## Task 2: Data Insights

### Targeting high value customers based on customer demographics and attributes.

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(lubridate)

##   
## Attaching package: 'lubridate'

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

library(readxl)  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.5 ✓ purrr 0.3.4  
## ✓ tibble 3.1.6 ✓ stringr 1.4.0  
## ✓ tidyr 1.1.4 ✓ forcats 0.5.1  
## ✓ readr 2.1.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x lubridate::as.difftime() masks base::as.difftime()  
## x lubridate::date() masks base::date()  
## x dplyr::filter() masks stats::filter()  
## x lubridate::intersect() masks base::intersect()  
## x dplyr::lag() masks stats::lag()  
## x lubridate::setdiff() masks base::setdiff()  
## x lubridate::union() masks base::union()

getwd()

## [1] "/Users/srushtidesai/Desktop"

setwd("/Users/srushtidesai/Desktop")

getwd()

## [1] "/Users/srushtidesai/Desktop"

df2<-read\_excel("KPMG\_VI/KPMG\_VI\_New\_raw\_data\_update\_final.xlsx",sheet="NewCustomerList")

## New names:  
## \* `` -> ...17  
## \* `` -> ...18  
## \* `` -> ...19  
## \* `` -> ...20  
## \* `` -> ...21

df3<-read\_excel("KPMG\_VI/KPMG\_VI\_New\_raw\_data\_update\_final.xlsx",sheet="CustomerDemographic")

### Data Exploration for New Customer List dataset

#Dropping the unnamed columns from the datasets  
df2 <- df2 %>% select(-...17,-...18,-...19,-...20,-...21)

colnames(df2)

## [1] "first\_name" "last\_name"   
## [3] "gender" "past\_3\_years\_bike\_related\_purchases"  
## [5] "DOB" "job\_title"   
## [7] "job\_industry\_category" "wealth\_segment"   
## [9] "deceased\_indicator" "owns\_car"   
## [11] "tenure" "address"   
## [13] "postcode" "state"   
## [15] "country" "property\_valuation"   
## [17] "Rank" "Value"

#Convert the datatype of DOB column from numeric to date format   
df2$DOB<-as.Date(as.numeric(df2$DOB,na.rm=TRUE), origin = "1899-12-30")

## Warning in as.Date(as.numeric(df2$DOB, na.rm = TRUE), origin = "1899-12-30"):  
## NAs introduced by coercion

df2 %>% select(DOB)

## # A tibble: 1,000 × 1  
## DOB   
## <date>   
## 1 NA   
## 2 NA   
## 3 1974-08-28  
## 4 NA   
## 5 NA   
## 6 NA   
## 7 NA   
## 8 NA   
## 9 NA   
## 10 NA   
## # … with 990 more rows

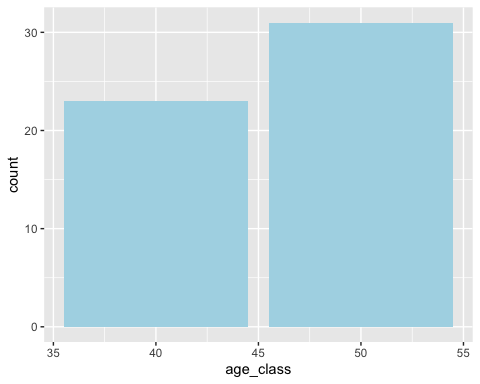
#Adding the calculated field like age and age\_class with the help of the DOB column  
df2 <- mutate(df2,age = as.numeric(difftime(Sys.Date(),df2$DOB, units = "weeks"))/52.25,age\_class = round(age/10)\*10 )

colnames(df2)

## [1] "first\_name" "last\_name"   
## [3] "gender" "past\_3\_years\_bike\_related\_purchases"  
## [5] "DOB" "job\_title"   
## [7] "job\_industry\_category" "wealth\_segment"   
## [9] "deceased\_indicator" "owns\_car"   
## [11] "tenure" "address"   
## [13] "postcode" "state"   
## [15] "country" "property\_valuation"   
## [17] "Rank" "Value"   
## [19] "age" "age\_class"

#Data Exploration: Age Distribution   
ggplot(data = df2, aes(x=age\_class)) + geom\_bar( fill="lightblue" )

## Warning: Removed 946 rows containing non-finite values (stat\_count).



### Data Exploration for Customer Demographics dataset

#### Categories are not correctly named. So we have to rename the categories.

#Rename the categories  
df3 <- df3 %>% mutate(gender = recode(gender, Femal = "Female",  
 Female = "Female",  
 F = "Female",  
 M = "Male",  
 Male = "Male"))

#to count number of occurrences grouped by gender  
table(df3$gender)

##   
## Female Male U   
## 2039 1873 88

#Convert the datatype of DOB column from numeric to date format   
df3$DOB<-as.Date(as.numeric(df3$DOB,na.rm=TRUE), origin = "1899-12-30")

## Warning in as.Date(as.numeric(df3$DOB, na.rm = TRUE), origin = "1899-12-30"):  
## NAs introduced by coercion

df3 %>% select(DOB)

## # A tibble: 4,000 × 1  
## DOB   
## <date>   
## 1 1953-10-12  
## 2 1980-12-16  
## 3 1954-01-20  
## 4 1961-10-03  
## 5 1977-05-13  
## 6 1966-09-16  
## 7 1976-02-23  
## 8 1962-03-30  
## 9 1973-03-10  
## 10 1988-10-11  
## # … with 3,990 more rows

#### The values are inconsistent, hence dropping the default column

#Dropping the default column from the datasets  
df3 <- df3 %>% select(-default)

colnames(df3)

## [1] "customer\_id" "first\_name"   
## [3] "last\_name" "gender"   
## [5] "past\_3\_years\_bike\_related\_purchases" "DOB"   
## [7] "job\_title" "job\_industry\_category"   
## [9] "wealth\_segment" "deceased\_indicator"   
## [11] "owns\_car" "tenure"

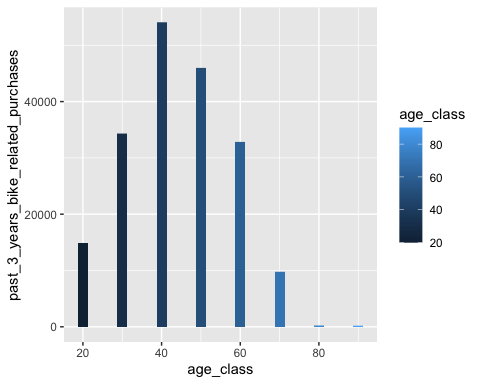
#Adding the calculated field like age and age\_class with the help of the DOB column  
df3 <- mutate(df3,age = as.numeric(difftime(Sys.Date(),df3$DOB, units = "weeks"))/52.25,age\_class = round(age/10)\*10 )

colnames(df3)

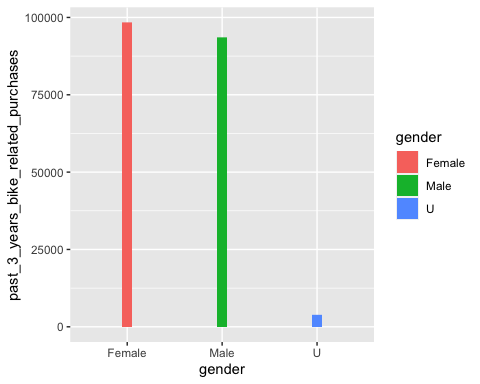
## [1] "customer\_id" "first\_name"   
## [3] "last\_name" "gender"   
## [5] "past\_3\_years\_bike\_related\_purchases" "DOB"   
## [7] "job\_title" "job\_industry\_category"   
## [9] "wealth\_segment" "deceased\_indicator"   
## [11] "owns\_car" "tenure"   
## [13] "age" "age\_class"

#Data Exploration: Age Distribution w.r.t. Bike Purchases  
ggplot(df3, aes(x=age\_class, y=past\_3\_years\_bike\_related\_purchases, fill=age\_class)) +   
geom\_col(width = 2.5)

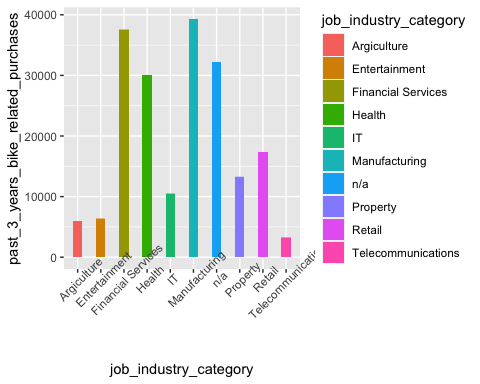
## Warning: Removed 88 rows containing missing values (position\_stack).



##Data Exploration: Gender Distribution w.r.t. Bike Purchases  
ggplot(df3, aes(x=gender, y=past\_3\_years\_bike\_related\_purchases, fill=gender)) +   
geom\_col(width = 0.1)



##Data Exploration: Job Industry Category w.r.t. Bike Purchases  
ggplot(df3, aes(x=job\_industry\_category, y=past\_3\_years\_bike\_related\_purchases, fill=job\_industry\_category)) +   
geom\_col(width = 0.4) +theme(axis.text.x = element\_text(angle = 45))



### RFM Analysis

#Reading the each file separately  
df1<-read\_excel("KPMG\_VI/KPMG\_VI\_New\_raw\_data\_update\_final.xlsx",sheet="Transactions")

colnames(df1)

## [1] "transaction\_id" "product\_id"   
## [3] "customer\_id" "transaction\_date"   
## [5] "online\_order" "order\_status"   
## [7] "brand" "product\_line"   
## [9] "product\_class" "product\_size"   
## [11] "list\_price" "standard\_cost"   
## [13] "product\_first\_sold\_date"

#to count number of occurrences grouped by product\_first\_size\_sold\_date  
table(df1$product\_first\_sold\_date)

##   
## 33259 33364 33429 33455 33549 33552 33879 33888 34071 34079 34115 34143 34165   
## 213 203 201 181 193 179 234 211 186 179 206 184 207   
## 34170 34244 34527 34556 34586 34996 35052 35160 35378 35455 35470 35560 35667   
## 205 193 200 206 162 209 206 189 191 206 206 195 213   
## 35707 36145 36146 36334 36361 36367 36498 36668 36833 37220 37337 37499 37539   
## 181 202 201 202 201 205 198 181 189 196 189 186 193   
## 37626 37659 37668 37698 37823 37838 37873 37874 38002 38193 38206 38216 38258   
## 179 163 186 203 227 191 207 189 215 189 210 220 201   
## 38339 38482 38573 38647 38693 38750 38859 38991 39031 39298 39427 39526 39880   
## 189 219 203 213 191 202 194 171 203 212 208 213 222   
## 39915 40303 40336 40410 40487 40553 40618 40649 40670 40672 40779 40784 41009   
## 199 184 218 203 211 199 207 182 193 176 193 192 215   
## 41047 41064 41167 41245 41345 41434 41533 41701 41848 41922 42105 42145 42172   
## 203 229 192 205 202 213 202 212 169 166 193 177 187   
## 42218 42226 42295 42404 42458 42560 42688 42696 42710   
## 193 205 191 168 206 216 211 182 207

#convert product\_first\_sold\_date from numeric to date format  
df1$product\_first\_sold\_date<-as.Date(as.numeric(df1$product\_first\_sold\_date,na.rm=TRUE), origin = "1899-12-30")

#to count number of occurrences grouped by product\_first\_size\_sold\_date  
table(df1$product\_first\_sold\_date)

##   
## 1991-01-21 1991-05-06 1991-07-10 1991-08-05 1991-11-07 1991-11-10 1992-10-02   
## 213 203 201 181 193 179 234   
## 1992-10-11 1993-04-12 1993-04-20 1993-05-26 1993-06-23 1993-07-15 1993-07-20   
## 211 186 179 206 184 207 205   
## 1993-10-02 1994-07-12 1994-08-10 1994-09-09 1995-10-24 1995-12-19 1996-04-05   
## 193 200 206 162 209 206 189   
## 1996-11-09 1997-01-25 1997-02-09 1997-05-10 1997-08-25 1997-10-04 1998-12-16   
## 191 206 206 195 213 181 202   
## 1998-12-17 1999-06-23 1999-07-20 1999-07-26 1999-12-04 2000-05-22 2000-11-03   
## 201 202 201 205 198 181 189   
## 2001-11-25 2002-03-22 2002-08-31 2002-10-10 2003-01-05 2003-02-07 2003-02-16   
## 196 189 186 193 179 163 186   
## 2003-03-18 2003-07-21 2003-08-05 2003-09-09 2003-09-10 2004-01-16 2004-07-25   
## 203 227 191 207 189 215 189   
## 2004-08-07 2004-08-17 2004-09-28 2004-12-18 2005-05-10 2005-08-09 2005-10-22   
## 210 220 201 189 219 203 213   
## 2005-12-07 2006-02-02 2006-05-22 2006-10-01 2006-11-10 2007-08-04 2007-12-11   
## 191 202 194 171 203 212 208   
## 2008-03-19 2009-03-08 2009-04-12 2010-05-05 2010-06-07 2010-08-20 2010-11-05   
## 213 222 199 184 218 203 211   
## 2011-01-10 2011-03-16 2011-04-16 2011-05-07 2011-05-09 2011-08-24 2011-08-29   
## 199 207 182 193 176 193 192   
## 2012-04-10 2012-05-18 2012-06-04 2012-09-15 2012-12-02 2013-03-12 2013-06-09   
## 215 203 229 192 205 202 213   
## 2013-09-16 2014-03-03 2014-07-28 2014-10-10 2015-04-11 2015-05-21 2015-06-17   
## 202 212 169 166 193 177 187   
## 2015-08-02 2015-08-10 2015-10-18 2016-02-04 2016-03-29 2016-07-09 2016-11-14   
## 193 205 191 168 206 216 211   
## 2016-11-22 2016-12-06   
## 182 207

#Adding the calculated field like profit  
df1 <- mutate(df1,profit = df1$list\_price - df1$standard\_cost)

colnames(df1)

## [1] "transaction\_id" "product\_id"   
## [3] "customer\_id" "transaction\_date"   
## [5] "online\_order" "order\_status"   
## [7] "brand" "product\_line"   
## [9] "product\_class" "product\_size"   
## [11] "list\_price" "standard\_cost"   
## [13] "product\_first\_sold\_date" "profit"

df1 %>% select(profit)

## # A tibble: 20,000 × 1  
## profit  
## <dbl>  
## 1 17.9  
## 2 1703.   
## 3 1545.   
## 4 817.   
## 5 1056.   
## 6 709.   
## 7 15.1  
## 8 1279.   
## 9 690.   
## 10 1070.   
## # … with 19,990 more rows

head(df1)

## # A tibble: 6 × 14  
## transaction\_id product\_id customer\_id transaction\_date online\_order  
## <dbl> <dbl> <dbl> <dttm> <lgl>   
## 1 1 2 2950 2017-02-25 00:00:00 FALSE   
## 2 2 3 3120 2017-05-21 00:00:00 TRUE   
## 3 3 37 402 2017-10-16 00:00:00 FALSE   
## 4 4 88 3135 2017-08-31 00:00:00 FALSE   
## 5 5 78 787 2017-10-01 00:00:00 TRUE   
## 6 6 25 2339 2017-03-08 00:00:00 TRUE   
## # … with 9 more variables: order\_status <chr>, brand <chr>, product\_line <chr>,  
## # product\_class <chr>, product\_size <chr>, list\_price <dbl>,  
## # standard\_cost <dbl>, product\_first\_sold\_date <date>, profit <dbl>

str(df1)

## tibble [20,000 × 14] (S3: tbl\_df/tbl/data.frame)  
## $ transaction\_id : num [1:20000] 1 2 3 4 5 6 7 8 9 10 ...  
## $ product\_id : num [1:20000] 2 3 37 88 78 25 22 15 67 12 ...  
## $ customer\_id : num [1:20000] 2950 3120 402 3135 787 ...  
## $ transaction\_date : POSIXct[1:20000], format: "2017-02-25" "2017-05-21" ...  
## $ online\_order : logi [1:20000] FALSE TRUE FALSE FALSE TRUE TRUE ...  
## $ order\_status : chr [1:20000] "Approved" "Approved" "Approved" "Approved" ...  
## $ brand : chr [1:20000] "Solex" "Trek Bicycles" "OHM Cycles" "Norco Bicycles" ...  
## $ product\_line : chr [1:20000] "Standard" "Standard" "Standard" "Standard" ...  
## $ product\_class : chr [1:20000] "medium" "medium" "low" "medium" ...  
## $ product\_size : chr [1:20000] "medium" "large" "medium" "medium" ...  
## $ list\_price : num [1:20000] 71.5 2091.5 1793.4 1198.5 1765.3 ...  
## $ standard\_cost : num [1:20000] 53.6 388.9 248.8 381.1 709.5 ...  
## $ product\_first\_sold\_date: Date[1:20000], format: "2012-12-02" "2014-03-03" ...  
## $ profit : num [1:20000] 17.9 1702.5 1544.6 817.4 1055.8 ...

max(df1$transaction\_date)

## [1] "2017-12-30 UTC"

### Date of the Analysis: The date relative to which you want the recency to be calculated.  
analysis\_date <-max(df1$transaction\_date)

analysis\_date

## [1] "2017-12-30 UTC"

install.packages("rfm", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/5r/9ds45bhj3wv0k4l6z6pvw9tc0000gn/T//Rtmpjuw5Pe/downloaded\_packages

# “rfm” package requires that you have developer tools set up in your environment.   
  
install.packages("devtools", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/5r/9ds45bhj3wv0k4l6z6pvw9tc0000gn/T//Rtmpjuw5Pe/downloaded\_packages

library ("rfm")

#Checking for null values for each column  
colSums(is.na(df1))

## transaction\_id product\_id customer\_id   
## 0 0 0   
## transaction\_date online\_order order\_status   
## 0 360 0   
## brand product\_line product\_class   
## 197 197 197   
## product\_size list\_price standard\_cost   
## 197 0 197   
## product\_first\_sold\_date profit   
## 197 197

#Removing NA values from the dataset  
df1 <- na.omit(df1)

#Checking for null values for each column  
colSums(is.na(df1))

## transaction\_id product\_id customer\_id   
## 0 0 0   
## transaction\_date online\_order order\_status   
## 0 0 0   
## brand product\_line product\_class   
## 0 0 0   
## product\_size list\_price standard\_cost   
## 0 0 0   
## product\_first\_sold\_date profit   
## 0 0

report <- rfm\_table\_order(df1, customer\_id, transaction\_date, profit,analysis\_date)

report

## # A tibble: 3,492 × 9  
## customer\_id date\_most\_recent recency\_days transaction\_count amount  
## <dbl> <dttm> <dbl> <dbl> <dbl>  
## 1 1 2017-12-23 00:00:00 7 11 3018.  
## 2 2 2017-08-24 00:00:00 128 3 2226.  
## 3 3 2017-09-19 00:00:00 102 7 3290.  
## 4 4 2017-06-18 00:00:00 195 2 221.  
## 5 5 2017-12-14 00:00:00 16 6 2395.  
## 6 6 2017-10-27 00:00:00 64 5 3947.  
## 7 7 2017-04-21 00:00:00 253 3 220.  
## 8 8 2017-12-08 00:00:00 22 9 5364.  
## 9 9 2017-10-13 00:00:00 78 6 2353.  
## 10 10 2017-11-17 00:00:00 43 5 3358.  
## # … with 3,482 more rows, and 4 more variables: recency\_score <int>,  
## # frequency\_score <int>, monetary\_score <int>, rfm\_score <dbl>

#Saving the report  
write.csv (report$rfm,"rfm\_report.csv", row.names = FALSE)

### Customer Segmentation

#The data being used here is ideally broken down into the segments  
segment\_titles <- c("Champions", "Loyal Customers", "Potential Loyalist","New Customers", "Promising", "Need Attention", "About To Sleep", "At Risk", "Can’t Lose Them", "Hibernating", "Lost")

#Numerical threshold for each component of RFM that categorizes customers in the segments.  
r\_low <- c(4, 2, 3, 4, 3, 2, 2, 1, 1, 1,1)  
r\_high <- c(5, 5, 5, 5, 4, 3, 3, 2, 1, 2,2)  
f\_low <- c(4, 3, 1, 1, 1, 2, 1, 2, 4,1, 1)  
f\_high <- c(5, 5, 3, 1, 1, 3, 2, 5, 5, 2, 2)  
m\_low <- c(4, 3, 1, 1, 1, 2, 1, 2, 4, 1, 1)  
m\_high <- c(5, 5, 3, 1, 1, 3, 2, 5, 5, 2, 2)

divisions<-rfm\_segment(report, segment\_titles, r\_low, r\_high, f\_low, f\_high, m\_low, m\_high)

head(divisions,10)

## # A tibble: 10 × 9  
## customer\_id segment rfm\_score transaction\_cou… recency\_days amount  
## <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 1 Loyal Customers 553 11 7 3018.  
## 2 2 Hibernating 112 3 128 2226.  
## 3 3 Loyal Customers 243 7 102 3290.  
## 4 4 Hibernating 111 2 195 221.  
## 5 5 Potential Loyalist 432 6 16 2395.  
## 6 6 At Risk 224 5 64 3947.  
## 7 7 Hibernating 111 3 253 220.  
## 8 8 Champions 455 9 22 5364.  
## 9 9 Need Attention 232 6 78 2353.  
## 10 10 Others 324 5 43 3358.  
## # … with 3 more variables: recency\_score <int>, frequency\_score <int>,  
## # monetary\_score <int>

#to count number of occurrences grouped by segments  
table(divisions$segment)

##   
## About To Sleep At Risk Champions Hibernating   
## 212 310 459 344   
## Loyal Customers Need Attention Others Potential Loyalist   
## 863 95 390 819

pdf <- divisions %>% count(segment) %>% arrange(desc(n)) %>% rename(Segment = segment, Count = n)

pdf

## # A tibble: 8 × 2  
## Segment Count  
## <chr> <int>  
## 1 Loyal Customers 863  
## 2 Potential Loyalist 819  
## 3 Champions 459  
## 4 Others 390  
## 5 Hibernating 344  
## 6 At Risk 310  
## 7 About To Sleep 212  
## 8 Need Attention 95

pdf <- pdf %>% mutate(percent = Count/ sum(Count) \* 100 )

pdf

## # A tibble: 8 × 3  
## Segment Count percent  
## <chr> <int> <dbl>  
## 1 Loyal Customers 863 24.7   
## 2 Potential Loyalist 819 23.5   
## 3 Champions 459 13.1   
## 4 Others 390 11.2   
## 5 Hibernating 344 9.85  
## 6 At Risk 310 8.88  
## 7 About To Sleep 212 6.07  
## 8 Need Attention 95 2.72

pdf$percent <- round(pdf$percent, digits=2)

pdf

## # A tibble: 8 × 3  
## Segment Count percent  
## <chr> <int> <dbl>  
## 1 Loyal Customers 863 24.7   
## 2 Potential Loyalist 819 23.4   
## 3 Champions 459 13.1   
## 4 Others 390 11.2   
## 5 Hibernating 344 9.85  
## 6 At Risk 310 8.88  
## 7 About To Sleep 212 6.07  
## 8 Need Attention 95 2.72

write.csv(pdf,"piechart\_viz.csv")

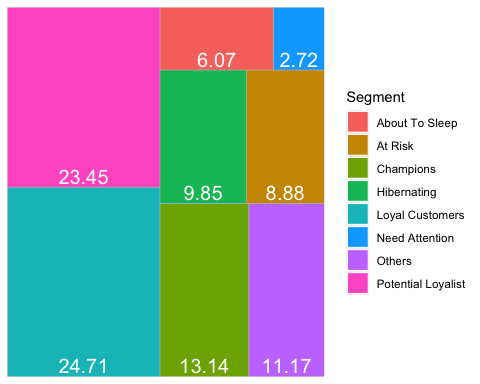
# To create a treemap visualization  
install.packages("treemapify", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/5r/9ds45bhj3wv0k4l6z6pvw9tc0000gn/T//Rtmpjuw5Pe/downloaded\_packages

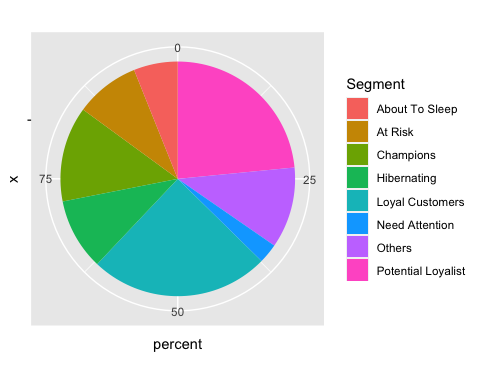
library(treemapify)

# Distribution of customer segments in percentages  
pdf %>% ggplot(aes(area = percent, fill = Segment, label = percent , label = paste(Segment))) +  
 geom\_treemap() +  
 geom\_treemap\_text(colour = "white",  
 place = "bottom",  
 size = 15)

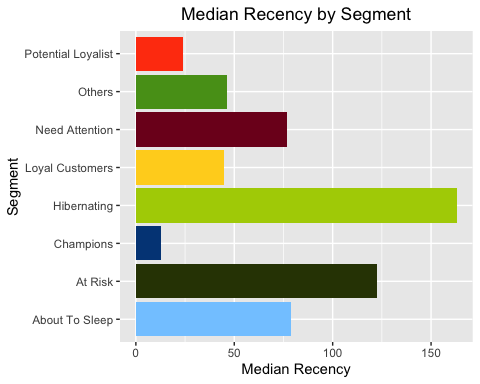
## Warning: Duplicated aesthetics after name standardisation: label



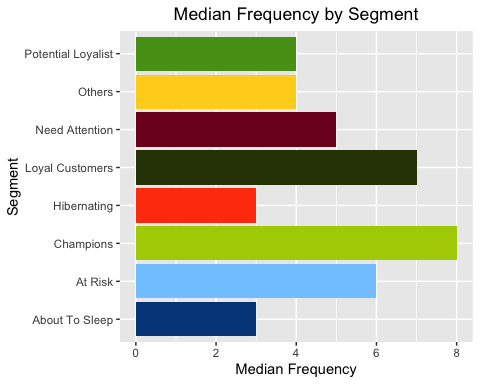
pdf %>% ggplot( aes(x = "", y = percent, fill = Segment)) +  
 geom\_col() +  
 coord\_polar(theta = "y")



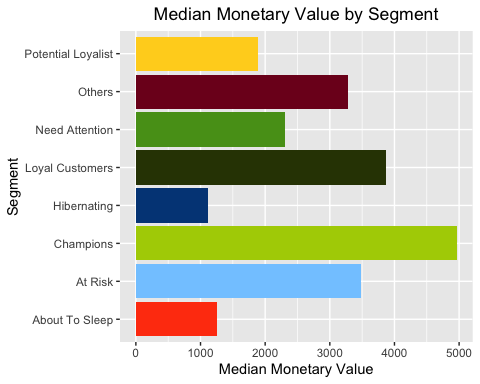
#Median Recency  
rfm\_plot\_median\_recency(divisions)



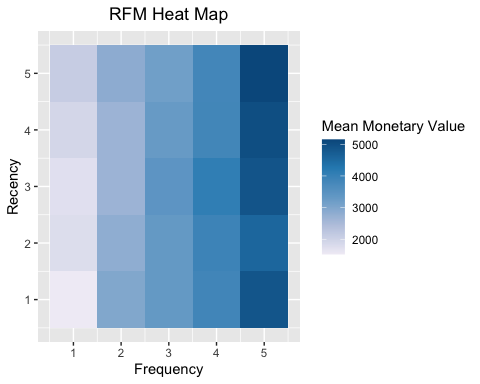
#Median Frequency  
rfm\_plot\_median\_frequency(divisions)



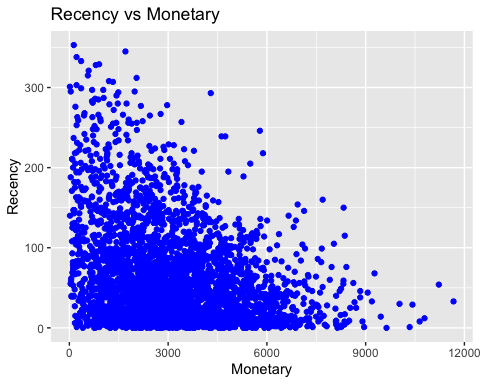
#Median Monetary Value  
rfm\_plot\_median\_monetary(divisions)



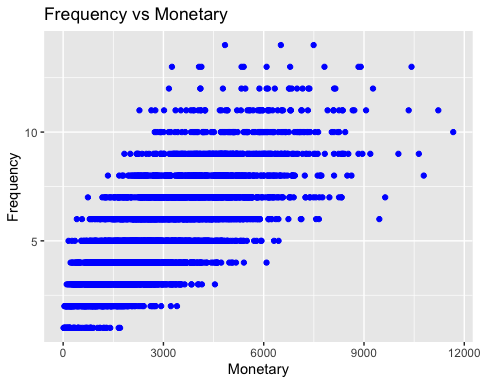
# Heat Map  
# Higher scores of frequency and recency are characterized by higher average monetary value as indicated by the darker areas in the heatmap.  
rfm\_heatmap(report)



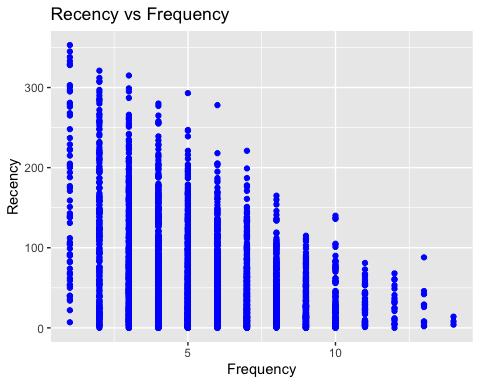
# Recency vs Monetary Value  
# Higher revenue would be associated with most recent visits (that is low recency value which implies most recent visits.)  
rfm\_rm\_plot(report)



# Frequency vs Monetary Value  
# As the frequency of visits increases, the revenue generated also increases.   
rfm\_fm\_plot(report)



# Recency vs Frequency  
# Higher frequency would be associated with the most recent visits.   
rfm\_rf\_plot(report)



# Merging the transactions(df1) and divisions dataframes for data visualization   
d23 <- merge(x = divisions, y = df1, by = "customer\_id", all = TRUE)

head(d23,10)

## customer\_id segment rfm\_score transaction\_count recency\_days amount  
## 1 1 Loyal Customers 553 11 7 3018.09  
## 2 1 Loyal Customers 553 11 7 3018.09  
## 3 1 Loyal Customers 553 11 7 3018.09  
## 4 1 Loyal Customers 553 11 7 3018.09  
## 5 1 Loyal Customers 553 11 7 3018.09  
## 6 1 Loyal Customers 553 11 7 3018.09  
## 7 1 Loyal Customers 553 11 7 3018.09  
## 8 1 Loyal Customers 553 11 7 3018.09  
## 9 1 Loyal Customers 553 11 7 3018.09  
## 10 1 Loyal Customers 553 11 7 3018.09  
## recency\_score frequency\_score monetary\_score transaction\_id product\_id  
## 1 5 5 3 14486 23  
## 2 5 5 3 3765 38  
## 3 5 5 3 15663 32  
## 4 5 5 3 18970 11  
## 5 5 5 3 9785 72  
## 6 5 5 3 16423 9  
## 7 5 5 3 14931 31  
## 8 5 5 3 5157 47  
## 9 5 5 3 13644 25  
## 10 5 5 3 94 86  
## transaction\_date online\_order order\_status brand product\_line  
## 1 2017-03-27 FALSE Approved Norco Bicycles Mountain  
## 2 2017-04-06 TRUE Approved Solex Standard  
## 3 2017-06-04 TRUE Approved Giant Bicycles Standard  
## 4 2017-03-29 TRUE Approved Giant Bicycles Standard  
## 5 2017-01-05 FALSE Approved Norco Bicycles Standard  
## 6 2017-12-09 TRUE Approved OHM Cycles Road  
## 7 2017-12-14 TRUE Approved Giant Bicycles Standard  
## 8 2017-05-11 TRUE Approved Trek Bicycles Road  
## 9 2017-05-19 FALSE Approved Giant Bicycles Road  
## 10 2017-12-23 FALSE Approved OHM Cycles Standard  
## product\_class product\_size list\_price standard\_cost product\_first\_sold\_date  
## 1 low small 688.63 612.88 1991-08-05  
## 2 medium medium 1577.53 826.51 2008-03-19  
## 3 medium medium 642.70 211.37 2011-03-16  
## 4 high medium 1274.93 764.96 1996-11-09  
## 5 medium medium 360.40 270.30 2003-09-09  
## 6 medium medium 742.54 667.40 2003-08-05  
## 7 medium medium 230.91 173.18 2002-03-22  
## 8 low small 1720.70 1531.42 2003-07-21  
## 9 medium medium 1538.99 829.65 1991-11-10  
## 10 medium medium 235.63 125.07 2005-05-10  
## profit  
## 1 75.75  
## 2 751.02  
## 3 431.33  
## 4 509.97  
## 5 90.10  
## 6 75.14  
## 7 57.73  
## 8 189.28  
## 9 709.34  
## 10 110.56

#Profit by each customer segments  
dataviz1 <- d23 %>% select(segment,profit) %>% group\_by(segment) %>% summarize(total\_profit = sum(profit)) %>% arrange(desc(total\_profit))

dataviz1

## # A tibble: 8 × 2  
## segment total\_profit  
## <chr> <dbl>  
## 1 Loyal Customers 3559897.  
## 2 Champions 2424068.  
## 3 Potential Loyalist 1506139.  
## 4 Others 1237971.  
## 5 At Risk 1116083.  
## 6 Hibernating 392100.  
## 7 About To Sleep 269393.  
## 8 Need Attention 223164.

write.csv(dataviz1,"dataviz1.csv")

df33<-read\_csv("CustomerDemographic\_update\_cleaned.csv")

## New names:  
## \* `` -> ...1

## Rows: 4000 Columns: 15

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (8): first\_name, last\_name, gender, job\_title, job\_industry\_category, w...  
## dbl (6): ...1, customer\_id, past\_3\_years\_bike\_related\_purchases, tenure, ag...  
## date (1): DOB

##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

head(df33)

## # A tibble: 6 × 15  
## ...1 customer\_id first\_name last\_name gender past\_3\_years\_bik… DOB   
## <dbl> <dbl> <chr> <chr> <chr> <dbl> <date>   
## 1 1 1 Laraine Medendorp Female 93 1953-10-12  
## 2 2 2 Eli Bockman Male 81 1980-12-16  
## 3 3 3 Arlin Dearle Male 61 1954-01-20  
## 4 4 4 Talbot <NA> Male 33 1961-10-03  
## 5 5 5 Sheila-kathryn Calton Female 56 1977-05-13  
## 6 6 6 Curr Duckhouse Male 35 1966-09-16  
## # … with 8 more variables: job\_title <chr>, job\_industry\_category <chr>,  
## # wealth\_segment <chr>, deceased\_indicator <chr>, owns\_car <chr>,  
## # tenure <dbl>, age <dbl>, age\_class <dbl>

#Removing NA values from the dataset  
df33 <- na.omit(df33)

head(df33)

## # A tibble: 6 × 15  
## ...1 customer\_id first\_name last\_name gender past\_3\_years\_bik… DOB   
## <dbl> <dbl> <chr> <chr> <chr> <dbl> <date>   
## 1 1 1 Laraine Medendorp Female 93 1953-10-12  
## 2 2 2 Eli Bockman Male 81 1980-12-16  
## 3 3 3 Arlin Dearle Male 61 1954-01-20  
## 4 5 5 Sheila-kathryn Calton Female 56 1977-05-13  
## 5 8 8 Rod Inder Male 31 1962-03-30  
## 6 9 9 Mala Lind Female 97 1973-03-10  
## # … with 8 more variables: job\_title <chr>, job\_industry\_category <chr>,  
## # wealth\_segment <chr>, deceased\_indicator <chr>, owns\_car <chr>,  
## # tenure <dbl>, age <dbl>, age\_class <dbl>

# Merging the transactions and divisions (that is d23) and customer demographic (df33) dataframes for data visualization   
d333 <- merge(x = df33, y = d23, by = "customer\_id", all = TRUE)

head(d333)

## customer\_id ...1 first\_name last\_name gender  
## 1 1 1 Laraine Medendorp Female  
## 2 1 1 Laraine Medendorp Female  
## 3 1 1 Laraine Medendorp Female  
## 4 1 1 Laraine Medendorp Female  
## 5 1 1 Laraine Medendorp Female  
## 6 1 1 Laraine Medendorp Female  
## past\_3\_years\_bike\_related\_purchases DOB job\_title  
## 1 93 1953-10-12 Executive Secretary  
## 2 93 1953-10-12 Executive Secretary  
## 3 93 1953-10-12 Executive Secretary  
## 4 93 1953-10-12 Executive Secretary  
## 5 93 1953-10-12 Executive Secretary  
## 6 93 1953-10-12 Executive Secretary  
## job\_industry\_category wealth\_segment deceased\_indicator owns\_car tenure  
## 1 Health Mass Customer N Yes 11  
## 2 Health Mass Customer N Yes 11  
## 3 Health Mass Customer N Yes 11  
## 4 Health Mass Customer N Yes 11  
## 5 Health Mass Customer N Yes 11  
## 6 Health Mass Customer N Yes 11  
## age age\_class segment rfm\_score transaction\_count recency\_days  
## 1 68.16131 70 Loyal Customers 553 11 7  
## 2 68.16131 70 Loyal Customers 553 11 7  
## 3 68.16131 70 Loyal Customers 553 11 7  
## 4 68.16131 70 Loyal Customers 553 11 7  
## 5 68.16131 70 Loyal Customers 553 11 7  
## 6 68.16131 70 Loyal Customers 553 11 7  
## amount recency\_score frequency\_score monetary\_score transaction\_id  
## 1 3018.09 5 5 3 14931  
## 2 3018.09 5 5 3 15663  
## 3 3018.09 5 5 3 18970  
## 4 3018.09 5 5 3 13644  
## 5 3018.09 5 5 3 3765  
## 6 3018.09 5 5 3 5157  
## product\_id transaction\_date online\_order order\_status brand  
## 1 31 2017-12-14 TRUE Approved Giant Bicycles  
## 2 32 2017-06-04 TRUE Approved Giant Bicycles  
## 3 11 2017-03-29 TRUE Approved Giant Bicycles  
## 4 25 2017-05-19 FALSE Approved Giant Bicycles  
## 5 38 2017-04-06 TRUE Approved Solex  
## 6 47 2017-05-11 TRUE Approved Trek Bicycles  
## product\_line product\_class product\_size list\_price standard\_cost  
## 1 Standard medium medium 230.91 173.18  
## 2 Standard medium medium 642.70 211.37  
## 3 Standard high medium 1274.93 764.96  
## 4 Road medium medium 1538.99 829.65  
## 5 Standard medium medium 1577.53 826.51  
## 6 Road low small 1720.70 1531.42  
## product\_first\_sold\_date profit  
## 1 2002-03-22 57.73  
## 2 2011-03-16 431.33  
## 3 1996-11-09 509.97  
## 4 1991-11-10 709.34  
## 5 2008-03-19 751.02  
## 6 2003-07-21 189.28

#Checking for null values for each column  
colSums(is.na(d333))

## customer\_id ...1   
## 0 3246   
## first\_name last\_name   
## 3246 3246   
## gender past\_3\_years\_bike\_related\_purchases   
## 3246 3246   
## DOB job\_title   
## 3246 3246   
## job\_industry\_category wealth\_segment   
## 3246 3246   
## deceased\_indicator owns\_car   
## 3246 3246   
## tenure age   
## 3246 3246   
## age\_class segment   
## 3246 409   
## rfm\_score transaction\_count   
## 409 409   
## recency\_days amount   
## 409 409   
## recency\_score frequency\_score   
## 409 409   
## monetary\_score transaction\_id   
## 409 409   
## product\_id transaction\_date   
## 409 409   
## online\_order order\_status   
## 409 409   
## brand product\_line   
## 409 409   
## product\_class product\_size   
## 409 409   
## list\_price standard\_cost   
## 409 409   
## product\_first\_sold\_date profit   
## 409 409

#Removing NA values from the dataset  
d333 <- na.omit(d333)

#Checking for null values for each column  
colSums(is.na(d333))

## customer\_id ...1   
## 0 0   
## first\_name last\_name   
## 0 0   
## gender past\_3\_years\_bike\_related\_purchases   
## 0 0   
## DOB job\_title   
## 0 0   
## job\_industry\_category wealth\_segment   
## 0 0   
## deceased\_indicator owns\_car   
## 0 0   
## tenure age   
## 0 0   
## age\_class segment   
## 0 0   
## rfm\_score transaction\_count   
## 0 0   
## recency\_days amount   
## 0 0   
## recency\_score frequency\_score   
## 0 0   
## monetary\_score transaction\_id   
## 0 0   
## product\_id transaction\_date   
## 0 0   
## online\_order order\_status   
## 0 0   
## brand product\_line   
## 0 0   
## product\_class product\_size   
## 0 0   
## list\_price standard\_cost   
## 0 0   
## product\_first\_sold\_date profit   
## 0 0

write.csv(d333,"3dataframes\_merged\_dataviz.csv")

#Profit by each gender  
dataviz2 <- d333 %>% select(gender,profit) %>% group\_by(gender) %>% summarize(total\_profit = sum(profit))

dataviz2

## # A tibble: 2 × 2  
## gender total\_profit  
## <chr> <dbl>  
## 1 Female 4566962.  
## 2 Male 4387828.

write.csv(dataviz2,"dataviz2.csv")

#Profit by wealth\_segment  
dataviz3 <- d333 %>% select(wealth\_segment,profit) %>% group\_by(wealth\_segment) %>% summarize(total\_profit = sum(profit))

dataviz3

## # A tibble: 3 × 2  
## wealth\_segment total\_profit  
## <chr> <dbl>  
## 1 Affluent Customer 2240658.  
## 2 High Net Worth 2239422.  
## 3 Mass Customer 4474711.

write.csv(dataviz3,"dataviz3.csv")