Factoring-Analysis.R

parth

2019-10-24

```
##Author: PARTH HINGU
########  FACTOR ANALYSIS ##########

library(data.table)
library(ggplot2)  # tidyverse data visualization package
library(stringr)
library(corrplot)

## corrplot 0.84 loaded

library(psych)

##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

#Importing csv file from my local computer
airbnbOriginalDF<-read.csv("C:/Users/yadav/Desktop/MVA proj/airbnb/airbnb_1/A
irbnb Host Data For Newyork City.csv")

##Converting data frame to data table
setDT(airbnbOriginalDF)

#Removing values which are null and storing in new table.
airbnbNoNADT = airbnbOriginalDF[airbnbOriginalDF$reviews_per_month != 'NA']

#Converting datatype of last review date to DAte Format.
airbnbNoNADT[,last_review:=as.Date(last_review, '%m/%d/%Y')]

#As the neighbourhood_group column has 5 categorical values, we can factor it
, and convert our string data type.
airbnbNoNADT[,neighbourhood_group:= factor(neighbourhood_group)]

#For room type, we get 3 unique categorical values. we can factor it, and con
vert our string datatype.
airbnbNoNADT[,room_type:= factor(room_type)]

#With earlier analysis/ summary and plot we found few ouliers, therefore that
data  we have dropped below, conforming it is not impact our main dataset.
```

```r
airbnbCleaned = airbnbNoNADT[price<2500 & number_of_reviews<400 & reviews_per
_month<10]
##Manhattan area dataset
airbnbManhattan = airbnbCleaned[neighbourhood_group=='Manhattan']
nrow(airbnbManhattan)
```

```
## [1] 16584
```

```r
library(dplyr)
```

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

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

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

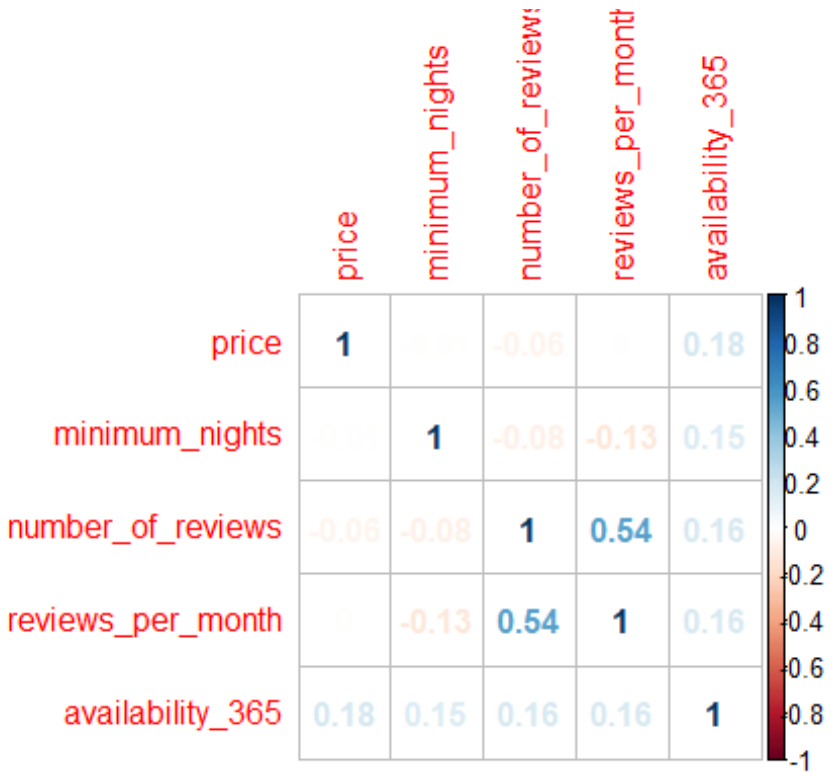
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(data.table)

##Taking the numeric columns that will contribute for variance in data
airbnbManhattanPCA = data.frame(
  airbnbManhattan$id,
  airbnbManhattan$host_id,
  airbnbManhattan$room_type,
  airbnbManhattan$price,
  airbnbManhattan$minimum_nights,
  airbnbManhattan$number_of_reviews,
  airbnbManhattan$reviews_per_month,
  airbnbManhattan$availability_365)

setDT(airbnbManhattanPCA)

##Setting column names for our new dataframe
names(airbnbManhattanPCA) <- c(
  'id',
  'host_id',
  'room_type',
  'price',
  'minimum_nights',
  'number_of_reviews',
  'reviews_per_month',
  'availability_365')
```

```
head(airbnbManhattanPCA, 5)
```

```
##      id host_id       room_type price minimum_nights number_of_reviews
## 1: 2595    2845 Entire home/apt   225              1                45
## 2: 5022    7192 Entire home/apt    80             10                 9
## 3: 5099    7322 Entire home/apt   200              3                74
## 4: 5203    7490    Private room    79              2               118
## 5: 5238    7549 Entire home/apt   150              1               160
##    reviews_per_month availability_365
## 1:              0.38              355
## 2:              0.10                0
## 3:              0.59              129
## 4:              0.99                0
## 5:              1.33              188
```

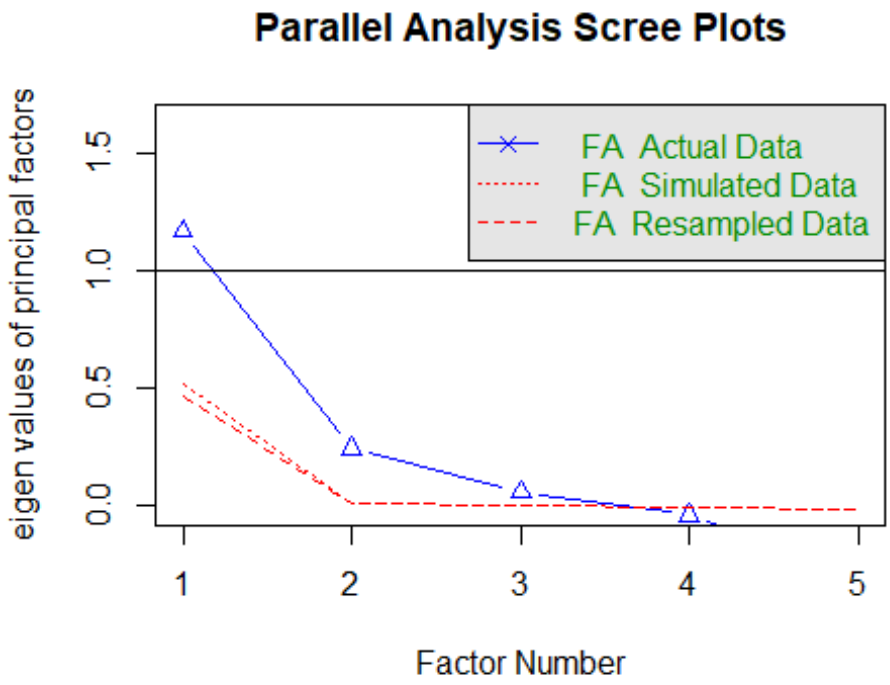
```
##Lets first check the correlation to see whether FA is good to apply
corrM = cor(airbnbManhattanPCA[,-1:-3])
corrplot(corrM, method = "number")
```




```
#The variables are not very correlated, however we see that no of reviews and
reviews_per_month are corelated
```

```
## To check acceptable number of factors and generate scree plot we use paral
lel analysis.
```

```
## Parallel analysis suggests that the number of factors =  5  and the number
of components =  NA"
parallel <- fa.parallel(airbnbManhattanPCA[,-1:-3], fm = 'minres', fa = 'fa')
```




```
## Parallel analysis suggests that the number of factors =  3  and the number
of components =  NA

##when look at the large drops in the actual data (its 2 in this case) and sp
ot the point where it levels off to the right.
##Also we locate the point of inflection - the point
##where the gap between simulated data and actual data tends to be minimum(it
s between 3 and 4)
##Factor we can take between 2 and 4
## So we will take 3 as factors here
threefactor <- principal(airbnbManhattanPCA[,-1:-3],nfactors = 3,rotate = "va
rimax")
print(threefactor)

## Principal Components Analysis
## Call: principal(r = airbnbManhattanPCA[, -1:-3], nfactors = 3, rotate = "v
arimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                      RC1   RC2   RC3   h2   u2 com
## price              -0.10  0.91 -0.09 0.86 0.14 1.0
## minimum_nights     -0.16 -0.10  0.90 0.84 0.16 1.1
## number_of_reviews   0.86 -0.06  0.01 0.75 0.25 1.0
## reviews_per_month   0.86  0.04 -0.09 0.75 0.25 1.0
```

```
## availability_365   0.33  0.53  0.53 0.67 0.33 2.7
##
##                        RC1  RC2  RC3
## SS loadings           1.63 1.13 1.11
## Proportion Var        0.33 0.23 0.22
## Cumulative Var        0.33 0.55 0.77
## Proportion Explained  0.42 0.29 0.29
## Cumulative Proportion 0.42 0.71 1.00
##
## Mean item complexity =  1.4
## Test of the hypothesis that 3 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.14
##  with the empirical chi square  6506.26  with prob <  NA
##
## Fit based upon off diagonal values = 0.54
```

```
class(threefactor)
```

```
## [1] "psych"     "principal"
```

```
#Displaying factor values.
threefactor$values
```

```
## [1] 1.6408485 1.2157045 1.0058219 0.6877868 0.4498382
```

```
round(threefactor$values, 3)
```

```
## [1] 1.641 1.216 1.006 0.688 0.450
```

```
#Displaying factor loadings
threefactor$loadings
```

```
##
## Loadings:
##                   RC1    RC2    RC3
## price             -0.104  0.914
## minimum_nights    -0.157         0.899
## number_of_reviews  0.865
## reviews_per_month  0.858
## availability_365   0.330  0.526  0.530
##
##                  RC1   RC2   RC3
## SS loadings    1.629 1.128 1.106
## Proportion Var 0.326 0.226 0.221
## Cumulative Var 0.326 0.551 0.772
```

```
# Communalities
threefactor$communality
```

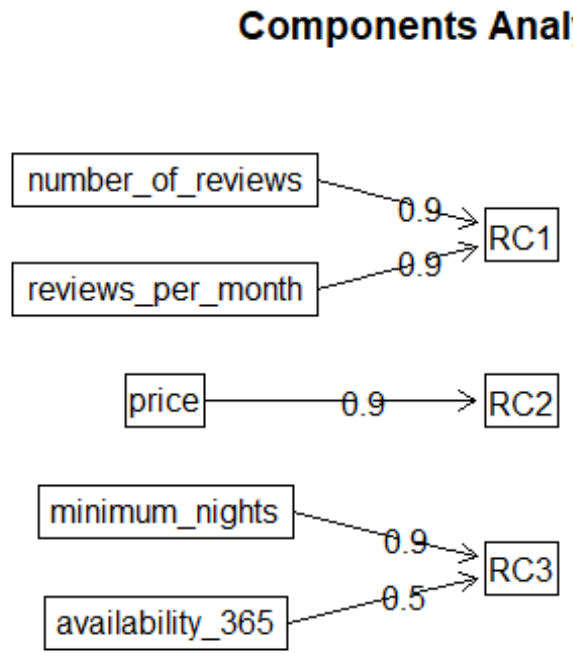
```
##             price    minimum_nights number_of_reviews reviews_per_month
##         0.8554337         0.8430710         0.7512177         0.7461790
```

```
##  availability_365
##         0.6664735
```

```
# Rotated factor scores.
head(threefactor$scores)
```

```
##               RC1        RC2          RC3
## [1,]  0.26020669  1.0711214  0.61387174
## [2,] -0.71854724 -0.8662059 -0.11774358
## [3,]  0.35433185  0.1306560 -0.05683767
## [4,]  0.91409167 -1.0337852 -0.40495262
## [5,]  1.76107882 -0.1146548  0.14415942
## [6,] -0.09323451 -0.5961246 -0.34141104
```

```
# Play with FA utilities


##Lets look at the factor mapping of different variables
fa.diagram(threefactor)
```

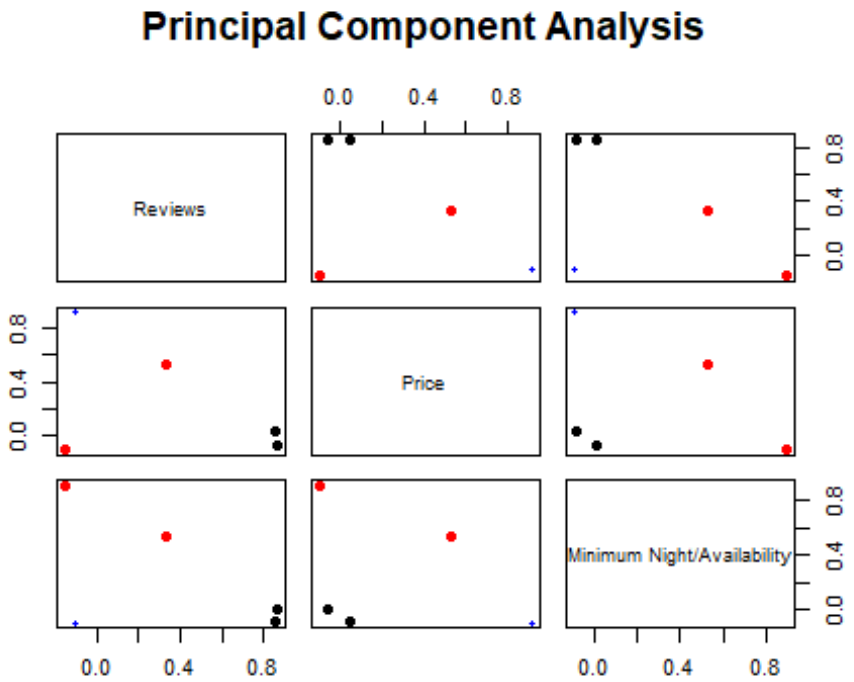



```
#Here we found that all the factors have good contribution in respective fact
ors and are singly mapped.
#Hence we can make three factor, i.e reduce 5 variable sto 3.


##Here we plot the factors and can rename the factors analyzed with three col
umn names
```

```
colnames(threefactor$loadings) <- c("Reviews","Price","Minimum Night/Availabi
lity")
colnames(threefactor$loadings)
```

```
## [1] "Reviews"                      "Price"
## [3] "Minimum Night/Availability"
```

```
plot(threefactor)
```




```
##In factor analysis we model the observed variables as linear functions of t
he "factors."
```