CUSTOMER PERSONALITY ANALYSIS

ABSTRACT:

Based on detailed data of company A' s customers, this report analysis and predict **customer response** to promotion campaigns, **customer preference** to wines and meat products and **amount of customer spent**. According the prediction, the report provides a **recommendation system** to realize personalized recommendation and promotion. Meanwhile, **customer cluster** has been done to help company A segment their customers into **3** groups and corresponding suggestions are also provided.

CONTENTS:

1.INTRODUCTION

DATA BACKFGROUND:

Resource link: https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis

Problem statement:

This dataset is a detailed analysis of a Company A’ s customers. It helps a business to better understand its customers and makes it easier for them to modify products according to the specific needs, behaviors and concerns of different types of customers. It collected **29** variables of **2240** customers, the attributes of the variables are attached to the appendix. There are 3 problems set to help Company A achieve better promotion and recommendation performance.

Problem1: How customers response to the promotion campaign? Is the campaign efficient? （Part 1）

Problem2: What’ s customers’ preference? Which product should Company A recommend to them? (Part 2)

Problem3: How to segment these customer? (Part 3)

2.DATA CLEANING AND PROCESSING

DATA CLEANING

DATA READING AND REMOVE NA: remove 24 rows with missing data

REDESIGN VARIABLES:

Age: change Year_Birth to age

Dt_Customer: change to days the customer has been to the company

Marital_Status: 8 levels to 2 levels; combine ‘Absurd Alone Divorced Single Widow YOLO’ to 1; combine ‘Married. Together’ to 2

Education: 5 levels to 4 levels; combine ‘2n Cycle’ & ‘basic’ to ‘undergraduate’

ADD VARIABLES:

ttlspend: total money spent on the 6 products

ttlnum: total numbers of purchase the customer made

child: = kid home + teen home

familymember: = child + Marital_Status

CREATE DUMMY VARIABLES:

Education : to Education_Master; Education_PhD; Education_Undergraduate

REMOVE VARIABLES:

Remove: ID, Year_Birth, Kidhome, Teenhome, Z_CostContact, Z_Revenue (figure 1)

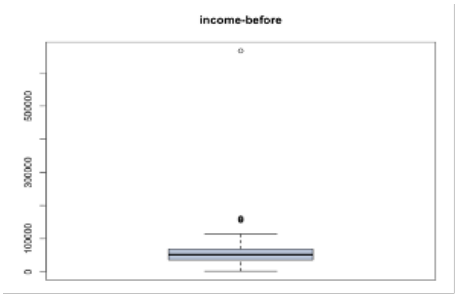
<u>After data cleaning, we now have data of **2216** rows and **30** variables.</u>

DATA PROCESSING:

REMOVE OUTLINERS:

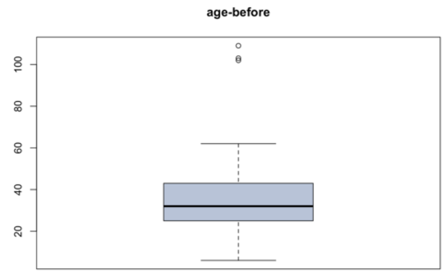
Income:

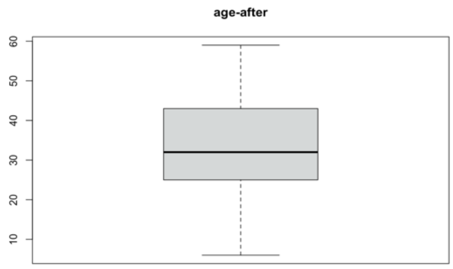
remove those whose income>= 15,000



Age:

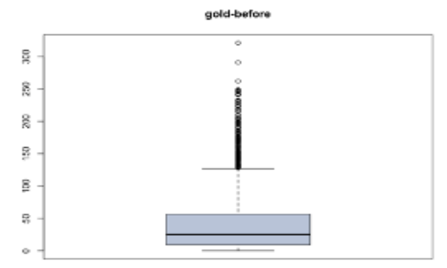
remove those whose age >=100

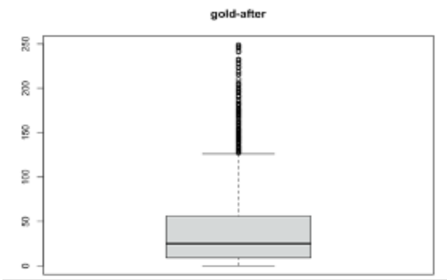




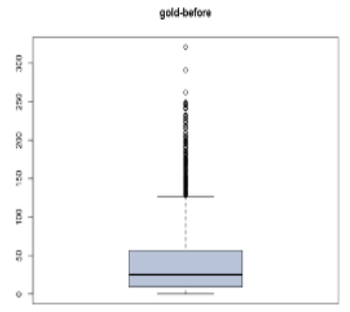
Gold:

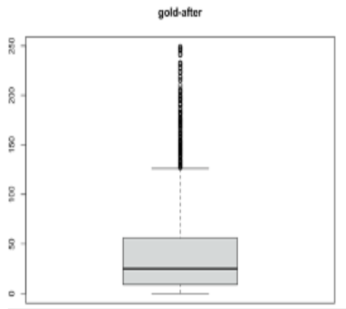
remove 'MntGoldProds' >= 250



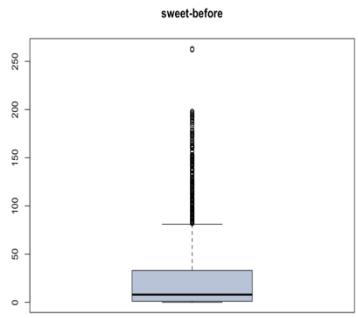


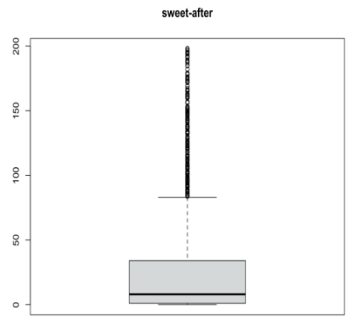
Gold: remove those whose 'MntGoldProds' >= 250





Sweat: remove those whose 'MntSweetProducts' >= 200



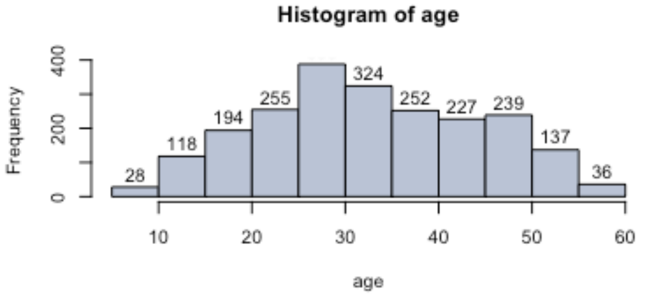


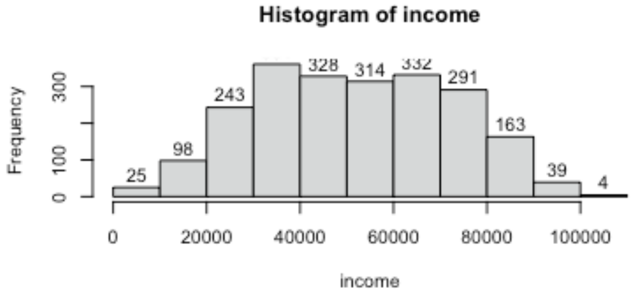
After data processing, we now have data of **2198** rows and **30** variables, which can be divided into **3** groups to help our further analysis.

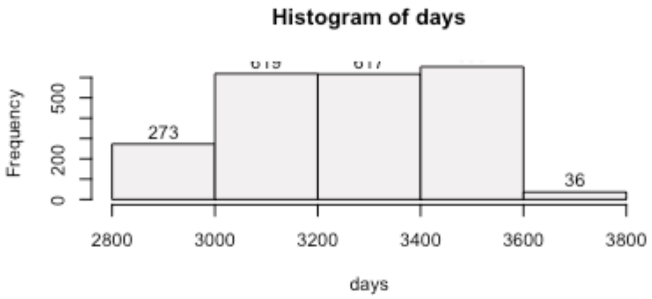
1. **Demographic** (11)：
   age, Education(3 dummies), Marital_Status, Income, child, family, Dt_Customer, Recency, complain
2. **Behaviors**：
   a) Products (7): MntWine, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds, ttlspend
   b) Promotion (7): NumDealsPurchases, AcceptedCmp1, AcceptedCmp2, AcceptedCmp3, AcceptedCmp4, AcceptedCmp5, Response
3. **Distribution** (5): NumWebPurchases, NumCatalogPurchases, NumStorePurchases, NumWebVisitsMonth, tllnum

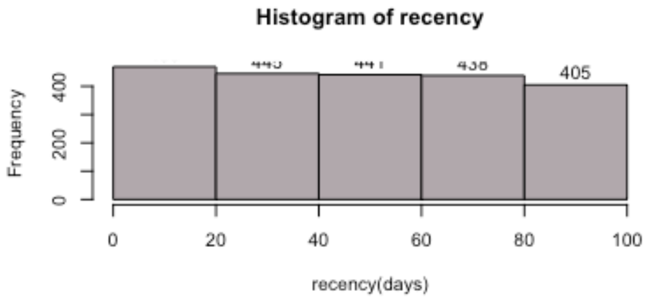
3.DESCRIPTIVE STATISTIC

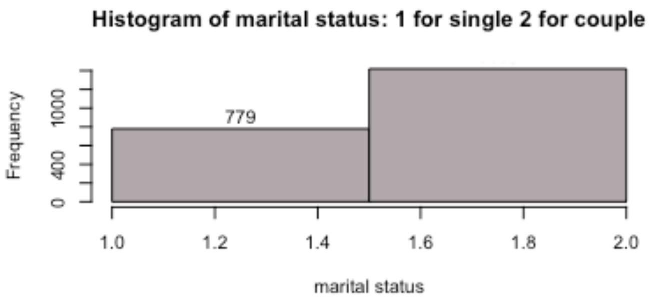
Group1 : Demographic

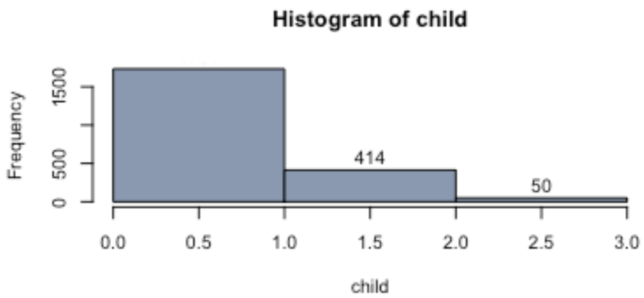












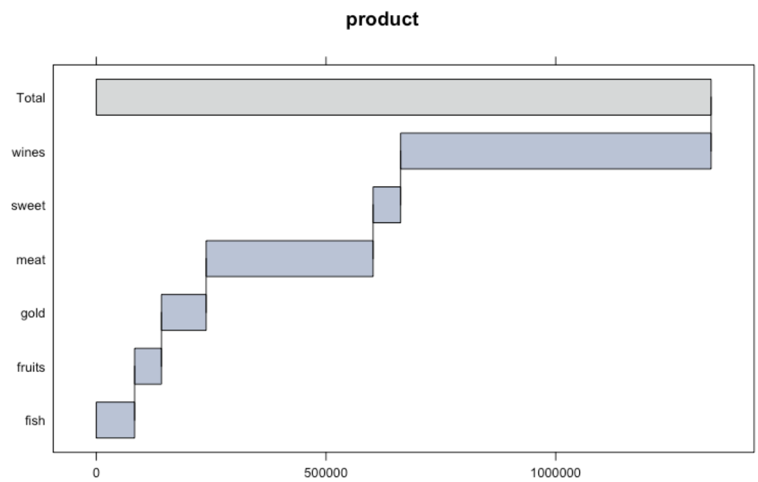
Group2.a: Behaviors--Product

We divide 6 products into **2** groups

according to the purchase amount:

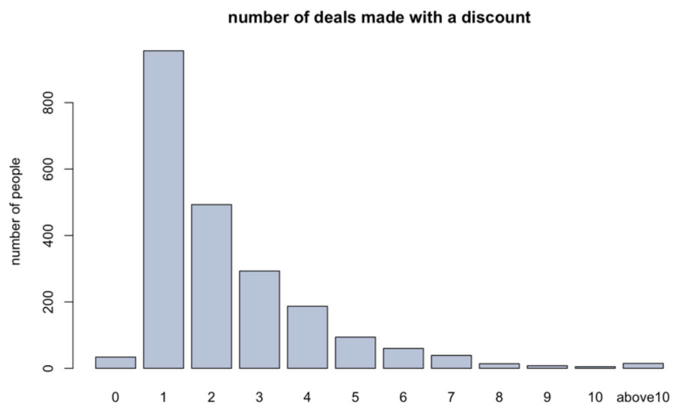
ace product: wines & meat

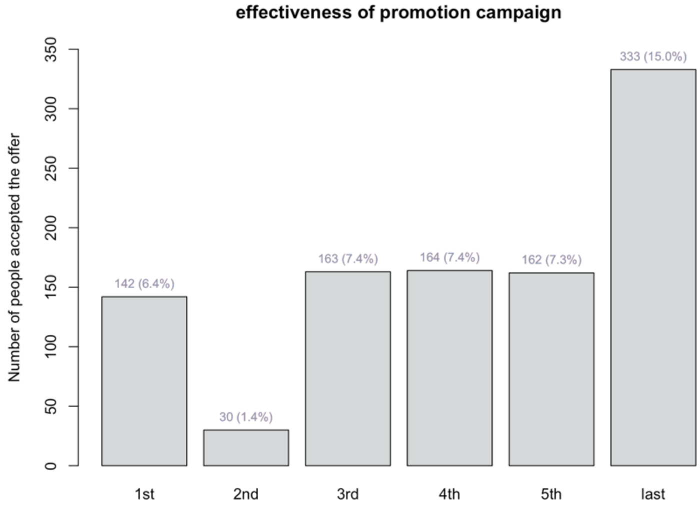
normal product: sweet & gold & fruits & fish



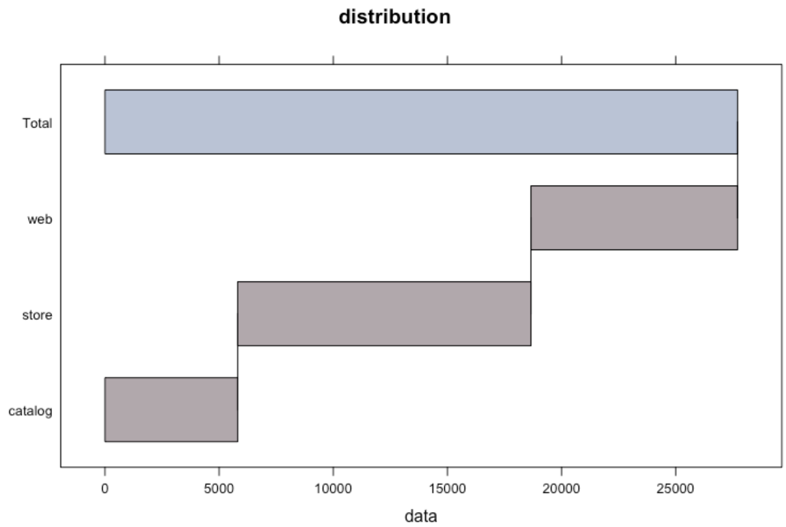
Group2.b: Behaviors—Promotion

Percentage: numbers of those who accepted the offer / total amount





Group 3: Distribution



The traditional **store** distribution is the most contributed one, followed by websites and catalog.

4.CONSUMER BEHAVIOR PREDICTION

We want to use data analysis to predict whether the customer will accept the offer when the promotion campaign is made **(Response-dependent variables)** , thus to help decide whether we should give the customer a discount or do the promotion.

We randomly select 30% of the data to be the test set, and the rest 70% to be the training set.

Key features used:

1. Accuracy: % correct prediction
2. Sensitivity: true positive rate
3. AIC: Akaike Information Criterion
4. Specificity: rue negative rate

GENERALIZED LINEAR REGRESSION

1. FULL MODEL:

Formula: Response ~ .

Summary of the regression result :

```
Call:
glm(formula = Response ~ ., family = "binomial", data = train.data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.36183  -0.41208  -0.20146  -0.08322   2.98000

Coefficients: (3 not defined because of singularities)
                      Estimate    Std. Error z value            Pr(>|z|)
(Intercept)        -15.86489251   2.01357475  -7.879  0.0000000000000033
Marital_Status      -1.45107483   0.19723202  -7.357  0.0000000000001878
Income               0.00001098   0.00001151   0.954            0.339836
Dt_Customer          0.00496497   0.00058440   8.496 < 0.0000000000000002
Recency             -0.03401140   0.00370109  -9.190 < 0.0000000000000002
MntWines            -0.00119915   0.00051496  -2.329            0.019879
MntFruits            0.00142554   0.00292663   0.487            0.626194

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1352.16  on 1537  degrees of freedom
Residual deviance:  773.45  on 1511  degrees of freedom
AIC: 827.45

Number of Fisher Scoring iterations: 6
```

Confusion Matrix :

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 548  40
         1  25  47

               Accuracy : 0.9015
                 95% CI : (0.8762, 0.9232)
    No Information Rate : 0.8682
    P-Value [Acc > NIR] : 0.005291

                  Kappa : 0.5358

 Mcnemar's Test P-Value : 0.082478

            Sensitivity : 0.54023
            Specificity : 0.95637
         Pos Pred Value : 0.65278
         Neg Pred Value : 0.93197
             Prevalence : 0.13182
         Detection Rate : 0.07121
   Detection Prevalence : 0.10909
      Balanced Accuracy : 0.74830

       'Positive' Class : 1
```

Key features: AIC: 827.45 Accuracy: 0.9015 Sensitivity: 0.54023 Specificity: 0.95637

2. REDUCED MODEL:

Formula: Response ~ Marital_Status + Dt_Customer + Recency + MntWines + MntMeatProducts + NumDealsPurchases + NumCatalogPurchases + NumStorePurchases + NumWebVisitsMonth + AcceptedCmp3 + AcceptedCmp4 + AcceptedCmp5 + AcceptedCmp1 + AcceptedCmp2 + child + Education_PhD + Education_undergraduate

Variables selection : backward elimination

```
                          Df Deviance    AIC
<none>                         779.73 815.73
- NumWebVisitsMonth        1   782.70 816.70
- child                    1   782.94 816.94
- MntWines                 1   783.17 817.17
- NumDealsPurchases        1   786.09 820.09
- Education_undergraduate  1   786.62 820.62
- AcceptedCmp2             1   787.50 821.50
- AcceptedCmp4             1   791.78 825.78
- Education_PhD            1   793.34 827.34
- MntMeatProducts          1   795.92 829.92
- NumCatalogPurchases      1   797.86 831.86
- AcceptedCmp1             1   800.26 834.26
- NumStorePurchases        1   810.80 844.80
- AcceptedCmp5             1   815.96 849.96
- Marital_Status           1   836.68 870.68
- AcceptedCmp3             1   841.56 875.56
- Dt_Customer              1   866.56 900.56
- Recency                  1   883.40 917.40
```

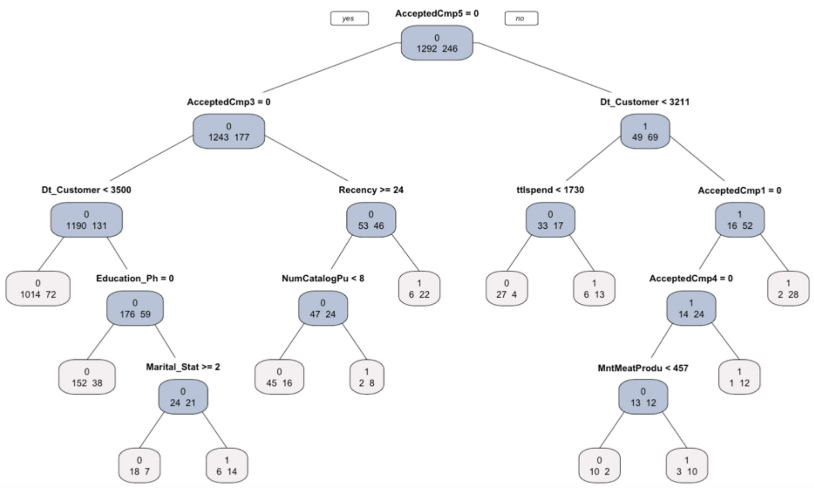
Confusion Matrix :

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 548  39
         1  25  48

               Accuracy : 0.903
                 95% CI : (0.8779, 0.9245)
    No Information Rate : 0.8682
    P-Value [Acc > NIR] : 0.003668

                  Kappa : 0.5453

 Mcnemar's Test P-Value : 0.104163

            Sensitivity : 0.55172
            Specificity : 0.95637
```

Key features: AIC: 815.73 Accuracy: 0.903 Sensitivity: 0.55172 Specificity: 0.95637

DEFAULT REGRESSION TREE

Tree Plot :



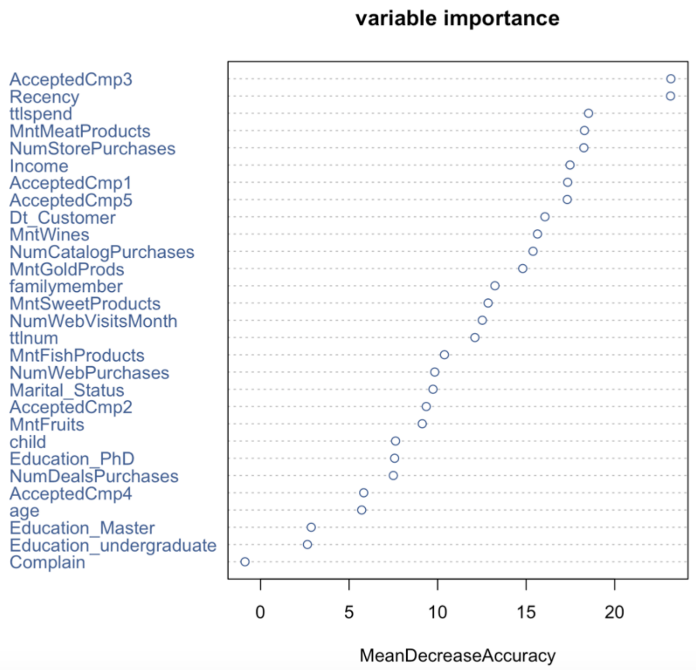
Confusion Matrix :

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 556  61
         1  17  26

               Accuracy : 0.8818
                 95% CI : (0.8547, 0.9055)
    No Information Rate : 0.8682
    P-Value [Acc > NIR] : 0.1641

                  Kappa : 0.3427

 Mcnemar's Test P-Value : 0.000001123

            Sensitivity : 0.29885
            Specificity : 0.97033
         Pos Pred Value : 0.60465
         Neg Pred Value : 0.90113
             Prevalence : 0.13182
         Detection Rate : 0.03939
   Detection Prevalence : 0.06515
      Balanced Accuracy : 0.63459

       'Positive' Class : 1
```

Key features: Accuracy: 0.8818 Sensitivity: 0.29885 Specificity: 0.97033

RANDOM FOREST

Variable Importance Plot :



Confusion Matrix :

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 559  53
         1  14  34

               Accuracy : 0.8985
                 95% CI : (0.8729, 0.9205)
    No Information Rate : 0.8682
    P-Value [Acc > NIR] : 0.0105

                  Kappa : 0.4524

 Mcnemar's Test P-Value : 0.000003443

            Sensitivity : 0.39080
            Specificity : 0.97557
         Pos Pred Value : 0.70833
         Neg Pred Value : 0.91340
             Prevalence : 0.13182
         Detection Rate : 0.05152
   Detection Prevalence : 0.07273
      Balanced Accuracy : 0.68319

       'Positive' Class : 1
```

Key features: Accuracy: 0.8985 Sensitivity: 0.39080 Specificity: 0.97557

MODEL COMPARISON

We can evaluate the efficiency of these models above by comparing their AIC, accuracy, sensitivity, specificity.

The comparison table:

	Full linear regression	Reduced linear regression	Default tree	Random forest
AIC	827.45	815.73	NA	NA
Accuracy	0.9015	0.903	0.8818	0.8985
Sensitivity	0.54023	0.55172	0.29885	0.39080
Specificity	0.95637	0.95637	0.97033	0.97557

Conclusion:

According to the table, we can see that the reduced linear regression model has the highest accuracy of 90.3%, the highest the sensitivity of 55,2%. It is also among the highest specificity. So we should choose the **reduced linear regression model** as our algorithms to predict consumer behavior.

Using the model, if the customer’s ‘response’ to the promotion campaign is ‘1’, which means promotion is efficient to this customer, than Company A should push the promotion campaign to him/her. Otherwise, Company A shouldn’t take the action.

5.CONSUMER PREFERENCE PREDICTION

After predicted whether we should push the promotion campaign, Company A would likely to know which of the 6 kinds of products is the one the customer most likely to purchase.

Model used: reduced linear regression (backward elimination)

Dependent variable (group2.a)：

MntWine, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds, ttlspend

	Wine	Meat	Fruit	Fish	Sweet	Gold	ttlspend
multiple R^2	0.6856	0.6517	0.389	0.4553	0.4171	0.4234	0.8262
adjusted R^2	0.6838	0.6499	0.385	0.4517	0.4129	0.4196	0.8253

After regression of the 7 variables separately, we found that the models of fruit\fish\sweet\gold don’t performance well, but the models of wine\meat/ttlspend are quite good. So we notice that the information provided **cannot** predict one’s preference on **normal products** (fruit, fish, sweet, gold), but can have a **good** prediction on **ace products** (wines , meat) and the **total amount** one would spend.

Following, let’s see the efficient models in details.

PREDICTION OF MNTWINES (AMOUNT SPENT ON WINE IN LAST 2 YEARS)

Formula: wine ~ Income + Dt_Customer + NumWebPurchases + NumCatalogPurchases + NumStorePurchases + NumWebVisitsMonth + child + Education_Master + Education_PhD

Summary of the regression result :

```
Call:
lm(formula = wine ~ Income + Dt_Customer + NumWebPurchases + 
    NumCatalogPurchases + NumStorePurchases + NumWebVisitsMonth + 
    child + Education_Master + Education_PhD, data = train.data)

Residuals:
    Min      1Q  Median      3Q     Max 
-792.03 -109.09  -10.91   82.04  913.26 

Coefficients:
                        Estimate  Std. Error t value             Pr(>|t|)    
(Intercept)         -907.2200163  84.0498614 -10.794 < 0.0000000000000002 ***
Income                 0.0083699   0.0004682  17.878 < 0.0000000000000002 ***
Dt_Customer            0.1189641   0.0263968   4.507   0.00000708376403745 ***
NumWebPurchases        9.4816695   2.6884575   3.527             0.000433 ***
NumCatalogPurchases   29.9296427   2.8445045  10.522 < 0.0000000000000002 ***
NumStorePurchases     18.0999787   2.2487494   8.049   0.00000000000000166 ***
NumWebVisitsMonth     33.3301922   3.4682962   9.610 < 0.0000000000000002 ***
child                -39.5864064   7.6390088  -5.182   0.00000024861529493 ***
Education_Master      62.1502598  13.3327646   4.661   0.00000341291978295 ***
Education_PhD         95.5450785  12.3179283   7.757   0.00000000000001584 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 189.1 on 1528 degrees of freedom
Multiple R-squared:  0.6856,	Adjusted R-squared:  0.6838 
F-statistic: 370.3 on 9 and 1528 DF,  p-value: < 0.00000000000000022
```

Predicted	Actual	Residual
415	173	-242
202	76	-126
160	14	-146
353	194	-159
244	84	-160
714	1012	298
590	867	277
-153	8	161
-38	6	44
-203	3	206

PREDICTION OF MNTMEAT (AMOUNT SPENT ON MEAT IN LAST 2 YEARS)

Formula: meat ~ Income + Dt_Customer + NumWebPurchases + NumCatalogPurchases + NumStorePurchases + age + child + Education_PhD

Predicted	Actual	Residual
178	118	-60
12	56	44
-8	24	32
303	480	177
127	38	-89
287	498	211
184	86	-98
-36	10	46
-2	14	16
-113	10	123

Summary of the regression result :

```
Call:
lm(formula = meat ~ Income + Dt_Customer + NumWebPurchases + 
    NumCatalogPurchases + NumStorePurchases + age + child + Education_PhD, 
    data = train.data)

Residuals:
    Min      1Q  Median      3Q     Max 
-513.14  -71.94  -10.38   48.67  522.39 

Coefficients:
                        Estimate  Std. Error t value             Pr(>|t|)    
(Intercept)         -302.2482157  57.6574482  -5.242          0.000000181 ***
Income                 0.0053157   0.0002779  19.131 < 0.0000000000000002 ***
Dt_Customer            0.0809520   0.0169061   4.788          0.000001845 ***
NumWebPurchases       -3.7055854   1.6105025  -2.301              0.02153 *  
NumCatalogPurchases   22.3429734   1.9010772  11.753 < 0.0000000000000002 ***
NumStorePurchases     -2.6732388   1.4950113  -1.788              0.07396 .  
age                   -0.9418859   0.2935538  -3.209              0.00136 ** 
child                -67.3331085   5.0875429 -13.235 < 0.0000000000000002 ***
Education_PhD        -24.1966276   8.1075249  -2.984              0.00289 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 128.3 on 1529 degrees of freedom
Multiple R-squared:  0.6517,	Adjusted R-squared:  0.6499 
F-statistic: 357.6 on 8 and 1529 DF,  p-value: < 0.00000000000000022
```

PERDICTION OF TTLSPEND (TOTAL AMOUNT SPENT ON COMPANY A' S PRODUCTS)

Formula: ttlspend ~ Income + Dt_Customer + NumWebPurchases + NumCatalogPurchases + NumStorePurchases + NumWebVisitsMonth + age + child

Summary of the regression result :

```
Call:
lm(formula = ttlspend ~ Income + Dt_Customer + NumWebPurchases +
    NumCatalogPurchases + NumStorePurchases + NumWebVisitsMonth +
    age + child, data = train.data)

Residuals:
     Min       1Q   Median       3Q      Max
-1310.64  -145.14   -19.55   117.00  1153.34

Coefficients:
                        Estimate    Std. Error t value               Pr(>|t|)
(Intercept)         -1270.0414017  114.4179994 -11.100 < 0.0000000000000002 ***
Income                  0.0148579    0.0006367  23.336 < 0.0000000000000002 ***
Dt_Customer             0.2585922    0.0356874   7.246   0.000000000000678 ***
NumWebPurchases        10.5902475    3.6136451   2.931             0.00343 **
NumCatalogPurchases    68.2663174    3.8579205  17.695 < 0.0000000000000002 ***
NumStorePurchases      19.8649863    3.0719240   6.467   0.000000000134489 ***
NumWebVisitsMonth      20.3805091    4.7074912   4.329   0.000015924753669 ***
age                    -1.5727677    0.5829754  -2.698             0.00706 **
child                -137.5313042   10.4031413 -13.220 < 0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 254.2 on 1529 degrees of freedom
Multiple R-squared:  0.8262,    Adjusted R-squared:  0.8253
F-statistic: 908.6 on 8 and 1529 DF,  p-value: < 0.00000000000000022
```

Predicted	Actual	Residual
1468	1617	149
960	716	-244
1551	1315	-236
502	317	-185
206	131	-75
331	302	-29
25	81	56
200	67	-133
-336	31	367
960	1319	359

6.RECOMMENDATION SYSTEM BASED ON THE PREDICTION

We assume that if **the predicted amount spent** is above the **0.75 quantile** of the original dataset, then the customer has preference on the product, and we should recommend this product to him/her.

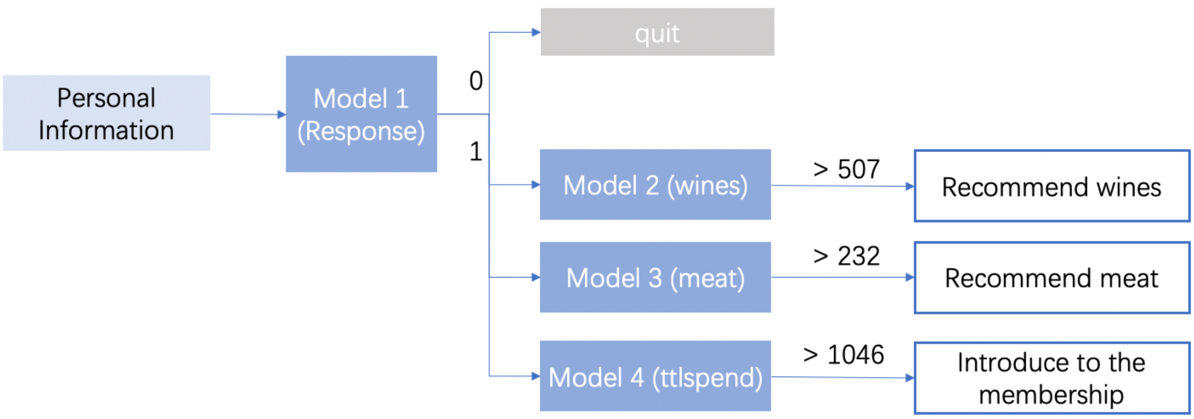
Models used: Model 1: reduced generalized linear regression of **Response**

Model 2: reduced linear regression of **MntWines (0.75 quantile=507)**

Model 3: reduced linear regression of **MntMeatProducts (0.75 quantile=232)**

Model 4: reduced linear regression of **ttlspend (0.75 quantile=1046)**

Recommendation logistic:



7.CUSTOMER CLUSTERING & MARKET SEGMENT

K-MEANS CLUSTERING

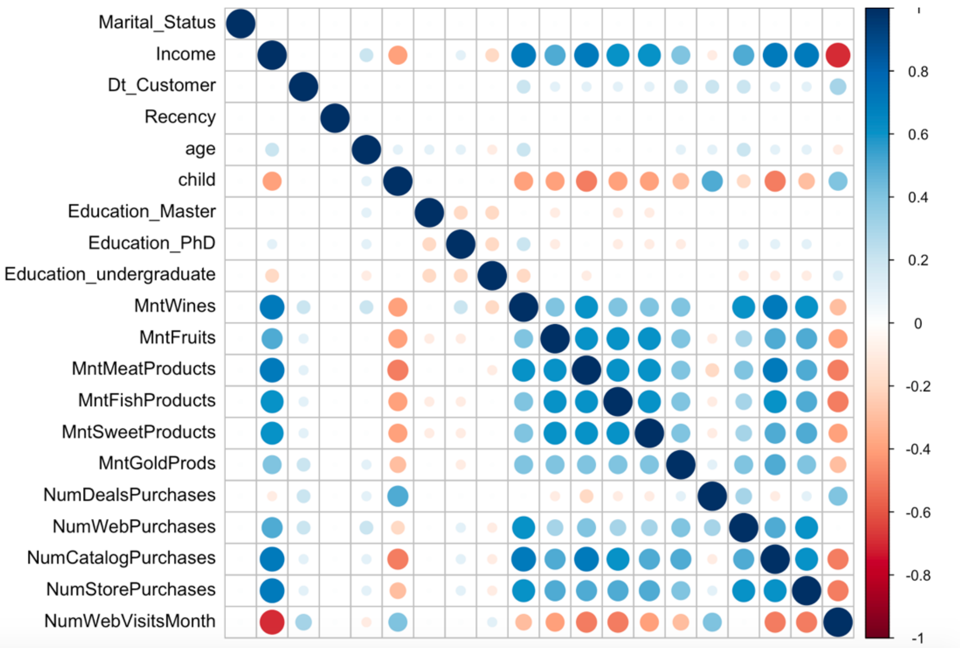
SELECT VARIABLES:

Group1: age, Education(3 dummies), Marital_Status, Income, child, family, Dt_Customer, Recency (10)

Group2.a: MntWine, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds (6)

Group3: NumWebPurchases, NumCatalogPurchases, NumStorePurchases, NumWebVisitsMonth (4)

Correlation table:



Kappa: 25.29492

After standardizing the data, the kappa of it is 25.29492.

Correlations between the variables are not so great.

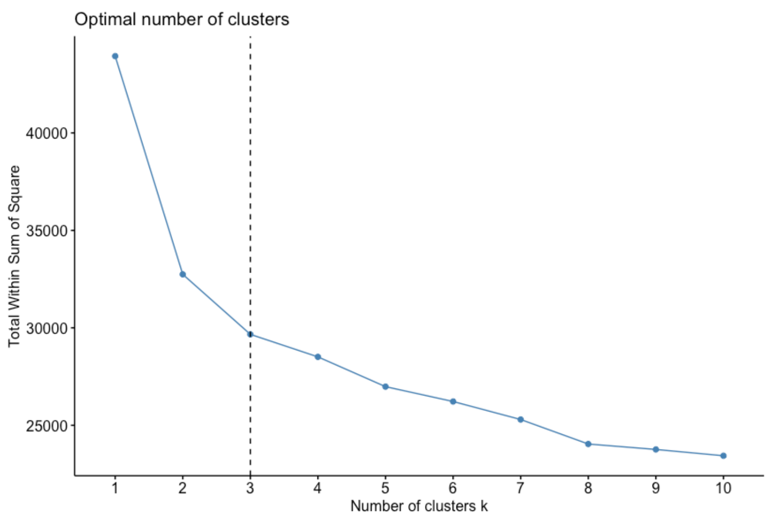
K-MEANS CLUSTERING:

1. **Determine cluster number**

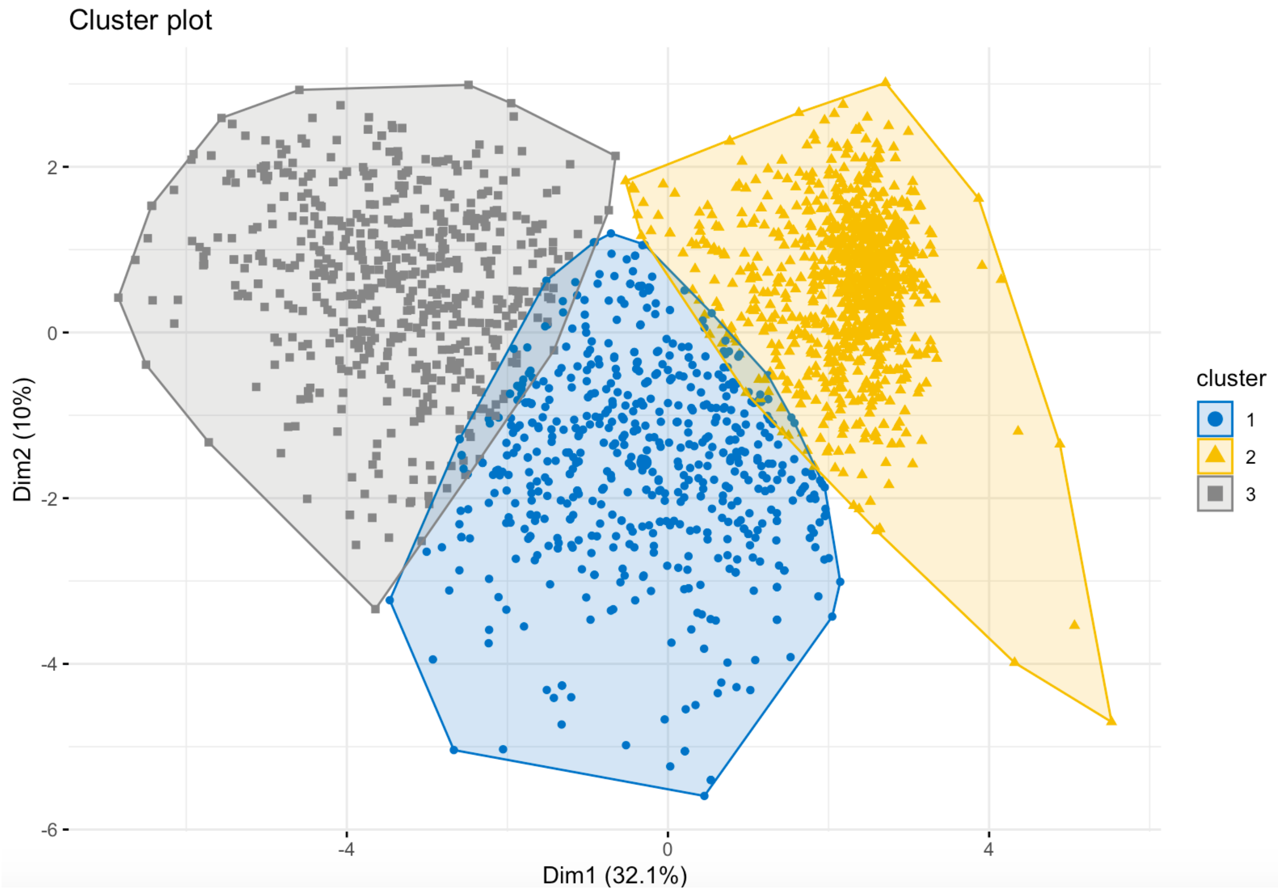
Looking at the elbow plot, we can see that it’ s decline tends to be gender from 3.

So, we choose to cluster the customers into **three** groups.

Elbow plot:



2. K-means clustering



Group	Marital_Status	Income	Age	Child	Master	PhD	Undergraduate
1	0.03	0.26	0.33	0.34	0.07	0.2	-0.2
2	0.01	-0.84	-0.23	0.37	-0.01	-0.09	0.15
3	-0.05	1.14	0.06	-0.94	-0.06	-0.05	-0.06
	Wines	**Fruits**	**Meat**	**Fish**	**Sweet**	**Gold**	
1	0.45	-0.17	-0.16	-0.21	-0.18	0.28	
2	-0.79	-0.54	-0.66	-0.56	-0.54	-0.57	
3	0.87	1.06	1.27	1.13	1.07	0.67	
	NumDealsPurchases	**NumWebPurchases**	**NumCatalogPurchases**	**NumStorePurchases**	**NumWebVisitsMonth**		
1	0.87	0.84	0.1	0.54	0.29		
2	-0.19	-0.78	-0.76	-0.82	0.44		
3	-0.54	0.46	1.17	0.82	-1.02		

low

high

MARKET SEGMENT AND SUGGESTION

CUSTOMER PROFILES OF THE THREE GROUPS:

Group 1:

Demographic: **middle** income; **high** education level

Behaviors: prefer **wines and gold** to other products

Distributions: prefer making purchases through **websites** and using **discount**

Group 2:

Demographic: **low** income; **younger** generation; most **undergraduate**

Behaviors: don' t like most products of company A

Distributions: like visiting the website but seldom make purchase

Group 3:

Demographic: **single; high** income; without child

Behaviors: like most products of company A

Distributions: prefer making purchase at **stores** and through catalogue

SUGGESTION TO COMPANY A

	Group 1	Group 2	Group 3
Attribute	Potential consumer base	not target consumer	Loyal consumer
How to improve	• Provide discount of wines and gold products • Focusing on website distribution	Drop	• Improve in-store experience • Establish membership system

8.APPENDIX

1. Attributes of original variables:

People

- ID: Customer's unique identifier
- Year_Birth: Customer's birth year
- Education: Customer's education level
- Marital_Status: Customer's marital status
- Income: Customer's yearly household income
- Kidhome: Number of children in customer's household
- Teenhome: Number of teenagers in customer's household
- Dt_Customer: Date of customer's enrollment with the company
- Recency: Number of days since customer's last purchase
- Complain: 1 if the customer complained in the last 2 years, 0 otherwise

Products

- MntWines: Amount spent on wine in last 2 years
- MntFruits: Amount spent on fruits in last 2 years
- MntMeatProducts: Amount spent on meat in last 2 years
- MntFishProducts: Amount spent on fish in last 2 years
- MntSweetProducts: Amount spent on sweets in last 2 years
- MntGoldProds: Amount spent on gold in last 2 years

Promotion

- NumDealsPurchases: Number of purchases made with a discount
- AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise
- AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise
- AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise
- AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise
- AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise
- Response: 1 if customer accepted the offer in the last campaign, 0 otherwise

Place

- NumWebPurchases: Number of purchases made through the company’ s website
- NumCatalogPurchases: Number of purchases made using a catalogue
- NumStorePurchases: Number of purchases made directly in stores
- NumWebVisitsMonth: Number of visits to company’ s website in the last month

2. Code：

Code from ‘consumer.R’: code for data cleaning and processing, descriptive statistic

```
##DATA READING
oridata<-read.csv('marketing_campaign.csv',sep='\t')
```

```
dim(oridata)

str(oridata)


##DATA CLEANING

clndata=na.omit(oridata)

dim(clndata) #2216     29


#add variables

clndata$ttlspend=
clndata$MntWines+clndata$MntFruits+clndata$MntMeatProducts+clndata$MntFishProducts+clndata$MntSweetProducts
+clndata$MntGoldProds

clndata$ttlnum= clndata$NumWebPurchases+clndata$NumCatalogPurchases+clndata$NumStorePurchases

clndata$age=2002-clndata$Year_Birth

clndata=clndata[,-2] #remove year birth

clndata$child=clndata$Kidhome+clndata$Teenhome

clndata=clndata[,-5] #remove kid home teen home

clndata=clndata[,-5]

#clndata$age=clndata$age-20

#resign date

clndata$Dt_Customer=as.Date(clndata$Dt_Customer,"%d-%m-%Y")

clndata$Dt_Customer=as.Date("2022-06-20")-clndata$Dt_Customer

clndata$Dt_Customer=as.numeric(clndata$Dt_Customer,units='days')

#clndata$Dt_Customer=round(clndata$Dt_Customer/365,digits=2)

# education

for(i in 1:length(clndata$Education)){

  if(clndata[i,]$Education=='2n Cycle')clndata[i,]$Education='undergraduate'

  if(clndata[i,]$Education=='Basic')clndata[i,]$Education='undergraduate'

}

# marriage

table(clndata$Marital_Status)

for(i in 1:length(clndata$Marital_Status)){

  if(clndata[i,]$Marital_Status=='Absurd')clndata[i,]$Marital_Status=1

  if(clndata[i,]$Marital_Status=='Alone')clndata[i,]$Marital_Status=1

  if(clndata[i,]$Marital_Status=='Divorced')clndata[i,]$Marital_Status=1

  if(clndata[i,]$Marital_Status=='Married')clndata[i,]$Marital_Status=2

  if(clndata[i,]$Marital_Status=='Single')clndata[i,]$Marital_Status=1

  if(clndata[i,]$Marital_Status=='Together')clndata[i,]$Marital_Status=2
```

```
    if(clndata[i,]$Marital_Status=='Widow')clndata[i,]$Marital_Status=1
    if(clndata[i,]$Marital_Status=='YOLO')clndata[i,]$Marital_Status=1
}
clndata$Marital_Status=as.numeric(clndata$Marital_Status)
clndata$familymember=clndata$Marital_Status+clndata$child
#clndata$accept=clndata$AcceptedCmp1+clndata$AcceptedCmp2+clndata$AcceptedCmp3+clndata$AcceptedCmp4+clndata$AcceptedCmp5+clndata$Response
clndata=clndata[,-1]#remove ID
#clndata$Income=round(clndata$Income/10000,digits=2)
clndata=clndata[,-23] #remove 2Z
clndata=clndata[,-23]
clndata <- dummy_cols(clndata ,select_columns = c("Education"),remove_first_dummy = TRUE,remove_selected_columns = TRUE )
dim(clndata)
str(clndata)
skim(clndata)


#options(scipen = 100)
#age outliner remove
boxplot(clndata$age,main='age-before',col='#bac3d5')
clndata=clndata[clndata$age<100,]
boxplot(clndata$age,main='age-after',col='#d6d8d8')
#income outliner remove
boxplot(oridata$Income,main='income-before',col='#bac3d5')
clndata=clndata[clndata$Income<150000,] #2208,30
boxplot(clndata$Income,main='income-after',col='#d6d8d8')
#meat outliner remove
boxplot(clndata$MntMeatProducts,main='meat-before',col='#bac3d5')
clndata=clndata[clndata$MntMeatProducts<1000,]
boxplot(clndata$MntMeatProducts,main='meat-after',col='#d6d8d8')
#gold outliner remove
boxplot(clndata$MntGoldProds,main='gold-before',col='#bac3d5')
clndata=clndata[clndata$MntGoldProds<250,]
boxplot(clndata$MntGoldProds,main='gold-after',col='#d6d8d8')
#sweet outliner remove
boxplot(clndata$MntSweetProducts,main='sweet-before',col='#bac3d5')
clndata=clndata[clndata$MntSweetProducts<200,]
```

```
boxplot(clndata$MntSweetProducts,main='sweet-after',col='#d6d8d8')


write.csv(clndata,file='cleaned.csv')


#descriptive statistics
par(mfrow=c(3,2))
hist(clndata$age,
     xlab = "age",
     main = "Histogram of age",
     col = "#bac3d5",
     breaks = 10,
     labels = TRUE)
hist(clndata$Income,
     xlab = "income",
     main = "Histogram of income",
     col = "#d6d8d8",
     breaks = 10,
     labels = TRUE)
hist(clndata$Dt_Customer,
     xlab = "days",
     main = "Histogram of days",
     col = "#f3f0f1",
     breaks = 4,
     labels = TRUE)
hist(clndata$Recency,
     xlab = "recency(days)",
     main = "Histogram of recency",
     col = "#b1a8ac",
     breaks = 5,
     labels = TRUE)
hist(clndata$Marital_Status,
     xlab = "marital status",
     main = "Histogram of marital status: 1 for single 2 for couple",
     col = "#b1a8ac",
     breaks = 2,
     labels = TRUE)
```

```
hist(clndata$child,
      xlab = "child",
      main = "Histogram of child",
      col = "#8a99af",
      breaks = 3,
      labels = TRUE)
#product
summary(clndata[,6:11])
spend=data.frame(Item=as.factor(c('wines','fruits','meat','fish','sweet',
                                  'gold')),
                 data=c(675860,58365,363357,83367,59883,97392))
#ace product:wines,meat
waterfallchart(Item~data,data=spend,col=c('#bac3d5','#d6d8d8'),main='product')
sum(clndata$MntFruits)
#promotion
sum(clndata$AcceptedCmp1) #142
sum(clndata$AcceptedCmp2) #30
sum(clndata$AcceptedCmp3) #163
sum(clndata$AcceptedCmp4) #164
sum(clndata$Response) #333
Campaign_column = c("1st","2nd","3rd","4th","5th","last")
Result_column = c(142,30,163,164,162,333)
Percentage_column = c("142 (6.4%)","30 (1.4%)","163 (7.4%)","164 (7.4%)","162 (7.3%)","333 (15.0%)")
CampaignResult = data.frame(Campaign_column,Result_column,Percentage_column)
plot =barplot( height = CampaignResult$Result_column,
                names.arg = CampaignResult$Campaign_column,
                ylim = c(0,350),
                ylab = "Number of people accepted the offer",
                main = "effectiveness of promotion campaign",
                col = "#d6d8d8")
text( x = plot,
       y = CampaignResult$Result_column,
       label = CampaignResult$Percentage_column,
       pos = 3, cex = 0.8, col = "#8a99af")
table(clndata$NumDealsPurchases)
hist(clndata$NumDealsPurchases)
```

```
dealsname=as.character(c(0:10))
dealsname=c(dealsname,"above10")
plot= barplot(names.arg=dealsname,
              height=c(34,956,493,293,187,94,60,39,14,8,5,15),
              ylab="number of people",
              main="number of deals made with a discount",
              col='#bac2d5')
#distribution
purchase=data.frame(Item=as.factor(c('web','catalog','store')),
                    data=c(9049,5812,12849))
waterfallchart(Item~data,data=purchase,col=c('#b1a8ac','#bac3d5'),main='distribution')
#most distribution:store
```

Code from 'lm2.R' : code for part1(customer behaviour prediction)

```
data<-read.csv('cleaned.csv',sep=',')
lmdata=data[,-1]

#GLM
set.seed(110)
train.rows <- sample(rownames(lmdata), nrow( lmdata )*0.7)
train.data <- lmdata[train.rows , ]
valid.rows <- setdiff(rownames( lmdata ), train.rows)
valid.data <- lmdata[valid.rows , ]
#glm1
#cor(lmdata)
result=glm(Response~.,data=train.data,family='binomial')
summary(result)
AIC(result)
result=step(result,direction="backward")
p=predict(result,newdata=valid.data,type='response')
confusionMatrix(factor(ifelse(p >= 0.5, 1, 0)), factor(valid.data$Response), positive = "1")
#
c <- as.data.frame(result$coefficients)
c$name <- rownames(c)
colnames(c)[1] <- "coef"
c$odds <- exp(c$coef)
```

```
c=c[order(c$odds),]
c=c[-1,]
ggplot(c,aes(x=odds,y=name))+
  geom_bar(stat='identity')+
  geom_vline(aes(xintercept=1),size=.25,linetype='dashed')+
  theme(panel.grid.minor=element_blank())+
  ylab("")+xlab("odds ratio")



#default tree
customer.default.tree = rpart(Response ~ .,
                              data = train.data,
                              method = "class")
prp( customer.default.tree,
     type = 1, extra = 1, varlen = -10,
     box.col = ifelse(customer.default.tree$frame$var == "<leaf>", '#f3f0f1', '#bac3d5'))
customer.default.tree.pred <- predict(customer.default.tree, valid.data, type = "class")
confusionMatrix(customer.default.tree.pred, as.factor(valid.data$Response), positive = "1")

#random forest
customer.rf <- randomForest(as.factor(Response) ~ .,
                            data = train.data,
                            ntree = 500,
                            mtry = 4,
                            nodesize = 5,
                            importance = TRUE)
summary(customer.rf)
varImpPlot(customer.rf, type = 1,main='variable importance',col='#486090')
customer.rf.pred <- predict(customer.rf, valid.data)
confusionMatrix(customer.rf.pred, as.factor(valid.data$Response), positive = "1")
```

code from ‘lm3.R’ : code for part2 (customer spend prediction)

```
data<-read.csv('cleaned.csv',sep=',')
lmdata=cbind(data[,2:5],data[,12:16],data[,25:31])
#wine
lmwine=cbind(lmdata,data[,6])
```

```
names(lmwine)[17]='wine'
set.seed(1120)
train.rows <- sample(rownames(lmwine), nrow( lmdata )*0.7)
train.data <- lmwine[train.rows , ]
valid.rows <- setdiff(rownames( lmwine ), train.rows)
valid.data <- lmwine[valid.rows , ]
result1=lm(wine~.,data=train.data)
summary(result1)
AIC(result1)
result=step(result1,direction = 'back')
summary(result)
AIC(result)
p= predict(result,newdata=valid.data,type='response')
valid.resid = valid.data$wine - p
plot(valid.resid)
accuracy(p,actual)
par(mfrow=c(2,2))
plot(result,1:4,col='grey')
res=data.frame( "Predicted" = p[1:10],
                "Actual" = valid.data$wine[1:10],
                "Residual" = valid.resid[1:10])
write.csv(res,file='re.csv')
train.data['1816',]

#meat
lmmeat=cbind(lmdata,data[,8])
names(lmmeat)[17]='meat'
set.seed(1120)
train.rows <- sample(rownames(lmmeat), nrow( lmdata )*0.7)
train.data <- lmmeat[train.rows , ]
valid.rows <- setdiff(rownames( lmmeat ), train.rows)
valid.data <- lmmeat[valid.rows , ]
result1=lm(meat~.,data=train.data)
summary(result1)
AIC(result1)
result=step(result1,direction = 'back')
```

```
summary(result)
p= predict(result,newdata=valid.data,type='response')
valid.resid = valid.data$meat- p
#plot(valid.resid)
#accuracy(p,actual)
par(mfrow=c(2,2))
plot(result,1:4,col='grey')
res=data.frame( "Predicted" = p[1:10],
                "Actual" = valid.data$meat[1:10],
                "Residual" = valid.resid[1:10])
#fish
lmfish=cbind(lmdata,data[,9])
names(lmfish)[16]='fish'
set.seed(1120)
train.rows <- sample(rownames(lmfish), nrow( lmdata )*0.7)
train.data <- lmfish[train.rows , ]
valid.rows <- setdiff(rownames( lmfish ), train.rows)
valid.data <- lmfish[valid.rows , ]
result1=lm(fish~.,data=train.data)
summary(result1)
AIC(result1)
result=step(result1,direction = 'back')
summary(result)

#sweet
lmsweet=cbind(lmdata,data[,10])
names(lmsweet)[16]='sweet'
set.seed(120)
train.rows <- sample(rownames(lmfish), nrow( lmdata )*0.7)
train.data <- lmsweet[train.rows , ]
valid.rows <- setdiff(rownames( lmfish ), train.rows)
valid.data <- lmsweet[valid.rows , ]
result1=lm(sweet~.,data=train.data)
summary(result1)
AIC(result1)
result=step(result1,direction = 'back')
```

```
summary(result)

#gold
lmgold=cbind(lmdata,data[,10])
names(lmgold)[16]='gold'
set.seed(110)
train.rows <- sample(rownames(lmfish), nrow( lmdata )*0.7)
train.data <- lmgold[train.rows , ]
valid.rows <- setdiff(rownames( lmfish ), train.rows)
valid.data <- lmgold[valid.rows , ]
result1=lm(gold~.,data=train.data)
summary(result1)
AIC(result)
result=step(result1,direction = 'back')
summary(result)

#ttlspned
lmttl=cbind(lmdata,data[,24])
names(lmttl)[17]='ttlspend'
set.seed(200)
train.rows <- sample(rownames(lmttl), nrow( lmdata )*0.7)
train.data <- lmttl[train.rows , ]
valid.rows <- setdiff(rownames( lmttl ), train.rows)
valid.data <- lmttl[valid.rows , ]
result=lm(ttlspend~.,data=train.data)
summary(result)
result1=step(result,direction='backward')
summary(result1)
result2=lm(ttlspend ~ Income + Dt_Customer + NumWebPurchases +
             NumCatalogPurchases + NumStorePurchases + NumWebVisitsMonth +
             age + child, data = train.data)
summary(result2)
AIC(result2)
p= predict(result2,newdata=valid.data,type='response')
valid.resid = valid.data$ttlspend- p
par(mfrow=c(2,2))
```

```
plot(result2,1:4,col='grey')
res=data.frame( "Predicted" = p[1:10],
                "Actual" = valid.data$ttlspend[1:10],
                "Residual" = valid.resid[1:10])

#
normal=cbind(lmdata,data[,7:11])
normal=normal[,-17]
fruit=lmdata
fruit$fruit=0
for(i in 1:2198){
  if(max(normal[i,16:19])==normal[i,16])fruit[i,]$fruit=1
}
set.seed(100)
train.rows <- sample(rownames(fruit), nrow( lmdata )*0.7)
train.data <- fruit[train.rows , ]
valid.rows <- setdiff(rownames(fruit), train.rows)
valid.data <- fruit[valid.rows , ]
re=glm(fruit~.,data=train.data,family='binomial')
pf=predict(re,newdata=valid.data,type='response')
confusionMatrix(factor(ifelse(p >= 0.5, 1, 0)), factor(valid.data$fruit), positive = "1")

summary(data$MntWines)
summary(data$MntMeatProducts)
summary(data$ttlspend)
```

code from ‘clustering.R’ : code for part 3

```
data<-read.csv('cleaned.csv',sep=',')
data=data[,-1]
scal.data=scale(data[,-1])
kappa(cor(scal.data))
corr=round(cor(data),1)
corrplot(corr,)
data.pr<-princomp(scal.data,cor=T)
summary(data.pr)
screeplot(data.pr,type="lines")
```

```
data.pr


cluster=data[,c(1:4,23:27,5:15)]
cluster=scale(cluster)
corr=round(cor(cluster),1)
corrplot(corr,tl.col='black',
         tl.srt=30 )
kappa(cor(cluster))
options(scipen = 100)
fviz_nbclust(cluster,kmeans,method="wss")+geom_vline(xintercept=3,linetype=2)
set.seed(90)
km<-kmeans(cluster,iter.max=100,center=3,nstart=20)
fviz_cluster(km, cluster, geom = "point",ellipse.type = "convex",
             repel = TRUE,
             labelsize=7,palette = "jco",
             ggtheme = theme_minimal())
cluster1<-round(aggregate(cluster,by=list(km$cluster),FUN=mean),2)
?fviz_cluster
write.csv(cluster1,file='cluser.csv')
```