

Problem Statement

Businesses like airlines have a very low profit margin. Help them increase profit by designing an optimal overbooking strategy

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

```
In [2]: id = "1PazlhissU63pozK0jJckyjIJuM_u-0JF"
print("https://drive.google.com/uc?export=download&id=" + id)
```

https://drive.google.com/uc?export=download&id=1PazlhissU63pozK0jJckyjIJuM_u-0JF

```
In [8]: !wget "https://drive.google.com/uc?export=download&id=1PazlhissU63pozK0jJckyjIJuM_u-0JF"
```

```
--2022-05-06 11:19:19-- https://drive.google.com/uc?export=download&id=1PazlhissU63pozK0jJckyjIJuM_u-0JF
Resolving drive.google.com (drive.google.com)... 142.250.196.14, 2404:6800:4007:805::200e
Connecting to drive.google.com (drive.google.com)|142.250.196.14|:443... connected.
HTTP request sent, awaiting response... 303 See Other
Location: https://doc-14-14-docs.googleusercontent.com/docs/securesc/ha0ro937gcuc7l7deffksulhg5h7mbp1/dk7j7j7tnc4j9u0epv5oa2geh9n5bo3k/1651816125000/06496627672658439642/*/1PazlhissU63pozK0jJckyjIJuM_u-0JF?e=download [following]
Warning: wildcards not supported in HTTP.
--2022-05-06 11:19:20-- https://doc-14-14-docs.googleusercontent.com/docs/securesc/ha0ro937gcuc7l7deffksulhg5h7mbp1/dk7j7j7tnc4j9u0epv5oa2geh9n5bo3k/1651816125000/06496627672658439642/*/1PazlhissU63pozK0jJckyjIJuM_u-0JF?e=download
Resolving doc-14-14-docs.googleusercontent.com (doc-14-14-docs.googleusercontent.com)... 142.250.76.33, 2404:6800:4007:814::2001
Connecting to doc-14-14-docs.googleusercontent.com (doc-14-14-docs.googleusercontent.com)|142.250.76.33|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12175 (12K) [text/csv]
Saving to: 'airline.csv'
```

```
airline.csv      100%[=====>]  11.89K  --.-KB/s    in 0.004s
```

```
2022-05-06 11:19:20 (2.75 MB/s) - 'airline.csv' saved [12175/12175]
```

```
In [9]: flights = pd.read_csv('airline.csv')
flights.head()
```

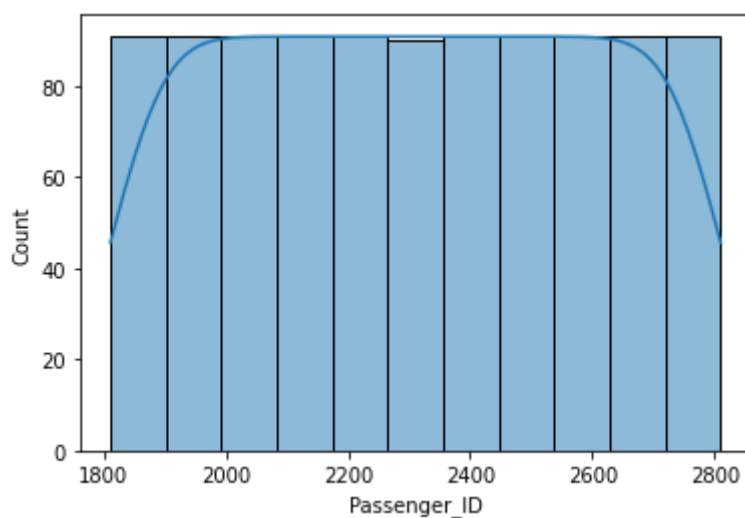
```
Out[9]:
```

	Passenger_ID	Flight_ID	Arrived
0	1811	A320	1
1	1812	A320	1
2	1813	B777	1
3	1814	B737	1
4	1815	B737	1

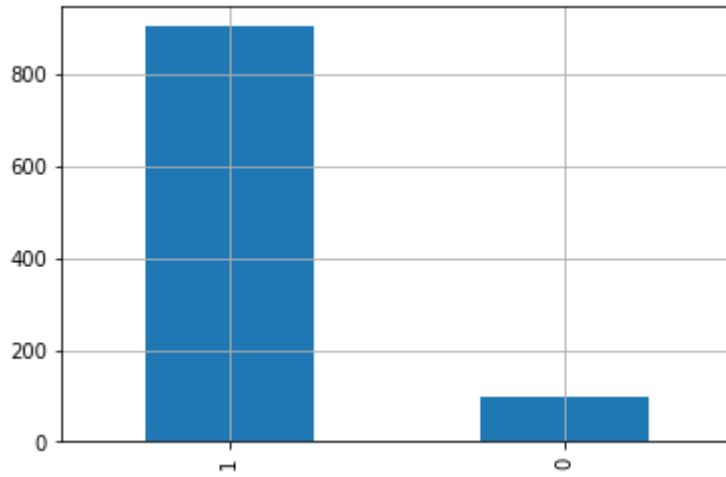
```
In [10]: flights.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1000 entries, 0 to 999  
Data columns (total 3 columns):  
#   Column          Non-Null Count  Dtype  
---  ---  
0   Passenger_ID    1000 non-null   int64  
1   Flight_ID       1000 non-null   object  
2   Arrived         1000 non-null   int64  
dtypes: int64(2), object(1)  
memory usage: 23.6+ KB
```

```
In [11]: sns.histplot(flights["Passenger_ID"], kde=True)  
plt.show()
```



```
In [12]: flights["Arrived"].value_counts().plot.bar()  
plt.grid()  
plt.show()
```



```
In [13]: showsup_probability = flights['Arrived'].value_counts(normalize=True)[1]
print(showsup_probability)
```

0.902

```
In [14]: flights['Arrived'].value_counts()
```

```
Out[14]: 1    902
0     98
Name: Arrived, dtype: int64
```

```
In [59]: import math

PENALTY = 10000

def comb(n, r):
    num1 = math.factorial(n)
    num2 = math.factorial(r)
    num3 = math.factorial(n-r)
    return num1/(num2*num3)
```

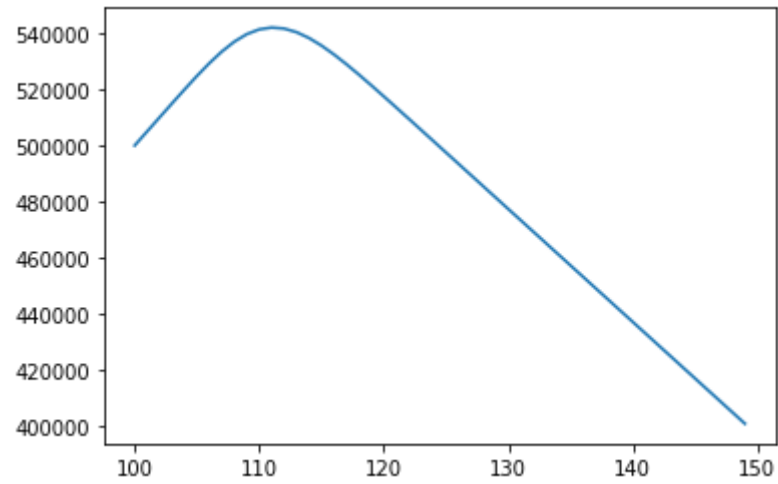
```
In [60]: def calculate_expected_penalty(ticket_sold):
    total_penalty = 0
    for i in range(1, ticket_sold - 100+1):
        ##pmf for k successes, n trials, p=success probab
        prob = stats.binom.pmf(k=100+i, n=ticket_sold, p=showsup_probability)
        penalty = prob * PENALTY * i
        total_penalty += penalty
    return total_penalty
```

```
In [61]: x = []
y = []
for i in range(100, 150):
    sales = 5000 * i
    penalty = calculate_expected_penalty(i)
    netsales = sales - penalty
    x.append(i)
    y.append(netsales)
    print("Total seats:", i, ", Net Sales :", round(netsales))
x[y.index(max(y))]
```

```
Total seats: 100 , Net Sales : 500000
Total seats: 101 , Net Sales : 505000
Total seats: 102 , Net Sales : 509996
Total seats: 103 , Net Sales : 514979
Total seats: 104 , Net Sales : 519913
Total seats: 105 , Net Sales : 524725
Total seats: 106 , Net Sales : 529288
Total seats: 107 , Net Sales : 533425
Total seats: 108 , Net Sales : 536929
Total seats: 109 , Net Sales : 539603
Total seats: 110 , Net Sales : 541302
Total seats: 111 , Net Sales : 541959
Total seats: 112 , Net Sales : 541595
Total seats: 113 , Net Sales : 540305
Total seats: 114 , Net Sales : 538233
Total seats: 115 , Net Sales : 535544
Total seats: 116 , Net Sales : 532393
Total seats: 117 , Net Sales : 528919
Total seats: 118 , Net Sales : 525227
Total seats: 119 , Net Sales : 521398
Total seats: 120 , Net Sales : 517484
Total seats: 121 , Net Sales : 513521
Total seats: 122 , Net Sales : 509531
Total seats: 123 , Net Sales : 505526
Total seats: 124 , Net Sales : 501514
Total seats: 125 , Net Sales : 497497
Total seats: 126 , Net Sales : 493479
Total seats: 127 , Net Sales : 489459
Total seats: 128 , Net Sales : 485440
Total seats: 129 , Net Sales : 481420
Total seats: 130 , Net Sales : 477400
Total seats: 131 , Net Sales : 473380
Total seats: 132 , Net Sales : 469360
Total seats: 133 , Net Sales : 465340
Total seats: 134 , Net Sales : 461320
Total seats: 135 , Net Sales : 457300
Total seats: 136 , Net Sales : 453280
Total seats: 137 , Net Sales : 449260
Total seats: 138 , Net Sales : 445240
Total seats: 139 , Net Sales : 441220
Total seats: 140 , Net Sales : 437200
Total seats: 141 , Net Sales : 433180
Total seats: 142 , Net Sales : 429160
Total seats: 143 , Net Sales : 425140
Total seats: 144 , Net Sales : 421120
Total seats: 145 , Net Sales : 417100
Total seats: 146 , Net Sales : 413080
Total seats: 147 , Net Sales : 409060
Total seats: 148 , Net Sales : 405040
Total seats: 149 , Net Sales : 401020
```

Out[61]: 111

```
In [62]: plt.plot(x, y)
plt.show()
```



In []:

In []: