CaseStudy3

Oscar Wieland, Pablo Huber

2024-05-15

Preparation

```
rm(list = ls()) #Clean the entire environment
cat("\014") # clean console
```

Install required packages

Import required packages

```
library(rcompanion)   # Histogram and Normal Curve
library(nortest) # Kolmogorov-Smirnov-Test
library(corrplot) #C orrelation matrix plot
#library(olsrr)  # VIF and Tolerance Values
library(dplyr)
library(pastecs)
library(REdaS) # Bartelett's Test
library(psych)
library(lm.beta)
library(mice) # MCAR plot
library(naniar) # MCAR test
library(VIM) # Visualisation of missing values
library(lavaan)
library(semPlot)
library(psy)
```

Context and goal

We have been contacted by Frederic Fuchsbau, the marketing manager of Galeries Lafayette, to understand customers perception of the company and to explore how these perceptions impact their loyalty and commitment. The main objective of the project is to pinpoint the primary factors that contribute to Galeries Lafayette's brand equity.

He provides us with a dataset containing questions and answers of a questionnaire from 553 customers. This dataset includes various items that evaluate perceptions of the company's image, loyalty, commitment, and other relevant factors. The question were answered on a scale ranging from 1 to 7. Here is the dataset:

```
survey = read.csv("Case Study III_Structural Equation Modeling.csv")
head(survey, n=1)
```

```
##   Im1 Im2 Im3 Im4 Im5 Im6 Im7 Im8 Im9 Im10 Im11 Im12 Im13 Im14 Im15 Im16 Im17
## 1   7   7   4   4   4   7 999 999   6    7    7    6    4    7    5    5    4
##   Im18 Im19 Im20 Im21 Im22 C_CR1 C_CR2 C_CR3 C_CR4 C_REP1 C_REP2 C_REP3 COM_A1
## 1    4    5    4    5    4     6     6     6     6      5    999      5      5
##   COM_A2 COM_A3 COM_A4 SAT_1 SAT_2 SAT_3 SAT_P1 SAT_P2 SAT_P3 SAT_P4 SAT_P5
## 1      6      6      4     6     6     6      6      7      7      6      6
```

```
##   SAT_P6 TRU_1 TRU_2 TRU_3
## 1      5     7     7     7
```

After a quick review of the data, we've noticed that some entries are recorded as 999. Since the scale we're using ranges only up to 7, we've decided to interpret these 999 values as representing missing or unavailable data (NA). Therefore, we'll proceed to identify and manage all NA entries appropriately.

```
survey =data.frame(sapply(survey,function(x) ifelse((x==999),NA,as.numeric(x))))
head(survey, n=1)
```

```
##   Im1 Im2 Im3 Im4 Im5 Im6 Im7 Im8 Im9 Im10 Im11 Im12 Im13 Im14 Im15 Im16 Im17
## 1   7   7   4   4   4   7  NA  NA   6    7    7    6    4    7    5    5    4
##   Im18 Im19 Im20 Im21 Im22 C_CR1 C_CR2 C_CR3 C_CR4 C_REP1 C_REP2 C_REP3 COM_A1
## 1    4    5    4    5    4     6     6     6     6      5     NA      5      5
##   COM_A2 COM_A3 COM_A4 SAT_1 SAT_2 SAT_3 SAT_P1 SAT_P2 SAT_P3 SAT_P4 SAT_P5
## 1      6      6      4     6     6     6      6      7      7      6      6
##   SAT_P6 TRU_1 TRU_2 TRU_3
## 1      5     7     7     7
```

First of all, we will focus on understanding the dimensions through which customers perceive Galeries Lafayette. As a result, we will specifically analyze the questions pertaining to these dimensions (Image1 to Image22):

```
IM = survey[,paste0("Im", seq(1, 22))]
head(IM)
```

```
##   Im1 Im2 Im3 Im4 Im5 Im6 Im7 Im8 Im9 Im10 Im11 Im12 Im13 Im14 Im15 Im16 Im17
## 1   7   7   4   4   4   7  NA  NA   6    7    7    6    4    7    5    5    4
## 2   4   4  NA   4   3   5   3   5   4    5    4    5    5    5    4    4    4
## 3   5   5   7   7   7   4  NA   6   6    7    7    7    7    7    6    6    6
## 4   5   5   5   5   5   4   4   4   4    4    6    6    6    5    6    5    4
## 5   4   4   4   3   5   4   4   4   3    6    5    4    4    6    4    3    3
## 6   4   4   5   5  NA   4   2   5   3    3    5    3    3   NA    3    3    5
##   Im18 Im19 Im20 Im21 Im22
## 1    4    5    4    5    4
## 2    3    2    2    6    2
## 3    5    6    7    7    6
## 4    4    5    4    3    2
## 5    3    3    3    3    3
## 6    5    3    4    5    5
```

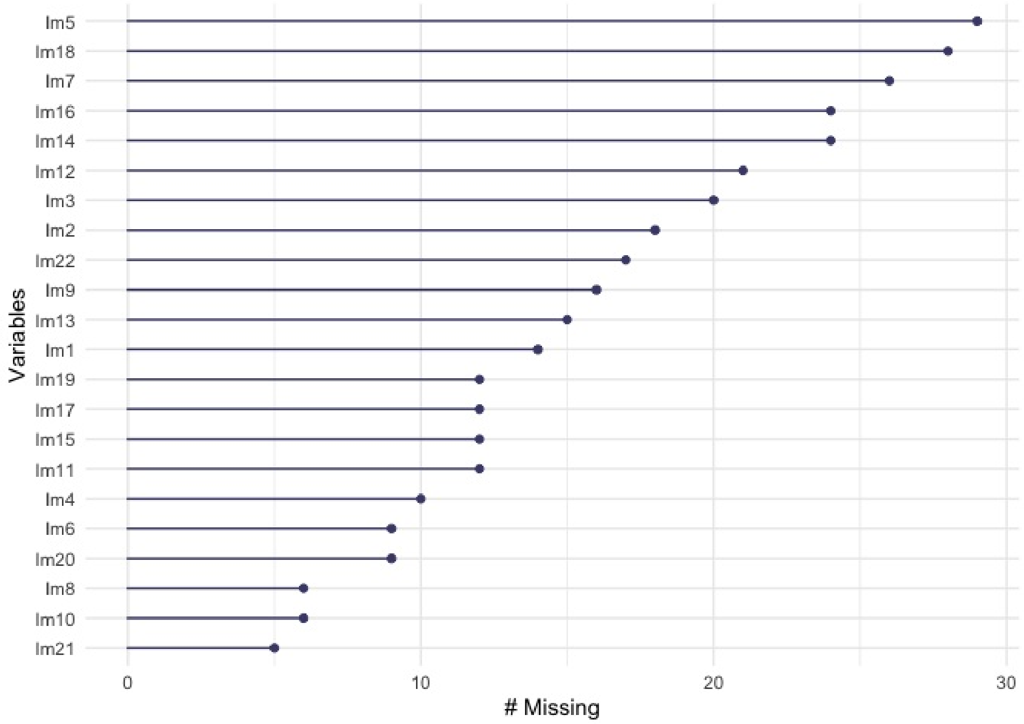
Missing values

Let's examine the NA values in the dataset for further analysis.

```
list_na =colnames(IM)[apply(IM,2,anyNA)]
list_na
```

```
##  [1] "Im1"  "Im2"  "Im3"  "Im4"  "Im5"  "Im6"  "Im7"  "Im8"  "Im9"  "Im10"
## [11] "Im11" "Im12" "Im13" "Im14" "Im15" "Im16" "Im17" "Im18" "Im19" "Im20"
## [21] "Im21" "Im22"
```

```
gg_miss_var(IM)
```



We have many missing values. While one approach could involve replacing these NAs with the mean or median, doing so might risk reducing the variance in the data. Therefore, we have opted for the simplest solution, which is to remove all instances of missing values. Even after this step, we still have a considerable amount of information available for analysis.

Removing the rows with missing values shrink the dataset from 553 rows to 385:

```
IM = na.omit(IM)
dim(IM)
```

```
## [1] 385  22
```

Question 1

First we will perform an exploratory factor analysis to get an initial idea by which dimensions customers perceive Galeries Lafayette.

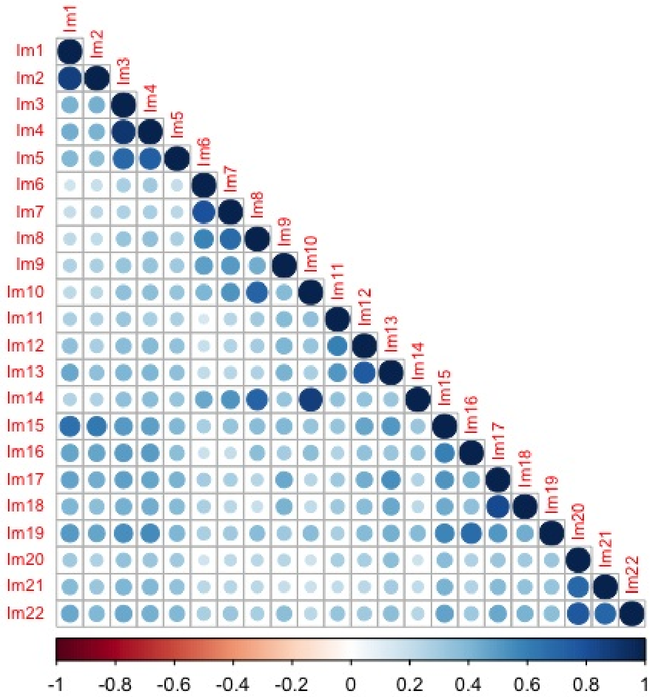
Exploratory factor analysis

Correlation among variables

The first step of a factor analysis is to analyze the correlation matrix. It will help us asses the relationships among variables.

If variables are highly correlated, it suggests that they are likely explaining similar phenomena, facilitating the extraction of common factors.

```
raqMatrix = cor(IM)
corrplot(as.matrix(raqMatrix), type = "lower", tl.cex=0.7)
```




```
# If you want the numbers:  corrplot(as.matrix(raqMatrix), type = "lower", tl.cex=0.7,met
```

They all seem to be correlated with other variables, we will now do some statistical tests to asses the significance of those correlations.

Bartlett's Test of sphericity

This test assesses whether there are significant correlations among the variables in the dataset. The null hypothesis of the test is that all variables are uncorrelated. Therefore, a small p-value from the test indicates evidence to reject the null hypothesis, suggesting that the variables are indeed correlated.

```
bart_spher(IM)
```

```
##  Bartlett's Test of Sphericity
##
## Call: bart_spher(x = IM)
##
##      X2 = 6451.238
##      df = 231
## p-value < 2.22e-16
```

We have a small p-value so we can reject the null hypothesis, which indicates that some variables are correlated.

KMO

KMO provides information about the proportion of variance in the variables that might be caused by underlying factors.

It is the ratio of the sum of squared correlations to the sum of squared correlations plus the sum of squared partial correlations:

$$KMO = \frac{\text{Sum of Squared Correlations}}{\text{Sum of Squared Correlations} + \text{Sum of Squared Partial Correlations}}$$

This means that a high KMO is associated with a small sum of squared partial correlations. This suggests that a greater proportion of the variance among the variables is shared, rather than being unique to individual variables.

Higher values indicate more suitable data for factor analysis. The criterion is that the KMO should be above 0.6.

```
KMOTEST=KMOS(IM)
KMOTEST$KMO
```

```
## [1] 0.8770975
```

We have a value higher than 0.6 which means it's good for factor analysis.

Anti-Image

If a value in the diagonal is below 0.5, the corresponding variable might be removed from factor analysis. The value on the diagonal represent the Measure of Sampling Adequacy (MSA) for each variable. It indicates how well each variable correlate with all the other variables and thus indicate if they share common factors, the closer to 1 the better (however a extremely high value might indicate that there is to much redundancy in some questions).

```
#Here the values are the diagonal elements for each vairable
sort(KMOTEST$MSA)
```

```
##       Im2       Im6       Im1      Im20      Im14      Im10       Im7       Im4
## 0.8224640 0.8224827 0.8244624 0.8266391 0.8267452 0.8285789 0.8448231 0.8542604
##      Im18       Im3      Im17      Im13      Im12      Im22      Im16      Im11
## 0.8550678 0.8640362 0.8644991 0.8722220 0.8789413 0.8793157 0.9092200 0.9113882
##      Im21       Im8       Im9      Im19       Im5      Im15
## 0.9149654 0.9300079 0.9380091 0.9400714 0.9546668 0.9647563
```

All variables are higher than 0.5. No need to remove some variables.

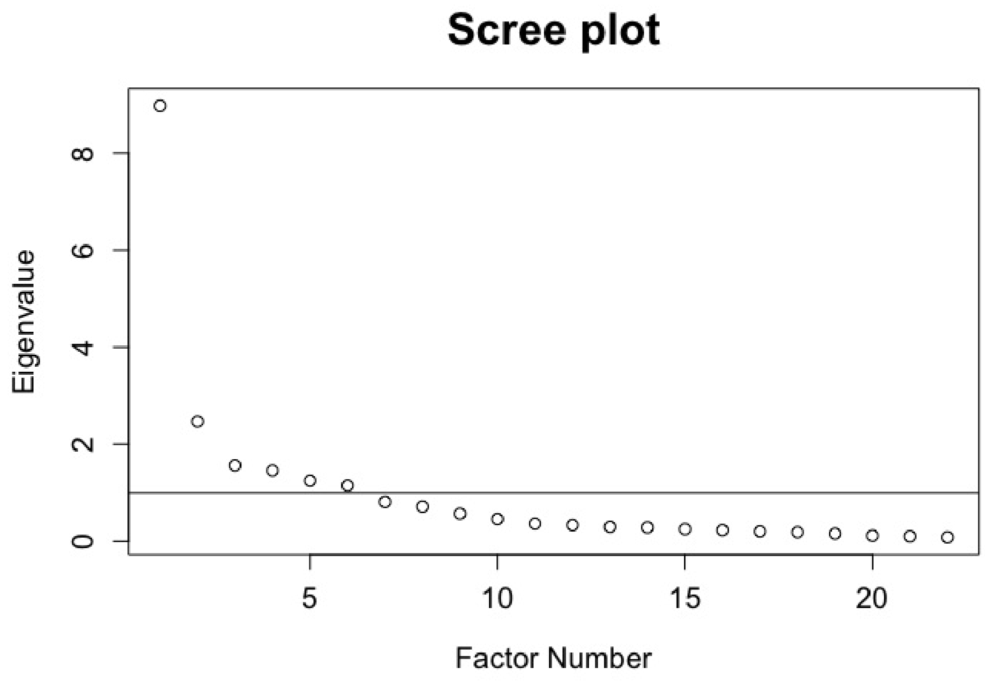
PCA

Before performing the factor analysis, we should carefully choose the number of factors. To do so, we use the kaiser criterion and the scree-test.

Scree-test and kaiser criterion

```
PC0 = principal(IM,
                        rotate="varimax", scores=TRUE)


plot(PC0$values,xlab="Factor Number",ylab="Eigenvalue",main="Scree plot",cex.lab=1.2,cex.
```




```
## integer(0)
```

```
EigenValue=PC0$values
Variance=EigenValue/ncol(IM)*100
SumVariance=cumsum(EigenValue/ncol(IM))
Total_Variance_Explained=cbind(EigenValue=EigenValue[EigenValue>0],Variance=Variance[Vari
Total_Variance_Explained
```

```
##        EigenValue   Variance Total_Variance
##  [1,] 8.97758636 40.8072107      0.4080721
##  [2,] 2.46726381 11.2148355      0.5202205
##  [3,] 1.56195916  7.0998144      0.5912186
##  [4,] 1.45683885  6.6219948      0.6574386
##  [5,] 1.24785174  5.6720533      0.7141591
##  [6,] 1.14733750  5.2151705      0.7663108
##  [7,] 0.81009930  3.6822696      0.8031335
##  [8,] 0.71161301  3.2346046      0.8354795
##  [9,] 0.56785521  2.5811600      0.8612911
## [10,] 0.45684420  2.0765645      0.8820568
## [11,] 0.36139965  1.6427257      0.8984840
## [12,] 0.33234747  1.5106703      0.9135907
## [13,] 0.29499718  1.3408963      0.9269997
## [14,] 0.28351700  1.2887137      0.9398868
## [15,] 0.24936387  1.1334721      0.9512216
## [16,] 0.22811058  1.0368663      0.9615902
## [17,] 0.20225224  0.9193284      0.9707835
## [18,] 0.18624143  0.8465520      0.9792490
## [19,] 0.15737216  0.7153280      0.9864023
## [20,] 0.11623773  0.5283533      0.9916858
## [21,] 0.10167221  0.4621464      0.9963073
## [22,] 0.08123935  0.3692698      1.0000000
```

The scree-plot is a graphical representation of eigen values by ordering them according to their size. To choose the number of factors, the rule of thumb is to choose the number before the elbow of the curve. Unfortunately, this criterion is not very clear and different people may have different choice of the elbow. Thus, it is important to include the kaiser criterion. This criterion says that only factors with eigen value higher than 1 should be extracted.

Using these two criteria and knowing the lack of precision, we decide to test different situations around the kaiser criterion. But first we will try the solution of the kaiser criterion which suggest the use of 6 factors.

The first component has an eigenvalue of 8.98 indicating that it can explain around 40.8% of the overall variance. The 6 derived factors explain a total variance of 76.6%, the 7 derived factors 80.3% and the 8 derived factord 83.5%.

PCA with 6 factors

```
PCA6 = principal(IM, rotate="varimax",nfactors=6, scores=TRUE)
print(PCA6$loadings, cutoff=0.3,sort=TRUE)
```

```
##
## Loadings:
##       RC2    RC1    RC5    RC3    RC4    RC6
## Im6    0.718                              0.494
## Im7    0.803                              0.371
## Im8    0.854
## Im10   0.791
## Im14   0.791
## Im1           0.853
## Im2           0.856
## Im15          0.685
## Im16          0.624  0.385
## Im19          0.568  0.415
## Im3                  0.829
## Im4                  0.847
## Im5                  0.778
## Im20                        0.874
## Im21                        0.830
## Im22                        0.817
## Im11                               0.770
## Im12                               0.817
## Im13                               0.738
## Im9    0.426                              0.554
## Im17          0.354                       0.687
## Im18                                      0.705
##
##                  RC2   RC1   RC5   RC3   RC4   RC6
## SS loadings    3.648 3.307 2.851 2.540 2.463 2.049
## Proportion Var 0.166 0.150 0.130 0.115 0.112 0.093
## Cumulative Var 0.166 0.316 0.446 0.561 0.673 0.766
```

We can see that the results are not very clear. There are many loadings smaller than 0.7 indicating sub-optimal performance. Moreover, certain variables load onto multiple factors, further complicating the analysis.Therefore, we will follow the suggestions of the kaiser criterion which suggest to explorie solutions involving an additional one or two factors.We experimented with subtracting one or two factors, but the outcomes did not yield satisfactory results.

First, let’s try with 1 additional factor

PCA with 7 factors

```
PCA7 = principal(IM, rotate="varimax",nfactors=7, scores=TRUE)
```

```
print(PCA7$loadings, cutoff=0.3,sort=TRUE)
```

```
##
## Loadings:
##      RC7    RC1    RC5    RC3    RC4    RC2    RC6
## Im8   0.716                                    0.497
## Im10  0.830
## Im14  0.808
## Im1          0.874
## Im2          0.890
## Im15         0.632                              0.301
## Im3                 0.830
## Im4                 0.853
## Im5                 0.813
## Im20                       0.886
## Im21                       0.835
## Im22                       0.811
## Im11                              0.776
## Im12                              0.822
## Im13                              0.751
## Im6                                      0.849
## Im7   0.375                              0.814
## Im9                               0.371  0.600
## Im17                                            0.780
## Im18                                            0.793
## Im19  0.438  0.399                              0.502
## Im16  0.483  0.446                              0.453
##
##                  RC7   RC1   RC5   RC3   RC4   RC2   RC6
## SS loadings    2.723 2.702 2.642 2.550 2.464 2.400 2.189
## Proportion Var 0.124 0.123 0.120 0.116 0.112 0.109 0.099
## Cumulative Var 0.124 0.247 0.367 0.483 0.595 0.704 0.803
```

Every variable has a loading, but things are a bit messy. Many variables are loading on more than one factor, and there are also some small loadings. So, we're thinking of trying a PCA with more factors since a lot of variables are loading on multiple factors.

PCA with 8 factors

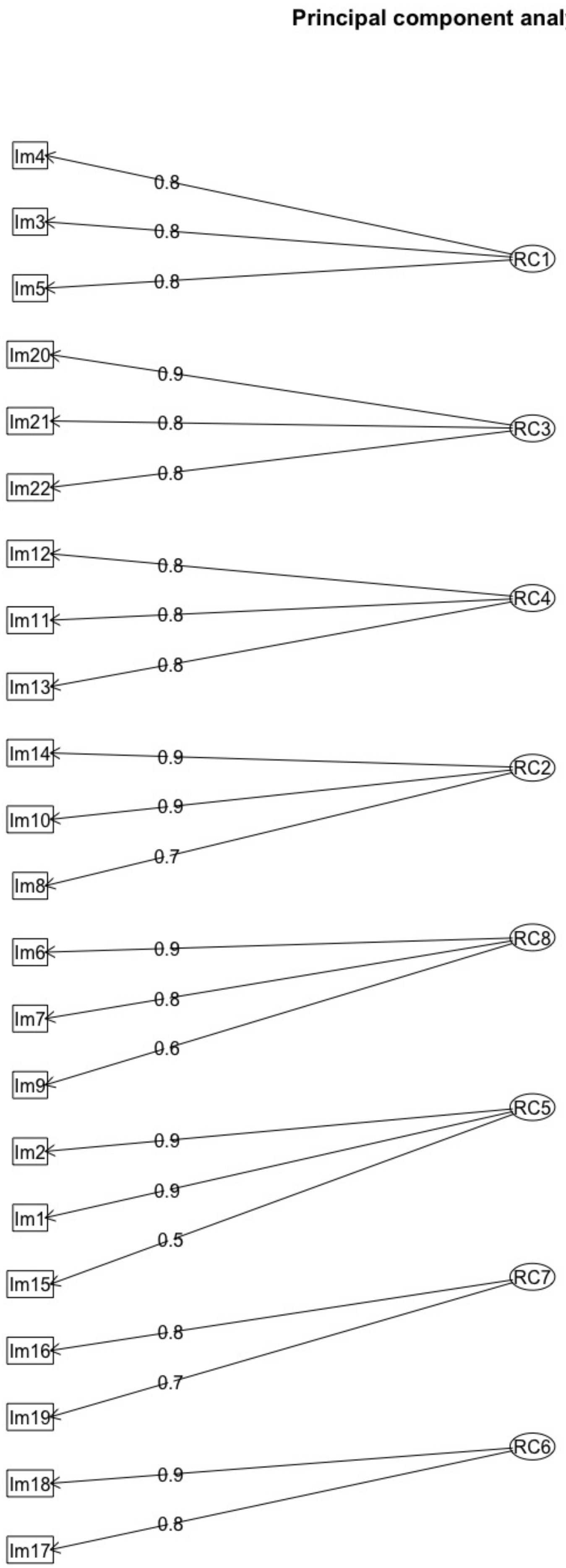
```
PCA8 = principal(IM, rotate="varimax",nfactors=8, scores=TRUE)
print(PCA8$loadings, cutoff=0.3,sort=TRUE)
```

```
##
## Loadings:
##      RC1    RC3    RC4    RC2    RC8    RC5    RC7    RC6
## Im3   0.822
## Im4   0.845
## Im5   0.814
## Im20         0.887
## Im21         0.835
## Im22         0.811
## Im11                0.768
## Im12                0.841
## Im13                0.758
## Im8                        0.674  0.515
## Im10                       0.883
## Im14                       0.884
## Im6                               0.870
```

```
## Im7                          0.366  0.825
## Im9                   0.388         0.635
## Im1                                        0.869
## Im2                                        0.891
## Im15                                       0.520  0.469
## Im16                                              0.822
## Im19                                              0.723
## Im17                                                     0.807
## Im18                                                     0.866
##
##                  RC1   RC3   RC4   RC2   RC8   RC5   RC7   RC6
## SS loadings    2.571 2.563 2.497 2.439 2.421 2.216 1.865 1.808
## Proportion Var 0.117 0.116 0.113 0.111 0.110 0.101 0.085 0.082
## Cumulative Var 0.117 0.233 0.347 0.458 0.568 0.669 0.753 0.835
```

We have observed a reduction in the number of variables loading on multiple factors, and the loadings have also increased compared to previous iterations. However, there are still a few variables with relatively low loadings on two factors (less than 0.7), indicating weaker correlations. Therefore, it's important to be cautious with these variables. Specifically, variables Im 8, 9, and 15 exhibit this behavior.

```
fa.diagram(PCA8$loadings, main="Principal component analysis with 8 factors")
```



Interpretation

Now that we have identified the factors, it's crucial to understand what these factors represent.

Therefore, we need to examine the items that constitute them and interpret the factors accordingly. We will help ourselves from the literature review on quality dimensions.

RC1 (Im 3,4,5): decoration
RC2 (Im 8,10,14): Haute cuisine.
In this case, it appears that Im8 is connected to the other two variables because it pertains to French culture, which is famous for its cuisine.

RC3 (Im 20,21,22): Ambiance
RC4 (Im 11,12,13): Prestigious brands
RC5 (Im 1,2,15): Assortment
Here, it's evident that Im15 doesn't align well with the other two variables. Recall that we previously observed Im15's relatively low loading on this factor. Hence, we've made the decision to remove this variable from consideration.
RC6 (Im 17,18): Stylish
RC7 (Im 16,19): Professional
RC8 (Im 6,7,9): France
We've chosen to retain im9 because it demonstrates a connection with the other two questions.

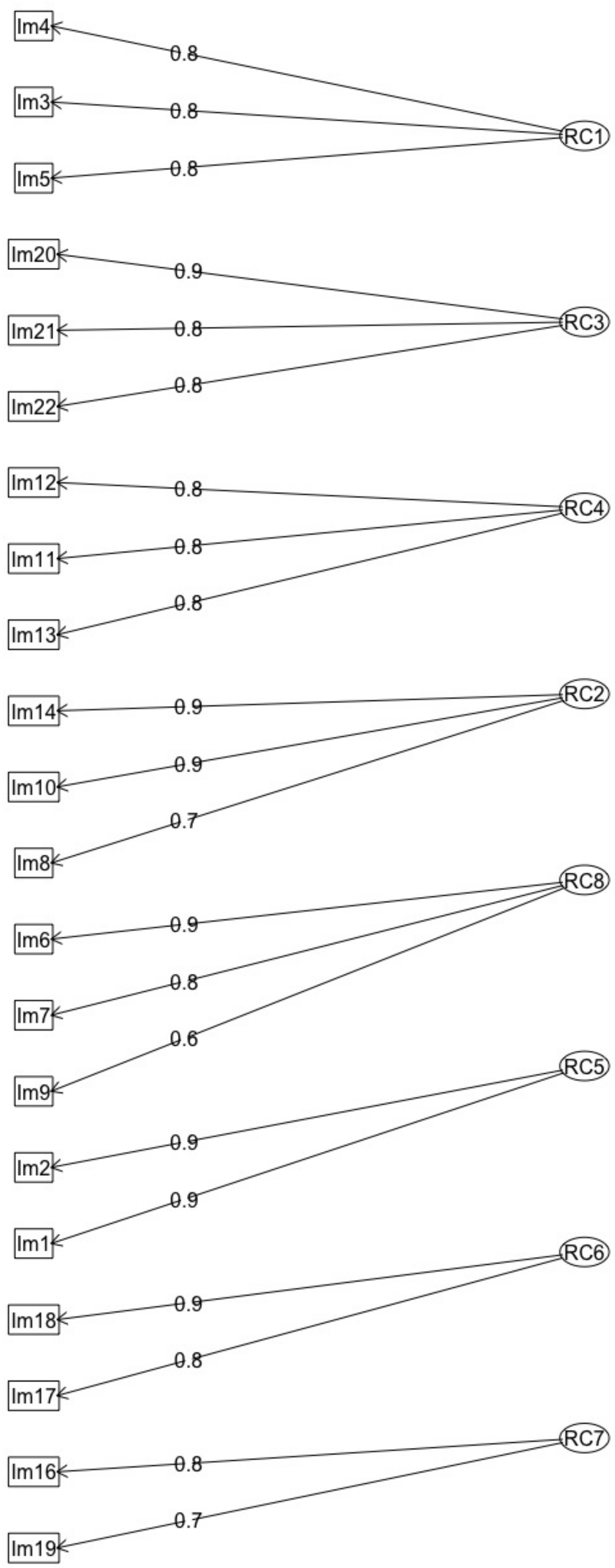
PCA with 8 factors Without 15

```
IM <- IM %>% select(-Im15)
PCA8 = principal(IM, rotate="varimax",nfactors=8, scores=TRUE)
print(PCA8$loadings, cutoff=0.3,sort=TRUE)
```

```
##
## Loadings:
##       RC1    RC3    RC4    RC2    RC8    RC5    RC6    RC7
## Im3   0.825
## Im4   0.847
## Im5   0.816
## Im20         0.889
## Im21         0.837
## Im22         0.814
## Im11                0.773
## Im12                0.842
## Im13                0.759
## Im8                        0.680  0.511
## Im10                       0.885
## Im14                       0.886
## Im6                               0.871
## Im7                        0.369  0.825
## Im9                 0.391         0.637
## Im1                                      0.865
## Im2                                      0.891
## Im17                                            0.812
## Im18                                            0.868
## Im16                                                   0.825
## Im19                                                   0.738
##
##                  RC1   RC3   RC4   RC2   RC8   RC5   RC6   RC7
## SS loadings    2.567 2.529 2.449 2.444 2.384 1.886 1.813 1.639
## Proportion Var 0.122 0.120 0.117 0.116 0.114 0.090 0.086 0.078
## Cumulative Var 0.122 0.243 0.359 0.476 0.589 0.679 0.765 0.843
```

```
fa.diagram(PCA8$loadings, main="Principal component analysis with 8 factors")
```

Principal component analysis with 8 factors



After deleting Im8, we can see that the factors didn't change. We have:

RC1 (Im 3,4,5): decoration

RC2 (Im 8,10,14): Haute_cuisine
RC3 (Im 20,21,22): Ambiance
RC4 (Im 11,12,13): Prestigious_brands
RC5 (Im 1,2): Assortment
RC6 (Im 17,18): Stylish
RC7 (Im 16,19): Professional
RC8 (Im 6,7,9): France

Now we can perform a confirmatory factor analysis to validate the factor structure identified in the exploratory analysis and assess the fit of the proposed model to the data.

Confirmatory factor analysis

Our model is:

```
model = "

decoration =~ Im3 + Im4 + Im5

Haute_cuisine =~ Im8 + Im10 + Im14

Ambiance =~ Im20 + Im21 + Im22

Prestigious_brands =~ Im11 + Im12 + Im13

Assortment =~ Im1 + Im2

Stylish =~ Im17 + Im18

Professional =~ Im16 + Im19

France =~ Im6 + Im7 + Im9

"
```

Fit assessment

Global fit

```
fit = cfa(model, data=IM, missing="ML")
summary_fit=summary(fit, fit.measures=TRUE,standardized=TRUE)
summary_fit
```

```
## lavaan 0.6.17 ended normally after 65 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        91
##
##   Number of observations                           385
##   Number of missing patterns                         1
##
## Model Test User Model:
##
##   Test statistic                               472.925
##   Degrees of freedom                               161
##   P-value (Chi-square)                           0.000
```

```
##
## Model Test Baseline Model:
##
##   Test statistic                              6231.365
##   Degrees of freedom                               210
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.948
##   Tucker-Lewis Index (TLI)                       0.932
##
##   Robust Comparative Fit Index (CFI)             0.948
##   Robust Tucker-Lewis Index (TLI)                0.932
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -10305.142
##   Loglikelihood unrestricted model (H1)     -10068.679
##
##   Akaike (AIC)                               20792.284
##   Bayesian (BIC)                             21152.029
##   Sample-size adjusted Bayesian (SABIC)      20863.297
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.071
##   90 Percent confidence interval - lower         0.064
##   90 Percent confidence interval - upper         0.078
##   P-value H_0: RMSEA <= 0.050                    0.000
##   P-value H_0: RMSEA >= 0.080                    0.022
##
##   Robust RMSEA                                   0.071
##   90 Percent confidence interval - lower         0.064
##   90 Percent confidence interval - upper         0.078
##   P-value H_0: Robust RMSEA <= 0.050             0.000
##   P-value H_0: Robust RMSEA >= 0.080             0.022
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.057
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                          Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   decoration =~
##     Im3                     1.000                               1.267    0.937
##     Im4                     1.061    0.030   35.714    0.000    1.345    0.966
##     Im5                     0.808    0.041   19.935    0.000    1.024    0.753
##   Haute_cuisine =~
##     Im8                     1.000                               0.830    0.789
##     Im10                    0.970    0.046   20.990    0.000    0.805    0.929
##     Im14                    0.969    0.046   20.942    0.000    0.805    0.930
##   Ambiance =~
##     Im20                    1.000                               1.240    0.840
##     Im21                    0.888    0.049   18.252    0.000    1.100    0.797
##     Im22                    1.133    0.056   20.207    0.000    1.404    0.908
```

```
##   Prestigious_brands =~
##     Im11              1.000                               0.734    0.642
##     Im12              1.335    0.098   13.584    0.000    0.980    0.863
##     Im13              1.449    0.114   12.718    0.000    1.064    0.871
##   Assortment =~
##     Im1               1.000                               1.310    0.978
##     Im2               0.872    0.040   21.673    0.000    1.142    0.888
##   Stylish =~
##     Im17              1.000                               1.225    0.973
##     Im18              0.951    0.047   20.079    0.000    1.165    0.847
##   Professional =~
##     Im16              1.000                               0.957    0.780
##     Im19              1.052    0.068   15.370    0.000    1.006    0.866
##   France =~
##     Im6               1.000                               1.033    0.844
##     Im7               1.083    0.054   20.112    0.000    1.119    0.930
##     Im9               0.745    0.064   11.727    0.000    0.769    0.570
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   decoration ~~
##     Haute_cuisine     0.442    0.064    6.898    0.000    0.420    0.420
##     Ambiance          0.760    0.099    7.694    0.000    0.484    0.484
##     Prestigs_brnds    0.447    0.064    6.975    0.000    0.480    0.480
##     Assortment        0.773    0.098    7.906    0.000    0.466    0.466
##     Stylish           0.792    0.094    8.435    0.000    0.510    0.510
##     Professional      0.805    0.089    9.005    0.000    0.664    0.664
##     France            0.450    0.079    5.724    0.000    0.344    0.344
##   Haute_cuisine ~~
##     Ambiance          0.288    0.061    4.726    0.000    0.280    0.280
##     Prestigs_brnds    0.255    0.042    6.066    0.000    0.419    0.419
##     Assortment        0.300    0.062    4.854    0.000    0.275    0.275
##     Stylish           0.286    0.058    4.925    0.000    0.281    0.281
##     Professional      0.385    0.055    6.964    0.000    0.485    0.485
##     France            0.544    0.062    8.713    0.000    0.634    0.634
##   Ambiance ~~
##     Prestigs_brnds    0.396    0.063    6.269    0.000    0.435    0.435
##     Assortment        0.761    0.100    7.629    0.000    0.468    0.468
##     Stylish           0.737    0.094    7.832    0.000    0.485    0.485
##     Professional      0.532    0.080    6.641    0.000    0.449    0.449
##     France            0.454    0.079    5.769    0.000    0.355    0.355
##   Prestigious_brands ~~
##     Assortment        0.449    0.065    6.887    0.000    0.466    0.466
##     Stylish           0.517    0.065    7.911    0.000    0.574    0.574
##     Professional      0.366    0.054    6.810    0.000    0.521    0.521
##     France            0.254    0.049    5.136    0.000    0.335    0.335
##   Assortment ~~
##     Stylish           0.799    0.095    8.417    0.000    0.498    0.498
##     Professional      0.761    0.089    8.588    0.000    0.607    0.607
##     France            0.329    0.077    4.261    0.000    0.243    0.243
##   Stylish ~~
##     Professional      0.694    0.081    8.540    0.000    0.592    0.592
##     France            0.445    0.077    5.801    0.000    0.351    0.351
##   Professional ~~
##     France            0.355    0.064    5.546    0.000    0.359    0.359
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Im3               4.919    0.069   71.357    0.000    4.919    3.637
##    .Im4               4.922    0.071   69.397    0.000    4.922    3.537
##    .Im5               4.932    0.069   71.156    0.000    4.932    3.626
##    .Im8               6.010    0.054  112.119    0.000    6.010    5.714
```

```
##    .Im10              6.096    0.044  137.953    0.000    6.096    7.031
##    .Im14              6.125    0.044  138.943    0.000    6.125    7.081
##    .Im20              4.670    0.075   62.070    0.000    4.670    3.163
##    .Im21              5.132    0.070   72.923    0.000    5.132    3.717
##    .Im22              4.226    0.079   53.602    0.000    4.226    2.732
##    .Im11              5.603    0.058   96.062    0.000    5.603    4.896
##    .Im12              5.626    0.058   97.243    0.000    5.626    4.956
##    .Im13              5.371    0.062   86.325    0.000    5.371    4.400
##    .Im1               4.764    0.068   69.744    0.000    4.764    3.554
##    .Im2               4.836    0.066   73.793    0.000    4.836    3.761
##    .Im17              4.966    0.064   77.379    0.000    4.966    3.944
##    .Im18              4.512    0.070   64.369    0.000    4.512    3.281
##    .Im16              5.104    0.063   81.603    0.000    5.104    4.159
##    .Im19              5.104    0.059   86.184    0.000    5.104    4.392
##    .Im6               5.823    0.062   93.417    0.000    5.823    4.761
##    .Im7               5.745    0.061   93.721    0.000    5.745    4.776
##    .Im9               5.042    0.069   73.293    0.000    5.042    3.735
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Im3               0.224    0.031    7.293    0.000    0.224    0.122
##    .Im4               0.128    0.031    4.144    0.000    0.128    0.066
##    .Im5               0.801    0.061   13.075    0.000    0.801    0.433
##    .Im8               0.417    0.035   12.045    0.000    0.417    0.377
##    .Im10              0.103    0.014    7.629    0.000    0.103    0.137
##    .Im14              0.101    0.013    7.503    0.000    0.101    0.135
##    .Im20              0.643    0.068    9.500    0.000    0.643    0.295
##    .Im21              0.697    0.066   10.479    0.000    0.697    0.365
##    .Im22              0.421    0.070    5.977    0.000    0.421    0.176
##    .Im11              0.770    0.063   12.309    0.000    0.770    0.588
##    .Im12              0.328    0.045    7.327    0.000    0.328    0.254
##    .Im13              0.359    0.052    6.911    0.000    0.359    0.241
##    .Im1               0.080    0.063    1.262    0.207    0.080    0.044
##    .Im2               0.349    0.054    6.456    0.000    0.349    0.211
##    .Im17              0.085    0.056    1.518    0.129    0.085    0.054
##    .Im18              0.534    0.063    8.429    0.000    0.534    0.282
##    .Im16              0.590    0.061    9.704    0.000    0.590    0.392
##    .Im19              0.337    0.054    6.260    0.000    0.337    0.250
##    .Im6               0.430    0.048    9.018    0.000    0.430    0.287
##    .Im7               0.196    0.047    4.206    0.000    0.196    0.135
##    .Im9               1.230    0.095   12.925    0.000    1.230    0.675
##     decoration        1.606    0.133   12.032    0.000    1.000    1.000
##     Haute_cuisine     0.689    0.076    9.090    0.000    1.000    1.000
##     Ambiance          1.537    0.158    9.727    0.000    1.000    1.000
##     Prestigs_brnds    0.539    0.082    6.607    0.000    1.000    1.000
##     Assortment        1.716    0.144   11.932    0.000    1.000    1.000
##     Stylish           1.501    0.127   11.824    0.000    1.000    1.000
##     Professional      0.916    0.109    8.409    0.000    1.000    1.000
##     France            1.066    0.109    9.742    0.000    1.000    1.000
```

First, we should assess the global fit. Different test exist:

```
summary_fit$fit[c("chisq","df","rmsea","cfi")]
```

```
##        chisq           df        rmsea          cfi
## 472.92463555 161.00000000   0.07093842   0.94819702
```

- Chi2-test: We have to look at the ratio Chi2-value/df. This should be below 5 for samples up to 1000. We have 553 observations. In our case, it is 472.925/161 = 2.937, which is less than 5 so the fit of the model is good.

- Root mean square error of approximation (RMSEA): It measures the discrepancy between the observed covariance matrix and the model-implied covariance matrix, with lower values indicating better fit. In our case, it is 0.071 which means that our result is acceptable.
- Comparative fit index (CFI): It measures the proportional improvement of fit by comparing the target model with a more restricted baseline model. Therefore, it evaluates how well the proposed model fits the data by comparing it to a baseline model. The goal is to have a high value. In our case, it is 0.948. This means we have a huge under rejection rates for missspecified models (close to acceptable)

Local fit measures

Individual item reliability

```
std.loadings<- inspect(fit, what="std")$lambda
check=std.loadings
check[check>0] <- 1
std.loadings[std.loadings==0] <- NA
std.loadings2 <- std.loadings^2
std.theta<- inspect(fit, what="std")$theta

#Individual item Reliability
IIR=std.loadings2/(colSums(std.theta)+std.loadings2)
IIR
```

```
##      decrtn Ht_csn Ambinc Prstg_ Assrtm Stylsh Prfssn France
## Im3   0.878     NA     NA     NA     NA     NA     NA     NA
## Im4   0.934     NA     NA     NA     NA     NA     NA     NA
## Im5   0.567     NA     NA     NA     NA     NA     NA     NA
## Im8      NA  0.623     NA     NA     NA     NA     NA     NA
## Im10     NA  0.863     NA     NA     NA     NA     NA     NA
## Im14     NA  0.865     NA     NA     NA     NA     NA     NA
## Im20     NA     NA  0.705     NA     NA     NA     NA     NA
## Im21     NA     NA  0.635     NA     NA     NA     NA     NA
## Im22     NA     NA  0.824     NA     NA     NA     NA     NA
## Im11     NA     NA     NA  0.412     NA     NA     NA     NA
## Im12     NA     NA     NA  0.746     NA     NA     NA     NA
## Im13     NA     NA     NA  0.759     NA     NA     NA     NA
## Im1      NA     NA     NA     NA  0.956     NA     NA     NA
## Im2      NA     NA     NA     NA  0.789     NA     NA     NA
## Im17     NA     NA     NA     NA     NA  0.946     NA     NA
## Im18     NA     NA     NA     NA     NA  0.718     NA     NA
## Im16     NA     NA     NA     NA     NA     NA  0.608     NA
## Im19     NA     NA     NA     NA     NA     NA  0.750     NA
## Im6      NA     NA     NA     NA     NA     NA     NA  0.713
## Im7      NA     NA     NA     NA     NA     NA     NA  0.865
## Im9      NA     NA     NA     NA     NA     NA     NA  0.325
```

The individual item reliability represents the true score variance of the item divided by total variance. This should be larger than 0.4. As we can see, we have generally good results. We still have to keep an eye on Im9.

Composite/construct reliability

```
#Composite/Construct Reliability
sum.std.loadings<-colSums(std.loadings, na.rm=TRUE)^2
sum.std.theta<-rowSums(std.theta)
```

```
sum.std.theta=check*sum.std.theta
CR=sum.std.loadings/(sum.std.loadings+colSums(sum.std.theta))
CR
```

```
##        decoration      Haute_cuisine          Ambiance Prestigious_brands
##         0.9190547          0.9153278         0.8856300          0.8390536
##        Assortment            Stylish      Professional             France
##         0.9316176          0.9080260         0.8085218          0.8335041
```

The composite or construct reliabilities exceed the threshold of 0.6, which indicates a satisfactory level of reliability.

Average variance extracted

```
#Average Variance Extracted
std.loadings<- inspect(fit, what="std")$lambda
std.loadings <- std.loadings^2
AVE=colSums(std.loadings)/(colSums(sum.std.theta)+colSums(std.loadings))
AVE
```

```
##        decoration      Haute_cuisine          Ambiance Prestigious_brands
##         0.7928639          0.7837198         0.7213475          0.6388766
##        Assortment            Stylish      Professional             France
##         0.8722443          0.8322095         0.6791861          0.6341149
```

The average variance extracted is comparable to the proportion of explained variance in factor analysis. It is higher than 0.5 which is good.

Standardized loadings

Now we can check the standardized loadings squared ("Std.all" in latent variables part). This indicates the variance in the items explain trough the constructs. It should be above 0.6. In our scenario, we observe a value of 0.567 for Im9, which falls below the threshold of 0.6. Consequently, we have decided to remove this question from further analysis.

Modification indices

We can also check the modification indices. It represents the anticipated reduction in the chi-square statistic when a specific parameter is liberated and the model is re-evaluated. The largest modification index shows the parameter that improves the fit most when set free.

```
modificationindices(fit) %>%filter(mi>10)
```

```
##                   lhs op  rhs     mi    epc sepc.lv sepc.all sepc.nox
## 1          decoration =~  Im9 18.758  0.219   0.277    0.205    0.205
## 2            Ambiance =~  Im9 15.343  0.210   0.261    0.193    0.193
## 3  Prestigious_brands =~  Im9 55.812  0.680   0.500    0.370    0.370
## 4          Assortment =~ Im20 16.250 -0.180  -0.235   -0.159   -0.159
## 5          Assortment =~ Im22 10.438  0.148   0.194    0.126    0.126
## 6          Assortment =~  Im9 12.774  0.167   0.219    0.162    0.162
## 7             Stylish =~ Im20 10.776 -0.160  -0.196   -0.133   -0.133
## 8             Stylish =~ Im12 13.138 -0.173  -0.212   -0.187   -0.187
## 9             Stylish =~ Im13 19.969  0.231   0.283    0.232    0.232
## 10            Stylish =~  Im7 14.354 -0.144  -0.177   -0.147   -0.147
## 11            Stylish =~  Im9 48.473  0.366   0.448    0.332    0.332
## 12       Professional =~  Im9 23.343  0.348   0.333    0.247    0.247
```

```
## 13                 France =~   Im8 89.061  0.461   0.476    0.453    0.453
## 14                 France =~  Im10 15.754 -0.132  -0.136   -0.157   -0.157
## 15                    Im8 ~~  Im14 13.559 -0.086  -0.086   -0.418   -0.418
## 16                    Im8 ~~   Im7 23.608  0.115   0.115    0.404    0.404
## 17                   Im10 ~~  Im14 71.396  0.256   0.256    2.508    2.508
## 18                   Im10 ~~   Im6 19.232 -0.070  -0.070   -0.331   -0.331
## 19                   Im20 ~~  Im21 14.760  0.279   0.279    0.417    0.417
## 20                   Im21 ~~  Im22 21.333 -0.405  -0.405   -0.749   -0.749
## 21                   Im11 ~~  Im12 11.127  0.151   0.151    0.300    0.300
## 22                   Im11 ~~  Im13 15.145 -0.191  -0.191   -0.363   -0.363
## 23                    Im6 ~~   Im7 14.785  0.413   0.413    1.422    1.422
```

As we can see, the error of Prestigious_brands and Im9 are highly correlated. This means that Im9 wants to load on Prestigious_brands (Im9 shares variance with Prestigious_brands that is not taken care of through the construct correlation). The same situation with Stylish.

The error of Im8 and France are also highly correlated. This means that Im8 want to load on France (share variance with France that is not taken care of through the construct correlation).

Same situation between Im10 and Im14. It means they share something that is not explained by the model which may be due to the fact that Im8 is also in the same factor.

Additionally for Im9, we already saw that we have some problem with it as his standardized loadings squared is below 0.6.

Therefore, we think deleting these two variables is appropriate.

We can re-perform a PCA with 8 factors and without these variables to be sure that the factors doesn't change.

PCA with 8 factors Without 15,8,9

```
IM <- IM %>% select(-c(Im9,Im8))
PCA8 = principal(IM, rotate="varimax",nfactors=8, scores=TRUE)
print(PCA8$loadings, cutoff=0.3,sort=TRUE)
```

```
##
## Loadings:
##      RC1   RC3   RC4   RC5   RC2   RC8   RC6   RC7
## Im3  0.827
## Im4  0.850
## Im5  0.818
## Im20       0.891
## Im21       0.833
## Im22       0.821
## Im11             0.758
## Im12             0.859
## Im13             0.781
## Im1                    0.866
## Im2                    0.892
## Im10                         0.892
## Im14                         0.869
## Im6                                0.917
## Im7                          0.309 0.859
## Im17                                     0.818
## Im18                                     0.876
## Im16                                           0.824
## Im19                                           0.751
##
```

```
##                      RC1   RC3   RC4   RC5   RC2   RC8   RC6   RC7
## SS loadings    2.556 2.525 2.334 1.881 1.875 1.843 1.786 1.645
## Proportion Var 0.135 0.133 0.123 0.099 0.099 0.097 0.094 0.087
## Cumulative Var 0.135 0.267 0.390 0.489 0.588 0.685 0.779 0.865
```

The results are much better now. Only one variable loads on two factors, but that's okay because it loads strongly on one of them. Also, the factors are the same as before. (the factors contain the same images so they have the same "title" as before)

Confirmatory factor analysis after deleting variables 15 + 8,9

```
model_final = "

decoration =~ Im3 + Im4 + Im5

Haute_cuisine =~ Im10 + Im14

Ambiance =~ Im20 + Im21 + Im22

Prestigious_brands =~ Im11 + Im12 + Im13

Assortment =~ Im1 + Im2

Stylish =~ Im17 + Im18

Professional =~ Im16 + Im19

France =~ Im6 + Im7

"
```

Fit Assessment

Global fit measures

```
fit_final = cfa(model_final, data=IM, missing="ML")
summary_fit_final=summary(fit_final, fit.measures=TRUE,standardized=TRUE)
summary_fit_final
```

```
## lavaan 0.6.17 ended normally after 60 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        85
##
##   Number of observations                           385
##   Number of missing patterns                         1
##
## Model Test User Model:
##
##   Test statistic                               243.685
##   Degrees of freedom                               124
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              5540.157
##   Degrees of freedom                               171
```

```
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.978
##   Tucker-Lewis Index (TLI)                       0.969
##
##   Robust Comparative Fit Index (CFI)             0.978
##   Robust Tucker-Lewis Index (TLI)                0.969
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -9308.631
##   Loglikelihood unrestricted model (H1)      -9186.789
##
##   Akaike (AIC)                               18787.263
##   Bayesian (BIC)                             19123.289
##   Sample-size adjusted Bayesian (SABIC)      18853.594
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.050
##   90 Percent confidence interval - lower         0.041
##   90 Percent confidence interval - upper         0.059
##   P-value H_0: RMSEA <= 0.050                    0.482
##   P-value H_0: RMSEA >= 0.080                    0.000
##
##   Robust RMSEA                                   0.050
##   90 Percent confidence interval - lower         0.041
##   90 Percent confidence interval - upper         0.059
##   P-value H_0: Robust RMSEA <= 0.050             0.482
##   P-value H_0: Robust RMSEA >= 0.080             0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.029
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                          Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   decoration =~
##     Im3                     1.000                               1.268    0.937
##     Im4                     1.061    0.030   35.721    0.000    1.345    0.966
##     Im5                     0.808    0.041   19.939    0.000    1.024    0.753
##   Haute_cuisine =~
##     Im10                    1.000                               0.806    0.929
##     Im14                    1.010    0.043   23.594    0.000    0.814    0.941
##   Ambiance =~
##     Im20                    1.000                               1.240    0.840
##     Im21                    0.887    0.049   18.265    0.000    1.101    0.797
##     Im22                    1.132    0.056   20.237    0.000    1.404    0.907
##   Prestigious_brands =~
##     Im11                    1.000                               0.734    0.641
##     Im12                    1.335    0.098   13.575    0.000    0.980    0.863
##     Im13                    1.450    0.114   12.708    0.000    1.065    0.872
##   Assortment =~
##     Im1                     1.000                               1.310    0.977
```

```
##     Im2                  0.873    0.040   21.670    0.000    1.143    0.889
##   Stylish =~
##     Im17                 1.000                               1.226    0.974
##     Im18                 0.950    0.048   19.920    0.000    1.164    0.847
##   Professional =~
##     Im16                 1.000                               0.955    0.778
##     Im19                 1.057    0.069   15.289    0.000    1.009    0.868
##   France =~
##     Im6                  1.000                               0.996    0.814
##     Im7                  1.172    0.079   14.776    0.000    1.167    0.971
##
## Covariances:
##                       Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   decoration ~~
##     Haute_cuisine        0.421    0.060    6.966    0.000    0.413    0.413
##     Ambiance             0.761    0.099    7.697    0.000    0.484    0.484
##     Prestigs_brnds       0.447    0.064    6.974    0.000    0.480    0.480
##     Assortment           0.774    0.098    7.909    0.000    0.466    0.466
##     Stylish              0.793    0.094    8.437    0.000    0.510    0.510
##     Professional         0.802    0.089    8.977    0.000    0.663    0.663
##     France               0.391    0.078    5.021    0.000    0.310    0.310
##   Haute_cuisine ~~
##     Ambiance             0.265    0.058    4.582    0.000    0.266    0.266
##     Prestigs_brnds       0.246    0.040    6.177    0.000    0.416    0.416
##     Assortment           0.288    0.059    4.886    0.000    0.273    0.273
##     Stylish              0.272    0.055    4.920    0.000    0.276    0.276
##     Professional         0.363    0.052    6.983    0.000    0.471    0.471
##     France               0.465    0.057    8.219    0.000    0.580    0.580
##   Ambiance ~~
##     Prestigs_brnds       0.396    0.063    6.270    0.000    0.435    0.435
##     Assortment           0.761    0.100    7.626    0.000    0.468    0.468
##     Stylish              0.737    0.094    7.831    0.000    0.485    0.485
##     Professional         0.531    0.080    6.634    0.000    0.448    0.448
##     France               0.413    0.076    5.400    0.000    0.334    0.334
##   Prestigious_brands ~~
##     Assortment           0.448    0.065    6.885    0.000    0.466    0.466
##     Stylish              0.517    0.065    7.913    0.000    0.574    0.574
##     Professional         0.365    0.054    6.799    0.000    0.521    0.521
##     France               0.221    0.047    4.744    0.000    0.303    0.303
##   Assortment ~~
##     Stylish              0.799    0.095    8.418    0.000    0.498    0.498
##     Professional         0.758    0.089    8.559    0.000    0.606    0.606
##     France               0.295    0.074    4.008    0.000    0.226    0.226
##   Stylish ~~
##     Professional         0.692    0.081    8.523    0.000    0.592    0.592
##     France               0.375    0.075    5.005    0.000    0.307    0.307
##   Professional ~~
##     France               0.316    0.062    5.095    0.000    0.332    0.332
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Im3               4.919    0.069   71.357    0.000    4.919    3.637
##    .Im4               4.922    0.071   69.397    0.000    4.922    3.537
##    .Im5               4.932    0.069   71.156    0.000    4.932    3.626
##    .Im10              6.096    0.044  137.953    0.000    6.096    7.031
##    .Im14              6.125    0.044  138.943    0.000    6.125    7.081
##    .Im20              4.670    0.075   62.070    0.000    4.670    3.163
##    .Im21              5.132    0.070   72.923    0.000    5.132    3.717
##    .Im22              4.226    0.079   53.602    0.000    4.226    2.732
##    .Im11              5.603    0.058   96.062    0.000    5.603    4.896
##    .Im12              5.626    0.058   97.243    0.000    5.626    4.956
##    .Im13              5.371    0.062   86.325    0.000    5.371    4.400
```

```
##     .Im1             4.764    0.068   69.744    0.000    4.764    3.554
##     .Im2             4.836    0.066   73.793    0.000    4.836    3.761
##     .Im17            4.966    0.064   77.379    0.000    4.966    3.944
##     .Im18            4.512    0.070   64.369    0.000    4.512    3.281
##     .Im16            5.104    0.063   81.603    0.000    5.104    4.159
##     .Im19            5.104    0.059   86.184    0.000    5.104    4.392
##     .Im6             5.823    0.062   93.418    0.000    5.823    4.761
##     .Im7             5.745    0.061   93.721    0.000    5.745    4.776
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     .Im3             0.223    0.031    7.275    0.000    0.223    0.122
##     .Im4             0.129    0.031    4.176    0.000    0.129    0.066
##     .Im5             0.801    0.061   13.072    0.000    0.801    0.433
##     .Im10            0.103    0.023    4.519    0.000    0.103    0.137
##     .Im14            0.086    0.023    3.759    0.000    0.086    0.115
##     .Im20            0.641    0.068    9.498    0.000    0.641    0.294
##     .Im21            0.696    0.066   10.490    0.000    0.696    0.365
##     .Im22            0.423    0.070    6.020    0.000    0.423    0.177
##     .Im11            0.771    0.063   12.308    0.000    0.771    0.589
##     .Im12            0.329    0.045    7.347    0.000    0.329    0.255
##     .Im13            0.357    0.052    6.886    0.000    0.357    0.240
##     .Im1             0.081    0.063    1.283    0.199    0.081    0.045
##     .Im2             0.348    0.054    6.435    0.000    0.348    0.210
##     .Im17            0.082    0.057    1.451    0.147    0.082    0.052
##     .Im18            0.536    0.064    8.391    0.000    0.536    0.283
##     .Im16            0.595    0.061    9.730    0.000    0.595    0.395
##     .Im19            0.332    0.054    6.116    0.000    0.332    0.246
##     .Im6             0.504    0.067    7.555    0.000    0.504    0.337
##     .Im7             0.084    0.077    1.088    0.277    0.084    0.058
##      decoration      1.607    0.133   12.037    0.000    1.000    1.000
##      Haute_cuisine   0.649    0.058   11.220    0.000    1.000    1.000
##      Ambiance        1.538    0.158    9.736    0.000    1.000    1.000
##      Prestigs_brnds  0.539    0.082    6.602    0.000    1.000    1.000
##      Assortment      1.715    0.144   11.924    0.000    1.000    1.000
##      Stylish         1.504    0.127   11.809    0.000    1.000    1.000
##      Professional    0.911    0.109    8.372    0.000    1.000    1.000
##      France          0.992    0.116    8.558    0.000    1.000    1.000
```

We should also assess the global fit:

```
summary_fit_final$fit[c("chisq","df","rmsea","cfi")]
```

```
##        chisq           df        rmsea          cfi
## 243.68544902 124.00000000   0.05007022   0.97770871
```

- Chi2-test: 243.685/124 = 1.965, which is less than 5 so the fit of the model is good.
- Root mean square error of approximation (RMSEA): It is 0.05 which means that our result is good and better than before.
- Comparative fit index (CFI): The CFI is now 0.978 which improved. The model can be accepted.

Standardized loadings

The standardized loadings squared are all above 0.6.

Local fit measures

Cronbach's Test

```
CronReli_decoration = cronbach(subset(IM, select = c(Im3, Im4, Im5)))
CronReli_haute_cuisine = cronbach(subset(IM, select = c(Im10, Im14)))
CronReli_ambiance = cronbach(subset(IM, select = c(Im20, Im21, Im22)))
CronReli_prestigious_brands = cronbach(subset(IM, select = c(Im11, Im12, Im13)))
CronReli_assortment = cronbach(subset(IM, select = c(Im1, Im2)))
CronReli_stylish = cronbach(subset(IM, select = c(Im17, Im18)))
CronReli_professional = cronbach(subset(IM, select = c(Im16, Im19)))
CronReli_france = cronbach(subset(IM, select = c(Im6, Im7)))

list(
  CronReli_decoration = CronReli_decoration$alpha,
  CronReli_haute_cuisine = CronReli_haute_cuisine$alpha,
  CronReli_ambiance = CronReli_ambiance$alpha,
  CronReli_prestigious_brands = CronReli_prestigious_brands$alpha,
  CronReli_assortment = CronReli_assortment$alpha,
  CronReli_stylish = CronReli_stylish$alpha,
  CronReli_professional = CronReli_professional$alpha,
  CronReli_france = CronReli_france$alpha
)
```

```
## $CronReli_decoration
## [1] 0.9127438
##
## $CronReli_haute_cuisine
## [1] 0.9328373
##
## $CronReli_ambiance
## [1] 0.8860165
##
## $CronReli_prestigious_brands
## [1] 0.8307632
##
## $CronReli_assortment
## [1] 0.9290584
##
## $CronReli_stylish
## [1] 0.9017849
##
## $CronReli_professional
## [1] 0.8055802
##
## $CronReli_france
## [1] 0.8828341
```

Cronbach's alpha coefficient measures the internal consistency, or reliability, of a set of survey items. Cronbach's alpha quantifies the level of agreement on a standardized 0 to 1 scale. Higher values indicating a higher agreement between items.

We can see that all the cronbach's alpha are above 0.8 which is really good because the threshold is 0.7. This means that we have high internal concistency reliability.

Individual item reliability

```
std.loadings<- inspect(fit_final, what="std")$lambda
check=std.loadings
check[check>0] <- 1
std.loadings[std.loadings==0] <- NA
```

```
std.loadings2 <- std.loadings^2
std.theta<- inspect(fit_final, what="std")$theta

#Individual item Reliability
IIR=std.loadings2/(colSums(std.theta)+std.loadings2)
IIR
```

```
##       decrtn Ht_csn Ambinc Prstg_ Assrtm Stylsh Prfssn France
## Im3    0.878     NA     NA     NA     NA     NA     NA     NA
## Im4    0.934     NA     NA     NA     NA     NA     NA     NA
## Im5    0.567     NA     NA     NA     NA     NA     NA     NA
## Im10      NA  0.863     NA     NA     NA     NA     NA     NA
## Im14      NA  0.885     NA     NA     NA     NA     NA     NA
## Im20      NA     NA  0.706     NA     NA     NA     NA     NA
## Im21      NA     NA  0.635     NA     NA     NA     NA     NA
## Im22      NA     NA  0.823     NA     NA     NA     NA     NA
## Im11      NA     NA     NA  0.411     NA     NA     NA     NA
## Im12      NA     NA     NA  0.745     NA     NA     NA     NA
## Im13      NA     NA     NA  0.760     NA     NA     NA     NA
## Im1       NA     NA     NA     NA  0.955     NA     NA     NA
## Im2       NA     NA     NA     NA  0.790     NA     NA     NA
## Im17      NA     NA     NA     NA     NA  0.948     NA     NA
## Im18      NA     NA     NA     NA     NA  0.717     NA     NA
## Im16      NA     NA     NA     NA     NA     NA  0.605     NA
## Im19      NA     NA     NA     NA     NA     NA  0.754     NA
## Im6       NA     NA     NA     NA     NA     NA     NA  0.663
## Im7       NA     NA     NA     NA     NA     NA     NA  0.942
```

As we can see, we have everything above 0.4 which is good. (This means we have small errors)

Composite/construct reliability

```
#Composite/Construct Reliability
sum.std.loadings<-colSums(std.loadings, na.rm=TRUE)^2
sum.std.theta<-rowSums(std.theta)
sum.std.theta=check*sum.std.theta
CR=sum.std.loadings/(sum.std.loadings+colSums(sum.std.theta))
CR
```

```
##          decoration      Haute_cuisine           Ambiance Prestigious_brands
##           0.9190648          0.9328763          0.8856736          0.8389902
##          Assortment            Stylish       Professional             France
##           0.9315832          0.9081337          0.8087486          0.8897228
```

The composite or construct reliabilities exceed the threshold of 0.6, which indicates a satisfactory level of reliability.

Average variance extracted

```
#Average Variance Extracted
std.loadings<- inspect(fit_final, what="std")$lambda
std.loadings <- std.loadings^2
AVE=colSums(std.loadings)/(colSums(sum.std.theta)+colSums(std.loadings))
AVE
```

```
##          decoration      Haute_cuisine           Ambiance Prestigious_brands
##           0.7928832          0.8742013          0.7214262          0.6387833
```

```
##          Assortment              Stylish        Professional              France
##           0.8721800            0.8324059           0.6795660           0.8025646
```

It is higher than 0.5 which is good.

Modification indices

We can also check the modification indices:

```
modificationindices(fit_final) %>%filter(mi>10)
```

```
##           lhs op  rhs     mi    epc  sepc.lv sepc.all sepc.nox
## 1  Assortment =~ Im20 16.337 -0.180  -0.236   -0.160   -0.160
## 2  Assortment =~ Im22 10.522  0.149   0.195    0.126    0.126
## 3     Stylish =~ Im20 10.800 -0.160  -0.196   -0.133   -0.133
## 4     Stylish =~ Im12 13.016 -0.172  -0.211   -0.186   -0.186
## 5     Stylish =~ Im13 19.790  0.230   0.281    0.231    0.231
## 6        Im10 ~~  Im6 11.183 -0.054  -0.054   -0.239   -0.239
## 7        Im20 ~~ Im21 14.308  0.274   0.274    0.410    0.410
## 8        Im21 ~~ Im22 20.740 -0.398  -0.398   -0.734   -0.734
## 9        Im11 ~~ Im12 11.346  0.152   0.152    0.302    0.302
## 10       Im11 ~~ Im13 15.256 -0.191  -0.191   -0.364   -0.364
```

Now the modification indices (MI) are fairly consistent and not particularly large.

```
std_fit=inspect(fit_final, "std")
std_fit$psi
```

```
##                     decrtn Ht_csn Ambinc Prstg_ Assrtm Stylsh Prfssn France
## decoration           1.000
## Haute_cuisine        0.413  1.000
## Ambiance             0.484  0.266  1.000
## Prestigious_brands   0.480  0.416  0.435  1.000
## Assortment           0.466  0.273  0.468  0.466  1.000
## Stylish              0.510  0.276  0.485  0.574  0.498  1.000
## Professional         0.663  0.471  0.448  0.521  0.606  0.592  1.000
## France               0.310  0.580  0.334  0.303  0.226  0.307  0.332  1.000
```

Here we can see how latent constructs correlate with each other. High correlations between them might indicate a potential issue as the constructs may not be as distinct as theoretically proposed. A common threshold is 0.7, in our case it is much lower which is good.

Question 1

In conclusion, the Galeries Lafayette's brand image can be best characterized by eight distinct dimensions: decoration, Haute cuisine, Ambiance, Prestigious brands, Assortment, Stylish, Professional, France. This model meets all the criteria required for both global and local fit.

Section: Question 2-3

Now, we want to understand if the mechanism driving the satisfaction and affective commitment are similar and if customer satisfaction and affective commitment are mediating the impact of image perceptions on outcome.

```
dataset = survey
```

```
model2<-"
## Measurement model
decoration =~ Im3 + Im4 + Im5

Haute_cuisine =~ Im10 + Im14

Ambiance =~ Im20 + Im21 + Im22

Prestigious_brands =~ Im11 + Im12 + Im13

Assortment =~ Im1 + Im2

Stylish =~ Im17 + Im18

Professional =~ Im16 + Im19

France =~ Im6 + Im7



Satisfaction = ~ SAT_1 + SAT_2 + SAT_3

Commitment = ~ COM_A1 + COM_A2 + COM_A3 + COM_A4

Repurchase = ~ C_REP1 + C_REP2 + C_REP3

Cocreation = ~ C_CR1 + C_CR3 + C_CR4


##Structural model


Repurchase ~ a * Satisfaction + c * Commitment

Cocreation ~ b * Satisfaction + d * Commitment



Satisfaction ~ e*decoration + f*Haute_cuisine + g*Ambiance + h*Prestigious_brands + i*Ass


Commitment ~ m*decoration + n*Haute_cuisine + o*Ambiance + p*Prestigious_brands + q*Assor

Repurchase ~ cc*decoration + dd*Haute_cuisine + ee*Ambiance + ff*Prestigious_brands + gg*

Cocreation ~ u*decoration + v*Haute_cuisine + w*Ambiance + x*Prestigious_brands + y*Assor




#Indirect effects


# -> satisfaction -> repurchase
ae:=a*e
af:=a*f
ag:=a*g
ah:=a*h
ai:=a*i
```

```
aj:=a*j
ak:=a*k
al:=a*l

# -> commitment -> repurchase
cm:=c*m
cn:=c*n
co:=c*o
cp:=c*p
cq:=c*q
cr:=c*r
cs:=c*s
ct:=c*t

# -> satisfaction -> cocreation
be:=b*e
bf:=b*f
bg:=b*g
bh:=b*h
bi:=b*i
bj:=b*j
bk:=b*k
bl:=b*l


# -> commitment -> cocreation
dm:=d*m
dn:=d*n
do:=d*o
dp:=d*p
dq:=d*q
dr:=d*r
ds:=d*s
dt:=d*t




#Total effect

# -> cocreation (direct + satisfaction -> cocreation + -> commitment -> cocreation)
te1c:= u+(b*e)+(d*m)
te2c:=v+(b*f)+(d*n)
te3c:=w+(b*g)+(d*o)
te4c:=x+(b*h)+(d*p)
te5c:=y+(b*i)+(d*q)
te6c:=z+(b*j)+(d*r)
te7c:=aa+(b*k)+(d*s)
te8c:=bb+(b*l)+(d*t)


# -> repurchase (direct + -> satisfaction -> repurchase + -> commitment -> repurchase)
te1r:= cc+(a*e)+(c*m)
te2r:=dd+(a*f)+(c*n)
te3r:=ee+(a*g)+(c*o)
te4r:=ff+(a*h)+(c*p)
te5r:=gg+(a*i)+(c*q)
te6r:=hh+(a*j)+(c*r)
te7r:=ii+(a*k)+(c*s)
te8r:=jj+(a*l)+(c*t)
```

```
#Total indirect effect
# -> repurchase
ti1r:= (a*e)+(c*m)
ti2r:=(a*f)+(c*n)
ti3r:=(a*g)+(c*o)
ti4r:=(a*h)+(c*p)
ti5r:=(a*i)+(c*q)
ti6r:=(a*j)+(c*r)
ti7r:=(a*k)+(c*s)
ti8r:=(a*l)+(c*t)

# -> cocreation
ti1c:=(b*e)+(d*m)
ti2c:=(b*f)+(d*n)
ti3c:=(b*g)+(d*o)
ti4c:=(b*h)+(d*p)
ti5c:=(b*i)+(d*q)
ti6c:=(b*j)+(d*r)
ti7c:=(b*k)+(d*s)
ti8c:=(b*l)+(d*t)

"

fit2 <- cfa(model2, data=dataset, missing="ML", estimator="MLR")

Sum_fit=summary(fit2, fit.measures=TRUE, standardized=TRUE)
Sum_fit
```

```
## lavaan 0.6.17 ended normally after 150 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                       161
##
##   Number of observations                           553
##   Number of missing patterns                       135
##
## Model Test User Model:
##                                               Standard      Scaled
##   Test Statistic                               700.455     632.247
##   Degrees of freedom                               399         399
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.108
##     Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                             11978.557    9969.592
##   Degrees of freedom                               496         496
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.202
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.974       0.975
##   Tucker-Lewis Index (TLI)                       0.967       0.969
##
##   Robust Comparative Fit Index (CFI)                         0.979
##   Robust Tucker-Lewis Index (TLI)                            0.973
##
```

```
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -22368.900  -22368.900
##   Scaling correction factor                                 1.404
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)     -22018.673  -22018.673
##   Scaling correction factor                                 1.193
##       for the MLR correction
##
##   Akaike (AIC)                               45059.800   45059.800
##   Bayesian (BIC)                             45754.573   45754.573
##   Sample-size adjusted Bayesian (SABIC)      45243.488   45243.488
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.037       0.033
##   90 Percent confidence interval - lower         0.032       0.028
##   90 Percent confidence interval - upper         0.041       0.037
##   P-value H_0: RMSEA <= 0.050                    1.000       1.000
##   P-value H_0: RMSEA >= 0.080                    0.000       0.000
##
##   Robust RMSEA                                              0.034
##   90 Percent confidence interval - lower                    0.029
##   90 Percent confidence interval - upper                    0.039
##   P-value H_0: Robust RMSEA <= 0.050                        1.000
##   P-value H_0: Robust RMSEA >= 0.080                        0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.041       0.041
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                          Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   decoration =~
##     Im3                     1.000                               1.235    0.936
##     Im4                     1.057    0.028   37.421    0.000    1.306    0.970
##     Im5                     0.818    0.046   17.727    0.000    1.011    0.760
##   Haute_cuisine =~
##     Im10                    1.000                               0.810    0.921
##     Im14                    1.021    0.041   24.871    0.000    0.827    0.955
##   Ambiance =~
##     Im20                    1.000                               1.262    0.844
##     Im21                    0.857    0.046   18.667    0.000    1.081    0.789
##     Im22                    1.056    0.048   21.798    0.000    1.333    0.873
##   Prestigious_brands =~
##     Im11                    1.000                               0.701    0.613
##     Im12                    1.414    0.113   12.568    0.000    0.991    0.872
##     Im13                    1.468    0.140   10.474    0.000    1.029    0.855
##   Assortment =~
##     Im1                     1.000                               1.297    0.974
##     Im2                     0.896    0.035   25.253    0.000    1.162    0.904
##   Stylish =~
##     Im17                    1.000                               1.205    0.970
##     Im18                    0.992    0.042   23.744    0.000    1.196    0.855
##   Professional =~
##     Im16                    1.000                               0.919    0.764
```

```
##     Im19                 1.043    0.071   14.680    0.000    0.959    0.853
##   France =~
##     Im6                  1.000                               0.987    0.822
##     Im7                  1.158    0.076   15.174    0.000    1.143    0.944
##   Satisfaction =~
##     SAT_1                1.000                               0.882    0.865
##     SAT_2                0.933    0.059   15.698    0.000    0.823    0.819
##     SAT_3                0.809    0.061   13.271    0.000    0.714    0.624
##   Commitment =~
##     COM_A1               1.000                               1.144    0.796
##     COM_A2               1.174    0.049   23.795    0.000    1.342    0.836
##     COM_A3               1.162    0.059   19.802    0.000    1.329    0.817
##     COM_A4               1.278    0.064   20.041    0.000    1.462    0.842
##   Repurchase =~
##     C_REP1               1.000                               0.596    0.816
##     C_REP2               0.971    0.048   20.251    0.000    0.579    0.931
##     C_REP3               0.702    0.057   12.368    0.000    0.419    0.756
##   Cocreation =~
##     C_CR1                1.000                               1.658    0.851
##     C_CR3                1.033    0.056   18.597    0.000    1.712    0.826
##     C_CR4                0.963    0.056   17.089    0.000    1.597    0.806
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Repurchase ~
##     Satisfctn  (a)    0.215    0.049    4.396    0.000    0.318    0.318
##     Commitmnt  (c)    0.184    0.031    5.882    0.000    0.354    0.354
##   Cocreation ~
##     Satisfctn  (b)   -0.357    0.143   -2.501    0.012   -0.190   -0.190
##     Commitmnt  (d)    0.546    0.094    5.824    0.000    0.377    0.377
##   Satisfaction ~
##     decoratin  (e)   -0.109    0.048   -2.285    0.022   -0.152   -0.152
##     Haute_csn  (f)    0.081    0.075    1.069    0.285    0.074    0.074
##     Ambiance   (g)    0.052    0.044    1.169    0.243    0.074    0.074
##     Prstgs_br  (h)   -0.038    0.095   -0.400    0.689   -0.030   -0.030
##     Assortmnt  (i)    0.134    0.053    2.512    0.012    0.197    0.197
##     Stylish    (j)    0.008    0.061    0.131    0.896    0.011    0.011
##     Professnl  (k)    0.459    0.105    4.382    0.000    0.479    0.479
##     France     (l)    0.103    0.053    1.934    0.053    0.115    0.115
##   Commitment ~
##     decoratin  (m)   -0.024    0.058   -0.413    0.680   -0.026   -0.026
##     Haute_csn  (n)    0.028    0.090    0.308    0.758    0.020    0.020
##     Ambiance   (o)    0.373    0.059    6.359    0.000    0.411    0.411
##     Prstgs_br  (p)   -0.187    0.116   -1.614    0.106   -0.115   -0.115
##     Assortmnt  (q)    0.101    0.055    1.840    0.066    0.114    0.114
##     Stylish    (r)   -0.018    0.068   -0.260    0.795   -0.019   -0.019
##     Professnl  (s)    0.160    0.129    1.240    0.215    0.129    0.129
##     France     (t)    0.223    0.067    3.327    0.001    0.192    0.192
##   Repurchase ~
##     decoratin (cc)    0.010    0.027    0.358    0.720    0.020    0.020
##     Haute_csn (dd)    0.038    0.047    0.806    0.420    0.051    0.051
##     Ambiance  (ee)    0.040    0.029    1.375    0.169    0.085    0.085
##     Prstgs_br (ff)    0.077    0.052    1.490    0.136    0.091    0.091
##     Assortmnt (gg)   -0.017    0.024   -0.688    0.491   -0.037   -0.037
##     Stylish   (hh)   -0.011    0.028   -0.385    0.701   -0.022   -0.022
##     Professnl (ii)   -0.037    0.056   -0.655    0.513   -0.056   -0.056
##     France    (jj)   -0.034    0.031   -1.073    0.283   -0.056   -0.056
##   Cocreation ~
##     decoratin  (u)   -0.031    0.100   -0.311    0.756   -0.023   -0.023
##     Haute_csn  (v)   -0.080    0.142   -0.560    0.575   -0.039   -0.039
##     Ambiance   (w)    0.152    0.093    1.633    0.103    0.116    0.116
##     Prstgs_br  (x)    0.197    0.149    1.318    0.187    0.083    0.083
```

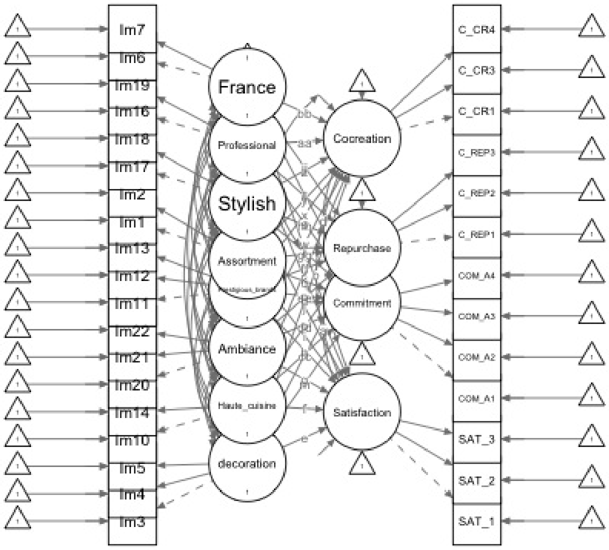
```
##     Assortmnt  (y)   -0.006    0.083   -0.074    0.941   -0.005   -0.005
##     Stylish    (z)    0.022    0.091    0.245    0.807    0.016    0.016
##     Professnl (aa)   -0.176    0.194   -0.908    0.364   -0.098   -0.098
##     France    (bb)   -0.127    0.110   -1.152    0.249   -0.075   -0.075
##
## Covariances:
##                         Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   decoration ~~
##     Haute_cuisine          0.417    0.053    7.922    0.000    0.417    0.417
##     Ambiance               0.728    0.086    8.422    0.000    0.467    0.467
##     Prestigs_brnds         0.407    0.054    7.531    0.000    0.470    0.470
##     Assortment             0.708    0.076    9.364    0.000    0.442    0.442
##     Stylish                0.769    0.080    9.568    0.000    0.516    0.516
##     Professional           0.744    0.078    9.528    0.000    0.655    0.655
##     France                 0.413    0.074    5.553    0.000    0.339    0.339
##   Haute_cuisine ~~
##     Ambiance               0.301    0.053    5.668    0.000    0.295    0.295
##     Prestigs_brnds         0.256    0.040    6.399    0.000    0.452    0.452
##     Assortment             0.327    0.050    6.583    0.000    0.312    0.312
##     Stylish                0.317    0.042    7.576    0.000    0.325    0.325
##     Professional           0.371    0.047    7.848    0.000    0.499    0.499
##     France                 0.469    0.050    9.313    0.000    0.587    0.587
##   Ambiance ~~
##     Prestigs_brnds         0.370    0.064    5.772    0.000    0.418    0.418
##     Assortment             0.732    0.084    8.665    0.000    0.447    0.447
##     Stylish                0.785    0.083    9.464    0.000    0.516    0.516
##     Professional           0.552    0.073    7.616    0.000    0.476    0.476
##     France                 0.415    0.074    5.579    0.000    0.333    0.333
##   Prestigious_brands ~~
##     Assortment             0.433    0.060    7.267    0.000    0.477    0.477
##     Stylish                0.477    0.064    7.438    0.000    0.565    0.565
##     Professional           0.342    0.044    7.744    0.000    0.531    0.531
##     France                 0.211    0.039    5.423    0.000    0.305    0.305
##   Assortment ~~
##     Stylish                0.814    0.088    9.249    0.000    0.521    0.521
##     Professional           0.717    0.075    9.576    0.000    0.602    0.602
##     France                 0.292    0.063    4.632    0.000    0.228    0.228
##   Stylish ~~
##     Professional           0.667    0.068    9.845    0.000    0.602    0.602
##     France                 0.389    0.068    5.721    0.000    0.327    0.327
##   Professional ~~
##     France                 0.336    0.051    6.602    0.000    0.370    0.370
##  .Repurchase ~~
##    .Cocreation            -0.015    0.034   -0.434    0.664   -0.020   -0.020
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Im3               4.995    0.056   88.622    0.000    4.995    3.786
##    .Im4               4.999    0.057   87.009    0.000    4.999    3.712
##    .Im5               5.036    0.057   87.765    0.000    5.036    3.787
##    .Im10              6.100    0.037  162.837    0.000    6.100    6.936
##    .Im14              6.138    0.037  165.572    0.000    6.138    7.093
##    .Im20              4.672    0.064   73.268    0.000    4.672    3.125
##    .Im21              5.139    0.058   88.093    0.000    5.139    3.750
##    .Im22              4.280    0.065   65.575    0.000    4.280    2.802
##    .Im11              5.653    0.049  115.355    0.000    5.653    4.944
##    .Im12              5.665    0.049  116.260    0.000    5.665    4.986
##    .Im13              5.448    0.052  105.700    0.000    5.448    4.527
##    .Im1               4.792    0.057   84.272    0.000    4.792    3.600
##    .Im2               4.858    0.055   88.352    0.000    4.858    3.781
##    .Im17              5.025    0.053   94.433    0.000    5.025    4.042
##    .Im18              4.595    0.060   76.160    0.000    4.595    3.287
```

```
##    .Im16              5.135    0.052   99.250    0.000    5.135    4.270
##    .Im19              5.145    0.048  106.953    0.000    5.145    4.576
##    .Im6               5.828    0.051  114.014    0.000    5.828    4.858
##    .Im7               5.754    0.052  110.958    0.000    5.754    4.756
##    .SAT_1             5.343    0.044  122.780    0.000    5.343    5.239
##    .SAT_2             5.482    0.043  127.736    0.000    5.482    5.455
##    .SAT_3             5.458    0.050  109.045    0.000    5.458    4.774
##    .COM_A1            4.287    0.062   69.635    0.000    4.287    2.983
##    .COM_A2            3.887    0.069   56.723    0.000    3.887    2.420
##    .COM_A3            3.543    0.070   50.824    0.000    3.543    2.178
##    .COM_A4            3.456    0.074   46.674    0.000    3.456    1.991
##    .C_REP1            4.283    0.031  136.245    0.000    4.283    5.859
##    .C_REP2            4.507    0.027  167.452    0.000    4.507    7.250
##    .C_REP3            4.677    0.024  193.058    0.000    4.677    8.445
##    .C_CR1             2.679    0.083   32.267    0.000    2.679    1.375
##    .C_CR3             3.261    0.088   37.085    0.000    3.261    1.572
##    .C_CR4             2.786    0.084   33.126    0.000    2.786    1.405
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Im3               0.214    0.042    5.082    0.000    0.214    0.123
##    .Im4               0.108    0.031    3.507    0.000    0.108    0.060
##    .Im5               0.747    0.066   11.357    0.000    0.747    0.422
##    .Im10              0.118    0.029    4.073    0.000    0.118    0.153
##    .Im14              0.066    0.022    3.010    0.003    0.066    0.088
##    .Im20              0.644    0.076    8.530    0.000    0.644    0.288
##    .Im21              0.708    0.093    7.634    0.000    0.708    0.377
##    .Im22              0.557    0.076    7.288    0.000    0.557    0.239
##    .Im11              0.817    0.091    8.950    0.000    0.817    0.625
##    .Im12              0.309    0.055    5.592    0.000    0.309    0.240
##    .Im13              0.389    0.055    7.021    0.000    0.389    0.269
##    .Im1               0.090    0.052    1.716    0.086    0.090    0.051
##    .Im2               0.302    0.051    5.914    0.000    0.302    0.183
##    .Im17              0.092    0.046    1.990    0.047    0.092    0.060
##    .Im18              0.524    0.088    5.965    0.000    0.524    0.268
##    .Im16              0.602    0.071    8.415    0.000    0.602    0.416
##    .Im19              0.345    0.052    6.650    0.000    0.345    0.273
##    .Im6               0.466    0.067    6.998    0.000    0.466    0.324
##    .Im7               0.158    0.072    2.198    0.028    0.158    0.108
##    .SAT_1             0.262    0.038    6.976    0.000    0.262    0.252
##    .SAT_2             0.332    0.061    5.485    0.000    0.332    0.329
##    .SAT_3             0.798    0.165    4.832    0.000    0.798    0.610
##    .COM_A1            0.757    0.074   10.270    0.000    0.757    0.366
##    .COM_A2            0.779    0.084    9.326    0.000    0.779    0.302
##    .COM_A3            0.880    0.079   11.170    0.000    0.880    0.333
##    .COM_A4            0.875    0.080   10.987    0.000    0.875    0.290
##    .C_REP1            0.179    0.027    6.733    0.000    0.179    0.334
##    .C_REP2            0.051    0.012    4.160    0.000    0.051    0.133
##    .C_REP3            0.131    0.012   10.620    0.000    0.131    0.428
##    .C_CR1             1.047    0.144    7.285    0.000    1.047    0.276
##    .C_CR3             1.369    0.192    7.125    0.000    1.369    0.318
##    .C_CR4             1.378    0.204    6.766    0.000    1.378    0.351
##     decoration        1.526    0.105   14.501    0.000    1.000    1.000
##     Haute_cuisine     0.655    0.066    9.882    0.000    1.000    1.000
##     Ambiance          1.591    0.138   11.533    0.000    1.000    1.000
##     Prestigs_brnds    0.491    0.088    5.558    0.000    1.000    1.000
##     Assortment        1.682    0.114   14.718    0.000    1.000    1.000
##     Stylish           1.453    0.116   12.551    0.000    1.000    1.000
##     Professional      0.845    0.101    8.367    0.000    1.000    1.000
##     France            0.974    0.111    8.760    0.000    1.000    1.000
##    .Satisfaction      0.449    0.063    7.166    0.000    0.576    0.576
##    .Commitment        0.862    0.088    9.757    0.000    0.659    0.659
```

```
##    .Repurchase         0.237    0.025    9.638    0.000    0.667    0.667
##    .Cocreation         2.280    0.220   10.373    0.000    0.829    0.829
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     ae               -0.023    0.012   -1.972    0.049   -0.048   -0.048
##     af                0.017    0.017    1.003    0.316    0.023    0.023
##     ag                0.011    0.010    1.134    0.257    0.024    0.024
##     ah               -0.008    0.021   -0.394    0.694   -0.010   -0.010
##     ai                0.029    0.013    2.212    0.027    0.063    0.063
##     aj                0.002    0.013    0.130    0.896    0.003    0.003
##     ak                0.099    0.033    2.945    0.003    0.152    0.152
##     al                0.022    0.012    1.860    0.063    0.037    0.037
##     cm               -0.004    0.011   -0.413    0.679   -0.009   -0.009
##     cn                0.005    0.017    0.308    0.758    0.007    0.007
##     co                0.069    0.015    4.475    0.000    0.145    0.145
##     cp               -0.035    0.022   -1.564    0.118   -0.041   -0.041
##     cq                0.019    0.010    1.777    0.076    0.040    0.040
##     cr               -0.003    0.012   -0.260    0.795   -0.007   -0.007
##     cs                0.030    0.024    1.217    0.223    0.045    0.045
##     ct                0.041    0.014    2.932    0.003    0.068    0.068
##     be                0.039    0.025    1.564    0.118    0.029    0.029
##     bf               -0.029    0.031   -0.943    0.346   -0.014   -0.014
##     bg               -0.019    0.018   -1.009    0.313   -0.014   -0.014
##     bh                0.014    0.034    0.406    0.685    0.006    0.006
##     bi               -0.048    0.023   -2.066    0.039   -0.038   -0.038
##     bj               -0.003    0.022   -0.131    0.896   -0.002   -0.002
##     bk               -0.164    0.078   -2.098    0.036   -0.091   -0.091
##     bl               -0.037    0.024   -1.532    0.126   -0.022   -0.022
##     dm               -0.013    0.032   -0.410    0.682   -0.010   -0.010
##     dn                0.015    0.049    0.307    0.759    0.007    0.007
##     do                0.204    0.046    4.425    0.000    0.155    0.155
##     dp               -0.102    0.066   -1.562    0.118   -0.043   -0.043
##     dq                0.055    0.031    1.787    0.074    0.043    0.043
##     dr               -0.010    0.037   -0.260    0.795   -0.007   -0.007
##     ds                0.087    0.073    1.204    0.229    0.048    0.048
##     dt                0.122    0.041    2.964    0.003    0.072    0.072
##     te1c             -0.006    0.096   -0.057    0.954   -0.004   -0.004
##     te2c             -0.093    0.152   -0.613    0.540   -0.045   -0.045
##     te3c              0.337    0.088    3.845    0.000    0.256    0.256
##     te4c              0.108    0.163    0.662    0.508    0.046    0.046
##     te5c              0.001    0.089    0.011    0.992    0.001    0.001
##     te6c              0.010    0.096    0.101    0.920    0.007    0.007
##     te7c             -0.253    0.181   -1.395    0.163   -0.140   -0.140
##     te8c             -0.042    0.111   -0.377    0.706   -0.025   -0.025
##     te1r             -0.018    0.029   -0.614    0.539   -0.037   -0.037
##     te2r              0.060    0.058    1.042    0.297    0.082    0.082
##     te3r              0.120    0.032    3.700    0.000    0.254    0.254
##     te4r              0.034    0.065    0.524    0.600    0.040    0.040
##     te5r              0.031    0.030    1.031    0.303    0.067    0.067
##     te6r             -0.012    0.038   -0.326    0.745   -0.025   -0.025
##     te7r              0.091    0.059    1.556    0.120    0.141    0.141
##     te8r              0.029    0.035    0.841    0.400    0.049    0.049
##     ti1r             -0.028    0.018   -1.515    0.130   -0.057   -0.057
##     ti2r              0.022    0.028    0.790    0.429    0.030    0.030
##     ti3r              0.080    0.020    3.940    0.000    0.169    0.169
##     ti4r             -0.043    0.037   -1.159    0.246   -0.050   -0.050
##     ti5r              0.047    0.019    2.448    0.014    0.103    0.103
##     ti6r             -0.002    0.022   -0.068    0.946   -0.003   -0.003
##     ti7r              0.128    0.049    2.625    0.009    0.197    0.197
##     ti8r              0.063    0.021    3.016    0.003    0.104    0.104
##     ti1c              0.026    0.035    0.725    0.468    0.019    0.019
```

```
##     ti2c              -0.014    0.047    -0.291    0.771    -0.007    -0.007
##     ti3c               0.185    0.043     4.282    0.000     0.141     0.141
##     ti4c              -0.089    0.057    -1.568    0.117    -0.037    -0.037
##     ti5c               0.007    0.032     0.220    0.826     0.006     0.006
##     ti6c              -0.012    0.031    -0.401    0.689    -0.009    -0.009
##     ti7c              -0.077    0.086    -0.887    0.375    -0.043    -0.043
##     ti8c               0.085    0.039     2.151    0.032     0.051     0.051
```

```
semPaths(fit2, nCharNodes = 0, style = "lisrel", rotation = 2)
```



Question 2

First, let's check if the mechanism driving satisfaction and affective commitment are similar. For this, we can take a look at the regression part, particularly at results of the factors on the two mediators. We can mention that not all the p-value are smaller than 0.05 which means that we cannot reject all the null hypothesis stating that there is an absence of a relationship between variables. Let's take a look at the significant factors for each mediator:

```
Regression:
                      Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  Satisfaction ~
    decoratin  (e)      -0.109    0.048   -2.285    0.022   -0.152   -0.152
    Assortmnt  (i)       0.134    0.053    2.512    0.012    0.197    0.197
    Professnl  (k)       0.459    0.105    4.382    0.000    0.479    0.479
  Commitment ~
    Ambiance   (o)       0.373    0.059    6.359    0.000    0.411    0.411
    France     (t)       0.223    0.067    3.327    0.001    0.192    0.192
```

We can see that it's not the same mechanism that drive satisfaction and affective commitment as it's not the same factors that are significant. For Satisfaction, the main driver is professional as it has the higher standardized estimate. We can also mention that decoration has a negative value which means it has a negative impact on satisfaction. For commitment, the main driver is ambiance as it has the higher standardized estimate.

We suggest to the galleries lafayette to have a large choice of assortment (Assortment factor). Additionally, they should select an organized and professional staff (profesional factor) to increase the satisfaction of the customers. To drive Commitment they should create a relaxing and intimate atmosphere that should be reflecting the french way of life. However, it seems that a too creative assortment of the shop, tend to drive down the Satisfaction of the customers. Maybe they are looking for something more simple regarding the assortment

Then, we want to know if satisfaction and affective commitment mediate the impact of image perceptions on outcomes. To know if satisfaction and affective commitment mediate the impact of image perceptions on outcomes, we should take a look at the regression part:

```
Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  Repurchase ~
    Satisfctn  (a)    0.215    0.049    4.396    0.000    0.318    0.318
    Commitmnt  (c)    0.184    0.031    5.882    0.000    0.354    0.354
  Cocreation ~
    Satisfctn  (b)   -0.357    0.143   -2.501    0.012   -0.190   -0.190
    Commitmnt  (d)    0.546    0.094    5.824    0.000    0.377    0.377
```

Here, the first thing to notice is that all the p-values are smaller than 0.05. This means we can reject the null hypothesis stating that there is an absence of a relationship between variables. This means that satisfaction and affective commitment have a significant impact on Repurchase and Commitment.

For repurchase intention, the two standardized estimates are higher than zero and close to 0.3. We can mention that it is higher for commitment which means that affective commitment is a bit more driving the repurchase intention compared to satisfaction.

For Cocreation intention, the standardized estimated of commitment is positive while for satisfaction it is negative. This means that satisfaction has a negative impact on Cocreation. In other words, the higher the customer's satisfaction with Galeries Lafayette, the less inclined they are to participate in its development, perhaps because they believe there is little room for improvement. On the other hand, customers who are deeply attached to Galeries Lafayette are more eager to contribute to its improvement.

If we want to know which Images have been mediated, we should take a look at this table (after deleting >0.05):

```
Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all

Satisfaction serves as a mediator for the different influences on Repurchase:
    ae               -0.023    0.012   -1.972    0.049   -0.048   -0.048 (mediator of decoration)
    ai                0.029    0.013    2.212    0.027    0.063    0.063 (mediator of assortment)
    ak                0.099    0.033    2.945    0.003    0.152    0.152 (mediator of professional)


Commitment serves as a mediator for the different influences on Repurchase:
    co                0.069    0.015    4.475    0.000    0.145    0.145 (mediator of ambiance)
    ct                0.041    0.014    2.932    0.003    0.068    0.068 (mediator of france)


Satisfaction serves as a mediator for the different influences on Cocreation:
    bi               -0.048    0.023   -2.066    0.039   -0.038   -0.038 (mediator of assortment)
    bk               -0.164    0.078   -2.098    0.036   -0.091   -0.091 (mediator of professional)

Commitment serves as a mediator for the different influences on Cocreation:
    do                0.204    0.046    4.425    0.000    0.155    0.155 (mediator of ambiance)
    dt                0.122    0.041    2.964    0.003    0.072    0.072 (mediator of france)
```

For example, we can see that satisfaction mediates the effect of assortment on Repurchase intention. This means that the more the customer is satisfied with Assortment, total satisfaction would be higher which lead to high likelihood of repurchase behavior intention. We can do the same for each line to understand how satisfaction or commitment mediates each factors for each outcomes.

We can conclude that both mediators mediate the impact of image perceptions on the two outcomes (Repurchase and cocreation)

Question 3

Now to understand what is driving the two distinct outcomes (repurchase and co-creation) we can rank the image dimensions with respect to the total effect which is equal to the sum of the indirect effect and the direct effect:

```
Example for Factor 1 on Repurchase:

Factor 1 -> Repurchase (DIRECT EFFECT)
+
Factor 1 -> Satisfaction -> Repurchase (INDIRECT EFFECT MEDIATOR 1)
+
Factor 1 -> Commitment -> Repruchase (INDIRECT EFFECT MEDIATOR 2)
=
TOTAL EFFECT
```

We can look at this table:

```
Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all

Total effect factor ... on Cocreation:
    te3c              0.337    0.088    3.845    0.000    0.256    0.256 (Total effect of Ambiance on Cocreation)

Total effect factor ... on Repurchase:
    te3r              0.120    0.032    3.700    0.000    0.254    0.254 (Total effect of Ambiance on Repurchase)
```

We can see that Ambiance is the only factor that has a significant total effect. The total effect are quit the same for cocreation and repurchase.

We suggest to the galleries lafayettes to have a nice ambiance in order to increase the repurchase and cocreation intention.