Project 2: Advanced Statistics

PCA / FA & Regression Models

Mercy Chrysolite
G3 PGPBABI

Contents

Cereal Data Factor Analysis

Objective

As part of a study of consumer consideration of ready-to-eat cereals sponsored by Kellogg

Australia, Roberts and Lattin (1991) surveyed consumers regarding their perceptions of their favorite brands of cereals. Each respondent was asked to evaluate three preferred brands on each of 25 different attributes. Respondents used a five point Likert scale to indicate the extent to which each brand possessed the given attribute.

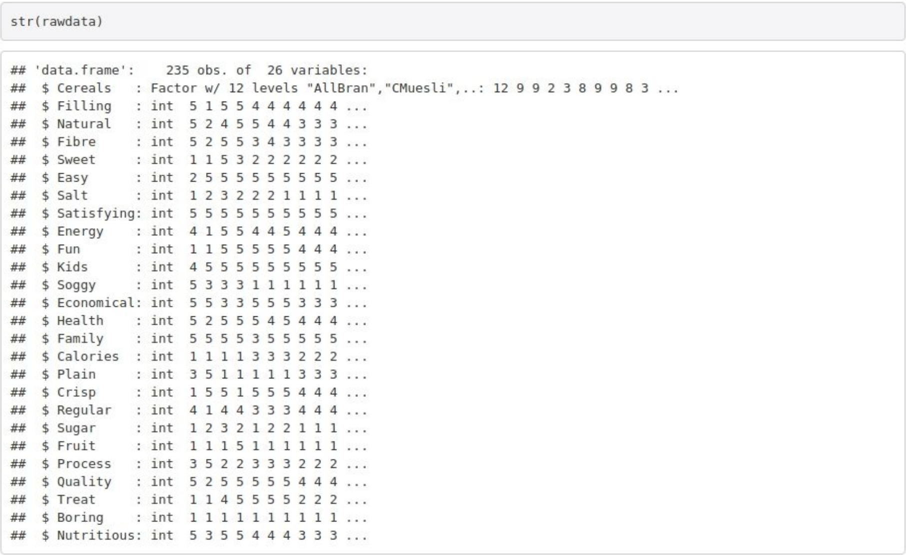
For the purpose of this assignment, a subset of the data collected by Roberts and Lattin, reflecting the evaluations of the 12 most frequently cited cereal brands in the sample (in the original study, a total of 40 different brands were evaluated by 121 respondents, but the majority of brands were rated by only a small number of consumers).

In total, 116 respondents provided 235 observations of the 12 selected brands. How do you characterize the consideration behavior of the 12 selected brands? Analyze and interpret your results using factor analysis.

Exploratory Data Analysis

The following inferences are made from analysing the data:

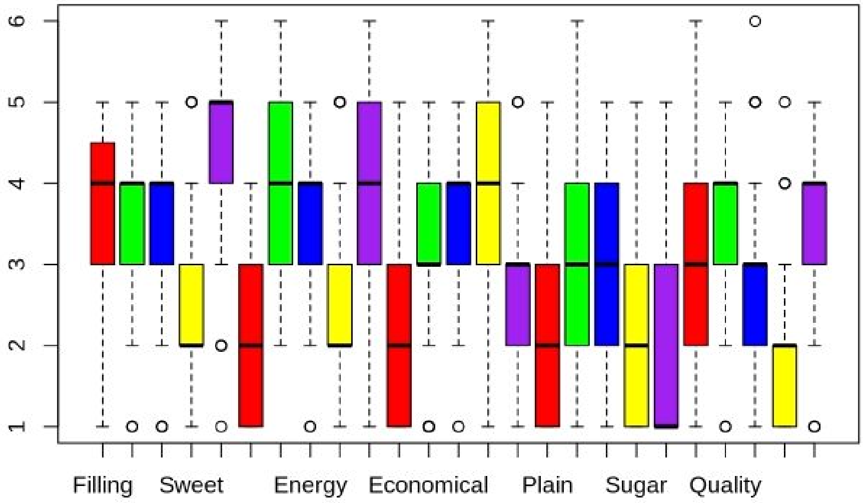
- There are 235 rows and 26 variables
- There are no missing values
- All variables are numerical; except the cereal brands. The remaining 25 variables represent attributes on which rating was done.



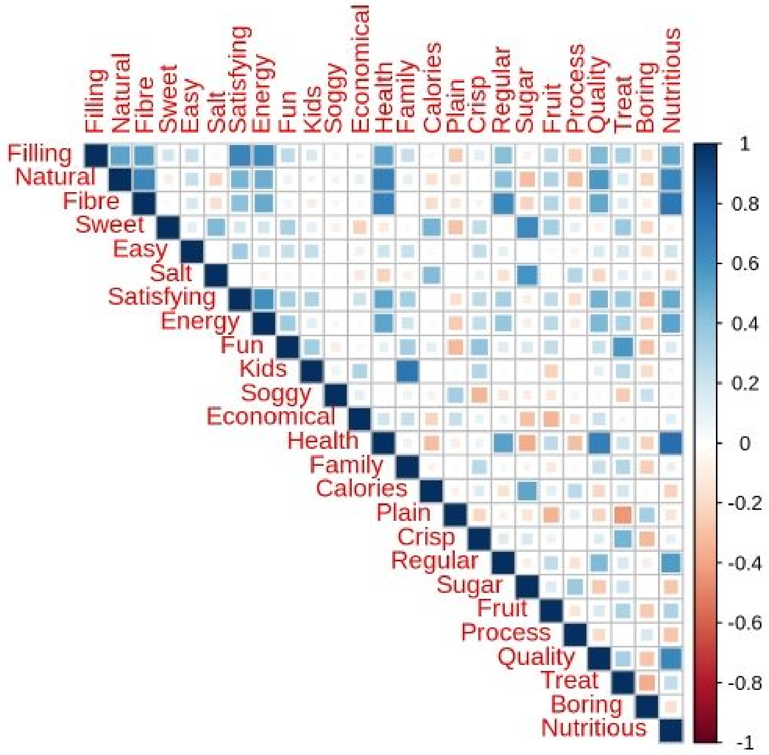
```
str(rawdata)
```

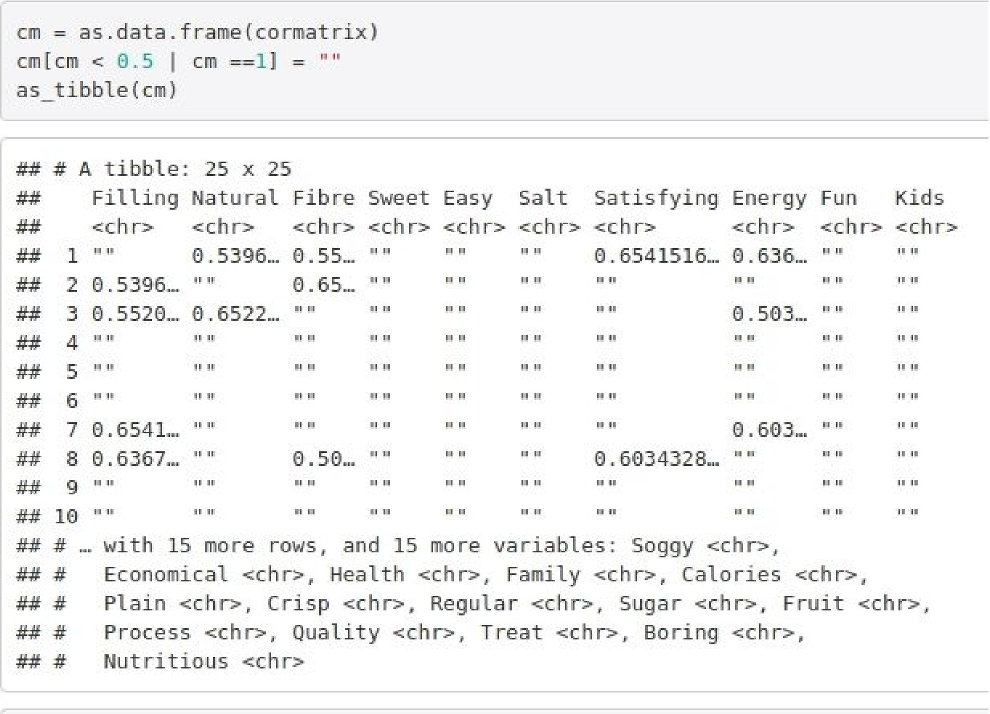
```
## 'data.frame':    235 obs. of  26 variables:
##  $ Cereals   : Factor w/ 12 levels "AllBran","CMuesli",..: 12 9 9 2 3 8 9 9 8 3 ...
##  $ Filling   : int  5 1 5 5 4 4 4 4 4 4 ...
##  $ Natural   : int  5 2 4 5 5 4 4 3 3 3 ...
##  $ Fibre     : int  5 2 5 5 3 4 3 3 3 3 ...
##  $ Sweet     : int  1 1 5 3 2 2 2 2 2 2 ...
##  $ Easy      : int  2 5 5 5 5 5 5 5 5 5 ...
##  $ Salt      : int  1 2 3 2 2 2 1 1 1 1 ...
##  $ Satisfying: int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Energy    : int  4 1 5 5 4 4 5 4 4 4 ...
##  $ Fun       : int  1 1 5 5 5 5 5 4 4 4 ...
##  $ Kids      : int  4 5 5 5 5 5 5 5 5 5 ...
##  $ Soggy     : int  5 3 3 3 1 1 1 1 1 1 ...
##  $ Economical: int  5 5 3 3 5 5 5 3 3 3 ...
##  $ Health    : int  5 2 5 5 5 4 5 4 4 4 ...
##  $ Family    : int  5 5 5 5 3 5 5 5 5 5 ...
##  $ Calories  : int  1 1 1 1 3 3 3 2 2 2 ...
##  $ Plain     : int  3 5 1 1 1 1 1 3 3 3 ...
##  $ Crisp     : int  1 5 5 1 5 5 5 4 4 4 ...
##  $ Regular   : int  4 1 4 4 3 3 3 4 4 4 ...
##  $ Sugar     : int  1 2 3 2 1 2 2 1 1 1 ...
##  $ Fruit     : int  1 1 1 5 1 1 1 1 1 1 ...
##  $ Process   : int  3 5 2 2 3 3 3 2 2 2 ...
##  $ Quality   : int  5 2 5 5 5 5 5 4 4 4 ...
##  $ Treat     : int  1 1 4 5 5 5 5 2 2 2 ...
##  $ Boring    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Nutritious: int  5 3 5 5 4 4 4 3 3 3 ...
```

- The box plot shows the distribution and skewness of the variables. Outliers exist but lie within 1-5. And hence are not removed. Values above 5 exist and are removed.



- Multicollinearity is highly prevalent between many variables. Showing variables with collinearity more the 0.5. Therefore PCA/FA has to be done.



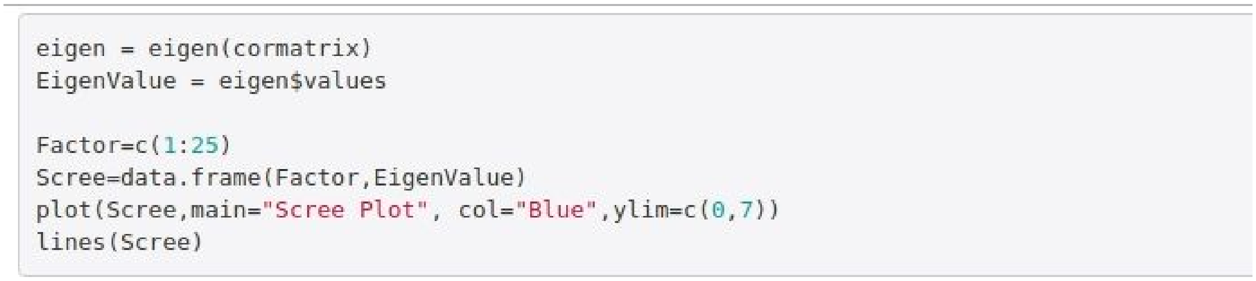


```
cm = as.data.frame(cormatrix)
cm[cm < 0.5 | cm ==1] = ""
as_tibble(cm)
```

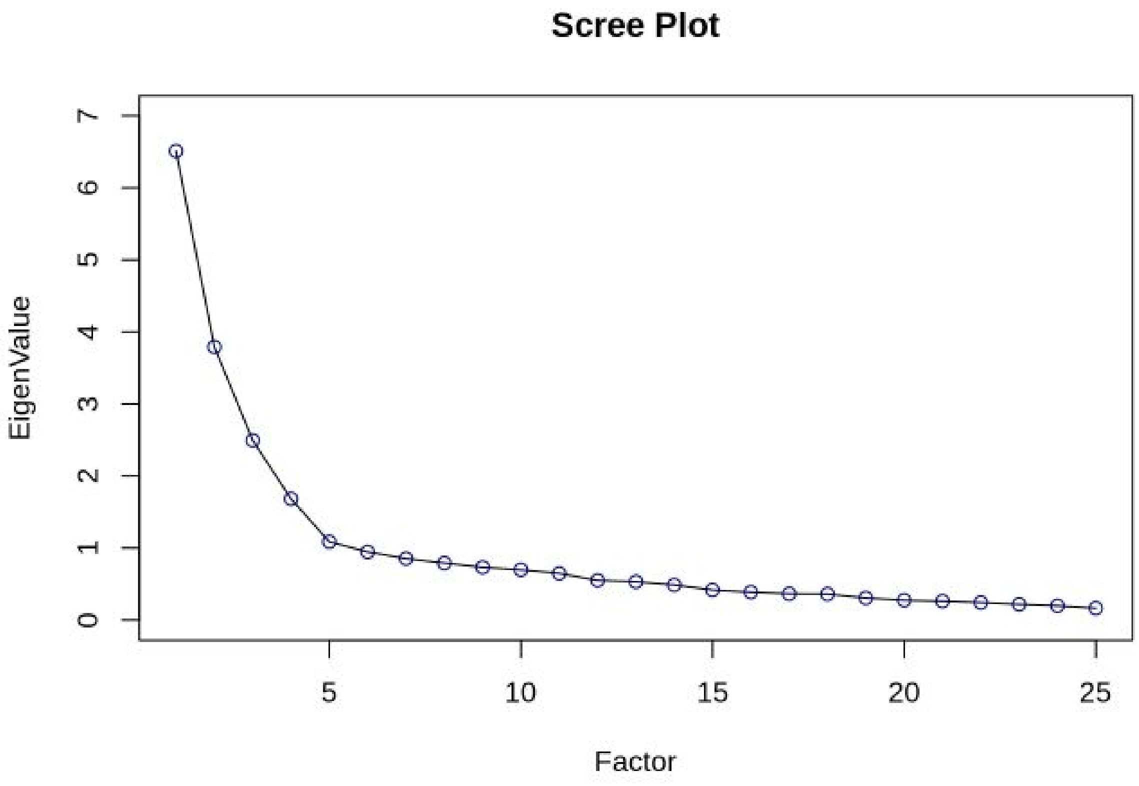
```
## # A tibble: 25 x 25
##    Filling Natural Fibre Sweet Easy  Salt  Satisfying Energy Fun   Kids
##    <chr>   <chr>   <chr> <chr> <chr> <chr> <chr>      <chr>  <chr> <chr>
##  1 ""      0.5396… 0.55… ""    ""    ""    0.6541516… 0.636… ""    ""
##  2 0.5396… ""      0.65… ""    ""    ""    ""         ""     ""    ""
##  3 0.5520… 0.6522… ""    ""    ""    ""    ""         0.503… ""    ""
##  4 ""      ""      ""    ""    ""    ""    ""         ""     ""    ""
##  5 ""      ""      ""    ""    ""    ""    ""         ""     ""    ""
##  6 ""      ""      ""    ""    ""    ""    ""         ""     ""    ""
##  7 0.6541… ""      ""    ""    ""    ""    ""         0.603… ""    ""
##  8 0.6367… ""      0.50… ""    ""    ""    0.6034328… ""     ""    ""
##  9 ""      ""      ""    ""    ""    ""    ""         ""     ""    ""
## 10 ""      ""      ""    ""    ""    ""    ""         ""     ""    ""
## # … with 15 more rows, and 15 more variables: Soggy <chr>,
## #   Economical <chr>, Health <chr>, Family <chr>, Calories <chr>,
## #   Plain <chr>, Crisp <chr>, Regular <chr>, Sugar <chr>, Fruit <chr>,
## #   Process <chr>, Quality <chr>, Treat <chr>, Boring <chr>,
## #   Nutritious <chr>
```

PCA

In order to remove multicollinearity, we can use either Principal Component Analysis or Factor Analysis. In order to proceed, we first need to find the eigen values and consquently, the best number of factors.

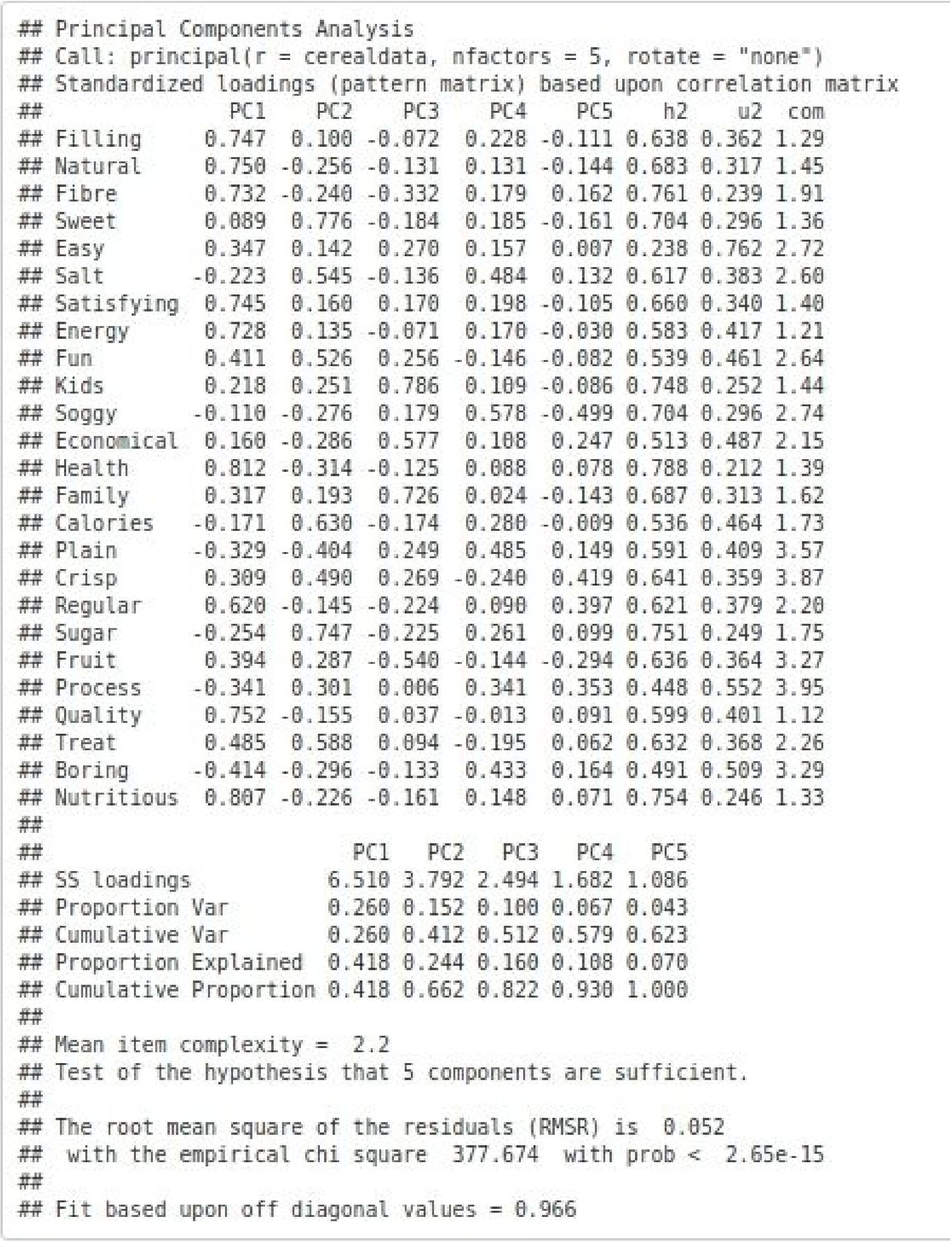


```
eigen = eigen(cormatrix)
EigenValue = eigen$values

Factor=c(1:25)
Scree=data.frame(Factor,EigenValue)
plot(Scree,main="Scree Plot", col="Blue",ylim=c(0,7))
lines(Scree)
```



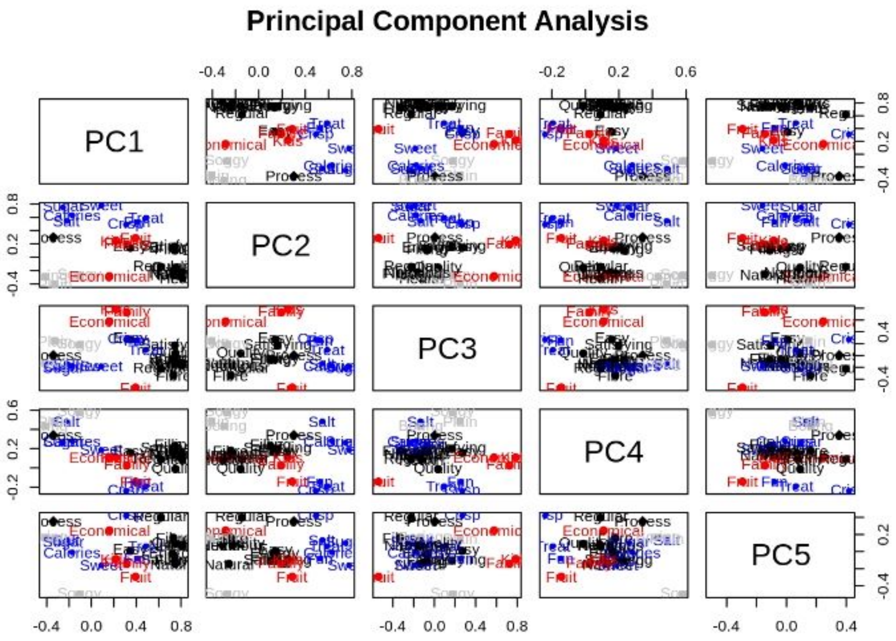
From above plot, we can observe that 4 factors have eigen value > 1 and as per Kaiser rule, we can consider ***5 factors***.

The loadings from unrotated computation are as follows:

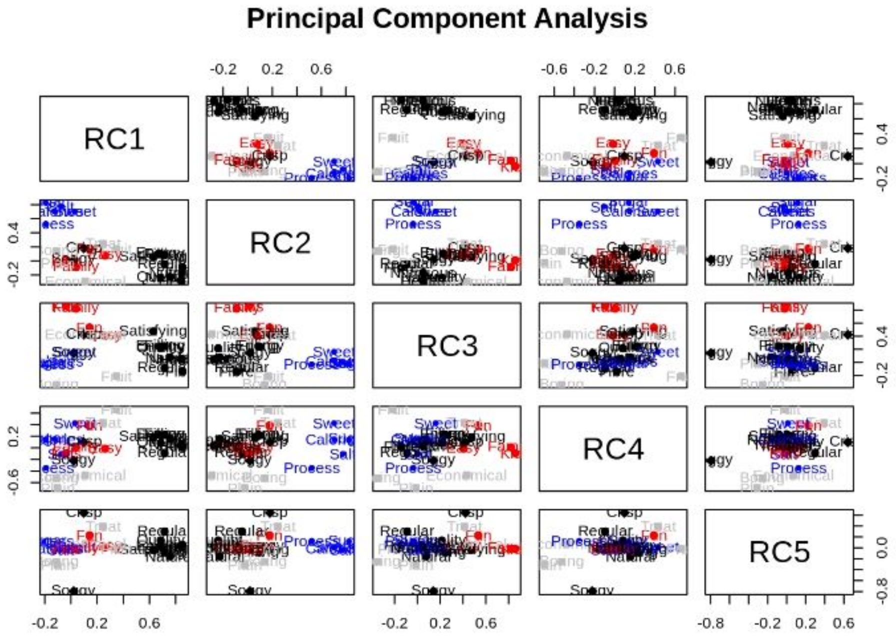


```
## Principal Components Analysis
## Call: principal(r = cerealdata, nfactors = 5, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##               PC1    PC2    PC3    PC4    PC5    h2    u2  com
## Filling     0.747  0.100 -0.072  0.228 -0.111 0.638 0.362 1.29
## Natural     0.750 -0.256 -0.131  0.131 -0.144 0.683 0.317 1.45
## Fibre       0.732 -0.240 -0.332  0.179  0.162 0.761 0.239 1.91
## Sweet       0.089  0.776 -0.184  0.185 -0.161 0.704 0.296 1.36
## Easy        0.347  0.142  0.270  0.157  0.007 0.238 0.762 2.72
## Salt       -0.223  0.545 -0.136  0.484  0.132 0.617 0.383 2.60
## Satisfying  0.745  0.160  0.170  0.198 -0.105 0.660 0.340 1.40
## Energy      0.728  0.135 -0.071  0.170 -0.030 0.583 0.417 1.21
## Fun         0.411  0.526  0.256 -0.146 -0.082 0.539 0.461 2.64
## Kids        0.218  0.251  0.786  0.109 -0.086 0.748 0.252 1.44
## Soggy      -0.110 -0.276  0.179  0.578 -0.499 0.704 0.296 2.74
## Economical  0.160 -0.286  0.577  0.108  0.247 0.513 0.487 2.15
## Health      0.812 -0.314 -0.125  0.088  0.078 0.788 0.212 1.39
## Family      0.317  0.193  0.726  0.024 -0.143 0.687 0.313 1.62
## Calories   -0.171  0.630 -0.174  0.280 -0.009 0.536 0.464 1.73
## Plain      -0.329 -0.404  0.249  0.485  0.149 0.591 0.409 3.57
## Crisp       0.309  0.490  0.269 -0.240  0.419 0.641 0.359 3.87
## Regular     0.620 -0.145 -0.224  0.090  0.397 0.621 0.379 2.20
## Sugar      -0.254  0.747 -0.225  0.261  0.099 0.751 0.249 1.75
## Fruit       0.394  0.287 -0.540 -0.144 -0.294 0.636 0.364 3.27
## Process    -0.341  0.301  0.006  0.341  0.353 0.448 0.552 3.95
## Quality     0.752 -0.155  0.037 -0.013  0.091 0.599 0.401 1.12
## Treat       0.485  0.588  0.094 -0.195  0.062 0.632 0.368 2.26
## Boring     -0.414 -0.296 -0.133  0.433  0.164 0.491 0.509 3.29
## Nutritious  0.807 -0.226 -0.161  0.148  0.071 0.754 0.246 1.33
##
##                         PC1   PC2   PC3   PC4   PC5
## SS loadings           6.510 3.792 2.494 1.682 1.086
## Proportion Var        0.260 0.152 0.100 0.067 0.043
## Cumulative Var        0.260 0.412 0.512 0.579 0.623
## Proportion Explained  0.418 0.244 0.160 0.108 0.070
## Cumulative Proportion 0.418 0.662 0.822 0.930 1.000
##
## Mean item complexity =  2.2
## Test of the hypothesis that 5 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.052
##  with the empirical chi square  377.674  with prob <  2.65e-15
##
## Fit based upon off diagonal values = 0.966
```

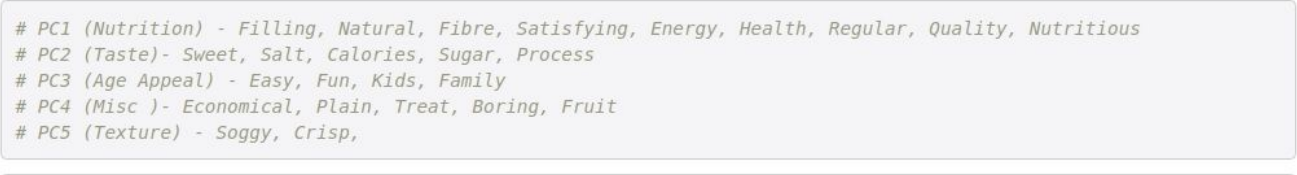
Since the *cumulative variance = 0.623,* it explains 62.3% of the variability. But from the graph, it is evident that the variables are not properly distributed.



To improve the graph, we use rotated method and come up with a better classification, with the same cumulative variance. This graph shows a clearer classification of the variables into factors.



We can thereby group the variables into the following components:



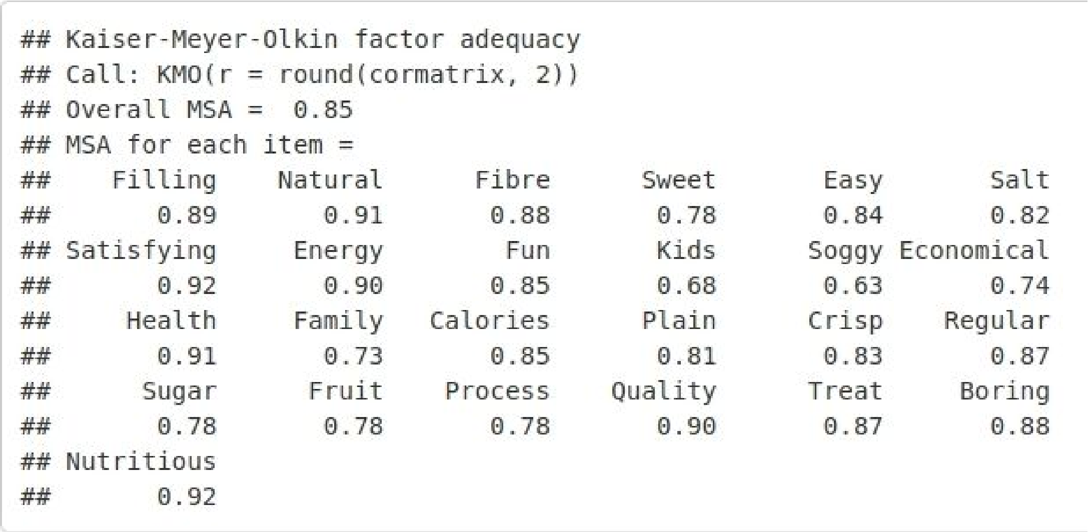
```
# PC1 (Nutrition) - Filling, Natural, Fibre, Satisfying, Energy, Health, Regular, Quality, Nutritious
# PC2 (Taste)- Sweet, Salt, Calories, Sugar, Process
# PC3 (Age Appeal) - Easy, Fun, Kids, Family
# PC4 (Misc )- Economical, Plain, Treat, Boring, Fruit
# PC5 (Texture) - Soggy, Crisp,
```

Factor Analysis

Before we find factors we need to find out if data is adequate and has enough sphericity. We will use KMO test for data adequacy and Bartlett test for sphericity.

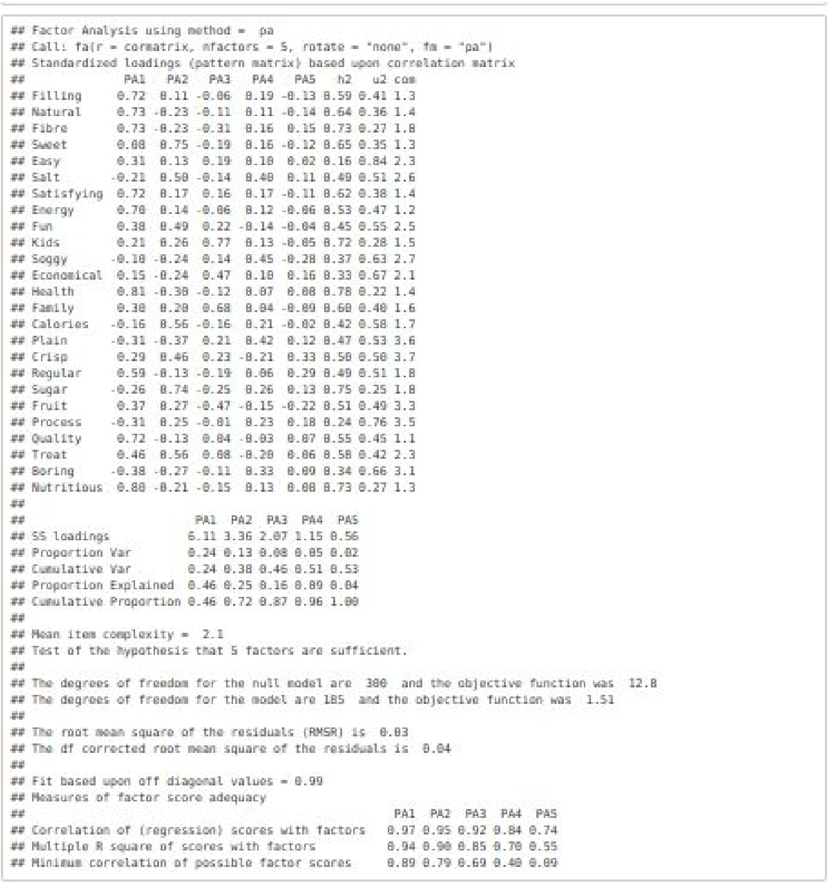
KMO test suggests data is adequate if MSA is between 0.5 & 1

Bartlett test suggest enough sphericity if p value is < 0.05

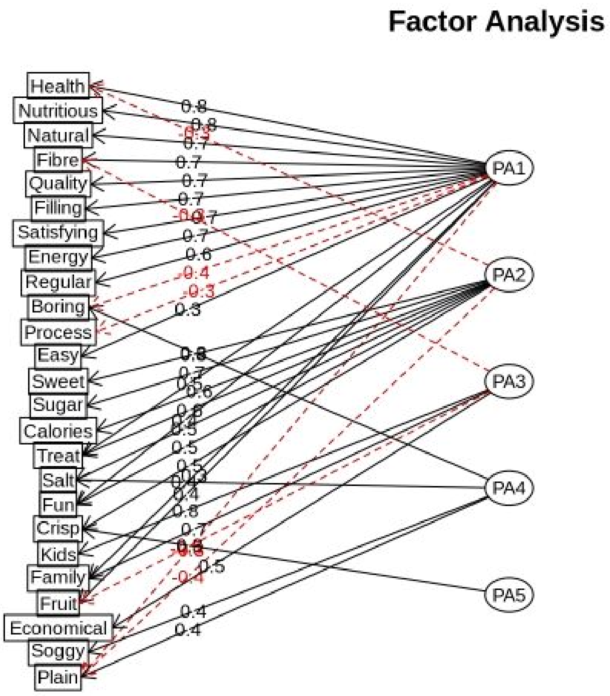


```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = round(cormatrix, 2))
## Overall MSA =  0.85
## MSA for each item =
##    Filling    Natural      Fibre      Sweet       Easy       Salt
##       0.89       0.91       0.88       0.78       0.84       0.82
## Satisfying     Energy        Fun       Kids      Soggy Economical
##       0.92       0.90       0.85       0.68       0.63       0.74
##     Health     Family   Calories      Plain      Crisp    Regular
##       0.91       0.73       0.85       0.81       0.83       0.87
##      Sugar      Fruit    Process    Quality      Treat     Boring
##       0.78       0.78       0.78       0.90       0.87       0.88
## Nutritious
##       0.92
```

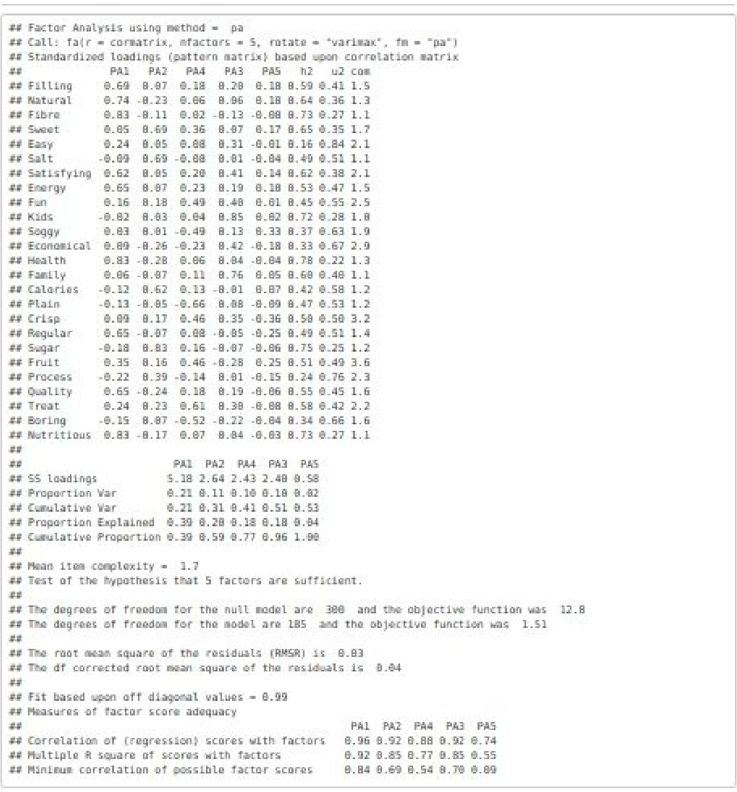
Hence we can proceed with factor analysis. The unrotated loadings and graph are given:



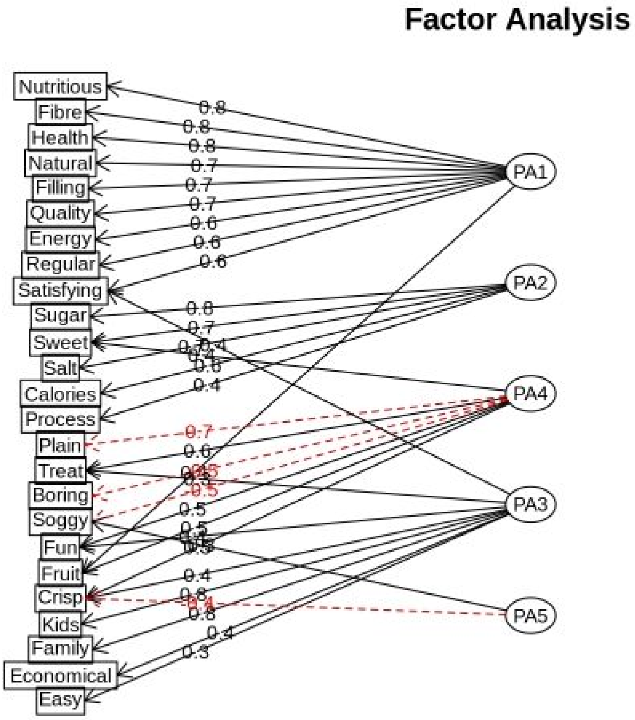
```
## Factor Analysis using method =  pa
## Call: fa(r = cormatrix, nfactors = 5, rotate = "none", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##              PA1   PA2   PA3   PA4   PA5   h2   u2 com
## Filling     0.72  0.11 -0.06  0.19 -0.13 0.59 0.41 1.3
## Natural     0.73 -0.23 -0.11  0.11 -0.14 0.64 0.36 1.4
## Fibre       0.73 -0.23 -0.31  0.16  0.15 0.73 0.27 1.8
## Sweet       0.08  0.75 -0.19  0.16 -0.12 0.65 0.35 1.3
## Easy        0.31  0.13  0.19  0.10  0.02 0.16 0.84 2.3
## Salt       -0.21  0.50 -0.14  0.40  0.11 0.49 0.51 2.6
## Satisfying  0.72  0.17  0.16  0.17 -0.11 0.62 0.38 1.4
## Energy      0.70  0.14 -0.06  0.12 -0.06 0.53 0.47 1.2
## Fun         0.38  0.49  0.22 -0.14 -0.04 0.45 0.55 2.5
## Kids        0.21  0.26  0.77  0.13 -0.05 0.72 0.28 1.5
## Soggy      -0.10 -0.24  0.14  0.45 -0.28 0.37 0.63 2.7
## Economical  0.15 -0.24  0.47  0.10  0.16 0.33 0.67 2.1
## Health      0.81 -0.30 -0.12  0.07  0.08 0.78 0.22 1.4
## Family      0.30  0.20  0.68  0.04 -0.09 0.60 0.40 1.6
## Calories   -0.16  0.56 -0.16  0.21 -0.02 0.42 0.58 1.7
## Plain      -0.31 -0.37  0.21  0.42  0.12 0.47 0.53 3.6
## Crisp       0.29  0.46  0.23 -0.21  0.33 0.50 0.50 3.7
## Regular     0.59 -0.13 -0.19  0.06  0.29 0.49 0.51 1.8
## Sugar      -0.26  0.74 -0.25  0.26  0.13 0.75 0.25 1.8
## Fruit       0.37  0.27 -0.47 -0.15 -0.22 0.51 0.49 3.3
## Process    -0.31  0.25 -0.01  0.23  0.18 0.24 0.76 3.5
## Quality     0.72 -0.13  0.04 -0.03  0.07 0.55 0.45 1.1
## Treat       0.46  0.56  0.08 -0.20  0.06 0.58 0.42 2.3
## Boring     -0.38 -0.27 -0.11  0.33  0.09 0.34 0.66 3.1
## Nutritious  0.80 -0.21 -0.15  0.13  0.08 0.73 0.27 1.3
##
##                        PA1  PA2  PA3  PA4  PA5
## SS loadings           6.11 3.36 2.07 1.15 0.56
## Proportion Var        0.24 0.13 0.08 0.05 0.02
## Cumulative Var        0.24 0.38 0.46 0.51 0.53
## Proportion Explained  0.46 0.25 0.16 0.09 0.04
## Cumulative Proportion 0.46 0.72 0.87 0.96 1.00
##
## Mean item complexity =  2.1
## Test of the hypothesis that 5 factors are sufficient.
##
## The degrees of freedom for the null model are  300  and the objective function was  12.8
## The degrees of freedom for the model are 185  and the objective function was  1.51
##
## The root mean square of the residuals (RMSR) is  0.03
## The df corrected root mean square of the residuals is  0.04
##
## Fit based upon off diagonal values = 0.99
## Measures of factor score adequacy
##                                                   PA1  PA2  PA3  PA4  PA5
## Correlation of (regression) scores with factors   0.97 0.95 0.92 0.84 0.74
## Multiple R square of scores with factors          0.94 0.90 0.85 0.70 0.55
## Minimum correlation of possible factor scores     0.89 0.79 0.69 0.40 0.09
```



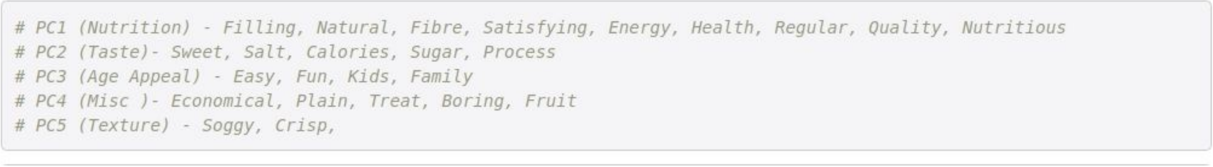
Once again, since the distribution is not clear, we do rotation.



```
## Factor Analysis using method =  pa
## Call: fa(r = cormatrix, nfactors = 5, rotate = "varimax", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##               PA1   PA2   PA4   PA3   PA5   h2   u2 com
## Filling      0.69  0.07  0.18  0.20  0.18 0.59 0.41 1.5
## Natural      0.74 -0.23  0.06  0.06  0.18 0.64 0.36 1.3
## Fibre        0.83 -0.11  0.02 -0.13 -0.08 0.73 0.27 1.1
## Sweet        0.05  0.69  0.36  0.07  0.17 0.65 0.35 1.7
## Easy         0.24  0.05  0.08  0.31 -0.01 0.16 0.84 2.1
## Salt        -0.09  0.69 -0.08  0.01 -0.04 0.49 0.51 1.1
## Satisfying   0.62  0.05  0.20  0.41  0.14 0.62 0.38 2.1
## Energy       0.65  0.07  0.23  0.19  0.10 0.53 0.47 1.5
## Fun          0.16  0.18  0.49  0.40  0.01 0.45 0.55 2.5
## Kids        -0.02  0.03  0.04  0.85  0.02 0.72 0.28 1.0
## Soggy        0.03  0.01 -0.49  0.13  0.33 0.37 0.63 1.9
## Economical   0.09 -0.26 -0.23  0.42 -0.18 0.33 0.67 2.9
## Health       0.83 -0.28  0.06  0.04 -0.04 0.78 0.22 1.3
## Family       0.06 -0.07  0.11  0.76  0.05 0.60 0.40 1.1
## Calories    -0.12  0.62  0.13 -0.01  0.07 0.42 0.58 1.2
## Plain       -0.13 -0.05 -0.66  0.08 -0.09 0.47 0.53 1.2
## Crisp        0.09  0.17  0.46  0.35 -0.36 0.50 0.50 3.2
## Regular      0.65 -0.07  0.08 -0.05 -0.25 0.49 0.51 1.4
## Sugar       -0.18  0.83  0.16 -0.07 -0.06 0.75 0.25 1.2
## Fruit        0.35  0.16  0.46 -0.28  0.25 0.51 0.49 3.6
## Process     -0.22  0.39 -0.14  0.01 -0.15 0.24 0.76 2.3
## Quality      0.65 -0.24  0.18  0.19 -0.06 0.55 0.45 1.6
## Treat        0.24  0.23  0.61  0.30 -0.08 0.58 0.42 2.2
## Boring      -0.15  0.07 -0.52 -0.22 -0.04 0.34 0.66 1.6
## Nutritious   0.83 -0.17  0.07  0.04 -0.03 0.73 0.27 1.1
##
##                        PA1  PA2  PA4  PA3  PA5
## SS loadings           5.18 2.64 2.43 2.40 0.58
## Proportion Var        0.21 0.11 0.10 0.10 0.02
## Cumulative Var        0.21 0.31 0.41 0.51 0.53
## Proportion Explained  0.39 0.20 0.18 0.18 0.04
## Cumulative Proportion 0.39 0.59 0.77 0.96 1.00
##
## Mean item complexity =  1.7
## Test of the hypothesis that 5 factors are sufficient.
##
## The degrees of freedom for the null model are  300  and the objective function was  12.8
## The degrees of freedom for the model are 185  and the objective function was  1.51
##
## The root mean square of the residuals (RMSR) is  0.03
## The df corrected root mean square of the residuals is  0.04
##
## Fit based upon off diagonal values = 0.99
## Measures of factor score adequacy
##                                                   PA1  PA2  PA4  PA3  PA5
## Correlation of (regression) scores with factors   0.96 0.92 0.88 0.92 0.74
## Multiple R square of scores with factors          0.92 0.85 0.77 0.85 0.55
## Minimum correlation of possible factor scores     0.84 0.69 0.54 0.70 0.09
```

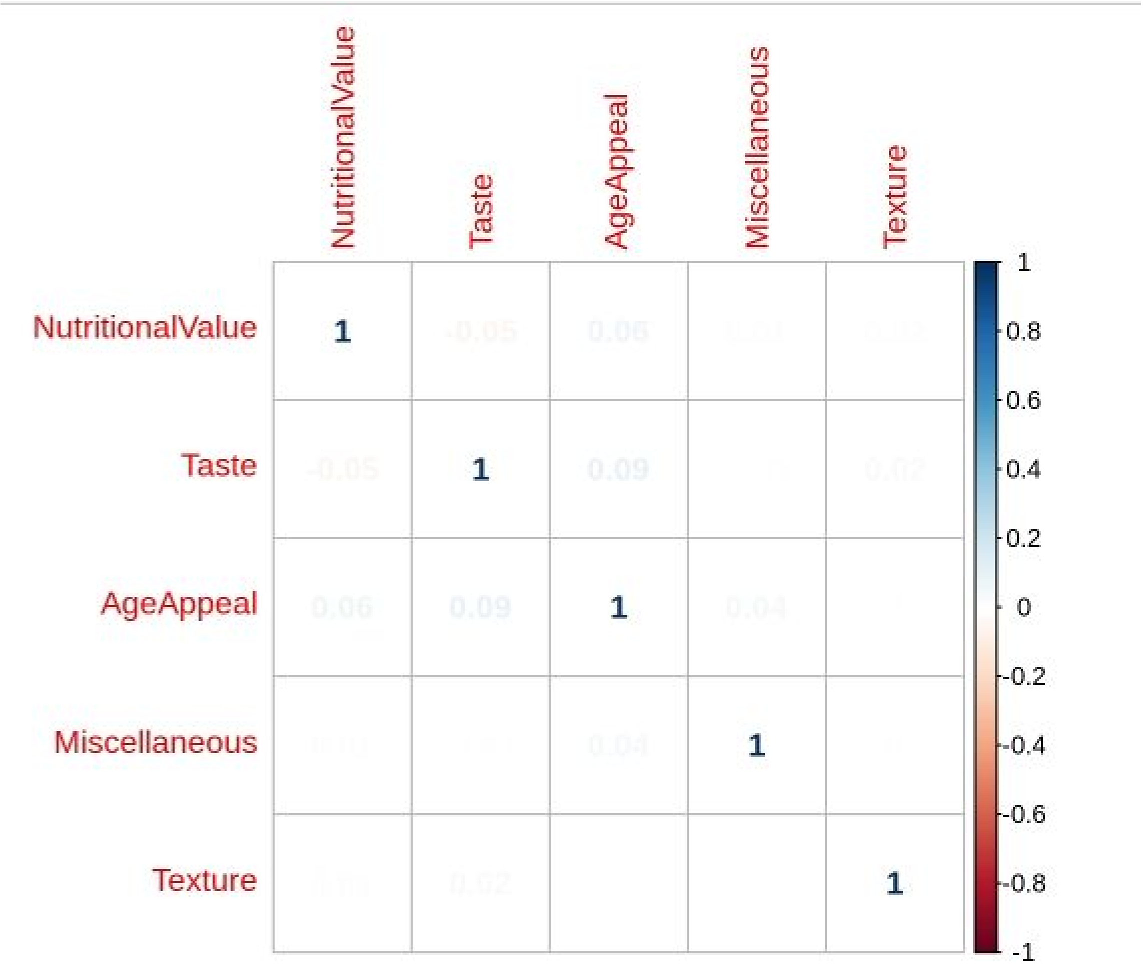


The rotated graph shows clearer distinction and each of the factors is mapped to similar variables as we got in PCA



```
# PC1 (Nutrition) - Filling, Natural, Fibre, Satisfying, Energy, Health, Regular, Quality, Nutritious
# PC2 (Taste)- Sweet, Salt, Calories, Sugar, Process
# PC3 (Age Appeal) - Easy, Fun, Kids, Family
# PC4 (Misc )- Economical, Plain, Treat, Boring, Fruit
# PC5 (Texture) - Soggy, Crisp,
```

Finally, we verify whether the collinearity is reduced. And results are as expected.



Leslie Salt Dataset

Objective

In 1968, the city of Mountain View, California began the necessary legal proceedings to acquire a parcel of land owned by the Leslie Salt Company. The Leslie property contained 246.8 acres and was located right on the San Francisco Bay. The land had been used for salt evaporation and had an elevation of exactly sea level. However, the property was diked so that the waters from the bay park were kept out. The city of Mountain View intended to fill the property and use it for a city park.

Ultimately, it fell into the hands of the courts to determine a fair market value for the property. Appraisers were hired, but what made the processes difficult was that there were very few sales of a byland property and none of them corresponded exactly to the characteristics of the Leslie property. The experts involved decided to build a regression model to better understand the factors that might have influenced the market valuation. They collected data on 31 byland properties that were sold during the last 10 years. In addition to the transaction price for each property, they collected data for a large number of other factors, including size, time of sale, elevation, location, and access to sewers. A listing of these data, including only those variables deemed relevant for this exercise.

Answer the following questions:

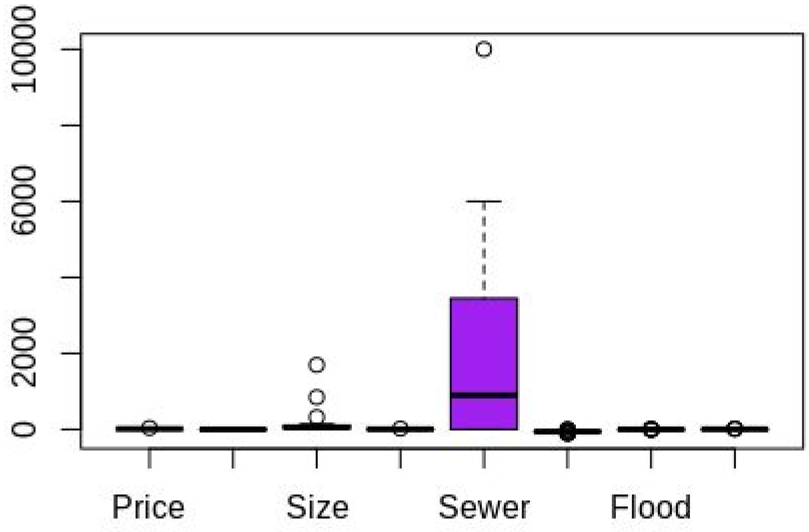
1. What is the nature of each of the variables? Which variable is dependent variable and

what are the independent variables in the model?

2. Check whether the variables require any transformation individually

3. Set up a regression equation, run the model and discuss your results

```
setwd("/home/dell/mne/BABI/Project2/2.1")

rawdata = readxl::read_excel("leslie.xlsx")


dim(rawdata)

## [1] 31 8

str(rawdata)

## Classes 'tbl_df', 'tbl' and 'data.frame': 31 obs. of 8 variables:

## $ Price : num 4.5 10.6 1.7 5 5 3.3 5.7 6.2 19.4 3.2 ...

## $ County : num 1 1 0 0 0 1 1 1 1 1 ...

## $ Size : num 138.4 52 16.1 1695.2 845 ...

## $ Elevation: num 10 4 0 1 1 2 4 4 20 0 ...

## $ Sewer : num 3000 0 2640 3500 1000 10000 0 0 1300 6000 ...

## $ Date : num -103 -103 -98 -93 -92 -86 -68 -64 -63 -62 ...

## $ Flood : num 0 0 1 0 1 0 0 0 0 0 ...

## $ Distance : num 0.3 2.5 10.3 14 14 0 0 0 1.2 0 ...

anyNA(rawdata)

## [1] FALSE

summary(rawdata)

## Price County Size Elevation

## Min. : 1.70 Min. :0.0000 Min. : 6.90 Min. : 0.000

## 1st Qu.: 5.35 1st Qu.:0.0000 1st Qu.: 20.35 1st Qu.: 2.000

## Median :11.70 Median :1.0000 Median : 51.40 Median : 4.000

## Mean :11.95 Mean :0.6129 Mean : 139.97 Mean : 4.645

## 3rd Qu.:16.05 3rd Qu.:1.0000 3rd Qu.: 104.10 3rd Qu.: 7.000
## Max. :37.20 Max. :1.0000 Max. :1695.20 Max. :20.000

## Sewer Date Flood Distance

## Min. : 0 Min. :-103.00 Min. :0.0000 Min. : 0.000

## 1st Qu.: 0 1st Qu.: -63.50 1st Qu.:0.0000 1st Qu.: 0.850

## Median : 900 Median : -59.00 Median :0.0000 Median : 4.900

## Mean : 1981 Mean : -58.65 Mean :0.1613 Mean : 5.132

## 3rd Qu.: 3450 3rd Qu.: -51.00 3rd Qu.:0.0000 3rd Qu.: 5.500

## Max. :10000 Max. : -4.00 Max. :1.0000 Max. :16.500

names(rawdata)

## [1] "Price" "County" "Size" "Elevation" "Sewer" "Date"

## [7] "Flood" "Distance"

boxplot(rawdata, col = rep(c("Red", "Green", "Blue", "Yellow", "Purple" )))
```

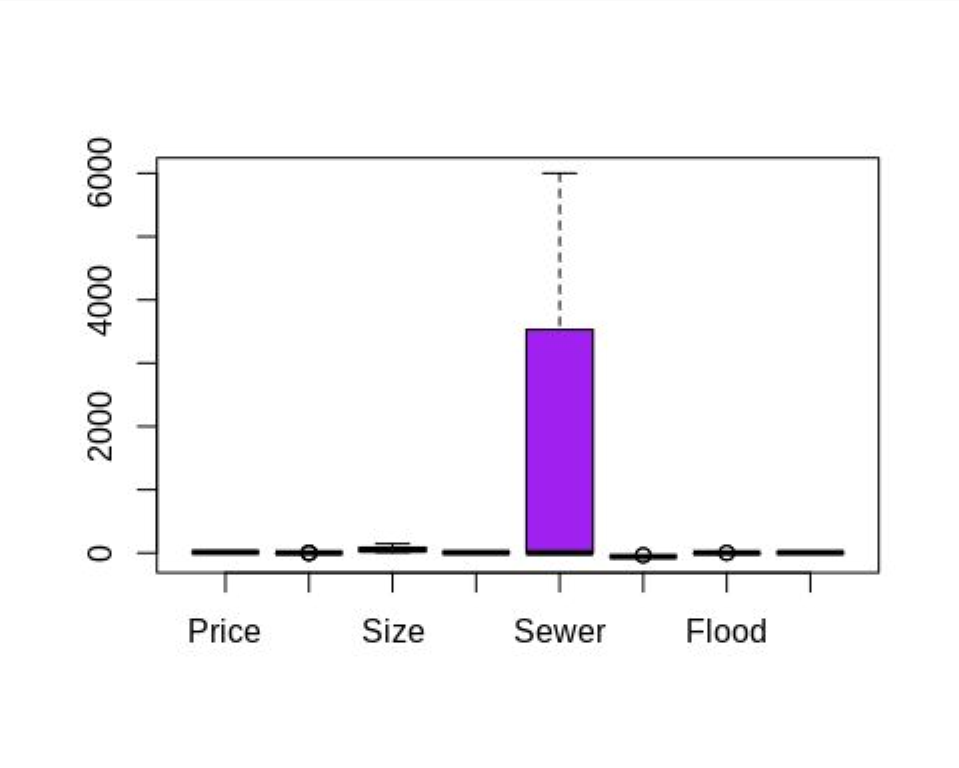


```
price.outliers = boxplot(rawdata$Price)$out

sewer.outliers = boxplot(rawdata$Sewer)$out

size.outliers = boxplot(rawdata$Size)$out

elev.outliers = boxplot(rawdata$Elevation)$out

date.outliers = boxplot(rawdata$Date)$out
```

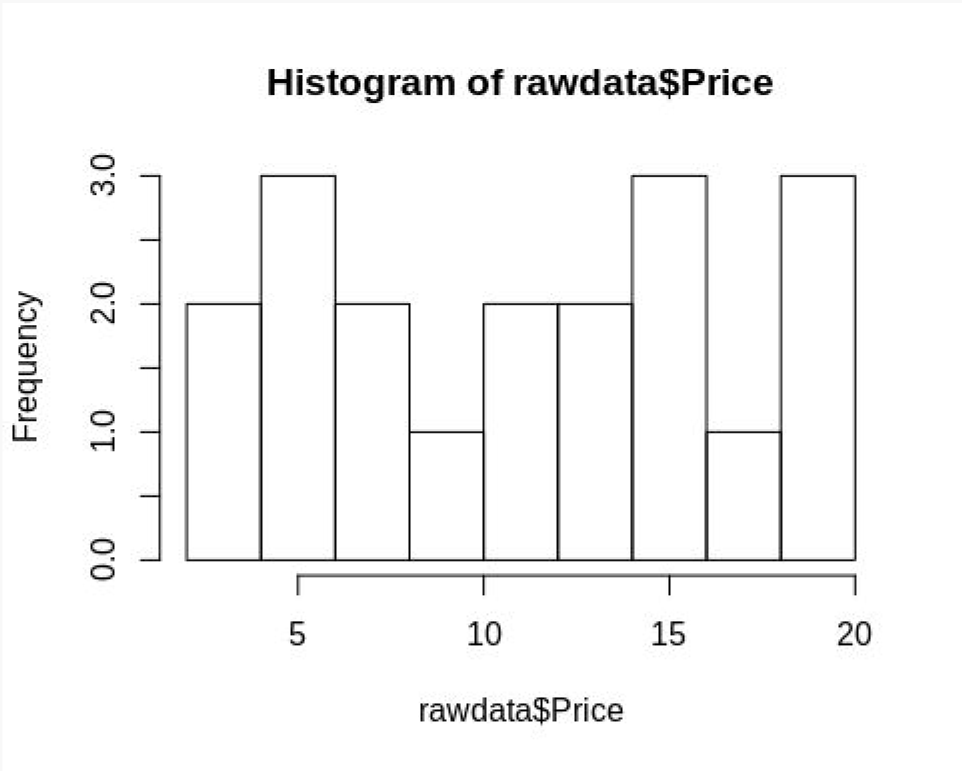
```
rawdata[which(rawdata$Price %in% price.outliers ),]

## # A tibble: 1 x 8

## Price County Size Elevation Sewer Date Flood Distance

## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

## 1 37.2 0 15 5 0 -39 0 7.2

rawdata[which(rawdata$Sewer %in% sewer.outliers ),]

## # A tibble: 1 x 8

## Price County Size Elevation Sewer Date Flood Distance

## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

## 1 3.3 1 6.9 2 10000 -86 0 0

rawdata[which(rawdata$Size %in% size.outliers ),]

## # A tibble: 3 x 8

## Price County Size Elevation Sewer Date Flood Distance

## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

## 1 5 0 1695. 1 3500 -93 0 14

## 2 5 0 845 1 1000 -92 1 14

## 3 12.2 0 321. 0 4000 -54 0 16.5

rawdata[which(rawdata$Elevation %in% elev.outliers ),]
```

```
## # A tibble: 1 x 8

## Price County Size Elevation Sewer Date Flood Distance

## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

## 1 19.4 1 51.4 20 1300 -63 0 1.2

rawdata[which(rawdata$Date %in% date.outliers ),]

## # A tibble: 9 x 8

## Price County Size Elevation Sewer Date Flood Distance

## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

## 1 4.5 1 138. 10 3000 -103 0 0.3

## 2 10.6 1 52 4 0 -103 0 2.5

## 3 1.7 0 16.1 0 2640 -98 1 10.3

## 4 5 0 1695. 1 3500 -93 0 14

## 5 5 0 845 1 1000 -92 1 14

## 6 3.3 1 6.9 2 10000 -86 0 0

## 7 22.9 0 12 5 3400 -16 0 5.5

## 8 15.2 0 67 2 900 -5 1 5.5

## 9 21.9 0 30.8 2 900 -4 0 5.5

rawdata = rawdata[-which(rawdata$Price %in% price.outliers ),]

rawdata = rawdata[-which(rawdata$Sewer %in% sewer.outliers ),]

rawdata = rawdata[-which(rawdata$Size %in% size.outliers ),]

rawdata = rawdata[-which(rawdata$Elevation %in% elev.outliers ),]

rawdata = rawdata[-which(rawdata$Date %in% date.outliers ),]
```

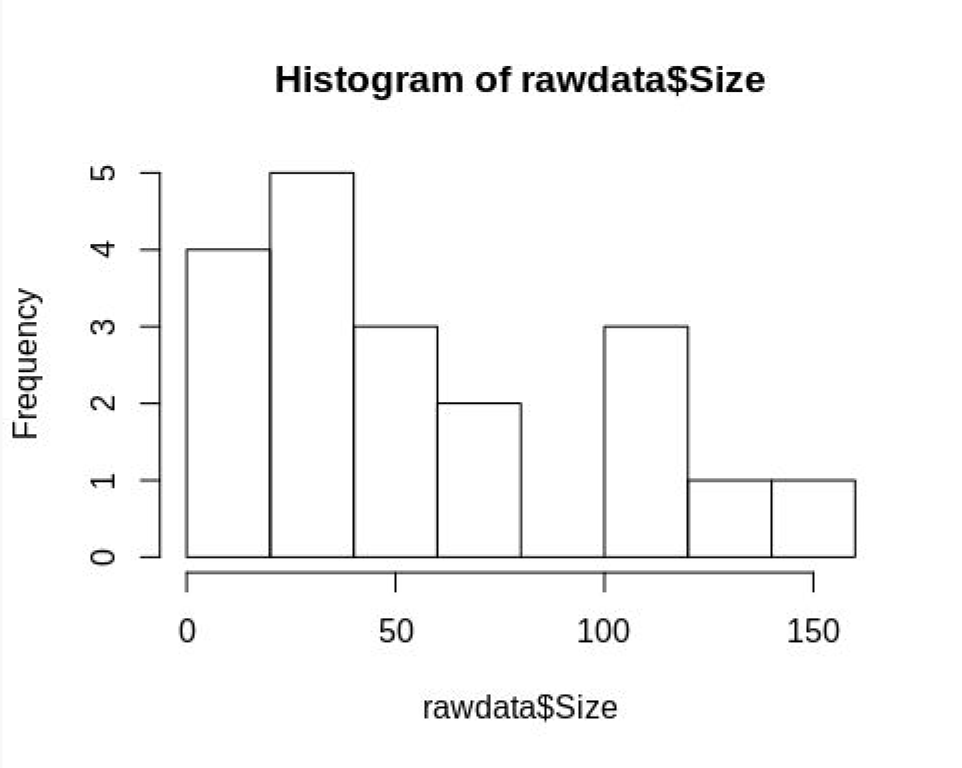
```
boxplot(rawdata, col = rep(c("Red", "Green", "Blue", "Yellow", "Purple" )))
```



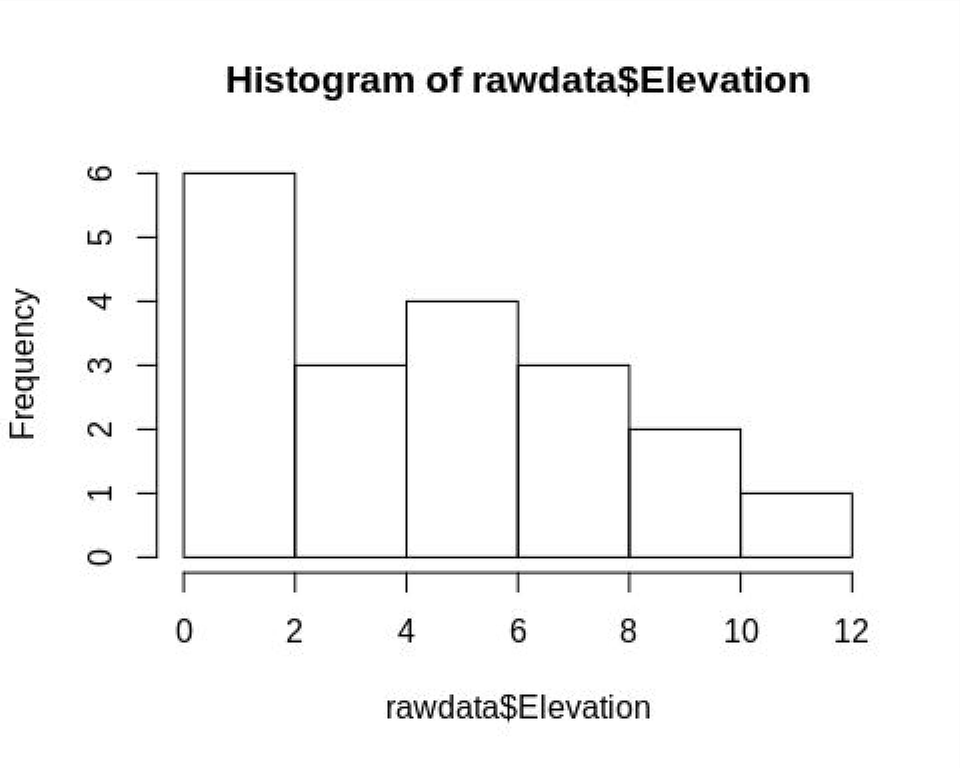
```
hist(rawdata$Price)
```



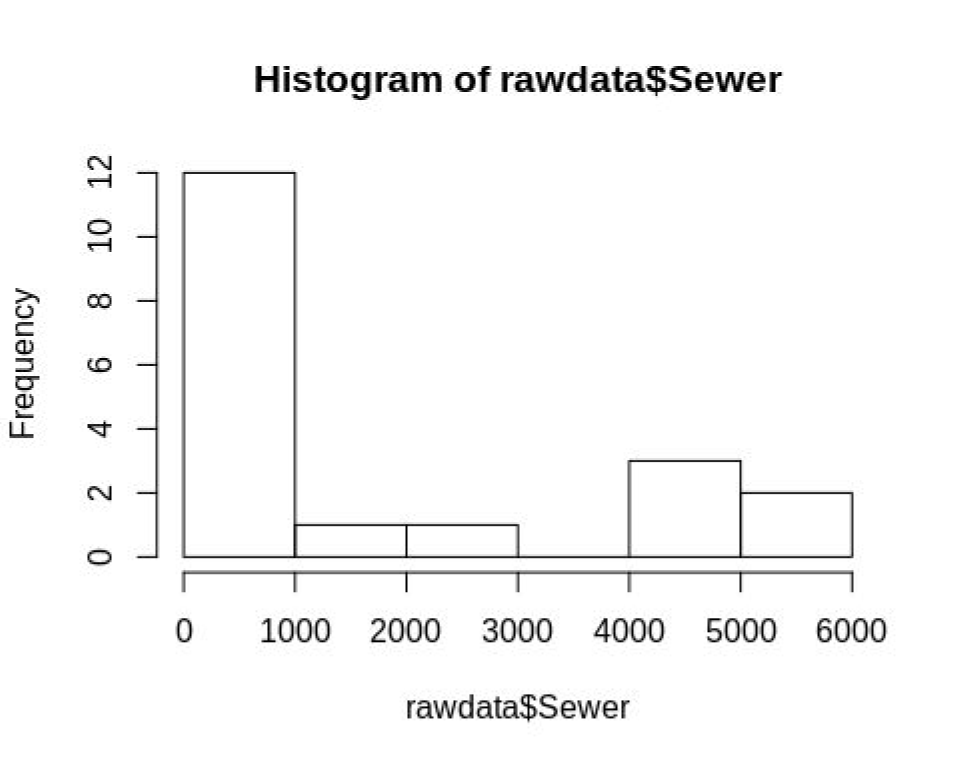
```
hist(rawdata$Size)
```



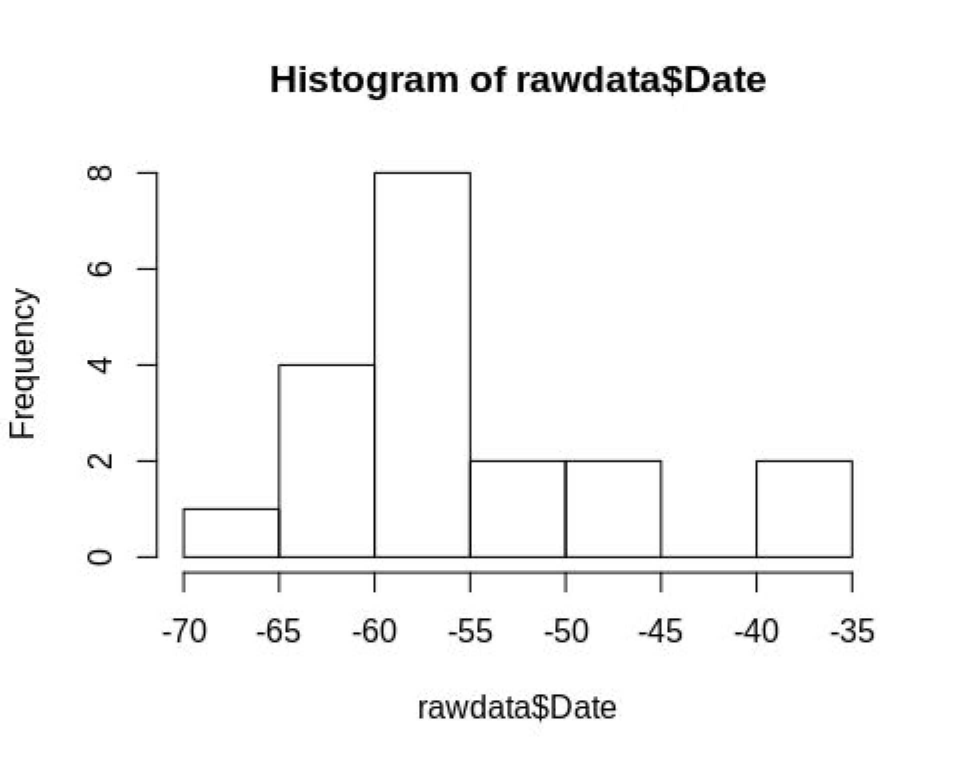
```
hist(rawdata$Elevation)
```



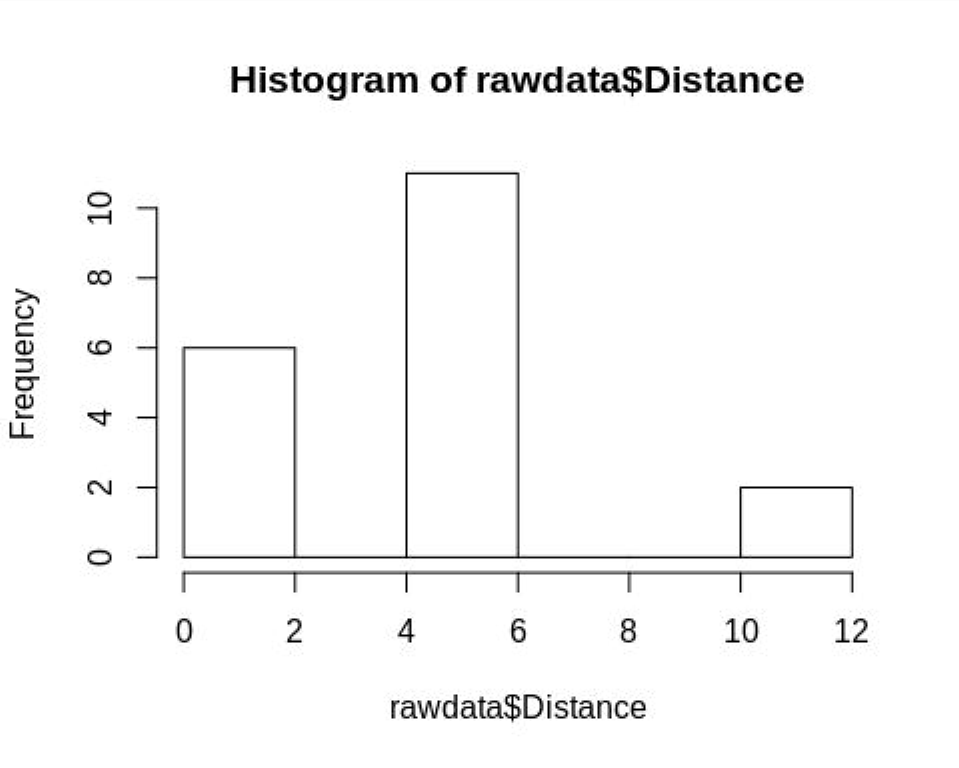
```
hist(rawdata$Sewer) # transform
```



```
hist(rawdata$Date)
```

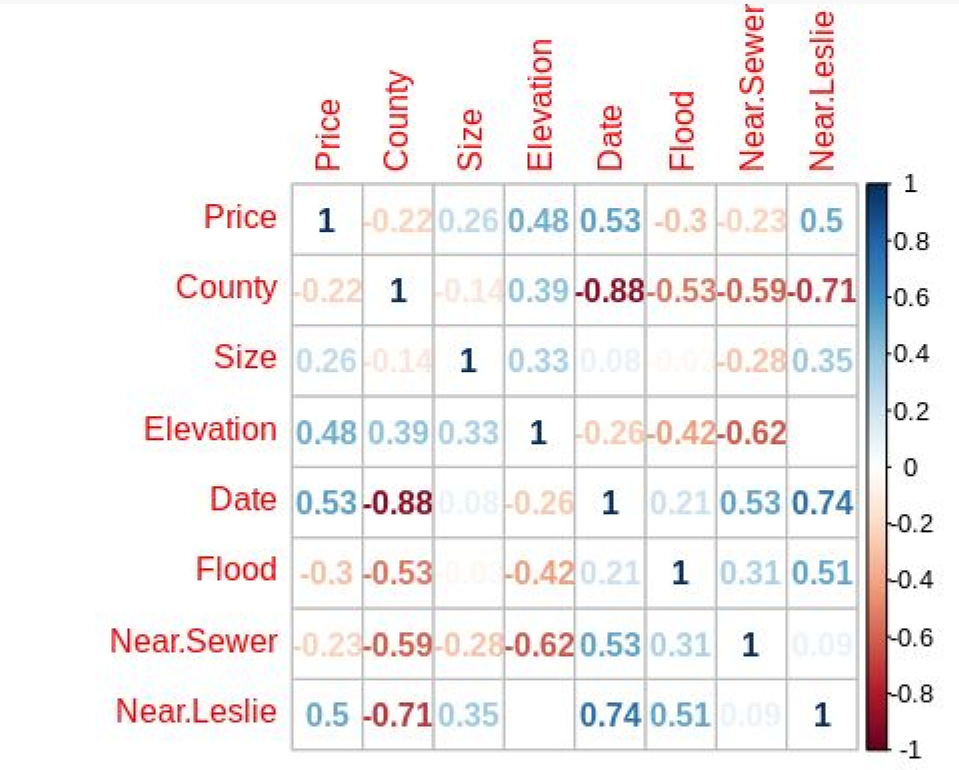


```
hist(rawdata$Distance) # transform
```



```
rawdata$Near.Sewer = ifelse( rawdata$Sewer <= 1000, 0, 1)

rawdata$Near.Sewer

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

rawdata$Near.Leslie = ifelse( rawdata$Distance <= 2, 0, ifelse( rawdata$Distance <= 6, 1, 2) )

rawdata$Near.Leslie

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

install.packages("caTools")

## Error in install.packages : Updating loaded packages

library(caTools)

set.seed(101)

data = rawdata[,-c(5,8)]

sample = sample.split(data$Price, SplitRatio = 0.7)
```

```
pricedata = subset(data, sample == TRUE)

test.pricedata = subset(data, sample == FALSE)



library(corrplot)

cormatrix = cor(pricedata)

corrplot::corrplot(cormatrix, method = "number")
```



	Price	County	Size	Elevation	Date	Flood	Near.Sewer	Near.Leslie
Price	1	-0.22	0.26	0.48	0.53	-0.3	-0.23	0.5
County	-0.22	1	-0.14	0.39	-0.88	-0.53	-0.59	-0.71
Size	0.26	-0.14	1	0.33	0.08	-0.03	-0.28	0.35
Elevation	0.48	0.39	0.33	1	-0.26	-0.42	-0.62	
Date	0.53	-0.88	0.08	-0.26	1	0.21	0.53	0.74
Flood	-0.3	-0.53	-0.03	-0.42	0.21	1	0.31	0.51
Near.Sewer	-0.23	-0.59	-0.28	-0.62	0.53	0.31	1	0.09
Near.Leslie	0.5	-0.71	0.35		0.74	0.51	0.09	1

```
price_lm = lm(Price ~. , data = pricedata)

reg_equation = paste0("Price = ",

round(price_lm$coefficients[1], 2)," + (",

round(price_lm$coefficients[2],2)," * County) + (",

round(price_lm$coefficients[3],2)," * Size) + (",

round(price_lm$coefficients[4],2)," * Elevation) + (",
round(price_lm$coefficients[5],2)," * Date) + (",

round(price_lm$coefficients[6],2)," * Flood) + (",

round(price_lm$coefficients[7],2)," * Near.Sewer) + (",

round(price_lm$coefficients[8],2)," * Near.Leslie)")

print(reg_equation)

## [1] "Price = 80.25 + (15.18 * County) + (0.03 * Size) + (0.74 * Elevation) + (1.42 * Date) + (12.62 *
Flood) + (-6.2 * Near.Sewer) + (-6.28 * Near.Leslie)"

predict(price_lm, newdata = data.frame(County = 0, Size = 246.8, Elevation = 0, Date = 0, Flood = 0,
Near.Sewer = 0, Near.Leslie = 0))

## 1

## 86.50387

# install.packages("car")

library(car)

VIF = vif(price_lm)

VIF

## County Size Elevation Date Flood Near.Sewer

## 22.026348 2.516412 2.184272 47.810958 14.082240 4.735237

## Near.Leslie

## 21.437107

price_lm_2 = lm(Price ~ Size + Elevation + Flood + Near.Sewer + Near.Leslie + Date , data = pricedata)

reg_equation_2 = paste0("Price = ",

round(price_lm_2$coefficients[1], 2)," + (",

round(price_lm_2$coefficients[2],2)," * Size) + (",

round(price_lm_2$coefficients[3],2)," * Elevation) + (",

round(price_lm_2$coefficients[4],2)," * Flood) + (",

round(price_lm_2$coefficients[5],2)," * Near.Sewer) + (",
round(price_lm_2$coefficients[6],2)," * Near.Leslie) + (",

round(price_lm_2$coefficients[7],2)," * Date)")

print(reg_equation_2)

## [1] "Price = 39.31 + (-0.01 * Size) + (0.63 * Elevation) + (-3.26 * Flood) + (-4.29 * Near.Sewer) +
(-0.26 * Near.Leslie) + (0.51 * Date)"

predict(price_lm_2, newdata = data.frame(Size = 246.8, Elevation = 0, Flood = 0, Near.Sewer = 0,
Near.Leslie = 0, Date = 0))

## 1

## 37.44861

VIF_2 = vif(price_lm_2)

VIF_2

## Size Elevation Flood Near.Sewer Near.Leslie Date

## 1.450418 2.120850 4.443278 4.247305 11.976313 10.582194

price_lm_3 = lm(Price ~ Size + Elevation + Flood + Near.Sewer + Near.Leslie , data = pricedata)

reg_equation_3 = paste0("Price = ",

round(price_lm_3$coefficients[1], 2)," + (",

round(price_lm_3$coefficients[2],2)," * Size) + (",

round(price_lm_3$coefficients[3],2)," * Elevation) + (",

round(price_lm_3$coefficients[4],2)," * Flood) + (",

round(price_lm_3$coefficients[5],2)," * Near.Sewer) + (",

round(price_lm_3$coefficients[6],2)," * Near.Leslie)")

print(reg_equation_3)

## [1] "Price = 6 + (-0.02 * Size) + (0.45 * Elevation) + (-11.98 * Flood) + (0.3 * Near.Sewer) + (6.29 *
Near.Leslie)"

predict(price_lm_3, newdata = data.frame(Size = 246.8, Elevation = 0, Flood = 0, Near.Sewer = 0,
Near.Leslie = 0))

## 1
## 1.553898

VIF_3 = vif(price_lm_3)

VIF_3

## Size Elevation Flood Near.Sewer Near.Leslie

## 1.351225 1.946746 1.788705 1.675740 1.715996

summary(price_lm_3)

##

## Call:

## lm(formula = Price ~ Size + Elevation + Flood + Near.Sewer +

## Near.Leslie, data = pricedata)

##

## Residuals:

## Min 1Q Median 3Q Max

## -3.220 -2.091 -0.237 0.000 5.166

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 5.99703 3.16290 1.896 0.0998 .

## Size -0.01800 0.02655 -0.678 0.5195

## Elevation 0.44718 0.42992 1.040 0.3329

## Flood -11.97711 4.91078 -2.439 0.0448 *

## Near.Sewer 0.29713 2.54068 0.117 0.9102

## Near.Leslie 6.28836 1.84002 3.418 0.0112 *

## ---

## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##

## Residual standard error: 3.528 on 7 degrees of freedom

## Multiple R-squared: 0.7272, Adjusted R-squared: 0.5324

## F-statistic: 3.732 on 5 and 7 DF, p-value: 0.05762
```

The predicted price at which the leslie salt land should be sold is : ~1500$ / acre

All Greens Dataset

Objective

Explain the importance of X2, X3, X4, X5, X6 on Annual Net Sales, X1.

The data (X1, X2, X3, X4, X5, X6) are for each franchise store.

X1 = annual net sales/$1000

X2 = number sq. ft./1000

X3 = inventory/$1000

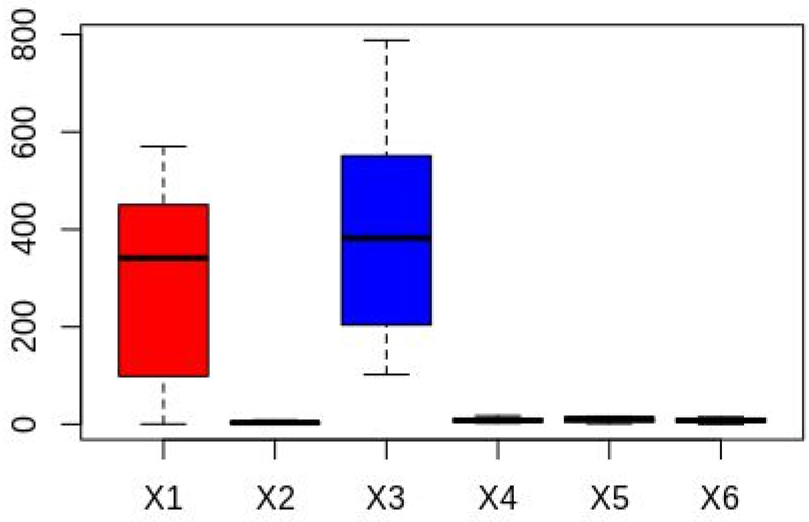
X4 = amount spent on advertising/$1000

X5 = size of sales district/1000 families

X6 = number of competing stores in district

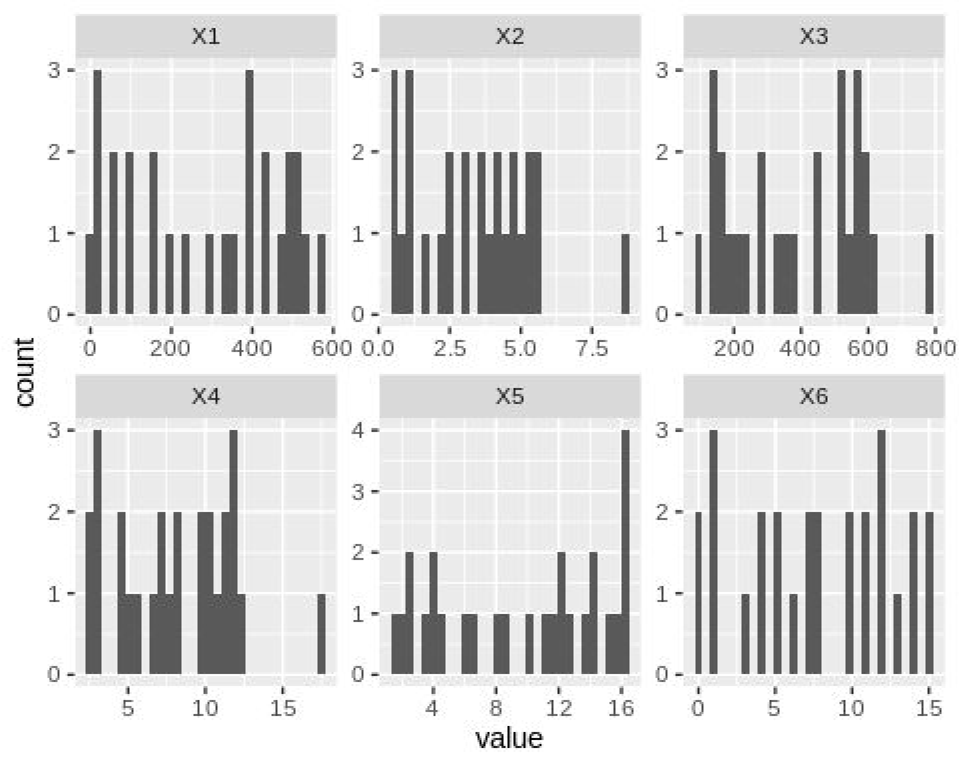
Analysis

```
setwd("/home/dell/mne/BABI/Project2/2.1")

install.packages("tidyverse")

## Installing package into '/home/dell/R/x86_64-pc-linux-gnu-library/3.6'

## (as 'lib' is unspecified)

rawdata = readxl::read_excel("allgreens.xls")

rawdata

## # A tibble: 27 x 6

## X1 X2 X3 X4 X5 X6

## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

## 1 231 3 294 8.20 8.20 11
## 2 156 2.20 232 6.90 4.10 12
## 3 10 0.5 149 3 4.30 15
## 4 519 5.5 600 12 16.1 1
## 5 437 4.40 567 10.6 14.1 5
## 6 487 4.80 571 11.8 12.7 4
## 7 299 3.10 512 8.1 10.1 10
## 8 195 2.5 347 7.70 8.4 12
## 9 20 1.20 212 3.30 2.10 15
## 10 68 0.600 102 4.90 4.70 8
## # … with 17 more rows
dim(rawdata)
## [1] 27 6
str(rawdata)
## Classes 'tbl_df', 'tbl' and 'data.frame': 27 obs. of 6 variables:
## $ X1: num 231 156 10 519 437 487 299 195 20 68 ...
## $ X2: num 3 2.2 0.5 5.5 4.4 ...
## $ X3: num 294 232 149 600 567 571 512 347 212 102 ...
## $ X4: num 8.2 6.9 3 12 10.6 ...
## $ X5: num 8.2 4.1 4.3 16.1 14.1 ...
## $ X6: num 11 12 15 1 5 4 10 12 15 8 ...
anyNA(rawdata)
## [1] FALSE
summary(rawdata)
## X1 X2 X3 X4
## Min. : 0.5 Min. :0.500 Min. :102.0 Min. : 2.50
## 1st Qu.: 98.5 1st Qu.:1.400 1st Qu.:204.0 1st Qu.: 4.80

## Median :341.0 Median :3.500 Median :382.0 Median : 8.10

## Mean :286.6 Mean :3.326 Mean :387.5 Mean : 8.10

## 3rd Qu.:450.5 3rd Qu.:4.750 3rd Qu.:551.0 3rd Qu.:10.95

## Max. :570.0 Max. :8.600 Max. :788.0 Max. :17.40

## X5 X6

## Min. : 1.600 Min. : 0.000

## 1st Qu.: 4.500 1st Qu.: 4.000

## Median :11.300 Median : 8.000

## Mean : 9.693 Mean : 7.741

## 3rd Qu.:14.050 3rd Qu.:12.000

## Max. :16.300 Max. :15.000

names(rawdata)

## [1] "X1" "X2" "X3" "X4" "X5" "X6"

boxplot(rawdata, col = rep(c("Red", "Green", "Blue", "Yellow", "Purple" )))
```



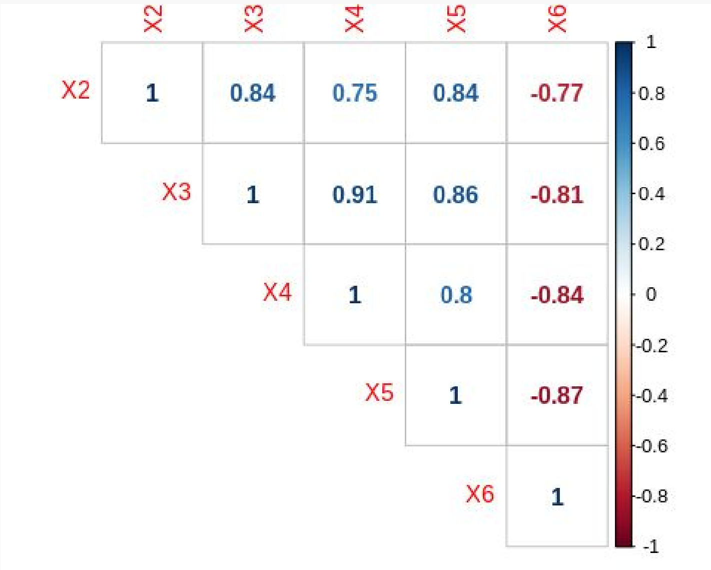
```
install.packages("purrr")

## Installing package into '/home/dell/R/x86_64-pc-linux-gnu-library/3.6'

## (as 'lib' is unspecified)

install.packages("tidyr")

## Installing package into '/home/dell/R/x86_64-pc-linux-gnu-library/3.6'

## (as 'lib' is unspecified)

install.packages("ggplot2")

## Installing package into '/home/dell/R/x86_64-pc-linux-gnu-library/3.6'

## (as 'lib' is unspecified)

library(purrr)

library(tidyr)

library(ggplot2)
```

```
rawdata %>%
gather() %>%
ggplot(aes(value)) +
facet_wrap(~ key, scales = "free") +
geom_histogram()
```



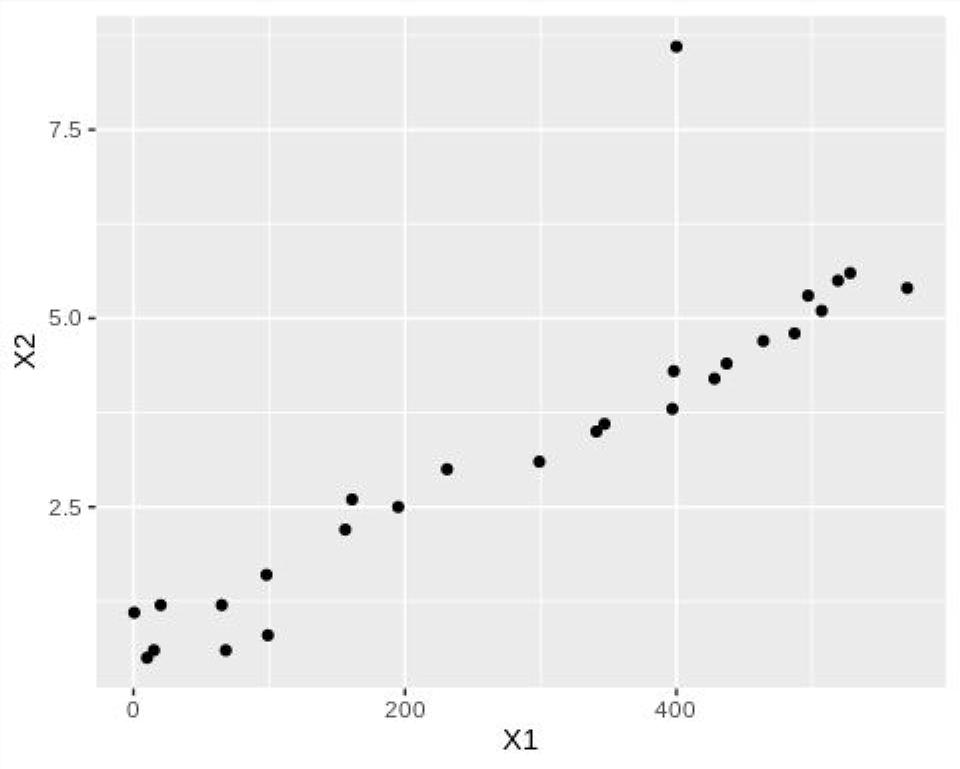
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
cormatrix = cor(rawdata[,2:6])
corrplot::corrplot(cormatrix, type="upper", method="number")
```

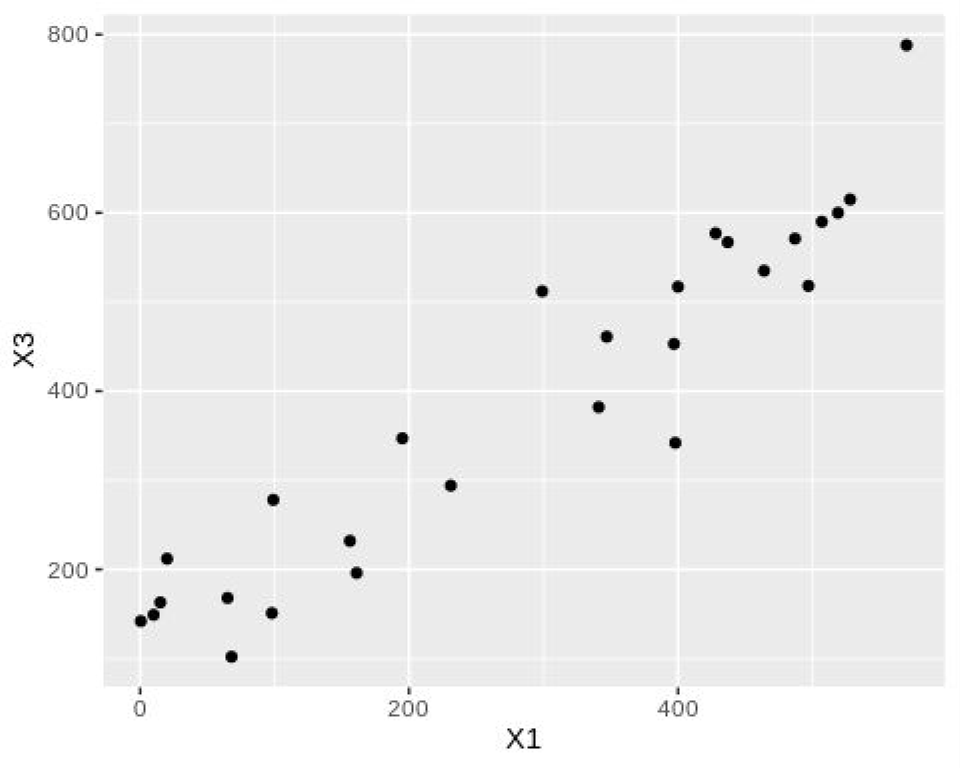


	X2	X3	X4	X5	X6
X2	1	0.84	0.75	0.84	-0.77
X3		1	0.91	0.86	-0.81
X4			1	0.8	-0.84
X5				1	-0.87
X6					1

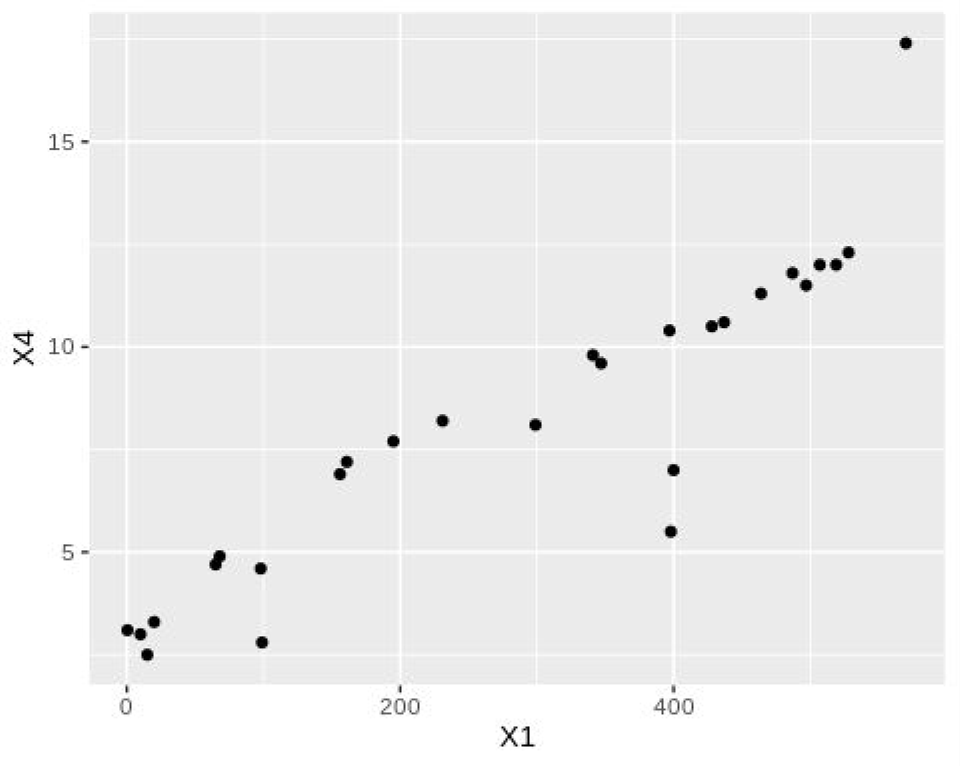
```
ggplot(data = rawdata) + geom_point( mapping = aes(x = X1, y = X2))
```



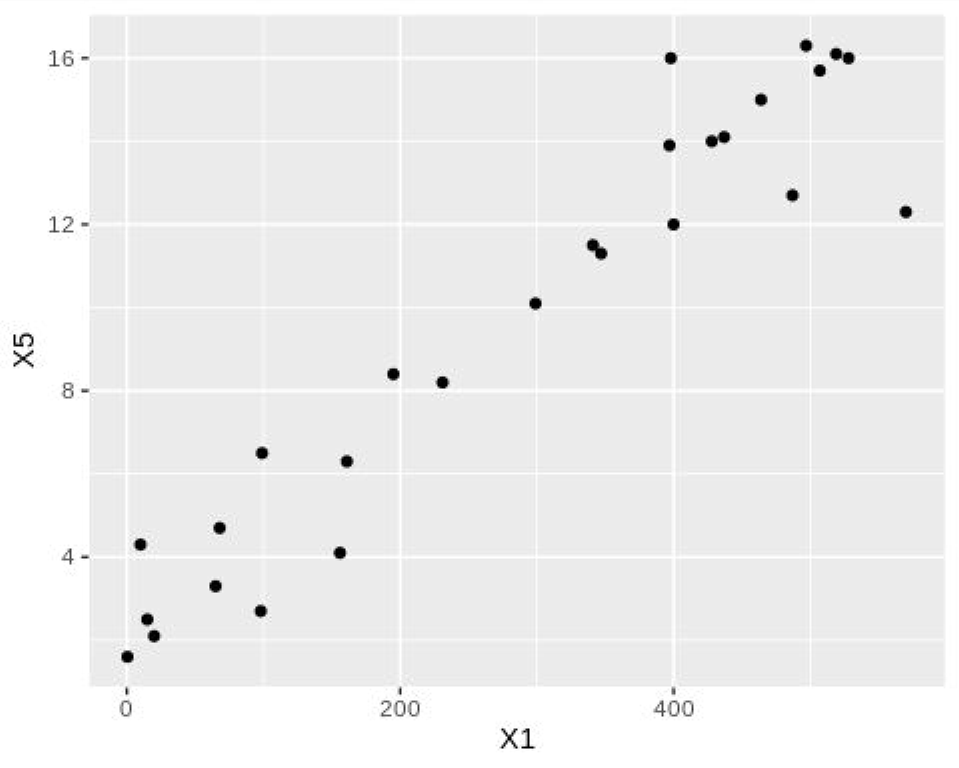
```
ggplot(data = rawdata) + geom_point( mapping = aes(x = X1, y = X3))
```



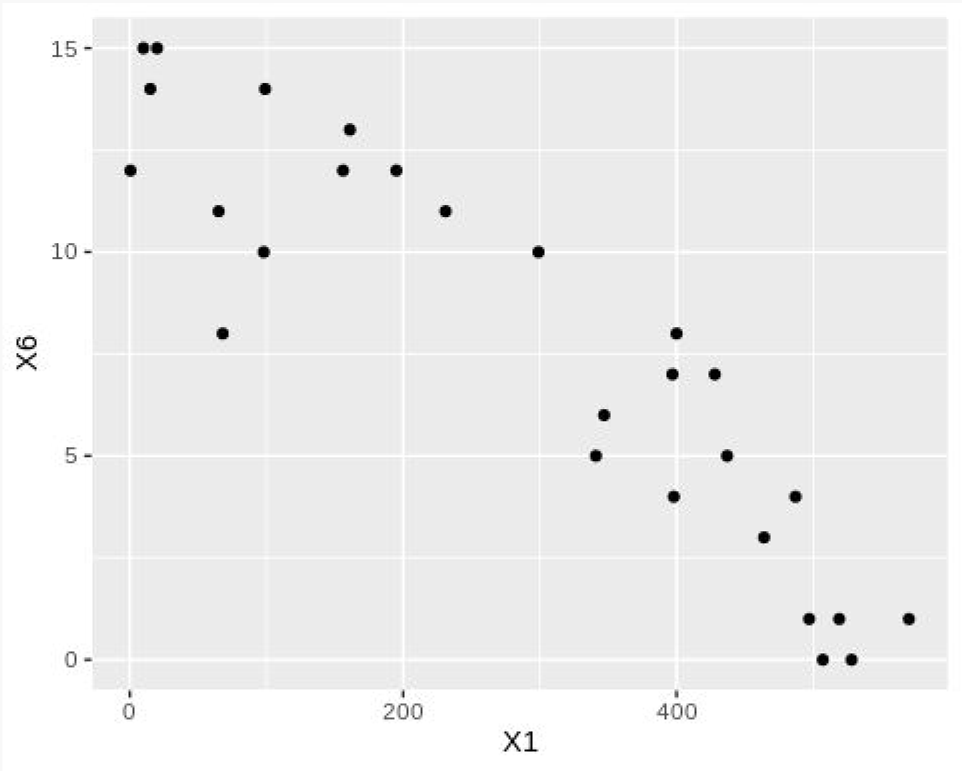
ggplot(data = rawdata) + geom_point(mapping = aes(x = X1, y = X4))



```
ggplot(data = rawdata) + geom_point( mapping = aes(x = X1, y = X5))
```



```
ggplot(data = rawdata) + geom_point( mapping = aes(x = X1, y = X6))
```



X1 is positively correlated with variables X2, X3, X4, X5

X1 is negatively correlated with variables X6

```
lm.X2 = lm(X1 ~ X2, data = rawdata)

lm.X2$coefficients

## (Intercept) X2

## 2.577006 85.388873

lm.X2.eqn = paste0("X1 = ",

round(lm.X2$coefficients[1],3), " + ",

round(lm.X2$coefficients[2],3), " * X2 ")

lm.X2.eqn

## [1] "X1 = 2.577 + 85.389 * X2 "
```

```
lm.X3 = lm(X1 ~ X3, data = rawdata)

lm.X3$coefficients

## (Intercept) X3

## -81.5043523 0.9499252

lm.X3.eqn = paste0("X1 = ",

round(lm.X3$coefficients[1],3), " + ",

round(lm.X3$coefficients[2],3), " * X3 ")

lm.X3.eqn

## [1] "X1 = -81.504 + 0.95 * X3 "

lm.X4 = lm(X1 ~ X4, data = rawdata)

lm.X4$coefficients

## (Intercept) X4

## -90.14962 46.50910

lm.X4.eqn = paste0("X1 = ",

round(lm.X4$coefficients[1],3), " + ",

round(lm.X4$coefficients[2],3), " * X4 ")

lm.X4.eqn

## [1] "X1 = -90.15 + 46.509 * X4 "

lm.X5 = lm(X1 ~ X5, data = rawdata)

lm.X5$coefficients

## (Intercept) X5

## -58.82321 35.63518

lm.X5.eqn = paste0("X1 = ",

round(lm.X5$coefficients[1],3), " + ",

round(lm.X5$coefficients[2],3), " * X5 ")
```

```
lm.X5.eqn

## [1] "X1 = -58.823 + 35.635 * X5 "

lm.X6 = lm(X1 ~ X6, data = rawdata)

lm.X6$coefficients

## (Intercept) X6

## 563.59262 -35.78709

lm.X6.eqn = paste0("X1 = ",

round(lm.X6$coefficients[1],3), " + ",

round(lm.X6$coefficients[2],3), " * X6 ")

lm.X6.eqn

## [1] "X1 = 563.593 + -35.787 * X6 "

lm = lm(X1 ~ ., data = rawdata)

lm$coefficients

## (Intercept) X2 X3 X4 X5 X6

## -18.8594142 16.2015736 0.1746352 11.5262690 13.5803129 -5.3109714

lm.eqn = paste0("X1 = ",

round(lm$coefficients[1],3), " + (",

round(lm$coefficients[2],3), " * X2) + (",

round(lm$coefficients[3],3), " * X3) + (",

round(lm$coefficients[4],3), " * X4) + (",

round(lm$coefficients[5],3), " * X5) + (",

round(lm$coefficients[2],3), " * X6) ")

lm.eqn

## [1] "X1 = -18.859 + (16.202 * X2) + (0.175 * X3) + (11.526 * X4) + (13.58 * X5) + (16.202 * X6) "

predict(lm, newdata = data.frame( X2 = 3, X3 = 294, X4 = 8.2, X5 = 8.2, X6 = 11))
```

```
## 1

## 228.5413
```

The predicted Net Sales when

X2 = 3000 sq. ft. of shop area

X3 =$294000 of inventory

X4 = $8200 spent in advertising

X5 = in a district of 8200 families

X6 = and 11 competing stores

Will give a net sales of X1 = ~ $228000