PrelimAnalysis

Jaffa Romain

2022-04-16

# Principal Component Analysis

* to reduce noise: removes redundant/useless variables
* minimizes collinearity - reduces number of independent variables that are highly correlated to each other
* Varimax rotation
* eigenvalues greater than 1

library(psych)

##   
## Attaching package: 'psych'

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

library(factoextra)

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

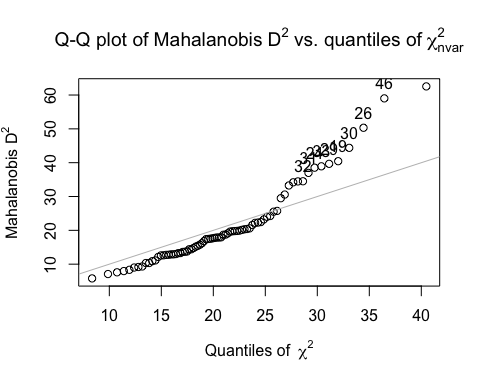
# remove categorical and unimportant variables  
PCAdata <- LIWCdata %>% dplyr::select(-c(Filename,Segment, WC))  
PCAdata2<- PCAdata[,apply(PCAdata, 2, function(x) all(x > 0))]  
   
PCA\_scaled <- PCAdata2 %>% mutate\_all(~scale(.) %>% as.vector)  
  
   
# STANDARDIZE DATA  
# to transform all variables to the same scale  
# Average scores below the mean of 0 indicated that participants used fewer words and phrases from the word categories compared with the mean  
# Factor scores that did not change significantly between the two writing assignments can be interpreted as representing whatever attitudes, experiences, or beliefs typical of participants that were unchanged by the educational intervention.   
  
  
###### Preliminary Checks ######  
# Bartlett's test for correlation matrix  
cortest.mat(PCA\_scaled)

## Warning in cortest.mat(PCA\_scaled): R1 matrix was not square, correlations found

## Bartlett's test of is R = I

## Tests of correlation matrices   
## Call:cortest.mat(R1 = PCA\_scaled)  
## Chi Square value 1736.27 with df = 210 with probability < 4.3e-238

####### squared multiple correlations (smc) #######  
smc1= smc(PCA\_scaled) # all > 0.3 - no factors need to be removed  
  
#### outliers #####  
outlier(PCA\_scaled, plot=T, bad=10, na.rm=T)



## 1 2 3 4 5 6 7 8   
## 25.508326 16.823914 17.875347 21.574751 12.585772 22.443265 13.245755 12.138694   
## 9 10 11 12 13 14 15 16   
## 20.396979 17.411363 8.332443 9.163619 14.775462 10.336261 13.674714 16.192006   
## 17 18 19 20 21 22 23 24   
## 7.082198 10.345332 40.441984 30.579311 29.499297 24.243175 22.296740 38.522555   
## 25 26 27 28 29 30 31 32   
## 62.550286 50.323763 17.938576 17.765723 39.659657 44.385465 36.979869 34.509268   
## 33 34 35 36 37 38 39 40   
## 38.928195 11.109585 25.742473 9.319790 20.312807 17.447577 12.897598 12.906388   
## 41 42 43 44 45 46 47 48   
## 12.619889 19.824915 9.004874 33.264790 15.165080 59.011208 19.099490 12.780833   
## 49 50 51 52 53 54 55 56   
## 19.628541 7.958966 12.989206 10.889560 19.924017 13.283874 17.526427 13.747196   
## 57 58 59 60 61 62 63 64   
## 7.593395 19.778113 13.611661 12.742253 15.422150 20.588089 5.792444 23.992732   
## 65 66 67 68 69 70 71 72   
## 34.200422 17.636569 18.692006 20.141612 18.736394 23.231861 19.719716 14.410164   
## 73 74 75 76 77   
## 17.884047 15.759646 34.462320 22.209093 14.410164

# Determining number of factors to extract  
library(nFactors)

## Loading required package: lattice

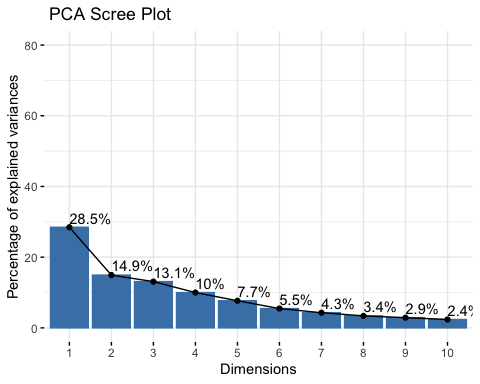
##   
## Attaching package: 'nFactors'

## The following object is masked from 'package:lattice':  
##   
## parallel

library(factoextra)  
# common rule - atleast 70% of total variation explained with eigenvalues of at least 1  
pca=prcomp(PCA\_scaled)  
# rotation - implement a prior shuffling of varimax rotation - better captures explained variance  
summary(pca)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 2.4448 1.7706 1.6578 1.44797 1.26963 1.07152 0.94838  
## Proportion of Variance 0.2846 0.1493 0.1309 0.09984 0.07676 0.05467 0.04283  
## Cumulative Proportion 0.2846 0.4339 0.5648 0.66461 0.74137 0.79605 0.83888  
## PC8 PC9 PC10 PC11 PC12 PC13 PC14  
## Standard deviation 0.84349 0.77876 0.70558 0.64777 0.59434 0.51329 0.43873  
## Proportion of Variance 0.03388 0.02888 0.02371 0.01998 0.01682 0.01255 0.00917  
## Cumulative Proportion 0.87276 0.90164 0.92534 0.94532 0.96214 0.97469 0.98386  
## PC15 PC16 PC17 PC18 PC19 PC20 PC21  
## Standard deviation 0.33247 0.29392 0.25700 0.18802 0.16206 0.09048 0.07895  
## Proportion of Variance 0.00526 0.00411 0.00315 0.00168 0.00125 0.00039 0.00030  
## Cumulative Proportion 0.98912 0.99323 0.99638 0.99806 0.99931 0.99970 1.00000

fviz\_screeplot(pca, type="lines", addlabels=TRUE, ylim = c(0, 80), main=" PCA Scree Plot"  
 )



get\_eig(pca)

## eigenvalue variance.percent cumulative.variance.percent  
## Dim.1 5.976940300 28.46162048 28.46162  
## Dim.2 3.134872011 14.92796196 43.38958  
## Dim.3 2.748431753 13.08777025 56.47735  
## Dim.4 2.096603835 9.98382779 66.46118  
## Dim.5 1.611971614 7.67605531 74.13724  
## Dim.6 1.148158237 5.46742018 79.60466  
## Dim.7 0.899428982 4.28299515 83.88765  
## Dim.8 0.711483316 3.38801579 87.27567  
## Dim.9 0.606461054 2.88790978 90.16358  
## Dim.10 0.497842188 2.37067709 92.53425  
## Dim.11 0.419606399 1.99812571 94.53238  
## Dim.12 0.353241858 1.68210409 96.21448  
## Dim.13 0.263471075 1.25462417 97.46911  
## Dim.14 0.192480354 0.91657311 98.38568  
## Dim.15 0.110536625 0.52636488 98.91205  
## Dim.16 0.086386332 0.41136349 99.32341  
## Dim.17 0.066048176 0.31451513 99.63792  
## Dim.18 0.035352046 0.16834308 99.80627  
## Dim.19 0.026264066 0.12506698 99.93133  
## Dim.20 0.008186601 0.03898381 99.97032  
## Dim.21 0.006233177 0.02968179 100.00000

# 5 is optimal  
  
library(ggfortify)  
  
# loadings  
  
  
res.pc <-principal(PCA\_scaled, nfactors=5, rotate="varimax")  
# cronbach alpha   
  
  
  
loadings <- res.pc$loadings  
  
loadings

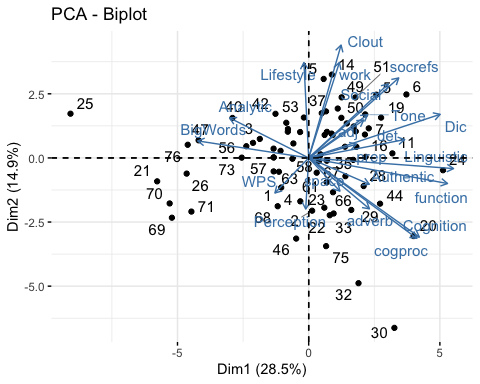
##   
## Loadings:  
## RC1 RC3 RC2 RC4 RC5   
## Analytic -0.751 0.135 -0.353 -0.162  
## Clout -0.262 0.129 0.902 0.105   
## Authentic 0.487 0.144   
## Tone 0.480 0.217 -0.227  
## WPS 0.190 -0.537   
## BigWords -0.452 -0.664   
## Dic 0.377 0.669 0.345 0.333   
## Linguistic 0.624 0.692 0.237   
## function 0.554 0.762 0.177  
## det 0.132 0.840 0.102 -0.124 -0.172  
## prep -0.238 0.720 -0.269 0.319  
## adverb 0.486 0.117 -0.114 0.363  
## adj 0.268 0.105 0.715 0.156  
## Cognition 0.927 0.165   
## cogproc 0.928 0.138   
## Social 0.216 0.100 0.930   
## socrefs 0.256 0.132 0.913 0.196   
## Lifestyle -0.241 0.903   
## work 0.104 0.147 0.850 -0.250  
## Perception 0.154 -0.346 0.118 -0.106 0.831  
## space 0.153 0.925  
##   
## RC1 RC3 RC2 RC4 RC5  
## SS loadings 4.250 3.959 2.798 2.532 2.030  
## Proportion Var 0.202 0.189 0.133 0.121 0.097  
## Cumulative Var 0.202 0.391 0.524 0.645 0.741

library(FactoMineR)  
res2 <- PCA(PCA\_scaled, graph = FALSE, scale.unit = T)  
res2$eig

## eigenvalue percentage of variance cumulative percentage of variance  
## comp 1 5.976940300 28.46162048 28.46162  
## comp 2 3.134872011 14.92796196 43.38958  
## comp 3 2.748431753 13.08777025 56.47735  
## comp 4 2.096603835 9.98382779 66.46118  
## comp 5 1.611971614 7.67605531 74.13724  
## comp 6 1.148158237 5.46742018 79.60466  
## comp 7 0.899428982 4.28299515 83.88765  
## comp 8 0.711483316 3.38801579 87.27567  
## comp 9 0.606461054 2.88790978 90.16358  
## comp 10 0.497842188 2.37067709 92.53425  
## comp 11 0.419606399 1.99812571 94.53238  
## comp 12 0.353241858 1.68210409 96.21448  
## comp 13 0.263471075 1.25462417 97.46911  
## comp 14 0.192480354 0.91657311 98.38568  
## comp 15 0.110536625 0.52636488 98.91205  
## comp 16 0.086386332 0.41136349 99.32341  
## comp 17 0.066048176 0.31451513 99.63792  
## comp 18 0.035352046 0.16834308 99.80627  
## comp 19 0.026264066 0.12506698 99.93133  
## comp 20 0.008186601 0.03898381 99.97032  
## comp 21 0.006233177 0.02968179 100.00000

loadings <-res2$var$cor # table with factor loadings  
  
# how much of each of the 5 dimensions are explained by the extracted factors.  
loadings <- as\_tibble(res2$var$cor) %>% # We need to capture the loadings as a data frame into a new object. Use as\_tibble(), otherwise we cannot access the different factors  
 mutate(variable = rownames(res2$var$cor), # keep track of the row names (these are removed when converting to tibble)  
 communality = Dim.1^2 + Dim.2^2,   
 uniqueness = 1 - communality) # The ^ operator elevates a value to a certain power. To calculate the communality, we need to sum the squares of the loadings on each factor.  
  
# communality is percentage of that variable's that is explained by the factors  
# uniqueness - measurement error - uniqueness > 0.6 is considered high - meaningt hat the variable is not well explained by the factors  
  
  
fviz\_pca\_biplot(res2, repel = TRUE) # plot the loadings and the brands together on one plot

## Warning: ggrepel: 28 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps



library(kableExtra)

##   
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':  
##   
## group\_rows

vars = as.data.frame(loadings$variable)  
vars$Variable = vars$`loadings$variable`   
vars <- vars %>% select(-`loadings$variable`)  
vars %>% cbind(loadings) %>% select(-c(communality, uniqueness, variable)) %>% kable(caption = "Correlations between Principal Components and the original scores.", digits = 2) %>% save\_kable("tbl1.jpg")

## save\_kable will have the best result with magick installed.