Marketing_Analysis

Owais Khan

11/17/2021

I can write professional reports using R Markdown. A marketining company has a data of its customer's purchases. They need to know which customers are interested in what product so that they can send them numerious promotions and deals specific to their interest. They also want to wake up those customers who hasn't made any purchase from the store by sending them discounted coupons and deal through their contact information.

I can explore new functions or packages and implement them into analyses

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.4     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

##
## Attaching package: 'lubridate'

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

##
## Attaching package: 'data.table'

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

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```r
marketing_data <- read_csv("~/STA 518/R-for-data-science/data/marketing_campaign.csv")
```

I can import data from a variety of file types

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   Education = col_character(),
##   Marital_Status = col_character(),
##   Dt_Customer = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
print(spec(marketing_data))
```

```
## cols(
##   ID = col_double(),
##   Year_Birth = col_double(),
##   Education = col_character(),
##   Marital_Status = col_character(),
##   Income = col_double(),
##   Kidhome = col_double(),
##   Teenhome = col_double(),
##   Dt_Customer = col_character(),
##   Recency = col_double(),
##   MntWines = col_double(),
##   MntFruits = col_double(),
##   MntMeatProducts = col_double(),
##   MntFishProducts = col_double(),
##   MntSweetProducts = col_double(),
##   MntGoldProds = col_double(),
##   NumDealsPurchases = col_double(),
##   NumWebPurchases = col_double(),
##   NumCatalogPurchases = col_double(),
##   NumStorePurchases = col_double(),
##   NumWebVisitsMonth = col_double(),
##   AcceptedCmp3 = col_double(),
##   AcceptedCmp4 = col_double(),
##   AcceptedCmp5 = col_double(),
##   AcceptedCmp1 = col_double(),
##   AcceptedCmp2 = col_double(),
##   Complain = col_double(),
##   Z_CostContact = col_double(),
##   Z_Revenue = col_double(),
##   Response = col_double()
## )
```

Write clear, efficient, and well-documented R programs.

```
#Eliminating unwanted variables
marketing_data <- marketing_data %>% select(ID:NumWebVisitsMonth)

#Calculating age of customers and the years since they're purchasing from given marketing data
marketing_data <- marketing_data %>%
  select(ID:NumWebVisitsMonth) %>%
  mutate(age = year(today()) - Year_Birth, customer_since = year(today()) - year(Dt_Customer))
```

```r
#Removing all NA values from marketing dataset
cleaned_marketing_data <- na.omit(marketing_data)

#Generating customer data with unique ID's from marketing dataset
customer_data <- marketing_data %>%
  select(ID) %>%
  mutate(first_name = randomNames(nrow(marketing_data), which.names="first"),
         last_name = randomNames(nrow(marketing_data), which.names="last"),
         email = paste0(first_name, "_", last_name, "@gmail.com"),
         phone_number = r_phone_numbers(nrow(marketing_data)))
```

```r
#Removing all NA values from marketing dataset
cleaned_marketing_data <- na.omit(marketing_data)
```

I can restructure information to be in a "tidy" format.

```r
merged_data <- merge(x=cleaned_marketing_data,y=customer_data,by="ID")
```

I can combine information from multiple data sources.

```r
#Finding out contact information of top 10 customers with highest purchase of Meat products
meat_lovers <- meat_lovers <- merged_data %>%
  arrange(desc(MntMeatProducts)) %>%
  head(10) %>%
  select(ID, first_name, last_name, phone_number)

print(meat_lovers)
```

I can restructure information to be in a "tidy" format.

```
##      ID  first_name last_name phone_number
## 1  4931      Julian    Holmes   7382541246
## 2  5376     Maurice   Jackson   8625833781
## 3  1501     Brandon   Oliveri   8219751974
## 4  8475       Sahar     Ortiz   4527312178
## 5  1065     Kenneth el-Kaleel   1872672845
## 6  1619 Javier-Luis      Bohm   2761763421
## 7  4947       Amari      Knox   1869144965
## 8  3104        Carl      Todd   8544613465
## 9  9220     Krystin  Schuppan   2488179481
## 10 2109         Ana   Mmorosa   2582574153
```

```r
#Finding out contact information of top 10 customers with highest purchase of Sweet products
sweet_lovers <- meat_lovers <- merged_data %>%
  arrange(desc(MntSweetProducts)) %>%
  head(10) %>%
  select(ID, first_name, last_name, phone_number)

print(sweet_lovers)
```

```
##      ID first_name last_name phone_number
## 1  4619     Travis     Beaty   5287648129
## 2  8923    Khaalid   Cordova   9762383527
## 3  9264  Dominique       Her   8714631768
## 4  1964       Pari    Suarez   6147926912
## 5   961       Alex      Baca   1629251375
## 6   500     Andrew      Eden   4519621768
## 7  4947      Amari      Knox   1869144965
## 8  8370   Samantha     Thang   7195475947
## 9  2021       Evan     Terry   1232964637
## 10 7428     Victor     Black   5187191435
```

```
#Finding contact information of top 100 lapsed customers (customer who hasn't returned to shop in a whi
lapsed_customers <- merged_data %>%
  arrange(desc(Recency)) %>%
  head(100) %>%
  mutate(no_purchase_since = Recency) %>%
  select(ID, first_name, last_name, phone_number, no_purchase_since)

print(lapsed_customers %>% head(10))
```

```
##      ID first_name last_name phone_number no_purchase_since
## 1    22    Jessica      Berg   2714816374                99
## 2   528    Whitney     Avila   4572151826                99
## 3   868    Brandon    Teuton   2163166387                99
## 4  1743 Tachayanne  Lolmaugh   3859824316                99
## 5  2106    Karissa   Burgess   8962691365                99
## 6  2415       Nada      Akon   9769543276                99
## 7  2831    Saamiqa al-Naderi   2574377268                99
## 8  3363   Angelica     Davis   5943714615                99
## 9  4070      Turki Pelletier   5135963549                99
## 10 5263    Jessica      Bron   1249438276                99
```

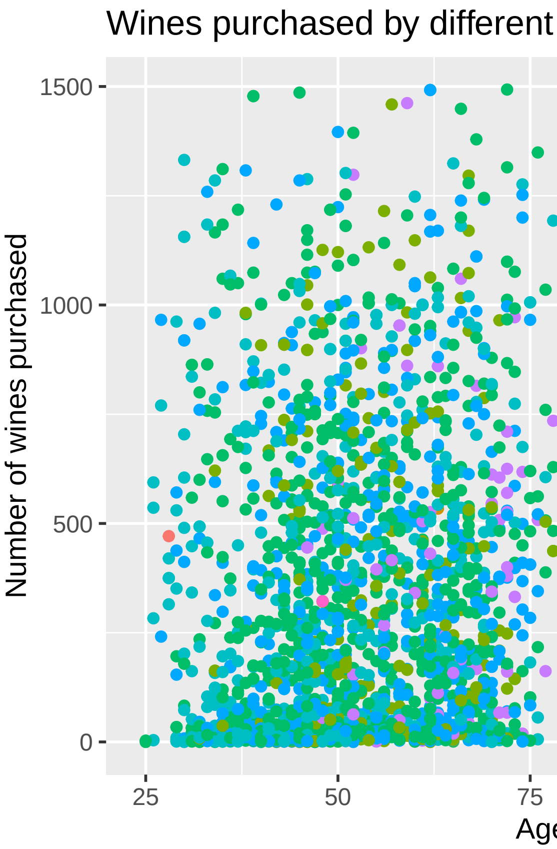
```
#Get mean of salaries as per education
educational_income_mean <-group_by(merged_data, Education) %>%
  summarise(mean_income = mean(Income)) %>%
  arrange(desc(mean_income))

print(educational_income_mean)
```

I can apply a function to groupings within a data source.

```
## # A tibble: 5 x 2
##   Education  mean_income
##   <chr>            <dbl>
## 1 PhD             56145.
## 2 Master          52918.
## 3 Graduation      52720.
## 4 2n Cycle        47633.
## 5 Basic           20306.
```

```
ggplot(data = merged_data) +
  geom_point(mapping = aes(x = age, y = MntWines,  color = Marital_Status)) +
  labs(title = "Wines purchased by different Age Groups", x = "Age", y = "Number of wines purchased")
```



I can create graphical displays of data that highlight key features.