OSEMN Assignment -Best Buy customers rate iPhones with larger memory more favorably

Overview

Best Buy Co., Inc. is an American multinational consumer electronics corporation. people can have good electronic goods with reasonable prices from best buy using it's website or by vising stroes. it isheadquarterd in Richfield, Minnesota, a Minneapolis suburb. It operates in the United States, Puerto Rico, Mexico, Canada and China.[5] The company was founded by Richard M. Schulze and Gary Smoliak in 1966 as an audio specialty store. In 1983, it was renamed and rebranded with more emphasis placed on consumer electronics.

Best Buy's subsidiaries include CinemaNow, Geek Squad, Magnolia Audio Video, Pacific Sales, and Cowboom. Best Buy operates under the Best Buy, Best Buy Mobile, Geek Squad, Magnolia Audio Video and Pacific Sales brands in the United States; the Best Buy, Geek Squad, Best Buy Mobile brands in Canada; Best Buy Mobile and Five Star in China; and Best Buy, Best Buy Express, and Geek Squad in Mexico.[5] Best Buy sells cellular phones with phones from Verizon Wireless, AT andT Mobility, Sprint PCS, Boost Mobile,and T-Mobile USA,[6] in regular stores and standalone Best Buy Mobile stores in shopping malls.

Best Buy was named "Company of the Year" by Forbes magazine in 2004,[7] "Specialty Retailer of the Decade" by Discount Store News in 2001,[8] ranked in the Top 10 of "America's Most Generous Corporations" by Forbes in 2005 (based on 2004 giving),[9] and made Fortune magazine's List of Most Admired Companies in 2006 source:wikipedia Best Buy Mobile stores in shopping malls.

This paper will check and confirm that wheather iphones which have larger memory haves higher ratings than low memory iphones or not .ratings of high phones with lower memory such as 16 Gb and 32Gb as well as highger memory phones such as 64 and 128GB.I started our work with collecting data about iphones such as their memory and ratings.Next step would be to take average of ratings memorywoie and compare them.

Gathering Data

To gather the data needed for this comparison, I utilized the Best buy review API . which returns JSON (JavaScript Object Notation) which has three data elements that I am interested in for this report. The ID ,SKU and Ratings.

ID works as a identifier assigned to a perticuler product. The SKU attribute provides the SKU identifier for the product of a review .Data type of SKU is The rating attribute provides an average of all the ratings submitted for a product by reviewers. The ratings that can be submitted using a range from zero to five where five is the best rating that can be given. Ratings may be returned using decimals (example 3.5).data type of ratings is float.

The Reviews API provides access to individual reviews that have been submitted on BESTBUY.COM for the products we sell. You can use it to retrieve information such as the author of the review, the date that the review was submitted, the reviewer's rating (0-5 where 5 is the best) of a product and the reviewer's comments about the product.Along with real-time calls archives are available for reviews data. These are updated daily. Refer to archives for documentation on our archives.

The following shows the code needed to pull the data from best buy review api

```
library(jsonlite)

##
## Attaching package:  'jsonlite'
##
## The following object is masked from 'package:utils':
##
##    View

library (plyr)

url  <- 'http://api.remix.bestbuy.com/'   # the Best Buy API
path <- 'v1/reviews'                      # Reviews path
search <- '(sku=1722036|sku=172463|sku=1751846|sku=1755302|sku=1752845|sku=1761054|sku=1755
format <- '?format=json&'                 # Results will be in json format
key <- 'apiKey=ztk9ycxpg6rcmu9wfbq7gqe9&' # here is my key
                                          # get your own at:
                                          # https://developer.bestbuy.com/documentation/rev
                                          # Select 'Get API Key' in top right hand corner
fields <- 'show=id,sku,rating&'           # I only want these fields
pageSize <- 'pageSize=100'                # 100 results per page is the max allowed

# Paste0 allows me to concatenate all of the above into one long URL
URLall <-paste0(url, path, search, format, key, fields, pageSize)

#Here is what it looks like all together
# The above URL will create a page in json format
# I can read the json and convert it to a list using the jsonlite package
l = jsonlite::fromJSON(URLall)
# If I look at l, I see that
# My search yielded 7291  total records over 73 total pages
# But the the API only returned 100 records (100 is the max per page)

head(l)
```

```
## $from
## [1] 1
##
## $to
## [1] 100
##
## $total
## [1] 7291
##
## $currentPage
## [1] 1
##
## $totalPages
## [1] 73
##
## $queryTime
## [1] "0.037"
```

```
# to get all 7291 records, I will need to access pages 1-73
# if you look at the documentation for the api:
# https://developer.bestbuy.com/documentation#pagination-pagination
# you will see that you can specify the page you want

pageText <- '&page=' # i will be the page number

# r allows many types of loops - including a for loop

for (i in  1:73)
{
  URLpage <-paste0(URLall, pageText, i)

  l = jsonlite::fromJSON(URLpage)

  # The ldply function in the plyr package allows us
  #  to convert our list (l) to a dataframe (df)
  dfnew <- ldply (l, data.frame)

  # the first 9 rows are unneeded header information
  # the first 10 columns are unneeded header information
  # we will drop both the unneeded columns and rows
  dfnew<-dfnew[-c(1:9),-c(1:10)]

  # we will write each page out to a text file
  # each page will be appended to the bestbuy.csv file
  write.table(dfnew, file="bestbuy.csv", col.names=F, row.names=F, append=T, sep=",")
  # ignore the warning messages
```

```
}

# Now we can read in the complete set of page data
iphone.data <-read.csv("bestbuy.csv", header = F)

# Since we set col.names=F above, we need to name our columns
iphone.data<-rename(iphone.data,c(V1="ID", V2="SKU", V3 = "Rating"))
# now, our data is ready to analyse
```

Scrubbung the Data

to srub the data means get useful data which is important to calculate the result for the qustion asked above.here we need the group of higher and lower memory iphone and thier average ratings.so we take ID, SKU ID's, ratings for iphone 16GB,32GB,64GB and 128GB and take the average of ratinns

```
h1_iphonedata<-iphone.data[grep(7640008,iphone.data$SKU),]
h2_iphonedata<-iphone.data[grep(7640035,iphone.data$SKU),]
h3_iphonedata<-iphone.data[grep(1752872,iphone.data$SKU),]
h4_iphonedata<-iphone.data[grep(1755715 ,iphone.data$SKU),]
mean_h1_iphonedata<-mean(h1_iphonedata$Rating,)
mean_h2_iphonedata<-mean(h2_iphonedata$Rating,)
mean_h3_iphonedata<-mean(h3_iphonedata$Rating,)
mean_h4_iphonedata<-mean(h4_iphonedata$Rating,)
```

Exploring the data

In this section, we will take a deeper look into our Useful data. Our data is presented in a tabular format with three columns. The first column gives the ID, the second column gives the desired SKU, and the third column gives raring. Let us begin by determining the data type of Data using the class function.

```
class(h1_iphonedata)

## [1] "data.frame"

class(h2_iphonedata)

## [1] "data.frame"

class(h3_iphonedata)

## [1] "data.frame"

class(h4_iphonedata)

## [1] "data.frame"
```

Now let us go ahead and examine the structure of useful Data using the str function. This function is used to the structure of an R object.

```
str(h1_iphonedata)

## 'data.frame': 6186 obs. of  3 variables:
##  $ ID    : int  49096232 47146465 47102302 47094567 45324375 45302510 44572353 44411807 4
##  $ SKU   : int  7640008 7640008 7640008 7640008 7640008 7640008 7640008 7640008 7640008 7
##  $ Rating: int  5 5 5 5 5 5 5 5 5 ...

str(h2_iphonedata)

## 'data.frame': 24181 obs. of  3 variables:
##  $ ID    : int  50562784 49721452 49707095 48364027 48033306 48018326 47673724 47451633 4
##  $ SKU   : int  7640035 7640035 7640035 7640035 7640035 7640035 7640035 7640035 7640035 7
##  $ Rating: int  5 5 4 5 5 3 5 5 5 5 ...

str(h3_iphonedata)

## 'data.frame': 2417 obs. of  3 variables:
##  $ ID    : int  41336160 39407850 41336160 39407850 38310323 38094006 37718426 35989100 3
##  $ SKU   : int  1752872 1752872 1752872 1752872 1752872 1752872 1752872 1752872 1752872 1
##  $ Rating: int  5 5 5 5 5 4 4 5 5 5 ...

str(h4_iphonedata)

## 'data.frame': 11732 obs. of  3 variables:
##  $ ID    : int  16780390 16779462 16779383 16778428 16780390 16779462 16779383 16778428 4
##  $ SKU   : int  1755715 1755715 1755715 1755715 1755715 1755715 1755715 1755715 1755715 1
##  $ Rating: int  4 5 5 5 4 5 5 5 5 5 ...
```

The summary of the dataset displays the minimum, first quarter, median, mean, third quarter, and maximum value of each variable. The summary is displayed as followed

```
summary(h1_iphonedata)

##        ID                SKU              Rating
##  Min.   :37230684   Min.   :7640008   Min.   :3.000
##  1st Qu.:38316781   1st Qu.:7640008   1st Qu.:5.000
##  Median :39404405   Median :7640008   Median :5.000
##  Mean   :40092620   Mean   :7640008   Mean   :4.856
##  3rd Qu.:41173868   3rd Qu.:7640008   3rd Qu.:5.000
##  Max.   :49096232   Max.   :7640008   Max.   :5.000

summary(h2_iphonedata)
```

```
##        ID                 SKU              Rating
##  Min.   :36906235   Min.   :7640035   Min.   :1.000
##  1st Qu.:38724484   1st Qu.:7640035   1st Qu.:5.000
##  Median :42183487   Median :7640035   Median :5.000
##  Mean   :41534137   Mean   :7640035   Mean   :4.831
##  3rd Qu.:43876564   3rd Qu.:7640035   3rd Qu.:5.000
##  Max.   :50562784   Max.   :7640035   Max.   :5.000

summary(h3_iphonedata)

##        ID                 SKU              Rating
##  Min.   :18315609   Min.   :1752872   Min.   :3.000
##  1st Qu.:20468079   1st Qu.:1752872   1st Qu.:5.000
##  Median :22251119   Median :1752872   Median :5.000
##  Mean   :25423821   Mean   :1752872   Mean   :4.745
##  3rd Qu.:32568988   3rd Qu.:1752872   3rd Qu.:5.000
##  Max.   :41336160   Max.   :1752872   Max.   :5.000

summary(h4_iphonedata)

##        ID                 SKU              Rating
##  Min.   :16778428   Min.   :1755715   Min.   :1.000
##  1st Qu.:20548244   1st Qu.:1755715   1st Qu.:4.000
##  Median :24163676   Median :1755715   Median :5.000
##  Mean   :28244691   Mean   :1755715   Mean   :4.618
##  3rd Qu.:35348344   3rd Qu.:1755715   3rd Qu.:5.000
##  Max.   :47363454   Max.   :1755715   Max.   :5.000
```

Result This section will display the result after analyzing the data

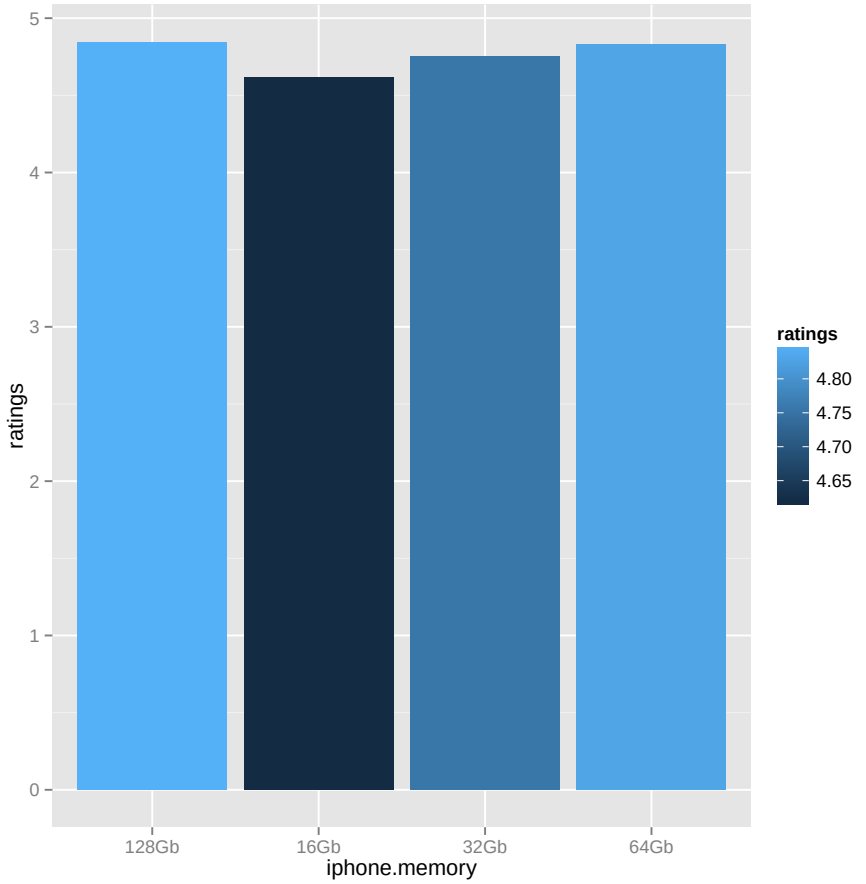
```
mean_h1_iphonedata

## [1] 4.855965

mean_h2_iphonedata

## [1] 4.831479

mean_h3_iphonedata

## [1] 4.745139

mean_h4_iphonedata

## [1] 4.617542
```

```
iphone.memory<-c("128Gb","64Gb","32Gb","16Gb")
ratings=c(4.845714,4.827815,4.75431,4.618387)
dfg=data.frame(iphone.memory,ratings)
dfg
```

```
##   iphone.memory  ratings
## 1         128Gb 4.845714
## 2          64Gb 4.827815
## 3          32Gb 4.754310
## 4          16Gb 4.618387
```

Creating graph

```
library(ggplot2)
g<-ggplot(dfg,aes(x=iphone.memory,y=ratings,fill=ratings))+geom_bar(stat="identity")
g
```



As from the graph it is clear that average rating 16Gb ais 4.65 and as memory is increasing from 32gb to 128gb average rating is also incgring(4.70,4.75ans4.80) Hence its cleare that best buy customers rate iphone with the larger memory more favorably.