Quantium Virtual Internship- Retail Strategy and Analytics- Task 1

2025-11-28

```
#### Load required libraries
library(data.table)

## Warning: package 'data.table' was built under R version 4.4.3

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.4.2

library(readr)

## Warning: package 'readr' was built under R version 4.4.3

#### Load the Dataset
transactionData = read.csv("QVI_transaction_data.csv")
customerData = read.csv("QVI_purchase_behaviour.csv")

#### Examine transaction data
str(transactionData)

## 'data.frame':    264836 obs. of  8 variables:
##  $ DATE          : int  43390 43599 43605 43329 43330 43604 43601 43601
43332 43330 ...
##  $ STORE_NBR     : int  1 1 1 2 2 4 4 4 5 7 ...
##  $ LYLTY_CARD_NBR: int  1000 1307 1343 2373 2426 4074 4149 4196 5026 7150
...
##  $ TXN_ID        : int  1 348 383 974 1038 2982 3333 3539 4525 6900 ...
##  $ PROD_NBR      : int  5 66 61 69 108 57 16 24 42 52 ...
##  $ PROD_NAME     : chr  "Natural Chip        Compny SeaSalt175g" "CCs
Nacho Cheese    175g" "Smiths Crinkle Cut  Chips Chicken 170g" "Smiths Chip
Thinly  S/Cream&Onion 175g" ...
##  $ PROD_QTY      : int  2 3 2 5 3 1 1 1 1 2 ...
##  $ TOT_SALES     : num  6 6.3 2.9 15 13.8 5.1 5.7 3.6 3.9 7.2 ...

#### Convert DATE column to a date format
transactionData<-as.data.table(transactionData)
transactionData[,DATA:=as.Date(DATE,origin="1899-12-30")]

#### Examine PROD_NAME
transactionData[, .N, PROD_NAME]

##                                       PROD_NAME     N
##                                          <char> <int>
##    1:   Natural Chip        Compny SeaSalt175g  1468
##    2:                  CCs Nacho Cheese    175g  1498
##   3:   Smiths Crinkle Cut  Chips Chicken 170g  1484
##   4:   Smiths Chip Thinly  S/Cream&Onion 175g  1473
##   5: Kettle Tortilla ChpsHny&Jlpno Chili 150g  3296
##  ---
## 110:    Red Rock Deli Chikn&Garlic Aioli 150g  1434
## 111:      RRD SR Slow Rst     Pork Belly 150g  1526
## 112:                 RRD Pc Sea Salt     165g  1431
## 113:       Smith Crinkle Cut   Bolognese 150g  1451
## 114:                 Doritos Salsa Mild  300g  1472
```

```
productWords<- data.table(unlist(strsplit(unique(transactionData[,
PROD_NAME]), " ")))
setnames(productWords, 'words')

productWords <- productWords[grepl("\\d", words) == FALSE, ]
productWords <- productWords[grepl("[:alpha:]", words), ]
productWords[, .N, words][order(N, decreasing = TRUE)]
```

```
##             words     N
##            <char> <int>
##   1:        Chips    21
##   2:       Smiths    16
##   3:      Crinkle    14
##   4:       Kettle    13
##   5:       Cheese    12
##  ---
## 127: Chikn&Garlic     1
## 128:        Aioli     1
## 129:         Slow     1
## 130:        Belly     1
## 131:    Bolognese     1
```

```
summary(transactionData)
```

```
##       DATE         STORE_NBR     LYLTY_CARD_NBR        TXN_ID
##  Min.   :43282   Min.   :  1.0   Min.   :   1000   Min.   :      1
##  1st Qu.:43373   1st Qu.: 70.0   1st Qu.:  70021   1st Qu.:  67602
##  Median :43464   Median :130.0   Median : 130358   Median : 135138
##  Mean   :43464   Mean   :135.1   Mean   : 135550   Mean   : 135158
##  3rd Qu.:43555   3rd Qu.:203.0   3rd Qu.: 203094   3rd Qu.: 202701
##  Max.   :43646   Max.   :272.0   Max.   :2373711   Max.   :2415841
##     PROD_NBR       PROD_NAME          PROD_QTY         TOT_SALES
##  Min.   :  1.00   Length:264836      Min.   :  1.000   Min.   :  1.500
##  1st Qu.: 28.00   Class :character   1st Qu.:  2.000   1st Qu.:  5.400
##  Median : 56.00   Mode  :character   Median :  2.000   Median :  7.400
##  Mean   : 56.58                      Mean   :  1.907   Mean   :  7.304
##  3rd Qu.: 85.00                      3rd Qu.:  2.000   3rd Qu.:  9.200
##  Max.   :114.00                      Max.   :200.000   Max.   :650.000
##       DATA
##  Min.   :2018-07-01
##  1st Qu.:2018-09-30
##  Median :2018-12-30
##  Mean   :2018-12-30
##  3rd Qu.:2019-03-31
##  Max.   :2019-06-30
```

```
transactionData[PROD_QTY == 200, ]
```

```
##     DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
##    <int>     <int>          <int>  <int>    <int>
## 1: 43331       226         226000 226201        4
## 2: 43605       226         226000 226210        4
##                           PROD_NAME PROD_QTY TOT_SALES       DATA
##                              <char>    <int>     <num>     <Date>
## 1: Dorito Corn Chp     Supreme 380g      200       650 2018-08-19
## 2: Dorito Corn Chp     Supreme 380g      200       650 2019-05-20
```

```
transactionData[LYLTY_CARD_NBR == 226000, ]
```

```
##     DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
##    <int>     <int>          <int>  <int>    <int>
## 1: 43331       226         226000 226201        4
## 2: 43605       226         226000 226210        4
##                           PROD_NAME PROD_QTY TOT_SALES       DATA
##                              <char>    <int>     <num>     <Date>
## 1: Dorito Corn Chp     Supreme 380g      200       650 2018-08-19
## 2: Dorito Corn Chp     Supreme 380g      200       650 2019-05-20
```

```
transactionData <- transactionData[LYLTY_CARD_NBR != 226000, ]
summary(transactionData)
```

```
##       DATE          STORE_NBR     LYLTY_CARD_NBR        TXN_ID
##  Min.   :43282   Min.   :  1.0   Min.   :   1000   Min.   :      1
##  1st Qu.:43373   1st Qu.: 70.0   1st Qu.:  70021   1st Qu.:  67601
##  Median :43464   Median :130.0   Median : 130357   Median : 135137
##  Mean   :43464   Mean   :135.1   Mean   : 135549   Mean   : 135158
##  3rd Qu.:43555   3rd Qu.:203.0   3rd Qu.: 203094   3rd Qu.: 202700
##  Max.   :43646   Max.   :272.0   Max.   :2373711   Max.   :2415841
##     PROD_NBR       PROD_NAME            PROD_QTY       TOT_SALES
##  Min.   :  1.00   Length:264834      Min.   :1.000   Min.   : 1.500
##  1st Qu.: 28.00   Class :character   1st Qu.:2.000   1st Qu.: 5.400
##  Median : 56.00   Mode  :character   Median :2.000   Median : 7.400
##  Mean   : 56.58                      Mean   :1.906   Mean   : 7.299
##  3rd Qu.: 85.00                      3rd Qu.:2.000   3rd Qu.: 9.200
##  Max.   :114.00                      Max.   :5.000   Max.   :29.500
##       DATA
##  Min.   :2018-07-01
##  1st Qu.:2018-09-30
##  Median :2018-12-30
##  Mean   :2018-12-30
##  3rd Qu.:2019-03-31
##  Max.   :2019-06-30
```

```r
transactionData[, .N, by = DATE]
```

```
##        DATE     N
##       <int> <int>
##   1: 43390   732
##   2: 43599   758
##   3: 43605   754
##   4: 43329   711
##   5: 43330   737
##  ---
## 360: 43425   700
## 361: 43595   710
## 362: 43442   672
## 363: 43495   738
## 364: 43505   718
```

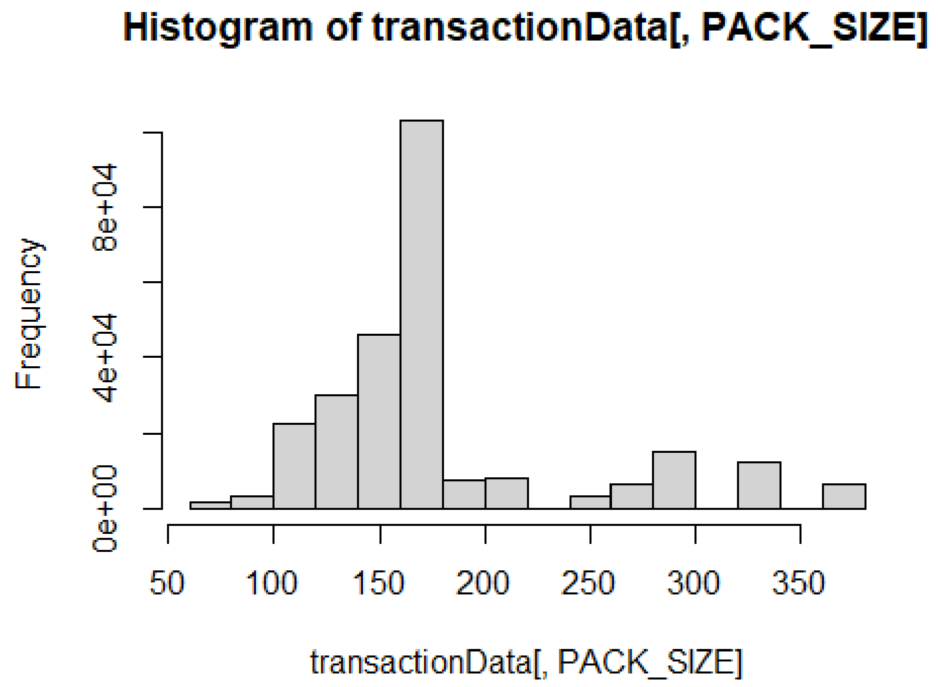
```r
transactionData[,PACK_SIZE:= parse_number(PROD_NAME)]
transactionData[,.N,PACK_SIZE][order(PACK_SIZE)]
```

```
##     PACK_SIZE     N
##         <num> <int>
##  1:        70  1507
##  2:        90  3008
##  3:       110 22387
##  4:       125  1454
##  5:       134 25102
##  6:       135  3257
##  7:       150 43131
##  8:       160  2970
##  9:       165 15297
## 10:       170 19983
## 11:       175 66390
## 12:       180  1468
## 13:       190  2995
## 14:       200  4473
## 15:       210  6272
## 16:       220  1564
## 17:       250  3169
## 18:       270  6285
## 19:       300 15166
## 20:       330 12540
## 21:       380  6416
##     PACK_SIZE     N
```

```r
transactionData
```

```
##           DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
##          <int>     <int>          <int>  <int>    <int>
##      1: 43390         1           1000      1        5
##      2: 43599         1           1307    348       66
##      3: 43605         1           1343    383       61
##      4: 43329         2           2373    974       69
##      5: 43330         2           2426   1038      108
##     ---
## 264830: 43533       272         272319 270088       89
## 264831: 43325       272         272358 270154       74
## 264832: 43410       272         272379 270187       51
## 264833: 43461       272         272379 270188       42
## 264834: 43365       272         272380 270189       74
##                                         PROD_NAME PROD_QTY TOT_SALES
DATA
##                                            <char>    <int>     <num>
<Date>
##      1:   Natural Chip        Compny SeaSalt175g        2       6.0 2018-
10-17
##      2:                 CCs Nacho Cheese    175g        3       6.3 2019-
05-14
##      3:   Smiths Crinkle Cut  Chips Chicken 170g        2       2.9 2019-
05-20
##      4:   Smiths Chip Thinly  S/Cream&Onion 175g        5      15.0 2018-
08-17
##      5: Kettle Tortilla ChpsHny&Jlpno Chili 150g        3      13.8 2018-
08-18
##     ---
## 264830:  Kettle Sweet Chilli And Sour Cream 175g        2      10.8 2019-
03-09
## 264831:            Tostitos Splash Of  Lime 175g        1       4.4 2018-
08-13
## 264832:                 Doritos Mexicana    170g        2       8.8 2018-
11-06
## 264833:  Doritos Corn Chip Mexican Jalapeno 150g        2       7.8 2018-
12-27
## 264834:            Tostitos Splash Of  Lime 175g        2       8.8 2018-
09-22
##         PACK_SIZE
##             <num>
##      1:       175
##      2:       175
##      3:       170
##      4:       175
##      5:       150
##     ---
## 264830:       175
## 264831:       175
## 264832:       170
## 264833:       150
## 264834:       175
```

```
hist(transactionData[,PACK_SIZE])
```



```
 transactionData[,BRAND:= toupper(substr(PROD_NAME,1, regexpr(pattern='',
 PROD_NAME)-1))]
 transactionData[,.N,by=BRAND][order(-N)]

##     BRAND      N
##    <char>  <int>
## 1:        264834

 transactionData[BRAND == "RED", BRAND := "RRD"]
 transactionData[BRAND == "SNBTS", BRAND := "SUNBITES"]
 transactionData[BRAND == "INFZNS", BRAND := "INFUZIONS"]
 transactionData[BRAND == "WW", BRAND := "WOOLWORTHS"]
 transactionData[BRAND == "SMITH", BRAND := "SMITHS"]
 transactionData[BRAND == "NCC", BRAND := "NATURAL"]
 transactionData[BRAND == "DORITO", BRAND := "DORITOS"]
 transactionData[BRAND == "GRAIN", BRAND := "GRNWVES"]

 transactionData[, .N, by = BRAND][order(BRAND)]

##     BRAND      N
##    <char>  <int>
## 1:        264834

 str(customerData)

## 'data.frame':    72637 obs. of  3 variables:
##  $ LYLTY_CARD_NBR  : int  1000 1002 1003 1004 1005 1007 1009 1010 1011
1012 ...
```

```
##  $ LIFESTAGE       : chr  "YOUNG SINGLES/COUPLES" "YOUNG SINGLES/COUPLES"
"YOUNG FAMILIES" "OLDER SINGLES/COUPLES" ...
##  $ PREMIUM_CUSTOMER: chr  "Premium" "Mainstream" "Budget" "Mainstream" ...
```

```
 summary(customerData)
```

```
##  LYLTY_CARD_NBR     LIFESTAGE          PREMIUM_CUSTOMER
##  Min.   :   1000   Length:72637       Length:72637
##  1st Qu.:  66202   Class :character   Class :character
##  Median : 134040   Mode  :character   Mode  :character
##  Mean   : 136186
##  3rd Qu.: 203375
##  Max.   :2373711
```

```
customerData<-as.data.table(customerData)
customerData[,.N, by = LIFESTAGE][order(-N)]
```

```
##                 LIFESTAGE     N
##                    <char> <int>
## 1:               RETIREES 14805
## 2:  OLDER SINGLES/COUPLES 14609
## 3:  YOUNG SINGLES/COUPLES 14441
## 4:         OLDER FAMILIES  9780
## 5:         YOUNG FAMILIES  9178
## 6: MIDAGE SINGLES/COUPLES  7275
## 7:           NEW FAMILIES  2549
```

```
customerData[, .N, by = PREMIUM_CUSTOMER][order(-N)]
```

```
##    PREMIUM_CUSTOMER     N
##              <char> <int>
## 1:       Mainstream 29245
## 2:           Budget 24470
## 3:          Premium 18922
```

```
 data <- merge(transactionData, customerData, all.x = TRUE)
```

```
data[is.null(LIFESTAGE), .N]
```

```
## [1] 0
```

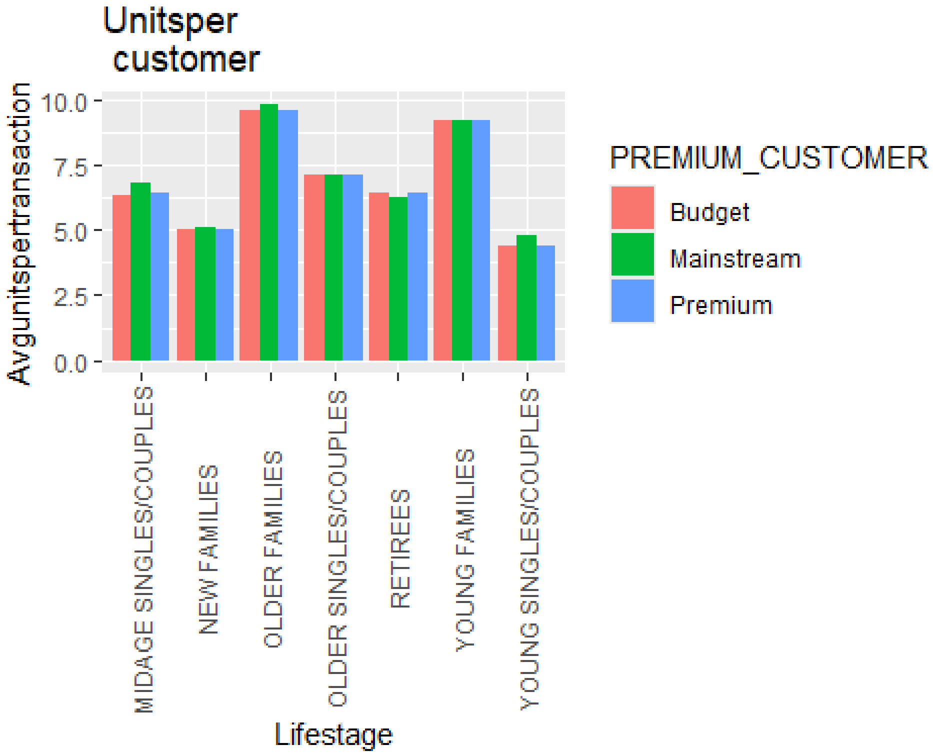
```
 data[is.null(PREMIUM_CUSTOMER), .N]
```

```
## [1] 0
```

```
fwrite(data,"QVI_data.csv")
```

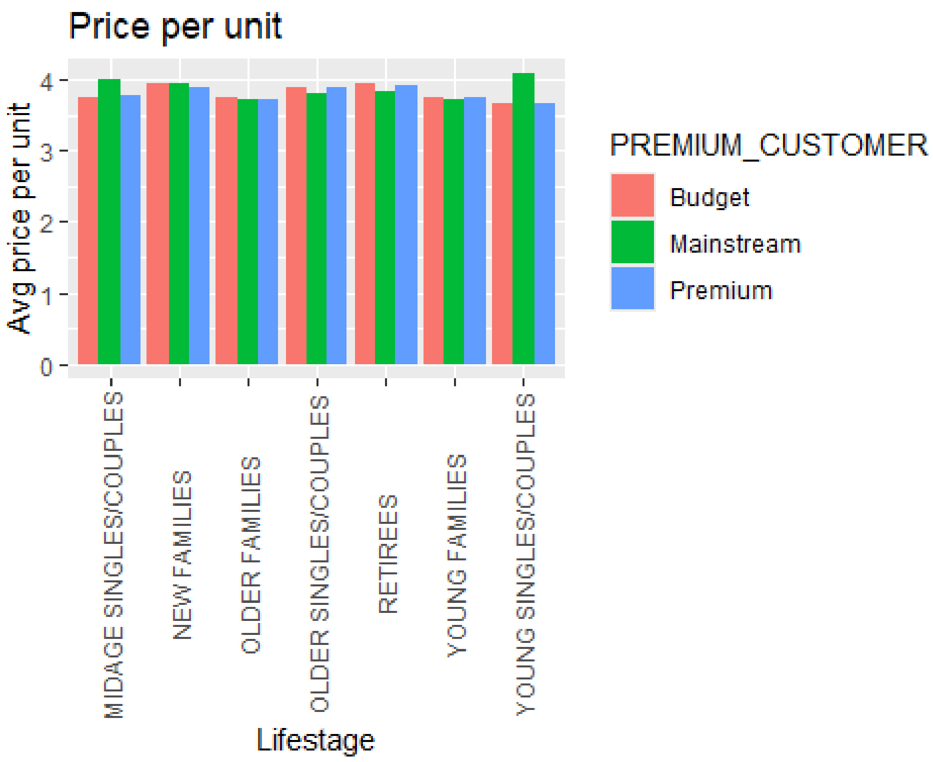
```
 customers<-data[,.(CUSTOMERS= uniqueN(LYLTY_CARD_NBR)),.(LIFESTAGE,
 PREMIUM_CUSTOMER)][order(-CUSTOMERS)]
```

```
 avg_units<-data[,.(AVG= sum(PROD_QTY)/uniqueN(LYLTY_CARD_NBR)),
 .(LIFESTAGE,PREMIUM_CUSTOMER)][order(-AVG)]
 ggplot(data=avg_units, aes(weight=AVG,x= LIFESTAGE,fill=
 PREMIUM_CUSTOMER))+
 geom_bar(position= position_dodge()) +
 labs(x= "Lifestage",y= "Avgunitspertransaction",title="Unitsper
 customer") +
 theme(axis.text.x= element_text(angle=90,vjust=0.5))
```



```
avg_price <- data[, .(AVG = sum(TOT_SALES)/sum(PROD_QTY)), .(LIFESTAGE,
PREMIUM_CUSTOMER)][order(-AVG)]

ggplot(data = avg_price, aes(weight = AVG, x = LIFESTAGE, fill =
PREMIUM_CUSTOMER)) +
geom_bar(position = position_dodge()) +
labs(x = "Lifestage", y = "Avg price per unit", title = "Price per unit") +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



```
 pricePerUnit <- data[, price := TOT_SALES/PROD_QTY]
 t.test(data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE
SINGLES/COUPLES")
 & PREMIUM_CUSTOMER == "Mainstream", price]

 , data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES")
 & PREMIUM_CUSTOMER != "Mainstream", price]
 , alternative = "greater")

##
##  Welch Two Sample t-test
##
## data:  data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE
SINGLES/COUPLES") & PREMIUM_CUSTOMER == "Mainstream", price] and
data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") &
PREMIUM_CUSTOMER != "Mainstream", price]
## t = 40.61, df = 58792, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.3429435       Inf
## sample estimates:
## mean of x mean of y
##  4.045586  3.688165

 segment1 <- data[LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM_CUSTOMER ==
 "Mainstream",]
 other <- data[!(LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM_CUSTOMER ==
 "Mainstream"),]
 quantity_segment1 <- segment1[, sum(PROD_QTY)]

quantity_other <- other[, sum(PROD_QTY)]
 quantity_segment1_by_brand <- segment1[, .(targetSegment =
 sum(PROD_QTY)/quantity_segment1), by = BRAND]

 quantity_other_by_brand <- other[, .(other = sum(PROD_QTY)/quantity_other),
by
 = BRAND]
 brand_proportions <- merge(quantity_segment1_by_brand,
 quantity_other_by_brand)[, affinityToBrand := targetSegment/other]

 brand_proportions[order(-affinityToBrand)]

## Key: <BRAND>
##     BRAND targetSegment other affinityToBrand
##    <char>         <num> <num>           <num>
## 1:                    1     1               1

 quantity_segment1_by_pack <- segment1[, .(targetSegment =
 sum(PROD_QTY)/quantity_segment1), by = PACK_SIZE]

 quantity_other_by_pack <- other[, .(other = sum(PROD_QTY)/quantity_other),
by =
 PACK_SIZE]
 pack_proportions <- merge(quantity_segment1_by_pack,
quantity_other_by_pack)[,
 affinityToPack := targetSegment/other]

 pack_proportions[order(-affinityToPack)]

##     PACK_SIZE targetSegment       other affinityToPack
##         <num>         <num>       <num>          <num>
##  1:       270   0.029845724 0.023377359      1.2766936
##  2:       380   0.030156347 0.023832205      1.2653612
##  3:       330   0.057465314 0.046726826      1.2298142
##  4:       134   0.111979706 0.093743295      1.1945356
##  5:       110   0.099658314 0.083642285      1.1914824
##  6:       210   0.027308967 0.023400959      1.1670020
##  7:       135   0.013848623 0.012179999      1.1369971
##  8:       250   0.013460344 0.011905375      1.1306107
##  9:       170   0.075740319 0.075440042      1.0039803
## 10:       300   0.054954442 0.057263373      0.9596787
## 11:       175   0.239102299 0.251516868      0.9506412
## 12:       150   0.155130462 0.163446272      0.9491221
## 13:       165   0.052184717 0.058003570      0.8996811
## 14:       190   0.007014910 0.011589987      0.6052561
## 15:       180   0.003365086 0.005651245      0.5954592
## 16:       160   0.006005384 0.011525622      0.5210464
## 17:        90   0.005953614 0.011718716      0.5080431
## 18:       125   0.002821495 0.005623353      0.5017460
## 19:       200   0.008412715 0.017378543      0.4840863
## 20:        70   0.002847380 0.005889395      0.4834759
## 21:       220   0.002743839 0.006144710      0.4465369
##     PACK_SIZE targetSegment       other affinityToPack
```

```
data[PACK_SIZE == 270, unique(PROD_NAME)]
```

```
## [1] "Twisties Cheese     270g" "Twisties Chicken270g"
```