week3R

Emmanuel Titi

2022-06-11

Loading our data.

```
df=read.csv("http://bit.ly/CarreFourDataset")
summary(df)
```

```
##   Invoice.ID           Branch          Customer.type        Gender
##  Length:1000        Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Product.line         Unit.price       Quantity          Tax
##  Length:1000        Min.   :10.08   Min.   : 1.00   Min.   : 0.5085
##  Class :character   1st Qu.:32.88   1st Qu.: 3.00   1st Qu.: 5.9249
##  Mode  :character   Median :55.23   Median : 5.00   Median :12.0880
##                     Mean   :55.67   Mean   : 5.51   Mean   :15.3794
##                     3rd Qu.:77.94   3rd Qu.: 8.00   3rd Qu.:22.4453
##                     Max.   :99.96   Max.   :10.00   Max.   :49.6500
##      Date               Time             Payment               cogs
##  Length:1000        Length:1000        Length:1000        Min.   : 10.17
##  Class :character   Class :character   Class :character   1st Qu.:118.50
##  Mode  :character   Mode  :character   Mode  :character   Median :241.76
##                                                           Mean   :307.59
##                                                           3rd Qu.:448.90
##                                                           Max.   :993.00
##  gross.margin.percentage  gross.income         Rating          Total
##  Min.   :4.762           Min.   : 0.5085   Min.   : 4.000   Min.   :  10.68
##  1st Qu.:4.762           1st Qu.: 5.9249   1st Qu.: 5.500   1st Qu.: 124.42
##  Median :4.762           Median :12.0880   Median : 7.000   Median : 253.85
##  Mean   :4.762           Mean   :15.3794   Mean   : 6.973   Mean   : 322.97
##  3rd Qu.:4.762           3rd Qu.:22.4453   3rd Qu.: 8.500   3rd Qu.: 471.35
##  Max.   :4.762           Max.   :49.6500   Max.   :10.000   Max.   :1042.65
```

Removing features not required

```
df1<-dplyr::select(df,-c("Invoice.ID","Date","Time"))
```

Onehot-encoding out categorical variables.

```r
#dummify the data
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
df_dummy<-dummyVars("~.",data=df1)
df2<-data.frame(predict(df_dummy,newdat=df1))
head(df2)
```

```
##   BranchA BranchB BranchC Customer.typeMember Customer.typeNormal GenderFemale
## 1       1       0       0                   1                   0            1
## 2       0       0       1                   0                   1            1
## 3       1       0       0                   0                   1            0
## 4       1       0       0                   1                   0            0
## 5       1       0       0                   0                   1            0
## 6       0       0       1                   0                   1            0
##   GenderMale Product.lineElectronic.accessories Product.lineFashion.accessories
## 1          0                                  0                               0
## 2          0                                  1                               0
## 3          1                                  0                               0
## 4          1                                  0                               0
## 5          1                                  0                               0
## 6          1                                  1                               0
##   Product.lineFood.and.beverages Product.lineHealth.and.beauty
## 1                              0                             1
## 2                              0                             0
## 3                              0                             0
## 4                              0                             1
## 5                              0                             0
## 6                              0                             0
##   Product.lineHome.and.lifestyle Product.lineSports.and.travel Unit.price
## 1                              0                             0      74.69
## 2                              0                             0      15.28
## 3                              1                             0      46.33
## 4                              0                             0      58.22
## 5                              0                             1      86.31
## 6                              0                             0      85.39
##   Quantity     Tax PaymentCash PaymentCredit.card PaymentEwallet   cogs
## 1        7 26.1415           0                  0              1 522.83
## 2        5  3.8200           1                  0              0  76.40
## 3        7 16.2155           0                  1              0 324.31
## 4        8 23.2880           0                  0              1 465.76
## 5        7 30.2085           0                  0              1 604.17
## 6        7 29.8865           0                  0              1 597.73
##   gross.margin.percentage gross.income Rating    Total
## 1                4.761905      26.1415    9.1 548.9715
## 2                4.761905       3.8200    9.6  80.2200
## 3                4.761905      16.2155    7.4 340.5255
## 4                4.761905      23.2880    8.4 489.0480
## 5                4.761905      30.2085    5.3 634.3785
## 6                4.761905      29.8865    4.1 627.6165
```

We need to seperate the encoded data and the original numerical data so as tho scale without getting an error

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df_scale<-df2%>%select( Total,Rating, gross.income,,cogs
                          ,Quantity,Tax,Unit.price)
df_encod<-df2%>%select(-Total,-Rating, -gross.income,
                       -gross.margin.percentage,-cogs
                          ,-Quantity,-Tax,-Unit.price)
```

scaling our data

```
library(tibble)
scaled<-scale(as.data.frame(df_scale,center = TRUE))
enframe(scaled)
```

```
## # A tibble: 1,000 x 2
##    name  value[,"Total"] [,"Rating"] [,"gross.income"] [,"cogs"] [,"Quantity"]
##    <chr>           <dbl>       <dbl>             <dbl>     <dbl>         <dbl>
##  1 1              0.919        1.24              0.919     0.919         0.510
##  2 2             -0.987        1.53             -0.987    -0.987        -0.174
##  3 3              0.0714       0.249             0.0714    0.0714        0.510
##  4 4              0.675        0.831             0.675     0.675         0.852
##  5 5              1.27        -0.973             1.27      1.27          0.510
##  6 6              1.24        -1.67              1.24      1.24          0.510
##  7 7              0.450       -0.682             0.450     0.450         0.168
##  8 8              1.83         0.598             1.83      1.83          1.54
##  9 9             -1.00         0.132            -1.00     -1.00         -1.20
## 10 10            -0.611       -0.624            -0.611    -0.611        -0.859
## # ... with 990 more rows, and 1 more variable: value[6:7] <dbl>
```

```
head(scaled)
```

```
##         Total     Rating gross.income        cogs   Quantity         Tax
## 1  0.91914693  1.2378240   0.91914693  0.91914693  0.5096752  0.91914693
## 2 -0.98723557  1.5287619  -0.98723557 -0.98723557 -0.1744526 -0.98723557
## 3  0.07141032  0.2486355   0.07141032  0.07141032  0.5096752  0.07141032
## 4  0.67544187  0.8305111   0.67544187  0.67544187  0.8517391  0.67544187
## 5  1.26649176 -0.9733034   1.26649176  1.26649176  0.5096752  1.26649176
```

```
## 6  1.23899114 -1.6715541   1.23899114  1.23899114  0.5096752  1.23899114
##    Unit.price
## 1  0.71780097
## 2 -1.52454035
## 3 -0.35260468
## 4  0.09616553
## 5  1.15638044
## 6  1.12165642
```

Lets now join the two the dataframes

```
#concatinate the scaled and dfencode
newdf<-cbind(scaled,df_encod)
head(newdf)
```

```
##         Total     Rating gross.income        cogs   Quantity         Tax
## 1  0.91914693  1.2378240   0.91914693  0.91914693  0.5096752  0.91914693
## 2 -0.98723557  1.5287619  -0.98723557 -0.98723557 -0.1744526 -0.98723557
## 3  0.07141032  0.2486355   0.07141032  0.07141032  0.5096752  0.07141032
## 4  0.67544187  0.8305111   0.67544187  0.67544187  0.8517391  0.67544187
## 5  1.26649176 -0.9733034   1.26649176  1.26649176  0.5096752  1.26649176
## 6  1.23899114 -1.6715541   1.23899114  1.23899114  0.5096752  1.23899114
##    Unit.price BranchA BranchB BranchC Customer.typeMember Customer.typeNormal
## 1  0.71780097       1       0       0                   1                   0
## 2 -1.52454035       0       0       1                   0                   1
## 3 -0.35260468       1       0       0                   0                   1
## 4  0.09616553       1       0       0                   1                   0
## 5  1.15638044       1       0       0                   0                   1
## 6  1.12165642       0       0       1                   0                   1
##   GenderFemale GenderMale Product.lineElectronic.accessories
## 1            1          0                                  0
## 2            1          0                                  1
## 3            0          1                                  0
## 4            0          1                                  0
## 5            0          1                                  0
## 6            0          1                                  1
##   Product.lineFashion.accessories Product.lineFood.and.beverages
## 1                               0                              0
## 2                               0                              0
## 3                               0                              0
## 4                               0                              0
## 5                               0                              0
## 6                               0                              0
##   Product.lineHealth.and.beauty Product.lineHome.and.lifestyle
## 1                             1                              0
## 2                             0                              0
## 3                             0                              1
## 4                             1                              0
## 5                             0                              0
## 6                             0                              0
##   Product.lineSports.and.travel PaymentCash PaymentCredit.card PaymentEwallet
## 1                             0           0                  0              1
## 2                             0           1                  0              0
## 3                             0           0                  1              0
```

```
## 4                                 0           0                 0            1
## 5                                 1           0                 0            1
## 6                                 0           0                 0            1
```

Lets run our pca function

```
#PCA
dfPCA<-prcomp(newdf)
#head(dfPCA)
```

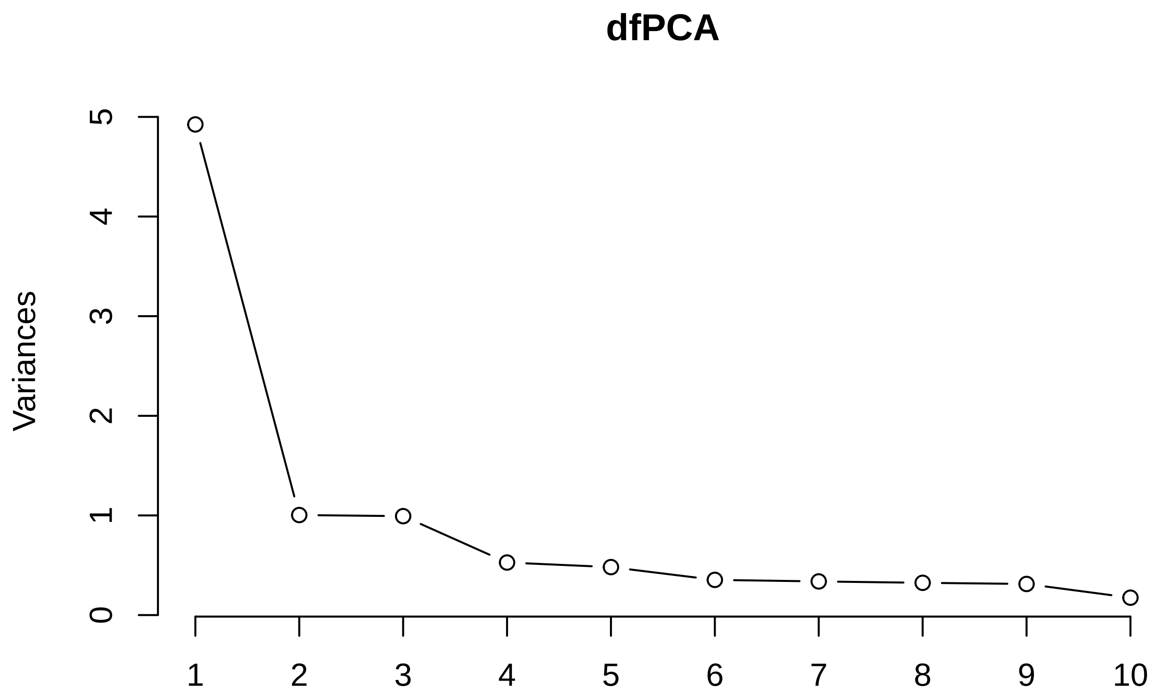
Lets make a short descreptive analysis of the PCs

```
summary(dfPCA)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.2191 1.00198 0.99639 0.72621 0.69392 0.59438 0.58110
## Proportion of Variance 0.4843 0.09873 0.09763 0.05186 0.04735 0.03474 0.03321
## Cumulative Proportion  0.4843 0.58300 0.68064 0.73250 0.77985 0.81460 0.84780
##                            PC8     PC9    PC10    PC11    PC12    PC13   PC14
## Standard deviation     0.56903 0.55868 0.41794 0.41052 0.40776 0.40066 0.3905
## Proportion of Variance 0.03184 0.03069 0.01718 0.01657 0.01635 0.01579 0.0150
## Cumulative Proportion  0.87965 0.91034 0.92752 0.94409 0.96044 0.97623 0.9912
##                           PC15      PC16     PC17      PC18      PC19      PC20
## Standard deviation     0.29863 4.594e-16 4.12e-16 2.817e-16 2.471e-16 1.986e-16
## Proportion of Variance 0.00877 0.000e+00 0.00e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.00000 1.000e+00 1.00e+00 1.000e+00 1.000e+00 1.000e+00
##                             PC21      PC22      PC23
## Standard deviation     1.683e-16 1.001e-16 3.444e-17
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00
```

We can start to visualize our PCs using the plot function

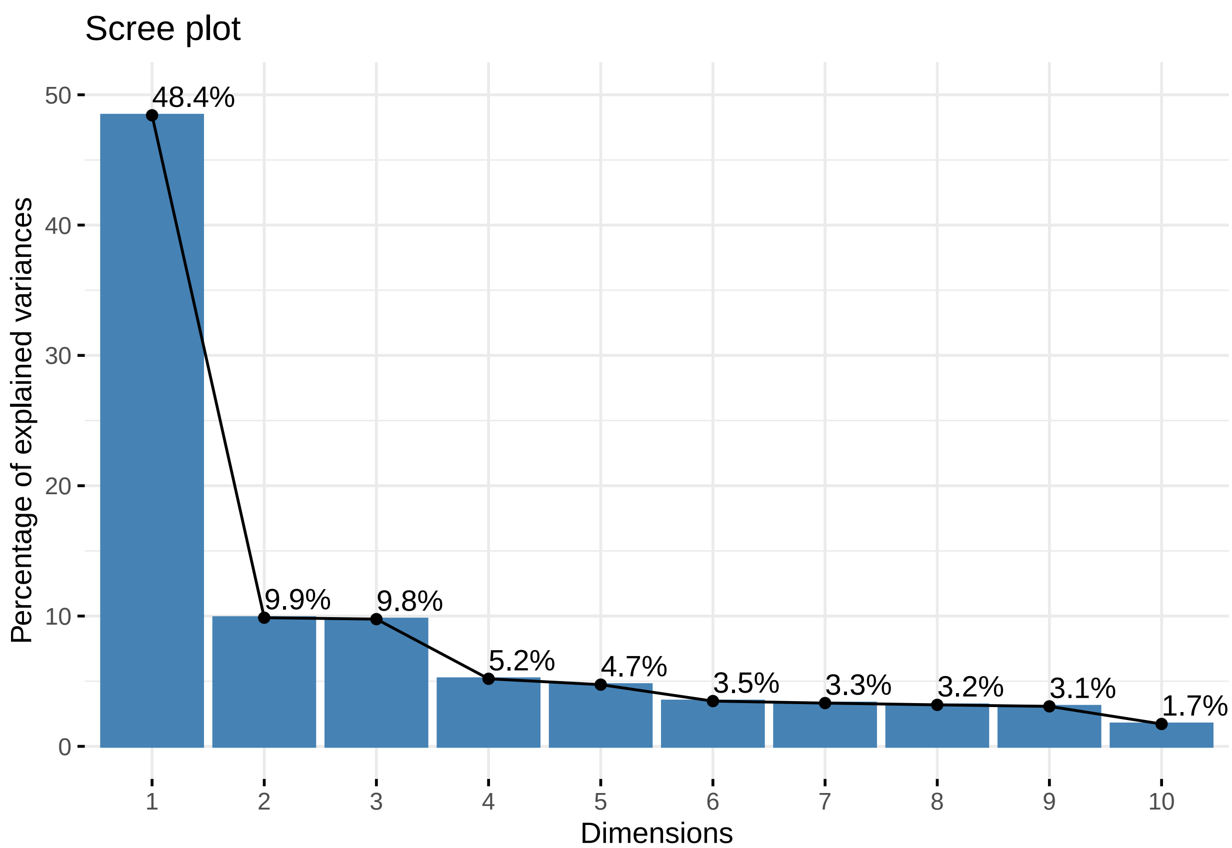
```
plot(dfPCA,type="l")
```




```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_eig(dfPCA ,addlabels = TRUE,ylim=c(0,50))
```

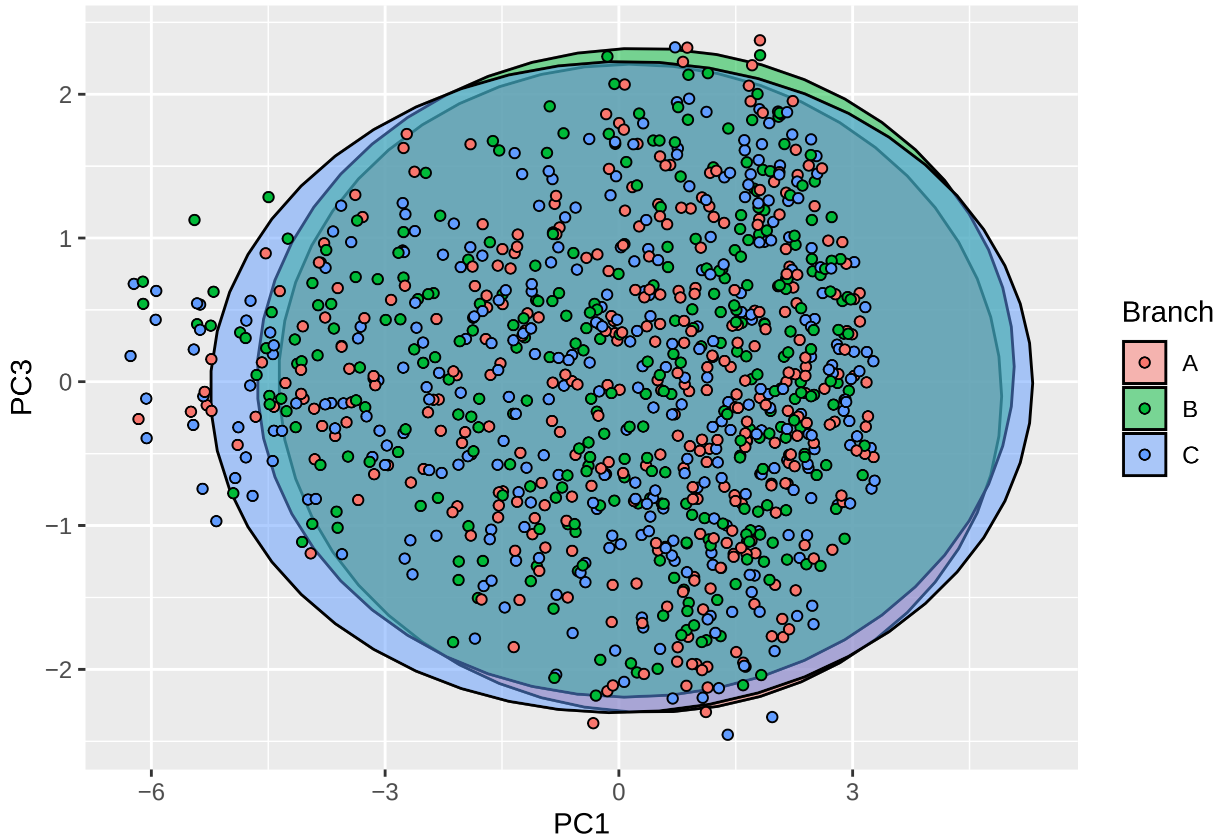


The scree plot shows percentage of variance contributed by each principal component.We migh want to stop at 6 since already 81% of variance is represented.

```
df3<-cbind(df,(dfPCA$x[,1:6]))
```

Here we have now combined our original data set to the 1st 6 components

```
library(ggplot2)
ggplot(df3,aes(PC1,PC3,col=Branch,fill=Branch))+
  stat_ellipse(geom="polygon",col="black",alpha=0.5)+
  geom_point(shape=21,col="black")
```



The plot shows the relationship Between PC1 and 2 separated/highlighted branch-wise.

Graph of variables

Here we use the get_pca_var()function that provides a list if matrices containing all results for active variables.

```
var_df<-get_pca_var(dfPCA)
var_df
```

```
## Principal Component Analysis Results for variables
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates for the variables"
## 2 "$cor"     "Correlations between variables and dimensions"
## 3 "$cos2"    "Cos2 for the variables"
## 4 "$contrib" "contributions of the variables"
```

We can now visualize variables and make conclusions

Correlation circle

The correlation between variables and principal component is used as the coordinates of the variable on the PC

```
#coordinates of variables
head(var_df$coord,6)
```

```
##                   Dim.1        Dim.2        Dim.3        Dim.4         Dim.5
## Total        -0.99782094  0.004441866 -0.001602519  0.006105305 -0.0009938612
## Rating        0.04147205  0.903838165 -0.424135766 -0.010112460 -0.0114491551
## gross.income -0.99782094  0.004441866 -0.001602519  0.006105305 -0.0009938612
## cogs         -0.99782094  0.004441866 -0.001602519  0.006105305 -0.0009938612
## Quantity     -0.72024050 -0.267716439 -0.605532437  0.031880602 -0.0339532773
## Tax          -0.99782094  0.004441866 -0.001602519  0.006105305 -0.0009938612
##                     Dim.6        Dim.7        Dim.8        Dim.9       Dim.10
## Total         0.001180199 -0.002108020  0.001165783 -0.005950078 -0.001750018
## Rating        0.005177509 -0.023970262  0.004548973  0.022249491 -0.001087060
## gross.income  0.001180199 -0.002108020  0.001165783 -0.005950078 -0.001750018
## cogs          0.001180199 -0.002108020  0.001165783 -0.005950078 -0.001750018
## Quantity     -0.008022062  0.005916874 -0.003612045  0.024485248 -0.005213397
## Tax           0.001180199 -0.002108020  0.001165783 -0.005950078 -0.001750018
##                    Dim.11        Dim.12       Dim.13       Dim.14       Dim.15
## Total        -0.001496901 -0.0001471787  0.002459825 -0.001946532 -0.065077266
## Rating        0.006446994  0.0017339896  0.005330317 -0.003339676 -0.005451258
## gross.income -0.001496901 -0.0001471787  0.002459825 -0.001946532 -0.065077266
## cogs         -0.001496901 -0.0001471787  0.002459825 -0.001946532 -0.065077266
## Quantity     -0.006063208  0.0005932810 -0.014573120  0.004593227  0.199331606
## Tax          -0.001496901 -0.0001471787  0.002459825 -0.001946532 -0.065077266
##                     Dim.16        Dim.17        Dim.18        Dim.19
## Total         1.151192e-17 -1.472107e-16 -2.859765e-17 -1.219424e-17
## Rating       -9.563348e-33  5.717951e-32  0.000000e+00  2.571745e-33
## gross.income -2.828127e-17  2.637773e-16  3.739402e-17 -3.174842e-17
## cogs          1.548519e-17 -2.294141e-16 -2.631499e-17  2.708411e-17
## Quantity     -6.534954e-32  1.029231e-31  2.443142e-32 -8.572485e-34
## Tax           1.284155e-18  1.128475e-16  1.751861e-17  1.685855e-17
##                     Dim.20        Dim.21        Dim.22        Dim.23
## Total         8.692806e-18 -8.429379e-18  7.634883e-17 -5.211008e-18
## Rating        3.444621e-33 -4.670443e-33  1.667131e-32  4.780040e-33
## gross.income  2.750408e-20 -9.840235e-18 -1.260040e-17 -1.832186e-17
## cogs          6.404734e-18  2.768578e-17 -6.120422e-17 -4.383107e-18
## Quantity     -8.267092e-33 -2.130890e-32  8.335654e-33 -1.027709e-32
## Tax          -1.512504e-17 -9.416165e-18 -2.544203e-18  2.791598e-17
```

Results above shows how the variables correlate to the PCs.

Quality of representation

Quality of representation on a factor map is **cos2** (squared cosine ,squared coordinates)

```
head(var_df$cos2,6)
```

```
##                   Dim.1        Dim.2        Dim.3        Dim.4        Dim.5
## Total        0.995646625 1.973017e-05 2.568066e-06 3.727475e-05 9.877601e-07
## Rating       0.001719931 8.169234e-01 1.798911e-01 1.022619e-04 1.310832e-04
## gross.income 0.995646625 1.973017e-05 2.568066e-06 3.727475e-05 9.877601e-07
## cogs         0.995646625 1.973017e-05 2.568066e-06 3.727475e-05 9.877601e-07
## Quantity     0.518746385 7.167209e-02 3.666695e-01 1.016373e-03 1.152825e-03
## Tax          0.995646625 1.973017e-05 2.568066e-06 3.727475e-05 9.877601e-07
##                     Dim.6        Dim.7        Dim.8        Dim.9       Dim.10
## Total        1.392869e-06 4.443749e-06 1.359051e-06 3.540343e-05 3.062563e-06
## Rating       2.680660e-05 5.745735e-04 2.069315e-05 4.950398e-04 1.181699e-06
```

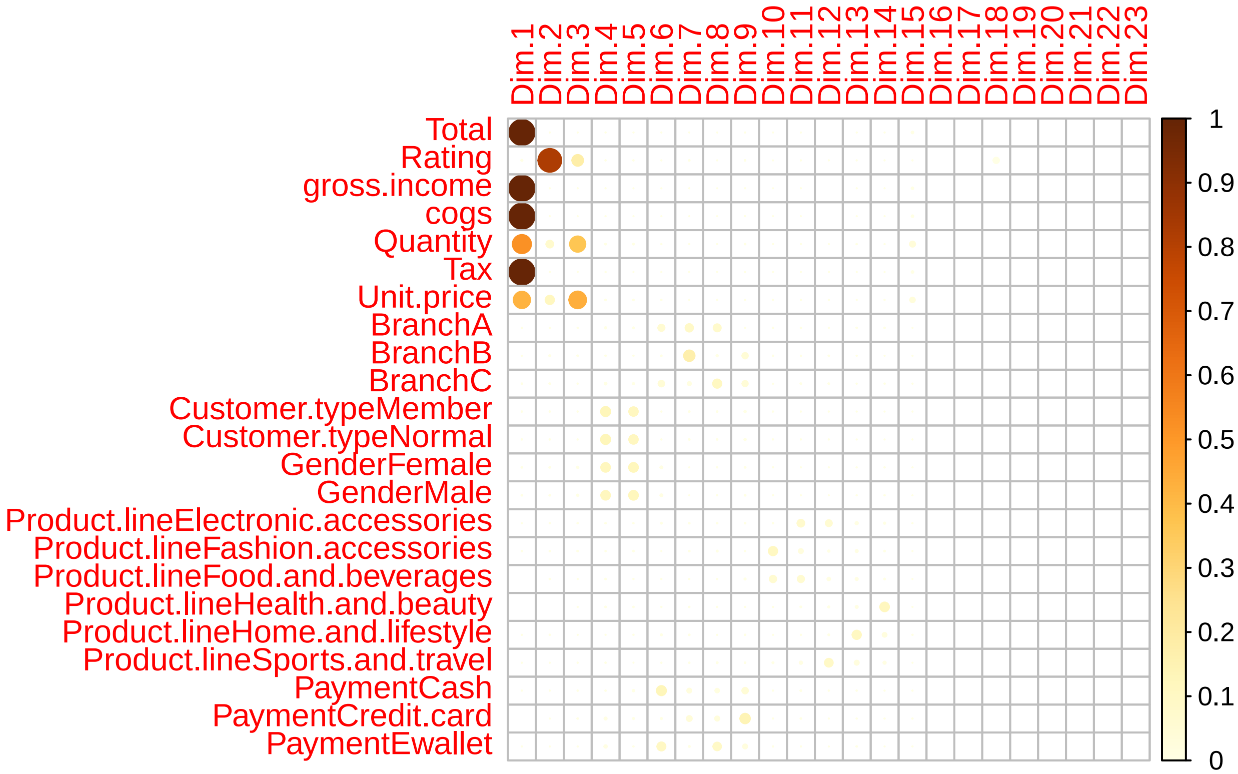
```
## gross.income 1.392869e-06 4.443749e-06 1.359051e-06 3.540343e-05 3.062563e-06
## cogs         1.392869e-06 4.443749e-06 1.359051e-06 3.540343e-05 3.062563e-06
## Quantity     6.435348e-05 3.500940e-05 1.304687e-05 5.995274e-04 2.717950e-05
## Tax          1.392869e-06 4.443749e-06 1.359051e-06 3.540343e-05 3.062563e-06
##                    Dim.11       Dim.12       Dim.13       Dim.14       Dim.15
## Total        2.240713e-06 2.166157e-08 6.050738e-06 3.788987e-06 4.235051e-03
## Rating       4.156373e-05 3.006720e-06 2.841228e-05 1.115344e-05 2.971621e-05
## gross.income 2.240713e-06 2.166157e-08 6.050738e-06 3.788987e-06 4.235051e-03
## cogs         2.240713e-06 2.166157e-08 6.050738e-06 3.788987e-06 4.235051e-03
## Quantity     3.676250e-05 3.519823e-07 2.123758e-04 2.109773e-05 3.973309e-02
## Tax          2.240713e-06 2.166157e-08 6.050738e-06 3.788987e-06 4.235051e-03
##                    Dim.16       Dim.17       Dim.18       Dim.19       Dim.20
## Total        1.325244e-34 2.167099e-32 8.178254e-34 1.486995e-34 7.556488e-35
## Rating       9.145762e-65 3.269496e-63 0.000000e+00 6.613874e-66 1.186542e-65
## gross.income 7.998303e-34 6.957846e-32 1.398313e-33 1.007962e-33 7.564745e-40
## cogs         2.397912e-34 5.263082e-32 6.924785e-34 7.335488e-34 4.102062e-35
## Quantity     4.270563e-63 1.059317e-62 5.968942e-64 7.348749e-67 6.834480e-65
## Tax          1.649054e-36 1.273455e-32 3.069018e-34 2.842109e-34 2.287670e-34
##                    Dim.21       Dim.22       Dim.23
## Total        7.105442e-35 5.829143e-33 2.715461e-35
## Rating       2.181304e-65 2.779325e-64 2.284879e-65
## gross.income 9.683022e-35 1.587701e-34 3.356906e-34
## cogs         7.665023e-34 3.745957e-33 1.921163e-35
## Quantity     4.540691e-64 6.948312e-65 1.056185e-64
## Tax          8.866416e-35 6.472970e-36 7.793018e-34
```

lets visualize our results on cos2

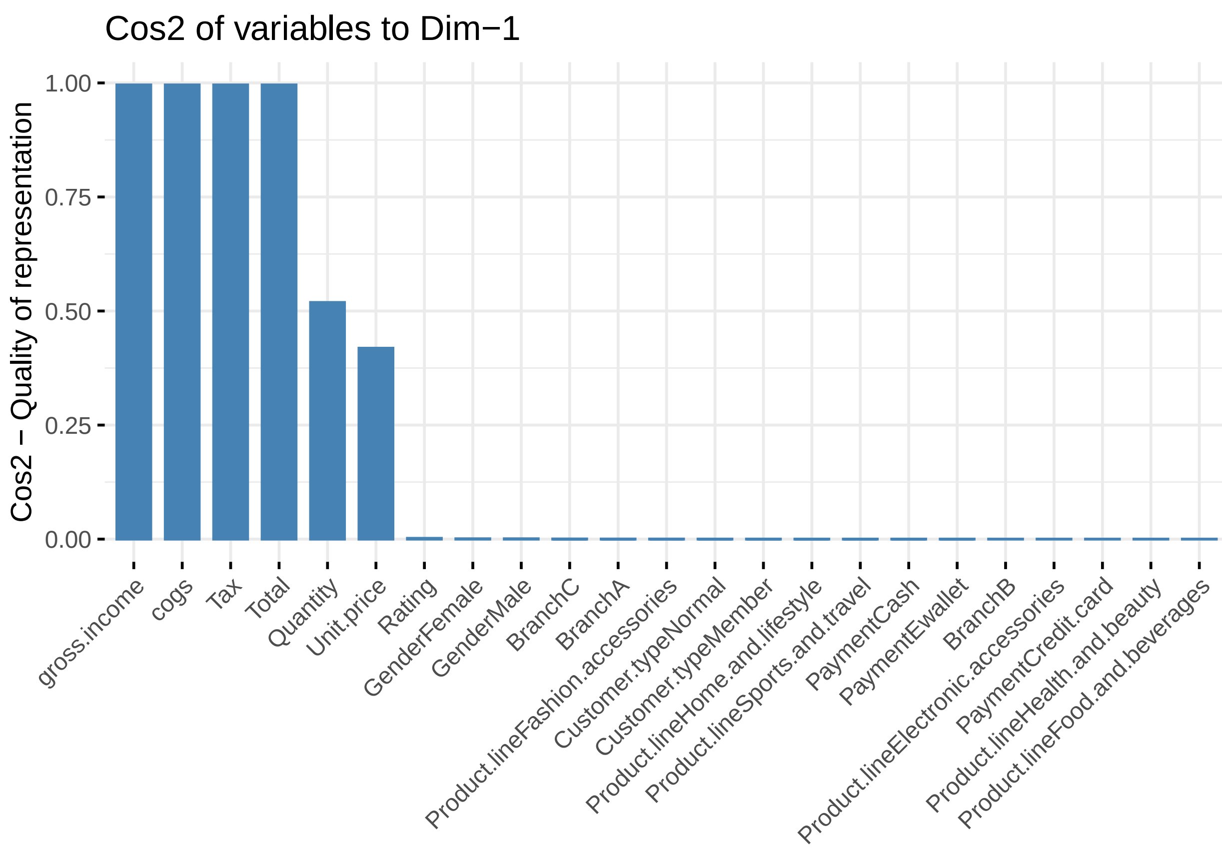
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(var_df$cos2,is.corr=FALSE)
```




```
fviz_cos2(dfPCA,choice="var",xes=1:2)
```



A high cos2 indicates good representation of the variable on the principal component.

We can hence conclude that the variables below are the more important features in our data.

```
str(newdf[,1:7])
```

```
## 'data.frame':    1000 obs. of  7 variables:
##  $ Total       : num  0.9191 -0.9872 0.0714 0.6754 1.2665 ...
##  $ Rating      : num  1.238 1.529 0.249 0.831 -0.973 ...
##  $ gross.income: num  0.9191 -0.9872 0.0714 0.6754 1.2665 ...
##  $ cogs        : num  0.9191 -0.9872 0.0714 0.6754 1.2665 ...
##  $ Quantity    : num  0.51 -0.174 0.51 0.852 0.51 ...
##  $ Tax         : num  0.9191 -0.9872 0.0714 0.6754 1.2665 ...
##  $ Unit.price  : num  0.7178 -1.5245 -0.3526 0.0962 1.1564 ...
```

FEATURE SELECTION

Random forest method

```
#libraries
library(party)
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
cf1 <- cforest(Total ~ . , data= newdf, control=cforest_unbiased(mtry=2,ntree=50))
```

```
#feature selection
varimp(cf1)
```

```
##                          Rating                      gross.income
##                    3.384917e-04                      3.907661e-01
##                            cogs                          Quantity
##                    3.301923e-01                      1.319855e-01
##                             Tax                        Unit.price
##                    2.620213e-01                      9.444527e-02
##                         BranchA                           BranchB
##                   -1.082237e-03                      7.928378e-04
##                         BranchC                Customer.typeMember
##                   -7.280935e-04                     -1.579422e-03
##             Customer.typeNormal                      GenderFemale
```

```
##                          -2.931581e-03                          4.768697e-04
##                             GenderMale Product.lineElectronic.accessories
##                           3.305157e-04                         -3.461942e-04
##     Product.lineFashion.accessories     Product.lineFood.and.beverages
##                          -6.611270e-04                         -4.664718e-03
##       Product.lineHealth.and.beauty     Product.lineHome.and.lifestyle
##                          -1.164853e-03                         -1.494883e-03
##       Product.lineSports.and.travel                            PaymentCash
##                           5.854174e-05                         -1.480815e-03
##                    PaymentCredit.card                         PaymentEwallet
##                           7.063618e-04                          5.464844e-04
```

```
# get variable importance, based on mean decrease in accuracy
```