Final Project - EDA

Keerthi Sreenivas Konjety, Vishnu Elangonvan

12/15/2021

```
mkt = read.csv("marketing_campaign.csv")
```

Abstract of my EDA Analysis:

1.Data Cleaning 2.Univariate Analysis - Questions Answered: ### 1.Which aged customers purchase highly? ### 2.Among all the products that we sell, which one is highly influenced by most of our customers? ### 3.How is the performance of the campaigns? 3.Bivariate Analysis - Questions Answered: ### Income and Expense variation based on Marital Status ### Education vs Category of Expenses

```
library(skimr)
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(DataExplorer)
library(plotly)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

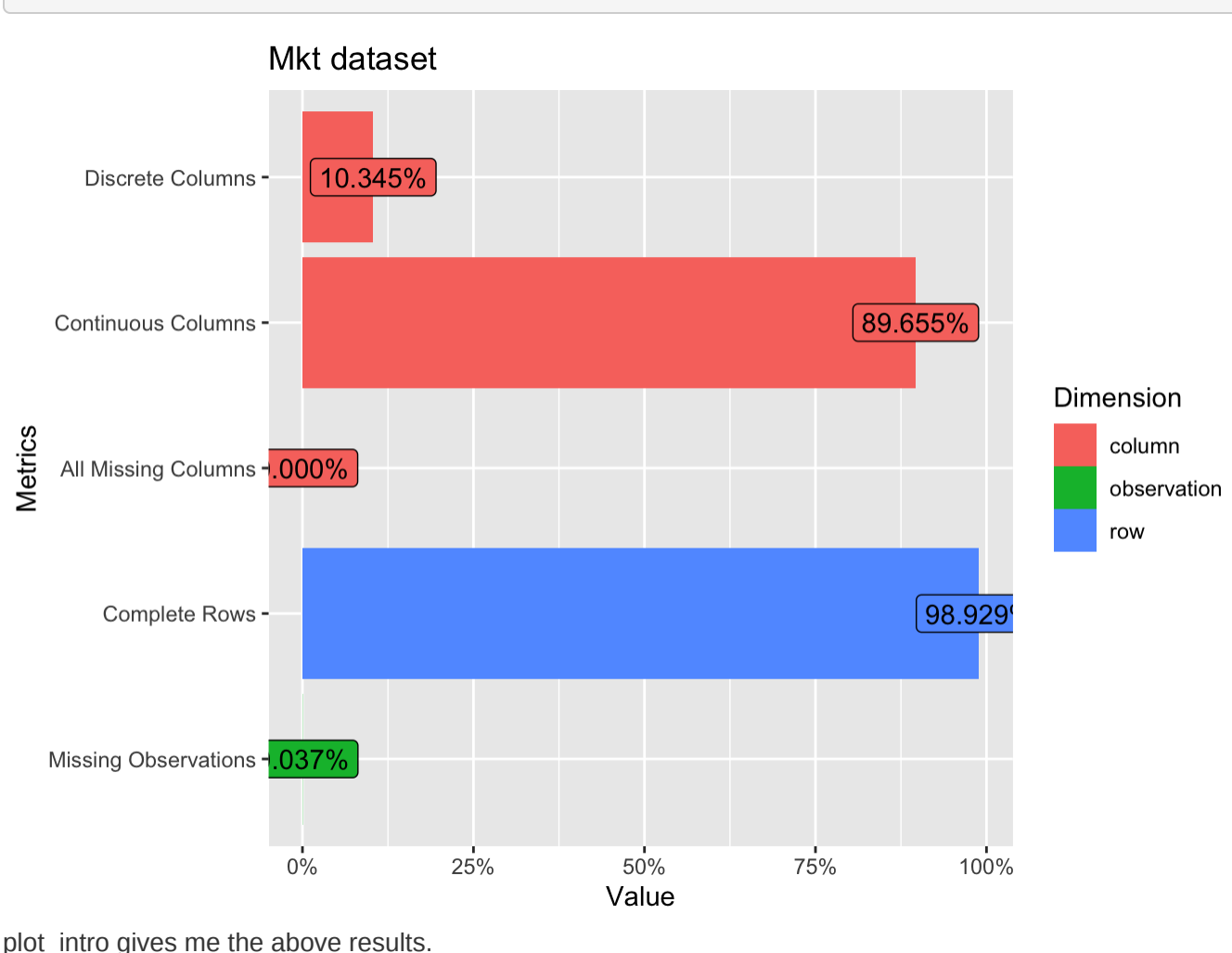
```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(ggplot2)
library(IRdisplay)
mkt = read.csv("marketing_campaign.csv")

mkt_df=data.frame(mkt)
copy_df = data.frame(mkt_df)
```

```
plot_intro(mkt_df,title='Mkt dataset')
```



Data Exploration function from

plot_intro gives me the above results.

```
head(mkt_df,5)
```

```
##     ID Year_Birth  Education Marital_Status Income Kidhome Teenhome Dt_Customer
## 1 5524       1957 Graduation         Single  58138       0        0      4/9/12
## 2 2174       1954 Graduation         Single  46344       1        1      8/3/14
## 3 4141       1965 Graduation       Together  71613       0        0  21-08-2013
## 4 6182       1984 Graduation       Together  26646       1        0     10/2/14
## 5 5324       1981        PhD        Married  58293       1        0  19-01-2014
##   Recency MntWines MntFruits MntMeatProducts MntFishProducts MntSweetProducts
## 1      58      635        88             546             172               88
## 2      38       11         1               6               2                1
## 3      26      426        49             127             111               21
## 4      26       11         4              20              10                3
## 5      94      173        43             118              46               27
##   MntGoldProds NumDealsPurchases NumWebPurchases NumCatalogPurchases
## 1           88                 3               8                  10
## 2            6                 2               1                   1
## 3           42                 1               8                   2
## 4            5                 2               2                   0
## 5           15                 5               5                   3
##   NumStorePurchases NumWebVisitsMonth AcceptedCmp3 AcceptedCmp4 AcceptedCmp5
## 1                 4                 7            0            0            0
## 2                 2                 5            0            0            0
## 3                10                 4            0            0            0
## 4                 4                 6            0            0            0
## 5                 6                 5            0            0            0
##   AcceptedCmp1 AcceptedCmp2 Complain Z_CostContact Z_Revenue Response
## 1            0            0        0             3        11        1
## 2            0            0        0             3        11        0
## 3            0            0        0             3        11        0
## 4            0            0        0             3        11        0
## 5            0            0        0             3        11        0
```

This gives me the first five rows of my data.

```
skim(mkt_df)
```

Data summary

Name	mkt_df
Number of rows	2240
Number of columns	29
_______________________	
Column type frequency:	
character	3
numeric	26
________________________	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Education	0	1	3	10	0	5	0
Marital_Status	0	1	4	8	0	8	0
Dt_Customer	0	1	6	10	0	663	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ID	0	1.00	5592.16	3246.66	0	2828.25	5458.5	8427.75	11191	▇▇▇▇▇
Year_Birth	0	1.00	1968.81	11.98	1893	1959.00	1970.0	1977.00	1996	▁▁▂▇▅
Income	24	0.99	52247.25	25173.08	1730	35303.00	51381.5	68522.00	666666	▇▁▁▁▁
Kidhome	0	1.00	0.44	0.54	0	0.00	0.0	1.00	2	▇▁▆▁▁
Teenhome	0	1.00	0.51	0.54	0	0.00	0.0	1.00	2	▇▁▇▁▁
Recency	0	1.00	49.11	28.96	0	24.00	49.0	74.00	99	▇▇▇▇▇
MntWines	0	1.00	303.94	336.60	0	23.75	173.5	504.25	1493	▇▂▂▁▁
MntFruits	0	1.00	26.30	39.77	0	1.00	8.0	33.00	199	▇▁▁▁▁
MntMeatProducts	0	1.00	166.95	225.72	0	16.00	67.0	232.00	1725	▇▁▁▁▁
MntFishProducts	0	1.00	37.53	54.63	0	3.00	12.0	50.00	259	▇▁▁▁▁
MntSweetProducts	0	1.00	27.06	41.28	0	1.00	8.0	33.00	263	▇▁▁▁▁
MntGoldProds	0	1.00	44.02	52.17	0	9.00	24.0	56.00	362	▇▁▁▁▁
NumDealsPurchases	0	1.00	2.33	1.93	0	1.00	2.0	3.00	15	▇▂▁▁▁
NumWebPurchases	0	1.00	4.08	2.78	0	2.00	4.0	6.00	27	▇▃▁▁▁
NumCatalogPurchases	0	1.00	2.66	2.92	0	0.00	2.0	4.00	28	▇▂▁▁▁
NumStorePurchases	0	1.00	5.79	3.25	0	3.00	5.0	8.00	13	▂▇▂▃▂
NumWebVisitsMonth	0	1.00	5.32	2.43	0	3.00	6.0	7.00	20	▅▇▁▁▁
AcceptedCmp3	0	1.00	0.07	0.26	0	0.00	0.0	0.00	1	▇▁▁▁▁
AcceptedCmp4	0	1.00	0.07	0.26	0	0.00	0.0	0.00	1	▇▁▁▁▁
AcceptedCmp5	0	1.00	0.07	0.26	0	0.00	0.0	0.00	1	▇▁▁▁▁
AcceptedCmp1	0	1.00	0.06	0.25	0	0.00	0.0	0.00	1	▇▁▁▁▁
AcceptedCmp2	0	1.00	0.01	0.11	0	0.00	0.0	0.00	1	▇▁▁▁▁
Complain	0	1.00	0.01	0.10	0	0.00	0.0	0.00	1	▇▁▁▁▁
Z_CostContact	0	1.00	3.00	0.00	3	3.00	3.0	3.00	3	▁▁▇▁▁
Z_Revenue	0	1.00	11.00	0.00	11	11.00	11.0	11.00	11	▁▁▇▁▁
Response	0	1.00	0.15	0.36	0	0.00	0.0	0.00	1	▇▁▁▁▂

Observations!

At the Initial observation - There are 2240 entries and 29 columns - In columns 'z_costcontact','z_revenue' all values are equal, we will drop this columns from the mkt. - No duplicated values and errors (at the first glance) - Only 'Income' column has missing values

```
#Filling the missing value with median value
mkt_df$Income[is.na(mkt_df$Income)]=median(mkt_df$Income,na.rm=T)
#Counting the missing value in Income column
sum(is.na(mkt_df$Income))
```

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

```
#Finding the no of unique values in each column
sapply(mkt_df, function(x) length(unique(x)))
```

```
##                  ID          Year_Birth           Education      Marital_Status
##                2240                  59                   5                   8
##              Income             Kidhome            Teenhome         Dt_Customer
##                1975                   3                   3                 663
##             Recency            MntWines           MntFruits     MntMeatProducts
##                 100                 776                 158                 558
##     MntFishProducts    MntSweetProducts        MntGoldProds   NumDealsPurchases
##                 182                 177                 213                  15
##     NumWebPurchases NumCatalogPurchases   NumStorePurchases   NumWebVisitsMonth
##                  15                  14                  14                  16
##        AcceptedCmp3        AcceptedCmp4        AcceptedCmp5        AcceptedCmp1
##                   2                   2                   2                   2
##        AcceptedCmp2            Complain       Z_CostContact           Z_Revenue
##                   2                   2                   1                   1
##            Response
##                   2
```

```
#Since the columns z_costcontact and z_revenue has one unique value, we will remove those columns
#Removing the unwanted columns
mkt_df=subset(mkt_df,select=-c(Z_CostContact,Z_Revenue))
```

```
#Old dimension of main_df is 2240 29
dim(mkt_df)
```

```
## [1] 2240   27
```

Univariate Analysis Qn 1-3

1.Which aged customers purchase highly?

```
current_date= Sys.Date()
current_year = format(current_date,format="%Y")
current_year = as.integer(current_year)
Age= c(current_year - mkt_df$Year_Birth)
#Adding Age column to main_df dataframe
mkt_df['Age']=Age
#Calculating maximum age of customers
max(mkt_df$Age)
```

```
## [1] 128
```

```
#Calculating minimum age of customers
min(mkt_df$Age)
```

```
## [1] 25
```

```
#Calculating average for age of customers
```

Calculating maximum age of customers is 128 Calculating minimum age of customers is 25

```
children = mkt_df %>% filter(Age < 15) %>% summarize(n())
young = mkt_df %>% filter(15 <= Age & Age <= 25) %>% summarize(n())
middle_aged = mkt_df %>% filter(25 < Age & Age <= 59) %>% summarize(n())
old  = mkt_df %>% filter( 60 <= Age & Age <=128 ) %>% summarize(n())


children = as.integer(children)
young = as.integer(young)
middle_aged = as.integer(middle_aged)
old = as.integer(old)

count = c(children,young,middle_aged,old)
labels_age = c('Children','Young','Middle Aged','Old')

children#No of Children
```

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

```
young#No of Children
```

```
## [1] 2
```

```
middle_aged#No of Children
```

```
## [1] 1583
```

```
old#No of Children
```

```
## [1] 655
```

Observations!

- 70% of customers in the data are middle aged
- Only 29% of customers in the date are old
- There is no significant amount of children and young customers ### 2.Among all the products that we sell, which one is highly influenced by most of our customers?

```
# Combining different dataframe into a single column to reduce the number of dimension

mkt_df['Expenses'] = mkt_df['MntWines'] + mkt_df['MntFruits'] + mkt_df['MntMeatProducts'] +
mkt_df['MntFishProducts'] + mkt_df['MntSweetProducts'] + mkt_df['MntGoldProds']

#Maximum Expenses
max(mkt_df$Expenses)
```

```
## [1] 2525
```

```
#Minimum Expenses
min(mkt_df$Expenses)
```

```
## [1] 5
```

```
#Average Expenses
mean(mkt_df$Expenses)
```

```
## [1] 605.7982
```

```
TotalExpense = sum(mkt_df$Expenses)
percent_w = (sum(mkt_df$MntWines)/TotalExpense)*100
percent_f = (sum(mkt_df$MntFruits)/TotalExpense)*100
percent_mp = (sum(mkt_df$MntMeatProducts)/TotalExpense)*100
percent_fp = (sum(mkt_df$MntFishProducts)/TotalExpense)*100
percent_sp = (sum(mkt_df$MntSweetProducts)/TotalExpense)*100
percent_gp = (sum(mkt_df$MntGoldProds)/TotalExpense)*100


percent_w
```

```
## [1] 50.17111
```

```
percent_f
```

```
## [1] 4.341748
```

```
percent_mp
```

```
## [1] 27.55868
```

```
percent_fp
```

```
## [1] 6.19438
```

```
percent_sp
```

```
## [1] 4.46732
```

```
percent_gp
```

```
## [1] 7.266755
```

```
per_ex = c(percent_w,percent_f,percent_mp,percent_fp,percent_sp,percent_gp)
labels_ex = c('MntWines','MntFruits','MntMeatProducts','MntFishProducts','MntSweetProducts','MntGoldProducts')
```

Percent_Wine is 50.17 Percent_Fruits is 4.34 Percent_MeatProducts is 27.55 Percent_FishProducts is 6.19 Percent_SweetProducts is 4.46 Percent_GoldProducts is 7.26

As we can see, people tend to spend more on wine products comparing to the rest of the products.

3.How is the performance of the campaigns?

```
#Finding unique values in AcceptedCmp1 column
unique(mkt_df$AcceptedCmp1)
```

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

```
##Same unique values for other AcceptedCmp variables


# Combining different dataframe into a single column to reduce the number of dimension

mkt_df['TotalAcceptedCmp'] = mkt_df['AcceptedCmp1'] + mkt_df['AcceptedCmp2'] + mkt_df['AcceptedCmp3'] +
mkt_df['AcceptedCmp4'] + mkt_df['AcceptedCmp5']
```

```
#Creating table for TotalAcceptedCmp column

table_ac = table(mkt_df$TotalAcceptedCmp)
per_ac = as.vector(prop.table(table_ac)*100)
labels_ac = c('0','1','2','3','4')
df_ac = data.frame(labels_ac,per_ac)
df_ac
```

```
##   labels_ac     per_ac
## 1         0 79.3303571
## 2         1 14.5089286
## 3         2  3.7053571
## 4         3  1.9642857
## 5         4  0.4910714
```

```
#Calculating percentage for each of the category in TotalAcceptedcmp column

TotalAcceptedOffers = sum(mkt_df$TotalAcceptedCmp)
percent_c1 = (sum(mkt_df$AcceptedCmp1)/TotalAcceptedOffers)*100
percent_c2 = (sum(mkt_df$AcceptedCmp2)/TotalAcceptedOffers)*100
percent_c3 = (sum(mkt_df$AcceptedCmp3)/TotalAcceptedOffers)*100
percent_c4 = (sum(mkt_df$AcceptedCmp4)/TotalAcceptedOffers)*100
percent_c5 = (sum(mkt_df$AcceptedCmp5)/TotalAcceptedOffers)*100

per_cmp = c(percent_c1,percent_c2,percent_c3,percent_c4,percent_c5)
labels_cmp = c('Campaign 1','Campaign 2','Campaign 3','Campaign 4','Campaign 5')
```

Observations!

- 79.33% of Customers accepted no offers in the campaigns
- 14.50% of Customers accepted only one offer in the campaigns
- 3.70% of Customers accepted two offer in the campaigns
- 1.96% of Customers accepted three offer in the campaigns
- 0.49% of Customers accepted four offer in the campaigns

Insights

1.Moreover, we observe that we don't have any customers who accepted all the five offers in the campaigns organized. 2.Having a high percentage about acceptance of no offers in the campaigns conducted shows that have to significantly improve the performance in the campaigns. 3.Most customers accepted the offers in the campaign 4 but only small amount of customers have accepted the offers in the 2nd campaign

Bivariate Analysis

```
#Creating a table for Education variable
table_ed = sort(table(mkt_df$Education),decreasing=T)
per_ed = as.vector((prop.table(table_ed)*100))
labels_ed = c('Graduation','PhD','Master','2n Cycle','Basic')
df_ed = data.frame(labels_ed,per_ed)
df_ed
```

```
##    labels_ed    per_ed
## 1 Graduation 50.312500
## 2        PhD 21.696429
## 3     Master 16.517857
## 4   2n Cycle  9.062500
## 5      Basic  2.410714
```

```
#Creating a table for Education variable
table_ms = sort(table(mkt_df$Marital_Status),decreasing=T)
per_ms = as.vector(prop.table(table_ms)*100)
labels_ms = c('Married','Together','Single','Divorced','Widow','Alone','Absurd','YOLO')
df_ms = data.frame(labels_ms,per_ms)
df_ms
```

```
##   labels_ms      per_ms
## 1   Married 38.57142857
## 2  Together 25.89285714
## 3    Single 21.42857143
## 4  Divorced 10.35714286
## 5     Widow  3.43750000
## 6     Alone  0.13392857
## 7    Absurd  0.08928571
## 8      YOLO  0.08928571
```

1.Income and Expense variation based on Marital Status

```
#Filtering the dataframe based on the categories of Marital Status column

married = mkt_df %>% filter(Marital_Status == 'Married') %>% select(Income, Expenses)
avg_married_ic = mean(married$Income)
avg_married_ex = mean(married$Expenses)

together = mkt_df %>% filter(Marital_Status == 'Together') %>% select(Income, Expenses)
avg_together_ic = mean(together$Income)
avg_together_ex = mean(together$Expenses)

single = mkt_df %>% filter(Marital_Status == 'Single') %>% select(Income, Expenses)
avg_single_ic = mean(single$Income)
avg_single_ex = mean(single$Expenses)

divorced = mkt_df %>% filter(Marital_Status == 'Divorced') %>% select(Income, Expenses)
avg_divorced_ic = mean(divorced$Income)
avg_divorced_ex = mean(divorced$Expenses)

widow = mkt_df %>% filter(Marital_Status == 'Widow') %>% select(Income, Expenses)
avg_widow_ic = mean(widow$Income)
avg_widow_ex = mean(widow$Expenses)

alone = mkt_df %>% filter(Marital_Status == 'Alone') %>% select(Income, Expenses)
avg_alone_ic = mean(alone$Income)
avg_alone_ex = mean(alone$Expenses)

absurd = mkt_df %>% filter(Marital_Status == 'Absurd') %>% select(Income, Expenses)
avg_absurd_ic = mean(absurd$Income)
avg_absurd_ex = mean(absurd$Expenses)

yolo = mkt_df %>% filter(Marital_Status == 'YOLO') %>% select(Income, Expenses)
avg_yolo_ic = mean(yolo$Income)
avg_yolo_ex = mean(yolo$Expenses)

avg_ms_ic=c(avg_married_ic,avg_together_ic ,avg_single_ic, avg_divorced_ic ,avg_widow_ic,avg_alone_ic,avg_absurd_
ic
,avg_yolo_ic)
avg_ms_ex=c(avg_married_ex,avg_together_ex ,avg_single_ex, avg_divorced_ex ,avg_widow_ex,avg_alone_ex,avg_absurd_
ex
,avg_yolo_ex)
df_ms_ic_ex = data.frame(labels_ms,avg_ms_ic,avg_ms_ex)
df_ms_ic_ex
```

```
##   labels_ms avg_ms_ic avg_ms_ex
## 1   Married  51722.20  590.8021
## 2  Together  53223.04  608.3879
## 3    Single  51002.59  606.4833
## 4  Divorced  52834.23  610.6293
## 5     Widow  56415.32  738.8182
## 6     Alone  43789.00  256.6667
## 7    Absurd  72365.50 1192.5000
## 8      YOLO  48432.00  424.0000
```

avg_ms_ic (Income Variable) avg_ms_ex (Expenses)

Observations!

- Customers with the marital status 'Absurd' has high income and they spend highly than the other customers.Looks like an Outlier though
- Customers who are alone, have low income and spendings. The reason might be these type of customers are too old or too young so that they cannot earn lot of money

2.Education vs Category of Expenses

```
#Filtering the dataframe based on the categories of Education column

graduation = mkt_df %>% filter(Education == 'Graduation') %>% select(MntWines,MntFruits, MntMeatProducts,MntSweet
Products,MntFishProducts,MntGoldProds)
avg_graduation_mw = mean(graduation$MntWines)
avg_graduation_mf = mean(graduation$MntFruits)
avg_graduation_mp = mean(graduation$MntMeatProducts)
avg_graduation_fp = mean(graduation$MntFishProducts)
avg_graduation_sp = mean(graduation$MntSweetProducts)
avg_graduation_gp = mean(graduation$MntGoldProds)

phd = mkt_df %>% filter(Education == 'PhD') %>% select(MntWines,MntFruits, MntMeatProducts,MntSweetProducts,MntFi
shProducts,MntGoldProds)
avg_phd_mw = mean(phd$MntWines)
avg_phd_mf = mean(phd$MntFruits)
avg_phd_mp = mean(phd$MntMeatProducts)
avg_phd_fp = mean(phd$MntFishProducts)
avg_phd_sp = mean(phd$MntSweetProducts)
avg_phd_gp = mean(phd$MntGoldProds)


master = mkt_df %>% filter(Education == 'Master') %>% select(MntWines,MntFruits, MntMeatProducts,MntSweetProduct
s,MntFishProducts,MntGoldProds)
avg_master_mw = mean(master$MntWines)
avg_master_mf = mean(master$MntFruits)
avg_master_mp = mean(master$MntMeatProducts)
avg_master_fp = mean(master$MntFishProducts)
avg_master_sp = mean(master$MntSweetProducts)
avg_master_gp = mean(master$MntGoldProds)

second_cycle = mkt_df %>% filter(Education == '2n Cycle') %>% select(MntWines,MntFruits, MntMeatProducts,MntSweet
Products,MntFishProducts,MntGoldProds)
avg_second_cycle_mw = mean(second_cycle$MntWines)
avg_second_cycle_mf = mean(second_cycle$MntFruits)
avg_second_cycle_mp = mean(second_cycle$MntMeatProducts)
avg_second_cycle_fp = mean(second_cycle$MntFishProducts)
avg_second_cycle_sp = mean(second_cycle$MntSweetProducts)
avg_second_cycle_gp = mean(second_cycle$MntGoldProds)

basic = mkt_df %>% filter(Education == 'Basic') %>% select(MntWines,MntFruits, MntMeatProducts,MntSweetProducts,M
ntFishProducts,MntGoldProds)
avg_basic_mw = mean(basic$MntWines)
avg_basic_mf = mean(basic$MntFruits)
avg_basic_mp = mean(basic$MntMeatProducts)
avg_basic_fp = mean(basic$MntFishProducts)
avg_basic_sp = mean(basic$MntSweetProducts)
avg_basic_gp = mean(basic$MntGoldProds)

avg_ed_mw=c(avg_graduation_mw,avg_phd_mw ,avg_master_mw, avg_second_cycle_mw ,avg_basic_mw)
avg_ed_mf=c(avg_graduation_mf,avg_phd_mf,avg_master_mf, avg_second_cycle_mf ,avg_basic_mf)
avg_ed_mp=c(avg_graduation_mp,avg_phd_mp ,avg_master_mp, avg_second_cycle_mp ,avg_basic_mp)
avg_ed_fp=c(avg_graduation_fp,avg_phd_fp ,avg_master_fp, avg_second_cycle_fp ,avg_basic_fp)
avg_ed_sp=c(avg_graduation_sp,avg_phd_sp ,avg_master_sp, avg_second_cycle_sp ,avg_basic_sp)
avg_ed_gp=c(avg_graduation_gp,avg_phd_gp ,avg_master_gp, avg_second_cycle_gp ,avg_basic_gp)
df_ed_prods = data.frame(labels_ed,avg_ed_mw,avg_ed_mf,avg_ed_mp,avg_ed_fp,avg_ed_sp,avg_ed_gp)
df_ed_prods
```

```
##    labels_ed  avg_ed_mw avg_ed_mf avg_ed_mp avg_ed_fp avg_ed_sp avg_ed_gp
## 1 Graduation 284.268855  30.77462 179.48891  43.14996  31.36735  50.84916
## 2        PhD 404.495885  20.04938 168.60288  26.72840  20.22222  32.31070
## 3     Master 333.075676  21.65405 163.37838  32.10000  21.17568  40.39730
## 4   2n Cycle 198.182266  28.95567 141.25616  47.48276  34.25123  46.39901
## 5      Basic   7.240741  11.11111  11.44444  17.05556  12.11111  22.83333
```

Observations!

avg_ed_mw Wines avg_ed_mf Fruits avg_ed_mp MeatProducts avg_ed_fp FishProducts avg_ed_sp SweetProducts avg_ed_gp GoldProds - Customers with a PhD degree tend to buy wine products largely than the other educated customers - Except the products wine and meat, customers tend to spend very little amount on all the other products - Customers with a basic education tend to spend very little amount to buy the products