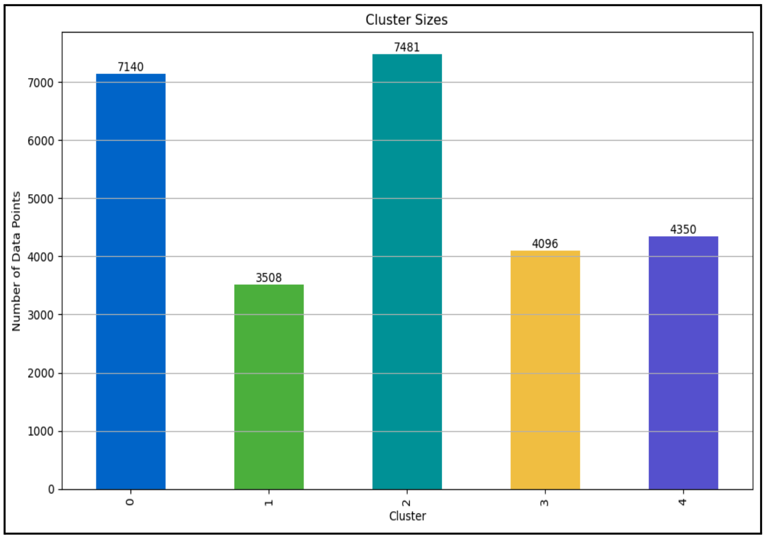
Project Report: Sun Country Case Study

To: Professor Mingdi Xin
~ Team 12A: Zhiwei Lu, Yuh-Shin Yen, Viraj Vijaywargiya, Anna Haroutounian, Danqi Zheng

For our team case assignment, we've analyzed the data provided by Sun Country Airlines and created five customer segments to gain a deeper understanding of which business objectives they should prioritize to become more profitable against their competitors. For our analysis, we created three categories that we will focus on: Customer Demographics, Customer Travel Patterns, and Customer Travel Status. Based on these insights, we've summarized our actionable marketing steps that Sun Country Airlines can take to improve their customer segmentation, marketing efforts and product development to better compete with larger national brands.

After receiving our clustering data, we ensured there were no missing values and created a copy of the dataset to perform the clustering and preserve the original data. From there, we found that **five** is our optimum number of clusters by using the K-Means clustering method and ran 10 iterations of an Elbow Curve. From this we applied the K-Means and then extracted the cluster "Assignments" to combine the clustering data and the customer data. We will be using the combined data for our segmentation analysis (please refer to the appendix for our Python code that demonstrates the above).

Distribution of Segments



Before we dive into our specific categories, we wanted to analyze the distribution of the segments. The bar chart on the left shows the **sizes of the five customer segments (clusters)** in terms of the number of data points. **Segments 2** and **0** represent the largest customer groups, suggesting that they represent common customer types. This could indicate that the airline serves a large group of customers with similar preferences or behaviors.

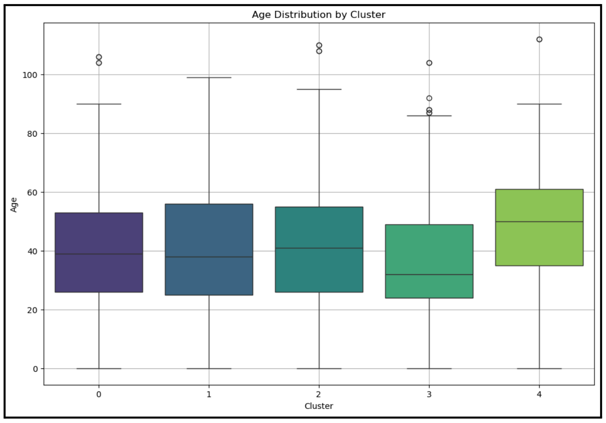
Clusters 3 and 4 are moderate in size and could suggest moderate diversity in customer behavior or preferences. These might represent a mix of common behaviors, with neither being as dominant as Clusters 2 and 0.

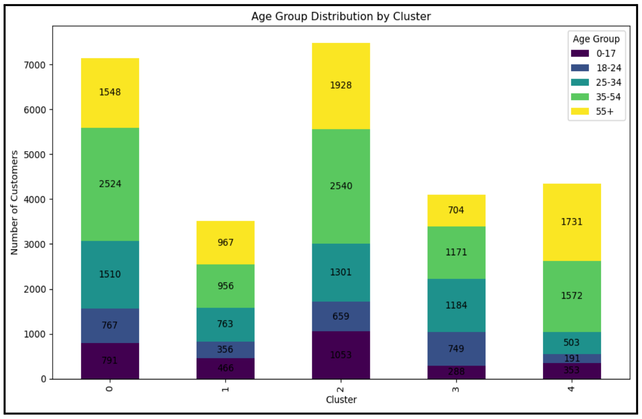
Cluster 1, the smallest, may represent a niche or less common segment of customers. This might indicate a group with more specific or unique preferences, potentially requiring tailored services.

Further, we will be analyzing the behavior and characteristics of each Customer Segment.

Customer Demographics

Age Distribution





For our first area of analysis in Demographics, we looked at the Age Distribution by using the **Age_Group** data. We created a box plot visualization to understand the age distribution by segment (cluster). The x-axis represents segments from 0 - 4. The y-axis represents the age of individuals within each segment, ranging from 0 - 100 (representing customers from the **55+ bracket**).

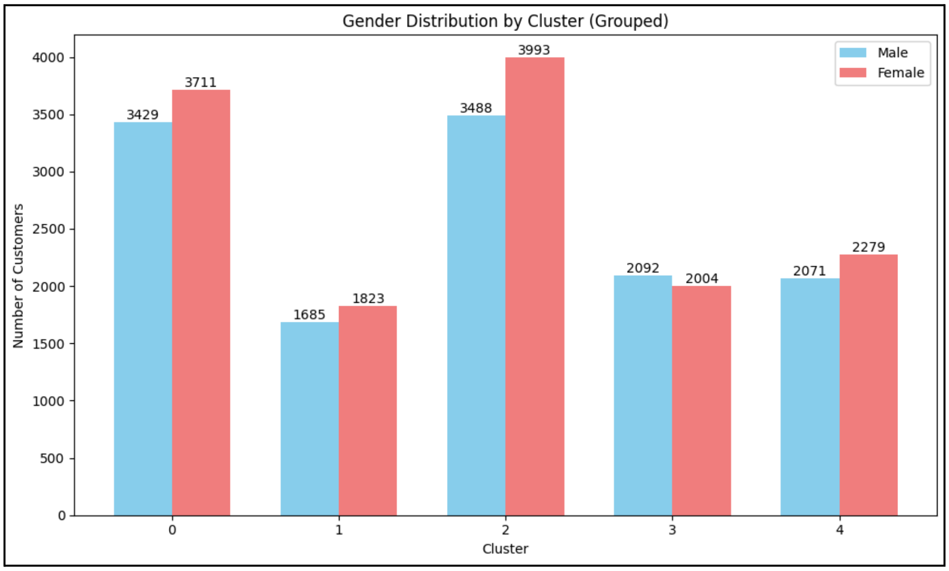
From this visualization we can see that segment 0 and segment 1 share a similar **median age of around 42**. Segment 2 has a slightly compressed age distribution with a **median age closer to 43**. It has **fewer outliers** of all the segments and less variability compared to segment 0 and 1. This suggests that the age group for this segment is more uniform with lesser extreme values.

Segment 3 tells us that the **median age is 35, the lowest** of all the segments. The **IQR is also narrower** indicating that there is less variation in ages with this segment, however it also has o**utliers above 80** which might indicate that there are older members in this group. Segment 4 has the **highest median age of around 50** with a moderate IQR range of ages. There are also **outliers present** however the overall distribution is less skewed than segment 0 and 1 which might indicate that this segment is an **older age group** with a fewer amount of extreme values.

When we take a look at the bar chart, we can confirm our findings by seeing that a small proportion of segment 3 is 55+, whereas a large proportion of Segment 4 is 35+.

In summary, we might want to focus on segment 3 if we are interested in targeting a **younger population**, and alternatively we can look at segment 4 which exhibits a more concentrated distribution of an **older population**, while segments 0 and 1 can help us target a middle-aged population.

Gender Distribution



Continuing the analysis of demographics, we analyzed the gender distribution with a bar chart by using the **GenderCode** data. The x-axis represents the clusters (segments) 0 - 4, and the y-axis indicates number of customers (0 to 4000), with male customers shown in blue and female customers in coral.

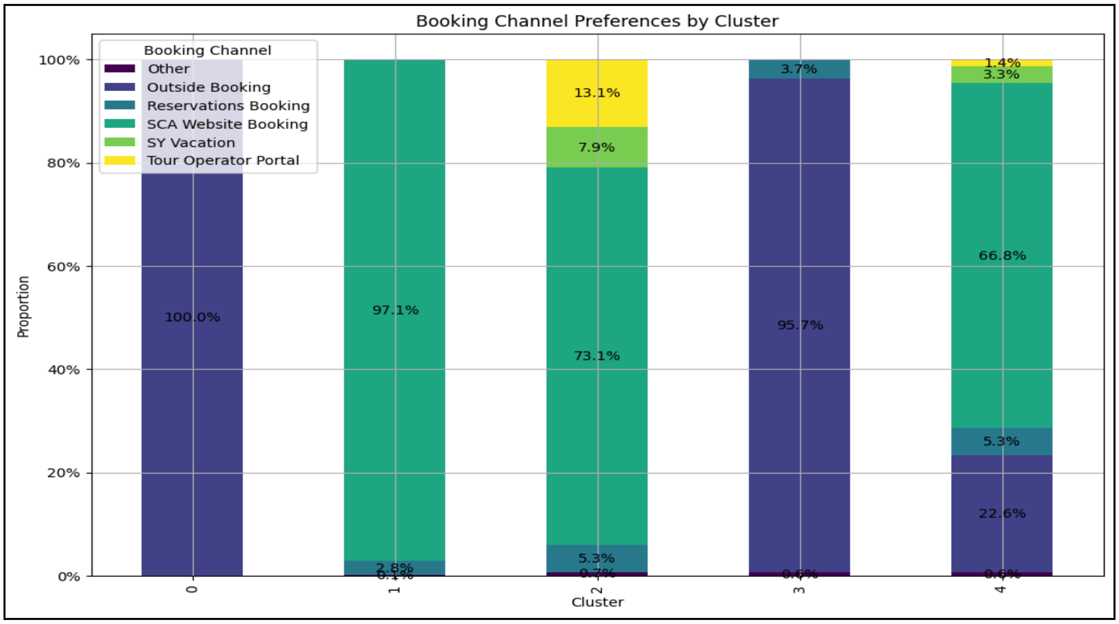
In segment 0, female customers (3711) **slightly outnumber male** customers (3429). For segment 2, there is a **significant female majority,** with 3993 female customers compared to 3488 males. The gender distribution in segment 3 is nearly balanced (2092 males, 2004 females) and in segment 4, Females (2279) slightly outnumber males (2071).

With this context, **segment 0 and 2** have the **highest volume of female customer**s and they both outnumber males. For these segments, Sun Country can focus on marketing strategies tailored to women, such as family vacation packages, female-focused travel groups, parent-child travel services, or wellness packages to attract more women.

Because segment 3 and 4 are **almost balanced** proportions, these segments could potentially be ideal for **gender-neutral promotions** and loyalty incentives aimed at both male and female customers. Marketing family or couple travel services could be effective in these segments.

Customer Travel Patterns

Booking Channel Preferences



From demographics we next decided to look at the travel patterns of the customers, starting by focusing on their booking channel preferences using the **BookingChannel** data. We created a stacked bar chart of customer booking channel preferences across the five segments (clusters). The x-axis represents the segment from 0 - 4, while the y-axis shows the percentage of customers using different choices such as Outside Booking, Reservations Booking, SCA Website, SY Vacation, and Tour Operator Portal.

First, **s**egment 0 stands out as all customers **(100%)** exclusively use **Outside Booking**. In contrast, **s**egment 1 shows a strong preference for the **SCA Website**, with **97.1%** of bookings made through this channel and only 2.8% using Reservations Booking. This highlights segment 1's reliance on the website for convenience and familiarity.

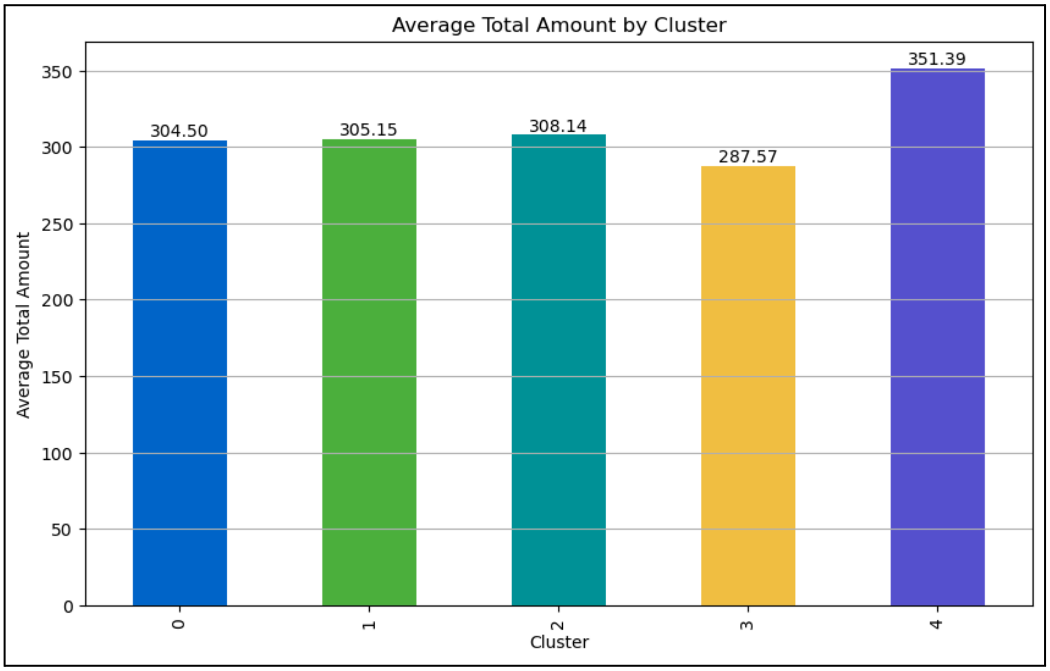
Moving on to segment 2, we see more varied booking behavior, with 73.1% using the SCA Website, 13.1% choosing the Tour Operator Portal, and 7.9% booking through SY Vacation.This suggests that customers in segment 2 are **more flexible and open** to exploring different booking methods. Similarly, **s**egment 4 mirrors segment 2's diversity, with **66.8%** bookings via the SCA

Website, **22.6%** using Outside Booking, 5.3% for Reservation booking and 3.3% for SY Vacation. Both segments 2 and 4 use the highest amount of Reservations booking, both at 5.3% which can suggest that at some times, these segments prefer to have someone else help them with their reservations.

Meanwhile, segment 3 behaves more like segment 0, with **95.7%** relying on Outside Booking, though it shows slight openness to other options.

In summary, Segment 0 and 3 **highly prefer Outside Bookin**g. In contrast, segment 1 remains **dependent on the SCA Website**, which indicates that they are not open to other choices. Alternatively, segments 2 and 4 display **very diverse booking habits**, which could demonstrate a **greater openness** to vacation packages or bundled deals or alternative choices like from third-party services.

Average Total Amount Spent

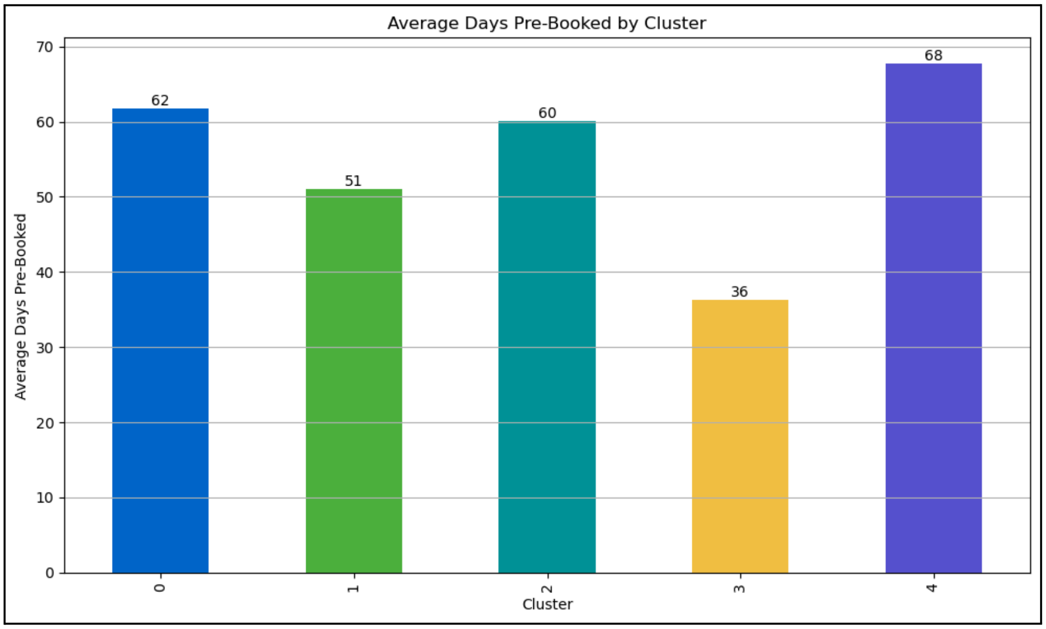


Continuing on our analysis of travel patterns, we took a look at the average total amount spent by looking at the **BaseFareAmount** data. For this visualization, we chose a bar chart that shows the average total amount spent by customers, categorized into five clusters (segments). The x-axis represents the segment 0 - 4. The y-axis represents the average total amount spent by the customers in each segment. The amounts range between 250 and 350 in dollars.

Segment 4 stands out as having the highest average total amount spent, exceeding **350**. This segment might represent **high-value customers** or frequent flyers who spend significantly more on their tickets or services. Segment 3 has the lowest average total amount, around **285**, which suggests that this group may consist of more **budget-conscious** customers or infrequent travelers. The remaining segments (0, 1, and 2) show a relatively similar average total amount around **300**, indicating that these customer segments are more **balanced** in their spending behavior which could mean that they are regular flyers with moderate spending patterns.

In summary, because segment 4 is the **highest spender** of the segments, Sun Country could focus on creating **retention strategies** to ensure that they continue their **customer loyalty**. With segments 0, 1, and 2 having similar average amounts, they appear to be more **stable** as customer groups, which is **also an important population to retain**. While all segments are important to retain for Sun Country, because of the differentiation of each of these segments, there can be targeted retention strategies created to ensure the loyalty of each group while at the same time, encouraging more spending. While segment 3 has the lowest average total, it could also indicate that they are a group that prioritizes **low-cost options** or they might not be as loyal to one airline like Sun Country. Marketing efforts can be made to target this sentiment by providing flash deals and incentives.

Average Days Pre-Booked



Our last analysis of travel patterns is by looking at the average days pre-booked, using the **Days_Pre_Booked** data. This bar chart shows the Average Days Pre-Booked by customers across five different segments (clusters), 0 - 4. Each bar represents the average number of days that customers from a specific segment booked their flights in advance. The y-axis represents the average days pre-booked, ranging from 0 to 70 days. The x-axis shows the five clusters, 0 - 4. The exact number of days pre-booked for each cluster is labeled on top of the corresponding bar for easier readability.

First, we can see that segments 0, 2, and 4 have very high averages for pre-booking days (62, 60, and 68, respectively). From this we can assume that these segments tend to be **highly organized travelers** who plan ahead. They could also be more **price-sensitive**, booking in advance to secure better fares, or they may belong to **frequent travelers or families** who require early planning.

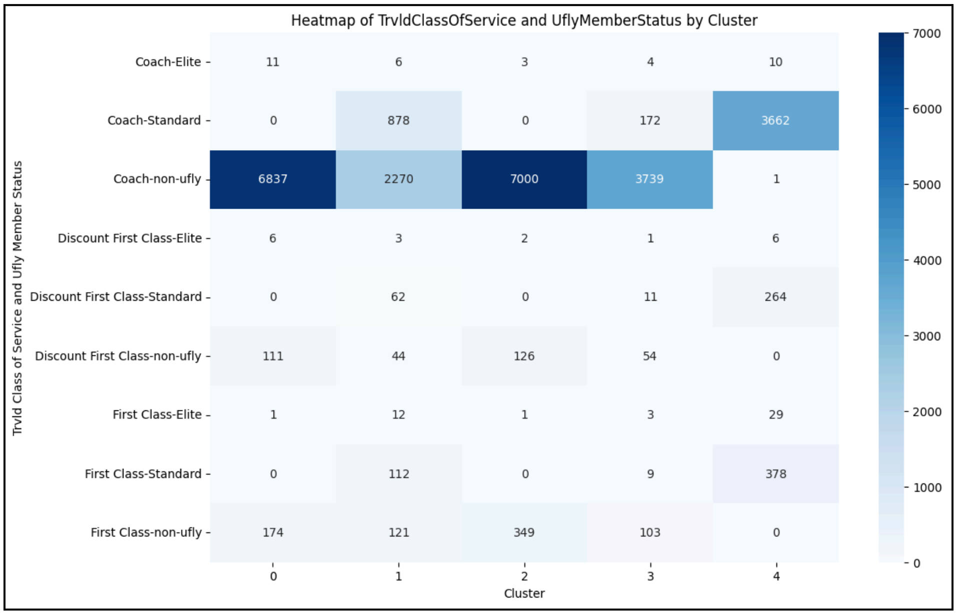
Segment 1, at 51 days, also prefers planning in advance, but not as early as Segments 0, 2, and 4. They may be a blend of **casual leisure travelers** who plan early but may not be as rigid in their booking patterns.

With the lowest pre-booking average of 36 days, Segment 3 stands out as a **spontaneous or last-minute** traveler group. They may be more **flexible** with travel dates, opting for deals closer to the departure date, or **business travelers** who often book based on unpredictable schedules.

Therefore, the plot highlights clear behavioral differences among the five customer segments. Understanding these characteristics can help Sun Country Airlines tailor marketing strategies, such as **early-bird promotions** for segments 0, 2, and 4, and **last-minute deals** for segment 3.

Customer Travel Status

Travel Class and Member Status



For our analysis of Customer Travel Status, we decided to focus mainly on the segments Travel Class and Member Status. We chose a heat map to visualize the relationship between traveled class of service (**TrvldClassOfService**), Ufly member status (**UflyMemberStatus**), and **cluster** (segment). The rows represent combinations of the traveled class of service (Coach, Discount First Class, First Class) and loyalty status (Elite, Standard, Non-Ufly). The columns represent the five different customer segments (clusters), 0 - 4. Darker shades represent higher numbers of customers in that particular combination of traveled class, loyalty status, and cluster, while lighter shades represent fewer customers.

In Segments 0 and 2, the Majority traveled in **Coach (Non-Ufly)** class, meaning most customers in these groups are not Ufly members and opt for standard coach class. Therefore, this segment appears primarily to be non-loyalty program members, **budget-conscious** travelers who typically opt for economy class without any elite or standard status. They could represent occasional or **infrequent travelers**.

Segment 3 is also largely **coach** travelers, but with a few customers upgrading to premium services like Coach Standard or First Class(non-loyalty), likely representing a mix of **budget and occasional business** travelers.

For segment 1, **Coach (Non-Ufly)** remains dominant, but there's a significant group of Coach-Standard travelers, showing that a notable portion of this segment may have **standard Ufly loyalty status**. Therefore, it consists of a blend of non-loyalty and standard loyalty travelers who prefer coach class but may use their loyalty benefits for occasional upgrades.

Finally, segment 4 stands out due to its **heavy representation in Coach-Standard**, which can suggest a strong presence of loyalty program members in the standard coach class. It is composed of **loyal and value-driven customers** who regularly travel coach class but also use loyalty benefits for occasional upgrades to first class. They likely represent frequent leisure or business travelers who use loyalty status to enhance their travel experience.

Segment Insights and Recommendations

- Starting with **Segment 0**, we can define this segment as the "***Vacationers***." They are a diverse group of travelers that mainly fall between the 35 - 53 age group with a slightly higher number of women than men. They exclusively prefer outside bookings to arrange their travel and spend on average about $300. These customers also book their trip an average of 62 days in advance and mostly opt in for Coach class, especially as they are primarily non-loyalty program members and more budget-conscious.

<u>**Recommendation**</u>: With this information, **Early-bird Promotions** could be effective with discounts or loyalty points with bookings made 60+ days in advance, combined with launching marketing campaigns at least six months in advance. Additionally, strategic **Holiday-Specific Marketing** by incentivizing them with bundled packages of accommodation, transportation, and guided tours will appeal to travelers' preference for well-planned trips.

❖ For **Segment 1**, we call this the "***Convenience***" segment as they consist mostly of pragmatic travelers who prefer the convenience of booking through the SCA website. With a balanced age and gender distribution, they also tend to book their trips ahead of time at about 51 days in advance. They prefer Coach class even though they are also a blend of non-loyalty and loyalty members.

Recommendation: With this information, we recommend focusing on **Early-bird Promotions** for off-peak travel as well as **Flexible Booking Policies**, such as refundable tickets or easy date changes to cater to their preference of convenience and year round travel plans.

❖ **Segment 2** is the largest of all the segments with mostly members aged 35 and above and women slightly outnumbering men. On average they book 30 days in advance and mostly use the SCA website, however they also use the other booking options more than the other customer segments. While they are primarily non-loyalty members and fly Coach, they also have a notable preference to book first class as well. With this information, we consider this segment the "***Family Flyers***" mainly due to the size of this group and the equal amounts split for gender.

Recommendation: As a result, we recommend strategizing **Family Travel Packages** with deals on travel plans and advertise on various platforms. Additionally, a targeted **Family Loyalty Program** could motivate this segment in becoming a member for exclusive benefits, discounts and perks.

❖ **Segment 3** is a group that mainly consists of middle-aged customers between 35 and 54, and have an equal amount in gender distribution. This group heavily relies on outside booking sites and have the lowest average amount of total spent of all segments at $275, assuming due to limited reimbursement. They also appear to book on average 36 days in advance and prefer to fly coach. With this blend of budget and emergent business travel, we would categorize this segment as the "***White Collar***".

Recommendation: We would recommend **Last-Minute Booking** strategies for these customers with same-day deals and flash sales. To cater to the business travelers, by providing corporate discounts, flexible changes and seat upgrades, it can help in targeting these customers in creating business convenience. Additionally, strategizing on **Corporate Loyalty Programs** like seat upgrades, lounge access and free Wi-Fi will incentivize the segment and increase loyal members.

- Finally in **Segment 4**, we found this group to be the oldest of our customers with travelers aged 35 and up, and a balanced distribution in gender. This group has the highest average of pre-booking at 68 days as well as the highest in spending at $350. They prefer to book from the SCA website and outside channels. We found that most customers in this segment are loyalty program members with a high proportion of customers flying first class. We categorize this segment as the "***Deluxe Comfort***."

  Recommendation: Since segment 4 is the highest-value segment by far, it is important to focus on retention strategies for this group. We recommend strategizing **Luxury Travel Promotions** like premium service and seating by emphasizing comfort, exclusivity, and luxury in promotions. Additionally, we promote **Personalized Travel Offers** to develop personalized offers based on past travel behavior, focusing on high-value services.

Conclusion

In conclusion, our analysis has revealed five distinct customer segments, each with unique behaviors and preferences. We believe Sun Country can benefit from our recommendations by developing targeted strategies to enhance customer engagement and drive loyalty. With these strategies, they can become more profitable and gain a competitive edge. As the world becomes more modernized and data-driven, implementing these new marketing plans will help Sun Country Airlines better cater to its diverse customer base, leading to sustained growth and long-term success.

Appendix_Team_12A_Project_Code

September 8, 2024

1 APPENDIX: TEAM 12A PROJECT CODE

1.1 Importing libraries and Connecting to Google Drive

```
[ ]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.cluster import KMeans

     # Connect to google drive
     from google.colab import drive
     drive.mount('/content/drive')
     # Increase the default max number of columns displayed
     from google.colab.data_table import DataTable
     DataTable.max_columns = 100
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

1.2 Loading the Clustering Dataset

```
[ ]: data = pd.read_csv('/content/drive/MyDrive/MSBA Team 12A/BANA 200A (Foundations␣
       ↪of Business Analytics )/Clustering Data.csv')

     data.head()
```

```
[ ]:                                                      uid PNRLocatorID   avg_amt  \
     0  5045544552446964204 93F7C2067657420746869732072…       AADMLF  0.019524
     1  46495853454E44696420493F7C20676574207468697320…       AAFBOM  0.081774
     2  53435554544469642049 3F7C2067657420746869732072…       AAFILI  0.026650
     3  53435554544469642049 3F7C2067657420746869732072…       AAFILI  0.026650
     4  44554D4D414E4E44696420493F7C20676574207468 6973…       AAFRQI  0.000000

        round_trip  group_size  group  days_pre_booked  BookingChannel_Other  \
     0           0       0.000      0         0.029703                     0
     1           1       0.000      0         0.039604                     0
     2           0       0.125      1         0.069307                     0
```

```
3           0       0.125      1         0.069307                     0
4           1       0.000      0         0.035361                     0

   BookingChannel_Outside_Booking  BookingChannel_Reservations_Booking  ...  \
0                               1                                    0  ...
1                               0                                    0  ...
2                               0                                    0  ...
3                               0                                    0  ...
4                               1                                    0  ...

   true_destination_dest_SXM  true_destination_dest_TPA  \
0                          0                          0
1                          0                          0
2                          0                          0
3                          0                          0
4                          0                          0

   true_destination_dest_ZIH  UflyMemberStatus_Elite  \
0                          0                       0
1                          0                       0
2                          0                       0
3                          0                       0
4                          0                       0

   UflyMemberStatus_non-ufly  UflyMemberStatus_Standard  seasonality_Q1  \
0                          0                          1               0
1                          0                          1               0
2                          1                          0               1
3                          1                          0               1
4                          1                          0               0

   seasonality_Q2  seasonality_Q3  seasonality_Q4
0               0               0               1
1               0               1               0
2               0               0               0
3               0               0               0
4               0               0               1

[5 rows x 90 columns]
```

1.3 Data Pre-Processing for Clustering

[ ]:

```
# checking for missing values
missing_values = data.isnull().sum()
missing_values
```

```
[ ]: uid                          0
     PNRLocatorID                 0
     avg_amt                      0
     round_trip                   0
     group_size                   0
                                 ..
     UflyMemberStatus_Standard    0
     seasonality_Q1               0
     seasonality_Q2               0
     seasonality_Q3               0
     seasonality_Q4               0
     Length: 90, dtype: int64
```

```
[ ]: # Preserving the original data
     clustering_data = data.copy()

     # preparing for clustering
     clustering_data.drop(columns = ['PNRLocatorID', 'uid'], inplace=True)
     clustering_data.head()
```

```
[ ]:     avg_amt  round_trip  group_size  group  days_pre_booked  \
     0  0.019524           0       0.000      0         0.029703
     1  0.081774           1       0.000      0         0.039604
     2  0.026650           0       0.125      1         0.069307
     3  0.026650           0       0.125      1         0.069307
     4  0.000000           1       0.000      0         0.035361

        BookingChannel_Other  BookingChannel_Outside_Booking  \
     0                     0                               1
     1                     0                               0
     2                     0                               0
     3                     0                               0
     4                     0                               1

        BookingChannel_Reservations_Booking  BookingChannel_SCA_Website_Booking  \
     0                                    0                                   0
     1                                    0                                   1
     2                                    0                                   1
     3                                    0                                   1
     4                                    0                                   0

        BookingChannel_SY_Vacation  …  true_destination_dest_SXM  \
     0                           0  …                          0
     1                           0  …                          0
     2                           0  …                          0
     3                           0  …                          0
     4                           0  …                          0
```

```
   true_destination_dest_TPA  true_destination_dest_ZIH  \
0                          0                          0
1                          0                          0
2                          0                          0
3                          0                          0
4                          0                          0

   UflyMemberStatus_Elite  UflyMemberStatus_non-ufly  \
0                       0                          0
1                       0                          0
2                       0                          1
3                       0                          1
4                       0                          1

   UflyMemberStatus_Standard  seasonality_Q1  seasonality_Q2  seasonality_Q3  \
0                          1               0               0               0
1                          1               0               0               1
2                          0               1               0               0
3                          0               1               0               0
4                          0               0               0               0

   seasonality_Q4
0               1
1               0
2               0
3               0
4               1

[5 rows x 88 columns]
```

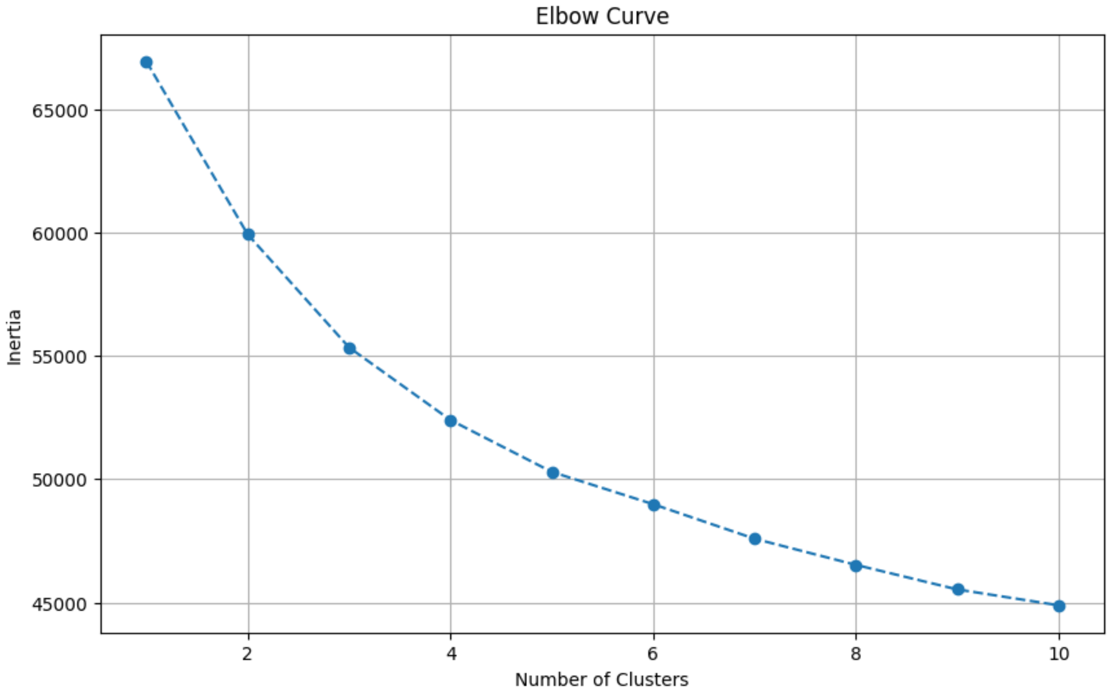
1.4 Clustering

[ ]:

```
# Finding the optimal K (Given to us as 5)

inertia = []                  # sum of squared distances between each data␣
 ↪point and its closest cluster centroid (wcss)
cluster_range = range(1,11)


for cluster_num in cluster_range:

    print(f'Iteration Number: {cluster_num}')

    kmeans = KMeans(n_clusters=cluster_num, n_init=10)
    kmeans.fit(clustering_data)
    inertia.append(kmeans.inertia_)
```

```
plt.figure(figsize=(10,6))
plt.plot(cluster_range, inertia, marker='o', linestyle='--')
plt.title('Elbow Curve')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.grid(True)
plt.show()
```

```
Iteration Number:  1
Iteration Number:  2
Iteration Number:  3
Iteration Number:  4
Iteration Number:  5
Iteration Number:  6
Iteration Number:  7
Iteration Number:  8
Iteration Number:  9
Iteration Number:  10
```



[ ]:

```
# applying the kmeans
```

```
kmeans = KMeans(n_clusters=5, random_state=42)
kmeans.fit(clustering_data)

####### For analysis, do NOT assign the labels to the transformed data!
data['Cluster'] = kmeans.labels_

data.head(10000)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

```
[ ]:                                                    uid PNRLocatorID  \
      0     50455445524469642049 3F7C2067657420746869732072…       AADMLF
```

```
                                                    uid PNRLocatorID  \
0     504554455244696420493F7C20676574207468697320 72…       AADMLF
```

```
                                                    uid PNRLocatorID  \
0     504554455244696420493F7C2067657420746869732072…       AADMLF
1     46495853454E44696420493F7C20676574207468697320…       AAFBOM
2     534355545444696420493F7C2067657420746869732072…       AAFILI
3     534355545444696420493F7C2067657420746869732072…       AAFILI
4     44554D4D414E4E44696420493F7C206765742074686973…       AAFRQI
…                                                   …          …
9995  465241484D44696420493F7C2067657420746869732072…       OVDEVE
9996  5649434B45525944696420493F7C206765742074686973…       OVDVCT
9997  535445504B414469642049 3F7C20676574207468697320…       OVGARJ
9998  4B4F53544B4F44696420493F7C20676574207468697320…       OVGBHE
9999  59414244696420493F7C2067657420746869732072696 7…       OVGBHE

       avg_amt  round_trip  group_size  group  days_pre_booked  \
0     0.019524           0       0.000      0         0.029703
1     0.081774           1       0.000      0         0.039604
2     0.026650           0       0.125      1         0.069307
3     0.026650           0       0.125      1         0.069307
4     0.000000           1       0.000      0         0.035361
…          …         …          …      …               …
9995  0.042032           0       0.125      1         0.036775
9996  0.049705           0       0.000      0         0.028289
9997  0.012305           1       0.000      0         0.019802
9998  0.067609           1       0.125      1         0.125884
9999  0.067609           1       0.125      1         0.125884

      BookingChannel_Other  BookingChannel_Outside_Booking  \
0                        0                               1
1                        0                               0
2                        0                               0
3                        0                               0
4                        0                               1
…                      …                              …
9995                     0                               0
```

```
9996                     0                               1
9997                     0                               1
9998                     0                               1
9999                     0                               1

      BookingChannel_Reservations_Booking  …  true_destination_dest_TPA  \
0                                       0  …                          0
1                                       0  …                          0
2                                       0  …                          0
3                                       0  …                          0
4                                       0  …                          0
…                                    …  …                        …
9995                                    0  …                          0
9996                                    0  …                          0
9997                                    0  …                          0
9998                                    0  …                          0
9999                                    0  …                          0

      true_destination_dest_ZIH  UflyMemberStatus_Elite  \
0                             0                       0
1                             0                       0
2                             0                       0
3                             0                       0
4                             0                       0
…                          …                     …
9995                          0                       0
9996                          0                       0
9997                          0                       0
9998                          0                       0
9999                          0                       0

      UflyMemberStatus_non-ufly  UflyMemberStatus_Standard  seasonality_Q1  \
0                             0                          1               0
1                             0                          1               0
2                             1                          0               1
3                             1                          0               1
4                             1                          0               0
…                          …                        …             …
9995                          0                          1               0
9996                          1                          0               0
9997                          1                          0               0
9998                          1                          0               0
9999                          1                          0               0

      seasonality_Q2  seasonality_Q3  seasonality_Q4  Cluster
0                  0               0               1        4
1                  0               1               0        4
```

```
2                  0               0               0        1
3                  0               0               0        1
4                  0               0               1        3
…              …             …             …      …
9995               0               1               0        4
9996               0               1               0        3
9997               0               1               0        0
9998               0               1               0        0
9999               0               1               0        0

[10000 rows x 91 columns]
```

```
[ ]: # cluster counts
     data['Cluster'].value_counts().sort_index()
```

```
[ ]: Cluster
     0    3843
     1    2353
     2    4127
     3    2436
     4    2385
     Name: count, dtype: int64
```

1.5 Merging the Cluster and Customer Data

```
[ ]: # loading the customer data
     customer_data = pd.read_csv('/content/drive/MyDrive/MSBA Team 12A/BANA 200A␣
      ↪(Foundations of Business Analytics )/sample_data_transformed.csv')
     customer_data.head()
```

```
<ipython-input-9-e1f0414e29a9>:2: DtypeWarning: Columns (13) have mixed types.
Specify dtype option on import or set low_memory=False.
  customer_data = pd.read_csv('/content/drive/MyDrive/MSBA Team 12A/BANA 200A
(Foundations of Business Analytics )/sample_data_transformed.csv')
```

```
[ ]:    Unnamed: 0 PNRLocatorID PaxName     TicketNum  CouponSeqNbr  \
     0           1       AADMLF  PETEJO  3.377490e+12             1
     1           2       AAFBOM  FIXSMO  3.372110e+12             2
     2           3       AAFBOM  FIXSMO  3.372110e+12             1
     3           4       AAFILI  SCUTKA  3.372110e+12             2
     4           5       AAFILI  SCUTKA  3.372110e+12             1

       ServiceStartCity ServiceEndCity PNRCreateDate ServiceStartDate  \
     0              MSP            DFW       9/15/14          10/6/14
     1              JFK            MSP       7/22/14          8/19/14
     2              MSP            JFK       7/22/14          8/14/14
     3              MSP            SEA        2/6/14          3/27/14
     4              LAN            MSP        2/6/14          3/27/14
```

```
                                       EncryptedName  …  \
0  504554455244696420493F7C2067657420746869732072…  …
1  46495853454E44696420493F7C20676574207468697320…  …
2  46495853454E44696420493F7C20676574207468697320…  …
3  534355545444696420493F7C2067657420746869732072…  …
4  534355545444696420493F7C2067657420746869732072…  …

                                                 uid  age_group  true_origins  \
0  504554455244696420493F7C2067657420746869732072…        55+           MSP
1  46495853454E44696420493F7C20676574207468697320…      35-54           MSP
2  46495853454E44696420493F7C20676574207468697320…      35-54           MSP
3  534355545444696420493F7C2067657420746869732072…      25-34           LAN
4  534355545444696420493F7C2067657420746869732072…      25-34           LAN

  final_destination round_trip group_size group  seasonality  days_pre_booked  \
0               DFW          0          1     0           Q4               21
1               MSP          1          1     0           Q3               28
2               MSP          1          1     0           Q3               23
3               SEA          0          2     1           Q1               49
4               SEA          0          2     1           Q1               49

  true_destination
0              DFW
1              JFK
2              JFK
3              MSP
4              MSP

[5 rows x 37 columns]
```

[ ]:
```
# cleaning the customer data
customer_data.drop(columns = ['Unnamed: 0'], inplace=True)

# merging the two dataframes
final_df = customer_data.merge(data[['uid', 'Cluster']], on='uid', how='left')
final_df.head()
```

[ ]:
```
  PNRLocatorID PaxName     TicketNum  CouponSeqNbr ServiceStartCity  \
0       AADMLF  PETEJO  3.377490e+12             1              MSP
1       AAFBOM  FIXSMO  3.372110e+12             2              JFK
2       AAFBOM  FIXSMO  3.372110e+12             1              MSP
3       AAFILI  SCUTKA  3.372110e+12             2              MSP
4       AAFILI  SCUTKA  3.372110e+12             1              LAN

  ServiceEndCity PNRCreateDate ServiceStartDate  \
0            DFW       9/15/14          10/6/14
```

```
1            MSP       7/22/14          8/19/14
2            JFK       7/22/14          8/14/14
3            SEA        2/6/14          3/27/14
4            MSP        2/6/14          3/27/14

                                       EncryptedName GenderCode  …  \
0  504554455244696420493F7C2067657420746869732072…         M  …
1  46495853454E44696420493F7C20676574207468697320…         F  …
2  46495853454E44696420493F7C20676574207468697320…         F  …
3  534355545444696420493F7C2067657420746869732072…         F  …
4  534355545444696420493F7C2067657420746869732072…         F  …

   age_group  true_origins final_destination round_trip group_size group  \
0        55+           MSP               DFW          0          1     0
1      35-54           MSP               MSP          1          1     0
2      35-54           MSP               MSP          1          1     0
3      25-34           LAN               SEA          0          2     1
4      25-34           LAN               SEA          0          2     1

   seasonality  days_pre_booked  true_destination Cluster
0           Q4               21               DFW       4
1           Q3               28               JFK       4
2           Q3               23               JFK       4
3           Q1               49               MSP       1
4           Q1               49               MSP       1

[5 rows x 37 columns]
```

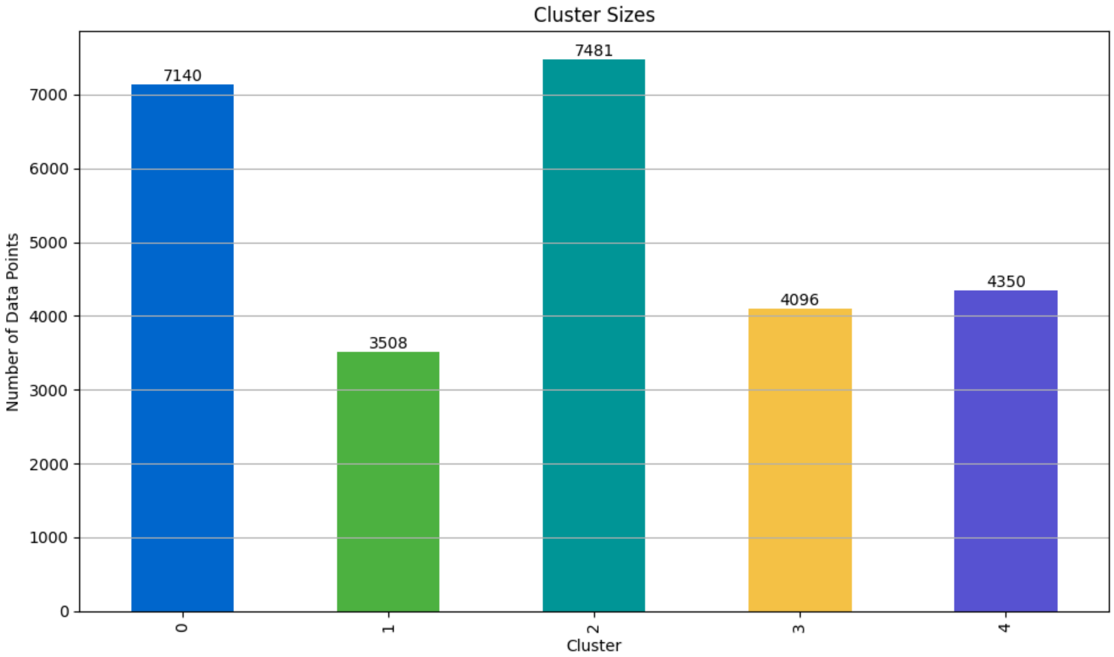
1.6 Data Visualizations for Segment Analysis

1.6.1 Cluster Distribution

[ ]:

```
colors = ['#06C', '#4CB140', '#009596', '#F4C145', '#5752D1']

cluster_sizes = final_df['Cluster'].value_counts().sort_index()

plt.figure(figsize=(10, 6))

# for cluster in range(len(cluster_sizes)):
#     plt.bar(cluster, cluster_sizes[cluster], color=colors[cluster])

ax = cluster_sizes.plot(kind='bar', color=colors)
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.
 ↪get_height()),
                ha='center', va='center', xytext=(0, 5), textcoords='offset␣
 ↪points')
```

```
plt.title('Cluster Sizes')
plt.xlabel('Cluster')
plt.ylabel('Number of Data Points')
plt.xticks(ticks=range(len(cluster_sizes)))
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

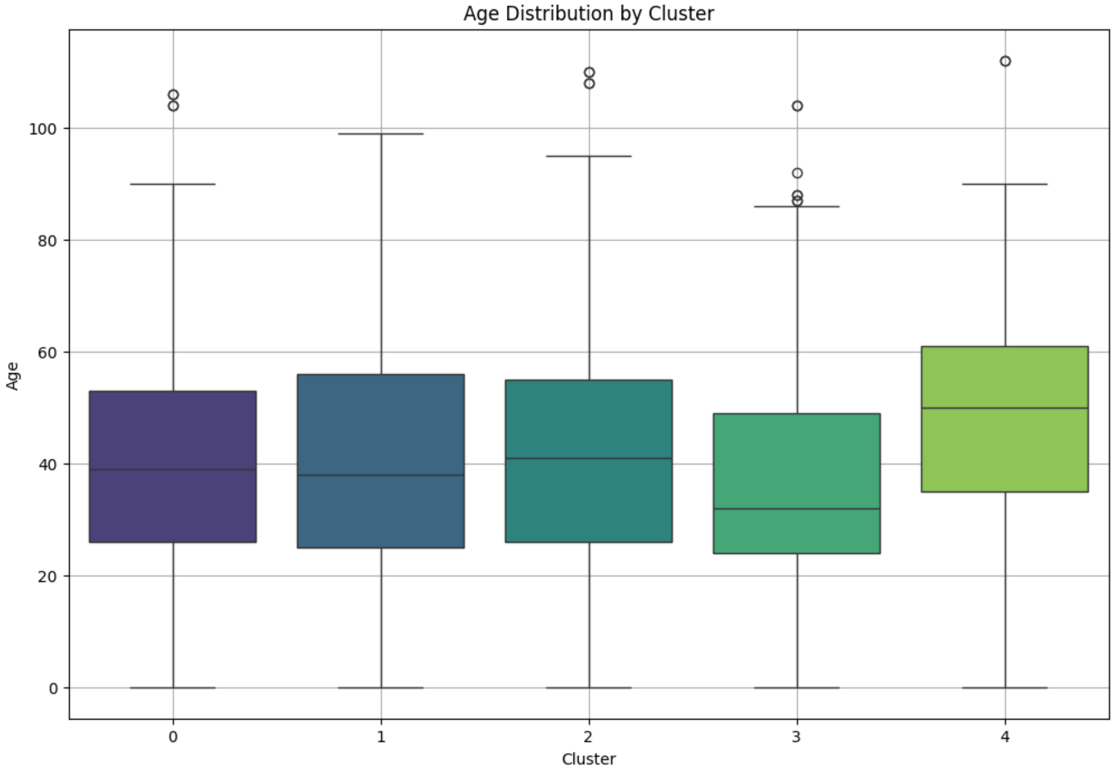


1.6.2 Customer Age by Segments

```
[ ]: # Age boxplot

plt.figure(figsize=(12, 8))
sns.boxplot(x='Cluster', y='Age', data=final_df, palette='viridis')
plt.title('Age Distribution by Cluster')
plt.xlabel('Cluster')
plt.ylabel('Age')
plt.grid(True)
plt.show()
```

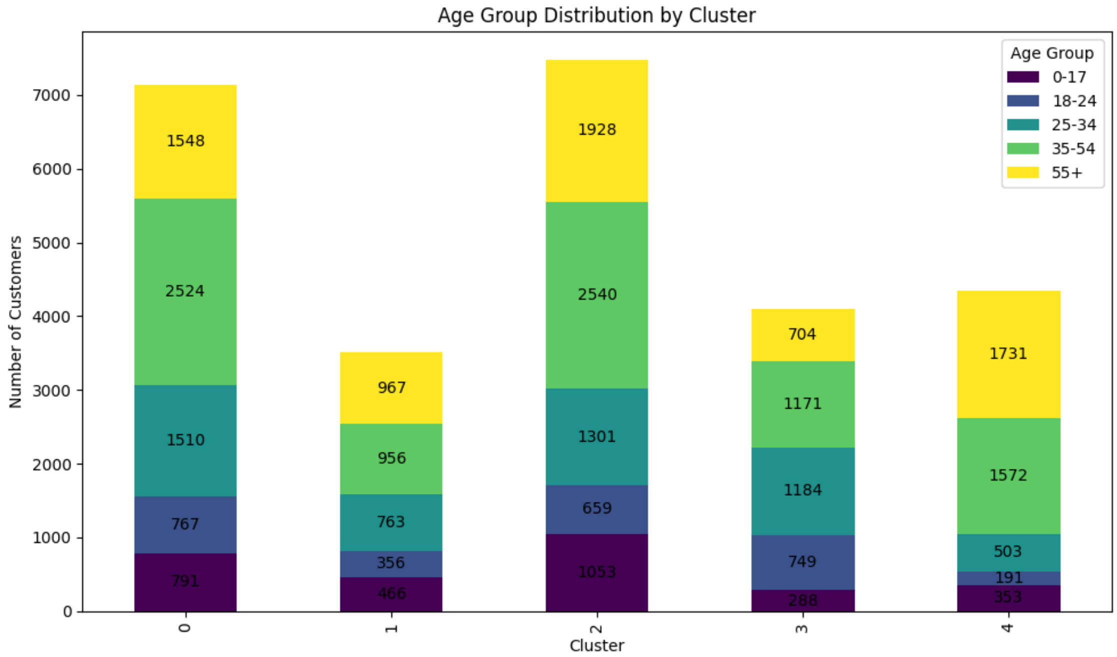
```
<ipython-input-12-a12d7373e189>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.
```

```
sns.boxplot(x='Cluster', y='Age', data=final_df, palette='viridis')
```



[ ]:

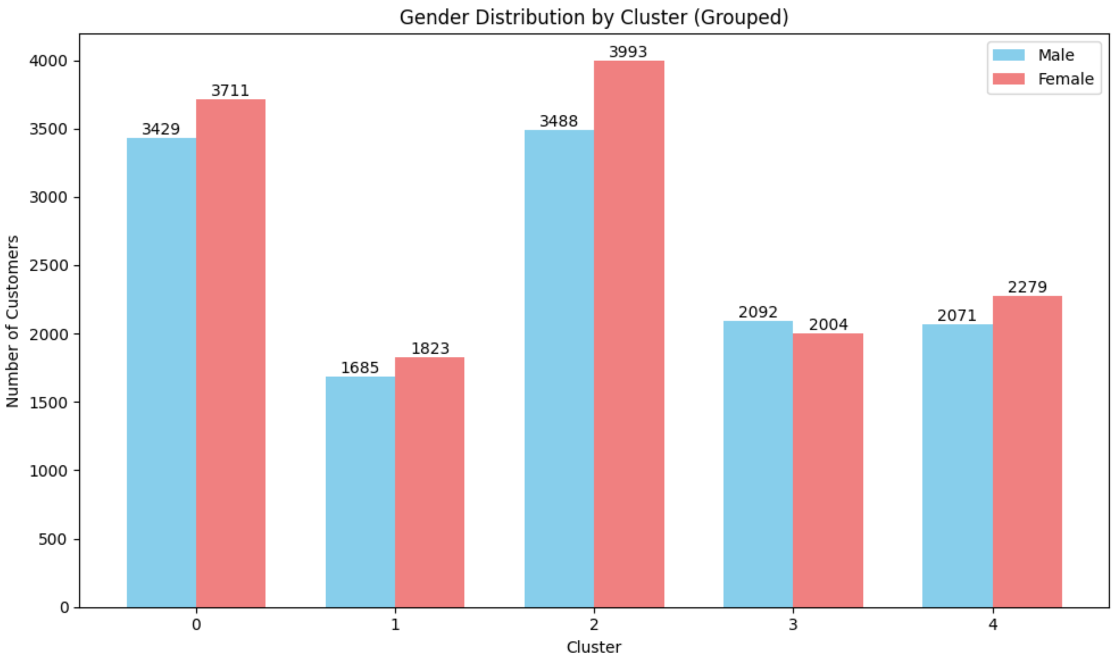
```
# Age Group Stacked bar chart

age_distribution = final_df.groupby(['Cluster', 'age_group']).size().unstack().
 ↪fillna(0)

# Plotting the stacked bar chart
ax = age_distribution.plot(kind='bar', stacked=True, figsize=(10, 6),␣
 ↪colormap='viridis')

# Adding labels to each segment of the stacked bar chart
for c in ax.containers:
    labels = [f'{int(v)}' if v > 0 else '' for v in c.datavalues]  # Only add␣
 ↪label if value > 0
    ax.bar_label(c, labels=labels, label_type='center')

# Plot
# age_distribution.plot(kind='bar', stacked=True, figsize=(10,6))
plt.title('Age Group Distribution by Cluster')
plt.xlabel('Cluster')
plt.ylabel('Number of Customers')
plt.legend(title='Age Group')
plt.tight_layout()
plt.show()
```



1.6.3 Customer Gender by Segment

```
[ ]: # Grouping data by clusters and gender
     gender_distribution = final_df.groupby(['Cluster', 'GenderCode']).size().
      ↪unstack().fillna(0)

     fig, ax = plt.subplots(figsize=(10, 6))

     # Setting bar width and positions for grouped bar chart
     bar_width = 0.35
     index = np.arange(len(gender_distribution.index))

     # bars for Male and Female
     bars1 = ax.bar(index, gender_distribution['M'], bar_width, label='Male', color␣
      ↪= 'skyblue')
     bars2 = ax.bar(index + bar_width, gender_distribution['F'], bar_width,␣
      ↪label='Female', color = 'lightcoral')

     # Adding number labels on the bars
     for bar in bars1:
```

```
    height = bar.get_height()
    ax.text(bar.get_x() + bar.get_width() / 2.0, height, f'{int(height)}',␣
  ↪ha='center', va='bottom')

for bar in bars2:
    height = bar.get_height()
    ax.text(bar.get_x() + bar.get_width() / 2.0, height, f'{int(height)}',␣
  ↪ha='center', va='bottom')

# plot
ax.set_xlabel('Cluster')
ax.set_ylabel('Number of Customers')
ax.set_title('Gender Distribution by Cluster (Grouped)')
ax.set_xticks(index + bar_width / 2)
ax.set_xticklabels(gender_distribution.index)
ax.legend()
plt.tight_layout()
plt.show()
```

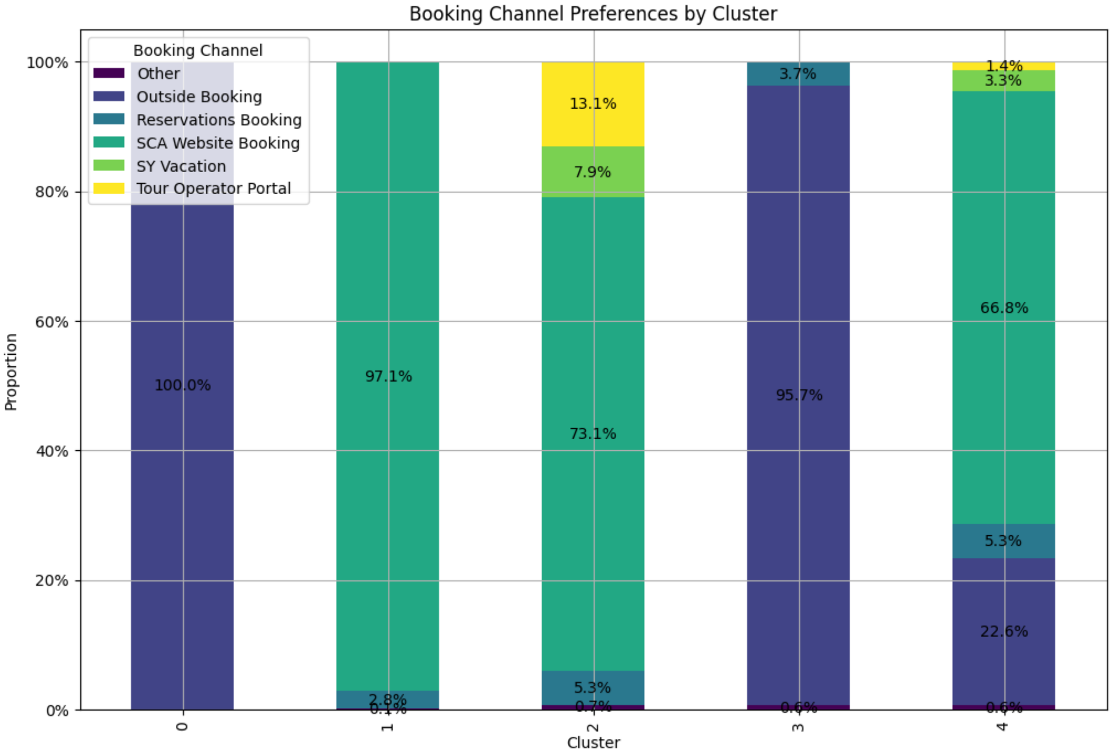


1.6.4 Customer Booking Channel Preferrences by Segment

[ ]:

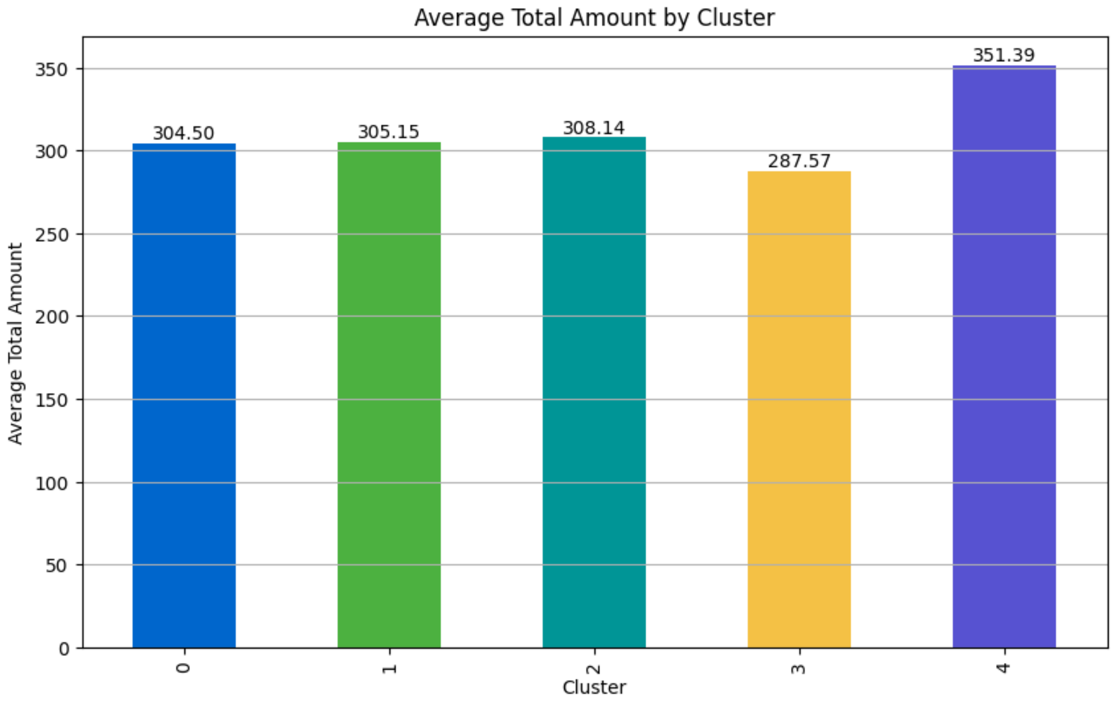
```
import matplotlib.ticker as mtick

# creating stacked and percentage data
```

```
booking_channel_distribution = final_df.groupby(['Cluster', 'BookingChannel']).
 ↪size().unstack().fillna(0)
booking_channel_distribution_percentage = booking_channel_distribution.
 ↪div(booking_channel_distribution.sum(axis=1), axis=0)

ax = booking_channel_distribution_percentage.plot(kind='bar', stacked=True,␣
 ↪figsize=(12, 8), colormap='viridis')

# adding percent labels to the bars
for i, cluster in enumerate(booking_channel_distribution_percentage.index):
    cumulative = 0
    for j, booking_channel in enumerate(booking_channel_distribution_percentage.
 ↪columns):
        value = booking_channel_distribution_percentage.iloc[i, j]
        if value > 0:
            cumulative += value
            plt.text(i, cumulative - value / 2, f'{value:.1%}', ha='center',␣
 ↪va='center', color='black', fontsize=10)

# plot
plt.title('Booking Channel Preferences by Cluster')
plt.xlabel('Cluster')
plt.ylabel('Proportion')
plt.legend(title='Booking Channel')
plt.gca().yaxis.set_major_formatter(mtick.PercentFormatter(1.0))
plt.grid(True)
plt.show()
```



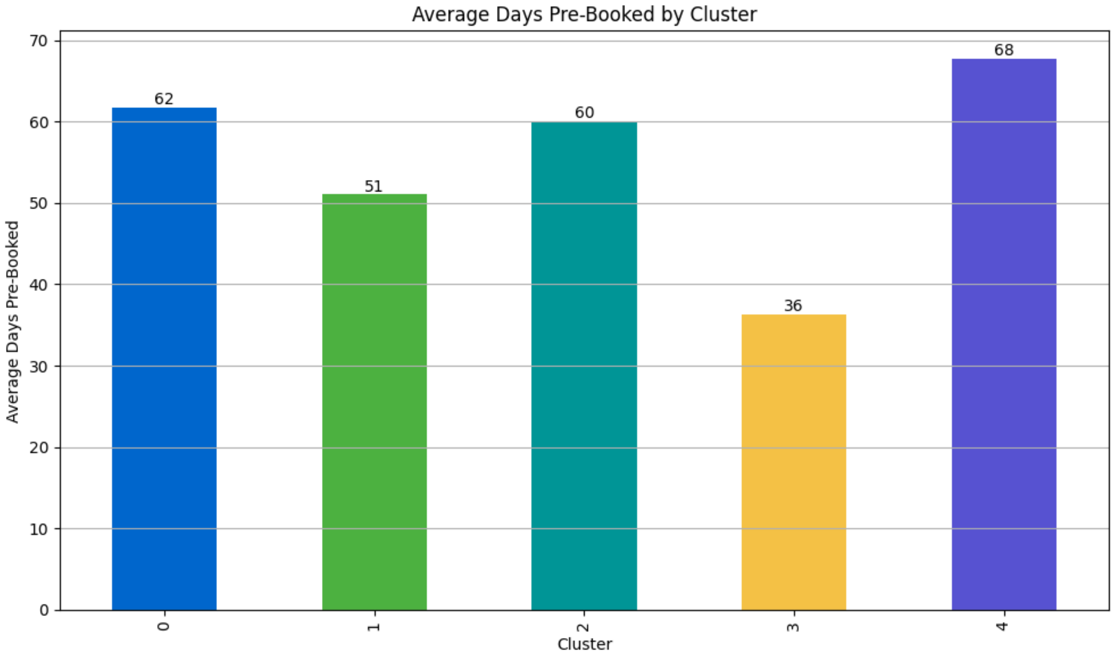
1.6.5 Average Total Amount Spent by Segment

```
[ ]: # preparing the data for plot
avg_amt_by_cluster = final_df.groupby('Cluster')['TotalDocAmt'].mean()
plt.figure(figsize=(10, 6))

ax = avg_amt_by_cluster.plot(kind='bar', color=colors)

# Adding number labels on the bars
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.
 ↪get_height()),
                ha='center', va='center', xytext=(0, 5), textcoords='offset␣
 ↪points')

plt.title('Average Total Amount by Cluster')
plt.xlabel('Cluster')
plt.ylabel('Average Total Amount')
plt.grid(axis='y')
plt.show()
```



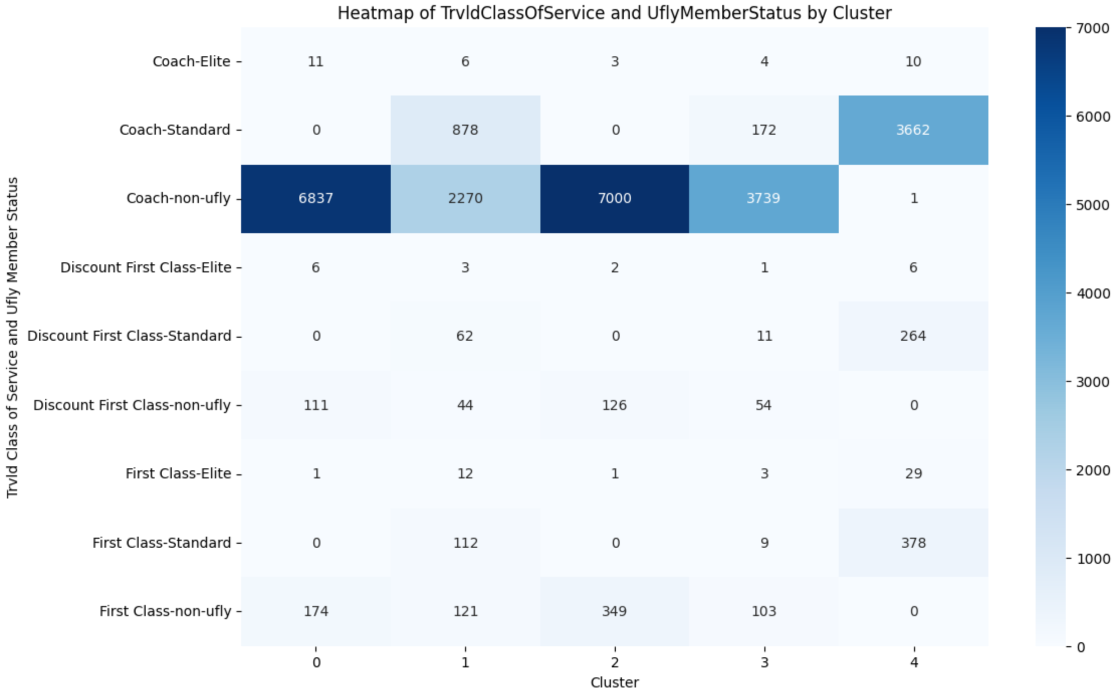
1.6.6 Average Days Pre-Booked by Segment

[ ]:
```
# Grouping data by clusters to calculate average days pre-booked
days_pre_booked_avg = final_df.groupby('Cluster')['days_pre_booked'].mean()

plt.figure(figsize=(10,6))
ax = days_pre_booked_avg.plot(kind='bar', color=colors)

# Adding number labels on the bars
for p in ax.patches:
    ax.annotate(f'{p.get_height():.0f}', (p.get_x() + p.get_width() / 2., p.
 ↪get_height()),
                ha='center', va='center', xytext=(0, 5), textcoords='offset␣
 ↪points')

# plot
plt.title('Average Days Pre-Booked by Cluster')
plt.xlabel('Cluster')
plt.ylabel('Average Days Pre-Booked')
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```



1.6.7 Travel Class and Memeber Status by Segment

```
[ ]: # setting up the data for the heatmap
     crosstab_data = pd.crosstab([final_df['TrvldClassOfService'],␣
      ↪final_df['UflyMemberStatus']], final_df['Cluster'])

     plt.figure(figsize=(12, 8))

     # Create the heatmap
     sns.heatmap(crosstab_data, annot=True, fmt="d", cmap="Blues")

     # Customize the plot
     plt.title('Heatmap of TrvldClassOfService and UflyMemberStatus by Cluster')
     plt.xlabel('Cluster')
     plt.ylabel('Trvld Class of Service and Ufly Member Status')

     plt.show()
```


Heatmap of TrvldClassOfService and UflyMemberStatus by Cluster
Trvld Class of Service and Ufly Member Status
Cluster
Coach-Elite
Coach-Standard
Coach-non-ufly
Discount First Class-Elite
Discount First Class-Standard
Discount First Class-non-ufly
First Class-Elite
First Class-Standard
First Class-non-ufly
0
1
2
3
4
11 6 3 4 10
0 878 0 172 3662
6837 2270 7000 3739 1
6 3 2 1 6
0 62 0 11 264
111 44 126 54 0
1 12 1 3 29
0 112 0 9 378
174 121 349 103 0
7000
6000
5000
4000
3000
2000
1000
0