

Homework2_FBDA

109090035_陳佑祥

2/25/2023

Read data as dataframe

```
library(skimr)
# Read CSV into a DataFrame named ' market '
market = read.csv('/Users/user/Downloads/marketing_data.csv')
head(market)
```

##	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome
## 1	1826	1970	Graduation	Divorced	\$84,835.00	0	0
## 2	1	1961	Graduation	Single	\$57,091.00	0	0
## 3	10476	1958	Graduation	Married	\$67,267.00	0	1
## 4	1386	1967	Graduation	Together	\$32,474.00	1	1
## 5	5371	1989	Graduation	Single	\$21,474.00	1	0
## 6	7348	1958	PhD	Single	\$71,691.00	0	0
##	Dt_Customer	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	
## 1	6/16/14	0	189	104	379	111	
## 2	6/15/14	0	464	5	64	7	
## 3	5/13/14	0	134	11	59	15	
## 4	5/11/14	0	10	0	1	0	
## 5	4/8/14	0	6	16	24	11	
## 6	3/17/14	0	336	130	411	240	
##	MntSweetProducts	MntGoldProds	NumDealsPurchases	NumWebPurchases			
## 1	189	218	1	4			
## 2	0	37	1	7			
## 3	2	30	1	3			
## 4	0	0	1	1			
## 5	0	34	2	3			
## 6	32	43	1	4			
##	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	AcceptedCmp3			
## 1	4	6	1	0			
## 2	3	7	5	0			
## 3	2	5	2	0			
## 4	0	2	7	0			
## 5	1	2	7	1			
## 6	7	5	2	0			
##	AcceptedCmp4	AcceptedCmp5	AcceptedCmp1	AcceptedCmp2	Response	Complain	Country
## 1	0	0	0	0	1	0	SP
## 2	0	0	0	1	1	0	CA
## 3	0	0	0	0	0	0	US
## 4	0	0	0	0	0	0	AUS
## 5	0	0	0	0	1	0	SP
## 6	0	0	0	0	1	0	SP

第一題

```
str(market)
```

```
## 'data.frame':    2240 obs. of  28 variables:
##  $ ID                : int  1826 1 10476 1386 5371 7348 4073 1991 4047 9477 ...
##  $ Year_Birth         : int  1970 1961 1958 1967 1989 1958 1954 1967 1954 1954 ...
##  $ Education          : chr   "Graduation" "Graduation" "Graduation" "Graduation" ...
##  $ Marital_Status     : chr   "Divorced" "Single" "Married" "Together" ...
##  $ Income             : chr   "$84,835.00 " "$57,091.00 " "$67,267.00 " "$32,474.00 " ...
##  $ Kidhome            : int    0 0 0 1 1 0 0 0 0 0 ...
##  $ Teenhome           : int    0 0 1 1 0 0 0 1 1 1 ...
##  $ Dt_Customer        : chr   "6/16/14" "6/15/14" "5/13/14" "5/11/14" ...
##  $ Recency            : int    0 0 0 0 0 0 0 0 0 0 ...
##  $ MntWines           : int   189 464 134 10 6 336 769 78 384 384 ...
##  $ MntFruits          : int   104 5 11 0 16 130 80 0 0 0 ...
##  $ MntMeatProducts    : int   379 64 59 1 24 411 252 11 102 102 ...
##  $ MntFishProducts    : int   111 7 15 0 11 240 15 0 21 21 ...
##  $ MntSweetProducts   : int   189 0 2 0 0 32 34 0 32 32 ...
##  $ MntGoldProds       : int   218 37 30 0 34 43 65 7 5 5 ...
##  $ NumDealsPurchases  : int    1 1 1 1 2 1 1 1 3 3 ...
##  $ NumWebPurchases    : int    4 7 3 1 3 4 10 2 6 6 ...
##  $ NumCatalogPurchases: int    4 3 2 0 1 7 10 1 2 2 ...
##  $ NumStorePurchases  : int    6 7 5 2 2 5 7 3 9 9 ...
##  $ NumWebVisitsMonth  : int    1 5 2 7 7 2 6 5 4 4 ...
##  $ AcceptedCmp3       : int    0 0 0 0 1 0 1 0 0 0 ...
##  $ AcceptedCmp4       : int    0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp5       : int    0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp1       : int    0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp2       : int    0 1 0 0 0 0 0 0 0 0 ...
##  $ Response           : int    1 1 0 0 1 1 1 0 0 0 ...
##  $ Complain           : int    0 0 0 0 0 0 0 0 0 0 ...
##  $ Country            : chr   "SP" "CA" "US" "AUS" ...
```

```
skim(market)
```

Data summary

Name	market
Number of rows	2240
Number of columns	28
<hr/>	
Column type frequency:	
character	5
numeric	23
<hr/>	
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
Income	0	1	0	12	24	1975	0
Dt_Customer	0	1	6	8	0	663	0
Country	0	1	2	3	0	8	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ID	0	1	5592.16	3246.66	0	2828.25	5458.5	8427.75	11191	■■■■■
Year_Birth	0	1	1968.81	11.98	1893	1959.00	1970.0	1977.00	1996	■■■■■

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Kidhome	0	1	0.44	0.54	0	0.00	0.0	1.00	2	
Teenhome	0	1	0.51	0.54	0	0.00	0.0	1.00	2	
Recency	0	1	49.11	28.96	0	24.00	49.0	74.00	99	
MntWines	0	1	303.94	336.60	0	23.75	173.5	504.25	1493	
MntFruits	0	1	26.30	39.77	0	1.00	8.0	33.00	199	
MntMeatProducts	0	1	166.95	225.72	0	16.00	67.0	232.00	1725	
MntFishProducts	0	1	37.53	54.63	0	3.00	12.0	50.00	259	
MntSweetProducts	0	1	27.06	41.28	0	1.00	8.0	33.00	263	
MntGoldProds	0	1	44.02	52.17	0	9.00	24.0	56.00	362	
NumDealsPurchases	0	1	2.33	1.93	0	1.00	2.0	3.00	15	
NumWebPurchases	0	1	4.08	2.78	0	2.00	4.0	6.00	27	
NumCatalogPurchases	0	1	2.66	2.92	0	0.00	2.0	4.00	28	
NumStorePurchases	0	1	5.79	3.25	0	3.00	5.0	8.00	13	
NumWebVisitsMonth	0	1	5.32	2.43	0	3.00	6.0	7.00	20	
AcceptedCmp3	0	1	0.07	0.26	0	0.00	0.0	0.00	1	
AcceptedCmp4	0	1	0.07	0.26	0	0.00	0.0	0.00	1	
AcceptedCmp5	0	1	0.07	0.26	0	0.00	0.0	0.00	1	
AcceptedCmp1	0	1	0.06	0.25	0	0.00	0.0	0.00	1	
AcceptedCmp2	0	1	0.01	0.11	0	0.00	0.0	0.00	1	
Response	0	1	0.15	0.36	0	0.00	0.0	0.00	1	
Complain	0	1	0.01	0.10	0	0.00	0.0	0.00	1	

第二題

```
ls(market)
```

```
## [1] "AcceptedCmp1"      "AcceptedCmp2"      "AcceptedCmp3"
## [4] "AcceptedCmp4"      "AcceptedCmp5"      "Complain"
## [7] "Country"           "Dt_Customer"       "Education"
## [10] "ID"                "Income"            "Kidhome"
## [13] "Marital_Status"    "MntFishProducts"   "MntFruits"
## [16] "MntGoldProds"      "MntMeatProducts"   "MntSweetProducts"
## [19] "MntWines"          "NumCatalogPurchases" "NumDealsPurchases"
## [22] "NumStorePurchases" "NumWebPurchases"    "NumWebVisitsMonth"
## [25] "Recency"           "Response"           "Teenhome"
## [28] "Year_Birth"
```

第三題

```
sum(is.na(market))
```

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

第四題

```
sapply(market, function(x) sum(is.na(x)))
```

```
##          ID          Year_Birth          Education          Marital_Status
##          0            0            0            0
##          Income          Kidhome          Teenhome          Dt_Customer
##          0            0            0            0
##          Recency          MntWines          MntFruits          MntMeatProducts
##          0            0            0            0
##          MntFishProducts          MntSweetProducts          MntGoldProds          NumDealsPurchases
##          0            0            0            0
##          NumWebPurchases          NumCatalogPurchases          NumStorePurchases          NumWebVisitsMonth
##          0            0            0            0
##          AcceptedCmp3          AcceptedCmp4          AcceptedCmp5          AcceptedCmp1
##          0            0            0            0
##          AcceptedCmp2          Response          Complain          Country
##          0            0            0            0
```

第四題（Hint方法）

```
colSums(is.na(market))
```

```
##          ID          Year_Birth          Education          Marital_Status
##          0            0            0            0
##          Income          Kidhome          Teenhome          Dt_Customer
##          0            0            0            0
##          Recency          MntWines          MntFruits          MntMeatProducts
##          0            0            0            0
##          MntFishProducts          MntSweetProducts          MntGoldProds          NumDealsPurchases
##          0            0            0            0
##          NumWebPurchases          NumCatalogPurchases          NumStorePurchases          NumWebVisitsMonth
##          0            0            0            0
##          AcceptedCmp3          AcceptedCmp4          AcceptedCmp5          AcceptedCmp1
##          0            0            0            0
##          AcceptedCmp2          Response          Complain          Country
##          0            0            0            0
```

```
#double check with skim()
skim(market)
```



















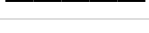



Data summary

Name	market
Number of rows	2240
Number of columns	28
<hr/>	
Column type frequency:	
character	5
numeric	23
<hr/>	
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
Income	0	1	0	12	24	1975	0
Dt_Customer	0	1	6	8	0	663	0
Country	0	1	2	3	0	8	0

Variable type: numeric

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

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Complain	0	1	0.01	0.10	0	0.00	0.0	0.00	1	

第五題

```
library(psych)
describe(market)
```

##	vars	n	mean	sd	median	trimmed	mad	min	max
## ID	1	2240	5592.16	3246.66	5458.5	5582.43	4137.94	0	11191
## Year_Birth	2	2240	1968.81	11.98	1970.0	1968.94	13.34	1893	1996
## Education*	3	2240	3.39	1.12	3.0	3.48	0.00	1	5
## Marital_Status*	4	2240	4.73	1.08	5.0	4.75	1.48	1	8
## Income*	5	2240	978.27	575.95	975.5	978.22	744.27	1	1975
## Kidhome	6	2240	0.44	0.54	0.0	0.40	0.00	0	2
## Teenhome	7	2240	0.51	0.54	0.0	0.48	0.00	0	2
## Dt_Customer*	8	2240	331.32	194.29	329.0	331.58	250.56	1	663
## Recency	9	2240	49.11	28.96	49.0	49.11	37.06	0	99
## MntWines	10	2240	303.94	336.60	173.5	248.99	243.89	0	1493
## MntFruits	11	2240	26.30	39.77	8.0	16.98	11.86	0	199
## MntMeatProducts	12	2240	166.95	225.72	67.0	119.33	87.47	0	1725
## MntFishProducts	13	2240	37.53	54.63	12.0	25.08	17.79	0	259
## MntSweetProducts	14	2240	27.06	41.28	8.0	17.35	11.86	0	263
## MntGoldProds	15	2240	44.02	52.17	24.0	33.41	26.69	0	362
## NumDealsPurchases	16	2240	2.32	1.93	2.0	1.96	1.48	0	15
## NumWebPurchases	17	2240	4.08	2.78	4.0	3.81	2.97	0	27
## NumCatalogPurchases	18	2240	2.66	2.92	2.0	2.21	2.97	0	28
## NumStorePurchases	19	2240	5.79	3.25	5.0	5.48	2.97	0	13
## NumWebVisitsMonth	20	2240	5.32	2.43	6.0	5.40	2.97	0	20
## AcceptedCmp3	21	2240	0.07	0.26	0.0	0.00	0.00	0	1
## AcceptedCmp4	22	2240	0.07	0.26	0.0	0.00	0.00	0	1
## AcceptedCmp5	23	2240	0.07	0.26	0.0	0.00	0.00	0	1
## AcceptedCmp1	24	2240	0.06	0.25	0.0	0.00	0.00	0	1
## AcceptedCmp2	25	2240	0.01	0.11	0.0	0.00	0.00	0	1
## Response	26	2240	0.15	0.36	0.0	0.06	0.00	0	1
## Complain	27	2240	0.01	0.10	0.0	0.00	0.00	0	1
## Country*	28	2240	5.46	2.20	7.0	5.72	1.48	1	8
##	range	skew	kurtosis	se					
## ID	11191	0.04	-1.19	68.60					
## Year_Birth	103	-0.35	0.71	0.25					
## Education*	4	-0.28	-0.16	0.02					
## Marital_Status*	7	0.17	-0.78	0.02					
## Income*	1974	0.00	-1.21	12.17					
## Kidhome	2	0.63	-0.78	0.01					
## Teenhome	2	0.41	-0.99	0.01					
## Dt_Customer*	662	0.00	-1.21	4.11					

## Recency	99	0.00	-1.20	0.61
## MntWines	1493	1.17	0.59	7.11
## MntFruits	199	2.10	4.03	0.84
## MntMeatProducts	1725	2.08	5.49	4.77
## MntFishProducts	259	1.92	3.08	1.15
## MntSweetProducts	263	2.13	4.36	0.87
## MntGoldProds	362	1.88	3.54	1.10
## NumDealsPurchases	15	2.42	8.90	0.04
## NumWebPurchases	27	1.38	5.68	0.06
## NumCatalogPurchases	28	1.88	8.02	0.06
## NumStorePurchases	13	0.70	-0.63	0.07
## NumWebVisitsMonth	20	0.21	1.81	0.05
## AcceptedCmp3	1	3.29	8.81	0.01
## AcceptedCmp4	1	3.24	8.48	0.01
## AcceptedCmp5	1	3.29	8.81	0.01
## AcceptedCmp1	1	3.55	10.61	0.01
## AcceptedCmp2	1	8.46	69.62	0.00
## Response	1	1.97	1.88	0.01
## Complain	1	10.18	101.58	0.00
## Country*	7	-0.91	-0.76	0.05

```
summary(market)
```

```

##      ID      Year_Birth      Education      Marital_Status
## Min.    :    0      Min.    :1893      Length:2240      Length:2240
## 1st Qu.: 2828      1st Qu.:1959      Class :character      Class :character
## Median : 5458      Median :1970      Mode  :character      Mode  :character
## Mean    : 5592      Mean    :1969
## 3rd Qu.: 8428      3rd Qu.:1977
## Max.    :11191      Max.    :1996
##      Income      Kidhome      Teenhome      Dt_Customer
## Length:2240      Min.    :0.0000      Min.    :0.0000      Length:2240
## Class :character      1st Qu.:0.0000      1st Qu.:0.0000      Class :character
## Mode  :character      Median :0.0000      Median :0.0000      Mode  :character
##                               Mean    :0.4442      Mean    :0.5062
##                               3rd Qu.:1.0000      3rd Qu.:1.0000
##                               Max.    :2.0000      Max.    :2.0000
##      Recency      MntWines      MntFruits      MntMeatProducts
## Min.    : 0.00      Min.    : 0.00      Min.    : 0.0      Min.    : 0
## 1st Qu.:24.00      1st Qu.: 23.75      1st Qu.: 1.0      1st Qu.: 16
## Median :49.00      Median : 173.50      Median : 8.0      Median : 67
## Mean    :49.11      Mean    : 303.94      Mean    : 26.3      Mean    : 167
## 3rd Qu.:74.00      3rd Qu.: 504.25      3rd Qu.: 33.0      3rd Qu.: 232
## Max.    :99.00      Max.    :1493.00      Max.    :199.0      Max.    :1725
## MntFishProducts      MntSweetProducts      MntGoldProds      NumDealsPurchases
## Min.    : 0.00      Min.    : 0.00      Min.    : 0.00      Min.    : 0.000
## 1st Qu.: 3.00      1st Qu.: 1.00      1st Qu.: 9.00      1st Qu.: 1.000
## Median : 12.00      Median : 8.00      Median : 24.00      Median : 2.000
## Mean    : 37.53      Mean    : 27.06      Mean    : 44.02      Mean    : 2.325
## 3rd Qu.: 50.00      3rd Qu.: 33.00      3rd Qu.: 56.00      3rd Qu.: 3.000
## Max.    :259.00      Max.    :263.00      Max.    :362.00      Max.    :15.000
## NumWebPurchases      NumCatalogPurchases      NumStorePurchases      NumWebVisitsMonth
## Min.    : 0.000      Min.    : 0.000      Min.    : 0.00      Min.    : 0.000
## 1st Qu.: 2.000      1st Qu.: 0.000      1st Qu.: 3.00      1st Qu.: 3.000
## Median : 4.000      Median : 2.000      Median : 5.00      Median : 6.000
## Mean    : 4.085      Mean    : 2.662      Mean    : 5.79      Mean    : 5.317
## 3rd Qu.: 6.000      3rd Qu.: 4.000      3rd Qu.: 8.00      3rd Qu.: 7.000
## Max.    :27.000      Max.    :28.000      Max.    :13.00      Max.    :20.000
## AcceptedCmp3      AcceptedCmp4      AcceptedCmp5      AcceptedCmp1
## Min.    :0.00000      Min.    :0.00000      Min.    :0.00000      Min.    :0.00000
## 1st Qu.:0.00000      1st Qu.:0.00000      1st Qu.:0.00000      1st Qu.:0.00000

```

```
## Median :0.00000 Median :0.00000 Median :0.00000 Median :0.00000
## Mean :0.07277 Mean :0.07455 Mean :0.07277 Mean :0.06429
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.00000
## AcceptedCmp2 Response Complain Country
## Min. :0.00000 Min. :0.0000 Min. :0.000000 Length:2240
## 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.000000 Class :character
## Median :0.00000 Median :0.0000 Median :0.000000 Mode :character
## Mean :0.01339 Mean :0.1491 Mean :0.009375
## 3rd Qu.:0.00000 3rd Qu.:0.0000 3rd Qu.:0.000000
## Max. :1.00000 Max. :1.0000 Max. :1.000000
```

The summary function in r, simply just print out the type of the character and state the number of length that the column value type is character have. On the other side, describe function giving out more detail information during handling character value such as “variation”, “number of character”, “mean”, “sd”, “median”, “trimmed”, “mad”, “min”, “max”, “range”, “skew”, “kurtosis” and “se”.

第六題

```
library(magrittr)
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
```

```
market %$%
  table(Education)
```

```
## Education
##   2n Cycle   Basic Graduation   Master   PhD
##       203         54       1127       370       486
```

```
print("This is the table of the existing frequency number of degree in the market data")
```

```
## [1] "This is the table of the existing frequency number of degree in the market data"
```

```
market %$%
  table(Marital_Status)
```

```
## Marital_Status
##   Absurd   Alone Divorced   Married   Single Together   Widow   YOLO
##       2         3       232       864       480       580       77       2
```

```
print("This is the table of the existing frequency number of marriage status in the market data")
```

```
## [1] "This is the table of the existing frequency number of marriage status in the market data"
```

```
market %$%
  table(Country)
```

```
## Country
##   AUS   CA   GER   IND   ME   SA   SP   US
##   160  268  120  148    3  337 1095  109
```

```
print("This is the table of the existing frequency number of country status in the market data")
```

```
## [1] "This is the table of the existing frequency number of country status in the market data"
```

```
market %$%
  table(Education,Marital_Status)
```

```
##           Marital_Status
## Education  Absurd Alone Divorced Married Single Together Widow YOLO
## 2n Cycle    0     0     23     81     37     57     5     0
## Basic       0     0      1     20     18     14     1     0
## Graduation  1     1    119    433    252    286    35     0
## Master      1     1     37    138     75    106    12     0
## PhD         0     1     52    192     98    117    24     2
```

```
print("This is the frequency table with two column in the market dataframe with the existing frequency number of degree and marriage status in the market data")
```

```
## [1] "This is the frequency table with two column in the market dataframe with the existing frequency number of degree and marriage status in the market data"
```

第七題

```
library(stringr)
head(market$Income)
```

```
## [1] "$84,835.00 " "$57,091.00 " "$67,267.00 " "$32,474.00 " "$21,474.00 "
## [6] "$71,691.00 "
```

```
#convert string to numeric vlaue
sub(", " , "" ,str_sub(market$Income,2)) %>%
  as.numeric()
```


##	[1]	84835	57091	67267	32474	21474	71691	63564	44931	65324	65324
##	[11]	81044	62499	67786	26872	4428	54809	54809	32173	47823	30523
##	[21]	36634	43456	40662	49544	57731	33168	54450	54450	35340	73170
##	[31]	65808	79529	34578	46374	18351	95169	65370	39228	84865	61314
##	[41]	21359	66465	81698	39146	25959	64260	32300	74806	72967	48904
##	[51]	14796	66991	52195	90369	18393	47139	38576	61905	83715	60597
##	[61]	6560	48330	38236	22701	53367	34728	63915	82504	38578	38578
##	[71]	79174	81975	43263	27242	76445	75276	34213	72335	79143	27469
##	[81]	68695	68695	50388	77622	46102	31859	23830	24639	71604	27255
##	[91]	55375	77037	24163	69476	72063	58646	58646	47570	80011	54998
##	[101]	60200	63211	48432	48432	48432	62845	62845	65220	74214	96547
##	[111]	53863	50447	37716	45203	37368	67225	65104	65104	81698	70566
##	[121]	69096	71434	33039	33039	46610	49967	53359	49505	68657	72071
##	[131]	72071	59235	21994	21994	NA	69209	31928	74004	30390	21355
##	[141]	26304	85710	102692	58350	7500	18890	20425	20425	41437	49912
##	[151]	49912	79823	85683	47025	68397	38175	46423	46423	30560	30560
##	[161]	60152	29672	29672	70321	56181	24279	68462	55424	49094	59184
##	[171]	67430	30522	31878	83528	90300	68092	53761	52845	24027	7500
##	[181]	42586	42586	9722	86836	30467	54549	19444	69245	72968	48686
##	[191]	73687	82716	35523	60554	64090	61559	71670	67680	54386	64108
##	[201]	69702	47958	67716	34600	34600	48752	15716	46610	74985	45576
##	[211]	113734	57100	24434	37126	45160	33178	64866	34176	50014	92344
##	[221]	55282	48799	84169	84169	39763	74165	42720	65275	36443	43776
##	[231]	56046	65148	49431	32583	16653	26850	51287	43050	73448	17256
##	[241]	86429	33569	25130	62998	26907	26907	45579	47320	70179	70179
##	[251]	28072	50501	58917	69372	35178	35178	75693	69109	69109	80134
##	[261]	64713	41411	NA	55521	43795	22070	36408	34026	34026	46049
##	[271]	35641	32414	59821	34968	56320	70091	46097	41003	44911	33249
##	[281]	33249	42873	51039	78952	29435	79865	34176	34176	65968	86424
##	[291]	67272	78499	77044	70515	17459	80336	15759	26326	62061	5305
##	[301]	75777	75777	71499	31626	42670	83917	41021	79456	78789	50870
##	[311]	35876	23539	32557	62466	62466	54198	82170	61798	15253	78285
##	[321]	62535	27190	48985	72940	36145	157146	81843	41883	45759	45759
##	[331]	78394	30279	30279	31880	60208	78687	37401	37401	75315	69263
##	[341]	78569	57705	74805	66426	51529	18100	59601	20981	74068	36790
##	[351]	18793	27421	30675	83145	34053	31788	45894	41986	42769	58582
##	[361]	49096	76412	18929	18929	18929	21888	41580	31605	25509	25509
##	[371]	33051	28764	32146	22148	36781	54197	93790	24762	30477	11448

##	[381]	70844	30828	62513	27733	39435	75345	20491	36715	71706	69063
##	[391]	24645	35797	44078	26954	NA	38887	65463	53537	36230	80763
##	[401]	60585	49166	75261	89891	72828	75278	38557	81300	60161	69389
##	[411]	64497	61014	44319	53233	77972	81361	81361	25707	42403	65704
##	[421]	37406	39767	54210	57136	40794	83003	89058	20193	72099	43824
##	[431]	66653	44359	79410	55842	79946	88194	80184	76982	65295	57100
##	[441]	29009	78931	7500	56939	64509	30093	30351	30351	38201	NA
##	[451]	43974	38200	87305	16860	27213	7500	4861	38196	41145	38513
##	[461]	45688	77863	42169	45906	35388	80952	39898	39898	44051	86718
##	[471]	80141	58597	70356	46904	37760	37760	61839	29791	48918	48918
##	[481]	58692	39356	76234	55434	66835	66835	71965	74538	50002	53977
##	[491]	53977	69930	62981	46014	18227	65991	38808	160803	43783	30396
##	[501]	65526	65526	55956	55563	57236	89694	57420	33622	29999	31497
##	[511]	31497	36802	61671	60182	29732	45921	16529	98777	98777	77027
##	[521]	69139	69139	65169	40049	69755	NA	54456	666666	60689	26487
##	[531]	28164	22585	64795	42000	42000	70713	82800	82800	30801	83891
##	[541]	38054	14045	44155	57642	31353	31353	41335	41335	40737	19789
##	[551]	35946	36262	44551	71367	71367	57957	50334	48150	56796	56796
##	[561]	52278	52278	7500	33444	20895	37509	88097	59041	43322	46015
##	[571]	46015	34242	65210	50520	90765	80589	49572	75437	44635	58512
##	[581]	62000	54252	45068	60474	60474	56337	22434	61416	92163	58656
##	[591]	NA	8940	26751	71427	26646	19346	56243	59892	71613	25252
##	[601]	42160	15862	82584	38360	75283	34941	61823	90226	44989	44989
##	[611]	43586	32233	62882	57288	75774	75774	90247	32218	84219	84219
##	[621]	48877	60905	75236	62551	70123	50785	15315	15315	64413	87171
##	[631]	30081	20180	20180	55686	26887	61996	51537	72025	30538	61618
##	[641]	77343	75433	82571	60033	39062	30023	30015	33228	22574	22574
##	[651]	55267	66886	72258	82733	45146	51250	32871	63855	42564	46344
##	[661]	50437	70971	37774	4023	33279	42693	70545	65486	77882	78468
##	[671]	77226	90638	90638	70666	59052	13084	49160	23272	49476	62694
##	[681]	57906	71853	71853	65031	30507	67419	59462	95529	53653	25965
##	[691]	22979	64950	44322	30096	30096	56628	82326	71163	71022	42607
##	[701]	43057	47808	29236	40590	40590	77520	40211	63516	53034	22669
##	[711]	79908	69867	70287	37070	37070	38179	39922	39922	39922	64191
##	[721]	41658	40780	77568	66503	83512	31686	71322	27938	22263	34633
##	[731]	42162	162397	54162	41769	44953	37717	37717	40548	67893	69267
##	[741]	9548	68281	92556	48526	90273	19414	79761	53312	41039	88347
##	[751]	50150	50150	10245	33454	40887	55914	37292	52914	24884	67384

##	[761]	51651	82497	32632	33562	79146	79146	36143	27159	41275	76800
##	[771]	75794	30833	26518	91065	45183	48070	30372	30372	57036	66731
##	[781]	35682	10979	10979	57912	36026	35788	42243	46086	61346	33316
##	[791]	33316	21063	33697	25804	72117	61286	61286	65196	55635	42021
##	[801]	75251	35322	53374	87195	42395	82427	44602	94642	28359	55801
##	[811]	92955	56534	69627	78579	44964	30899	78825	25545	47111	49667
##	[821]	66000	46910	25224	77382	28420	32952	46149	83532	73059	73059
##	[831]	35701	89616	105471	78427	22518	59809	39660	52203	35860	35860
##	[841]	35860	35860	76842	15033	67353	17003	65735	45684	21918	51267
##	[851]	59868	51373	46692	157733	54432	79607	51813	17487	40706	62859
##	[861]	61074	43638	58116	62187	46344	77353	69759	21846	76532	70053
##	[871]	75342	27590	51148	33378	32644	33581	12393	42710	82347	82347
##	[881]	22554	17323	62204	78028	63887	32727	29103	61482	20587	20587
##	[891]	52190	48186	19329	34109	23477	47472	14849	26224	27238	NA
##	[901]	57967	51569	72550	44511	62972	62972	77845	36663	53204	21059
##	[911]	67087	73892	75114	42192	31454	22775	22775	31590	80134	80134
##	[921]	14188	18169	62503	46734	68126	52869	26576	75825	57045	34984
##	[931]	27803	69283	18746	70924	70924	69401	75012	85738	60934	27161
##	[941]	64325	18978	41124	39684	53378	37040	78710	92910	36283	79632
##	[951]	28587	88325	51412	63998	38593	27889	45889	56575	16014	61331
##	[961]	70893	2447	64849	42387	80617	54006	16531	30843	18988	81051
##	[971]	39747	34554	52854	81205	69084	69084	65352	22063	47691	29187
##	[981]	72504	33986	42386	42386	36957	13724	42664	37334	78497	78497
##	[991]	58821	58398	80950	30732	89572	57107	76467	NA	51315	43602
##	[1001]	38643	49090	70617	67536	27573	13533	50725	25851	57113	62847
##	[1011]	68627	25271	63810	63810	65488	36927	22108	72025	33183	23661
##	[1021]	25315	45938	66951	77298	77298	43018	75759	33996	33996	54137
##	[1031]	61223	69674	49494	51195	37284	37284	42011	70044	46053	24480
##	[1041]	54233	36130	56559	92859	44010	80124	80124	50183	64722	37395
##	[1051]	29543	19514	19514	59481	91712	80144	59973	68352	81169	61456
##	[1061]	22634	22634	23957	33471	52157	30298	56551	56551	65487	65487
##	[1071]	44213	40344	57867	25410	13260	42691	36138	57136	58275	54237
##	[1081]	34596	34596	74881	74881	44267	68487	68487	80982	69508	63342
##	[1091]	63342	82014	76320	64587	74190	32765	NA	22212	18358	26067
##	[1101]	22390	70440	60199	38547	19107	37929	16813	62187	57811	50943
##	[1111]	50943	53201	54414	26290	61250	28389	55250	37758	37758	37633
##	[1121]	33629	36947	69142	53154	65695	69805	36975	26228	37787	37087
##	[1131]	57247	36930	48721	43641	75903	71391	16927	27215	53858	30983

##	[1141]	43462	63943	26816	75484	33564	54348	54984	34738	68118	68118
##	[1151]	22682	51948	25293	25293	41551	56067	67433	22010	64831	62820
##	[1161]	62820	78353	57183	23478	30168	37155	16581	41850	81380	56386
##	[1171]	25818	15345	57954	50300	38725	10404	26868	72298	72905	47025
##	[1181]	29938	46681	46681	29548	62058	NA	36736	8820	14918	56715
##	[1191]	24882	24882	40442	64176	23536	35246	28442	48195	74116	78093
##	[1201]	59354	36301	77598	62710	36317	42767	43815	44421	48948	48948
##	[1211]	63120	55517	36864	NA	82582	79803	52569	82032	82032	65685
##	[1221]	65685	40451	31163	31163	35704	7500	36075	70038	70038	71796
##	[1231]	47821	45503	28647	37760	85431	72635	72635	77142	53172	36065
##	[1241]	73926	35790	39552	68316	102160	51390	72066	21675	48006	65106
##	[1251]	58401	29315	49389	51717	46779	28520	34377	54959	82384	46998
##	[1261]	58554	58554	57333	80685	52117	62220	64014	64014	46390	75507
##	[1271]	75507	59412	50616	35924	57937	73356	21255	58494	72460	68682
##	[1281]	68682	36959	46463	61825	61825	28691	28691	54058	84196	38620
##	[1291]	38620	3502	47743	80360	66303	52413	60714	31385	83257	80427
##	[1301]	38823	42014	42014	76140	41713	62670	70932	78939	44512	63967
##	[1311]	90842	25358	NA	28567	66373	66373	83844	83844	83844	83664
##	[1321]	27450	54880	71322	60000	65176	61010	31089	20518	34469	28332
##	[1331]	33456	38443	65333	82623	73691	74485	65748	24367	91700	79244
##	[1341]	82017	58138	59354	58482	29478	64474	31160	40321	40321	55239
##	[1351]	57513	85693	71855	7500	71969	28973	80872	84117	38741	38961
##	[1361]	75922	56962	56962	70165	65308	29298	63246	33402	96843	82333
##	[1371]	72643	32313	55284	15287	15287	51563	81929	53593	60432	66726
##	[1381]	62450	49514	43269	33762	60894	87771	87771	50737	26150	57304
##	[1391]	71107	53253	81320	87000	40049	63684	69016	7500	73803	46094
##	[1401]	85844	38452	38452	80395	87679	35416	44124	50729	94384	94384
##	[1411]	34412	31158	34838	56775	54356	72159	8028	55951	67911	67911
##	[1421]	20427	32303	55954	49187	49854	63693	63693	19510	52332	55593
##	[1431]	54591	74859	40851	58607	7500	52614	52614	67445	67445	67445
##	[1441]	36778	39453	42731	40760	69969	80317	66636	54730	42835	53843
##	[1451]	79530	63841	63841	63841	27100	27100	72309	50664	20194	76618
##	[1461]	45207	56721	27038	23763	37085	49605	1730	34824	34824	63206
##	[1471]	26759	55412	33590	70886	72679	65073	44300	56129	56129	38590
##	[1481]	53083	79689	34704	65316	41014	19740	23724	55212	55212	79800
##	[1491]	70647	38410	28071	54803	74293	74293	35684	34320	41967	49681
##	[1501]	49681	53187	86610	70951	24206	55357	58113	70643	64355	82576
##	[1511]	41644	72570	30630	22944	86979	NA	82072	52034	72354	28427

##	[1521]	28427	39858	67381	23529	51479	76068	22507	52413	50116	76624
##	[1531]	34587	62159	66294	5648	37235	85620	23976	41020	36108	57530
##	[1541]	62905	65846	34380	41638	63207	28320	43140	52531	40246	35196
##	[1551]	35196	70596	70596	52597	42394	81657	49154	34074	NA	70638
##	[1561]	38136	67432	40689	40689	101970	44325	61467	48178	16005	79593
##	[1571]	79593	61180	64449	77437	17649	53103	71819	74290	72282	47352
##	[1581]	47352	46524	50200	18222	77610	38097	44377	58684	66480	49638
##	[1591]	50616	26642	33235	23228	36732	82657	64140	44392	16185	38702
##	[1601]	54072	61787	34935	34935	59292	59292	57959	92491	42207	46984
##	[1611]	23442	46757	14515	80910	30545	44802	79941	79734	50353	77376
##	[1621]	78075	42473	25008	56223	47850	91820	40637	23295	84574	54165
##	[1631]	28510	67131	56242	86580	76005	54693	52750	36997	34230	47916
##	[1641]	60839	79930	55158	85485	75865	44375	60504	65492	67506	68805
##	[1651]	61209	66825	87188	84953	74637	86111	77632	31814	70116	48240
##	[1661]	40101	70503	79205	51766	62994	65640	38361	38361	19986	19986
##	[1671]	72903	45072	59062	61794	45143	45143	26877	36921	54342	62772
##	[1681]	75032	22419	22419	36550	75027	63159	54108	81795	59594	18492
##	[1691]	94472	70337	34853	NA	16269	50272	22804	22804	30631	78420
##	[1701]	81741	25721	31907	37859	30261	21645	21282	32144	62637	16626
##	[1711]	54690	54690	61284	23718	33419	82025	15056	48192	67309	66033
##	[1721]	28769	6835	32880	23616	19419	34961	64892	16248	43776	52074
##	[1731]	22123	35544	59060	65665	29819	81217	81217	26095	65706	31859
##	[1741]	93027	58710	49618	58308	40800	18690	18690	18690	28087	75702
##	[1751]	32892	32892	89120	27116	46107	54466	59686	82460	76045	83829
##	[1761]	83829	32889	33462	36627	28718	77981	60230	74918	40464	64857
##	[1771]	64857	63381	63381	66664	66664	86358	41443	57072	54178	48767
##	[1781]	51124	43020	49980	72906	25176	96876	31086	24072	76773	72190
##	[1791]	83837	17345	75154	28249	28249	69661	27922	45903	36273	70777
##	[1801]	17148	66476	45057	41120	NA	19485	21840	41473	31395	68117
##	[1811]	47682	83151	58086	84460	66565	23910	35893	71128	38683	37150
##	[1821]	61064	51650	55249	46854	14421	54132	153924	64813	61872	58025
##	[1831]	83790	34350	33438	21955	60491	59304	34421	34421	51411	71964
##	[1841]	64504	49678	60482	55260	67369	47175	68743	56937	83033	37697
##	[1851]	37697	44689	23162	63033	40304	40059	80067	36038	NA	40521
##	[1861]	66334	11012	31615	NA	69098	70792	17688	25443	57091	24336
##	[1871]	51111	44503	56850	66582	23148	82224	61917	43482	68274	42373
##	[1881]	NA	57537	80995	30992	30992	78642	74268	75072	56253	59925
##	[1891]	62807	62807	23626	46106	81168	27244	26091	26091	43142	38232

##	[1901]	92533	63285	38946	38946	40233	46831	67605	70379	77297	55759
##	[1911]	58217	51369	51369	91249	52513	44159	76081	76081	39996	55012
##	[1921]	77457	80695	71866	73450	55614	156924	30753	54603	22280	59385
##	[1931]	76998	80573	80573	24711	38829	71466	28839	53790	73705	22448
##	[1941]	12571	73113	35765	46734	66313	68148	44393	38197	51012	51012
##	[1951]	13672	41452	42081	14906	33812	46098	46098	46098	53230	88420
##	[1961]	39548	61278	81246	58330	58330	65777	70829	NA	59247	25930
##	[1971]	54753	28440	23559	62335	63777	59666	50965	71488	72228	46231
##	[1981]	75702	43300	78618	NA	29760	29760	50127	29604	49413	51876
##	[1991]	45837	85696	85696	59432	43185	60631	20559	50898	73807	14661
##	[2001]	36807	46772	81574	30772	46377	78128	83443	50523	47009	37054
##	[2011]	32011	82122	70300	21024	34916	27943	39791	62745	18589	85606
##	[2021]	26997	82332	38853	27071	90933	65814	74250	33181	45006	42315
##	[2031]	48699	59111	48726	37244	74854	60896	67546	67546	46923	38988
##	[2041]	49118	27683	39190	56981	23228	90000	55707	9255	42997	71232
##	[2051]	57744	33585	46891	46891	50387	76653	76542	76995	38415	22304
##	[2061]	75127	75127	39722	7144	39771	42618	51518	74716	33955	45736
##	[2071]	31632	60093	30822	41728	80739	49767	34445	67046	73538	80398
##	[2081]	52973	38998	38998	26490	26490	23331	60544	60544	49269	27203
##	[2091]	45204	67023	77583	76630	47353	48920	67472	64100	38872	72217
##	[2101]	15038	13624	42554	71952	71952	78041	63972	52852	62010	83151
##	[2111]	77870	46931	85072	73395	62307	58293	47889	24594	24594	69882
##	[2121]	24221	22327	61923	35791	35791	91172	35441	19656	7500	71626
##	[2131]	75330	34529	48789	34487	35688	35688	53700	73454	71113	NA
##	[2141]	31535	80812	40479	29440	38508	69520	18701	18701	69901	47703
##	[2151]	86037	71847	42033	51983	68655	68655	38285	79419	57338	42523
##	[2161]	42523	31761	65747	17144	17144	NA	84618	84618	66375	42213
##	[2171]	NA	51141	15072	7500	17117	59754	65569	28457	86857	86857
##	[2181]	68142	48794	63404	49669	55761	78331	37971	30368	34043	54111
##	[2191]	69932	39665	93404	38680	24570	77766	77766	64961	23748	45989
##	[2201]	23331	70421	23091	24683	157243	54222	73455	84906	84906	47025
##	[2211]	44529	31560	90687	66973	41154	64590	50611	50611	83273	81702
##	[2221]	24401	42557	7500	78416	44794	44794	69719	62568	42231	20130
##	[2231]	20130	42429	42429	36640	78901	66476	31056	46310	65819	94871

The income data is string instead of numeric value is due to the input format of the data, since it turns out it input as a character we can see that the format is not numeric. But we can change it through as.numeric function .

第八題

```
Nunique = apply(market,2,n_unique)
print(Nunique)
```

```
##           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           Response           Complain           Country
##           2              2              2              8
```

```
print('n_unique() counts the number of unique combinations in a set of one or more vectors in market data ')
```

```
## [1] "n_unique() counts the number of unique combinations in a set of one or more vectors in market data "
```

```
col = colnames(market)[Nunique == 2]
print(col)
```

```
## [1] "AcceptedCmp3" "AcceptedCmp4" "AcceptedCmp5" "AcceptedCmp1" "AcceptedCmp2"
## [6] "Response"      "Complain"
```

```
print('col print out the column with a n_unique() count equals to 2 ')
```

```
## [1] "col print out the column with a n_unique() count equals to 2 "
```

```
tbl = apply(market[,col],2,table)
print(tbl)
```

```
## AcceptedCmp3 AcceptedCmp4 AcceptedCmp5 AcceptedCmp1 AcceptedCmp2 Response
## 0          2077          2073          2077          2096          2210          1906
## 1          163          167          163          144          30          334
## Complain
## 0          2219
## 1           21
```

```
print('tbl print out the above variable "col" using the table function and make them as a frequency table. ')
```

```
## [1] "tbl print out the above variable \"col\" using the table function and make them as a frequency table. "
```

```
ratio = apply(market[,col],2,mean)
print(ratio)
```

```
## AcceptedCmp3 AcceptedCmp4 AcceptedCmp5 AcceptedCmp1 AcceptedCmp2      Response
## 0.07276786  0.07455357  0.07276786  0.06428571  0.01339286  0.14910714
## Complain
## 0.00937500
```

```
print('tbl print out the above variable "col" using the mean function and calculate their mean as a ratio ')
```

```
## [1] "tbl print out the above variable \"col\" using the mean function and calculate their mean as a ratio "
```


第九題

```
myf = function(var,byvar)
{
  by = unique(byvar)
  mout = NULL
  for (i in by) {
    s = subset(var,byvar == i)
    m = colMeans(s)
    mout = cbind(mout,m)
  }
  colnames(mout) = by
  mout
}

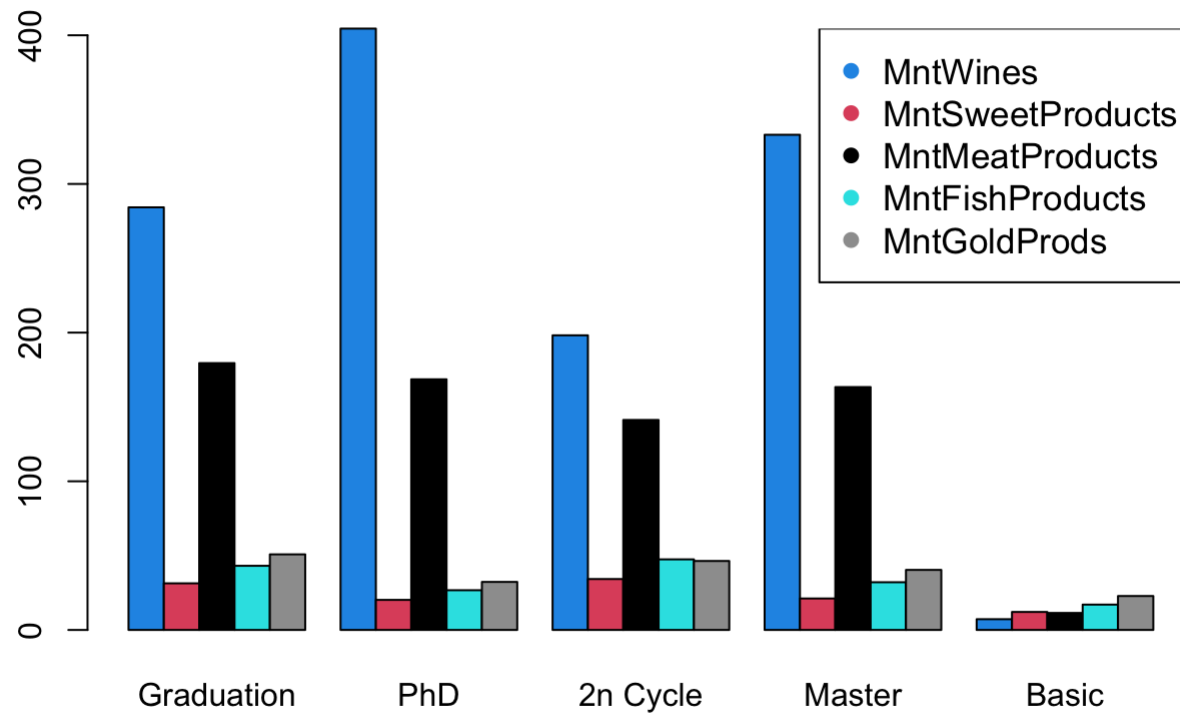
varlist = c("MntWines","MntSweetProducts",
            "MntMeatProducts","MntFishProducts",
            "MntGoldProds")
colspec = c(4,2,1,5,8)

tbl = myf(subset(market,,varlist),market$Education)

tbl
```

##	Graduation	PhD	2n Cycle	Master	Basic
## MntWines	284.26886	404.49588	198.18227	333.07568	7.240741
## MntSweetProducts	31.36735	20.22222	34.25123	21.17568	12.111111
## MntMeatProducts	179.48891	168.60288	141.25616	163.37838	11.444444
## MntFishProducts	43.14996	26.72840	47.48276	32.10000	17.055556
## MntGoldProds	50.84916	32.31070	46.39901	40.39730	22.833333

```
barplot(tbl, beside = T, col = colspec)
legend("topright", inset = c(0,0), legend = varlist,
      col = colspec, pch = 16, cex = 1.1)
```



```
agegrp = ifelse(market$Year_Birth < 1960,"<1960",
               ifelse(market$Year_Birth < 1970,"[1960,1970)",
                     ifelse(market$Year_Birth < 1980,"[1970,1980)", ">=1980")))

table(agegrp)
```

```
## agegrp
## [1960,1970) [1970,1980)    <1960    >=1980
##           506         740         570         424
```

agegrp is a condition statement, if the Year_birth data in market lower than 1960, then assign it as character "<1960", if the Year_birth data in market higher than 1960 and lower than 1970, then assign it as character "[1960,1970)", if the Year_birth data in market higher than 1970 and lower than 1980, then assign it as character "[1970,1980)", finally if the Year_birth data in market higher than 1980, then assign it as character ">=1980".

```
#Redefining myf()
myf = function(var, agegrp)
{
  by = unique(agegrp)
  mout = NULL
  for (i in by) {
    s = subset(var, agegrp == i)
    m = colMeans(s)
    mout = cbind(mout, m)
  }
  colnames(mout) = by
  mout
}

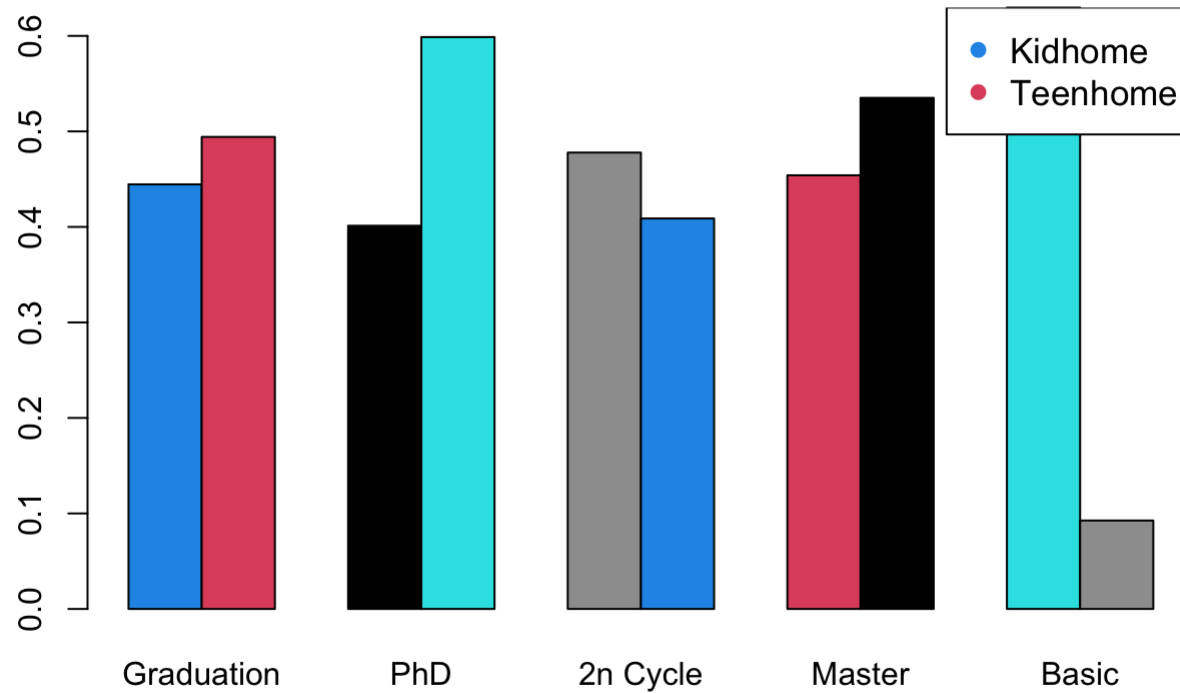
varlist = c("Kidhome", "Teenhome")
colspec = c(4, 2, 1, 5, 8)

tbl = myf(subset(market, , varlist), market$Education)

tbl
```

```
##           Graduation      PhD 2n Cycle      Master      Basic
## Kidhome   0.4445430 0.4012346 0.4778325 0.4540541 0.62962963
## Teenhome  0.4942325 0.5987654 0.4088670 0.5351351 0.09259259
```

```
barplot(tbl, beside = T, col = colspec)
legend("topright", inset = c(0,0), legend = varlist,
      col = colspec, pch = 16, cex = 1.1)
```



The bar plot showing the degree of

graduation that Kidhome and Teenhome frequency table information, we can see that In the basic degree,Kidhome has outperform Teenhome, and in the master,2n Cycle and graduation degree,Kidhome and Teenhome frequency table mean information are similar, In the PhD degree, Teenhome has outperform Kidhome in frequency table mean.