Perceptual Mapping

Marketing Analytics

Professor Kamel Jedidi
Columbia University

Overview

- Marketing creates value
  - Makes products desirable by endowing them with awareness, beliefs, associations, and status
  - The brand captures these intangible values/perceptions and differentiates the product from others

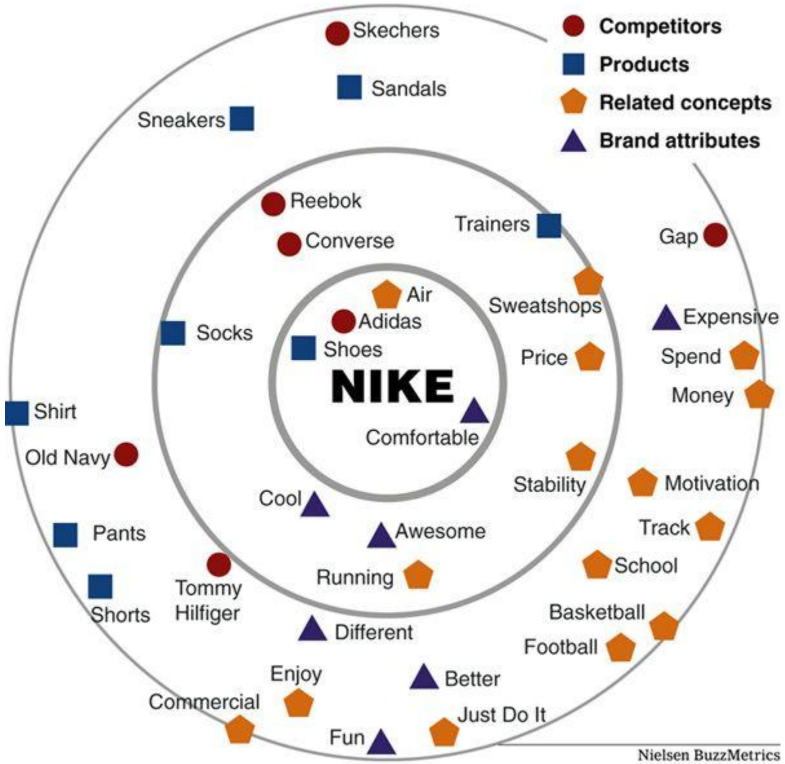
- Managers need to know how their products are perceived by consumers so that they can (re)position them appropriately

Outline

- Goal: discuss perceptual mapping methods for measuring and portraying how consumers perceive brands. We will discuss:
  - Perceptual mapping examples
  - Factor analysis
  - An application

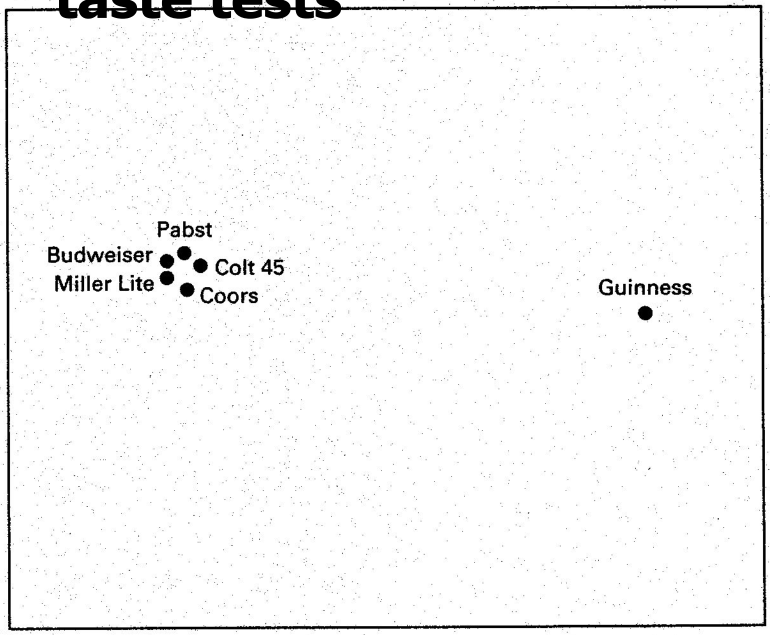
A perceptual map provides a picture of how consumers perceive different competitors

A brand association map drawn from online commentary about Nike



Consumer perception of beers without revealing brand names (blind taste test)

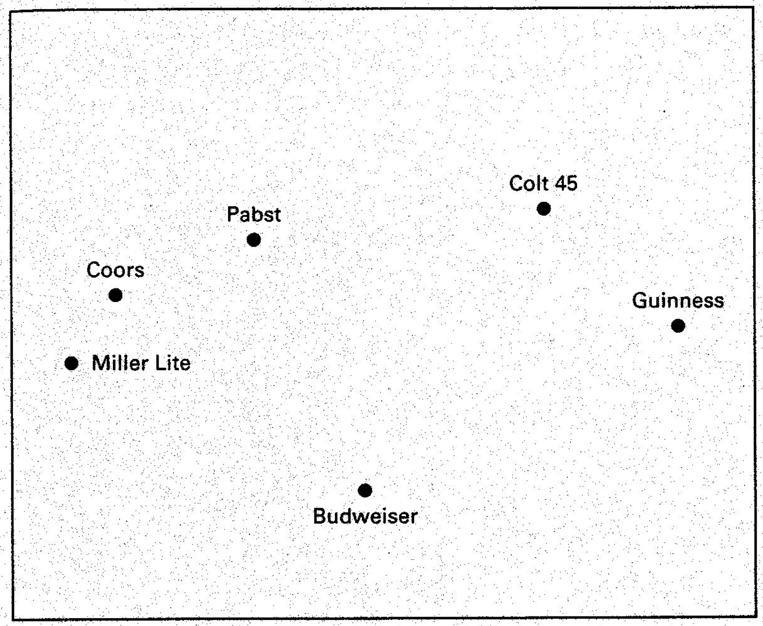
Results of "blind" beer taste tests



B. Taste Perceptions of Six Beer Brands When the Drinker Does **Not** Know What He Is Drinking

Consumer perception of beer brands (brand names revealed)

Results of beer taste tests



A. Taste Perceptions of Six Beer Brands When the Drinker Knows What He Is Drinking

What is Factor Analysis

- A data-analytic method that is used for:
  - Data reduction
  - Understanding the structure underlying the data
  - Constructing perceptual maps for product positioning
  - Dealing with multicolinearity

Factor Analysis intuition

Summarizing students' test scores

Student	Grammar	Spelling	Composition	Algebra	Geometry	Trigonometry
1	90	85	88	56	60	49
2	23	33	31	90	78	79
3	65	59	49	60	58	55
4	88	92	94	90	93	98
5	22	19	33	45	34	12
6	82	78	79	85	79	90
7	96	95	94	45	67	53
8	43	36	45	89	93	87
9	66	76	45	34	44	40
10	70	80	75	60	65	70

How would you summarize such data?

A factor is a group of variables measuring something in common

	Grammar	Spelling	Composition	Algebra	Geometry	Trigonometry
Grammar	1.00					
Spelling	0.97	1.00				
Composition	0.92	0.89	1.00			
Algebra	-0.14	-0.16	0.04	1.00		
Geometry	0.21	0.19	0.34	0.88	1.00	
Trigonometry	0.25	0.27	0.34	0.87	0.95	1.00

A factor is a group of variables that are highly correlated.

Factor Analysis intuition

Grouping variables (Factor Loading Matrix)

Variable	Verbal Ability Factor 1	Math. Ability Factor 2
Grammar	X	
Spelling	X	
Composition	X	
Algebra		X
Geometry		X
Trigonometry		X

Factor scores

Factor1 = 1/3 Gram +1/3 Spel + 1/3 Comp + 0 Alg + 0 Geom + 0 Trig
= w1 Gram + w2 Spel + w3 Comp + w4 Alg + w5 Geom + w6 Trig
= Verbal Ability Score

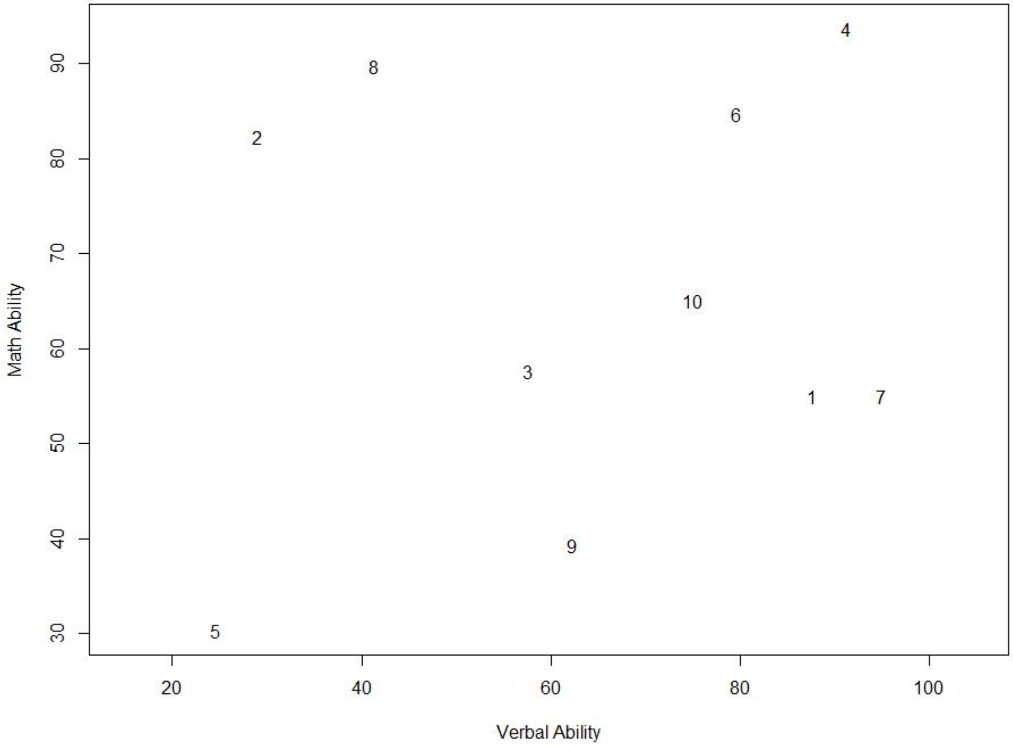
Factor2 = 0 Gram + 0 Spel + 0 Comp + 1/3 Alg + 1/3 Geom + 1/3 Trig
= Mathematical Ability Score

- A factor is a weighted average of the variables.
- Select weights to summarize data in the best way possible.

Summarized data—factor scores

							Verbal Ability	Math. Ability
Student	Grammar	Spelling	Composition	Algebra	Geometry	Trigonometry	Factor1	Factor2
1	90	85	88	56	60	49	87.67	55.00
2	23	33	31	90	78	79	29.00	82.33
3	65	59	49	60	58	55	57.67	57.67
4	88	92	94	90	93	98	91.33	93.67
5	22	19	33	45	34	12	24.67	30.33
6	82	78	79	85	79	90	79.67	84.67
7	96	95	94	45	67	53	95.00	55.00
8	43	36	45	89	93	87	41.33	89.67
9	66	76	45	34	44	40	62.33	39.33
10	70	80	75	60	65	70	75.00	65.00

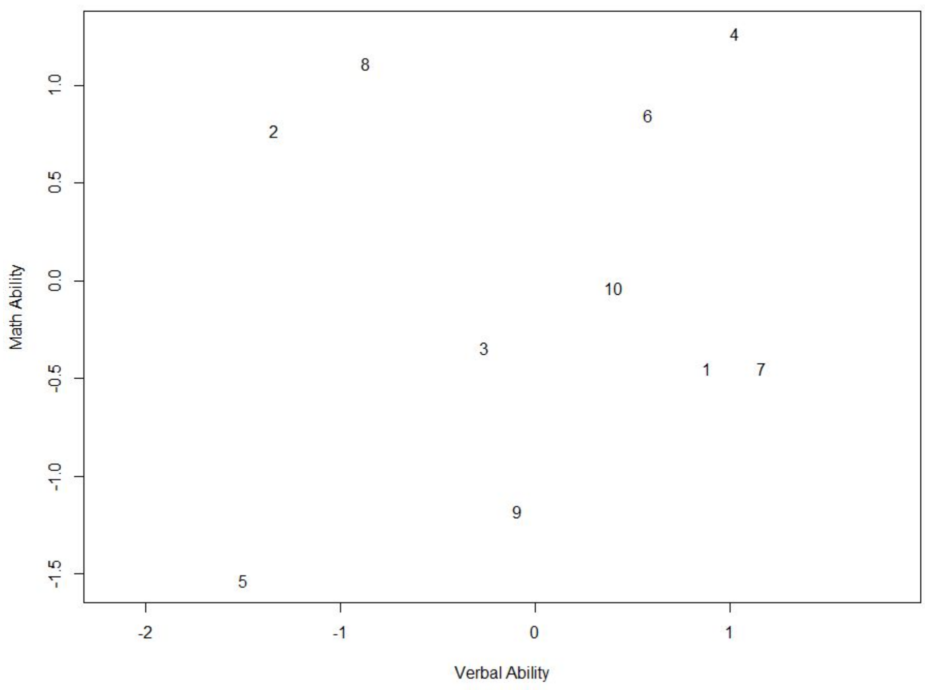
A summary map of student test scores



Standardized test scores data

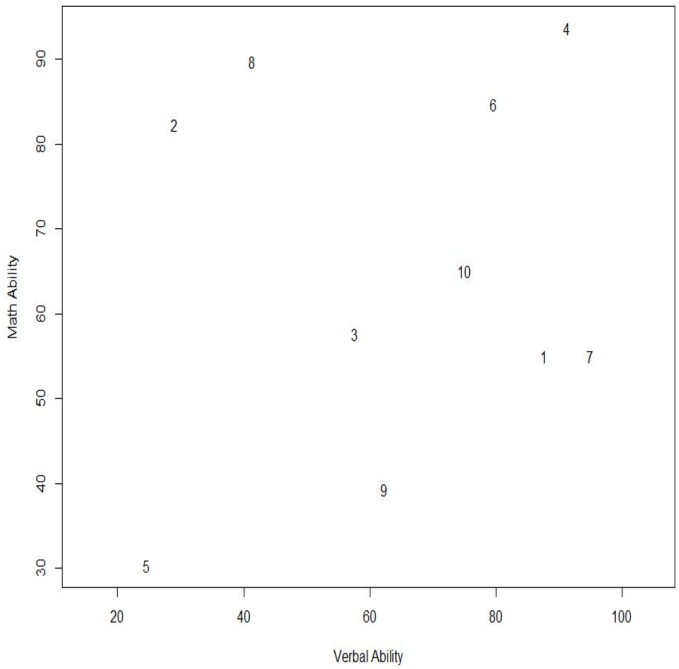
							Verbal Ability	Math Ability
Student	Grammar	Spelling	Composition	Algebra	Geometry	Trigonometry	Factor 1	Factor 2
1	0.95	0.73	0.98	-0.44	-0.37	-0.54	0.89	-0.45
2	-1.54	-1.20	-1.28	1.15	0.56	0.59	-1.34	0.77
3	0.02	-0.23	-0.57	-0.25	-0.47	-0.31	-0.26	-0.35
4	0.87	0.99	1.22	1.15	1.34	1.31	1.03	1.27
5	-1.58	-1.71	-1.20	-0.95	-1.71	-1.94	-1.50	-1.53
6	0.65	0.47	0.62	0.92	0.62	1.01	0.58	0.85
7	1.17	1.10	1.22	-0.95	-0.01	-0.39	1.16	-0.45
8	-0.80	-1.09	-0.73	1.10	1.34	0.89	-0.87	1.11
9	0.06	0.40	-0.73	-1.47	-1.19	-0.88	-0.09	-1.18
10	0.20	0.54	0.46	-0.25	-0.11	0.25	0.40	-0.04

A summary map of standardized test scores

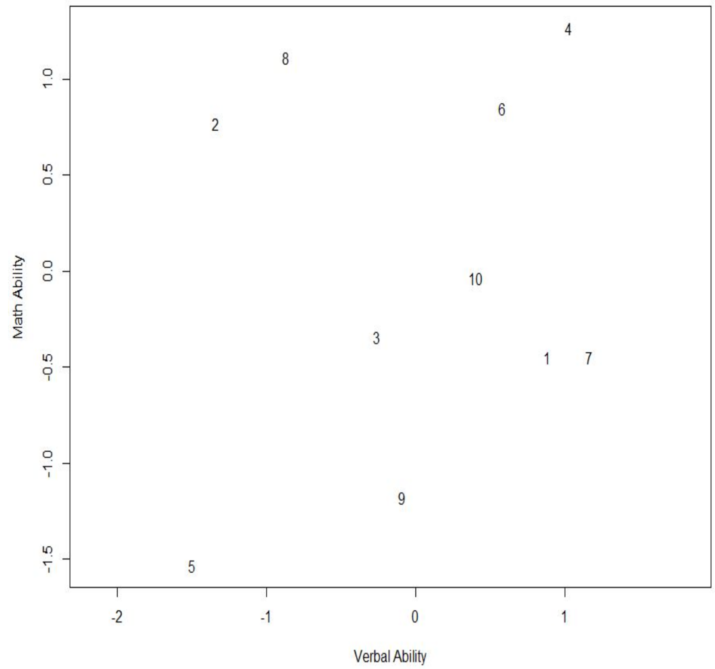


Scaled and unscaled maps

Non standardized data



Standardized data



Summarizing data with Factor Analysis using R

- There are two analysis decisions in factor analysis:
  - What’s the number of factors?
  - What’s the factor structure?

- In R
  - Use nScree(input data, cor=TRUE) to determine the number of factors to retain
  - Use principal(input data, nfactors=n, rotate="varimax") to obtain the solution for n factors (say n=2)

Determining the number of factors

```
library(nFactors)
nScree(test_scores[, 2:7], cor=TRUE)
```

```
  noc naf nparallel nkaiser
1   2   2         2       2
```

- The maximal number of factors is the number of variables
- All four methods (Optimal Coordinates, Acceleration Factor, Parallel, Kaiser) suggest n=2 factors

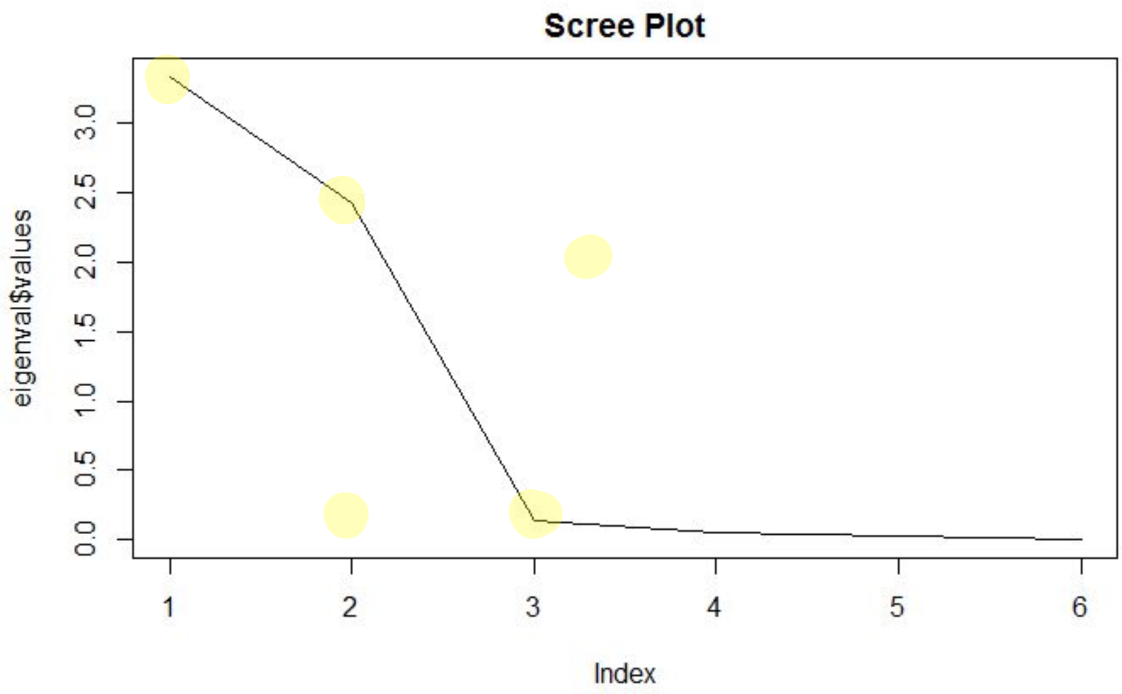
Numbers of factors to retain: Eigenvalue greater than 1

```
eigenval=eigen(cor(test_scores[,2:7]))
eigenval$values
```

```
[1] 3.334873852 2.431236985 0.137298334 0.057348762 0.031841202 0.007400866
```

Numbers of factors to retain: Elbow in the scree plot

```
plot(eigenval$values, main="Scree Plot", type="l")
```



Numbers of factors to retain: Cumulative variance explained

```
library(psych)
fit <- principal(test_scores[, 2:7],nfactors=5, rotate="none")
fit$loadings #print results
```

Cumulative Variance Explained

	PC1	PC2	PC3	PC4	PC5
SS loadings	3.335	2.431	0.137	0.057	0.032
Proportion Var	0.556	0.405	0.023	0.010	0.005
Cumulative Var	0.556	0.961	0.984	0.993	0.999

Factor structure: The factor loading matrix

```r
library(psych)
fit <- principal(test_scores[, 2:7],nfactors=2, rotate="varimax")
```

```r
fit$loadings # print the factor loading results
```

```
             RC1    RC2
Grammar       0.989
Spelling      0.981
Composition   0.947  0.174
Algebra      -0.166  0.969
Geometry      0.184  0.965
Trigonometry  0.227  0.956
```

This matrix gives the correlations between each of the variables and the derived factors.

Interpreting the factors

- Examine the factor loadings and identify the significant loadings (> 0.5) in each row/column

- Label the factor by finding a collective name to describe the items most strongly associated with this factor

Summarizing data with Factor Analysis
The estimated factor weights

```
colnames(fit$weights) = c("Verbal", "Math")
fit$weights # print the factor weights
```

```
                  Verbal        Math
Grammar       0.34191828 -0.04456066
Spelling      0.33963570 -0.04646619
Composition   0.31925112  0.01117554
Algebra      -0.11086057  0.36157268
Geometry      0.01083039  0.34104688
Trigonometry  0.02604433  0.33554399
```

Verbal Score= .342Gram+.340Spel + .319Comp – .111Alg+.010Geom+.026Trig

Math Score = -.044Gram+.046Spel + .011Comp + .361Alg+.341Geom+.335Trig

Summarizing data with Factor Analysis

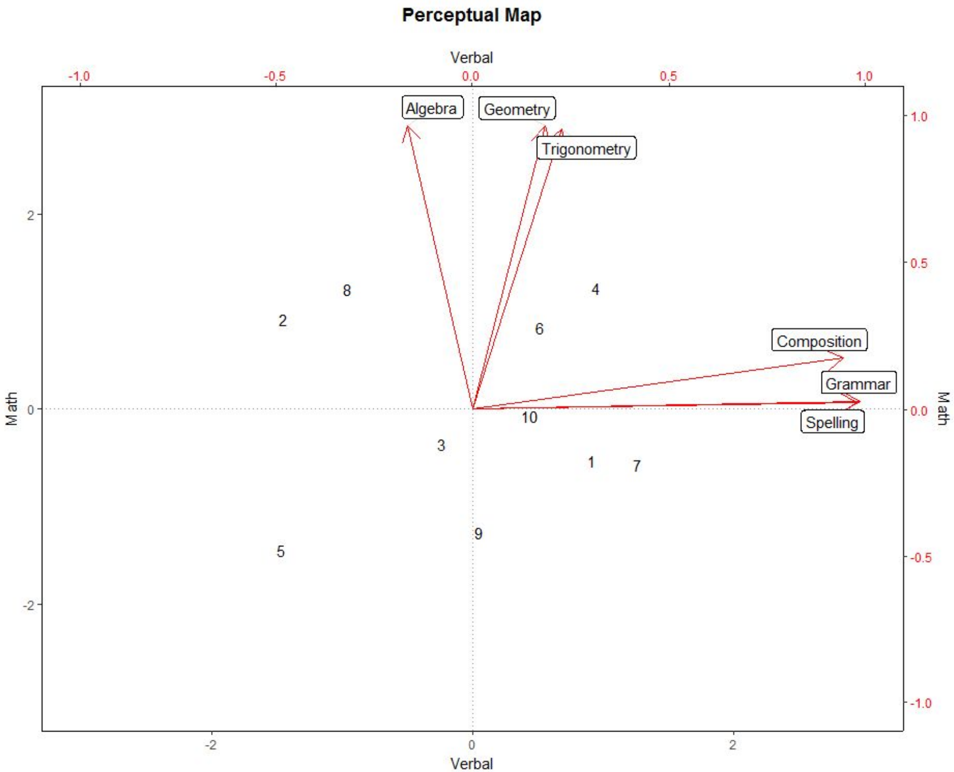
The estimated factor scores

```
colnames(fit$scores) = c("Verbal", "Math")
fit$scores #print the factor scores
```

```
         Verbal       Math
 [1,]  0.9153038 -0.5302432
 [2,] -1.4485777  0.9167256
 [3,] -0.2393435 -0.3530964
 [4,]  0.9443561  1.2404663
 [5,] -1.4692481 -1.4411465
 [6,]  0.5122126  0.8352546
 [7,]  1.2581872 -0.5666709
 [8,] -0.9580358  1.2334432
 [9,]  0.0487361 -1.2620155
[10,]  0.4364093 -0.0727173
```

Verbal Ability	Math Ability
Factor 1	**Factor 2**
0.89	-0.45
-1.34	0.77
-0.26	-0.35
1.03	1.27
-1.50	-1.53
0.58	0.85
1.16	-0.45
-0.87	1.11
-0.09	-1.18
0.40	-0.04

Putting it all together



Red: Factor loadings
Black: Factor scores

Factor Analysis
Definitions of terms

- Eigenvalue is the variance explained by the factor. A factor should have an eigenvalue greater than one to be retained.

- Factor loading is the correlation between a variable (an attribute) and a factor. Variables that have high loadings with a specific factor are used to name the factor.

- Factor score is a weighted sum of the variables. The weights are regression-like coefficients which are output by the computer program.

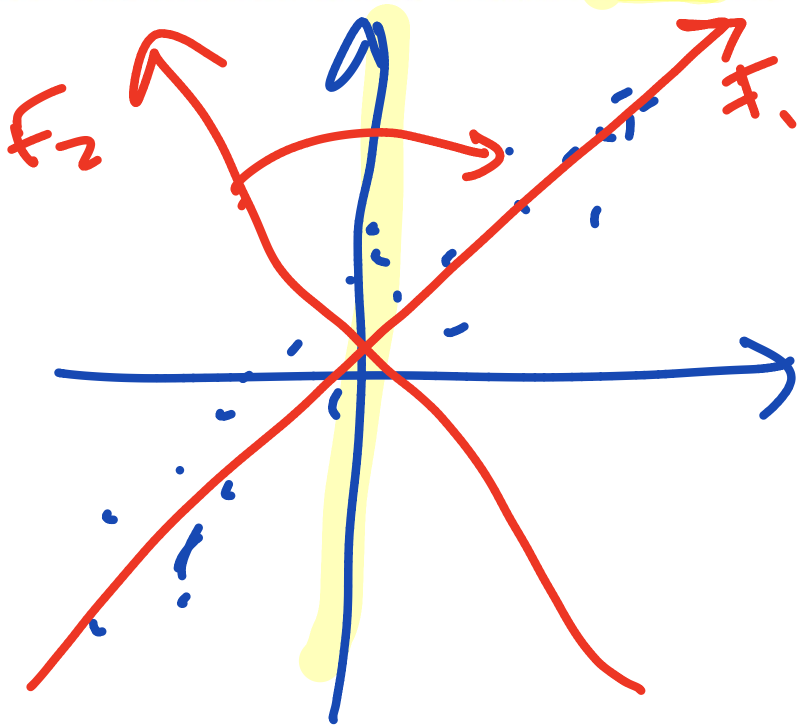
Other issues:

The unrotated loading matrix

```
fit <- principal(test_scores[, 2:7], nfactors=2, rotate="none")
fit$loadings # print results
```

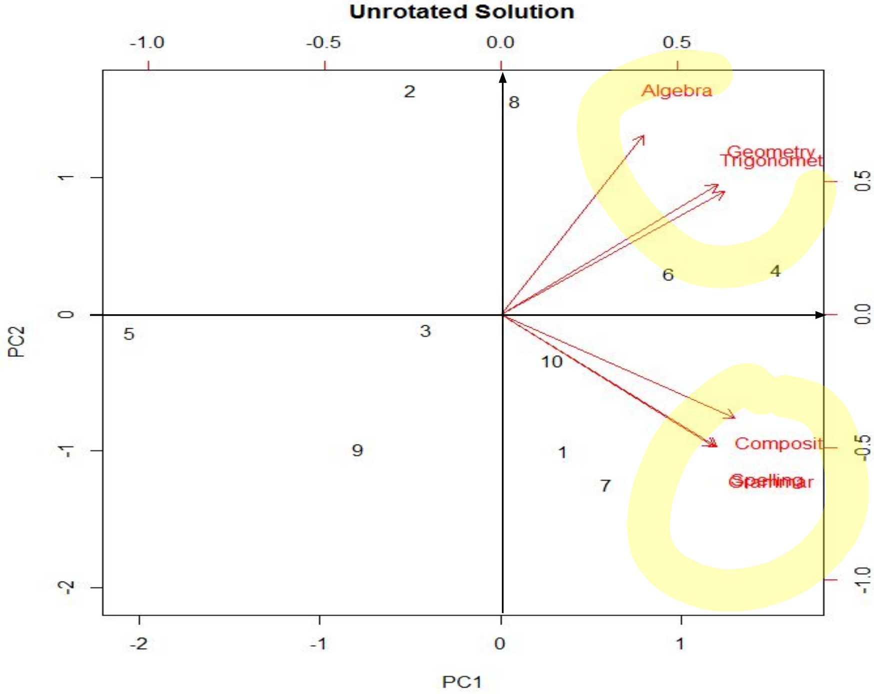
Loadings:

	PC1	PC2
Grammar	0.767	-0.624
Spelling	0.758	-0.624
Composition	0.831	-0.486
Algebra	0.506	0.842
Geometry	0.769	0.611
Trigonometry	0.796	0.577



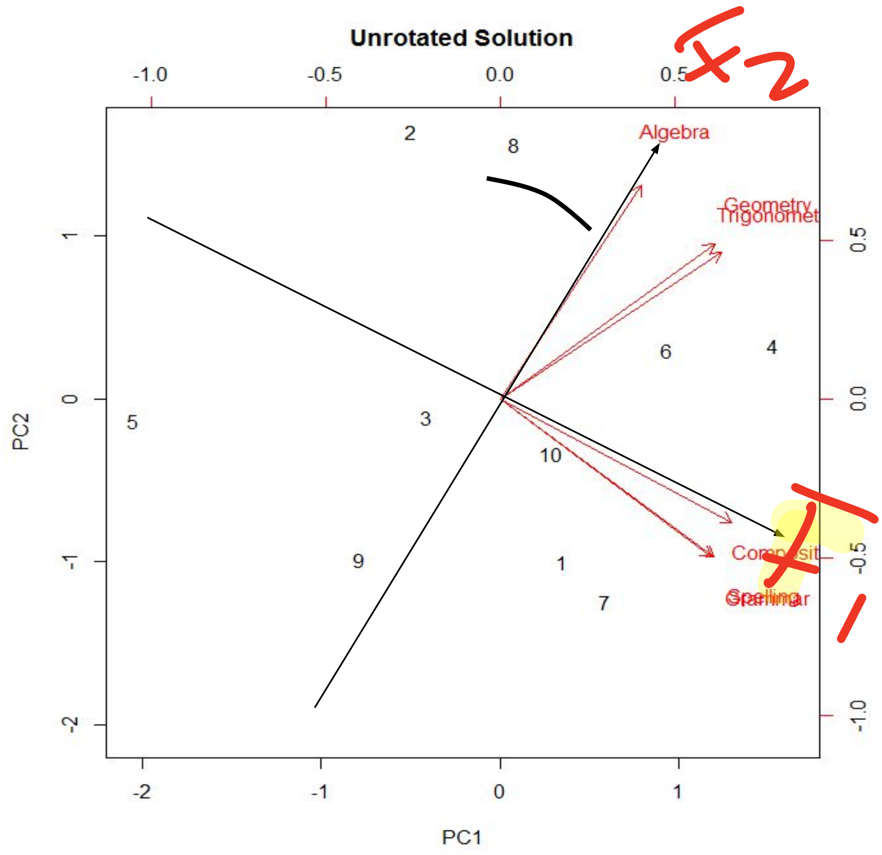
Other issues:
Plot of the unrotated solution

```
biplot(fit$scores, fit$loadings, xlabs=test_scores[,1],
       main = "Unrotated Solution\n")
```

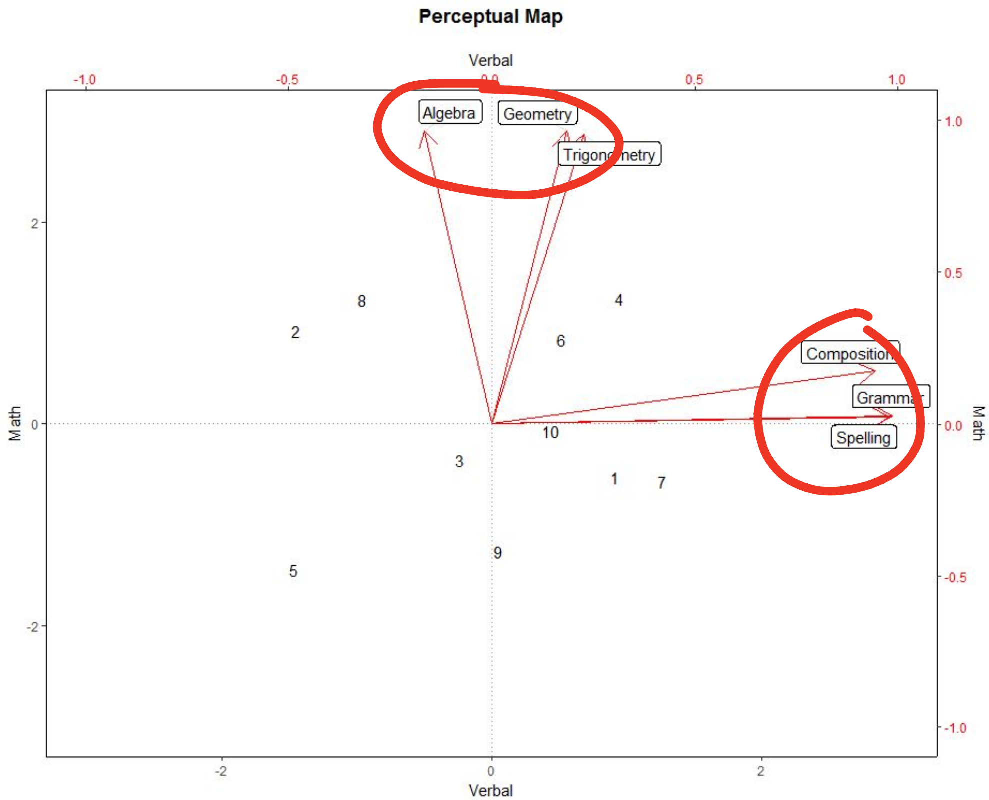


Other issues:
The Varimax rotation

Rotation is a transformation of the initial solution into a new solution which is easier to interpret



Rotated factor solution



Red: Factor loadings

Black: Factor scores

An application

- Attribute ratings data collected from 92 MBA and undergraduate students
- Six luxury sedans are included:
  - Acura, BMW, Cadillac, Lexus, Lincoln, Mercedes
- Each student rated each sedan on six attributes
  - Trendy/innovative, Styling, Reliability, Sportiness, Performance, Comfort
- Each respondent also provided overall preference for each luxury sedan

Data

ID	Education	Brand	Trendy	Styling	Sportiness	Reliability	Performance	Comfort	Preference
1	MBA	Acura	6	5	6	8	6	6	7
1	MBA	BMW	9	10	7	8	8	7	10
1	MBA	Cadillac	4	6	3	5	6	9	5
1	MBA	Lexus	6	7	2	9	7	8	8
1	MBA	Lincoln	4	5	1	5	5	9	4
1	MBA	Mercedes	8	9	5	8	9	8	9

92	Undergrad	Acura	7	8	9	7	7	7	6
92	Undergrad	BMW	8	9	8	8	8	8	9
92	Undergrad	Cadillac	6	4	5	9	9	10	8
92	Undergrad	Lexus	6	5	5	9	8	8	8
92	Undergrad	Lincoln	7	6	5	8	8	9	8
92	Undergrad	Mercedes	9	9	7	9	9	9	9

Correlations

Halo effect

	Trendy	*Styling*	*Sportiness*	*Reliability*	*Performance*	*Comfort*
Trendy	1.00					
Styling	0.83	1.00				
Sportiness	0.75	0.70	1.00			
Reliability	0.55	0.57	0.48	1.00		
Performance	0.67	0.68	0.59	0.69	1.00	
Comfort	0.35	0.38	0.17	0.46	0.50	1.00

Determining the number of factors

```
library(nFactors)
nScree(cardata[, 4:9], cor=TRUE) #This function help you determine the number of factors
```

```
  noc naf nparallel nkaiser
1   1   1         1       1
```

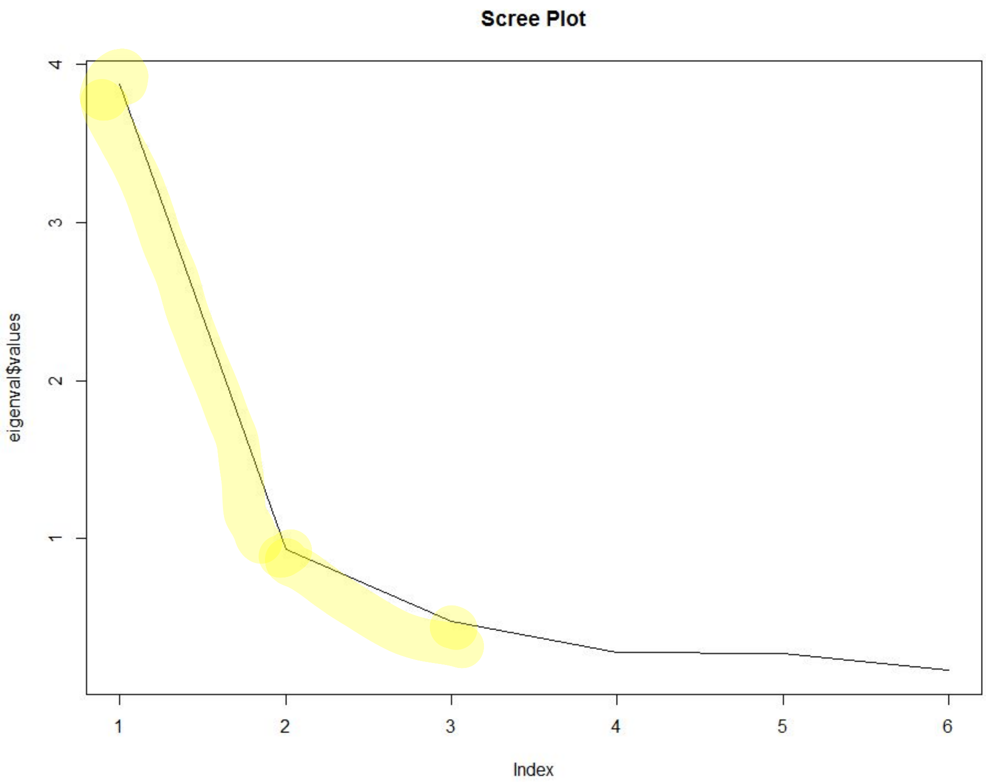
- All four methods (Optimal Coordinates, Acceleration Factor, Parallel, Kaiser) suggest n=1 factor

How many factors to retain?
The eigenvalue > 1 criterion

```
eigenval=eigen(cor(cardata[,4:9]))
eigenval$values
```

```
[1] 3.8726005 0.9347620 0.4751468 0.2803722 0.2715605 0.1655580
```

How many factors to retain?
The elbow in scree plot criterion



How many factors to retain? *Cumulative variance explained*

```r
fit <- principal(cardata[,4:9], nfactors=5, rotate="none")
fit$loadings
```

Cumulative Variance Explained

	PC1	PC2	PC3	PC4	PC5
SS loadings	3.873	0.935	0.475	0.280	0.272
Proportion Var	0.645	0.156	0.079	0.047	0.045
Cumulative Var	0.645	0.801	0.880	0.927	0.972

Interpreting the factors
Rotated solution

```
fit <- principal(cardata[,4:9], nfactors=3, rotate="varimax")
fit$loadings
```

```
Loadings:
              RC1    RC3    RC2
Trendy        0.881  0.249  0.209
Styling       0.839  0.281  0.261
Sportiness    0.870  0.271
Reliability   0.294  0.902  0.211
Performance   0.549  0.621  0.330
Comfort       0.126  0.234  0.951
```

Computing factor scores

```
colnames(fit$weights) = c("Appearance", "Performance", "Comfort")
fit$weights
```

	Appearance	Performance	Comfort
Trendy	0.451769686	-0.27497882	0.07367905
Styling	0.400831739	-0.23133003	0.12214951
Sportiness	0.443894424	-0.08208276	-0.27113948
Reliability	-0.350452175	1.06183977	-0.26457070
Performance	-0.009305856	0.41731508	0.02843823
Comfort	-0.099899402	-0.26153179	1.03312622

Summarized data

```
colnames(fit$scores) = c("Appearance", "Performance", "Comfort")
reduced_data = cbind(cardata[,1:3],fit$scores)
head(reduced_data)
```

	ID	Education	Brand	Appearance	Performance	Comfort
	<int>	<fctr>	<fctr>	<dbl>	<dbl>	<dbl>
1	1	MBA	Acura	-0.4633312	0.60738739	-1.3943949
2	1	MBA	BMW	1.1217549	-0.03578809	-0.4699704
3	1	MBA	Cadillac	-0.8329368	-1.35020394	1.2683913
4	1	MBA	Lexus	-1.1427352	0.98872908	0.3067639
5	1	MBA	Lincoln	-1.3701566	-1.39009793	1.4202708
6	1	MBA	Mercedes	0.3216445	0.30087651	0.3175364

6 rows

Hide

```
tail(reduced_data)
```

	ID	Education	Brand	Appearance	Performance	Comfort
	<int>	<fctr>	<fctr>	<dbl>	<dbl>	<dbl>
547	92	Undergrad	Acura	0.9322268	-0.43984547	-0.73622276
548	92	Undergrad	BMW	0.8692587	-0.01020726	-0.02844505
549	92	Undergrad	Cadillac	-1.2725592	1.29820884	1.11569928
550	92	Undergrad	Lexus	-0.9644658	1.30676132	-0.12019289
551	92	Undergrad	Lincoln	-0.4670737	0.35696647	0.74555472
552	92	Undergrad	Mercedes	0.6263164	0.52329852	0.62448733

6 rows

Average factor scores by brand

```
brand.mean=aggregate(reduced_data[,c(4:6)], by=list(Brand), FUN=mean, na.rm=TRUE)
brand.mean
```

Group.1 <fctr>	Appearance <dbl>	Performance <dbl>	Comfort <dbl>
Acura	0.15079326	-0.07427787	-0.70211543
BMW	0.94436691	0.26409425	-0.01513218
Cadillac	-0.56757751	-0.37379770	0.22053742
Lexus	0.09714827	0.40356450	0.23730124
Lincoln	-1.11480605	-0.32346058	-0.14560705
Mercedes	0.49007512	0.10387740	0.40501600

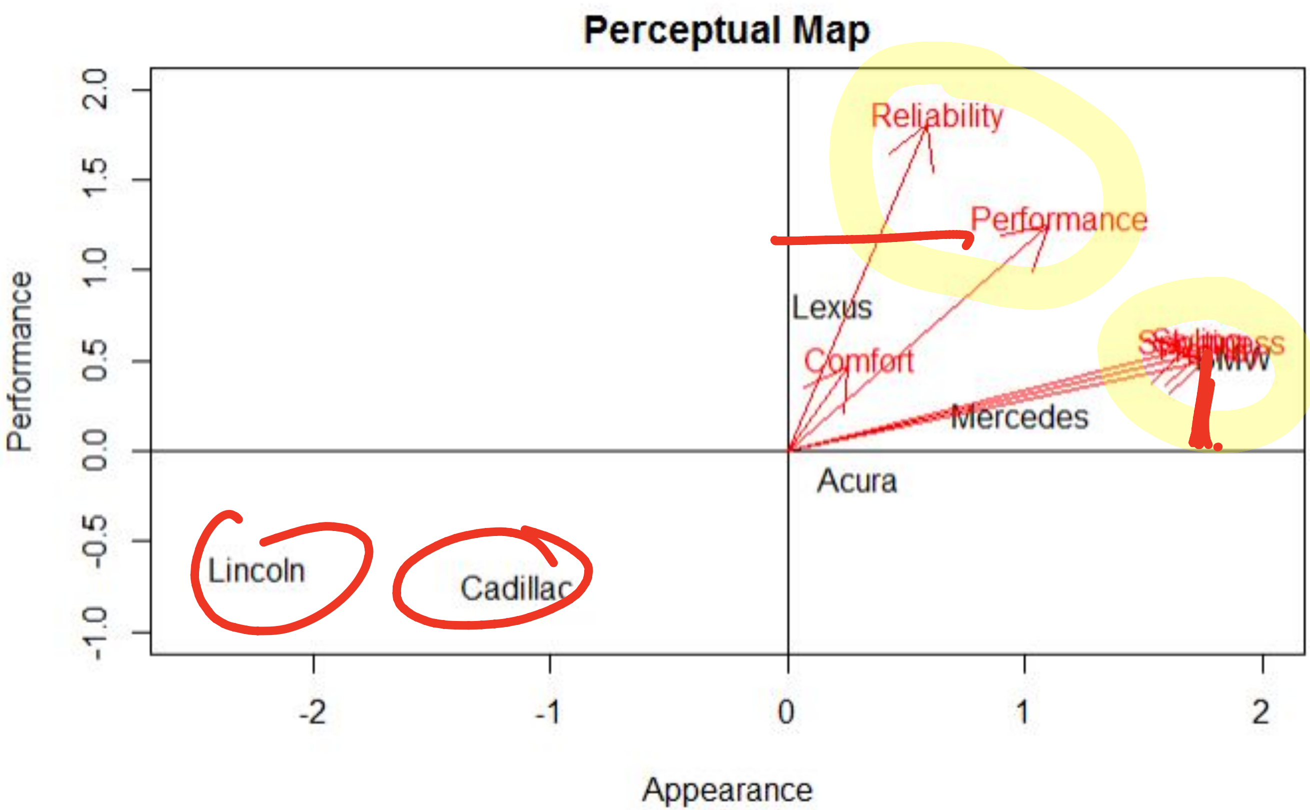
Perceptual map in two dimensions

```
plot(Appearance, Performance, col = 'white',
     xlim= c(-2.5,2), ylim = c(-1,2),
     main="Perceptual Map")
abline(h=0) #add a horizontal line in the graph
```

```
abline(v=0) #add a vertical line in the graph
text(Appearance*2, Performance*2, brand.mean[,1])
```

```
tmp <-matrix(fit$loadings, 6,3)
for(i in 1:dim(tmp)[1])
{arrows(x0 = 0,y0 = 0,x1 = tmp[i,1]*2, y1 = tmp[i,2]*2, col= 'red')}
text(tmp[,1]*2+0.05, tmp[,2]*2+.05,rownames(fit$loadings), col= 'red')
```

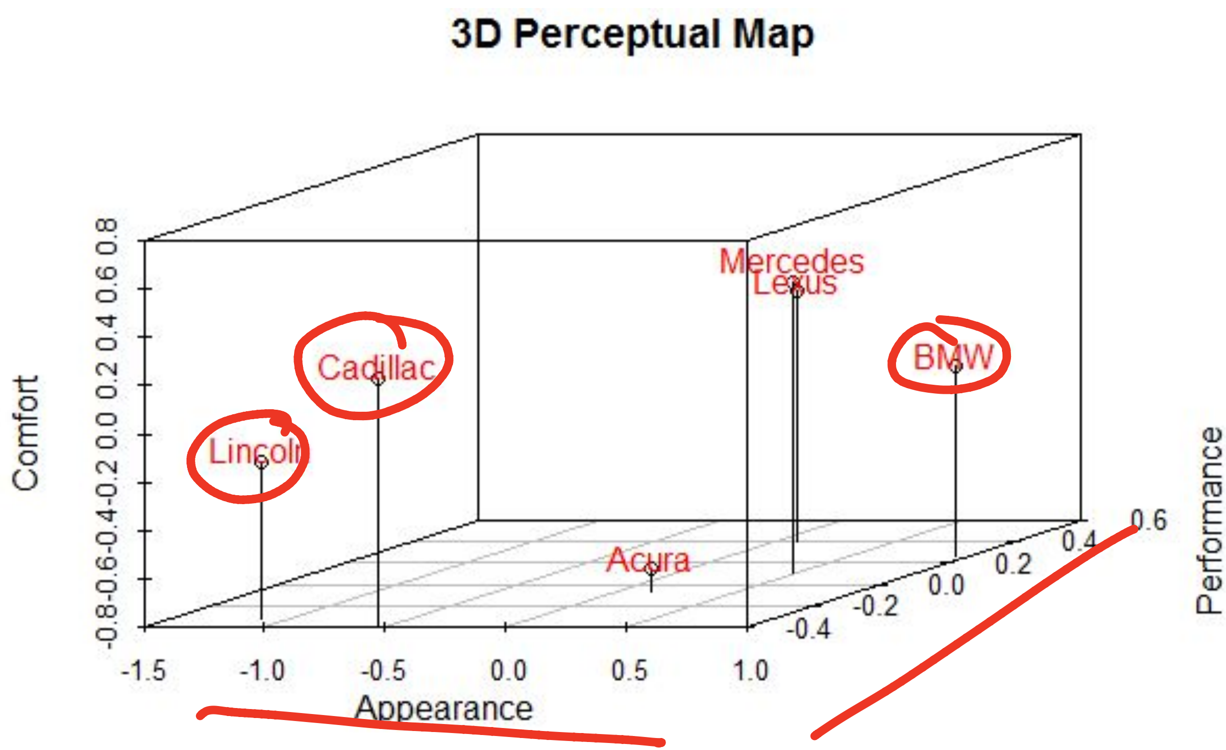
Perceptual map in two dimensions



Perceptual map in 3D

```
# 3D Scatterplot
library(scatterplot3d)
s3d <- scatterplot3d(Appearance, Performance, Comfort,
                     scale.y = 1, type='h', asp = 1,
                     main="3D Perceptual Map")
text(s3d$xyz.convert(Appearance, Performance, Comfort + c(rep(0.05,5),0.1)),
     labels=(brand.mean[,1]),
     col = 'red')
```

Perceptual map in 3D



Average Factor Scores by Brand and Education

```r
brand_by_edu=aggregate(summarized_data[,c(3:5)], by=list(Brand, Education),
                       FUN=mean, na.rm=TRUE)
colnames(brand_by_edu) = c("Brand", "Edu", "Appearance", "Performance", "Comfort")
brand_by_edu # Print the average factor scores by brand and education
```

Brand <fctr>	Edu <fctr>	Appearance <dbl>	Performance <dbl>	Comfort <dbl>
Acura	MBA	0.03433972	0.13653402	[illegible]
BMW	MBA	0.91392623	0.37221942	-0.15632088
Cadillac	MBA	-0.83545768	-0.50688479	0.19605853
Lexus	MBA	0.04216595	0.50547105	0.05818948
Lincoln	MBA	-1.23476083	-0.77846877	[illegible]
Mercedes	MBA	0.22722894	0.05009002	0.35085023
Acura	Undergrad	0.19666890	-0.15732498	-0.70903563
BMW	Undergrad	0.95635870	0.22149948	[illegible]
Cadillac	Undergrad	-0.46204896	-0.32136945	0.23018062
Lexus	Undergrad	0.11880797	0.36341949	[illegible]

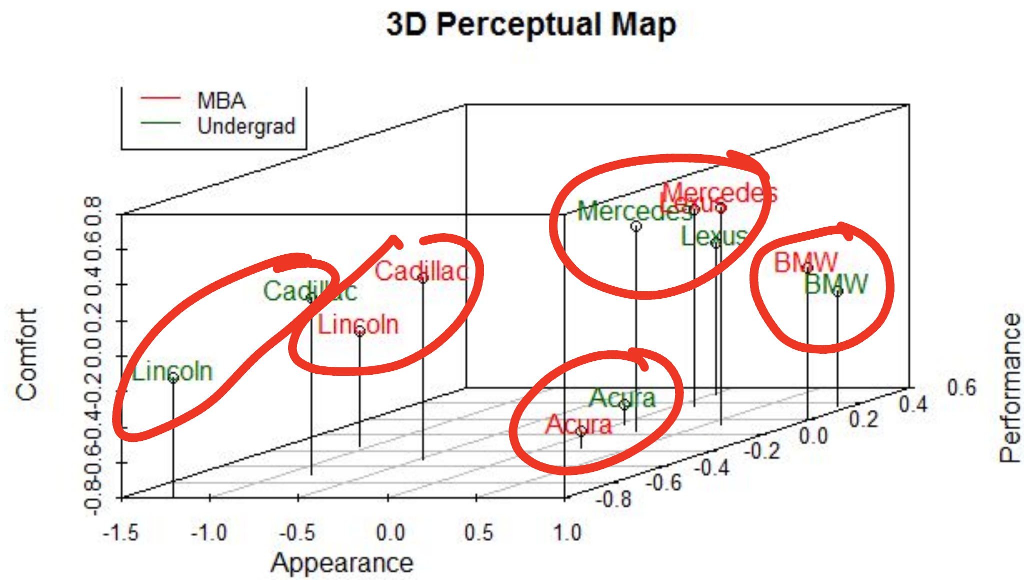
1-10 of 12 rows   Previous 1 2 Next

Perceptual map in 3D by education level

```r
# 3D Scatterplot
library(scatterplot3d)
s3d1 <- scatterplot3d(brand_by_edu[,3], brand_by_edu[,4],
        brand_by_edu[,5], xlab = "Appearance",
        ylab = "Performance", zlab = "Comfort",
        scale.y = 1, type='h', asp = 1,
        main="3D Perceptual Map")
tmp <- brand_by_edu[which(brand_by_edu$Edu == 'MBA'),]
text(s3d1$xyz.convert(tmp$Appearance, tmp$Performance,
       tmp$Comfort + c(rep(0.05,5),0.1)),
       labels=(tmp$Brands), col = 'darkgreen')
```

```r
tmp <- brand_by_edu[which(brand_by_edu$Edu=='Undergrad'),]
text(s3d1$xyz.convert(tmp$Appearance, tmp$Performance,
       tmp$Comfort + c(rep(0.05,5),0.1)),
       labels=(tmp$Brands), col = 'red')
legend(-3, 8,
       legend=c("MBA", "Undergrad"),
       col=c("red", "darkgreen"), lty=1, cex=0.8)
```

Perceptual map in 3D by education level



Drivers of brand preference

```
# Append preference data
red_data = cbind(cardata[,c(2,3,10)], fit$scores)
colnames(red_data) = c("Brand", "Edu", "Preference", "Appearance", "Performance", "Comfort")
head(red_data)
```

	Brand	Edu	Preference	Appearance	Performance	Comfort
	<fctr>	<fctr>	<int>	<dbl>	<dbl>	<dbl>
1	MBA	Acura	7	-0.4633312	0.60738739	-1.3943949
2	MBA	BMW	10	1.1217549	-0.03578809	-0.4699704
3	MBA	Cadillac	5	-0.8329368	-1.35020394	1.2683913
4	MBA	Lexus	8	-1.1427352	0.98872908	0.3067639
5	MBA	Lincoln	4	-1.3701566	-1.39009793	1.4202708
6	MBA	Mercedes	9	0.3216445	0.30087651	0.3175364

6 rows

Drivers of brand preference

```
# Multiple Linear Regression
regfit <- lm(Preference ~ Appearance + Performance + Comfort, data=red_data)
summary(regfit) # show results
```

```
Call:
lm(formula = Preference ~ Appearance + Performance + Comfort,
    data = red_data)

Residuals:
   Min     1Q Median     3Q    Max
-6.616 -0.982 -0.381  0.344 44.218

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.5181     0.1752  42.918  < 2e-16 ***
Appearance    1.5609     0.1753   8.903  < 2e-16 ***
Performance   1.1064     0.1753   6.310 5.75e-10 ***
Comfort       0.5364     0.1753   3.059  0.00233 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.116 on 548 degrees of freedom
Multiple R-squared:  0.1899,    Adjusted R-squared:  0.1854
F-statistic: 42.81 on 3 and 548 DF,  p-value: < 2.2e-16
```

Preference= 7.51+ 1.56Appearance +1.10Performance +0.53Comfort

Summary: Factor Analysis is useful for

- Data reduction (reducing the number of variables)
- Substantive interpretation (allows for understanding the underlying structure in the data)
- Can be used for subsequent analysis (dealing with multicolineraity in regression, perceptual maps)

Interpreting a factor analysis output

- Decide on the number of factors to be retained
  - ✓ Use eigenvalue criterion and own judgment
  - ✓ Evaluate the percent of variance explained
- Use the rotated factor loadings to interpret the factor structure
  - ✓ Loadings should be greater than 0.5
- Save factor scores for subsequent analyses