Independent Project

```
install.packages('cowplot')
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages('parallel')
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)

## Warning: package 'parallel' is not available (for R version 3.6.3)

## Warning: package 'parallel' is a base package, and should not be updated
```

```
install.packages('foreach')
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages('doParallel')
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages('e1071', dependencies=TRUE)
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```


tinytex::install_tinytex()
```

Ill import the necessary libraries.

```
## tlmgr option sys_bin ~/bin
```

```
library(doParallel)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
library(parallel)
library(ggplot2)
library(cowplot)
library(magrittr)
library(caret)
```

```
## Loading required package: lattice
```

```
library(ggcorrplot)
library(ggExtra)
theme_set(theme_classic())
options(warn = -1)
```

```
greg <- read.csv('http://bit.ly/IPAdvertisingData')
head(greg)
```

Ill first import the dataset and display the head of the dataset

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35    61833.90               256.09
## 2                    80.23  31    68441.85               193.77
## 3                    69.47  26    59785.94               236.50
## 4                    74.15  29    54806.18               245.89
## 5                    68.37  35    73889.99               225.58
## 6                    59.99  23    59761.56               226.74
##                          Ad.Topic.Line           City Male    Country
## 1    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunisia
## 2    Monitored national standardization      West Jodi    1      Nauru
## 3      Organic bottom-line service-desk       Davidton    0 San Marino
## 4 Triple-buffered reciprocal time-frame West Terrifurt    1      Italy
## 5         Robust logistical utilization   South Manuel    0    Iceland
## 6       Sharable client-driven software      Jamieberg    1     Norway
##             Timestamp Clicked.on.Ad
## 1 2016-03-27 00:53:11             0
## 2 2016-04-04 01:39:02             0
## 3 2016-03-13 20:35:42             0
## 4 2016-01-10 02:31:19             0
## 5 2016-06-03 03:36:18             0
## 6 2016-05-19 14:30:17             0
```

```
  tail(greg)
```

Ill then display the tail of the dataset

```
##      Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 995                     43.70  28    63126.96               173.01
## 996                     72.97  30    71384.57               208.58
## 997                     51.30  45    67782.17               134.42
## 998                     51.63  51    42415.72               120.37
## 999                     55.55  19    41920.79               187.95
## 1000                    45.01  26    29875.80               178.35
##                            Ad.Topic.Line          City Male
## 995         Front-line bifurcated ability  Nicholasland    0
## 996         Fundamental modular algorithm     Duffystad    1
## 997       Grass-roots cohesive monitoring   New Darlene    1
## 998          Expanded intangible solution South Jessica    1
## 999  Proactive bandwidth-monitored policy   West Steven    0
## 1000      Virtual 5thgeneration emulation   Ronniemouth    0
##                      Country           Timestamp Clicked.on.Ad
## 995                  Mayotte 2016-04-04 03:57:48             1
## 996                  Lebanon 2016-02-11 21:49:00             1
## 997  Bosnia and Herzegovina 2016-04-22 02:07:01             1
## 998                 Mongolia 2016-02-01 17:24:57             1
## 999                Guatemala 2016-03-24 02:35:54             0
## 1000                  Brazil 2016-06-03 21:43:21             1
```

```
rownames(greg, do.NULL = TRUE, prefix = "row")
```

I'll then check the rows of the dataset

```
##    [1] "1"    "2"    "3"    "4"    "5"    "6"    "7"    "8"    "9"    "10"
##   [11] "11"   "12"   "13"   "14"   "15"   "16"   "17"   "18"   "19"   "20"
##   [21] "21"   "22"   "23"   "24"   "25"   "26"   "27"   "28"   "29"   "30"
##   [31] "31"   "32"   "33"   "34"   "35"   "36"   "37"   "38"   "39"   "40"
##   [41] "41"   "42"   "43"   "44"   "45"   "46"   "47"   "48"   "49"   "50"
##   [51] "51"   "52"   "53"   "54"   "55"   "56"   "57"   "58"   "59"   "60"
##   [61] "61"   "62"   "63"   "64"   "65"   "66"   "67"   "68"   "69"   "70"
##   [71] "71"   "72"   "73"   "74"   "75"   "76"   "77"   "78"   "79"   "80"
##   [81] "81"   "82"   "83"   "84"   "85"   "86"   "87"   "88"   "89"   "90"
##   [91] "91"   "92"   "93"   "94"   "95"   "96"   "97"   "98"   "99"   "100"
##  [101] "101"  "102"  "103"  "104"  "105"  "106"  "107"  "108"  "109"  "110"
##  [111] "111"  "112"  "113"  "114"  "115"  "116"  "117"  "118"  "119"  "120"
##  [121] "121"  "122"  "123"  "124"  "125"  "126"  "127"  "128"  "129"  "130"
##  [131] "131"  "132"  "133"  "134"  "135"  "136"  "137"  "138"  "139"  "140"
##  [141] "141"  "142"  "143"  "144"  "145"  "146"  "147"  "148"  "149"  "150"
##  [151] "151"  "152"  "153"  "154"  "155"  "156"  "157"  "158"  "159"  "160"
##  [161] "161"  "162"  "163"  "164"  "165"  "166"  "167"  "168"  "169"  "170"
##  [171] "171"  "172"  "173"  "174"  "175"  "176"  "177"  "178"  "179"  "180"
##  [181] "181"  "182"  "183"  "184"  "185"  "186"  "187"  "188"  "189"  "190"
##  [191] "191"  "192"  "193"  "194"  "195"  "196"  "197"  "198"  "199"  "200"
##  [201] "201"  "202"  "203"  "204"  "205"  "206"  "207"  "208"  "209"  "210"
##  [211] "211"  "212"  "213"  "214"  "215"  "216"  "217"  "218"  "219"  "220"
##  [221] "221"  "222"  "223"  "224"  "225"  "226"  "227"  "228"  "229"  "230"
##  [231] "231"  "232"  "233"  "234"  "235"  "236"  "237"  "238"  "239"  "240"
##  [241] "241"  "242"  "243"  "244"  "245"  "246"  "247"  "248"  "249"  "250"
```

```
##  [251] "251"  "252"  "253"  "254"  "255"  "256"  "257"  "258"  "259"  "260"
##  [261] "261"  "262"  "263"  "264"  "265"  "266"  "267"  "268"  "269"  "270"
##  [271] "271"  "272"  "273"  "274"  "275"  "276"  "277"  "278"  "279"  "280"
##  [281] "281"  "282"  "283"  "284"  "285"  "286"  "287"  "288"  "289"  "290"
##  [291] "291"  "292"  "293"  "294"  "295"  "296"  "297"  "298"  "299"  "300"
##  [301] "301"  "302"  "303"  "304"  "305"  "306"  "307"  "308"  "309"  "310"
##  [311] "311"  "312"  "313"  "314"  "315"  "316"  "317"  "318"  "319"  "320"
##  [321] "321"  "322"  "323"  "324"  "325"  "326"  "327"  "328"  "329"  "330"
##  [331] "331"  "332"  "333"  "334"  "335"  "336"  "337"  "338"  "339"  "340"
##  [341] "341"  "342"  "343"  "344"  "345"  "346"  "347"  "348"  "349"  "350"
##  [351] "351"  "352"  "353"  "354"  "355"  "356"  "357"  "358"  "359"  "360"
##  [361] "361"  "362"  "363"  "364"  "365"  "366"  "367"  "368"  "369"  "370"
##  [371] "371"  "372"  "373"  "374"  "375"  "376"  "377"  "378"  "379"  "380"
##  [381] "381"  "382"  "383"  "384"  "385"  "386"  "387"  "388"  "389"  "390"
##  [391] "391"  "392"  "393"  "394"  "395"  "396"  "397"  "398"  "399"  "400"
##  [401] "401"  "402"  "403"  "404"  "405"  "406"  "407"  "408"  "409"  "410"
##  [411] "411"  "412"  "413"  "414"  "415"  "416"  "417"  "418"  "419"  "420"
##  [421] "421"  "422"  "423"  "424"  "425"  "426"  "427"  "428"  "429"  "430"
##  [431] "431"  "432"  "433"  "434"  "435"  "436"  "437"  "438"  "439"  "440"
##  [441] "441"  "442"  "443"  "444"  "445"  "446"  "447"  "448"  "449"  "450"
##  [451] "451"  "452"  "453"  "454"  "455"  "456"  "457"  "458"  "459"  "460"
##  [461] "461"  "462"  "463"  "464"  "465"  "466"  "467"  "468"  "469"  "470"
##  [471] "471"  "472"  "473"  "474"  "475"  "476"  "477"  "478"  "479"  "480"
##  [481] "481"  "482"  "483"  "484"  "485"  "486"  "487"  "488"  "489"  "490"
##  [491] "491"  "492"  "493"  "494"  "495"  "496"  "497"  "498"  "499"  "500"
##  [501] "501"  "502"  "503"  "504"  "505"  "506"  "507"  "508"  "509"  "510"
##  [511] "511"  "512"  "513"  "514"  "515"  "516"  "517"  "518"  "519"  "520"
##  [521] "521"  "522"  "523"  "524"  "525"  "526"  "527"  "528"  "529"  "530"
##  [531] "531"  "532"  "533"  "534"  "535"  "536"  "537"  "538"  "539"  "540"
##  [541] "541"  "542"  "543"  "544"  "545"  "546"  "547"  "548"  "549"  "550"
##  [551] "551"  "552"  "553"  "554"  "555"  "556"  "557"  "558"  "559"  "560"
##  [561] "561"  "562"  "563"  "564"  "565"  "566"  "567"  "568"  "569"  "570"
##  [571] "571"  "572"  "573"  "574"  "575"  "576"  "577"  "578"  "579"  "580"
##  [581] "581"  "582"  "583"  "584"  "585"  "586"  "587"  "588"  "589"  "590"
##  [591] "591"  "592"  "593"  "594"  "595"  "596"  "597"  "598"  "599"  "600"
##  [601] "601"  "602"  "603"  "604"  "605"  "606"  "607"  "608"  "609"  "610"
##  [611] "611"  "612"  "613"  "614"  "615"  "616"  "617"  "618"  "619"  "620"
##  [621] "621"  "622"  "623"  "624"  "625"  "626"  "627"  "628"  "629"  "630"
##  [631] "631"  "632"  "633"  "634"  "635"  "636"  "637"  "638"  "639"  "640"
##  [641] "641"  "642"  "643"  "644"  "645"  "646"  "647"  "648"  "649"  "650"
##  [651] "651"  "652"  "653"  "654"  "655"  "656"  "657"  "658"  "659"  "660"
##  [661] "661"  "662"  "663"  "664"  "665"  "666"  "667"  "668"  "669"  "670"
##  [671] "671"  "672"  "673"  "674"  "675"  "676"  "677"  "678"  "679"  "680"
##  [681] "681"  "682"  "683"  "684"  "685"  "686"  "687"  "688"  "689"  "690"
##  [691] "691"  "692"  "693"  "694"  "695"  "696"  "697"  "698"  "699"  "700"
##  [701] "701"  "702"  "703"  "704"  "705"  "706"  "707"  "708"  "709"  "710"
##  [711] "711"  "712"  "713"  "714"  "715"  "716"  "717"  "718"  "719"  "720"
##  [721] "721"  "722"  "723"  "724"  "725"  "726"  "727"  "728"  "729"  "730"
##  [731] "731"  "732"  "733"  "734"  "735"  "736"  "737"  "738"  "739"  "740"
##  [741] "741"  "742"  "743"  "744"  "745"  "746"  "747"  "748"  "749"  "750"
##  [751] "751"  "752"  "753"  "754"  "755"  "756"  "757"  "758"  "759"  "760"
##  [761] "761"  "762"  "763"  "764"  "765"  "766"  "767"  "768"  "769"  "770"
##  [771] "771"  "772"  "773"  "774"  "775"  "776"  "777"  "778"  "779"  "780"
##  [781] "781"  "782"  "783"  "784"  "785"  "786"  "787"  "788"  "789"  "790"
```

```
##  [791] "791"  "792"  "793"  "794"  "795"  "796"  "797"  "798"  "799"  "800"
##  [801] "801"  "802"  "803"  "804"  "805"  "806"  "807"  "808"  "809"  "810"
##  [811] "811"  "812"  "813"  "814"  "815"  "816"  "817"  "818"  "819"  "820"
##  [821] "821"  "822"  "823"  "824"  "825"  "826"  "827"  "828"  "829"  "830"
##  [831] "831"  "832"  "833"  "834"  "835"  "836"  "837"  "838"  "839"  "840"
##  [841] "841"  "842"  "843"  "844"  "845"  "846"  "847"  "848"  "849"  "850"
##  [851] "851"  "852"  "853"  "854"  "855"  "856"  "857"  "858"  "859"  "860"
##  [861] "861"  "862"  "863"  "864"  "865"  "866"  "867"  "868"  "869"  "870"
##  [871] "871"  "872"  "873"  "874"  "875"  "876"  "877"  "878"  "879"  "880"
##  [881] "881"  "882"  "883"  "884"  "885"  "886"  "887"  "888"  "889"  "890"
##  [891] "891"  "892"  "893"  "894"  "895"  "896"  "897"  "898"  "899"  "900"
##  [901] "901"  "902"  "903"  "904"  "905"  "906"  "907"  "908"  "909"  "910"
##  [911] "911"  "912"  "913"  "914"  "915"  "916"  "917"  "918"  "919"  "920"
##  [921] "921"  "922"  "923"  "924"  "925"  "926"  "927"  "928"  "929"  "930"
##  [931] "931"  "932"  "933"  "934"  "935"  "936"  "937"  "938"  "939"  "940"
##  [941] "941"  "942"  "943"  "944"  "945"  "946"  "947"  "948"  "949"  "950"
##  [951] "951"  "952"  "953"  "954"  "955"  "956"  "957"  "958"  "959"  "960"
##  [961] "961"  "962"  "963"  "964"  "965"  "966"  "967"  "968"  "969"  "970"
##  [971] "971"  "972"  "973"  "974"  "975"  "976"  "977"  "978"  "979"  "980"
##  [981] "981"  "982"  "983"  "984"  "985"  "986"  "987"  "988"  "989"  "990"
##  [991] "991"  "992"  "993"  "994"  "995"  "996"  "997"  "998"  "999"  "1000"
```

```
colnames(greg, do.NULL = TRUE, prefix = "col")
```

Ill then check the columns of the dataset

```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "Ad.Topic.Line"            "City"
##  [7] "Male"                     "Country"
##  [9] "Timestamp"                "Clicked.on.Ad"
```

```
sum(is.na(greg))
```

I'll then check for missing values in the dataset

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

The output shows no missing values after summation

```
sum(duplicated(greg))
```

I'll then check for duplicates

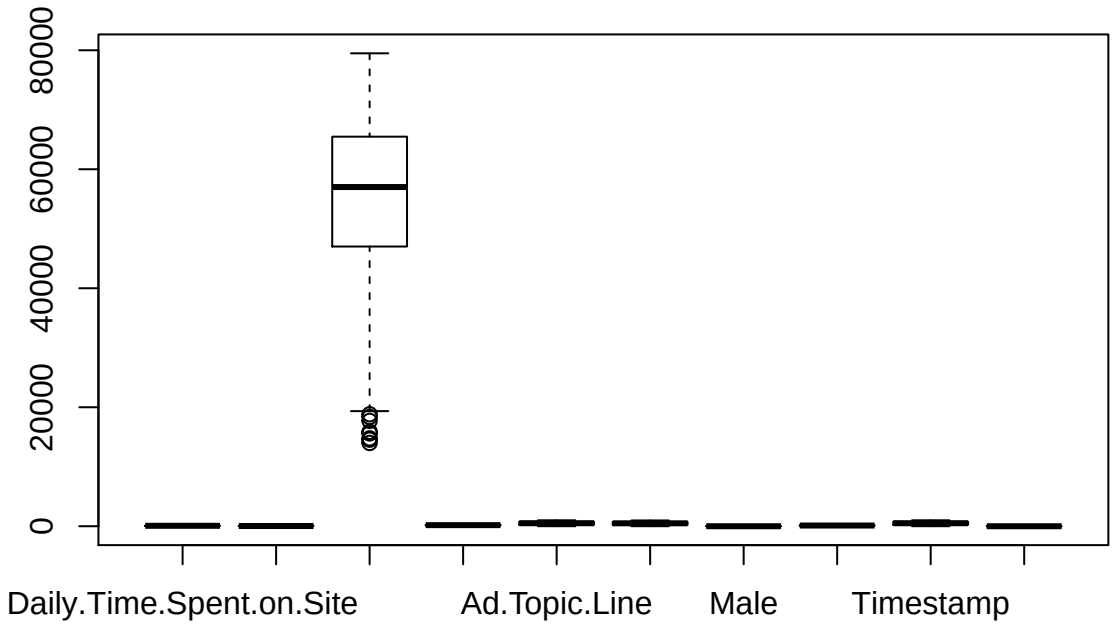
```
## [1] 0
```

The output shows no duplicates.

UNIVARIATIVE ANALYSIS

Ill check for outliers in the dataset

```
boxplot(greg)
```



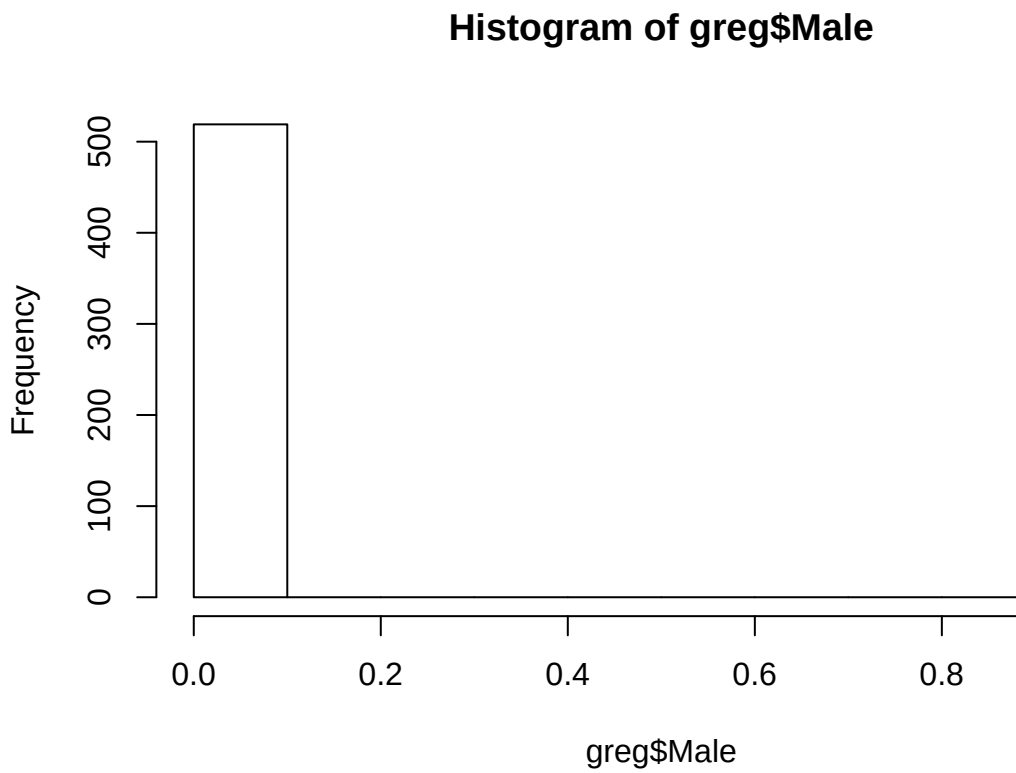
Their is presence of outliers,ill not drop them

```
hist(greg$Age)
```

Ill then show distribution of age

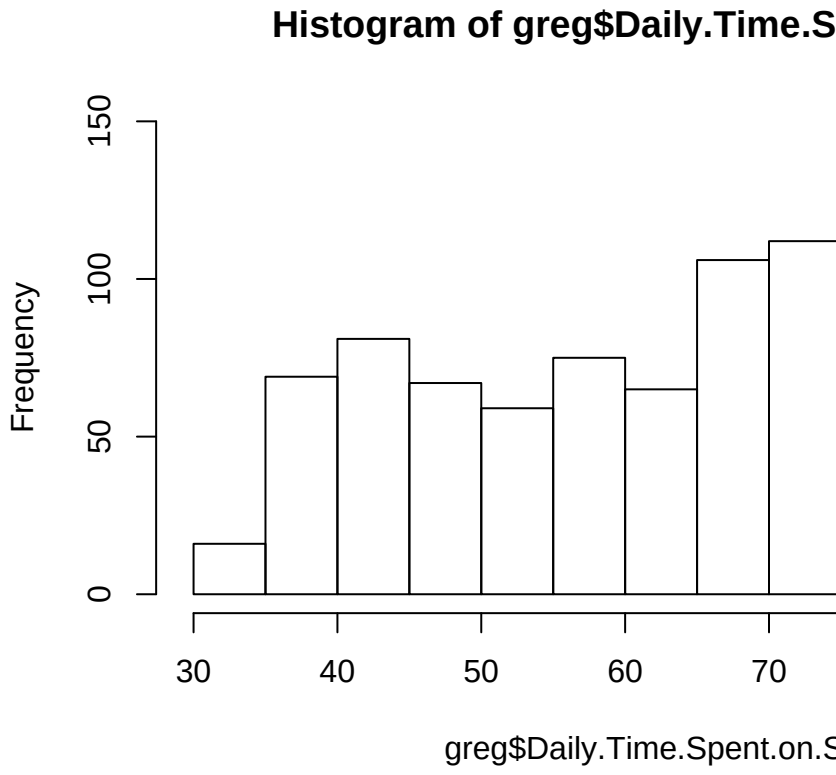
Ill then show various distribution in the

```
hist(greg$Male)
```



dataset like income,gender,etc

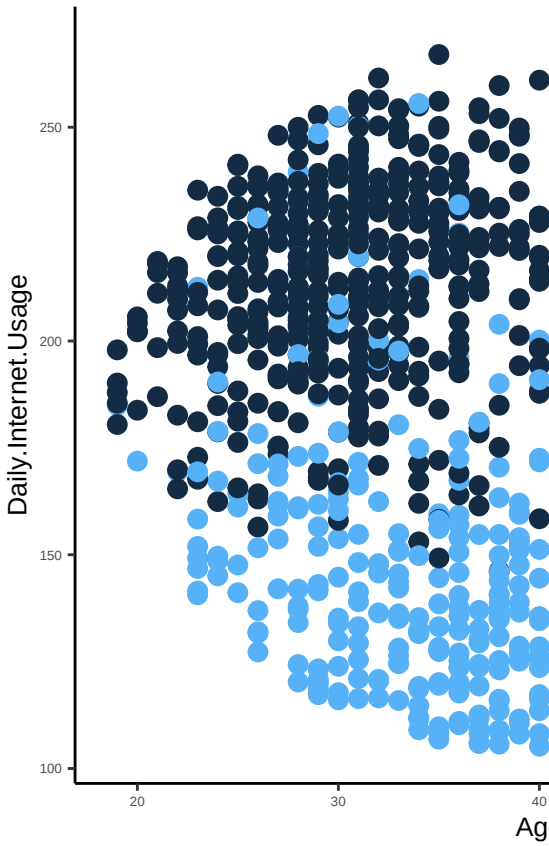
```
hist(greg$Daily.Time.Spent.on.Site)
```



Ill then distribution of income in the dataset
Bivariate Analysis

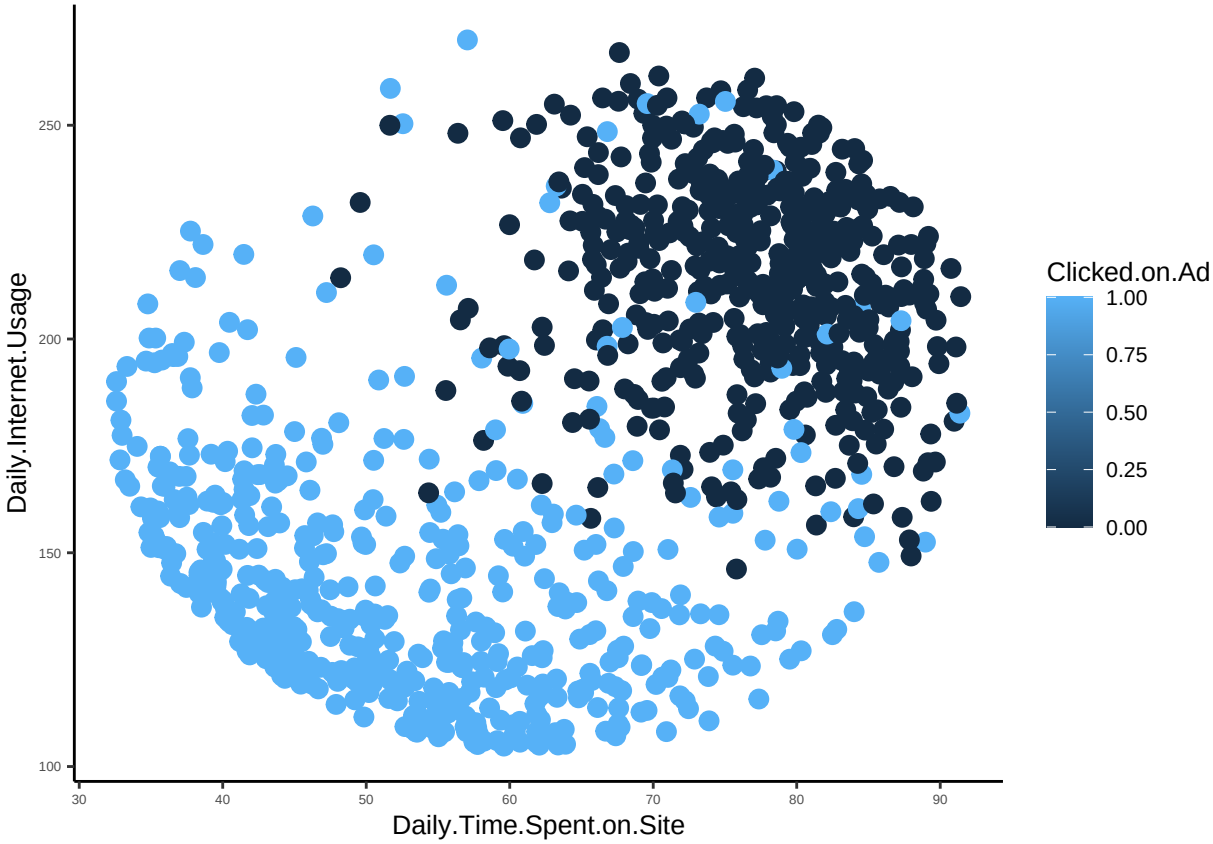
Ill then use the plot_grid() function which provides a simple

```
library(ggplot2)
plot1 <- ggplot(greg, aes(x = Age, y = Daily.Internet.Usage, color = Clicked.on.Ad)) + geom_point(size =
  theme(text = element_text(size = 10), axis.text.x = element_text(size = 5),axis.text.y = element_text
plot_grid(plot1)
```

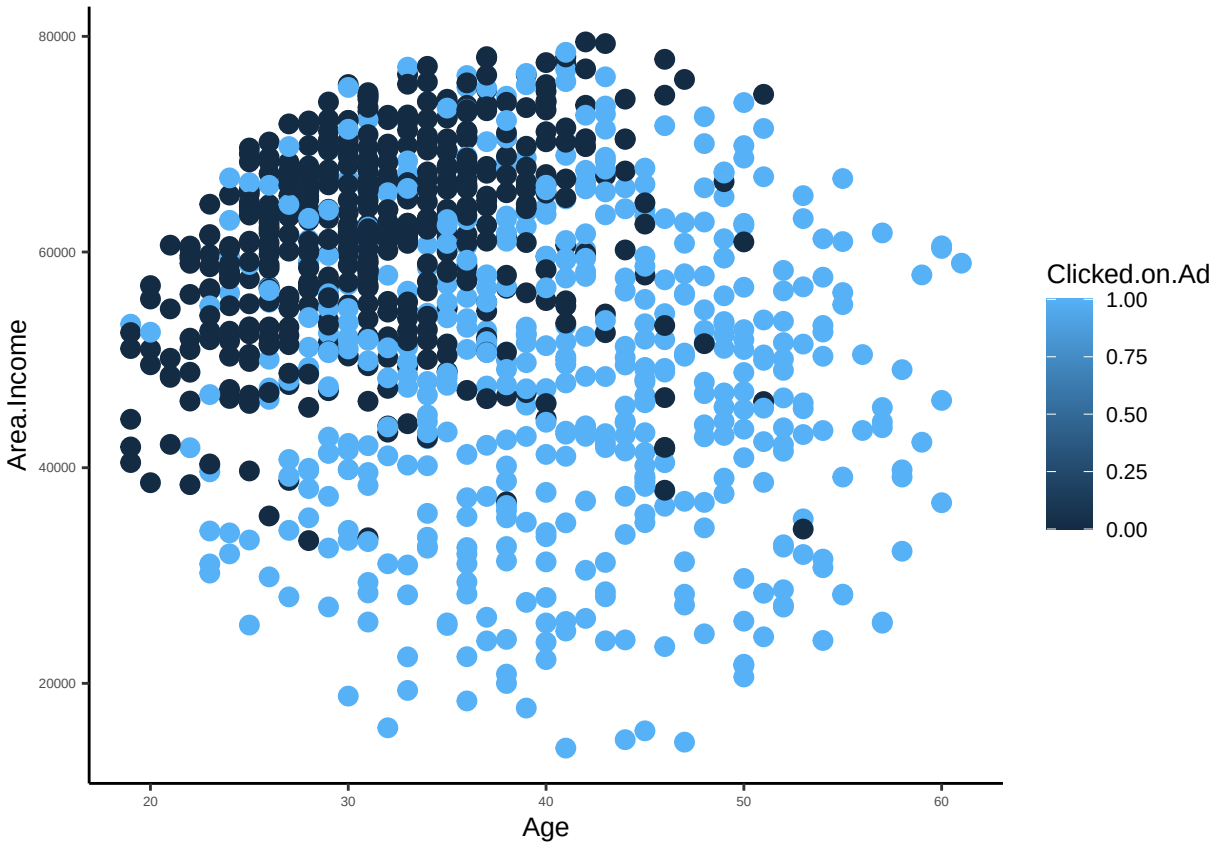


interface for arranging plots into a grid and adding labels to them.

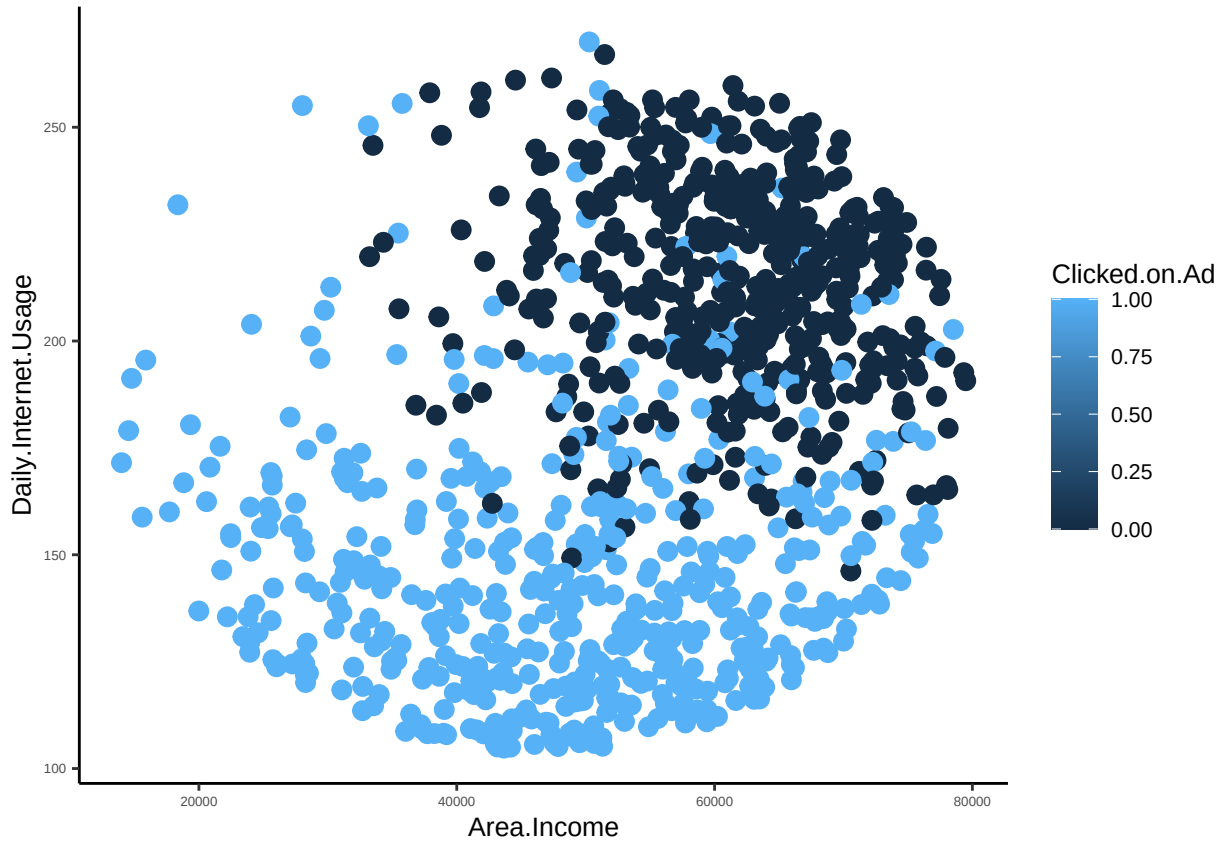
```
plot2 <- ggplot(greg, aes(x = Daily.Time.Spent.on.Site, y=Daily.Internet.Usage, color=Clicked.on.Ad)) +
  theme(text = element_text(size=10) ,axis.text.x = element_text(size = 5),axis.text.y = element_text(s
plot_grid(plot2)
```




```
plot3 <- ggplot(greg, aes(x = Age, y = Area.Income, color=Clicked.on.Ad)) + geom_point(size=3)+
  theme(text = element_text(size=10) ,axis.text.x = element_text(size = 5),axis.text.y = element_text(s
plot_grid(plot3)
```




```
plot4 <- ggplot(greg, aes(x = Area.Income, y = Daily.Internet.Usage, color = Clicked.on.Ad)) + geom_poi
  theme(text = element_text(size=10) ,axis.text.x = element_text(size = 5),axis.text.y = element_text(s
plot_grid(plot4)
```



Modelling

```
greg$Clicked.on.Ad = as.factor(greg$Clicked.on.Ad)
training1 <- createDataPartition(y = greg$Clicked.on.Ad, p = .75, list = FALSE)
```

```
training <- greg[training1,]
testing  <- greg[-training1,]
```

```
cluster <- makeCluster(detectCores() - 1)
registerDoParallel(cluster)
controlknn <- trainControl(method = "repeatedcv", number = 10, repeats = 3, verboseIter = TRUE)
KNNall <- train(Clicked.on.Ad ~ .,data = training, method = "knn",trControl = controlknn,preProc = c("c
```

I'll first split the dataset into train and test set

```
## Aggregating results
## Selecting tuning parameters
## Fitting k = 19 on full training set
```

```
KNNall
```

```
## k-Nearest Neighbors
##
## 750 samples
##   9 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (3207), scaled (3207)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 675, 675, 675, 674, 676, 676, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.6111623  0.2223081
##    7  0.6300458  0.2596929
##    9  0.6375429  0.2749510
##   11  0.6301688  0.2609002
##   13  0.6572352  0.3140698
##   15  0.6500955  0.2993288
##   17  0.6674260  0.3336526
##   19  0.7017934  0.4021256
##   21  0.6945978  0.3873357
##   23  0.6932516  0.3846678
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 19.
```

Random Forest

```
sample<- sample(c(TRUE, FALSE),nrow(greg),replace= T, prob = c(0.75, 0.25))
train<- greg[sample, ]
test<- greg[!sample, ]
dim(test)
```

```
## [1] 225  10
```

```
dim(train)
```

```
## [1] 775  10
```

```
var1 = c('Male', 'Clicked.on.Ad', 'Ad.Topic.Line')
for (i in var1){
    greg[,i] = as.factor(greg[,i])
}
```

```
var2 = c('Area.Income')
for (i in var2){
    greg[,i] = as.integer(greg[,i])
}
```

```
names(greg)
```

```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "Ad.Topic.Line"            "City"
##  [7] "Male"                     "Country"
##  [9] "Timestamp"                "Clicked.on.Ad"
```