Creative Gaming Group Assignment 2

Section 81 Meaghan Fudge Shiv Krishna

Install new packages

```
#install.packages("skimr")

#install.packages("ranger")
#devtools::install_github("imbs-hl/ranger", dependencies = TRUE)
library(ranger)
library(skimr)

#devtools::install_github("RcppCore/Rcpp", dependencies = TRUE)
```

```
#install.packages("remotes")
library(remotes)
#install_version("vip", version="0.3.2", upgrade="never")
#devtools::install_github("blakemcshane/kelloggmktg482", upgrade = "never", force = TRUE)
#install.packages("splitstackshape")
library(splitstackshape)
```

```
## Read in the data:
# use load("filename.Rdata") for .Rdata files
rm(list = ls())
load("creative_gaming.Rdata")

#head(intuit)
#training_first_100_rows <- intuit$training[1:100]
#print(head(intuit$training, 100))
#rm(training_first_100_rows)
```

Set seed

```
set.seed(1000)
```

Convert categorical variables into factors

```
cg_organic <- cg_organic %>%
mutate(AcquiredSpaceship = factor(AcquiredSpaceship),
AcquiredIonWeapon = factor(AcquiredIonWeapon),
PurchasedCoinPackSmall = factor(PurchasedCoinPackSmall),
PurchasedCoinPackLarge = factor(PurchasedCoinPackLarge),
UserNoConsole = factor(UserNoConsole),
UserHasOldOS = factor(UserHasOldOS),
converted = factor(converted))

cg_organic_control <- cg_organic_control %>%
```

```r
mutate(AcquiredSpaceship = factor(AcquiredSpaceship),
AcquiredIonWeapon = factor(AcquiredIonWeapon),
PurchasedCoinPackSmall = factor(PurchasedCoinPackSmall),
PurchasedCoinPackLarge = factor(PurchasedCoinPackLarge),
UserNoConsole = factor(UserNoConsole),
UserHasOldOS = factor(UserHasOldOS),
converted = factor(converted))

cg_ad_treatment <- cg_ad_treatment %>%
mutate(AcquiredSpaceship = factor(AcquiredSpaceship),
AcquiredIonWeapon = factor(AcquiredIonWeapon),
PurchasedCoinPackSmall = factor(PurchasedCoinPackSmall),
PurchasedCoinPackLarge = factor(PurchasedCoinPackLarge),
UserNoConsole = factor(UserNoConsole),
UserHasOldOS = factor(UserHasOldOS),
converted = factor(converted))
```

Assignment

Question 1

Prepare your data: Hint: Please visualize what you are doing by looking at the new data frames you create in each step. a) Create "Group 2" by sampling from cg_ad_treatment. You can use this syntax: cg_ad_random <- cg_ad_treatment[sample_random_30000,] b) Create a stacked data set for the uplift analysis by combining cg_organic_control (Group 1) and cg_ad_random (Group 2). You can use this syntax: expdata_stacked <- rbind(cg_organic_control %>% mutate(ad = 0), cg_ad_random %>% mutate(ad = 1)) c) Split the stacked dataset into a training and test dataset. You can use this syntax: set.seed(1234) split.index <- stratified(expdata_stacked, c("ad", "converted"), 0.7, bothSets=TRUE) expdata_stacked.train <- split.index[[1]] expdata_stacked.test <- split.index[[2]]

a)

```r
cg_ad_random <- cg_ad_treatment[sample_random_30000,]
```

b)

```r
expdata_stacked <- rbind(cg_organic_control %>% mutate(ad = 0),cg_ad_random
                         %>% mutate(ad = 1))
```

c)

```r
set.seed(1234)
split.index <- stratified(expdata_stacked, c("ad","converted"), 0.7,
                          bothSets=TRUE)
expdata_stacked.train <- split.index[[1]]
expdata_stacked.test <- split.index[[2]]
```

```r
expdata_stacked %>% tabyl(converted, ad) %>% adorn_percentages("all")
```

```
 converted          0          1
         0 0.47156667 0.43478333
         1 0.02843333 0.06521667
```

Question 2 Train an uplift model using random forests. Add the predicted scores for the treatment and control models to expdata_stacked.test and calculate the uplift score.

```r
#we exclude the treatment indicator - 'ad' (it is constant for each sample and
#it is not something we want to use to predict conversion).
```

```r
rfcon_treatment <- ranger(converted ~ GameLevel + NumGameDays +
                            NumGameDays4Plus +
NumInGameMessagesSent
+ NumFriends + NumFriendRequestIgnored + NumSpaceHeroBadges +
AcquiredSpaceship + AcquiredIonWeapon +
TimesLostSpaceship + TimesKilled + TimesCaptain + TimesNavigator +
PurchasedCoinPackSmall + PurchasedCoinPackLarge + NumAdsClicked +
DaysUser + UserNoConsole + UserHasOldOS, data=expdata_stacked.train%>%
  filter(ad==1) %>% select(-ad),
probability=TRUE, mtry=2, min.node.size=1)

rfcon_control <- ranger(converted ~ GameLevel + NumGameDays + NumGameDays4Plus +
NumInGameMessagesSent
+ NumFriends + NumFriendRequestIgnored + NumSpaceHeroBadges +
AcquiredSpaceship + AcquiredIonWeapon +
TimesLostSpaceship + TimesKilled + TimesCaptain + TimesNavigator +
PurchasedCoinPackSmall + PurchasedCoinPackLarge + NumAdsClicked +
DaysUser + UserNoConsole + UserHasOldOS, data=expdata_stacked.train%>%
  filter(ad==0) %>% select(-ad),
probability=TRUE, mtry=2, min.node.size=1)
```

For each customer in the test data (regardless of whether they were in the treatment group or the control group), we get Prob(conversion | with ad) and Prob(conversion | without ad) from the above models and then take their difference (i.e., the uplift score):

```r
expdata_stacked.test <- expdata_stacked.test %>%
mutate(pred_rfcon_treat = predict(rfcon_treatment, data=expdata_stacked.test,
                                  type="response")[[1]][,2],
       pred_rfcon_control = predict(rfcon_control,data=expdata_stacked.test,
                                    type="response")[[1]][,2],
       uplift_score = pred_rfcon_treat - pred_rfcon_control)
```

```r
expdata_stacked.test %>%
arrange(-uplift_score) %>%
select(converted, ad, pred_rfcon_treat, pred_rfcon_control, uplift_score) %>%
head()
```

```
   converted ad pred_rfcon_treat pred_rfcon_control uplift_score
1:         0  1        0.5995861         0.04582619    0.5537599
2:         0  0        0.5949630         0.05915099    0.5358120
3:         0  0        0.6134722         0.08555849    0.5279137
4:         0  0        0.6301030         0.10799387    0.5221092
5:         1  0        0.6008486         0.08319413    0.5176544
6:         0  0        0.5983574         0.08505642    0.5133010
```

Calculate the Uplift (%) and Incremental Uplift (%) for the uplift model (use 20 instead of the standard 10 groups) and plot performance metrics. Interpret theplots.

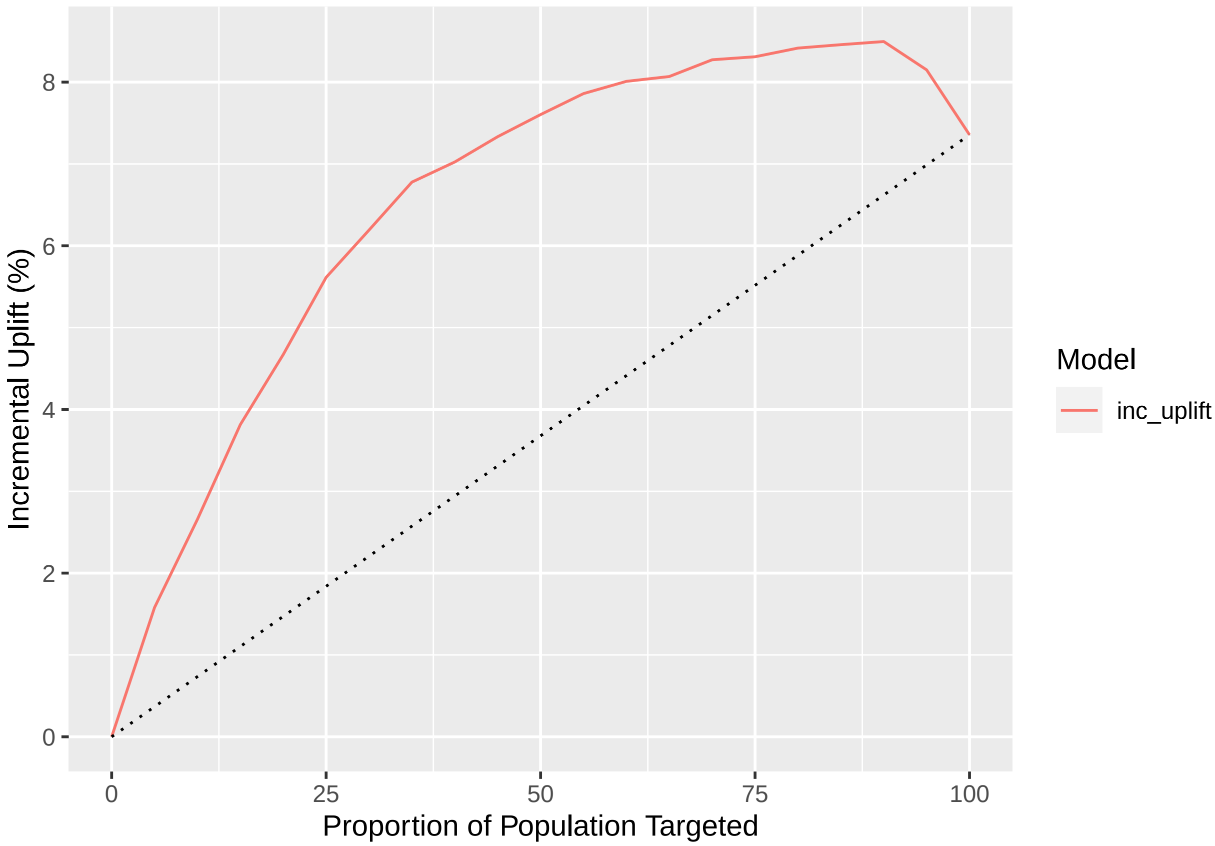
```r
PerfTable_uplift <- QiniTable(
expdata_stacked.test,
treat = "ad",
outcome = "converted",
prediction = "uplift_score",
nb.group = 20
)
PerfTable_uplift
```

```
   cum_per T_Y1  T_n C_Y1  C_n incremental_Y1 inc_uplift       uplift
1     0.05  197  450   69  566       142.1413   1.579348  0.315869651
2     0.10  328  900  110 1115       239.2108   2.657897  0.216429872
3     0.15  457 1350  141 1675       343.3582   3.815091  0.231309524
4     0.20  558 1800  170 2223       420.3482   4.670535  0.171524736
5     0.25  657 2250  186 2758       505.2596   5.613996  0.190093458
6     0.30  723 2700  203 3307       557.2607   6.191785  0.115701275
7     0.35  789 3150  217 3819       610.0134   6.777926  0.119322917
8     0.40  829 3600  232 4243       632.1581   7.023979  0.053511530
9     0.45  868 4050  243 4731       659.9784   7.333094  0.064125683
10    0.50  903 4500  251 5164       684.2742   7.603047  0.059302027
11    0.55  932 4950  255 5619       707.3604   7.859560  0.055653236
12    0.60  952 5400  259 6050       720.8264   8.009183  0.035163702
13    0.65  965 5850  265 6490       726.1325   8.068139  0.015252525
14    0.70  987 6300  268 6964       744.5531   8.272813  0.042559775
15    0.75  998 6750  275 7421       747.8652   8.309614  0.009127158
16    0.80 1012 7200  276 7806       757.4266   8.415851  0.028513709
17    0.85 1022 7650  282 8271       761.1730   8.457478  0.009318996
18    0.90 1071 8100  325 8592       764.6103   8.495670 -0.025067497
19    0.95 1139 8550  419 8833       733.4243   8.149159 -0.238930383
20    1.00 1174 9000  512 9000       662.0000   7.355556 -0.479108450
```

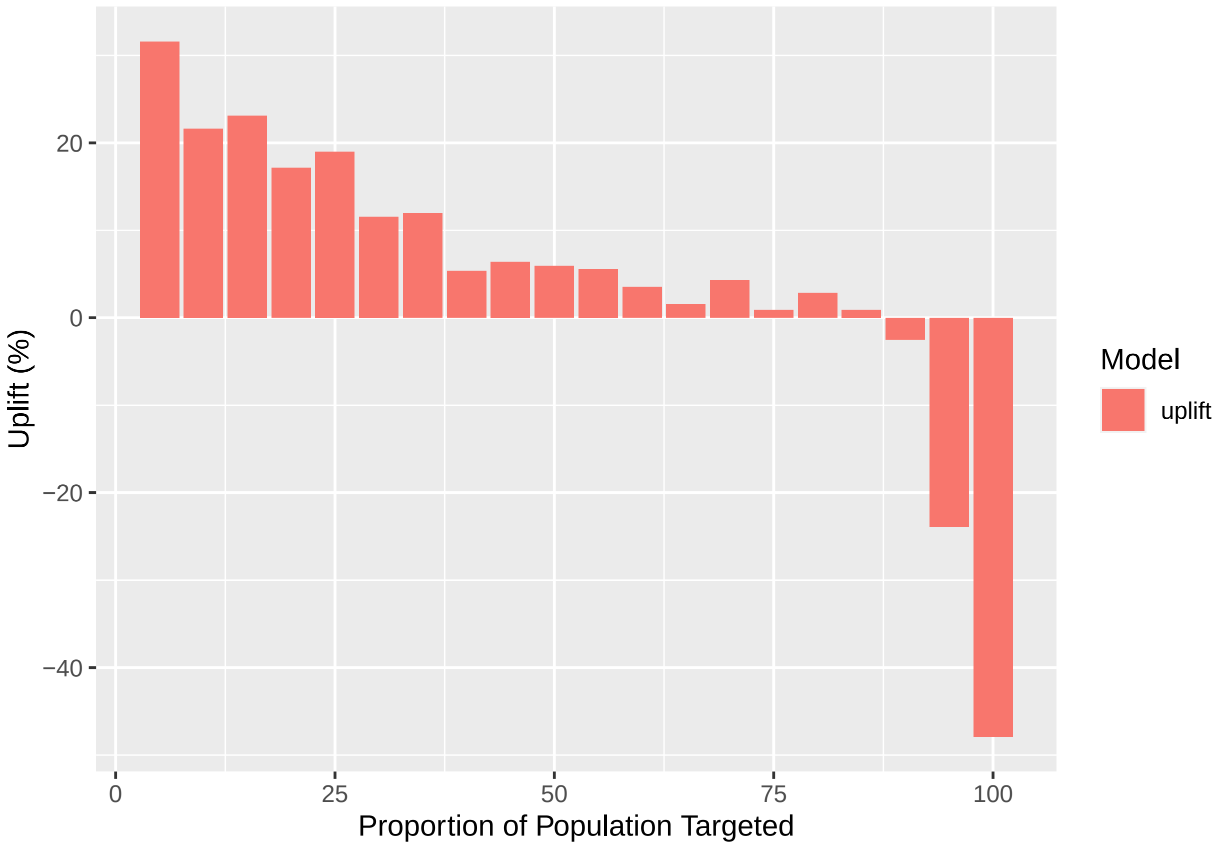
```r
PerfTable_uplift$incremental_Y1[5]
```

```
[1] 505.2596
```

```r
QiniCurve(PerfTable_uplift)
```




```
QiniBarPlot(PerfTable_uplift)
```



There's a concentrated proportion of do not disturbs. Additionally, there's a significant portion of the population where there's some impact of the ad, but it is small. Finally there's roughly 25-35% of the population that falls into >10% uplift which means there is a decent proportion that are potentially persuadable and may make sense for targeting.

Calculate the incremental profit you'd expect to make if you targeted the best 30,000 consumers of 120,000 using the uplift model. Hint: For every n-tile, the incremental_Y1 tells you how many incremental purchases were made when consumers up to that n-tile were targeted. To extrapolate correctly to picking the best 30,000 from 120,000, notice that there are a total of 9,000 consumers who got the ad in the test sample expdata_stacked.test.

```
revenue <- 14.99
cost <- 1.50

#find profit of 25% of 9000 from uplift
profit_expdata_uplift <- (revenue * PerfTable_uplift$incremental_Y1[5]) -
  (cost * PerfTable_uplift$T_n[5])

profit_expdata_uplift
```

```
[1] 4198.842
```

```
# Extrapolate to get the best 30000 out of 120000
profit_uplift <- profit_expdata_uplift* (30000 / PerfTable_uplift$T_n[5])
profit_uplift
```

```
[1] 55984.55
```

```r
cat( paste("The profit generated by targeting top 25% of customers
           based on uplift within the expdata.test is",
           profit_expdata_uplift, ". \n We then used this to calculate the \n
           profit generated when we target the top 30000 customers from \n
           the 120000 customers -- For \n",
           PerfTable_uplift$T_n[5],"customers,
           the profit was ",profit_expdata_uplift,
           "\n then for 30000 customers,
           the profit would be 30000 / ",PerfTable_uplift$T_n[5], " * " ,
           profit_expdata_uplift, " = $ \n " ,profit_uplift,  "\n"))
```

```
The profit generated by targeting top 25% of customers
           based on uplift within the expdata.test is 4198.84153009427 .
 We then used this to calculate the

           profit generated when we target the top 30000 customers from

           the 120000 customers -- For
 2250 customers,
           the profit was  4198.84153009427
 then for 30000 customers,
           the profit would be 30000 /  2250  *  4198.84153009427  = $
  55984.5537345903
```

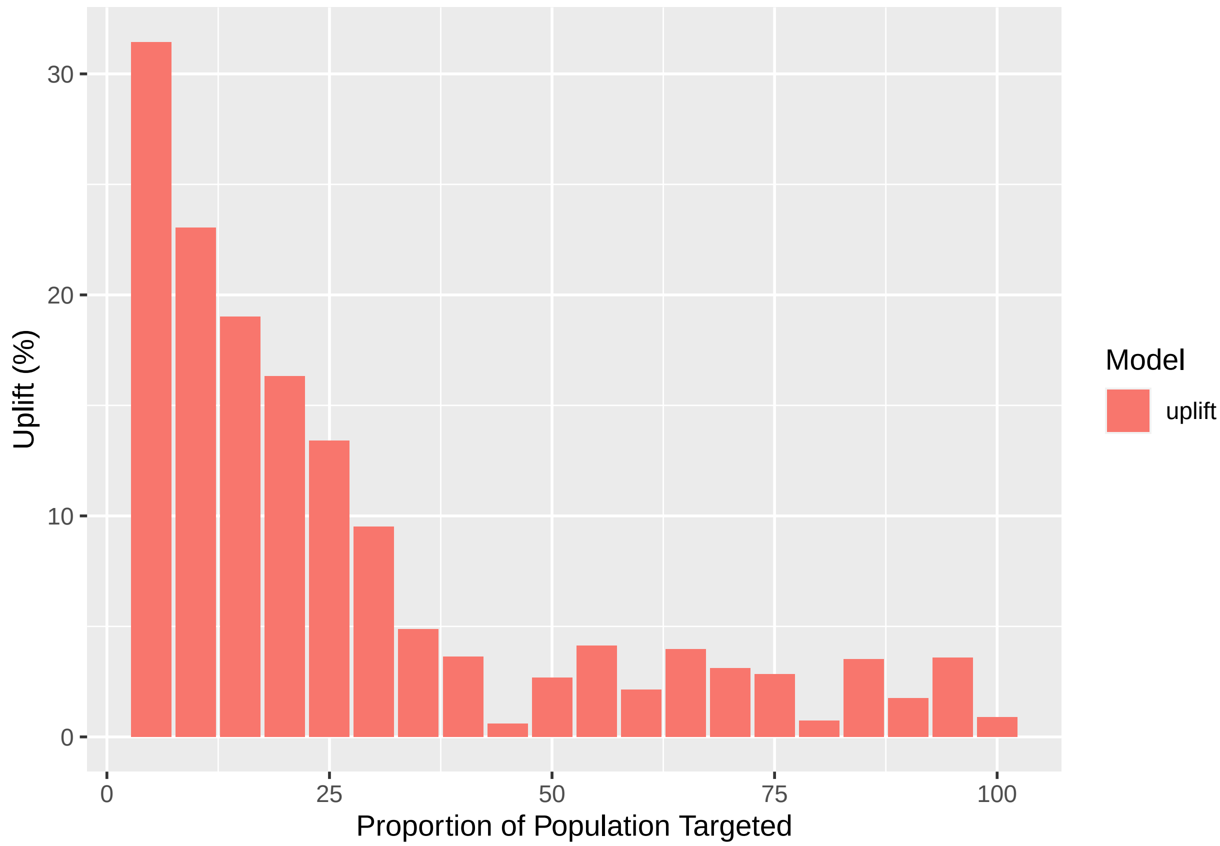
Calculate the Uplift (%) and Incremental Uplift (%) you would get if you used a propensity model (use 20 instead of the standard 10 groups). Compare the Uplift (%) performance metric between the uplift and propensity models. Interpret the difference. Hint: To compare the performance of the uplift and propensity models, use the functions QiniCurve() and QiniBarPlot().

```r
PerfTable_propensity <- QiniTable(
expdata_stacked.test,
treat = "ad",
outcome = "converted",
prediction = "pred_rfcon_treat",
nb.group = 20
)
PerfTable_propensity
```

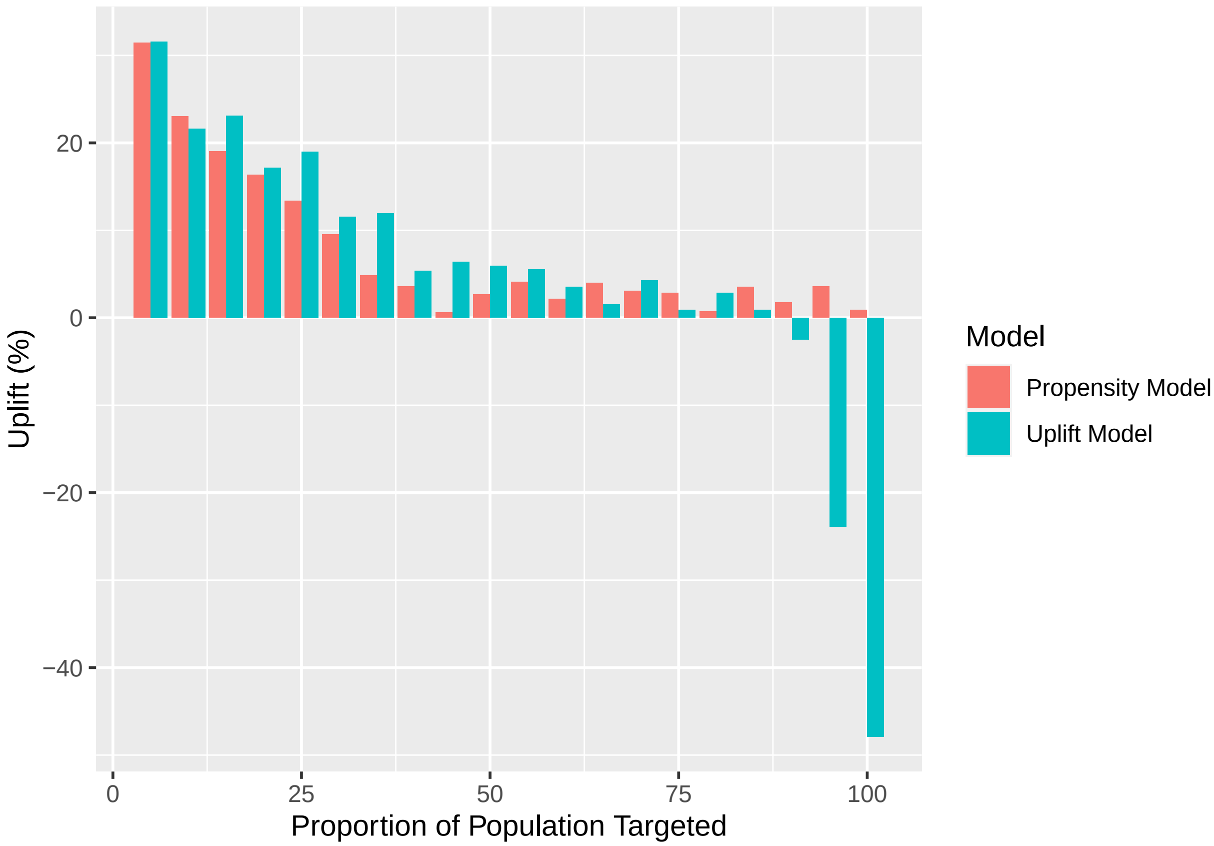
```
   cum_per T_Y1  T_n C_Y1  C_n incremental_Y1 inc_uplift      uplift
1     0.05  213  450   89  560       141.4821   1.572024 0.314404762
2     0.10  354  900  133 1091       244.2841   2.714268 0.230470810
3     0.15  477 1350  176 1608       329.2388   3.658209 0.190161186
4     0.20  587 1800  213 2064       401.2442   4.458269 0.163304094
5     0.25  692 2250  261 2547       461.4346   5.127051 0.133954451
6     0.30  777 2700  298 2942       503.5126   5.594584 0.095218003
7     0.35  851 3150  348 3374       526.1037   5.845597 0.048703704
8     0.40  912 3600  391 3807       542.2600   6.025112 0.036248396
9     0.45  960 4050  435 4244       544.8845   6.054273 0.005980168
10    0.50 1005 4500  466 4668       555.7712   6.175236 0.026886792
11    0.55 1040 4950  480 5052       569.6912   6.329902 0.041319444
12    0.60 1068 5400  498 5493       578.4315   6.427016 0.021405896
13    0.65 1090 5850  502 5931       594.8558   6.609509 0.039756469
14    0.70 1105 6300  503 6372       607.6836   6.752040 0.031065760
15    0.75 1120 6750  505 6782       617.3828   6.859809 0.028455285
16    0.80 1126 7200  508 7293       624.4780   6.938644 0.007462492
```

```
17     0.85 1143 7650  509 7693       636.8451   7.076056 0.035277778
18     0.90 1152 8100  510 8110       642.6289   7.140321 0.017601918
19     0.95 1170 8550  512 8581       659.8497   7.331663 0.035753715
20     1.00 1174 9000  512 9000       662.0000   7.355556 0.008888889
```

```
QiniBarPlot(PerfTable_propensity)
```




```
QiniBarPlot(PerfTable_uplift,
PerfTable_propensity,
modelnames = c("Uplift Model", "Propensity Model"))
```



the uplift model better places customers with high incrementality in earlier deciles. The incrementality is lower for the propensity model because it targets Persuadables and Sure Things whereas the uplift model targets only the former. That said, the propensity model does not do terribly here; this is because the customers who have the best propensity also tend to have the best uplift in this data:

```
cor(expdata_stacked.test$pred_rfcon_treat, expdata_stacked.test$uplift_score)
```

```
[1] 0.6302737
```

Using the incremental_Y1 column from the performance metric table created by QiniTable() for the propensity model, calculate the incremental profit you'd expect to make if you targeted the best 30,000 consumers of 120,000 using the propensity model. How much more money do you expect to make from using an uplift instead of a propensity model?

```
revenue <- 14.99
cost <- 1.50

#find profit of 25% of 9000 from uplift
profit_expdata_propensity <- (revenue * PerfTable_propensity$incremental_Y1[5])
- (cost * PerfTable_propensity$T_n[5])
```

```
[1] -3375
```

```
profit_expdata_propensity
```

```
[1] 6916.905
```

```
# Extrapolate to get the best 30000 out of 120000
profit_propensity <- profit_expdata_propensity* (30000 / PerfTable_uplift$T_n[5])
profit_propensity
```

```
[1] 92225.4
```

```
Net_profit_delta <- profit_uplift - profit_propensity
```

```
cat( paste("The profit generated by targeting top 25% of customers based on
           PROPENSITY within the expdata.test is", profit_expdata_propensity,
           ".\n  We then used this to calculate the profit generated when we \n
           target the top 30000 customers from the 120000 customers -- For \n  ", PerfTable_propensity$
           profit_expdata_propensity,"then for 30000 customers, the profit
           would be 30000 / ",PerfTable_propensity$T_n[5], " * " ,
           profit_expdata_propensity, " = $ \n" ,profit_propensity,  "\n"))
```

```
The profit generated by targeting top 25% of customers based on
           PROPENSITY within the expdata.test is 6916.90508833922 .
  We then used this to calculate the profit generated when we

           target the top 30000 customers from the 120000 customers -- For
   2250
 customers, the profit was
  6916.90508833922 then for 30000 customers, the profit
           would be 30000 /  2250  *  6916.90508833922  = $
 92225.4011778563
```

```
cat(paste("Net profit increase when using an Uplift model instead of \n
          Propensity model for targeting the top 25% customers \n
          (i.e 30000 of 120000) is ", Net_profit_delta, "\n"))
```

```
Net profit increase when using an Uplift model instead of

          Propensity model for targeting the top 25% customers

          (i.e 30000 of 120000) is  -36240.847443266
```

```
#Net_profit_delta
```

Part 2: Targeting the optimal percent of customers So far we have always targeted a 25% of model-selected customers (by picking the best 30,000 out of 120,000 customers). We now want to evaluate whether we should target fewer or more than 25% of customers. 1. What formula would you use to select which consumers to target using a propensity model where your goal is to maximize profits? What percentage of customers in the ad treatment group of expdata_stacked.test would you target using the propensity model?

```
revenue <- 14.99
cost <- 1.50

#we use for loop method to calculate incremental profits for each group
#-- as it applies the same profitability formula to each group separated
#by probability of conversion from the ad to consistently and comparably
#derive incremental profit in each model

# Create a vector to store incremental profits for each group
incremental_profits <- numeric(length(PerfTable_propensity$incremental_Y1))

# Calculate and print incremental profits for each group
for (i in 1:length(PerfTable_propensity$incremental_Y1)) {
  incremental_profit <- (revenue * PerfTable_propensity$incremental_Y1[i]) -
    (cost * PerfTable_propensity$T_n[i])
  incremental_profits[i] <- incremental_profit
```

```r
  cat(paste("Group", i, "- Incremental Profit: $",
            round(incremental_profit, 2), "\n"))
}
```

```
Group 1 - Incremental Profit: $ 1445.82
Group 2 - Incremental Profit: $ 2311.82
Group 3 - Incremental Profit: $ 2910.29
Group 4 - Incremental Profit: $ 3314.65
Group 5 - Incremental Profit: $ 3541.91
Group 6 - Incremental Profit: $ 3497.65
Group 7 - Incremental Profit: $ 3161.29
Group 8 - Incremental Profit: $ 2728.48
Group 9 - Incremental Profit: $ 2092.82
Group 10 - Incremental Profit: $ 1581.01
Group 11 - Incremental Profit: $ 1114.67
Group 12 - Incremental Profit: $ 570.69
Group 13 - Incremental Profit: $ 141.89
Group 14 - Incremental Profit: $ -340.82
Group 15 - Incremental Profit: $ -870.43
Group 16 - Incremental Profit: $ -1439.07
Group 17 - Incremental Profit: $ -1928.69
Group 18 - Incremental Profit: $ -2516.99
Group 19 - Incremental Profit: $ -2933.85
Group 20 - Incremental Profit: $ -3576.62
```

```r
# Find the group that provides the maximum incremental profit
max_profit_group <- which.max(incremental_profits)
cat(paste("Group", max_profit_group, "provides the maximum incremental
          profit when using the Propensity model.\n"))
```

```
Group 5 provides the maximum incremental
          profit when using the Propensity model.
```

```r
# Determine the percentage of customers to target (e.g., 25th percentile)
targeted_percent <- max_profit_group /
  length(PerfTable_propensity$incremental_Y1) * 100
cat(paste("Target approximately", round(targeted_percent, 2),
          "\n % of customers for maximum incremental profits based on \n
          the Propensity model. \n"))
```

```
Target approximately 25
 % of customers for maximum incremental profits based on

          the Propensity model.
```

What formula would you use to select which consumers to target using an uplift model where your goal is to maximize incremental profits. What percentage of customers in the ad treatment group of expdata_stacked.test would you target using the uplift model?

```r
revenue <- 14.99
cost <- 1.50

#we use for loop method to calculate incremental profits for each group --
#as it applies the same profitability formula to each group separated
#by probability of conversion from the ad to consistently and comparably
#derive incremental profit in each model
```

```r
# Create a vector to store incremental profits for each group
incremental_profits <- numeric(length(PerfTable_uplift$incremental_Y1))

# Calculate and print incremental profits for each group
for (i in 1:length(PerfTable_uplift$incremental_Y1)) {
  incremental_profit <- (revenue * PerfTable_uplift$incremental_Y1[i]) -
    (cost * PerfTable_uplift$T_n[i])
  incremental_profits[i] <- incremental_profit
  cat(paste("Group", i, "- Incremental Profit: $",
            round(incremental_profit, 2), "\n"))
}
```

```
Group 1 - Incremental Profit: $ 1455.7
Group 2 - Incremental Profit: $ 2235.77
Group 3 - Incremental Profit: $ 3121.94
Group 4 - Incremental Profit: $ 3601.02
Group 5 - Incremental Profit: $ 4198.84
Group 6 - Incremental Profit: $ 4303.34
Group 7 - Incremental Profit: $ 4419.1
Group 8 - Incremental Profit: $ 4076.05
Group 9 - Incremental Profit: $ 3818.08
Group 10 - Incremental Profit: $ 3507.27
Group 11 - Incremental Profit: $ 3178.33
Group 12 - Incremental Profit: $ 2705.19
Group 13 - Incremental Profit: $ 2109.73
Group 14 - Incremental Profit: $ 1710.85
Group 15 - Incremental Profit: $ 1085.5
Group 16 - Incremental Profit: $ 553.82
Group 17 - Incremental Profit: $ -65.02
Group 18 - Incremental Profit: $ -688.49
Group 19 - Incremental Profit: $ -1830.97
Group 20 - Incremental Profit: $ -3576.62
```

```r
# Find the group that provides the maximum incremental profit
max_profit_group <- which.max(incremental_profits)
cat(paste("Group", max_profit_group, "provides the maximum incremental
          profit when using the Uplift model.\n"))
```

```
Group 7 provides the maximum incremental
          profit when using the Uplift model.
```

```r
# Determine the percentage of customers to target (e.g., 25th percentile)
targeted_percent <- max_profit_group /
  length(PerfTable_uplift$incremental_Y1) * 100
cat(paste("Target approximately", round(targeted_percent, 2),
          "% of customers for maximum incremental profits
          based on the Uplift model.\n"))
```

```
Target approximately 35 % of customers for maximum incremental profits
          based on the Uplift model.
```

```r
## have to check this breakeven calculation
revenue <- 14.99
cost <- 1.50
breakeven <- cost/revenue
```

```r
#Get size of ad treatment group
expdata_stacked.test %>%
  filter(ad==1) %>%
  summarize(total = n())
```

```
  total
1  9000
```

```r
expdata_stacked.test %>%
  filter(ad==1) %>%
  mutate(target = 1*(pred_rfcon_treat >= breakeven)) %>%
  filter(target==1) %>%
  summarise(num_targeted = n(), frac_targeted = n()/9000)
```

```
  num_targeted frac_targeted
1         4643     0.5158889
```

Rounding the targeting percentage numbers you calculated in 1. and 2. to the nearest 5%, use the QiniTable() you calculated for the propensity and uplift models in Part 1 to calculate the incremental profits you would have obtained in the expdata_stacked.test dataset if you had targeted the optimal percentage of customers suggested by each model.

```r
#Propensity Model - Targeting 25%

revenue <- 14.99
cost <- 1.50

#find profit of 25% of 9000 from uplift
profit_expdata_propensity <- (revenue * PerfTable_propensity$incremental_Y1[5])
- (cost * PerfTable_propensity$T_n[5])
```

```
[1] -3375
```

```r
profit_expdata_propensity
```

```
[1] 6916.905
```

```r
#Uplift Model - Targeting 35%

revenue <- 14.99
cost <- 1.50

#find profit of 35% of 9000 from uplift
profit_expdata_uplift <- (revenue * PerfTable_uplift$incremental_Y1[7])
- (cost * PerfTable_uplift$T_n[7])
```

```
[1] -4725
```

```r
profit_expdata_uplift
```

```
[1] 9144.1
```

```r
Net_expdata_profit_delta <- profit_expdata_uplift - profit_expdata_propensity


cat( paste("Based on the optimal percentage of incremental users \n
           we found for Propensity Model and Uplift model, the profit \n
           generated from PROPENSITY model is", profit_expdata_propensity,
```

```
          "and Profit generated \n from UPLIFT model is \n  ", profit_expdata_uplift,".
          \n The Net delta profit is",Net_expdata_profit_delta,  "\n"))
```

```
Based on the optimal percentage of incremental users

          we found for Propensity Model and Uplift model, the profit

          generated from PROPENSITY model is 6916.90508833922 and Profit generated
 from UPLIFT model is
   9144.10018067557 .

 The Net delta profit is 2227.19509233635
```

Give two reasons for why one model beats the other in incremental profits

1) The uplift model better places customers with high incrementality in earlier deciles. The incrementality is lower for the propensity model because it targets Persuadables and Sure Things whereas the uplift model targets only the former. From above, Uplift model can target top 35% of consumers by better placing higher incrementality customers in the earlier deciles, whereas propensity model only target top 25%, thus losing almost 10% of the customers who could have been targeted and converted.
2) Similar to the example in class, it appears the uplift model better places customers with high incrementality in earlier deciles. However, the propensity model also correlates to this portion of the graph pretty closely, indicating those with higher uplift also in general have a higher likelihood of purchase. Where the propensity model appears to differ is detecting those who may have a negative impact from advertising. The propensity model indicates all uplift is positive where the uplift model indicates there are some individuals you definitely may want to avoid in advertising as their likelihood to purchase without an advertisement is higher.