Winery

Weihan Weng

2023-11-22

Clean and modify dataset

Market Analysis Make two more data frame, just want avoid cluttering the original database. winery_unique is for doing marketing analysis, avoid double counting sales.

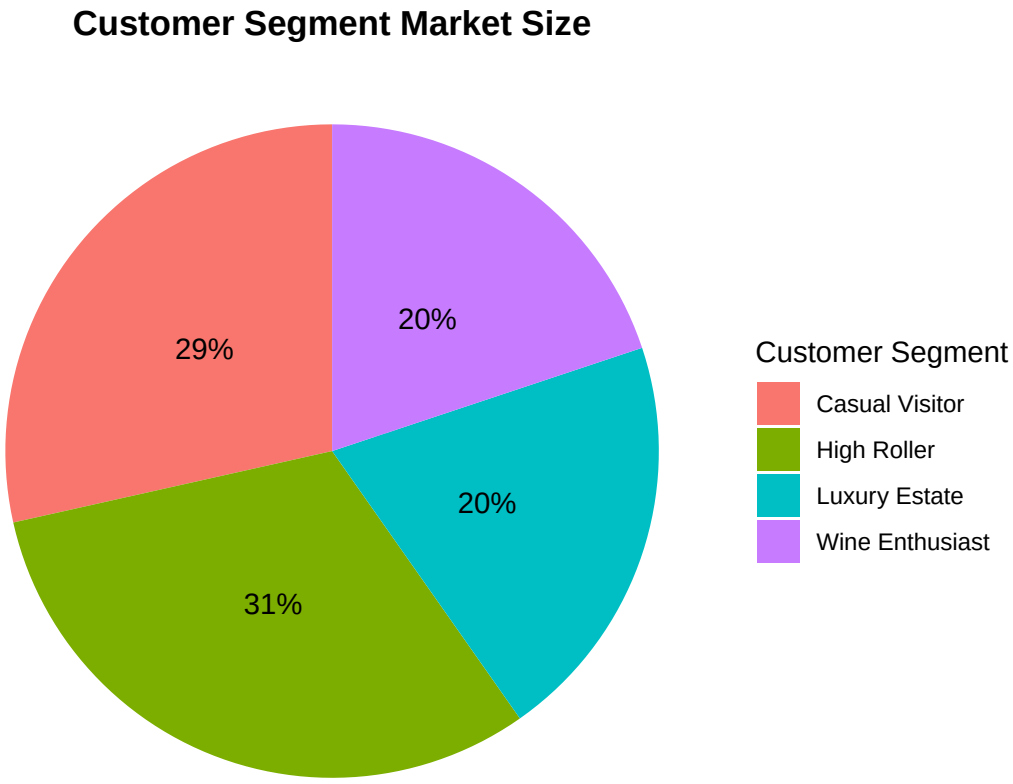
```
winery = read.csv("Winery_Data_Students.csv")

# Clean dataset
winery[winery == '#N/A'] <- NA
winery[winery == ""] <- NA
winery <- winery[complete.cases(winery), ]

winery_analysis <- data.frame(Customer.ID = winery$Customer.ID,
                              Customer.Segment = winery$Customer.Segment,
                              State = winery$State,
                              Sales.2008 = winery$Sales.2008,
                              Sales.2009 = winery$Sales.2009,
                              Sales.2010 = winery$Sales.2010)
duplicates <- duplicated(winery_analysis$Customer.ID)
winery_unique <- winery_analysis[!duplicates, ]
head(winery_unique,5)
```

```
##    Customer.ID Customer.Segment State Sales.2008 Sales.2009 Sales.2010
## 1            1      High Roller    FL        213   30903.10   13340.94
## 17           2      High Roller    WA         56   18729.56   23416.11
## 26           3      High Roller    CA          0    3022.00   25371.49
## 34           4  Wine Enthusiast    ME          0    3560.00   19950.13
## 41           5   Casual Visitor    WA          0   10415.23   10146.87
```

```
sales_2008 <- aggregate(Sales.2008 ~ Customer.Segment, winery_unique, FUN = sum)
sales_2009 <- aggregate(Sales.2009 ~ Customer.Segment, winery_unique, FUN = sum)
sales_2010 <- aggregate(Sales.2010 ~ Customer.Segment, winery_unique, FUN = sum)

sales <- merge(sales_2008, sales_2009, by = "Customer.Segment",
               suffixes = c(".2008", ".2009"))
sales <- merge(sales, sales_2010, by = "Customer.Segment",
               suffixes = c(".2008", ".2010"))
names(sales)[names(
  sales) == "Sales.2008"] <- "2008"
names(sales)[names(
  sales) == "Sales.2009"] <- "2009"
names(sales)[names(
  sales) == "Sales.2010"] <- "2010"

sales$TotalSales <- sales[["2008"]] + sales[["2009"]] + sales[["2010"]]

sales
```

```
##   Customer.Segment     2008     2009     2010 TotalSales
## 1   Casual Visitor 355186.8 713695.0 312100.0  1380981.8
## 2      High Roller 127050.9 887089.6 498976.8  1513117.3
## 3    Luxury Estate 208635.7 536016.0 242124.6   986776.3
## 4  Wine Enthusiast 148352.8 502976.9 311760.9   963090.5
```

```
library(ggplot2)
sales$MarketShare <- sales$TotalSales / sum(sales$TotalSales) * 100

ggplot(sales, aes(x = "", y = TotalSales, fill = Customer.Segment)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0) +
  labs(title = "Customer Segment Market Size", fill = "Customer Segment") +
  geom_text(aes(label = paste0(round(MarketShare), "%")),
            position = position_stack(vjust = 0.5)) +
  theme_void() +
  theme(legend.position = "right",
        plot.title = element_text(hjust = 0.5, face = "bold"))
```



Based on Location, find out the top 5 states on sales

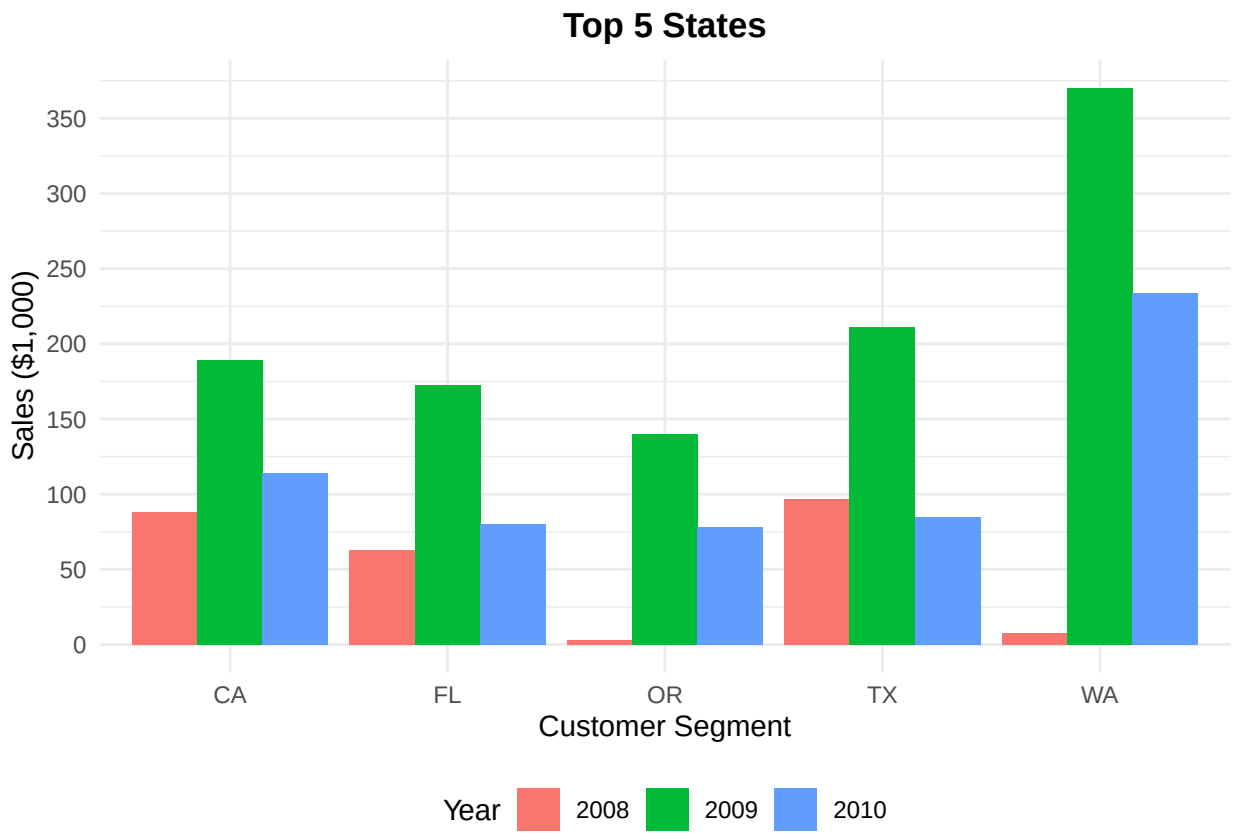
```
sales2008 <- aggregate(Sales.2008 ~ State, winery_unique, FUN = sum)
sales2009 <- aggregate(Sales.2009 ~ State, winery_unique, FUN = sum)
sales2010 <- aggregate(Sales.2010 ~ State, winery_unique, FUN = sum)

state <- merge(sales2008, sales2009, by = "State",
               suffixes = c(".2008", ".2009"))
state <- merge(state, sales2010, by = "State", suffixes = c(".2008", ".2010"))

names(state)[names(
  state) == "Sales.2008"] <- "2008"
names(state)[names(
```

```r
  state) == "Sales.2009"] <- "2009"
names(state)[names(
  state) == "Sales.2010"] <- "2010"

state$Total_Sales <- rowSums(state[, c("2008", "2009", "2010")])
state <- state[order(state$Total_Sales, decreasing = TRUE), ]
top_5_states <- head(state, 5)
top_5_states <- subset(top_5_states, select = -c(Total_Sales))
```

```r
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.2
```

```r
state_convert <- gather(top_5_states, key = "Year", value = "Sales", -State)

ggplot(state_convert, aes(x = State, y = Sales, fill = Year)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Top 5 States", x = "Customer Segment", y = "Sales ($1,000)") +
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(hjust = 0.5, face = "bold")) +
  scale_y_continuous(labels = function(x) x/1000,
                     breaks = seq(0, max(state_convert$Sales), by = 50000),
                     limits = c(0, max(state_convert$Sales)))
```



```r
# Detect whether customers are affected by marketing avtivities
winery$Response <- ifelse((winery$Newsletter.Subscr + winery$Email.Subscr
                           + winery$Winemaker.call) > 1
                          & winery$Email.Sales + winery$Newsletter.Sales
                          + winery$Winemaker.Call.Sales > 0, 1, 0)

# Convert Customer.Segment to dummy variable and keep other columns
winery$Customer.Segment <- gsub("High Roller", "HighRoller",
                                winery$Customer.Segment)
winery$Customer.Segment <- gsub("Casual Visitor", "CasualVisitor",
                                winery$Customer.Segment)
winery$Customer.Segment <- gsub("Luxury Estate", "LuxuryEstate",
                                winery$Customer.Segment)
winery$Customer.Segment <- gsub("Wine Enthusiast", "WineEnthusiast",
                                winery$Customer.Segment)
winery$Customer.Segment <- as.factor(winery$Customer.Segment)
dummy_vars <- model.matrix(~ Customer.Segment - 1, data = winery)
winery <- cbind(winery[, !names(winery) %in% "Customer.Segment"], dummy_vars)

head(winery, 5)
```

```
##   Customer.ID Order.ID      Date Zip.Code State Sales.2008 Sales.2009
## 1           1     1532  8-Jul-08    33467    FL        213    30903.1
## 2           1    14378  5-Oct-08    33467    FL        213    30903.1
## 3           1    17690 26-Oct-08    33467    FL        213    30903.1
## 4           1    19808  8-Nov-08    33467    FL        213    30903.1
## 5           1    25406  2-Jan-09    33467    FL        213    30903.1
##   Sales.2010 Sale.Amount Orders.2008 Orders.2009 Orders.2010 Year.Acquired
## 1   13340.94          44           4           8           4          2008
## 2   13340.94          47           4           8           4          2008
## 3   13340.94          57           4           8           4          2008
## 4   13340.94          65           4           8           4          2008
## 5   13340.94        3889           4           8           4          2008
##   Email.Subscr Newsletter.Subscr Winemaker.call Email.Sales Newsletter.Sales
## 1            1                 1              1           0                0
## 2            1                 1              1           0                0
## 3            1                 1              1           0               57
## 4            1                 1              1           0                0
## 5            1                 1              1           0                0
##   Tasting.Room.Sales Winemaker.Call.Sales Response
## 1                 44                    0        0
## 2                 47                    0        0
## 3                  0                    0        1
## 4                 65                    0        0
## 5               3889                    0        0
##   Customer.SegmentCasualVisitor Customer.SegmentHighRoller
## 1                             0                         1
## 2                             0                         1
## 3                             0                         1
## 4                             0                         1
## 5                             0                         1
##   Customer.SegmentLuxuryEstate Customer.SegmentWineEnthusiast
## 1                            0                             0
```

```
## 2                             0                             0
## 3                             0                             0
## 4                             0                             0
## 5                             0                             0
```

Divide into training and testing datasets

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.2
```

```
## Loading required package: lattice
```

```
set.seed(123)
index <- createDataPartition(y = 1:nrow(winery), p = 0.7, list = FALSE)

train <- winery[index, ]
test <- winery[-index, ]

head(train, 3)
```

```
##   Customer.ID Order.ID     Date Zip.Code State Sales.2008 Sales.2009 Sales.2010
## 1           1     1532 8-Jul-08    33467    FL        213    30903.1   13340.94
## 2           1    14378 5-Oct-08    33467    FL        213    30903.1   13340.94
## 5           1    25406 2-Jan-09    33467    FL        213    30903.1   13340.94
##   Sale.Amount Orders.2008 Orders.2009 Orders.2010 Year.Acquired Email.Subscr
## 1          44           4           8           4          2008            1
## 2          47           4           8           4          2008            1
## 5        3889           4           8           4          2008            1
##   Newsletter.Subscr Winemaker.call Email.Sales Newsletter.Sales
## 1                 1              1           0                0
## 2                 1              1           0                0
## 5                 1              1           0                0
##   Tasting.Room.Sales Winemaker.Call.Sales Response
## 1                 44                    0        0
## 2                 47                    0        0
## 5               3889                    0        0
##   Customer.SegmentCasualVisitor Customer.SegmentHighRoller
## 1                             0                          1
## 2                             0                          1
## 5                             0                          1
##   Customer.SegmentLuxuryEstate Customer.SegmentWineEnthusiast
## 1                            0                              0
## 2                            0                              0
## 5                            0                              0
```

```
head(test, 3)
```

```
##   Customer.ID Order.ID      Date Zip.Code State Sales.2008 Sales.2009
## 3           1    17690 26-Oct-08    33467    FL        213    30903.1
## 4           1    19808  8-Nov-08    33467    FL        213    30903.1
```

```
## 8           1    40916  8-Jun-09    33467    FL        213    30903.1
##   Sales.2010 Sale.Amount Orders.2008 Orders.2009 Orders.2010 Year.Acquired
## 3   13340.94       57.00           4           8           4          2008
## 4   13340.94       65.00           4           8           4          2008
## 8   13340.94     1928.73           4           8           4          2008
##   Email.Subscr Newsletter.Subscr Winemaker.call Email.Sales Newsletter.Sales
## 3            1                 1              1           0               57
## 4            1                 1              1           0                0
## 8            1                 1              1           0                0
##   Tasting.Room.Sales Winemaker.Call.Sales Response
## 3               0.00                    0        1
## 4              65.00                    0        0
## 8            1928.73                    0        0
##   Customer.SegmentCasualVisitor Customer.SegmentHighRoller
## 3                             0                          1
## 4                             0                          1
## 8                             0                          1
##   Customer.SegmentLuxuryEstate Customer.SegmentWineEnthusiast
## 3                            0                             0
## 4                            0                             0
## 8                            0                             0
```

Response Analysis (For Entire Sales Process) Make a logistic regression model for entire sales channels, find out the most important/valuable channel can let customer buy.

Based on the outcome of model, Newsletter and Email are the most valuable channels.

```
glm <- glm(Response ~ Newsletter.Subscr + Email.Subscr + Winemaker.call +
             Year.Acquired + Customer.SegmentCasualVisitor +
             Customer.SegmentCasualVisitor + Customer.SegmentHighRoller +
             Customer.SegmentLuxuryEstate + Customer.SegmentWineEnthusiast,
           family = binomial(link = "logit"), data = train)
summary(glm)
```

```
##
## Call:
## glm(formula = Response ~ Newsletter.Subscr + Email.Subscr + Winemaker.call +
##     Year.Acquired + Customer.SegmentCasualVisitor + Customer.SegmentCasualVisitor +
##     Customer.SegmentHighRoller + Customer.SegmentLuxuryEstate +
##     Customer.SegmentWineEnthusiast, family = binomial(link = "logit"),
##     data = train)
##
## Coefficients: (1 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   46.02531   32.38999   1.421    0.155
## Newsletter.Subscr              1.85968    0.08270  22.486   <2e-16 ***
## Email.Subscr                   2.17108    0.07733  28.075   <2e-16 ***
## Winemaker.call                 2.62839    0.06288  41.802   <2e-16 ***
## Year.Acquired                 -0.02601    0.01613  -1.612    0.107
## Customer.SegmentCasualVisitor  1.15795    0.05942  19.488   <2e-16 ***
## Customer.SegmentHighRoller    -1.70323    0.04905 -34.722   <2e-16 ***
## Customer.SegmentLuxuryEstate  -3.18810    0.13072 -24.388   <2e-16 ***
## Customer.SegmentWineEnthusiast       NA         NA      NA       NA
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 36440  on 45392  degrees of freedom
## Residual deviance: 17555  on 45385  degrees of freedom
## AIC: 17571
##
## Number of Fisher Scoring iterations: 8
```

Input testing dataset to model and sort based on lift

```
df <- data.frame(
  Order.ID = test$Order.ID,
  Newsletter.Subscr = test$Newsletter.Subscr,
  Email.Subscr = test$Email.Subscr,
  Winemaker.call = test$Winemaker.call,
  Year.Acquired = test$Year.Acquired,
  Customer.SegmentCasualVisitor = test$Customer.SegmentCasualVisitor,
  Customer.SegmentHighRoller = test$Customer.SegmentHighRoller,
  Customer.SegmentLuxuryEstate = test$Customer.SegmentLuxuryEstate,
  Customer.SegmentWineEnthusiast = test$Customer.SegmentWineEnthusiast
)

prediction <- data.frame(
  Order.ID = df$Order.ID,
  ResponseProb = predict(glm, df, type = c("response")),
  ResponsePredict = round(predict(glm, df,
                                  type = c("response")), digits = 0)
)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
prediction$ActualResponse = test$Response #add actual response
prediction$Lift = prediction$ResponseProb/mean(train$Response) #add lift

prediction.sorting <- prediction[order(prediction$Lift, decreasing = TRUE), ]

prediction.sorting$cumsum_prob = cumsum(prediction.sorting$ResponseProb)
prediction.sorting$cumsum_actualresponse = cumsum(
  prediction.sorting$ActualResponse
  )

print(head(prediction.sorting, 5))
```

```
##       Order.ID ResponseProb ResponsePredict ActualResponse     Lift cumsum_prob
## 365      92887    0.8444562               1              0 6.117523   0.8444562
## 7585     76041    0.8444562               1              1 6.117523   1.6889125
## 8847     69486    0.8444562               1              1 6.117523   2.5333687
```

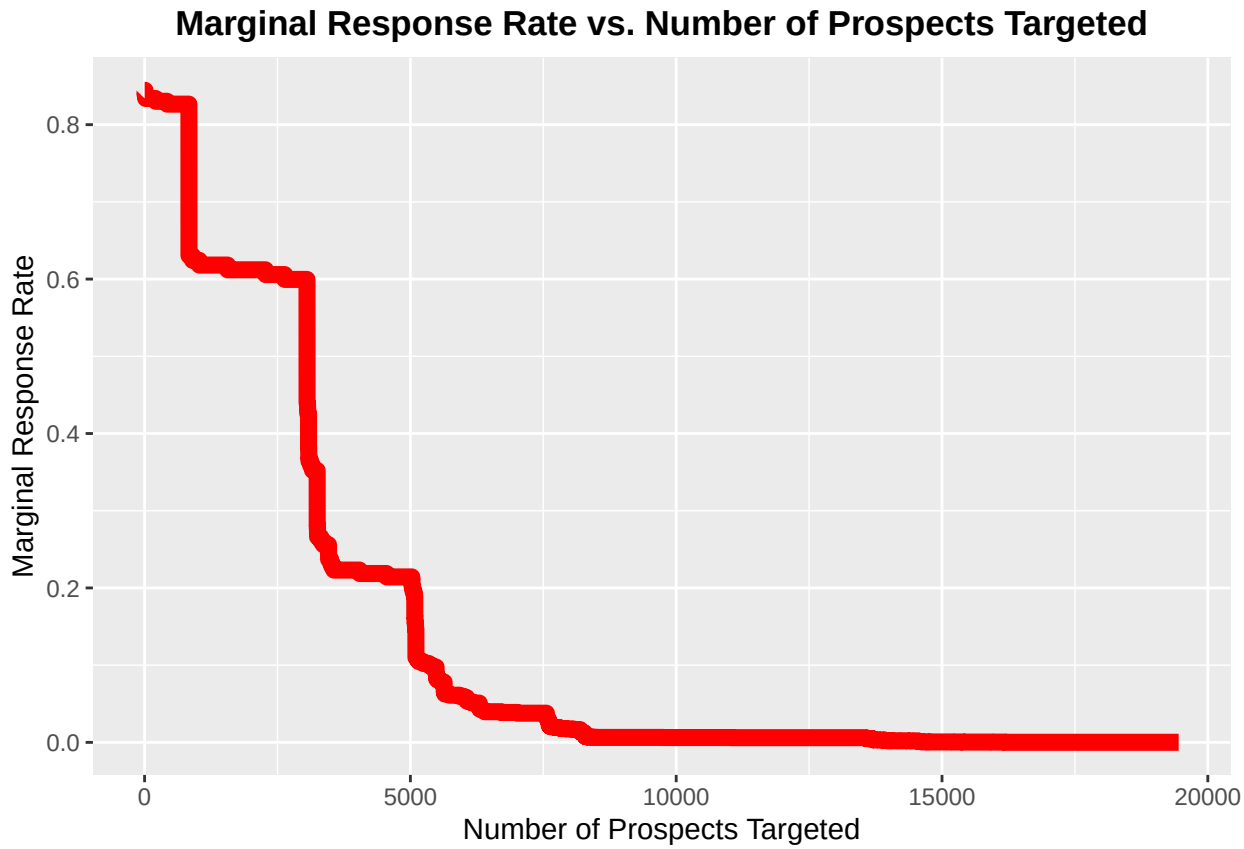
```
## 11227    86705    0.8444562              1              1 6.117523   3.3778249
## 11228    92676    0.8444562              1              1 6.117523   4.2222812
##       cumsum_actualresponse
## 365                       0
## 7585                      1
## 8847                      2
## 11227                     3
## 11228                     4
```

Use ggplot to make margin response graph

```
library(ggplot2)

x_values <- seq(1, nrow(prediction.sorting))

ggplot(prediction.sorting, aes(x = x_values, y = ResponseProb)) +
  geom_line(color = "red", shape = 1, size = 3) +
  labs(title = "Marginal Response Rate vs. Number of Prospects Targeted",
       x = "Number of Prospects Targeted", y = "Marginal Response Rate") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```

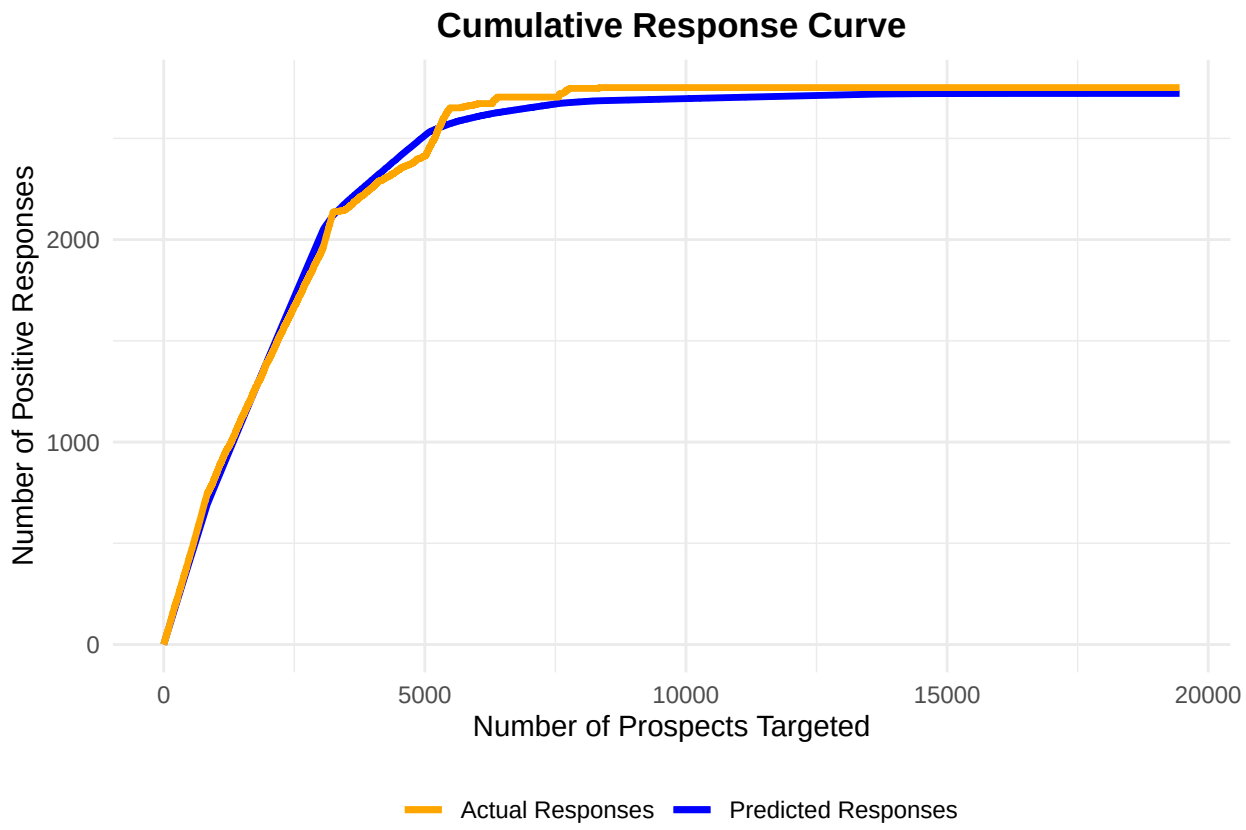
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## Warning in geom_line(color = "red", shape = 1, size = 3): Ignoring unknown
## parameters: ‘shape‘
```



Use ggplot make cumulative plot It looks like the predicted values are in good agreement with the actual values, which may indicate that the model is good.

```
x_values <- seq(1, nrow(prediction.sorting))

ggplot(prediction.sorting) +
  geom_line(aes(x = x_values, y = cumsum_prob, color = "Predicted Responses"),
            linewidth = 1.2) +
  geom_line(aes(x = x_values, y = cumsum_actualresponse,
                color = "Actual Responses"), linewidth = 1.2) +
  labs(
    title = "Cumulative Response Curve",
    x = "Number of Prospects Targeted",
    y = "Number of Positive Responses"
  ) +
  scale_color_manual(values = c("Predicted Responses" = "blue",
                                "Actual Responses" = "orange")) +
  theme_minimal() +
  theme(legend.position = "bottom", legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, face = "bold"))
```



RFM Analysis I only target on 2010 data

```
winery_2010 <- subset(winery, format(
  as.Date(Date, format = "%d-%b-%y"), "%Y") == "2010")
winery_2010$Date <- as.Date(winery_2010$Date, format = "%d-%b-%y")
winery_2010 <- winery_2010[order(winery_2010$Date), ]

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
winery_2010 <- winery_2010 %>%
  group_by(Customer.ID) %>%
  slice(tail(row_number(), 1)) %>%
  ungroup()
```

```r
max_date <- winery_2010$Date[which.max(winery_2010$Date)]
winery_2010$Recency <- max_date - winery_2010$Date

head(winery_2010,5)
```

```
## # A tibble: 5 x 26
##   Customer.ID Order.ID Date       Zip.Code State Sales.2008 Sales.2009
##         <int>    <int> <date>        <int> <chr>      <dbl>      <dbl>
## 1           1    94016 2010-06-17    33467 FL           213     30903.
## 2           2    94019 2010-06-17    98683 WA            56     18730.
## 3           3    89832 2010-05-13    90247 CA             0      3022
## 4           4    92069 2010-06-04     4572 ME             0      3560
## 5           5    90764 2010-05-21    98042 WA             0     10415.
## # i 19 more variables: Sales.2010 <dbl>, Sale.Amount <dbl>, Orders.2008 <chr>,
## #   Orders.2009 <chr>, Orders.2010 <chr>, Year.Acquired <int>,
## #   Email.Subscr <int>, Newsletter.Subscr <int>, Winemaker.call <int>,
## #   Email.Sales <dbl>, Newsletter.Sales <dbl>, Tasting.Room.Sales <dbl>,
## #   Winemaker.Call.Sales <dbl>, Response <dbl>,
## #   Customer.SegmentCasualVisitor <dbl>, Customer.SegmentHighRoller <dbl>,
## #   Customer.SegmentLuxuryEstate <dbl>, ...
```

```r
RFM <- data.frame(ID = winery_2010$Customer.ID, Recency = winery_2010$Recency,
                  Frequency = winery_2010$Orders.2010,
                  Monetary = winery_2010$Sales.2010)
RFM$Frequency <- as.integer(RFM$Frequency)
RFM$Recency <- as.integer(RFM$Recency)

head(RFM,5)
```

```
##   ID Recency Frequency Monetary
## 1  1      13         4 13340.94
## 2  2      13         5 23416.11
## 3  3      48         7 25371.49
## 4  4      26         5 19950.13
## 5  5      40         2 10146.87
```

Calculate the scores

```r
source("RFM_Functions.R")
RFM.score <-getIndependentScore(RFM)
head(RFM.score, 10)
```

```
##    ID Recency Frequency Monetary R_Score F_Score M_Score Total_Score
## 2   2      13         5 23416.11       5       5       5         555
## 4   4      26         5 19950.13       5       5       5         555
## 1   1      13         4 13340.94       5       5       5         555
## 10 12      19         3 12568.60       5       5       5         555
## 14 17      26         5 12499.74       5       5       5         555
## 43 49      12         4 11217.78       5       5       5         555
## 39 44      22         2  5745.52       5       5       5         555
## 40 45      27         2  5219.35       5       5       5         555
## 69 89      23         3  4400.73       5       5       5         555
## 13 15      13         3  3648.17       5       5       5         555
```

```r
tail(RFM.score, 10)
```

```
##         ID Recency Frequency Monetary R_Score F_Score M_Score Total_Score
## 6017 14410     177         1        0       1       4       1         141
## 537   1007     178         1        0       1       4       1         141
## 1490  2651     178         1        0       1       4       1         141
## 2053  3647     178         1        0       1       4       1         141
## 5942 14240     178         1        0       1       4       1         141
## 6634 18125     178         1        0       1       4       1         141
## 526    988     179         1        0       1       4       1         141
## 1538  2761     179         1        0       1       4       1         141
## 5427 11671     179         1        0       1       4       1         141
## 6807 18518     179         1        0       1       4       1         141
```

I am not sure if Monetary can be negative, it makes sense, like customers want to return products (2010).

```r
count_555 <- sum(RFM.score$Total_Score == 555)
paste("There are", count_555, "customers are high-value customers.")
```

```
## [1] "There are 358 customers are high-value customers."
```

RFM based on 2009 data

```r
winery_2009 <- subset(winery, format(
  as.Date(Date, format = "%d-%b-%y"), "%Y") == "2009")
winery_2009$Date <- as.Date(winery_2009$Date, format = "%d-%b-%y")
winery_2009 <- winery_2009[order(winery_2009$Date), ]

winery_2009 <- winery_2009 %>%
  group_by(Customer.ID) %>%
  mutate(Sales.2009 = sum(Sale.Amount, na.rm = TRUE)) %>%
  slice(tail(row_number(), 1)) %>%
  ungroup()

max_date2009 <- "12/31/2009"
max_date2009 <- as.Date(max_date2009, format = "%m/%d/%Y")
winery_2009$Recency <- max_date2009 - winery_2009$Date

head(winery_2009,5)
```

```
## # A tibble: 5 x 26
##   Customer.ID Order.ID Date       Zip.Code State Sales.2008 Sales.2009
##         <int>    <int> <date>        <int> <chr>      <dbl>      <dbl>
## 1           1    62622 2009-11-08    33467 FL           213     30903.
## 2           2    67481 2009-12-08    98683 WA            56     18730.
## 3           3    67950 2009-12-11    90247 CA             0      3022
## 4           4    60575 2009-10-26     4572 ME             0      3560
## 5           5    55182 2009-09-21    98042 WA             0     10414.
## # i 19 more variables: Sales.2010 <dbl>, Sale.Amount <dbl>, Orders.2008 <chr>,
## #   Orders.2009 <chr>, Orders.2010 <chr>, Year.Acquired <int>,
## #   Email.Subscr <int>, Newsletter.Subscr <int>, Winemaker.call <int>,
```

```
## #   Email.Sales <dbl>, Newsletter.Sales <dbl>, Tasting.Room.Sales <dbl>,
## #   Winemaker.Call.Sales <dbl>, Response <dbl>,
## #   Customer.SegmentCasualVisitor <dbl>, Customer.SegmentHighRoller <dbl>,
## #   Customer.SegmentLuxuryEstate <dbl>, ...
```

```
RFM2009 <- data.frame(ID = winery_2009$Customer.ID,
                      Recency = winery_2009$Recency,
                      Frequency = winery_2009$Orders.2009,
                      Monetary = winery_2009$Sales.2009)
RFM2009$Frequency <- as.integer(RFM2009$Frequency)
RFM2009$Recency <- as.integer(RFM2009$Recency)

source("RFM_Functions.R")
RFM2009.score <-getIndependentScore(RFM2009)
head(RFM2009.score, 10)
```

```
##    ID Recency Frequency Monetary R_Score F_Score M_Score Total_Score
## 7   7       0         5 19299.98       5       5       5         555
## 2   2      23         3 18729.56       5       5       5         555
## 16 16       2         5 17885.71       5       5       5         555
## 9   9      15         2 17349.63       5       5       5         555
## 40 40      35         3 15978.08       5       5       5         555
## 6   6       2         5 12755.46       5       5       5         555
## 50 50      21         4 12104.01       5       5       5         555
## 43 43      36         5 11961.46       5       5       5         555
## 78 78      48         3  9718.35       5       5       5         555
## 26 26       6         3  9328.27       5       5       5         555
```

```
tail(RFM2009.score, 10)
```

```
##          ID Recency Frequency Monetary R_Score F_Score M_Score Total_Score
## 16059 17527     265         0       24       1       1       3         113
## 16090 17561     266         0       24       1       1       3         113
## 16301 17785     358         0       24       1       1       3         113
## 17002 18508     225         0       23       1       1       3         113
## 17171 18817     260         0       23       1       1       3         113
## 16848 18336     345         0       23       1       1       3         113
## 16982 18470     349         0       23       1       1       3         113
## 18665 20475     226         0       21       1       1       3         113
## 18713 20528     297         0       21       1       1       3         113
## 18444 20238     323         0       21       1       1       3         113
```

```
RFM2009.score_sorted <- RFM2009.score[order(RFM2009.score$ID),]
merged_data <- inner_join(winery, RFM2009.score, by = c("Customer.ID" = "ID"))
merged_data <- distinct(merged_data, Customer.ID)

data2010 <- data.frame(Customer.ID = winery_2010$Customer.ID)

merged_data <- merged_data %>%
  mutate(`Response2010` = ifelse(Customer.ID %in% data2010$Customer.ID, 1, 0))

head(merged_data)
```

```
##   Customer.ID Response2010
## 1           1            1
## 2           2            1
## 3           3            1
## 4           4            1
## 5           5            1
## 6           6            1
```

```r
# Crosstab of Recency Score vs. Buyer (did or did not buy offer)
library(gmodels)
```

```
## Warning: package 'gmodels' was built under R version 4.3.2
```

```r
#For Recency
data_crosstab_Recency <- CrossTable(RFM2009.score_sorted$R_Score,
                                    merged_data$Response2010,prop.r=TRUE,
                                    prop.c=FALSE, prop.t=FALSE,
                                    prop.chisq=FALSE, dnn = c("R","Response"))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |-------------------------|
##
##
## Total Observations in Table:  20919
##
##
##              | Response
##            R |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            1 |      2841 |      1316 |      4157 |
##              |     0.683 |     0.317 |     0.199 |
## -------------|-----------|-----------|-----------|
##            2 |      2906 |      1268 |      4174 |
##              |     0.696 |     0.304 |     0.200 |
## -------------|-----------|-----------|-----------|
##            3 |      2989 |      1139 |      4128 |
##              |     0.724 |     0.276 |     0.197 |
## -------------|-----------|-----------|-----------|
##            4 |      3058 |      1114 |      4172 |
##              |     0.733 |     0.267 |     0.199 |
## -------------|-----------|-----------|-----------|
##            5 |      2671 |      1617 |      4288 |
##              |     0.623 |     0.377 |     0.205 |
## -------------|-----------|-----------|-----------|
## Column Total |     14465 |      6454 |     20919 |
## -------------|-----------|-----------|-----------|
##
##
```

```r
# For Frequency
data_crosstab_Frequency <- CrossTable(RFM2009.score_sorted$F_Score,
                                      merged_data$Response2010,prop.r=TRUE,
                                      prop.c=FALSE, prop.t=FALSE,
                                      prop.chisq=FALSE, dnn = c("F","Response"))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |-------------------------|
##
##
## Total Observations in Table:  20919
##
##
##              | Response
##            F |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            1 |         0 |      1159 |      1159 |
##              |     0.000 |     1.000 |     0.055 |
## -------------|-----------|-----------|-----------|
##            4 |     10179 |      3073 |     13252 |
##              |     0.768 |     0.232 |     0.633 |
## -------------|-----------|-----------|-----------|
##            5 |      4286 |      2222 |      6508 |
##              |     0.659 |     0.341 |     0.311 |
## -------------|-----------|-----------|-----------|
## Column Total |     14465 |      6454 |     20919 |
## -------------|-----------|-----------|-----------|
##
##
```

```r
# For Monetary
data_crosstab_Monetary <- CrossTable(RFM2009.score_sorted$M_Score,
                                     merged_data$Response2010,prop.r=TRUE,
                                     prop.c=FALSE, prop.t=FALSE,
                                     prop.chisq=FALSE, dnn = c("M","Response"))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |-------------------------|
##
##
## Total Observations in Table:  20919
##
##
```

```
##               | Response
##             M |         0 |         1 | Row Total |
## --------------|-----------|-----------|-----------|
##             1 |        53 |         9 |        62 |
##               |     0.855 |     0.145 |     0.003 |
## --------------|-----------|-----------|-----------|
##             2 |      5861 |       341 |      6202 |
##               |     0.945 |     0.055 |     0.296 |
## --------------|-----------|-----------|-----------|
##             3 |      3083 |      1848 |      4931 |
##               |     0.625 |     0.375 |     0.236 |
## --------------|-----------|-----------|-----------|
##             4 |      3148 |      2392 |      5540 |
##               |     0.568 |     0.432 |     0.265 |
## --------------|-----------|-----------|-----------|
##             5 |      2320 |      1864 |      4184 |
##               |     0.554 |     0.446 |     0.200 |
## --------------|-----------|-----------|-----------|
## Column Total |     14465 |      6454 |     20919 |
## --------------|-----------|-----------|-----------|
##
##
```

```
# For Full RFM
data_crosstab_RFM <- CrossTable(RFM2009.score_sorted$Total_Score,
                                merged_data$Response2010,prop.r=TRUE,
                                prop.c=FALSE, prop.t=FALSE, prop.chisq=FALSE,
                                dnn = c("RFM","Response"))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |-------------------------|
##
##
## Total Observations in Table:  20919
##
##
##               | Response
##          RFM |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##          113 |         0 |        67 |        67 |
##              |     0.000 |     1.000 |     0.003 |
## -------------|-----------|-----------|-----------|
##          114 |         0 |       239 |       239 |
##              |     0.000 |     1.000 |     0.011 |
## -------------|-----------|-----------|-----------|
##          115 |         0 |        43 |        43 |
##              |     0.000 |     1.000 |     0.002 |
## -------------|-----------|-----------|-----------|
##          141 |         3 |         1 |         4 |
```

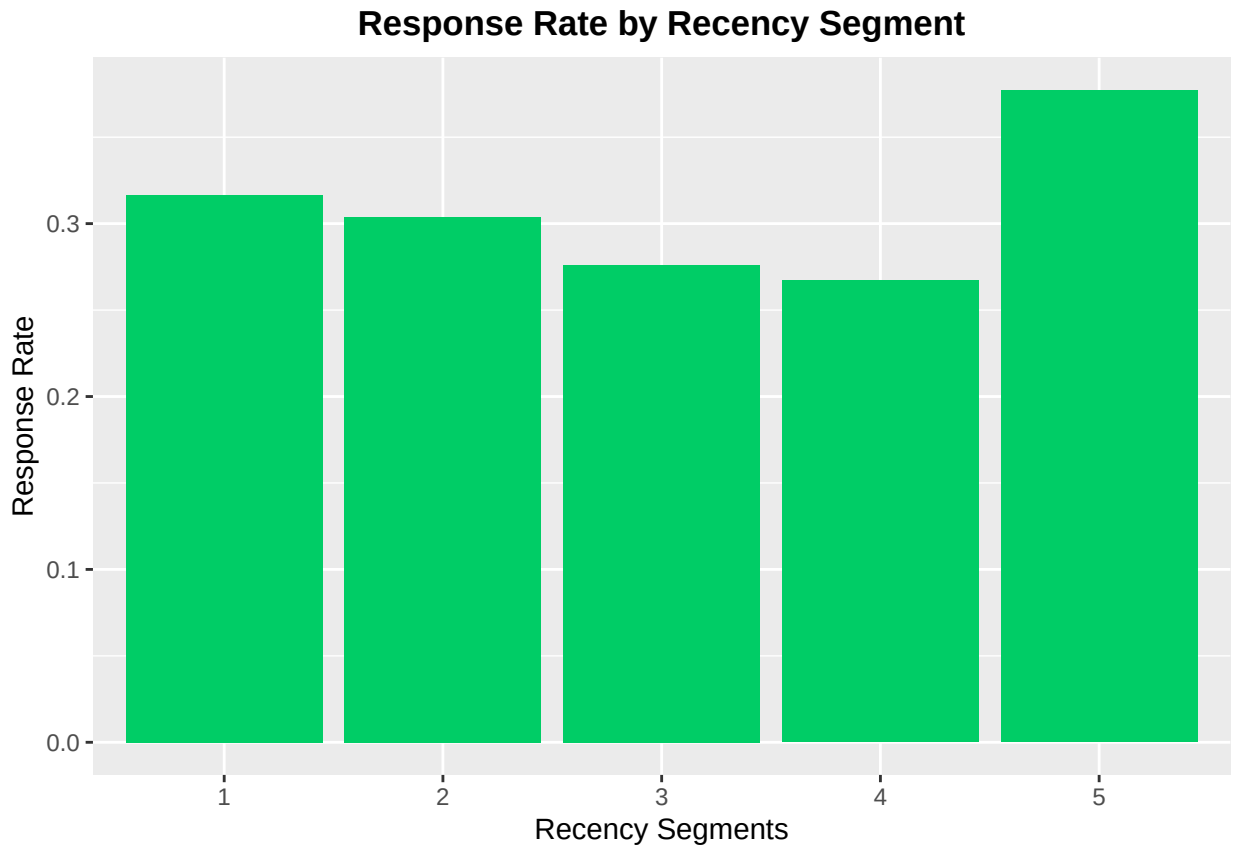
```
##              |     0.750 |     0.250 |     0.000 |
## -------------|-----------|-----------|-----------|
##          142 |      1509 |        70 |      1579 |
##              |     0.956 |     0.044 |     0.075 |
## -------------|-----------|-----------|-----------|
##          143 |       497 |       450 |       947 |
##              |     0.525 |     0.475 |     0.045 |
## -------------|-----------|-----------|-----------|
##          144 |       337 |       303 |       640 |
##              |     0.527 |     0.473 |     0.031 |
## -------------|-----------|-----------|-----------|
##          145 |       195 |        52 |       247 |
##              |     0.789 |     0.211 |     0.012 |
## -------------|-----------|-----------|-----------|
##          151 |         4 |         0 |         4 |
##              |     1.000 |     0.000 |     0.000 |
## -------------|-----------|-----------|-----------|
##          152 |         2 |         0 |         2 |
##              |     1.000 |     0.000 |     0.000 |
## -------------|-----------|-----------|-----------|
##          153 |        73 |         5 |        78 |
##              |     0.936 |     0.064 |     0.004 |
## -------------|-----------|-----------|-----------|
##          154 |       134 |        45 |       179 |
##              |     0.749 |     0.251 |     0.009 |
## -------------|-----------|-----------|-----------|
##          155 |        87 |        41 |       128 |
##              |     0.680 |     0.320 |     0.006 |
## -------------|-----------|-----------|-----------|
##          213 |         0 |        63 |        63 |
##              |     0.000 |     1.000 |     0.003 |
## -------------|-----------|-----------|-----------|
##          214 |         0 |       154 |       154 |
##              |     0.000 |     1.000 |     0.007 |
## -------------|-----------|-----------|-----------|
##          215 |         0 |        23 |        23 |
##              |     0.000 |     1.000 |     0.001 |
## -------------|-----------|-----------|-----------|
##          241 |         0 |         4 |         4 |
##              |     0.000 |     1.000 |     0.000 |
## -------------|-----------|-----------|-----------|
##          242 |      1333 |       109 |      1442 |
##              |     0.924 |     0.076 |     0.069 |
## -------------|-----------|-----------|-----------|
##          243 |       639 |       390 |      1029 |
##              |     0.621 |     0.379 |     0.049 |
## -------------|-----------|-----------|-----------|
##          244 |       222 |       249 |       471 |
##              |     0.471 |     0.529 |     0.023 |
## -------------|-----------|-----------|-----------|
##          245 |       161 |        51 |       212 |
##              |     0.759 |     0.241 |     0.010 |
## -------------|-----------|-----------|-----------|
##          251 |        12 |         0 |        12 |
```

```
##              |     1.000 |     0.000 |     0.001 |
## -------------|-----------|-----------|-----------|
##          252 |         4 |         0 |         4 |
##              |     1.000 |     0.000 |     0.000 |
## -------------|-----------|-----------|-----------|
##          253 |       117 |        11 |       128 |
##              |     0.914 |     0.086 |     0.006 |
## -------------|-----------|-----------|-----------|
##          254 |       244 |        94 |       338 |
##              |     0.722 |     0.278 |     0.016 |
## -------------|-----------|-----------|-----------|
##          255 |       174 |       120 |       294 |
##              |     0.592 |     0.408 |     0.014 |
## -------------|-----------|-----------|-----------|
##          313 |         0 |        39 |        39 |
##              |     0.000 |     1.000 |     0.002 |
## -------------|-----------|-----------|-----------|
##          314 |         0 |        68 |        68 |
##              |     0.000 |     1.000 |     0.003 |
## -------------|-----------|-----------|-----------|
##          315 |         0 |        15 |        15 |
##              |     0.000 |     1.000 |     0.001 |
## -------------|-----------|-----------|-----------|
##          341 |         0 |         2 |         2 |
##              |     0.000 |     1.000 |     0.000 |
## -------------|-----------|-----------|-----------|
##          342 |      1267 |        65 |      1332 |
##              |     0.951 |     0.049 |     0.064 |
## -------------|-----------|-----------|-----------|
##          343 |       538 |       289 |       827 |
##              |     0.651 |     0.349 |     0.040 |
## -------------|-----------|-----------|-----------|
##          344 |       274 |       168 |       442 |
##              |     0.620 |     0.380 |     0.021 |
## -------------|-----------|-----------|-----------|
##          345 |       145 |        54 |       199 |
##              |     0.729 |     0.271 |     0.010 |
## -------------|-----------|-----------|-----------|
##          351 |         7 |         0 |         7 |
##              |     1.000 |     0.000 |     0.000 |
## -------------|-----------|-----------|-----------|
##          352 |         3 |         0 |         3 |
##              |     1.000 |     0.000 |     0.000 |
## -------------|-----------|-----------|-----------|
##          353 |       163 |        30 |       193 |
##              |     0.845 |     0.155 |     0.009 |
## -------------|-----------|-----------|-----------|
##          354 |       340 |       160 |       500 |
##              |     0.680 |     0.320 |     0.024 |
## -------------|-----------|-----------|-----------|
##          355 |       252 |       249 |       501 |
##              |     0.503 |     0.497 |     0.024 |
## -------------|-----------|-----------|-----------|
##          413 |         0 |        31 |        31 |
```

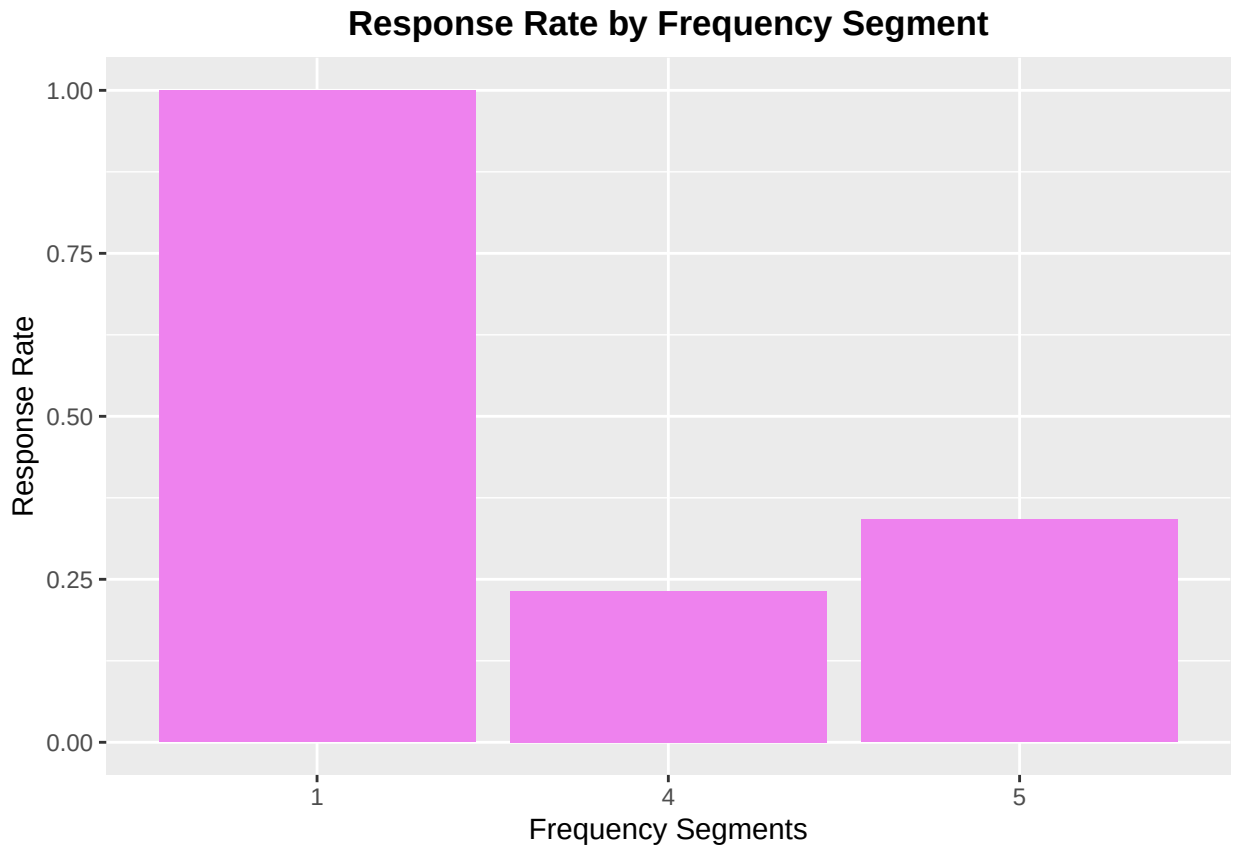
```
##              |     0.000 |     1.000 |     0.001 |
## -------------|-----------|-----------|-----------|
##          414 |         0 |        61 |        61 |
##              |     0.000 |     1.000 |     0.003 |
## -------------|-----------|-----------|-----------|
##          415 |         0 |        17 |        17 |
##              |     0.000 |     1.000 |     0.001 |
## -------------|-----------|-----------|-----------|
##          441 |         1 |         1 |         2 |
##              |     0.500 |     0.500 |     0.000 |
## -------------|-----------|-----------|-----------|
##          442 |      1168 |        61 |      1229 |
##              |     0.950 |     0.050 |     0.059 |
## -------------|-----------|-----------|-----------|
##          443 |       408 |       198 |       606 |
##              |     0.673 |     0.327 |     0.029 |
## -------------|-----------|-----------|-----------|
##          444 |       263 |       133 |       396 |
##              |     0.664 |     0.336 |     0.019 |
## -------------|-----------|-----------|-----------|
##          445 |       133 |        38 |       171 |
##              |     0.778 |     0.222 |     0.008 |
## -------------|-----------|-----------|-----------|
##          451 |        12 |         0 |        12 |
##              |     1.000 |     0.000 |     0.001 |
## -------------|-----------|-----------|-----------|
##          452 |         7 |         0 |         7 |
##              |     1.000 |     0.000 |     0.000 |
## -------------|-----------|-----------|-----------|
##          453 |       215 |        30 |       245 |
##              |     0.878 |     0.122 |     0.012 |
## -------------|-----------|-----------|-----------|
##          454 |       467 |       155 |       622 |
##              |     0.751 |     0.249 |     0.030 |
## -------------|-----------|-----------|-----------|
##          455 |       384 |       389 |       773 |
##              |     0.497 |     0.503 |     0.037 |
## -------------|-----------|-----------|-----------|
##          513 |         0 |        98 |        98 |
##              |     0.000 |     1.000 |     0.005 |
## -------------|-----------|-----------|-----------|
##          514 |         0 |       193 |       193 |
##              |     0.000 |     1.000 |     0.009 |
## -------------|-----------|-----------|-----------|
##          515 |         0 |        48 |        48 |
##              |     0.000 |     1.000 |     0.002 |
## -------------|-----------|-----------|-----------|
##          541 |         0 |         1 |         1 |
##              |     0.000 |     1.000 |     0.000 |
## -------------|-----------|-----------|-----------|
##          542 |       560 |        36 |       596 |
##              |     0.940 |     0.060 |     0.028 |
## -------------|-----------|-----------|-----------|
##          543 |       176 |       116 |       292 |
```

```
##              |     0.603 |     0.397 |     0.014 |
## -------------|-----------|-----------|-----------|
##          544 |       167 |       145 |       312 |
##              |     0.535 |     0.465 |     0.015 |
## -------------|-----------|-----------|-----------|
##          545 |       183 |        87 |       270 |
##              |     0.678 |     0.322 |     0.013 |
## -------------|-----------|-----------|-----------|
##          551 |        14 |         0 |        14 |
##              |     1.000 |     0.000 |     0.001 |
## -------------|-----------|-----------|-----------|
##          552 |         8 |         0 |         8 |
##              |     1.000 |     0.000 |     0.000 |
## -------------|-----------|-----------|-----------|
##          553 |       257 |        31 |       288 |
##              |     0.892 |     0.108 |     0.014 |
## -------------|-----------|-----------|-----------|
##          554 |       700 |       225 |       925 |
##              |     0.757 |     0.243 |     0.044 |
## -------------|-----------|-----------|-----------|
##          555 |       606 |       637 |      1243 |
##              |     0.488 |     0.512 |     0.059 |
## -------------|-----------|-----------|-----------|
## Column Total |     14465 |      6454 |     20919 |
## -------------|-----------|-----------|-----------|
##
##
```

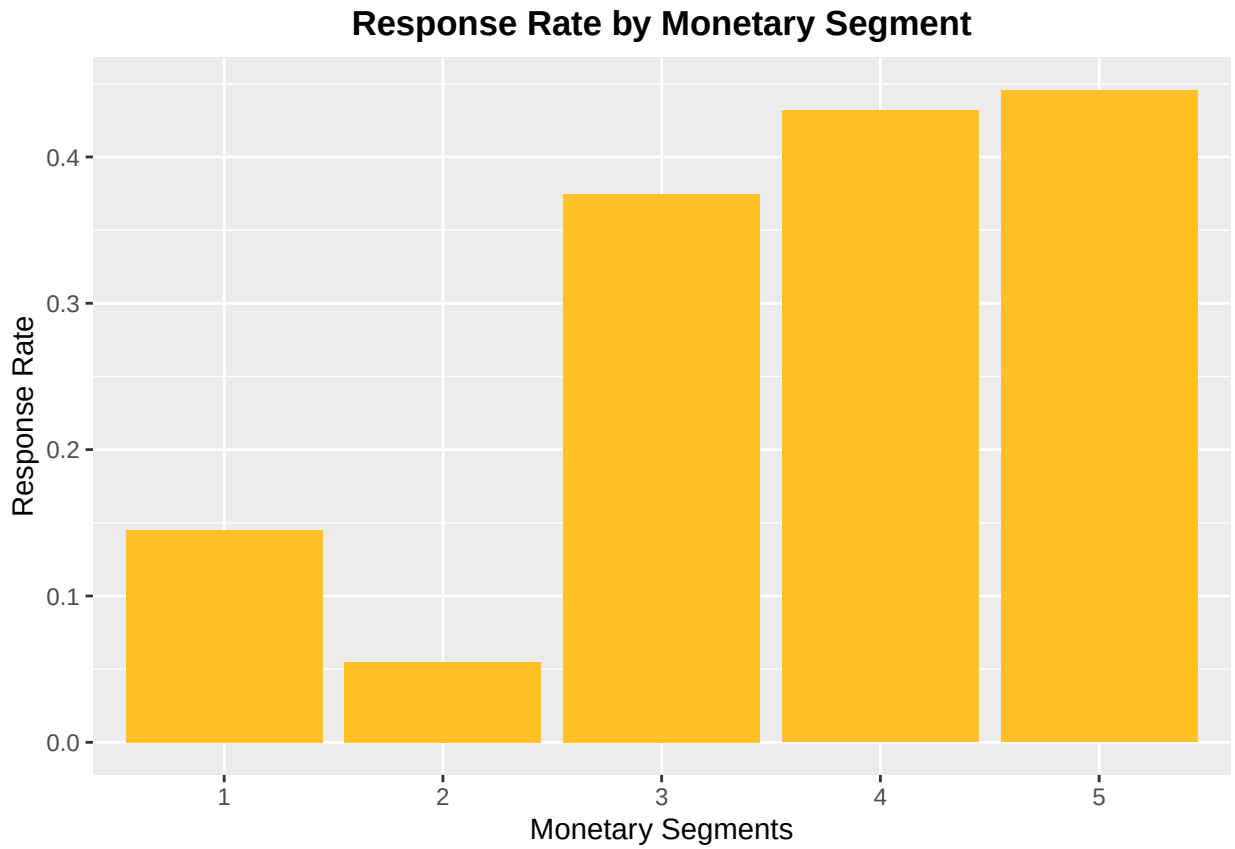
```r
plot_Recency <- data.frame(
  Recency_Segments = rownames(data_crosstab_Recency$prop.row),
  Response_rate = data_crosstab_Recency$prop.row[, 2]
)

ggplot(plot_Recency, aes(x = Recency_Segments, y = Response_rate,
                         fill = Recency_Segments)) +
  geom_bar(stat = "identity") +
  labs(title = "Response Rate by Recency Segment", x = "Recency Segments",
       y = "Response Rate") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold")) +
  scale_fill_manual(values = rep("springgreen3", nrow(plot_Recency))) +
  guides(fill = "none")
```



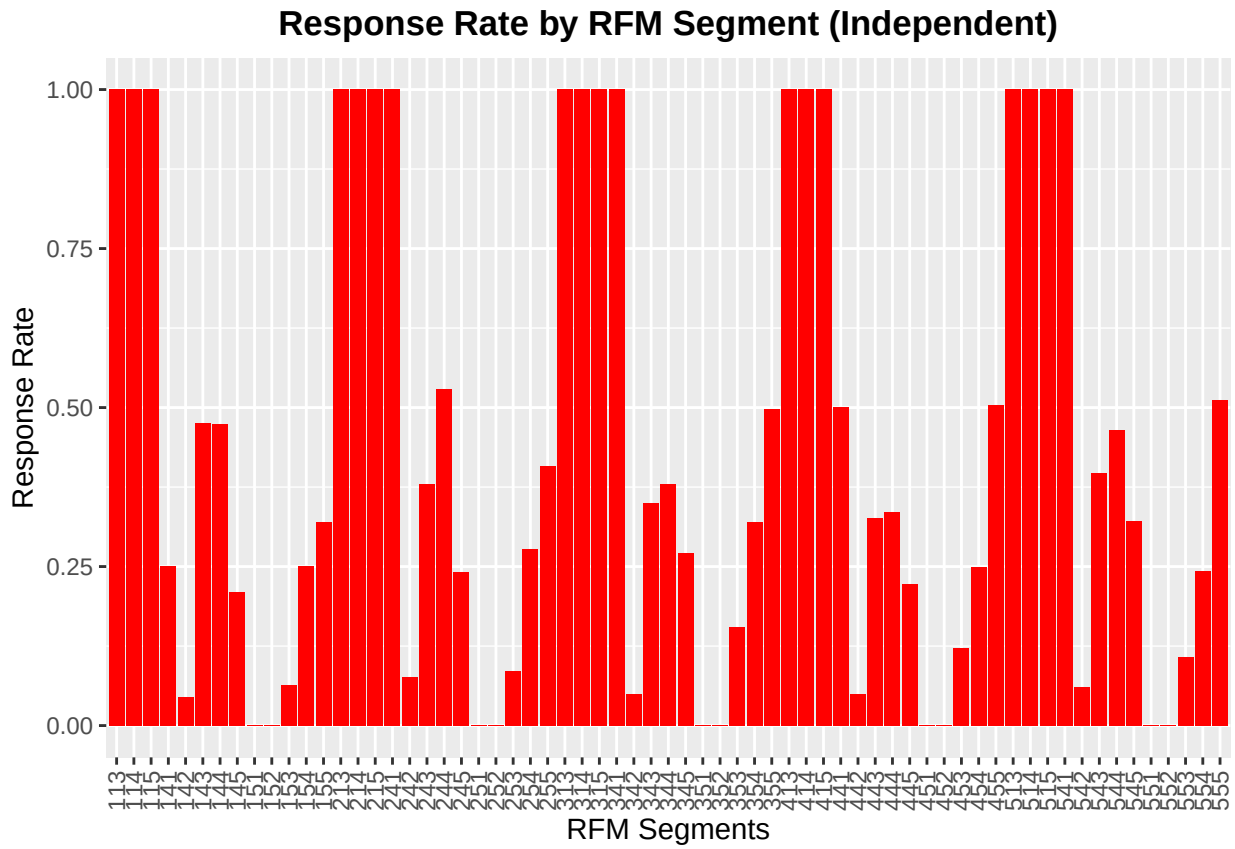

```
plot_Frequency <- data.frame(
  Frequency_Segments = rownames(data_crosstab_Frequency$prop.row),
  Response_rate = data_crosstab_Frequency$prop.row[, 2]
)

ggplot(plot_Frequency, aes(x = Frequency_Segments, y = Response_rate, fill = Frequency_Segments)) +
  geom_bar(stat = "identity") +
  labs(title = "Response Rate by Frequency Segment", x = "Frequency Segments",
       y = "Response Rate") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold")) +
  scale_fill_manual(values = rep("violet", nrow(plot_Frequency))) +
  guides(fill = "none")
```




```
plot_Monetary <- data.frame(
  Monetary_Segments = rownames(data_crosstab_Monetary$prop.row),
  Response_rate = data_crosstab_Monetary$prop.row[, 2]
)

ggplot(plot_Monetary, aes(x = Monetary_Segments, y = Response_rate, fill = Monetary_Segments)) +
  geom_bar(stat = "identity") +
  labs(title = "Response Rate by Monetary Segment", x = "Monetary Segments",
       y = "Response Rate") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold")) +
  scale_fill_manual(values = rep("goldenrod1", nrow(plot_Monetary))) +
  guides(fill = "none")
```




```
plot_data <- data.frame(
  RFM_Segments = rownames(data_crosstab_RFM$prop.row),
  Response_rate = data_crosstab_RFM$prop.row[, 2]
)

ggplot(plot_data, aes(x = RFM_Segments, y = Response_rate, fill = RFM_Segments)) +
  geom_bar(stat = "identity") +
  labs(title = "Response Rate by RFM Segment (Independent)", x = "RFM Segments",
       y = "Response Rate") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  scale_fill_manual(values = rep("red", nrow(plot_data))) +
  guides(fill = "none")
```




```
RFM_names = names(data_crosstab_RFM$t[,1])
# Remove names from data
non_buyers = unname(data_crosstab_RFM$t[,1], force = FALSE)
buyers     = unname(data_crosstab_RFM$t[,2], force = FALSE)

data.liftgains <-data.frame(RFM=RFM_names, ncustomers=non_buyers+buyers,
                            nbuyers=buyers)
data.liftgains$Percentage <- round(data.liftgains$nbuyers/
                                     data.liftgains$ncustomers * 100, 2)
data.liftgains <- data.liftgains[order(- data.liftgains$Percentage),]

best.RFM.seg <- data.liftgains[data.liftgains$Percentage == 100, ]
best.RFM.seg
```

```
##    RFM ncustomers nbuyers Percentage
## 1  113         67      67        100
## 2  114        239     239        100
## 3  115         43      43        100
## 14 213         63      63        100
## 15 214        154     154        100
## 16 215         23      23        100
## 17 241          4       4        100
## 27 313         39      39        100
## 28 314         68      68        100
## 29 315         15      15        100
```

```
## 30 341          2       2        100
## 40 413         31      31        100
## 41 414         61      61        100
## 42 415         17      17        100
## 53 513         98      98        100
## 54 514        193     193        100
## 55 515         48      48        100
## 56 541          1       1        100
```