Dimension Reduction
Week 13

PH 700A, Spring 2025

Rick Calvo

Table of contents

1 Dimensionality Reduction

1.1 Topics

- Packages
- Background
- Methods Overview
- Principal Components Analysis
- Factor Analysis

1.2 Packages

Management and exploration packages: `tidyverse`, `explore`, and `gtsummary`

PCAmixdata: Multivariate Analysis of Mixed Data

`library(PCAmixdata)` - PCA for non-numeric data

FactoMineR: Multivariate Exploratory Data Analysis and Data Mining

`library(FactoMineR)` - Accessory package for Factor Analysis and related methods

factoextra: Extract and Visualize the Results of Multivariate Data Analyses

`library(factoextra)` - Extra functions to visualize results

Analysis of Ecological Data: Exploratory and Euclidean Methods in Environmental Sciences

`library(ade4)` - Accessory package to display multidimensional results

lavaan: Latent Variable Analysis

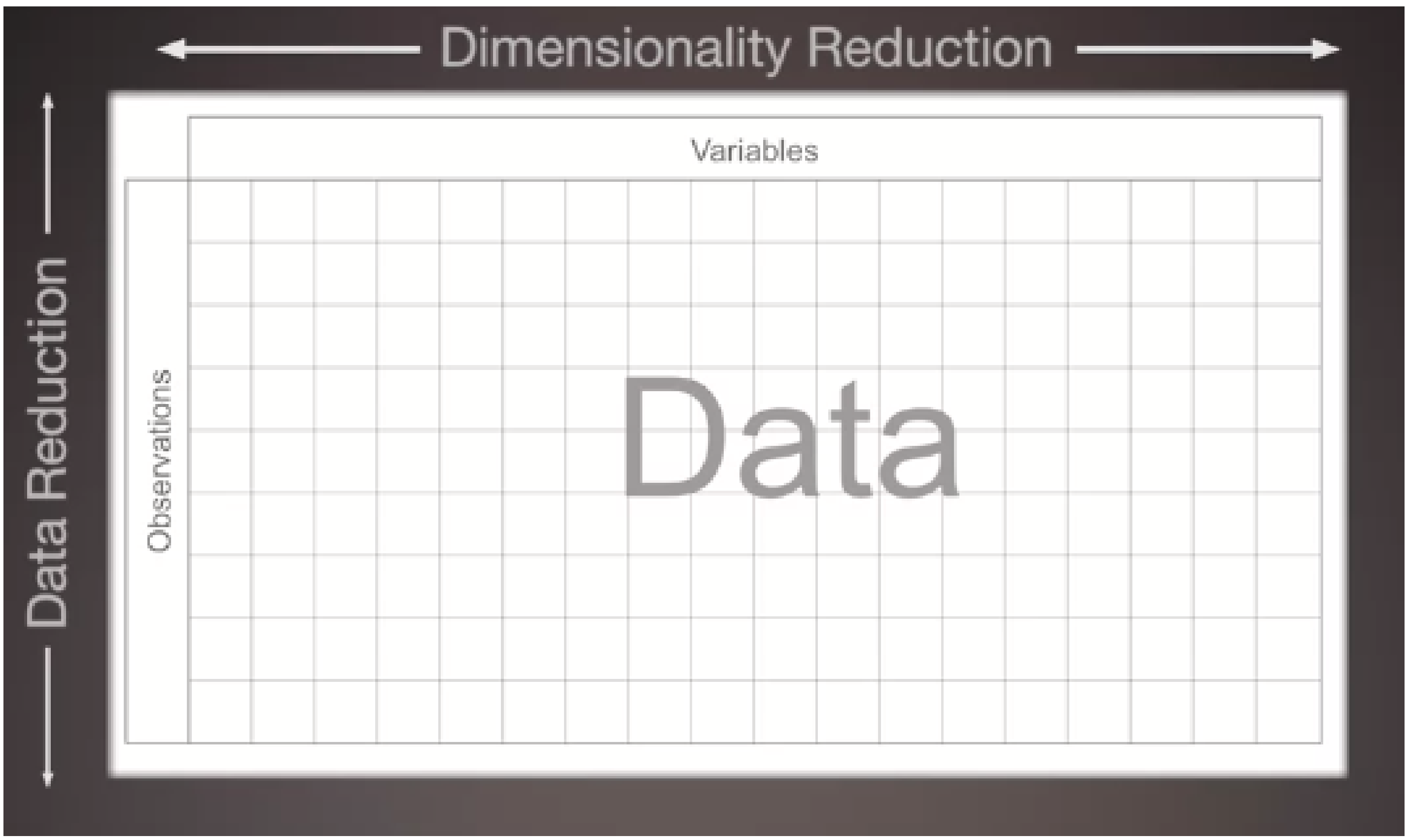
`library(lavaan)` - Structural Equation Modeling

2 Dimensionality Reduction

2.1 Background

Second part of the *two-part series* on *data reduction* and *dimensionality reduction.*

- Last week's statistical foundations on distance and similarity still apply
- Focus is on addressing multitudes of variables



2.2 Reduction Methods

Data (Sample) Reduction	**Dimensionality Reduction**
Cluster Analysis	*Principal Component Analysis*
Neural Network	*Factor Analysis*
Latent Class Analysis	Correspondence Analysis

Data (Sample) Reduction	**Dimensionality Reduction**
Discriminant Analysis	Total Correlation Explanation
Propensity Score Analysis	Structural Equation Modeling

2.3 Dimensionality Reduction

Datasets with *a lot* of variables may have unknown or highly complex intervariable relationships.

- Often occurs with survey/questionnaire data
- Secondary data variable intercorrelation tends to be unknown
- Ideally, we want to group "like" variables together

Addressing the complexity improves your understanding regarding how and which factors are important

- This is *Data Exploration* with data-driven pattern recognition
- Helps determine groups of *consequential* variables

2.4 Methods of Focus

Principal Components Analysis (PCA)

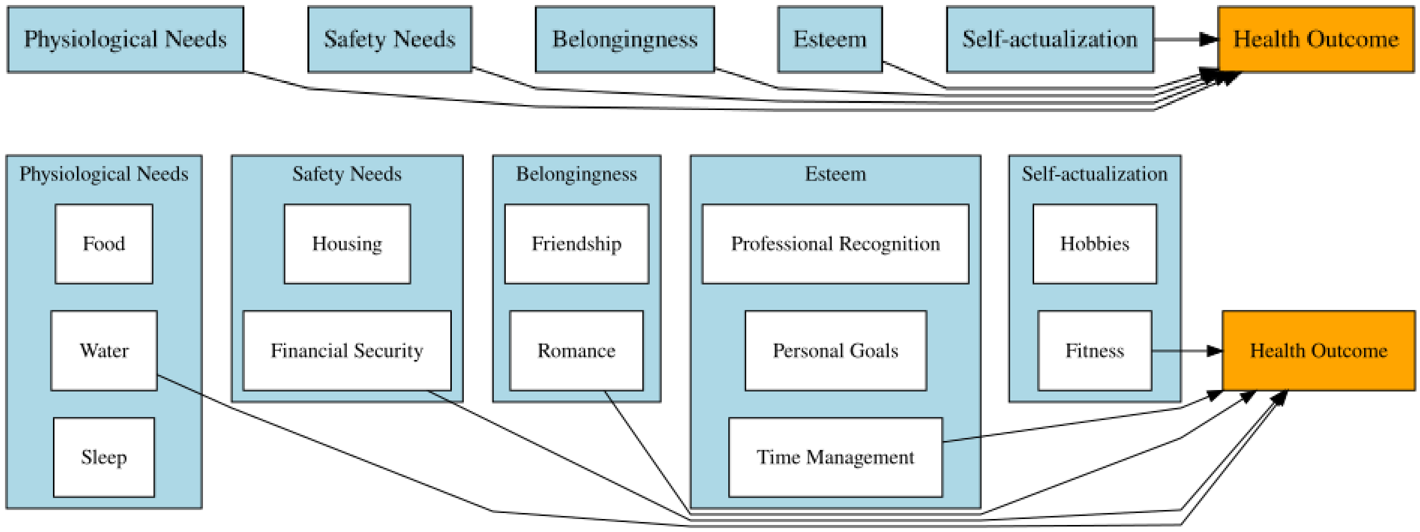
- PCA was made for *numeric* variables, but there are analogous methods that can handle *categorical* or *mixed* data
- Objective is to reduce the number of variables into domain groupings based on *correlation* with like variables

Factor Analysis (FA)

- Used to determine which variable(s) belong in a grouping schema
- Domains could be *known* or *hidden*
- Two forms:
  - Confirmatory Factor Analysis (CFA)
  - Exploratory Factor Analysis (EFA)

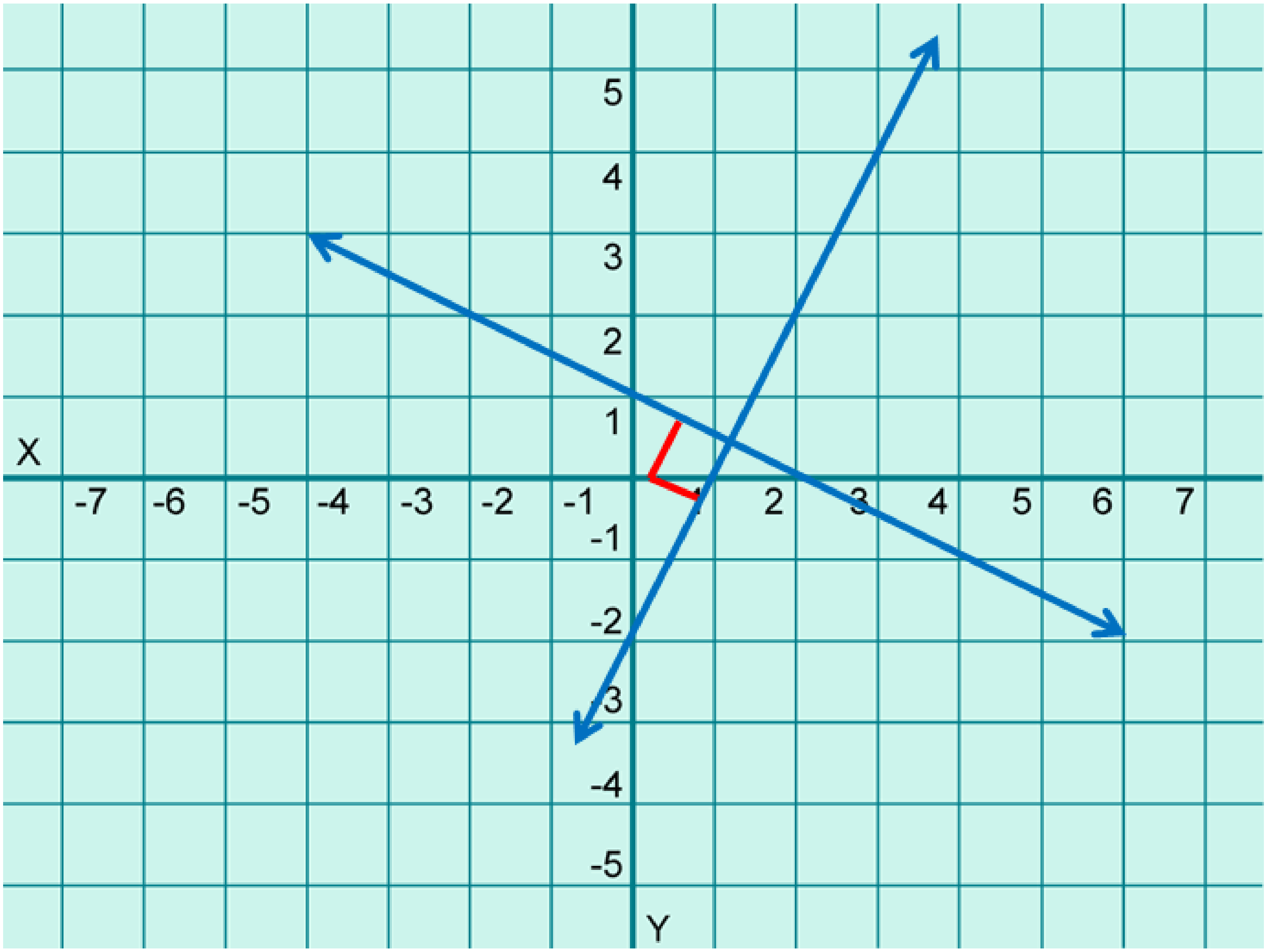
2.5 Domain Representation in Modeling

From *Maslow's Hierarchy of Needs*.



2.6 Statistics Recap

- *Euclidean Space*: a geometrically-valid multidimensional space to map data and evaluate relationships
- *Orthogonality*: Lines that are perpendicular have 0 correlation to each other.
  - For any change in Y, there is 0 change in X and vice versa for two groupings.
- *Transformations* can alter the degree of correlation between factors
  - think: z-scoring and standardization prior to model development



Here, each line represents a *best-fit* relationship (*magnitude* and *direction*) between two variables (X and Y) by a third factor

2.7 Transposition

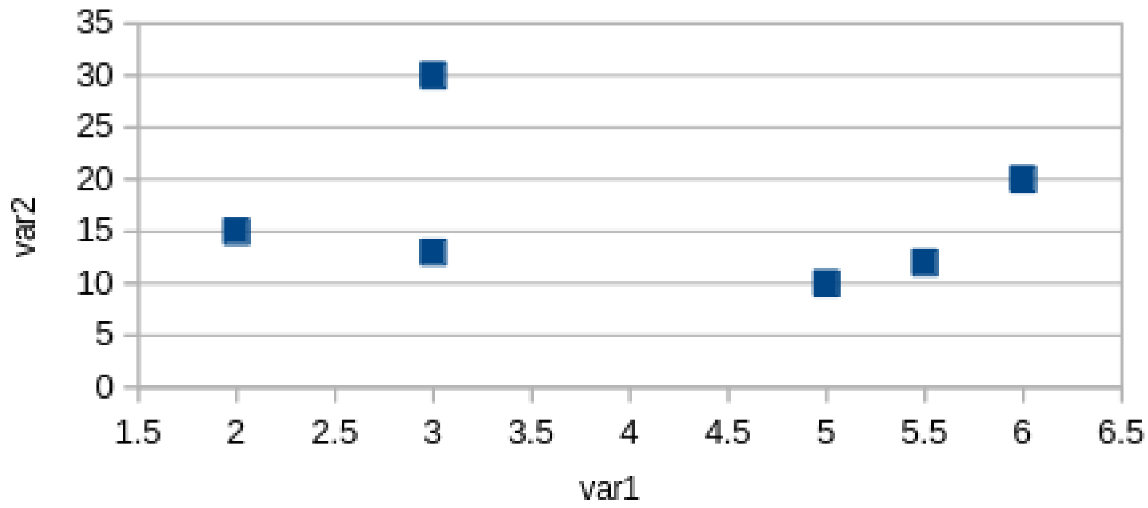
- *Dimensionality Reduction* focuses on variables instead of observations
- Transpose "flips" the data frame so that *variables become the unit of research*

id	var1	var2	var3	…	varN
1	5	10	A	…	1
2	6	20	B	…	0
3	3	30	B	…	0
…	…	…	…	…	…
K	2	15	A	…	0

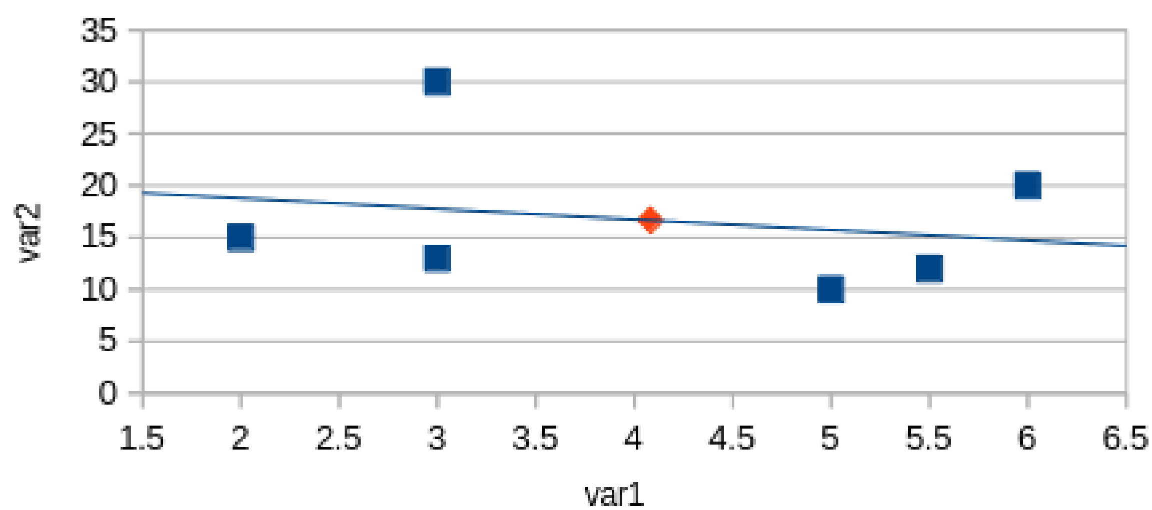
varname	id1	id2	id3	…	idK
var1	5	6	3	…	2
var2	10	20	30	…	15
var3	A	B	B	…	A
…	…	…	…	…	…
varN	1	0	0	…	0

2.8 Two-variable Relationship

varname	id1	id2	id3	id4	id5	id6
var1	5	6	3	5.5	3	2
var2	10	20	30	12	13	15

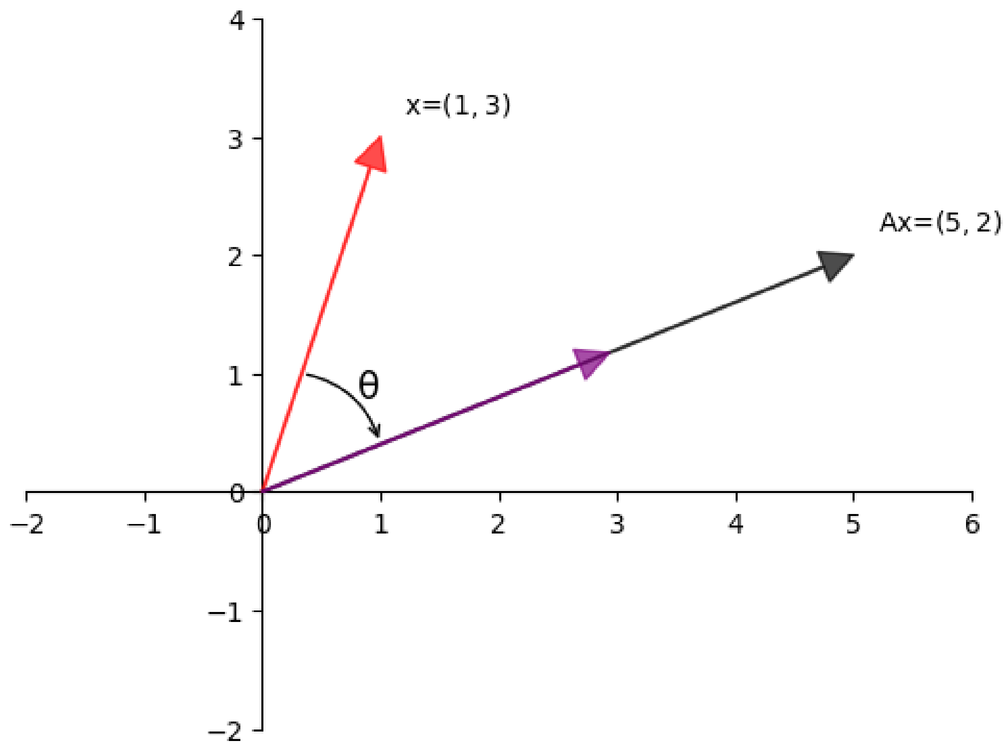


varname	id1	id2	id3	id4	id5	id6
var1	5	6	3	5.5	3	2
var2	10	20	30	12	13	15



2.9 Vectors

- *Vectors* represent relationships between variables in *multidimensional space*
- Each has *magnitude* and *direction*



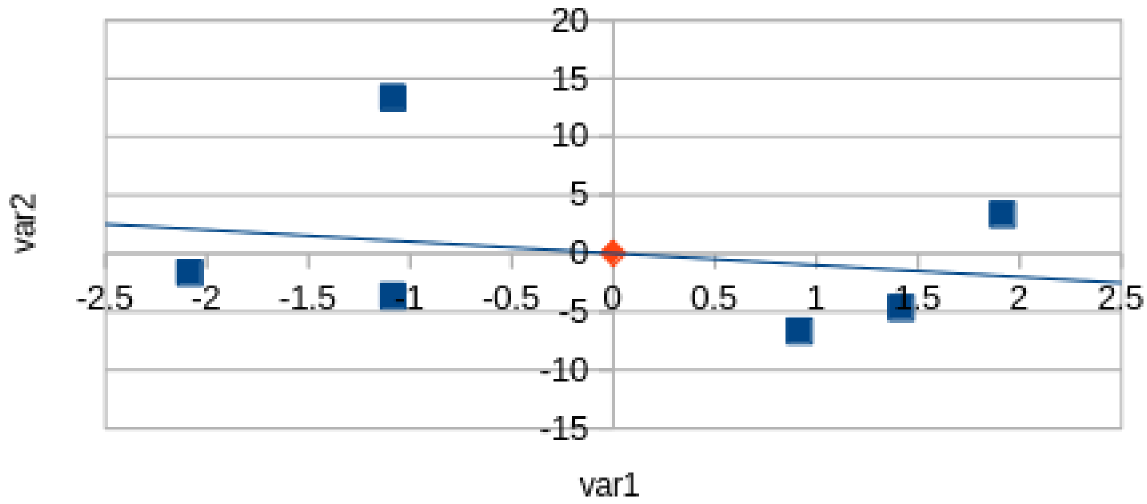
- Mathematical transformation of values typically alters the *magnitude* and *direction* of a vector

2.10 Technical Details

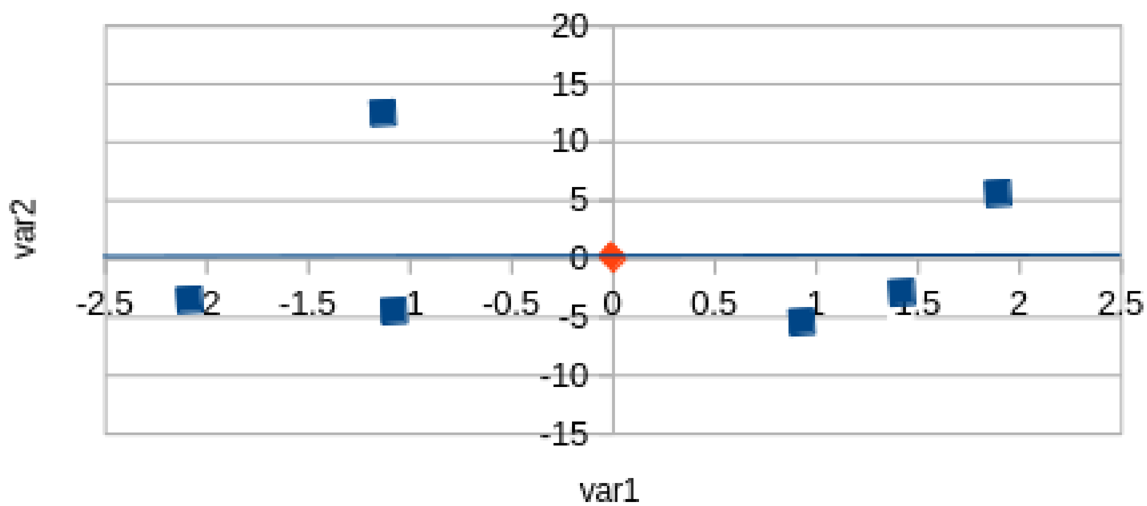
- **Eigenvectors** are transformed vectors with *special properties*:
  - *direction* is unchanged (with reference to its own dimensional axes)
  - *magnitude* can be mathematically transformed
- **Rotation Matrix** is a matrix containing transformations to perform a rotation in euclidean space
- **Eigenvalues** quantifies the amount of transformation on a vector
  - Every eigenvector has an eigenvalue
- **Factor Loading** is the correlation between member variables and their assigned group
  - Indicates the strength and direction of the relationship between each variable and the factor itself

2.11 Centering and Rotation

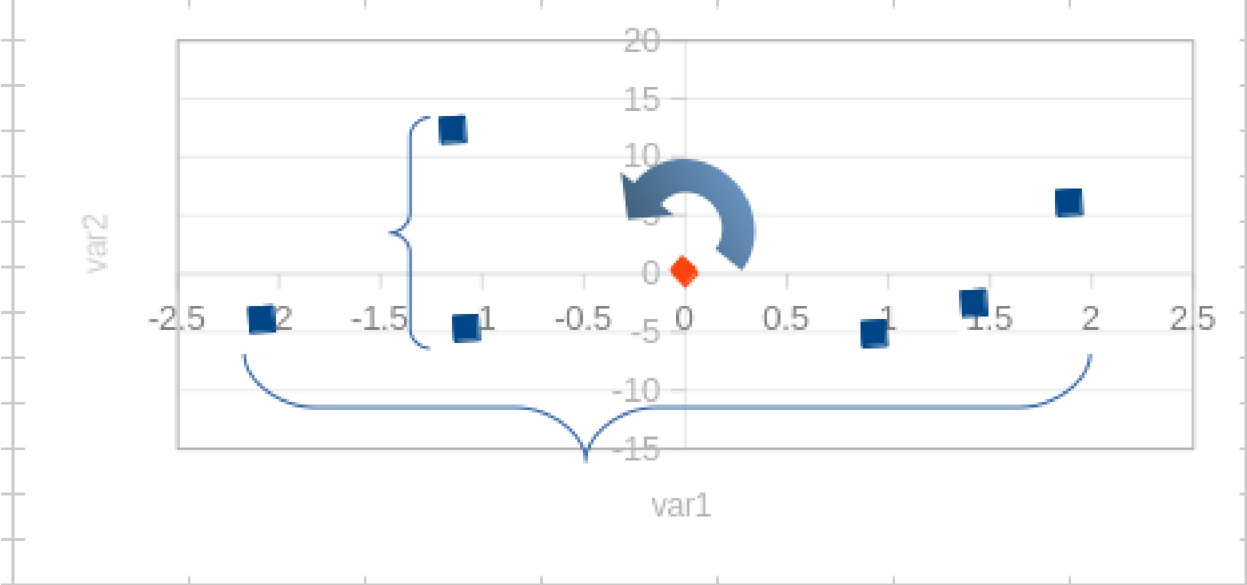
varname	id1	id2	id3	id4	id5	id6
var1	5	6	3	5.5	3	2
var2	10	20	30	12	13	15



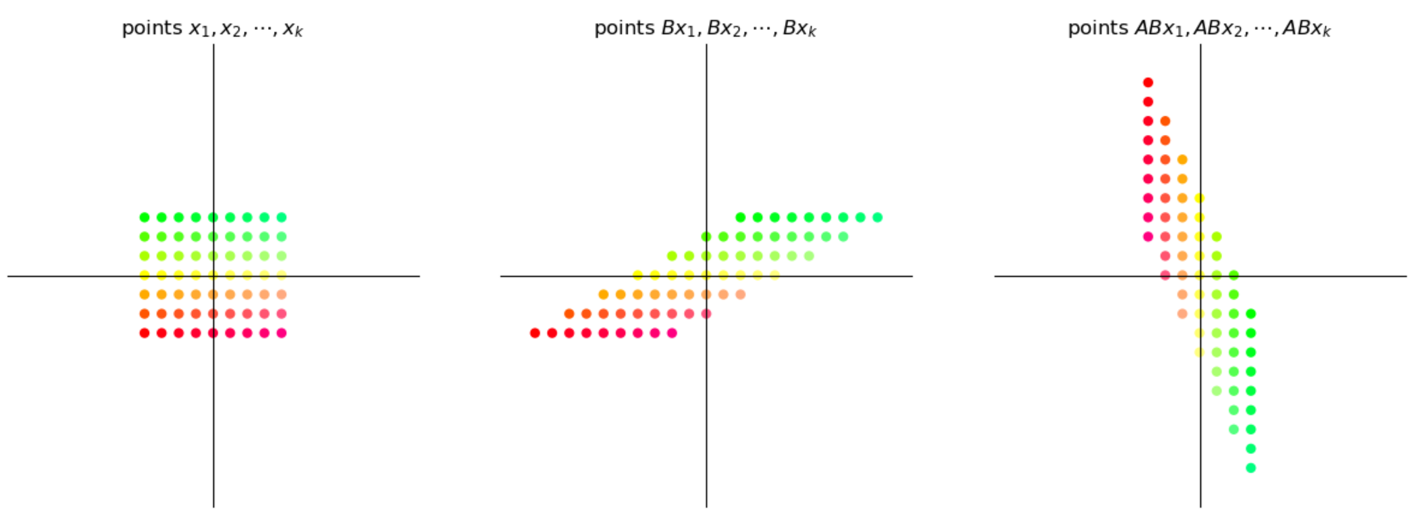
varname	id1	id2	id3	id4	id5	id6
var1	5	6	3	5.5	3	2
var2	10	20	30	12	13	15



varname	id1	id2	id3	id4	id5	id6
var1	5	6	3	5.5	3	2
var2	10	20	30	12	13	15



2.12 Rotating In Multiple Dimensions



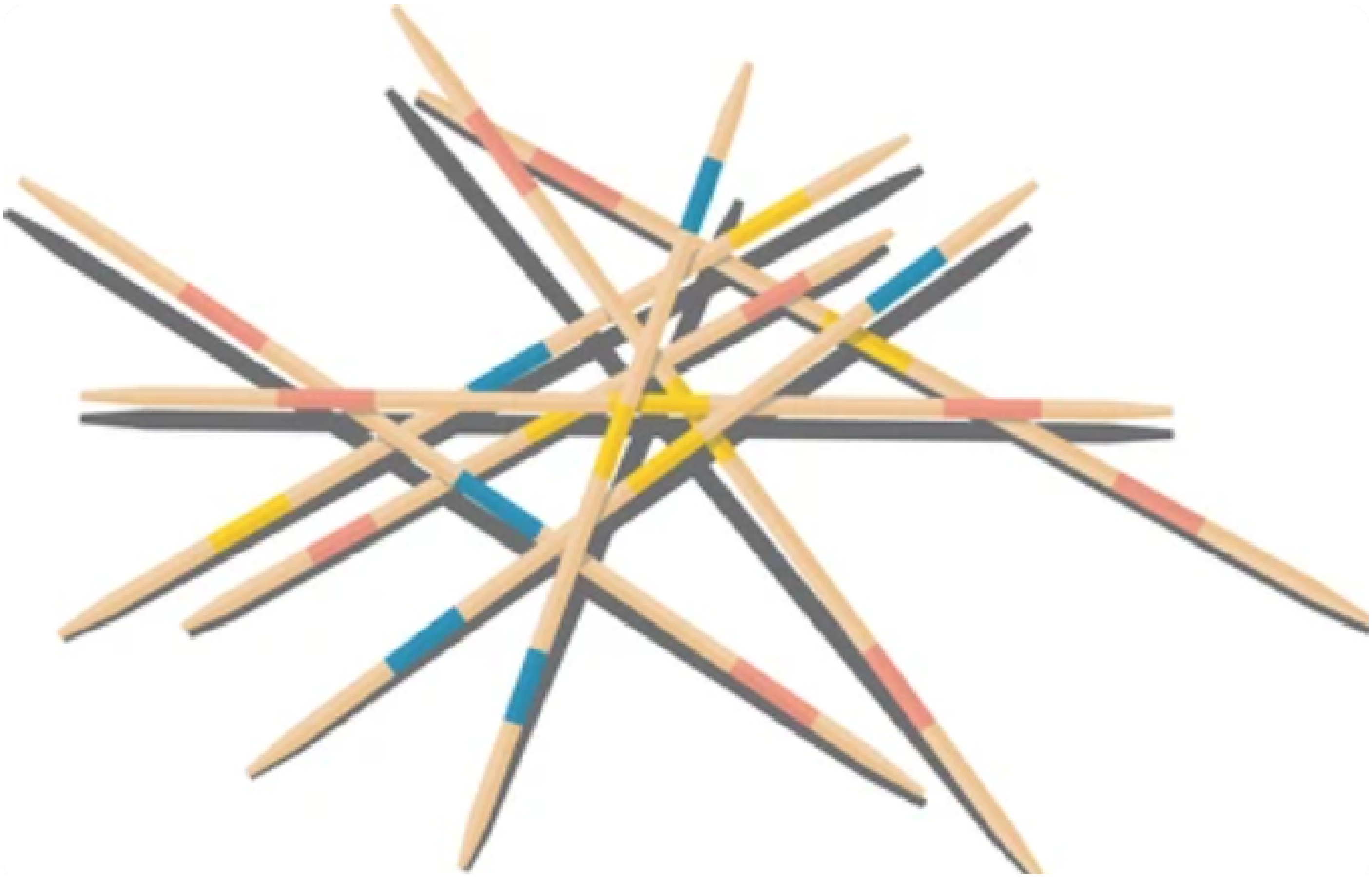
The orientation of each respective point never changes compared to the other points.

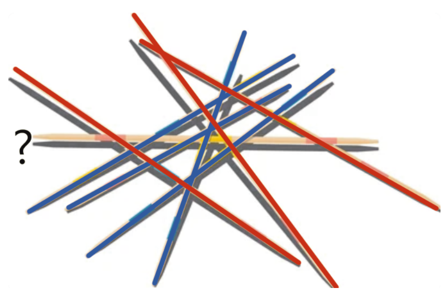
2.13 Uses

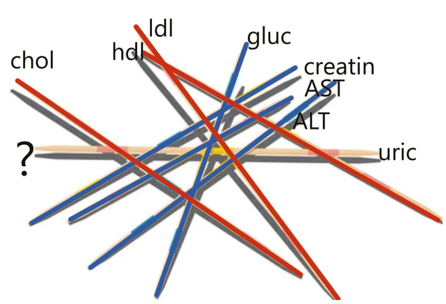
Data Reduction is for:

- Identifying similar variables
- Grouping variables by similarities
- Addressing variable intercorrelation
- Minimizing the number of variables you need to address
- Validating variable group membership

2.14 Multivariable Relationships







2.15 Principal Components Analysis

Compiles original sets of variables into *principal components* (PC) based on correlation.

- PCs are new variables
- By definition, they are uncorrelated to other PCs
- Combines elements of several original and potentially correlated variables

Utilizes *eigenvalues* to quantify the amount of variance captured

- Maximizing captured variance = minimizing unallocated variability
- Addresses data fitness

2.16 Data Preparation Commands

General PCA can be done using function `princomp()` from the `stats` package

- Only works on numeric data

Package `PCAmixdata` can be used for mixed data

- Categorical and numeric variables need to be analyzed separately
- Categorical data PCA is called a *Correspondence Analysis*
- Results are subsequently merged
- PCA should not be done on missing data
- Variables with no variability must be removed

```
library(PCAmixdata)

df2 <- df %>%
  select(var1, var2, var3, ... varN)

df2 <- na.omit(df2)

pca.splitmatrix <- splitmix(dfx)
```

- `splitmix()` generates a matrix `pcaVars` that tags numeric and factor variables
- `pcaVars` contains two main sub-objects
  - `X.quanti` lists quantitative variables (numerics)
  - `X.quali` lists qualitative variables (factors)

2.17 PCA Commands

```
pcamix.1 <- PCAmix(X.quanti=pca.splitmatrix$X.quanti, X.quali=pca.splitmatrix$X.quali, rename.level=
```

- `PCAmix()` is the primary command
- `X.quanti=` specifies the list of numeric variables
- `X.quali=` specifies the list of factor variables
- `rename.level = TRUE` tells R to use the category values for each categorical factor when generating labels in the output
- `graph = FALSE` suppresses the automatic display of the component X/Y plots
- `ndim =` specifies the maximum number of components to be generated in the output

2.18 PCA Results Assessment

Results were stored in `pcamix.1` above. This object contains several sub-objects of importance.

```
summary(pcamix.1$scores)

pcamix.1$eig

pcamix.1$sqload

plot(pcamix.1, choice = "cor")

plot(pcamix.1, choice = "levels")
```

- `scores` are composite values of correlation across all variables for each patient
- `eig` is a matrix containing the *eigenvalues* and variance captured for each component
  - *Cumulative proportion* tells us how "important" the component is at capturing the available variance
- `sqload` contains the *factor loadings* for each variable and component
- `plot()` will plot the first two components by default for:
  - `choice = "cor"` plots the numeric variables only
  - 'choice = "levels", plots the factor variables only

2.19 Factor Analysis

Exploratory Factor Analysis - `library(FactoMineR)` for mixed data

- Identifies interrelationships among variables and factors
- No *a priori* assumptions about relationships

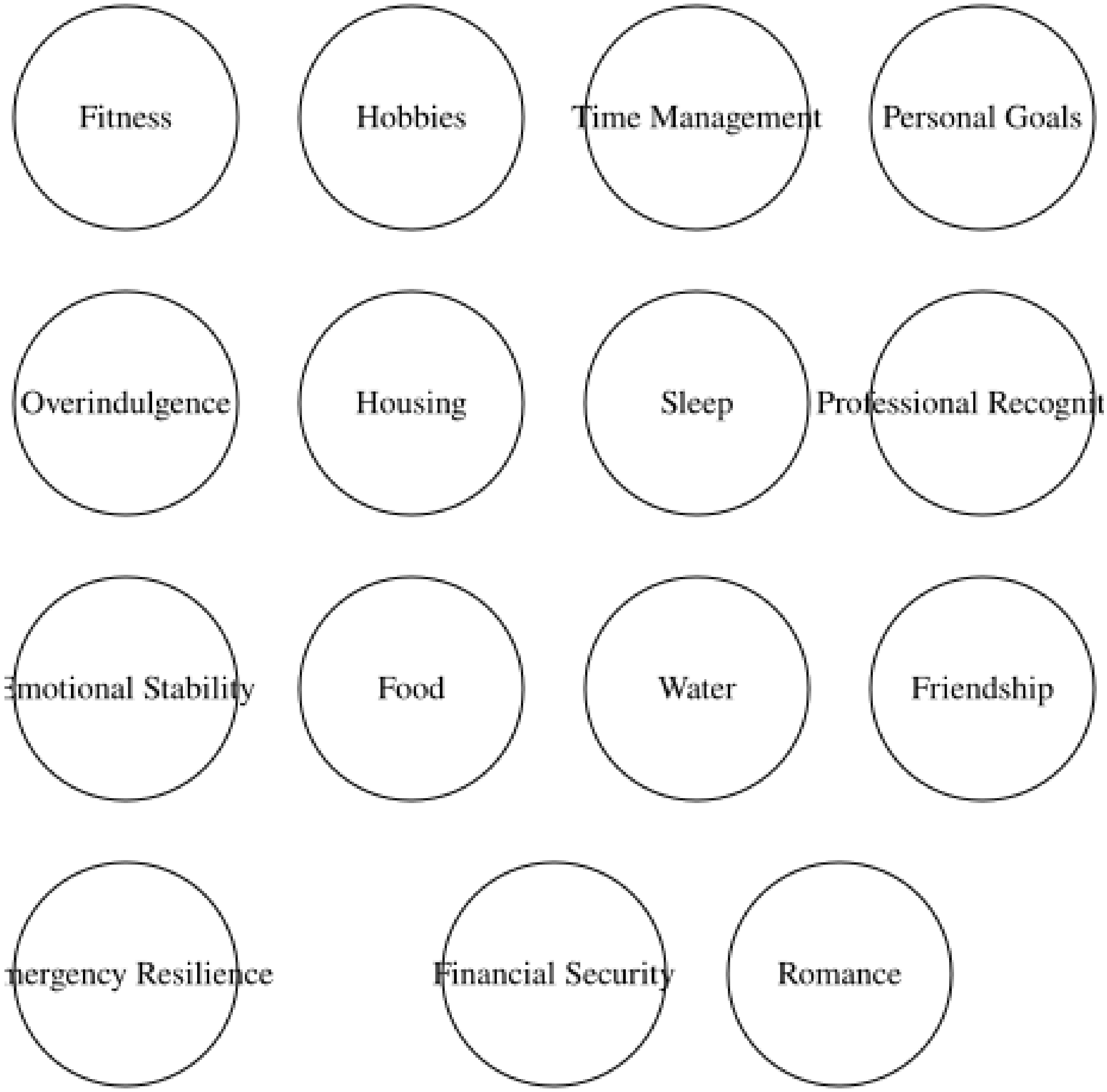
Confirmatory Factory Analysis - `library(lavaan)`

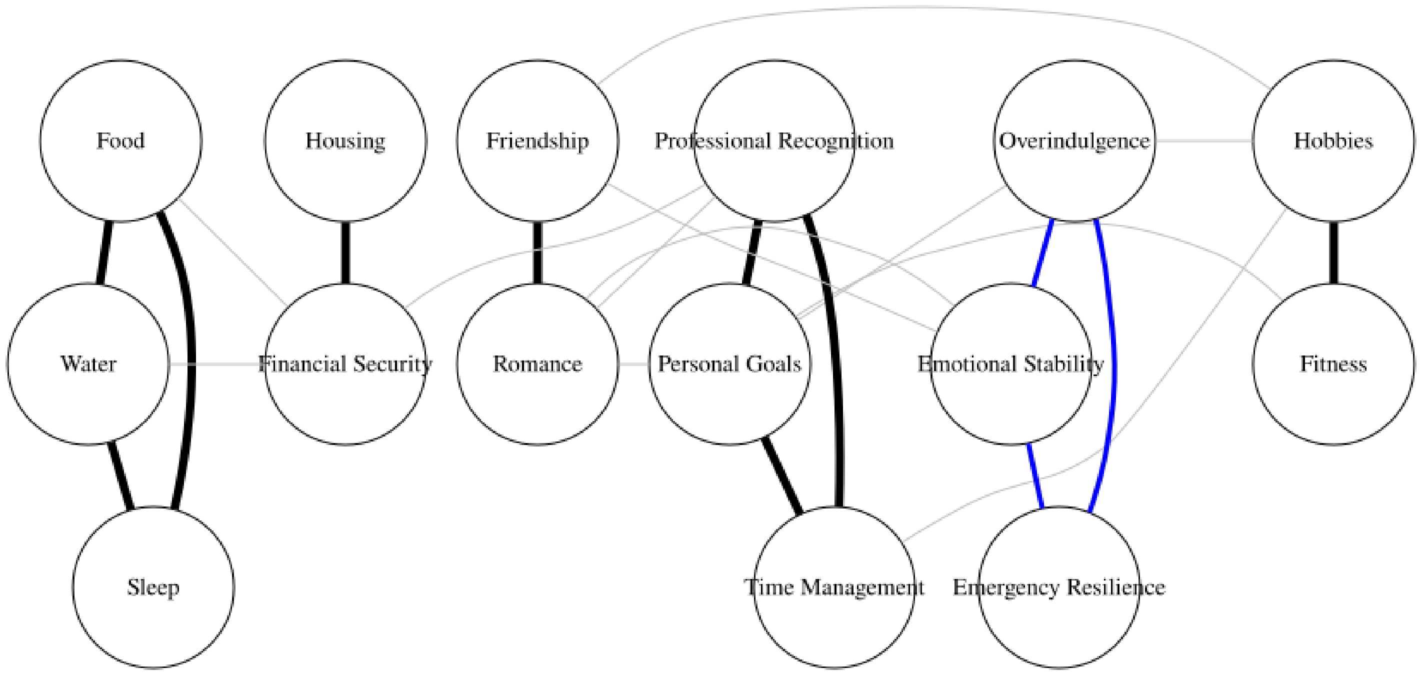
- Tests a hypothesis that a specific variable(s) is a member of a factor
- Leans on *Structural Equation Modeling* to perform

Structural Equation Modeling - `library(lavaan)`

- Identify *latent variables* and their member variables
- Requires some *a priori* information on assumed relationship

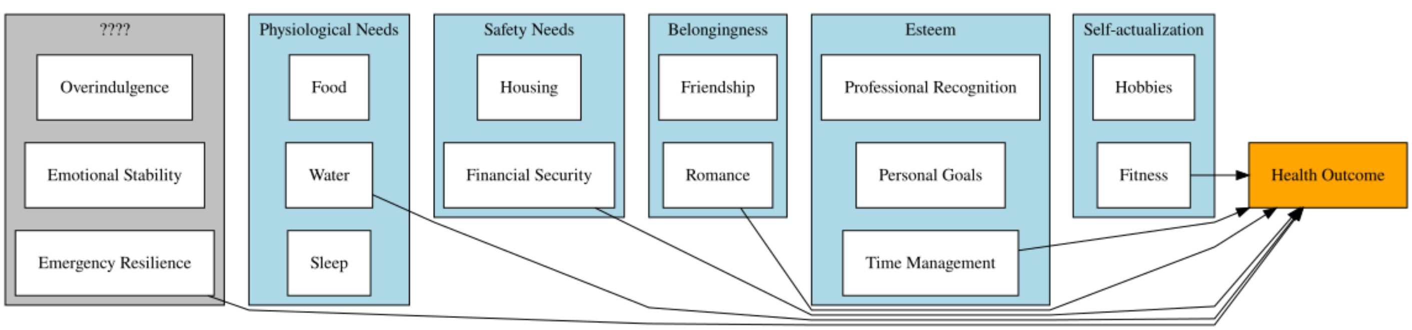
2.20 Identifying Variable Relationships





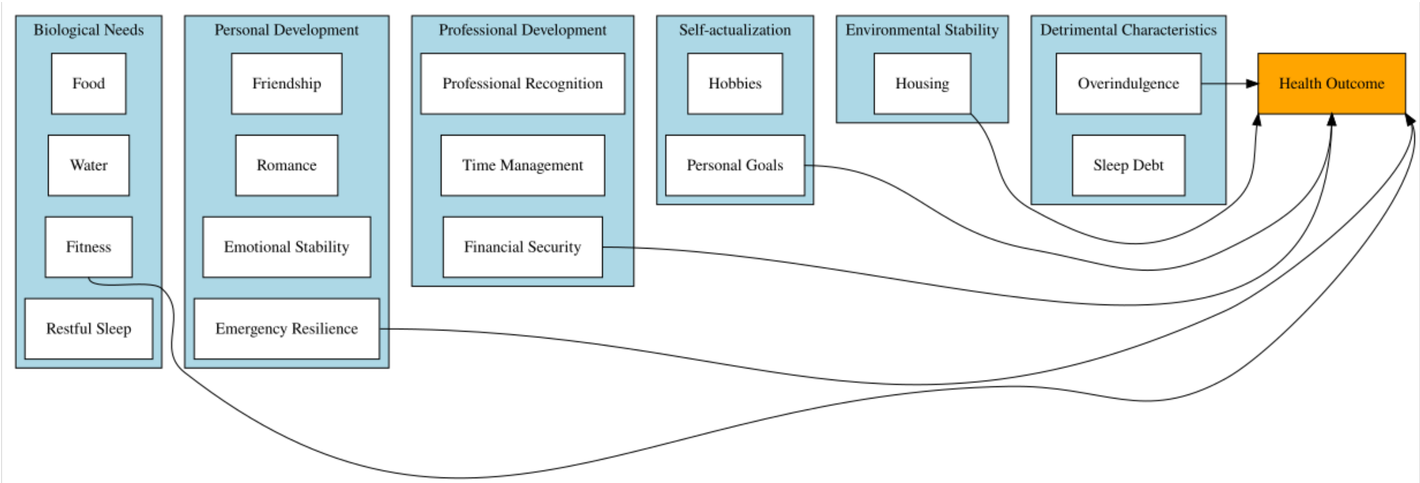
2.21 Latent Variables

Not directly observed, but their existence can be inferred through other variables.



Measures of *intercorrelation* can be used to measure *relatedness* of variables to domains and *exclusivity* of domains from each other.

2.22 Domain Specification



2.23 EFA Commands

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

efa.1 <- FAMD(df, graph = FALSE)

print(efa.1)
```

- `FAMD()` performs the factor analysis on data frame `df`
  - FAMD stands for Factor Analysis of Mixed Data
- `graph = FALSE` suppresses the automatic factor plots

2.24 Evaluating Captured Variance

```r
get_eigenvalue(efa.1)

fviz_screeplot(efa.1)
```

- `get_eigenvalue()` outputs the amount of variance captured by each factor
- `fviz_screeplot()` generates a screeplot (bar graph) of factors by the amount of captured variance in descending order

2.25 Evaluating Factor Membership

```r
fviz_famd_var(efa.1, repel = TRUE, axes = c(1, 2))

fviz_contrib(efa.1, choice = "var", axes = 1)

fviz_contrib(efa.1, choice = "var", axes = 2)
```

- `fviz_famd_var()` plots *correlation* of each variable and factors 1 and 2
  - `axes = c(1, 2)` specifies factor 1 as the X-axis and factor 2 as the Y-axis
  - Only two factors can be assessed per graph; can be changed in `axes = c()` parameter
  - `repel = TRUE` prevents variable labels from overlapping too much if there are many
- `fviz_contrib()` displays a screeplot of the hierarchy of member elements for a specified factor
  - `choice = "var"` specifies that you want to evaluate *variable* contributions to the factor
  - `axes =` specifies the factor that you want to evaluate contributions for

2.26 Inter-factor Variable Relationships

- Modifying the `choice =` parameter can output information specific to the member variables
- Correlation circle plots depict the *magnitude*, *direction*, and *correlation* between variables

2.26.1 For Numeric Variables

```r
fviz_famd_var(efa.1, choice = "quanti.var", repel = TRUE)
```

2.26.2 For Categorical Variables

```r
fviz_famd_var(efa.1, choice = "quali.var", repel = TRUE)
```

Results are analogous to PCA in terms of variable intercorrelation.

2.27 Example

I could run a ton of bivariate analyses but model development will be a pain because there is so much room for collinearity issues given our sample size.

I also don't necessarily understand the purpose or nuance behind every individual variable, but I don't want to omit important things.

We used several of the variables for the cluster analysis but never addressed the string variables.

Could we identify a pattern to the triage variables with more information regarding the *chief complaint*?

Data were manipulated and contain the standard emergency department `triage` variables plus many dichotomous features from the `chiefcomplaint` column.

There are 12 total variables for this analysis:

- 7 continuous variables
- 5 categorical variables with 10 categories

2.28 Tagging Categorical and Continuous

```
library(PCAmixdata)
splitVars <- splitmix(dfx)
```

```
Warning in splitmix(dfx): Columns of class integer are considered as
quantitative
```

2.29 Perform PCA

```
pcamix <- PCAmix(X.quanti=splitVars$X.quanti, X.quali=splitVars$X.quali, rename.level=TRUE, graph=FA
head(pcamix$scores)
```

```
        dim 1      dim 2      dim 3       dim 4      dim 5
1  0.47188238 -0.2280557  0.3050047 -0.96222769 -0.2245945
2 -0.54135573 -0.7268893  0.3720344 -0.20720713 -0.2151302
3  4.71424500  4.3509443  0.9761558  0.51735660  1.3454705
4  2.18558565 -0.4014262  0.6355010  0.07116661 -1.2321934
5  2.20387821  0.1382816 -3.8086910 -0.41444233 -1.0309084
6  0.01014832  1.1024173 -0.9016809 -0.25197885 -0.8272652
```

- Eight (8) PCs were evaluated at this step
- `head()` displays the top 6 observations and their correlation `score` values for each of the PCs
  - This is done just to get a sense of the range and scale of the scores

2.30 Evaluate the Eigenvalues

```
pcamix$eig
```

```
       Eigenvalue Proportion Cumulative
dim 1   1.7382864  14.485720   14.48572
dim 2   1.5649839  13.041533   27.52725
dim 3   1.3780160  11.483467   39.01072
dim 4   1.1109194   9.257662   48.26838
dim 5   1.0423805   8.686504   56.95489
dim 6   0.9979125   8.315938   65.27082
dim 7   0.9416637   7.847198   73.11802
dim 8   0.8531387   7.109489   80.22751
dim 9   0.7374707   6.145589   86.37310
dim 10  0.6122151   5.101793   91.47489
dim 11  0.5537681   4.614734   96.08963
dim 12  0.4692448   3.910373  100.00000
```

- It takes all 12 PCs to get all 100% of the variance in this data
- Each PC captures between 3.9% and 14.5% percent of the variance
- 7 PCs will capture $>$ 70% of the variance

2.31 Assess Loadings

Squared `loadings` represent the proportion of a variable's variance explained by a specific factor.

- r^2 for continuous variables and the respective component
- correlation ratios for each categorical variable with the respective component

```
pcamix$sqload
```

```
                  dim 1        dim 2        dim 3       dim 4        dim 5
temperature  0.04035892 0.0094850936 0.0795270129 0.026179754 8.827917e-02
heartrate    0.20943348 0.0460169153 0.1694943802 0.212850608 1.555976e-05
resprate     0.17516632 0.2816209861 0.0035820510 0.003692274 1.715446e-01
o2sat        0.18570129 0.0682606319 0.0001226608 0.210085626 3.821548e-02
sbp          0.27535678 0.1870432732 0.1467237085 0.150030146 1.063303e-02
dbp          0.01953659 0.0002857642 0.2053287616 0.213288135 9.834549e-03
lnlos        0.22382524 0.0095229122 0.0000410005 0.057826046 2.165240e-01
weakness     0.02420867 0.0639687649 0.0169305220 0.124448176 4.608646e-01
highpain     0.07303564 0.2534533227 0.1178238685 0.034874695 4.197263e-02
pain_noscore 0.08153774 0.4845855843 0.0119834188 0.016691719 1.636655e-04
hypotensive  0.24487838 0.0613284905 0.3243918271 0.056249889 4.071646e-03
racewhite    0.18524736 0.0994121833 0.3020668299 0.004702348 2.615231e-04
```

- Factor 1 main correlates include: `sbp` (0.275), `hypotensive` (0.245), `lnlos` (0.224), `heartrate` (0.209)
- Factor 2 main correlates include: `pain_noscore` (0.485), `resprate` (0.282), `highpain` (0.253)
- Factor 3 main correlates include: `hypotensive` (0.324), `racewhite` (0.302), `dbp` (0.205)

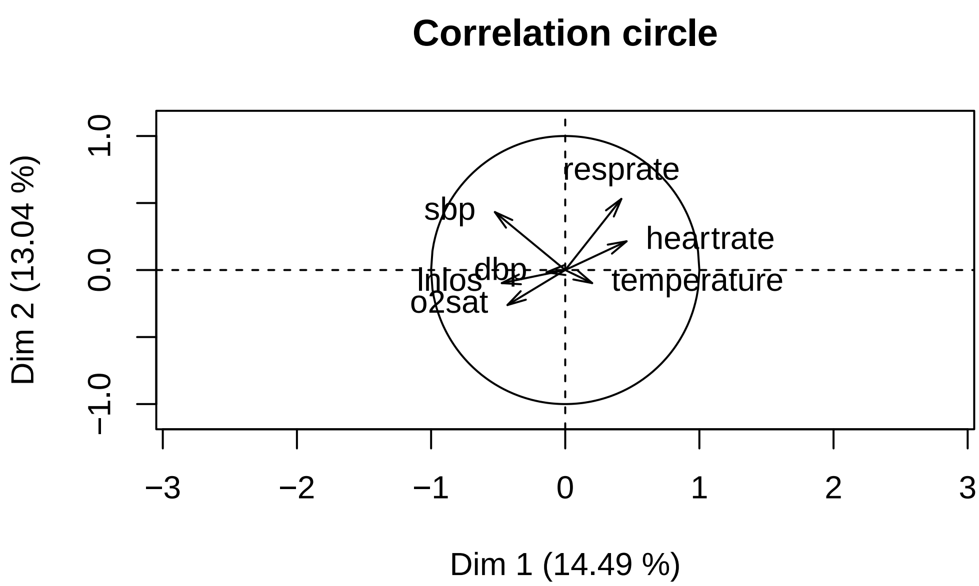
2.32 Assess Component Membership

Only plots the first two dimensions by default.

Categorical and continuous must be done separately.

2.32.1 Continuous Variables

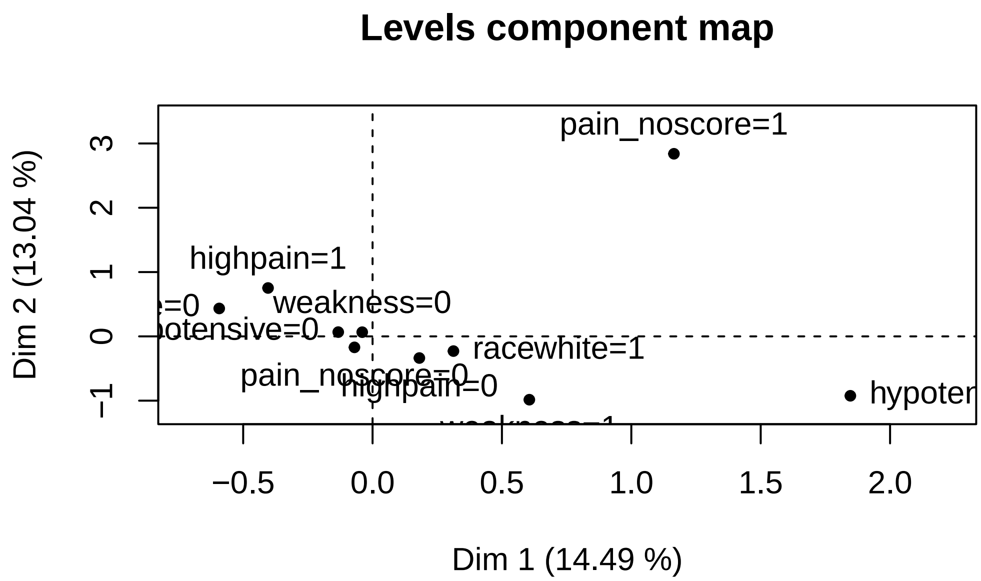
```
plot(pcamix, choice = "cor")
```



- Variables that are close and direction-aligned are positively correlated to each other
- Longer arrow distance means better variable representation between these dimensions
- Arrows in opposing directions are variables that are negatively correlated

2.33 Categorical Variables

```
plot(pcamix, choice = "levels")
```



- Categorical variable levels are individually graphed
- Distance from origin means better representation of that category for the dimension

2.34 EFA Example

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

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

```
famd <- FAMD(dfy, graph = FALSE)
print(famd)
```

```
*The results are available in the following objects:

  name          description
1 "$eig"        "eigenvalues and inertia"
2 "$var"        "Results for the variables"
3 "$ind"        "results for the individuals"
4 "$quali.var"  "Results for the qualitative variables"
5 "$quanti.var" "Results for the quantitative variables"
```

2.35 Checking Eigenvalues

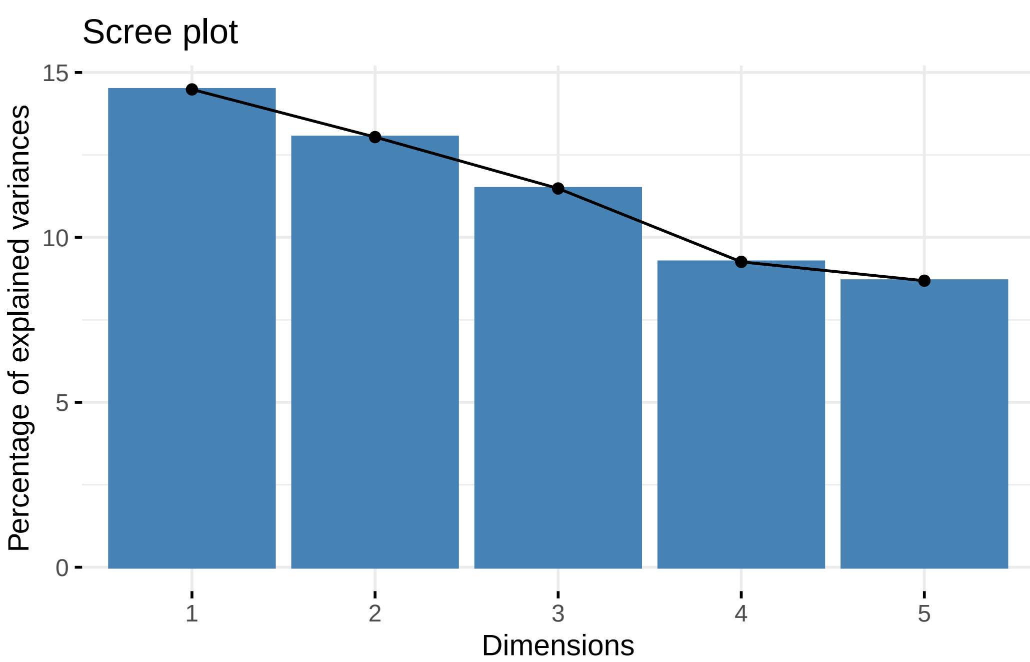
```
eigenvals <- get_eigenvalue(famd)
head(eigenvals)
```

```
      eigenvalue variance.percent cumulative.variance.percent
Dim.1   1.738286        14.485720                    14.48572
Dim.2   1.564984        13.041533                    27.52725
Dim.3   1.378016        11.483467                    39.01072
Dim.4   1.110919         9.257662                    48.26838
Dim.5   1.042380         8.686504                    56.95489
```

- 5 dimensions captures 56.95% of the variance.
- this is the same as the PCA

2.36 Scree Plot

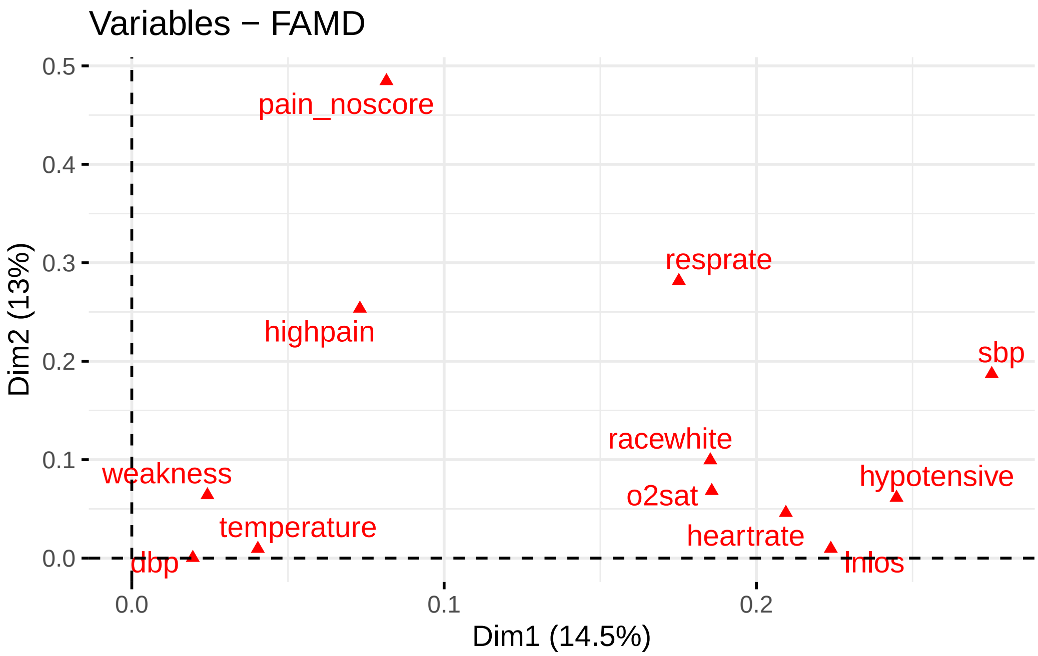
```
fviz_screeplot(famd)
```



2.36.1 Evaluating Variables

Right now, we'll focus on just the first two dimensions for simplicity

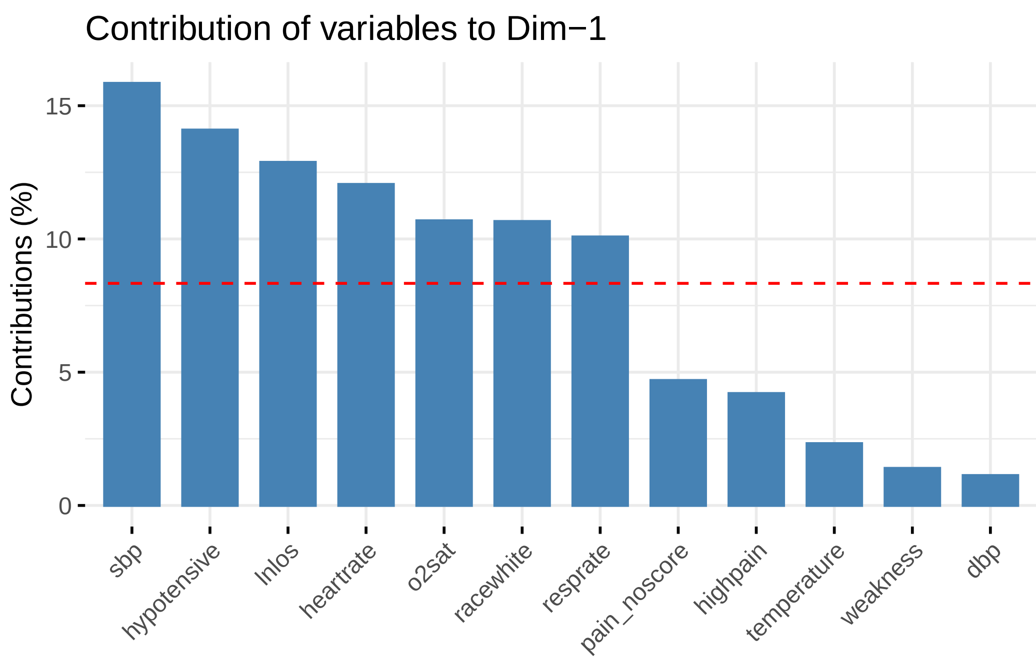
```r
fviz_famd_var(famd, repel = TRUE)
```



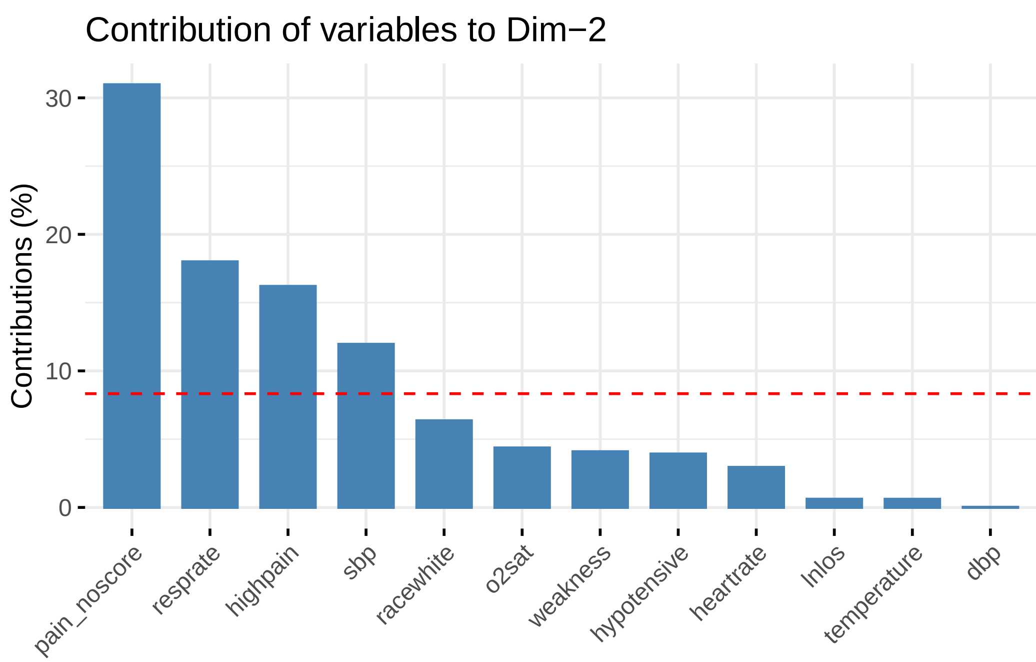
- Factor Analysis will visualize categorical and continuous variables on the same dimension plot
- Categorical variables are taken as whole; levels are not parsed yet

2.37 Scree Plots

```r
fviz_contrib(famd, "var", axes = 1)
```



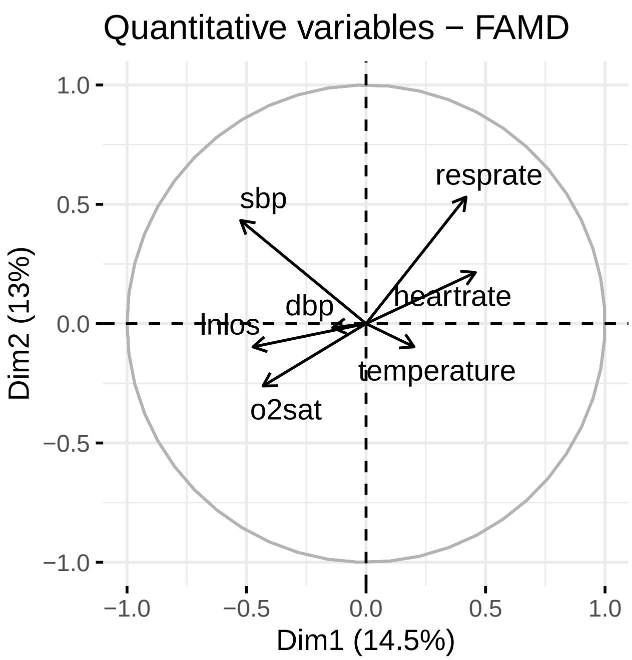

```r
fviz_contrib(famd, "var", axes = 2)
```



Primary contributing factors for each dimension are listed in order.

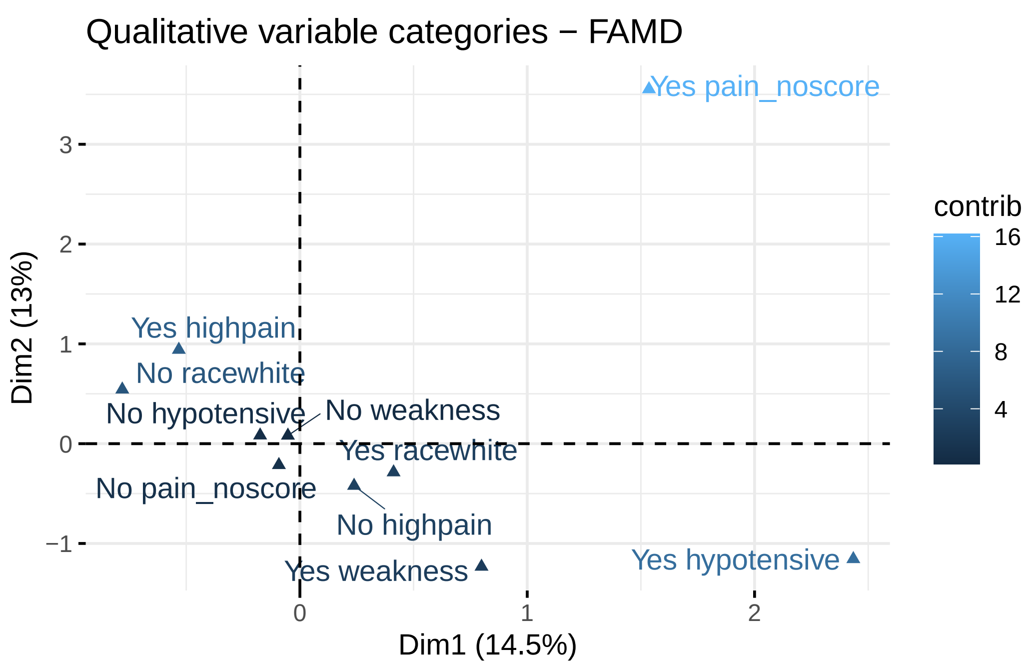
2.38 Correlation Circle Graphs

```r
fviz_famd_var(famd, choice = "quanti.var", repel = TRUE, col.var = "black")
```



Same type of correlation circle showing "polarity" of vars within these two dimensions.

```r
fviz_famd_var(famd, choice = "quali.var", repel = TRUE, col.var = "contrib")
```



Parsed values for categorical variables are graphed.

Color shows the intensity of the contribution based on distance from the origin.

3 References

3.1 Tutorials

http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/115-famd-factor-analysis-of-mixed-data-in-r-essentials/

https://bookdown.org/sz_psyc490/r4psychometics/factor-analysis.html

https://chavent.github.io/PCAmixdata/PCAmixcompare.html

3.2 Package Documentation

https://rdrr.io/github/chavent/PCAmixdata/man/PCAmix.html

http://factominer.free.fr/

https://lavaan.ugent.be/