

IE6600 Data Visualization - Reema

Reema Yadav

12/7/2021

```
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(magrittr)  
library(tidyr)
```

```
##  
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:magrittr':  
##  
##   extract
```

```
library(ggplot2)  
library(skimr)
```

```
## Warning: package 'skimr' was built under R version 4.1.2
```

```
library(DataExplorer)
```

```
## Warning: package 'DataExplorer' was built under R version 4.1.2
```

```
library(plotly)
```

```
## Warning: package 'plotly' was built under R version 4.1.2
```

```
##  
## 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)
```

```
## Warning: package 'IRdisplay' was built under R version 4.1.2
```

```
#setwd("C:/Users/18572/Documents/Data Visualization/Assignment/HW6")  
#df <- read.csv(file="C:/Users/18572/Documents/Data Visualization/HW6/marketing_campaign.csv")  
main_df <- read.csv(file="marketing_campaign.csv", sep = "\t", header = T)
```

```
#plot_intro(main_df, title='Dataset')
```

```
head(main_df)
```

ID <int>	Year_Birth <int>	Education <chr>	Marital_Status <chr>	Income <int>	Kidhome <int>	Teenhome <int>	Dt_Customer <chr>	Recency <int>
1 5524	1957	Graduation	Single	58138	0	0	04-09-2012	58
2 2174	1954	Graduation	Single	46344	1	1	08-03-2014	38
3 4141	1965	Graduation	Together	71613	0	0	21-08-2013	26
4 6182	1984	Graduation	Together	26646	1	0	10-02-2014	26
5 5324	1981	PhD	Married	58293	1	0	19-01-2014	94
6 7446	1967	Master	Together	62513	0	1	09-09-2013	16

6 rows | 1-10 of 30 columns

```
skim(main_df)
```

Data summary

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

Variable type: character

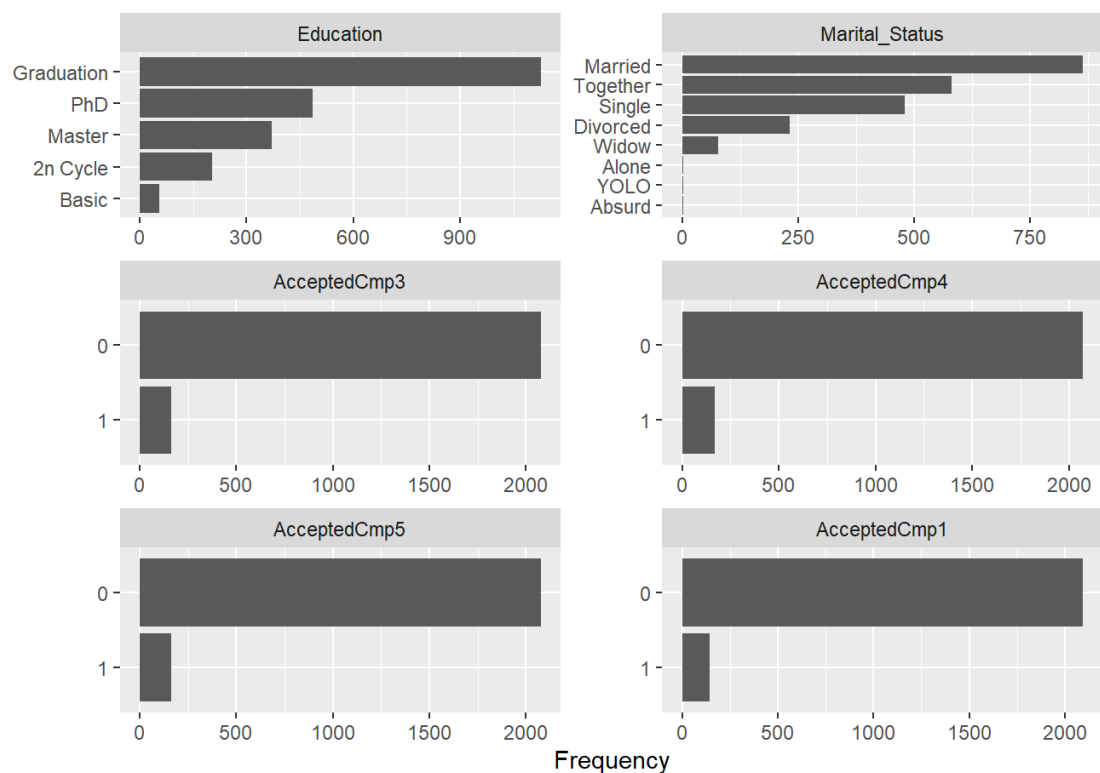
skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Education	0	1	3	10	0	5	0
Marital_Status	0	1	4	8	0	8	0
Dt_Customer	0	1	10	10	0	663	0

Variable type: numeric

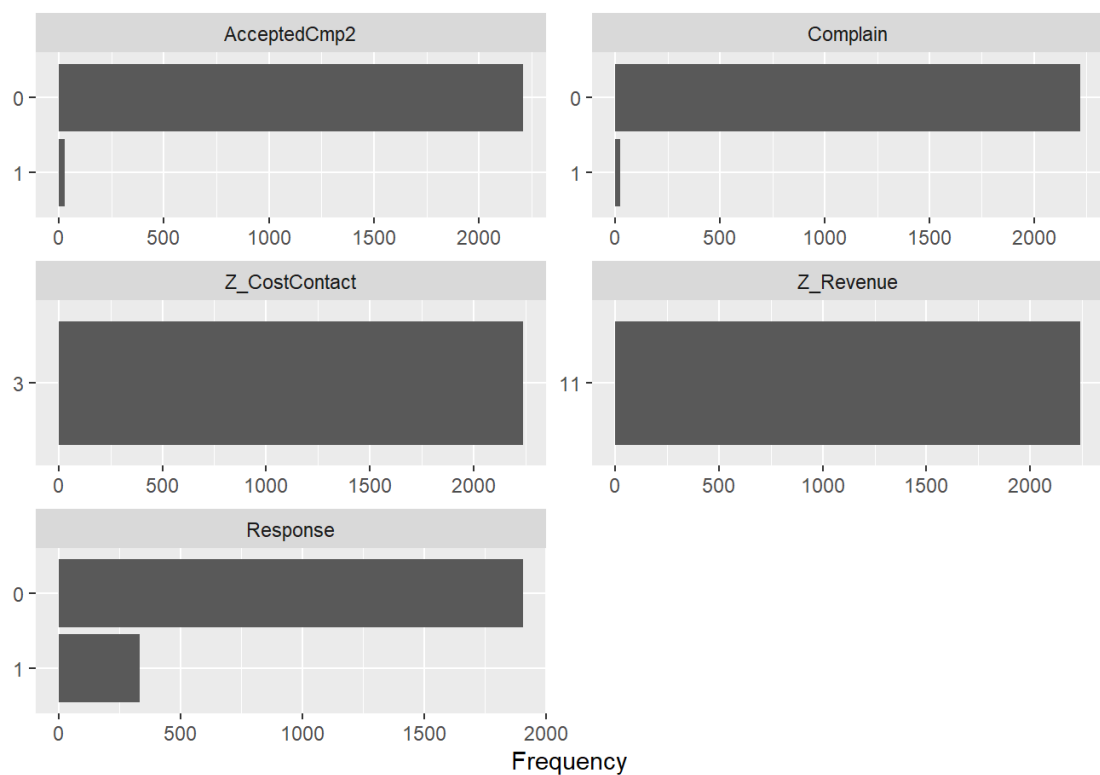
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ID	0	1.00	5592.16	3246.66	0	2828.25	5458.5	8427.75	11191	
Year_Birth	0	1.00	1968.81	11.98	1893	1959.00	1970.0	1977.00	1996	
Income	24	0.99	52247.25	25173.08	1730	35303.00	51381.5	68522.00	666666	
Kidhome	0	1.00	0.44	0.54	0	0.00	0.0	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	

```
plot_bar(main_df,ncol=2)
```

```
## 1 columns ignored with more than 50 categories.
## Dt_Customer: 663 categories
```



Page 1



Page 2

```
#Filling the missing value with median value
main_df$Income[is.na(main_df$Income)]=median(main_df$Income,na.rm=T)
```

```
#Calculating the age for each customer as a vector  
current_date= Sys.Date()  
current_year = format(current_date,format="%Y")  
current_year = as.integer(current_year)  
Age= c(current_year - main_df$Year_Birth)  
Age
```

```
## [1] 64 67 56 37 40 54 50 36 47 71 38 45 62 69 34 75 41 75
## [19] 72 36 39 42 72 67 70 52 45 35 32 56 32 58 51 69 75 70
## [37] 51 45 48 78 41 36 64 62 46 45 25 53 70 67 64 57 44 44
## [55] 43 66 55 33 39 53 39 46 69 70 73 50 54 42 63 51 67 48
## [73] 62 46 69 44 49 45 45 56 36 52 48 58 34 61 51 64 48 54
## [91] 64 64 48 49 51 38 61 49 70 38 61 62 42 76 62 42 57 52
## [109] 43 61 42 30 58 30 41 59 51 43 54 52 68 54 36 45 38 56
## [127] 37 68 60 65 69 63 52 58 57 54 45 62 48 56 29 45 59 47
## [145] 33 45 46 40 48 55 68 52 49 33 69 72 63 43 44 66 63 76
## [163] 37 46 48 55 68 62 33 64 56 39 51 62 54 46 73 51 62 70
## [181] 45 33 75 37 60 38 51 55 45 57 43 32 121 33 48 44 41 67
## [199] 67 48 69 53 43 44 56 37 33 58 55 67 65 67 53 47 36 44
## [217] 43 73 67 41 33 52 51 41 59 43 45 69 49 35 55 59 50 70
## [235] 56 46 39 63 48 128 53 63 50 31 63 57 49 68 33 52 46 36
## [253] 47 63 54 47 60 57 57 64 42 53 51 43 69 55 58 51 51 51
## [271] 68 38 46 61 58 58 65 40 35 53 40 45 75 47 46 47 69 65
## [289] 55 72 37 52 41 50 56 69 33 35 39 56 63 65 43 33 37 60
## [307] 55 74 57 29 45 54 32 45 55 65 49 36 70 51 50 47 56 60
## [325] 68 64 45 37 71 35 38 32 65 50 38 61 53 49 56 122 33 35
## [343] 52 71 48 69 56 51 50 45 64 35 55 47 45 43 66 52 78 73
## [361] 64 52 43 43 54 52 61 45 59 46 37 46 51 55 49 46 51 32
## [379] 60 52 52 36 67 48 40 50 53 61 72 53 42 42 28 75 53 30
## [397] 50 37 34 41 65 56 47 61 66 56 49 50 47 58 39 45 66 34
## [415] 59 78 46 27 40 50 48 38 39 47 80 68 35 32 32 53 60 32
## [433] 54 57 51 65 55 72 50 35 32 73 75 32 47 62 49 57 45 68
## [451] 66 52 50 41 45 49 55 37 69 39 57 72 63 58 46 52 66 51
## [469] 70 50 46 41 40 43 33 65 49 44 49 44 48 49 47 46 61 57
## [487] 45 47 52 69 63 45 66 57 77 59 43 50 72 77 71 60 31 36
## [505] 65 68 45 32 29 62 40 42 57 62 75 46 60 31 62 77 56 34
## [523] 30 68 34 35 63 44 57 61 46 52 44 59 39 70 54 54 69 40
## [541] 44 52 37 39 45 31 57 41 53 48 58 32 50 58 72 69 43 39
## [559] 51 62 43 41 50 51 58 71 59 70 42 51 47 53 43 51 40 44
## [577] 53 51 55 65 47 70 54 70 38 50 66 60 66 62 37 46 56 44
## [595] 49 46 52 52 43 40 60 61 66 73 55 35 48 65 46 56 52 77
## [613] 48 54 52 47 56 45 43 61 60 50 72 43 66 45 46 61 44 44
## [631] 51 63 51 61 48 59 69 59 71 50 61 73 62 57 74 50 51 51
## [649] 44 63 61 36 47 71 66 46 68 63 74 55 58 36 46 41 65 62
## [667] 47 34 53 50 57 59 51 41 46 47 58 42 36 41 58 54 39 66
## [685] 54 45 49 39 32 52 61 51 63 69 72 60 26 72 60 61 48 43
## [703] 43 33 53 35 51 70 73 50 35 56 46 68 39 46 61 43 51 63
## [721] 56 67 37 56 45 55 66 45 56 30 72 68 43 73 48 72 48 69
## [739] 32 52 68 65 71 43 57 69 46 26 51 67 49 63 46 52 42 43
## [757] 49 57 42 62 40 54 33 43 69 66 44 34 56 65 42 70 56 49
## [775] 64 68 56 63 66 50 47 45 51 49 49 35 61 53 43 45 76 64
## [793] 71 43 60 59 67 56 48 53 51 65 70 77 63 45 50 64 49 49
## [811] 40 49 46 66 35 51 45 67 49 43 41 49 60 58 62 62 67 59
## [829] 63 37 45 29 54 41 54 65 49 41 41 44 71 47 33 40 61 51
## [847] 30 65 58 41 47 67 64 55 47 42 48 36 55 32 52 52 34 32
## [865] 46 48 55 35 43 42 38 62 45 40 43 68 50 38 65 43 49 46
## [883] 46 40 49 51 68 62 59 43 48 56 38 30 78 49 65 63 47 35
## [901] 63 51 54 48 52 44 60 50 47 35 55 29 58 49 27 67 61 39
## [919] 67 67 66 28 52 73 35 42 43 39 28 45 47 32 70 32 69 50
## [937] 50 51 50 46 69 59 38 68 35 34 39 42 44 46 42 45 48 50
## [955] 45 45 56 69 48 64 54 59 52 46 41 62 48 48 47 41 33 53
## [973] 33 62 50 59 62 56 69 67 61 56 42 43 71 47 43 52 45 65
## [991] 67 49 40 32 67 26 43 64 48 65 65 34 40 34 67 67 68 69
## [1009] 53 58 42 45 43 53 50 70 56 52 48 45 48 47 68 55 43 54
## [1027] 43 34 63 57 50 50 52 44 48 41 63 49 50 68 62 67 30 59
## [1045] 35 56 55 61 37 55 56 55 72 51 38 36 47 70 69 34 72 63
## [1063] 34 48 57 60 29 56 65 43 50 55 66 68 63 60 40 45 53 43
## [1081] 56 50 65 39 51 44 51 63 38 63 48 47 62 49 56 42 63 35
```

```

## [1099] 39 42 46 42 35 51 49 61 69 51 47 56 67 38 48 61 37 50
## [1117] 38 38 55 43 53 42 48 42 67 59 55 44 36 69 70 54 43 45
## [1135] 36 47 53 59 32 60 68 32 43 59 76 47 40 65 67 35 78 57
## [1153] 42 47 61 58 45 47 60 70 56 65 56 27 47 49 53 53 73 52
## [1171] 25 47 31 64 42 61 69 40 40 30 71 64 60 63 50 36 56 69
## [1189] 59 47 51 52 61 69 52 53 56 52 49 60 44 62 63 42 64 65
## [1207] 56 31 64 56 64 66 49 52 49 46 56 58 30 58 35 48 64 40
## [1225] 63 62 66 35 46 66 50 35 43 38 45 50 65 49 32 69 37 54
## [1243] 59 74 43 50 29 53 48 46 56 69 54 49 58 51 66 46 45 52
## [1261] 48 38 39 61 69 46 58 51 73 63 77 58 49 55 53 50 46 57
## [1279] 43 46 38 50 64 70 70 51 49 52 49 50 65 50 48 44 58 67
## [1297] 43 55 75 43 50 68 36 51 42 55 45 56 39 49 61 65 56 66
## [1315] 32 59 56 48 49 37 61 63 53 44 47 59 39 46 39 76 42 52
## [1333] 59 73 43 46 39 38 69 39 59 44 72 31 39 46 62 38 45 59
## [1351] 65 28 34 56 39 59 43 75 58 50 53 29 47 71 44 63 45 48
## [1369] 53 63 46 61 56 61 52 61 46 36 50 51 66 65 63 57 48 73
## [1387] 49 45 41 42 49 48 49 46 49 65 47 49 53 52 60 49 68 64
## [1405] 51 54 70 68 71 68 44 49 52 77 31 42 55 73 49 48 49 49
## [1423] 55 52 62 51 47 49 63 49 38 48 36 48 54 42 60 35 57 38
## [1441] 69 72 42 73 75 63 50 64 39 48 49 56 68 54 56 52 50 49
## [1459] 51 47 71 62 66 43 35 58 68 69 71 51 51 44 55 52 63 67
## [1477] 65 51 34 60 42 61 40 41 45 45 51 75 66 49 50 71 33 43
## [1495] 31 62 44 52 62 33 46 48 44 48 66 47 49 46 50 63 33 46
## [1513] 56 35 44 41 59 43 49 51 38 50 45 48 48 45 73 63 34 61
## [1531] 46 63 46 46 33 33 32 62 50 54 32 60 43 68 57 35 34 58
## [1549] 64 46 70 56 65 59 47 54 59 45 60 45 38 45 33 67 36 69
## [1567] 57 67 71 64 75 67 30 35 44 61 39 74 29 45 65 74 71 55
## [1585] 34 48 69 56 41 34 59 66 33 56 54 48 61 37 50 46 56 65
## [1603] 66 61 60 58 44 31 48 57 33 43 31 46 44 37 38 34 56 65
## [1621] 56 50 39 62 58 69 50 58 39 45 74 52 62 43 73 66 67 76
## [1639] 52 45 52 57 40 42 40 61 47 40 54 71 39 60 58 44 74 43
## [1657] 67 67 49 40 63 53 45 55 32 36 54 62 51 54 36 70 35 66
## [1675] 51 30 46 42 45 41 43 50 66 38 69 52 41 53 50 32 70 34
## [1693] 36 40 31 69 50 38 64 70 45 66 32 48 48 37 65 56 43 48
## [1711] 35 62 43 55 29 37 65 44 47 45 42 38 30 59 39 42 66 56
## [1729] 60 70 57 62 66 55 63 67 54 58 62 50 57 38 76 74 52 69
## [1747] 74 75 52 72 60 47 41 72 50 42 53 50 66 49 57 42 46 44
## [1765] 35 65 41 39 60 46 47 75 36 69 54 66 36 44 48 37 42 45
## [1783] 53 62 50 54 48 38 42 42 71 58 59 45 69 61 66 44 57 40
## [1801] 45 47 70 55 70 63 55 57 65 47 55 58 52 29 67 38 39 70
## [1819] 65 60 59 50 50 53 43 52 56 35 63 46 34 62 59 47 44 43
## [1837] 51 64 66 67 39 68 53 64 56 46 58 31 55 33 26 46 43 60
## [1855] 37 50 70 45 49 36 55 46 42 50 39 48 53 37 50 44 54 57
## [1873] 49 73 69 34 62 63 70 46 37 52 53 52 36 37 64 50 46 39
## [1891] 33 32 36 53 47 72 43 67 76 71 57 38 34 50 53 67 50 35
## [1909] 62 41 61 51 50 54 58 53 31 40 45 56 44 49 65 78 62 44
## [1927] 52 51 64 56 47 74 39 53 70 46 65 72 62 48 37 71 44 37
## [1945] 45 70 39 47 53 67 81 45 70 45 57 71 45 65 31 71 56 59
## [1963] 43 69 55 74 49 58 72 43 55 38 50 46 50 52 58 53 42 37
## [1981] 37 47 70 59 46 64 43 30 35 65 58 50 74 63 61 41 46 46
## [1999] 48 41 51 54 51 55 49 44 51 57 62 55 63 35 44 40 63 46
## [2017] 38 54 70 49 43 39 65 69 56 42 56 68 72 36 74 66 45 46
## [2035] 37 71 44 70 37 39 30 51 40 42 50 51 62 48 51 74 70 46
## [2053] 62 66 68 56 48 45 71 52 52 40 39 43 47 70 73 69 37 42
## [2071] 72 65 52 34 38 51 44 61 50 67 48 66 61 33 78 52 29 62
## [2089] 41 50 47 52 45 28 72 71 46 65 45 46 45 72 37 47 49 35
## [2107] 59 51 72 60 66 40 68 47 67 52 40 69 59 50 45 45 41 65
## [2125] 64 49 55 35 56 47 45 45 72 41 64 56 52 51 67 55 47 50
## [2143] 53 64 48 64 50 61 48 59 63 49 45 52 45 38 57 46 66 56
## [2161] 57 41 64 72 29 47 67 31 37 40 56 51 46 70 66 46 54 48
## [2179] 50 66 47 49 57 43 64 61 49 65 64 49 38 50 74 64 64 69
## [2197] 47 48 67 46 31 52 48 43 31 53 41 67 58 58 66 40 71 26

```

```
## [2215] 45 67 38 73 43 36 42 39 42 58 51 53 42 59 43 49 37 51
## [2233] 35 44 47 54 75 40 65 67
```

```
#Adding Age column and then calculating maximum age
main_df['Age']=Age
max(main_df$Age)
```

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

```
#Calculating median age
median(main_df$Age)
```

```
## [1] 51
```

```
#Calculating average age
mean(main_df$Age)
```

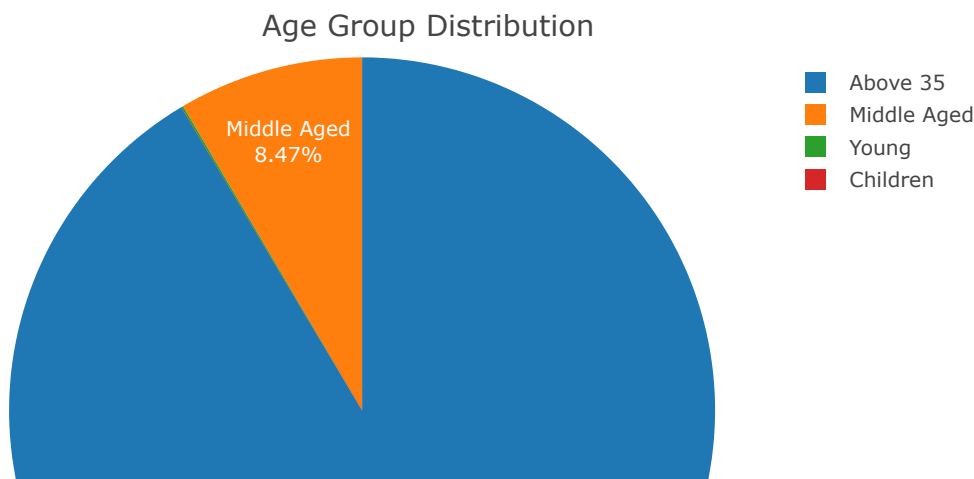
```
## [1] 52.1942
```

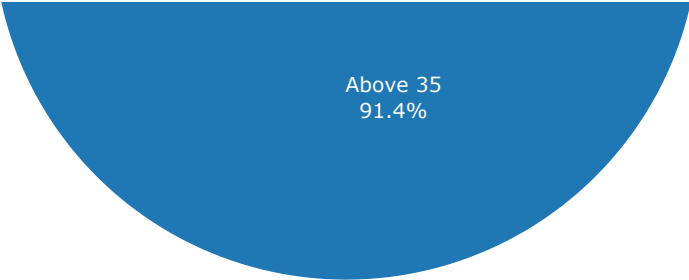
```
#Understanding the income based on Marital_Status
#ggplot(df) +
# geom_point(aes(x = Income, y = Marital_Status), color = '#FF0000') + labs(x="Income")
```

```
children = main_df %>% filter(Age < 15) %>% summarize(n())
young = main_df %>% filter(15 <= Age & Age <= 25) %>% summarize(n())
middle_aged = main_df %>% filter(25 < Age & Age <= 35) %>% summarize(n())
above_35 = main_df %>% filter(Age > 36) %>% summarize(n())
children = as.integer(children)
young = as.integer(young)
middle_aged = as.integer(middle_aged)
above_35 = as.integer(above_35)

count = c(children,young,middle_aged,above_35)
labels_age = c('Children','Young','Middle Aged','Above 35')
```

```
fig1 = plot_ly(labels = ~labels_age, values = ~count, type = 'pie',textposition='inside',
textinfo = 'label+percent',insidetextfont = list(color = '#FFFFFF'))
fig1 = fig1 %>% layout(title = 'Age Group Distribution',
xaxis = list(showgrid = FALSE), yaxis = list(showgrid = FALSE))
fig1
```





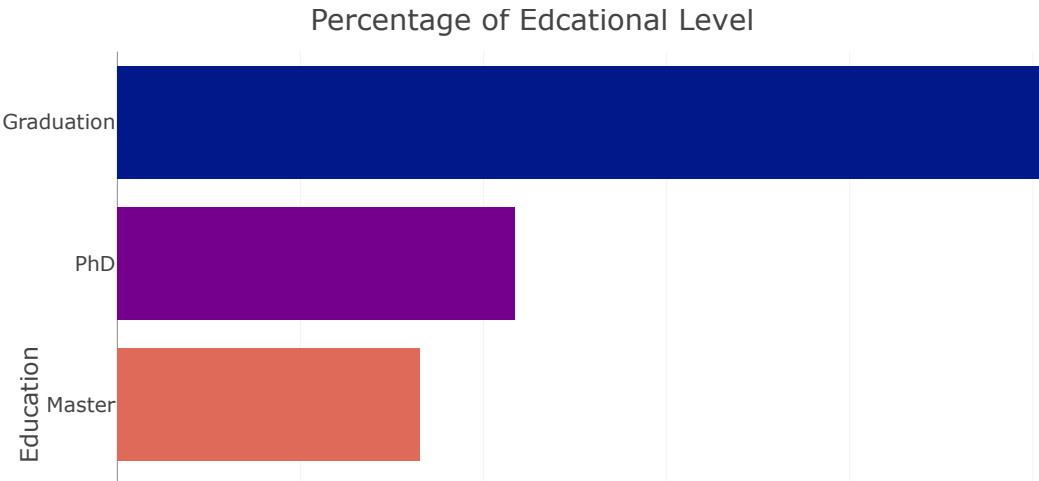
```
#Unique Education values
unique(main_df$Education)
```

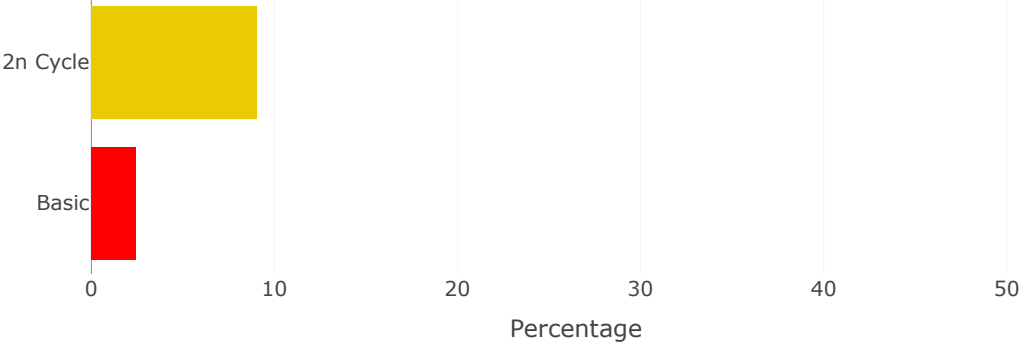
```
## [1] "Graduation" "PhD"      "Master"     "Basic"      "2n Cycle"
```

```
#Creating a table for Education
table_ed = sort(table(main_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<chr>	per_ed<dbl>
Graduation	50.312500
PhD	21.696429
Master	16.517857
2n Cycle	9.062500
Basic	2.410714
5 rows	

```
fig2 = plot_ly(df_ed, y=~labels_ed,x=~per_ed,type='bar',marker= list(color=c('#001889','#74008D','#E06A5A','#EACA00','#FF0000'))))
fig2 = fig2 %>% layout(title="Percentage of Edcational Level",xaxis = list(title='Percentage'),
yaxis = list(title='Education',categoryorder = "array",
categoryarray = rev(labels_ed)))
fig2
```





```
#How many children's customers have?
#Unique values in Kidhome
unique(main_df$Kidhome)
```

```
## [1] 0 1 2
```

```
#Unique values in Teenhome
unique(main_df$Teenhome)
```

```
## [1] 0 1 2
```

```
#Merge
main_df['Children']=main_df['Kidhome']+main_df['Teenhome']
```

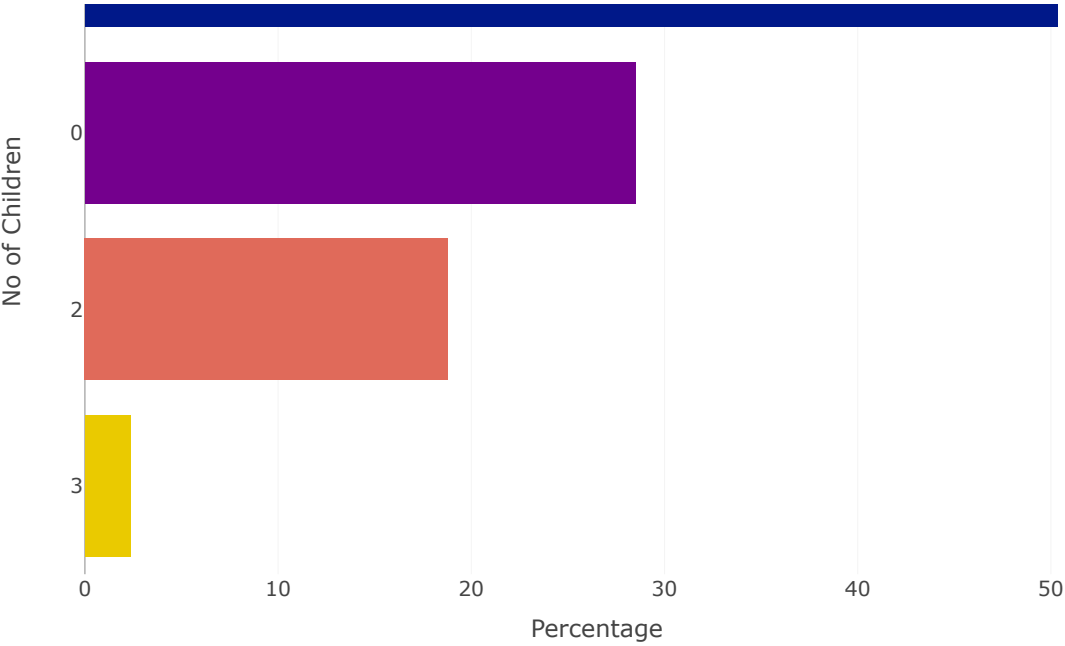
```
#Creating table function for children column
table_cd=sort(table(main_df$Children),decreasing=T)
per_cd = as.vector(prop.table(table_cd)*100)
labels_cd = c('1','0','2','3')
df_cd = data.frame(labels_cd,per_cd)
df_cd
```

labels_cd<chr>	per_cd<dbl>
1	50.357143
0	28.482143
2	18.794643
3	2.366071
4 rows	

```
fig3 = plot_ly(df_cd, y=~labels_cd,x=~per_cd,type='bar',orientation='h',
marker= list(color=c('#001889','#74008D','#E06A5A','#EACA00')))
fig3 = fig3 %>% layout(title="Percentage of No of Children",xaxis = list(title='Percentage'),
yaxis = list(title='No of Children',categoryorder = "array",
categoryarray = rev(labels_cd)))
fig3
```

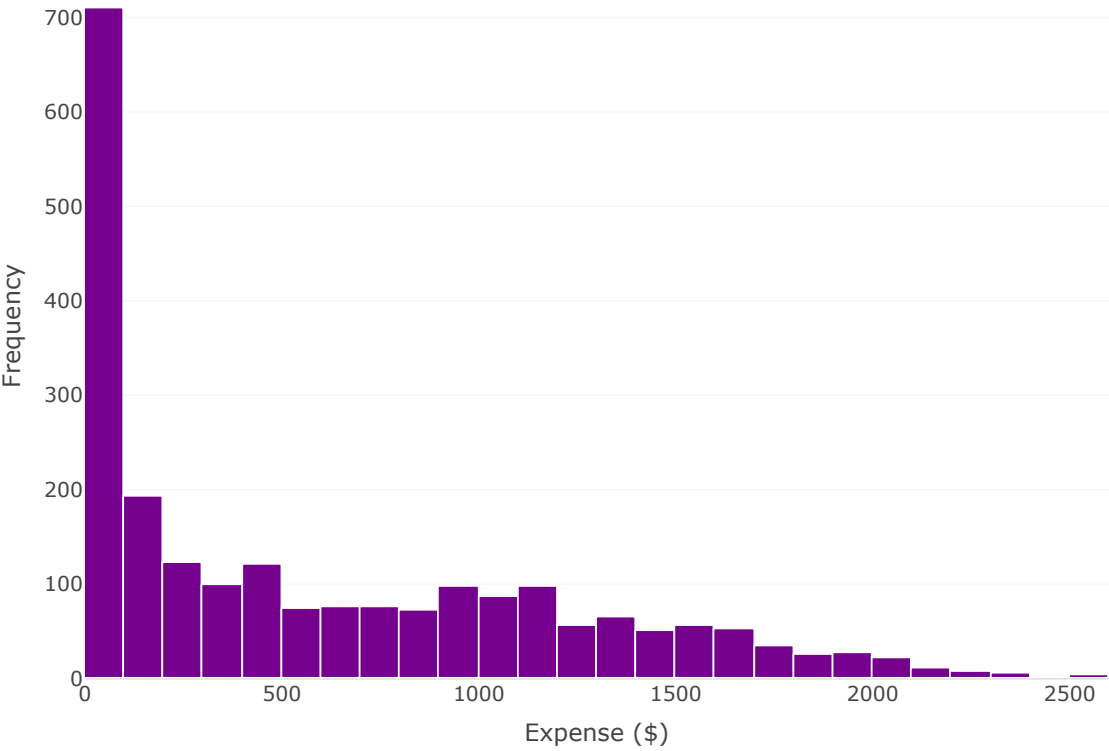
Percentage of No of Children





```
main_df['Expenses'] = main_df['MntWines'] + main_df['MntFruits'] + main_df['MntMeatProducts'] +
main_df['MntFishProducts'] + main_df['MntSweetProducts'] + main_df['MntGoldProds']
fig4 = plot_ly(main_df,x=~Expenses,type='histogram',marker = list(color = '#74008D',
line = list(color = '#FFFFFF',width = 1)))
fig4 = fig4 %>% layout(title="Frequency of Customer's Expenses",xaxis=list(title='Expense ($)'),yaxis=list(title=
'Frequency'),xaxis = list(gridcolor = 'ffff'))
fig4
```

Frequency of Customer's Expenses



```
#Maximum Income
max(main_df$Income)
```

```
## [1] 666666
```

```
#Average Income  
mean(main_df$Income)
```

```
## [1] 52237.98
```

```
fig5= plot_ly(main_df,x=~Income,type='histogram',marker = list(color = '#EACA00',  
line = list(color = '#FFFFFF',width = 1)))  
fig5 = fig5 %>% layout(title="Frequency of Income",xaxis=list(title='Income'),yaxis=list(title='Frequency'))  
fig5
```

Frequency of Income

