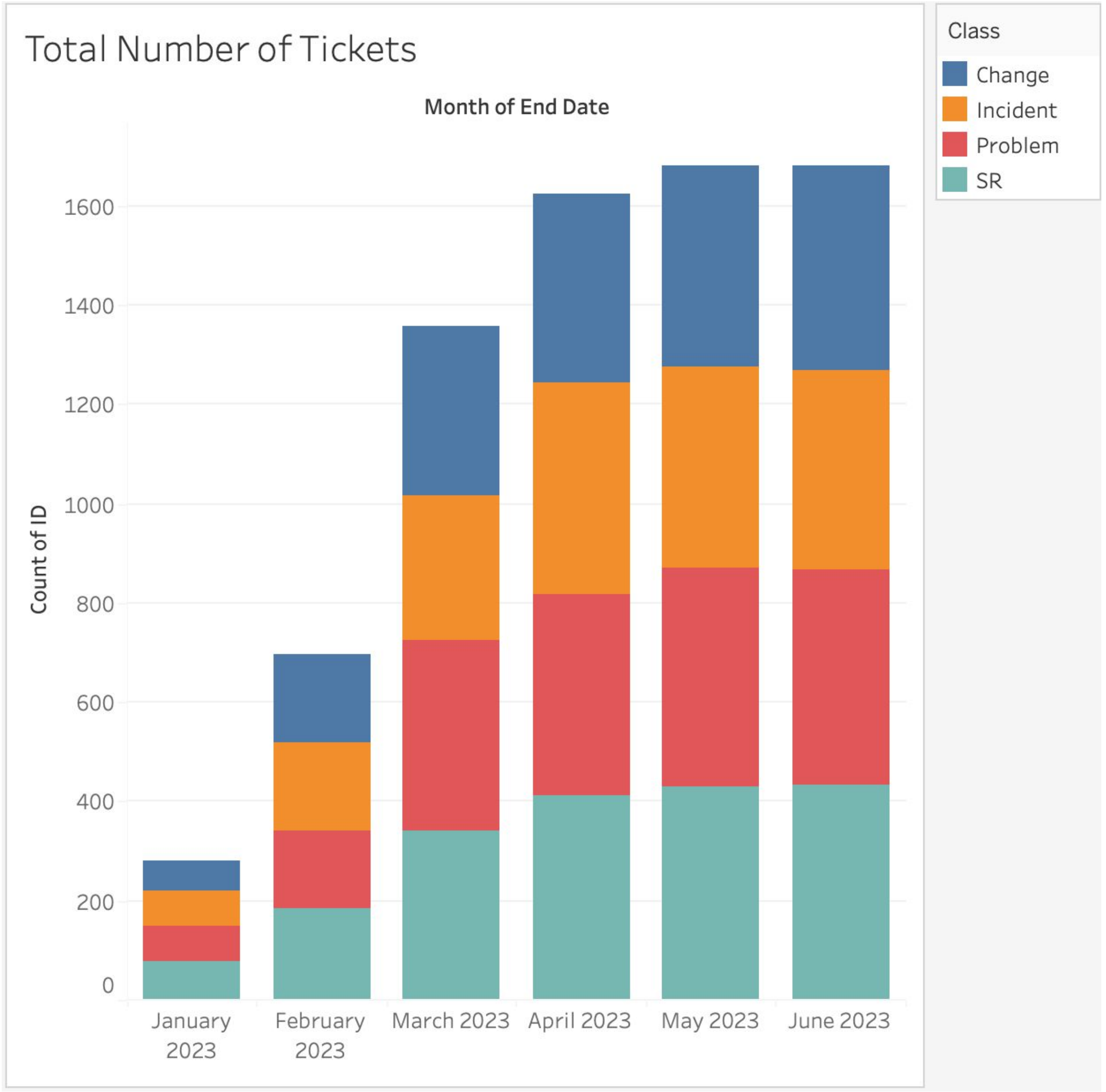

MTTR by Month and Class
Class
Change
Incident
Problem
SR
MTTR
55
50
January
February
March
April
May
June
July
August
September
October
November
December
Month


Total Number of Tickets over Time by Class
Class
Change
Incident
Problem
SR
Count of Tickets
100
80
60
January
February
March
April
May
June
July
August
September
October
November
December
Month

Total Number of Tickets Over Time

This visualization depicts a progressive increase in the number of tickets over time for each class, evident by the rising count of tickets with their end date falling within each month, starting from January 1. This trend may be attributed to several factors, such as resource constraints or the implementation of a new system, leading to a backlog of service tickets across all classes.

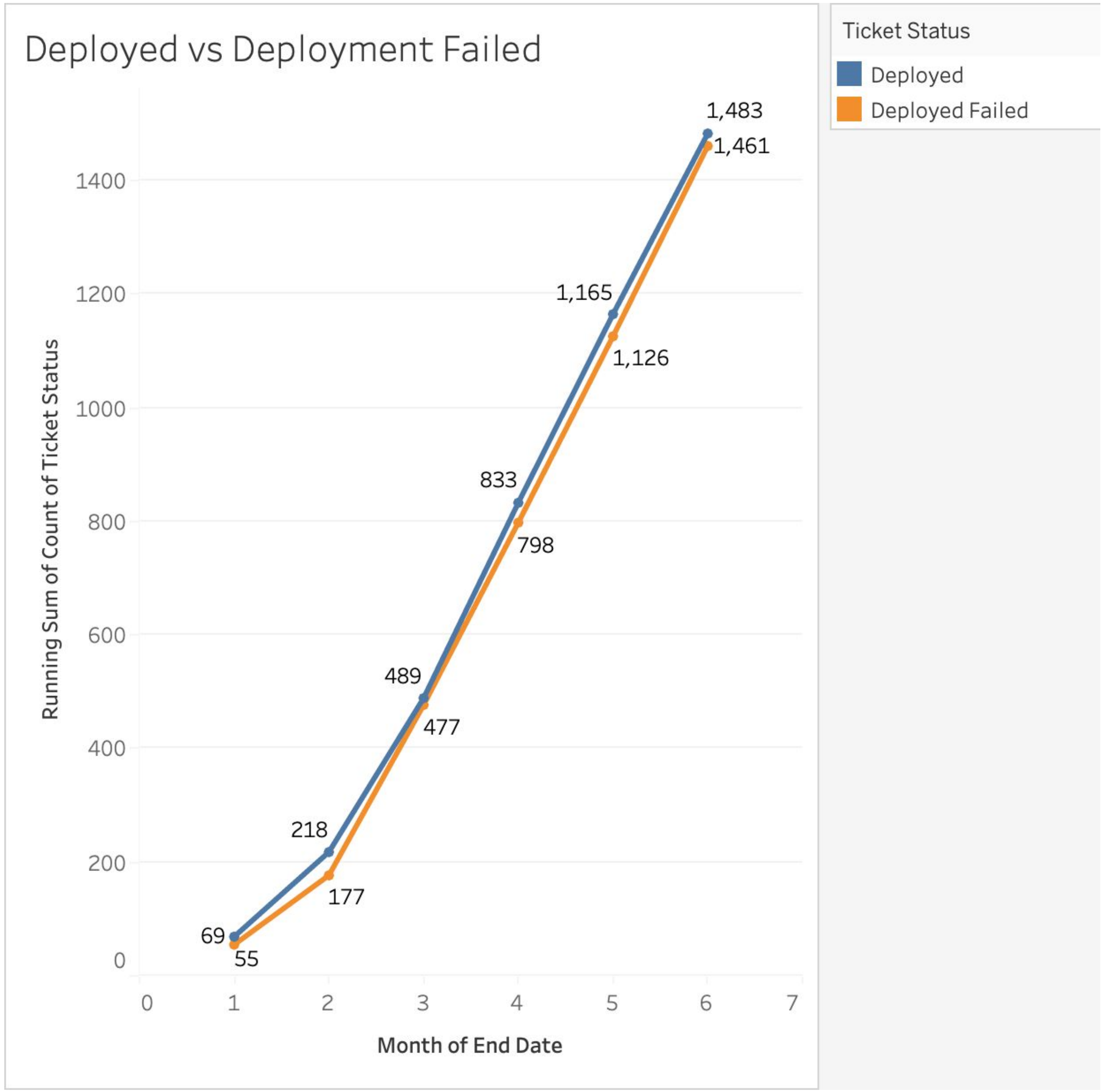
In real-world scenarios, a uniform distribution of tickets across classes is unlikely. Service requests (SR) and incidents are expected to constitute the majority of tickets, given their nature of being smaller issues with quick resolutions. Conversely, problems are likely to be the least frequently reported ticket type, as they represent the root causes of many incidents and occur less frequently by definition.



Deployment Success vs Failed

This visualization illustrates a consistent rise in the total number of tickets over time, with the trend commencing in January. Disparities between test environments and real-life production settings often lead to numerous instances of deployment failures.

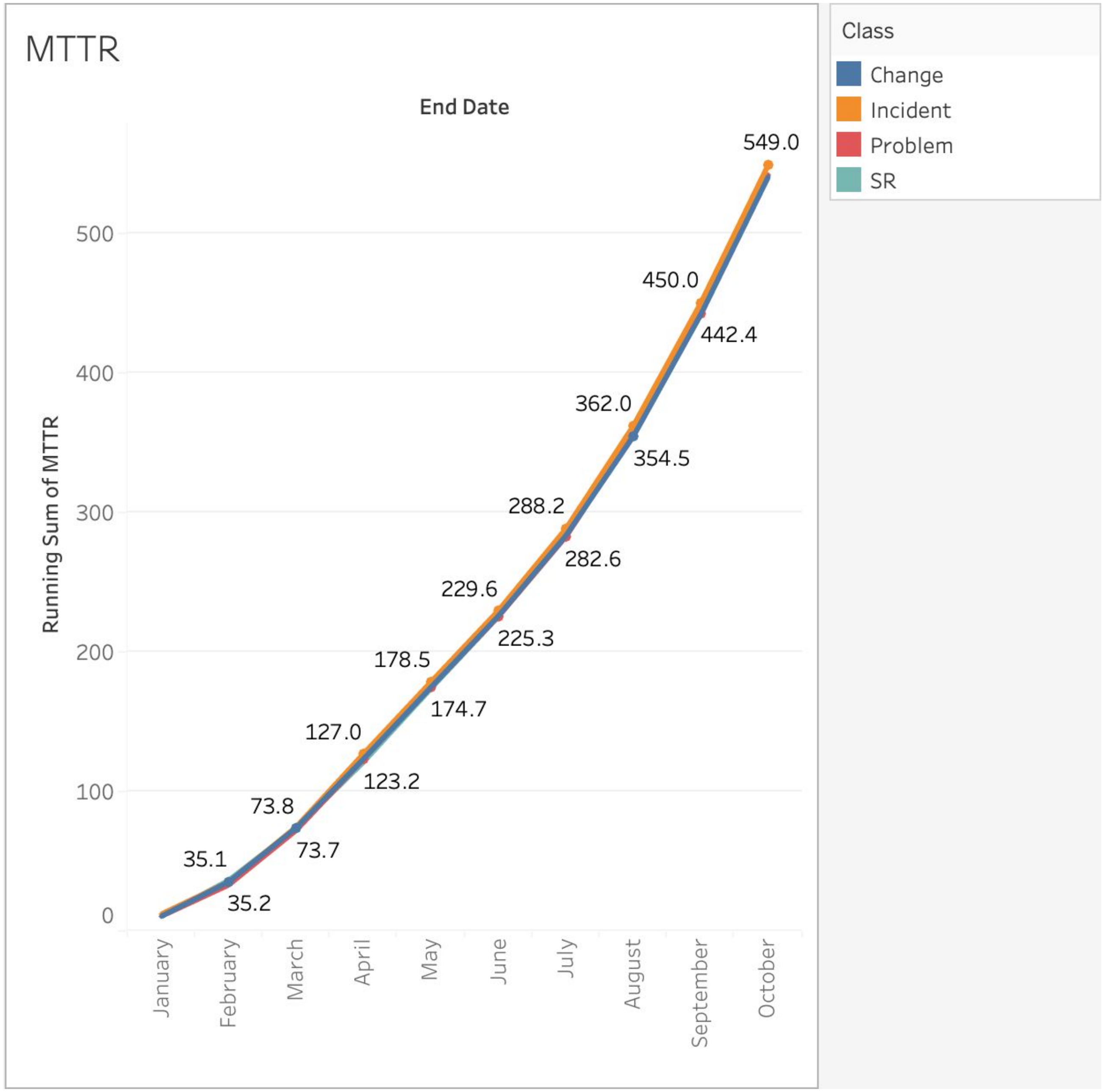
In an optimal scenario, the number of successful deployments should greatly outweigh the number of deployment failures.



MTTR

The mean time to repair (MTTR) is observed to be escalating over time, paralleling the exponential growth in the number of tickets, with a consistent MTTR observed across all classes. This rise in MTTR may be attributed to resource constraints and the escalating complexity of the system.

In practice, the MTTR is anticipated to be lowest for service requests, such as a password change, which typically require minimal time for resolution. Incidents, being unforeseen, can vary widely in their resolution time, ranging from swift resolutions to more protracted resolutions. Changes are expected to have the lengthiest MTTR as they necessitate thorough testing and deployment processes.

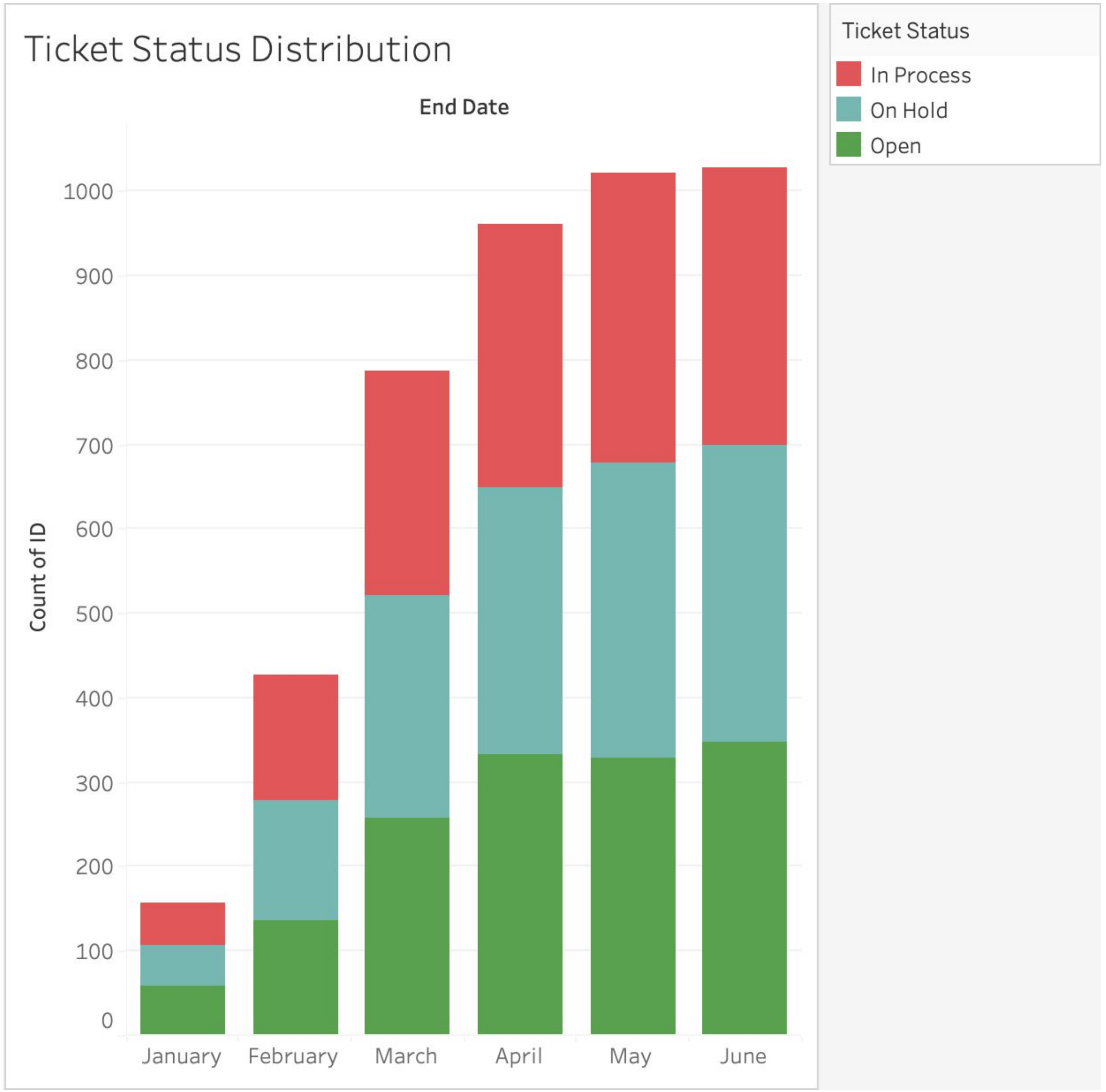


Ticket Status Distribution Over Time

The proportion of ticket duration, categorized by status (in process, on hold, and open), appears to be even each month.

Potential factors contributing to this uniform distribution of duration may include resource constraints and inefficiencies within the ticketing system.

From a business timeline perspective, this even distribution is unrealistic. Ideally, tickets would remain open for a brief period before progressing to the "in process" status. The duration of being in process would then align with the complexity of the ticket class. Tickets placed on hold would likely have the longest duration, as this status indicates that the ticket cannot proceed to deployment and is not actively being addressed.

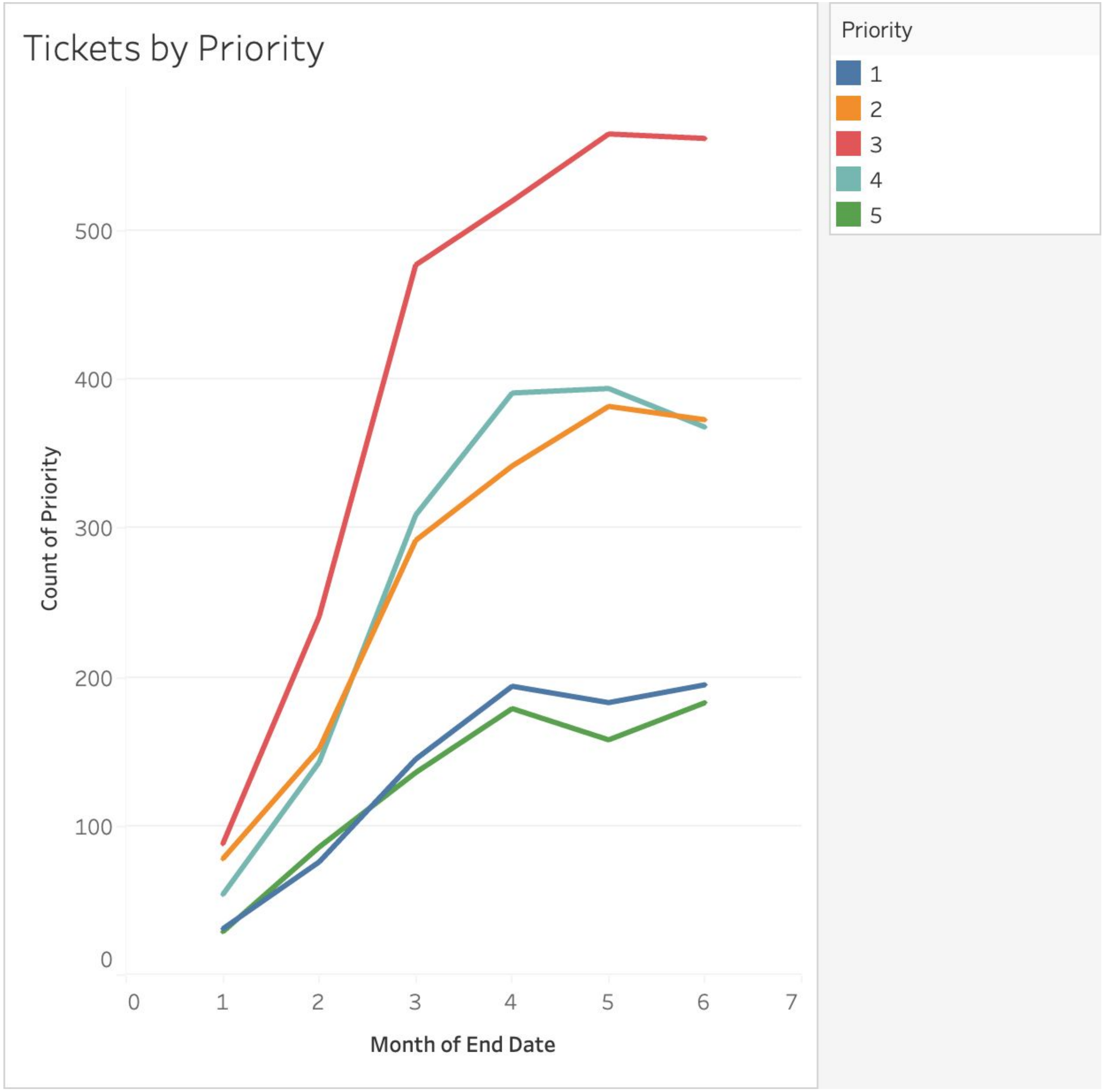


Number of Tickets by Priority

This visualization indicates a progressive increase in tickets over time, with a notable predominance of tickets rated as priority 3. This observation aligns with both the random nature of ticket generation and the impact/urgency priority matrix, where the majority of cells correspond to a priority rating of 3.

The prevalence of priority 3 tickets could potentially be attributed to a system that has been in operation for an extended period, undergoing continual improvements through updates rather than major overhauls.

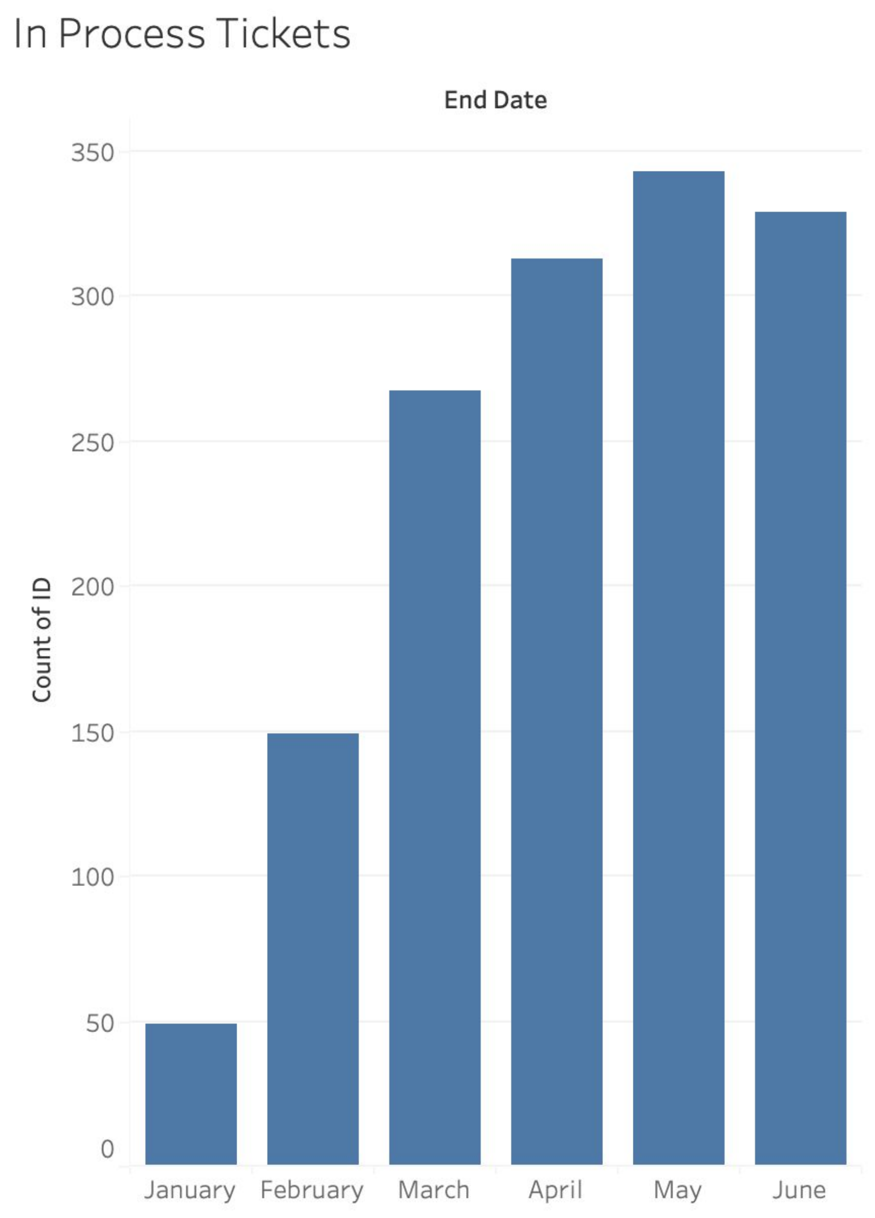
However, it is important to note that this representation may not necessarily reflect real-world data. In practice, while priority 3 tickets may still constitute a significant portion of the total, priority 1 (highest priority) tickets would ideally be the least numerous. Conversely, priority 5 tickets would likely outnumber priority 1 and 2 tickets.



In Process Tickets

The quantity of tickets in the "in process" status demonstrates a consistent increase over time, peaking in May before declining in June. This pattern suggests a mounting backlog of service tickets, highlighting the importance of a system that efficiently processes tickets upon receipt, aiming for swift deployment or closure.

The backlog may stem from inadequacies in the ticket service system or a shortage of staff dedicated to ticket resolution.



Conclusion

The analysis of the ticketing data reveals a consistent pattern indicating a company facing challenges of resource constraints and an inefficient ticketing system. This conclusion is supported by various key performance indicator visualizations, including the distribution of ticket duration, the mean time to repair, and the prioritization of tickets.

To address the randomness in ticket generation and enhance the realism of the dataset, consulting industry trends and implementing weighted logic for random selections would be beneficial. This approach can help in creating a more representative dataset for further analysis and decision-making processes.