Book purchase prediction

Funalytics - Anisha, Jamie, Lindsay, Spencer, Veronica

1/21/2018

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```r
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.3.2
```

```r
library(MASS)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.2
```

```r
library(car)
```

```
## Warning: package 'car' was built under R version 3.3.2
```

```r
#source("util.R")
```

Overall approach

1. Read in data and run EDA, focus on categories and price and quantity

2. Feature engineering:

a. create RFM features
b. additional features based on EDA findings
c. merge with booktrain data for additional EDA, and see if features need transformation for better linear relationships with logtargdol

3. Model fitting - regressions

a. baseline model fitting
b. additional tries by adding/removing features

4. Model fitting - logistic and regression

a. train regressions model based on those whose logtargdol >0, apply stepwise to select final subset of vars: log(monetary_avg + 1), log(avg_ord + 1), dummy vars on cat19 and cat20
b. train logistic model based on buyer or not buyer (logtargdol >0 buyer)
c. multiple a * b for final predicted logtargdol

Findings & Conclusion

During feature creation, certain book categories seemed to have an association with a customer making another purchase. Therefore, indicator variables were added to flag whether a customer made a purchase or not for categories 17, 19, and 20.

For the regression model, we saw that numerical variables around the actual purchase amount were significant predictors for how much a customer would spend on their next order (e.g. average price of an item, average order size), along with the indicator variables for customers who made purchases in book categories 19 & 20.

For the logistic model, numerical variables which described a customers purchasing behavior (e.g. frequency of orders and purchase rate) along with the indicator variable for customers purchasing books in category 20 were significant predictors of whether the customer would make a next purchase.

1. Reading data and describe

```
#read orders
dat = read.csv("data/orders.csv")
dat$orddt = as.Date(dat$orddate, "%d%b%Y")
```

```
## Warning in strptime(x, format, tz = "GMT"): unknown timezone 'default/
## America/Chicago'
```

```
dat$orddate = NULL
#head(dat)
str(dat)
```

```
## 'data.frame':    627955 obs. of  6 variables:
##  $ id      : int  914 914 914 914 914 914 914 914 914 914 ...
##  $ ordnum  : int  314037 314037 499719 499719 499719 499719 499719 638467 638467 638467 ...
##  $ category: int  20 20 36 20 31 12 20 31 20 20 ...
##  $ qty     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ price   : num  9.2 10.2 10.17 10.2 6.14 ...
##  $ orddt   : Date, format: "2009-12-02" "2009-12-02" ...
```

```
dim(dat)
```

```
## [1] 627955      6
```

```
#min date = "2007-11-04"
min(dat$orddt)
```

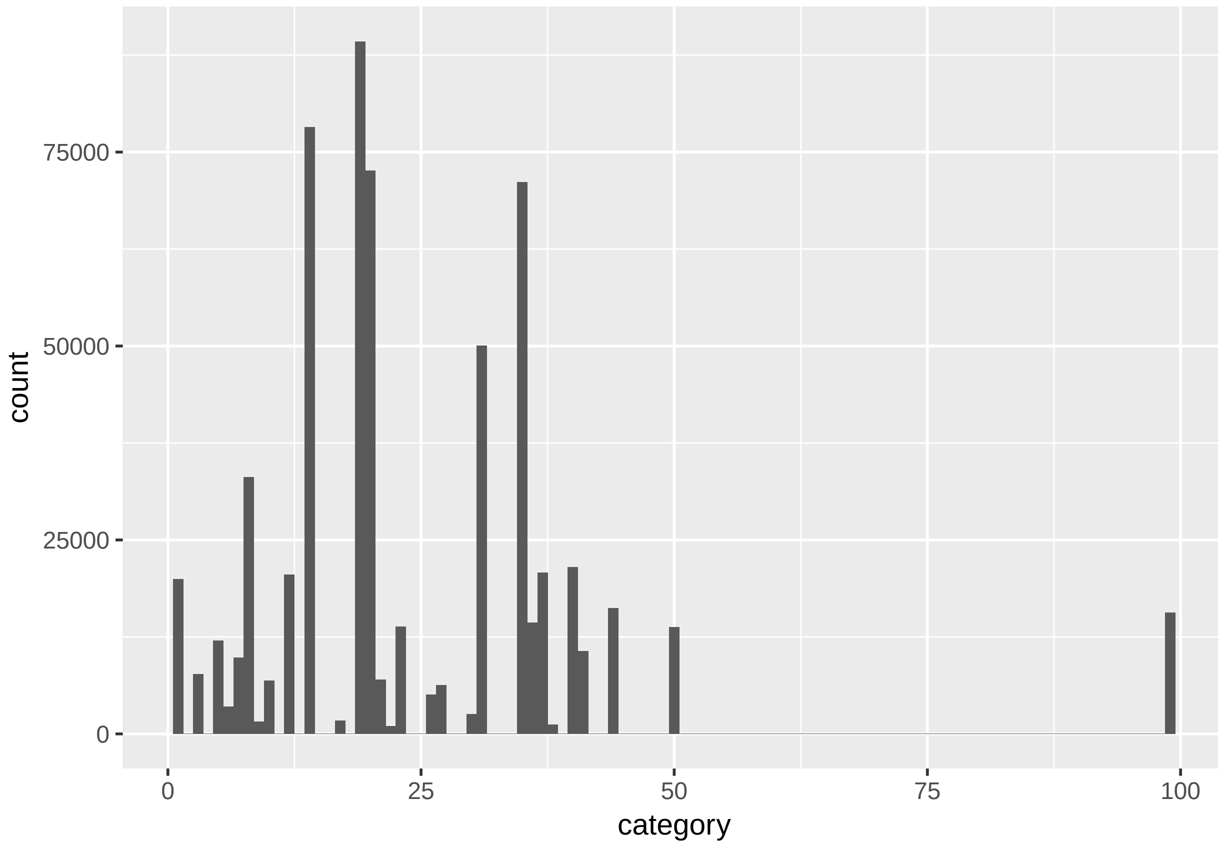
```
## [1] "2007-11-04"
```

```
#max date = "2014-07-31"
max(dat$orddt)
```

```
## [1] "2014-07-31"
```

Initial EDA and data checks on orders a. qualitative var - category => category 99 has some oddities (most qty with price - $0, max price = 1533) b. quantitative vars - qty,price

```
missing = dat[!complete.cases(dat),] #no missing value

#category frequency
ggplot(data=dat, aes(x=category)) + geom_histogram(binwidth = 1)
```




```r
#category by price (most $)
result = tapply(dat$price, dat$category,mean)
sort_price = result[order(result)] #category 17 (art prints) is $$$
art_collect = dat[dat$category==17,] #these people buy at most 3 items

#category by Q (most popular category)
result2 = tapply(dat$qty, dat$category,mean)
sort_q = result2[order(result2)] #note - cat99 nonbooks has the highest avg
result3 = tapply(dat$qty, dat$category,median)
sort_q3 = result3[order(result3)] #median is all 1
result4 = tapply(dat$qty, dat$category,max)
sort_q4 = result4[order(result4)] #category 99 is nonbooks, ID 8070857 has price =0, Q = max.

#category by price * Q (most popular category)
qp = tapply(dat$qty * dat$price, dat$category,mean)
sort_qp = qp[order(qp)] #align with expectation - 17 has the largest avg order size

qpm = tapply(dat$qty * dat$price, dat$category,max)
sort_qpm = qpm[order(qpm)] #8,14,35,37 have the max one-time order amounts, >$140k; makes sense, 37 is
```

```r
#Descriptive stats
summary(dat[,-1])
```

```
##      ordnum           category          qty               price        
##  Min.   :   1012   Min.   : 1.00   Min.   :     0.00   Min.   :   0.000  
##  1st Qu.: 360118   1st Qu.:14.00   1st Qu.:     1.00   1st Qu.:   5.113  
##  Median : 670449   Median :20.00   Median :     1.00   Median :   8.666  
```

```
##  Mean   : 646013   Mean   :24.76   Mean   :     1.55   Mean   :  11.215
##  3rd Qu.: 945367   3rd Qu.:35.00   3rd Qu.:     1.00   3rd Qu.:  12.731
##  Max.   :1191704   Max.   :99.00   Max.   :134872.00   Max.   :3834.688
##      orddt
##  Min.   :2007-11-04
##  1st Qu.:2010-03-03
##  Median :2011-11-08
##  Mean   :2011-09-11
##  3rd Qu.:2013-05-12
##  Max.   :2014-07-31
```

```
#investigate items with $0 in price
percent_price0 = count(dat[dat$price == 0,])/count(dat)

#The majority of items with 0 price are non-books, add flag to indicate: if category = 99, book = 0
dat$book = 0
dat$book[dat$category!=99]=1
table(dat$book)
```

```
##
##      0      1
##  15615 612340
```

2. feature engineer -recency: max/min time since last purchase, indicates inactivity -frequency: count of previous behaviors, indicates loyalty -monetary: sum/total spend of $ or time over a past period -time of file: time since first purchase (min/max)

```
# do simple roll up
x = dat %>%
 group_by(id) %>%
 summarise(f=n(),
           # ORIGINAL FEATURES, ADDED BY JAMIE
           recency_first = as.numeric(as.Date('2014-08-01') - min(orddt)), #time since first purchase -
           recency_last = as.numeric(as.Date('2014-08-01') - max(orddt)), #time since last purchase - r
           date_duration = recency_first - recency_last, #time between 1st and last purchases
           p_qty = sum(qty), #number of items
           frequency_ord = n_distinct(ordnum), #number of distinct orders, which <= f
           monetary_tot = sum(price * qty), #total spent
           monetary_avg = mean(price), #how expensive is each ordered item

           # FEATURES ADDED BY SPENCER
           count_cat = n_distinct(category) #number of distinct categories ordered
          )%>%
  dplyr::select(id, recency_first, recency_last, date_duration, p_qty, frequency_ord, monetary_tot, mone
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.2
```

```
#head(x)
dim(x)
```

```
## [1] 33355    10
```

Additional features

```
# ADDED BY JAMIE
#avg order size
x$avg_ord = x$monetary_tot/x$frequency_ord
```

```r
#purchase rate = purchases/period
x$prate = x$frequency_ord/x$recency_first

# ADDED BY SPENCER
#diversity of order
x$catrate = x$count_cat/x$frequency_ord
x$prate2 = x$frequency_ord/(x$date_duration + 1)

# ADDED BY ANISHA
#dummy variable - ordered category 20
cat20 = dat %>%
  filter(category == 20) %>%
  distinct(id) %>%
  mutate(cat20 = 1)
x = left_join(x,cat20,by="id")
x$cat20[is.na(x$cat20)] = 0

#dummy variable - ordered category 19
cat19 = dat %>%
  filter(category == 19) %>%
  distinct(id) %>%
  mutate(cat19 = 1)
x = left_join(x,cat19,by="id")
x$cat19[is.na(x$cat19)] = 0

#dummy variable - ordered category 17
cat17 = dat %>%
  filter(category == 17) %>%
  distinct(id) %>%
  mutate(cat17 = 1)
x = left_join(x,cat17,by="id")
x$cat17[is.na(x$cat17)] = 0

#check predictors cor
cor_mat = cor(x[2:14])
cor_mat > 0.6 #f & freq_ord are colinear as expected, avg_ord and monetary_tot
```

```
##               recency_first recency_last date_duration p_qty frequency_ord
## recency_first          TRUE        FALSE          TRUE FALSE         FALSE
## recency_last          FALSE         TRUE         FALSE FALSE         FALSE
## date_duration          TRUE        FALSE          TRUE FALSE          TRUE
## p_qty                 FALSE        FALSE         FALSE  TRUE         FALSE
## frequency_ord         FALSE        FALSE          TRUE FALSE          TRUE
## monetary_tot          FALSE        FALSE         FALSE FALSE         FALSE
## monetary_avg          FALSE        FALSE         FALSE FALSE         FALSE
## count_cat             FALSE        FALSE          TRUE FALSE          TRUE
## f                     FALSE        FALSE         FALSE FALSE          TRUE
## avg_ord               FALSE        FALSE         FALSE FALSE         FALSE
## prate                 FALSE        FALSE         FALSE FALSE         FALSE
## catrate               FALSE        FALSE         FALSE FALSE         FALSE
## prate2                FALSE        FALSE         FALSE FALSE         FALSE
##               monetary_tot monetary_avg count_cat     f avg_ord prate
## recency_first        FALSE        FALSE     FALSE FALSE   FALSE FALSE
## recency_last         FALSE        FALSE     FALSE FALSE   FALSE FALSE
```

```
## date_duration        FALSE        FALSE      TRUE FALSE   FALSE FALSE
## p_qty                FALSE        FALSE     FALSE FALSE   FALSE FALSE
## frequency_ord        FALSE        FALSE      TRUE  TRUE   FALSE FALSE
## monetary_tot          TRUE        FALSE     FALSE FALSE    TRUE FALSE
## monetary_avg         FALSE         TRUE     FALSE FALSE   FALSE FALSE
## count_cat            FALSE        FALSE      TRUE  TRUE   FALSE FALSE
## f                    FALSE        FALSE      TRUE  TRUE   FALSE FALSE
## avg_ord               TRUE        FALSE     FALSE FALSE    TRUE FALSE
## prate                FALSE        FALSE     FALSE FALSE   FALSE  TRUE
## catrate              FALSE        FALSE     FALSE FALSE   FALSE FALSE
## prate2               FALSE        FALSE     FALSE FALSE   FALSE FALSE
##               catrate prate2
## recency_first   FALSE  FALSE
## recency_last    FALSE  FALSE
## date_duration   FALSE  FALSE
## p_qty           FALSE  FALSE
## frequency_ord   FALSE  FALSE
## monetary_tot    FALSE  FALSE
## monetary_avg    FALSE  FALSE
## count_cat       FALSE  FALSE
## f               FALSE  FALSE
## avg_ord         FALSE  FALSE
## prate           FALSE  FALSE
## catrate          TRUE  FALSE
## prate2          FALSE   TRUE
```

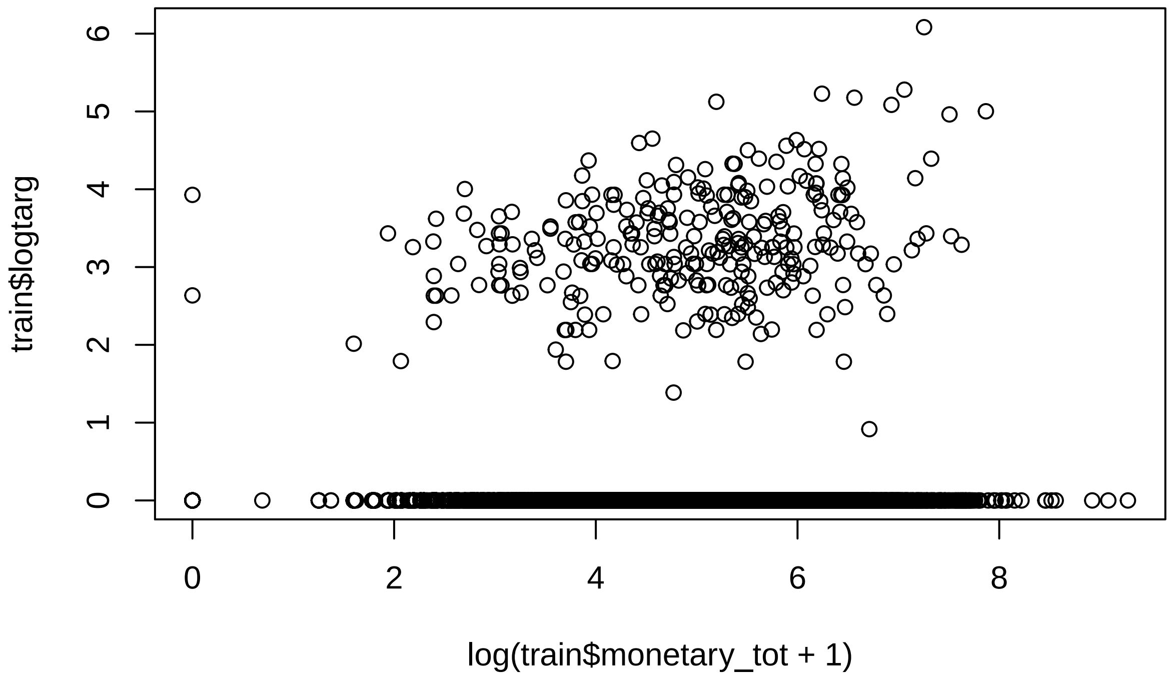
```r
#f, date_duration, recency_first, frequency_ord, count_cat have high correlation
```

```r
# read in dependent variable
y = read.csv("data/booktrain.csv")
#head(y)

#Left join booktrain table with orders, add a flag on buyer or not
all = left_join(x,y,by="id")
all$responseflag = ifelse(all$logtarg > 0, 1, 0)
dim(all)
```

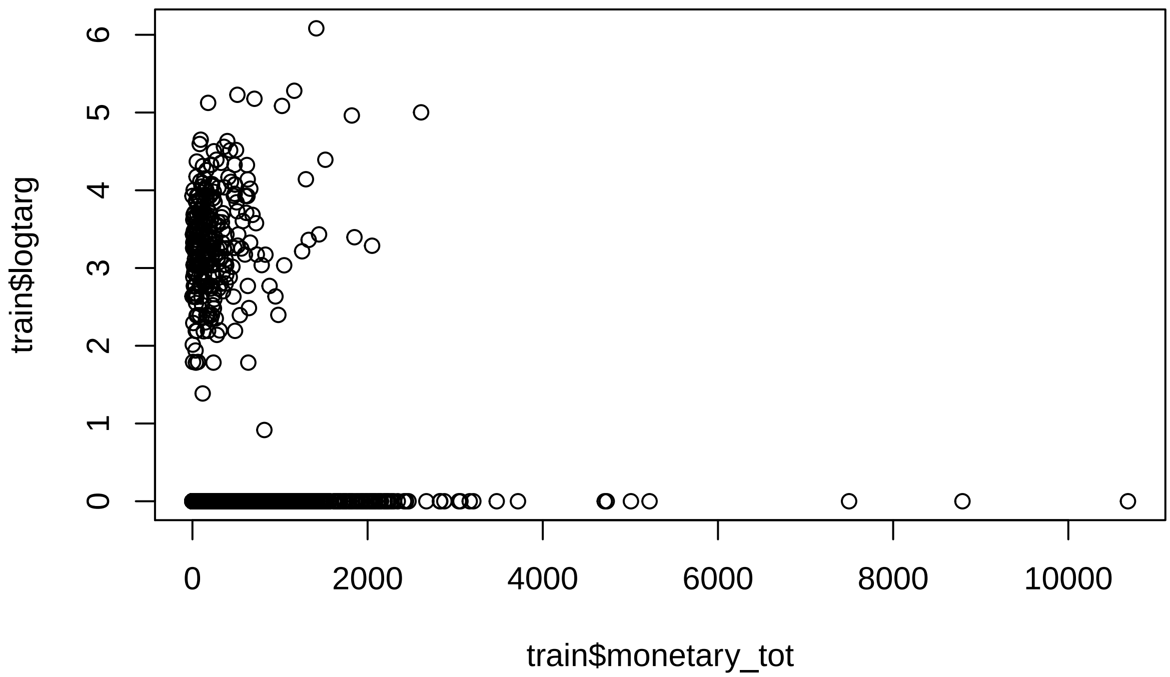
```
## [1] 33355    19
```

Variable transformation based on EDA - Create log transormation for F and M because of right skew

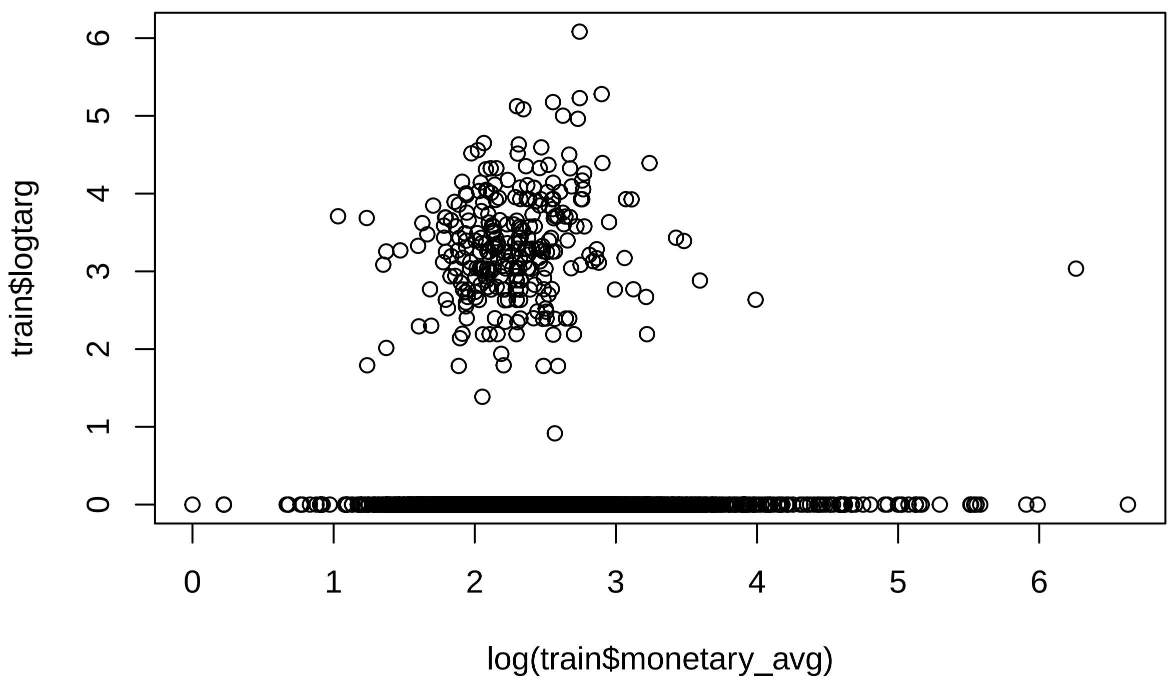
```r
train = all[!is.na(all$logtarg),] #8224 obs instead of 8311

#plot(log(train$monetary_tot), train$logtarg)
plot(log(train$monetary_tot +1), train$logtarg)
```



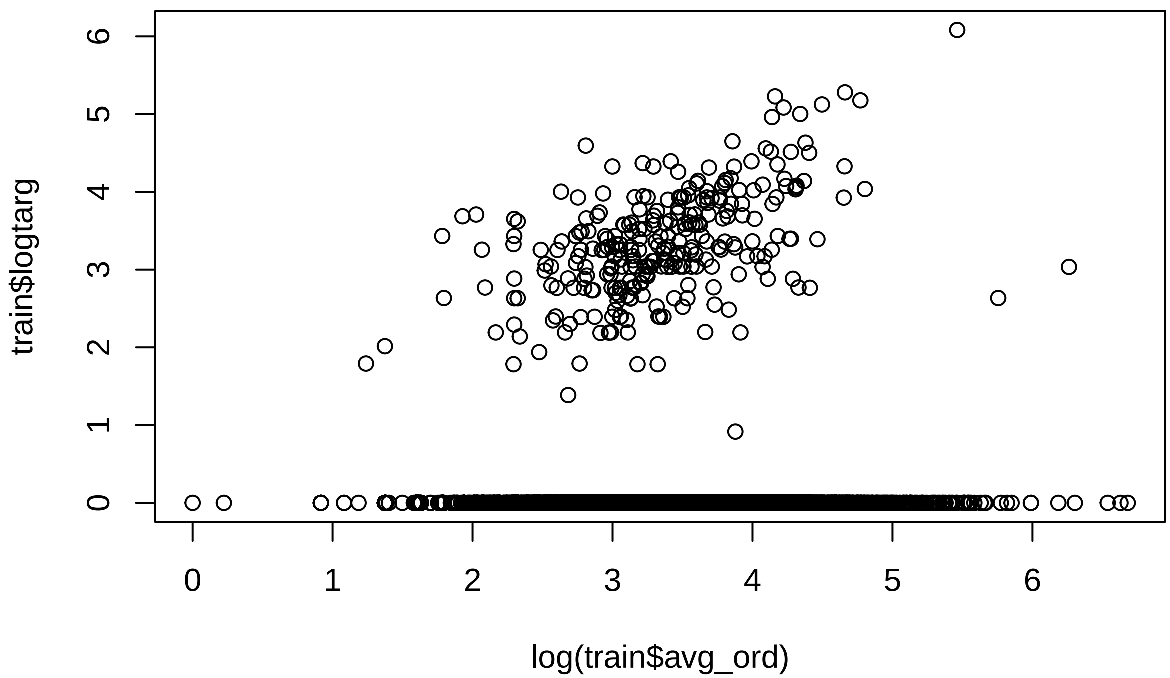

```
plot(train$monetary_tot, train$logtarg)
```



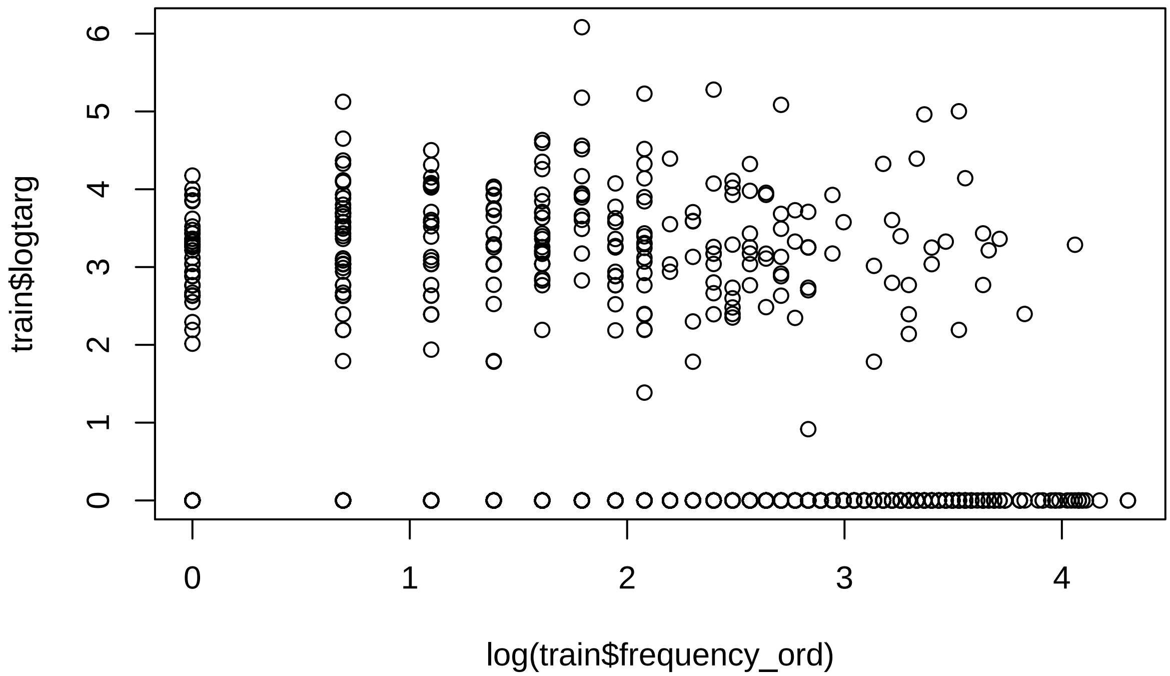

```
plot(log(train$monetary_avg), train$logtarg)
```



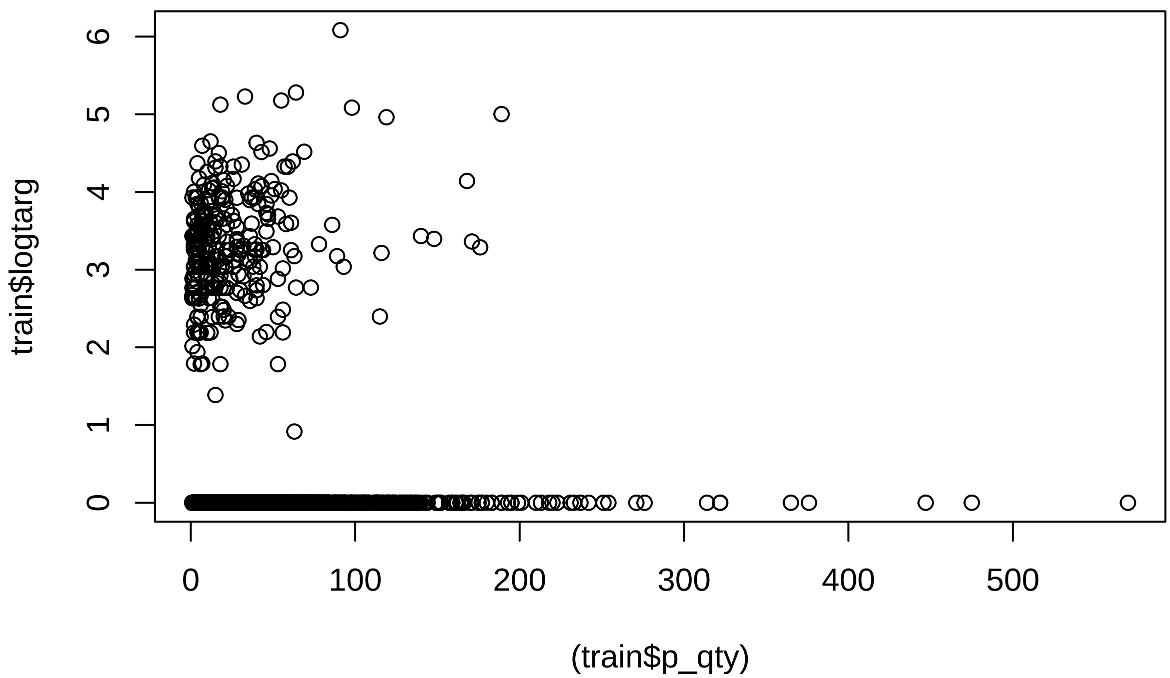

```
plot(log(train$avg_ord), train$logtarg)
```



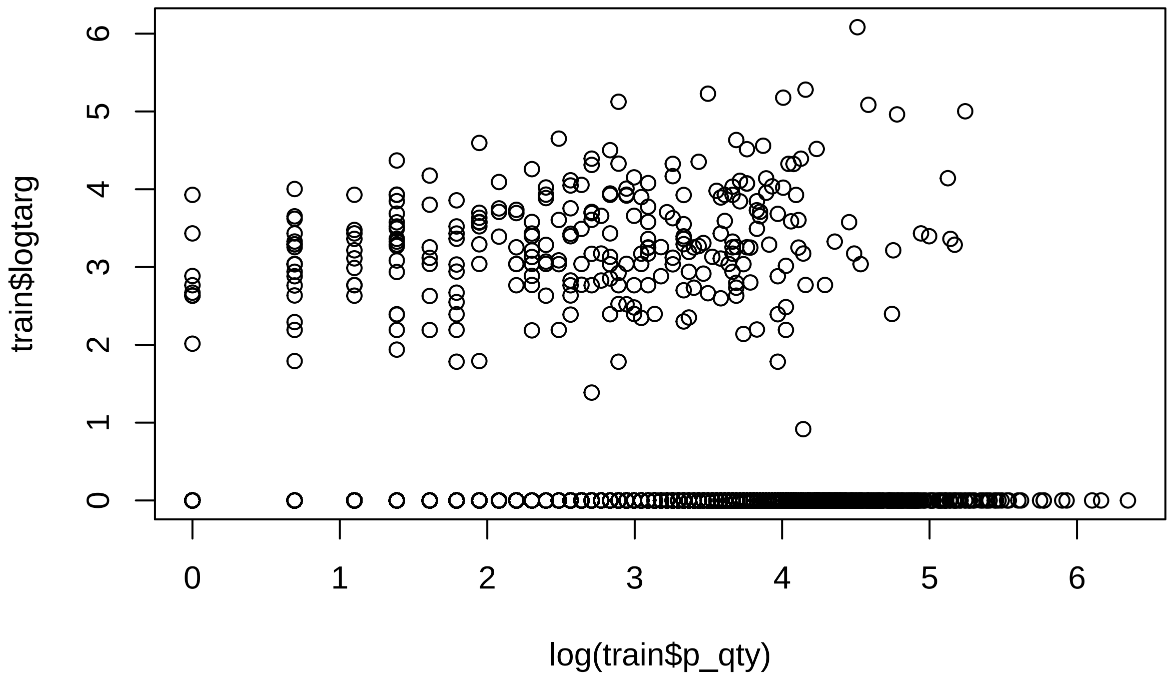

```
plot(log(train$frequency_ord), train$logtarg)
```



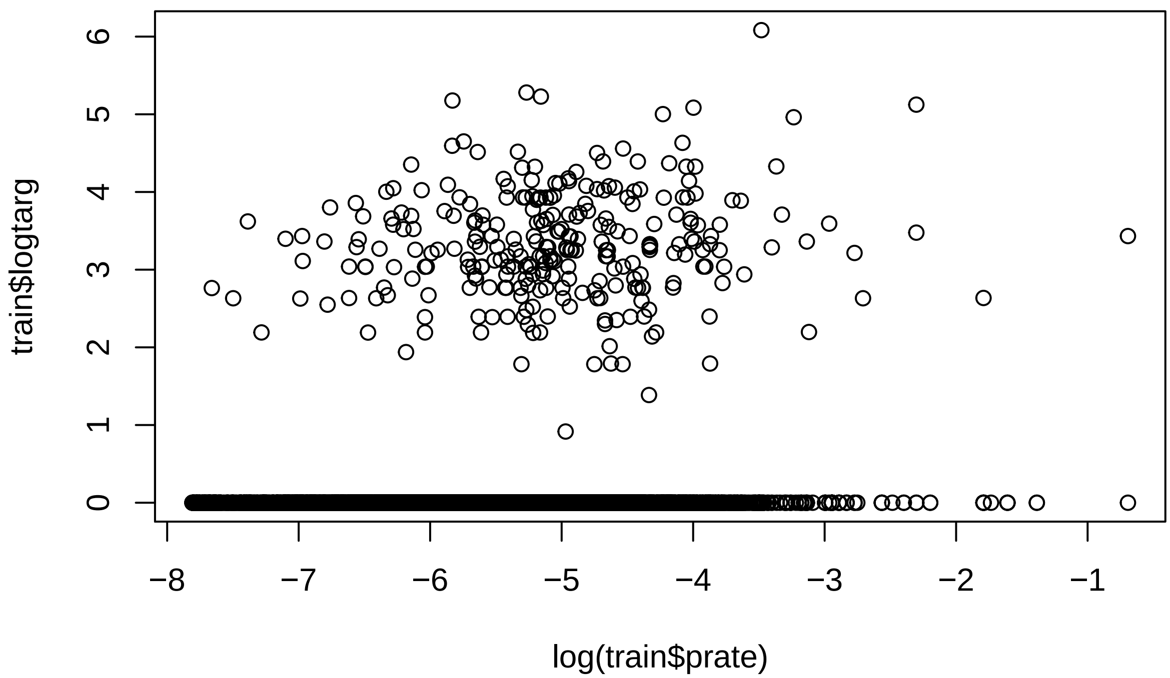

```r
plot((train$p_qty), train$logtarg)
```




```r
plot(log(train$p_qty), train$logtarg)
```




```
plot(log(train$prate),train$logtarg)
```



Additional EDAs on category

```
train2 = inner_join(dat, y, by="id")

cats = train2 %>%
  group_by(category) %>%
  summarize(qty_0 = sum(qty[logtarg == 0]), qty_1 = sum(qty[logtarg > 0])) %>% #Why Anisha added cat19
  #summarize(qty_0 = sum(qty*price[logtarg == 0]), qty_1 = sum(qty*price[logtarg > 0])) %>% #try add ca
  mutate(pct_0 = qty_0/sum(qty_0), pct_1 = qty_1/sum(qty_1), diff = abs(pct_1 - pct_0)) %>%
```

```r
  select(category, qty_0, qty_1, pct_0, pct_1, diff)

#ggplot(data = cats, aes(category, diff)) + geom_point()
cats[cats$diff > 0.01,]
```

```
## # A tibble: 4 x 6
##   category qty_0 qty_1     pct_0      pct_1       diff
##      <int> <int> <int>     <dbl>      <dbl>      <dbl>
## 1        8  7642   477 0.0513527 0.06504841 0.01369572
## 2       12  4763   160 0.0320064 0.02181917 0.01018722
## 3       19 21554   940 0.1448385 0.12818764 0.01665088
## 4       20 17157  1143 0.1152916 0.15587072 0.04057915
```

3. Model fitting

a. Baseline model => submitted with score 0.61844

```r
fit1 = lm(logtarg ~ log(monetary_avg+1) + log(avg_ord+1) + log(frequency_ord) + recency_first + recency_
#summary(fit1)
vif(fit1)
```

```
## log(monetary_avg + 1)      log(avg_ord + 1)    log(frequency_ord)
##              2.099127              2.125480              3.064547
##         recency_first          recency_last
##              3.295646              2.272089
```

```r
#plot(fit1)
```

b. Model fit2 => submitted with score 0.61887

```r
fit2 = lm(logtarg ~ log(monetary_avg+1) + log(avg_ord+1) + log(frequency_ord) + log(prate) + recency_fi
#summary(fit2)
vif(fit2)
```

```
## log(monetary_avg + 1)      log(avg_ord + 1)    log(frequency_ord)
##              2.106627              2.134549              7.102740
##            log(prate)         recency_first          recency_last
##              5.020810              6.717000              2.405834
```

c. Model fit3 ADDED BY SPENCER

```r
full = lm(logtarg ~ recency_first + recency_last + date_duration + log(p_qty) + log(count_cat)
          + log(catrate) + log(monetary_avg + 1)  + log(avg_ord + 1)
          + log(frequency_ord) + log(prate)
        , data = train)
#summary(full)

adj = step(full, scope = list(upper=full), data = train, direction="both")
```

```
## Start:  AIC=-8085.1
## logtarg ~ recency_first + recency_last + date_duration + log(p_qty) +
##     log(count_cat) + log(catrate) + log(monetary_avg + 1) + log(avg_ord +
##     1) + log(frequency_ord) + log(prate)
##
##
## Step:  AIC=-8085.1
## logtarg ~ recency_first + recency_last + date_duration + log(p_qty) +
##     log(count_cat) + log(catrate) + log(monetary_avg + 1) + log(avg_ord +
```

```
##     1) + log(prate)
##
##
## Step:  AIC=-8085.1
## logtarg ~ recency_first + recency_last + log(p_qty) + log(count_cat) +
##     log(catrate) + log(monetary_avg + 1) + log(avg_ord + 1) +
##     log(prate)
##
##                         Df Sum of Sq    RSS     AIC
## - log(p_qty)             1   0.02828 3070.3 -8087.0
## - log(monetary_avg + 1)  1   0.18327 3070.4 -8086.6
## - log(count_cat)         1   0.18762 3070.4 -8086.6
## - log(avg_ord + 1)       1   0.18822 3070.4 -8086.6
## - recency_last           1   0.20248 3070.4 -8086.6
## - log(catrate)           1   0.25040 3070.5 -8086.4
## <none>                               3070.2 -8085.1
## - log(prate)             1   2.11771 3072.4 -8081.4
## - recency_first          1   2.18617 3072.4 -8081.2
##
## Step:  AIC=-8087.02
## logtarg ~ recency_first + recency_last + log(count_cat) + log(catrate) +
##     log(monetary_avg + 1) + log(avg_ord + 1) + log(prate)
##
##                         Df Sum of Sq    RSS     AIC
## - recency_last           1   0.20258 3070.5 -8088.5
## <none>                               3070.3 -8087.0
## + log(p_qty)             1   0.02828 3070.2 -8085.1
## - log(monetary_avg + 1)  1   1.84710 3072.1 -8084.1
## - log(count_cat)         1   1.92351 3072.2 -8083.9
## - log(prate)             1   2.13046 3072.4 -8083.3
## - recency_first          1   2.17433 3072.4 -8083.2
## - log(catrate)           1   2.45732 3072.7 -8082.4
## - log(avg_ord + 1)       1   2.51979 3072.8 -8082.3
##
## Step:  AIC=-8088.48
## logtarg ~ recency_first + log(count_cat) + log(catrate) + log(monetary_avg +
##     1) + log(avg_ord + 1) + log(prate)
##
##                         Df Sum of Sq    RSS     AIC
## <none>                               3070.5 -8088.5
## + recency_last           1   0.20258 3070.3 -8087.0
## + date_duration          1   0.20258 3070.3 -8087.0
## + log(p_qty)             1   0.02837 3070.4 -8086.6
## - log(count_cat)         1   1.72977 3072.2 -8085.8
## - log(monetary_avg + 1)  1   1.83603 3072.3 -8085.6
## - log(prate)             1   1.93894 3072.4 -8085.3
## - recency_first          1   1.97229 3072.4 -8085.2
## - log(catrate)           1   2.27682 3072.8 -8084.4
## - log(avg_ord + 1)       1   2.55803 3073.0 -8083.6
```

```
summary(adj)
```

```
##
## Call:
## lm(formula = logtarg ~ recency_first + log(count_cat) + log(catrate) +
```

```
##     log(monetary_avg + 1) + log(avg_ord + 1) + log(prate), data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3573 -0.1603 -0.1103 -0.0565  5.7499
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.881e-01  9.036e-02   3.188  0.00144 **
## recency_first         -4.679e-05  2.037e-05  -2.297  0.02162 *
## log(count_cat)         4.086e-02  1.899e-02   2.152  0.03146 *
## log(catrate)          -5.322e-02  2.156e-02  -2.468  0.01359 *
## log(monetary_avg + 1) -4.714e-02  2.127e-02  -2.217  0.02668 *
## log(avg_ord + 1)       4.617e-02  1.764e-02   2.616  0.00890 **
## log(prate)             3.633e-02  1.595e-02   2.278  0.02276 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6113 on 8217 degrees of freedom
## Multiple R-squared:  0.01476,    Adjusted R-squared:  0.01404
## F-statistic: 20.52 on 6 and 8217 DF,  p-value: < 2.2e-16
```

```r
vif(adj)
```

```
##         recency_first        log(count_cat)          log(catrate)
##              6.035272              5.418038              3.454667
## log(monetary_avg + 1)      log(avg_ord + 1)            log(prate)
##              3.265707              4.173752              4.774073
```

4. Model fitting - logistic and regression a. Linear + Logistic Part 1: Linear - trained on logtarg > 0

```r
train_lm = all[!is.na(all$logtarg) & all$logtarg > 0,] #280 obs instead of 8311

colnames(train_lm)
```

```
##  [1] "id"              "recency_first" "recency_last"  "date_duration"
##  [5] "p_qty"           "frequency_ord" "monetary_tot"  "monetary_avg"
##  [9] "count_cat"       "f"             "avg_ord"       "prate"
## [13] "catrate"         "prate2"        "cat20"         "cat19"
## [17] "cat17"           "logtarg"       "responseflag"
```

```r
#cor(train_lm[-1])

full_lm = lm(logtarg ~ recency_first
             + recency_last
             #+ date_duration
             #+ log(p_qty)
             + log(frequency_ord)
             #+ log(monetary_tot)
             + log(monetary_avg + 1)
             + log(avg_ord + 1)
             + log(count_cat)
             + log(prate)
             #+ log(catrate)
             + log(prate2)
             + cat19
```

```r
             + cat20
             + cat17, data = train_lm)
summary(full_lm)
```

```
##
## Call:
## lm(formula = logtarg ~ recency_first + recency_last + log(frequency_ord) +
##     log(monetary_avg + 1) + log(avg_ord + 1) + log(count_cat) +
##     log(prate) + log(prate2) + cat19 + cat20 + cat17, data = train_lm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.49188 -0.37911  0.02536  0.38890  1.63170
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.586e+00  3.757e-01   6.882 4.18e-11 ***
## recency_first         -4.796e-05  9.900e-05  -0.484   0.6284
## recency_last           6.685e-05  1.402e-04   0.477   0.6340
## log(frequency_ord)    -2.829e-02  9.858e-02  -0.287   0.7744
## log(monetary_avg + 1) -5.596e-01  1.293e-01  -4.328 2.13e-05 ***
## log(avg_ord + 1)       6.978e-01  9.573e-02   7.289 3.51e-12 ***
## log(count_cat)        -4.752e-02  9.826e-02  -0.484   0.6291
## log(prate)             5.560e-02  7.333e-02   0.758   0.4490
## log(prate2)           -3.663e-02  2.980e-02  -1.229   0.2201
## cat19                  2.090e-01  1.018e-01   2.052   0.0411 *
## cat20                 -2.100e-01  8.401e-02  -2.500   0.0130 *
## cat17                  9.783e-02  1.950e-01   0.502   0.6162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6142 on 268 degrees of freedom
## Multiple R-squared:  0.2959, Adjusted R-squared:  0.267
## F-statistic: 10.24 on 11 and 268 DF,  p-value: 1.358e-15
```

```r
#vif(full_lm)

adj_lm = step(full_lm, scope = list(upper=full_lm), data = train_lm, direction="both")
```

```
## Start:  AIC=-261.25
## logtarg ~ recency_first + recency_last + log(frequency_ord) +
##     log(monetary_avg + 1) + log(avg_ord + 1) + log(count_cat) +
##     log(prate) + log(prate2) + cat19 + cat20 + cat17
##
##                        Df Sum of Sq    RSS     AIC
## - log(frequency_ord)     1    0.0311 101.12 -263.16
## - recency_last           1    0.0857 101.18 -263.01
## - log(count_cat)         1    0.0882 101.18 -263.00
## - recency_first          1    0.0885 101.18 -263.00
## - cat17                  1    0.0950 101.19 -262.99
## - log(prate)             1    0.2169 101.31 -262.65
## - log(prate2)            1    0.5698 101.66 -261.68
## <none>                               101.09 -261.25
## - cat19                  1    1.5883 102.68 -258.88
```

```
## - cat20                 1    2.3574 103.45 -256.79
## - log(monetary_avg + 1)  1    7.0661 108.16 -244.33
## - log(avg_ord + 1)       1   20.0420 121.14 -212.61
##
## Step:  AIC=-263.16
## logtarg ~ recency_first + recency_last + log(monetary_avg + 1) +
##     log(avg_ord + 1) + log(count_cat) + log(prate) + log(prate2) +
##     cat19 + cat20 + cat17
##
##                         Df Sum of Sq    RSS     AIC
## - cat17                  1    0.0950 101.22 -264.90
## - recency_last           1    0.1023 101.23 -264.88
## - log(prate)             1    0.1972 101.32 -264.62
## - log(count_cat)         1    0.2062 101.33 -264.59
## - recency_first          1    0.2733 101.40 -264.41
## - log(prate2)            1    0.5413 101.67 -263.67
## <none>                               101.12 -263.16
## + log(frequency_ord)     1    0.0311 101.09 -261.25
## - cat19                  1    1.6794 102.80 -260.55
## - cat20                  1    2.4614 103.59 -258.43
## - log(monetary_avg + 1)  1    7.7680 108.89 -244.44
## - log(avg_ord + 1)       1   21.8300 122.95 -210.43
##
## Step:  AIC=-264.9
## logtarg ~ recency_first + recency_last + log(monetary_avg + 1) +
##     log(avg_ord + 1) + log(count_cat) + log(prate) + log(prate2) +
##     cat19 + cat20
##
##                         Df Sum of Sq    RSS     AIC
## - recency_last           1    0.0894 101.31 -266.65
## - log(count_cat)         1    0.1865 101.41 -266.38
## - log(prate)             1    0.1935 101.41 -266.37
## - recency_first          1    0.2560 101.47 -266.19
## - log(prate2)            1    0.5192 101.74 -265.47
## <none>                               101.22 -264.90
## + cat17                  1    0.0950 101.12 -263.16
## + log(frequency_ord)     1    0.0311 101.19 -262.99
## - cat19                  1    1.7261 102.95 -262.17
## - cat20                  1    2.5371 103.76 -259.97
## - log(monetary_avg + 1)  1    7.6739 108.89 -246.44
## - log(avg_ord + 1)       1   21.7618 122.98 -212.37
##
## Step:  AIC=-266.65
## logtarg ~ recency_first + log(monetary_avg + 1) + log(avg_ord +
##     1) + log(count_cat) + log(prate) + log(prate2) + cat19 +
##     cat20
##
##                         Df Sum of Sq    RSS     AIC
## - log(prate)             1    0.1086 101.42 -268.35
## - recency_first          1    0.1988 101.51 -268.10
## - log(count_cat)         1    0.2171 101.53 -268.05
## - log(prate2)            1    0.4346 101.74 -267.45
## <none>                               101.31 -266.65
## + recency_last           1    0.0894 101.22 -264.90
```

```
## + cat17                  1    0.0821 101.23 -264.88
## + log(frequency_ord)     1    0.0465 101.26 -264.78
## - cat19                  1    1.7592 103.07 -263.83
## - cat20                  1    2.5733 103.88 -261.63
## - log(monetary_avg + 1)  1    7.7480 109.06 -248.02
## - log(avg_ord + 1)       1   22.0526 123.36 -213.51
##
## Step:  AIC=-268.35
## logtarg ~ recency_first + log(monetary_avg + 1) + log(avg_ord +
##     1) + log(count_cat) + log(prate2) + cat19 + cat20
##
##                         Df Sum of Sq    RSS     AIC
## - log(count_cat)         1    0.1365 101.55 -269.98
## - log(prate2)            1    0.3826 101.80 -269.30
## - recency_first          1    0.5517 101.97 -268.83
## <none>                               101.42 -268.35
## + log(prate)             1    0.1086 101.31 -266.65
## + cat17                  1    0.0878 101.33 -266.60
## + recency_last           1    0.0046 101.41 -266.37
## + log(frequency_ord)     1    0.0038 101.41 -266.36
## - cat19                  1    2.0127 103.43 -264.85
## - cat20                  1    2.4700 103.89 -263.62
## - log(monetary_avg + 1)  1    7.6394 109.06 -250.02
## - log(avg_ord + 1)       1   22.6951 124.11 -213.81
##
## Step:  AIC=-269.98
## logtarg ~ recency_first + log(monetary_avg + 1) + log(avg_ord +
##     1) + log(prate2) + cat19 + cat20
##
##                         Df Sum of Sq    RSS     AIC
## - log(prate2)            1    0.2824 101.84 -271.20
## <none>                               101.55 -269.98
## - recency_first          1    0.9745 102.53 -269.30
## + log(count_cat)         1    0.1365 101.42 -268.35
## + cat17                  1    0.0660 101.49 -268.16
## + recency_last           1    0.0311 101.52 -268.06
## + log(frequency_ord)     1    0.0308 101.52 -268.06
## + log(prate)             1    0.0281 101.53 -268.05
## - cat19                  1    1.9644 103.52 -266.61
## - cat20                  1    2.6832 104.24 -264.67
## - log(monetary_avg + 1)  1    8.4013 109.95 -249.72
## - log(avg_ord + 1)       1   27.3032 128.86 -205.30
##
## Step:  AIC=-271.2
## logtarg ~ recency_first + log(monetary_avg + 1) + log(avg_ord +
##     1) + cat19 + cat20
##
##                         Df Sum of Sq    RSS     AIC
## - recency_first          1    0.6978 102.53 -271.29
## <none>                               101.84 -271.20
## + log(prate2)            1    0.2824 101.55 -269.98
## + cat17                  1    0.0647 101.77 -269.38
## + log(count_cat)         1    0.0363 101.80 -269.30
## + log(prate)             1    0.0244 101.81 -269.26
```

```
## + recency_last           1    0.0002 101.84 -269.20
## + log(frequency_ord)     1    0.0001 101.84 -269.20
## - cat19                  1    2.2284 104.06 -267.14
## - cat20                  1    2.5491 104.39 -266.28
## - log(monetary_avg + 1)  1    8.3126 110.15 -251.23
## - log(avg_ord + 1)       1   27.8697 129.71 -205.47
##
## Step:  AIC=-271.29
## logtarg ~ log(monetary_avg + 1) + log(avg_ord + 1) + cat19 +
##     cat20
##
##                         Df Sum of Sq    RSS     AIC
## <none>                               102.53 -271.29
## + recency_first          1     0.698 101.84 -271.20
## + log(count_cat)         1     0.383 102.15 -270.33
## + log(prate)             1     0.298 102.24 -270.10
## + log(frequency_ord)     1     0.230 102.30 -269.92
## + recency_last           1     0.083 102.45 -269.51
## + cat17                  1     0.033 102.50 -269.38
## + log(prate2)            1     0.006 102.53 -269.30
## - cat19                  1     1.708 104.24 -268.66
## - cat20                  1     3.049 105.58 -265.08
## - log(monetary_avg + 1)  1    10.351 112.89 -246.36
## - log(avg_ord + 1)       1    31.798 134.33 -197.65
```

```r
summary(adj_lm)
```

```
##
## Call:
## lm(formula = logtarg ~ log(monetary_avg + 1) + log(avg_ord +
##     1) + cat19 + cat20, data = train_lm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5951 -0.3928 -0.0188  0.3972  1.6146
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.31935    0.20505  11.311  < 2e-16 ***
## log(monetary_avg + 1) -0.56089    0.10645  -5.269 2.78e-07 ***
## log(avg_ord + 1)       0.69858    0.07565   9.235  < 2e-16 ***
## cat19                  0.17800    0.08316   2.140  0.03320 *
## cat20                 -0.22742    0.07952  -2.860  0.00456 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6106 on 275 degrees of freedom
## Multiple R-squared:  0.2858, Adjusted R-squared:  0.2755
## F-statistic: 27.52 on 4 and 275 DF,  p-value: < 2.2e-16
```

```r
vif(adj_lm)
```

```
## log(monetary_avg + 1)      log(avg_ord + 1)                 cat19
##              1.850510              2.033360              1.212502
##                 cat20
```

```
##               1.139740
```

```
#plot(adj_lm)
```

b. Linear + Logistic Part 2: Logistic - trained on logtarg not NA

```
train_log = all[!is.na(all$logtarg) & all$logtarg >= 0,]

log_fit <- glm(responseflag ~ recency_first
             + recency_last
             #+ date_duration
             #+ log(p_qty)
             + log(frequency_ord)
             #+ log(monetary_tot)
             + log(monetary_avg + 1)
             + log(avg_ord + 1)
             + log(count_cat)
             + log(prate)
             #+ log(catrate)
             + log(prate2)
             + cat19
             + cat20
             + cat17,
family = "binomial", data = train_log)

summary(log_fit)
```

```
##
## Call:
## glm(formula = responseflag ~ recency_first + recency_last + log(frequency_ord) +
##     log(monetary_avg + 1) + log(avg_ord + 1) + log(count_cat) +
##     log(prate) + log(prate2) + cat19 + cat20 + cat17, family = "binomial",
##     data = train_log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5915  -0.3018  -0.2354  -0.1765   3.4286
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.6363529  0.7325481  -2.234   0.0255 *
## recency_first         -0.0003735  0.0001705  -2.190   0.0285 *
## recency_last          -0.0003233  0.0002336  -1.384   0.1663
## log(frequency_ord)     0.3383590  0.1782424   1.898   0.0577 .
## log(monetary_avg + 1) -0.2387428  0.2016492  -1.184   0.2364
## log(avg_ord + 1)       0.1610573  0.1532267   1.051   0.2932
## log(count_cat)        -0.0776744  0.1784815  -0.435   0.6634
## log(prate)             0.3422574  0.1381963   2.477   0.0133 *
## log(prate2)           -0.0613095  0.0504804  -1.215   0.2245
## cat19                 -0.0925998  0.1583811  -0.585   0.5588
## cat20                  0.3395080  0.1401172   2.423   0.0154 *
## cat17                  0.0257278  0.3290538   0.078   0.9377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2443.2  on 8223  degrees of freedom
## Residual deviance: 2315.4  on 8212  degrees of freedom
## AIC: 2339.4
##
## Number of Fisher Scoring iterations: 7
```

```r
adj_log_fit <- glm(responseflag ~ recency_first
             + log(frequency_ord)
             + log(prate)
             + cat20,
family = binomial("logit"), data = train_log)

summary(adj_log_fit)
```

```
##
## Call:
## glm(formula = responseflag ~ recency_first + log(frequency_ord) +
##     log(prate) + cat20, family = binomial("logit"), data = train_log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6452  -0.2971  -0.2323  -0.1793   3.2735
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.7961271  0.6178941  -2.907  0.00365 **
## recency_first      -0.0004902  0.0001628  -3.011  0.00260 **
## log(frequency_ord)  0.4266458  0.1323657   3.223  0.00127 **
## log(prate)          0.3284698  0.1177097   2.791  0.00526 **
## cat20               0.3529968  0.1382241   2.554  0.01066 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2443.2  on 8223  degrees of freedom
## Residual deviance: 2321.7  on 8219  degrees of freedom
## AIC: 2331.7
##
## Number of Fisher Scoring iterations: 6
```

```r
vif(adj_log_fit)
```

```
##      recency_first log(frequency_ord)         log(prate)
##           4.786082           4.989725           2.774465
##              cat20
##           1.226207
```

Choose threshold p for logistic model

```r
predicted_vals <- predict(adj_log_fit, data = train_log, type = "response")
#get_logit_details(train_log$responseflag, predicted_vals, 0.10) #0.1
#get_logit_details(train_log$responseflag, predicted_vals, 0.071) #0.1355
```

CURRENT FINAL OUTPUT WITH THE HIGHEST SCORE!!!

```r
#Predict and output
test = all[is.na(all$logtarg),]

test$yhat = predict(adj_lm, test)
prob = predict(adj_log_fit, test, type = "response")

#output test values
out = cbind(test[,c('id', 'yhat')], prob)
out$logtarg = out$yhat * out$prob
final = out[,c('id','logtarg')]
colnames(final) <- c("id", "yhat")
head(final)
```

```
##     id       yhat
## 1  914 0.12780023
## 2  957 0.13065758
## 3 1406 0.13788920
## 4 1414 0.09167945
## 5 1546 0.09449043
## 6 1651 0.04248661
```

```r
write.csv(final, "output/test_lmlog.csv", row.names=F)
```

OLD Testing with Choosing threshold p

```r
#Predict and output
test = all[is.na(all$logtarg),]

test$yhat = predict(adj_lm, test)
prob = predict(adj_log_fit, test, type = "response")


#output test values
out = cbind(test[,c('id', 'yhat')], prob)
out$flag = ifelse(out$prob >= 0.071, 1, 0)
out$logtarg = out$yhat * out$flag
final = out[,c('id','logtarg')]
colnames(final) <- c("id", "yhat")
head(final)
```

```
##     id yhat
## 1  914    0
## 2  957    0
## 3 1406    0
## 4 1414    0
## 5 1546    0
## 6 1651    0
```

```r
#write.csv(final, "../output/test_threshold.csv", row.names=F)
```