R Notebook

Code

Stephanie Omwanda

Hide

```
# load libraries
library(readr)
library(dplyr)
```

Hide

```
# Loading the csv file
df = read_csv('advertising.csv')
```

```
Parsed with column specification:
cols(
  `Daily Time Spent on Site` = [32mcol_double()[39m,
  Age = [32mcol_double()[39m,
  `Area Income` = [32mcol_double()[39m,
  `Daily Internet Usage` = [32mcol_double()[39m,
  `Ad Topic Line` = [31mcol_character()[39m,
  City = [31mcol_character()[39m,
  Male = [32mcol_double()[39m,
  Country = [31mcol_character()[39m,
  Timestamp = [34mcol_datetime(format = "")[39m,
  `Clicked on Ad` = [32mcol_double()[39m
)
```

Hide

```
# Previewing the first five rows of the dataframe
head(df)
```

Daily Time Spent on Site <dbl>	... <dbl>	Area Income <dbl>	Daily Internet Usage <dbl>
68.95	35	61833.90	256.09
80.23	31	68441.85	193.77
69.47	26	59785.94	236.50
74.15	29	54806.18	245.89
68.37	35	73889.99	225.58
59.99	23	59761.56	226.74

6 rows | 1-4 of 10 columns

Hide

```
# show information on dataset
str(df)
```

```
Classes ‘spec_tbl_df’, ‘tbl_df’, ‘tbl’ and 'data.frame':    1000 obs. of  10 varia
bles:
 $ Daily Time Spent on Site: num  69 80.2 69.5 74.2 68.4 ...
 $ Age                     : num  35 31 26 29 35 23 33 48 30 20 ...
 $ Area Income             : num  61834 68442 59786 54806 73890 ...
 $ Daily Internet Usage    : num  256 194 236 246 226 ...
 $ Ad Topic Line           : chr  "Cloned 5thgeneration orchestration" "Monitored
national standardization" "Organic bottom-line service-desk" "Triple-buffered reci
procal time-frame" ...
 $ City                    : chr  "Wrightburgh" "West Jodi" "Davidton" "West Terri
furt" ...
 $ Male                    : num  0 1 0 1 0 1 0 1 1 1 ...
 $ Country                 : chr  "Tunisia" "Nauru" "San Marino" "Italy" ...
 $ Timestamp               : POSIXct, format: "2016-03-27 00:53:11" "2016-04-04 01
:39:02" "2016-03-13 20:35:42" ...
 $ Clicked on Ad           : num  0 0 0 0 0 0 0 1 0 0 ...
 - attr(*, "spec")=
  .. cols(
  ..   `Daily Time Spent on Site` = [32mcol_double()[39m,
  ..   Age = [32mcol_double()[39m,
  ..   `Area Income` = [32mcol_double()[39m,
  ..   `Daily Internet Usage` = [32mcol_double()[39m,
  ..   `Ad Topic Line` = [31mcol_character()[39m,
  ..   City = [31mcol_character()[39m,
  ..   Male = [32mcol_double()[39m,
  ..   Country = [31mcol_character()[39m,
  ..   Timestamp = [34mcol_datetime(format = "")[39m,
  ..   `Clicked on Ad` = [32mcol_double()[39m
  .. )
```

Hide

```
# checking for the statistical summary
summary(df)
```

```
 Daily Time Spent on Site      Age         Area Income    Daily Internet Usage Ad
Topic Line
 Min.   :32.60            Min.   :19.00   Min.   :13996   Min.   :104.8        Len
gth:1000
 1st Qu.:51.36            1st Qu.:29.00   1st Qu.:47032   1st Qu.:138.8        Cla
ss :character
 Median :68.22            Median :35.00   Median :57012   Median :183.1        Mod
e  :character
 Mean   :65.00            Mean   :36.01   Mean   :55000   Mean   :180.0
 3rd Qu.:78.55            3rd Qu.:42.00   3rd Qu.:65471   3rd Qu.:218.8
 Max.   :91.43            Max.   :61.00   Max.   :79485   Max.   :270.0
     City                 Male         Country            Timestamp
 Length:1000        Min.   :0.000   Length:1000        Min.   :2016-01-01 02:52:10
 Class :character   1st Qu.:0.000   Class :character   1st Qu.:2016-02-18 02:55:42
 Mode  :character   Median :0.000   Mode  :character   Median :2016-04-07 17:27:29
                    Mean   :0.481                      Mean   :2016-04-10 10:34:06
                    3rd Qu.:1.000                      3rd Qu.:2016-05-31 03:18:14
                    Max.   :1.000                      Max.   :2016-07-24 00:22:16
 Clicked on Ad
 Min.   :0.0
 1st Qu.:0.0
 Median :0.5
 Mean   :0.5
 3rd Qu.:1.0
 Max.   :1.0
```

Hide

```
# determine the dimensions of the dataset
dim(df)
```

```
[1] 1000   10
```

The dataset is seen to have 1000 observations and 10 variables.

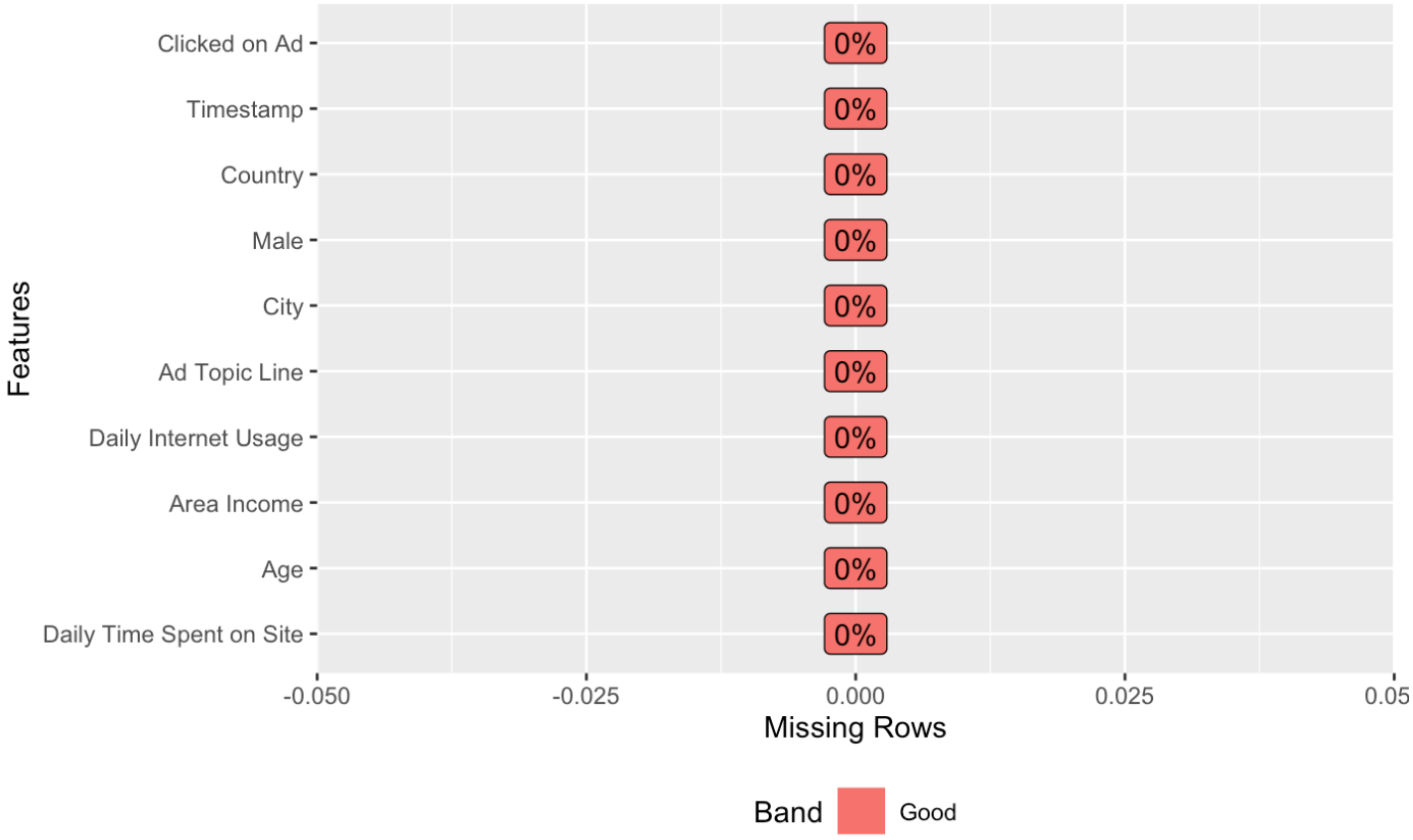
Hide

```
# checking if there exists null values by calculating the sum of the null values p
er column
colSums((is.na(df)))
```

```
Daily Time Spent on Site                 Age                Area Income
                       0                   0                           0
    Daily Internet Usage         Ad Topic Line                        City
                       0                   0                           0
                    Male             Country                   Timestamp
                       0                   0                           0
           Clicked on Ad
                       0
```

Hide

```
# a plot showing missing values
library(DataExplorer)
plot_missing(df)
```



Hide

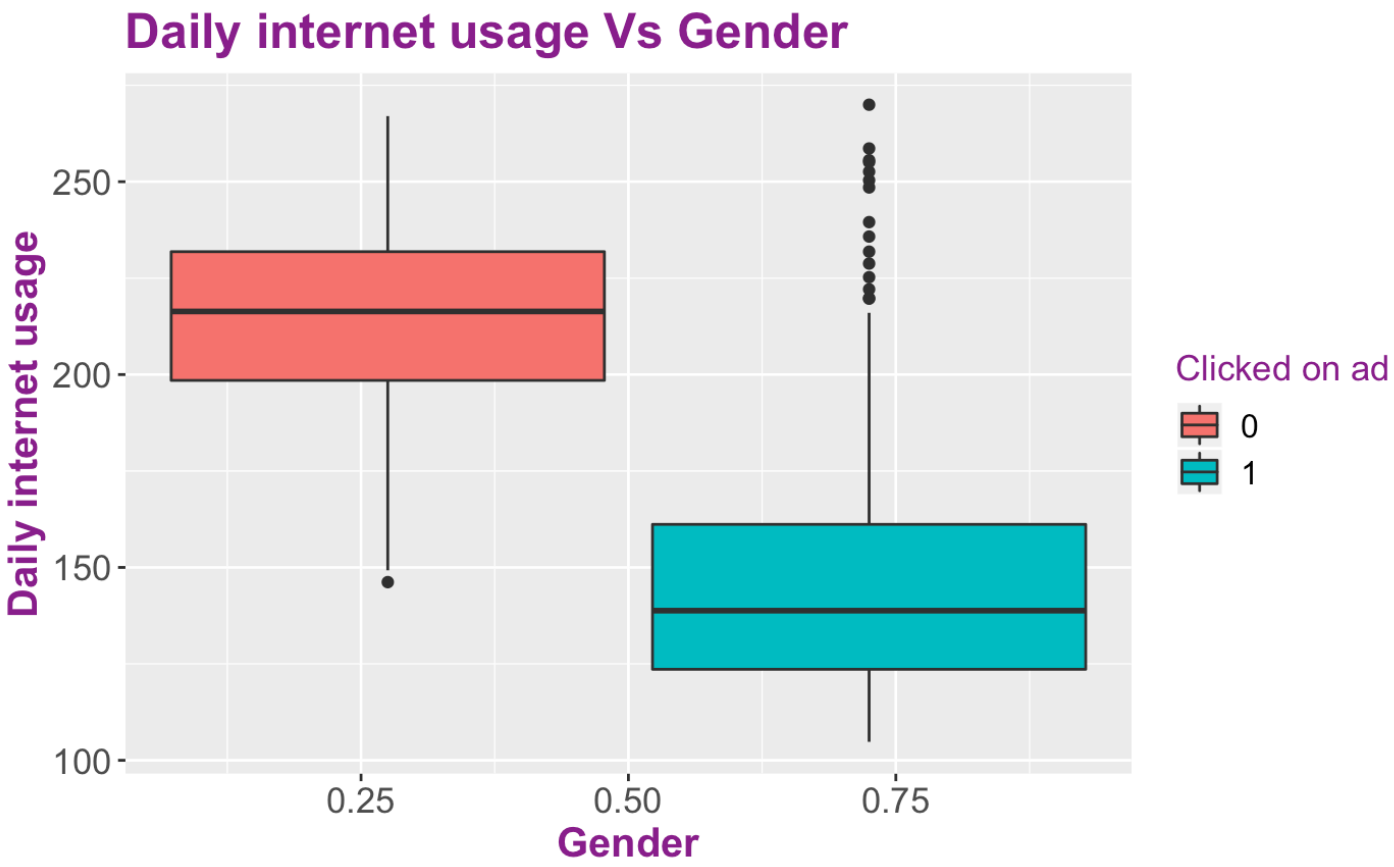
```
# checking for duplicates in the dataset by assigning a variable 'duplicates'
duplicates <- df[duplicated(df),]
duplicates
```

0 rows | 1-8 of 10 columns

Hide

```
# Plotting boxplots
options(repr.plot.width = 13, repr.plot.height = 7)
ggplot(data = df, aes(x = gender, y = daily_internet_usage)) +
    geom_boxplot(aes(fill = factor(clicked_on_ad))) +
    labs(title = 'Daily internet usage Vs Gender', y = 'Daily internet usage', x =
'Gender', fill = 'Clicked on ad') +
    scale_color_brewer(palette = 'cool') +
    theme(plot.title = element_text(size = 18, face = 'bold', color = 'darkmagenta
'),
              axis.title.x = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.title.y = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.text.x = element_text(size = 13),
              axis.text.y = element_text(size = 13),
              legend.title = element_text(size = 13, color = 'darkmagenta'),
              legend.text = element_text(size = 12))
```

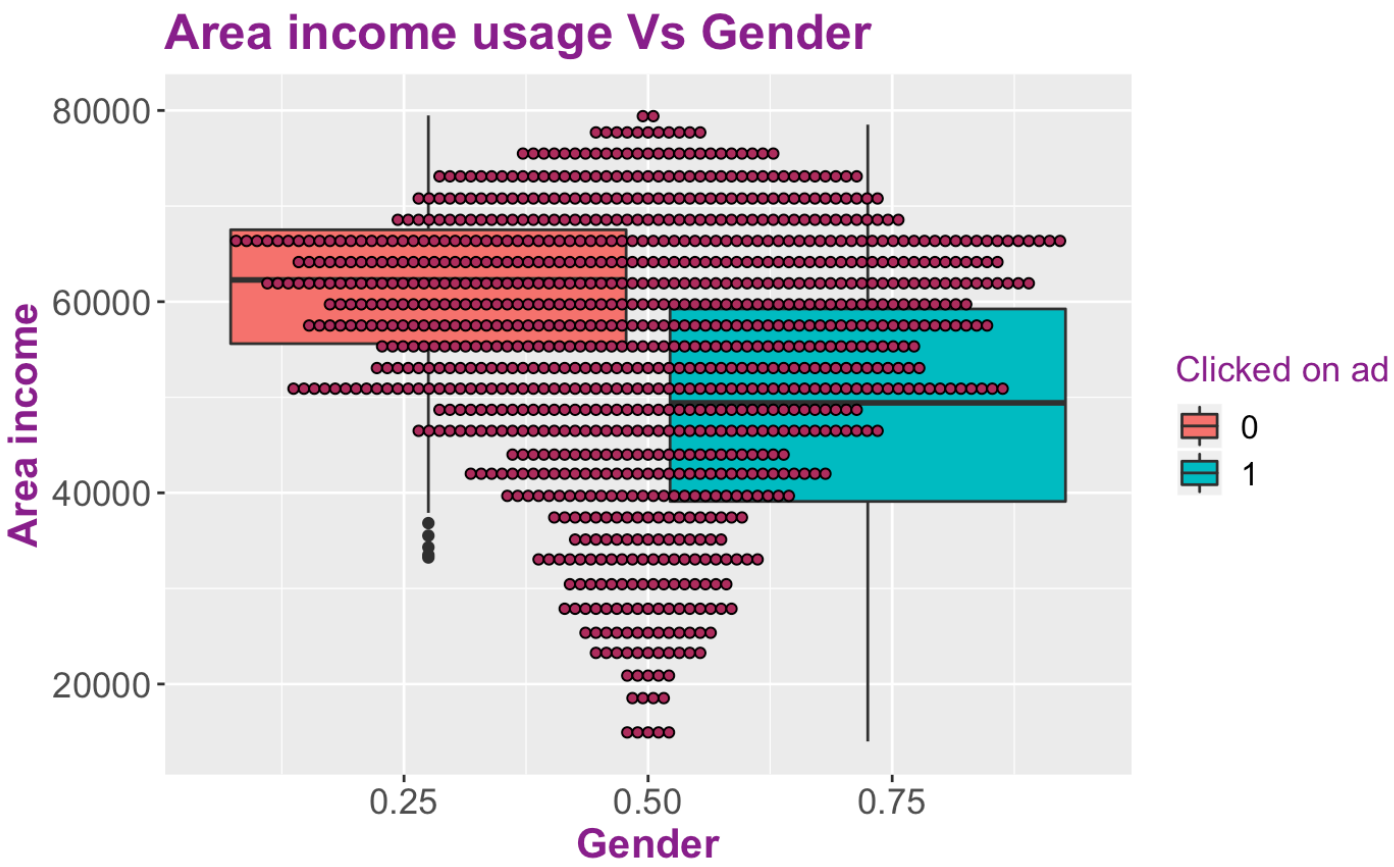
```
Unknown palette cool
```



Hide

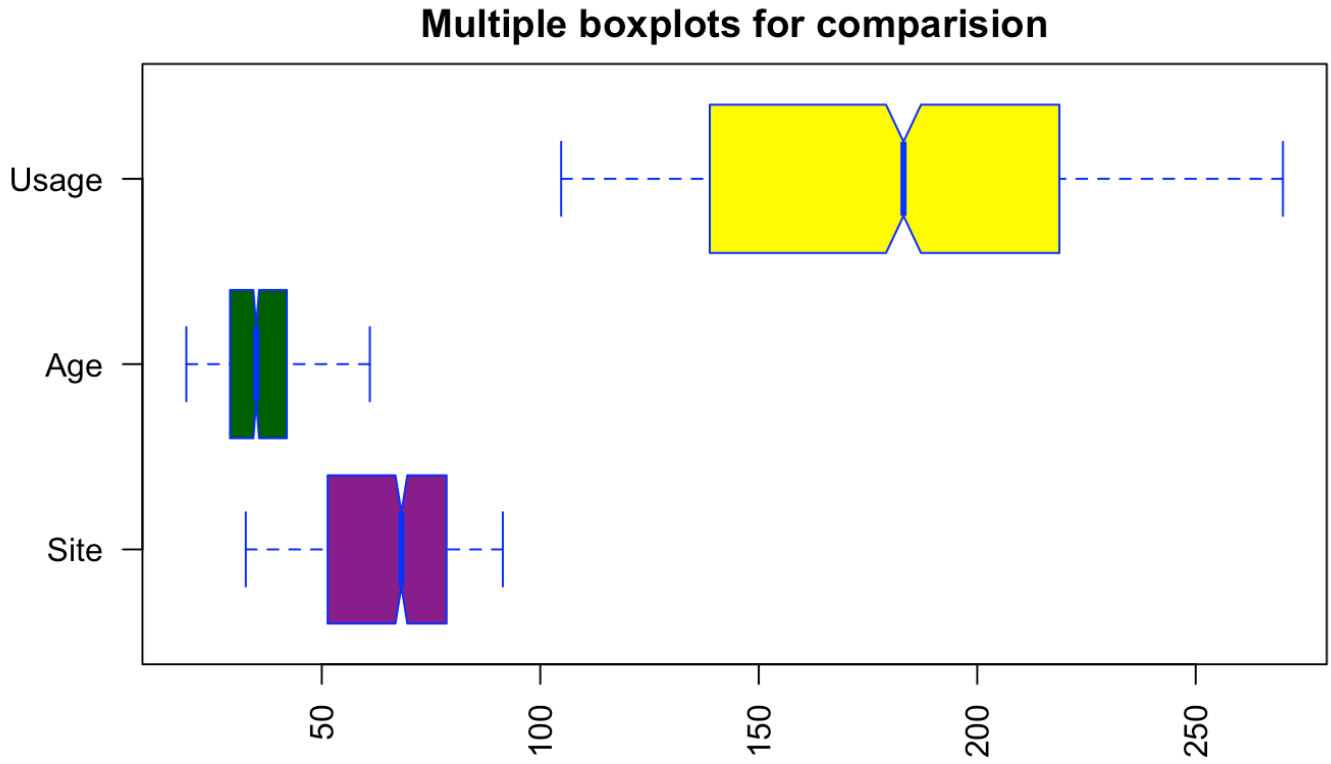
```
# a plot showing income usage in relation to gender
options(repr.plot.width = 13, repr.plot.height = 7)
ggplot(data = df, aes(x = gender, y = area_income)) +
    geom_boxplot(aes(fill = factor(clicked_on_ad))) +
    geom_dotplot(binwidth = NULL, binaxis = 'y', stackdir = 'center', dotsize = .5
, fill = 'maroon') +
    labs(title = 'Area income usage Vs Gender', y = 'Area income', x = 'Gender', f
ill = 'Clicked on ad') +
    scale_color_brewer(palette = 'cool') +
    theme(plot.title = element_text(size = 18, face = 'bold', color = 'darkmagenta
'),
              axis.title.x = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.title.y = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.text.x = element_text(size = 13),
              axis.text.y = element_text(size = 13),
              legend.title = element_text(size = 13, color = 'darkmagenta'),
              legend.text = element_text(size = 12))
```

```
Unknown palette cool
```



Hide

```
# plotting multiple boxplots
options(repr.plot.width = 13, repr.plot.height = 7)
boxplot(df$daily_time_spent_on_site, df$age, df$daily_internet_usage,
main = "Multiple boxplots for comparision",
at = c(1,2,3),
names = c("Site", "Age","Usage"),
las = 2,
col = c("darkmagenta","darkgreen","yellow"),
border = "blue",
horizontal = TRUE,
notch = TRUE
)
```



Hide

```
# The male column should be renamed to gender
colnames(df)[colnames(df) == 'male'] = 'gender'
```

Hide

```
library(tidyverse)
# Changing column names to lower case
colnames(df) = tolower(str_replace_all(colnames(df), c(' ' = '_')))

# Checking whether the column names have been renames appriopriately
print(colnames(df))
```

```
 [1] "daily_time_spent_on_site" "age"                      "area_income"
 [4] "daily_internet_usage"     "ad_topic_line"            "city"
 [7] "gender"                   "country"                  "timestamp"
[10] "clicked_on_ad"
```

Hide

```
# Checking the datatypes for each column

columns = colnames(df)
for (column in seq(length(colnames(df)))){
    print(columns[column])
    print(str(df[, column]))
    cat('\n')
}
```

```
[1] "daily_time_spent_on_site"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ daily_time_spent_on_site: num  69 80.2 69.5 74.2 68.4 ...
NULL

[1] "age"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ age: num  35 31 26 29 35 23 33 48 30 20 ...
NULL

[1] "area_income"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ area_income: num  61834 68442 59786 54806 73890 ...
NULL

[1] "daily_internet_usage"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ daily_internet_usage: num  256 194 236 246 226 ...
NULL

[1] "ad_topic_line"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ ad_topic_line: chr  "Cloned 5thgeneration orchestration" "Monitored national st
andardization" "Organic bottom-line service-desk" "Triple-buffered reciprocal time
-frame" ...
NULL

[1] "city"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ city: chr  "Wrightburgh" "West Jodi" "Davidton" "West Terrifurt" ...
NULL
```

```
[1] "gender"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ gender: num  0 1 0 1 0 1 0 1 1 1 ...
NULL

[1] "country"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ country: chr  "Tunisia" "Nauru" "San Marino" "Italy" ...
NULL

[1] "timestamp"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ timestamp: POSIXct, format: "2016-03-27 00:53:11" "2016-04-04 01:39:02" "2016-0
3-13 20:35:42" ...
NULL

[1] "clicked_on_ad"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ clicked_on_ad: num  0 0 0 0 0 0 0 1 0 0 ...
NULL
```

Hide

```
library(magrittr)
# Changing column names to their appriopriate data type
# Creating a lists of categorical and numerical columns

# List of categorical columns
cat_cols = c("ad_topic_line", "city", "gender", "country", "clicked_on_ad" )

# List of numerical columns
num_cols = c("daily_time_spent_on_site", "age", "area_income", "daily_internet_usa
ge")

# Changing columns to factors
df[,cat_cols] %<>% lapply(function(x) as.factor(as.character(x)))
```

Hide

```
# Checking whether the datatypes for each column have been changed apprippriately
columns = colnames(df)
for (column in seq(length(colnames(df)))){
    print(columns[column])
    print(str(df[, column]))
    print(nlevels(df[, column]))
    cat('\n')
}
```

```
[1] "daily_time_spent_on_site"
```

```
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ daily_time_spent_on_site: num  69 80.2 69.5 74.2 68.4 ...
NULL
[1] 0

[1] "age"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ age: num  35 31 26 29 35 23 33 48 30 20 ...
NULL
[1] 0

[1] "area_income"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ area_income: num  61834 68442 59786 54806 73890 ...
NULL
[1] 0

[1] "daily_internet_usage"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ daily_internet_usage: num  256 194 236 246 226 ...
NULL
[1] 0

[1] "ad_topic_line"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ ad_topic_line: Factor w/ 1000 levels "Adaptive 24hour Graphic Interface",..: 92
465 567 904 767 806 223 724 108 455 ...
NULL
[1] 0

[1] "city"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ city: Factor w/ 969 levels "Adamsbury","Adamside",..: 962 904 112 940 806 283 4
7 672 885 713 ...
NULL
[1] 0

[1] "gender"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ gender: Factor w/ 2 levels "0","1": 1 2 1 2 1 2 1 2 2 2 ...
NULL
[1] 0

[1] "country"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ country: Factor w/ 237 levels "Afghanistan",..: 216 148 185 104 97 159 146 13 8
3 79 ...
NULL
[1] 0
```

```
[1] "timestamp"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ timestamp: POSIXct, format: "2016-03-27 00:53:11" "2016-04-04 01:39:02" "2016-0
3-13 20:35:42" ...
NULL
[1] 0

[1] "clicked_on_ad"
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   1000 obs. of  1 variable:
 $ clicked_on_ad: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
NULL
[1] 0
```

Hide

```
# Frequency tables
# 0-female, 1-male
levels(df$gender) = c("Female", "Male")
table(df$gender)
```

```
Female   Male
   519    481
```

The gender column is seen to be almost evenly distributed with Females being slightly higher than the males.

Hide

```
#0-yes,1-no
levels(df$clicked_on_ad) = c("Yes", "No")
table(df$clicked_on_ad)
```

```
Yes  No
500 500
```

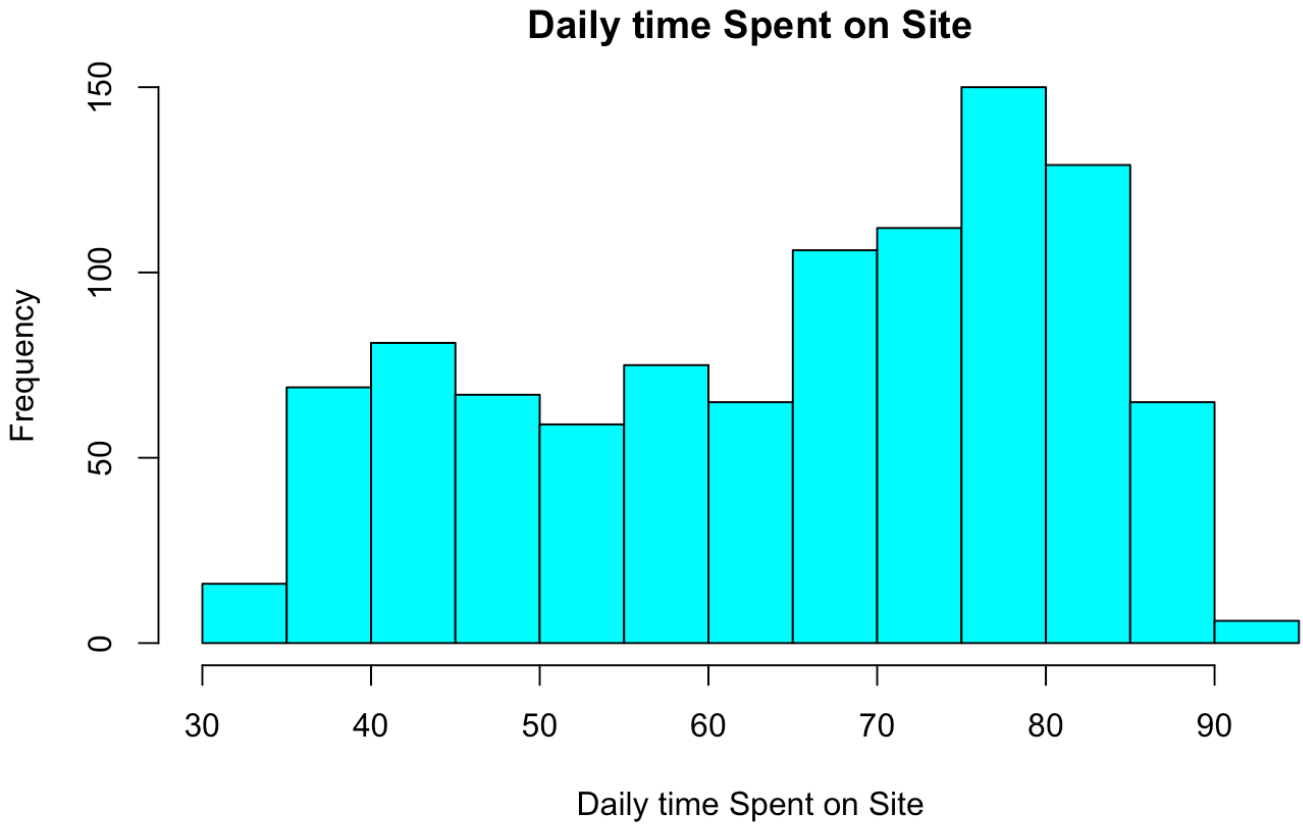
Exploratory Data Analysis

#This is where we explore the data so as to: *maximize insights on the data set* uncover underlying structure *extract important variables* detect outliers and anomalies *test underlying assumptions* develop models with great explanatory predictive power *determine optimal factor settings

Here we will perform : univariate analysis bivariate analysis multivariate analysis

Hide

```
# Daily time spent on site distribution
#
x = hist(df$daily_time_spent_on_site,
         main = "Daily time Spent on Site",
         xlab = "Daily time Spent on Site",
         col = "cyan",
)
```



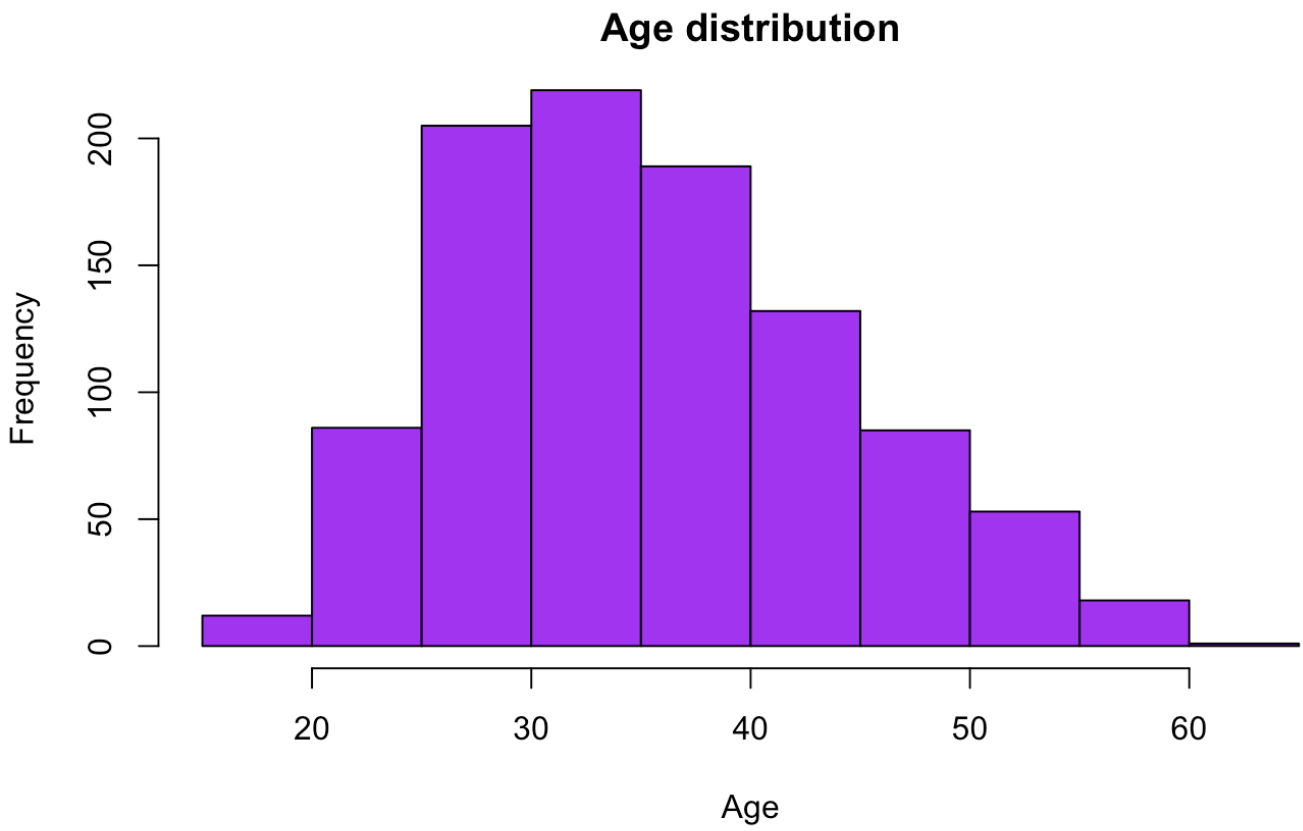
Hide

```
summary(df$daily_time_spent_on_site)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  32.60   51.36   68.22   65.00   78.55   91.43 
```

Hide

```
# Age distribution.
#
y = hist(df$age,
         main = "Age distribution",
         xlab = "Age",
         col = "purple",
)
```



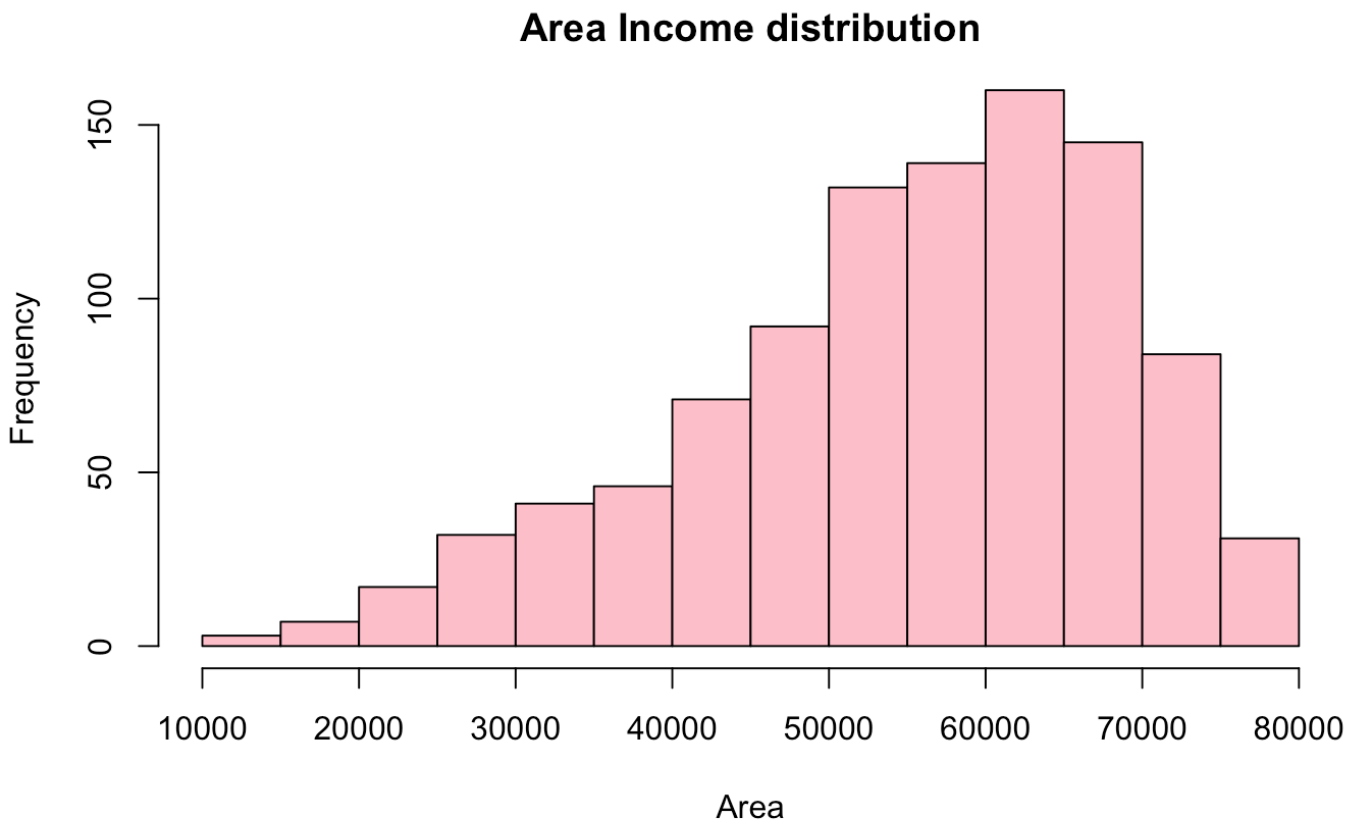
Hide

```
summary(df$age)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  19.00   29.00   35.00   36.01   42.00   61.00 
```

Hide

```
z = hist(df$area_income,
         main = "Area Income distribution",
         xlab = "Area",
         col = "pink",
)
```



Hide

```
summary(df$area_income)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  13996   47032   57012   55000   65471   79485 
```

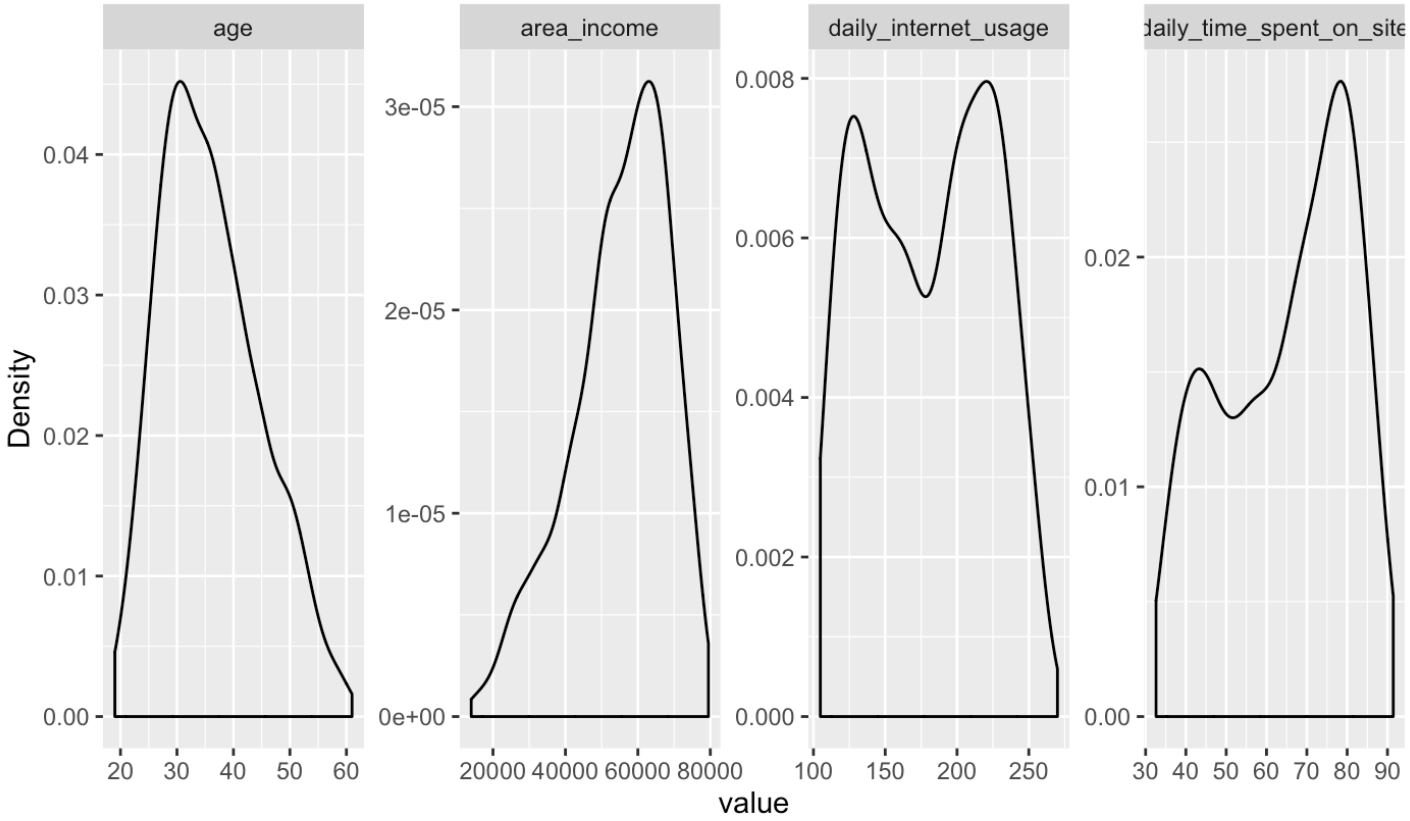
Hide

```
#density plots fr univariate analysis
library(DataExplorer)
```

```
Registered S3 method overwritten by 'data.table':
  method           from
  print.data.table
Registered S3 method overwritten by 'htmlwidgets':
  method           from
  print.htmlwidget tools:rstudio
```

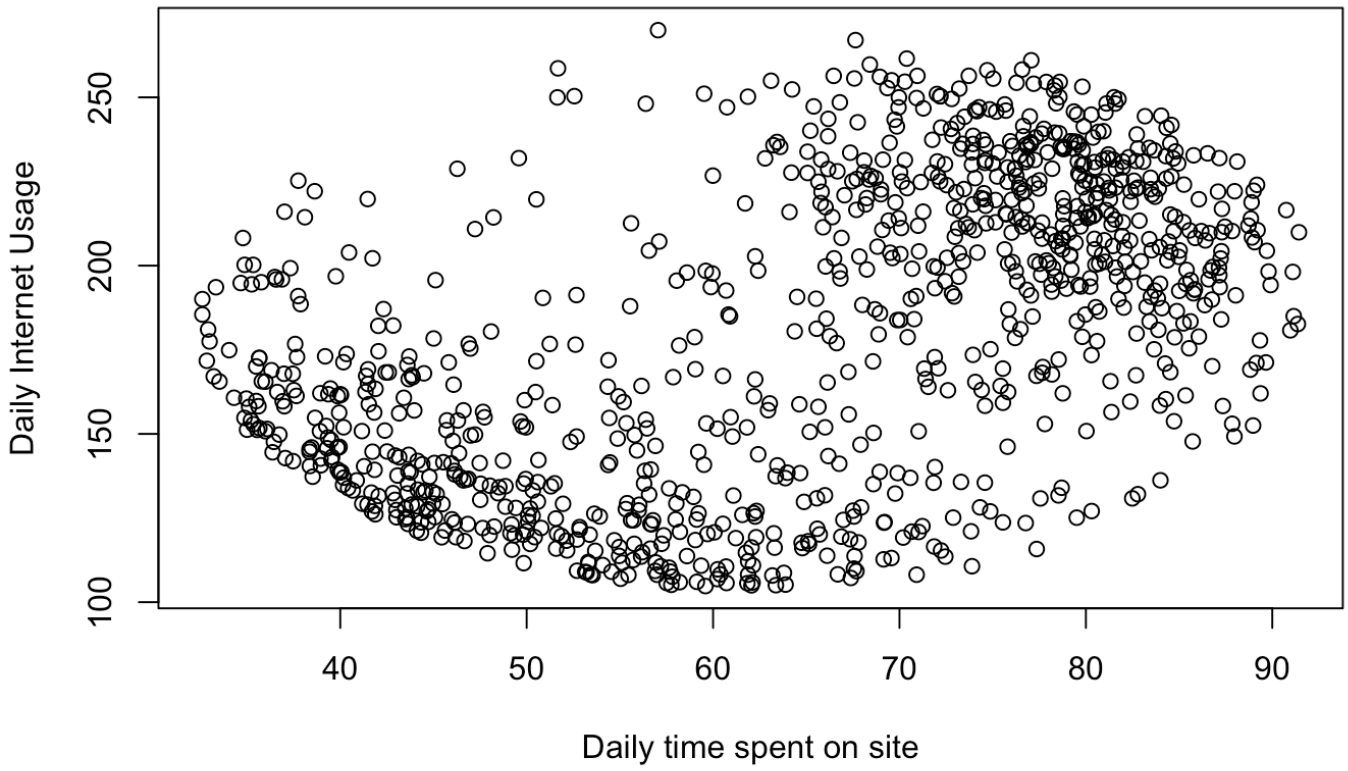
Hide

```
plot_density(df)
```



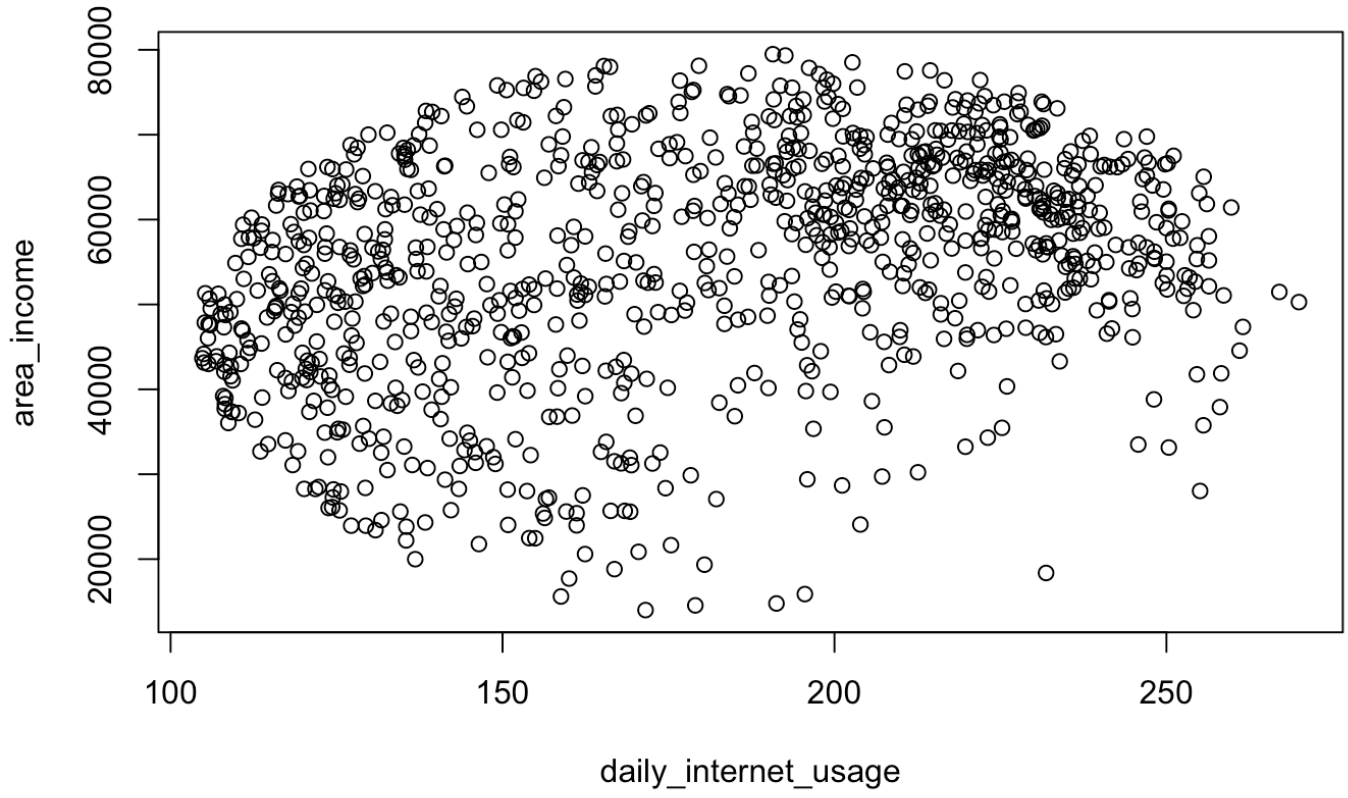
Hide

```
# bivariate plots
library(DataExplorer)
timespent <- df$daily_time_spent_on_site
internetusage<- df$daily_internet_usage
plot(timespent, internetusage, xlab="Daily time spent on site", ylab="Daily Intern
et Usage")
```



Hide

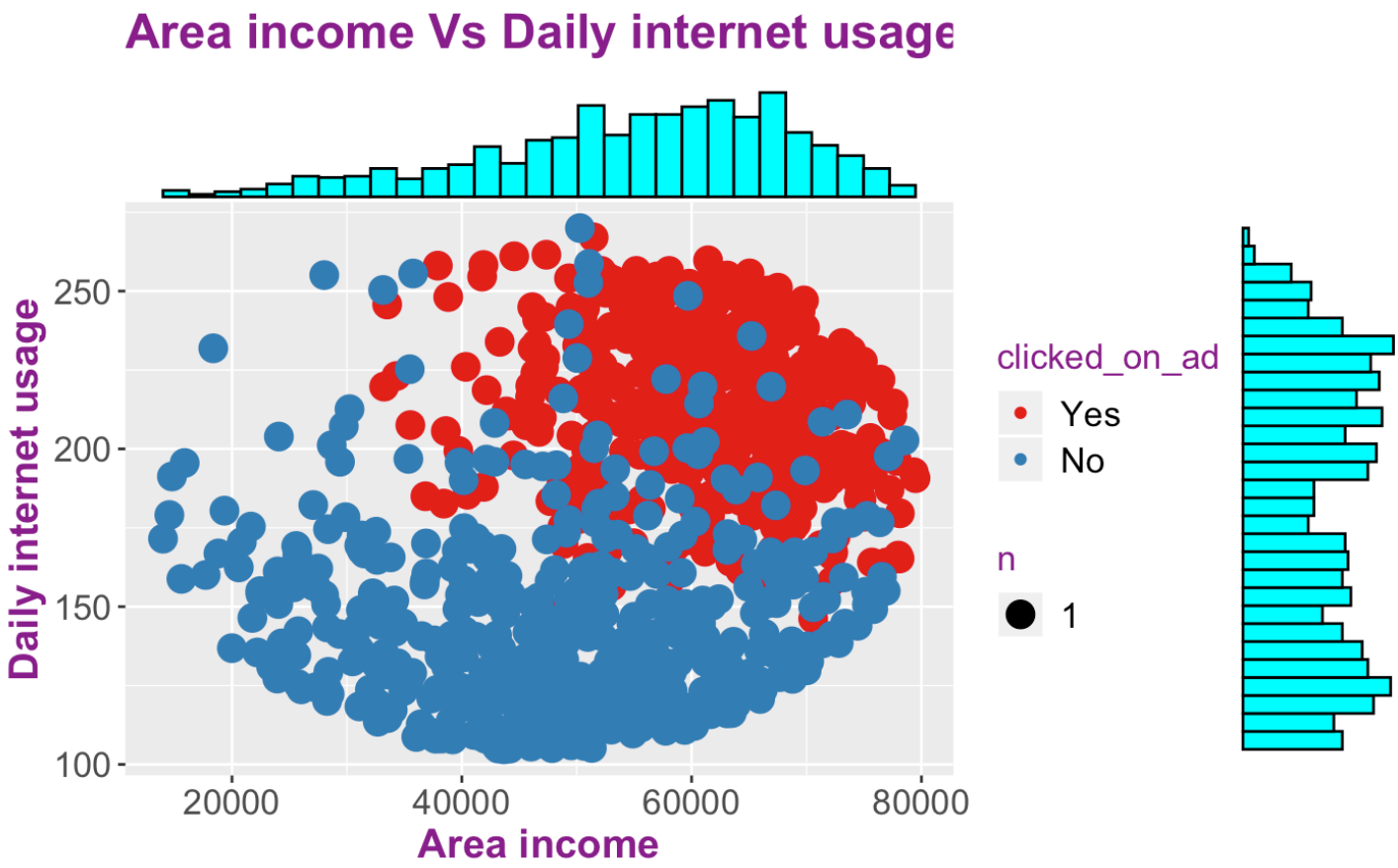
```
plot(area_income ~ daily_internet_usage, data = df)
```



Hide

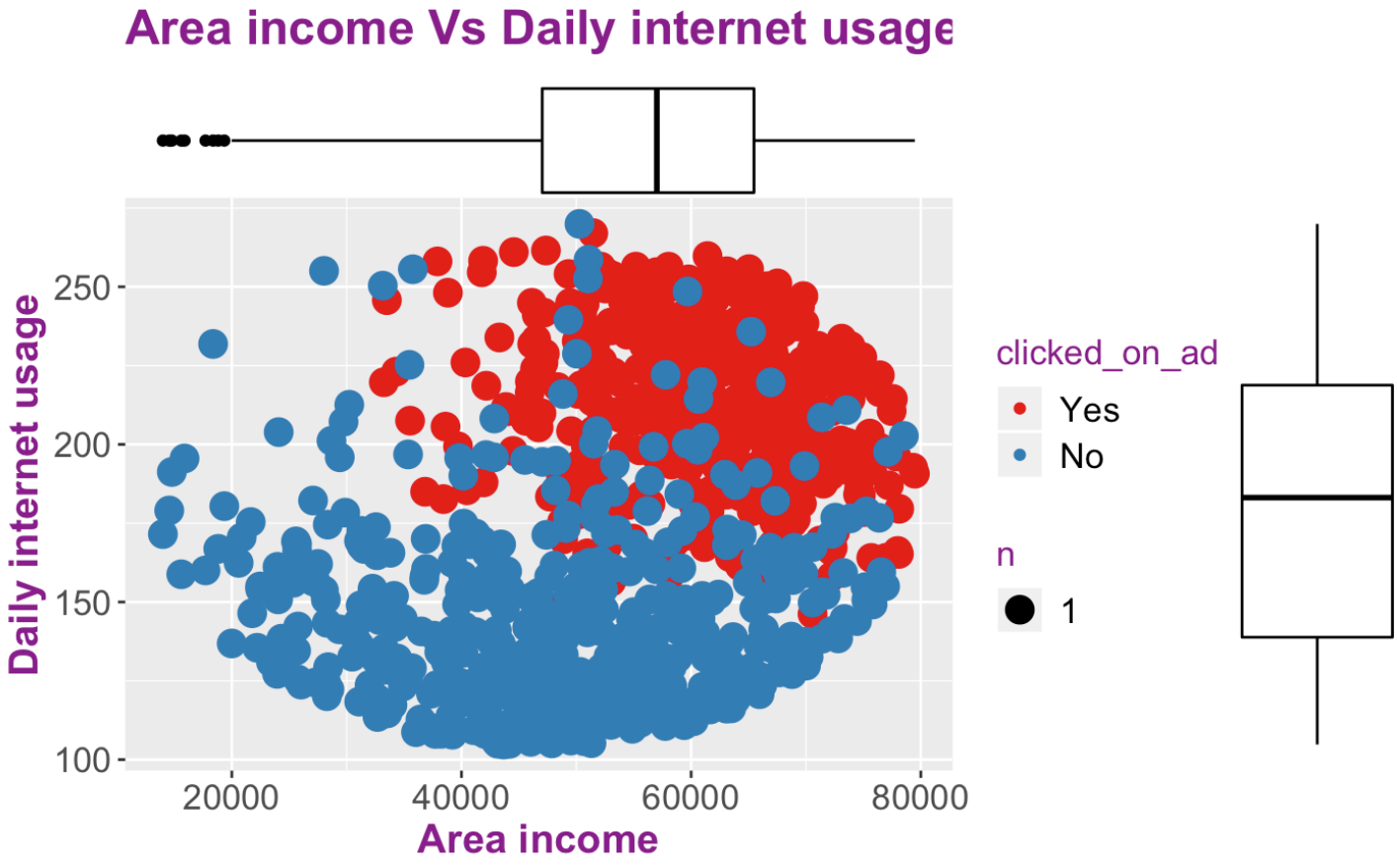
```
library (ggExtra)
options(repr.plot.width = 13, repr.plot.height = 7)
g = ggplot(data =df, aes(x =area_income, y = daily_internet_usage, col= clicked_on
_ad)) +
    geom_count() +
    labs(title = 'Area income Vs Daily internet usage', y = 'Daily internet usage'
, x = 'Area income', fill = 'Clicked on ad') +
    scale_color_brewer(palette = 'Set1') +
    theme(plot.title = element_text(size = 18, face = 'bold', color = 'darkmagenta
'),
             axis.title.x = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
             axis.title.y = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
             axis.text.x = element_text(size = 13),
             axis.text.y = element_text(size = 13),
             legend.title = element_text(size = 13, color = 'darkmagenta'),
             legend.text = element_text(size = 13))

ggMarginal(g, type = "histogram", fill="cyan")
```



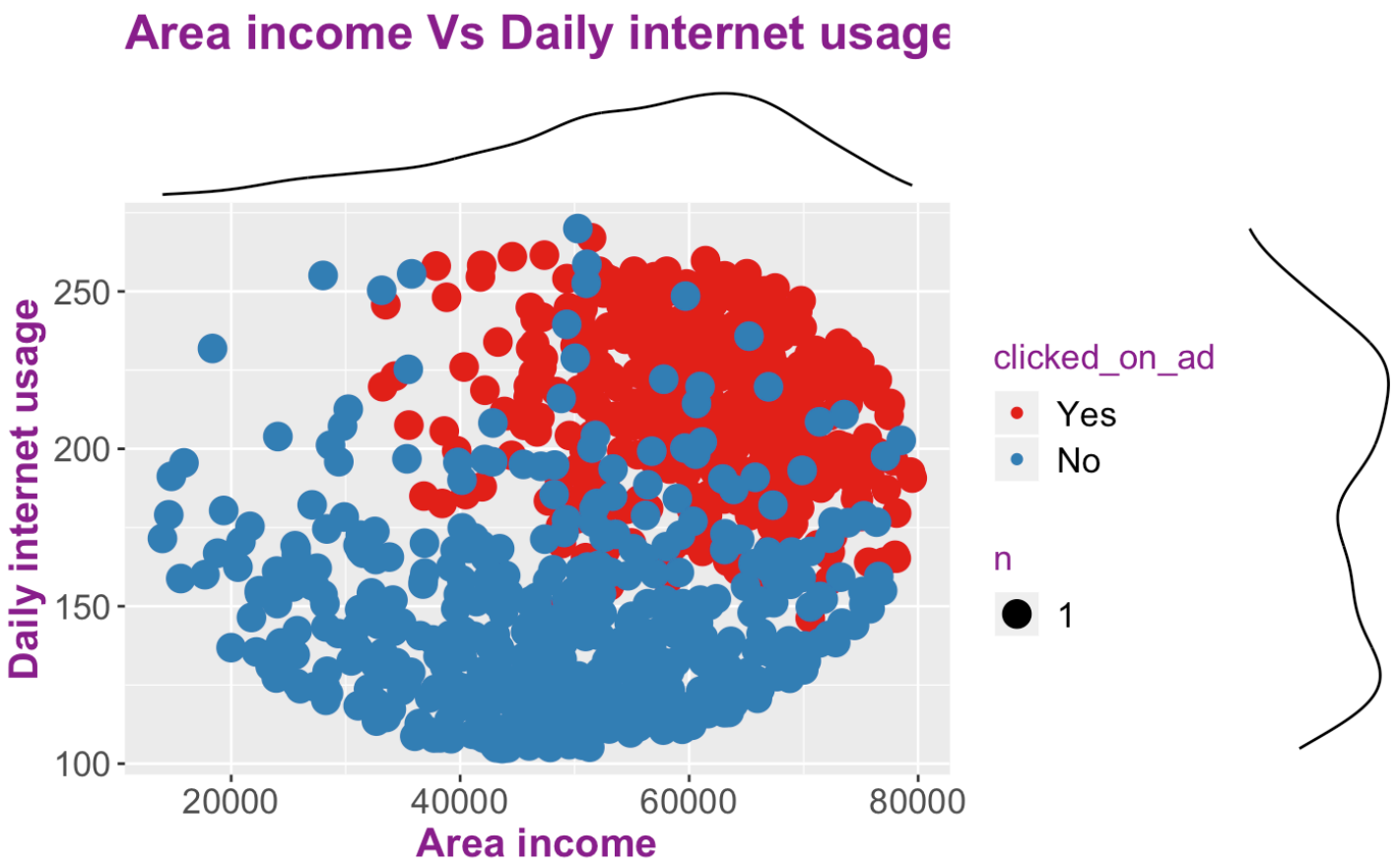
Hide

```
ggMarginal(g, type = "boxplot", fill="transparent")
```



Hide

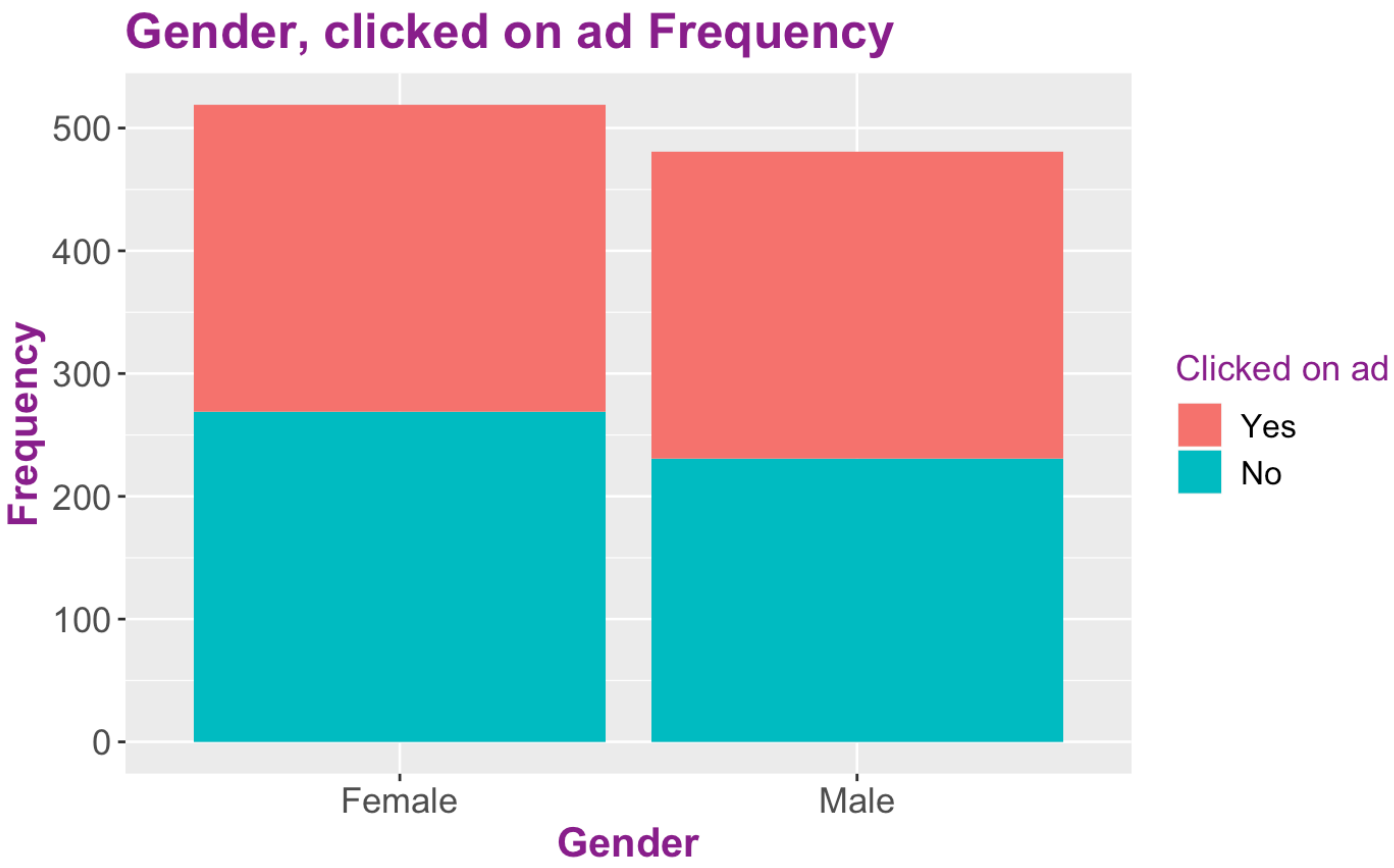
```
ggMarginal(g, type = "density", fill="transparent")
```



Hide

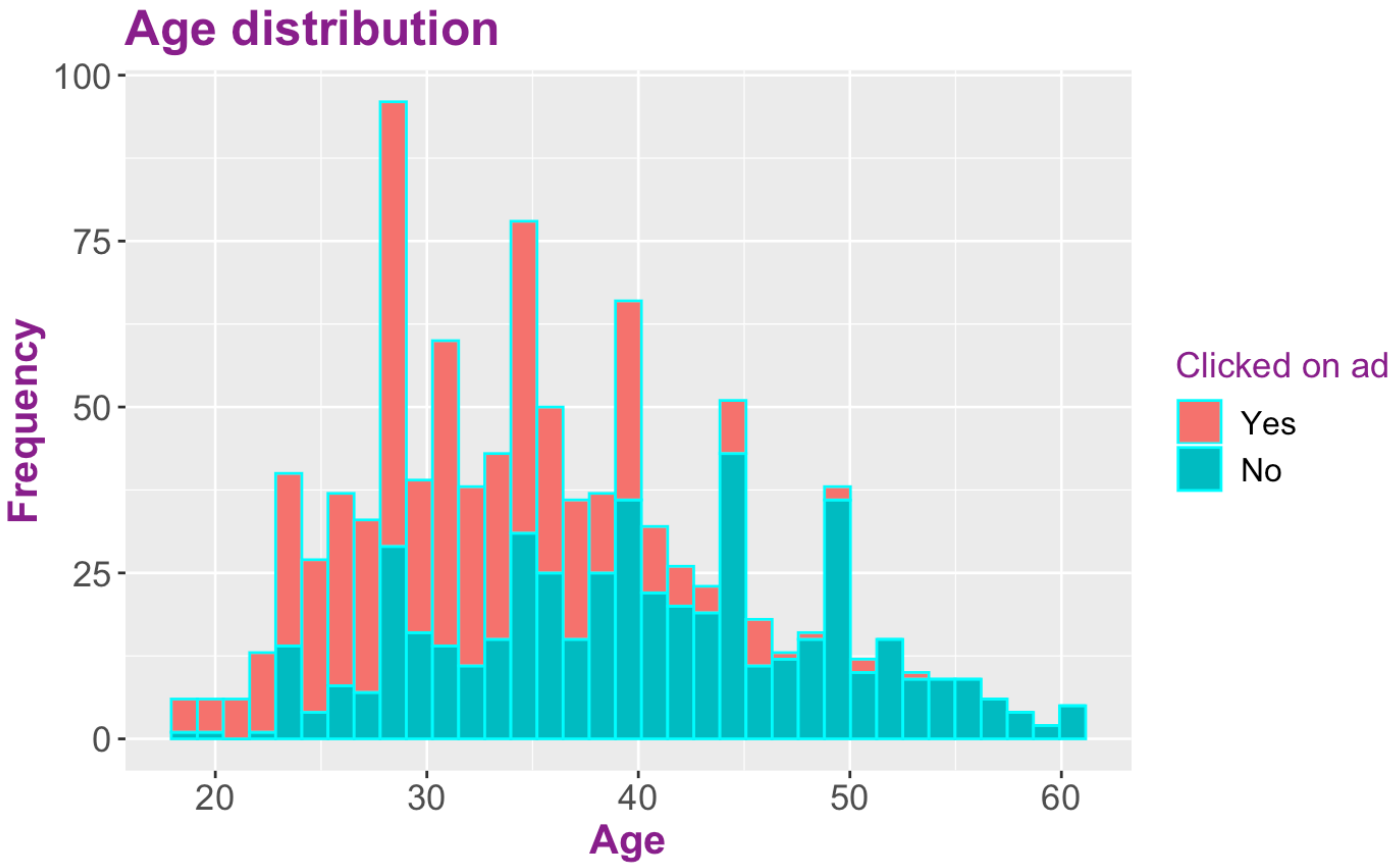
```
# A frequency plot
options(repr.plot.width = 13, repr.plot.height = 7)
ggplot(data = df, aes(x = gender))+
    geom_bar(aes(fill = clicked_on_ad))+
    labs(title = 'Gender, clicked on ad Frequency', y = 'Frequency', x = 'Gender',
fill = 'Clicked on ad') +
    scale_color_brewer(palette = 'cool') +
    theme(plot.title = element_text(size = 18, face = 'bold', color = 'darkmagenta
'),
              axis.title.x = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.title.y = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.text.x = element_text(size = 13),
              axis.text.y = element_text(size = 13),
              legend.title = element_text(size = 13, color = 'darkmagenta'),
              legend.text = element_text(size = 12))
```

```
Unknown palette cool
```



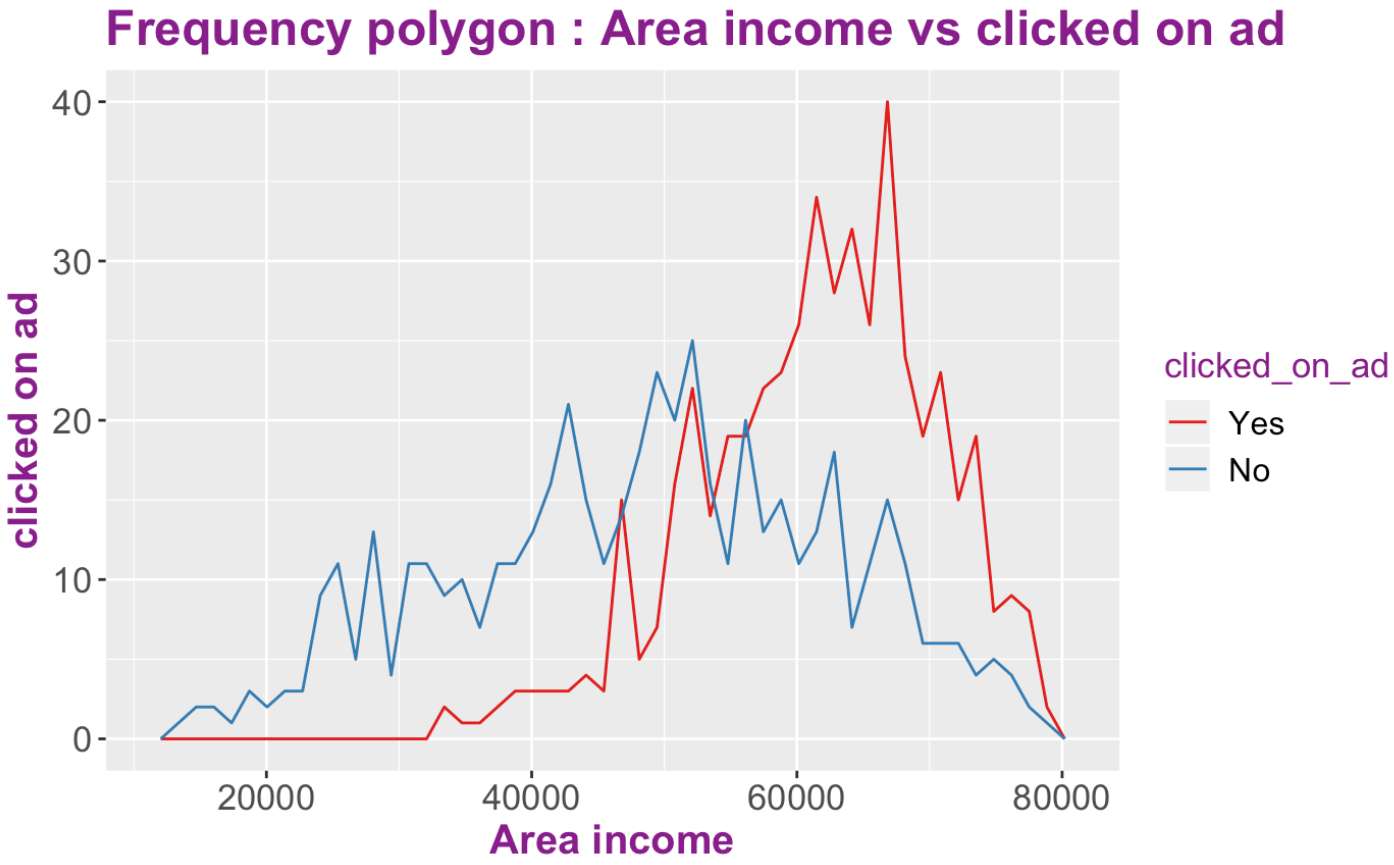
Hide

```
# Plotting a a pair of histograms
options(repr.plot.width = 13, repr.plot.height = 7)
ggplot(data = df, aes(x = age, fill = clicked_on_ad))+
    geom_histogram(bins = 35, color = 'cyan') +
    labs(title = 'Age distribution', x = 'Age', y = 'Frequency', fill = 'Clicked o
n ad') +
        scale_color_brewer(palette = 'Set1') +
        theme(plot.title = element_text(size = 18, face = 'bold', color = 'darkmag
enta'),
              axis.title.x = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.title.y = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.text.x = element_text(size = 13, angle = 0),
              axis.text.y = element_text(size = 13),
              legend.title = element_text(size = 13, color = 'darkmagenta'),
              legend.text = element_text(size = 12))
```



Hide

```
# Frequency polygon
options(repr.plot.width = 13, repr.plot.height = 7)
ggplot(data = df, aes(x = area_income, col = clicked_on_ad))+
    geom_freqpoly(bins = 50)+
    labs(title = 'Frequency polygon : Area income vs clicked on ad', x = 'Area inc
ome', y = 'clicked on ad', fill = 'Clicked on ad') +
        scale_color_brewer(palette = 'Set1') +
        theme(plot.title = element_text(size = 18, face = 'bold', color = 'darkmag
enta'),
              axis.title.x = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.title.y = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.text.x = element_text(size = 13),
              axis.text.y = element_text(size = 13),
              legend.title = element_text(size = 13, color = 'darkmagenta'),
              legend.text = element_text(size = 12))
```



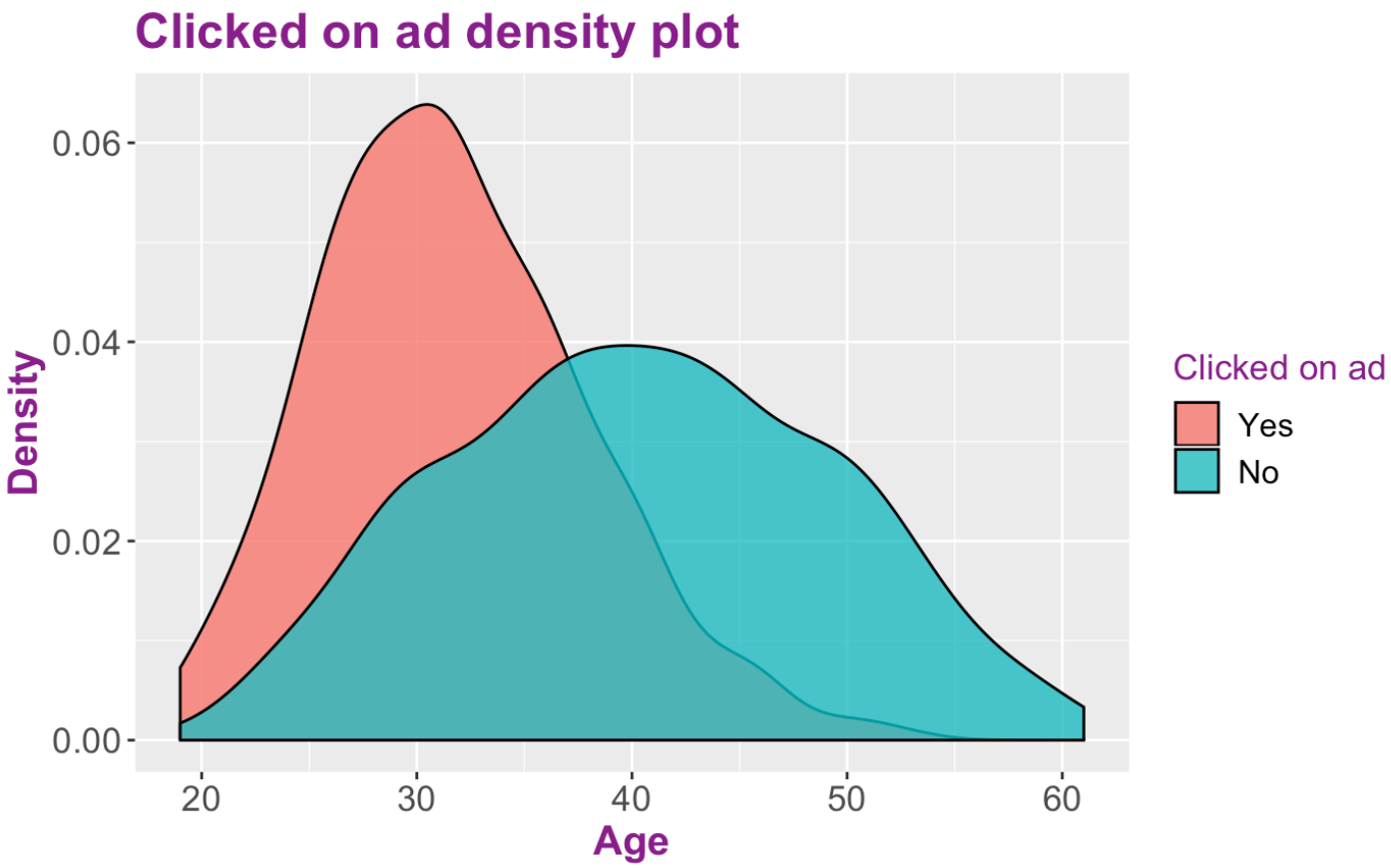
Hide

```
# Plotting density plot
options(repr.plot.width = 13, repr.plot.height = 7)
p1 = ggplot(data = df, aes(age)) +
        geom_density(aes(fill=factor(clicked_on_ad)), alpha = 0.8) +
        labs(title = 'Clicked on ad density plot', x = 'Age', y = 'Density', fill
= 'Clicked on ad') +
        scale_color_brewer(palette = 'cool') +
        theme(plot.title = element_text(size = 18, face = 'bold', color = 'darkmag
enta'),
              axis.title.x = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.title.y = element_text(size = 15, face = 'bold', color = 'darkma
genta'),
              axis.text.x = element_text(size = 13, angle = 0),
              axis.text.y = element_text(size = 13),
              legend.title = element_text(size = 13, color = 'darkmagenta'),
              legend.text = element_text(size = 12))
```

```
Unknown palette cool
```

Hide

```
plot(p1)
```

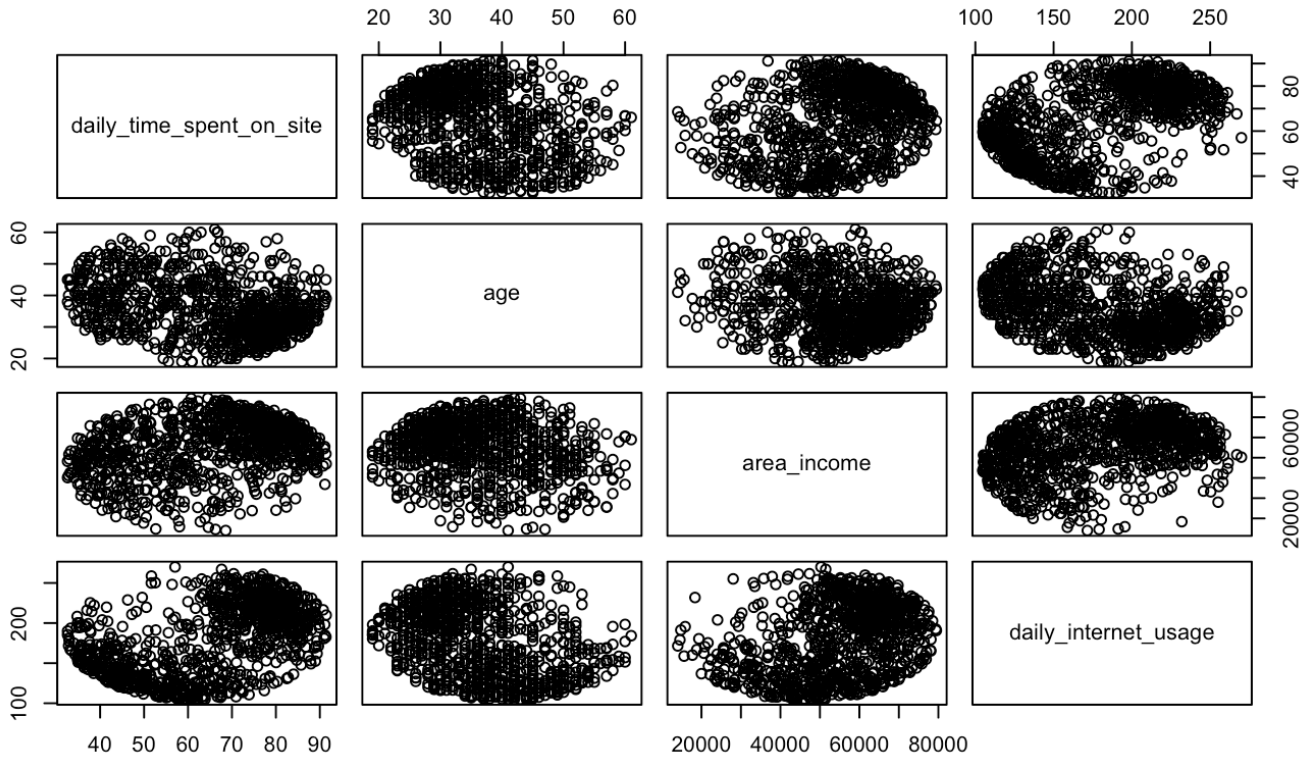


Hide

```
# multivariate analysis
# a correlation matrix showing the correlation between each variable
library(ggcorrplot)
corr = round(cor(select_if(df, is.numeric)), 2)
ggcorrplot(corr, hc.order = T, ggtheme = ggplot2::theme_grey,
   colors = c("cyan", "peachpuff4", "pink"), lab = T)
```

Hide

```
# Pairplot
pairs(df[,c(1,2,3,4)])
```


daily_time_spent_on_site
age
area_income
daily_internet_usage
20 30 40 50 60
100 150 200 250
40 60 80
20 40 60
20000 60000
100 200
40 50 60 70 80 90
20000 40000 60000 80000