Project 2
(Optimization II-RM294)

Submitted by:

Teagan Milford, Shrishty Mishra, Ruby Nie, Sanjhana Rangaraj

March 2024

Table of Contents :

Project Overview

This project aims to investigate airline ticket sales scenarios, considering the impacts of price on purchasing behavior and overbooking on overall profits. The outcome of the project is an optimized strategy for airline ticket sales that maximizes the expected discounted profit from a particular price. Importantly, for the context of this project, only ticket revenue and overbooking costs are considered in calculating profits. Other costs are considered fixed costs that cannot be avoided or optimized.

The main issue at hand is the impact of overbooking. While overbooking can increase revenue in the case that not all booked passengers show up for the flight, it introduces the risk of having insufficient seats. If there are more booked passengers than seats available, the airline must pay a stipend to the passengers who are unable to board.

Using dynamic programming, we tested the optimal pricing policies based on different overbooking policies to identify the policy with the largest expected discounted profit.

Problem Definition

In order to identify the optimal strategy, we considered different pricing and overbooking scenarios. There are two ticket types: coach and first class. As a general rule, first class seats cannot be overbooked, to ensure that high paying customers are not disrupted. Coach seats can be overbooked. Any first class seats that are not booked can be used to accommodate customers who purchased coach seats.

All scenarios make the following assumptions:

There are 365 days until the plane departs, and up to 1 ticket of each type can be sold each day. On the day of the flight, no additional tickets can be sold for revenue, so there are only costs. The demand for coach and first class seats are independent. The probability that a ticket is sold for each ticket type depends on the price of the ticket. Initially, there are 100 coach seats and 20 first class seats available. The annual discount rate is 17%, so the daily discount factor is $\frac{1}{\frac{1.17}{365}}$.

<u>*State Variables:*</u>

- **Time (*T*)**: The current day relative to the flight departure, with $T = 0$ being the start and $T = 364$ being the day before departure.
- **Number of Coach Tickets Still Available at the High and Low Prices (*sc*)**: The number of coach tickets available up to time *T*.
- **Number of First Class Tickets Still Available at the High and Low Prices (*sf*)**: The number of first-class tickets available up to time *T*.

The state variables consist of the number of available seats in coach, the current time, and the number of available seats in first class. This can be symbolized as (*sc, t, sf*).

<u>*Choice Variables:*</u>

The choice variables are described as options or decisions and can be denoted as follows:

Variable	Coach Price	First-Class Price
$L_{sc}L_{sf}$	Low	Low
$L_{sc}H_{sf}$	Low	High
$H_{sc}L_{sf}$	High	Low
$H_{sc}H_{sf}$	High	High

<u>*Dynamics:*</u>

To transition from one state to another, it is necessary to determine the quantity of tickets sold today and the total number of tickets sold to date for both coach and first-class sections. Time

will invariably progress from one state to the next. The potential subsequent states can be depicted as follows:
- (*sc, t+1, sf*) indicating unsold tickets in coach and first-class for the day,
- (*sc-1, t+1, sf*) indicating a ticket sold in coach but not in first-class for the day,
- (*sc-1, t+1, sf-1*) indicating tickets sold in both coach and first-class,
- (*sc, t+1, sf-1*) indicating a ticket sold in first-class but not in coach for the day.

Value Function: Represents the maximum expected discounted profit that can be obtained from time to onward, given the current state of available seats in coach and first-class. It aims to find the optimal combination of pricing decisions to maximize profits over the remaining time.

$$V(sc,t,sf)$$

Terminal Condition: The terminal condition occurs at the time of departure (t = 364),

$$V(sc,0,sf) = -\text{ Overbooking Costs.}$$

Bellman Equation:

$$V(sc, sf, t) = \max \{ E\,[revenue + \delta v_{t+1} \text{ if } L_{sc}L_{sf}],$$
$$[revenue + \delta v_{t+1} \text{ if } L_{sc}H_{sf}],\ [revenue + \delta v_{t+1} \text{ if } H_{sc}L_{sf}],$$
$$[revenue + \delta v_{t+1} \text{ if } H_{sc}H_{sf}] \}$$

Pricing Scenarios

Coach Tickets

When first-class seats are sold out, the chance of sale in coach increases by 3%, regardless of price.

Ticket Price	\$300	\$350
Probability of Sale (first-class available)	65%	30%
Probability of Sale (first-class sold out)	68%	33%

First Class Tickets

Ticket Price	$425	$500
Probability of Sale	8%	4%

Each coach passenger shows up with a probability of 95%. Each first-class passenger shows up with a probability of 97%.

	Coach Seat	First-Class Seat
Probability of Shows Up	95%	97%

Overbook Cost

	Bump Coach to First-Class	Bump Coach Off the Plane
Bumping Cost	$50	$425

Given the constraints, we implement the rules into code as below:

```
# initializing variables
capacity = {'coach':100,'firstclass':20} # capacity of the flight in each of the ticket classes
overbook = {'coach':5,'firstclass':0} # over-booking policy in each of the ticket classes.
bumpingcosts = {'tofirst':50, 'planeoff':425} # cost of bumping off passengers, to first class and off plane
ticketprice = {'coach':[300,350],'firstclass':[425,500]} # low and high ticket price options for each of the ticket classes
sale_prob = {'coach':[0.65,0.3],'firstclass':[0.08,0.04]} # probabilities of sale for each of the pricing
delta = 1/(1+0.17/365) # discount rate
T = 365 # time from start of sale of tickets to flight departure
```

Analysis

Strategy 1

Scenario 1: Oversell by 5 Seats

Utilizing dynamic programming, the optimal pricing policy listed above is sought to maximize the expected discounted profit, considering the probabilities of ticket sales and the likelihood of passengers showing up. When overbooking by 5 coach seats, we get a significant expected discounted profit of approximately **$41,886.16**.

The following set of plots provided depict the state of available seats in both the coach and first-class sections of a flight at various points in time before departure. Each plot is a heatmap representing the distribution of available seats in the two classes over time.

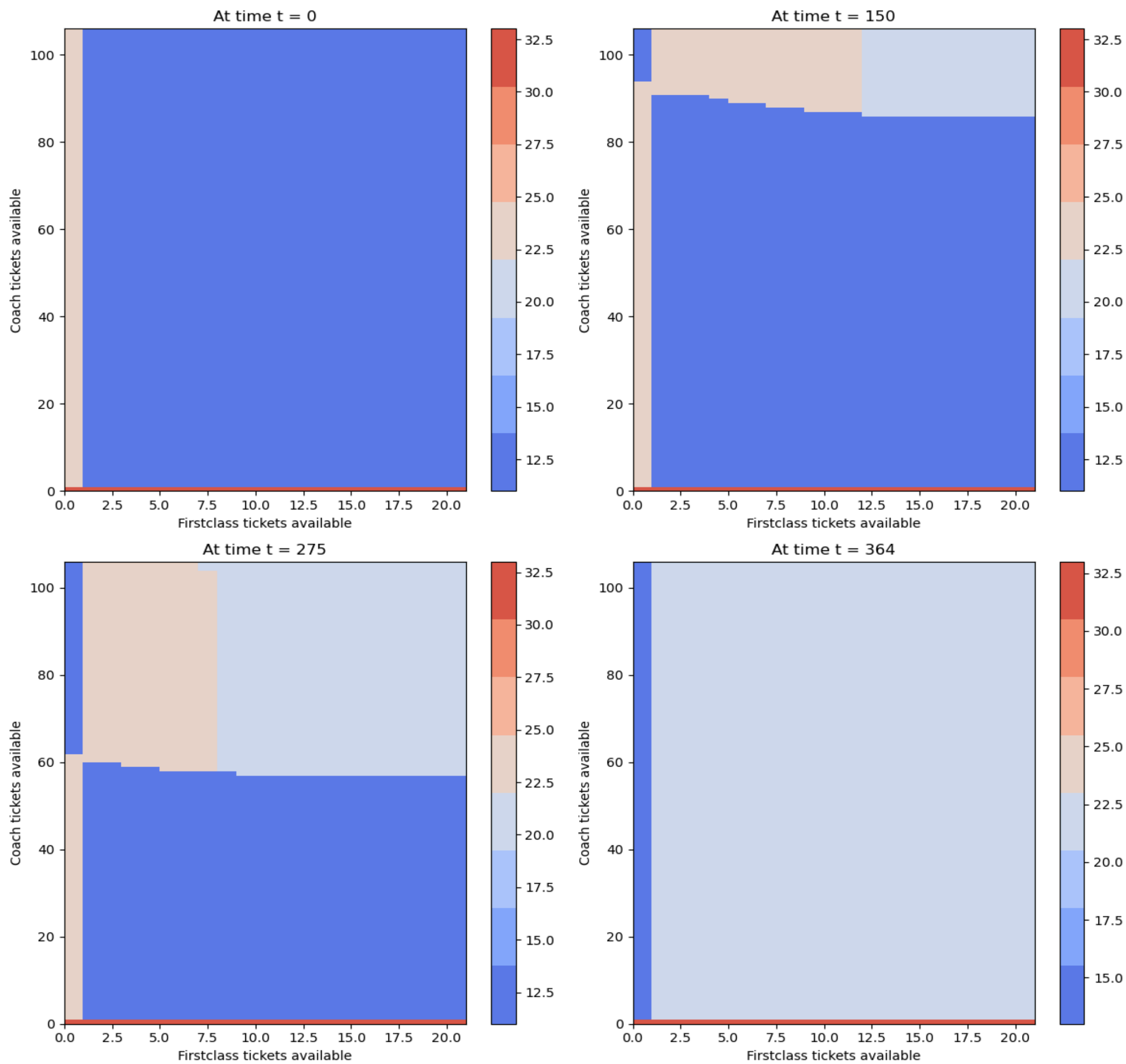

Figure 1. Distribution of seats in first-class and coach over time.

At Time $t = 0$: The plot indicates that all seats in both coach and first-class are available. This is the starting condition of the model, representing a full capacity of 100 coach and 20 first-class seats ready to be sold.

At Time $t = 150$:
The distribution shows a variance in the number of coach seats sold with some first-class seats also being sold. As time progresses, the airline has started to sell tickets, and the seats in both

classes are being filled. The shaded area indicates the combination of seats sold in both coach and first-class at this time point.

At Time $t = 275$: As time continues to progress, more tickets are being sold. The plot shows that the number of available seats in coach has decreased significantly, indicating that the airline has been successful in selling coach tickets over time. First-class seats also continue to be sold but at a slower rate than coach seats. The different shades suggest variations in the remaining capacity across different pricing strategies or demand conditions.

At Time $t = 364$:
The day before the flight, the plot suggests that nearly all coach and first-class seats have been sold. The few remaining seats, particularly in first-class, imply that the airline has reached its overbooking limit.

Customers would start getting bumped to first-class when the coach class is overbooked (approaching or exceeding the number of seats available) and there are still available seats in first-class.

In the heatmap series, we can see that the coach class appears to be full or nearly full closer to the departure date (e.g., at time t=364), while there is still some availability in first-class. This suggests that at some point between time t =275 and t=364, if overbooking occurs, customers from the coach may be bumped to the available first-class seats.

Code

The following code snippets define a function called 'optimize_profit', which takes parameters including airplane capacity, overbook constraint, overbook bumping costs, high and low ticket prices, sales probability, discount rate, and time *T*. By feeding in the parameters, we get the expected profit mentioned above for this case.

```
def optimize_profit(capacity,overbook,bumpingcosts,ticketprice,sale_prob,delta,T):

    scValues = np.arange(capacity['coach']+overbook['coach']+1) # all possible number of seats left in coach
    sfValues = np.arange(capacity['firstclass']+overbook['firstclass']+1) # all possible number of seats left in firstclass
    tValues = np.arange(T+1) # all possible days until takeoff
    scN = len(scValues)# count possible state values
    sfN = len(sfValues)# count possible state values
    tN = len(tValues)
    V = np.zeros((scN,tN,sfN)) # initialize value function
```

```
    U = np.zeros((scN,tN,sfN)) # initialize optimal choice variable

    LC_price = ticketprice['coach'][0] # lower choice of ticket price of a coach ticket
    HC_price = ticketprice['coach'][1]  # higher choice of ticket price of a coach ticket
    LF_price = ticketprice['firstclass'][0] # lower choice of ticket price of a first class ticket
    HF_price = ticketprice['firstclass'][1] # higher choice of ticket price of a first class ticket

    pcL = [1-sale_prob['coach'][0],sale_prob['coach'][0]] # demand probabilities for low ticket price
coach ticket
    pcH = [1-sale_prob['coach'][1],sale_prob['coach'][1]] # demand probabilities for high ticket price
coach ticket
    pfL = [1-sale_prob['firstclass'][0],sale_prob['firstclass'][0]] # demand probabilities for low ticket
price first class ticket
    pfH = [1-sale_prob['firstclass'][1],sale_prob['firstclass'][1]] # demand probabilities for high ticket
price first class ticket
    Coach_cap = capacity['coach'] # number of seats in coach
    First_cap = capacity['firstclass'] # number of seats in first class
```

```
#boundary/terminal condition
    U[:,tN-1,:] = 33  # 1: lower ticket price, # 2: higher ticket price, # 3  value for no tickets for sale.

    # airline cost calculation for each of the scenarios of number of seats still available in coach and
first class at the time of takeoff.
    for sc in range(scN):
        tc = scN - sc-1 # number of coach tickets booked
        for sf in range(sfN):
            tf = sfN - sf-1 # number of first class tickets booked
            airline_cost = 0

            for tc_actual in range(tc+1): # looping over possible coach people actually showing up
                pr_tc_act = scipy.stats.binom.pmf(tc_actual,tc,0.95) # probability of  actual number of

                for tf_actual in range(tf+1): # looping over  first class passengers actually showing up
                    pr_tf_act = scipy.stats.binom.pmf(tf_actual,tf,0.97)# probability no. of people showing up

                    if tc_actual<=Coach_cap: #no (people) who showed up is less than the capacity of coach
                        airline_cost = airline_cost+0
                    else:
                        if tc_actual-Coach_cap <= First_cap-tf_actual:# seats in first class > if number (coach
passengers) showing up > capacity(coach)
                        # airline cost = airline cost to bump to first class
                            airline_cost = airline_cost -
```

```
pr_tc_act*pr_tf_act*bumpingcosts['tofirst']*(tc_actual-Coach_cap)
                    else:
                    # Costs include upgrading to first class until full, then offloading the rest, if coach
arrivals exceed both coach and first-class capacities.
                     airline_cost = airline_cost -
pr_tc_act*pr_tf_act*(bumpingcosts['tofirst']*(First_cap-tf_actual)
                                                    +
bumpingcosts['planeoff']*(tc_actual-Coach_cap-(First_cap-tf_actual)))
          V[sc,tN-1,sf] = airline_cost
```

```
  # using Bellman equation to calculate the expected Value backward in time for each of the scenarios
   for t in reversed(range(tN-1)):
      for sc in range(scN):
         for sf in range(sfN):
            if sc==0 and sf==0: # is the flight full (0 seats left)
               V[sc,t,sf]= delta*V[sc,t+1,sf] # if so, you can't make any more money
               U[sc,t,sf]=33 # No tickets can be sold.

            elif sc !=0 and sf==0: # coach tickets available but not first class, increasing the  probability
of sale coach tickets by 0.03.
               # value if coach ticket price low
               CL_value = (pcL[1]+0.03)*LC_price + delta* ((pcL[0]-0.03)*V[sc,t+1,sf] +
(pcL[1]+0.03)*V[sc-1,t+1,sf])
               # value if coach ticket price high
               CH_value = (pcH[1]+0.03)*HC_price + delta* ((pcH[0]-0.03)*V[sc,t+1,sf] +
(pcH[1]+0.03)*V[sc-1,t+1,sf])

               V[sc,t,sf]=max(CL_value,CH_value) # value function maximizes expected profit
               U[sc,t,sf]=(np.argmax([CL_value,CH_value])+1)*10+3

            elif sc == 0 and sf!=0: # there are first class tickets available but not coach
               # value if you set the first class ticket price low
               LF_value = (pfL[1])*LF_price + delta* ((pfL[0])*V[sc,t+1,sf] + (pfL[1])*V[sc,t+1,sf-1])
               # value if you set the first class ticket price high
               HF_value = (pfH[1])*HF_price + delta* ((pfH[0])*V[sc,t+1,sf] + (pfH[1])*V[sc,t+1,sf-1])

               V[sc,t,sf]=max(LF_value,HF_value) # value function maximizes expected profit
               U[sc,t,sf]=30+np.argmax([LF_value,HF_value])+1

            else:
               # value if you set both coach and first class ticket prices low
               LL_value = pcL[1]*LC_price + pfL[1]*LF_price + delta* (pfL[0]*pcL[0]*V[sc,t+1,sf]
```

```
                                    +pfL[0]*pcL[1]*V[sc-1,t+1,sf]
                                +pfL[1]*pcL[1]*V[sc-1,t+1,sf-1]
                                    +pfL[1]*pcL[0]*V[sc,t+1,sf-1])

            # value if you set coach ticket price low and first class ticket price high
            LH_value= pcL[1]*LC_price + pfH[1]*HF_price + delta* (pfH[0]*pcL[0]*V[sc,t+1,sf]
                                    +pfH[0]*pcL[1]*V[sc-1,t+1,sf]
                                    +pfH[1]*pcL[1]*V[sc-1,t+1,sf-1]
                                    +pfH[1]*pcL[0]*V[sc,t+1,sf-1])

            # value if you set coach ticket price high and first class ticket price low
            HL_value = pcH[1]*HC_price + pfL[1]*LF_price + delta* (pfL[0]*pcH[0]*V[sc,t+1,sf]
                                    +pfL[0]*pcH[1]*V[sc-1,t+1,sf]
                                    +pfL[1]*pcH[1]*V[sc-1,t+1,sf-1]
                                    +pfL[1]*pcH[0]*V[sc,t+1,sf-1])

            # value if you set both coach and first class ticket prices high
            HH_value = pcH[1]*HC_price + pfH[1]*HF_price + delta* (pfH[0]*pcH[0]*V[sc,t+1,sf]
                                    +pfH[0]*pcH[1]*V[sc-1,t+1,sf]
                                    +pfH[1]*pcH[1]*V[sc-1,t+1,sf-1]
                                    +pfH[1]*pcH[0]*V[sc,t+1,sf-1])

            V[sc,t,sf]=max(LL_value,LH_value,HL_value,HH_value) # value function maximizes
expected profit

            U[sc,t,sf]=(int(np.argmax([LL_value,LH_value,HL_value,HH_value])/2)+1)*10\
            +int((np.argmax([LL_value,LH_value,HL_value,HH_value])%2)!=0)+1
  return U,V
```

Scenario 2: Oversell by 6-15 Seats

In this scenario, coach seats are allowed to be oversold by 6 to 15 seats.

Overbook Seats	Profit
5	$41,886.16
6	$42,011.22
7	$42,085.54
8	$42,122.17

9	**$42,134.62**
10	$42,132.90
11	$42,123.67
12	$42,111.03
13	$42,097.42
14	$42,084.11
15	$42,071.74

The expected profit is the highest at allowing 9 overbooked coach seats, $42,134.62.

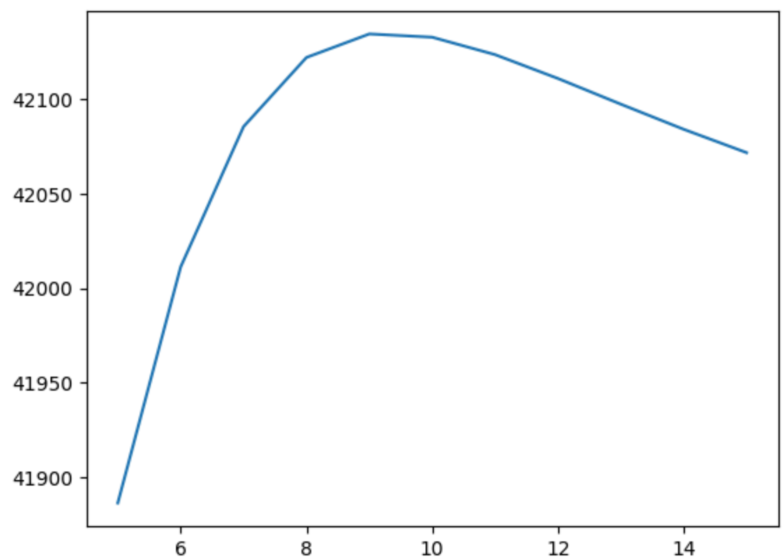

Figure 2. Expected profit based on number of overbooked seats

The plot above shows the different profits for different overbooked coach seats. The x-axis displays the number of overbooked coach seats allowed. The y-axis represents the expected profits.

Starting from 6 overbooked seats, as the number of overbooked seats increases, there is a corresponding rise in expected profits. This suggests that, up to a certain point, overbooking seats in coach is financially beneficial for the airline. The additional revenue generated from selling more tickets than the actual number of available seats compensates for the costs incurred from potential overbooking scenarios, such as having to bump passengers to a higher class or compensate them.

The profit reaches a peak at 9 overbooked seats, which suggests that this is the optimal overbooking level from a profit maximization standpoint.

As the number of overbooked seats increases beyond the optimal level, the expected profit begins to decline. This indicates that the costs associated with overbooking start to outweigh the additional revenue from selling these extra seats.

Code

```
import pandas as pd

overbookext = [i for i in range(5, 16)]
profitinc = {}

for ob in overbookext:
    overbook = {'coach': ob, 'firstclass': 0}
    U, V = optimize_profit(capacity, overbook, bumpingcosts, ticketprice, sale_prob, delta, T)
    profit = V[capacity['coach'] + overbook['coach'], 0, capacity['firstclass'] + overbook['firstclass']]
    profitinc[ob] = profit

# Convert profitinc to a pandas DataFrame
df_profitinc = pd.DataFrame(list(profitinc.items()), columns=['Overbooking Extension', 'Profit Increase'])

# Round the 'Profit Increase' column to 2 decimal places
df_profitinc['Profit Increase'] = df_profitinc['Profit Increase'].round(2)

# Display the DataFrame
print(df_profitinc)
```

Strategy 2

Scenario 3: Option to Sell No Coach Ticket

In scenario 3, a new strategy is introduced. The airline now has the option of forcing the demand for coach tickets to 0 on a day.

- *Coach Seat*: Not Selling, Low Price, High Price
- *First-Class*: Low Price, High Price

Given the assumption that the airline will never overbook more than 20 coach seats and 0 first-class seats, we get the expected profit of $42,139.89.

In scenario 2, the expected profit is $42,134.62. Thus, this policy has a better expected profit than the best policy (hard cap on overbooking 9 coach seats) in scenario 2.

Overbook Seats	Profit
5	$41886.16
6	$42011.22
7	$42085.54
8	$42122.17
9	$42134.62
10	$42138.14
11	$42139.33
12	$42139.71
13	$42139.83
14	$42139.87
15	**$42139.89**
16	$42139.89
17	$42139.89
18	$42139.89
19	$42139.89
20	$42139.89

Code

```
def optimize_profit2(capacity,overbook,bumpingcosts,ticketprice,sale_prob,delta,T):

    scValues = np.arange(capacity['coach']+overbook['coach']+1) # all possible number of seats left in
coach
    sfValues = np.arange(capacity['firstclass']+overbook['firstclass']+1) # all possible number of seats
left in first class
    tValues = np.arange(T+1) # all possible days until takeoff
    scN = len(scValues)# count possible state values
    sfN = len(sfValues)# count possible state values
    tN = len(tValues)
    V = np.zeros((scN,tN,sfN)) # initialize value function
    U = np.zeros((scN,tN,sfN)) # initialize optimal choice variable

    LC_price = ticketprice['coach'][0] # lower choice of ticket price of a coach ticket
    HC_price = ticketprice['coach'][1]  # higher choice of ticket price of a coach ticket
    LF_price = ticketprice['firstclass'][0] # lower choice of ticket price of a first class ticket
    HF_price = ticketprice['firstclass'][1] # higher choice of ticket price of a first class ticket

    pcL = [1-sale_prob['coach'][0],sale_prob['coach'][0]] # demand probabilities for low ticket price
coach ticket
    pcH = [1-sale_prob['coach'][1],sale_prob['coach'][1]] # demand probabilities for high ticket price
coach ticket
    pfL = [1-sale_prob['firstclass'][0],sale_prob['firstclass'][0]] # demand probabilities for low ticket
price first class ticket
    pfH = [1-sale_prob['firstclass'][1],sale_prob['firstclass'][1]] # demand probabilities for high ticket
price first class ticket
    Coach_cap = capacity['coach'] # number of seats in coach
    First_cap = capacity['firstclass'] # number of seats in first class
```

```
   #boundary/terminal condition
    U[:,tN-1,:] = 33 # 1: lower ticket price, # 2: higher ticket price, # 3  value for no tickets for sale.

    # airline cost calculation for each of the scenarios of number of seats still available in coach and first
class at the
    # time of takeoff.
    for sc in range(scN):
        tc = scN - sc-1 # number of coach tickets booked
        for sf in range(sfN):
            tf = sfN - sf-1 # number of first class tickets booked
            airline_cost = 0
            for tc_actual in range(tc+1):# looping over possible number of coach passengers actually
```

```
showing up on the day of departure
            pr_tc_act = scipy.stats.binom.pmf(tc_actual,tc,0.95)# probability of actual number of

            for tf_actual in range(tf+1):# looping over possible number of first class passengers showing
up for departure
               pr_tf_act = scipy.stats.binom.pmf(tf_actual,tf,0.97)# probability of actual number of people
showing up
               if tc_actual<=Coach_cap:#no (people) who showed up is less than the capacity of coach
                  airline_cost = airline_cost+0
               else:
                  if tc_actual-Coach_cap <=First_cap-tf_actual   #seats in first class > if number (coach
passengers) showing up > capacity(coach)
                     airline_cost = airline_cost -
pr_tc_act*pr_tf_act*bumpingcosts['tofirst']*(tc_actual-Coach_cap)
                  else:
                 # Costs include upgrading to first class until full, then offloading the rest, if coach arrivals
exceed both coach and first-class capacities
                     airline_cost = airline_cost -
pr_tc_act*pr_tf_act*(bumpingcosts['tofirst']*(First_cap-tf_actual)
                                                +
bumpingcosts['planeoff']*(tc_actual-Coach_cap-(First_cap-tf_actual)))
         V[sc,tN-1,sf] = airline_cost
```

Comparison of Strategies

In this section, we compare the effectiveness of the overbooking policies.

Strategy 1

When applying strategy 1, we expect every flight to be overbooked.

This does not impact first-class ticket holders. Using this strategy we expect to book just under our capacity for first-class tickets. Additionally, we expect most (97%) of our first-class ticket holders to show up for the flight, meaning that we cannot upgrade overbooked coach passengers to first-class in most cases (Figure 3).

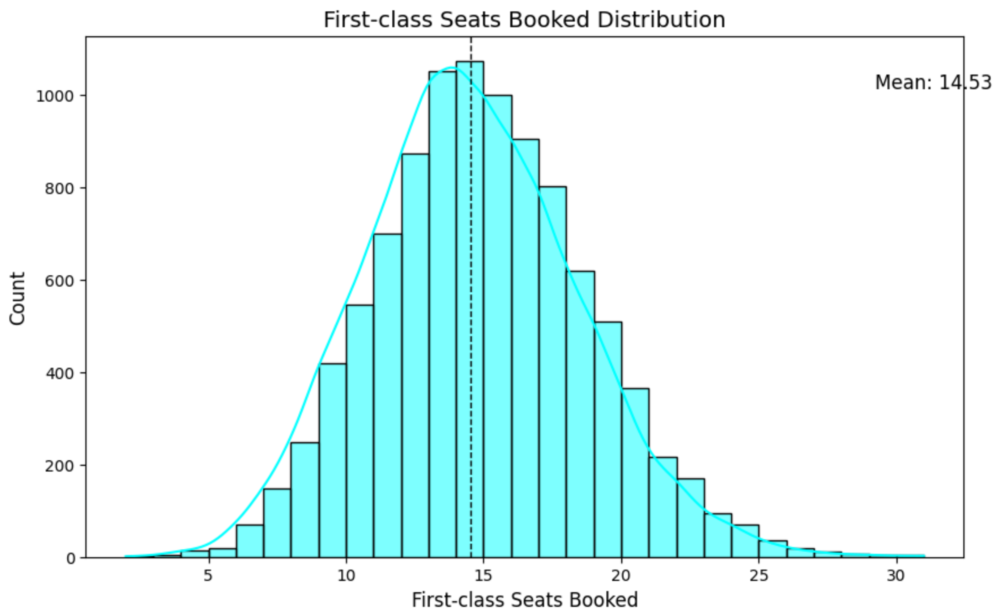

Figure 3. Distribution of the number of first-class seats booked at time of flight.

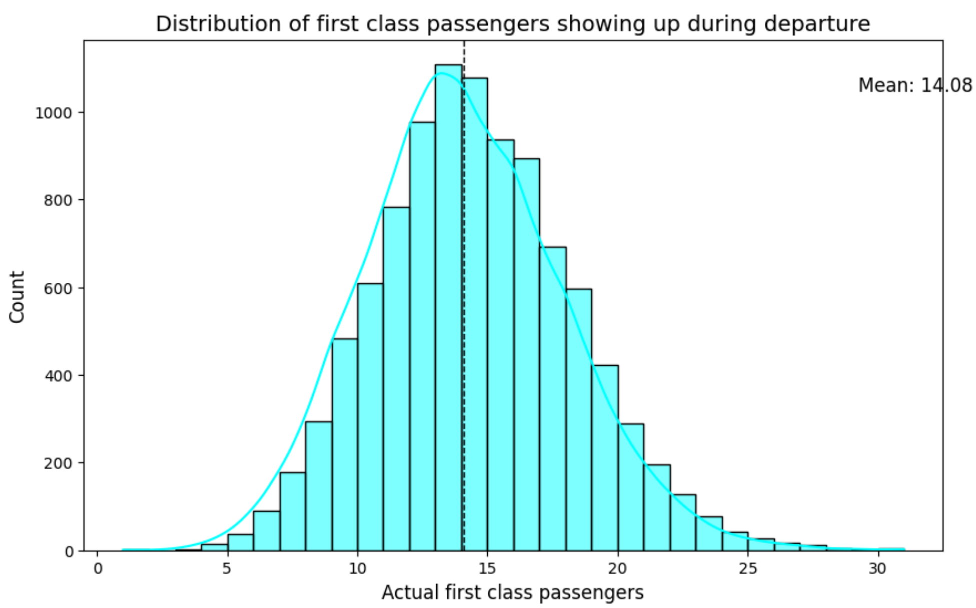

Figure 4. Distribution of first-class ticket holders who show up to the flight.

Given the rules of overbooking, coach passengers present a different story. As seen in Figure 5, the mean number of coach passengers booked is just over 108, which is 8 above the actual capacity of the plane. There is an obvious skew left that suggests a high count of between 108 to 109 seats being sold.

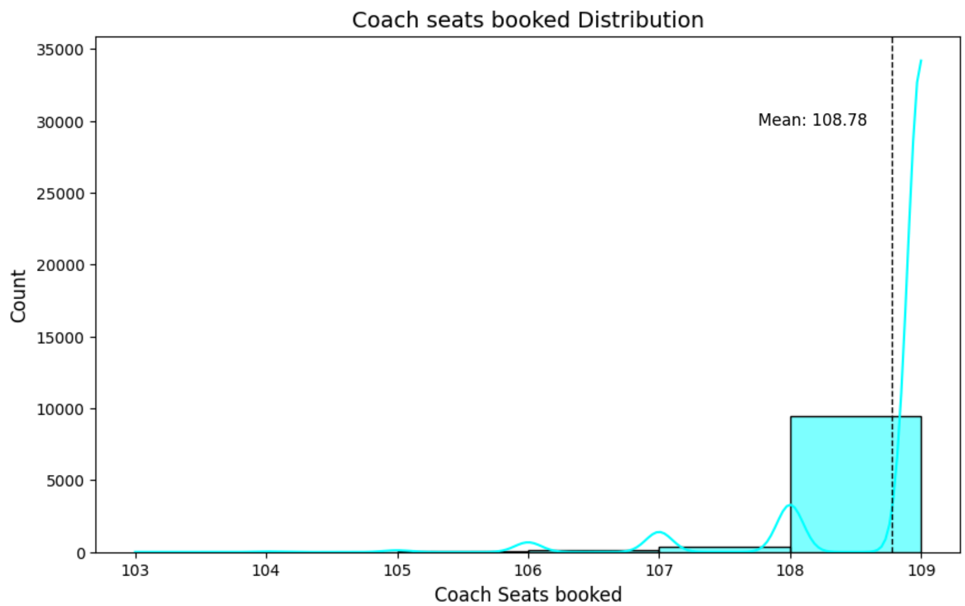

Figure 5. Distribution of the number of coach seats booked at time of flight.

The actual distribution shown in Figure 6 conveys that while we expect to overbook by about 8 seats, we expect around 3 extra coach passengers to actually show up to the flight, which suggests that there *may* be a greater benefit to overbooking than there is risk of cost.

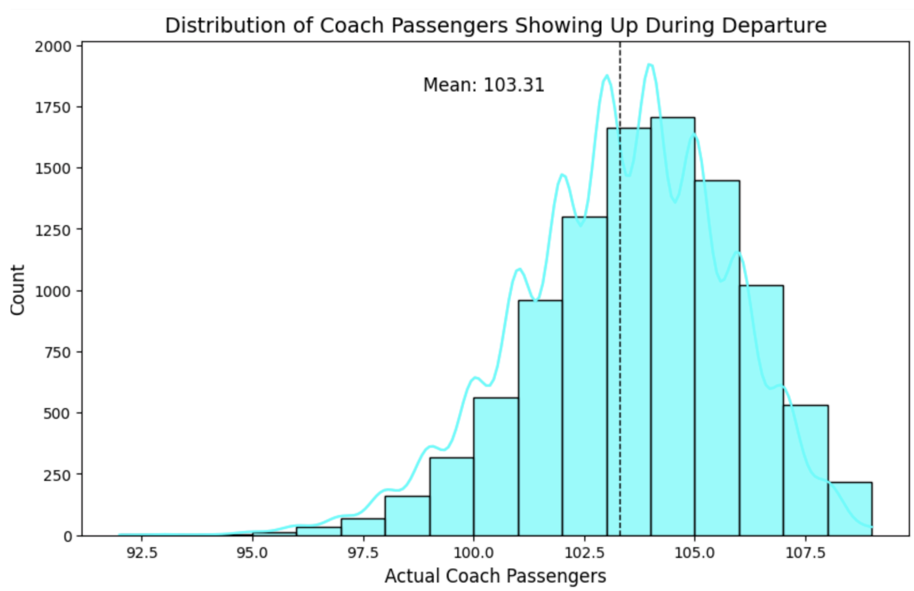

Figure 6. Distribution of coach ticket holders who show up to the flight.

Figure 7 suggests that this strategy of overbooking results in just over 2 coach passengers being kicked off the flight at departure. With every coach passenger that is unable to board the flight, the airline must grant compensation of $425. Applying the number presented by Figure 7, we can expect to pay over $900 in compensation per flight. *(2.17 * $425 = $922.25)*

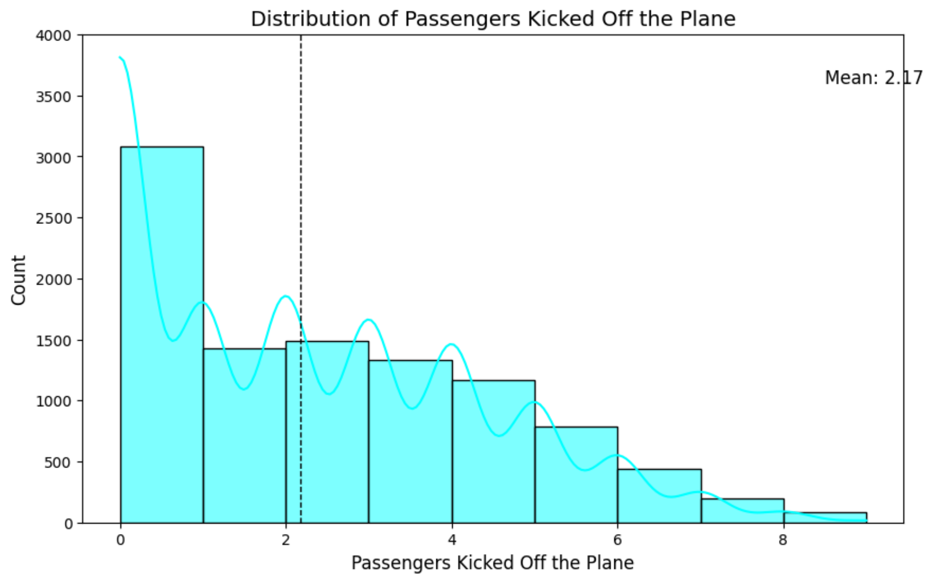

Figure 7. Distribution of coach ticket holders who are bumped from the flight.

Figure 8 confirms this assertion. The mean amount of compensation we should expect to pay is $960.62, and the highest amount of compensation we should expect to pay is just under $4000. While this is certainly a drastic case, we should keep this consideration in mind and perhaps investigate further into these extreme cases.

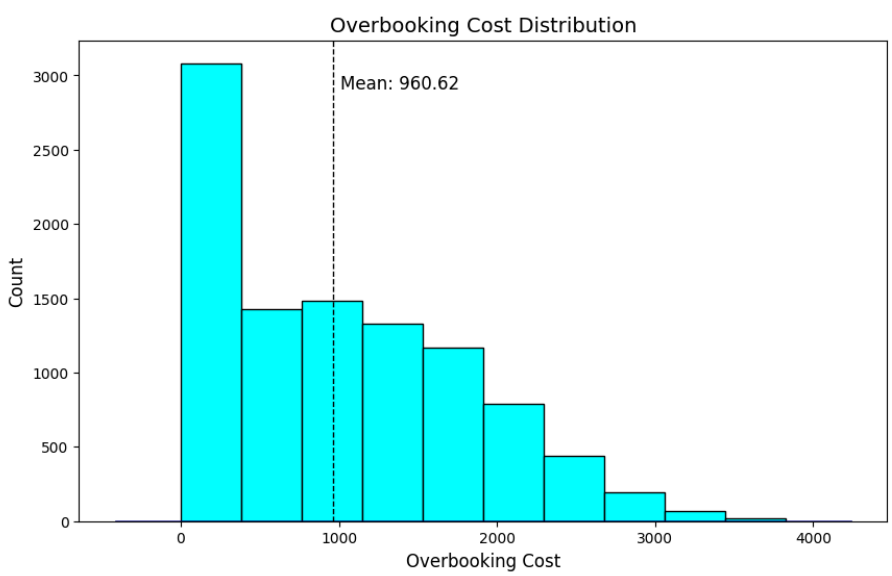

Figure 8. Distribution of overbooking costs per flight.

Interestingly, a deeper dive into the data reveals that passenger removal is not distributed evenly among flights. In fact, out of 10,000 trials, there were 3,082 instances where no passenger was removed from the plane. While it is notable that instances without any passenger removals are apparent, the occurrences where at least one passenger gets removed are even more prevalent, amounting to 6,918. So, while there may not be total consistency among which flight bookings result in passenger bumping, the overwhelming majority include at least one instance of bumping a passenger.

Understanding that 9 seats is the optimal level for overbooking, we should expect to make a mean profit of **$41,076.81**.

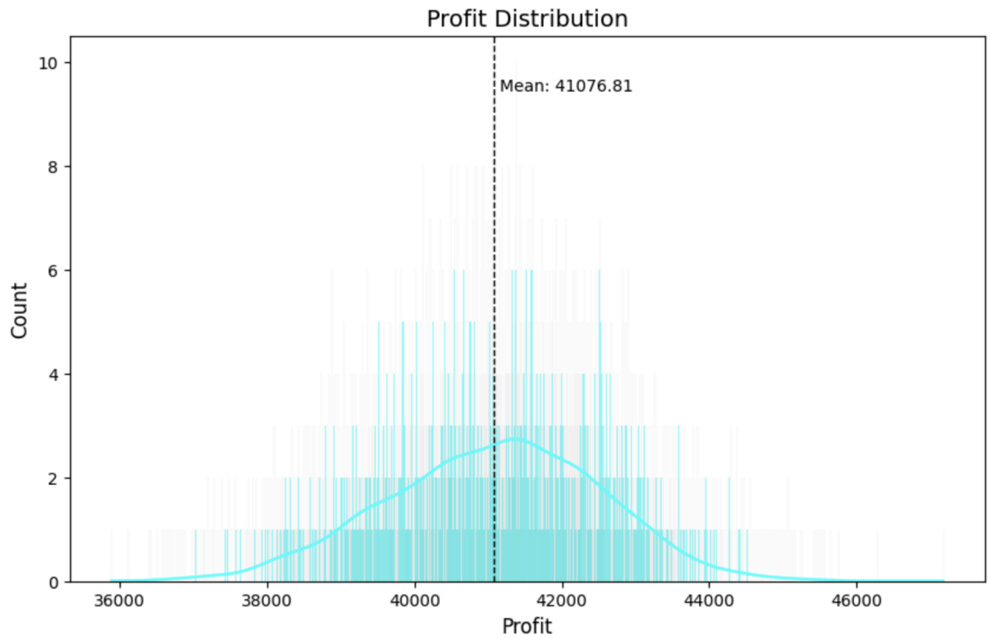

Figure 9. Distribution of expected profits from strategy 1.

Strategy 2

When applying strategy 2, we understand that we have the option to *not* sell coach tickets.

Under this strategy, we see a slightly lower mean number of first-class seats being booked, a comparison of 14.97 for strategy 2 to 19.62 for strategy 1.

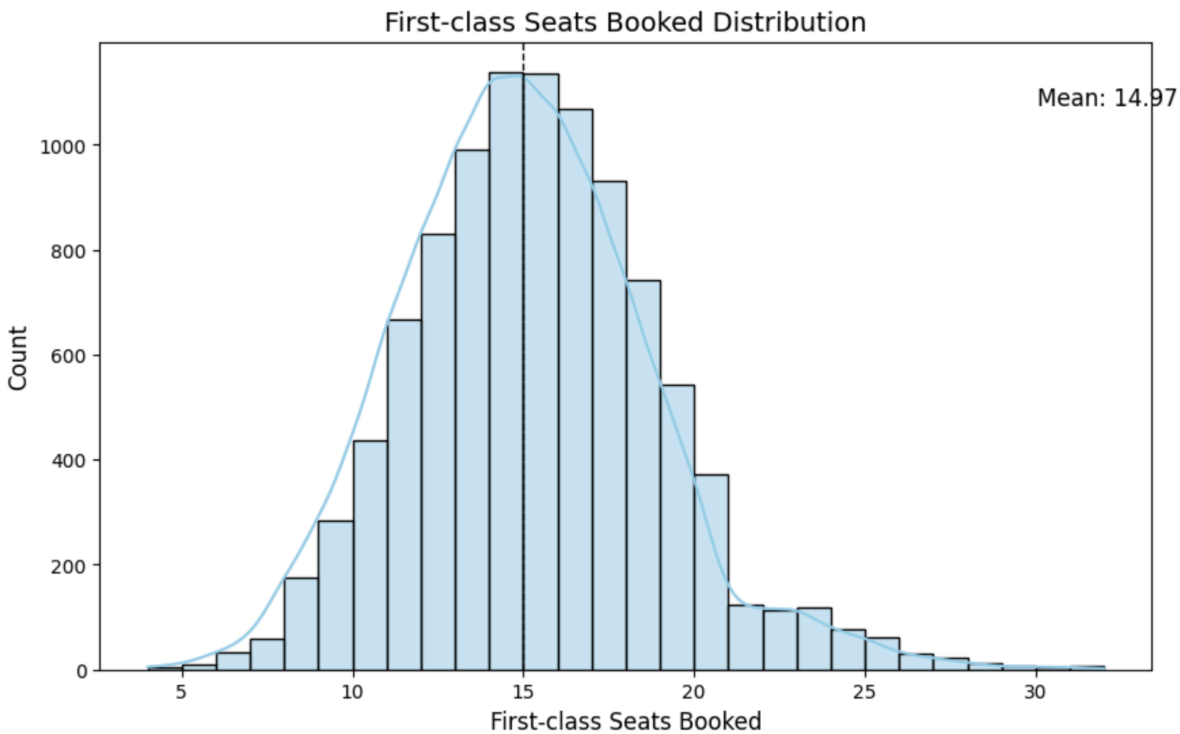

Figure 10. Distribution of first-class seats booked at time of flight.

Similarly, we expect a high ratio of first-class passengers to show up for the flight, a mean comparison of 14.52 for strategy 2 to 14.08 for strategy 1.

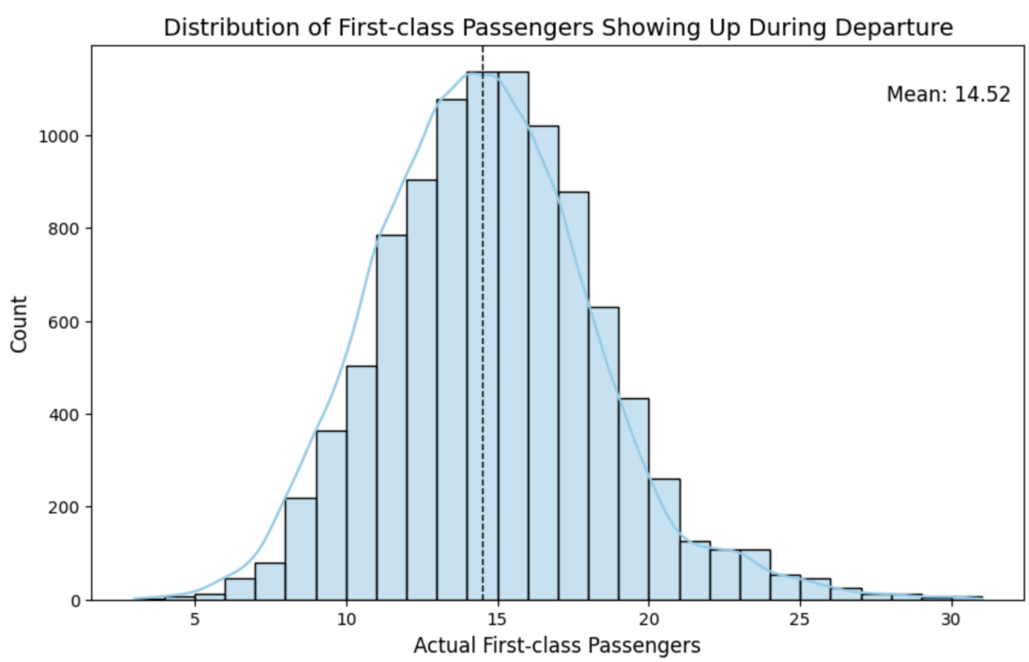

Figure 11. Distribution of first-class ticket holders who show up to the flight.

The mean number of coach seats booked for strategy 2 is slightly higher than strategy 1, suggesting that the optimal booking value truly fell within the range of overbooking address (i.e. overbooking by between 5-15 seats). Strategy 2's distribution of coach seats booked, as shown in Figure 12, displays a wider range of seats booked from 105 to 120, with a notable lift between 109 and 110 seats.

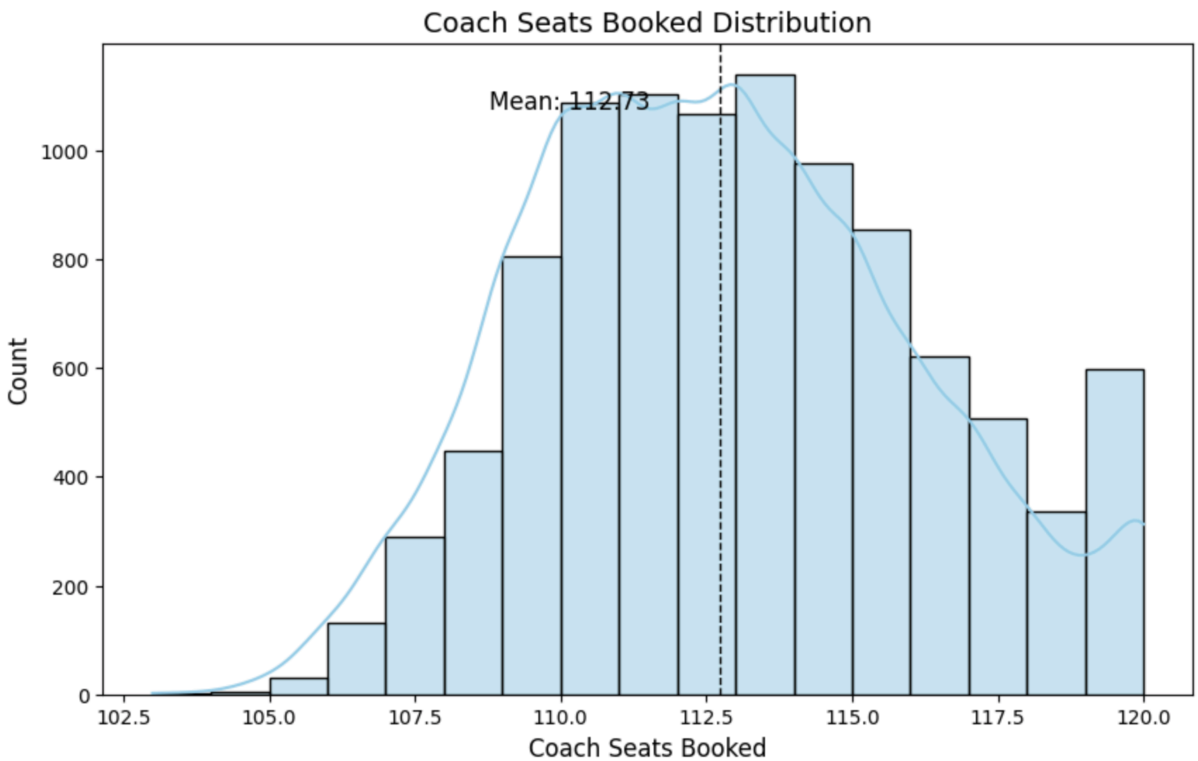

Figure 12. Distribution of coach seats booked at the time of flight.

Again, we see that approximately 6-7 extra passengers will show up to the flight, slightly higher than strategy 1 in figure 13

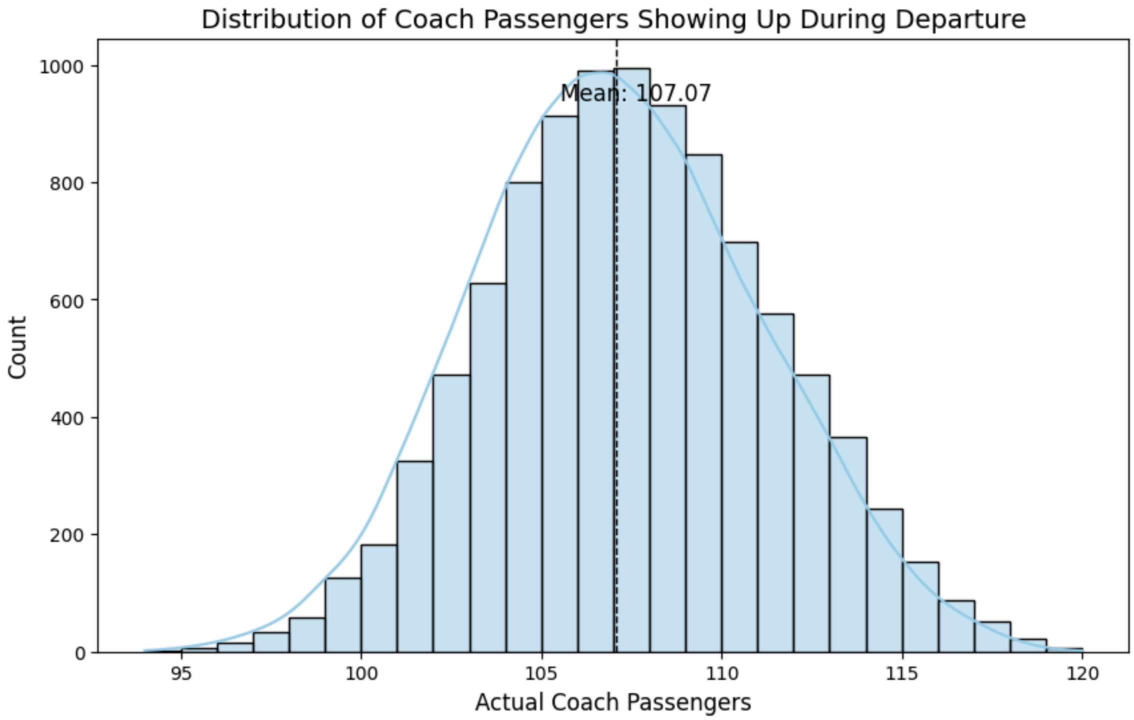

Figure 13. Distribution of coach ticket holders who show up to the flight.

Figure 14 suggests that this strategy of overbooking results in approximately 2 coach passengers being kicked off the flight at departure. Again, the airline must pay $425 in compensation for every passenger that cannot board the flight. Based on figure 14, for strategy 2 we can expect to pay almost $1,000 in compensation per flight. *(2.23 * $425 = 947.75)*

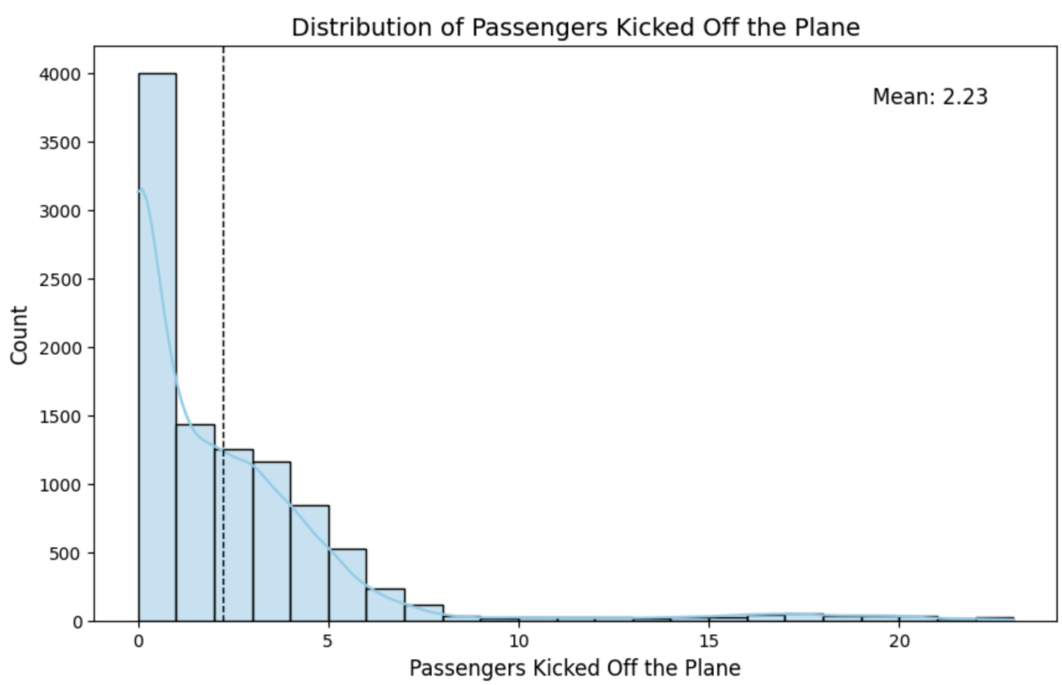

Figure 14. Distribution of coach ticket holders who are bumped from the flight.

Figure 15 actually asserts that we will have to pay more than $1,000 in compensation, with a mean of $1,191.85. The highest amount of compensation we should expect to pay is, again, just under $4,000. This mean payment is 231 higher than that of strategy 1, meaning that on average we would pay approximately an additional $230 per flight in compensation, informing our decision going forward.

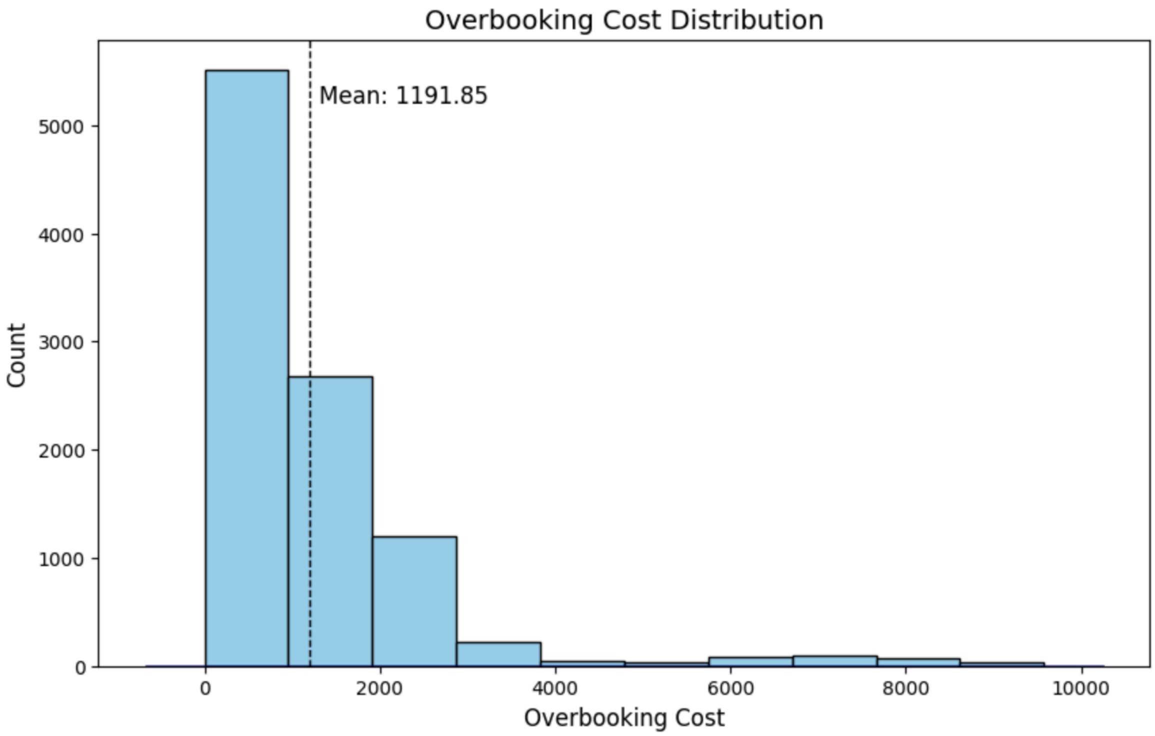

Figure 15. Distribution of overbooking costs per flight.

In strategy 2, we again find that a deeper dive into the data reveals an uneven distribution of passenger bumping among flights. Out of 10,000 trials, there were 3082 instances where no passenger was removed from the plane. Despite the high frequency of these instances, the occasions where at least one passenger was removed are more, amounting to 7085. The overwhelming majority of flights - over 70% - include at least one instance of passenger bumping.

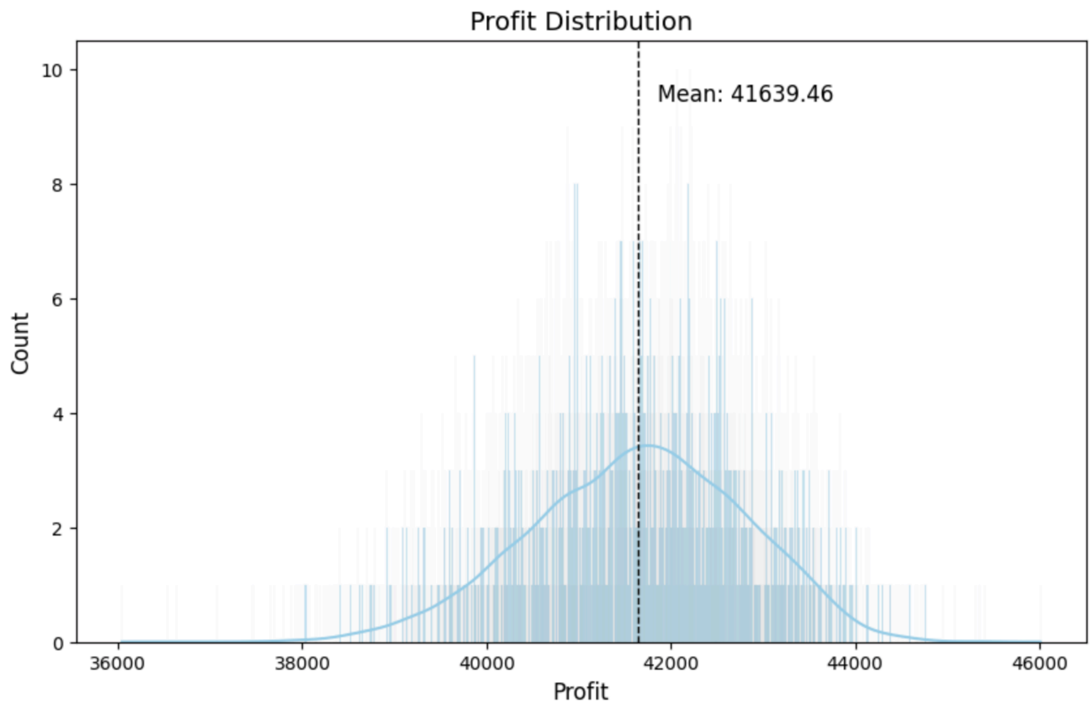

Figure 16. Distribution of expected profits from strategy 2.

Understanding that 20 seats is the optimal level for overbooking for this strategy, we should expect to make a mean profit of **$41,639**.

Summary of Findings

The two strategies have slightly differing outcomes. We can expect a profit of approximately $41,076 from strategy 1, which is slightly lower than the expected profit from strategy 2 of $41,639.

Additionally, we can expect to pay, on average, approximately $230 more in compensation for strategy 2, suggesting that strategy 1 at 9 overbooked seats is the optimal policy, and less number of passengers are being bumped off the plane in strategy 1.

Overall, we should choose strategy 1 because of its relative consistency. While strategy 2 may be projected to make a marginally higher amount of profit, it also experiences a greater degree of variation in the number of coach passengers who are bumped off the plane. This creates a greater amount of variation in the compensation required, making it more difficult to confidently build out a budget per plane as well as potentially eroding more customers' trust in our brand.

It does not appear that the different strategies have much variation in the number of coach ticket holders who show up for the flight, though the variation in the number of passengers bumped from the plane makes a large difference in the amount of compensation owed.

Conclusions

Overall, both overbooking strategies do not have much difference in profits, though strategy 2 has a slightly high expected profit. However, the cost to the airline in terms of compensation is also higher and the number of coach ticket holders being bumped from the flight is greater. .

It is important to recognize that there are two mitigation options when the coach is overbooked: bumping passengers to first-class, if available, and bumping passengers from the plane. Bumping passengers to first class is contingent on there being first-class seats available. Our strategy investigations suggest that this is uncommon, as most first-class ticket seats are already taken. In fact, first-class being sold out increases the probability that coach ticket sales by 3%, due to increased demand for any tickets in light of a shortage, potentially heightening the propensity for our airline to be overbooked and thus being required to pay compensation.

As a general approach, we should be cautious of how often our overbooked customers are bumped from flights. While we do not have data that suggest that this is negatively impacting ticket sales, we may be losing goodwill that will cost us in the future. From the customer's point of view, it seems a bit troublesome if they are being adjusted to other flights because it might clash with their predetermined schedules, assuming in the case of a business person when he or she has back to back meetings in different cities. This may be a good topic for future investigations.