Travel Tide

By: Satakshi Salaria

Summary

This exploratory data analysis was conducted for Travel Tide, a hypothetical online travel agency. Our main goal is to identify appropriate benefits to improve client retention through customer segmentation. This notebook illustrates the initial stage of our study using Python, which includes data interpretation, cleaning, and preparation for next analytical stages. A SQL query is produced to connect four tables:

- **Users:** user demographic information
- **Sessions:** information about individual browsing sessions
- **Flights:** information about purchased flights
- **Hotels:** information about purchased hotel stays

Context

We only included people who had more than 7 sessions within the same time period, and we included sessions from after the New Year's holiday (2023-01-04) to the latest date in the database that was still accessible (2023-07-23). This was done at the Marketing Manager's request. This enables us to examine consumer behaviour within a predetermined time range.

These are the perks most likely to attract customers:

• Free hotel meal
• Free checked bag
• No cancellation fees
• Exclusive discounts
• 1-night free hotel with a flight

Test Parameters

We ran this experiment with the following parameters:

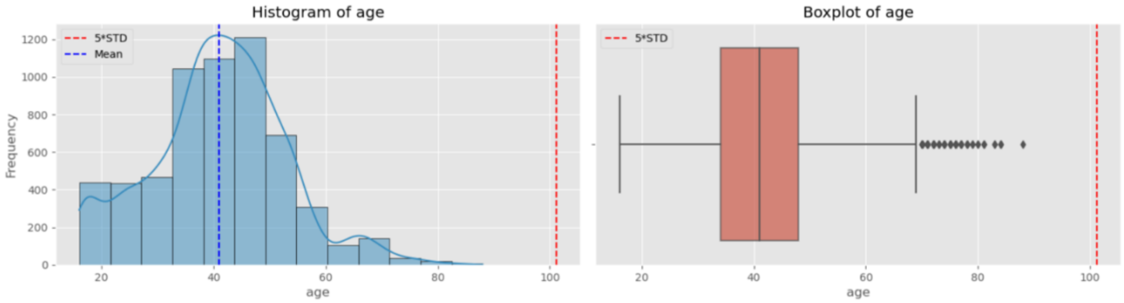
- Date Range: January 4’2023 – July 7’2023(Last date available)
- Total Users: 5998
- Included users with more than 7 sessions
- Tables: Users, Sessions, Flights, Hotels

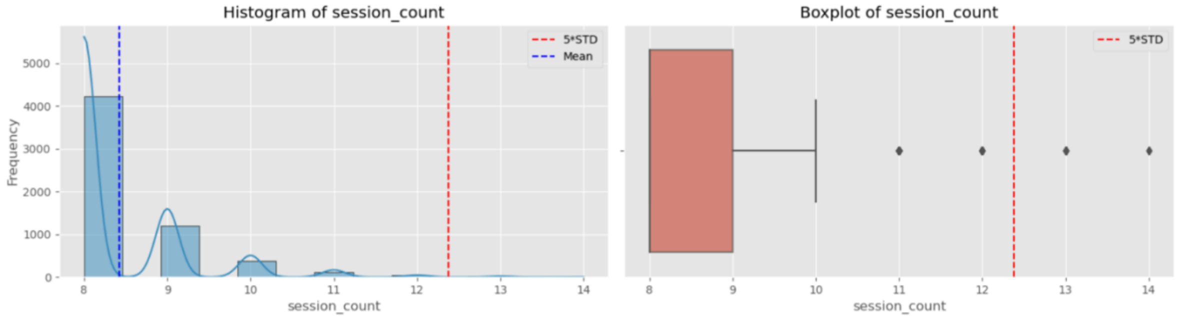
Outlier Treatment:

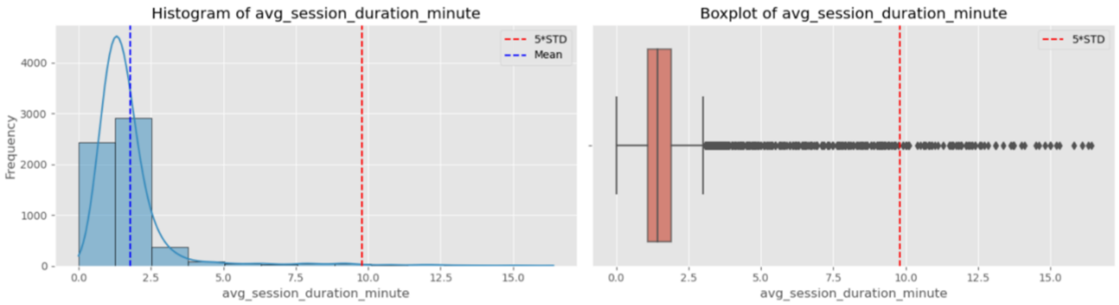
Examining the statistical summary, it's apparent that the majority of columns have mean and median values that are fairly similar. To further explore the data's distribution and identify any potential outliers, we will visualize it using both histograms and box plots. This approach aims to provide a more comprehensive view of our data's distribution characteristics.

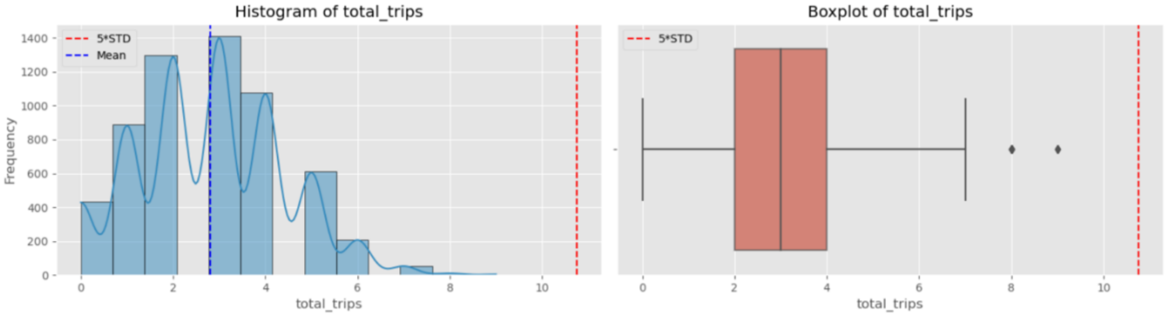
- I've established a criteria for identifying outliers, which involves considering data points that are 5 standard deviations distant from the mean of the sample. This criteria is significant because, assuming a normal distribution, such extreme data points occur at a rarity of approximately one in 500 million instances. Hence, we can confidently assert that these data points are highly unlikely outliers.

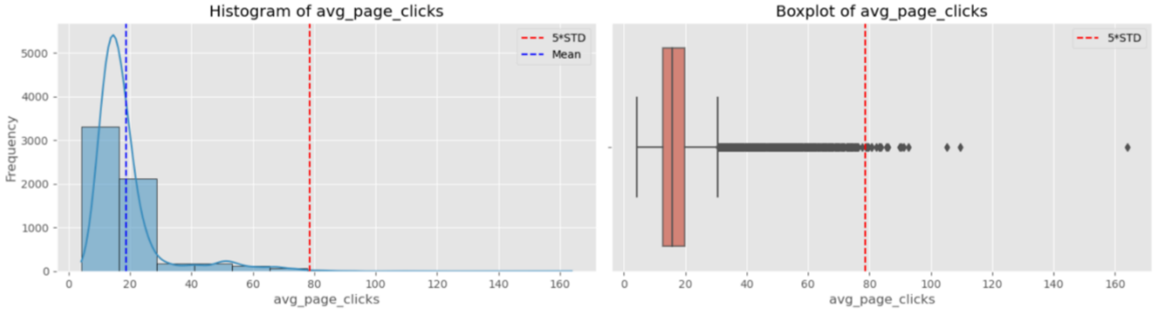
- It's worth noting that our choice of using a threshold of 5 times the standard deviation (5*STD) is relatively permissive. Data points that fall beyond this range are considered highly likely to be outliers, indicating that they significantly deviate from the norm within our dataset.

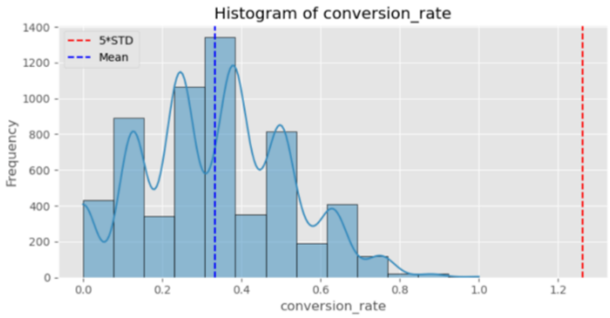

Histogram of age
5*STD
Mean
Frequency
0
200
400
600
800
1000
1200
20
40
60
80
100
age
Boxplot of age
5*STD
20
40
60
80
100
age

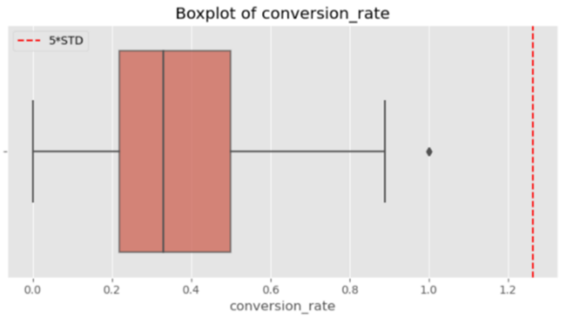

Histogram of session_count
5*STD
Mean
Frequency
0
1000
2000
3000
4000
5000
8
9
10
11
12
13
14
session_count
Boxplot of session_count
5*STD
8
9
10
11
12
13
14
session_count

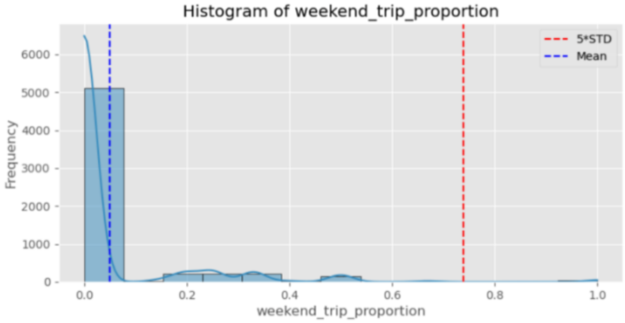

Histogram of avg_session_duration_minute
5*STD
Mean
Frequency
0
1000
2000
3000
4000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
avg_session_duration_minute
Boxplot of avg_session_duration_minute
5*STD
0.0
2.5
5.0
7.5
10.0
12.5
15.0
avg_session_duration_minute

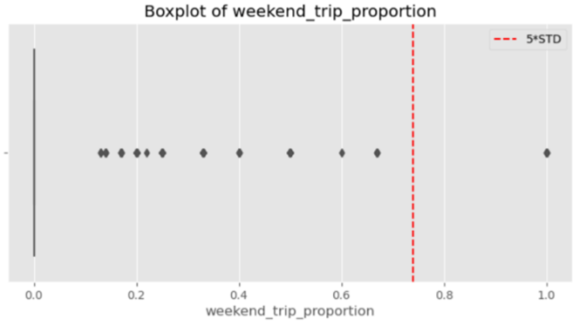

Histogram of total_trips
5*STD
Mean
Frequency
0
200
400
600
800
1000
1200
1400
0
2
4
6
8
10
total_trips
Boxplot of total_trips
5*STD
0
2
4
6
8
10
total_trips

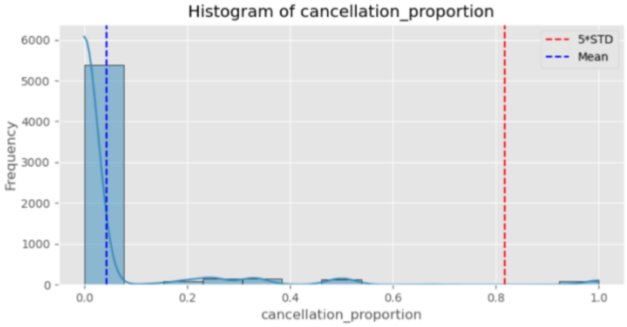

Histogram of avg_page_clicks
5*STD
Mean
Frequency
0
1000
2000
3000
4000
5000
0
20
40
60
80
100
120
140
160
avg_page_clicks
Boxplot of avg_page_clicks
5*STD
0
20
40
60
80
100
120
140
160
avg_page_clicks

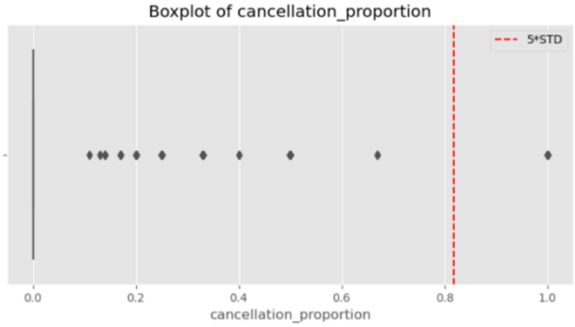

Histogram of conversion_rate
5*STD
Mean
Frequency
0
200
400
600
800
1000
1200
1400
0.0
0.2
0.4
0.6
0.8
1.0
1.2
conversion_rate

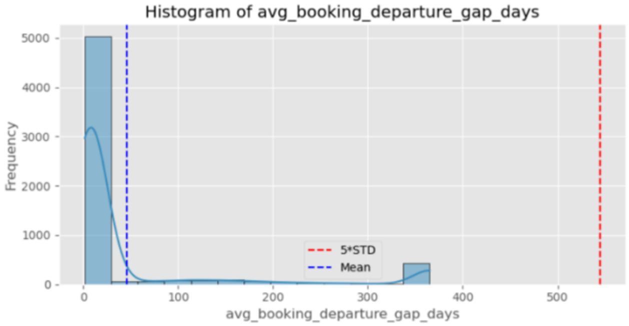

Boxplot of conversion_rate
5*STD
0.0
0.2
0.4
0.6
0.8
1.0
1.2
conversion_rate

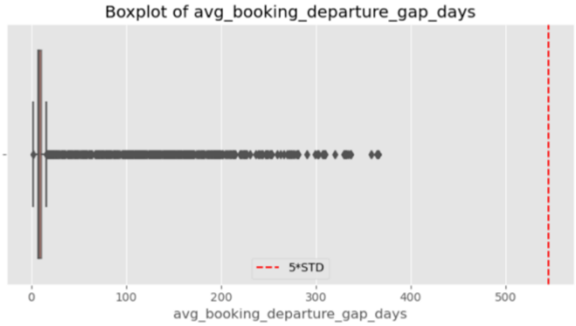

Histogram of weekend_trip_proportion
5*STD
Mean
Frequency
0
1000
2000
3000
4000
5000
6000
0.0
0.2
0.4
0.6
0.8
1.0
weekend_trip_proportion

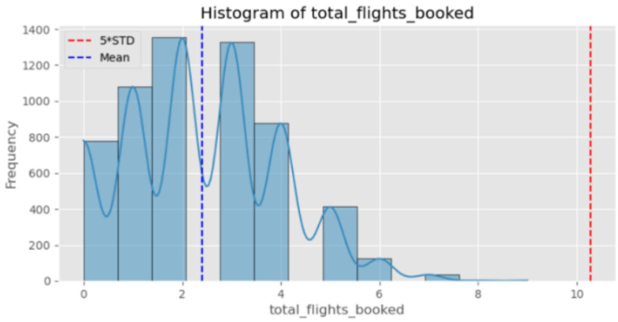

Boxplot of weekend_trip_proportion
5*STD
0.0
0.2
0.4
0.6
0.8
1.0
weekend_trip_proportion

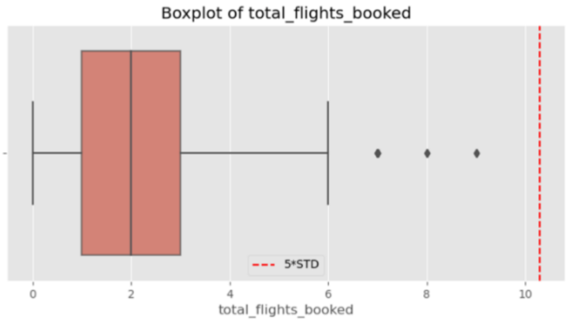

Histogram of cancellation_proportion
5*STD
Mean
Frequency
0
1000
2000
3000
4000
5000
6000
0.0
0.2
0.4
0.6
0.8
1.0
cancellation_proportion

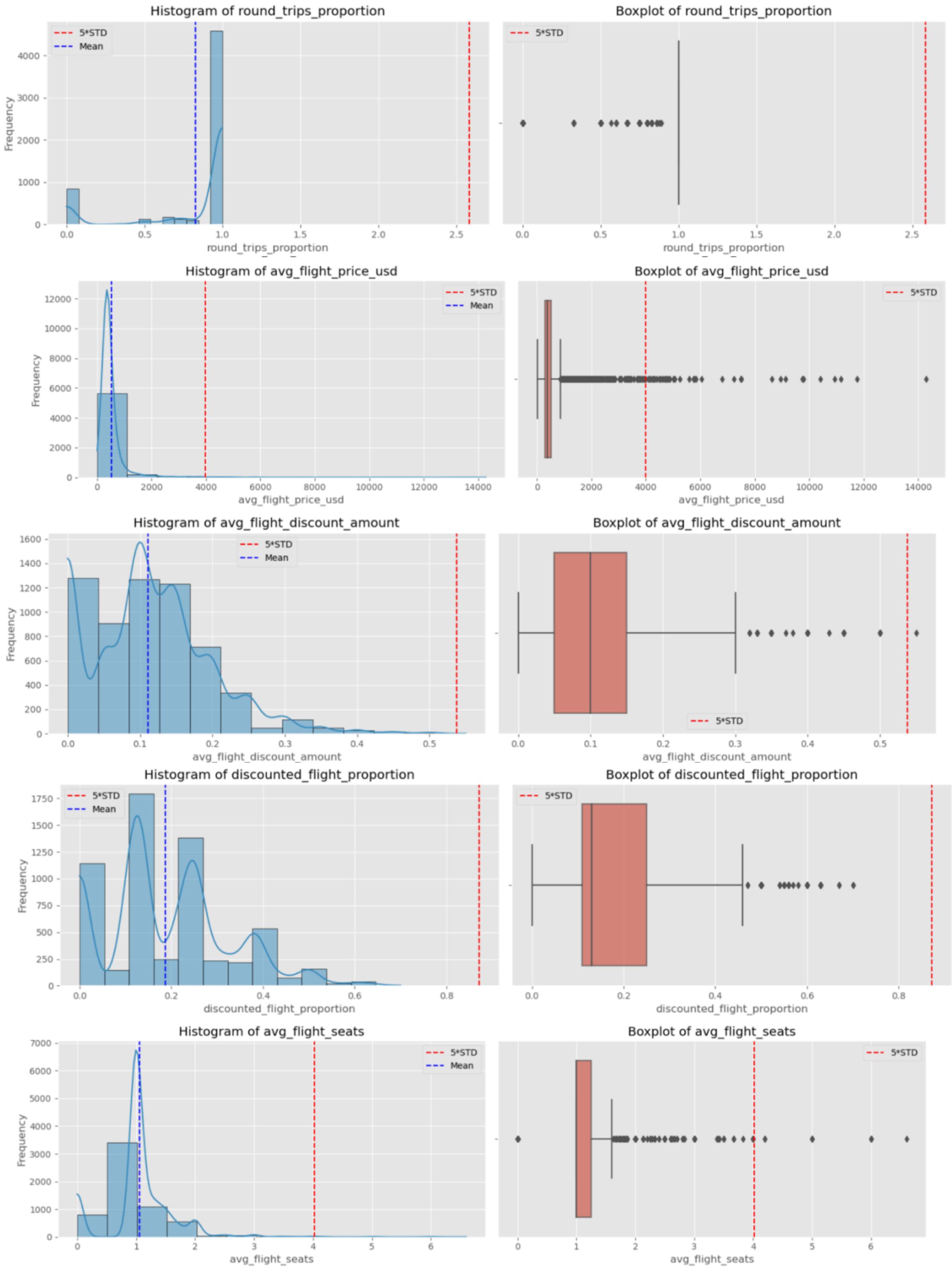

Boxplot of cancellation_proportion
5*STD
0.0
0.2
0.4
0.6
0.8
1.0
cancellation_proportion

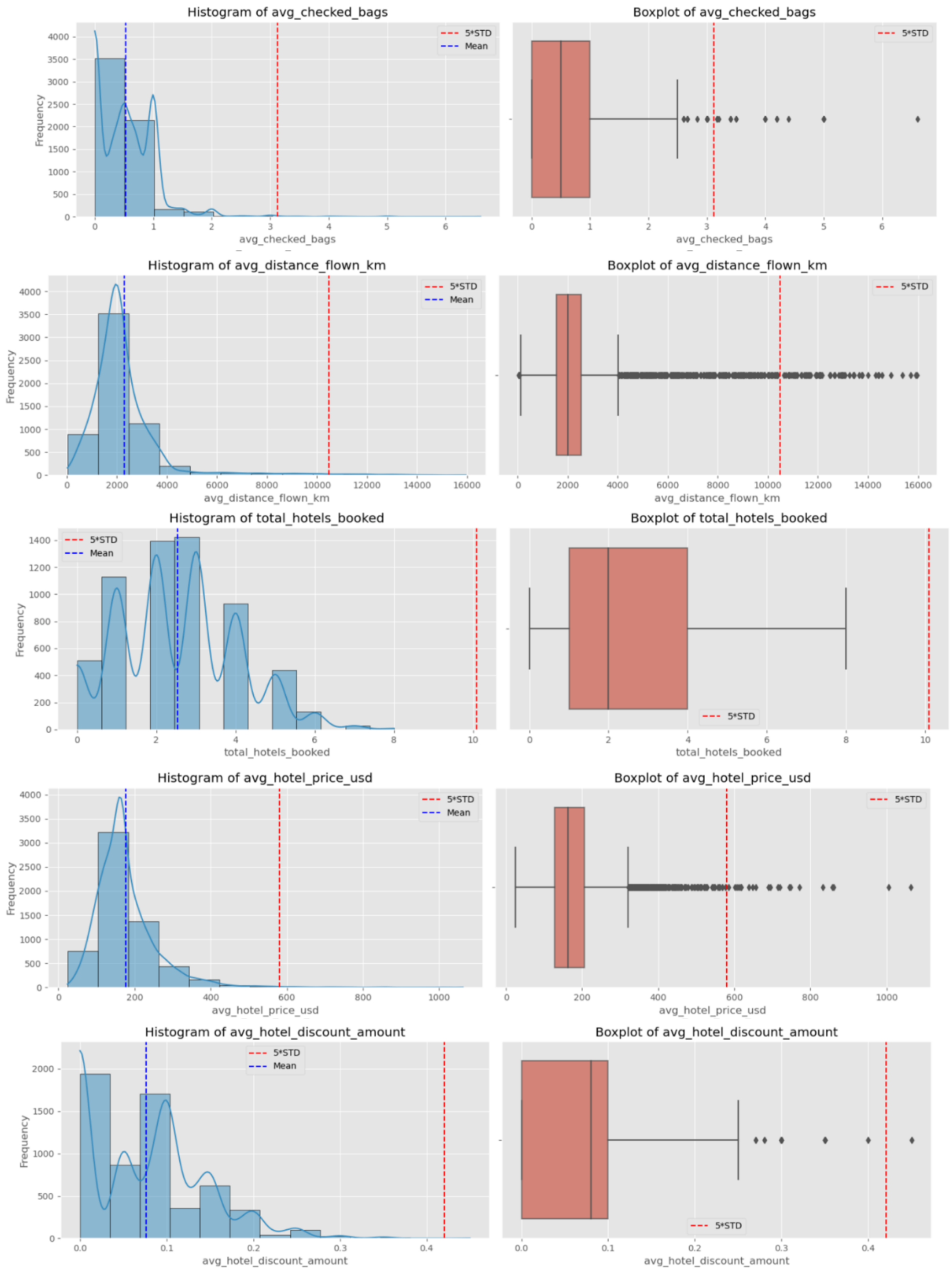

Histogram of avg_booking_departure_gap_days
5*STD
Mean
Frequency
0
1000
2000
3000
4000
5000
0
100
200
300
400
500
avg_booking_departure_gap_days


Boxplot of avg_booking_departure_gap_days
5*STD
0
100
200
300
400
500
avg_booking_departure_gap_days


Histogram of total_flights_booked
5*STD
Mean
Frequency
0
200
400
600
800
1000
1200
1400
0
2
4
6
8
10
total_flights_booked


Boxplot of total_flights_booked
5*STD
0
2
4
6
8
10
total_flights_booked


Histogram of round_trips_proportion
5*STD
Mean
Frequency
0
1000
2000
3000
4000
0.0
0.5
1.0
1.5
2.0
2.5
round_trips_proportion
Boxplot of round_trips_proportion
5*STD
0.0
0.5
1.0
1.5
2.0
2.5
round_trips_proportion
Histogram of avg_flight_price_usd
5*STD
Mean
Frequency
0
2000
4000
6000
8000
10000
12000
0
2000
4000
6000
8000
10000
12000
14000
avg_flight_price_usd
Boxplot of avg_flight_price_usd
5*STD
0
2000
4000
6000
8000
10000
12000
14000
avg_flight_price_usd
Histogram of avg_flight_discount_amount
5*STD
Mean
Frequency
0
200
400
600
800
1000
1200
1400
1600
0.0
0.1
0.2
0.3
0.4
0.5
avg_flight_discount_amount
Boxplot of avg_flight_discount_amount
5*STD
0.0
0.1
0.2
0.3
0.4
0.5
avg_flight_discount_amount
Histogram of discounted_flight_proportion
5*STD
Mean
Frequency
0
250
500
750
1000
1250
1500
1750
0.0
0.2
0.4
0.6
0.8
discounted_flight_proportion
Boxplot of discounted_flight_proportion
5*STD
0.0
0.2
0.4
0.6
0.8
discounted_flight_proportion
Histogram of avg_flight_seats
5*STD
Mean
Frequency
0
1000
2000
3000
4000
5000
6000
7000
0
1
2
3
4
5
6
avg_flight_seats
Boxplot of avg_flight_seats
5*STD
0
1
2
3
4
5
6
avg_flight_seats


Histogram of avg_checked_bags

Boxplot of avg_checked_bags

Histogram of avg_distance_flown_km

Boxplot of avg_distance_flown_km

Histogram of total_hotels_booked

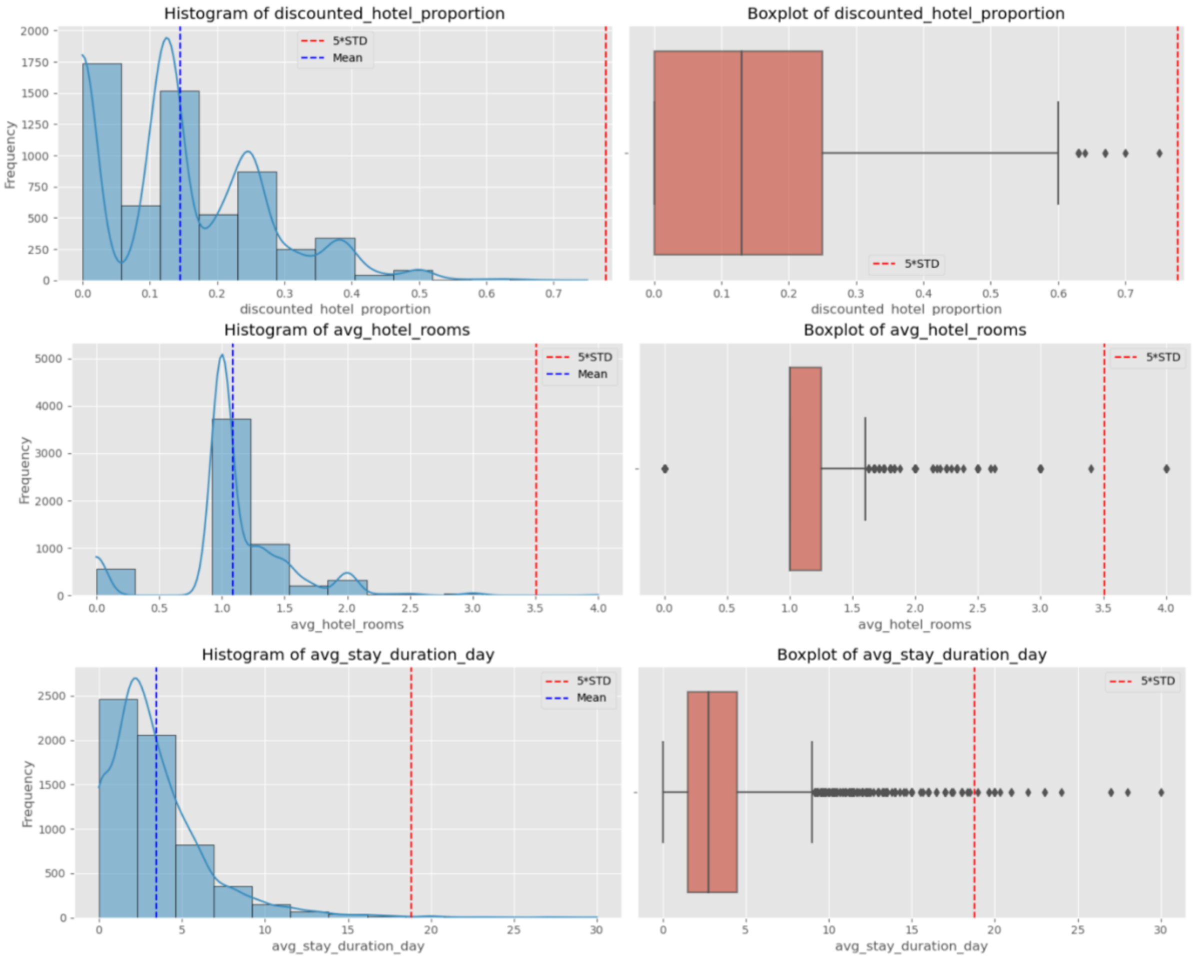
Boxplot of total_hotels_booked

Histogram of avg_hotel_price_usd

Boxplot of avg_hotel_price_usd

Histogram of avg_hotel_discount_amount

Boxplot of avg_hotel_discount_amount



Upon closer examination, we have identified several columns that contain data points surpassing this 5*STD threshold. Notably, in the case of columns like "avg_session_duration_minute," a substantial portion of the data points extend beyond this threshold. This suggests that there are significant variations or extreme values within these columns, which may warrant further investigation and potentially outlier treatment in our data analysis.

Outlier Removal:

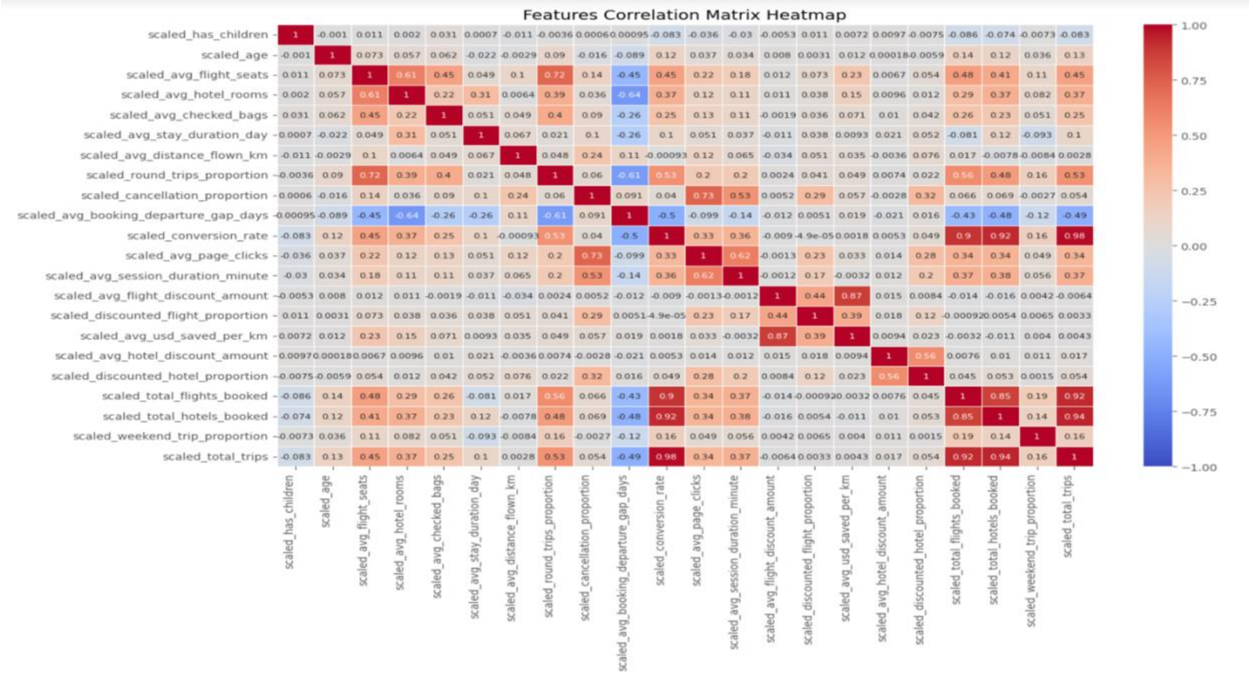
I employ a masking method to simplify our process of identifying and handling outliers:

- I start with a universal mask where all data points are initially marked as valid (set to True).
- For each individual column, I generate a column-specific mask that identifies potential outliers.
- These column-specific masks are combined with the universal mask, refining our selection.

- Only data points that are consistently marked as True across all columns are kept, ensuring a thorough removal of outliers.
- This approach not only effectively eliminates outliers but also maintains the integrity of our core dataset.
- In total, 329 records (5.49% of the dataset) were identified as outliers.

Fitting the k-means & choosing best value for k

Our goal is to prepare our data for clustering analysis. To do this, we carefully choose specific features and create a heatmap displaying their correlation matrix. This heatmap provides insights into the relationships between these chosen features. This step holds significant importance as it helps enhance the clustering algorithm's performance by uncovering and, if necessary, resolving issues related to feature correlation.



After conducting a heatmap analysis and carefully considering the process of feature selection, I have identified the following set of features that we will utilize in our clustering analysis:

- **scaled_avg_hotel_rooms:** This feature represents the average number of hotel rooms and will be employed to categorize users who are eligible to receive the Free Hotel Meal perk.
- **scaled_avg_checked_bags:** Reflecting the average number of checked bags, this feature will be used to determine which travelers meet the criteria for the Free Checked Bag perk.
- **scaled_cancellation_proportion:** This feature measures the proportion of cancellations and will be used to identify customers eligible for the No Cancellation Fees perk.
- **scaled_conversion_rate:** The conversion rate feature will be utilized to differentiate users who qualify for the Exclusive Discounts perk.

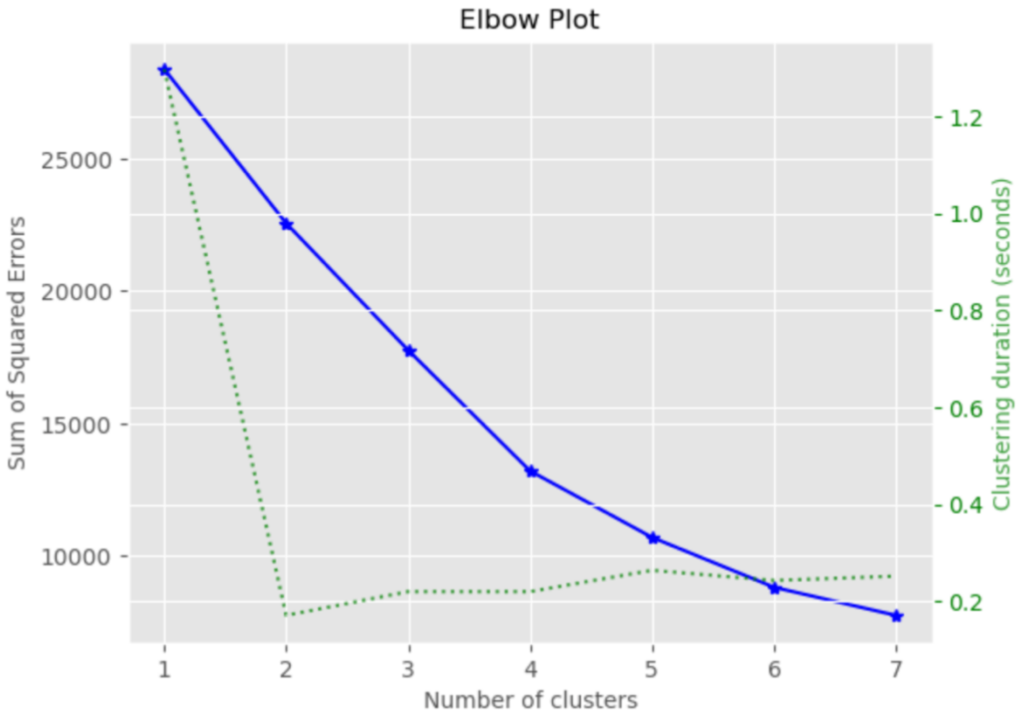
- **scaled_weekend_trip_proportion:** This feature, indicating the proportion of weekend trips, will guide our selection of users eligible for the 1-night free hotel with a flight perk.

Fitting K-means and finding appropriate amount of customer types

After conducting an analysis of the within-cluster sum of squared errors (WCSS) for various cluster numbers, we have observed interesting trends. The most substantial reductions in WCSS occur when transitioning from 5 to 6 clusters and from 6 to 7 clusters. This suggests that, from a purely data-driven perspective using the elbow method, having either 6 or 7 clusters might be considered optimal.

However, it's crucial to consider TravelTide's specific objective of categorizing customers into five distinct groups to offer five unique perks. This aligns with the company's business strategy. While the reduction in WCSS between 4 and 5 clusters is smaller in comparison, it is still significant. This indicates that having a fifth cluster captures important variations in the data.

Given the need to strike a balance between data-driven insights from the elbow method and the company's specific business requirements, the decision to proceed with five clusters is justified. This approach not only aligns with TravelTide's marketing strategy but also ensures that the rewards program is tailored to diverse customer behaviors and preferences, thereby enhancing its potential for success.



Observations:

- The silhouette score is at its peak when we have 7 clusters. This suggests that the data could be most effectively divided into seven distinct groups, considering how similar data points are within clusters and how different they are from points in other clusters.
- As we transition from 4 to 5 clusters, the silhouette score declines, but it rises again when moving to 6 and 7 clusters.
- When solely considering the silhouette scores, it appears that having 7 clusters is the optimal choice. However, our primary business objective is to categorize customers into five distinct groups for the perks program, introducing a trade-off:
- 7 Clusters offers the highest data clustering quality (highest silhouette score).
- 5 Clusters aligns with the business goal but has a slightly lower silhouette score compared to 7 clusters.

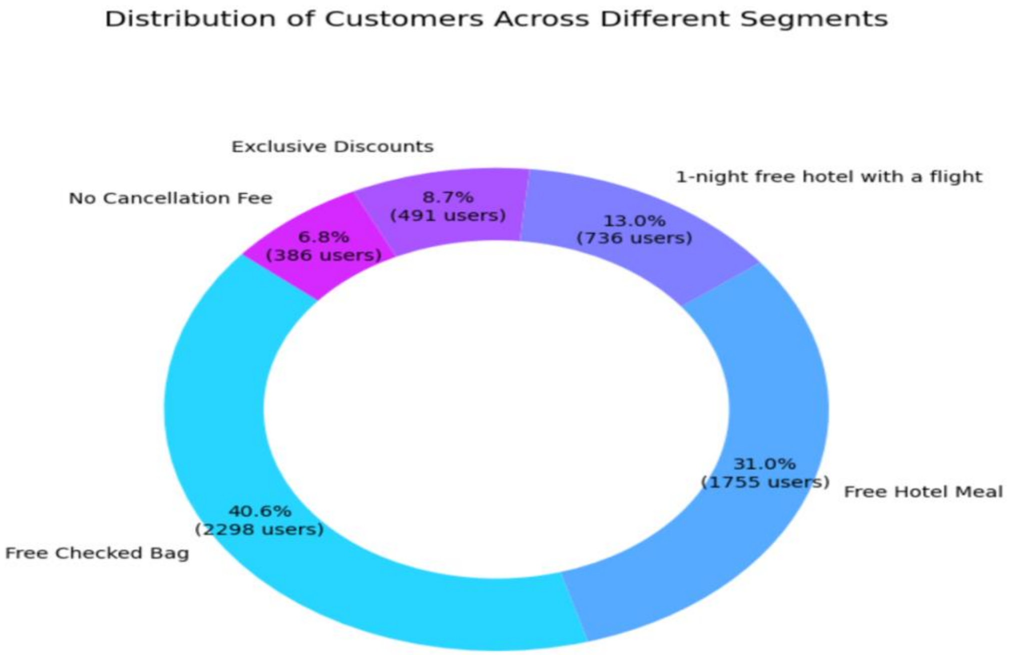
Despite the slightly higher silhouette score for 7 clusters, the difference compared to 5 clusters (0.394 vs. 0.351) is relatively small. A silhouette score of 0.351 is still respectable and could result in more meaningful customer segments for our marketing objectives. Balancing data-driven quality and business alignment is a key consideration in our decision-making process.

Segmentation:

Distribution of Customers:

- **Free Checked Bag:** This category holds the largest number of consumers, underscoring that a significant portion of our customer base appreciates a feature that offers complimentary checked bags when booking travel. Our analysis indicates that 40.6% of consumers, equivalent to 2298 users, favor the inclusion of free checked bags.
- **Free Hotel Meal:** Following the first segment, it appears that 31.0% (1755 users) of consumers perceive a complimentary hotel meal to be an attractive bonus.
- **1-night Free Hotel with a Flight:** Comprising 13% of the total user base (735 users), this group represents the third-largest segment. This statistic implies the existence of a considerable number of frequent travelers who would likely welcome the offer of a 1-night free hotel stay with a flight.
- **Exclusive Discounts:** Within this category, which accounts for 8.7% of our user base (comprising 491 users), it is evident that a significant number of customers prioritize pricing and would find exclusive discounts to be valuable.

- **No Cancellation Fee:** Comprising just 6.8% of our user base (equivalent to 386 users), this represents the smallest segment. It suggests that a limited portion of our customers highly values the flexibility offered in terms of cancellations.

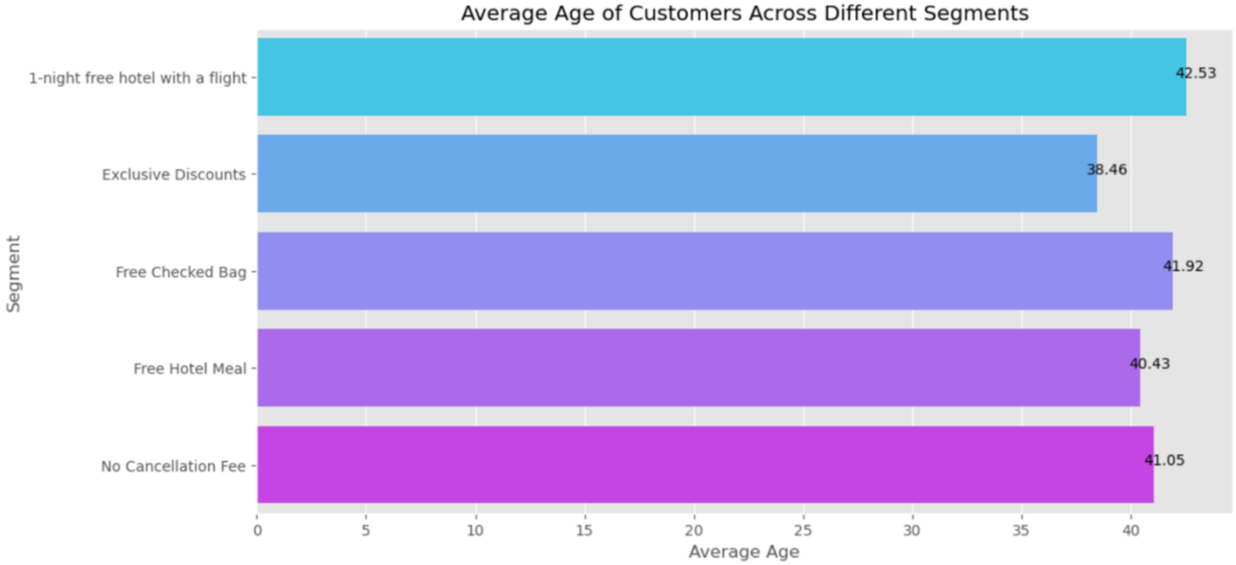

Distribution of Customers Across Different Segments

Age:

Understanding the age demographics associated with these perk preferences can be instrumental in crafting targeted marketing strategies and tailoring perk offerings to better meet the expectations and desires of specific customer groups. This knowledge enables businesses to enhance their customer engagement and satisfaction levels. The bar chart illustrates the average age of customers within various segments. It's noticeable that the average ages across all segments are quite similar.

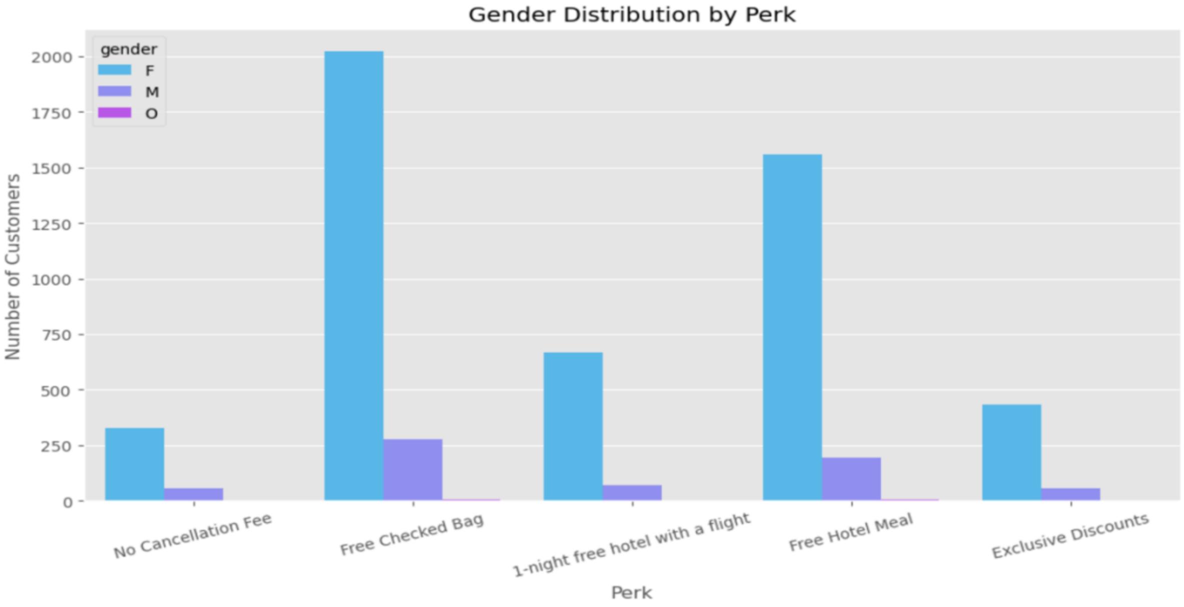
- **1-night Free Hotel with a Flight:** This segment comprises customers who are particularly interested in receiving a one-night free hotel stay when booking a flight. The relatively higher average age of 42.53 suggests that older customers may be more inclined toward this perk, possibly valuing the added comfort and relaxation associated with a hotel stay.
- **Exclusive Discounts:** Customers in this segment are characterized by their preference for exclusive discounts on travel-related services. The lower average age of 38.46 indicates that younger individuals, likely budget-conscious travelers, find value in obtaining discounts, potentially to save on their overall travel expenses.
- **Free Checked Bag:** This segment includes customers who appreciate the benefit of having a checked bag included in their travel arrangements. With an average age of 41.92, it suggests that a slightly older demographic is more inclined toward the convenience of not having to pay extra for checked baggage.
- **Free Hotel Meal:** Customers in this category prioritize receiving complimentary hotel meals as part of their travel package. The average age of 40.43 indicates that customers of varying age groups value the convenience and cost savings associated with complimentary meals.

- **No Cancellation Fee:** This segment represents customers who highly value the flexibility of not incurring cancellation fees when altering their travel plans. With an average age of 41.05, it suggests that individuals across different age groups appreciate the peace of mind and flexibility provided by this perk.



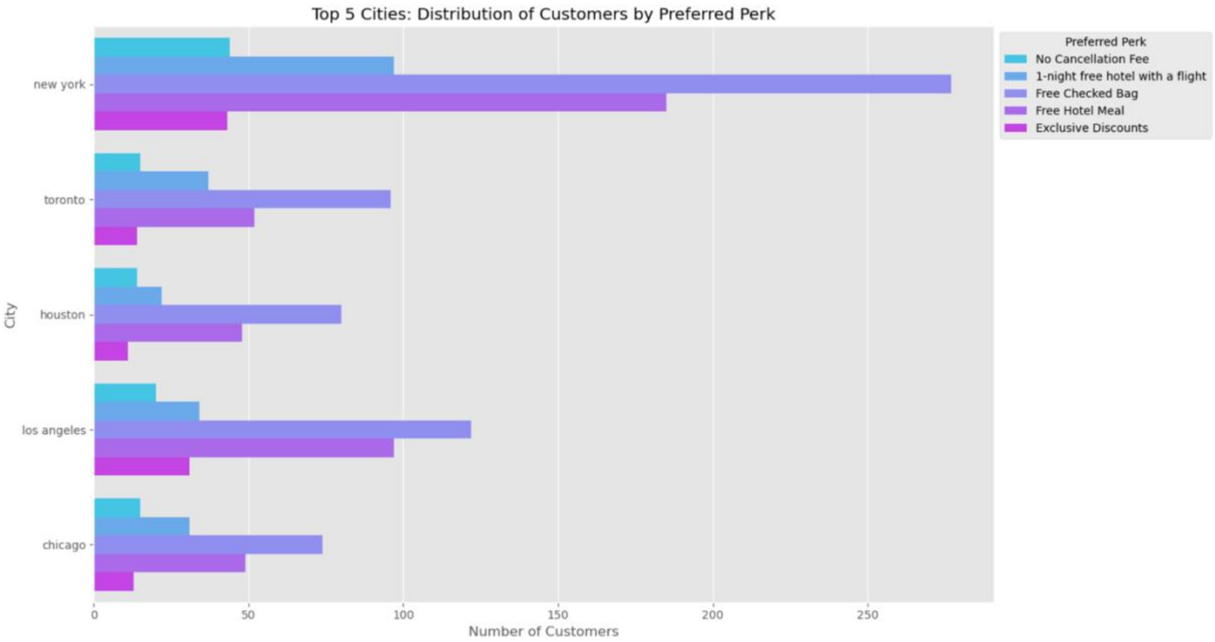
Gender:

Across all segments, there is a noticeable trend of a higher number of female users showing interest in each perk compared to male users. This suggests that the appeal of these perks is more prominent among female customers in the dataset. Understanding this gender distribution can be valuable when designing targeted marketing strategies or making adjustments to the perks to better cater to the preferences of both male and female audiences.



Geographical:

It's evident that, for these cities (Toronto, Houston, Los Angeles, and Chicago), there is a consistent pattern in the preferences for the top three perks, with "Free Checked Bag," "Free Hotel Meal," and "1-night Free Hotel with a Flight" being the most favored perks. This pattern mirrors that of New York City, indicating a similarity in customer preferences for these specific perks across these cities. Understanding these preferences can be valuable for tailoring marketing strategies and perk offerings in these locations.



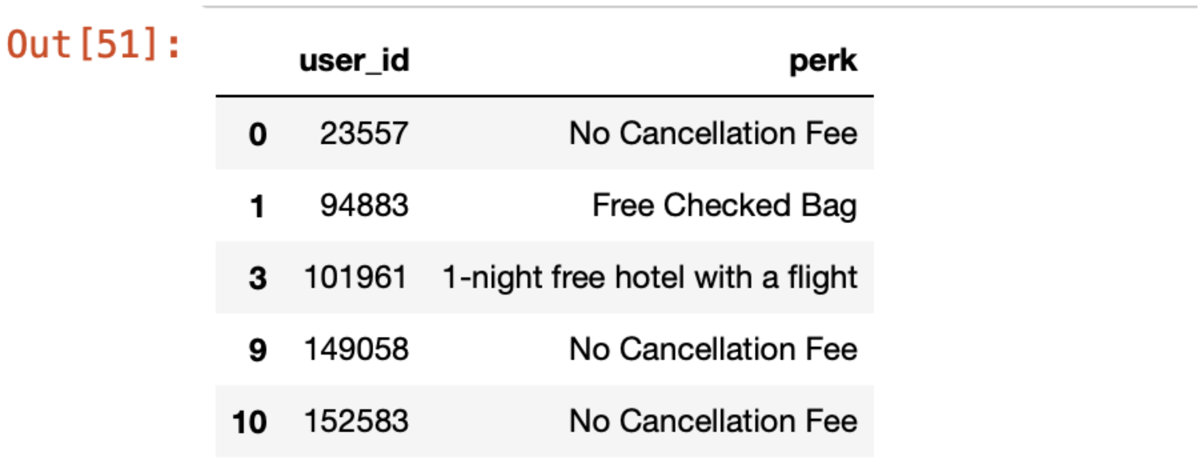
Recommendations:

- Given the overlap between certain perks, consider bundling perks or offering tiered rewards. For instance, a combined perk of "Free Checked Bag + 1-night Free Hotel with a Flight" could be tested for its appeal.
- For the "No Cancellation Fee" segment, consider additional perks or incentives that cater to their need for flexibility, such as "Flexible Dates" or "Priority Rescheduling".
- Design marketing campaigns that cater to the specific behaviors and preferences of each segment. For instance, target the "Exclusive Discounts" segment with special limited time offers to boost their conversion rate.
- For the gender disparity observed, consider gender-specific campaigns or investigate the reasons for the disparity to ensure a balanced appeal.
- Given the consistent perk preferences across major cities, consider a uniform rewards program rollout in these cities. However, monitor regional preferences and be ready to adjust based on feedback.

- Keep an eye on KPIs such as engagement rate, conversion rate, and customer lifetime value to measure the success of these personalized campaigns.
- Recognize and engage with segments that exhibit high-value behaviors, such as booking for larger groups (reflected in higher "avg_hotel_rooms") or frequent flying. Personalized loyalty programs or premium services could be offered to these customers.
- Maintain a vigilant eye on key performance indicators (KPIs) like engagement rate, conversion rate, and customer lifetime value to gauge the effectiveness of these personalized campaigns.
- Given the consistent preference for certain perks across major cities, contemplate the possibility of implementing a standardized rewards program in these locations. However, remain flexible and responsive to regional variations in preferences, adapting strategies based on feedback and local insights.
- Identify and engage with segments that exhibit valuable behaviors, such as those booking for larger groups (indicated by higher "avg_hotel_rooms") or frequent flyers. Consider offering personalized loyalty programs or premium services to incentivize and retain these high-value customers.

Conclusion:

The segmentation analysis has provided invaluable insights into the diverse preferences and behaviors of TravelTide's customer base. By aligning perks with these insights and continuously refining the offerings, TravelTide has the potential to enhance customer satisfaction, foster loyalty, and drive business growth. Here below is the first few rows of the final dataframe including customer's user_id and their corresponding segment label.



Out[51]:

	user_id	perk
0	23557	No Cancellation Fee
1	94883	Free Checked Bag
3	101961	1-night free hotel with a flight
9	149058	No Cancellation Fee
10	152583	No Cancellation Fee

Appendix

SQL Query

Cohort definition:

```
WITH cohort_users AS (
  SELECT user_id
  FROM sessions
  WHERE session_start > '2023-01-04'
  GROUP BY user_id
  HAVING COUNT(session_id) > 7
),
```

Determining the distance between two airports:

```
VincentyDistance AS (
    SELECT
        s.user_id,
        f.trip_id,
        6378137.0 AS a, -- semi-major axis in meters
        6356752.3142 AS b, -- semi-minor axis in meters
        1/298.257223563 AS f, -- flattening
```

Latitude and longitude conversion from degrees to radians:

```
        RADIANS(u.home_airport_lat) AS lat1,
        RADIANS(u.home_airport_lon) AS lon1,
        RADIANS(f.destination_airport_lat) AS lat2,
        RADIANS(f.destination_airport_lon) AS lon2
    FROM sessions AS s
    JOIN flights AS f ON s.trip_id = f.trip_id
    JOIN users AS u ON s.user_id = u.user_id
    WHERE s.user_id IN (SELECT user_id FROM cohort_users)
),
```

Utilizing the following techniques to determine distance:

```
VincentyInitialComputations AS (
    SELECT user_id,
        trip_id,
        a,
        b,
        f,
        lat1,
        lon1,
        lat2,
        lon2,
        -- Compute delta longitude
        lon2 - lon1 AS L,
        ATAN((1 - f) * TAN(lat1)) AS U1,
        ATAN((1 - f) * TAN(lat2)) AS U2
    FROM VincentyDistance
),

VincentyIntermediateComputations AS (
    SELECT
        user_id,
        trip_id,
        a,
        b,
        f,
        lat1,
        lon1,
        lat2,
        lon2,
        L,
        U1,
        U2,
        SQRT(
            (COS(U2)*SIN(L)) * (COS(U2)*SIN(L)) +
            (COS(U1)*SIN(U2) - SIN(U1)*COS(U2)*COS(L)) * (COS(U1)*SIN(U2) -
SIN(U1)*COS(U2)*COS(L))
        ) AS sinSigma,
        SIN(U1)*SIN(U2) + COS(U1)*COS(U2)*COS(L) AS cosSigma
    FROM VincentyInitialComputations
),

VincentyComputations AS (
    SELECT
        user_id,
        trip_id,
        a,
        b,
        f,
        lat1,
        lon1,
        lat2,
        lon2,
        L,
        U1,
        U2,
        sinSigma,
        cosSigma,
        COS(U1)*COS(U2)*SIN(L) AS sinAlpha,
        SQRT(1 - COS(U1)*COS(U2)*SIN(L)*COS(U1)*COS(U2)*SIN(L)) AS cos2Alpha,
        cosSigma - 2*SIN(U1)*SIN(U2) / (SQRT(1 -
COS(U1)*COS(U2)*SIN(L)*COS(U1)*COS(U2)*SIN(L))) AS cos2SigmaM,
        a*1.0/(1.0-f) AS u_sq
    FROM VincentyIntermediateComputations
),
```

Aggregate and Merge Data:

```
aggregated_data AS (
    SELECT
        s.user_id,
            -- total number of user's sessions
        COUNT(DISTINCT s.session_id) AS session_count,
            -- average duration of sessions in minute
            ROUND(AVG(EXTRACT(MINUTE FROM (session_end - session_start))),2) AS
avg_session_duration_minute,
            -- average number of clicks in all browsing sessions
            ROUND(AVG(page_clicks),2) AS avg_page_clicks,
            -- total number of booked trips
            COUNT(DISTINCT CASE WHEN NOT cancellation THEN s.trip_id END) AS total_trips,
            -- conversion rate, dividing the number of booked trips (in case of no cancellation) by total number of browising sessions
            ROUND(
              CASE WHEN COUNT(DISTINCT s.session_id) > 0 THEN
                 1.0 * COUNT(DISTINCT CASE WHEN NOT cancellation THEN s.trip_id END) / COUNT(DISTINCT s.session_id)
                 ELSE 0 END
              ,2) AS conversion_rate,
            -- Cancellation proportion, returns NULL for users who didn't book any trip to not get division by zero error
            ROUND(
              1.0 * COUNT(DISTINCT CASE WHEN cancellation THEN s.trip_id END) /
                 NULLIF(COUNT(DISTINCT CASE WHEN NOT cancellation THEN s.trip_id END), 0)
              ,2) AS cancellation_proportion,
            -- calculating the booking to departure time gap in seconds and then days by dividing by 86400
            ROUND(
                 AVG(EXTRACT(EPOCH FROM (f.departure_time - s.session_end)) / 86400)
              ,2) AS avg_booking_departure_gap_days_flights,
            -- As some users only booked hotels, I add another calculation considering hotel check_in_time and later in Python will merge these two columns
            ROUND(
                 AVG(EXTRACT(EPOCH FROM (h.check_in_time - s.session_end)) / 86400)
              ,2) AS avg_booking_departure_gap_days_hotels,
            -- total number of flights
        COUNT(DISTINCT CASE WHEN flight_booked THEN s.trip_id END) AS total_flights_booked,
            /* Weekend trips proportion, to distinguish weekened gateway travelers,
             when the departure time is on Fridays or Saturdays, and return_time is on Sundays or Mondays
            and the duration of the trip is less than three days
            */
            ROUND(
              CASE WHEN COUNT(DISTINCT CASE WHEN NOT cancellation THEN s.trip_id END) > 0 THEN
                 1.0 * COUNT(DISTINCT CASE WHEN EXTRACT(DOW FROM departure_time) IN (5,6)
              AND return_flight_booked IS TRUE
              AND EXTRACT(DOW FROM return_time) IN (0,1)
              AND EXTRACT(DAY FROM (return_time - departure_time)) < 3
              THEN f.trip_id
              ELSE NULL END) / COUNT(DISTINCT CASE WHEN NOT cancellation THEN s.trip_id END) ELSE 0 END
             ,2) AS weekend_trip_proportion,

-- Round trips proportion, users who booked two ways flights
            ROUND(
             CASE WHEN COUNT(DISTINCT CASE WHEN flight_booked THEN s.trip_id END) > 0
THEN
             1.0 * COUNT(DISTINCT CASE WHEN return_flight_booked THEN s.trip_id END) /
             COUNT(DISTINCT CASE WHEN flight_booked THEN s.trip_id END) ELSE 0 END
             ,2) AS round_trips_proportion,
            -- average flight price
            ROUND(AVG(base_fare_usd),2) AS avg_flight_price_usd,
            -- average flight discount amount
        ROUND(AVG(flight_discount_amount),2) AS avg_flight_discount_amount,
            -- discounted flights proportion
            ROUND(SUM(CASE WHEN flight_discount THEN 1 ELSE 0 END) :: NUMERIC /
COUNT(*),2) AS discounted_flight_proportion,
            -- average number of booked flight seats
            ROUND(AVG(seats),2) AS avg_flight_seats,
            -- average number of checked bags in flights
            ROUND(AVG(checked_bags),2) AS avg_checked_bags,
        -- Vincenty formula for average distance between airports (average distance flown in km)
        ROUND(
            AVG(
               v.a * ATAN2(
                  SQRT((v.cos2Alpha)*(v.sinSigma*v.sinSigma)),
                  v.cosSigma - v.f*v.cos2Alpha*(v.cos2SigmaM)
               )/1000
            )::NUMERIC, 2
        ) AS avg_distance_flown_km,
            -- total number of booked hotels
            COUNT(DISTINCT CASE WHEN hotel_booked THEN s.trip_id END) AS
total_hotels_booked,
            -- average hotel price
            ROUND(AVG(hotel_per_room_usd),2) AS avg_hotel_price_usd,
            -- average hotel discount amount
            ROUND(AVG(hotel_discount_amount),2) AS avg_hotel_discount_amount,
            -- discounted hotel proportion
            ROUND(SUM(CASE WHEN hotel_discount THEN 1 ELSE 0 END) :: NUMERIC /
COUNT(*),2) AS discounted_hotel_proportion,

-- average number of rooms in booked hotels
            ROUND(AVG(rooms),2) AS avg_hotel_rooms,
            -- average duration of hotel stays in days
            ROUND(AVG(EXTRACT(DAY FROM (check_out_time - check_in_time))),2) AS
avg_stay_duration_day

   FROM sessions AS s
   LEFT JOIN flights AS f ON s.trip_id = f.trip_id
   LEFT JOIN hotels AS h ON s.trip_id = h.trip_id
   LEFT JOIN users AS u ON s.user_id = u.user_id
   LEFT JOIN VincentyComputations AS v ON s.trip_id = v.trip_id
   WHERE s.user_id IN (SELECT user_id FROM cohort_users)
   GROUP BY s.user_id
)
```

Final Selection

```
SELECT
    -- user demographic information
    u.user_id,
    u.sign_up_date,
    EXTRACT(YEAR FROM AGE(u.birthdate)) AS age,
    u.gender,
    u.married,
    u.has_children,
    u.home_country,
    u.home_city,
    -- browsing sessions info
    ad.session_count,
    ad.avg_session_duration_minute,
    ad.avg_page_clicks,
    -- booking behaviour
    ad.total_trips,
    ad.conversion_rate,
    ad.weekend_trip_proportion,
    ad.cancellation_proportion,
    ad.avg_booking_departure_gap_days_flights,
    ad.avg_booking_departure_gap_days_hotels,
    -- booked flights info
    ad.total_flights_booked,
    ad.round_trips_proportion,
    ad.avg_flight_price_usd,
    ad.avg_flight_discount_amount,
    ad.discounted_flight_proportion,
    ad.avg_flight_seats,
    ad.avg_checked_bags,
    ad.avg_distance_flown_km,
    -- booked hotels info
    ad.total_hotels_booked,
    ad.avg_hotel_price_usd,
    ad.avg_hotel_discount_amount,
    ad.discounted_hotel_proportion,
    ad.avg_hotel_rooms,
    ad.avg_stay_duration_day
FROM users AS u
JOIN aggregated_data AS ad ON u.user_id = ad.user_id;
```

File for References:

1.) Spreadsheet_Link
2.) Jupyter_Link