Prediction of Online Product Sales

Dong Kim

March 4, 2018

**Load the data from csv file.**

#Load the Data set   
setwd("C:/Users/Dongsung/Documents/")  
OnlineProductSale <- read.csv("OnlineProductSale.csv",sep=",",header = TRUE)  
summary(OnlineProductSale)

## Outcome\_M1 Outcome\_M2 Outcome\_M3 Outcome\_M4   
## Min. : 2000 Min. : 500 Min. : 500 Min. : 500   
## 1st Qu.: 2000 1st Qu.: 2000 1st Qu.: 500 1st Qu.: 500   
## Median : 5000 Median : 3000 Median : 2000 Median : 2000   
## Mean : 20621 Mean : 10838 Mean : 5194 Mean : 3495   
## 3rd Qu.: 16000 3rd Qu.: 8000 3rd Qu.: 5000 3rd Qu.: 3000   
## Max. :795000 Max. :315000 Max. :108000 Max. :74000   
## NA's :3 NA's :5 NA's :5   
## Outcome\_M5 Outcome\_M6 Outcome\_M7 Outcome\_M8   
## Min. : 500 Min. : 500 Min. : 500 Min. : 500   
## 1st Qu.: 500 1st Qu.: 500 1st Qu.: 500 1st Qu.: 500   
## Median : 2000 Median : 2000 Median : 2000 Median : 500   
## Mean : 2852 Mean : 2018 Mean : 1751 Mean : 1493   
## 3rd Qu.: 2000 3rd Qu.: 2000 3rd Qu.: 2000 3rd Qu.: 2000   
## Max. :81000 Max. :27000 Max. :21000 Max. :21000   
## NA's :24 NA's :46 NA's :57 NA's :64   
## Outcome\_M9 Outcome\_M10 Outcome\_M11 Outcome\_M12   
## Min. : 500 Min. : 500 Min. : 500 Min. : 500   
## 1st Qu.: 500 1st Qu.: 500 1st Qu.: 500 1st Qu.: 500   
## Median : 500 Median : 500 Median : 500 Median : 500   
## Mean : 1411 Mean : 1321 Mean : 1191 Mean : 1072   
## 3rd Qu.: 2000 3rd Qu.: 2000 3rd Qu.: 2000 3rd Qu.: 2000   
## Max. :23000 Max. :23000 Max. :10000 Max. :10000   
## NA's :74 NA's :95 NA's :106 NA's :112   
## Date\_1 Date\_2 Quan\_1 Quan\_2   
## Min. :1420 Min. : 365 Min. : 31600 Min. :1600   
## 1st Qu.:2254 1st Qu.:1584 1st Qu.: 922700 1st Qu.:6100   
## Median :2737 Median :2234 Median :1864300 Median :7200   
## Mean :2703 Mean :2177 Mean :2015039 Mean :6987   
## 3rd Qu.:3220 3rd Qu.:2716 3rd Qu.:2789600 3rd Qu.:8000   
## Max. :3682 Max. :3508 Max. :5268500 Max. :9700   
## NA's :3 NA's :12   
## Quan\_3 Quan\_4 Quan\_5 Quan\_6   
## Min. : 1.00 Min. : 2 Min. : 0.00 Min. : 0.000   
## 1st Qu.: 37.00 1st Qu.: 67368 1st Qu.: 4.00 1st Qu.: 1.000   
## Median : 60.00 Median : 245364 Median : 16.00 Median : 3.000   
## Mean : 62.17 Mean : 438943 Mean : 29.66 Mean : 8.603   
## 3rd Qu.: 87.00 3rd Qu.: 600178 3rd Qu.: 39.00 3rd Qu.:11.000   
## Max. :185.00 Max. :5417231 Max. :313.00 Max. :73.000   
## NA's :5 NA's :79 NA's :474 NA's :474   
## Quan\_7 Quan\_8 Quan\_9 Quan\_10   
## Min. : 0.00 Min. : 2.00 Min. : 0.00 Min. : 0.000   
## 1st Qu.: 0.00 1st Qu.: 10.00 1st Qu.: 2.00 1st Qu.: 1.000   
## Median : 4.00 Median : 25.00 Median : 7.00 Median : 2.000   
## Mean : 10.43 Mean : 41.39 Mean : 15.97 Mean : 6.025   
## 3rd Qu.: 11.00 3rd Qu.: 43.00 3rd Qu.: 19.00 3rd Qu.: 8.000   
## Max. :273.00 Max. :544.00 Max. :288.00 Max. :46.000   
## NA's :474 NA's :522 NA's :474 NA's :474   
## Quan\_11 Quan\_12 Quan\_13 Quan\_14   
## Min. : 0.00 Min. : 2 Min. : 3 Min. : 2.000   
## 1st Qu.: 0.00 1st Qu.: 728 1st Qu.: 493 1st Qu.: 4.000   
## Median : 1.00 Median : 4096 Median : 3072 Median : 8.000   
## Mean : 3.56 Mean : 497433 Mean : 512329 Mean : 9.447   
## 3rd Qu.: 2.00 3rd Qu.: 31667 3rd Qu.: 16486 3rd Qu.: 12.000   
## Max. :265.00 Max. :7340032 Max. :16777216 Max. :256.000   
## NA's :474 NA's :245 NA's :11 NA's :308   
## Quan\_15 Quan\_16 Cat\_1 Cat\_2   
## Min. : 2.00 Min. : 2.00 Min. :1.000 Min. :0.0000   
## 1st Qu.: 8.00 1st Qu.: 8.00 1st Qu.:1.000 1st Qu.:0.0000   
## Median : 24.00 Median : 26.00 Median :1.000 Median :0.0000   
## Mean : 52.81 Mean : 55.49 Mean :1.033 Mean :0.2117   
## 3rd Qu.: 43.25 3rd Qu.: 46.00 3rd Qu.:1.000 3rd Qu.:0.0000   
## Max. :1039.00 Max. :1039.00 Max. :2.000 Max. :1.0000   
## NA's :407 NA's :438   
## Cat\_3 Cat\_4 Cat\_5 Cat\_6   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:1.0000 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :1.0000 Median :1.0000 Median :0.0000 Median :0.0000   
## Mean :0.9933 Mean :0.8469 Mean :0.1704 Mean :0.3036   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
##   
## Cat\_7 Cat\_8 Cat\_9 Cat\_10 Cat\_11   
## Min. :1 Min. :1 Min. :0.0000 Min. :0 Min. :0.0000   
## 1st Qu.:1 1st Qu.:1 1st Qu.:0.0000 1st Qu.:0 1st Qu.:0.0000   
## Median :1 Median :1 Median :0.0000 Median :0 Median :0.0000   
## Mean :1 Mean :1 Mean :0.1212 Mean :0 Mean :0.3515   
## 3rd Qu.:1 3rd Qu.:1 3rd Qu.:0.0000 3rd Qu.:0 3rd Qu.:1.0000   
## Max. :1 Max. :1 Max. :1.0000 Max. :0 Max. :1.0000   
##   
## Cat\_12 Cat\_13 Cat\_14 Cat\_15   
## Min. :0.0000 Min. :1 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.:1 1st Qu.:0.0000 1st Qu.:0.00000   
## Median :0.0000 Median :1 Median :0.0000 Median :0.00000   
## Mean :0.3023 Mean :1 Mean :0.2783 Mean :0.09587   
## 3rd Qu.:1.0000 3rd Qu.:1 3rd Qu.:1.0000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :1 Max. :1.0000 Max. :1.00000   
##   
## Cat\_16 Cat\_17 Cat\_18 Cat\_19   
## Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:1.0000   
## Median :0.00000 Median :0.0000 Median :0.0000 Median :1.0000   
## Mean :0.01332 Mean :0.1704 Mean :0.2597 Mean :0.9228   
## 3rd Qu.:0.00000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :2.00000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
##   
## Cat\_20   
## Min. :0.0000   
## 1st Qu.:0.0000   
## Median :0.0000   
## Mean :0.4154   
## 3rd Qu.:1.0000   
## Max. :1.0000   
##

**Dataset:** 751 Observations and 50 variables

**List of Libraries:**

library(tidyr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

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

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

library(ggplot2)  
library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

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

library(ridge)  
library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':  
##   
## expand

## Loading required package: foreach

## Loaded glmnet 2.0-13

library(broom)  
library(tidyverse)

## -- Attaching packages -- tidyverse 1.2.1 --

## v tibble 1.4.2 v purrr 0.2.4  
## v readr 1.1.1 v stringr 1.2.0  
## v tibble 1.4.2 v forcats 0.3.0

## -- Conflicts ----- tidyverse\_conflicts() --  
## x purrr::accumulate() masks foreach::accumulate()  
## x plyr::arrange() masks dplyr::arrange()  
## x purrr::compact() masks plyr::compact()  
## x plyr::count() masks dplyr::count()  
## x Matrix::expand() masks tidyr::expand()  
## x plyr::failwith() masks dplyr::failwith()  
## x dplyr::filter() masks stats::filter()  
## x plyr::id() masks dplyr::id()  
## x dplyr::lag() masks stats::lag()  
## x plyr::mutate() masks dplyr::mutate()  
## x plyr::rename() masks dplyr::rename()  
## x plyr::summarise() masks dplyr::summarise()  
## x plyr::summarize() masks dplyr::summarize()  
## x purrr::when() masks foreach::when()

**CLEANING THE DATA**

**Adding a new column to the Data Set**

* Each row represents different consumer product but added new column called “Product\_Number” in the front of the data set.

OnlineProductSale["Product\_Number"] <- NA  
OnlineProductSale$Product\_Number <- 1:nrow(OnlineProductSale)  
OnlineProductSale <- OnlineProductSale[colnames(OnlineProductSale)[c(51,1:50)]]

**Variables with Missing Values**

* There are many missing values “Nan” throughout the data set in every columns.
* Replaced all the “Nan” with mean of each column

for(i in 1:ncol(OnlineProductSale))  
{  
 OnlineProductSale[is.na(OnlineProductSale[,i]), i] <- colMeans(OnlineProductSale[,i, drop = FALSE], na.rm = TRUE)  
 OnlineProductSale[]<-lapply(OnlineProductSale, as.integer)  
}

**Saving Cleaned Data Set**

write.csv(OnlineProductSale,"OnlineProductSale(Clean).csv",row.names = FALSE)

* Separately saved the cleaned dataset for better visualization of the data.

**Statistical Analysis**

* Calculated the minimum, median, maximum, and mean of outcome columns

dfnew1 <-OnlineProductSale[,c(2,3,4,5,6,7,8,9,10,11,12,13)]  
minimum\_outcome <- apply(dfnew1,2,min)  
minimum\_outcome

## Outcome\_M1 Outcome\_M2 Outcome\_M3 Outcome\_M4 Outcome\_M5 Outcome\_M6   
## 2000 500 500 500 500 500   
## Outcome\_M7 Outcome\_M8 Outcome\_M9 Outcome\_M10 Outcome\_M11 Outcome\_M12   
## 500 500 500 500 500 500

median\_outcome <- apply(dfnew1,2,median)  
median\_outcome

## Outcome\_M1 Outcome\_M2 Outcome\_M3 Outcome\_M4 Outcome\_M5 Outcome\_M6   
## 5000 3000 2000 2000 2000 2000   
## Outcome\_M7 Outcome\_M8 Outcome\_M9 Outcome\_M10 Outcome\_M11 Outcome\_M12   
## 1751 500 500 500 500 500

maximum\_outcome <- apply(dfnew1,2,max)  
maximum\_outcome

## Outcome\_M1 Outcome\_M2 Outcome\_M3 Outcome\_M4 Outcome\_M5 Outcome\_M6   
## 795000 315000 108000 74000 81000 27000   
## Outcome\_M7 Outcome\_M8 Outcome\_M9 Outcome\_M10 Outcome\_M11 Outcome\_M12   
## 21000 21000 23000 23000 10000 10000

mean\_outcome <- apply(dfnew1,2,mean)  
mean\_outcome

## Outcome\_M1 Outcome\_M2 Outcome\_M3 Outcome\_M4 Outcome\_M5 Outcome\_M6   
## 20620.506 10837.565 5194.368 3494.634 2852.128 2018.413   
## Outcome\_M7 Outcome\_M8 Outcome\_M9 Outcome\_M10 Outcome\_M11 Outcome\_M12   
## 1751.407 1492.660 1411.337 1320.772 1190.599 1071.840

* Minimum outcome of each sales months are represented. Outcome\_M1 has the highest mimimum sales outcome compared to other months.
* Outcome\_M1 has the products that were sold the most throughout the year.
* Outcome\_M1 has the highest average outcome.

**Statistical Analysis**

* What products are being sold below median throughout the year? Low Outcome Products

minim1 <- data.frame(OnlineProductSale$Product\_Number)  
LowOutcome\_Prod <-minim1[OnlineProductSale$Outcome\_M1<=5000 & OnlineProductSale$Outcome\_M2<=3000   
 & OnlineProductSale$Outcome\_M3<=2000 & OnlineProductSale$Outcome\_M4<=2000  
 & OnlineProductSale$Outcome\_M5<=2000 & OnlineProductSale$Outcome\_M6<=3000  
 & OnlineProductSale$Outcome\_M7<=1751 & OnlineProductSale$Outcome\_M8<=500   
 & OnlineProductSale$Outcome\_M9<=500 & OnlineProductSale$Outcome\_M10<=500   
 & OnlineProductSale$Outcome\_M11<=500 & OnlineProductSale$Outcome\_M12<=500,]  
LowOutcome\_Prod

## [1] 8 10 12 14 16 18 21 23 30 39 47 50 67 72 73 74 75  
## [18] 76 80 84 87 90 102 103 105 106 116 117 118 119 120 121 122 126  
## [35] 127 129 133 135 139 140 161 170 175 177 183 184 186 189 190 191 192  
## [52] 207 209 214 215 217 224 225 229 234 237 240 241 250 253 257 266 273  
## [69] 280 287 297 299 310 311 312 316 321 328 329 330 351 362 365 367 371  
## [86] 385 387 397 407 409 412 423 438 439 452 456 459 474 478 480 483 484  
## [103] 486 495 498 499 504 508 516 526 527 528 530 533 539 540 541 543 546  
## [120] 547 551 552 554 555 561 564 566 572 574 575 580 581 584 588 593 597  
## [137] 599 600 610 611 612 615 616 621 624 625 626 628 632 636 637 640 644  
## [154] 646 647 653 656 657 660 668 670 676 677 679 685 686 687 693 694 695  
## [171] 702 703 714 733 743

* Total of 175 products were sold below median
* What products are being sold above median throughout the year? BEST SELLERS: High Outcome Products

HighOutcome\_Prod <-minim1[OnlineProductSale$Outcome\_M1>=5000 & OnlineProductSale$Outcome\_M2>=3000   
 & OnlineProductSale$Outcome\_M3>=2000 & OnlineProductSale$Outcome\_M4>=2000  
 & OnlineProductSale$Outcome\_M5>=2000 & OnlineProductSale$Outcome\_M6>=3000  
 & OnlineProductSale$Outcome\_M7>=1751 & OnlineProductSale$Outcome\_M8>=500   
 & OnlineProductSale$Outcome\_M9>=500 & OnlineProductSale$Outcome\_M10>=500   
 & OnlineProductSale$Outcome\_M11>=500 & OnlineProductSale$Outcome\_M12>=500,]  
HighOutcome\_Prod

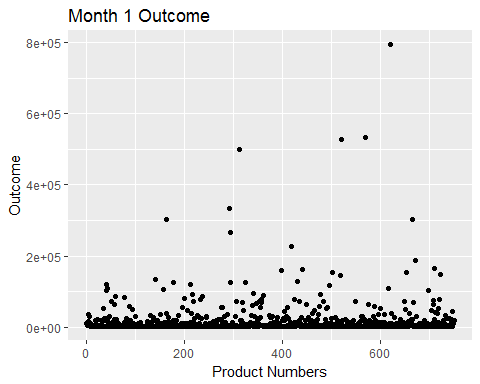
## [1] 4 32 36 41 43 44 57 58 59 83 85 89 91 94 101 109 131  
## [18] 137 142 146 157 164 165 176 179 182 197 206 216 218 220 222 231 238  
## [35] 252 260 276 294 295 298 307 313 320 322 324 325 340 341 344 348 353  
## [52] 355 356 357 358 361 391 395 404 411 417 419 440 442 460 462 465 477  
## [69] 479 485 493 496 503 513 518 520 538 542 545 549 562 563 573 578 587  
## [86] 591 601 603 609 619 620 630 641 643 648 650 654 655 659 661 664 666  
## [103] 673 689 690 698 705 708 709 710 711 719 722 729 734 735 739

* Total of 117 products were sold above median throughout the year.

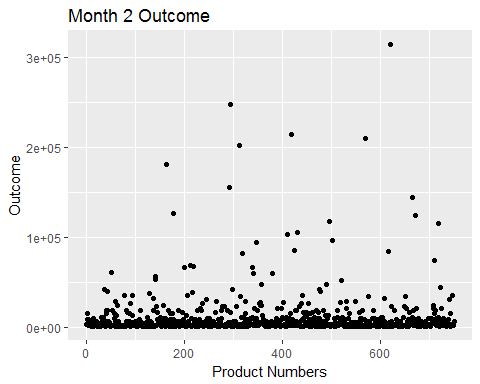
**Data Visualization**

* Graphs of each sales months

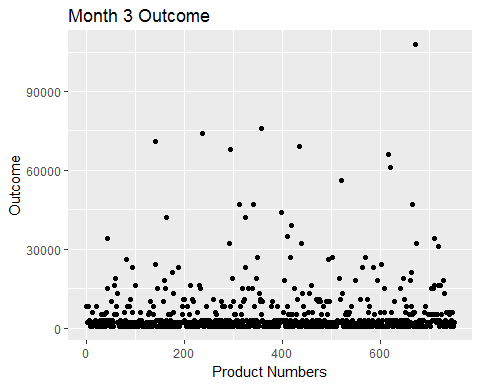
ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M1))+geom\_point()+  
 labs(title= "Month 1 Outcome",x = "Product Numbers", y = "Outcome")



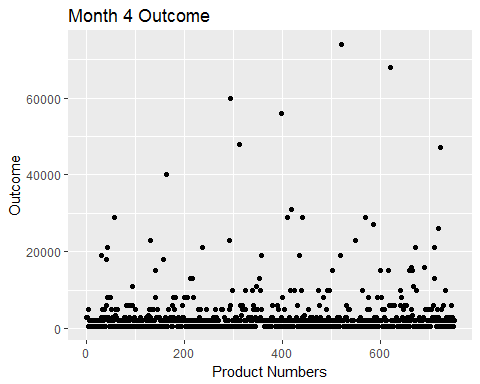
ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M2))+geom\_point()+  
 labs(title= "Month 2 Outcome",x = "Product Numbers", y = "Outcome")



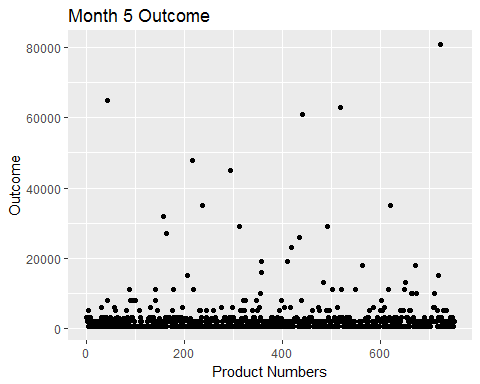
ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M3))+geom\_point()+  
 labs(title= "Month 3 Outcome",x = "Product Numbers", y = "Outcome")



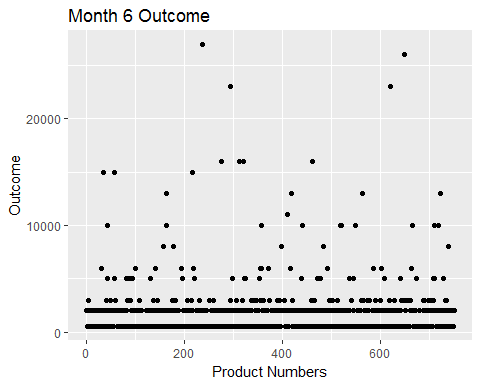
ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M4))+geom\_point()+  
 labs(title= "Month 4 Outcome",x = "Product Numbers", y = "Outcome")



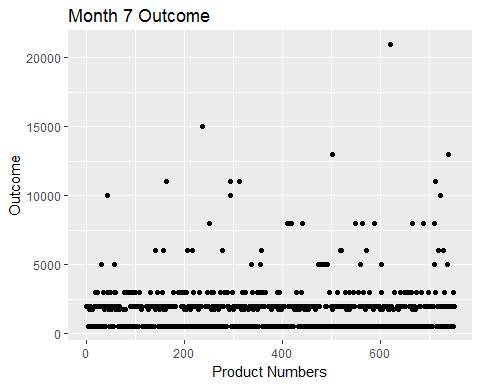
ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M5))+geom\_point()+  
 labs(title= "Month 5 Outcome",x = "Product Numbers", y = "Outcome")



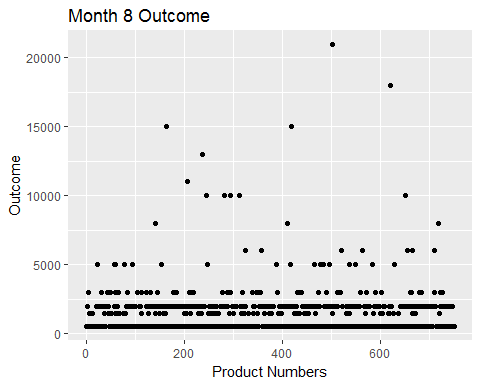
ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M6))+geom\_point()+  
 labs(title= "Month 6 Outcome",x = "Product Numbers", y = "Outcome")



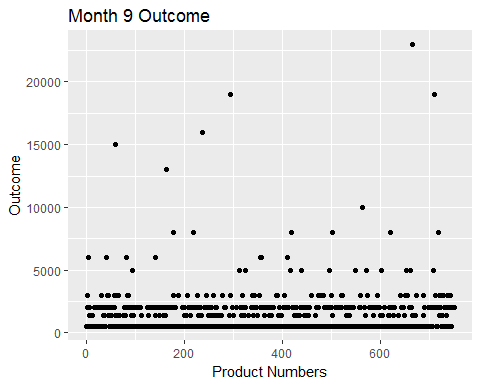
ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M7))+geom\_point()+  
 labs(title= "Month 7 Outcome",x = "Product Numbers", y = "Outcome")



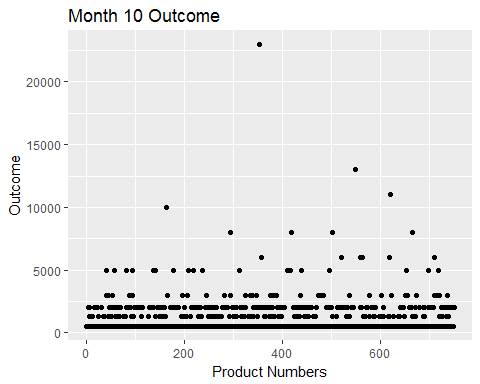
ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M8))+geom\_point()+  
 labs(title= "Month 8 Outcome",x = "Product Numbers", y = "Outcome")



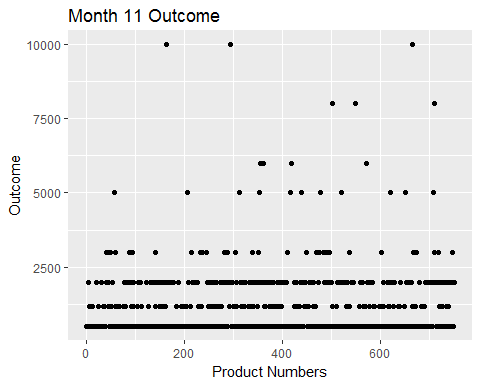
ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M9))+geom\_point()+  
 labs(title= "Month 9 Outcome",x = "Product Numbers", y = "Outcome")



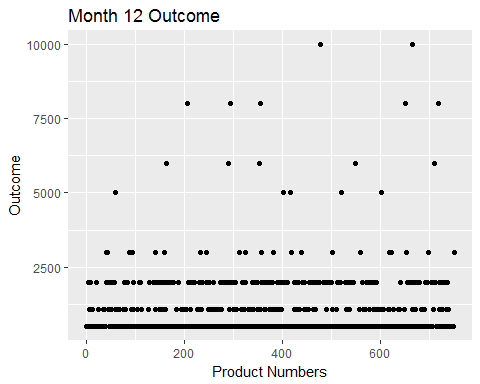
ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M10))+geom\_point()+  
 labs(title= "Month 10 Outcome",x = "Product Numbers", y = "Outcome")



ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M11))+geom\_point()+  
 labs(title= "Month 11 Outcome",x = "Product Numbers", y = "Outcome")



ggplot(OnlineProductSale,aes(x=Product\_Number, y=Outcome\_M12))+geom\_point()+  
 labs(title= "Month 12 Outcome",x = "Product Numbers", y = "Outcome")



* Compared to low and high values, it’s very challenging to find linear for these graphs with 751 products. Most of the plots were at very bottom of the graph. High selling products are highly scattered compared to low selling products. High selling products are more distinguishable compare to other products (outliers). Limitation for these graphs are there are too many products in the dataset which it is very challenging to distinguish between different products.

**Filtering Advertising Campaign**

* Calculated minimum, maximum, median, and mean of advertising campaign days.
* Date\_1 is the day number the major advertising campaign began and the product launched.
* Date\_2 is the day number the product was announced and a pre-release advertising campaign began.

dfnew2 <-OnlineProductSale[,c(14,15)]  
minimum\_adver <- apply(dfnew2,2,min)  
minimum\_adver

## Date\_1 Date\_2   
## 1420 365

max\_adver <- apply(dfnew2,2,max)  
max\_adver

## Date\_1 Date\_2   
## 3682 3508

median\_adver <-apply(dfnew2,2,median)  
median\_adver

## Date\_1 Date\_2   
## 2737 2233

mean\_adver <-apply(dfnew2,2,mean)  
mean\_adver

## Date\_1 Date\_2   
## 2702.860 2177.269

* Minimum: Date\_1: 1420 Date\_2:365
* Maximum: Date\_1: 3682 Date\_2:3508
* Median: Date\_1:2737 Date\_2:2233
* Mean: Date\_1:2702.9 Date\_2:2177.3
* Filtered what products have been majorly advertised and launched (Date\_1) the shortest based on median number.

minim1 <- data.frame(OnlineProductSale$Product\_Number)  
Least\_adver <-minim1[OnlineProductSale$Date\_1<2737,]   
Least\_adver

## [1] 1 2 3 4 12 14 15 18 19 20 21 22 23 24 26 28 29  
## [18] 32 34 36 38 41 42 44 45 47 48 53 54 55 56 58 64 69  
## [35] 70 71 75 79 85 86 88 89 91 93 94 95 98 99 101 102 107  
## [52] 108 109 110 112 115 117 123 124 125 126 129 132 133 135 136 137 138  
## [69] 142 146 147 150 151 157 160 161 163 165 166 167 169 170 172 173 174  
## [86] 176 178 180 181 182 183 187 189 190 191 192 193 194 198 199 203 210  
## [103] 211 212 215 218 221 222 224 226 228 234 235 237 241 242 245 246 247  
## [120] 248 253 257 259 260 264 265 266 267 268 273 277 278 280 282 284 285  
## [137] 286 288 290 291 294 295 297 299 300 302 303 304 305 307 308 310 314  
## [154] 315 321 324 325 327 329 330 331 333 336 337 339 340 343 345 346 347  
## [171] 350 351 352 353 354 356 357 358 360 362 364 367 370 371 372 373 376  
## [188] 378 381 382 386 387 390 391 393 394 395 404 405 408 409 410 412 415  
## [205] 416 417 418 419 421 422 423 424 426 427 429 430 432 434 435 437 439  
## [222] 442 443 446 447 448 451 452 457 458 459 460 461 465 469 471 472 473  
## [239] 476 478 480 482 484 486 488 489 490 493 495 497 498 500 504 505 506  
## [256] 507 508 509 512 515 518 519 521 523 524 527 528 529 530 534 538 539  
## [273] 543 544 545 547 549 550 552 553 555 556 559 563 564 570 571 573 575  
## [290] 577 578 582 583 584 588 590 591 596 599 600 602 603 608 609 610 612  
## [307] 613 615 619 622 628 629 630 631 632 633 636 640 641 643 648 649 652  
## [324] 654 655 658 660 661 664 666 667 668 669 673 674 675 679 680 681 685  
## [341] 686 692 694 695 696 697 698 700 701 702 704 705 706 707 708 711 712  
## [358] 714 715 716 723 730 732 735 738 740 741 742 743 745 746 748 749 750  
## [375] 751

* Total of 375 products were advertised less than the median number of days.

Most\_adver <-minim1[OnlineProductSale$Date\_1>2737,]   
Most\_adver

## [1] 5 6 7 8 9 10 11 13 16 17 25 27 30 31 33 35 37  
## [18] 39 40 43 46 49 50 51 52 57 59 60 61 62 63 65 66 67  
## [35] 68 72 73 74 76 77 78 80 81 82 83 84 87 90 92 96 97  
## [52] 100 103 104 105 106 111 113 114 116 118 119 120 121 122 127 128 130  
## [69] 131 134 139 140 141 143 145 148 149 152 153 154 155 156 158 159 162  
## [86] 164 168 171 175 177 179 184 185 186 188 195 196 197 200 201 202 204  
## [103] 205 206 207 208 209 213 214 216 217 219 220 223 225 227 229 230 231  
## [120] 232 233 236 238 239 240 243 244 249 250 251 252 254 255 256 258 261  
## [137] 262 263 269 270 271 272 274 275 276 279 281 283 287 289 292 293 296  
## [154] 298 301 306 309 311 312 313 316 317 318 319 320 322 323 326 328 332  
## [171] 334 335 338 341 342 344 348 349 355 359 361 363 365 366 368 369 374  
## [188] 375 377 379 380 383 384 385 388 389 392 396 397 398 399 400 401 402  
## [205] 403 406 407 411 413 414 420 425 428 431 433 436 438 440 441 444 445  
## [222] 449 450 453 454 455 456 462 463 464 466 467 468 470 474 475 477 479  
## [239] 481 483 485 487 491 492 494 496 499 501 502 503 510 511 513 514 516  
## [256] 517 520 522 525 526 531 532 533 535 536 537 540 541 542 546 548 551  
## [273] 554 557 558 560 561 562 565 566 567 568 569 572 574 579 580 581 585  
## [290] 586 587 589 592 593 594 595 597 598 601 604 605 606 607 611 614 616  
## [307] 617 618 620 621 623 624 625 626 627 634 635 637 638 639 642 644 645  
## [324] 646 647 650 651 653 656 657 659 662 663 665 670 671 672 676 677 678  
## [341] 682 683 684 687 688 689 690 691 693 699 703 709 710 717 718 719 720  
## [358] 721 722 724 725 726 727 728 729 731 733 734 736 737 739 744 747

* Total of 373 products were advertised more than the median number of days.
* About 1:1 ratio of products were advertised more than another products
* Filtered what products have been announced and pre-release advertised (Date\_2) the shortest.

minim1 <- data.frame(OnlineProductSale$Product\_Number)  
LeastPre\_adver <-minim1[OnlineProductSale$Date\_2<2233,]   
LeastPre\_adver

## [1] 1 2 3 4 5 9 10 12 14 15 16 18 20 21 22 23 24  
## [18] 27 28 29 32 34 36 38 41 42 44 45 47 48 53 54 55 56  
## [35] 58 63 64 69 70 71 75 78 79 81 85 86 88 89 90 91 93  
## [52] 94 97 98 99 101 107 108 109 111 112 117 122 123 124 125 129 132  
## [69] 133 135 137 138 142 145 146 147 150 151 157 161 163 165 166 167 169  
## [86] 171 172 173 176 178 180 181 182 183 184 189 190 192 193 194 198 199  
## [103] 201 203 204 210 212 215 218 222 226 228 232 234 235 236 237 242 244  
## [120] 245 246 247 248 250 257 259 265 266 267 268 273 274 277 278 279 280  
## [137] 281 282 284 285 286 288 289 290 291 294 295 297 299 300 304 305 307  
## [154] 308 310 314 315 324 325 326 327 329 330 331 333 334 335 336 337 338  
## [171] 339 340 343 344 345 346 347 348 350 351 352 353 354 356 357 358 360  
## [188] 362 363 366 370 371 372 373 376 379 386 387 388 390 391 393 395 403  
## [205] 404 405 408 409 410 412 413 415 417 418 419 421 422 423 424 426 427  
## [222] 429 430 432 435 437 439 442 443 444 446 448 451 452 456 460 461 465  
## [239] 467 470 471 472 473 476 477 478 482 486 488 490 493 495 497 500 503  
## [256] 504 505 507 508 509 512 513 515 516 518 519 521 523 524 526 528 530  
## [273] 534 538 539 540 543 544 545 547 548 549 550 553 555 556 559 563 565  
## [290] 568 570 571 573 575 577 578 580 584 588 590 591 596 599 600 601 603  
## [307] 608 611 612 615 617 619 621 623 629 630 631 633 640 641 643 648 649  
## [324] 652 654 655 657 658 660 666 667 668 669 672 673 674 675 676 678 679  
## [341] 680 681 685 692 696 697 698 701 702 704 705 706 707 708 711 714 715  
## [358] 716 720 721 723 729 735 737 738 740 741 742 743 745 746 748 749 750

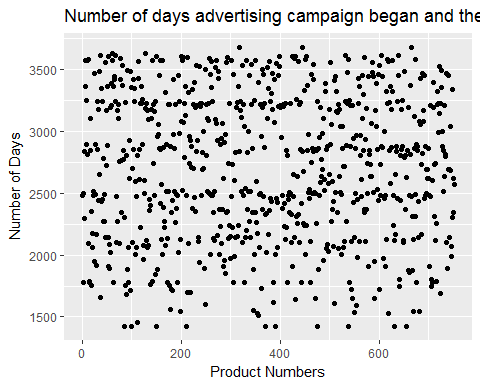
* Filtered what products have been announced and pre-release advertised (Date\_2) the shortest.

minim1 <- data.frame(OnlineProductSale$Product\_Number)  
MostPre\_adver <-minim1[OnlineProductSale$Date\_2>2233,]   
MostPre\_adver

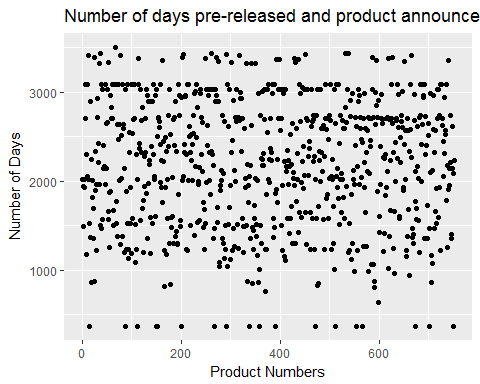
## [1] 6 7 8 11 13 17 19 25 26 30 31 33 35 37 39 40 43  
## [18] 46 49 50 51 52 57 59 60 61 62 65 66 67 68 72 73 74  
## [35] 76 77 80 82 83 84 87 92 95 96 100 102 103 104 105 106 110  
## [52] 113 114 115 116 118 119 120 121 126 127 128 130 131 134 136 139 140  
## [69] 141 143 144 148 149 152 153 154 155 156 158 159 160 162 164 168 170  
## [86] 174 175 177 179 185 186 187 188 191 195 196 197 200 202 205 206 207  
## [103] 208 209 211 213 214 216 217 219 220 221 223 224 225 227 229 230 231  
## [120] 233 238 239 240 241 243 249 251 252 253 254 255 256 258 260 261 262  
## [137] 263 264 269 270 271 272 275 276 283 287 292 293 296 298 301 302 303  
## [154] 306 309 311 312 313 316 317 318 319 320 321 322 323 328 332 341 342  
## [171] 349 355 359 361 364 365 367 368 369 374 375 377 378 380 381 382 384  
## [188] 385 389 392 394 396 397 398 399 400 401 402 406 407 411 414 416 420  
## [205] 425 428 431 433 434 436 438 440 441 445 447 449 450 453 454 455 457  
## [222] 458 459 462 463 464 466 468 469 474 475 480 481 483 484 485 487 489  
## [239] 491 492 494 496 498 499 501 502 506 510 511 514 517 520 522 525 527  
## [256] 529 531 532 533 535 536 537 541 542 546 551 552 554 557 558 560 561  
## [273] 562 564 566 567 569 572 574 576 579 581 582 583 585 586 587 589 592  
## [290] 593 594 595 597 598 602 604 605 606 607 609 610 613 614 616 618 620  
## [307] 624 625 626 627 628 632 634 635 636 637 638 639 642 644 645 646 647  
## [324] 650 651 653 656 659 661 662 663 664 665 670 671 677 682 683 684 686  
## [341] 687 688 689 690 691 693 694 695 699 700 703 709 710 712 713 717 718  
## [358] 719 722 724 725 726 727 728 730 731 732 733 734 736 739 744 747 751

* Scatterplot of High seller products advertising campaign

ggplot(OnlineProductSale,aes(x=Product\_Number, y=Date\_1))+geom\_point()+  
 labs(title= "Number of days advertising campaign began and the product launched",x = "Product Numbers", y = "Number of Days")



ggplot(OnlineProductSale,aes(x=Product\_Number, y=Date\_2))+geom\_point()+  
 labs(title= "Number of days pre-released and product announced",x = "Product Numbers", y = "Number of Days")



* Both pre-release advertising and post advertising not comparable. Number of days were very different from each products.
* From the figures above, it was found that products were advertised longer period of time when the products were launched than number of days which products were pre-released and products were announced.
* Overall, more advertisements before the products were launched had more effect on the sales outcome.

**Machine Learning**

* For OnlineProductSales, linear regression was used to predict how monthly outcome based on the products were related to the advertising campign.

filtered\_df <-OnlineProductSale[c(4,32,36,41,43,44,57,58,59,83,  
 85,89,91,94,101,109,131, 137, 142, 146,  
 157,164,165,176,179, 182, 197, 206, 216, 218,   
 220, 222,231, 238, 252, 260, 276, 294, 295, 298,   
 307, 313, 320, 322, 324, 325, 340, 341, 344, 348,   
 353, 355, 356, 357, 358, 361,391, 395, 404, 411,   
 417, 419, 440, 442, 460, 462, 465, 477, 479, 485,   
 493, 496, 503, 513, 518, 520, 538, 542, 545, 549,   
 562, 563, 573,578, 587, 591, 601, 603, 609, 619,   
 620, 630, 641, 643, 648, 650, 654, 655, 659, 661,   
 664, 666, 673, 689, 690, 698, 705, 708, 709, 710,  
 711, 719, 722, 729, 734, 735, 739),c(2:15)]  
filtered\_df

## Outcome\_M1 Outcome\_M2 Outcome\_M3 Outcome\_M4 Outcome\_M5 Outcome\_M6  
## 4 35000 8000 8000 5000 5000 3000  
## 32 19000 5000 5000 19000 6000 6000  
## 36 53000 11000 3000 2000 3000 15000  
## 41 103000 15000 6000 18000 2000 3000  
## 43 108000 40000 15000 21000 65000 10000  
## 44 16000 18000 34000 8000 8000 5000  
## 57 21000 15000 16000 29000 6000 5000  
## 58 65000 10000 5000 3000 6000 15000  
## 59 87000 29000 19000 5000 5000 3000  
## 83 24000 18000 26000 6000 5000 5000  
## 85 18000 16000 8000 6000 3000 3000  
## 89 58000 15000 8000 3000 11000 5000  
## 91 18000 26000 11000 6000 8000 3000  
## 94 11000 35000 23000 11000 8000 5000  
## 101 6000 6000 16000 5000 8000 6000  
## 109 10000 8000 3000 2000 5000 3000  
## 131 34000 6000 10000 23000 6000 5000  
## 137 26000 18000 5000 2000 2000 3000  
## 142 19000 56000 24000 15000 11000 6000  
## 146 18000 8000 15000 5000 5000 3000  
## 157 105000 24000 10000 18000 32000 8000  
## 164 302000 181000 42000 40000 27000 13000  
## 165 39000 6000 3000 2000 3000 10000  
## 176 16000 15000 21000 6000 5000 3000  
## 179 127000 126000 13000 8000 11000 8000  
## 182 8000 3000 2000 8000 3000 3000  
## 197 56000 16000 8000 6000 6000 5000  
## 206 48000 35000 8000 8000 15000 3000  
## 216 39000 10000 2000 2000 2000 3000  
## 218 92000 39000 11000 13000 48000 15000  
## 220 73000 68000 10000 8000 11000 6000  
## 222 24000 6000 3000 2000 2000 5000  
## 231 10000 8000 16000 3000 5000 3000  
## 238 87000 26000 74000 21000 35000 27000  
## 252 11000 10000 3000 2000 2000 3000  
## 260 16000 6000 11000 3000 5000 3000  
## 276 55000 18000 10000 5000 6000 16000  
## 294 126000 16000 8000 6000 3000 3000  
## 295 266000 248000 68000 60000 45000 23000  
## 298 15000 42000 19000 10000 6000 5000  
## 307 73000 23000 3000 3000 2000 3000  
## 313 500000 202000 47000 48000 29000 16000  
## 320 69000 82000 15000 6000 5000 3000  
## 322 48000 11000 10000 5000 8000 16000  
## 324 5000 5000 11000 5000 5000 5000  
## 325 127000 29000 42000 10000 8000 5000  
## 340 61000 66000 15000 10000 5000 3000  
## 341 94000 60000 47000 6000 3000 3000  
## 344 31000 21000 8000 5000 3000 3000  
## 348 66000 94000 19000 11000 8000 3000  
## 353 74000 26000 13000 13000 6000 5000  
## 355 55000 27000 10000 3000 3000 3000  
## 356 29000 24000 11000 10000 10000 6000  
## 357 69000 47000 76000 19000 16000 10000  
## 358 79000 24000 5000 5000 19000 6000  
## 361 90000 8000 2000 2000 2000 3000  
## 391 27000 21000 8000 6000 3000 3000  
## 395 6000 5000 3000 2000 5000 3000  
## 404 44000 10000 18000 5000 6000 3000  
## 411 56000 103000 35000 29000 19000 11000  
## 417 15000 15000 27000 10000 6000 6000  
## 419 227000 215000 39000 31000 23000 13000  
## 440 29000 26000 32000 10000 8000 5000  
## 442 163000 35000 13000 29000 61000 10000  
## 460 18000 6000 16000 3000 3000 3000  
## 462 39000 8000 6000 3000 5000 16000  
## 465 10000 11000 2000 2000 2000 3000  
## 477 60000 42000 11000 6000 5000 5000  
## 479 92000 40000 10000 3000 2000 5000  
## 485 73000 19000 8000 8000 13000 8000  
## 493 55000 10000 5000 10000 29000 6000  
## 496 119000 118000 10000 10000 5000 3000  
## 503 155000 97000 27000 15000 11000 3000  
## 513 19000 18000 2000 3000 6000 3000  
## 518 145000 27000 11000 19000 63000 10000  
## 520 527000 52000 56000 74000 11000 10000  
## 538 13000 15000 6000 5000 3000 5000  
## 542 10000 5000 8000 2000 2000 3000  
## 545 18000 5000 2000 2000 2000 5000  
## 549 74000 29000 18000 23000 11000 10000  
## 562 32000 10000 2000 2000 2000 3000  
## 563 32000 18000 23000 10000 18000 13000  
## 573 11000 5000 11000 5000 3000 3000  
## 578 19000 6000 8000 5000 5000 3000  
## 587 32000 19000 23000 27000 6000 6000  
## 591 18000 5000 2000 2000 2000 3000  
## 601 53000 6000 6000 15000 5000 3000  
## 603 13000 11000 24000 8000 8000 6000  
## 609 13000 6000 15000 3000 6000 5000  
## 619 11000 5000 3000 6000 3000 3000  
## 620 795000 315000 61000 68000 35000 23000  
## 630 29000 19000 10000 6000 3000 3000  
## 641 16000 6000 15000 6000 5000 5000  
## 643 8000 3000 3000 8000 3000 3000  
## 648 15000 11000 19000 5000 5000 3000  
## 650 74000 23000 11000 6000 11000 26000  
## 654 155000 21000 8000 6000 3000 3000  
## 655 6000 6000 3000 3000 3000 3000  
## 659 40000 16000 5000 15000 5000 3000  
## 661 18000 8000 18000 3000 5000 3000  
## 664 13000 6000 21000 5000 10000 6000  
## 666 302000 144000 47000 15000 10000 10000  
## 673 31000 21000 32000 10000 10000 5000  
## 689 11000 3000 3000 3000 2000 3000  
## 690 19000 5000 6000 16000 3000 3000  
## 698 103000 10000 5000 5000 2000 3000  
## 705 48000 10000 15000 5000 3000 3000  
## 708 65000 21000 5000 3000 2000 3000  
## 709 76000 24000 15000 6000 6000 5000  
## 710 21000 15000 16000 21000 6000 5000  
## 711 165000 74000 34000 13000 10000 10000  
## 719 52000 115000 31000 26000 15000 10000  
## 722 148000 44000 16000 47000 81000 13000  
## 729 19000 10000 18000 6000 5000 5000  
## 734 34000 5000 3000 10000 2000 3000  
## 735 5000 5000 6000 3000 5000 3000  
## 739 6000 8000 6000 3000 2000 8000  
## Outcome\_M7 Outcome\_M8 Outcome\_M9 Outcome\_M10 Outcome\_M11 Outcome\_M12  
## 4 2000 3000 6000 2000 2000 2000  
## 32 5000 3000 3000 2000 500 500  
## 36 3000 2000 2000 500 500 500  
## 41 2000 2000 2000 3000 3000 2000  
## 43 10000 3000 2000 500 2000 2000  
## 44 3000 2000 3000 3000 3000 3000  
## 57 5000 5000 3000 5000 5000 2000  
## 58 3000 2000 2000 500 500 500  
## 59 3000 3000 3000 2000 3000 5000  
## 83 3000 2000 6000 5000 2000 2000  
## 85 3000 3000 3000 2000 2000 2000  
## 89 3000 2000 2000 3000 3000 3000  
## 91 2000 500 500 2000 2000 500  
## 94 3000 5000 5000 5000 3000 3000  
## 101 3000 2000 2000 500 500 500  
## 109 3000 2000 2000 2000 2000 2000  
## 131 3000 3000 2000 2000 500 500  
## 137 2000 2000 2000 5000 500 500  
## 142 6000 8000 6000 5000 3000 3000  
## 146 2000 2000 2000 2000 2000 2000  
## 157 6000 2000 2000 500 500 500  
## 164 11000 15000 13000 10000 10000 6000  
## 165 2000 2000 2000 500 500 500  
## 176 2000 2000 2000 500 500 2000  
## 179 3000 3000 8000 5000 2000 2000  
## 182 2000 2000 2000 2000 500 500  
## 197 2000 2000 2000 3000 500 500  
## 206 6000 11000 3000 3000 5000 8000  
## 216 2000 2000 2000 2000 3000 2000  
## 218 6000 3000 2000 2000 2000 500  
## 220 3000 3000 8000 5000 2000 2000  
## 222 2000 2000 2000 2000 2000 2000  
## 231 2000 2000 2000 2000 500 500  
## 238 15000 13000 16000 5000 3000 2000  
## 252 8000 2000 2000 2000 2000 2000  
## 260 3000 2000 3000 2000 2000 2000  
## 276 2000 2000 1411 1320 1190 1071  
## 294 11000 3000 3000 3000 2000 2000  
## 295 10000 10000 19000 8000 10000 8000  
## 298 3000 2000 2000 2000 2000 2000  
## 307 2000 2000 2000 2000 2000 500  
## 313 11000 10000 5000 5000 5000 3000  
## 320 3000 3000 3000 2000 2000 2000  
## 322 3000 2000 500 500 500 500  
## 324 2000 2000 2000 500 500 500  
## 325 3000 6000 5000 3000 2000 3000  
## 340 3000 3000 3000 3000 3000 2000  
## 341 2000 2000 3000 2000 2000 2000  
## 344 2000 2000 3000 2000 2000 2000  
## 348 2000 3000 2000 3000 2000 2000  
## 353 3000 3000 3000 23000 5000 6000  
## 355 2000 2000 6000 2000 2000 1071  
## 356 5000 3000 2000 2000 6000 8000  
## 357 6000 6000 6000 6000 6000 3000  
## 358 3000 2000 2000 2000 2000 2000  
## 361 3000 2000 2000 2000 6000 2000  
## 391 3000 3000 3000 2000 2000 2000  
## 395 3000 2000 2000 2000 2000 2000  
## 404 2000 2000 3000 2000 2000 2000  
## 411 8000 8000 6000 5000 3000 2000  
## 417 8000 5000 5000 5000 5000 5000  
## 419 8000 15000 8000 8000 6000 3000  
## 440 3000 3000 5000 5000 5000 3000  
## 442 8000 2000 2000 2000 2000 2000  
## 460 2000 2000 3000 2000 2000 2000  
## 462 2000 2000 500 500 500 500  
## 465 2000 2000 500 500 500 500  
## 477 2000 2000 3000 3000 3000 2000  
## 479 5000 5000 3000 3000 5000 10000  
## 485 5000 5000 3000 3000 3000 2000  
## 493 5000 2000 2000 2000 2000 500  
## 496 3000 5000 5000 5000 3000 2000  
## 503 13000 21000 8000 8000 8000 3000  
## 513 3000 3000 2000 2000 2000 2000  
## 518 6000 3000 2000 2000 2000 2000  
## 520 6000 6000 3000 6000 5000 5000  
## 538 3000 5000 2000 3000 3000 2000  
## 542 2000 2000 2000 2000 2000 2000  
## 545 3000 2000 2000 2000 2000 2000  
## 549 8000 5000 5000 13000 8000 6000  
## 562 2000 500 500 2000 2000 500  
## 563 8000 6000 10000 6000 2000 2000  
## 573 2000 2000 3000 2000 2000 2000  
## 578 3000 2000 2000 3000 2000 2000  
## 587 1751 1492 1411 1320 1190 1071  
## 591 2000 2000 2000 500 500 2000  
## 601 1751 1492 1411 1320 1190 1071  
## 603 5000 3000 5000 3000 3000 5000  
## 609 2000 2000 2000 500 500 500  
## 619 2000 2000 2000 6000 2000 3000  
## 620 21000 18000 8000 11000 5000 3000  
## 630 3000 5000 2000 500 500 500  
## 641 3000 2000 3000 2000 2000 500  
## 643 2000 2000 2000 2000 500 500  
## 648 2000 2000 2000 2000 500 500  
## 650 3000 2000 1411 1320 1190 1071  
## 654 3000 2000 5000 5000 2000 3000  
## 655 3000 6000 3000 3000 2000 2000  
## 659 2000 2000 2000 2000 2000 2000  
## 661 3000 3000 5000 2000 2000 2000  
## 664 3000 3000 2000 500 500 500  
## 666 8000 6000 23000 8000 10000 10000  
## 673 3000 2000 3000 3000 3000 2000  
## 689 8000 2000 3000 2000 500 500  
## 690 2000 2000 2000 2000 500 500  
## 698 3000 2000 2000 5000 2000 3000  
## 705 2000 2000 2000 2000 2000 2000  
## 708 2000 2000 2000 2000 5000 2000  
## 709 2000 2000 5000 2000 2000 1071  
## 710 5000 3000 2000 3000 3000 2000  
## 711 8000 6000 19000 6000 8000 6000  
## 719 6000 8000 8000 5000 3000 8000  
## 722 10000 3000 2000 500 2000 500  
## 729 6000 2000 3000 2000 2000 2000  
## 734 2000 2000 2000 2000 2000 2000  
## 735 2000 2000 2000 2000 500 500  
## 739 13000 2000 2000 1320 1190 1071  
## Date\_1 Date\_2  
## 4 2296 2018  
## 32 2443 2215  
## 36 1659 1508  
## 41 2450 1569  
## 43 2782 2401  
## 44 2490 2162  
## 57 3192 2960  
## 58 2023 1885  
## 59 3206 2831  
## 83 2849 2639  
## 85 1421 1195  
## 89 1680 1231  
## 91 1883 1526  
## 94 1778 1231  
## 101 2100 1584  
## 109 2064 1231  
## 131 3171 2960  
## 137 2401 1190  
## 142 1784 1525  
## 146 2485 2025  
## 157 2415 1987  
## 164 2870 2450  
## 165 2387 2226  
## 176 1757 1301  
## 179 2877 2521  
## 182 1715 1603  
## 197 2982 2738  
## 206 2793 2417  
## 216 2961 2500  
## 218 1694 1508  
## 220 2877 2521  
## 222 2380 2004  
## 231 2835 2488  
## 238 3199 2988  
## 252 3101 2628  
## 260 2478 2337  
## 276 3481 3348  
## 294 2354 1044  
## 295 2134 1511  
## 298 3229 3026  
## 307 2604 2123  
## 313 3234 2689  
## 320 3241 2934  
## 322 3115 2926  
## 324 2100 1584  
## 325 2485 1197  
## 340 2142 1354  
## 341 3220 2626  
## 344 2863 857  
## 348 3249 1728  
## 353 1528 1171  
## 355 3395 2985  
## 356 1511 860  
## 357 2492 365  
## 358 2058 1010  
## 361 2961 2261  
## 391 1421 365  
## 395 2058 1757  
## 404 2478 1955  
## 411 3229 3026  
## 417 2485 1961  
## 419 2506 2162  
## 440 2856 2676  
## 442 2415 1987  
## 460 2107 1654  
## 462 2751 2610  
## 465 2499 1583  
## 477 2863 857  
## 479 2947 2233  
## 485 2982 2590  
## 493 2051 1802  
## 496 2877 2708  
## 503 3059 1231  
## 513 3157 2057  
## 518 2051 1802  
## 520 3178 2429  
## 538 1421 1197  
## 542 3213 2908  
## 545 1646 1301  
## 549 1539 983  
## 562 2968 2598  
## 563 2471 2039  
## 573 2107 1654  
## 578 1904 983  
## 587 3556 3388  
## 591 1652 802  
## 601 3535 1605  
## 603 2485 1961  
## 609 2471 2264  
## 619 1526 1016  
## 620 3234 2689  
## 630 1421 959  
## 641 1745 1584  
## 643 2443 2215  
## 648 2121 1900  
## 650 3481 3348  
## 654 2628 1158  
## 655 1421 1227  
## 659 3192 2569  
## 661 2478 2337  
## 664 2471 2264  
## 666 2310 1470  
## 673 2492 365  
## 689 3080 2942  
## 690 3171 2960  
## 698 2618 1730  
## 705 1750 1171  
## 708 2604 2123  
## 709 3395 2582  
## 710 3192 2960  
## 711 2310 1470  
## 719 3229 3026  
## 722 3143 2738  
## 729 2849 2052  
## 734 2800 2528  
## 735 2100 1781  
## 739 3451 3361

* Filtered (high sellers) products were made from Outcome\_M1 through Date\_2.
* There are 117 products that are filtered from 12 months period and 2 advertising days.

filtered\_df\_train <- filtered\_df[c(1:75),c(1:14)]  
filtered\_df\_train

## Outcome\_M1 Outcome\_M2 Outcome\_M3 Outcome\_M4 Outcome\_M5 Outcome\_M6  
## 4 35000 8000 8000 5000 5000 3000  
## 32 19000 5000 5000 19000 6000 6000  
## 36 53000 11000 3000 2000 3000 15000  
## 41 103000 15000 6000 18000 2000 3000  
## 43 108000 40000 15000 21000 65000 10000  
## 44 16000 18000 34000 8000 8000 5000  
## 57 21000 15000 16000 29000 6000 5000  
## 58 65000 10000 5000 3000 6000 15000  
## 59 87000 29000 19000 5000 5000 3000  
## 83 24000 18000 26000 6000 5000 5000  
## 85 18000 16000 8000 6000 3000 3000  
## 89 58000 15000 8000 3000 11000 5000  
## 91 18000 26000 11000 6000 8000 3000  
## 94 11000 35000 23000 11000 8000 5000  
## 101 6000 6000 16000 5000 8000 6000  
## 109 10000 8000 3000 2000 5000 3000  
## 131 34000 6000 10000 23000 6000 5000  
## 137 26000 18000 5000 2000 2000 3000  
## 142 19000 56000 24000 15000 11000 6000  
## 146 18000 8000 15000 5000 5000 3000  
## 157 105000 24000 10000 18000 32000 8000  
## 164 302000 181000 42000 40000 27000 13000  
## 165 39000 6000 3000 2000 3000 10000  
## 176 16000 15000 21000 6000 5000 3000  
## 179 127000 126000 13000 8000 11000 8000  
## 182 8000 3000 2000 8000 3000 3000  
## 197 56000 16000 8000 6000 6000 5000  
## 206 48000 35000 8000 8000 15000 3000  
## 216 39000 10000 2000 2000 2000 3000  
## 218 92000 39000 11000 13000 48000 15000  
## 220 73000 68000 10000 8000 11000 6000  
## 222 24000 6000 3000 2000 2000 5000  
## 231 10000 8000 16000 3000 5000 3000  
## 238 87000 26000 74000 21000 35000 27000  
## 252 11000 10000 3000 2000 2000 3000  
## 260 16000 6000 11000 3000 5000 3000  
## 276 55000 18000 10000 5000 6000 16000  
## 294 126000 16000 8000 6000 3000 3000  
## 295 266000 248000 68000 60000 45000 23000  
## 298 15000 42000 19000 10000 6000 5000  
## 307 73000 23000 3000 3000 2000 3000  
## 313 500000 202000 47000 48000 29000 16000  
## 320 69000 82000 15000 6000 5000 3000  
## 322 48000 11000 10000 5000 8000 16000  
## 324 5000 5000 11000 5000 5000 5000  
## 325 127000 29000 42000 10000 8000 5000  
## 340 61000 66000 15000 10000 5000 3000  
## 341 94000 60000 47000 6000 3000 3000  
## 344 31000 21000 8000 5000 3000 3000  
## 348 66000 94000 19000 11000 8000 3000  
## 353 74000 26000 13000 13000 6000 5000  
## 355 55000 27000 10000 3000 3000 3000  
## 356 29000 24000 11000 10000 10000 6000  
## 357 69000 47000 76000 19000 16000 10000  
## 358 79000 24000 5000 5000 19000 6000  
## 361 90000 8000 2000 2000 2000 3000  
## 391 27000 21000 8000 6000 3000 3000  
## 395 6000 5000 3000 2000 5000 3000  
## 404 44000 10000 18000 5000 6000 3000  
## 411 56000 103000 35000 29000 19000 11000  
## 417 15000 15000 27000 10000 6000 6000  
## 419 227000 215000 39000 31000 23000 13000  
## 440 29000 26000 32000 10000 8000 5000  
## 442 163000 35000 13000 29000 61000 10000  
## 460 18000 6000 16000 3000 3000 3000  
## 462 39000 8000 6000 3000 5000 16000  
## 465 10000 11000 2000 2000 2000 3000  
## 477 60000 42000 11000 6000 5000 5000  
## 479 92000 40000 10000 3000 2000 5000  
## 485 73000 19000 8000 8000 13000 8000  
## 493 55000 10000 5000 10000 29000 6000  
## 496 119000 118000 10000 10000 5000 3000  
## 503 155000 97000 27000 15000 11000 3000  
## 513 19000 18000 2000 3000 6000 3000  
## 518 145000 27000 11000 19000 63000 10000  
## Outcome\_M7 Outcome\_M8 Outcome\_M9 Outcome\_M10 Outcome\_M11 Outcome\_M12  
## 4 2000 3000 6000 2000 2000 2000  
## 32 5000 3000 3000 2000 500 500  
## 36 3000 2000 2000 500 500 500  
## 41 2000 2000 2000 3000 3000 2000  
## 43 10000 3000 2000 500 2000 2000  
## 44 3000 2000 3000 3000 3000 3000  
## 57 5000 5000 3000 5000 5000 2000  
## 58 3000 2000 2000 500 500 500  
## 59 3000 3000 3000 2000 3000 5000  
## 83 3000 2000 6000 5000 2000 2000  
## 85 3000 3000 3000 2000 2000 2000  
## 89 3000 2000 2000 3000 3000 3000  
## 91 2000 500 500 2000 2000 500  
## 94 3000 5000 5000 5000 3000 3000  
## 101 3000 2000 2000 500 500 500  
## 109 3000 2000 2000 2000 2000 2000  
## 131 3000 3000 2000 2000 500 500  
## 137 2000 2000 2000 5000 500 500  
## 142 6000 8000 6000 5000 3000 3000  
## 146 2000 2000 2000 2000 2000 2000  
## 157 6000 2000 2000 500 500 500  
## 164 11000 15000 13000 10000 10000 6000  
## 165 2000 2000 2000 500 500 500  
## 176 2000 2000 2000 500 500 2000  
## 179 3000 3000 8000 5000 2000 2000  
## 182 2000 2000 2000 2000 500 500  
## 197 2000 2000 2000 3000 500 500  
## 206 6000 11000 3000 3000 5000 8000  
## 216 2000 2000 2000 2000 3000 2000  
## 218 6000 3000 2000 2000 2000 500  
## 220 3000 3000 8000 5000 2000 2000  
## 222 2000 2000 2000 2000 2000 2000  
## 231 2000 2000 2000 2000 500 500  
## 238 15000 13000 16000 5000 3000 2000  
## 252 8000 2000 2000 2000 2000 2000  
## 260 3000 2000 3000 2000 2000 2000  
## 276 2000 2000 1411 1320 1190 1071  
## 294 11000 3000 3000 3000 2000 2000  
## 295 10000 10000 19000 8000 10000 8000  
## 298 3000 2000 2000 2000 2000 2000  
## 307 2000 2000 2000 2000 2000 500  
## 313 11000 10000 5000 5000 5000 3000  
## 320 3000 3000 3000 2000 2000 2000  
## 322 3000 2000 500 500 500 500  
## 324 2000 2000 2000 500 500 500  
## 325 3000 6000 5000 3000 2000 3000  
## 340 3000 3000 3000 3000 3000 2000  
## 341 2000 2000 3000 2000 2000 2000  
## 344 2000 2000 3000 2000 2000 2000  
## 348 2000 3000 2000 3000 2000 2000  
## 353 3000 3000 3000 23000 5000 6000  
## 355 2000 2000 6000 2000 2000 1071  
## 356 5000 3000 2000 2000 6000 8000  
## 357 6000 6000 6000 6000 6000 3000  
## 358 3000 2000 2000 2000 2000 2000  
## 361 3000 2000 2000 2000 6000 2000  
## 391 3000 3000 3000 2000 2000 2000  
## 395 3000 2000 2000 2000 2000 2000  
## 404 2000 2000 3000 2000 2000 2000  
## 411 8000 8000 6000 5000 3000 2000  
## 417 8000 5000 5000 5000 5000 5000  
## 419 8000 15000 8000 8000 6000 3000  
## 440 3000 3000 5000 5000 5000 3000  
## 442 8000 2000 2000 2000 2000 2000  
## 460 2000 2000 3000 2000 2000 2000  
## 462 2000 2000 500 500 500 500  
## 465 2000 2000 500 500 500 500  
## 477 2000 2000 3000 3000 3000 2000  
## 479 5000 5000 3000 3000 5000 10000  
## 485 5000 5000 3000 3000 3000 2000  
## 493 5000 2000 2000 2000 2000 500  
## 496 3000 5000 5000 5000 3000 2000  
## 503 13000 21000 8000 8000 8000 3000  
## 513 3000 3000 2000 2000 2000 2000  
## 518 6000 3000 2000 2000 2000 2000  
## Date\_1 Date\_2  
## 4 2296 2018  
## 32 2443 2215  
## 36 1659 1508  
## 41 2450 1569  
## 43 2782 2401  
## 44 2490 2162  
## 57 3192 2960  
## 58 2023 1885  
## 59 3206 2831  
## 83 2849 2639  
## 85 1421 1195  
## 89 1680 1231  
## 91 1883 1526  
## 94 1778 1231  
## 101 2100 1584  
## 109 2064 1231  
## 131 3171 2960  
## 137 2401 1190  
## 142 1784 1525  
## 146 2485 2025  
## 157 2415 1987  
## 164 2870 2450  
## 165 2387 2226  
## 176 1757 1301  
## 179 2877 2521  
## 182 1715 1603  
## 197 2982 2738  
## 206 2793 2417  
## 216 2961 2500  
## 218 1694 1508  
## 220 2877 2521  
## 222 2380 2004  
## 231 2835 2488  
## 238 3199 2988  
## 252 3101 2628  
## 260 2478 2337  
## 276 3481 3348  
## 294 2354 1044  
## 295 2134 1511  
## 298 3229 3026  
## 307 2604 2123  
## 313 3234 2689  
## 320 3241 2934  
## 322 3115 2926  
## 324 2100 1584  
## 325 2485 1197  
## 340 2142 1354  
## 341 3220 2626  
## 344 2863 857  
## 348 3249 1728  
## 353 1528 1171  
## 355 3395 2985  
## 356 1511 860  
## 357 2492 365  
## 358 2058 1010  
## 361 2961 2261  
## 391 1421 365  
## 395 2058 1757  
## 404 2478 1955  
## 411 3229 3026  
## 417 2485 1961  
## 419 2506 2162  
## 440 2856 2676  
## 442 2415 1987  
## 460 2107 1654  
## 462 2751 2610  
## 465 2499 1583  
## 477 2863 857  
## 479 2947 2233  
## 485 2982 2590  
## 493 2051 1802  
## 496 2877 2708  
## 503 3059 1231  
## 513 3157 2057  
## 518 2051 1802

filtered\_df\_train['Outcome\_M1'] = log10(filtered\_df\_train['Outcome\_M1'])  
filtered\_df\_train['Outcome\_M2'] = log10(filtered\_df\_train['Outcome\_M2'])  
filtered\_df\_train['Outcome\_M3'] = log10(filtered\_df\_train['Outcome\_M3'])  
filtered\_df\_train['Outcome\_M4'] = log10(filtered\_df\_train['Outcome\_M4'])  
filtered\_df\_train['Outcome\_M5'] = log10(filtered\_df\_train['Outcome\_M5'])  
filtered\_df\_train['Outcome\_M6'] = log10(filtered\_df\_train['Outcome\_M6'])  
filtered\_df\_train['Outcome\_M7'] = log10(filtered\_df\_train['Outcome\_M7'])  
filtered\_df\_train['Outcome\_M8'] = log10(filtered\_df\_train['Outcome\_M8'])  
filtered\_df\_train['Outcome\_M9'] = log10(filtered\_df\_train['Outcome\_M9'])  
filtered\_df\_train['Outcome\_M10'] = log10(filtered\_df\_train['Outcome\_M10'])  
filtered\_df\_train['Outcome\_M11'] = log10(filtered\_df\_train['Outcome\_M11'])  
filtered\_df\_train['Outcome\_M12'] = log10(filtered\_df\_train['Outcome\_M12'])  
  
head(filtered\_df\_train)

## Outcome\_M1 Outcome\_M2 Outcome\_M3 Outcome\_M4 Outcome\_M5 Outcome\_M6  
## 4 4.544068 3.903090 3.903090 3.698970 3.698970 3.477121  
## 32 4.278754 3.698970 3.698970 4.278754 3.778151 3.778151  
## 36 4.724276 4.041393 3.477121 3.301030 3.477121 4.176091  
## 41 5.012837 4.176091 3.778151 4.255273 3.301030 3.477121  
## 43 5.033424 4.602060 4.176091 4.322219 4.812913 4.000000  
## 44 4.204120 4.255273 4.531479 3.903090 3.903090 3.698970  
## Outcome\_M7 Outcome\_M8 Outcome\_M9 Outcome\_M10 Outcome\_M11 Outcome\_M12  
## 4 3.301030 3.477121 3.778151 3.301030 3.301030 3.301030  
## 32 3.698970 3.477121 3.477121 3.301030 2.698970 2.698970  
## 36 3.477121 3.301030 3.301030 2.698970 2.698970 2.698970  
## 41 3.301030 3.301030 3.301030 3.477121 3.477121 3.301030  
## 43 4.000000 3.477121 3.301030 2.698970 3.301030 3.301030  
## 44 3.477121 3.301030 3.477121 3.477121 3.477121 3.477121  
## Date\_1 Date\_2  
## 4 2296 2018  
## 32 2443 2215  
## 36 1659 1508  
## 41 2450 1569  
## 43 2782 2401  
## 44 2490 2162

filtered\_df\_test <- filtered\_df[c(76:117),c(1:12)]  
filtered\_df\_test

## Outcome\_M1 Outcome\_M2 Outcome\_M3 Outcome\_M4 Outcome\_M5 Outcome\_M6  
## 520 527000 52000 56000 74000 11000 10000  
## 538 13000 15000 6000 5000 3000 5000  
## 542 10000 5000 8000 2000 2000 3000  
## 545 18000 5000 2000 2000 2000 5000  
## 549 74000 29000 18000 23000 11000 10000  
## 562 32000 10000 2000 2000 2000 3000  
## 563 32000 18000 23000 10000 18000 13000  
## 573 11000 5000 11000 5000 3000 3000  
## 578 19000 6000 8000 5000 5000 3000  
## 587 32000 19000 23000 27000 6000 6000  
## 591 18000 5000 2000 2000 2000 3000  
## 601 53000 6000 6000 15000 5000 3000  
## 603 13000 11000 24000 8000 8000 6000  
## 609 13000 6000 15000 3000 6000 5000  
## 619 11000 5000 3000 6000 3000 3000  
## 620 795000 315000 61000 68000 35000 23000  
## 630 29000 19000 10000 6000 3000 3000  
## 641 16000 6000 15000 6000 5000 5000  
## 643 8000 3000 3000 8000 3000 3000  
## 648 15000 11000 19000 5000 5000 3000  
## 650 74000 23000 11000 6000 11000 26000  
## 654 155000 21000 8000 6000 3000 3000  
## 655 6000 6000 3000 3000 3000 3000  
## 659 40000 16000 5000 15000 5000 3000  
## 661 18000 8000 18000 3000 5000 3000  
## 664 13000 6000 21000 5000 10000 6000  
## 666 302000 144000 47000 15000 10000 10000  
## 673 31000 21000 32000 10000 10000 5000  
## 689 11000 3000 3000 3000 2000 3000  
## 690 19000 5000 6000 16000 3000 3000  
## 698 103000 10000 5000 5000 2000 3000  
## 705 48000 10000 15000 5000 3000 3000  
## 708 65000 21000 5000 3000 2000 3000  
## 709 76000 24000 15000 6000 6000 5000  
## 710 21000 15000 16000 21000 6000 5000  
## 711 165000 74000 34000 13000 10000 10000  
## 719 52000 115000 31000 26000 15000 10000  
## 722 148000 44000 16000 47000 81000 13000  
## 729 19000 10000 18000 6000 5000 5000  
## 734 34000 5000 3000 10000 2000 3000  
## 735 5000 5000 6000 3000 5000 3000  
## 739 6000 8000 6000 3000 2000 8000  
## Outcome\_M7 Outcome\_M8 Outcome\_M9 Outcome\_M10 Outcome\_M11 Outcome\_M12  
## 520 6000 6000 3000 6000 5000 5000  
## 538 3000 5000 2000 3000 3000 2000  
## 542 2000 2000 2000 2000 2000 2000  
## 545 3000 2000 2000 2000 2000 2000  
## 549 8000 5000 5000 13000 8000 6000  
## 562 2000 500 500 2000 2000 500  
## 563 8000 6000 10000 6000 2000 2000  
## 573 2000 2000 3000 2000 2000 2000  
## 578 3000 2000 2000 3000 2000 2000  
## 587 1751 1492 1411 1320 1190 1071  
## 591 2000 2000 2000 500 500 2000  
## 601 1751 1492 1411 1320 1190 1071  
## 603 5000 3000 5000 3000 3000 5000  
## 609 2000 2000 2000 500 500 500  
## 619 2000 2000 2000 6000 2000 3000  
## 620 21000 18000 8000 11000 5000 3000  
## 630 3000 5000 2000 500 500 500  
## 641 3000 2000 3000 2000 2000 500  
## 643 2000 2000 2000 2000 500 500  
## 648 2000 2000 2000 2000 500 500  
## 650 3000 2000 1411 1320 1190 1071  
## 654 3000 2000 5000 5000 2000 3000  
## 655 3000 6000 3000 3000 2000 2000  
## 659 2000 2000 2000 2000 2000 2000  
## 661 3000 3000 5000 2000 2000 2000  
## 664 3000 3000 2000 500 500 500  
## 666 8000 6000 23000 8000 10000 10000  
## 673 3000 2000 3000 3000 3000 2000  
## 689 8000 2000 3000 2000 500 500  
## 690 2000 2000 2000 2000 500 500  
## 698 3000 2000 2000 5000 2000 3000  
## 705 2000 2000 2000 2000 2000 2000  
## 708 2000 2000 2000 2000 5000 2000  
## 709 2000 2000 5000 2000 2000 1071  
## 710 5000 3000 2000 3000 3000 2000  
## 711 8000 6000 19000 6000 8000 6000  
## 719 6000 8000 8000 5000 3000 8000  
## 722 10000 3000 2000 500 2000 500  
## 729 6000 2000 3000 2000 2000 2000  
## 734 2000 2000 2000 2000 2000 2000  
## 735 2000 2000 2000 2000 500 500  
## 739 13000 2000 2000 1320 1190 1071

* Training and Test dataset were made to find suitable linear regression for Outcomes followed by two different advertising had effects.
* Test dataset does not contain advertising variables which only predict the outcomes.
* Log10 was used to see how log10 could help reducing the p-values which it can find the significance of the predicting variables.

model1 = lm(Date\_1 ~ Outcome\_M1 + Outcome\_M2 + Outcome\_M3 + Outcome\_M4 + Outcome\_M5 + Outcome\_M6 + Outcome\_M7 +   
 Outcome\_M8 + Outcome\_M9 + Outcome\_M10 + Outcome\_M11 + Outcome\_M12,data=filtered\_df\_train)  
summary(model1)

##   
## Call:  
## lm(formula = Date\_1 ~ Outcome\_M1 + Outcome\_M2 + Outcome\_M3 +   
## Outcome\_M4 + Outcome\_M5 + Outcome\_M6 + Outcome\_M7 + Outcome\_M8 +   
## Outcome\_M9 + Outcome\_M10 + Outcome\_M11 + Outcome\_M12, data = filtered\_df\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1144.76 -318.29 27.98 384.08 933.21   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 1206.7 1180.7 1.022 0.311  
## Outcome\_M1 371.5 225.5 1.647 0.105  
## Outcome\_M2 190.6 259.5 0.734 0.465  
## Outcome\_M3 411.1 260.9 1.575 0.120  
## Outcome\_M4 -227.9 319.8 -0.713 0.479  
## Outcome\_M5 -432.7 287.1 -1.507 0.137  
## Outcome\_M6 -147.9 347.1 -0.426 0.672  
## Outcome\_M7 144.8 460.1 0.315 0.754  
## Outcome\_M8 449.8 445.7 1.009 0.317  
## Outcome\_M9 -208.2 363.3 -0.573 0.569  
## Outcome\_M10 -76.1 322.1 -0.236 0.814  
## Outcome\_M11 152.8 410.9 0.372 0.711  
## Outcome\_M12 -434.0 355.7 -1.220 0.227  
##   
## Residual standard error: 531.6 on 62 degrees of freedom  
## Multiple R-squared: 0.1955, Adjusted R-squared: 0.03978   
## F-statistic: 1.255 on 12 and 62 DF, p-value: 0.2678

* Outcomes and Date\_1 (number of days which products were pre-released and products launched) linear model was created to find the corelations.
* Model 1 has high P-value of 0.2678, which it’s higher than 0.05. High p-value result in unable to find the significant differences.
* Using log10 method for training test test have improved the p-value from 0.4641 to 0.2678 but still significance cannot be found due to higher p-value than 0.05.

filtered\_df\_new <- filtered\_df[c(76:117),c(1:14)]  
filtered\_df\_new <- log10(filtered\_df\_new)

* New filtered\_df\_new was created to include Date\_1 and Date\_2 variables for linear models

predictTest\_date1 = predict(model1, newdata =filtered\_df\_test)  
predictTest\_date1

## 520 538 542 545 549 562 563   
## 206710502 8608827 6251607 6606411 28937388 13200438 13542097   
## 573 578 587 591 601 603 609   
## 6531885 7872891 15577964 6642285 17524720 9714222 8715778   
## 619 620 630 641 643 648 650   
## 2485104 354539830 17487194 9631217 1693776 12214779 27085144   
## 654 655 659 661 664 666 673   
## 60687309 3899727 14000471 12350511 9442307 145395910 21402188   
## 689 690 698 705 708 709 710   
## 5041016 5572151 39322830 23084121 28732891 34112333 10137844   
## 711 719 722 729 734 735 739   
## 78680796 39496701 24699738 12524805 11290048 2467496 5471706

SSE = sum((filtered\_df\_new$Date\_1 - predictTest\_date1)^2)  
SST = sum((filtered\_df\_new$Date\_1 - mean(filtered\_df\_train$Date\_1))^2)  
1-SSE/SST

## [1] -786102625

RMSE = sqrt(SSE/nrow(filtered\_df\_test))  
RMSE

## [1] 70808507

* To make prediction, predictTest was used to predict the outcome.
* R2 value was measured but it the value came out as negative value.
* Both with and without log, the p-value was much higher than 0.05.
* From the P-value and R2 value of model1, it was conlcuded that the outcome and date\_1 has no linear relationship.

**Ridge Regression**

outcome<-filtered\_df\_train[c(1,2,3,4,5,6,7,8,9,10,11,12),]  
date1<- filtered\_df\_train[c(13),]  
  
y <- as.matrix(filtered\_df\_train[,13])   
x <- as.matrix(filtered\_df\_train[,1:12])  
  
lambdas <- 10^seq(3, -2, by = -.1)  
  
fit <- glmnet(x, y, alpha = 0, lambda = lambdas)  
summary(fit)

## Length Class Mode   
## a0 51 -none- numeric  
## beta 612 dgCMatrix S4   
## df 51 -none- numeric  
## dim 2 -none- numeric  
## lambda 51 -none- numeric  
## dev.ratio 51 -none- numeric  
## nulldev 1 -none- numeric  
## npasses 1 -none- numeric  
## jerr 1 -none- numeric  
## offset 1 -none- logical  
## call 5 -none- call   
## nobs 1 -none- numeric

x.tr <- model.matrix(Date\_1 ~ Outcome\_M1 + Outcome\_M2 + Outcome\_M3 + Outcome\_M4 + Outcome\_M5 + Outcome\_M6 + Outcome\_M7 + Outcome\_M8 + Outcome\_M9 + Outcome\_M10 + Outcome\_M11 + Outcome\_M12,data=filtered\_df\_train)  
  
y.tr <- filtered\_df\_train$Date\_1  
  
fit <- glmnet(x.tr, y.tr, alpha = 50)

## Warning in glmnet(x.tr, y.tr, alpha = 50): alpha >1; set to 1

summary(fit)

## Length Class Mode   
## a0 73 -none- numeric  
## beta 949 dgCMatrix S4   
## df 73 -none- numeric  
## dim 2 -none- numeric  
## lambda 73 -none- numeric  
## dev.ratio 73 -none- numeric  
## nulldev 1 -none- numeric  
## npasses 1 -none- numeric  
## jerr 1 -none- numeric  
## offset 1 -none- logical  
## call 4 -none- call   
## nobs 1 -none- numeric

y\_predicted <- predict(fit, newx = x.tr)  
  
SSE = sum((y\_predicted- filtered\_df\_train$Date\_1 )^2)  
SST = sum((filtered\_df\_train$Date\_1)^2)  
  
rsq <- 1 - SSE / SST  
rsq

## [1] -1.667025

* Ridge regression was used to lower the p-value of model 1.
* Tuning alpha values of ridge regression model did not help to change the R2 value.
* Therefore, outcome, Date\_1 and Date\_2 has no linear relationship.

**Date\_2 and Outcomes linear model**

model2 = lm(Date\_2 ~ Outcome\_M1 + Outcome\_M2 + Outcome\_M3 + Outcome\_M4 + Outcome\_M5 + Outcome\_M6 + Outcome\_M7 +   
 Outcome\_M8 + Outcome\_M9 + Outcome\_M10 + Outcome\_M11 + Outcome\_M12,data=filtered\_df\_train)  
summary(model2)

##   
## Call:  
## lm(formula = Date\_2 ~ Outcome\_M1 + Outcome\_M2 + Outcome\_M3 +   
## Outcome\_M4 + Outcome\_M5 + Outcome\_M6 + Outcome\_M7 + Outcome\_M8 +   
## Outcome\_M9 + Outcome\_M10 + Outcome\_M11 + Outcome\_M12, data = filtered\_df\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1809.29 -526.84 69.88 547.62 1114.58   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 196.689 1611.075 0.122 0.903  
## Outcome\_M1 160.551 307.730 0.522 0.604  
## Outcome\_M2 26.178 354.089 0.074 0.941  
## Outcome\_M3 201.726 356.049 0.567 0.573  
## Outcome\_M4 -33.474 436.357 -0.077 0.939  
## Outcome\_M5 -317.466 391.773 -0.810 0.421  
## Outcome\_M6 531.460 473.677 1.122 0.266  
## Outcome\_M7 6.278 627.772 0.010 0.992  
## Outcome\_M8 153.060 608.159 0.252 0.802  
## Outcome\_M9 56.733 495.746 0.114 0.909  
## Outcome\_M10 -55.371 439.480 -0.126 0.900  
## Outcome\_M11 5.418 560.603 0.010 0.992  
## Outcome\_M12 -339.861 485.332 -0.700 0.486  
##   
## Residual standard error: 725.4 on 62 degrees of freedom  
## Multiple R-squared: 0.07899, Adjusted R-squared: -0.09927   
## F-statistic: 0.4431 on 12 and 62 DF, p-value: 0.939

* Model 2 was created to see the linear relationship between Date\_2 and Outcomes
* P- value was also high along with Model 1 above.
* Ridge-regression, Log10, without log10 was used to see the significance of the linear relationship but the p-value was significantly greater than 0.05.

# Test linear model for Date 2 and outcomes

predictTest\_date2 = predict(model2, newdata = filtered\_df\_test)  
predictTest\_date2

## 520 538 542 545 549 562 563   
## 95735566 5295932 3895416 5038664 15707298 6543169 11643378   
## 573 578 587 591 601 603 609   
## 4299991 4264638 10543846 4044396 9270528 6523050 6159633   
## 619 620 630 641 643 648 650   
## 2034628 148764670 8314642 6846411 2496683 6523070 24801395   
## 654 655 659 661 664 666 673   
## 26813269 2539699 7007160 6619283 6723975 61861163 10795715   
## 689 690 698 705 708 709 710   
## 3557573 4652485 17752233 11121431 12522158 16546323 6816055   
## 711 719 722 729 734 735 739   
## 36710174 16031619 8210762 7546996 6472213 2205003 5970175

SSE\_date2 = sum((filtered\_df\_new$Date\_2 - predictTest\_date2)^2)  
SST\_date2 = sum((filtered\_df\_new$Date\_2 - mean(filtered\_df\_train$Date\_2))^2)  
1-SSE\_date2/SST\_date2

## [1] -245891331

RMSE\_date2 = sqrt(SSE\_date2/nrow(filtered\_df\_test))  
RMSE\_date2

## [1] 31042629

* Similar to Model 1 above, the R2 value came out as negative value.

**Conclusion of linear regression Model 1 and Model 2**

As a result, it was concluded that both linear model 1 and model 2 has no linear relationship to Date\_1 and Date\_2. Mainly, 3 different methods were used to predict the outcomes. First method was not using the log method. Due to high values of the dataset, P-values were also greater than 0.05. Secondly, log was used in both training and testset to see the possiblity of reducing the P-values. This method slightly reduced the p-value was no signicance were measured. Lastly, ridge regression was used to measure the R2 but changing alpha values did not changed the R2 value. Therefore, Date\_1 and Date\_2 has no linear relationship with Outcomes.

**Quantitative and Date\_1&Date\_2**

#Date\_1 &2 and Quantitative variables   
filtered\_quan\_df <-OnlineProductSale[c(4,32,36,41,43,44,57,58,59,83,  
 85,89,91,94,101,109,131, 137, 142, 146,  
 157,164,165,176,179, 182, 197, 206, 216, 218,   
 220, 222,231, 238, 252, 260, 276, 294, 295, 298,   
 307, 313, 320, 322, 324, 325, 340, 341, 344, 348,   
 353, 355, 356, 357, 358, 361,391, 395, 404, 411,   
 417, 419, 440, 442, 460, 462, 465, 477, 479, 485,   
 493, 496, 503, 513, 518, 520, 538, 542, 545, 549,   
 562, 563, 573,578, 587, 591, 601, 603, 609, 619,   
 620, 630, 641, 643, 648, 650, 654, 655, 659, 661,   
 664, 666, 673, 689, 690, 698, 705, 708, 709, 710,  
 711, 719, 722, 729, 734, 735, 739),c(14:31)]  
  
filtered\_quan\_df

## Date\_1 Date\_2 Quan\_1 Quan\_2 Quan\_3 Quan\_4 Quan\_5 Quan\_6 Quan\_7  
## 4 2296 2018 685300 7900 101 1416697 29 8 10  
## 32 2443 2215 1804500 8800 62 28203 29 8 10  
## 36 1659 1508 356000 7900 61 652377 29 8 10  
## 41 2450 1569 1804500 7200 105 856200 29 8 10  
## 43 2782 2401 1332300 8400 63 1649272 29 8 10  
## 44 2490 2162 918600 8500 63 2053329 29 8 10  
## 57 3192 2960 3518300 8600 92 1172663 36 13 17  
## 58 2023 1885 950000 8100 44 438659 29 8 10  
## 59 3206 2831 3570700 7300 100 872255 76 33 26  
## 83 2849 2639 1463400 9000 82 802286 29 8 10  
## 85 1421 1195 52200 8300 110 870647 29 8 10  
## 89 1680 1231 379300 8400 144 1173440 29 8 10  
## 91 1883 1526 805400 7800 43 697011 29 8 10  
## 94 1778 1231 31600 8600 121 1399606 29 8 10  
## 101 2100 1584 1139100 7300 49 788296 29 8 10  
## 109 2064 1231 281900 8300 90 372875 29 8 10  
## 131 3171 2960 3518300 8800 68 16504 23 9 14  
## 137 2401 1190 820700 8400 95 336930 29 8 10  
## 142 1784 1525 538300 8800 110 527439 29 8 10  
## 146 2485 2025 1864300 8400 113 538046 29 8 10  
## 157 2415 1987 850500 8500 58 1238684 29 8 10  
## 164 2870 2450 1578000 9300 114 3220430 29 8 10  
## 165 2387 2226 820700 8000 30 539483 29 8 10  
## 176 1757 1301 456200 8200 107 689611 29 8 10  
## 179 2877 2521 2789600 8900 142 1394197 29 8 10  
## 182 1715 1603 421100 7900 67 90029 29 8 10  
## 197 2982 2738 1950700 8900 53 676207 32 4 10  
## 206 2793 2417 1332300 9100 131 582649 29 8 10  
## 216 2961 2500 1900100 7400 112 941734 22 7 9  
## 218 1694 1508 379300 8000 86 1951543 29 8 10  
## 220 2877 2521 1578000 8900 118 1394197 29 8 10  
## 222 2380 2004 1717000 8400 81 212135 29 8 10  
## 231 2835 2488 2657500 8000 61 13671 29 8 10  
## 238 3199 2988 3570700 8800 74 696622 313 26 273  
## 252 3101 2628 3311300 7900 81 940341 30 10 7  
## 260 2478 2337 918600 8700 58 222511 29 8 10  
## 276 3481 3348 2801100 8200 28 171279 102 27 53  
## 294 2354 1044 784400 9300 144 1650772 29 8 10  
## 295 2134 1511 1263700 9300 120 1808023 29 8 10  
## 298 3229 3026 3792300 7500 65 1969756 80 8 42  
## 307 2604 2123 2344300 9200 124 752566 29 8 10  
## 313 3234 2689 2301100 8600 107 4365238 155 57 31  
## 320 3241 2934 2301100 8900 102 1529105 121 43 37  
## 322 3115 2926 2088600 8400 38 359080 48 9 19  
## 324 2100 1584 320800 7400 40 788296 29 8 10  
## 325 2485 1197 1864300 8800 149 483316 29 8 10  
## 340 2142 1354 396300 8000 87 1168518 29 8 10  
## 341 3220 2626 3570700 7900 118 457014 62 22 21  
## 344 2863 857 1578000 8600 77 1348716 29 8 10  
## 348 3249 1728 2301100 8300 111 931417 107 20 48  
## 353 1528 1171 194600 9000 133 823479 29 8 10  
## 355 3395 2985 4355300 9400 112 169441 58 27 11  
## 356 1511 860 163700 8500 115 467424 29 8 10  
## 357 2492 365 1864300 9200 137 1329383 29 8 10  
## 358 2058 1010 994700 9500 149 1005204 29 8 10  
## 361 2961 2261 3122700 8700 151 834327 29 8 10  
## 391 1421 365 52200 8100 145 1003761 29 8 10  
## 395 2058 1757 281900 8500 39 2006285 29 8 10  
## 404 2478 1955 1864300 8200 108 975913 29 8 10  
## 411 3229 3026 3792300 8200 79 5417231 134 11 79  
## 417 2485 1961 1864300 8000 91 883301 29 8 10  
## 419 2506 2162 1999100 8500 125 2408730 29 8 10  
## 440 2856 2676 2657500 9200 132 725087 29 8 10  
## 442 2415 1987 1748500 8300 76 1238684 29 8 10  
## 460 2107 1654 1139100 8200 66 18034 29 8 10  
## 462 2751 2610 1298400 7900 36 483256 29 8 10  
## 465 2499 1583 1999100 6700 104 335300 29 8 10  
## 477 2863 857 2789600 8600 115 1348716 29 8 10  
## 479 2947 2233 3054600 9500 166 1784743 29 8 10  
## 485 2982 2590 1950700 8800 104 1247336 29 8 10  
## 493 2051 1802 281900 8100 55 1503860 29 8 10  
## 496 2877 2708 2789600 8800 126 649910 29 8 10  
## 503 3059 1231 3238500 9400 142 2363963 29 8 10  
## 513 3157 2057 3440400 7400 116 850888 29 8 10  
## 518 2051 1802 994700 8300 77 1503860 29 8 10  
## 520 3178 2429 3518300 9000 136 2180621 29 8 10  
## 538 1421 1197 52200 8800 141 873287 29 8 10  
## 542 3213 2908 3570700 7500 59 438942 23 7 7  
## 545 1646 1301 356000 7900 81 123327 29 8 10  
## 549 1539 983 194600 9300 156 922361 29 8 10  
## 562 2968 2598 3122700 6600 97 405694 29 8 10  
## 563 2471 2039 1864300 8300 49 411958 29 8 10  
## 573 2107 1654 320800 8100 49 18034 29 8 10  
## 578 1904 983 192800 9200 80 922361 29 8 10  
## 587 3556 3388 2896100 9000 64 233160 27 14 6  
## 591 1652 802 356000 7100 94 123467 29 8 10  
## 601 3535 1605 2896100 7200 75 225894 36 13 3  
## 603 2485 1961 918600 8100 73 883301 29 8 10  
## 609 2471 2264 918600 7600 39 513282 29 8 10  
## 619 1526 1016 194600 8800 100 561962 29 8 10  
## 620 3234 2689 3792300 8600 137 4365238 204 73 39  
## 630 1421 959 52200 7300 82 1566700 29 8 10  
## 641 1745 1584 456200 8000 71 1164838 29 8 10  
## 643 2443 2215 887900 8800 34 28203 29 8 10  
## 648 2121 1900 1139100 8000 77 385852 29 8 10  
## 650 3481 3348 4525300 8300 37 171279 131 28 79  
## 654 2628 1158 2397500 8400 136 1345876 29 8 10  
## 655 1421 1227 52200 7000 64 437694 29 8 10  
## 659 3192 2569 3518300 7900 109 669770 29 8 10  
## 661 2478 2337 1864300 8700 74 222511 29 8 10  
## 664 2471 2264 1864300 7700 48 513282 29 8 10  
## 666 2310 1470 1618300 9600 146 2479325 29 8 10  
## 673 2492 365 918600 9000 96 1329383 29 8 10  
## 689 3080 2942 2054000 7900 64 145077 47 4 24  
## 690 3171 2960 2175300 8800 55 16504 26 11 13  
## 698 2618 1730 2397500 8100 135 451502 29 8 10  
## 705 1750 1171 456200 8500 97 648530 29 8 10  
## 708 2604 2123 1175400 9200 113 752566 29 8 10  
## 709 3395 2582 2704600 8600 85 243024 91 36 21  
## 710 3192 2960 2175300 8700 78 1172663 34 14 17  
## 711 2310 1470 685300 9700 112 2089715 29 8 10  
## 719 3229 3026 3792300 7400 66 1242470 99 10 48  
## 722 3143 2738 3440400 8400 72 3030132 89 22 30  
## 729 2849 2052 1463400 8200 103 907724 29 8 10  
## 734 2800 2528 2617100 8500 107 793713 29 8 10  
## 735 2100 1781 320800 8300 26 534158 29 8 10  
## 739 3451 3361 4480600 4500 6 144409 13 0 7  
## Quan\_8 Quan\_9 Quan\_10 Quan\_11 Quan\_12 Quan\_13 Quan\_14 Quan\_15 Quan\_16  
## 4 41 15 6 3 2048 5121 16 76 76  
## 32 120 15 6 3 12288 9216 12 30 30  
## 36 13 15 6 3 1024 8192 2 119 119  
## 41 41 15 6 3 288 320 9 52 55  
## 43 41 15 6 3 1572864 282624 2 32 32  
## 44 3 15 6 3 60 2355 8 52 55  
## 57 41 7 5 1 8763 30720 22 52 55  
## 58 41 15 6 3 5767168 12288 2 119 119  
## 59 41 37 24 3 750 1280 24 52 55  
## 83 24 15 6 3 10240 30720 20 544 544  
## 85 41 15 6 3 4096 125 4 52 55  
## 89 41 15 6 3 5693 5120 9 52 55  
## 91 41 15 6 3 90 2816 2 52 55  
## 94 30 15 6 3 2816 122880 40 52 55  
## 101 41 15 6 3 664576 4096 4 30 30  
## 109 156 15 6 3 8192 829440 32 52 55  
## 131 41 6 3 1 51200 13312 12 52 55  
## 137 41 15 6 3 7168 2048 2 33 33  
## 142 22 15 6 3 400 120 16 17 17  
## 146 2 15 6 3 5120 6144 16 9 55  
## 157 2 15 6 3 6144 225280 2 32 32  
## 164 109 15 6 3 479 51200 18 52 55  
## 165 3 15 6 3 1048576 97280 2 120 120  
## 176 41 15 6 3 10317 264 4 33 33  
## 179 41 15 6 3 1024 4096 9 52 55  
## 182 41 15 6 3 2048 512329 2 35 35  
## 197 19 10 3 0 5120 5242880 2 30 30  
## 206 41 15 6 3 1024 1258291 9 52 55  
## 216 41 11 4 2 2560 1218 9 52 55  
## 218 41 15 6 3 4096 6144 2 32 32  
## 220 7 15 6 3 128 1887436 9 52 55  
## 222 41 15 6 3 2697216 2048 4 16 16  
## 231 41 15 6 3 7111 4096 10 34 34  
## 238 41 288 19 265 3890216 4096 10 30 30  
## 252 41 18 8 0 256 1024 9 167 167  
## 260 41 15 6 3 31744 23552 20 512 512  
## 276 41 37 21 5 497432 168960 2 52 55  
## 294 41 15 6 3 1024 5233664 9 52 55  
## 295 30 15 6 3 262144 4096 18 52 55  
## 298 25 24 5 4 4096 5120 9 11 11  
## 307 41 15 6 3 7168 140 2 19 19  
## 313 20 68 39 1 68 51200 18 52 55  
## 320 29 61 27 5 3072 4194304 8 52 15  
## 322 41 32 8 7 400 164864 2 52 55  
## 324 12 15 6 3 5355 4096 8 30 30  
## 325 16 15 6 3 2048 10240 9 52 55  
## 340 31 15 6 3 1047 2097152 9 52 55  
## 341 30 27 14 2 512 12288 2 2 2  
## 344 41 15 6 3 4096 3145728 9 6 55  
## 348 53 41 7 9 8192 262144 16 1039 1039  
## 353 41 15 6 3 1024 4096 16 52 4  
## 355 41 44 25 6 15360 21504 2 52 55  
## 356 41 15 6 3 300 132 2 27 27  
## 357 41 15 6 3 1271 8192 9 52 55  
## 358 36 15 6 3 2936012 15360 9 52 55  
## 361 41 15 6 3 3584000 2048 10 52 5  
## 391 41 15 6 3 40960 100 32 6 6  
## 395 41 15 6 3 6656 3072 16 52 2  
## 404 41 15 6 3 10240 4096 12 52 55  
## 411 41 28 5 4 5120 3072 9 9 9  
## 417 41 15 6 3 4096 600 16 46 46  
## 419 20 15 6 3 3145728 4096 18 52 55  
## 440 41 15 6 3 552 7168 8 405 405  
## 442 35 15 6 3 51200 41984 2 32 32  
## 460 41 15 6 3 2048 4096 4 621 621  
## 462 41 15 6 3 200 164864 12 52 55  
## 465 41 15 6 3 60 1536 12 52 55  
## 477 11 15 6 3 240 34816 9 6 55  
## 479 41 15 6 3 1048576 3072 9 11 55  
## 485 20 15 6 3 10240 1143 24 52 55  
## 493 40 15 6 3 2048 225280 2 32 32  
## 496 41 15 6 3 100 1024 8 4 4  
## 503 41 15 6 3 5120 800 16 52 55  
## 513 41 15 6 3 4096 36864 9 52 55  
## 518 41 15 6 3 54272 11264 2 32 32  
## 520 41 15 6 3 497432 512329 16 52 55  
## 538 41 15 6 3 497432 380 8 80 80  
## 542 41 6 2 1 497432 240 2 42 42  
## 545 41 15 6 3 497432 2048 4 52 7  
## 549 41 15 6 3 497432 8192 9 20 55  
## 562 41 15 6 3 497432 60 18 3 22  
## 563 41 15 6 3 497432 4096 10 30 30  
## 573 41 15 6 3 497432 2611 4 621 621  
## 578 41 15 6 3 497432 4710400 9 20 55  
## 587 41 19 13 0 497432 30720 22 52 55  
## 591 41 15 6 3 497432 240 12 52 55  
## 601 41 21 11 0 497432 52224 4 4 4  
## 603 41 15 6 3 497432 55296 16 46 46  
## 609 41 15 6 3 497432 4096 10 54 54  
## 619 41 15 6 3 497432 80 6 72 72  
## 620 41 83 46 2 497432 4096 18 52 55  
## 630 41 15 6 3 497432 4096 2 32 32  
## 641 41 15 6 3 497432 448 4 53 53  
## 643 41 15 6 3 497432 9216 12 30 30  
## 648 41 15 6 3 497432 1024 16 26 26  
## 650 41 103 22 73 497432 7372 2 52 55  
## 654 41 15 6 3 497432 140 2 10 10  
## 655 41 15 6 3 497432 288 4 25 25  
## 659 41 15 6 3 497432 356 4 52 55  
## 661 41 15 6 3 497432 23552 20 512 512  
## 664 41 15 6 3 497432 4096 10 54 54  
## 666 41 15 6 3 497432 1024 16 52 55  
## 673 41 15 6 3 497432 5120000 9 52 55  
## 689 41 12 3 2 497432 71680 24 52 55  
## 690 41 9 5 1 497432 22528 12 52 55  
## 698 41 15 6 3 497432 1024 6 52 55  
## 705 41 15 6 3 497432 4096 6 52 9  
## 708 41 15 6 3 497432 55296 2 19 19  
## 709 41 66 33 9 497432 4194304 8 27 27  
## 710 41 9 6 1 497432 30720 22 52 55  
## 711 41 15 6 3 497432 3424256 16 52 55  
## 719 41 28 6 4 497432 1024 4 52 55  
## 722 41 61 17 12 497432 16035 6 32 32  
## 729 41 15 6 3 497432 2831155 4 52 55  
## 734 41 15 6 3 497432 4096 8 52 55  
## 735 41 15 6 3 497432 4000 10 52 55  
## 739 41 4 0 1 497432 1024 9 12 12

filtered\_quan\_train <- filtered\_quan\_df[c(1:75),c(1:18)]  
filtered\_quan\_train

## Date\_1 Date\_2 Quan\_1 Quan\_2 Quan\_3 Quan\_4 Quan\_5 Quan\_6 Quan\_7  
## 4 2296 2018 685300 7900 101 1416697 29 8 10  
## 32 2443 2215 1804500 8800 62 28203 29 8 10  
## 36 1659 1508 356000 7900 61 652377 29 8 10  
## 41 2450 1569 1804500 7200 105 856200 29 8 10  
## 43 2782 2401 1332300 8400 63 1649272 29 8 10  
## 44 2490 2162 918600 8500 63 2053329 29 8 10  
## 57 3192 2960 3518300 8600 92 1172663 36 13 17  
## 58 2023 1885 950000 8100 44 438659 29 8 10  
## 59 3206 2831 3570700 7300 100 872255 76 33 26  
## 83 2849 2639 1463400 9000 82 802286 29 8 10  
## 85 1421 1195 52200 8300 110 870647 29 8 10  
## 89 1680 1231 379300 8400 144 1173440 29 8 10  
## 91 1883 1526 805400 7800 43 697011 29 8 10  
## 94 1778 1231 31600 8600 121 1399606 29 8 10  
## 101 2100 1584 1139100 7300 49 788296 29 8 10  
## 109 2064 1231 281900 8300 90 372875 29 8 10  
## 131 3171 2960 3518300 8800 68 16504 23 9 14  
## 137 2401 1190 820700 8400 95 336930 29 8 10  
## 142 1784 1525 538300 8800 110 527439 29 8 10  
## 146 2485 2025 1864300 8400 113 538046 29 8 10  
## 157 2415 1987 850500 8500 58 1238684 29 8 10  
## 164 2870 2450 1578000 9300 114 3220430 29 8 10  
## 165 2387 2226 820700 8000 30 539483 29 8 10  
## 176 1757 1301 456200 8200 107 689611 29 8 10  
## 179 2877 2521 2789600 8900 142 1394197 29 8 10  
## 182 1715 1603 421100 7900 67 90029 29 8 10  
## 197 2982 2738 1950700 8900 53 676207 32 4 10  
## 206 2793 2417 1332300 9100 131 582649 29 8 10  
## 216 2961 2500 1900100 7400 112 941734 22 7 9  
## 218 1694 1508 379300 8000 86 1951543 29 8 10  
## 220 2877 2521 1578000 8900 118 1394197 29 8 10  
## 222 2380 2004 1717000 8400 81 212135 29 8 10  
## 231 2835 2488 2657500 8000 61 13671 29 8 10  
## 238 3199 2988 3570700 8800 74 696622 313 26 273  
## 252 3101 2628 3311300 7900 81 940341 30 10 7  
## 260 2478 2337 918600 8700 58 222511 29 8 10  
## 276 3481 3348 2801100 8200 28 171279 102 27 53  
## 294 2354 1044 784400 9300 144 1650772 29 8 10  
## 295 2134 1511 1263700 9300 120 1808023 29 8 10  
## 298 3229 3026 3792300 7500 65 1969756 80 8 42  
## 307 2604 2123 2344300 9200 124 752566 29 8 10  
## 313 3234 2689 2301100 8600 107 4365238 155 57 31  
## 320 3241 2934 2301100 8900 102 1529105 121 43 37  
## 322 3115 2926 2088600 8400 38 359080 48 9 19  
## 324 2100 1584 320800 7400 40 788296 29 8 10  
## 325 2485 1197 1864300 8800 149 483316 29 8 10  
## 340 2142 1354 396300 8000 87 1168518 29 8 10  
## 341 3220 2626 3570700 7900 118 457014 62 22 21  
## 344 2863 857 1578000 8600 77 1348716 29 8 10  
## 348 3249 1728 2301100 8300 111 931417 107 20 48  
## 353 1528 1171 194600 9000 133 823479 29 8 10  
## 355 3395 2985 4355300 9400 112 169441 58 27 11  
## 356 1511 860 163700 8500 115 467424 29 8 10  
## 357 2492 365 1864300 9200 137 1329383 29 8 10  
## 358 2058 1010 994700 9500 149 1005204 29 8 10  
## 361 2961 2261 3122700 8700 151 834327 29 8 10  
## 391 1421 365 52200 8100 145 1003761 29 8 10  
## 395 2058 1757 281900 8500 39 2006285 29 8 10  
## 404 2478 1955 1864300 8200 108 975913 29 8 10  
## 411 3229 3026 3792300 8200 79 5417231 134 11 79  
## 417 2485 1961 1864300 8000 91 883301 29 8 10  
## 419 2506 2162 1999100 8500 125 2408730 29 8 10  
## 440 2856 2676 2657500 9200 132 725087 29 8 10  
## 442 2415 1987 1748500 8300 76 1238684 29 8 10  
## 460 2107 1654 1139100 8200 66 18034 29 8 10  
## 462 2751 2610 1298400 7900 36 483256 29 8 10  
## 465 2499 1583 1999100 6700 104 335300 29 8 10  
## 477 2863 857 2789600 8600 115 1348716 29 8 10  
## 479 2947 2233 3054600 9500 166 1784743 29 8 10  
## 485 2982 2590 1950700 8800 104 1247336 29 8 10  
## 493 2051 1802 281900 8100 55 1503860 29 8 10  
## 496 2877 2708 2789600 8800 126 649910 29 8 10  
## 503 3059 1231 3238500 9400 142 2363963 29 8 10  
## 513 3157 2057 3440400 7400 116 850888 29 8 10  
## 518 2051 1802 994700 8300 77 1503860 29 8 10  
## Quan\_8 Quan\_9 Quan\_10 Quan\_11 Quan\_12 Quan\_13 Quan\_14 Quan\_15 Quan\_16  
## 4 41 15 6 3 2048 5121 16 76 76  
## 32 120 15 6 3 12288 9216 12 30 30  
## 36 13 15 6 3 1024 8192 2 119 119  
## 41 41 15 6 3 288 320 9 52 55  
## 43 41 15 6 3 1572864 282624 2 32 32  
## 44 3 15 6 3 60 2355 8 52 55  
## 57 41 7 5 1 8763 30720 22 52 55  
## 58 41 15 6 3 5767168 12288 2 119 119  
## 59 41 37 24 3 750 1280 24 52 55  
## 83 24 15 6 3 10240 30720 20 544 544  
## 85 41 15 6 3 4096 125 4 52 55  
## 89 41 15 6 3 5693 5120 9 52 55  
## 91 41 15 6 3 90 2816 2 52 55  
## 94 30 15 6 3 2816 122880 40 52 55  
## 101 41 15 6 3 664576 4096 4 30 30  
## 109 156 15 6 3 8192 829440 32 52 55  
## 131 41 6 3 1 51200 13312 12 52 55  
## 137 41 15 6 3 7168 2048 2 33 33  
## 142 22 15 6 3 400 120 16 17 17  
## 146 2 15 6 3 5120 6144 16 9 55  
## 157 2 15 6 3 6144 225280 2 32 32  
## 164 109 15 6 3 479 51200 18 52 55  
## 165 3 15 6 3 1048576 97280 2 120 120  
## 176 41 15 6 3 10317 264 4 33 33  
## 179 41 15 6 3 1024 4096 9 52 55  
## 182 41 15 6 3 2048 512329 2 35 35  
## 197 19 10 3 0 5120 5242880 2 30 30  
## 206 41 15 6 3 1024 1258291 9 52 55  
## 216 41 11 4 2 2560 1218 9 52 55  
## 218 41 15 6 3 4096 6144 2 32 32  
## 220 7 15 6 3 128 1887436 9 52 55  
## 222 41 15 6 3 2697216 2048 4 16 16  
## 231 41 15 6 3 7111 4096 10 34 34  
## 238 41 288 19 265 3890216 4096 10 30 30  
## 252 41 18 8 0 256 1024 9 167 167  
## 260 41 15 6 3 31744 23552 20 512 512  
## 276 41 37 21 5 497432 168960 2 52 55  
## 294 41 15 6 3 1024 5233664 9 52 55  
## 295 30 15 6 3 262144 4096 18 52 55  
## 298 25 24 5 4 4096 5120 9 11 11  
## 307 41 15 6 3 7168 140 2 19 19  
## 313 20 68 39 1 68 51200 18 52 55  
## 320 29 61 27 5 3072 4194304 8 52 15  
## 322 41 32 8 7 400 164864 2 52 55  
## 324 12 15 6 3 5355 4096 8 30 30  
## 325 16 15 6 3 2048 10240 9 52 55  
## 340 31 15 6 3 1047 2097152 9 52 55  
## 341 30 27 14 2 512 12288 2 2 2  
## 344 41 15 6 3 4096 3145728 9 6 55  
## 348 53 41 7 9 8192 262144 16 1039 1039  
## 353 41 15 6 3 1024 4096 16 52 4  
## 355 41 44 25 6 15360 21504 2 52 55  
## 356 41 15 6 3 300 132 2 27 27  
## 357 41 15 6 3 1271 8192 9 52 55  
## 358 36 15 6 3 2936012 15360 9 52 55  
## 361 41 15 6 3 3584000 2048 10 52 5  
## 391 41 15 6 3 40960 100 32 6 6  
## 395 41 15 6 3 6656 3072 16 52 2  
## 404 41 15 6 3 10240 4096 12 52 55  
## 411 41 28 5 4 5120 3072 9 9 9  
## 417 41 15 6 3 4096 600 16 46 46  
## 419 20 15 6 3 3145728 4096 18 52 55  
## 440 41 15 6 3 552 7168 8 405 405  
## 442 35 15 6 3 51200 41984 2 32 32  
## 460 41 15 6 3 2048 4096 4 621 621  
## 462 41 15 6 3 200 164864 12 52 55  
## 465 41 15 6 3 60 1536 12 52 55  
## 477 11 15 6 3 240 34816 9 6 55  
## 479 41 15 6 3 1048576 3072 9 11 55  
## 485 20 15 6 3 10240 1143 24 52 55  
## 493 40 15 6 3 2048 225280 2 32 32  
## 496 41 15 6 3 100 1024 8 4 4  
## 503 41 15 6 3 5120 800 16 52 55  
## 513 41 15 6 3 4096 36864 9 52 55  
## 518 41 15 6 3 54272 11264 2 32 32

filtered\_quan\_test <- filtered\_quan\_df[c(76:117),c(1:18)]  
filtered\_quan\_test

## Date\_1 Date\_2 Quan\_1 Quan\_2 Quan\_3 Quan\_4 Quan\_5 Quan\_6 Quan\_7  
## 520 3178 2429 3518300 9000 136 2180621 29 8 10  
## 538 1421 1197 52200 8800 141 873287 29 8 10  
## 542 3213 2908 3570700 7500 59 438942 23 7 7  
## 545 1646 1301 356000 7900 81 123327 29 8 10  
## 549 1539 983 194600 9300 156 922361 29 8 10  
## 562 2968 2598 3122700 6600 97 405694 29 8 10  
## 563 2471 2039 1864300 8300 49 411958 29 8 10  
## 573 2107 1654 320800 8100 49 18034 29 8 10  
## 578 1904 983 192800 9200 80 922361 29 8 10  
## 587 3556 3388 2896100 9000 64 233160 27 14 6  
## 591 1652 802 356000 7100 94 123467 29 8 10  
## 601 3535 1605 2896100 7200 75 225894 36 13 3  
## 603 2485 1961 918600 8100 73 883301 29 8 10  
## 609 2471 2264 918600 7600 39 513282 29 8 10  
## 619 1526 1016 194600 8800 100 561962 29 8 10  
## 620 3234 2689 3792300 8600 137 4365238 204 73 39  
## 630 1421 959 52200 7300 82 1566700 29 8 10  
## 641 1745 1584 456200 8000 71 1164838 29 8 10  
## 643 2443 2215 887900 8800 34 28203 29 8 10  
## 648 2121 1900 1139100 8000 77 385852 29 8 10  
## 650 3481 3348 4525300 8300 37 171279 131 28 79  
## 654 2628 1158 2397500 8400 136 1345876 29 8 10  
## 655 1421 1227 52200 7000 64 437694 29 8 10  
## 659 3192 2569 3518300 7900 109 669770 29 8 10  
## 661 2478 2337 1864300 8700 74 222511 29 8 10  
## 664 2471 2264 1864300 7700 48 513282 29 8 10  
## 666 2310 1470 1618300 9600 146 2479325 29 8 10  
## 673 2492 365 918600 9000 96 1329383 29 8 10  
## 689 3080 2942 2054000 7900 64 145077 47 4 24  
## 690 3171 2960 2175300 8800 55 16504 26 11 13  
## 698 2618 1730 2397500 8100 135 451502 29 8 10  
## 705 1750 1171 456200 8500 97 648530 29 8 10  
## 708 2604 2123 1175400 9200 113 752566 29 8 10  
## 709 3395 2582 2704600 8600 85 243024 91 36 21  
## 710 3192 2960 2175300 8700 78 1172663 34 14 17  
## 711 2310 1470 685300 9700 112 2089715 29 8 10  
## 719 3229 3026 3792300 7400 66 1242470 99 10 48  
## 722 3143 2738 3440400 8400 72 3030132 89 22 30  
## 729 2849 2052 1463400 8200 103 907724 29 8 10  
## 734 2800 2528 2617100 8500 107 793713 29 8 10  
## 735 2100 1781 320800 8300 26 534158 29 8 10  
## 739 3451 3361 4480600 4500 6 144409 13 0 7  
## Quan\_8 Quan\_9 Quan\_10 Quan\_11 Quan\_12 Quan\_13 Quan\_14 Quan\_15 Quan\_16  
## 520 41 15 6 3 497432 512329 16 52 55  
## 538 41 15 6 3 497432 380 8 80 80  
## 542 41 6 2 1 497432 240 2 42 42  
## 545 41 15 6 3 497432 2048 4 52 7  
## 549 41 15 6 3 497432 8192 9 20 55  
## 562 41 15 6 3 497432 60 18 3 22  
## 563 41 15 6 3 497432 4096 10 30 30  
## 573 41 15 6 3 497432 2611 4 621 621  
## 578 41 15 6 3 497432 4710400 9 20 55  
## 587 41 19 13 0 497432 30720 22 52 55  
## 591 41 15 6 3 497432 240 12 52 55  
## 601 41 21 11 0 497432 52224 4 4 4  
## 603 41 15 6 3 497432 55296 16 46 46  
## 609 41 15 6 3 497432 4096 10 54 54  
## 619 41 15 6 3 497432 80 6 72 72  
## 620 41 83 46 2 497432 4096 18 52 55  
## 630 41 15 6 3 497432 4096 2 32 32  
## 641 41 15 6 3 497432 448 4 53 53  
## 643 41 15 6 3 497432 9216 12 30 30  
## 648 41 15 6 3 497432 1024 16 26 26  
## 650 41 103 22 73 497432 7372 2 52 55  
## 654 41 15 6 3 497432 140 2 10 10  
## 655 41 15 6 3 497432 288 4 25 25  
## 659 41 15 6 3 497432 356 4 52 55  
## 661 41 15 6 3 497432 23552 20 512 512  
## 664 41 15 6 3 497432 4096 10 54 54  
## 666 41 15 6 3 497432 1024 16 52 55  
## 673 41 15 6 3 497432 5120000 9 52 55  
## 689 41 12 3 2 497432 71680 24 52 55  
## 690 41 9 5 1 497432 22528 12 52 55  
## 698 41 15 6 3 497432 1024 6 52 55  
## 705 41 15 6 3 497432 4096 6 52 9  
## 708 41 15 6 3 497432 55296 2 19 19  
## 709 41 66 33 9 497432 4194304 8 27 27  
## 710 41 9 6 1 497432 30720 22 52 55  
## 711 41 15 6 3 497432 3424256 16 52 55  
## 719 41 28 6 4 497432 1024 4 52 55  
## 722 41 61 17 12 497432 16035 6 32 32  
## 729 41 15 6 3 497432 2831155 4 52 55  
## 734 41 15 6 3 497432 4096 8 52 55  
## 735 41 15 6 3 497432 4000 10 52 55  
## 739 41 4 0 1 497432 1024 9 12 12

filtered\_quan\_test\_new <- filtered\_quan\_df[c(76:117),c(1:18)]  
filtered\_quan\_test\_new <- log10(filtered\_quan\_test\_new)

* Training and test sets were made before the real test with Test.csv.
* There are 16 quantitative variables available which it measures temperature, area, discounts, and ext.

**(Model3) Date\_1 and Quan linear model**

model3 = lm(Date\_1 ~ Quan\_1 + Quan\_2 + Quan\_3 + Quan\_4 + Quan\_5 + Quan\_6 + Quan\_7 + Quan\_8 + Quan\_9 +   
 Quan\_10 + Quan\_11 + Quan\_12 + Quan\_13 + Quan\_14 + Quan\_15 + Quan\_16, data=filtered\_quan\_train)  
summary(model3)

##   
## Call:  
## lm(formula = Date\_1 ~ Quan\_1 + Quan\_2 + Quan\_3 + Quan\_4 + Quan\_5 +   
## Quan\_6 + Quan\_7 + Quan\_8 + Quan\_9 + Quan\_10 + Quan\_11 + Quan\_12 +   
## Quan\_13 + Quan\_14 + Quan\_15 + Quan\_16, data = filtered\_quan\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -313.15 -141.38 -26.22 106.55 507.00   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.334e+03 4.014e+02 3.324 0.001543 \*\*   
## Quan\_1 4.307e-04 2.847e-05 15.125 < 2e-16 \*\*\*  
## Quan\_2 7.012e-02 5.232e-02 1.340 0.185408   
## Quan\_3 -3.632e+00 9.272e-01 -3.917 0.000239 \*\*\*  
## Quan\_4 6.921e-05 4.355e-05 1.589 0.117453   
## Quan\_5 -1.116e+01 1.086e+01 -1.028 0.308441   
## Quan\_6 1.996e+01 2.048e+01 0.974 0.333883   
## Quan\_7 7.852e+00 1.292e+01 0.608 0.545634   
## Quan\_8 -3.295e-01 1.189e+00 -0.277 0.782647   
## Quan\_9 2.191e+01 1.734e+01 1.263 0.211550   
## Quan\_10 -2.525e+01 2.757e+01 -0.916 0.363582   
## Quan\_11 -1.914e+01 1.765e+01 -1.084 0.282677   
## Quan\_12 -1.021e-05 2.691e-05 -0.380 0.705676   
## Quan\_13 8.581e-05 2.650e-05 3.239 0.001990 \*\*   
## Quan\_14 5.323e+00 3.787e+00 1.405 0.165223   
## Quan\_15 -1.351e+00 1.784e+00 -0.758 0.451720   
## Quan\_16 1.700e+00 1.764e+00 0.964 0.339243   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 216.6 on 58 degrees of freedom  
## Multiple R-squared: 0.8751, Adjusted R-squared: 0.8407   
## F-statistic: 25.4 on 16 and 58 DF, p-value: < 2.2e-16

predictTest\_quan\_date1 = predict(model3,newdata = filtered\_quan\_test)  
predictTest\_quan\_date1

## 520 538 542 545 549 562 563 573   
## 3305.416 1608.112 3220.479 1734.453 1697.988 2962.365 2649.259 2117.273   
## 578 587 591 601 603 609 619 620   
## 2369.751 3233.787 1755.184 2901.632 2215.296 2244.583 1783.356 3580.437   
## 630 641 643 648 650 654 655 659   
## 1716.866 1969.746 2302.810 2242.659 3778.837 2584.641 1690.950 3113.981   
## 661 664 666 673 689 690 698 705   
## 2796.396 2626.201 2469.667 2630.272 2581.381 2843.122 2546.454 1812.135   
## 708 709 710 711 719 722 729 734   
## 2164.752 3511.428 2863.255 2465.133 3215.363 3737.556 2531.181 2805.378   
## 735 739   
## 2089.265 3579.576

SSE\_quan\_date1 = sum((filtered\_quan\_test$Date\_1 - predictTest\_quan\_date1)^2)  
SST\_quan\_date1 = sum((filtered\_quan\_df$Date\_1 - mean(filtered\_quan\_train$Date\_1))^2)  
1-SSE\_quan\_date1/SST\_quan\_date1

## [1] 0.9289686

RMSE\_quan\_date1 = sqrt(SSE\_quan\_date1/nrow(filtered\_quan\_test))  
RMSE\_quan\_date1

## [1] 260.3271

* The adjusted R-squared and p-value of the Model3 shows that Quantitative variables are strongly significant to the Date\_1.
* P-value is less than 0.05 therefore stong significance were measured.
* R-squared value of Model3 is 0.929 which it tells that the model has a good fit.

**Date\_2 and Quan linear model**

model4 = lm(Date\_2 ~ Quan\_1 + Quan\_2 + Quan\_3 + Quan\_4 + Quan\_5 + Quan\_6 + Quan\_7 + Quan\_8 + Quan\_9 +   
 Quan\_10 + Quan\_11 + Quan\_12 + Quan\_13 + Quan\_14 + Quan\_15 + Quan\_16, data=filtered\_quan\_train)  
summary(model4)

##   
## Call:  
## lm(formula = Date\_2 ~ Quan\_1 + Quan\_2 + Quan\_3 + Quan\_4 + Quan\_5 +   
## Quan\_6 + Quan\_7 + Quan\_8 + Quan\_9 + Quan\_10 + Quan\_11 + Quan\_12 +   
## Quan\_13 + Quan\_14 + Quan\_15 + Quan\_16, data = filtered\_quan\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1367.94 -193.08 10.38 190.21 878.53   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.693e+02 8.511e+02 0.786 0.4349   
## Quan\_1 4.211e-04 6.037e-05 6.974 3.24e-09 \*\*\*  
## Quan\_2 1.680e-01 1.109e-01 1.515 0.1353   
## Quan\_3 -9.666e+00 1.966e+00 -4.917 7.58e-06 \*\*\*  
## Quan\_4 3.267e-05 9.234e-05 0.354 0.7248   
## Quan\_5 -7.531e+00 2.303e+01 -0.327 0.7449   
## Quan\_6 8.646e+00 4.343e+01 0.199 0.8429   
## Quan\_7 1.088e+01 2.739e+01 0.397 0.6927   
## Quan\_8 -1.721e+00 2.521e+00 -0.683 0.4974   
## Quan\_9 3.611e-01 3.677e+01 0.010 0.9922   
## Quan\_10 1.647e+01 5.846e+01 0.282 0.7791   
## Quan\_11 -4.506e+00 3.743e+01 -0.120 0.9046   
## Quan\_12 -8.218e-06 5.706e-05 -0.144 0.8860   
## Quan\_13 1.018e-05 5.618e-05 0.181 0.8568   
## Quan\_14 5.754e+00 8.030e+00 0.717 0.4765   
## Quan\_15 7.992e+00 3.782e+00 2.113 0.0389 \*   
## Quan\_16 -7.882e+00 3.741e+00 -2.107 0.0395 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 459.2 on 58 degrees of freedom  
## Multiple R-squared: 0.6548, Adjusted R-squared: 0.5596   
## F-statistic: 6.876 on 16 and 58 DF, p-value: 1.969e-08

* Model4 was created to see the linear relationship between Date\_2 (number of days advertising campaign began and product launched) and Quantitative variables.

predictTest\_quan\_date2 = predict(model4,newdata = filtered\_quan\_test)  
predictTest\_quan\_date2

## 520 538 542 545 549 562 563   
## 2474.5697 865.9846 2812.5432 1726.7514 589.9290 2098.5158 2425.2370   
## 573 578 587 591 601 603 609   
## 1759.4170 1354.8766 3028.2642 1134.3565 2392.9603 1813.6512 2012.0384   
## 619 620 630 641 643 648 650   
## 1299.7003 2838.3141 1167.1294 1561.8444 2242.1522 1832.0198 3721.9068   
## 654 655 659 661 664 666 673   
## 1807.8089 1264.5305 2427.1089 2355.4316 2340.0379 1683.2707 1745.4201   
## 689 690 698 705 708 709 710   
## 2280.3747 2631.2965 1741.8549 1728.0215 1632.1574 2837.7626 2513.2397   
## 711 719 722 729 734 735 739   
## 1658.0033 2796.5411 2990.2233 1706.8771 2194.9078 1980.4306 3212.3632

SSE\_quan\_date2 = sum((filtered\_quan\_test$Date\_2 - predictTest\_quan\_date2)^2)  
SST\_quan\_date2 = sum((filtered\_quan\_df$Date\_2 - mean(filtered\_quan\_train$Date\_2))^2)  
1 -SSE\_quan\_date2/SST\_quan\_date2

## [1] 0.8942442

RMSE\_quan\_date2 = sqrt(SSE\_quan\_date2/nrow(filtered\_quan\_test))  
RMSE\_quan\_date2

## [1] 391.8497

* The model was predicted and shows that the R squared and RMSE are very strong fit for the dataset.

**Categorical Variable**

filtered\_cat <-OnlineProductSale[c(4,32,36,41,43,44,57,58,59,83,  
 85,89,91,94,101,109,131, 137, 142, 146,  
 157,164,165,176,179, 182, 197, 206, 216, 218,   
 220, 222,231, 238, 252, 260, 276, 294, 295, 298,   
 307, 313, 320, 322, 324, 325, 340, 341, 344, 348,   
 353, 355, 356, 357, 358, 361,391, 395, 404, 411,   
 417, 419, 440, 442, 460, 462, 465, 477, 479, 485,   
 493, 496, 503, 513, 518, 520, 538, 542, 545, 549,   
 562, 563, 573,578, 587, 591, 601, 603, 609, 619,   
 620, 630, 641, 643, 648, 650, 654, 655, 659, 661,   
 664, 666, 673, 689, 690, 698, 705, 708, 709, 710,  
 711, 719, 722, 729, 734, 735, 739),c(32:51)]  
filtered\_cat

## Cat\_1 Cat\_2 Cat\_3 Cat\_4 Cat\_5 Cat\_6 Cat\_7 Cat\_8 Cat\_9 Cat\_10 Cat\_11  
## 4 1 0 1 0 0 1 1 1 0 0 0  
## 32 1 0 1 1 0 0 1 1 0 0 1  
## 36 1 0 1 1 0 0 1 1 0 0 1  
## 41 1 1 1 1 0 1 1 1 0 0 0  
## 43 1 0 1 1 0 0 1 1 0 0 1  
## 44 1 0 1 1 0 0 1 1 1 0 1  
## 57 1 0 1 1 0 0 1 1 0 0 1  
## 58 1 0 1 1 0 0 1 1 0 0 1  
## 59 1 0 1 1 0 0 1 1 1 0 0  
## 83 1 0 1 1 0 0 1 1 0 0 1  
## 85 1 0 1 1 0 1 1 1 0 0 1  
## 89 1 0 1 1 0 1 1 1 0 0 1  
## 91 1 0 1 1 0 0 1 1 0 0 1  
## 94 1 0 1 1 1 0 1 1 1 0 0  
## 101 1 0 1 1 0 0 1 1 0 0 1  
## 109 1 0 1 1 0 1 1 1 0 0 0  
## 131 1 0 1 1 0 0 1 1 0 0 1  
## 137 1 1 1 1 1 0 1 1 0 0 1  
## 142 1 0 1 1 0 1 1 1 1 0 0  
## 146 1 0 1 1 0 1 1 1 0 0 0  
## 157 1 0 1 1 0 0 1 1 0 0 1  
## 164 1 0 1 1 0 1 1 1 0 0 0  
## 165 1 0 1 1 0 0 1 1 0 0 1  
## 176 1 1 1 1 1 0 1 1 0 0 1  
## 179 1 0 1 1 0 1 1 1 1 0 0  
## 182 1 0 1 1 0 0 1 1 0 0 1  
## 197 1 0 1 1 0 0 1 1 0 0 0  
## 206 1 0 1 1 0 0 1 1 0 0 0  
## 216 1 1 1 1 0 1 1 1 0 0 0  
## 218 1 0 1 1 0 0 1 1 0 0 1  
## 220 1 0 1 1 0 1 1 1 1 0 0  
## 222 1 0 1 1 0 1 1 1 0 0 0  
## 231 1 0 1 1 0 0 1 1 0 0 1  
## 238 1 0 1 1 0 0 1 1 0 0 1  
## 252 1 1 1 1 1 0 1 1 0 0 0  
## 260 1 0 1 1 0 0 1 1 0 0 1  
## 276 1 0 1 1 0 1 1 1 0 0 1  
## 294 1 0 1 1 0 0 1 1 1 0 0  
## 295 1 0 1 1 0 1 1 1 1 0 0  
## 298 1 0 1 1 0 0 1 1 0 0 0  
## 307 1 1 1 1 1 1 1 1 1 0 0  
## 313 1 0 1 1 0 1 1 1 0 0 0  
## 320 1 0 1 1 0 1 1 1 0 0 0  
## 322 1 0 1 1 0 0 1 1 0 0 1  
## 324 1 0 1 1 0 0 1 1 0 0 1  
## 325 1 0 1 1 0 1 1 1 1 0 0  
## 340 1 0 1 1 0 1 1 1 1 0 0  
## 341 1 1 1 1 0 1 1 1 0 0 0  
## 344 1 1 1 1 1 1 1 1 0 0 1  
## 348 1 0 1 1 0 0 1 1 0 0 1  
## 353 1 0 1 1 0 1 1 1 1 0 1  
## 355 1 0 1 1 0 0 1 1 0 0 0  
## 356 1 0 1 1 0 1 1 1 0 0 1  
## 357 1 0 1 1 1 1 1 1 0 0 0  
## 358 1 0 1 1 0 1 1 1 0 0 0  
## 361 1 0 1 1 1 1 1 1 1 0 0  
## 391 1 0 1 1 0 1 1 1 0 0 0  
## 395 2 0 1 1 0 1 1 1 0 0 0  
## 404 1 0 1 1 0 1 1 1 1 0 1  
## 411 1 0 1 1 0 0 1 1 0 0 0  
## 417 1 0 1 0 0 1 1 1 0 0 1  
## 419 1 0 1 1 0 1 1 1 0 0 0  
## 440 1 0 1 0 0 0 1 1 1 0 1  
## 442 1 0 1 1 0 0 1 1 0 0 1  
## 460 1 0 1 1 0 0 1 1 0 0 1  
## 462 1 0 1 1 0 0 1 1 0 0 1  
## 465 1 0 1 1 0 1 1 1 0 0 0  
## 477 1 1 1 1 1 1 1 1 0 0 1  
## 479 1 0 1 1 1 0 1 1 0 0 1  
## 485 1 0 1 1 0 0 1 1 1 0 0  
## 493 1 0 1 1 0 0 1 1 0 0 1  
## 496 1 0 1 1 0 0 1 1 0 0 0  
## 503 1 0 1 1 0 1 1 1 1 0 1  
## 513 1 0 1 1 0 1 1 1 0 0 0  
## 518 1 0 1 1 0 0 1 1 0 0 1  
## 520 1 0 1 1 1 1 1 1 0 0 1  
## 538 1 0 1 1 0 0 1 1 0 0 0  
## 542 1 1 1 1 1 0 1 1 0 0 0  
## 545 1 1 1 1 1 0 1 1 0 0 0  
## 549 1 1 1 1 1 1 1 1 1 0 1  
## 562 1 0 1 1 1 1 1 1 1 0 0  
## 563 1 0 1 1 0 1 1 1 0 0 1  
## 573 1 0 1 1 0 0 1 1 1 0 1  
## 578 1 1 1 1 1 0 1 1 0 0 1  
## 587 1 0 1 1 0 0 1 1 0 0 1  
## 591 1 0 1 0 0 0 1 1 0 0 1  
## 601 1 0 1 1 0 0 1 1 0 0 0  
## 603 1 0 1 1 0 1 1 1 1 0 1  
## 609 1 0 1 1 0 0 1 1 0 0 1  
## 619 1 0 1 0 0 1 1 1 0 0 0  
## 620 1 0 1 1 0 1 1 1 0 0 0  
## 630 1 0 1 1 0 0 1 1 0 0 1  
## 641 1 0 1 1 0 0 1 1 0 0 1  
## 643 1 0 1 1 0 0 1 1 0 0 1  
## 648 1 0 1 1 0 0 1 1 0 0 0  
## 650 1 0 1 1 0 1 1 1 0 0 1  
## 654 1 0 1 1 0 1 1 1 1 0 0  
## 655 1 0 1 1 0 0 1 1 0 0 1  
## 659 1 0 1 1 0 1 1 1 0 0 1  
## 661 1 0 1 1 0 0 1 1 1 0 0  
## 664 1 0 1 1 0 0 1 1 0 0 1  
## 666 1 0 1 1 0 1 1 1 1 0 0  
## 673 1 0 1 1 1 0 1 1 0 0 1  
## 689 1 0 1 1 0 1 1 1 0 0 1  
## 690 1 0 1 1 0 0 1 1 0 0 1  
## 698 1 0 1 1 1 1 1 1 1 0 0  
## 705 1 0 1 1 0 0 1 1 0 0 0  
## 708 1 1 1 1 1 0 1 1 1 0 0  
## 709 1 1 1 1 1 1 1 1 0 0 0  
## 710 1 0 1 1 0 0 1 1 0 0 1  
## 711 1 0 1 1 0 0 1 1 1 0 0  
## 719 1 0 1 0 0 0 1 1 0 0 1  
## 722 1 0 1 1 0 0 1 1 0 0 1  
## 729 1 0 1 1 1 1 1 1 0 0 1  
## 734 1 0 1 1 0 0 1 1 1 0 1  
## 735 1 0 1 1 0 0 1 1 0 0 1  
## 739 1 0 1 1 0 1 1 1 0 0 0  
## Cat\_12 Cat\_13 Cat\_14 Cat\_15 Cat\_16 Cat\_17 Cat\_18 Cat\_19 Cat\_20  
## 4 1 1 1 0 0 0 0 0 0  
## 32 0 1 0 0 0 0 0 1 0  
## 36 0 1 0 0 0 0 0 1 0  
## 41 1 1 1 0 0 1 1 1 0  
## 43 0 1 0 0 0 0 0 1 0  
## 44 0 1 0 0 0 0 0 1 1  
## 57 0 1 0 0 0 0 1 1 0  
## 58 0 1 0 0 0 0 0 1 0  
## 59 0 1 0 0 0 0 1 1 0  
## 83 0 1 0 0 0 1 1 1 0  
## 85 1 1 1 0 0 0 1 1 0  
## 89 1 1 1 0 0 0 0 1 0  
## 91 0 1 0 0 0 0 0 1 0  
## 94 0 1 0 0 0 0 0 1 0  
## 101 0 1 0 0 0 0 0 1 0  
## 109 1 1 1 1 0 0 0 1 0  
## 131 0 1 0 0 0 0 0 1 0  
## 137 0 1 0 0 0 1 1 1 1  
## 142 1 1 1 0 0 1 1 1 0  
## 146 1 1 1 0 0 1 1 1 0  
## 157 0 1 0 1 0 0 1 1 0  
## 164 1 1 1 0 0 1 1 1 0  
## 165 0 1 0 0 0 0 0 1 0  
## 176 0 1 0 1 0 1 0 1 1  
## 179 1 1 1 0 0 1 1 1 0  
## 182 0 1 0 0 0 0 0 1 0  
## 197 0 1 0 0 0 0 0 1 0  
## 206 0 1 0 1 0 1 1 1 0  
## 216 1 1 1 0 0 0 1 1 0  
## 218 0 1 0 0 0 0 1 1 0  
## 220 1 1 1 0 0 1 1 1 0  
## 222 1 1 1 0 0 0 0 1 1  
## 231 0 1 0 0 0 0 0 1 0  
## 238 0 1 0 0 0 0 0 1 0  
## 252 0 1 0 0 0 1 1 1 1  
## 260 0 1 0 0 0 0 0 1 0  
## 276 1 1 1 0 0 0 0 1 0  
## 294 0 1 0 0 0 1 1 1 0  
## 295 1 1 1 1 0 1 1 1 0  
## 298 0 1 0 0 0 1 1 0 0  
## 307 1 1 1 0 0 1 1 1 1  
## 313 1 1 1 1 0 1 1 1 0  
## 320 1 1 1 1 0 1 1 1 1  
## 322 0 1 0 0 0 0 0 1 0  
## 324 0 1 0 0 0 0 0 1 0  
## 325 1 1 1 0 0 1 1 1 1  
## 340 1 1 1 0 0 1 1 1 0  
## 341 1 1 1 0 0 1 1 1 1  
## 344 1 1 1 1 0 1 1 1 1  
## 348 0 1 0 1 0 1 1 1 0  
## 353 1 1 1 0 0 0 1 1 0  
## 355 0 1 0 0 0 0 0 0 1  
## 356 1 1 1 0 0 0 0 1 0  
## 357 1 1 1 0 0 1 1 1 1  
## 358 1 1 1 0 0 0 1 1 0  
## 361 1 1 1 0 0 1 1 1 1  
## 391 1 1 1 1 0 1 1 1 1  
## 395 1 1 1 0 0 0 1 1 1  
## 404 1 1 1 0 0 0 1 1 1  
## 411 0 1 0 0 0 0 0 1 1  
## 417 1 1 1 0 0 0 0 0 0  
## 419 1 1 1 0 0 1 1 1 0  
## 440 0 1 0 0 0 1 1 0 0  
## 442 0 1 0 1 0 0 1 1 0  
## 460 0 1 0 0 0 0 0 1 0  
## 462 0 1 0 0 0 0 0 1 0  
## 465 1 1 1 0 0 1 1 1 0  
## 477 1 1 1 0 0 1 1 1 1  
## 479 0 1 0 0 0 1 1 1 1  
## 485 0 1 0 1 0 1 1 1 0  
## 493 0 1 0 0 0 0 0 1 0  
## 496 0 1 0 0 0 0 0 1 1  
## 503 1 1 0 1 0 1 1 1 1  
## 513 1 1 1 1 0 1 1 1 0  
## 518 0 1 0 0 0 0 0 1 0  
## 520 1 1 1 0 0 1 1 1 1  
## 538 0 1 0 0 0 0 0 1 0  
## 542 0 1 0 0 0 0 1 1 1  
## 545 0 1 0 0 0 0 0 1 1  
## 549 1 1 1 1 0 1 1 1 1  
## 562 1 1 0 0 0 1 1 1 0  
## 563 1 1 1 0 0 0 0 1 0  
## 573 0 1 0 0 0 0 0 1 0  
## 578 0 1 0 1 0 1 1 1 1  
## 587 0 1 0 0 0 0 1 1 0  
## 591 0 1 0 0 0 0 0 0 0  
## 601 0 1 0 1 0 0 1 1 1  
## 603 1 1 1 1 0 0 0 1 0  
## 609 0 1 0 0 0 0 0 1 0  
## 619 1 1 1 0 0 0 0 0 0  
## 620 1 1 1 1 0 1 1 1 0  
## 630 0 1 0 0 0 0 0 1 0  
## 641 0 1 0 0 0 0 0 1 1  
## 643 0 1 0 0 0 0 0 1 0  
## 648 0 1 0 0 0 0 0 1 0  
## 650 1 1 1 0 0 0 0 1 0  
## 654 1 1 1 1 0 1 1 1 1  
## 655 0 1 0 0 0 0 0 1 1  
## 659 1 1 1 0 0 1 1 1 0  
## 661 0 1 0 0 0 0 1 1 0  
## 664 0 1 0 0 0 0 0 1 0  
## 666 1 1 1 1 0 1 1 1 1  
## 673 0 1 0 0 0 1 1 1 1  
## 689 0 1 0 0 0 0 0 1 1  
## 690 0 1 0 0 0 0 0 1 0  
## 698 1 1 1 0 0 1 1 1 1  
## 705 0 1 0 0 0 0 1 1 1  
## 708 0 1 0 0 0 1 1 1 1  
## 709 1 1 1 0 0 1 1 1 1  
## 710 0 1 0 0 0 0 1 1 0  
## 711 0 1 0 0 0 1 1 1 1  
## 719 0 1 0 0 0 0 0 1 1  
## 722 0 1 0 0 0 0 0 1 0  
## 729 1 1 1 1 0 0 0 1 1  
## 734 0 1 0 0 0 0 0 1 1  
## 735 0 1 0 0 0 0 0 1 0  
## 739 1 1 0 0 0 0 0 1 1

head(filtered\_cat)

## Cat\_1 Cat\_2 Cat\_3 Cat\_4 Cat\_5 Cat\_6 Cat\_7 Cat\_8 Cat\_9 Cat\_10 Cat\_11  
## 4 1 0 1 0 0 1 1 1 0 0 0  
## 32 1 0 1 1 0 0 1 1 0 0 1  
## 36 1 0 1 1 0 0 1 1 0 0 1  
## 41 1 1 1 1 0 1 1 1 0 0 0  
## 43 1 0 1 1 0 0 1 1 0 0 1  
## 44 1 0 1 1 0 0 1 1 1 0 1  
## Cat\_12 Cat\_13 Cat\_14 Cat\_15 Cat\_16 Cat\_17 Cat\_18 Cat\_19 Cat\_20  
## 4 1 1 1 0 0 0 0 0 0  
## 32 0 1 0 0 0 0 0 1 0  
## 36 0 1 0 0 0 0 0 1 0  
## 41 1 1 1 0 0 1 1 1 0  
## 43 0 1 0 0 0 0 0 1 0  
## 44 0 1 0 0 0 0 0 1 1

* Categorical variables were not deeply focused during this project due to number of 0’s in the data set.
* Binary categorical variables are measured as (1) if the product had the feature and (0) if it did not.
* Most of the values for above products were nearly 0 which tells no features.

**Conclusion:**

The main question of this project was to see what products should be recommended to the customers based on sales outcome. The limitation of this dataset was unable to predict the outcome with testset. The actual testset file does not include the outcome variables. Outcome variables were only predicted using values from testset. The challenges to this project was how 751 different products can be filtered by their sales outcome. We can conclude that low-outcome products should be bundled with other products to increase the sales margin. The advertising of products does not have effect on the sales outcomes. Although there were many challenges finding correlation between variables, Quantitative variables which had strong relationship with Date\_1 and Date\_2. Therefore, quantitative variables are the better predictors than outcomes of each month for the future predictions.

**Future Research & Recommendations:**

* Customers should look for quantitative variables when shopping online.
* Customers also lookout for what bundled products are available when shopping.
* High outcome products should be recommended to the customers compared to the low outcome products.
* Outcome from Test dataset should be given in order to predict actual sales outcome.