

Hello R!

An Introduction to R



Eghe Rice Osagie
[\(eghe.osagie@han.nl\)](mailto:eghe.osagie@han.nl)

22 nov. 2018

Who R we?

Eghe Osagie

- Assistant lector (professor) at HAN University of AS
- Lecturer Bachelor HRM, Master HRM, Master CE
- Coordinator Minor HR Analytics
- **Interests:** HR Analytics, Sustainability, HRM, Research methodology

Witek ten Hove

- Instructor at HAN University of AS
- Coordinator of MSI
- **Interests:** Business Economics, Data Engineering, Data Mining, AI, Web Dev.

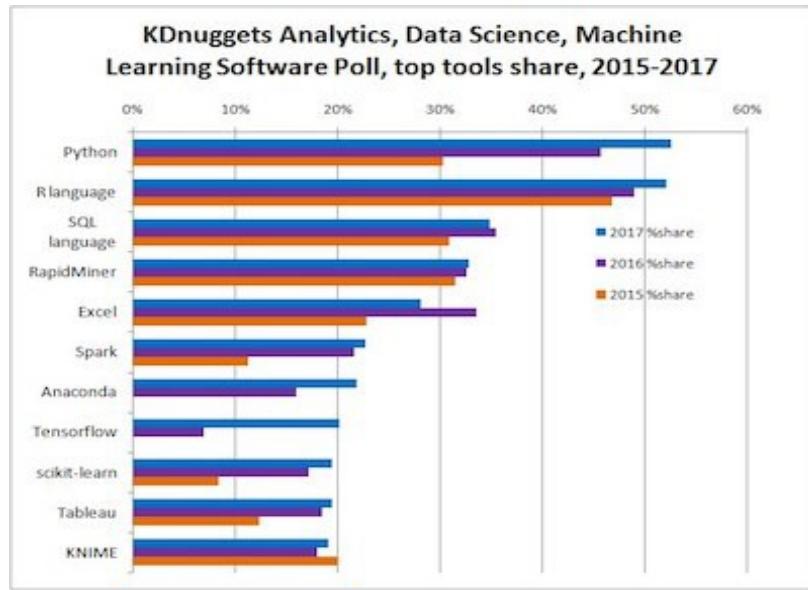
Programma

- 1. Intro R**
- 2. Practicum**
- 3. Confirmatory factor analysis**

Link naar alle docs:

Intro R

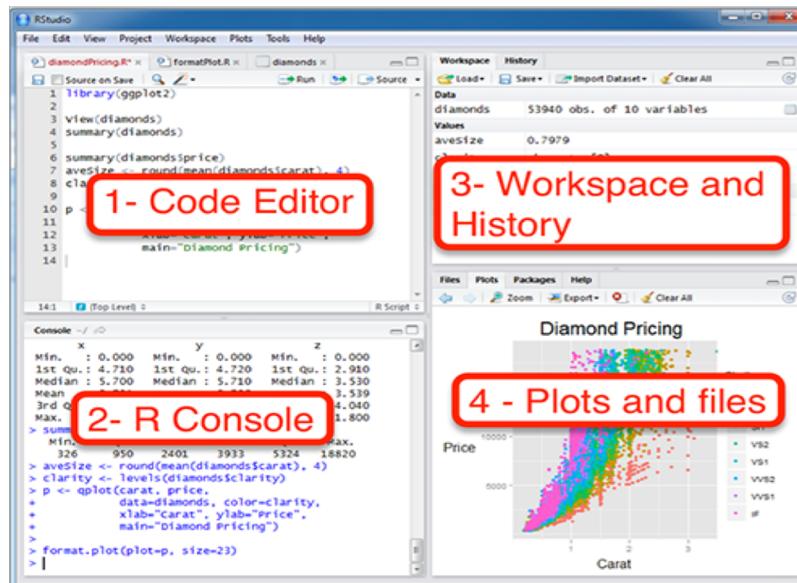
R - software



- Ranking second as tool for data science (after Python)

- Upcoming tool in Social sciences!

- Rule of Thumb: Play with the R program before you work on anything professional and know your data !!



1- Code Editor

```
library(ggplot2)
view(diamonds)
summary(diamonds$price)
avesize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)
p <- qplot(carat, price,
+   data=diamonds, color=clarity,
+   xlab="Carat", ylab="Price",
+   main="Diamond Pricing")
format.plot(plot=p, size=23)
```

2- R Console

```
summary(diamonds)
Min. : 0.000 Min. : 0.000 Min. : 0.000
1st Qu.: 4.710 1st Qu.: 4.720 1st Qu.: 2.980
Mean : 5.700 Median : 5.710 Median : 3.530
3rd Qu.: 6.780 3rd Qu.: 6.780 3rd Qu.: 4.040
Max. : 10.000 Max. : 10.000 Max. : 18.820
avesize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)
p <- qplot(carat, price,
+   data=diamonds, color=clarity,
+   xlab="Carat", ylab="Price",
+   main="Diamond Pricing")
format.plot(plot=p, size=23)
```

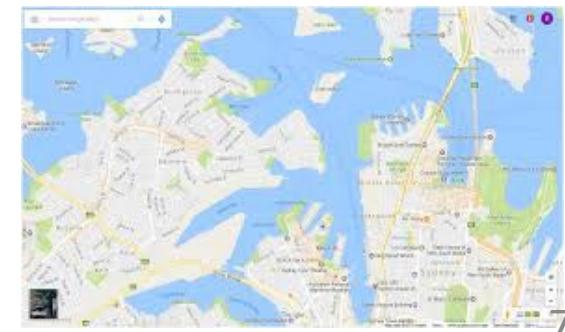
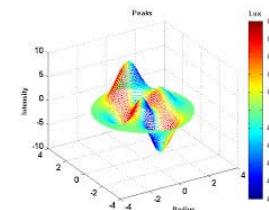
3- Workspace and History

Diamond Pricing

4 - Plots and files

Characteristics R

- **Created in:** 1995 by **Ross Ihaka & Robert Gentleman** at the University of Auckland
 - **Free**
 - Computer language
 - Windows, Mac, Linux
 - and object oriented
-
- **Extending software via ‘packages’**
 - Each package is maintained and supported by the author, but not warranted (!)
 - CRAN checks report any potential notes, warnings, and errors associated with a package
 - **Numorous Output options**



Who can read this?

Command 1:

```
install.packages("threejs")
library(threejs)
data(ego)
graphjs(ego, bg="black")
```



Command 2:

```
HS.model <- ' Visual =~ x1 + x2 + x3
              Textual =~ x4 + x5 + x6
              Speed =~ x7 + x8 + x9 '
```

1. Install R
2. Install R-studio – or
rstudio.cloud
3. Set working directory
4. Save workspace
5. Install packages and
load them
6. Read tutorial
7. Amend commands



Exemplary Packages

| Package | description |
|-------------------------|---|
| LAVAAN | Latent Variable Analysis (SEM,CFA) |
| <u>AcousticNDLCodeR</u> | Coding Sound Files for Use with NDL |
| <u>abd</u> | The Analysis of Biological Data |
| RQDA | R-Based Qualitative Data Analysis |
| <u>RSmartlyIO</u> | Loading Facebook and Instagram Advertising Data from 'Smartly.io' |
| <u>qdap</u> | Bridging the Gap Between Qualitative Data and Quantitative Analysis |
| <u>gha</u> | Qualitative Harmonic Analysis |
| <u>quanteda</u> | Quantitative Analysis of Textual Data |

See for more packages:

https://cran.r-project.org/web/packages/available_packages_by_name.html

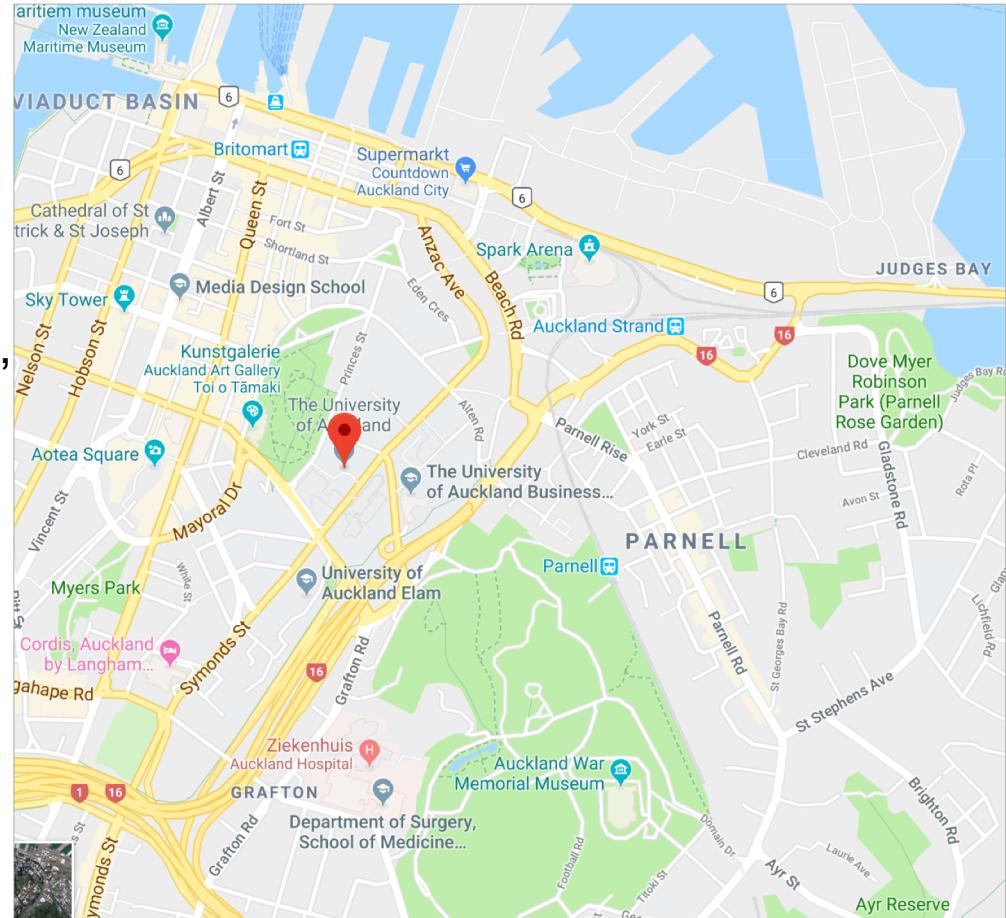
Exemplary output

Copy paste command in R

```
install.packages("leaflet");  
library(leaflet);
```

```
m <- leaflet() %>% addTiles() %>% #  
Add default OpenStreetMap map tiles  
addMarkers(lng=174.768, lat=-36.852,  
label= "The birthplace of R",  
labelOptions = labelOptions(noHide =  
T));
```

```
m # Print the map
```



Amending commands

Replace red.....

```
install.packages("leaflet");
library(leaflet);
```

```
m <- leaflet() %>% addTiles() %>% #
Add default OpenStreetMap map tiles
addMarkers(lng=174.768, lat=-36.852,
label= "The birthplace of R",
labelOptions = labelOptions(noHide =
T));
```

```
m # Print the map
```

....with green.

```
install.packages("leaflet");
library(leaflet);
```

```
m <- leaflet() %>%
addTiles() %>% # Add default
OpenStreetMap map tiles
addMarkers(lng= 5.949481,
lat=51.989683, label= "An introduction
to R", labelOptions =
labelOptions(noHide = T));
```

```
m # Print the map
```

LET OP: Google maps toont eerst "Lng" en dan "Lat", dus net andersom invoeren.

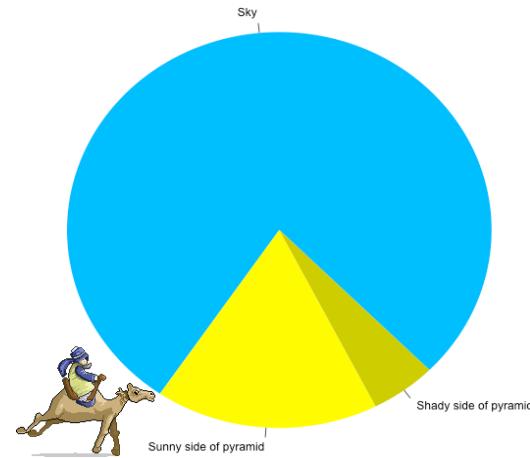
Exemplary output

Copy paste command in R

```
pie(c(a=78, b=17, c=5), init.angle =  
315, col = c("deepskyblue", "yellow",  
"yellow3"), border = FALSE, radius =  
1.0)
```

Copy paste command in R

```
install.packages("threejs");  
library(threejs);  
data(ego);  
graphjs(ego, bg="red")
```



More examples:

<https://github.com/witusj/hellor/blob/master/hellor.Rmd>

Practicum R

Practicum

First some initial information

Go to: witusj.github.io/WorkshopSI/

Perform the following exercises:

- Voorbereiding
- Basis R

Remaining exercises can be performed at home

Hello R!

This presentation can be found online:

witusj.github.io/hellor/hellor.html

press F for fullscreen

For the Workshop R (Dutch) go to:

<https://witusj.github.io/WorkshopSI/>

Workshop documents can be found here (docs folder): <https://github.com/witusj/hellor>

Who can read this?

Command 1:

- `install.packages("threejs")`
- `library(threejs)`
- `data(ego)`
- `graphjs(ego, bg="black")`



Command 2:

```
HS.model <- ' Visual =~ x1 + x2 + x3  
          Textual =~ x4 + x5 + x6  
          Speed =~ x7 + x8 + x9 '
```

Confirmatory Factor Analysis (Dutch)

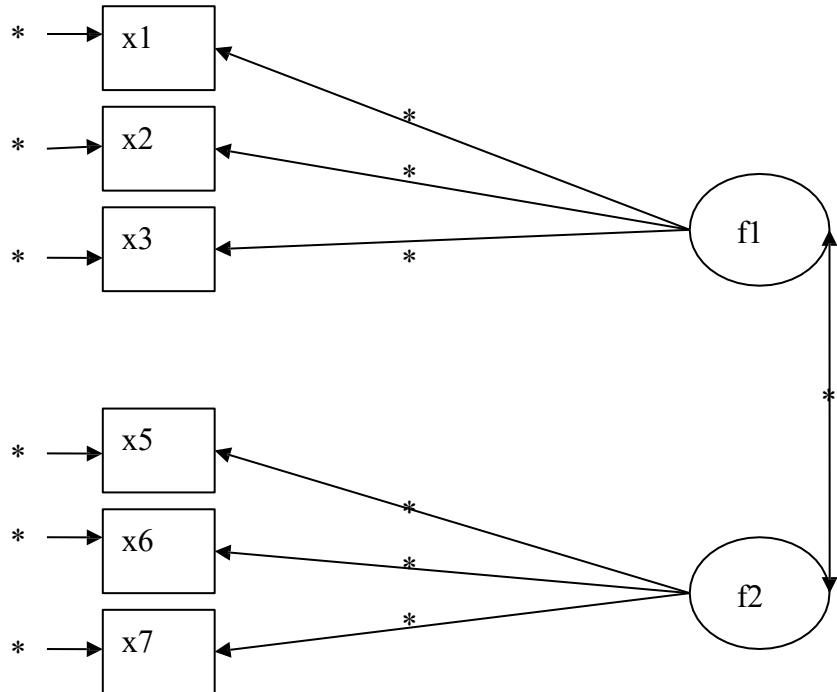
CFA

Doel confirmatory factor analysis:
bevestiging krijgen voor van te voren bepaald
model/structuur

CFA model:

- **Kenmerken:**
 - **NIET** elke manifeste variabele een lading op elke factor
 - **WEL** relatie tussen de componenten
 - **WEL** meetfouten

CFA model:



De asterixen verwijzen naar de te schatten parameters

Parameters = die delen van het model die nog onbekend zijn voor de onderzoeker, en dus berekend moeten worden

Hier:

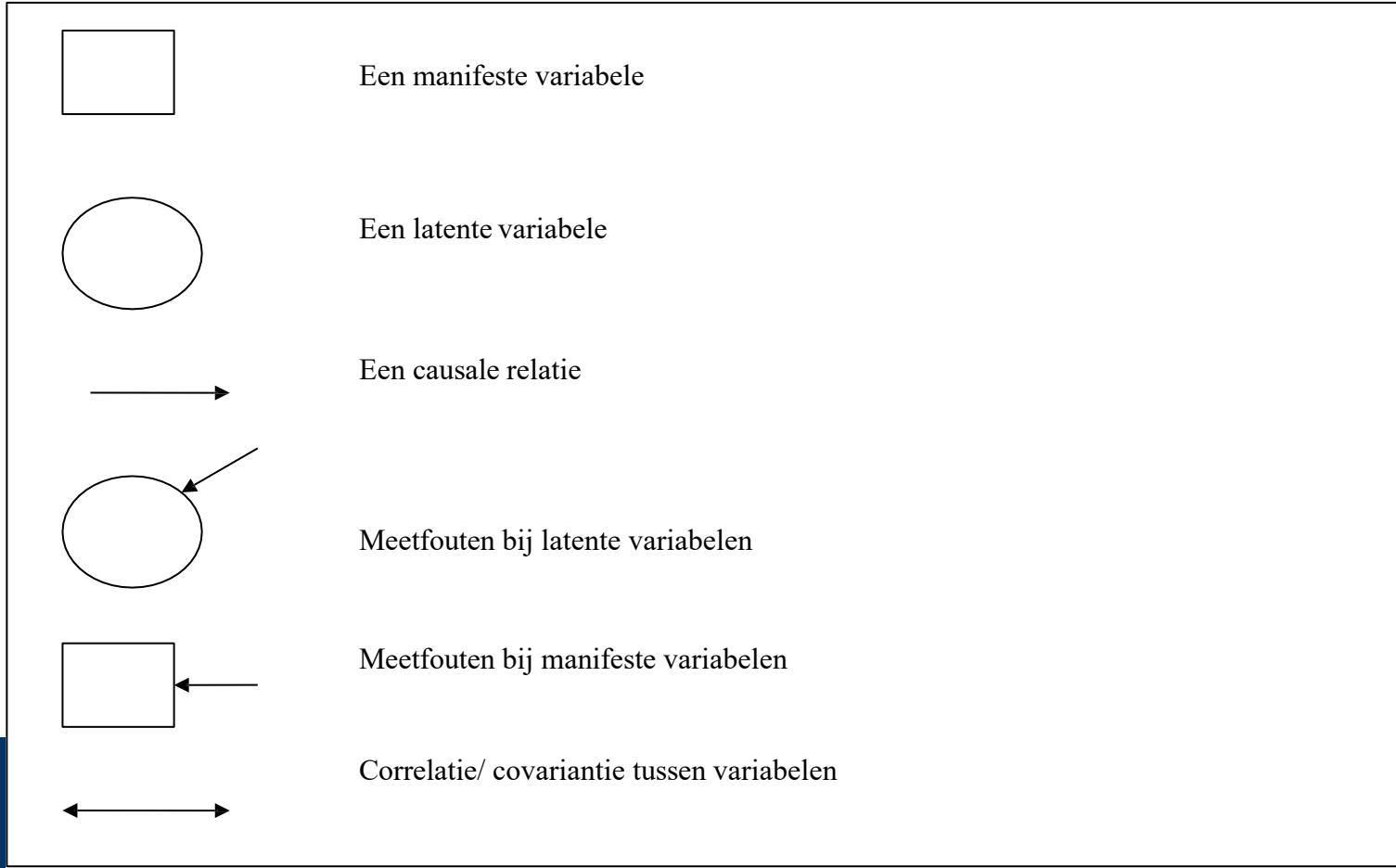
- meetfouten,
- factorladingen,
- correlaties tussen factoren,
- variantie van factoren,

...

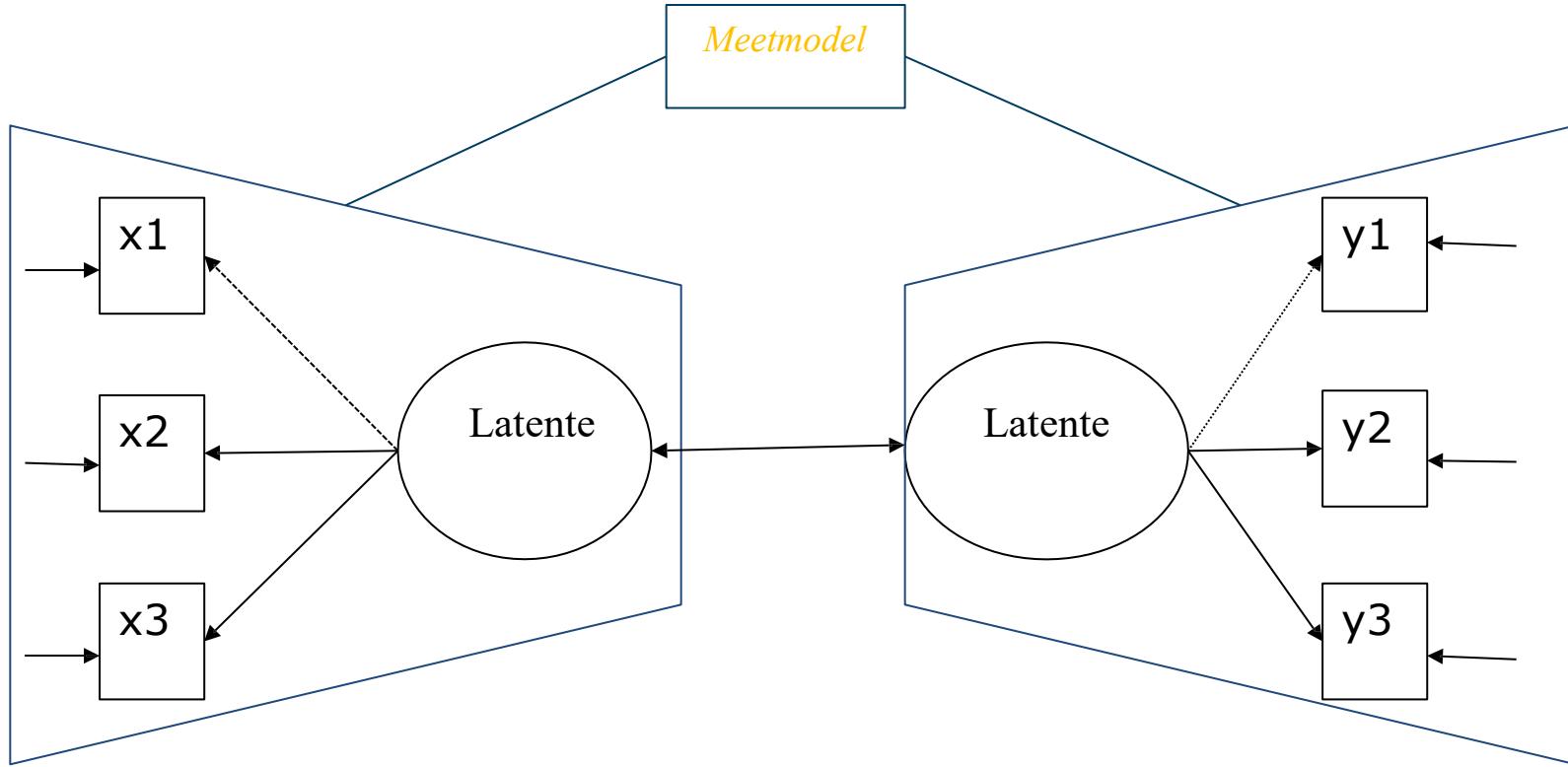
Belangrijke bergippen in CFA

- **Rondje** = niet direct gemeten (latente var. [f])
- **Vierkant** = direct gemeten (manifeste var./indicator/item [x])
- ***ind.*** = indicator[x]
- → = impact van 1 variabele/factor op een andere variabele/factor
- ←→ = covariantie of correlatie tussen variabelen/factoren.
- **Meetmodel** = relatie tussen latente variabelen en indicatoren
- **Structuurmodel** = relaties tussen latente variabelen
- **EXO** = Exogene construct/factor (pijltje exit)
- e = meetfout

Notatie voor tekenen van modellen



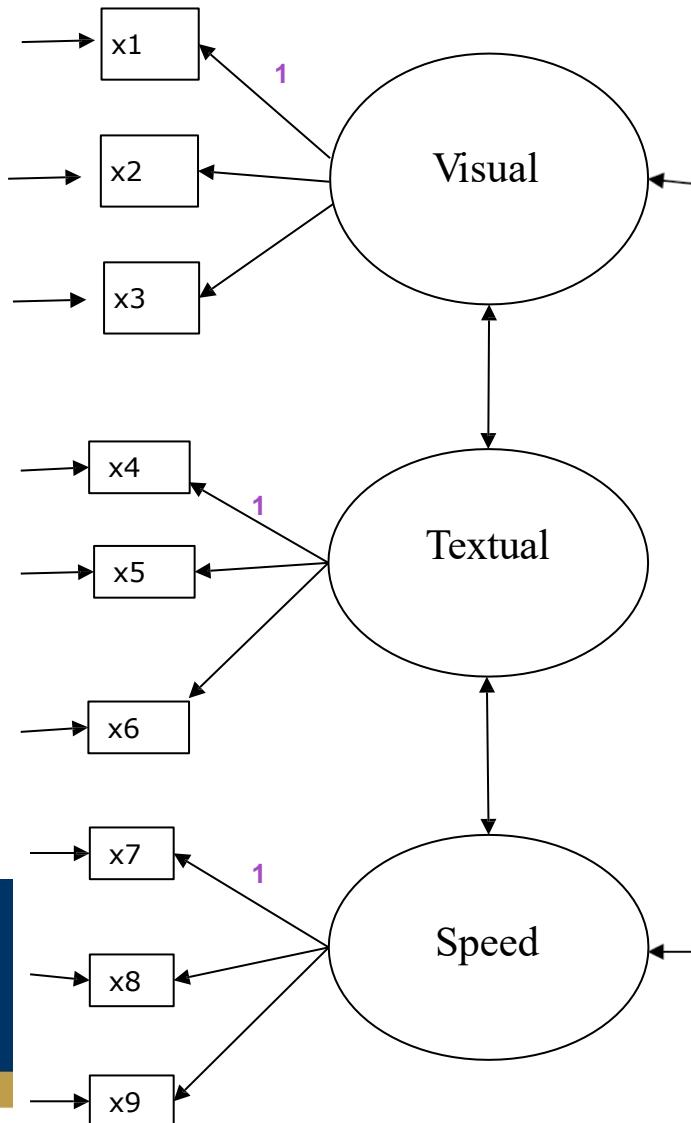
Voorbeeld model CFA



Belangerijke Commands LAVAAN

| Formule type | Operator | Betekenis |
|-----------------------------------|-----------------|--|
| • Definitie van latente variabele | \equiv | Is measured by/ Is gemeten door |
| • regressie | \sim | Is regressed on |
| • (residu) (co)variantie | $\sim\sim$ | Is correlated with/ gecorrelateerd met |
| • intercept | ~ 1 | intercept |
| | f | Latente variabele |
| | y | Afhankelijke var |
| | x | Onafhank. Var/observed variable/indicator |
| | cfa() | Voer een CFA analyse uit. Met help("cfa"), krijg je uitleg over de functie |
| | sem() | Voer een SEM analyse uit. Met help("sem"), krijg je uitleg over de functie |
| | Growth() | Voer een Growth curve analyse uit. Met help("growth"), krijg je uitleg over de functie |

Voorbeeld met LAVAAN in R



1. Bepaal model(len)

Visual $\sim x1 + x2 + x3$

Textual $\sim x4 + x5 + x6$

Speed $\sim x7 + x8 + x9$

Wat staat hier: Latent variable \sim indicator1 + indicator2 + indicator3

2. Specificeer model(len) in R

`HS.model <- ' Visual $\sim x1 + x2 + x3$`

`Textual $\sim x4 + x5 + x6$`

`Speed $\sim x7 + x8 + x9'$`

3. Fit model(len) in R

`....<- cfa (...., data =)`

Bijv.: `fitM1 <- cfa (HS.model, data = HolzingerSwineford1939)`

4. Lees Fit indices af/vergelijk ze

`summary (..., fit.measures = TRUE)`

Bijv.: `summary (fit, fit.measures = TRUE)`

In het geval van niet normal verdeelde data en n=200+ (estimator MLM test = SB):

`fitM1 <- cfa(HS.model, data =
HolzingerSwineford1939, estimator= "MLM",
= "satorra.bentler")`

Mocht je specifieke fitmaken willen opvragen:

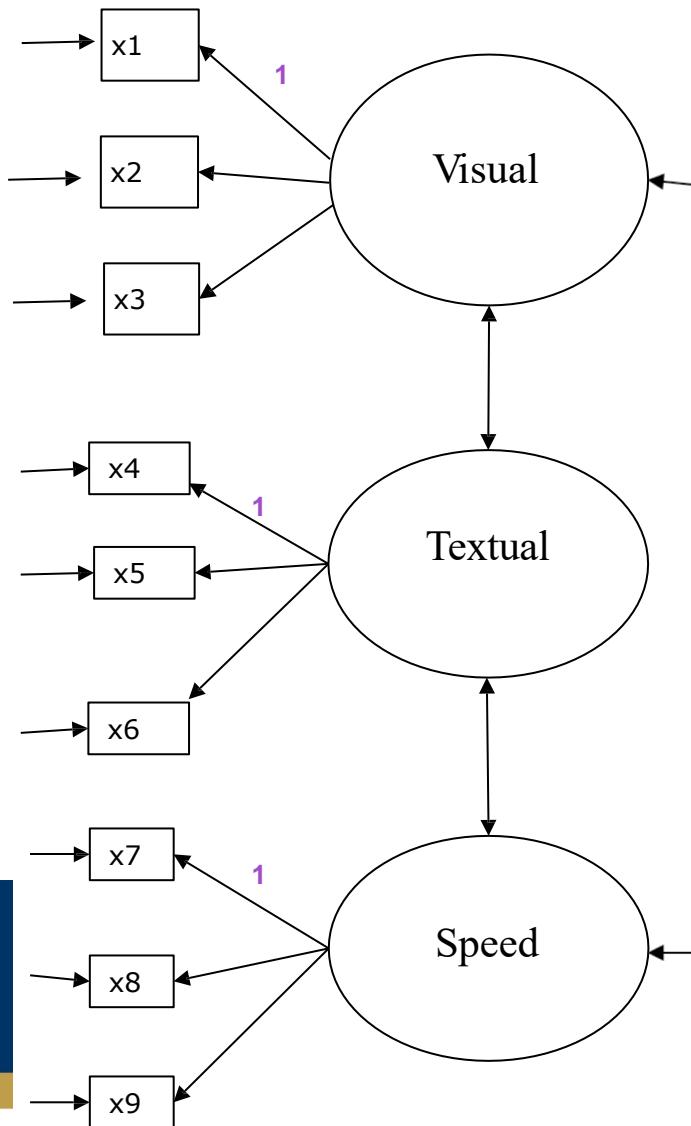
`fitMeasures(fitM1, c("cfi.scaled", "rmsea", "gfi",
"agfi", "nnfi", "chisq", "chisq.scaled",
"df.scaled"))`

Voor de modificatie indeces:

`modindices(fitM1)`

Rood = aanpassen aan eigen variabelen/items/namen

Voorbeeld met LAVAAN in R



1. Bepaal model(len)

Visual $\sim x_1 + x_2 + x_3$

Textual $\sim x_4 + x_5 + x_6$

Speed $\sim x_7 + x_8 + x_9$

Wat staat hier: Latent variable \sim indicator1 + indicator2 + indicator3

2. Specificeer model(len) in R

HS.model <- 'Visual $\sim x_1 + x_2 + x_3$

Textual $\sim x_4 + x_5 + x_6$

Speed $\sim x_7 + x_8 + x_9'$

3. Fit model(len) in R

fit1<- cfa (..., data =)

Bijv.: fitM1 <- cfa (HS.model, data = HolzingerSwineford1939)

4. Lees Fit indices af/vergelijk ze

summary (..., fit.measures = TRUE)

Bijv.: summary (fit1, fit.measures = TRUE)

5. Bepaal beste model nadat je meerdere modellen hebt gefit

anova(fit1, fit3)

Kijk naar AIC waarde....lagere AIC of chi kwardaat is beter model

Rood = aanpassen aan eigen variabelen/items/namen

Fit indices

| Fit indices | Thresholds (cut-offs) |
|---|--|
| • Relative Chi square (Chi-square-df; cmin/df) | < 2 ^a of <3= good ^b (soms is <5 ook toegelaten ^c) |
| • p value of the model | >.05 |
| • RMSEA | <.05=good; .05-.10=moderate; >.10=bad ^b |
| • CFI | >.95=great; >.90 traditional; >.80 sommige gevallen toelaatbaar ^bstreven >.93 ^d |
| • GFI | >.90 ^d ...liefst >.95 ^b |
| • (N)NFI | >.90 ^d ...of >.95 ^c |
| • AGFI | >.80 ^b |

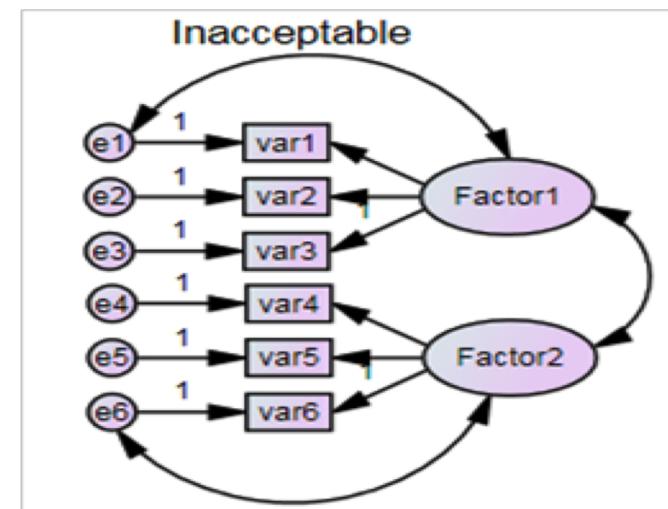
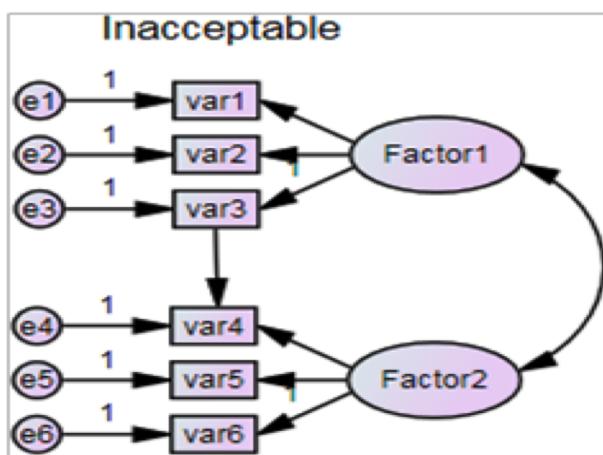
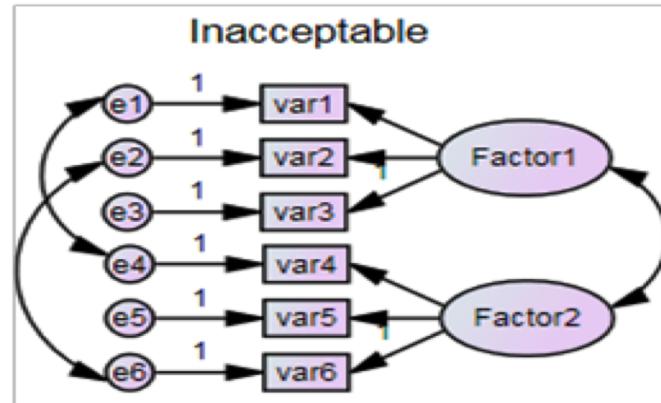
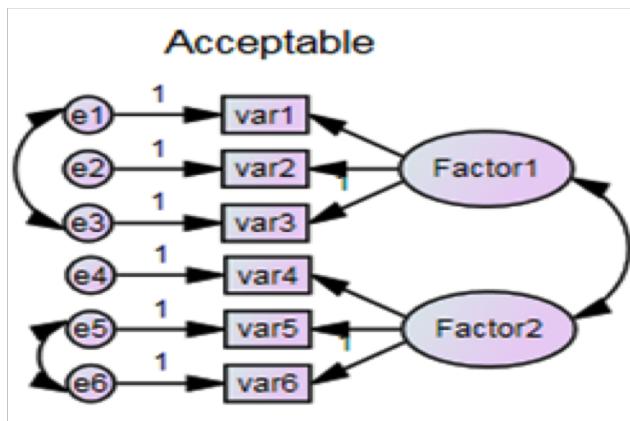
a = Ullman(2001). b = Hu & Bentler (1999). c = Schumacker & Lomax (2004). d = Byrne (1994)

Modification indices (MI) & Standardized residuals covar (SRC)

Aanpassen model : doe je bij geen goede fit. Theoretische onderbouwing belangrijk!!

- **Theorie**
- **MI**
 - Error van verschillende constructen mogen niet correleren
 - Error mag niet correleren met latente of observerd constructen
- **SRC**
 - Error van verschillende constructen mogen niet correleren
 - Error mag niet correleren met latente of observerd constructen

MI rules



CFA samengevat

- *CFA om model-fit te schatten:* past model bij de data?
=> fit indices: Chi², GFI, AGFI, NNFI, CFI, RMSEA
- *CFA om modellen onderling te vergelijken:* kijk naar AIC waarde, lagere waarde dan past model beter bij data
- *En hoe het model interpreteren?* => interpreteren van parameterschattingen

Practicum

- CFA → open Tutorial LAVAAN, perform excises on p. 4-8

Remaining exercises can be performed at home

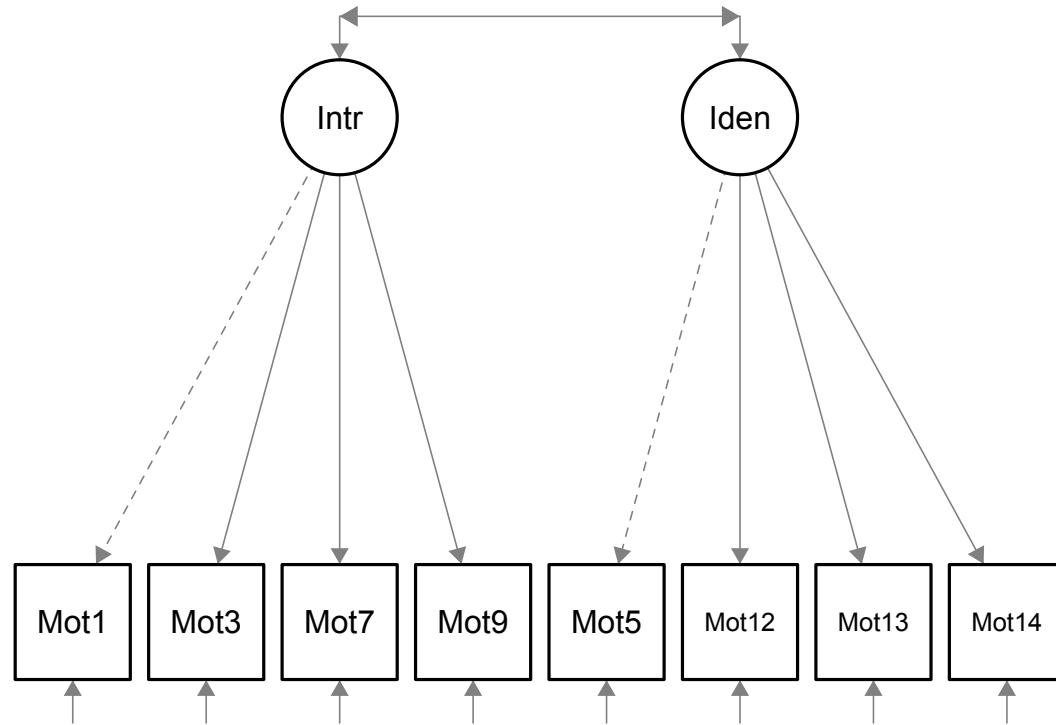
Questions?

Confirmatory Factor Analysis (Dutch)

CFA

Doel confirmatory factor analysis:
bevestiging krijgen voor van te voren bepaald
model/structuur

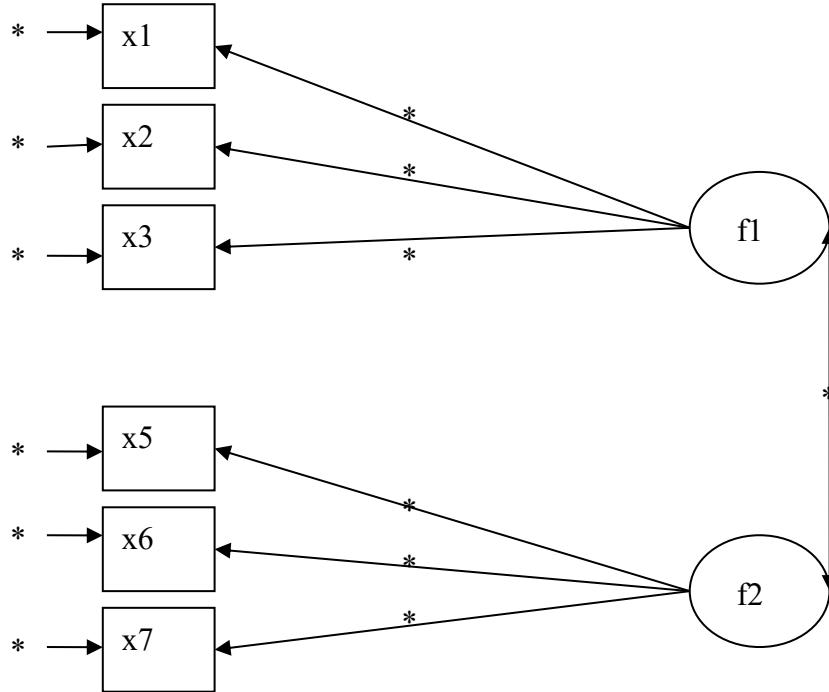
CFA model:



Kenmerken:

- NIET elke manifeste variabele een lading op elke factor
- WEL relatie tussen de componenten
- WEL meetfouten

CFA model:



De asterixen verwijzen naar de te schatten parameters

Parameters = die delen van het model die nog onbekend zijn voor de onderzoeker, en dus berekend moeten worden

Hier:

- meetfouten,
- factorladingen,
- correlaties tussen factoren,
- variantie van factoren,

...

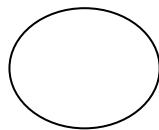
Belangrijke bergippen in CFA

- **Rondje** = niet direct gemeten (latente var. [f])
- **Vierkant** = direct gemeten (manifeste var./indicator/item [x])
- ***ind.*** = indicator[x]
- **→** = impact van 1 variabele/factor op een andere variabele/factor
- **↔** = covariantie of correlatie tussen variabelen/factoren.
- **Meetmodel** = relatie tussen latente variabelen en indicatoren
- **Structuurmodel** = relaties tussen latente variabelen
- **EXO** = Exogene construct/factor (pijltje exit)
- **e** = meetfout

Notatie voor tekenen van modellen



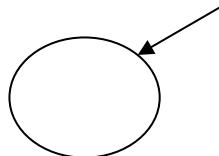
Een manifeste variabele



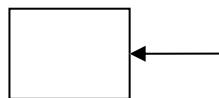
Een latente variabele



Een causale relatie



Meetfouten bij latente variabelen

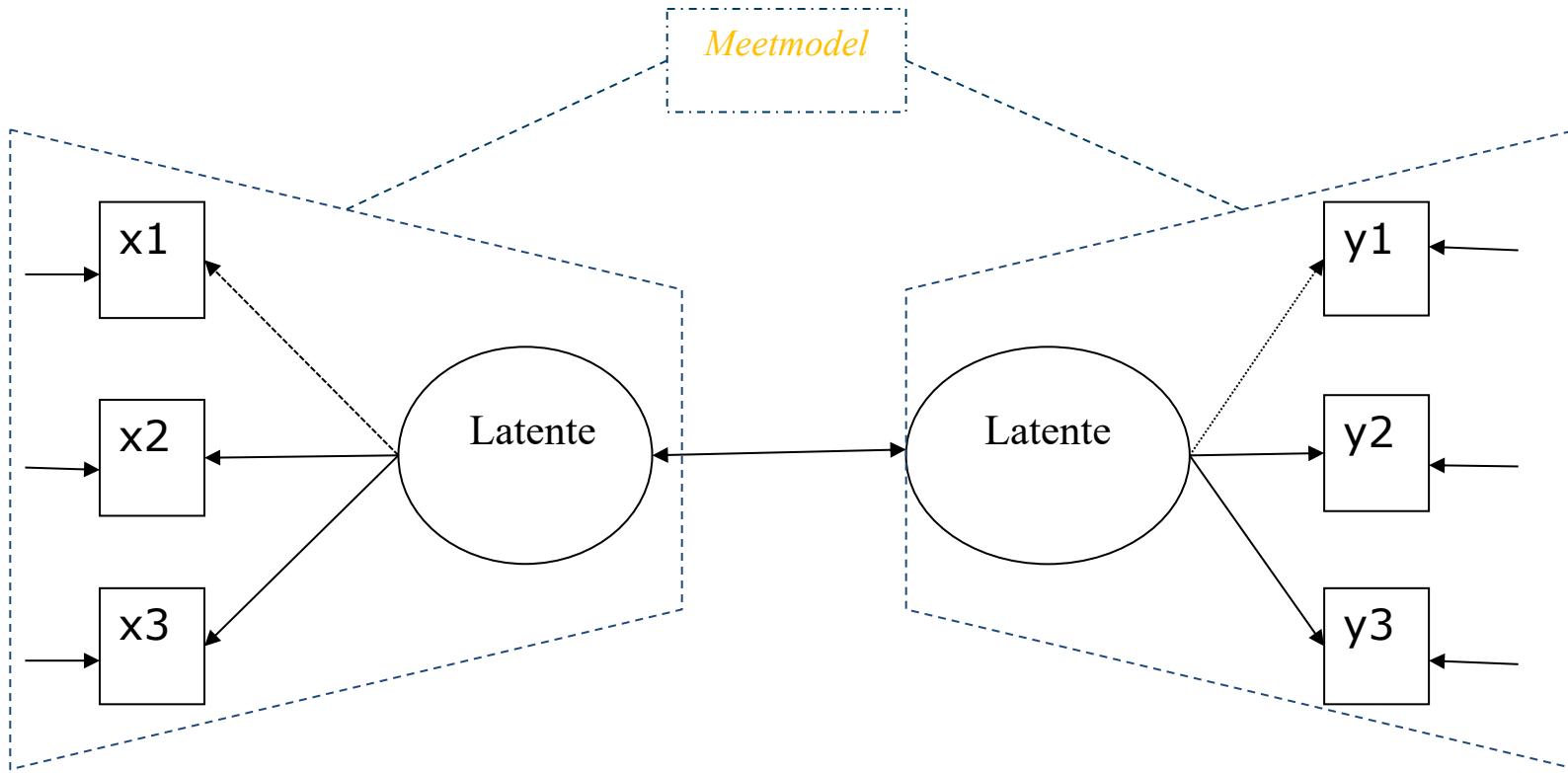


Meetfouten bij manifeste variabelen



Correlatie/ covariantie tussen variabelen

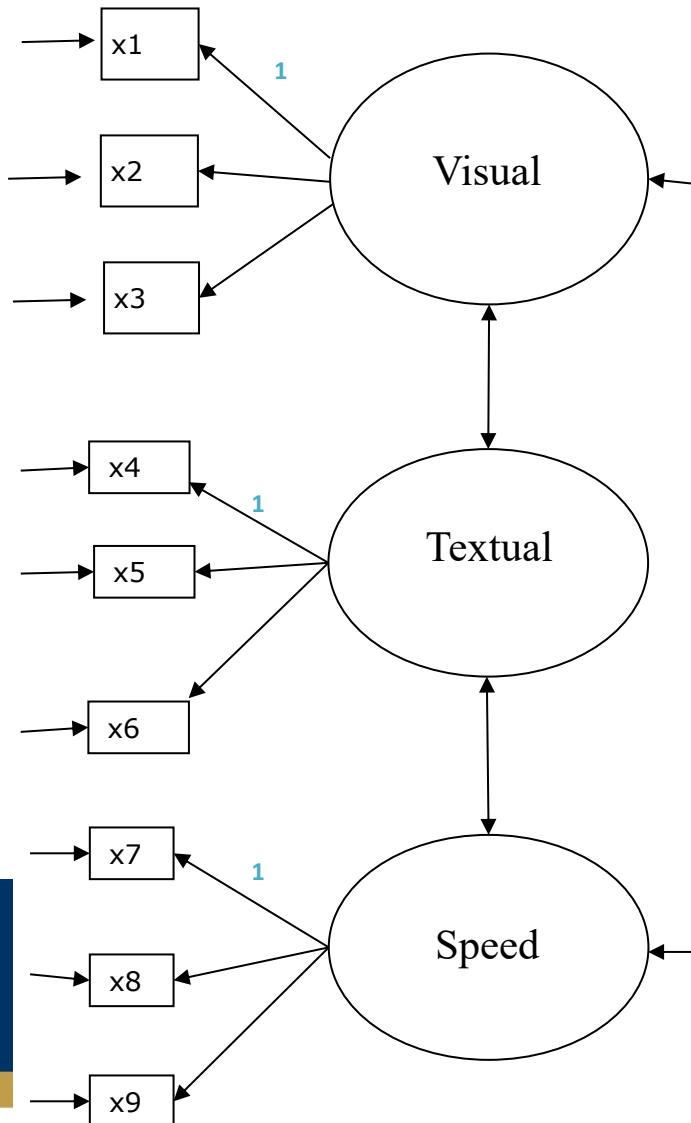
Voorbeeld model CFA



Belangerijke Commands LAVAAN

| Formule type | Operator | Betekenis |
|-----------------------------------|----------|--|
| • Definitie van latente variabele | =~ | Is measured by/ Is gemeten door |
| • regressie | ~ | Is regressed on |
| • (residu) (co)variantie | ~~ | Is correlated with/ gecorroeerd met |
| • intercept | ~1 | intercept |
| | f | Latente variabele |
| | y | Afhankelijke var |
| | x | Onafhank. Var/observed variable/indicator |
| | cfa() | Voer een CFA analyse uit. Met help("cfa"), krijg je uitleg over de functie |
| | sem() | Voer een SEM analyse uit. Met help("sem"), krijg je uitleg over de functie |
| | Growth() | Voer een Growth curve analyse uit. Met help("growth"), krijg je uitleg over de functie |

Voorbeeld met LAVAAN in R



1. Bepaal model(len)

Visual =~ x1 + x2 + x3

Textual =~ x4 + x5 + x6

Speed =~ x7 + x8 + x9

Wat staat hier: Latent variable =~ indicator1 + indicator2 + indicator3

2. Specificeer model(len) in R

```
HS.model <- ' Visual =~ x1 + x2 + x3
              Textual =~ x4 + x5 + x6
              Speed =~ x7 + x8 + x9 '
```

3. Fit model(len) in R

```
....<- cfa (..., data = .....)
```

Bijv.: `fitM1 <- cfa (HS.model, data = HolzingerSwineford1939)`

4. Lees Fit indices af/vergelijk ze

```
summary (... , fit.measures = TRUE )
```

Bijv.: `summary (fit, fit.measures = TRUE)`

In het geval van niet normal verdeelde data en n=200+ (estimator MLM test = SB):

```
fitM1 <- cfa(HS.model, data =
  HolzingerSwineford1939, estimator= "MLM",
  = "satorra.bentler")
```

Mocht je specifieke fitmaken willen opvragen:

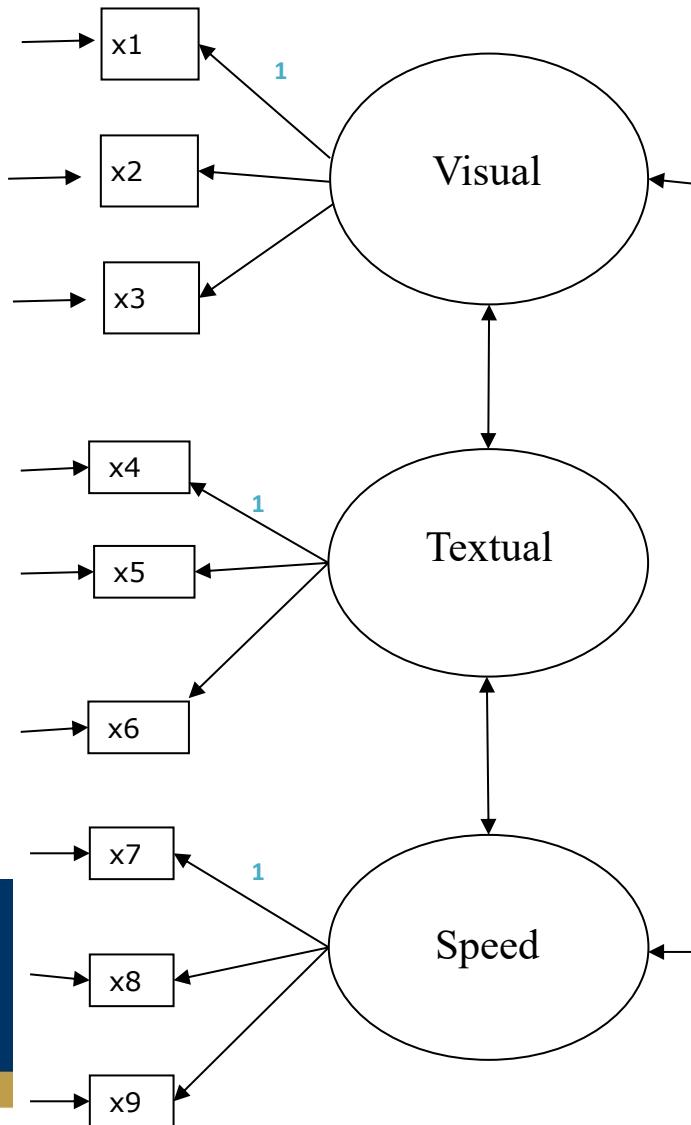
```
fitMeasures(fitM1, c("cfi.scaled", "rmsea", "gfi",
  "agfi", "nnfi", "chisq", "chisq.scaled",
  "df.scaled"))
```

Voor de modificatie indexen:

```
modindices(fitM1)
```

Rood = aanpassen aan eigen variabelen/items/namen

Voorbeeld met LAVAAN in R



1. Bepaal model(len)

Visual =~ x1 + x2 + x3

Textual =~ x4 + x5 + x6

Speed =~ x7 + x8 + x9

Wat staat hier: Latent variable =~ indicator1 + indicator2 + indicator3

2. Specificeer model(len) in R

HS.model <- ' Visual =~ x1 + x2 + x3

Textual =~ x4 + x5 + x6

Speed =~ x7 + x8 + x9 '

3. Fit model(len) in R

fit1<- cfa (...., data =)

Bijv.: fitM1 <- cfa (HS.model, data = HolzingerSwineford1939)

4. Lees Fit indices af/vergelijk ze

summary (..., fit.measures = TRUE)

Bijv.: summary (fit1, fit.measures = TRUE)

5. Bepaal beste model nadat je meerdere modellen hebt gefit

anova(fit1,fit3)

Kijk naar AIC waarde....lagere AIC of chi kwardaat is beter model

Rood = aanpassen aan eigen variabelen/items/namen

Fit indices

| Fit indices | Tresholds (cut-offs) |
|---|--|
| • Relative Chi square (Chi-square-df; cmin/df) | < 2 ^a of <3= good ^b (soms is <5 ook toegelaten ^c) |
| • p value of the model | >.05 |
| • RMSEA | <.05=good; .05-.10=moderate; >.10=bad ^b |
| • CFI | >.95=great; >.90 traditional; > .80 sommige gevallen toelaatbaar ^bstreven >.93 ^d |
| • GFI | >.90 ^d ...liefst >.95 ^b |
| • (N)NFI | >.90 ^d ...of >.95 ^c |
| • AGFI | >.80 ^b |

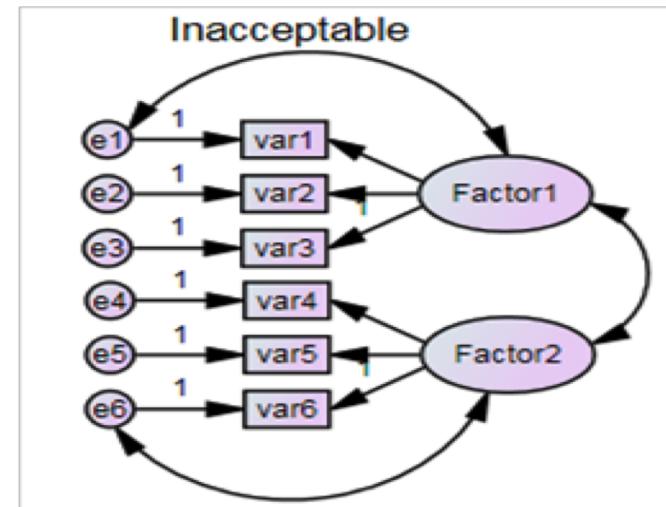
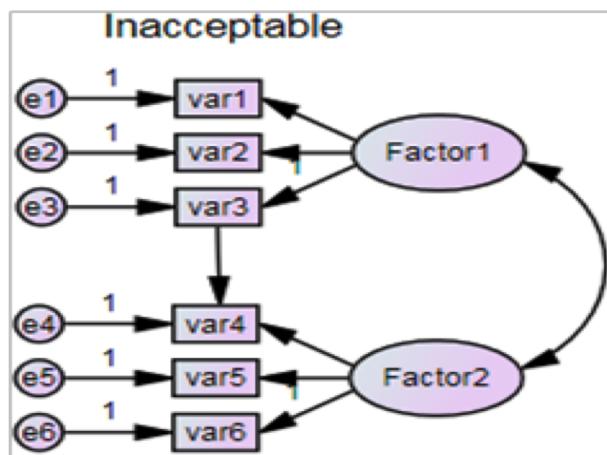
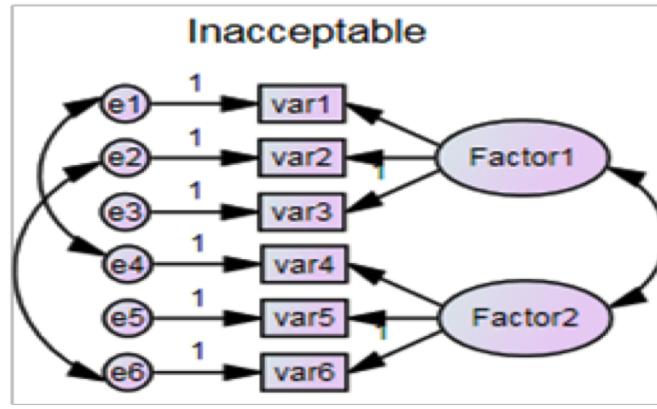
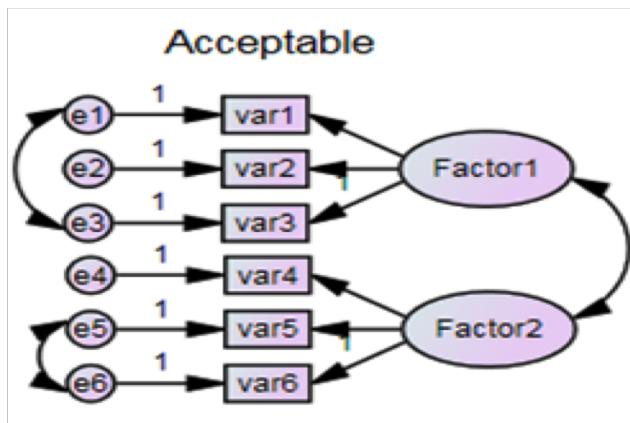
a = Ullman(2001). b = Hu & Bentler (1999). c = Schumacker & Lomax (2004). d = Byrne (1994)

Modification indices (MI) & Standardized residuals covar (SRC)

Aanpassen model : doe je bij geen goede fit. Theoretische onderbouwing belangrijk!!

- *Theorie*
- *MI*
 - Error van verschillende constructen mogen niet correleren
 - Error mag niet correleren met latente of observerd constructen
- *SRC*
 - Error van verschillende constructen mogen niet correleren
 - Error mag niet correleren met latente of observerd constructen

MI rules



CFA om model-fit te schatten: past model bij de data? => fit indices: Chi², GFI, AGFI, NNFI, CFI, RMSEA

CFA om modellen onderling te vergelijken: kijk naar AIC waarde, lagere waarde dan past model beter bij data

En hoe het model interpreteren? => interpreteren van parameterschattingen

Practicum

- CFA → open Tutorial LAVAAN, perform excises on p. 4-8

Remaining exercises can be performed at home

Questions?