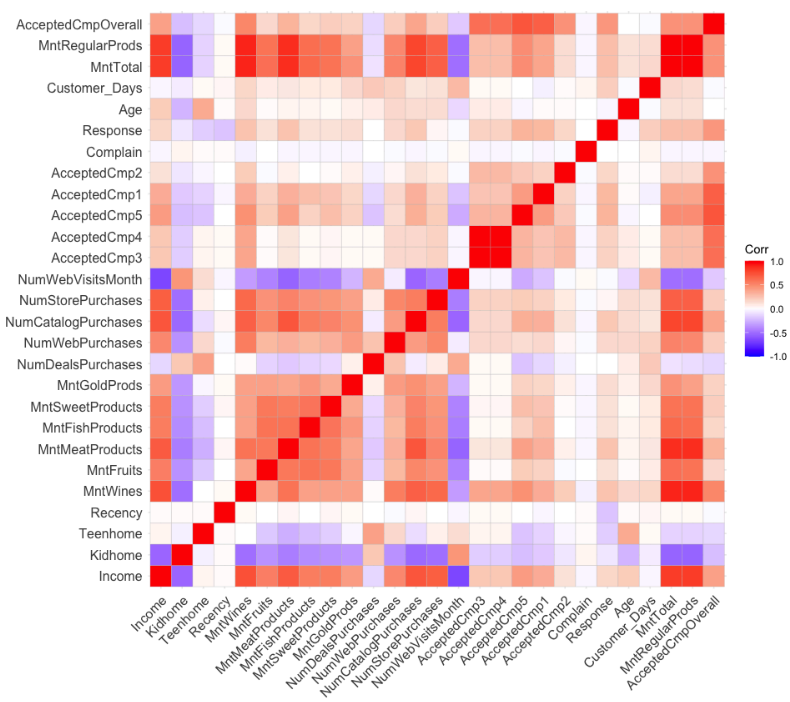
 # CORRELATION MATRIX

INPUT

```
CorA <- cor(ifoodR)   # creating correlation matrix
library(ggcorrplot)
> ggcorrplot(CorA)    # plotting corr. matrix
```

OUTPUT



INPUT

```
a <- as.data.frame(as.table(CorA))
> suba <- subset(a, abs(Freq) > 0.5) # creating subset of corr.
Matrix with the correlation is above 0.5 or below -0.5

> suba
```

OUTPUT

	Var1	Var2	Freq
1	Income	Income	1.0000000
2	Kidhome	Income	-0.5316989
5	MntWines	Income	0.7304952
6	MntFruits	Income	0.5379203
7	MntMeatProducts	Income	0.7024996
8	MntFishProducts	Income	0.5517580
9	MntSweetProducts	Income	0.5556010
12	NumWebPurchases	Income	0.5031842
13	NumCatalogPurchases	Income	0.7100565
14	NumStorePurchases	Income	0.6872057
15	NumWebVisitsMonth	Income	-0.6483063
19	MntTotal	Income	0.8230660
20	MntRegularProds	Income	0.8168792
22	Income	Kidhome	-0.5316989
23	Kidhome	Kidhome	1.0000000
34	NumCatalogPurchases	Kidhome	-0.5198133
35	NumStorePurchases	Kidhome	-0.5065432
40	MntTotal	Kidhome	-0.5511520
41	MntRegularProds	Kidhome	-0.5398280
45	Teenhome	Teenhome	1.0000000
67	Recency	Recency	1.0000000
85	Income	MntWines	0.7304952
89	MntWines	MntWines	1.0000000
91	MntMeatProducts	MntWines	0.5931189
96	NumWebPurchases	MntWines	0.5523421
97	NumCatalogPurchases	MntWines	0.6732338
98	NumStorePurchases	MntWines	0.6393728
103	MntTotal	MntWines	0.9023096
104	MntRegularProds	MntWines	0.9018484
105	AcceptedCmpOverall	MntWines	0.5099130
106	Income	MntFruits	0.5379203
111	MntFruits	MntFruits	1.0000000
112	MntMeatProducts	MntFruits	0.5681001
113	MntFishProducts	MntFruits	0.5925564
114	MntSweetProducts	MntFruits	0.5709861
118	NumCatalogPurchases	MntFruits	0.5136863
124	MntTotal	MntFruits	0.6066577
125	MntRegularProds	MntFruits	0.5941803
127	Income	MntMeatProducts	0.7024996
131	MntWines	MntMeatProducts	0.5931189
132	MntFruits	MntMeatProducts	0.5681001
133	MntMeatProducts	MntMeatProducts	1.0000000

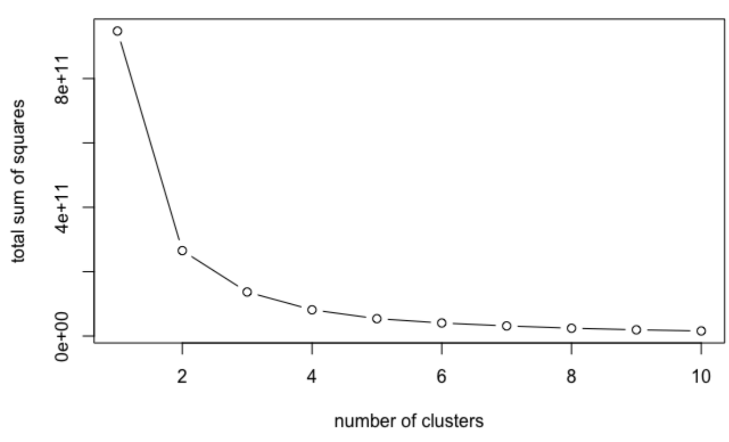
____

CUSTOMER SEGMENTATION

INPUT

```
library(purrr)

# dropping attributes with no variance or non numerical
> clustering_ifood <- subset(ifood_df, select = -Z_CostContact )
> clustering_ifood <- subset(clustering_ifood, select = -Z_Revenue )
> clustering_ifood <- subset(clustering_ifood, select = -Response )
> clustering_ifood <- subset(clustering_ifood, select = -ID )
> clustering_ifood <- subset(clustering_ifood, select =
-MaritalStatus )
> clustering_ifood <- subset(clustering_ifood, select = -Education )

# elbow method to find optimal number of clusters for this dataset
> fun <- function(k){
+     kmeans(clustering_ifood,k,iter.max=100,nstart =
100,algorithm='Lloyd')$tot.withinss}
> k.values <- 1:10
> fun_value <- map_dbl(k.values,fun)
> plot(k.values,fun_value,type='b',xlab='number of
clusters',ylab='total sum of squares')
```

OUTPUT



INPUT

```
# K-means clustering with 3 clusters created
> k3 <- kmeans(clustering_ifood,3,iter.max = 100,nstart =
50,algorithm = 'Lloyd')
> print(k3)
```

OUTPUT

```
K-means clustering with 3 clusters of sizes 724, 766, 715

Cluster means:
    Income      Age    Kidhome  Teenhome  Recency     Wines    Fruits
1 28077.80 46.79282 0.81215470 0.3080110 48.41436  29.86188  5.872928
2 51572.19 53.73107 0.41906005 0.8146214 49.57180 271.36423 17.592689
3 75516.22 52.62937 0.09230769 0.3776224 49.00839 623.22797 56.630769
  MeatProducts FishProducts SweetProducts GoldProds NumDealsPurchases
1     25.30525     9.063536      6.022099  17.50276          2.146409
2     91.58355    24.113577     16.938642  43.96214          3.093995
3    386.06853    81.426573     59.416783  71.04755          1.661538
  NumWebPurchases NumCatalogPurchases NumStorePurchases NumWebVisitsMonth
1        2.142265           0.5290055          3.067680          6.911602
2        4.612272           2.1422977          5.934726          5.744125
3        5.535664           5.3272727          8.495105          3.306294
  AcceptedCmp1 AcceptedCmp2 AcceptedCmp3 AcceptedCmp4 AcceptedCmp5
1  0.001381215   0.00000000   0.08425414  0.004143646  0.000000000
2  0.016971279   0.01436031   0.06657963  0.083550914  0.003916449
3  0.179020979   0.02657343   0.07132867  0.135664336  0.220979021
     Complain Customer_Days   MntTotal MntRegularProds AcceptedCmpOverall
1 0.015193370      2511.544   76.12569        58.62293         0.08977901
2 0.005221932      2522.389  421.59269       377.63055         0.18537859
3 0.006993007      2503.547 1206.77063      1135.72308         0.63356643
```

______

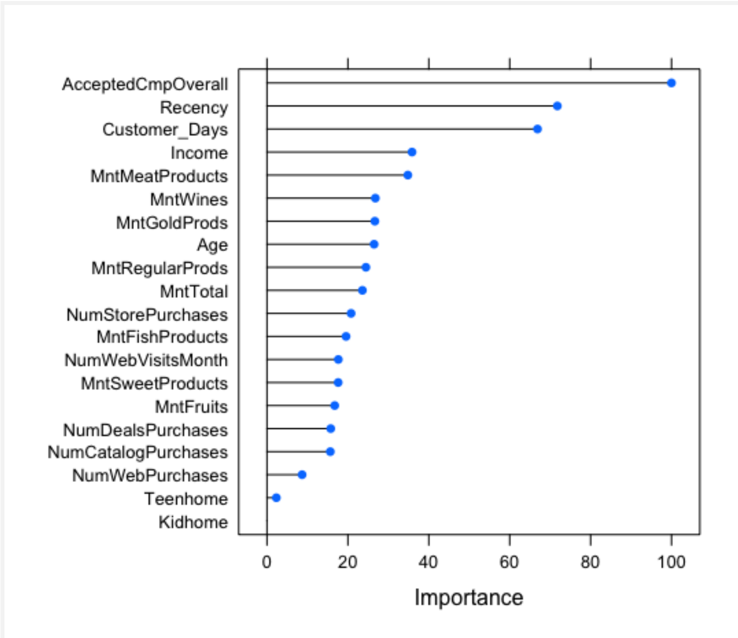
 # PREDICTIVE CLASSIFICATION MODEL

INPUT

```
> library(caret)

# 5-fold CV
> fitControl <- trainControl(
method = "repeatedcv",
number = 5,
repeats = 5,
search = "random")

# random forest classification method
> model.cv <- train(Response ~ .,
data = ifooddf,
method = "randomforest",
trControl = fitControl,
preProcess = c('scale', 'center'),
na.action = na.omit)

# rank importance of variables in model
> VarImpiFood <- varImp(model.cv)
```

```
> ggplot2(VarImpiFood) # plot variable importance
```

OUTPUT



INPUT

```
> VarImpiFood
```

OUTPUT

```
rf variable importance

                    Overall
AcceptedCmpOverall  100.000
Recency              71.786
Customer_Days        66.877
Income               35.854
MntMeatProducts      34.819
MntWines             26.783
MntGoldProds         26.663
Age                  26.474
MntRegularProds      24.452
MntTotal             23.587
NumStorePurchases    20.790
MntFishProducts      19.540
NumWebVisitsMonth    17.632
MntSweetProducts     17.594
MntFruits            16.739
NumDealsPurchases    15.758
NumCatalogPurchases  15.650
NumWebPurchases       8.678
Teenhome              2.309
Kidhome               0.000
```