Intro to EFA and data factorability

DIMENSIONALITY REDUCTION IN R

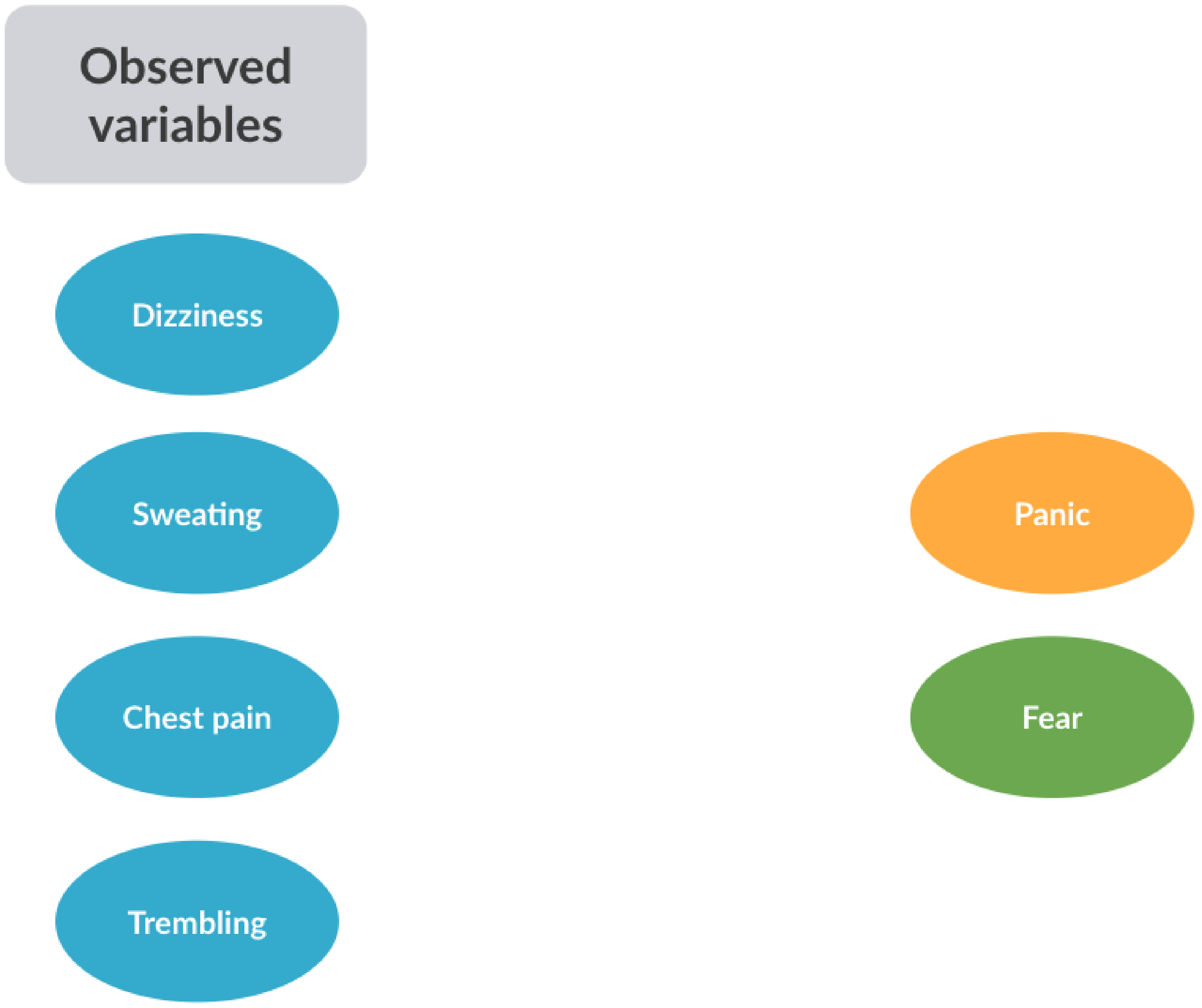
Alexandros Tantos
Assistant Professor, Aristotle University of Thessaloniki

EFA: a realistic model for reducing and exploring

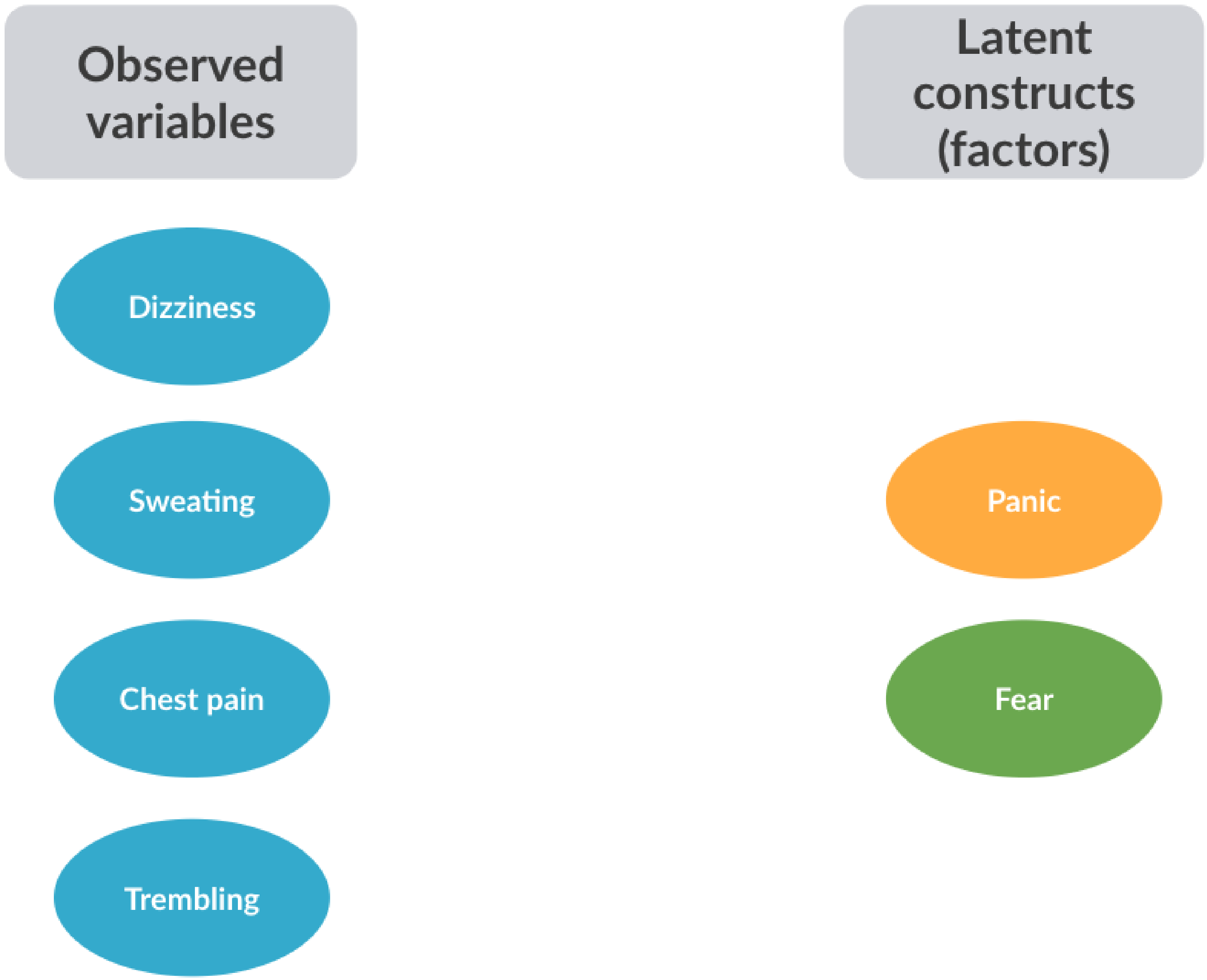
- Variance/covariance are **only partially** explained by *factors*
- Factors are labels for the underlying constructs
- Causal relationship between factors and observed variables

EFA: Measuring the unobserved

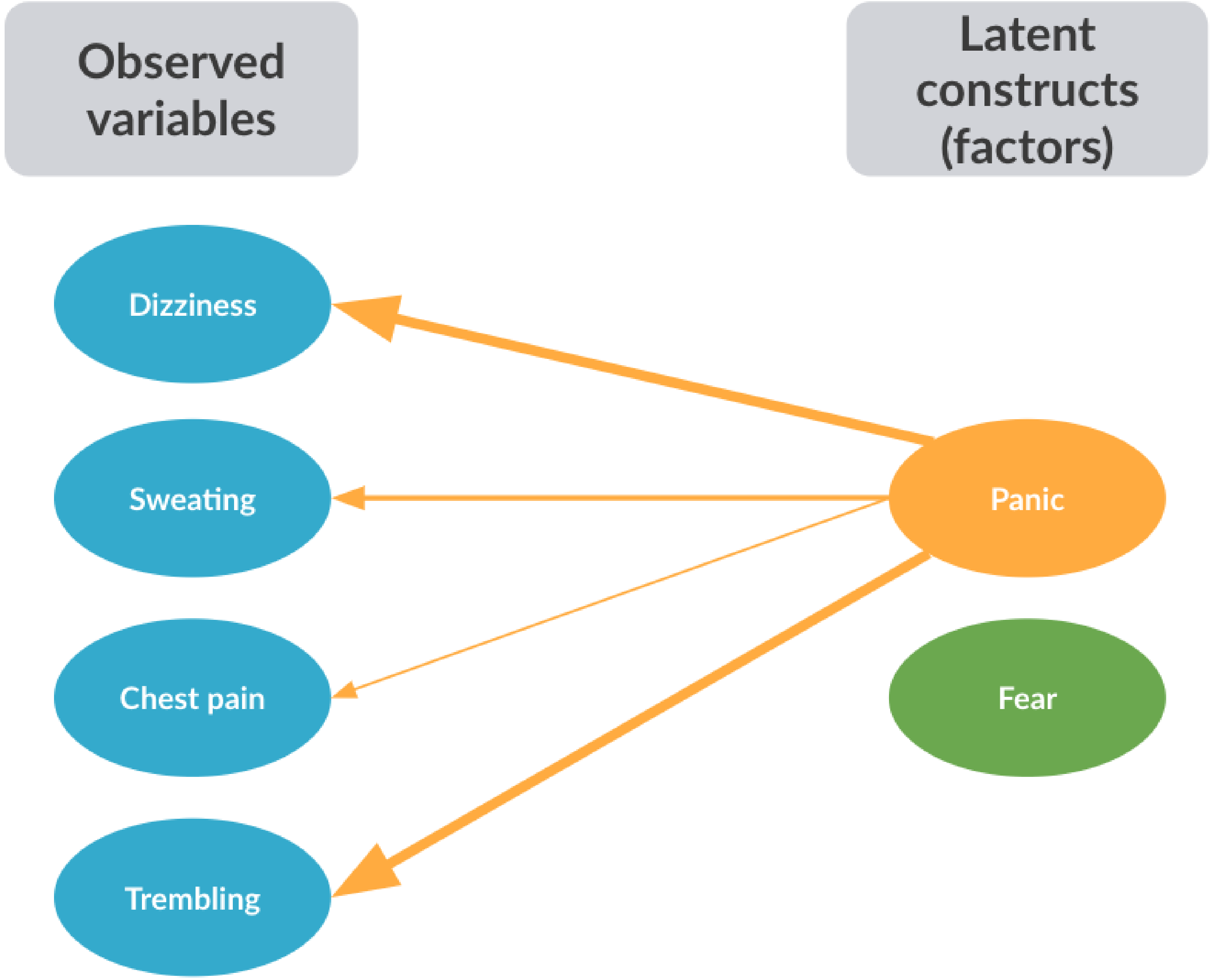
EFA: Measuring the unobserved



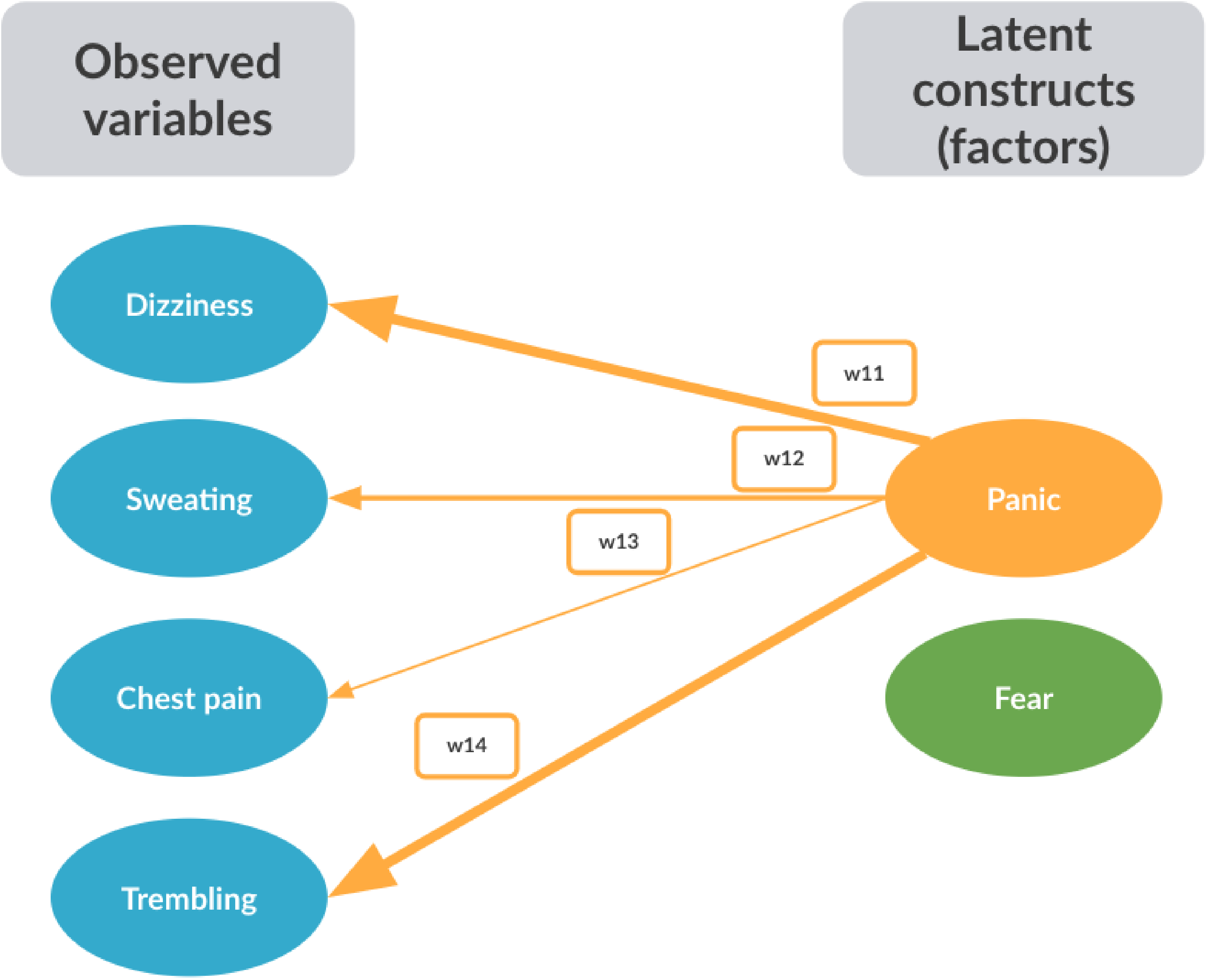
EFA: Measuring the unobserved



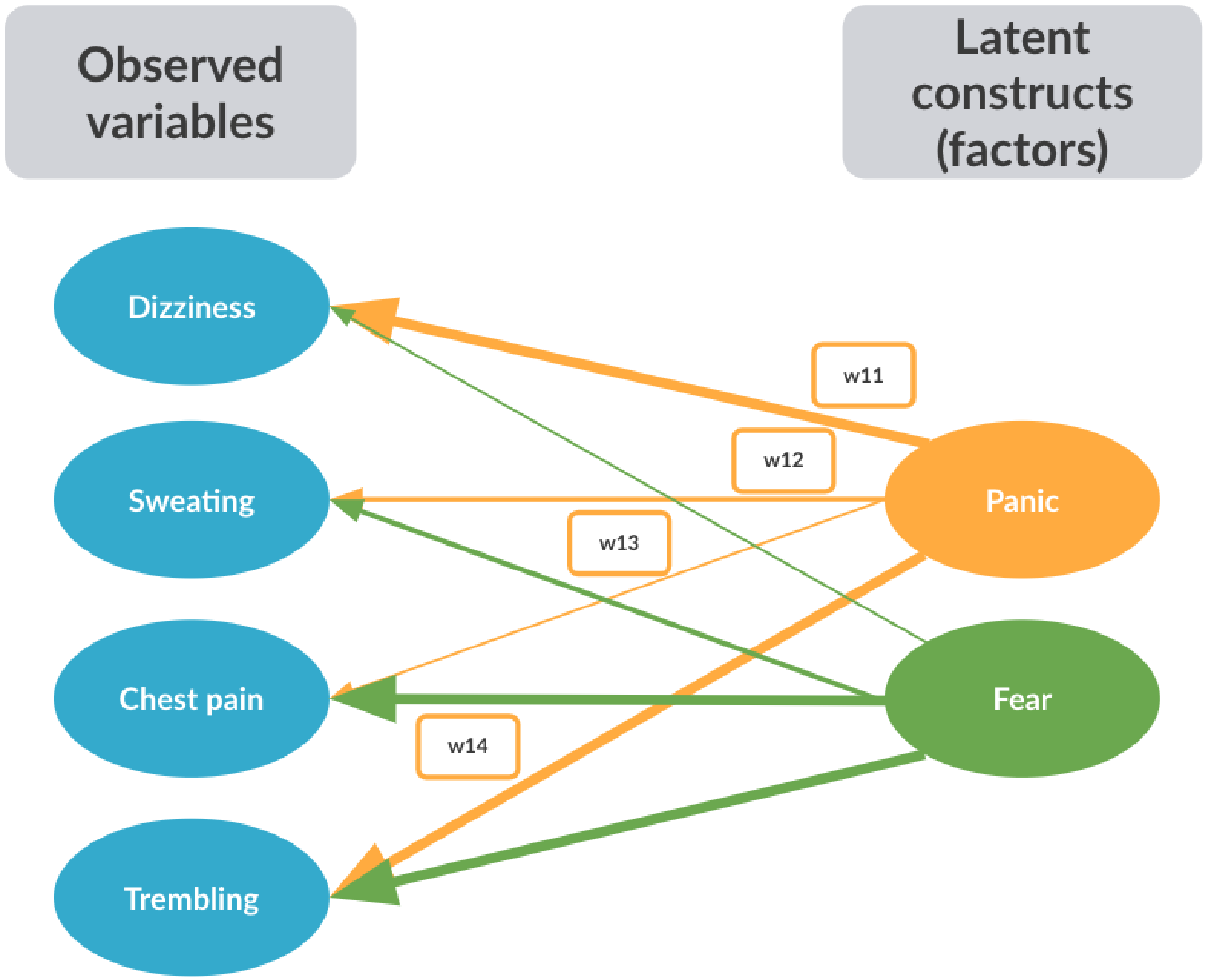
EFA: Measuring the unobserved



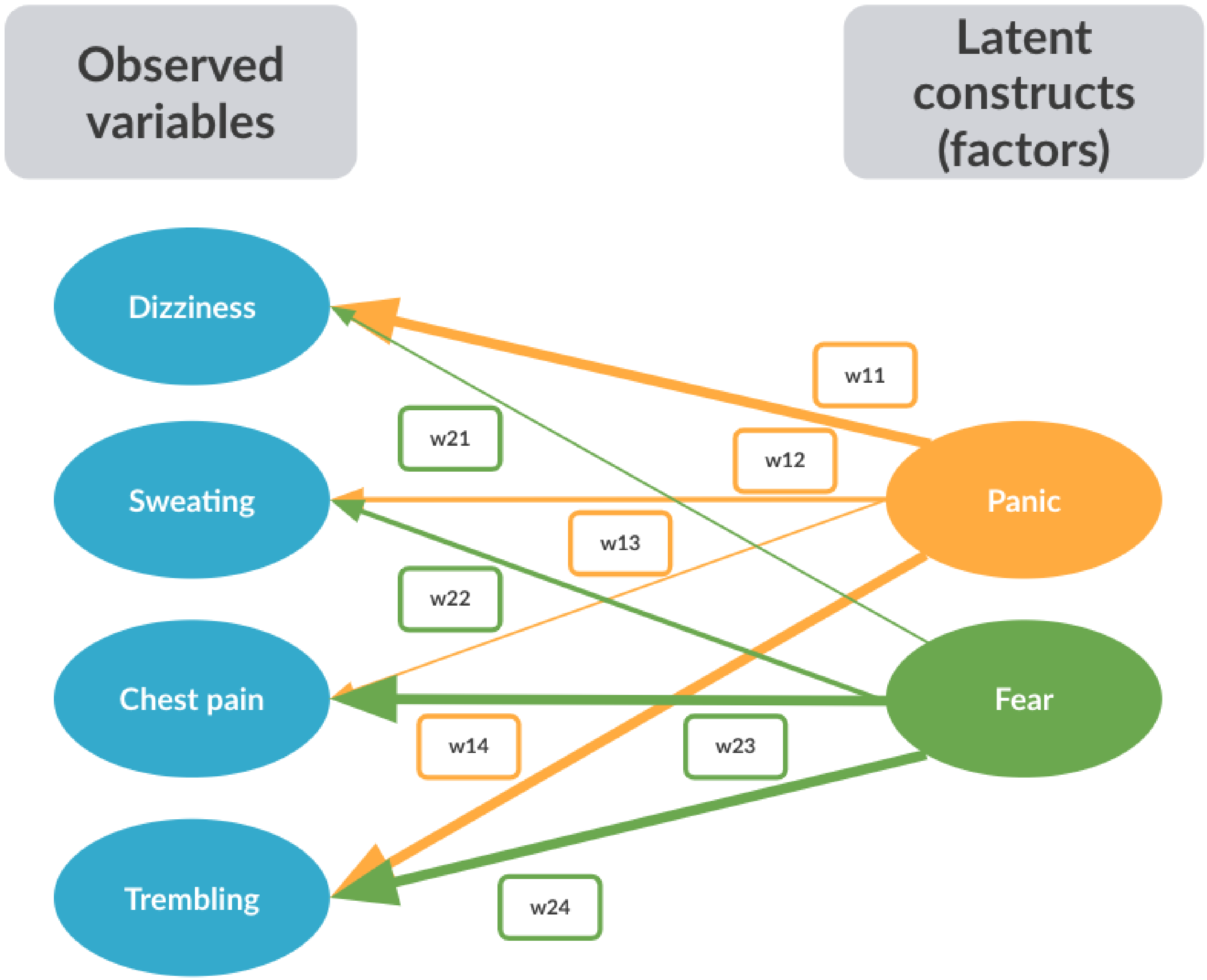
EFA: Measuring the unobserved



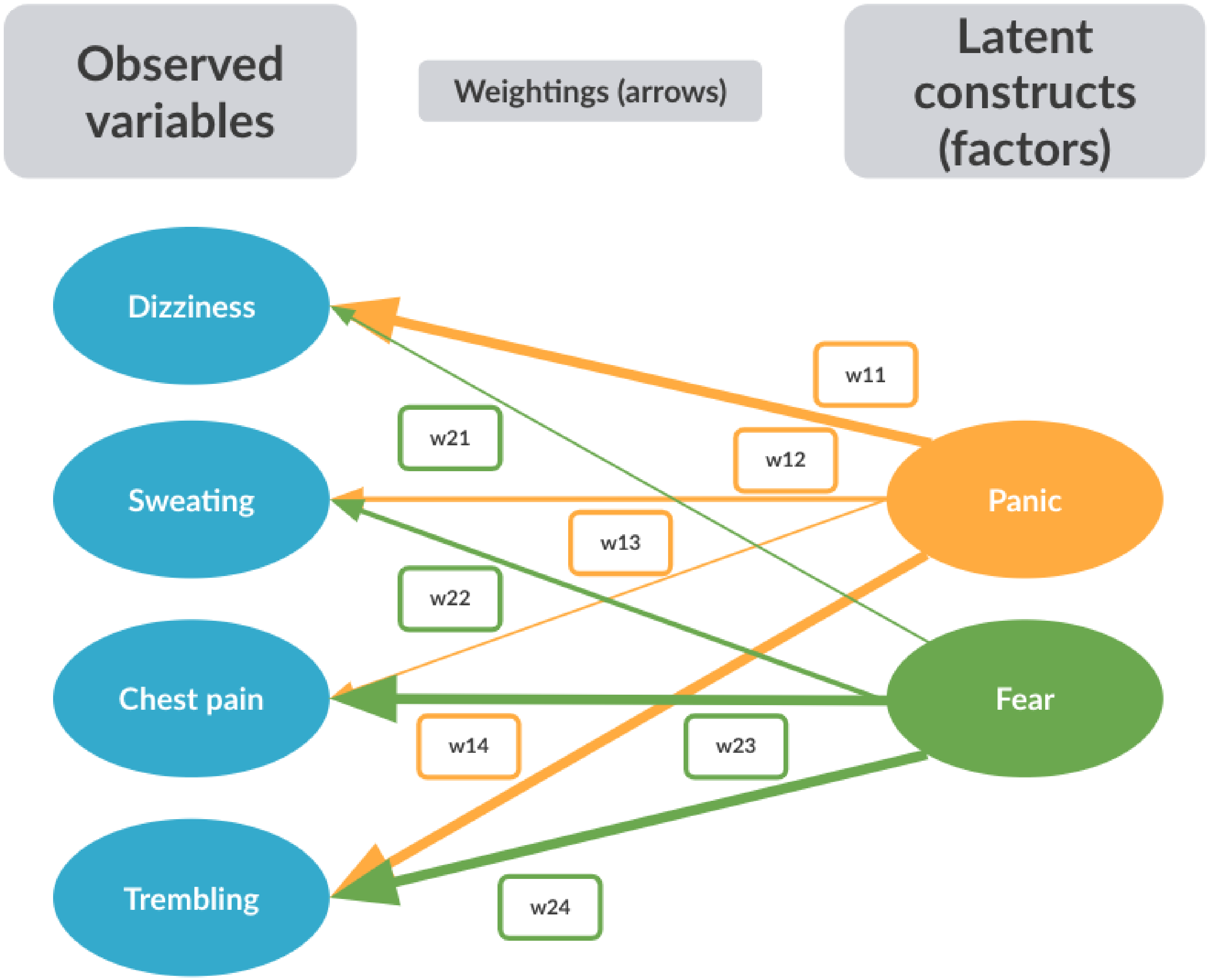
EFA: Measuring the unobserved



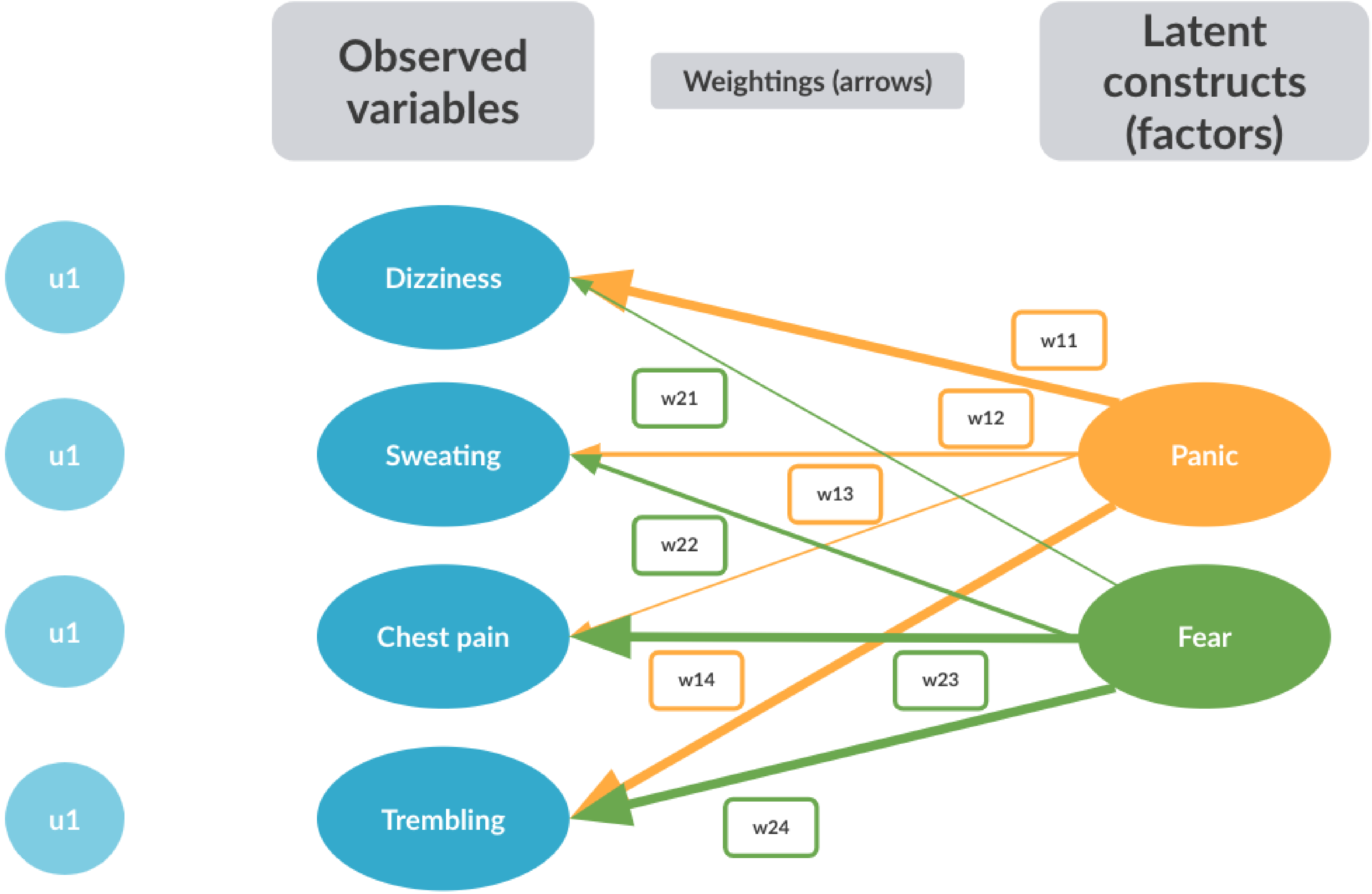
EFA: Measuring the unobserved



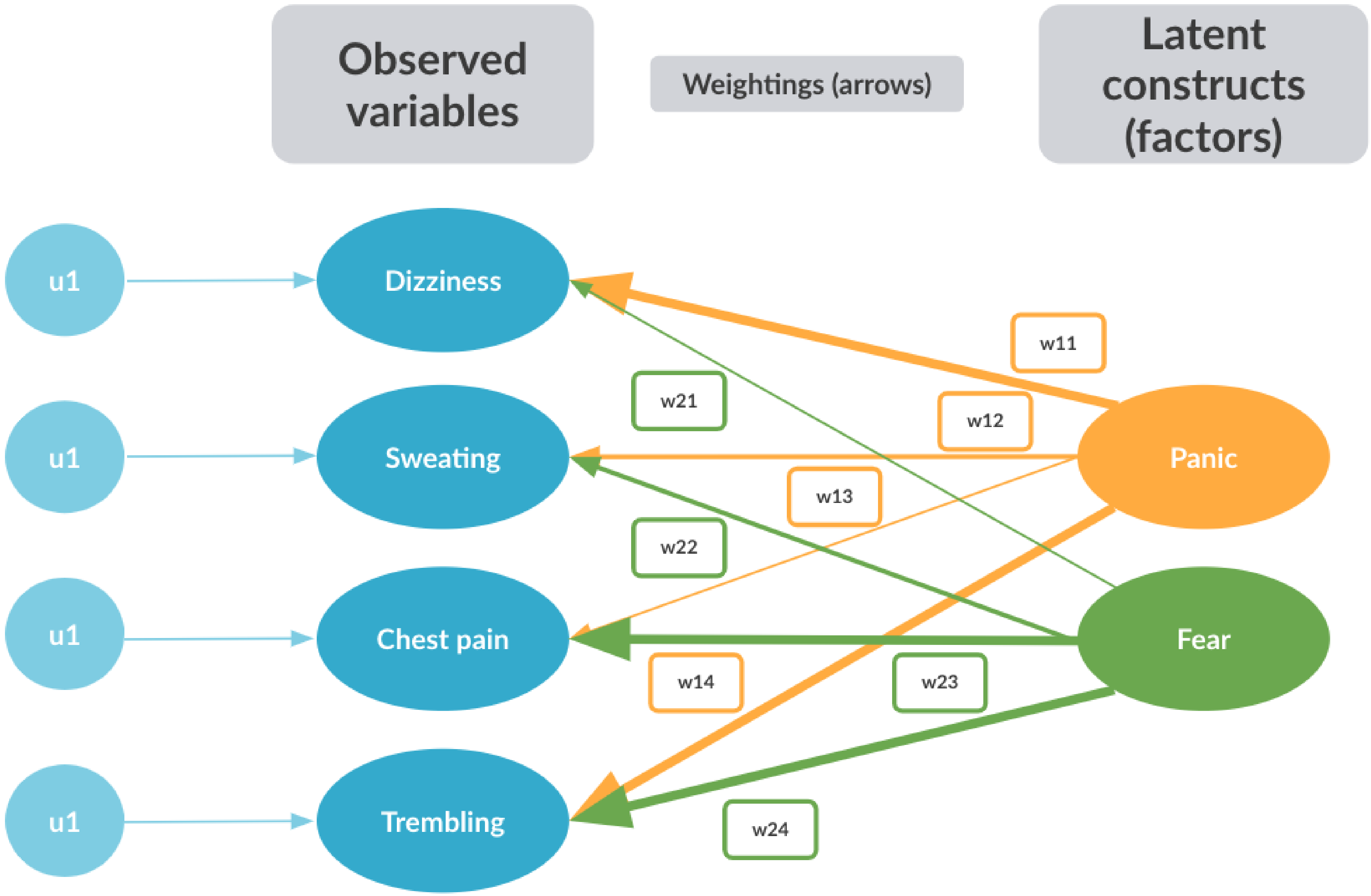
EFA: Measuring the unobserved



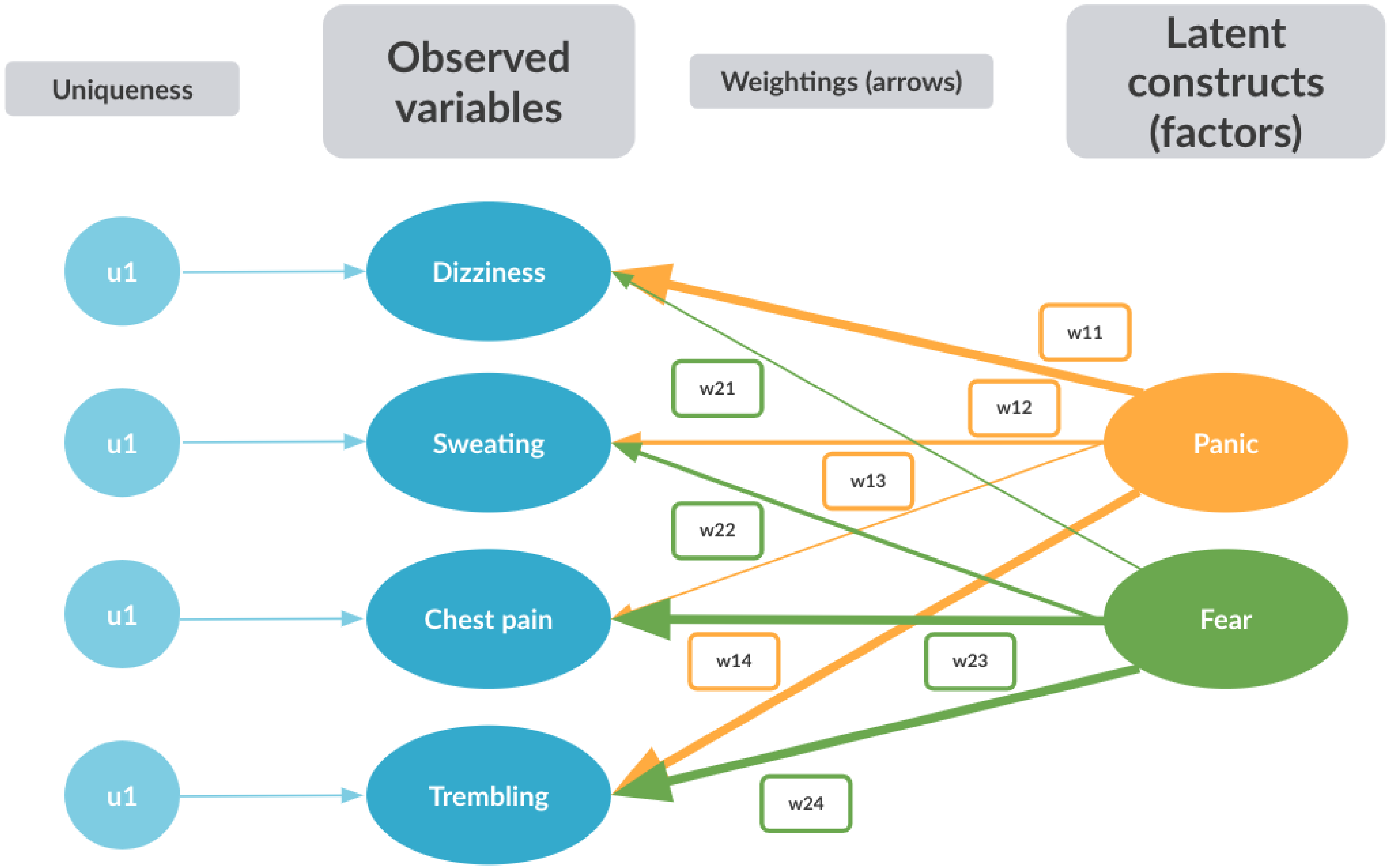
EFA: Measuring the unobserved



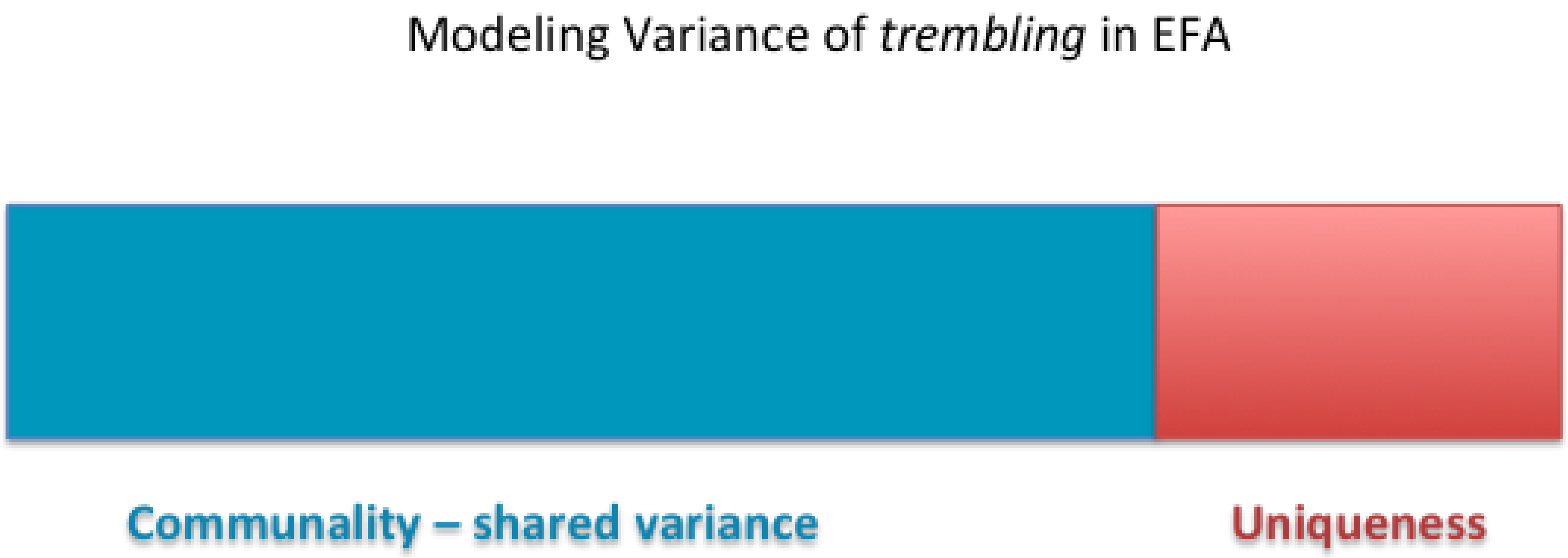
EFA: Measuring the unobserved



EFA: Measuring the unobserved



EFA: A realistic model of explaining variance



Steps to perform EFA

- Check for data factorability
- Extract factors
- Choose the "right" number of factors to retain
- Rotate factors
- Interpret the results

A first look at the bfi dataset

```r
library(psych)
data(bfi)
# Take a look at the head of bfi dataset.
head(bfi)
```

	A1	A2	A3	A4	A5	C1	C2	C3	C4	C5	E1	E2	E3	E4	E5	N1	N2	N3	N4	N5	O1	O2	O3	O4	O5	gender	education	age
61617	2	4	3	4	4	2	3	3	4	4	3	3	3	4	4	3	4	2	2	3	3	6	3	4	3	1	NA	16
61618	2	4	5	2	5	5	4	4	3	4	1	1	6	4	3	3	3	3	5	5	4	2	4	3	3	2	NA	18
61620	5	4	5	4	4	4	5	4	2	5	2	4	4	4	5	4	5	4	2	3	4	2	5	5	2	2	NA	17
61621	4	4	6	5	5	4	4	3	5	5	5	3	4	4	4	2	5	2	4	1	3	3	4	3	5	2	NA	17
61622	2	3	3	4	5	4	4	5	3	2	2	2	5	4	5	2	3	4	4	3	3	3	4	3	3	1	NA	17
61623	6	6	5	6	5	6	6	6	1	3	2	1	6	5	6	3	5	2	2	3	4	3	5	6	1	2	3	21

Let's practice!

DIMENSIONALITY REDUCTION IN R

Intro to EFA: data factorability

DIMENSIONALITY REDUCTION IN R

Alexandros Tantos
Assistant Professor, Aristotle University of Thessaloniki

Steps to perform EFA

- **Check for data factorability**
- Extract factors
- Choose the "right" number of factors to retain
- Rotate factors
- Interpret the results

Factorability tests:

- The `Bartlett sphericity` test
- The `Kaiser-Meyer-Olkin (KMO)` test

The Bartlett sphericity test

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

A 4X4 identity matrix

- H0: There is **no** significant difference between the correlation matrix and the identity matrix of the same dimensionality.
- H1: There is significant difference betweeen them and, thus, we have strong evidence that there are underlying factors.

```r
library(polycor)
# A subset of the bfi dataset.
bfi_s <- bfi[1:200, 1:25]
# Calculate the correlations.
bfi_hetcor <- hetcor(bfi_s)
# Retrieve the correlation matrix.
bfi_c <- bfi_hetcor$correlations
# Apply the Bartlett test.
bfi_factorability <- cortest.bartlett(bfi_c)
bfi_factorability
```

```
$chisq
[1] 891.1536

$p.value
[1] 5.931663e-60

$df
[1] 300
```

The Kaiser-Meyer-Olkin (KMO) test for sampling adequacy

```r
library(psych)
KMO(bfi_c)
```

```
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = bfi_c)
Overall MSA =  0.76
MSA for each item =
  A1   A2   A3   A4   A5   C1   C2   C3   C4   C5   E1   E2   E3   E4   E5   N1   N2
0.66 0.77 0.69 0.73 0.75 0.74 0.79 0.76 0.76 0.74 0.80 0.81 0.79 0.81 0.83 0.70 0.67
  N3   N4   N5   O1   O2   O3   O4   O5
0.82 0.79 0.82 0.79 0.65 0.81 0.62 0.77
```

Let's practice!

DIMENSIONALITY REDUCTION IN R

Extraction methods

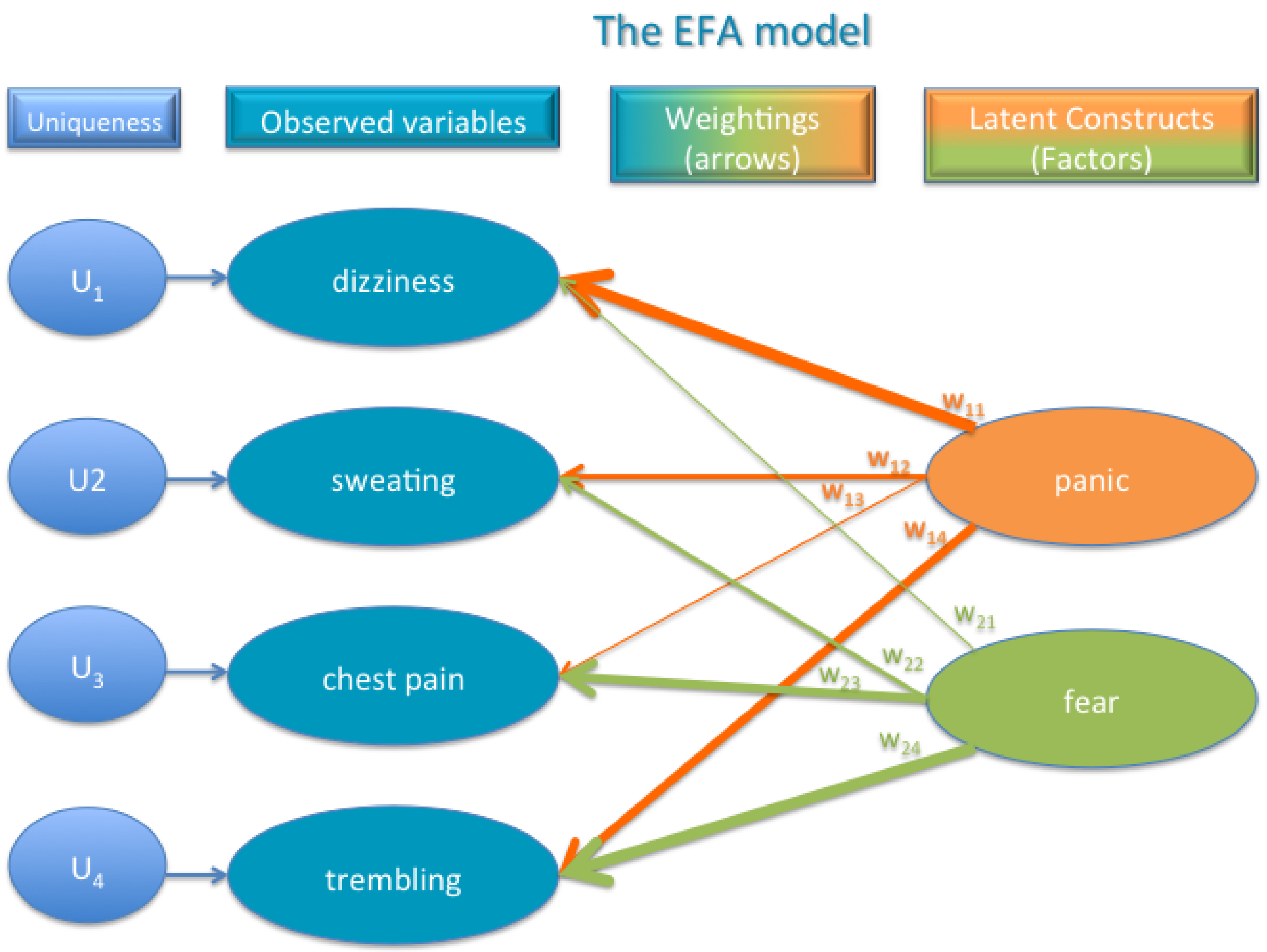
DIMENSIONALITY REDUCTION IN R

Alexandros Tantos
Assistant Professor, Aristotle University of Thessaloniki

Steps to perform EFA

- Check for data factorability
- **Extract factors**
- Choose the "right" number of factors to retain
- Rotate factors
- Interpret the results

Methods for extracting factors



`EFA` aims to:

- **extract factors**
- estimate factor loadings

Factor extraction with fa()

Extraction methods:

- `minres` : minimum residual [default] (slightly modified methods: `ols` , `wls` , `gls`)
- **`mle` : Maximum Likelihood Estimation (MLE)**
- **`paf` : Principal Axes Factor (PAF) extraction**
- `minchi` : minimum sample size weighted chi square
- `minrank` : minimum rank
- `alpha` : alpha factoring

Commonality:

- First extract the factor that accounts for the most variance, and then successively for factors that account for the most remaining variance.

The minres extraction method

```r
library(psych)
library(GPArotation)
# EFA with 3 factors
f_bfi_minres <- fa(bfi_c,
                   nfactors = 3,
                   rotate = "none")
# Sorted communality
f_bfi_minres_common <- sort(
                f_bfi_minres$communality,
                decreasing = TRUE
                )
# create a dataframe for an improved overview
data.frame(f_bfi_minres_common)
```

```
   f_bfi_minres_common
N1           0.6809294
E2           0.6564523
N2           0.5866483
N3           0.5394762
N4           0.4942059
E1           0.4744005
E5           0.4586935
E4           0.4580264
C1           0.4364326
N5           0.4119905
A5           0.3526680
C2           0.3256829
E3           0.3088069
A3           0.3051018
A2           0.2911182
O4           0.2818333
O3           0.2784802
C4           0.2478325
O1           0.2293049
C3           0.2095333
O5           0.2068315
C5           0.1727959
A1           0.1177920
A4           0.1091156
O2           0.0706517
```

The minres extraction method

```r
# Sorted uniqueness
f_bfi_minres_unique <- sort(
        f_bfi_minres$uniqueness,
        decreasing = TRUE)

# create a dataframe for an improved overview
data.frame(f_bfi_minres_unique)
```

```
   f_bfi_minres_unique
O2           0.9293483
A4           0.8908844
A1           0.8822080
C5           0.8272041
O5           0.7931685
C3           0.7904667
O1           0.7706951
C4           0.7521675
O3           0.7215198
O4           0.7181667
A2           0.7088818
A3           0.6948982
E3           0.6911931
C2           0.6743171
A5           0.6473320
N5           0.5880095
C1           0.5635674
E4           0.5419736
E5           0.5413065
E1           0.5255995
N4           0.5057941
N3           0.4605238
N2           0.4133517
E2           0.3435477
N1           0.3190706
```

The MLE extraction method

```r
# MLE factor extraction.
f_bfi_mle <- fa(bfi_c, nfactors = 3, fm =

# Sorted communality of the f_bfi_mle.
f_bfi_mle_common <- sort(
              f_bfi_mle$communality,
              decreasing = TRUE
              )

# create a dataframe for an improved over
data.frame(f_bfi_mle_common)
```

	f_bfi_mle_common
N1	0.7576920
E2	0.6802809
N2	0.6797943
E1	0.5219674
N3	0.5198285
N4	0.4839516
E5	0.4551183
E4	0.4343674
C1	0.4169230
N5	0.3831067
A5	0.3457303
C2	0.3253636
A2	0.3021875
A3	0.3005114
E3	0.2983111
O4	0.2573411
O3	0.2457782
C4	0.2303769
C3	0.2252758
O1	0.2013092
C5	0.1863400
O5	0.1466519
A1	0.1392760
A4	0.1071108
O2	0.0539446

Let's practice!

Choosing the right number of factors

DIMENSIONALITY REDUCTION IN R

?lexandros Tantos
Assistant Professor, Aristotle University of Thessaloniki

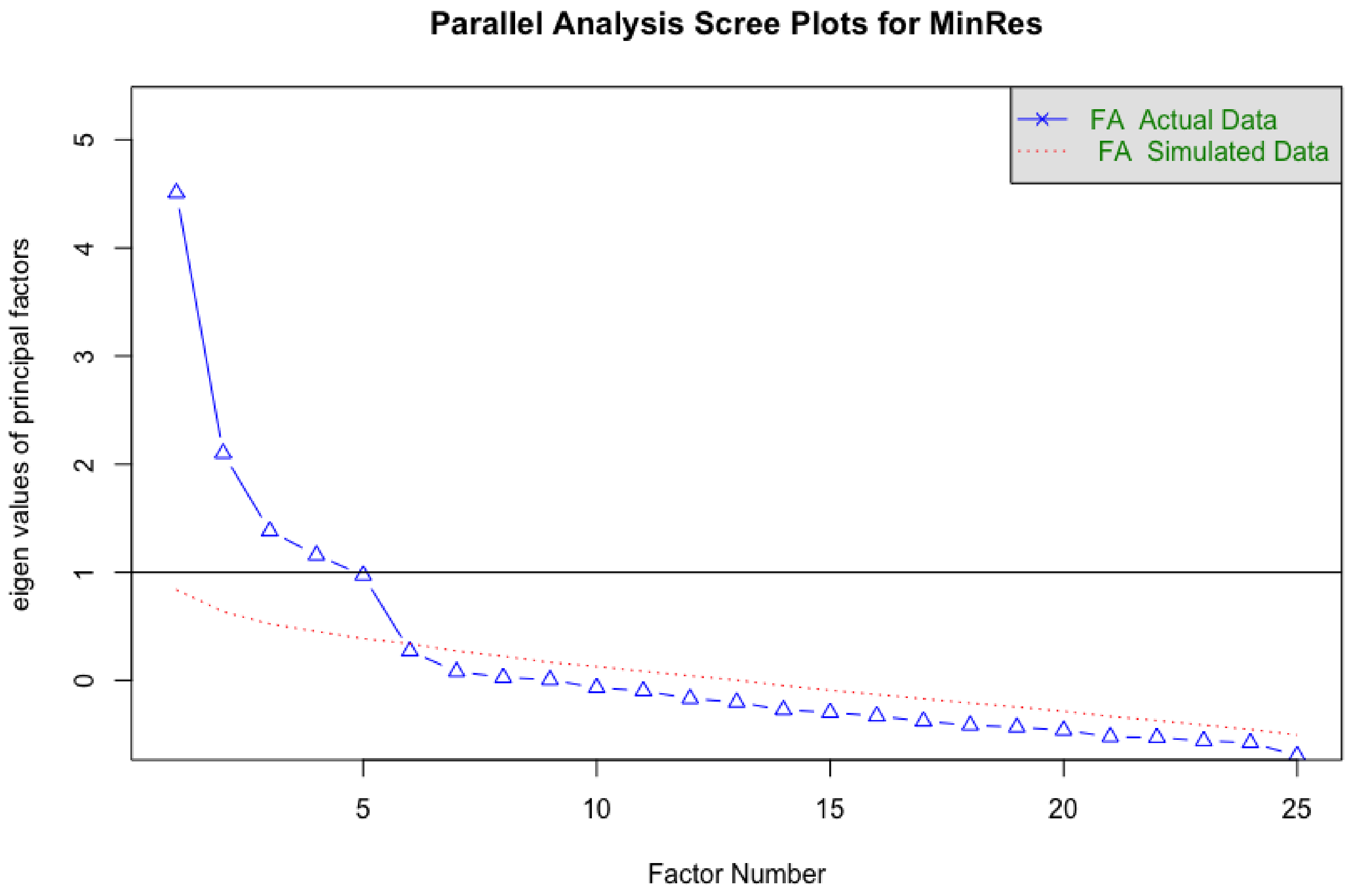
EFA: How many factors to retain?

"Solving the number of factors problem is easy, I do it everyday before breakfast. But knowing the right solution is harder" (Kaiser, 195x).

- Kaiser-Guttman criterion
- the Scree test
- Parallel analysis
- *very simple structure (`VSS`) criterion (`vss()` function in `psych`)*
- *Wayne Velicer's Minimum Average Partial (`MAP`) criterion (`vss()` function in `psych`)*

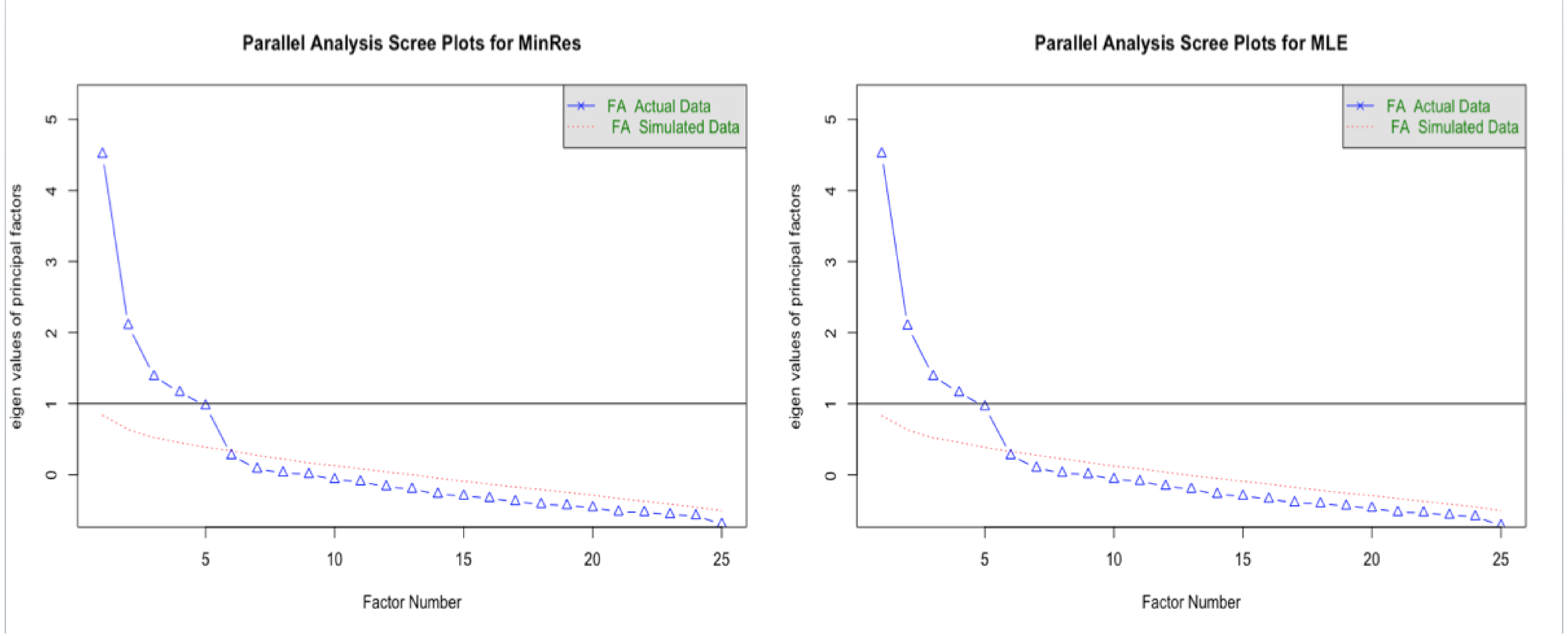
Determining the number of factors: fa.parallel()

```
# Based on the "minres" method.
fa.parallel(bfi_c, n.obs = 200,
            fa = "fa", fm = "minres")
```



Determining the number of factors: fa.parallel()

```
# Based on the "mle" method.
fa.parallel(bfi_c, n.obs = 200,
            fa = "fa", fm = "mle")
```



Let's practice!

DIMENSIONALITY REDUCTION IN R