Q1

```{r}
test_df = read.csv("RFM_data.csv")
```

```{r}
percentage_used_offer <- 100 * mean(test_df$offer_used)
total_spending_used_offer <- sum(test_df$normal_paid_price[test_df$offer_used == 1])

cat("Percentage of customers who used the offer:", percentage_used_offer, "%\n")
cat("Total spending by customers who used the offer: $", total_spending_used_offer, "\n")
```

```
Percentage of customers who used the offer: 9.622857 %
Total spending by customers who used the offer: $ 201333
```

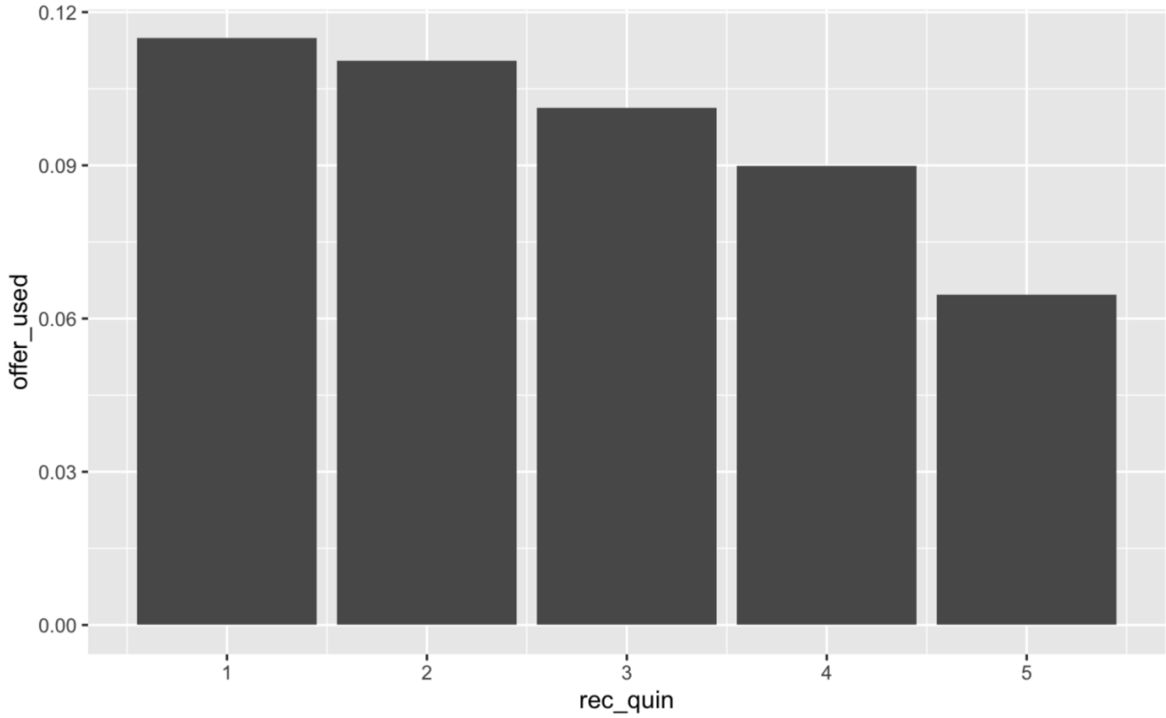
Q2

```{r}
test_df$rec_quin <- ntile(test_df$recency, 5)
test_df$freq_quin <- 6 - ntile(test_df$frequency, 5)
test_df$m_quin <- 6 - ntile(test_df$monetary, 5)

test_df$rfmindex_iq <- 100*test_df$rec_quin + 10*test_df$freq_quin + test_df$m_quin
```

```{r}
library (dplyr)
ggplot(test_df) + stat_summary(aes(x =
                                     rec_quin, y = offer_used), fun = "mean",
                               geom = "bar")

ggplot(test_df) + stat_summary(aes(x =
                                     freq_quin, y = offer_used), fun = "mean",
                               geom = "bar")

ggplot(test_df) + stat_summary(aes(x =
                                     m_quin, y = offer_used), fun = "mean",
                               geom = "bar")
```

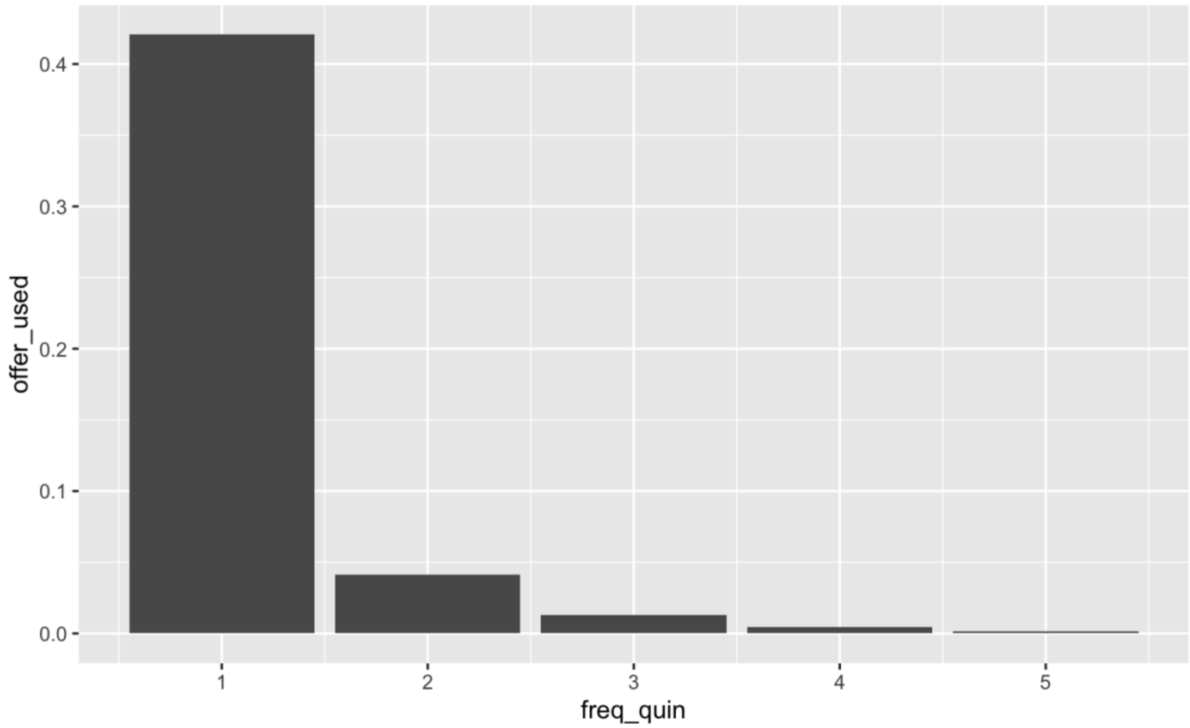
	user_id	recency	frequency	monetary	offer_used	normal_paid_price	rec_quin	freq_quin	m_quin	rfmindex_iq
1	1	43	49	954.52	0	0.00000	5	2	2	522
2	2	16	37	500.79	0	0.00000	2	4	4	244
3	3	11	59	722.28	0	0.00000	1	1	3	113
4	4	25	53	1137.38	0	0.00000	3	2	2	322
5	5	17	44	876.49	0	0.00000	2	3	2	232
6	6	10	48	722.51	0	0.00000	1	2	3	123
7	7	12	44	580.34	0	0.00000	2	3	4	234
8	8	53	32	661.55	0	0.00000	5	4	3	543
9	9	27	32	523.22	0	0.00000	4	4	4	444
10	10	33	37	414.87	0	0.00000	4	4	5	445
11	11	13	58	626.83	0	0.00000	2	1	4	214
12	12	28	51	438.28	0	0.00000	4	2	5	425
13	13	30	64	1290.00	0	0.00000	4	1	1	411
14	14	6	24	579.18	0	0.00000	1	5	4	154
15	15	26	29	566.54	0	0.00000	4	5	4	454
16	16	18	101	1428.12	1	78.02790	3	1	1	311

howing 1 to 15 of 35,000 entries, 13 total columns

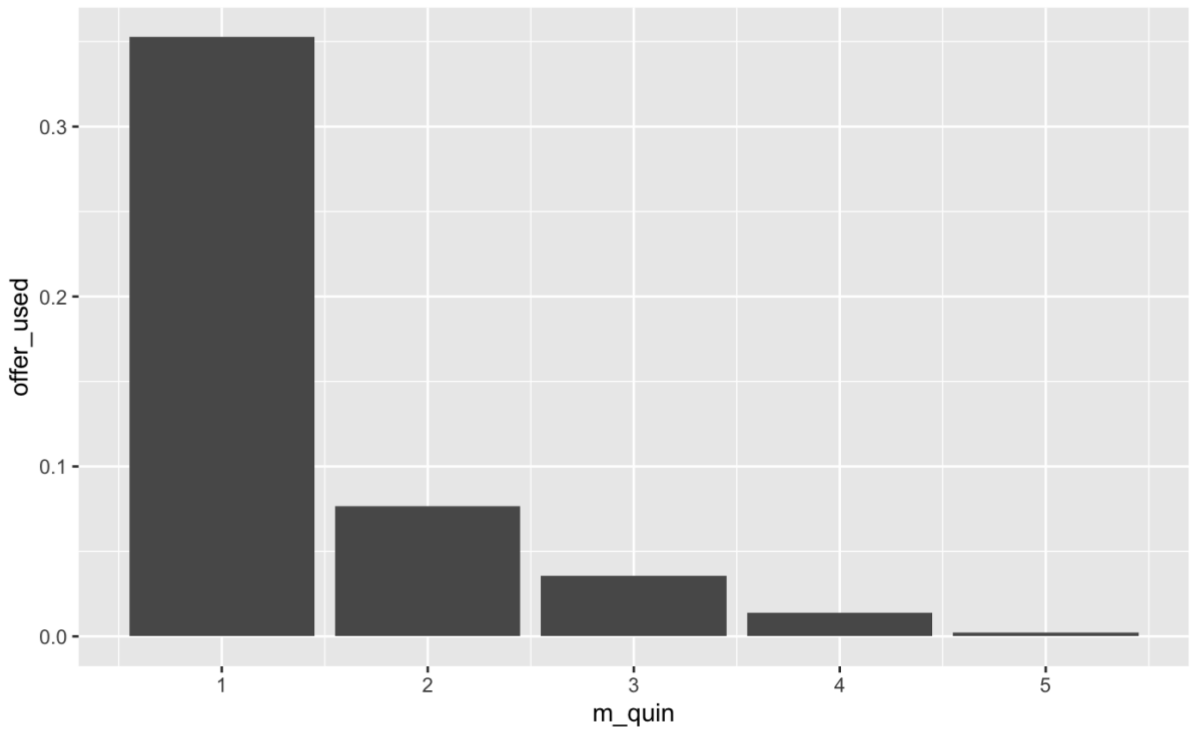


Customers are assigned RFM scores using a 5-quantile method. A score of 1 in R, F, or M indicates the best case (e.g., recent purchase, highest frequency, or highest monetary value), while 5 indicates the worst case. This creates 125 unique combinations.

Customers with a R score of 1 have the highest offer used rate, which is expected as these customers have purchased more recently. We observe a general incremental trend in the likelihood of the offer acceptance as R score goes from 5 to 1. The rise is rather steady and we don't see any unexpected or surprising findings.



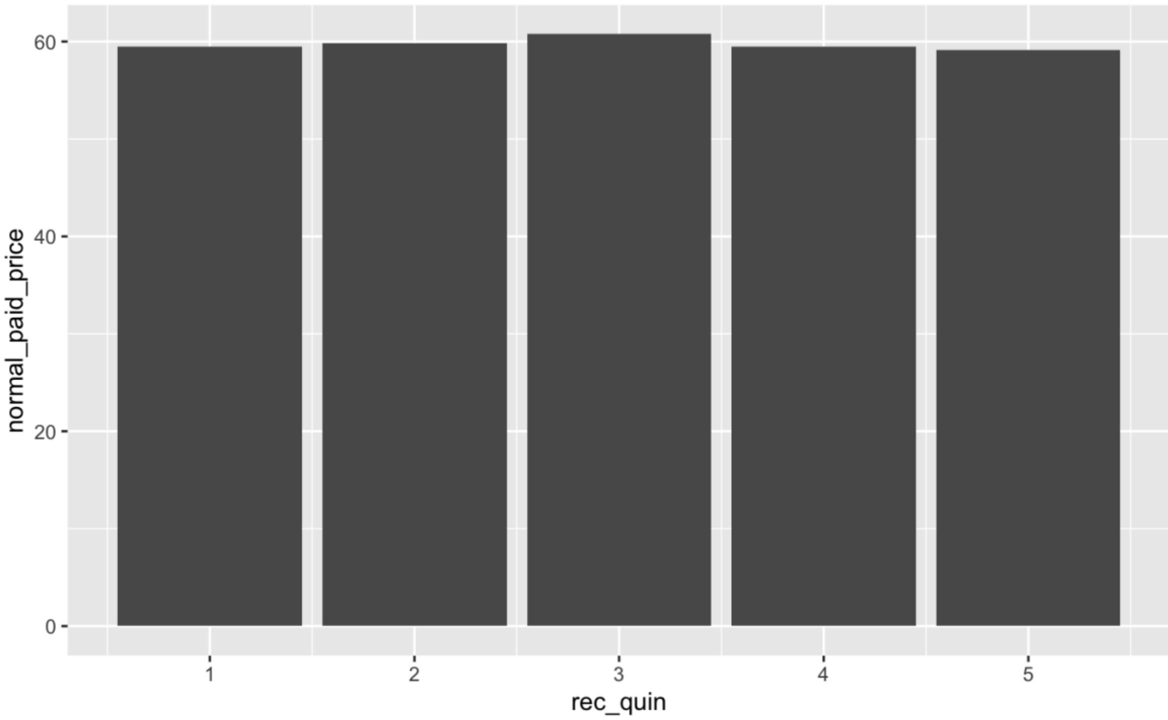
Customers with the best-case frequency, which is 1, have the highest chances of redeeming the offer. These are the most engaged customers, and the graph shows that it could be very profitable if these users are targeted. Moreover, we notice a steep decline in the probability of offer acceptance as F goes from 1 to 5.

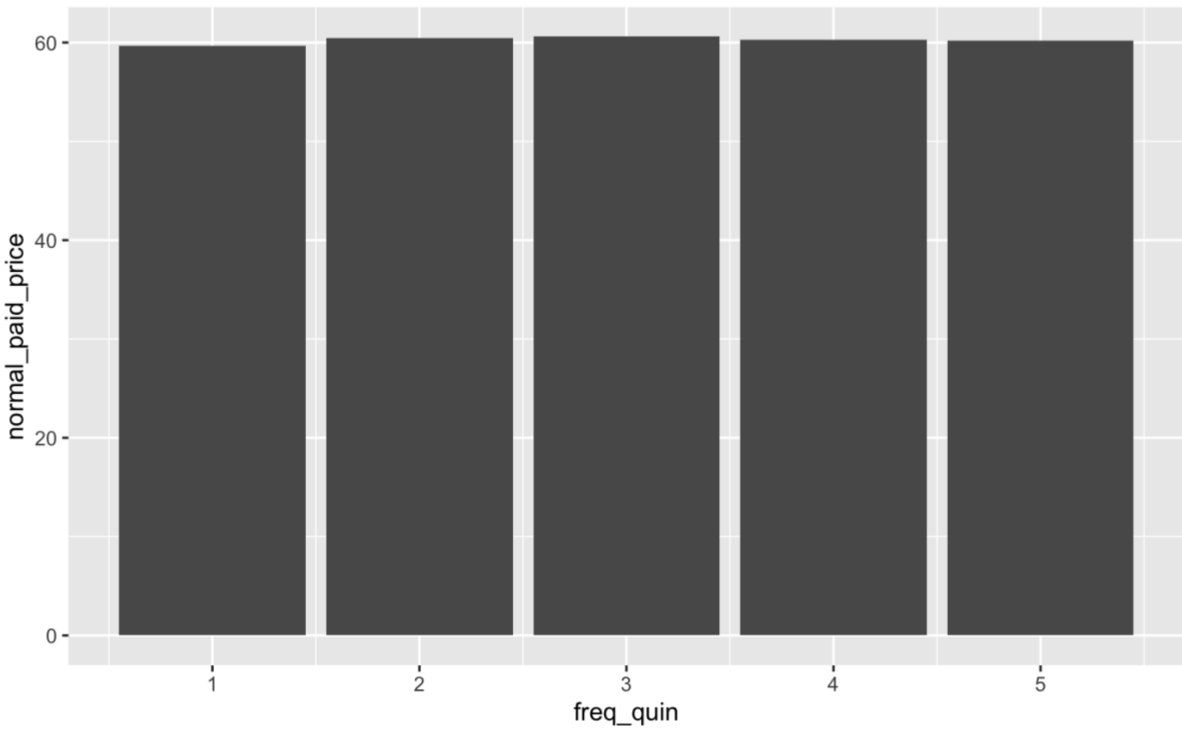


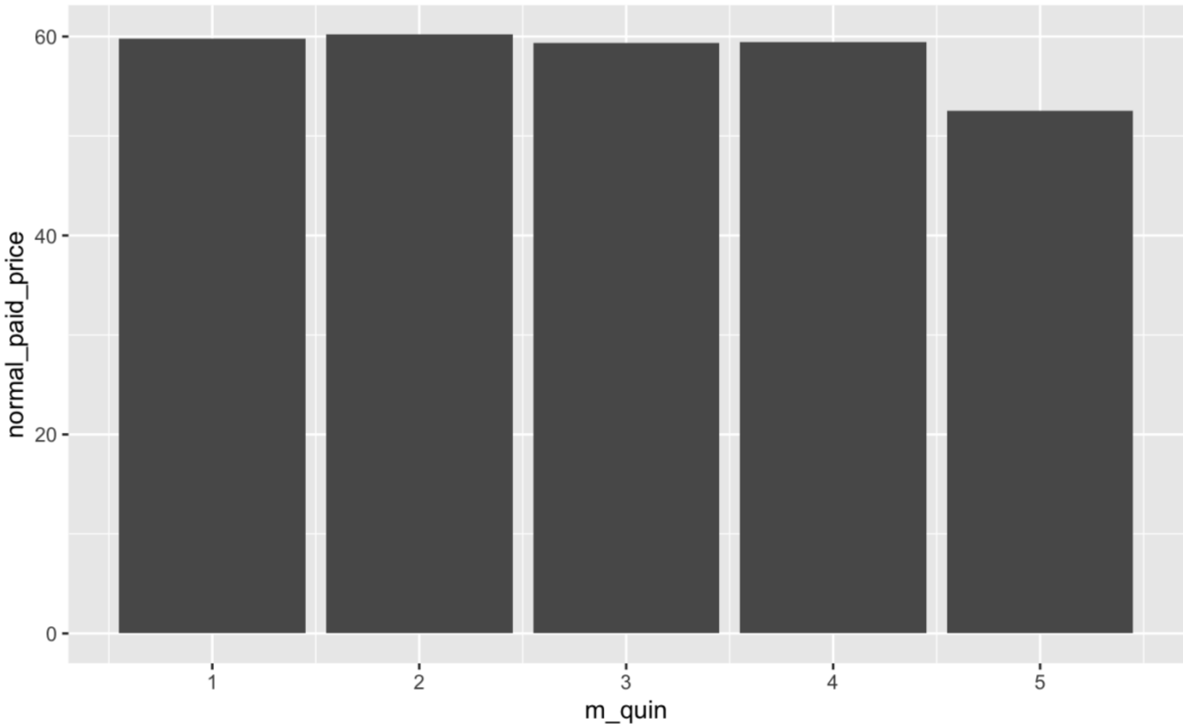
Likewise, we notice a similar trend in the monetary graph with the best performing segment as M = 1. It means that customers who have spent the highest amount are most likely to use the offer.

Q3

```{r}
library (dplyr)

subset_df <- filter(test_df, offer_used == 1)

ggplot(subset_df) + stat_summary(aes(x =
                                   rec_quin, y = normal_paid_price), fun = "mean",
                              geom = "bar")

ggplot(subset_df) + stat_summary(aes(x =
                                   freq_quin, y = normal_paid_price), fun = "mean",
                              geom = "bar")

ggplot(subset_df) + stat_summary(aes(x =
                                   m_quin, y = normal_paid_price), fun = "mean",
                              geom = "bar")
```


normal_paid_price
60
40
20
0
1
2
3
4
5
rec_quin


normal_paid_price
60
40
20
0
1
2
3
4
5
freq_quin



Interestingly, we could say from the above graphs that the normal price paid is more or less very similar across customer of varying RFM indices. It means that the customers are likely purchasing items with similar prices as they use the offer. We also see an exception wherein M = 5 customers are on and average purchase less costly items with the offer coupon. The exception is understandable as this segment has the least purchasing power, historically, in terms of monetary amount.

Q4

```{r}
avg_revenue <- mean(subset_df$normal_paid_price)
avg_cost <- 0.76 * avg_revenue
avg_profit <- 0.24 * avg_revenue

cat("Average Revenue: $", avg_revenue, "\n")
cat("Average Cost: $", avg_cost, "\n")
cat("Average Profit: $", avg_profit, "\n")
```

```
Average Revenue: $ 59.77821
Average Cost: $ 45.43144
Average Profit: $ 14.34677
```

Q5

```{r}
offer_cost <- 1.08
breakeven_rr <- offer_cost/avg_profit
cat("Breakeven Response Rate:", 100 * breakeven_rr, "%\n")
```

```
Breakeven Response Rate: 7.527826 %
```

Q6

```{r}
customer_base <- 39968762
overall_cost <-  customer_base * offer_cost
total_profit <- avg_profit * (customer_base * percentage_used_offer/100) - overall_cost

return_on_marketing <- total_profit/overall_cost

cat("Profit:", total_profit, "\n")
cat("Return on Marketing Cost:", 100 * return_on_marketing, "%\n")
```

```
Profit: 12013382
Return on Marketing Cost: 27.83049 %
```

Q7

```{r}
test_df <- test_df %>%
  group_by(rfmindex_iq) %>%
  mutate(buyprob_iq = mean(offer_used))

test_df$mailto_iq <- ifelse(test_df$buyprob_iq > breakeven_rr, 1, 0)
target_customer_share <- mean(test_df$mailto_iq)
```

```{r}
target_customers <- target_customer_share * customer_base
cat("How many customers to target?", target_customers, "\n")

RFM_response_rate <- mean(subset(test_df, mailto_iq == 1)$offer_used)

cat("RFM Response Rate (New):", 100 * RFM_response_rate, "%\n")
cat("Response Rate in absence of RFM:", percentage_used_offer, "%\n")
```

```
How many customers to target? 9313864
RFM Response Rate (New): 37.359 %
Response Rate in absence of RFM: 9.622857 %
```

```{r}
RFM_profit <- (RFM_response_rate * target_customers * avg_profit) - (target_customers * offer_cost)
return_on_marketing_RFM <- RFM_profit / (target_customers * offer_cost)

cat("Profit after RFM: $", RFM_profit, "\n")
cat("Return on Marketing Cost after RFM:", 100 * return_on_marketing_RFM, "%\n")
```

```
Profit after RFM: $ 39861567
Return on Marketing Cost after RFM: 396.2787 %
```

Values	
avg_cost	45.4314416962896
avg_profit	14.3467710619862
avg_revenue	59.7782127582757
breakeven_rr	0.0752782626372017
customer_base	39968762
offer_cost	1.08
overall_cost	43166262.96
percentage_used_offer	9.62285714285714
return_on_marketing	0.278304892507128
return_on_marketing_RFM	3.96278715804224
RFM_profit	39861567.4086113
RFM_response_rate	0.373589995095635
target_customer_share	0.233028571428572
target_customers	9313863.51062858
total_profit	12013382.1730172
total_spending_used_offer	201333.020569872

In the previous case, we were sending offer coupons to Sephora's entire customer base and wasting a lot of money on people who were not likely to respond to the offer. RFM analysis is based on the principle of reducing these costs and increasing the overall profitability of the marketing campaign. After RFM, we only target customers who are above the breakeven rate, and hence significantly decrease campaign costs. The target_customer_share is the corresponding share of customers we want to target. We can see the Return on Marketing cost has significantly jumped from 27.83% to 396.28% while the profits have increased to $39861567.

Q8

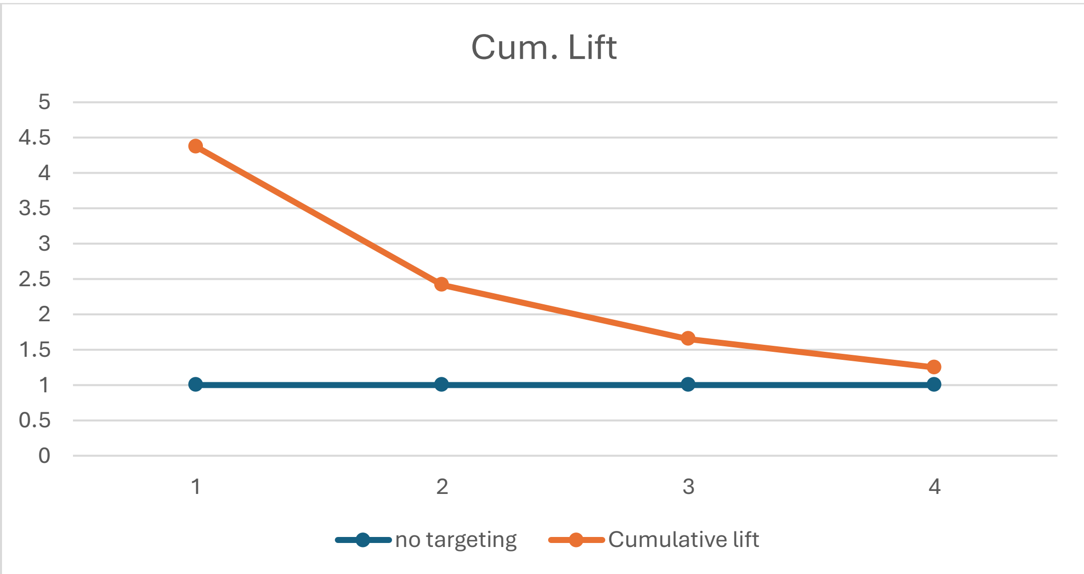
```{r}
test_df$rfm_score_tiles <- 6 - ntile(test_df$buyprob_iq, 5)

test_df %>%
  group_by(rfm_score_tiles) %>%
  summarise(
    count = length(user_id),
    buyers = sum(offer_used))
```

A tibble: 5 × 3

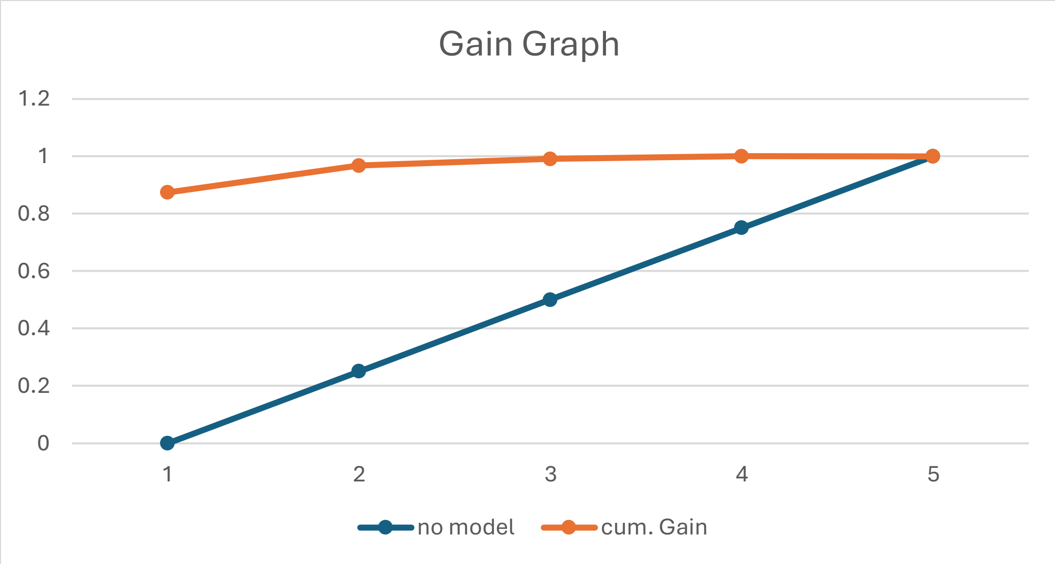
rfm_score_tiles <dbl>	count <int>	buyers <int>
1	7000	2945
2	7000	313
3	7000	81
4	7000	28
5	7000	1

5 rows

quantile	number of customers	number of respondents	cummulative number of customers	cummulative number of respondents	response rate of this group	cummulative response rate	Lift	Cumulative lift
1	7000	2945	7000	2945	42.07%	42.07%	4.372031	4.372030944
2	7000	313	14000	3258	4.47%	23.27%	0.464667	2.418349205
3	7000	81	21000	3339	1.16%	15.90%	0.120249	1.652315939
4	7000	28	28000	3367	0.40%	12.03%	0.041568	1.249628878
5	7000	1	35000	3368	0.01%	9.62%	0.001485	1.000000015



quantile	number of customers	number of respondents	cummulati	cummulative number of respondents	gain	cum. Gain
1	7000	2945	7000	2945	0.874406	0.874406
2	7000	313	14000	3258	0.092933	0.96734
3	7000	81	21000	3339	0.02405	0.99139
4	7000	28	28000	3367	0.008314	0.999703
5	7000	1	35000	3368	0.000297	1
		3368				



Lift is based on comparing the response rate within each quantile to the overall average response rate. Higher lift values are observed in top quantiles. Whereas, gain measures the cumulative percentage of total responses captured as more quantiles are included.

Life curve steeply declines after the top quantile, showing that the highest-probability segments drive most of the response. Gain curve demonstrates that a small portion of high-value customers account for the majority of responses.

Both Lift and gain analyses confirm that targeting lower performing quantiles adds minimal value while incurring higher costs.

Q9

Shortcomings:

- Equal RFM Weighting: We have treated R, F, and M as equally important. Now this is oversimplifying customer behavior. Some metrics might hold greater predictive power.
- Static Customer Behavior: Assumes that RFM scores and response probabilities remain constant, ignoring seasonal changes.
- Fixed Marginal Costs: Also assumed a static cost of annoyance, which may vary based on email content or campaign frequency.

Assumptions:

- Uniform Revenue Margins: Assuming the same profit margin (24%) across all purchases may not account for product-specific variability.
- Predictive Accuracy: Relies on historical behavior as a perfect predictor of future actions, which may not account for evolving preferences.
- Exclusion Effects: Ignores potential alienation of lower RFM segments who might still respond positively under certain conditions.

Conclusion: While the analysis provides actionable insights, adding dynamic RFM weighting, temporal behavior changes, or campaign-specific variations would improve accuracy in real-world scenarios.