Product Sales and Promotions Analysis

#Importing Libraries  
library(readxl)  
library(data.table)

## Warning: package 'data.table' was built under R version 4.0.3

library(stargazer)

## Warning: package 'stargazer' was built under R version 4.0.3

##   
## Please cite as:

## Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

## R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

#Setting the Working Directory and Importing the Dataset  
setwd("C:/Users/surya/Downloads")  
  
rc\_stores <- read\_excel("RetailChain.xlsx", sheet = "stores")  
names(rc\_stores) <- tolower(colnames(rc\_stores))  
rc\_products <- read\_excel("RetailChain.xlsx", sheet = "products")  
names(rc\_products) <- tolower(colnames(rc\_products))  
rc\_transactions <- read\_excel("RetailChain.xlsx", sheet = "transactions")  
names(rc\_transactions) <- tolower(colnames(rc\_transactions))  
#attach(rc\_stores)  
#attach(rc\_products)  
#attach(rc\_transactions)  
  
#NA Values Column-Wise & Pre-Processing  
sapply(rc\_transactions, function(x) sum(is.na(x)))

## week\_end\_date store\_num upc units visits   
## 0 0 0 0 0   
## hhs spend price base\_price feature   
## 0 0 23 185 0   
## display tpr\_only   
## 0 0

colSums(is.na(rc\_transactions))

## week\_end\_date store\_num upc units visits   
## 0 0 0 0 0   
## hhs spend price base\_price feature   
## 0 0 23 185 0   
## display tpr\_only   
## 0 0

rc\_transactions <- rc\_transactions[complete.cases(rc\_transactions), ]  
str(rc\_transactions)

## tibble [524,742 x 12] (S3: tbl\_df/tbl/data.frame)  
## $ week\_end\_date: POSIXct[1:524742], format: "2009-01-14" "2009-01-14" ...  
## $ store\_num : num [1:524742] 367 367 367 367 367 367 367 367 367 367 ...  
## $ upc : num [1:524742] 1.11e+09 1.11e+09 1.11e+09 1.11e+09 1.11e+09 ...  
## $ units : num [1:524742] 13 20 14 4 3 2 14 29 35 50 ...  
## $ visits : num [1:524742] 13 18 14 3 3 2 13 26 27 40 ...  
## $ hhs : num [1:524742] 13 18 14 3 3 2 13 25 25 40 ...  
## $ spend : num [1:524742] 18.1 27.8 19.3 14 7.5 ...  
## $ price : num [1:524742] 1.39 1.39 1.38 3.5 2.5 2.59 1.88 1.88 1.98 3.36 ...  
## $ base\_price : num [1:524742] 1.57 1.39 1.38 4.49 2.5 2.59 1.88 1.88 1.98 3.94 ...  
## $ feature : num [1:524742] 0 0 0 0 0 0 0 0 0 0 ...  
## $ display : num [1:524742] 0 0 0 0 0 0 0 0 0 1 ...  
## $ tpr\_only : num [1:524742] 1 0 0 1 0 0 0 0 0 0 ...

#Remove Oral Hygiene Products  
ohp\_temp <- rc\_products[rc\_products$category == "ORAL HYGIENE PRODUCTS", ]  
rc\_transactions <- rc\_transactions[!(rc\_transactions$upc %in% ohp\_temp$upc), ]  
  
#Extracting Year, Month & Week Number  
rc\_transactions$year <- format(rc\_transactions$week\_end\_date, "%Y")  
rc\_transactions$month <- format(rc\_transactions$week\_end\_date, "%B")  
rc\_transactions$month <- as.factor(rc\_transactions$month)  
rc\_transactions$month <- relevel(rc\_transactions$month, "January")  
#rc\_transactions$weeknum <- strftime(rc\_transactions$week\_end\_date, format = "%V")  
  
library(lubridate)

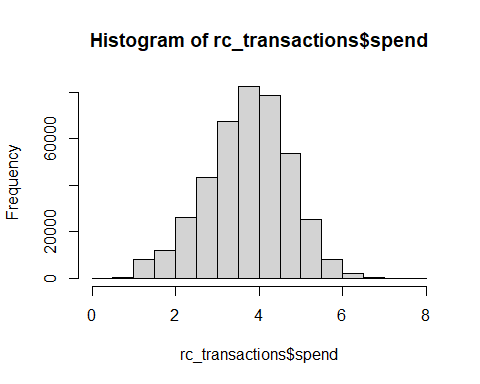
## Warning: package 'lubridate' was built under R version 4.0.3

##   
## Attaching package: 'lubridate'

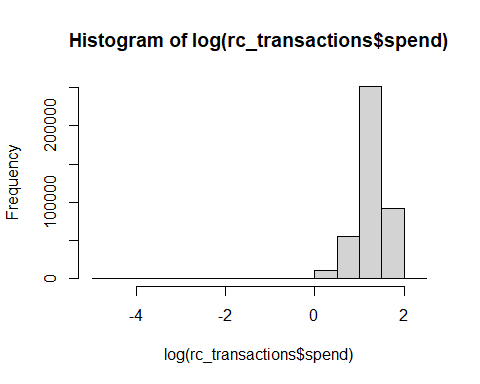
## The following objects are masked from 'package:data.table':  
##   
## hour, isoweek, mday, minute, month, quarter, second, wday, week,  
## yday, year

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

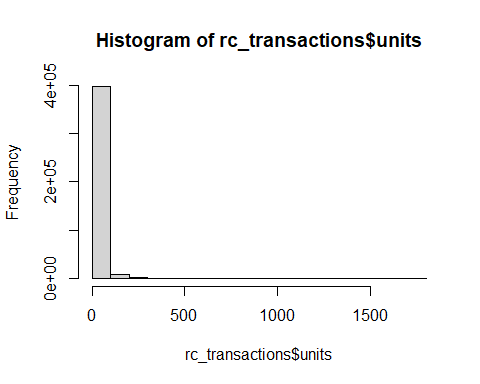
rc\_transactions$weeknum <- (interval(min(rc\_transactions$week\_end\_date), rc\_transactions$week\_end\_date) %/% weeks(1)) + 1  
rc\_transactions$spend <- log(rc\_transactions$spend + min(rc\_transactions$spend) + 1)  
  
#Factorizing Variables  
rc\_transactions$store\_num <- as.factor(rc\_transactions$store\_num)  
rc\_transactions$upc <- as.factor(rc\_transactions$upc)  
rc\_transactions$feature <- as.factor(rc\_transactions$feature)  
rc\_transactions$display <- as.factor(rc\_transactions$display)  
rc\_transactions$tpr\_only <- as.factor(rc\_transactions$tpr\_only)  
  
#Data Visualizations  
hist(rc\_transactions$spend)



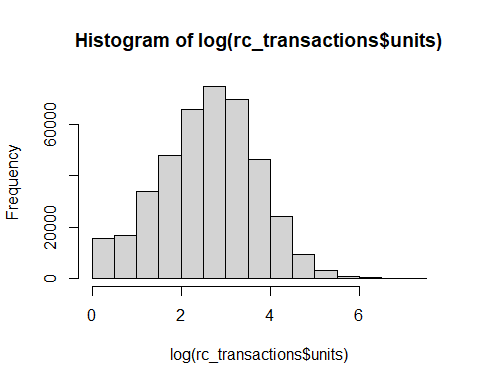
hist(log(rc\_transactions$spend))



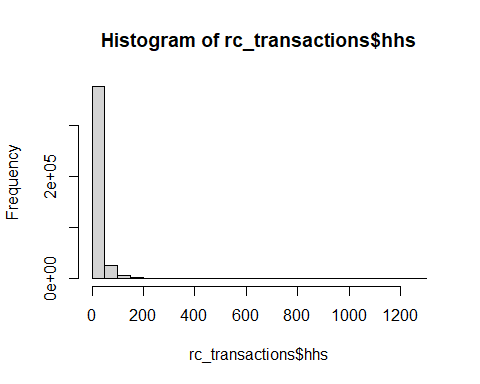
hist(rc\_transactions$units)



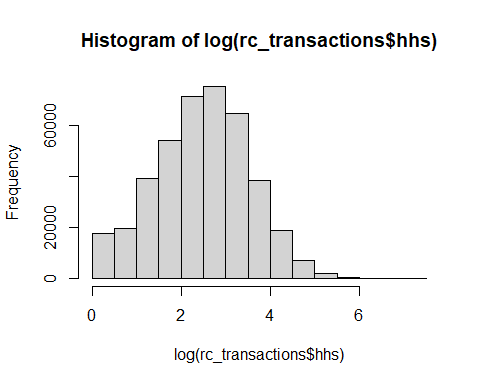
hist(log(rc\_transactions$units))



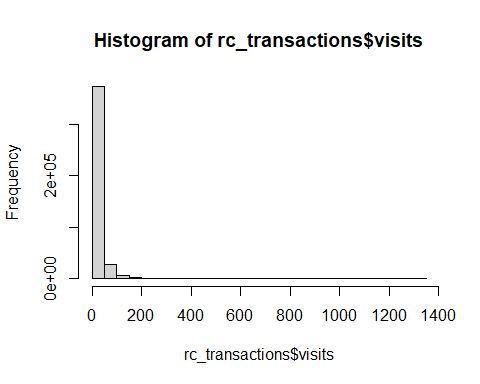
hist(rc\_transactions$hhs)



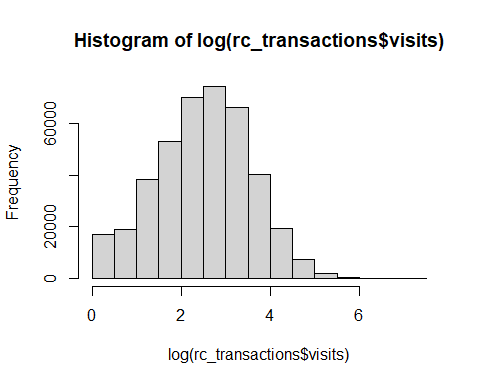
hist(log(rc\_transactions$hhs))



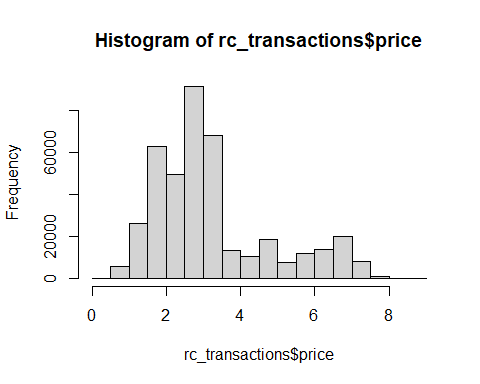
hist(rc\_transactions$visits)



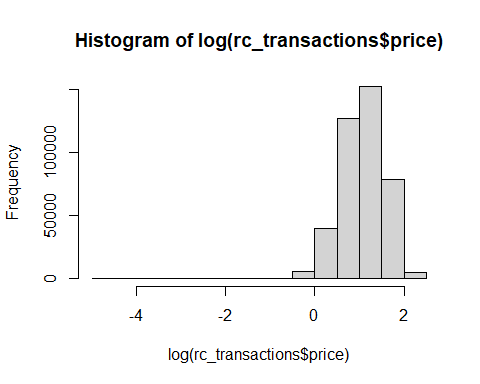
hist(log(rc\_transactions$visits))



hist(rc\_transactions$price)



hist(log(rc\_transactions$price))



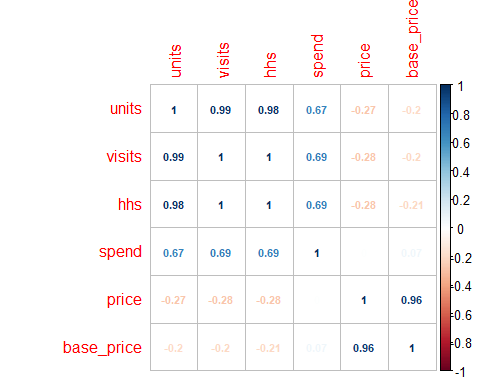
#Correlations  
rct\_corr <- rc\_transactions[, c(4:9)]  
  
#library(PerformanceAnalytics)  
#chart.Correlation(rct\_corr)  
  
library(corrplot)

## corrplot 0.84 loaded

rct\_corplot <- cor(rct\_corr)  
corrplot(rct\_corplot, method = "number", number.cex= 0.7)  
  
#Q1  
#Multi-Level Analysis  
library(lme4)

## Warning: package 'lme4' was built under R version 4.0.4

## Loading required package: Matrix



library(arm)

## Warning: package 'arm' was built under R version 4.0.4

## Loading required package: MASS

##   
## arm (Version 1.11-2, built: 2020-7-27)

## Working directory is C:/Users/surya/Downloads/Career/M.S. [2020-22]/3.Spring\_2021/[ISM6137.001S21.20445 Statistical Data Mining]/Week\_11/Assignment

##   
## Attaching package: 'arm'

## The following object is masked from 'package:corrplot':  
##   
## corrplot

any(is.na(log(rc\_transactions$spend)))

## [1] FALSE

any(is.nan(log(rc\_transactions$spend)))

## [1] FALSE

any(is.infinite(log(rc\_transactions$spend)))

## [1] TRUE

spend\_model <- lmer(spend ~ log(visits) + feature + display + tpr\_only + price +   
 year + weeknum + (1 | store\_num), data = rc\_transactions)  
units\_model <- lmer(log(units) ~ log(visits) + feature + display + tpr\_only + price +   
 year + weeknum + (1 | store\_num), data = rc\_transactions)  
hhs\_model <- lmer(log(hhs) ~ log(visits) + feature + display + tpr\_only + price +   
 year + weeknum + (1 | store\_num), data = rc\_transactions)  
  
#Stargazer  
stargazer(spend\_model, units\_model, hhs\_model, type="text", single.row=TRUE)

##   
## ===============================================================================  
## Dependent variable:   
## -----------------------------------------------------------  
## spend log(units) log(hhs)   
## (1) (2) (3)   
## -------------------------------------------------------------------------------  
## log(visits) 0.956\*\*\* (0.0003) 1.000\*\*\* (0.0003) 0.985\*\*\* (0.0001)   
## feature1 0.094\*\*\* (0.001) 0.052\*\*\* (0.001) 0.014\*\*\* (0.001)   
## display1 -0.023\*\*\* (0.001) 0.008\*\*\* (0.001) 0.013\*\*\* (0.0005)   
## tpr\_only1 0.019\*\*\* (0.001) 0.063\*\*\* (0.001) 0.011\*\*\* (0.0004)   
## price 0.257\*\*\* (0.0002) -0.010\*\*\* (0.0002) -0.005\*\*\* (0.0001)   
## year2010 -0.005\*\*\* (0.001) -0.006\*\*\* (0.001) 0.001 (0.001)   
## year2011 -0.017\*\*\* (0.002) -0.015\*\*\* (0.002) 0.004\*\*\* (0.001)   
## year2012 -0.0001 (0.004) -0.006\* (0.003) 0.001 (0.002)   
## weeknum 0.0002\*\*\* (0.00002) 0.0001\*\*\* (0.00002) -0.00002\* (0.00001)  
## Constant 0.484\*\*\* (0.002) 0.130\*\*\* (0.002) 0.024\*\*\* (0.003)   
## -------------------------------------------------------------------------------  
## Observations 408,238 408,238 408,238   
## Log Likelihood 138,848.600 213,635.200 446,500.400   
## Akaike Inf. Crit. -277,673.200 -427,246.400 -892,976.900   
## Bayesian Inf. Crit. -277,542.100 -427,115.300 -892,845.800   
## ===============================================================================  
## Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

#Q2  
#Merging Data Frames  
step <- merge(rc\_transactions, rc\_products, by = "upc")  
merged\_rc <- merge(step, rc\_stores, by.x = "store\_num", by.y = "store\_id")  
str(merged\_rc)

## 'data.frame': 418555 obs. of 28 variables:  
## $ store\_num : Factor w/ 77 levels "367","387","389",..: 27 27 27 27 27 27 27 27 27 27 ...  
## $ upc : Factor w/ 42 levels "1111009477","1111009497",..: 5 10 9 21 4 18 23 17 23 4 ...  
## $ week\_end\_date : POSIXct, format: "2011-02-09" "2011-09-28" ...  
## $ units : num 19 24 25 7 8 2 308 20 244 6 ...  
## $ visits : num 16 24 14 7 7 2 281 17 227 6 ...  
## $ hhs : num 16 24 14 7 7 2 276 17 217 6 ...  
## $ spend : num 3.44 4.18 4.41 3.08 2.72 ...  
## $ price : num 1.59 2.68 3.26 2.95 1.78 2.89 1.98 2.18 1.99 1.59 ...  
## $ base\_price : num 1.59 2.68 3.26 2.95 1.78 2.89 2.89 2.89 2.9 1.59 ...  
## $ feature : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 2 1 ...  
## $ display : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 2 1 ...  
## $ tpr\_only : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ year : chr "2011" "2011" "2010" "2011" ...  
## $ month : Factor w/ 12 levels "January","April",..: 5 12 9 2 4 6 9 5 4 5 ...  
## $ weeknum : num 109 142 71 120 48 26 71 109 152 109 ...  
## $ description : chr "PL RAISIN BRAN" "GM HONEY NUT CHEERIOS" "PL SR CRUST PEPPRN PIZZA" "QKER CAP N CRUNCH" ...  
## $ manufacturer : chr "PRIVATE LABEL" "GENERAL MI" "PRIVATE LABEL" "QUAKER" ...  
## $ category : chr "COLD CEREAL" "COLD CEREAL" "FROZEN PIZZA" "COLD CEREAL" ...  
## $ sub\_category : chr "ADULT CEREAL" "ALL FAMILY CEREAL" "PIZZA/PREMIUM" "KIDS CEREAL" ...  
## $ product\_size : chr "20 OZ" "12.25 OZ" "29.6 OZ" "14 OZ" ...  
## $ store\_name : chr "AT EASTEX FRWY" "AT EASTEX FRWY" "AT EASTEX FRWY" "AT EASTEX FRWY" ...  
## $ city : chr "HOUSTON" "HOUSTON" "HOUSTON" "HOUSTON" ...  
## $ state : chr "TX" "TX" "TX" "TX" ...  
## $ msa : num 26420 26420 26420 26420 26420 ...  
## $ segment : chr "VALUE" "VALUE" "VALUE" "VALUE" ...  
## $ parking : num 221 221 221 221 221 221 221 221 221 221 ...  
## $ size : num 35675 35675 35675 35675 35675 ...  
## $ avg\_weekly\_baskets: num 19068 19068 19068 19068 19068 ...

merged\_rc <- merged\_rc[complete.cases(merged\_rc), ]  
  
merged\_rc$category <- as.factor(merged\_rc$category)  
merged\_rc$city <- as.factor(merged\_rc$city)  
merged\_rc$state <- as.factor(merged\_rc$state)  
merged\_rc$segment <- as.factor(merged\_rc$segment)  
  
#Multi-Level Analysis Based on Category & Segment  
spend\_model2 <- lmer(spend ~ log(visits) + feature + display + tpr\_only + price +   
 year + weeknum + segment + category + (1 | store\_num), data = merged\_rc)  
units\_model2 <- lmer(log(units) ~ log(visits) + feature + display + tpr\_only + price +   
 year + weeknum + segment + category + (1 | store\_num), data = merged\_rc)  
hhs\_model2 <- lmer(log(hhs) ~ log(visits) + feature + display + tpr\_only + price +   
 year + weeknum + segment + category + (1 | store\_num), data = merged\_rc)  
  
#Stargazer  
stargazer(spend\_model2, units\_model2, hhs\_model2, type="text", single.row=TRUE)

##   
## ================================================================================  
## Dependent variable:   
## -----------------------------------------------------------  
## spend log(units) log(hhs)   
## (1) (2) (3)   
## --------------------------------------------------------------------------------  
## log(visits) 0.937\*\*\* (0.001) 1.012\*\*\* (0.0005) 0.983\*\*\* (0.0004)   
## feature1 0.059\*\*\* (0.002) 0.016\*\*\* (0.002) 0.013\*\*\* (0.001)   
## display1 0.006\*\*\* (0.002) -0.013\*\*\* (0.001) 0.008\*\*\* (0.001)   
## tpr\_only1 0.012\*\*\* (0.001) 0.042\*\*\* (0.001) 0.008\*\*\* (0.001)   
## price 0.236\*\*\* (0.001) -0.037\*\*\* (0.0005) -0.020\*\*\* (0.0004)   
## year2010 -0.011\*\*\* (0.002) -0.009\*\*\* (0.002) 0.0003 (0.001)   
## year2011 -0.021\*\*\* (0.003) -0.017\*\*\* (0.003) 0.007\*\*\* (0.002)   
## year2012 -0.002 (0.007) -0.012\* (0.006) 0.001 (0.005)   
## weeknum 0.0002\*\*\* (0.00003) 0.0001\*\*\* (0.00003) -0.00004\* (0.00002)  
## segmentUPSCALE 0.001 (0.003) -0.001 (0.003) 0.0002 (0.002)   
## segmentVALUE -0.006 (0.009) 0.004 (0.003) -0.049\*\*\* (0.011)   
## categoryCOLD CEREAL 0.085\*\*\* (0.001) -0.025\*\*\* (0.001) -0.015\*\*\* (0.001)   
## categoryFROZEN PIZZA 0.120\*\*\* (0.002) 0.125\*\*\* (0.002) 0.055\*\*\* (0.001)   
## Constant 0.533\*\*\* (0.007) 0.172\*\*\* (0.003) 0.081\*\*\* (0.007)   
## --------------------------------------------------------------------------------  
## Observations 136,093 136,093 136,093   
## Log Likelihood 45,471.510 71,580.540 105,196.600   
## Akaike Inf. Crit. -90,911.030 -143,129.100 -210,361.300   
## Bayesian Inf. Crit. -90,753.890 -142,971.900 -210,204.100   
## ================================================================================  
## Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

#Q3  
#Five Most Price Elastic & Five Least Price Elastic  
elasticity\_model <- glm(units ~ price + upc, data = rc\_transactions, family = poisson(link = log))   
summary(elasticity\_model)

##   
## Call:  
## glm(formula = units ~ price + upc, family = poisson(link = log),   
## data = rc\_transactions)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -23.771 -2.602 -0.791 1.239 72.930   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.1212538 0.0015217 3365.44 <2e-16 \*\*\*  
## price -0.8520167 0.0007179 -1186.75 <2e-16 \*\*\*  
## upc1111009497 -0.2334462 0.0018369 -127.08 <2e-16 \*\*\*  
## upc1111009507 -0.9067250 0.0023315 -388.91 <2e-16 \*\*\*  
## upc1111085319 -0.5405056 0.0023296 -232.02 <2e-16 \*\*\*  
## upc1111085345 -0.2025788 0.0020782 -97.48 <2e-16 \*\*\*  
## upc1111085350 0.3497437 0.0019965 175.18 <2e-16 \*\*\*  
## upc1111087395 0.3419301 0.0033026 103.53 <2e-16 \*\*\*  
## upc1111087396 0.3712986 0.0032797 113.21 <2e-16 \*\*\*  
## upc1111087398 0.5898481 0.0030670 192.32 <2e-16 \*\*\*  
## upc1600027527 1.3862062 0.0019209 721.65 <2e-16 \*\*\*  
## upc1600027528 1.7813601 0.0027594 645.57 <2e-16 \*\*\*  
## upc1600027564 0.9241095 0.0020772 444.88 <2e-16 \*\*\*  
## upc2066200530 1.7700980 0.0064490 274.48 <2e-16 \*\*\*  
## upc2066200531 1.7975779 0.0071439 251.62 <2e-16 \*\*\*  
## upc2066200532 1.4820047 0.0092139 160.84 <2e-16 \*\*\*  
## upc2840002333 -0.5534338 0.0039319 -140.76 <2e-16 \*\*\*  
## upc2840004768 -0.0473095 0.0029795 -15.88 <2e-16 \*\*\*  
## upc2840004770 -0.3302716 0.0033233 -99.38 <2e-16 \*\*\*  
## upc3000006340 -0.0602225 0.0028464 -21.16 <2e-16 \*\*\*  
## upc3000006560 0.4592943 0.0022672 202.58 <2e-16 \*\*\*  
## upc3000006610 0.1260421 0.0024724 50.98 <2e-16 \*\*\*  
## upc3800031829 1.0254575 0.0023713 432.44 <2e-16 \*\*\*  
## upc3800031838 1.1337930 0.0020931 541.68 <2e-16 \*\*\*  
## upc3800039118 0.9481731 0.0021912 432.73 <2e-16 \*\*\*  
## upc7027312504 -0.7135593 0.0037217 -191.73 <2e-16 \*\*\*  
## upc7027316204 -0.5024945 0.0031578 -159.13 <2e-16 \*\*\*  
## upc7027316404 -0.4964281 0.0031632 -156.94 <2e-16 \*\*\*  
## upc7110410455 -1.2643368 0.0056640 -223.22 <2e-16 \*\*\*  
## upc7110410470 -1.3273755 0.0059628 -222.61 <2e-16 \*\*\*  
## upc7110410471 -1.5562879 0.0066798 -232.99 <2e-16 \*\*\*  
## upc7192100336 2.1845380 0.0041696 523.92 <2e-16 \*\*\*  
## upc7192100337 2.4930177 0.0039371 633.22 <2e-16 \*\*\*  
## upc7192100339 2.7972979 0.0037618 743.60 <2e-16 \*\*\*  
## upc7218063052 1.9458519 0.0047400 410.51 <2e-16 \*\*\*  
## upc7218063979 1.8647576 0.0048703 382.88 <2e-16 \*\*\*  
## upc7218063983 1.7101248 0.0053059 322.30 <2e-16 \*\*\*  
## upc7797502248 -0.8069794 0.0032753 -246.38 <2e-16 \*\*\*  
## upc7797508004 -0.1404243 0.0031738 -44.24 <2e-16 \*\*\*  
## upc7797508006 -0.2320555 0.0032251 -71.95 <2e-16 \*\*\*  
## upc88491201426 0.9219386 0.0025644 359.51 <2e-16 \*\*\*  
## upc88491201427 0.8330132 0.0026643 312.66 <2e-16 \*\*\*  
## upc88491212971 0.3680219 0.0023741 155.01 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 10399079 on 408237 degrees of freedom  
## Residual deviance: 5376355 on 408195 degrees of freedom  
## AIC: 7200527  
##   
## Number of Fisher Scoring iterations: 5

coeff <- data.frame(elasticity\_model$coefficients)  
coeff$upc <- row.names(coeff)  
row.names(coeff) <- NULL  
colnames(coeff) <- c("standing", "UPC")  
coeff$UPC <- as.character(coeff$UPC)  
coeff$UPC <- gsub("^.{0,3}", "", coeff$UPC)  
coeff <- coeff[3:43,]  
coeff$UPC <- as.numeric(coeff$UPC)  
coeff <- merge(coeff, rc\_products, by.x = "UPC", by.y = "upc")  
coeff <- coeff[order(coeff$standing, decreasing = TRUE), ]  
row.names(coeff) <- NULL  
head(coeff,5)

## UPC standing description manufacturer category  
## 1 7192100339 2.797298 DIGRN PEPP PIZZA TOMBSTONE FROZEN PIZZA  
## 2 7192100337 2.493018 DIGRN SUPREME PIZZA TOMBSTONE FROZEN PIZZA  
## 3 7192100336 2.184538 DIGIORNO THREE MEAT TOMBSTONE FROZEN PIZZA  
## 4 7218063052 1.945852 FRSC BRCK OVN ITL PEP PZ TONYS FROZEN PIZZA  
## 5 7218063979 1.864758 FRSC PEPPERONI PIZZA TONYS FROZEN PIZZA  
## sub\_category product\_size  
## 1 PIZZA/PREMIUM 28.3 OZ  
## 2 PIZZA/PREMIUM 32.7 OZ  
## 3 PIZZA/PREMIUM 29.8 OZ  
## 4 PIZZA/PREMIUM 22.7 OZ  
## 5 PIZZA/PREMIUM 27.35 OZ

tail(coeff,5)

## UPC standing description manufacturer category  
## 37 7797502248 -0.8069794 SNYDR PRETZEL RODS SNYDER S BAG SNACKS  
## 38 1111009507 -0.9067250 PL TWIST PRETZELS PRIVATE LABEL BAG SNACKS  
## 39 7110410455 -1.2643368 MKSL MINI TWIST PRETZELS MKSL BAG SNACKS  
## 40 7110410470 -1.3273755 MKSL DUTCH PRETZELS MKSL BAG SNACKS  
## 41 7110410471 -1.5562879 MKSL PRETZEL STICKS MKSL BAG SNACKS  
## sub\_category product\_size  
## 37 PRETZELS 10 OZ  
## 38 PRETZELS 15 OZ  
## 39 PRETZELS 16 OZ  
## 40 PRETZELS 16 OZ  
## 41 PRETZELS 16 OZ

#Assumptions  
#Multicollinearity  
library(car)

## Loading required package: carData

## Registered S3 methods overwritten by 'car':  
## method from  
## influence.merMod lme4  
## cooks.distance.influence.merMod lme4  
## dfbeta.influence.merMod lme4  
## dfbetas.influence.merMod lme4

##   
## Attaching package: 'car'

## The following object is masked from 'package:arm':  
##   
## logit

vif(spend\_model2)

## GVIF Df GVIF^(1/(2\*Df))  
## log(visits) 1.682765 1 1.297214  
## feature 1.380449 1 1.174925  
## display 1.368067 1 1.169644  
## tpr\_only 1.084037 1 1.041171  
## price 3.288506 1 1.813424  
## year 8.875698 3 1.438910  
## weeknum 8.840835 1 2.973354  
## segment 1.001002 2 1.000250  
## category 3.868661 2 1.402459

vif(units\_model2)

## GVIF Df GVIF^(1/(2\*Df))  
## log(visits) 1.678309 1 1.295496  
## feature 1.379963 1 1.174718  
## display 1.368181 1 1.169693  
## tpr\_only 1.083945 1 1.041127  
## price 3.282864 1 1.811867  
## year 8.876861 3 1.438942  
## weeknum 8.841984 1 2.973547  
## segment 1.005622 2 1.001403  
## category 3.857445 2 1.401441

vif(hhs\_model2)

## GVIF Df GVIF^(1/(2\*Df))  
## log(visits) 1.683657 1 1.297558  
## feature 1.380537 1 1.174962  
## display 1.368049 1 1.169636  
## tpr\_only 1.084053 1 1.041179  
## price 3.289515 1 1.813702  
## year 8.875503 3 1.438905  
## weeknum 8.840641 1 2.973322  
## segment 1.000283 2 1.000071  
## category 3.870682 2 1.402642

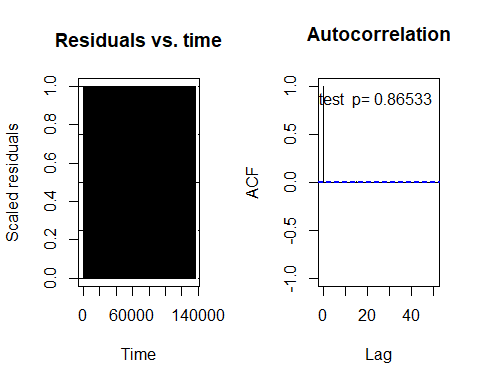
#Autocorrelation (Independence)  
#Durbin-Watson Test  
#library(lmtest)  
#dwtest(spend\_model2)  
#dwtest(units\_model2)  
#dwtest(hhs\_model2)  
  
library(DHARMa)

## Warning: package 'DHARMa' was built under R version 4.0.5

## This is DHARMa 0.4.1. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa') Note: Syntax of plotResiduals has changed in 0.3.0, see ?plotResiduals for details

testTemporalAutocorrelation(spend\_model2, time = NULL)

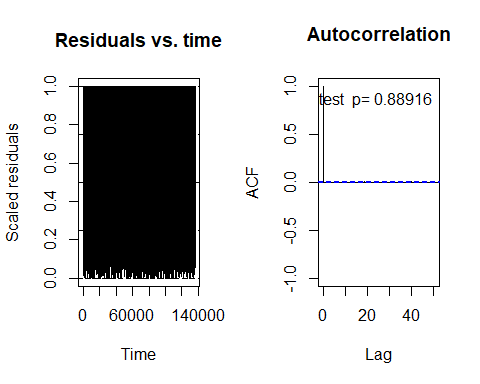
## DHARMa::testTemporalAutocorrelation - no time argument provided, using random times for each data point



##   
## Durbin-Watson test  
##   
## data: simulationOutput$scaledResiduals ~ 1  
## DW = 2.0009, p-value = 0.8653  
## alternative hypothesis: true autocorrelation is not 0

testTemporalAutocorrelation(units\_model2, time = NULL)

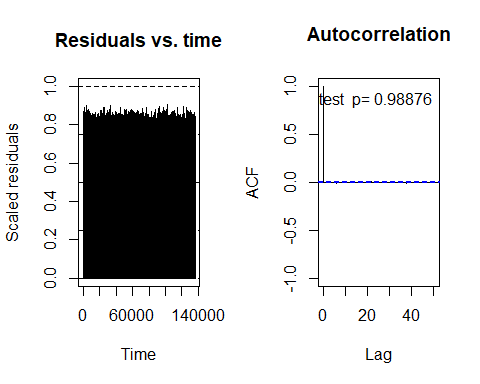
## DHARMa::testTemporalAutocorrelation - no time argument provided, using random times for each data point



##   
## Durbin-Watson test  
##   
## data: simulationOutput$scaledResiduals ~ 1  
## DW = 2.0008, p-value = 0.8892  
## alternative hypothesis: true autocorrelation is not 0

testTemporalAutocorrelation(hhs\_model2, time = NULL)

## DHARMa::testTemporalAutocorrelation - no time argument provided, using random times for each data point



##   
## Durbin-Watson test  
##   
## data: simulationOutput$scaledResiduals ~ 1  
## DW = 2.0001, p-value = 0.9888  
## alternative hypothesis: true autocorrelation is not 0