ASSIGNMENT NAME

true true true

Preliminaries

Determine notebook defaults:

Load packages:

Read in the data:

```
# use load("filename.Rdata") for .Rdata files
load("/Users/dain/Programs/R_Projects/MKTG_482_HW5/creative_gaming.Rdata")

cg_organic <- cg_organic %>%
   mutate(converted=factor(converted))
cg_organic_control <- cg_organic_control %>%
   mutate(converted=factor(converted))
cg_ad_treatment <- cg_ad_treatment %>%
   mutate(converted=factor(converted))
```

Assignment answers

Part 1: Uplift Modeling Using Machine Learning (Random Forests)

Question 1

Prepare your data: Hint: Please visualize what you are doing by looking at the new data frames you create in each step.

```
cg_ad_random <- cg_ad_treatment[sample_random_30000,]

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

set.seed(1234)
split.index <- stratified(expdata_stacked, c("ad", "converted"), 0.7, bothSets=TRUE)

expdata_stacked.train <- split.index$SAMP1
expdata_stacked.test <- split.index$SAMP2

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

```
 ad         0          1
  0 0.4715667 0.02843333
  1 0.4347833 0.06521667
```

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

```
 ad         0          1
  0 0.4715714 0.02842857
  1 0.4347857 0.06521429
```

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

```
 ad         0          1
  0 0.4715556 0.02844444
  1 0.4347778 0.06522222
```

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.

Hint: Please see the handout "Random Forests in R.pdf" for how to run a random forest.

```
rf_treatment <- ranger(converted ~ . - ad,
                       data=expdata_stacked.train %>% filter(ad==1),
                       probability=TRUE, mtry=4, min.node.size=1)
rf_control <- ranger(converted ~ . - ad,
                     data=expdata_stacked.train %>% filter(ad==0),
                     probability=TRUE, mtry=4, min.node.size=1)

expdata_stacked.test <- expdata_stacked.test %>%
   mutate(
      pred_treat = predict(rf_treatment, data=expdata_stacked.test, type="response")[[1]][,2],
      pred_control = predict(rf_control, data=expdata_stacked.test, type="response")[[1]][,2],
      uplift_score = pred_treat - pred_control
   )
expdata_stacked.test %>%
   arrange(-uplift_score) %>%
   select(converted, ad, pred_treat, pred_control, uplift_score) %>%
   head()
```

```
  converted ad pred_treat pred_control uplift_score
1         0  1  0.7752152   0.03181502    0.7434002
2         0  0  0.7754119   0.06801639    0.7073955
3         1  0  0.7445429   0.04501156    0.6995314
4         0  0  0.7562667   0.07640000    0.6798667
5         1  1  0.7480350   0.09735931    0.6506757
6         0  1  0.7348289   0.08455832    0.6502706
```

Question 3

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

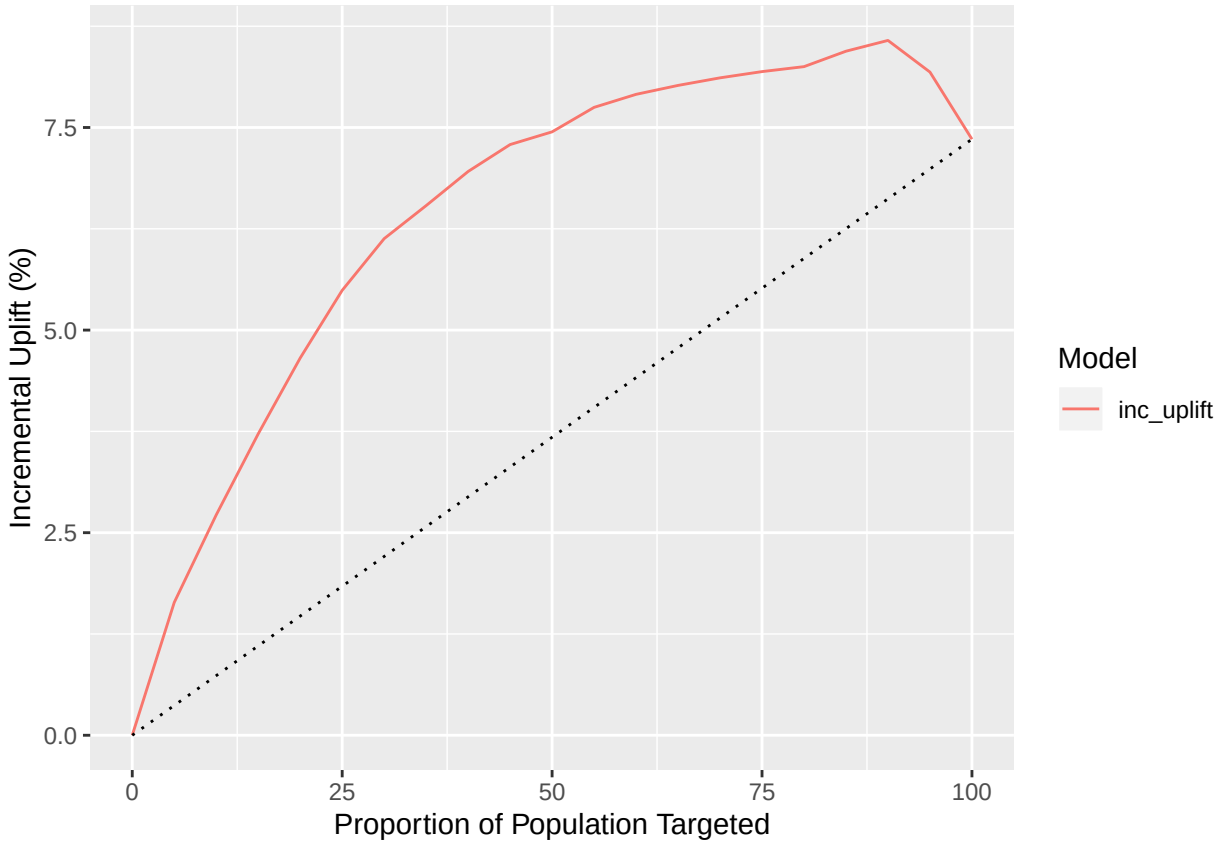
```
# expdata_stacked.test %>%
#   arrange(-uplift_score) %>%
#   select(converted, ad, pred_treat, pred_control, uplift_score) %>% head()

PerfTable_uplift <- QiniTable(
   expdata_stacked.test,
   treat = "ad",
   outcome = "converted",
   prediction = "uplift_score",
```

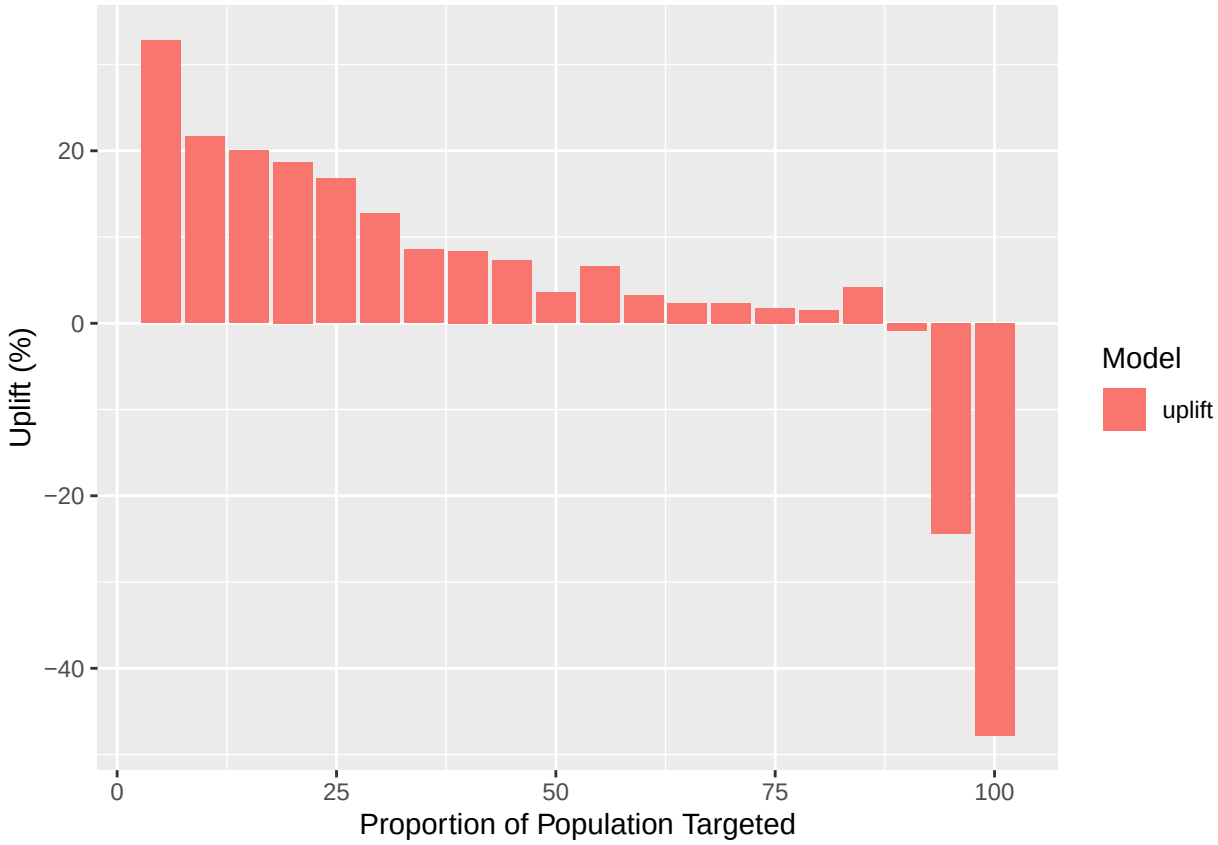
```
   nb.group = 20
)
PerfTable_uplift
```

```
   cum_per T_Y1  T_n C_Y1  C_n incremental_Y1 inc_uplift       uplift
1     0.05  197  450   62  564       147.5319   1.639243  0.327848700
2     0.10  328  900  103 1114       244.7864   2.719848  0.216565657
3     0.15  444 1350  134 1656       334.7609   3.719565  0.200582206
4     0.20  547 1800  157 2207       418.9529   4.655032  0.187146602
5     0.25  642 2250  180 2739       494.1358   5.490398  0.167878028
6     0.30  713 2700  197 3295       551.5736   6.128596  0.127202238
7     0.35  762 3150  208 3771       588.2530   6.536144  0.085779645
8     0.40  812 3600  223 4320       626.1667   6.957407  0.083788707
9     0.45  854 4050  231 4726       656.0419   7.289354  0.073628900
10    0.50  882 4500  242 5141       670.1735   7.446372  0.035716198
11    0.55  917 4950  247 5567       697.3754   7.748616  0.066040689
12    0.60  936 5400  252 6070       711.8155   7.909061  0.032281864
13    0.65  954 5850  260 6549       721.7508   8.019453  0.023298539
14    0.70  969 6300  264 6964       730.1717   8.113019  0.023694779
15    0.75  984 6750  271 7409       737.1043   8.190048  0.017602996
16    0.80  997 7200  277 7840       742.6122   8.251247  0.014967775
17    0.85 1018 7650  279 8265       759.7604   8.441783  0.041960784
18    0.90 1080 8100  327 8592       771.7249   8.574721 -0.009011213
19    0.95 1134 8550  410 8820       736.5510   8.183900 -0.244035088
20    1.00 1174 9000  512 9000       662.0000   7.355556 -0.477777778
```

```
QiniCurve(PerfTable_uplift)
```



```
QiniBarPlot(PerfTable_uplift)
```




```
"
Interpretation: Target ~85% of creative gaming players with ads to maximize uplift
"
```

```
[1] "\nInterpretation: Target ~85% of creative gaming players with ads to maximize uplift\n"
```

Question 4

Using the incremental_Y1 column from the performance metric table created by QiniTable(), calculate the incremental profit you 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_per <- 14.99
cost_per <- 1.50

# TBD: uplift calcs (compare against output from PerfTable_uplift)
# 147.5319 / 450                # ~0.3278 [OK]
# (244.7864 - 147.5319) / 450   # ~0.2165 [OK]
# (334.7609 - 244.7864) / 450   # ~0.2005 [OK]

# NOTE: T_n/C_n*C_Y1*revenue_per = "weighted control conv"
PerfTable_uplift <- PerfTable_uplift %>%
   mutate(inc_profit=revenue_per*incremental_Y1 - cost_per*T_n) %>%
```

```
   mutate(inc_profit_2=T_Y1*revenue_per - T_n*cost_per - T_n/C_n*C_Y1*revenue_per)

# PerfTable_uplift

upliftquarterPerc <- PerfTable_uplift %>% slice(5,)
upliftProfitFor30k <- upliftquarterPerc$inc_profit / (upliftquarterPerc$T_n/30000)

paste("Incremental Uplift over Nothing Profit: ", dollar(upliftProfitFor30k))
```

```
[1] "Incremental Uplift over Nothing Profit:  $53,761.28"
```

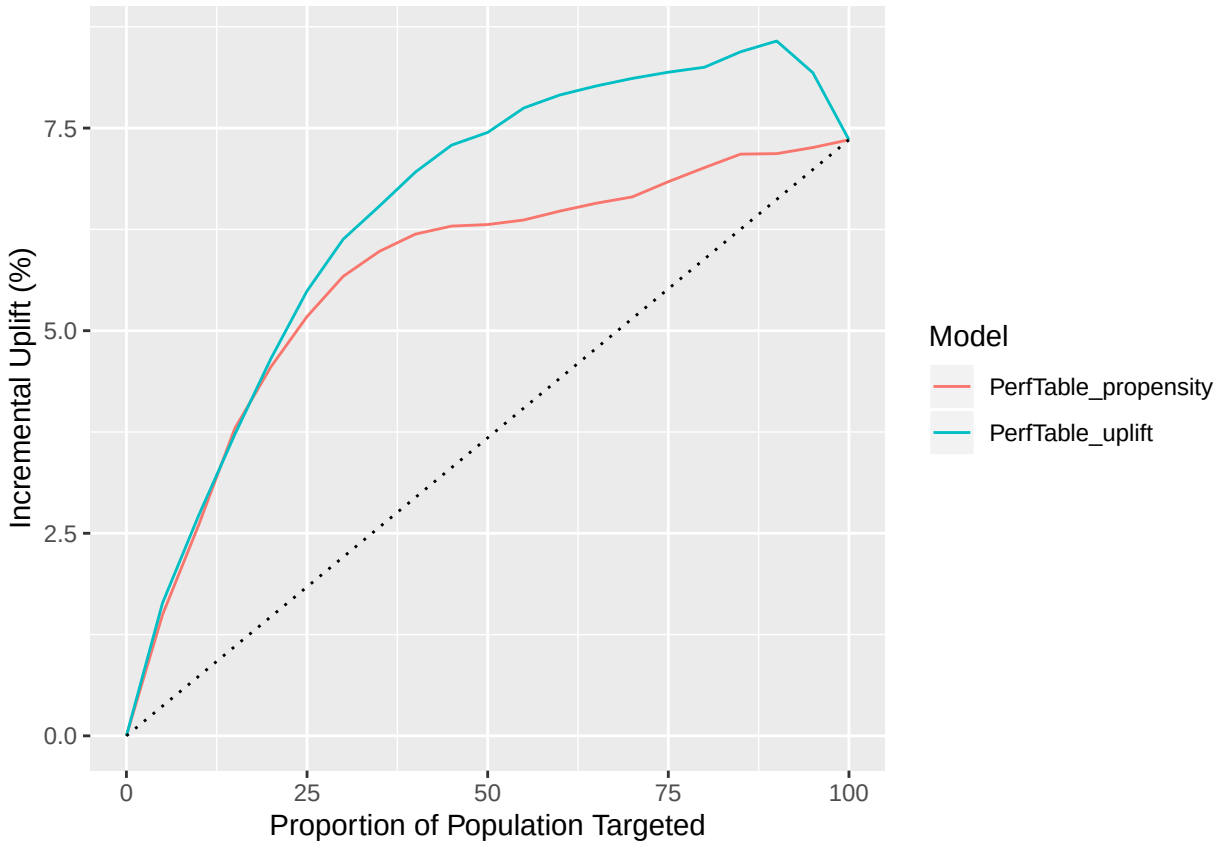
Question 5

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.

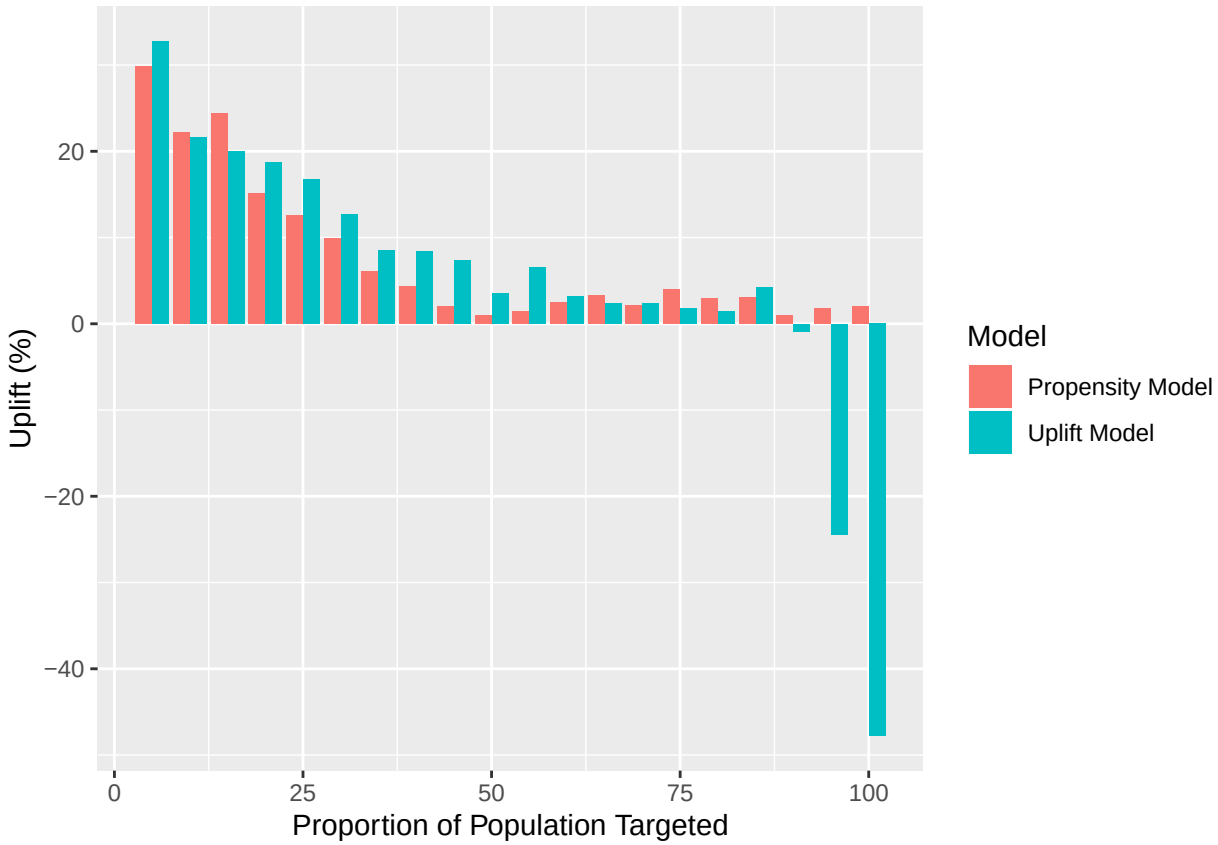
Hint: To compare the performance of the uplift and propensity models, use the functions QiniCurve() and QiniBarPlot()

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

QiniCurve(PerfTable_uplift, PerfTable_propensity)
```




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




```
# cor(expdata_stacked.test$pred_treat, expdata_stacked.test$uplift_score)
```

Question 6

Using the incremental_Y1 column from the performance metric table created by QiniTable() for the propensity model, calculate the incremental profit you 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?

```
# NOTE: T_n/C_n*C_Y1*revenue_per = "weighted control conv"
PerfTable_propensity <- PerfTable_propensity %>%
   mutate(inc_profit=revenue_per*incremental_Y1 - cost_per*T_n) %>%
   mutate(inc_profit_2=T_Y1*revenue_per - T_n*cost_per - T_n/C_n*C_Y1*revenue_per)

# PerfTable_propensity

propensity_quarterPerc <- PerfTable_propensity %>% slice(5,)
propensity_profit_30k <- propensity_quarterPerc$inc_profit / (propensity_quarterPerc$T_n/30000)

paste("Incremental Propensity over Nothing Profit: ", dollar(propensity_profit_30k))
```

```
[1] "Incremental Propensity over Nothing Profit:  $48,075.94"
```

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.

Question 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?

```
# TBD: Why not stay with the QiniTable results?
# propensityTarget <- PerfTable_propensity %>% slice(which.max(inc_profit))

propensityTarget <- expdata_stacked.test %>%
   mutate(ExpRev_prop=pred_treat*revenue_per)%>%
   filter(ExpRev_prop>cost_per)

cum_perc_prop <- nrow(propensityTarget)/nrow(expdata_stacked.test)

paste("Propensity profit is maximized at", percent(cum_perc_prop, 0.1))
```

```
[1] "Propensity profit is maximized at 51.2%"
```

Question 2

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?

```
# TBD: Why not stay with the QiniTable results?
# upliftTarget <- PerfTable_uplift %>% slice(which.max(inc_profit))

upliftTarget <- expdata_stacked.test %>%
   mutate(ExpRev_up=uplift_score*revenue_per)%>%
   filter(ExpRev_up>cost_per)

cum_perc_uplift <- nrow(upliftTarget)/nrow(expdata_stacked.test)

paste("Uplift profit is maximized at", percent(cum_perc_uplift, 0.1))
```

```
[1] "Uplift profit is maximized at 34.4%"
```

Question 3

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.

```
cum_perc_prop_r <- 0.5
cum_perc_uplift_r <- 0.35


propensity_target <- PerfTable_propensity %>% filter(cum_per==cum_perc_prop_r)
propensity_profit_30k <- propensity_target$inc_profit / (propensity_target$T_n/30000)

uplift_target <- PerfTable_uplift %>% filter(cum_per==cum_perc_uplift_r)
uplift_profit_30k <- uplift_target$inc_profit / (uplift_target$T_n/30000)

paste("Incremental Uplift over Propensity Profit: ", dollar(propensity_profit_30k))
```

```
[1] "Incremental Uplift over Propensity Profit:  $11,750.68"
```

Question 4

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

```
"
1. The uplift model allows you filter out lesser performing targets, targeting fewer people resulting ir
2. The uplift model may 'rebucket' previously lower-performing targets, allowing you to target differen
"
```

```
[1] "\n1. The uplift model allows you filter out lesser performing targets, targeting fewer people resul
```