Lab09

Jeffrey Wong

1/15/2020

```
setwd("/Users/jw-mba/Desktop/r-projects")
#source("GCP_local_IP_connection_setup.R")
#resale_item <- dbGetQuery(con, "SELECT * FROM econ_621.resale_item_data; ")
#write.csv(resale_item, "resale_item_data.csv", row.names = F)
#resale_page <- dbGetQuery(con, "SELECT * FROM econ_621.resale_pageviews;")
#write.csv(resale_page, "resale_pageviews.csv", row.names = F)
```

Build a predictive model for the percent of original price of the used items in this dataset that have been sold. An item's sale price is the used_list_price and original price is msrp_new. Items with an NA value for first_ordered_date have not been sold and should not be part of this analysis

a. Describe your independent variables (pageviews should be included):

- the percentage can be of the dependance of pageviews, department, category, color, msrp_new (original price)

```
library(plyr)
library(dplyr)
```

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

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

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

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

```
library(varhandle)
library(reshape2)
library(olsrr)
```

```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':
##
##     rivers
```

```
library(Metrics)
resale_item <- read.csv("resale_item_data.csv", stringsAsFactors = F)
colnames(resale_item)
```

```
##  [1] "unique_item_id"              "used_list_price"
##  [3] "used_condition"              "department"
##  [5] "category"                    "item_parent_sku"
##  [7] "color"                       "msrp_new"
##  [9] "last_known_retail_price_new" "first_approved_date"
## [11] "first_ordered_date"
```

What types of variables are they? Which is the default value for binary/category variables?

- pageviews and msrp_new are quantitative, others are category variables, there's no binary variables such as gender here.

Are there any variables you want to create (eg, discount offered)?

- yes, days on the market, which can be defined as days b/w first order date and first approve date

```
#data clean up
key_cols <- c("unique_item_id", "item_parent_sku" ,"used_list_price", "department", "category", "color"
resale_item <- resale_item[!is.na(resale_item$first_ordered_date), key_cols]

factor_var <- c("department", "category", "color")
resale_item[, factor_var] <- data.frame(sapply(resale_item[, factor_var], as.factor))
date_var <- c("first_approved_date", "first_ordered_date")
resale_item[, date_var] <- data.frame(sapply(resale_item[, date_var], as.Date))
resale_item <- data.frame(resale_item[, -which(colnames(resale_item) %in% date_var)],
                          days_on_mkt = as.integer(resale_item$first_ordered_date -
                                                     resale_item$first_approved_date))

# define the discount percentage dependent variables
resale_item <- data.frame(resale_item[,-which(colnames(resale_item) == "used_list_price")], perc =
                                       (resale_item$used_list_price / resale_item$msrp_new *100))

# merge the pageviwes data
pageviews <- read.csv("resale_pageviews.csv", stringsAsFactors = F)
resale_item <- merge(resale_item, pageviews, by = "item_parent_sku", all.x = T)
resale_item[is.na(resale_item$pageviews), which(colnames(resale_item) == "pageviews")] <- 0
resale_item <- resale_item[,-1]

head(resale_item)
```

```
##   unique_item_id         department   category     color msrp_new days_on_mkt
## 1         296096 Kitchen and Dining       Pots Not color  279.960           2
## 2         296105 Kitchen and Dining       Pots Not color  454.935           1
## 3         279648 Kitchen and Dining Appliances      Gray   68.000           4
## 4         339318 Kitchen and Dining       Pots Not color   44.760           0
## 5         298767 Kitchen and Dining       Pans      Blue   25.935           6
## 6         276644    Outdoor Storage    Shelves       Red  100.000         130
##        perc pageviews
## 1 103.12402       213
## 2  47.24875       213
## 3  96.70294        46
## 4  92.79714        29
## 5  42.00000       595
## 6  50.93000     11282
```

b. Create training, testing, and validation subsets. Construct, constrain, and tune a simple linear regression model. Describe your process.

-convert the categorical variables into its own binary ones

```
colnames(resale_item) <- c("id","dpt","cat","col","origin_price", "days_on_mkt", "perc", "pageviews")

category_cols <- c("dpt", "cat", "col")

resale_item$col <- as.character(resale_item$col)
resale_item[resale_item$col == "Multi-colored", which(colnames(resale_item) == "col")] <- "Multicolored
resale_item$col <- as.factor(resale_item$col)

# For each category variable, create a set of dummies and append to the data set
resale_item_wide <- resale_item
for (i in 1:length(category_cols)) {
  dummies <- to.dummy(resale_item_wide[, category_cols[i]], category_cols[i])
resale_item_wide <- cbind(resale_item_wide, dummies)
}

# examine NA items, no NA items,
sapply(resale_item_wide, function(x) {sum(is.na(x))})
```

```
##                     id                    dpt                    cat
##                      0                      0                      0
##                    col           origin_price            days_on_mkt
##                      0                      0                      0
##                   perc              pageviews          dpt.Furniture
##                      0                      0                      0
##     dpt.Kids_Furniture       dpt.Kids_Storage dpt.Kitchen_and_Dining
##                      0                      0                      0
##  dpt.Outdoor_Furniture    dpt.Outdoor_Storage            dpt.Storage
##                      0                      0                      0
##         cat.Appliances               cat.Bags               cat.Beds
##                      0                      0                      0
##              cat.Bowls              cat.Boxes           cat.Cabinets
##                      0                      0                      0
##     cat.Casual_Glasses             cat.Chairs              cat.Desks
```

```
##                     0                     0                     0
##     cat.Dining_Tables          cat.Envelopes             cat.Forks
##                     0                     0                     0
##           cat.Gadgets             cat.Knives              cat.Pans
##                     0                     0                     0
##            cat.Plates              cat.Pots              cat.Rugs
##                     0                     0                     0
##   cat.Serving_Utensils            cat.Shelves       cat.Side_Tables
##                     0                     0                     0
##            cat.Spoons       cat.Wine_Glasses             col.Black
##                     0                     0                     0
##              col.Blue              col.Brown              col.Gray
##                     0                     0                     0
##             col.Green              col.Khaki           col.Metallic
##                     0                     0                     0
##       col.Multicolored          col.Not_color             col.Orange
##                     0                     0                     0
##             col.other              col.Pink             col.Purple
##                     0                     0                     0
##               col.Red         col.Unassigned              col.White
##                     0                     0                     0
##            col.Yellow
##                     0
```

Building the first regression model

```
# Examine category variables by sorting by frequency
sapply(resale_item_wide[, category_cols], function(x) {
  names(table(x))[order(table(x), decreasing = T)]
  })
```

```
## $dpt
## [1] "Kitchen and Dining" "Outdoor Furniture"  "Outdoor Storage"
## [4] "Kids Storage"       "Kids Furniture"     "Furniture"
## [7] "Storage"
##
## $cat
##  [1] "Spoons"           "Shelves"          "Boxes"            "Rugs"
##  [5] "Side Tables"      "Desks"            "Forks"            "Envelopes"
##  [9] "Beds"             "Appliances"       "Dining Tables"    "Chairs"
## [13] "Wine Glasses"     "Bags"             "Gadgets"          "Serving Utensils"
## [17] "Plates"           "Knives"           "Bowls"            "Pots"
## [21] "Cabinets"         "Casual Glasses"   "Pans"
##
## $col
##  [1] "Blue"         "Gray"         "Black"        "Green"        "Red"
##  [6] "Brown"        "Purple"       "Orange"       "Khaki"        "Yellow"
## [11] "White"        "Not color"    "Multicolored" "Pink"         "other"
## [16] "Metallic"     "Unassigned"
```

```
# create the first round regression model
model <- lm(perc ~
              origin_price +
```

```
              days_on_mkt +
              pageviews +
              dpt.Kitchen_and_Dining + dpt.Outdoor_Furniture + dpt.Outdoor_Storage +
              dpt.Kids_Storage + dpt.Kids_Furniture + dpt.Furniture +
              cat.Spoons + cat.Shelves + cat.Boxes + cat.Rugs + cat.Side_Tables +
              cat.Desks + cat.Forks + cat.Envelopes + cat.Beds + cat.Appliances +
              cat.Dining_Tables + cat.Wine_Glasses + cat.Bags + cat.Gadgets +
              cat.Serving_Utensils + cat.Plates + cat.Knives + cat.Bowls + cat.Pots +
              cat.Casual_Glasses +
              col.Blue + col.Gray + col.Black + col.Green + col.Red + col.Brown + col.Purple +
              col.Orange + col.Khaki + col.Yellow + col.White + col.Not_color + col.Multicolored +
              col.Pink + col.other + col.Metallic,
            data = resale_item_wide)
summary(model)
```

```
##
## Call:
## lm(formula = perc ~ origin_price + days_on_mkt + pageviews +
##     dpt.Kitchen_and_Dining + dpt.Outdoor_Furniture + dpt.Outdoor_Storage +
##     dpt.Kids_Storage + dpt.Kids_Furniture + dpt.Furniture + cat.Spoons +
##     cat.Shelves + cat.Boxes + cat.Rugs + cat.Side_Tables + cat.Desks +
##     cat.Forks + cat.Envelopes + cat.Beds + cat.Appliances + cat.Dining_Tables +
##     cat.Wine_Glasses + cat.Bags + cat.Gadgets + cat.Serving_Utensils +
##     cat.Plates + cat.Knives + cat.Bowls + cat.Pots + cat.Casual_Glasses +
##     col.Blue + col.Gray + col.Black + col.Green + col.Red + col.Brown +
##     col.Purple + col.Orange + col.Khaki + col.Yellow + col.White +
##     col.Not_color + col.Multicolored + col.Pink + col.other +
##     col.Metallic, data = resale_item_wide)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.766 -15.549  -2.591  15.325  97.948
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.944e+01  2.048e+01   3.391 0.000699 ***
## origin_price           -7.693e-02  1.891e-03 -40.693  < 2e-16 ***
## days_on_mkt            -3.283e-02  5.823e-03  -5.637 1.76e-08 ***
## pageviews               1.083e-04  1.117e-04   0.969 0.332525
## dpt.Kitchen_and_Dining  2.001e+00  6.962e+00   0.287 0.773783
## dpt.Outdoor_Furniture  -1.585e+01  4.834e+00  -3.279 0.001043 **
## dpt.Outdoor_Storage     1.014e+01  2.244e+00   4.518 6.28e-06 ***
## dpt.Kids_Storage        1.281e+01  2.264e+00   5.658 1.56e-08 ***
## dpt.Kids_Furniture     -1.306e+01  4.852e+00  -2.692 0.007101 **
## dpt.Furniture          -1.763e+01  5.063e+00  -3.482 0.000499 ***
## cat.Spoons              4.366e+00  5.174e+00   0.844 0.398811
## cat.Shelves            -2.909e+01  4.138e+00  -7.029 2.18e-12 ***
## cat.Boxes              -3.127e+01  4.146e+00  -7.543 4.87e-14 ***
## cat.Rugs                1.910e+00  1.372e+00   1.392 0.163970
## cat.Side_Tables        -3.148e+00  1.360e+00  -2.315 0.020607 *
## cat.Desks              -9.605e+00  1.436e+00  -6.687 2.36e-11 ***
## cat.Forks               7.998e+00  5.256e+00   1.522 0.128092
## cat.Envelopes          -3.361e+01  4.243e+00  -7.922 2.52e-15 ***
```

```
## cat.Beds               -5.708e+00  1.607e+00  -3.551 0.000385 ***
## cat.Appliances         -1.303e+01  5.284e+00  -2.467 0.013655 *
## cat.Dining_Tables      -4.493e+00  1.740e+00  -2.583 0.009814 **
## cat.Wine_Glasses       -8.532e+00  5.318e+00  -1.604 0.108648
## cat.Bags               -3.143e+01  4.372e+00  -7.190 6.84e-13 ***
## cat.Gadgets            -3.614e+00  5.407e+00  -0.668 0.503883
## cat.Serving_Utensils   -3.424e+00  5.431e+00  -0.630 0.528466
## cat.Plates             -5.638e+00  5.545e+00  -1.017 0.309252
## cat.Knives             -1.159e+01  5.615e+00  -2.065 0.038954 *
## cat.Bowls              -4.563e+00  5.640e+00  -0.809 0.418534
## cat.Pots               -4.792e+00  5.681e+00  -0.844 0.398898
## cat.Casual_Glasses      6.144e+00  6.624e+00   0.928 0.353645
## col.Blue                6.956e+00  1.992e+01   0.349 0.726938
## col.Gray                5.615e+00  1.992e+01   0.282 0.777999
## col.Black               6.576e+00  1.992e+01   0.330 0.741287
## col.Green               4.323e+00  1.992e+01   0.217 0.828201
## col.Red                 5.842e+00  1.993e+01   0.293 0.769396
## col.Brown               5.731e+00  1.992e+01   0.288 0.773542
## col.Purple              4.101e+00  1.994e+01   0.206 0.837055
## col.Orange              6.878e+00  1.994e+01   0.345 0.730151
## col.Khaki               2.629e+00  1.994e+01   0.132 0.895112
## col.Yellow              4.535e+00  1.996e+01   0.227 0.820282
## col.White               2.382e+00  1.997e+01   0.119 0.905022
## col.Not_color           9.967e+00  2.001e+01   0.498 0.618348
## col.Multicolored        4.985e+00  2.002e+01   0.249 0.803423
## col.Pink                5.445e+00  2.004e+01   0.272 0.785797
## col.other               6.702e+00  2.005e+01   0.334 0.738177
## col.Metallic           -2.302e+01  2.229e+01  -1.033 0.301733
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.9 on 13805 degrees of freedom
## Multiple R-squared:  0.3054, Adjusted R-squared:  0.3032
## F-statistic: 134.9 on 45 and 13805 DF,  p-value: < 2.2e-16
```

Creating Train, Test, and Validation Subsets

```
# Subset the data: 70% training, 20% test, 10% validation
  # Create a dataframe of student ids and random values
sets <- data.frame(id = unique(resale_item_wide$id),
                   rand = runif(length(unique(resale_item_wide$id))))

  # Assign status based on unique values and merge into data
sets$set <- ifelse(sets$rand < 0.7, 'train', ifelse(sets$rand >= 0.9, 'validate', 'test'))
resale_item_wide <- merge(resale_item_wide, sets[, c('id', 'set')], by = 'id')

  # Subset by status
train <- resale_item_wide[resale_item_wide$set == "train",]
test <- resale_item_wide[resale_item_wide$set == "test",]
validate <- resale_item_wide[resale_item_wide$set == "validate",]

# Evaluate distributions of some variables we might want to stratify by
strats <- c("dpt", "cat", "col")
for (i in 1:length(strats)){
```

```
  print(table(resale_item_wide[, strats[i]])/nrow(resale_item_wide))
  print(table(train[, strats[i]])/nrow(train))
  print(table(test[, strats[i]])/nrow(test))
  print(table(validate[, strats[i]])/nrow(validate))
}
```

```
##
##          Furniture     Kids Furniture       Kids Storage Kitchen and Dining
##        0.011912497        0.127716410        0.136741030        0.407551801
##  Outdoor Furniture    Outdoor Storage            Storage
##        0.168435492        0.141578225        0.006064544
##
##          Furniture     Kids Furniture       Kids Storage Kitchen and Dining
##        0.012068076        0.128210418        0.140278494        0.409283136
##  Outdoor Furniture    Outdoor Storage            Storage
##        0.165755544        0.138421867        0.005982465
##
##          Furniture     Kids Furniture       Kids Storage Kitchen and Dining
##        0.008934954        0.128305933        0.125446748        0.407791279
##  Outdoor Furniture    Outdoor Storage            Storage
##        0.181200858        0.141887062        0.006433167
##
##          Furniture     Kids Furniture       Kids Storage Kitchen and Dining
##        0.016936672        0.122974963        0.134756996        0.394698085
##  Outdoor Furniture    Outdoor Storage            Storage
##        0.161266568        0.163475700        0.005891016
##
##       Appliances             Bags             Beds            Bowls
##      0.021009313      0.013573027      0.027290448      0.005559165
##            Boxes         Cabinets   Casual Glasses           Chairs
##      0.101725507      0.001732727      0.001660530      0.019059996
##            Desks    Dining Tables        Envelopes            Forks
##      0.057107790      0.019854162      0.030106130      0.031405675
##          Gadgets           Knives             Pans           Plates
##      0.010757346      0.005847953      0.001082954      0.006858711
##             Pots             Rugs Serving Utensils          Shelves
##      0.005414771      0.098043463      0.010107573      0.137246408
##      Side Tables           Spoons     Wine Glasses
##      0.086708541      0.290376146      0.017471663
##
##       Appliances             Bags             Beds            Bowls
##      0.021041774      0.014234141      0.027127385      0.005982465
##            Boxes         Cabinets   Casual Glasses           Chairs
##      0.101083032      0.002062919      0.001444043      0.019288293
##            Desks    Dining Tables        Envelopes            Forks
##      0.056730273      0.019700877      0.030221764      0.032800413
##          Gadgets           Knives             Pans           Plates
##      0.010727179      0.005776173      0.001340897      0.007529654
##             Pots             Rugs Serving Utensils          Shelves
##      0.005466735      0.097472924      0.010727179      0.137080970
##      Side Tables           Spoons     Wine Glasses
##      0.085714286      0.287880351      0.018566271
##
```

```
##       Appliances             Bags             Beds            Bowls
##     0.0214438885     0.0110793424     0.0264474625     0.0057183703
##            Boxes         Cabinets   Casual Glasses           Chairs
##     0.0997140815     0.0010721944     0.0025017870     0.0210864904
##            Desks    Dining Tables        Envelopes            Forks
##     0.0582558971     0.0196568978     0.0282344532     0.0257326662
##          Gadgets            Knives             Pans           Plates
##     0.0125089350     0.0053609721     0.0007147963     0.0042887777
##             Pots             Rugs Serving Utensils          Shelves
##     0.0050035740     0.1022158685     0.0082201573     0.1336669049
##      Side Tables           Spoons     Wine Glasses
##     0.0907791279     0.3002144389     0.0160829164
##
##       Appliances             Bags             Beds            Bowls
##      0.019882180      0.013991163      0.030191458      0.002209131
##            Boxes         Cabinets   Casual Glasses           Chairs
##      0.110456554      0.000736377      0.001472754      0.013254786
##            Desks    Dining Tables        Envelopes            Forks
##      0.057437408      0.021354934      0.033136966      0.033136966
##          Gadgets            Knives             Pans           Plates
##      0.007363770      0.007363770      0.000000000      0.007363770
##             Pots             Rugs Serving Utensils          Shelves
##      0.005891016      0.093519882      0.009572901      0.145802651
##      Side Tables           Spoons     Wine Glasses
##      0.085419735      0.287923417      0.012518409
##
##        Black         Blue        Brown         Gray        Green        Khaki
## 1.732727e-01 2.329074e-01 6.988665e-02 1.887228e-01 1.188362e-01 2.382499e-02
##     Metallic Multicolored    Not color       Orange        other         Pink
## 2.887878e-04 6.714317e-03 9.241210e-03 3.025052e-02 5.414771e-03 6.208938e-03
##       Purple          Red   Unassigned        White       Yellow
## 3.328280e-02 7.133059e-02 7.219695e-05 1.436719e-02 1.537795e-02
##
##        Black         Blue        Brown         Gray        Green        Khaki
## 0.1768953069 0.2296028881 0.0711707065 0.1872099020 0.1181021145 0.0225889634
##     Metallic Multicolored    Not color       Orange        other         Pink
## 0.0002062919 0.0071170707 0.0099020113 0.0289840124 0.0054667354 0.0067044868
##       Purple          Red   Unassigned        White       Yellow
## 0.0324909747 0.0710675606 0.0000000000 0.0155750387 0.0169159360
##
##        Black         Blue        Brown         Gray        Green        Khaki
## 0.1701215154 0.2419585418 0.0679056469 0.1890636169 0.1186561830 0.0278770550
##     Metallic Multicolored    Not color       Orange        other         Pink
## 0.0003573981 0.0057183703 0.0085775554 0.0296640457 0.0046461758 0.0039313796
##       Purple          Red   Unassigned        White       Yellow
## 0.0353824160 0.0704074339 0.0003573981 0.0117941387 0.0135811294
##
##        Black         Blue        Brown         Gray        Green        Khaki
##  0.153902798  0.237849779  0.064801178  0.198821797  0.124447717  0.024300442
##     Metallic Multicolored    Not color       Orange        other         Pink
##  0.000736377  0.005891016  0.005891016  0.040500736  0.006627393  0.007363770
##       Purple          Red   Unassigned        White       Yellow
##  0.034609720  0.075110457  0.000000000  0.011045655  0.008100147
```

- removing col.metallic due to singularity

```
#train the model
model <- lm(perc ~
              origin_price +
              days_on_mkt +
              pageviews +
              dpt.Kitchen_and_Dining + dpt.Outdoor_Furniture + dpt.Outdoor_Storage +
              dpt.Kids_Storage + dpt.Kids_Furniture + dpt.Furniture +
              cat.Spoons + cat.Shelves + cat.Boxes + cat.Rugs + cat.Side_Tables +
              cat.Desks + cat.Forks + cat.Envelopes + cat.Beds + cat.Appliances +
              cat.Dining_Tables + cat.Wine_Glasses + cat.Bags + cat.Gadgets +
              cat.Serving_Utensils + cat.Plates + cat.Knives + cat.Bowls + cat.Pots +
              cat.Casual_Glasses +
              col.Blue + col.Gray + col.Black + col.Green + col.Red + col.Brown + col.Purple +
              col.Orange + col.Khaki + col.Yellow + col.White + col.Not_color + col.Multicolored +
              col.Pink + col.other,
            data = train)
summary(model)
```

```
##
## Call:
## lm(formula = perc ~ origin_price + days_on_mkt + pageviews +
##     dpt.Kitchen_and_Dining + dpt.Outdoor_Furniture + dpt.Outdoor_Storage +
##     dpt.Kids_Storage + dpt.Kids_Furniture + dpt.Furniture + cat.Spoons +
##     cat.Shelves + cat.Boxes + cat.Rugs + cat.Side_Tables + cat.Desks +
##     cat.Forks + cat.Envelopes + cat.Beds + cat.Appliances + cat.Dining_Tables +
##     cat.Wine_Glasses + cat.Bags + cat.Gadgets + cat.Serving_Utensils +
##     cat.Plates + cat.Knives + cat.Bowls + cat.Pots + cat.Casual_Glasses +
##     col.Blue + col.Gray + col.Black + col.Green + col.Red + col.Brown +
##     col.Purple + col.Orange + col.Khaki + col.Yellow + col.White +
##     col.Not_color + col.Multicolored + col.Pink + col.other,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.503 -15.566  -2.405  15.387  97.112
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.290e+01  1.511e+01   2.839 0.004531 **
## origin_price            -7.697e-02  2.258e-03 -34.082  < 2e-16 ***
## days_on_mkt             -2.731e-02  6.996e-03  -3.904 9.53e-05 ***
## pageviews                1.037e-04  1.366e-04   0.759 0.448016
## dpt.Kitchen_and_Dining -2.688e+00  7.665e+00  -0.351 0.725845
## dpt.Outdoor_Furniture  -2.092e+01  5.470e+00  -3.824 0.000132 ***
## dpt.Outdoor_Storage     8.489e+00  2.713e+00   3.129 0.001761 **
## dpt.Kids_Storage        1.089e+01  2.732e+00   3.985 6.81e-05 ***
## dpt.Kids_Furniture     -1.789e+01  5.491e+00  -3.258 0.001125 **
## dpt.Furniture          -2.344e+01  5.755e+00  -4.073 4.68e-05 ***
## cat.Spoons              4.161e+00  5.590e+00   0.744 0.456702
## cat.Shelves            -3.224e+01  4.571e+00  -7.054 1.86e-12 ***
## cat.Boxes              -3.472e+01  4.576e+00  -7.588 3.54e-14 ***
## cat.Rugs                1.441e+00  1.636e+00   0.881 0.378390
```

```
## cat.Side_Tables        -3.417e+00  1.625e+00  -2.103 0.035531 *
## cat.Desks              -9.485e+00  1.714e+00  -5.532 3.24e-08 ***
## cat.Forks               7.352e+00  5.693e+00   1.292 0.196548
## cat.Envelopes          -3.751e+01  4.704e+00  -7.974 1.71e-15 ***
## cat.Beds               -6.608e+00  1.923e+00  -3.437 0.000591 ***
## cat.Appliances         -1.277e+01  5.735e+00  -2.228 0.025935 *
## cat.Dining_Tables      -4.255e+00  2.084e+00  -2.042 0.041209 *
## cat.Wine_Glasses       -9.183e+00  5.769e+00  -1.592 0.111494
## cat.Bags               -3.472e+01  4.859e+00  -7.146 9.57e-13 ***
## cat.Gadgets            -2.654e+00  5.898e+00  -0.450 0.652777
## cat.Serving_Utensils   -2.704e+00  5.909e+00  -0.458 0.647302
## cat.Plates             -4.385e+00  6.033e+00  -0.727 0.467382
## cat.Knives             -1.070e+01  6.178e+00  -1.732 0.083245 .
## cat.Bowls              -6.132e+00  6.163e+00  -0.995 0.319749
## cat.Pots               -2.628e+00  6.255e+00  -0.420 0.674456
## cat.Casual_Glasses      6.092e+00  7.718e+00   0.789 0.429995
## col.Blue                3.844e+01  1.417e+01   2.713 0.006688 **
## col.Gray                3.706e+01  1.417e+01   2.615 0.008930 **
## col.Black               3.798e+01  1.418e+01   2.679 0.007400 **
## col.Green               3.613e+01  1.417e+01   2.549 0.010807 *
## col.Red                 3.691e+01  1.418e+01   2.602 0.009280 **
## col.Brown               3.792e+01  1.419e+01   2.673 0.007537 **
## col.Purple              3.524e+01  1.421e+01   2.479 0.013180 *
## col.Orange              4.034e+01  1.421e+01   2.839 0.004531 **
## col.Khaki               3.328e+01  1.422e+01   2.340 0.019282 *
## col.Yellow              3.658e+01  1.424e+01   2.568 0.010231 *
## col.White               3.281e+01  1.426e+01   2.300 0.021444 *
## col.Not_color           4.269e+01  1.433e+01   2.978 0.002906 **
## col.Multicolored        3.756e+01  1.437e+01   2.615 0.008939 **
## col.Pink                3.684e+01  1.439e+01   2.561 0.010466 *
## col.other               3.886e+01  1.443e+01   2.692 0.007110 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.96 on 9650 degrees of freedom
## Multiple R-squared:  0.3084, Adjusted R-squared:  0.3053
## F-statistic: 97.82 on 44 and 9650 DF,  p-value: < 2.2e-16
```

```
#constraint the model
constrained_model <- ols_step_backward_p(model)$model
```

```
## Backward Elimination Method
## ---------------------------
##
## Candidate Terms:
##
## 1 . origin_price
## 2 . days_on_mkt
## 3 . pageviews
## 4 . dpt.Kitchen_and_Dining
## 5 . dpt.Outdoor_Furniture
## 6 . dpt.Outdoor_Storage
## 7 . dpt.Kids_Storage
## 8 . dpt.Kids_Furniture
```

```
## 9 . dpt.Furniture
## 10 . cat.Spoons
## 11 . cat.Shelves
## 12 . cat.Boxes
## 13 . cat.Rugs
## 14 . cat.Side_Tables
## 15 . cat.Desks
## 16 . cat.Forks
## 17 . cat.Envelopes
## 18 . cat.Beds
## 19 . cat.Appliances
## 20 . cat.Dining_Tables
## 21 . cat.Wine_Glasses
## 22 . cat.Bags
## 23 . cat.Gadgets
## 24 . cat.Serving_Utensils
## 25 . cat.Plates
## 26 . cat.Knives
## 27 . cat.Bowls
## 28 . cat.Pots
## 29 . cat.Casual_Glasses
## 30 . col.Blue
## 31 . col.Gray
## 32 . col.Black
## 33 . col.Green
## 34 . col.Red
## 35 . col.Brown
## 36 . col.Purple
## 37 . col.Orange
## 38 . col.Khaki
## 39 . col.Yellow
## 40 . col.White
## 41 . col.Not_color
## 42 . col.Multicolored
## 43 . col.Pink
## 44 . col.other
##
## We are eliminating variables based on p value...
##
## Variables Removed:
##
## - dpt.Kitchen_and_Dining
## - cat.Casual_Glasses
## - cat.Spoons
## - pageviews
## - cat.Rugs
##
## No more variables satisfy the condition of p value = 0.3
##
##
## Final Model Output
## ------------------
##
##                            Model Summary
```

```
## ----------------------------------------------------------------
## R                       0.555       RMSE                19.958
## R-Squared               0.308       Coef. Var           36.769
## Adj. R-Squared          0.305       MSE                398.306
## Pred R-Squared          0.302       MAE                 16.851
## ----------------------------------------------------------------
##  RMSE: Root Mean Square Error
##  MSE: Mean Square Error
##  MAE: Mean Absolute Error
##
##                                 ANOVA
## --------------------------------------------------------------------------
##                   Sum of
##                  Squares          DF    Mean Square       F          Sig.
## --------------------------------------------------------------------------
## Regression    1713855.160          39      43945.004     110.33    0.0000
## Residual      3845648.708        9655        398.306
## Total         5559503.868        9694
## --------------------------------------------------------------------------
##
##                                     Parameter Estimates
## ----------------------------------------------------------------------------------------------------
##                 model       Beta    Std. Error    Std. Beta       t        Sig       lower      upper
## ----------------------------------------------------------------------------------------------------
##           (Intercept)     44.590        14.164                3.148    0.002     16.826     72.355
##          origin_price     -0.077         0.002       -0.326  -34.592    0.000     -0.081     -0.07
##           days_on_mkt     -0.027         0.007       -0.036   -3.879    0.000     -0.041     -0.013
## dpt.Outdoor_Furniture    -21.358         0.841       -0.332  -25.386    0.000    -23.007    -19.709
##   dpt.Outdoor_Storage      8.048         2.363        0.116    3.405    0.001      3.415     12.68
##      dpt.Kids_Storage     10.425         2.362        0.151    4.413    0.000      5.794     15.055
##    dpt.Kids_Furniture    -18.301         0.889       -0.256  -20.577    0.000    -20.045    -16.558
##         dpt.Furniture    -23.909         1.939       -0.109  -12.327    0.000    -27.710    -20.107
##           cat.Shelves    -33.405         2.386       -0.480  -14.002    0.000    -38.082    -28.729
##             cat.Boxes    -35.918         2.384       -0.452  -15.067    0.000    -40.591    -31.245
##       cat.Side_Tables     -4.638         0.942       -0.054   -4.923    0.000     -6.485     -2.791
##             cat.Desks    -10.748         1.092       -0.104   -9.841    0.000    -12.889     -8.607
##             cat.Forks      3.013         1.227        0.022    2.456    0.014      0.608      5.418
##         cat.Envelopes    -38.742         2.494       -0.277  -15.533    0.000    -43.631    -33.852
##              cat.Beds     -7.844         1.393       -0.053   -5.631    0.000    -10.574     -5.113
##        cat.Appliances    -17.102         1.462       -0.103  -11.694    0.000    -19.969    -14.236
##     cat.Dining_Tables     -5.505         1.613       -0.032   -3.412    0.001     -8.668     -2.343
##      cat.Wine_Glasses    -13.537         1.542       -0.076   -8.776    0.000    -16.560    -10.513
##              cat.Bags    -35.945         2.877       -0.178  -12.496    0.000    -41.584    -30.306
##           cat.Gadgets     -6.944         1.999       -0.030   -3.473    0.001    -10.863     -3.025
##  cat.Serving_Utensils     -7.043         2.050       -0.030   -3.437    0.001    -11.061     -3.026
##             cat.Plates     -8.727         2.391       -0.032   -3.651    0.000    -13.413     -4.041
##             cat.Knives    -15.053         2.704       -0.048   -5.566    0.000    -20.355     -9.751
##              cat.Bowls    -10.502         2.681       -0.034   -3.917    0.000    -15.758     -5.247
##               cat.Pots     -6.951         2.925       -0.021   -2.377    0.017    -12.685     -1.218
##              col.Blue      38.454        14.170        0.675    2.714    0.007     10.677     66.231
##              col.Gray      37.046        14.168        0.603    2.615    0.009      9.273     64.819
##             col.Black      37.909        14.174        0.604    2.674    0.007     10.124     65.693
##             col.Green      36.147        14.171        0.487    2.551    0.011      8.369     63.926
##               col.Red      36.924        14.182        0.396    2.604    0.009      9.124     64.725
```

```
##              col.Brown     37.923        14.184        0.407      2.674    0.008     10.119     65.727
##             col.Purple     35.204        14.211        0.261      2.477    0.013      7.348     63.061
##             col.Orange     40.337        14.206        0.283      2.839    0.005     12.491     68.184
##              col.Khaki     33.310        14.218        0.207      2.343    0.019      5.441     61.180
##             col.Yellow     36.555        14.241        0.197      2.567    0.010      8.640     64.470
##              col.White     32.738        14.259        0.169      2.296    0.022      4.787     60.689
##          col.Not_color     42.623        14.331        0.176      2.974    0.003     14.530     70.716
##       col.Multicolored     37.259        14.361        0.131      2.594    0.009      9.108     65.410
##               col.Pink     36.733        14.384        0.125      2.554    0.011      8.537     64.928
##              col.other     38.871        14.431        0.120      2.694    0.007     10.584     67.159
## ------------------------------------------------------------------------------------------------------
```

```
summary(constrained_model)
```

```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.913 -15.540  -2.438  15.433  96.720
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             44.590493  14.164238   3.148 0.001648 **
## origin_price            -0.076698   0.002217 -34.592  < 2e-16 ***
## days_on_mkt             -0.026953   0.006948  -3.879 0.000106 ***
## dpt.Outdoor_Furniture -21.357659   0.841314 -25.386  < 2e-16 ***
## dpt.Outdoor_Storage     8.048069   2.363339   3.405 0.000663 ***
## dpt.Kids_Storage       10.424638   2.362313   4.413 1.03e-05 ***
## dpt.Kids_Furniture    -18.301448   0.889428 -20.577  < 2e-16 ***
## dpt.Furniture         -23.908651   1.939487 -12.327  < 2e-16 ***
## cat.Shelves           -33.405121   2.385675 -14.002  < 2e-16 ***
## cat.Boxes             -35.917829   2.383817 -15.067  < 2e-16 ***
## cat.Side_Tables        -4.638094   0.942171  -4.923 8.67e-07 ***
## cat.Desks             -10.747788   1.092116  -9.841  < 2e-16 ***
## cat.Forks               3.013365   1.226858   2.456 0.014060 *
## cat.Envelopes         -38.741591   2.494212 -15.533  < 2e-16 ***
## cat.Beds               -7.843607   1.392820  -5.631 1.84e-08 ***
## cat.Appliances        -17.102493   1.462459 -11.694  < 2e-16 ***
## cat.Dining_Tables      -5.505093   1.613318  -3.412 0.000647 ***
## cat.Wine_Glasses      -13.536639   1.542462  -8.776  < 2e-16 ***
## cat.Bags              -35.944887   2.876585 -12.496  < 2e-16 ***
## cat.Gadgets            -6.944337   1.999262  -3.473 0.000516 ***
## cat.Serving_Utensils   -7.043396   2.049569  -3.437 0.000592 ***
## cat.Plates             -8.726891   2.390511  -3.651 0.000263 ***
## cat.Knives            -15.053306   2.704445  -5.566 2.67e-08 ***
## cat.Bowls             -10.502268   2.681021  -3.917 9.02e-05 ***
## cat.Pots               -6.951364   2.924899  -2.377 0.017492 *
## col.Blue               38.453884  14.170487   2.714 0.006666 **
## col.Gray               37.045899  14.168400   2.615 0.008945 **
## col.Black              37.908601  14.174235   2.674 0.007497 **
## col.Green              36.147330  14.171240   2.551 0.010764 *
```

```
## col.Red                36.924138  14.182364   2.604 0.009241 **
## col.Brown              37.922839  14.184103   2.674 0.007516 **
## col.Purple             35.204448  14.210842   2.477 0.013255 *
## col.Orange             40.337195  14.205884   2.839 0.004528 **
## col.Khaki              33.310323  14.217521   2.343 0.019154 *
## col.Yellow             36.555074  14.241011   2.567 0.010277 *
## col.White              32.737944  14.259251   2.296 0.021702 *
## col.Not_color          42.622988  14.331440   2.974 0.002946 **
## col.Multicolored       37.258989  14.361266   2.594 0.009490 **
## col.Pink               36.732513  14.383823   2.554 0.010673 *
## col.other              38.871258  14.430856   2.694 0.007080 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.96 on 9655 degrees of freedom
## Multiple R-squared:  0.3083, Adjusted R-squared:  0.3055
## F-statistic: 110.3 on 39 and 9655 DF,  p-value: < 2.2e-16
```

```
coefficients <- constrained_model$coefficients[!is.na(constrained_model$coefficients)]
pred_perc <- function(obs, coefficients){
  pred <- rbind.fill(obs[names(train) %in% names(coefficients)],
                     as.data.frame(t(coefficients))) %>% t %>% as.data.frame %>% subset(!is.na(V1))
  pred$product <- pred$V1 * pred$V2
  sum(pred$product, unname(constrained_model$coefficients["(Intercept)"]))
}


# Test prediction function on a single row from the 'train' data set
i <- 3  #row number
round1_df <- data.frame(Actual = constrained_model$model$perc,
                        Prediction = constrained_model$fitted.values)
round1_df[i,]
```

```
##     Actual Prediction
## 5 54.74792   47.85603
```

c. How well does your model perform on the test set compared to the training set? Are you able to improve performance in any way?

- Since the RMSE of traning data set is pretty close to the one of testing set in round #1, 0.83X vs 0.84X. No further fine tune is necessary.
- Nonetheless, let us try on putting a lower p-value, 0.1 anyway to check. The result has gotten worse-off, 0.83X vs >1.X, so we will keep the round 1 constraint model.

Round #1

```
#check RMSE on TRAINING aata set round 1
train_rmse <- rmse(constrained_model$model$perc, constrained_model$fitted.values)
train_rmse/sd(constrained_model$model$perc)
```

```
## [1] 0.8316573
```

```r
# Calculate predicted values for test data set
test_pred <- ldply(lapply(split(test, test$id), function(x) {
  data.frame(pred = pred_perc(x, coefficients), actual = x$perc)
}), rbind)
test_pred <- test_pred[!is.na(test_pred$pred) & !is.na(test_pred$actual),]
test_rmse <- rmse(test_pred$actual, test_pred$pred)

#check RMSE on TESTING data set round 1
test_rmse/sd(test_pred$actual)
```

```
## [1] 0.8413432
```

Round #2

```r
# Increase p-value requirement
constrained_model_2 <- ols_step_backward_p(model, prem = 0.1, details = F)$model
```

```
## Backward Elimination Method
## ---------------------------
##
## Candidate Terms:
##
## 1 . origin_price
## 2 . days_on_mkt
## 3 . pageviews
## 4 . dpt.Kitchen_and_Dining
## 5 . dpt.Outdoor_Furniture
## 6 . dpt.Outdoor_Storage
## 7 . dpt.Kids_Storage
## 8 . dpt.Kids_Furniture
## 9 . dpt.Furniture
## 10 . cat.Spoons
## 11 . cat.Shelves
## 12 . cat.Boxes
## 13 . cat.Rugs
## 14 . cat.Side_Tables
## 15 . cat.Desks
## 16 . cat.Forks
## 17 . cat.Envelopes
## 18 . cat.Beds
## 19 . cat.Appliances
## 20 . cat.Dining_Tables
## 21 . cat.Wine_Glasses
## 22 . cat.Bags
## 23 . cat.Gadgets
## 24 . cat.Serving_Utensils
## 25 . cat.Plates
## 26 . cat.Knives
## 27 . cat.Bowls
## 28 . cat.Pots
## 29 . cat.Casual_Glasses
## 30 . col.Blue
```

```
## 31 . col.Gray
## 32 . col.Black
## 33 . col.Green
## 34 . col.Red
## 35 . col.Brown
## 36 . col.Purple
## 37 . col.Orange
## 38 . col.Khaki
## 39 . col.Yellow
## 40 . col.White
## 41 . col.Not_color
## 42 . col.Multicolored
## 43 . col.Pink
## 44 . col.other
##
## We are eliminating variables based on p value...
##
## Variables Removed:
##
## - dpt.Kitchen_and_Dining
## - cat.Casual_Glasses
## - cat.Spoons
## - pageviews
## - cat.Rugs
##
## No more variables satisfy the condition of p value = 0.1
##
##
## Final Model Output
## ------------------
##
##                            Model Summary
## ----------------------------------------------------------------
## R                        0.555       RMSE                19.958
## R-Squared                0.308       Coef. Var           36.769
## Adj. R-Squared           0.305       MSE                398.306
## Pred R-Squared           0.302       MAE                 16.851
## ----------------------------------------------------------------
##  RMSE: Root Mean Square Error
##  MSE: Mean Square Error
##  MAE: Mean Absolute Error
##
##                                  ANOVA
## --------------------------------------------------------------------------------
##                     Sum of
##                    Squares          DF    Mean Square       F         Sig.
## --------------------------------------------------------------------------------
## Regression    1713855.160          39      43945.004     110.33    0.0000
## Residual      3845648.708        9655        398.306
## Total         5559503.868        9694
## --------------------------------------------------------------------------------
##
##                                        Parameter Estimates
## ----------------------------------------------------------------------------------------------------
```

```
##                 model       Beta    Std. Error    Std. Beta       t        Sig       lower       upper
## ---------------------------------------------------------------------------------------------------------
##           (Intercept)     44.590        14.164                3.148    0.002     16.826     72.355
##          origin_price     -0.077         0.002       -0.326  -34.592    0.000     -0.081     -0.072
##           days_on_mkt     -0.027         0.007       -0.036   -3.879    0.000     -0.041     -0.013
## dpt.Outdoor_Furniture    -21.358         0.841       -0.332  -25.386    0.000    -23.007    -19.709
##   dpt.Outdoor_Storage      8.048         2.363        0.116    3.405    0.001      3.415     12.681
##      dpt.Kids_Storage     10.425         2.362        0.151    4.413    0.000      5.794     15.055
##    dpt.Kids_Furniture    -18.301         0.889       -0.256  -20.577    0.000    -20.045    -16.558
##         dpt.Furniture    -23.909         1.939       -0.109  -12.327    0.000    -27.710    -20.107
##           cat.Shelves    -33.405         2.386       -0.480  -14.002    0.000    -38.082    -28.729
##             cat.Boxes    -35.918         2.384       -0.452  -15.067    0.000    -40.591    -31.245
##       cat.Side_Tables     -4.638         0.942       -0.054   -4.923    0.000     -6.485     -2.791
##             cat.Desks    -10.748         1.092       -0.104   -9.841    0.000    -12.889     -8.607
##             cat.Forks      3.013         1.227        0.022    2.456    0.014      0.608      5.418
##         cat.Envelopes    -38.742         2.494       -0.277  -15.533    0.000    -43.631    -33.852
##              cat.Beds     -7.844         1.393       -0.053   -5.631    0.000    -10.574     -5.113
##        cat.Appliances    -17.102         1.462       -0.103  -11.694    0.000    -19.969    -14.236
##     cat.Dining_Tables     -5.505         1.613       -0.032   -3.412    0.001     -8.668     -2.343
##      cat.Wine_Glasses    -13.537         1.542       -0.076   -8.776    0.000    -16.560    -10.513
##             cat.Bags     -35.945         2.877       -0.178  -12.496    0.000    -41.584    -30.306
##           cat.Gadgets     -6.944         1.999       -0.030   -3.473    0.001    -10.863     -3.025
##  cat.Serving_Utensils     -7.043         2.050       -0.030   -3.437    0.001    -11.061     -3.026
##            cat.Plates     -8.727         2.391       -0.032   -3.651    0.000    -13.413     -4.041
##            cat.Knives    -15.053         2.704       -0.048   -5.566    0.000    -20.355     -9.752
##             cat.Bowls    -10.502         2.681       -0.034   -3.917    0.000    -15.758     -5.247
##              cat.Pots     -6.951         2.925       -0.021   -2.377    0.017    -12.685     -1.218
##              col.Blue     38.454        14.170        0.675    2.714    0.007     10.677     66.231
##              col.Gray     37.046        14.168        0.603    2.615    0.009      9.273     64.819
##             col.Black     37.909        14.174        0.604    2.674    0.007     10.124     65.693
##             col.Green     36.147        14.171        0.487    2.551    0.011      8.369     63.926
##               col.Red     36.924        14.182        0.396    2.604    0.009      9.124     64.725
##             col.Brown     37.923        14.184        0.407    2.674    0.008     10.119     65.727
##            col.Purple     35.204        14.211        0.261    2.477    0.013      7.348     63.061
##            col.Orange     40.337        14.206        0.283    2.839    0.005     12.491     68.184
##             col.Khaki     33.310        14.218        0.207    2.343    0.019      5.441     61.180
##            col.Yellow     36.555        14.241        0.197    2.567    0.010      8.640     64.470
##             col.White     32.738        14.259        0.169    2.296    0.022      4.787     60.689
##         col.Not_color     42.623        14.331        0.176    2.974    0.003     14.530     70.716
##      col.Multicolored     37.259        14.361        0.131    2.594    0.009      9.108     65.410
##              col.Pink     36.733        14.384        0.125    2.554    0.011      8.537     64.928
##             col.other     38.871        14.431        0.120    2.694    0.007     10.584     67.159
## ---------------------------------------------------------------------------------------------------------
```

```
#check RMSE on TRAINING data set round 2
train_rmse_2 <- rmse(constrained_model_2$model$perc, constrained_model_2$fitted.values)
train_rmse_2/sd(constrained_model$model$perc)
```

```
## [1] 0.8316573
```

```
# Calculate predicted values for test data set
coefficients <- constrained_model_2$coefficients[!is.na(constrained_model_2$coefficients)]
test_pred <- ldply(lapply(split(test, test$id), function(x) {
```

```
  data.frame(pred = pred_perc(x, coefficients), actual = x$perc)
}), rbind)
test_pred <- test_pred[!is.na(test_pred$pred) & !is.na(test_pred$actual),]
test_rmse_2 <- rmse(test_pred$actual, test_pred$pred)

#check RMSE on TESTING data set round 2
test_rmse_2/sd(test_pred$actual)
```

```
## [1] 0.8413432
```