Untitled

group1

18/03/2022

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(moderndive)
library(skimr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(dplyr)
library(readr)
library(Stat2Data)
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

load data from csv files

```
data <- read.csv("dataset1.csv", na.strings = "") %>% rename("Number_of_Family"=7,
                                                            "FoodExpenditure" = 3,
                                                            "Gender" = 4,
                                                            "Age" = 5,
                                                            "Type" = 6,
                                                            "Area" = 8,
                                                            "HouseAge" = 9,
                                                            "bedrooms" = 10)
glimpse(data)
```

```
## Rows: 1,725
## Columns: 11
## $ Total.Household.Income <int> 480332, 198235, 82785, 107589, 189322, 152883, ~
## $ Region                 <chr> "CAR", "CAR", "CAR", "CAR", "CAR", "CAR", "CAR"~
## $ FoodExpenditure        <int> 117848, 67766, 61609, 78189, 94625, 73326, 1046~
## $ Gender                 <chr> "Female", "Male", "Male", "Male", "Male", "Male~
## $ Age                    <int> 49, 40, 39, 52, 65, 46, 45, 33, 17, 53, 49, 35,~
## $ Type                   <chr> "Extended Family", "Single Family", "Single Fam~
## $ Number_of_Family       <int> 4, 3, 6, 3, 4, 4, 5, 5, 2, 6, 4, 7, 7, 3, 2, 4,~
## $ Area                   <int> 80, 42, 35, 30, 54, 40, 35, 35, 35, 70, 40, 35,~
## $ HouseAge               <int> 75, 15, 12, 15, 16, 7, 18, 48, 8, 12, 9, 17, 5,~
## $ bedrooms               <int> 3, 2, 1, 1, 3, 2, 1, 2, 1, 3, 2, 3, 1, 3, 1, 1,~
## $ Electricity            <int> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,~
```

FoodExpenditure is the annual expenditure by the household on food (in Philippine peso) *Gender* is the head of the households sex *Age* is the head of the households age (in years) *Type* is the relationship between the group of people living in the house $Number_of_Family$ is the number of people living in the house *Area* is the floor area of the house (in m^2) *HouseAge* is the age of the building (in years) *bedrooms* is the number of bedrooms in the house *Electricity* indicates that if the house have electricity? (1=Yes, 0=No)

convert chr into factor

```
data$Region <- factor(data$Region)
data$Gender <- factor(data$Gender)
data$Type <- factor(data$Type)
```

Check continuous variables

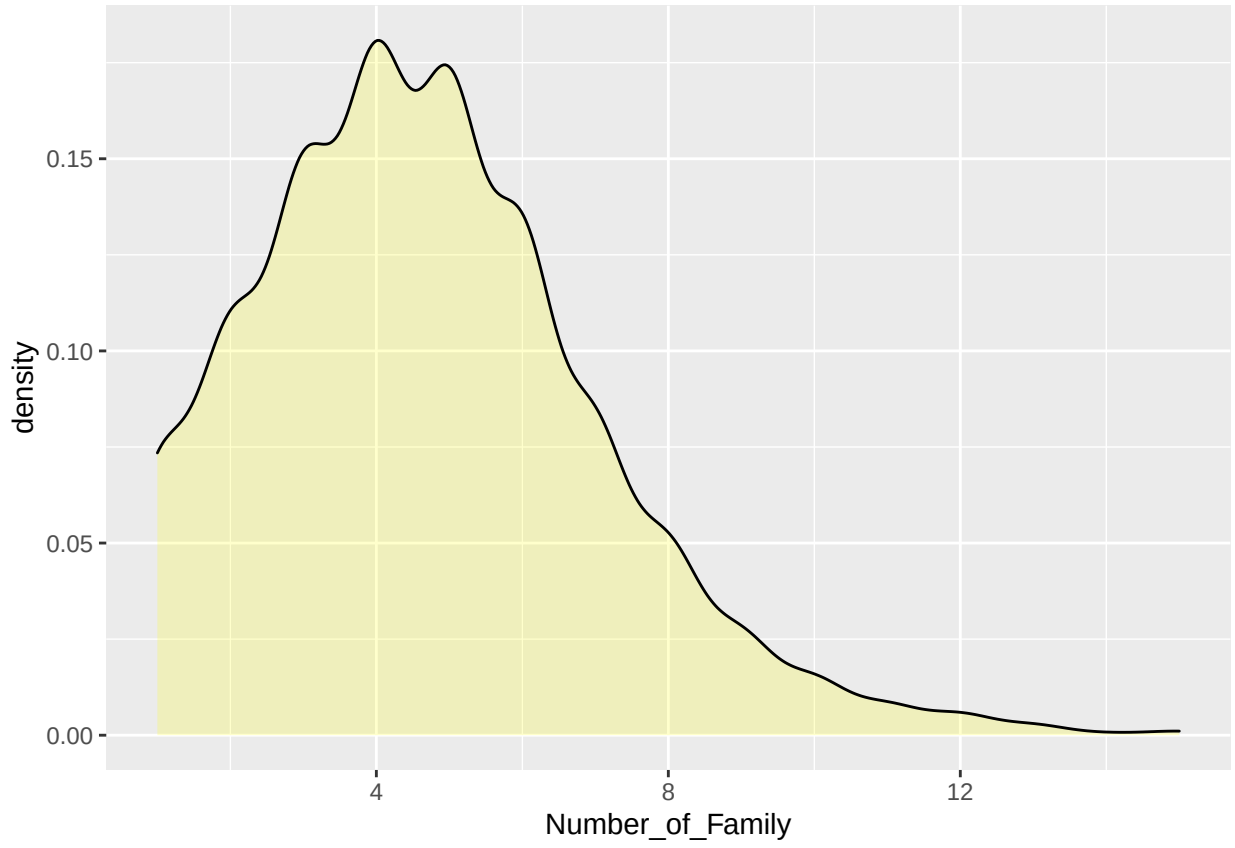
```
continuous <-select_if(data, is.numeric)
summary(continuous)
```

```
##  Total.Household.Income FoodExpenditure       Age        Number_of_Family
##  Min.   :  11988        Min.   :  6781   Min.   :17.00   Min.   : 1.000
##  1st Qu.: 118565        1st Qu.: 51922   1st Qu.:41.00   1st Qu.: 3.000
##  Median : 188580        Median : 73578   Median :52.00   Median : 4.000
##  Mean   : 269540        Mean   : 80353   Mean   :52.23   Mean   : 4.669
##  3rd Qu.: 328335        3rd Qu.: 98493   3rd Qu.:63.00   3rd Qu.: 6.000
##  Max.   :6042860        Max.   :327724   Max.   :99.00   Max.   :15.000
##       Area           HouseAge         bedrooms      Electricity
##  Min.   :  5.00   Min.   :  0.00   Min.   :0.000   Min.   :0.0000
##  1st Qu.: 32.00   1st Qu.: 12.00   1st Qu.:1.000   1st Qu.:1.0000
##  Median : 54.00   Median : 20.00   Median :2.000   Median :1.0000
```

```
##  Mean   : 90.92   Mean   : 22.98   Mean   :2.259   Mean   :0.9252
##  3rd Qu.:102.00   3rd Qu.: 31.00   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :900.00   Max.   :100.00   Max.   :9.000   Max.   :1.0000
```

data have totally different scales and many of them have large outliers, may need to standardize them?

```
ggplot(continuous, aes(x = Number_of_Family )) + geom_density(alpha = .2, fill = "yellow")
```



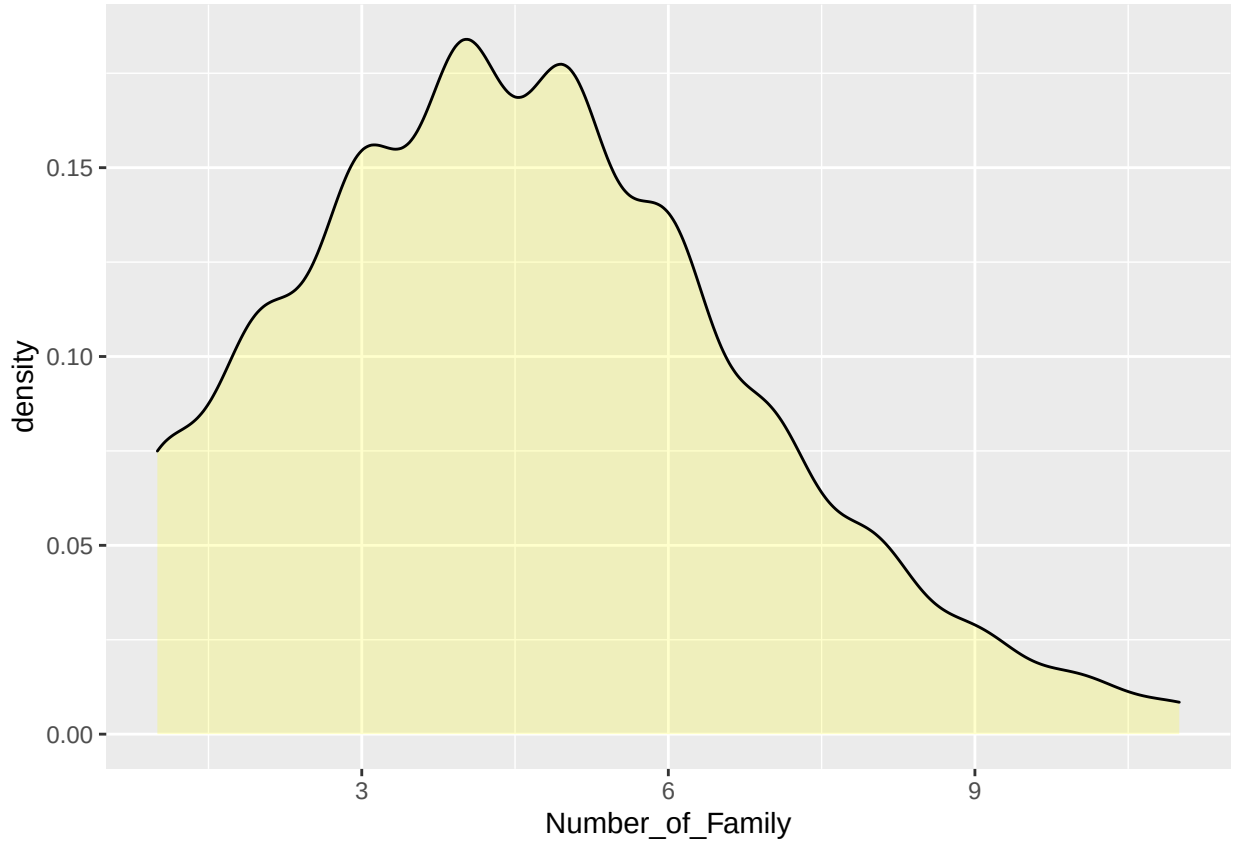

```
top_one_percent <- quantile(data$Number_of_Family , .99)
top_one_percent
```

```
##   99%
## 11.76
```

```
data_drop <-data %>%
  filter(Number_of_Family <top_one_percent)
dim(data_drop)
```

```
## [1] 1707   11
```

```
ggplot(data_drop, aes(x = Number_of_Family)) + geom_density(alpha = .2, fill = "yellow")
```




```
data_rescale <- data_drop %>%
  mutate_if(is.numeric, funs(as.numeric(scale(.))))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
glimpse(data_rescale)
```

```
## Rows: 1,707
## Columns: 11
## $ Total.Household.Income <dbl> 0.77848837, -0.25717086, -0.68102103, -0.589958~
## $ Region                 <fct> CAR, CAR, CAR, CAR, CAR, CAR, CAR, CAR, CAR, CA~
## $ FoodExpenditure        <dbl> 0.93182887, -0.29990887, -0.45133671, -0.043561~
## $ Gender                 <fct> Female, Male, Male, Male, Male, Male, Male, Mal~
```

```
## $ Age                    <dbl> -0.21983038, -0.83803541, -0.90672486, -0.01376~
## $ Type                   <fct> Extended Family, Single Family, Single Family, ~
## $ Number_of_Family       <dbl> -0.2666227, -0.7231170, 0.6463661, -0.7231170, ~
## $ Area                   <dbl> -0.1065545, -0.4940938, -0.5654826, -0.6164746,~
## $ HouseAge               <dbl> 3.3900496, -0.5194760, -0.7149523, -0.5194760, ~
## $ bedrooms               <dbl> 0.5165344, -0.1799302, -0.8763949, -0.8763949, ~
## $ Electricity            <dbl> 0.2858341, 0.2858341, -3.4964828, 0.2858341, 0.~
```

99% of the family member is below 11.76, drop the observations above this threshold?

check factor varialbles

```
factor <- data.frame(select_if(data_rescale, is.factor))
ncol(factor)
```

```
## [1] 3
```

Create graph for each column

```
data$Number_of_Family <- factor(data$Number_of_Family)

graph <- lapply(names(factor),
                function(x)
                  ggplot(factor, aes(get(x))) +
                  geom_bar(width = 0.1) +
                  theme(axis.text.x = element_text(angle = 90)))
```

Recast Feature

Change level family number as it has too many levels.

ver.1

```
#recast_data <- data_rescale %>%
#  select(-x) %>%
#  mutate(Number_of_Family = factor(ifelse(Number_of_Family == "1" | Number_of_Family == "2" | Number_o
#                                   ifelse(Number_of_Family == "7"| Number_of_Family == "8"| Number_of_F
#                                    ifelse( Number_of_Family == "13" | Number_of_Family == "14"| Number
```

ver.2

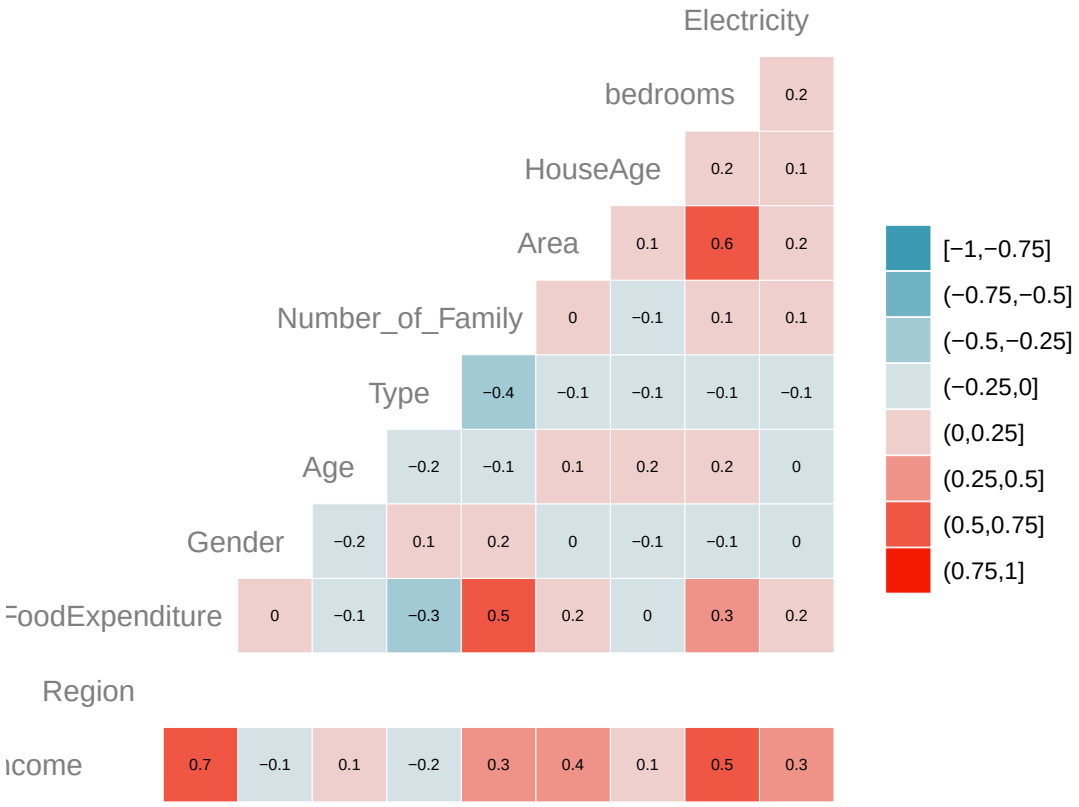
```
#recast_data <- data_rescale %>%
#  mutate(Number_of_Family = factor(ifelse(Number_of_Family == "1" | Number_of_Family == "2" | Number_o
#                                   ifelse(Number_of_Family == "7" | Number_of_Family == "8" | Number_o
#                                   ifelse(Number_of_Family == "13"| Number_of_Family == "14"| Number_o
```

Summary Statistic

visualize the correlation between the variables

```r
corr <- data.frame(lapply(data, as.integer)) #Convert data to numeric
ggcorr(corr, method = c("pairwise", "spearman"),
             nbreaks = 8,
             hjust = 0.9,
             label = TRUE,
             label_size = 2,
             color = "grey50")
```

```
## Warning in cor(data, use = method[1], method = method[2]): the standard
## deviation is zero

## Warning in cor(data, use = method[1], method = method[2]): the standard
## deviation is zero

## Warning in cor(data, use = method[1], method = method[2]): the standard
## deviation is zero

## Warning in cor(data, use = method[1], method = method[2]): the standard
## deviation is zero

## Warning in cor(data, use = method[1], method = method[2]): the standard
## deviation is zero

## Warning in cor(data, use = method[1], method = method[2]): the standard
## deviation is zero

## Warning in cor(data, use = method[1], method = method[2]): the standard
## deviation is zero

## Warning in cor(data, use = method[1], method = method[2]): the standard
## deviation is zero

## Warning in cor(data, use = method[1], method = method[2]): the standard
## deviation is zero

## Warning in cor(data, use = method[1], method = method[2]): the standard
## deviation is zero

## Warning in cor(data, use = method[1], method = method[2]): the standard
## deviation is zero
```



Train/test set

split the data between a train set and a test set (for machine learning task if needed)

```
set.seed(1234)
create_train_test <- function(data1, size = 0.8, train = TRUE) {
  n_row = nrow(data1)
  total_row = size * n_row
  train_sample <- 1: total_row
  if (train == TRUE) {
    return (data1[train_sample, ])
  } else {
    return (data1[-train_sample, ])}}

data_train <- create_train_test(data, 0.8, train = TRUE)
data_test <- create_train_test(data, 0.8, train = FALSE)
```

Generalized Linear Model

```
model <- glm(Number_of_Family~FoodExpenditure+Gender+Age+Type+Area+HouseAge+bedrooms+Electricity,
             data = data_train, family = 'binomial')
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model)
```

```
##
## Call:
## glm(formula = Number_of_Family ~ FoodExpenditure + Gender + Age +
##     Type + Area + HouseAge + bedrooms + Electricity, family = "binomial",
##     data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.0427   0.0000   0.0329   0.2044   1.8371
##
## Coefficients:
##                                           Estimate Std. Error z value
## (Intercept)                               1.499e+01  6.338e+02   0.024
## FoodExpenditure                           8.974e-05  9.283e-06   9.667
## GenderMale                                1.168e+00  3.102e-01   3.764
## Age                                      -1.091e-02  8.681e-03  -1.257
## TypeSingle Family                        -1.754e+01  6.338e+02  -0.028
## TypeTwo or More Nonrelated Persons/Members -2.565e+00  5.723e+03   0.000
## Area                                     -1.477e-03  1.759e-03  -0.840
## HouseAge                                  3.459e-03  9.312e-03   0.372
## bedrooms                                 -3.564e-01  1.219e-01  -2.924
## Electricity                               4.130e-01  4.173e-01   0.990
##                                          Pr(>|z|)
## (Intercept)                              0.981134
## FoodExpenditure                           < 2e-16 ***
## GenderMale                               0.000167 ***
## Age                                      0.208775
## TypeSingle Family                        0.977922
## TypeTwo or More Nonrelated Persons/Members 0.999642
## Area                                     0.400875
## HouseAge                                 0.710259
## bedrooms                                 0.003457 **
## Electricity                              0.322331
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 717.51  on 1379  degrees of freedom
## Residual deviance: 353.10  on 1370  degrees of freedom
## AIC: 373.1
##
## Number of Fisher Scoring iterations: 19
```

The summary of our model reveals interesting information. The performance of a logistic regression is evaluated with specific key metrics.

Assess the performance of the model

The logistic regression can be evaluated through the output of the glm() function which stored in a list. Below we print the first five elements to see the results.

```r
lapply(model, class)[1:5]
```

```
## $coefficients
## [1] "numeric"
##
## $residuals
## [1] "numeric"
##
## $fitted.values
## [1] "numeric"
##
## $effects
## [1] "numeric"
##
## $R
## [1] "matrix" "array"
```

```r
model$aic
```

```
## [1] 373.0977
```

```r
predict <- predict(model, data_test, type = 'response')
```

```r
table_mat <- table(data_test$Number_of_Family, predict > 0.5)
table_mat
```

```
##
##       FALSE TRUE
##   1      23    5
##   2      21   18
##   3       2   47
##   4       0   43
##   5       1   55
##   6       0   54
##   7       0   29
##   8       0   24
##   9       0   10
##   10      0    7
##   11      0    1
##   12      0    3
##   13      0    1
##   14      0    0
##   15      0    1
```

check model accuracy

```r
accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
accuracy_Test
```

```
## [1] 0.1188406
```