Aircraft Occupancy Optimization

Data-Driven Approaches to Strengthen Airline Profitability

Introduction

- **Business Overview:**

  We operate a fleet of small business jets and medium-sized aircraft, delivering safe, comfortable, and reliable air transportation.

- **Business Challenges:**

  - **Stricter environmental regulations → Higher compliance costs**
  - **Rising fuel prices → Increased operational expenses**
  - **Higher taxes → Additional financial pressure**
  - **Tight labor market → Higher wages and turnover**
  - **Elevated interest rates → Costlier financing**

- **Goal of the Project:**

  - **Increase aircraft occupancy rates**
  - **Refine pricing strategies**
  - **Improve profitability while maintaining service quality**

Tools & Technology

- **Python (Pandas, NumPy, Matplotlib, Seaborn)** → Data analysis & visualization
- **Jupyter Notebook** → Interactive data exploration
- **Excel** → Supplementary calculations, data checks

Data Cleaning & Preprocessing

- Removed duplicates and irrelevant entries.
- Addressed missing values in ticket and price data.
- Standardized data types for dates, fares, and seat counts.
- Filtered out inactive aircraft and cancelled flights.
- Created new fields: occupancy rate, average revenue per seat.

Key Challenges

Stricter Environmental Regulations:

The demand on the airlines industry to decrease its carbon footprint is growing, which has resulted in more stringent environmental laws that raise operating costs and restrict expansion potential.

Higher Flight Taxes:

To solve environmental issues and increase money, governments all around the world are taxiing aircraft more heavily, which raises the cost of flying and decreases demand.

Tight labor market resulting in increased labor costs:

The lack of trained people in the aviation sector has increased labor costs and increased turnover rates.

Basic Analysis

- **How many planes have more than 100 seats?**

```
pd.read_sql_query(f"""SELECT aircraft_code, COUNT(*) as num_seats FROM seats
                      GROUP BY aircraft_code
                      HAVING num_seats > 100
                      ORDER BY num_seats DESC""", connection)
```

	aircraft_code	num_seats
0	773	402
1	763	222
2	321	170
3	320	140
4	733	130
5	319	116

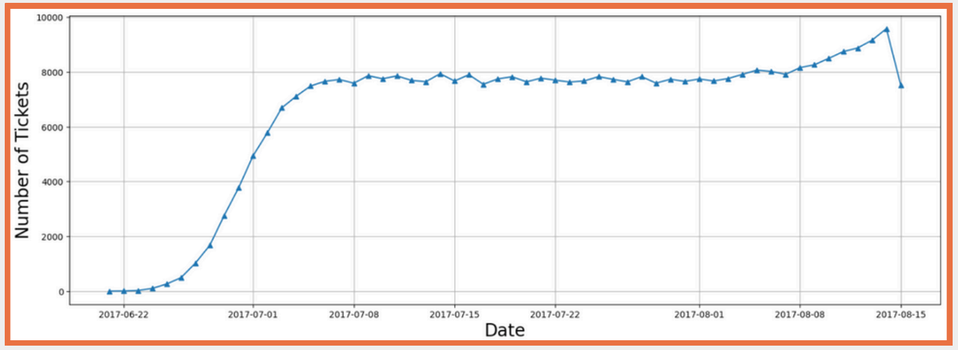
Focus:

Improve seat utilization across this fleet.

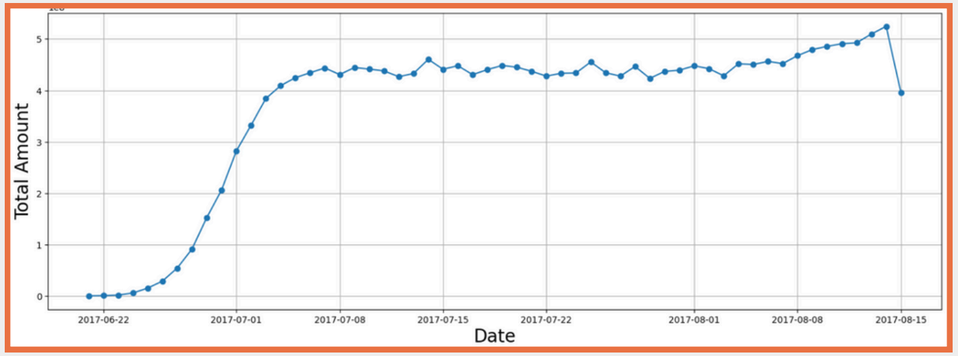
Basic Analysis

- **How the number of tickets booked and total amount earned changed with the time?**

```
tickets = pd.read_sql_query(f"""SELECT *
                                FROM tickets
                                INNER JOIN bookings
                                ON tickets.book_ref=bookings.book_ref;""", connection)
tickets['book_date'] = pd.to_datetime(tickets['book_date'])
tickets['date'] = tickets['book_date'].dt.date
x =  tickets.groupby('date')[['date']].count()
plt.figure(figsize = (18,6))
plt.plot(x.index,x['date'], marker = '^')
plt.xlabel('Date', fontsize = 20)
plt.ylabel('Number of Tickets', fontsize = 20)
plt.grid('b')
plt.show()
```




```
bookings = pd.read_sql_query(f"""SELECT * FROM bookings""", connection)
bookings['book_date'] = pd.to_datetime(bookings['book_date'])
bookings['date'] = bookings['book_date'].dt.date
y =  bookings.groupby('date')[['total_amount']].sum()
plt.figure(figsize = (18,6))
plt.plot(y.index,y['total_amount'], marker = 'o')
plt.xlabel('Date', fontsize = 20)
plt.ylabel('Total Amount', fontsize = 20)
plt.grid('b')
plt.show()
```



Insights:

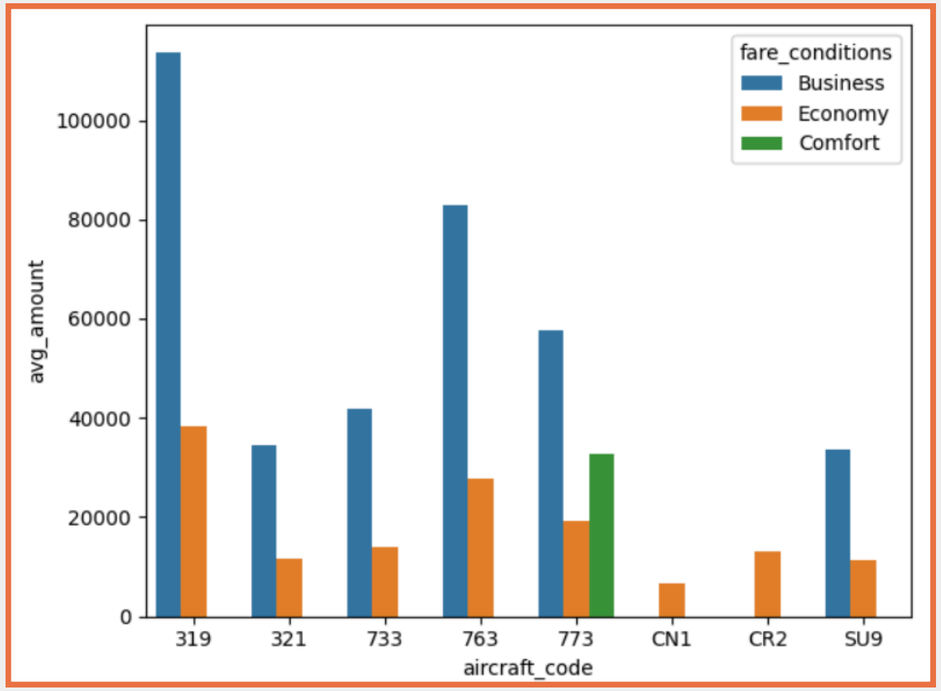
The business saw rapid growth in revenue and ticket sales from late June to early July 2017, followed by a stable period. A sharp spike and drop in mid-August suggest the end of a promotion or a significant event. Monitoring these trends can help optimize future campaigns and address sudden changes.

Basic Analysis

- **Calculate the average charges for each aircraft with different fare conditions.**

```
df = pd.read_sql_query(f"""SELECT fare_conditions, aircraft_code, AVG(amount) as avg_amount FROM ticket_flights
                        JOIN flights
                        ON ticket_flights.flight_id=flights.flight_id
                        GROUP BY aircraft_code, fare_conditions""", connection)

sns.barplot(data = df, x = 'aircraft_code', y ='avg_amount', hue = 'fare_conditions')
plt.tight_layout()
```



Insights:

Business class generates the highest average revenue across most aircraft types, especially for aircraft code 319. Economy fares are consistently lower, while Comfort class appears only for aircraft 773 with moderate revenue. Focusing on Business class offerings, particularly for high-performing aircraft, could maximize profitability.

Analyzing Occupancy Rate

- **For each aircraft, calculate the total revenue per year and the average revenue per ticket.**

```
pd.set_option('display.float_format', str)

pd.read_sql_query(f"""SELECT aircraft_code, total_revenue, ticket_count, total_revenue/ticket_count as avg_revenue_per_ticket
                      FROM
                      (SELECT aircraft_code, COUNT(*) as ticket_count, SUM(amount) as total_revenue  FROM ticket_flights
                      JOIN flights
                      ON ticket_flights.flight_id=flights.flight_id
                      GROUP BY aircraft_code)""", connection)
```

	aircraft_code	total_revenue	ticket_count	avg_revenue_per_ticket
0	319	2706163100	52853	51201
1	321	1638164100	107129	15291
2	733	1426552100	86102	16568
3	763	4371277100	124774	35033
4	773	3431205500	144376	23765
5	CN1	96373800	14672	6568
6	CR2	1982760500	150122	13207
7	SU9	5114484700	365698	13985

Analyzing Occupancy Rate

- **Calculate the average occupancy per aircraft.**

```
occupancy_rate = pd.read_sql_query(f"""SELECT a.aircraft_code, AVG(a.seats_count) as booked_seats, b.num_seats,
AVG(a.seats_count)/b.num_seats as occupancy_rate
                            FROM (
                                SELECT aircraft_code, flights.flight_id, COUNT(*) as seats_count
                                FROM boarding_passes
                                INNER JOIN flights
                                ON boarding_passes.flight_id=flights.flight_id
                                GROUP BY aircraft_code, flights.flight_id
                                ) as a INNER JOIN
                                (
                                SELECT aircraft_code, COUNT(*) as num_seats FROM seats
                                GROUP BY aircraft_code
                                ) as b
                                ON a.aircraft_code = b.aircraft_code
                            GROUP BY a.aircraft_code""", connection)
occupancy_rate
```

	aircraft_code	booked_seats	num_seats	occupancy_rate
0	319	53.58318098720292	116	0.46192397402761143
1	321	88.80923076923077	170	0.5224072398190045
2	733	80.25546218487395	130	0.617349709114415
3	763	113.93729372937294	222	0.5132310528350132
4	773	264.9258064516129	402	0.659019419033863
5	CN1	6.004431314623338	12	0.5003692762186115
6	CR2	21.48284690220174	50	0.42965693804403476
7	SU9	56.81211267605634	97	0.5856918832583128

Analyzing Occupancy Rate

- **Calculate by how much the total annual turnover could increase by giving all aircraft a 10% higher occupancy rate.**

```
occupancy_rate['Inc occupancy rate'] = occupancy_rate['occupancy_rate'] + occupancy_rate['occupancy_rate']*0.1
occupancy_rate
```

	aircraft_code	booked_seats	num_seats	occupancy_rate	Inc occupancy rate
0	319	53.58318098720292	116	0.46192397402761143	0.5081163714303726
1	321	88.80923076923077	170	0.5224072398190045	0.574647963800905
2	733	80.25546218487395	130	0.617349709114415	0.6790846800258565
3	763	113.93729372937294	222	0.5132310528350132	0.5645541581185146
4	773	264.9258064516129	402	0.659019419033863	0.7249213609372492
5	CN1	6.004431314623338	12	0.5003692762186115	0.5504062038404727
6	CR2	21.48284690220174	50	0.42965693804403476	0.4726226318484382
7	SU9	56.81211267605634	97	0.5856918832583128	0.644261071584144

Analyzing Occupancy Rate

- **Calculate by how much the total annual turnover could increase by giving all aircraft a 10% higher occupancy rate.**

```
occupancy_rate['Inc occupancy rate'] = occupancy_rate['occupancy_rate'] + occupancy_rate['occupancy_rate']*0.1
occupancy_rate
```

	aircraft_code	booked_seats	num_seats	occupancy_rate	Inc occupancy rate
0	319	53.58318098720292	116	0.46192397402761143	0.5081163714303726
1	321	88.80923076923077	170	0.5224072398190045	0.574647963800905
2	733	80.25546218487395	130	0.617349709114415	0.6790846800258565
3	763	113.93729372937294	222	0.5132310528350132	0.5645541581185146
4	773	264.9258064516129	402	0.659019419033863	0.7249213609372492
5	CN1	6.004431314623338	12	0.5003692762186115	0.5504062038404727
6	CR2	21.48284690220174	50	0.42965693804403476	0.4726226318484382
7	SU9	56.81211267605634	97	0.5856918832583128	0.644261071584144

Analyzing Occupancy Rate

- **Calculate by how much the total annual turnover could increase by giving all aircraft a 10% higher occupancy rate.**

```
total_revenue = pd.read_sql_query("""SELECT aircraft_code, SUM(amount) as total_revenue  FROM ticket_flights
                        JOIN flights
                        ON ticket_flights.flight_id=flights.flight_id
                        GROUP BY aircraft_code""", connection)

occupancy_rate['Inc Total Annual Turnover'] = (total_revenue['total_revenue']/occupancy_rate['occupancy_rate'])
*occupancy_rate['Inc occupancy rate']
occupancy_rate
```

	aircraft_code	booked_seats	num_seats	occupancy_rate	Inc occupancy rate	Inc Total Annual Turnover
0	319	53.58318098720292	116	0.46192397402761143	0.5081163714303726	2976779410.0
1	321	88.80923076923077	170	0.5224072398190045	0.574647963800905	1801980510.0
2	733	80.25546218487395	130	0.617349709114415	0.6790846800258565	1569207310.0000002
3	763	113.93729372937294	222	0.5132310528350132	0.5645541581185146	4808404810.0
4	773	264.9258064516129	402	0.659019419033863	0.7249213609372492	3774326050.0
5	CN1	6.004431314623338	12	0.5003692762186115	0.5504062038404727	106011180.00000001
6	CR2	21.48284690220174	50	0.42965693804403476	0.4726226318484382	2181036550.0
7	SU9	56.81211267605634	97	0.5856918832583128	0.644261071584144	5625933169.999999

Key Insights from Analysis

- **Ticket bookings rose June 22 – July 7, then stabilized.**
- **SU9 had highest revenue due to high volume despite low prices.**
- **CN1 had lowest revenue: economy-only, least demand.**
- **Business class fares consistently higher but underutilized on some aircraft.**

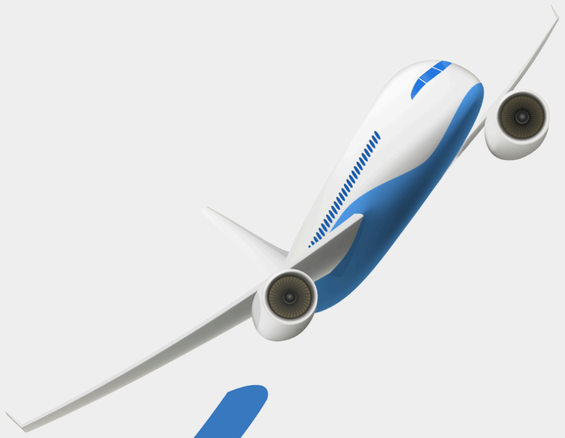
Impact of Occupancy Optimization

- **A 10% occupancy increase could drive clear, gradual revenue gains.**
- **Better alignment of price, demand, and service can unlock untapped profitability.**
- **Focus areas: pricing, flight scheduling, customer targeting.**

Key Outcomes

- **Identified specific aircraft/routes with low occupancy and low yield.**
- **Highlighted pricing mismatches that limit revenue potential.**
- **Modeled impact of higher occupancy on total turnover.**
- **Delivered actionable recommendations for pricing and service improvements.**

Conclusion



- **Raising occupancy is key to protecting margins under rising costs.**
- **Smart pricing, better scheduling, and improved customer experience can lift performance.**
- **Data-driven strategies will help balance profit, customer satisfaction, and compliance.**
- **Next steps: Test pricing adjustments, monitor results, iterate.**

Thank You

bhaskarpal.official@gmail.com