m21_pca

Joanna Morris

2024-10-28

This script computes the PCA for Morph21.

1. First we load the libraries we need

```
library(readr)
library(psych)
library(dplyr)
```

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

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

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

```
library(tidyr)
```

Compute PCA

Following Andrews and Lo (2013) this script computes a PCA for our spelling and vocabulary measures. Because the standardised spelling and vocabulary scores were correlated, to facilitate interpretation, two orthogonal measures of individual differences were derived from a principal components analysis. Analysis based on this tutorial

First we import the data, remove missing values adn standardize the scores.

```
library(readr)
library(dplyr)
library(datawizard)
```

```
##
## Attaching package: 'datawizard'

## The following object is masked from 'package:psych':
##
##     rescale
```

```
langdat_1_202410 <- read_csv("m21_langdat_1.csv")
```

```
## Rows: 70 Columns: 5

## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr (1): Sex
## dbl (4): SubjID, ART, Spelling, Vocabulary
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
langdat_2_202410 <- read_csv("m21_langdat_2.csv")
```

```
## Rows: 45 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Sex
## dbl (4): SubjID, ART, Spelling, Vocabulary
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
langdat_1.na <- na.omit(langdat_1_202410)
langdat_2.na <- na.omit(langdat_2_202410)

describe(langdat_1.na)
```

```
##            vars  n   mean    sd median trimmed   mad min max range  skew
## SubjID        1 61 137.49 23.13    137  137.02 29.65 101 177    76  0.16
## ART           2 61  10.20  4.80     10   10.04  4.45  -3  23    26  0.20
## Spelling      3 61  63.89  6.17     63   63.84  7.41  51  77    26  0.04
## Vocabulary    4 61  40.89  5.60     42   41.20  5.93  29  49    20 -0.46
## Sex*          5 61   1.23  0.42      1    1.16  0.00   1   2     1  1.25
##            kurtosis   se
## SubjID        -1.28 2.96
## ART            0.13 0.61
## Spelling      -0.66 0.79
## Vocabulary    -0.96 0.72
## Sex*          -0.43 0.05
```

```r
describe(langdat_2.na)
```

```
##            vars  n   mean    sd median trimmed   mad min max range  skew
## SubjID        1 45 223.00 13.13    223  223.00 16.31 201 245    44  0.00
## ART           2 45   5.40  4.10      5    5.51  4.45  -7  15    22 -0.33
## Spelling      3 45  61.00  5.90     61   61.03  5.93  47  73    26 -0.09
## Vocabulary    4 45  31.64  6.49     32   31.70  7.41  17  44    27 -0.04
## Sex*          5 45   1.40  0.50      1    1.38  0.00   1   2     1  0.39
##            kurtosis   se
## SubjID        -1.28 1.96
## ART            0.56 0.61
## Spelling      -0.46 0.88
## Vocabulary    -0.65 0.97
## Sex*          -1.88 0.07
```

```r
langdat_1.na <- mutate(langdat_1.na,
                       z_ART = standardise(ART),
                       z_Vocabulary = standardise(Vocabulary),
                       z_Spelling = standardise(Spelling))

langdat_2.na <- mutate(langdat_2.na,
```

```
                        z_ART = standardise(ART),
                        z_Vocabulary = standardise(Vocabulary),
                        z_Spelling = standardise(Spelling))
```

Now we can put the three standardized measures into a separate data frame and compute the correlations, using the `cor()` function. NB. A correlation coefficient is a standardized covariance statistic. We can run the `cov()` function on the standardized values or the `cor()` function on the unstandardized ones. Both methods will give the same results.

```
art_vcb_spl_raw_1 <- langdat_1.na |> select(Vocabulary, Spelling, ART)
art_vcb_spl_z_1 <- langdat_1.na |> select( z_Vocabulary, z_Spelling, z_ART)

cor(art_vcb_spl_raw_1, use = "everything", method = "pearson")
```

```
##            Vocabulary  Spelling       ART
## Vocabulary  1.0000000 0.2171216 0.5136866
## Spelling    0.2171216 1.0000000 0.2349441
## ART         0.5136866 0.2349441 1.0000000
```

```
cov(art_vcb_spl_z_1, use = "everything", method = "pearson")
```

```
##              z_Vocabulary z_Spelling     z_ART
## z_Vocabulary    1.0000000  0.2171216 0.5136866
## z_Spelling      0.2171216  1.0000000 0.2349441
## z_ART           0.5136866  0.2349441 1.0000000
```

```
art_vcb_spl_raw_2 <- langdat_2.na |> select(Vocabulary, Spelling, ART)
art_vcb_spl_z_2 <- langdat_2.na |> select( z_Vocabulary, z_Spelling, z_ART)

cor(art_vcb_spl_raw_2, use = "everything", method = "pearson")
```

```
##            Vocabulary  Spelling       ART
## Vocabulary  1.0000000 0.4933601 0.6163415
## Spelling    0.4933601 1.0000000 0.5351840
## ART         0.6163415 0.5351840 1.0000000
```

```
cov(art_vcb_spl_z_2, use = "everything", method = "pearson")
```

```
##              z_Vocabulary z_Spelling     z_ART
## z_Vocabulary    1.0000000  0.4933601 0.6163415
## z_Spelling      0.4933601  1.0000000 0.5351840
## z_ART           0.6163415  0.5351840 1.0000000
```

Once we have generated the correlation coefficients we can test them for statistical significance. You can only test one correlation at a time using the `cor.test()` function, but the `corr.test()` function in the `psych` package will test a matrix of correlation coefficients.

```
library(psych)
corr.test(art_vcb_spl_z_1)
```

```
## Call:corr.test(x = art_vcb_spl_z_1)
## Correlation matrix
##              z_Vocabulary z_Spelling z_ART
## z_Vocabulary         1.00       0.22  0.51
## z_Spelling           0.22       1.00  0.23
## z_ART                0.51       0.23  1.00
## Sample Size
## [1] 61
```

```
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##              z_Vocabulary z_Spelling z_ART
## z_Vocabulary         0.00       0.14  0.00
## z_Spelling           0.09       0.00  0.14
## z_ART                0.00       0.07  0.00
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```r
corr.test(art_vcb_spl_z_2)
```

```
## Call:corr.test(x = art_vcb_spl_z_2)
## Correlation matrix
##              z_Vocabulary z_Spelling z_ART
## z_Vocabulary         1.00       0.49  0.62
## z_Spelling           0.49       1.00  0.54
## z_ART                0.62       0.54  1.00
## Sample Size
## [1] 45
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##              z_Vocabulary z_Spelling z_ART
## z_Vocabulary            0          0     0
## z_Spelling              0          0     0
## z_ART                   0          0     0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

Now we can do the PCA. It turns out that by default, the function `PCA()` in `FactoMineR`, standardizes the data automatically, so we didn't actually need do the standardization. Oh well. ¯_()_/¯

Here are the arguments to the `PCA()` function:

- `X`: a data frame. Rows are individuals and columns are numeric variables
- `scale.unit`: a logical value. If TRUE, the data are scaled to unit variance before the analysis. This standardization to the same scale avoids some variables to become dominant just because of their large measurement units. It makes variables comparable.
- `ncp`: number of dimensions kept in the final results.
- `graph`: a logical value. If TRUE a graph is displayed.

The plot shows the relationships between all variables. It can be interpreted as follow:

- Positively correlated variables are grouped together.
- Negatively correlated variables are positioned on opposite sides of the plot origin (opposed quadrants).
- The distance between variables and the origin measures the quality of the variables on the factor map. Variables that are away from the origin are well represented on the factor map.

```r
library(FactoMineR)
library(factoextra)
```

```
## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

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

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

```
res.pca_1 <- PCA(langdat_1.na[,2:4], scale.unit = TRUE, ncp = 2, graph = FALSE)
plot(res.pca_1, choix = "varcor", graph.type = c("ggplot"))
```

m21_pca_files/figure-latex/unnamed-chunk-5-1.pdf

```
res.pca_2 <- PCA(langdat_2.na[,2:4], scale.unit = TRUE, ncp = 2, graph = FALSE)
plot(res.pca_2, choix = "varcor", graph.type = c("ggplot"))
```

m21_pca_files/figure-latex/unnamed-chunk-5-2.pdf

The eigenvalues measure the amount of variation retained by each principal component. Eigenvalues are large for the first PCs and small for the subsequent PCs. That is, the first PCs corresponds to the directions with the maximum amount of variation in the data set.

We examine the eigenvalues to determine the number of principal components to be considered. The sum of all the eigenvalues give a total variance of 3, the number of variables. An eigenvalue > 1 indicates that PCs account for more variance than accounted by one of the original variables in standardized data. This is commonly used as a cutoff point for which PCs are retained. This holds true only when the data are standardized.

```
(eig.val_1 <- get_eigenvalue(res.pca_1))
```

```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1  1.6669296         55.56432                    55.56432
## Dim.2  0.8471400         28.23800                    83.80232
## Dim.3  0.4859304         16.19768                   100.00000
```

```
(eig.val_2 <- get_eigenvalue(res.pca_2))
```

```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1  2.0982128         69.94043                    69.94043
## Dim.2  0.5226527         17.42176                    87.36218
## Dim.3  0.3791345         12.63782                   100.00000
```

The quality of representation of the variables on factor map is called cos2 (square cosine, squared coordinates). *A high cos2 indicates a good representation of the variable on the principal component.* In this case the variable is positioned close to the circumference of the correlation circle. *A low cos2 indicates that the variable is not perfectly represented by the PCs.* In this case the variable is close to the center of the circle. If a variable is perfectly represented by only two principal components (Dim.1 & Dim.2), the sum of the cos2 on these two PCs is equal to one. In this case the variables will be positioned on the circle of correlations.

```
res.pca_1$var$cos2
```

```
##                Dim.1      Dim.2
## ART        0.6846519 0.06801824
## Spelling   0.3114976 0.68805400
## Vocabulary 0.6707801 0.09106777
```

```r
res.pca_2$var$cos2
```

```
##                Dim.1      Dim.2
## ART        0.7457531 0.03592741
## Spelling   0.6400313 0.35136755
## Vocabulary 0.7124284 0.13535772
```

The contributions of variables in accounting for the variability in a given principal component are expressed in percentages. Variables that are correlated with PC1 (i.e., Dim.1) and PC2 (i.e., Dim.2) are the most important in explaining the variability in the data set. The larger the value of the contribution, the more the variable contributes to the component. It's possible to use the function corrplot() [corrplot package] to highlight the most contributing variables for each dimension.

```r
library('corrplot')
```

```
## corrplot 0.95 loaded
```

```r
res.pca_1$var$contrib
```

```
##               Dim.1     Dim.2
## ART        41.07264  8.029161
## Spelling   18.68691 81.220813
## Vocabulary 40.24046 10.750026
```

```r
res.pca_2$var$contrib
```

```
##               Dim.1    Dim.2
## ART        35.54230  6.87405
## Spelling   30.50364 67.22773
## Vocabulary 33.95406 25.89822
```

```r
corrplot(res.pca_1$var$contrib, is.corr=FALSE)
```

m21_pca_files/figure-latex/unnamed-chunk-7-1.pdf

```r
corrplot(res.pca_2$var$contrib, is.corr=FALSE)
```

m21_pca_files/figure-latex/unnamed-chunk-7-2.pdf

The correlation between a variable and a principal component (PC) is used as the coordinates of the variable on the PC.

```r
(res.pca_1$var$coord)
```

```
##                Dim.1      Dim.2
## ART        0.8274370 -0.2608031
## Spelling   0.5581197  0.8294902
## Vocabulary 0.8190116 -0.3017744
```

```r
(res.desc <- dimdesc(res.pca_1, axes = c(1,2), proba = 0.05))
```

```
## $Dim.1
##
## Link between the variable and the continuous variables (R-square)
## =================================================================================
##             correlation      p.value
## ART           0.8274370 2.032489e-16
## Vocabulary    0.8190116 7.307938e-16
## Spelling      0.5581197 2.962142e-06
##
## $Dim.2
##
## Link between the variable and the continuous variables (R-square)
## =================================================================================
##             correlation      p.value
## Spelling      0.8294902 1.472427e-16
## ART          -0.2608031 4.234863e-02
## Vocabulary   -0.3017744 1.810003e-02
```

```r
(res.pca_2$var$coord)
```

```
##                Dim.1      Dim.2
## ART        0.8635700 -0.1895453
## Spelling   0.8000196  0.5927626
## Vocabulary 0.8440547 -0.3679099
```

```r
(res.desc <- dimdesc(res.pca_2, axes = c(1,2), proba = 0.05))
```

```
## $Dim.1
##
## Link between the variable and the continuous variables (R-square)
## =================================================================================
##             correlation      p.value
## ART           0.8635700 2.270635e-14
## Vocabulary    0.8440547 3.277432e-13
## Spelling      0.8000196 4.305234e-11
##
## $Dim.2
##
## Link between the variable and the continuous variables (R-square)
## =================================================================================
##             correlation      p.value
## Spelling      0.5927626 1.784157e-05
## Vocabulary   -0.3679099 1.290211e-02
```

The fviz_pca_ind() is used to produce the graph of individuals.

```r
ind.1 <- get_pca_ind(res.pca_1)
fviz_pca_ind(res.pca_1)
```

m21_pca_files/figure-latex/c6-1.pdf

```r
ind.2 <- get_pca_ind(res.pca_2)
fviz_pca_ind(res.pca_2)
```

m21_pca_files/figure-latex/c6-2.pdf

```r
langdat_1.na<-bind_cols(langdat_1.na,res.pca_1$ind$coord)
langdat_2.na<-bind_cols(langdat_2.na,res.pca_2$ind$coord)

#Divide participants based on median split of Dim2.  Higher values on this factor indicate that spellin

langdat_1.na <- langdat_1.na |>
  mutate(lang_type_ortho = case_when(
    Dim.2 <= 0 ~ "Low Orthographic",
    Dim.2 > 0 ~ "High Orthographic"
  ))
langdat_1.na <- langdat_1.na |>
  mutate(lang_type_semantic = case_when(
    Dim.1 <= 0 ~ "Low Semantic",
    Dim.1 > 0 ~ "High Semantic"
  ))

langdat_2.na <- langdat_2.na |>
  mutate(lang_type_ortho = case_when(
    Dim.2 <= 0 ~ "Low Orthographic",
    Dim.2 > 0 ~ "High Orthographic"
  ))
langdat_2.na <- langdat_2.na |>
  mutate(lang_type_semantic = case_when(
    Dim.1 <= 0 ~ "Low Semantic",
    Dim.1 > 0 ~ "High Semantic"
  ))
```

We can then write the indivdiual pca values to a file

```r
write_csv(langdat_1.na, "m21_langdat_1_pca.csv")
write_csv(langdat_2.na, "m21_langdat_2_pca.csv")
```