CLIENT SELECTION
REGENCY BANK

TEAM 4 ICY, NEIL, HAYLEY, FOSTER

01

BUSINESS UNDERSTANDING

BUSINESS UNDERSTANDING

In January 2010, Regency Bank undertook a significant expansion by acquiring a portfolio of 2000 clients, worth $80 million in annual revenues, from Continental Bank.

A key challenge for Regency was client selection from client inheritance. Regency faced the dilemma of **optimizing client selection** to balance **risk**, **profitability**, and its strategic goal of expanding its presence in the corporate card market.

REVENUE BREAKDOWN

CLIENTS

- A total of 210 clients
- 1 - 7 Risk Rating
- 1 - 3 Complexity Level
- An annual spending from $89.62 to $64.73 million
- 1 - 782 cards per client (professional & travel card)

FIXED REVENUE

- A flat rate of $5,000 charged to each client

VARIABLE REVENUE

- 1% of the total annual spend per client

COST BREAKDOWN

FIXED COST

- Migration Cost: $500,000
- Annual Operating Cost: $200,000

COMPLEXITY LEVELS

- **Level 1:** Migration $2,000, Annual Operation $1,500
- **Level 2**: Migration $ 5,000, Annual Operation $ 2,000
- **Level 3**: Migration $7,000, Annual Operation $ 3,000

VARIABLE COST

Upon Action

- Migration Cost: per account according to complexity level
- Card Issuance: $45 per card

Annual

- Annual Operation Cost: per account according to complexity level
- Annual Card Service: $40 per card

RISK & GROWTH

DEFAULT & ATTRITION RISK

- Different risk ratings with varying probabilities of default and associated costs
- Natural Attrition Rate: 10% per annum

LEVEL	1	2	3	4	5	6	7
RISK OF DEFAULT	0.1%	0.5%	1%	2%	3%	5%	10%
# OF MONTHS	3	3	3	3	3	4	6

SPENDING AND CARD GROWTH

- Annual spending and the number of cards expected to grow at a mean rate of 8% and 10% per annum, respectively, with a standard deviation of 1%.

OBJECTIVE

Accounting the **risk with client attrition and default** and the **fluctuation in the growth factors**, Regency Bank aims to evaluate the profit by assessing the **revenue** and **cost** associated with the client selection decisions

02

QUESTION

A

Describe a simulation model to evaluate the attractiveness of one potential client over the first three years.

TIMELINE FOR ONE CLIENT

attrition?　attrition?

Year 1 → Year 2 → Year 3

Year 1

Revenue
1% total annual spend, plus a flat rate of $5,000

Cost

- Cost of issuing new cards to all potential clients
- Cost of cards service
- Migration cost ($2000/$5000/$7000)
- Operation cost ($1500/$2000/$3000)
- Risk of default (risk * (months / 12) * annual_spend)

Year 2

Revenue
1% total annual spend, plus a flat rate of $5,000

Cost

- Cost of issuing new cards to new clients
- Cost of cards service
- Operation cost ($1500/$2000/$3000)
- Risk of default (risk * (months / 12) * annual_spend)

Year 3

Revenue
1% total annual spend, plus a flat rate of $5,000

Cost

- Cost of issuing new cards to new clients
- Cost of cards service
- Operation cost ($1500/$2000/$3000)
- Risk of default (risk * (months / 12) * annual_spend)

For Question A(1 client), we only considered the variable costs

Starting from the second year, the model need to determine if the client has experienced attrition first, if yes, the profit of the client will be counted as a zero for that year and years followed.

SIMULATION MODEL

```
attrition_factor == 0:
 # Growth rates with fluctuations ---------------------------------------------------
 annual_spend_growth_rate = np.random.normal(0.08, 0.01)
 spend *= (1 + annual_spend_growth_rate)

 # Growth rates with fluctuations ---------------------------------------------------
 card_growth_rate = np.random.normal(0.10, 0.01)
 total_card_issue_cost += card_issue_cost * ncards * card_growth_rate #-------------
 ncards = ncards  * (1 + card_growth_rate)

 attrition_prob = 0.10  # 10% probability of attrition ------------------------------
 attrition_factor = np.random.choice([0, 1], p=[1 - attrition_prob, attrition_prob])
```

- The fluctuations of the growth in the # of cards and spending of the client is on **client-year level**, and follow a **normal distribution** with mean of 0.1 and 0.08 respectively with a standard deviation of 0.01
- The attrition rate is 10%, meaning the clients in the client list have a 10% probability of terminating their service with the bank, with no return probability assumed

All of the previously reviewed constraints were taken into consideration, in addition to an attrition rate of 10%, and random variables for the annual spend growth rate and card growth rate. We looped through three years of possibility and simulated 1000 times to find what we could expect from customer value

EXPECTED VALUES

```
# Risk of default -----------------------------------------------
risk_data = {
    'Rating': [7, 6, 5, 4, 3, 2, 1],
    'Risk of Default': [0.10, 0.05, 0.03, 0.02, 0.01, 0.005, 0.001],
    'Number of months': [6, 4, 3, 3, 3, 3, 3]
}
```

- The risk level according to the client risk rating, and the fraction of the year subject to default is shown accordingly (measured in number of month)
- The function here calculates the expected value of the loss of revenue (default_cost) from the clients

```
for rating, risk_prob, months in zip(risk_data['Rating'], risk_data['Risk of Default'], risk_data['Number of months']):
    if client_data['Risk Rating'] == rating:
        default_cost += risk_prob * (months / 12) * spend
        break
```

The default rate according to each client's rating in incorporated in this step. We used expected values for the loss of revenue because each of the revenue is associated with a fixed probability. By applying the expected value calculations, we can determine the utility clients with different risk ratings bring to the overall selection.

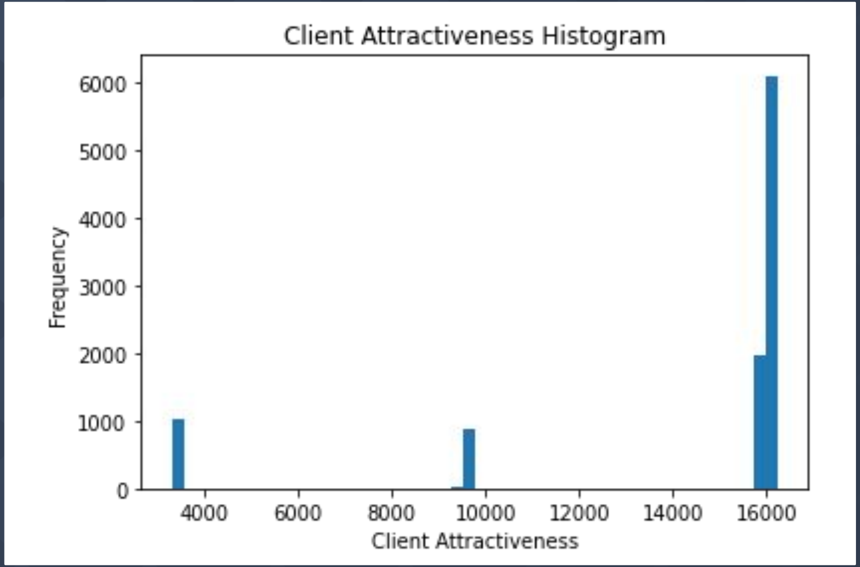
MODEL TEST – ONLY ONE CLIENT

As you can see, the resulting distribution of attractiveness is heavily skewed to the left. Our metric of client attractiveness most often fell into a range between \$15,000-\$16,000, as the most frequent scenario is when the client still in the client list after three years.

There are some but very rare scenarios where this client do attrite in the three years. Therefore, there are some lower numbers around \$3,500 and \$9,000 that observe some moderate peaks in value

10,000 Simulations Result:

For client 1, client attractiveness distribution



03

Question B

Assume that Regency Bank will migrate all of the potential clients. Build a three-year simulation model to evaluate this decision.

MIGRATING ALL CLIENTS

We are able to obtain the expected profit of migrating all the existing clients to Regency Bank by adding up the profits' results of Question A and minus the fixed cost

The simulation results showed that the profit for Regency Bank over 3 years are likely to be around $10,000,000

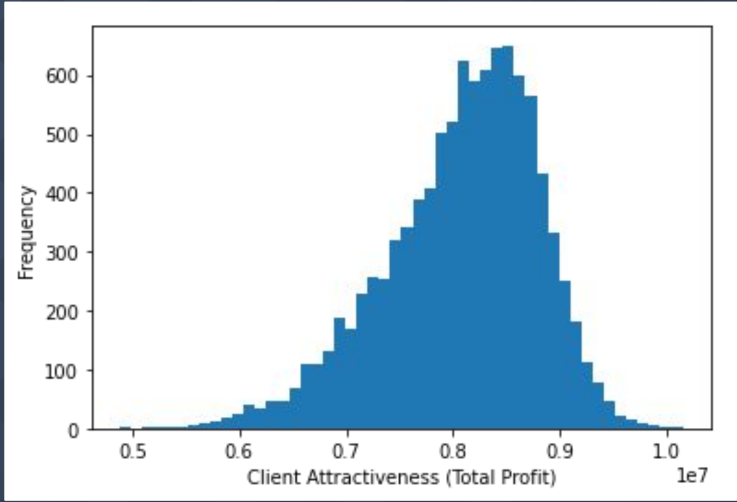
The variation is mostly likely to be caused by the randomness in clients attrition,
Clients are different in

- spending volume
- complexity level
- risk level

Spending volume will affect revenue directly, complexity level and risk level will affect cost directly

10,000 Simulations Result:

For all clients, client attractiveness distribution



04

Question

C

If Regency Bank were to only accept customers of a particular risk level (or better), what is the best risk level to use for the migration decision?

CLIENT SELECTION BASED ON RISK LEVEL

The Regency Bank has a more conservative preference for clients when it comes to client risk ratings. The bank would normally accept only those accounts with a maximum risk rating of 5. We are investigating if raising or lowering the risk tolerance would generate a better profit

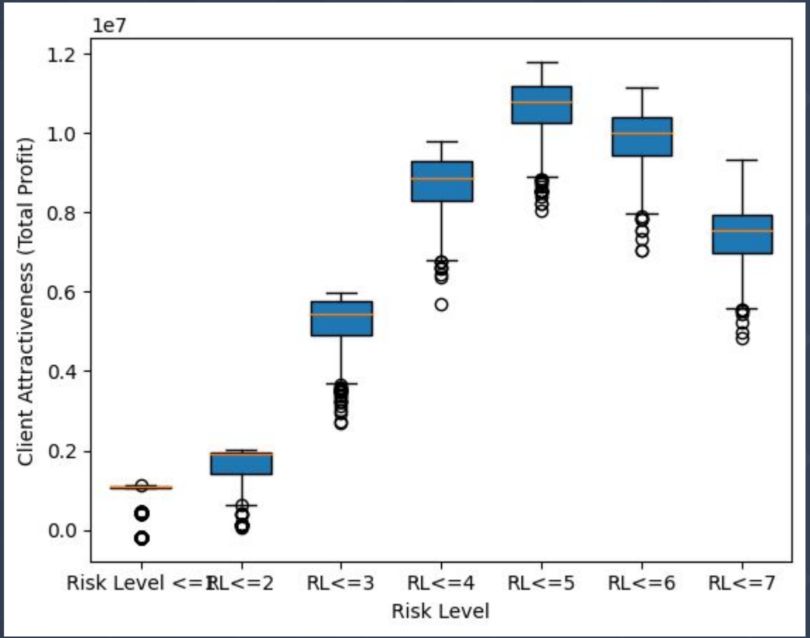
```python
for risk_level in range(1, 8):

        def simulate_all_clients_byRisk(client_data,rl):
            total_attractiveness = 0

            for _, client_data in data.iterrows():
                if client_data['Risk Rating'] <= rl:
                    total_attractiveness += simulate_client(client_data)
            return total_attractiveness

        # run simulations and plot the profit
        client_attractiveness_list_c = []
        for _ in range(10000):
            client_attractiveness_list_c.append(simulate_all_clients_byRisk(data,risk_level) - (migration_cost_fixed + operation_cost_fixed))

        #print sumaary statistics of client_attractiveness_list_c
        print('Summary Statistics of Client Attractiveness for Risk Level ' + str(risk_level))
        print('Mean: ' + str(np.mean(client_attractiveness_list_c)))
        print('Standard Deviation: ' + str(np.std(client_attractiveness_list_c)))
        print('Minimum: ' + str(np.min(client_attractiveness_list_c)))
        print('Maximum: ' + str(np.max(client_attractiveness_list_c)))
        print('Median: ' + str(np.median(client_attractiveness_list_c)))

        #plot boxplot
        #don't plot in new axis but add the new box below in the same scale: ax
        plt.boxplot(client_attractiveness_list_c, positions=[risk_level], widths=0.6, patch_artist=True)

plt.xticks(range(1, 8), ['Risk Level <=1', 'RL<=2', 'RL<=3', 'RL<=4', 'RL<=5', 'RL<=6', 'RL<=7'])
plt.ylabel('Client Attractiveness (Total Profit)')
plt.xlabel('Risk Level')
```

We used a for loop to run our function 6 times, one for each risk level within our constraints.

For 10,000 iterations, we store each client who falls within the risk level filter for the current iteration of the loop and simulate them based on their data.

Then, we plot the results of the simulations in a histogram, creating a graph for each risk level analyzed.

CLIENT ATTRACTIVENESS BY RISK LEVEL



As we increase the threshold for the client Risk Level, we see the client attractiveness increase, but the standard deviation of client attractiveness increase with it. This increased spread indicates a greater risk. However, with the increase in Client Attractiveness being so great, it is still a net benefit to increase the threshold.

According to the distribution of the final profit of each risk level, we can tell that **Risk Level 5 generates the best profit outcome and has a reasonable level of variation**

CLIENT SELECTION

Given that the most favorable risk level threshold is **Risk Level 5**, it means a total of **189 clients** will be remained in the client lists after applying the criteria, so in total 21 clients are excluded from the client list as a result.

Profit mean at Risk Level 7 (clients of all risk levels included): $7,364,814

Profit mean at Risk Level 5: $10,627,779

Therefore, having a risk level threshold at level 5 would improve the profit by **$3,262,965**

```
def simulate_all_clients_byRisk(client_data, rl):
    total_attractiveness = 0
    for _, client in client_data.iterrows():
        if client['Risk Rating'] <= rl:
            total_attractiveness += simulate_client(client)
    return total_attractiveness

def calculate_mean_attractiveness(data, risk_level):
    client_attractiveness_list = []
    for _ in range(1000):
        client_attractiveness_list.append(simulate_all_clients_byRisk(data, risk_level))

    return np.mean(client_attractiveness_list)# Calculate mean client attractiveness at risk level 7
mean_attractiveness_level_7 = calculate_mean_attractiveness(data, 7)
print('Mean Client Attractiveness at Risk Level 7:', mean_attractiveness_level_7)

# Calculate mean client attractiveness at risk level 5
mean_attractiveness_level_5 = calculate_mean_attractiveness(data, 5)
print('Mean Client Attractiveness at Risk Level 5:', mean_attractiveness_level_5)
```

05

Question

D

What is
the best migration strategy for the bank?

ITERATION MODEL DESIGN

Complexity Level

Higher complexity level means higher variable costs in operation, thus we coded the filter to be client['Complexity Level'] <= complexity_level_threshold ,and rerun the function to test different threshold of a range of 1 to 3

Risk Rating

Higher risk rating means higher probability of default. client['Risk Rating'] <= risk_level_threshold is to test different threshold values from 1-5

Annual Spend Volume

Higher annual spend will bring Regency more annual income, thus we coded the filter as client['Annual Spend Volume'] >= size_threshold , and created 5 thresholds between min and max annual spend

of Cards

Because we want to use number of cards to represent the size of client, we coded the filter as client['# of Cards'] >= num_cards_threshold, and created 5 thresholds between min and max number of cards

Size

MODELING STRATEGY

CALCULATION

Define a function to simulate profits given the 4 thresholds

RANGE

Define the whole possible range of value that each threshold can take

ITERATE OVER

Iterate over all possibilities calculating both mean and spread then sort to find best

MODEL **RESULTS**

Best solution: Risk Level <=5, Complexity Level no constraint, Spending Size >= 89.62, Num of Card no constraint

The above mix of criteria generates the best outcome in terms of profit value, which has a mean of **$10.6 million,** serving 189 clients and a reasonable standard deviation of $0.68 million (which can't be significantly reduced until the 30th high mean profit), outperforming all other combinations

This strategy improves the average total profit by **$3.27 million** compared to no selection criteria

1	Risk Level	Complexity Level	Size Threshold	Num Card Threshold	Mean	Standard Deviation	Minimum	Maximum	Mean # Client
2	5	3	89.62	1	10631246.66	680959.2031	7746004	11736862	189
3	6	3	89.62	1	9852006.73	671617.0963	7573590	10993307	200
4	4	3	89.62	1	8710673.54	682584.2651	5793242	9800645	88
5	7	3	89.62	1	7386849.529	731487.5279	4631107	9103499	210
6	5	2	89.62	1	6185522.73	509422.3355	4355402	6950487	143
7	6	3	16183652.85	1	5802724.07	645011.5531	3227511	6578978	10
8	6	2	89.62	1	5799270.263	515858.1927	3872410	6582622	151
9	5	3	16183652.85	1	5770055.699	636594.0706	2959945	6591737	10
10	7	3	16183652.85	1	5749417.926	674313.3173	3045255	6576713	10
11	4	3	16183652.85	1	5361191.916	593944.1041	2909499	6072940	8
12	3	3	89.62	1	5269159.955	603165.4416	2725396	5996978	27
13	4	2	89.62	1	5194206.806	492560.8769	3344644	5862362	65
14	7	2	89.62	1	4922404.237	513634.7496	2858265	5979957	156
15	3	3	16183652.85	1	3601266.668	612020.9325	1212383	4186135	4

THANK YOU

Appendix - Question D Code

```
import itertools
from tqdm import tqdm

# Assuming you have defined simulate_client and have client_data loaded

def simulate_policy(data, risk_level_threshold, complexity_level_threshold, size_threshold, num_cards_threshold):
    total_profit = 0
    number_of_migrated_clients = 0
    for _, client in data.iterrows():
        if (client['Risk Rating'] <= risk_level_threshold and
            client['Complexity Level'] <= complexity_level_threshold and
            client['Annual Spend Volume'] >= size_threshold and
            client['# of Cards'] >= num_cards_threshold):
                client_profit = simulate_client(client)
                total_profit += client_profit
                number_of_migrated_clients += 1
    total_profit -= (migration_cost_fixed + operation_cost_fixed)
    return total_profit, number_of_migrated_clients

# Define ranges for risk levels, complexity levels, and size thresholds
risk_levels = range(1, 8)
complexity_levels = range(1, 4)  # Example: 1-3
size_thresholds = np.linspace(max(data['Annual Spend Volume']),
                              min(data['Annual Spend Volume']),
                              num=5)  # Example: Create 5 thresholds between min and max annual spend
num_cards_threshold  = np.linspace(max(data['# of Cards']),
                                   min(data['# of Cards']),
                                   num=5) # Example: Create 5 thresholds between min and max number of cards

# create a table to store these 3 variables and the mean, sd, min, max, median of results
client_attractiveness_list_d = pd.DataFrame(columns=['Risk Level', 'Complexity Level', 'Size Threshold', 'Num Card Threshold','Mean', 'Standard Deviation', 'Minimum', 'Maximum', 'Median'])

for risk_level, complexity_level, size_threshold, num_cards_threshold in tqdm(itertools.product(risk_levels, complexity_levels, size_thresholds, num_cards_threshold)):
    profit = []
    numClient = []
    for _ in range(1000):
        tmp = simulate_policy(data, risk_level, complexity_level, size_threshold, num_cards_threshold)
        profit.append(tmp[0])
        numClient.append(tmp[1])
    client_attractiveness_list_d = pd.concat([client_attractiveness_list_d, pd.DataFrame({'Risk Level': risk_level, 'Complexity Level': complexity_level, 'Size Threshold': size_threshold,'Num Card Threshol
                                             'Mean': np.mean(profit), 'Standard Deviation': np.std(profit), 'Minimum': np.min(profit), 'Maximum': np.max(profit), 'Median': np.median(profit), \
                                            'Mean # Client': np.mean(numClient), 'Standard Deviation # Client': np.std(numClient), 'Minimum # Client': np.min(numClient), 'Maximum # Client': np.max(numClient),
```