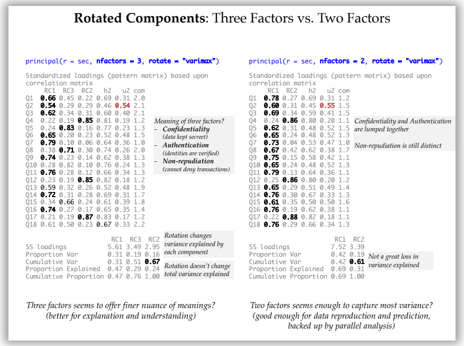
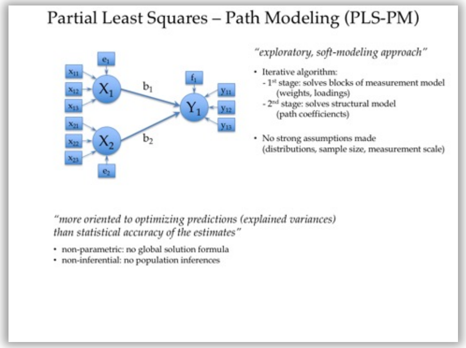
Business Analytics Using
Computational Statistics

PCA and Factors → **Structural Equation Modeling** → Predictions

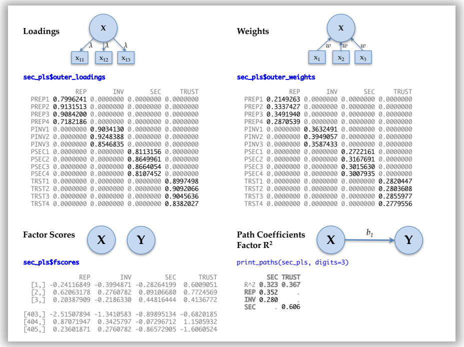
PCA Revisited



Structural Equation Modeling



SEMinR



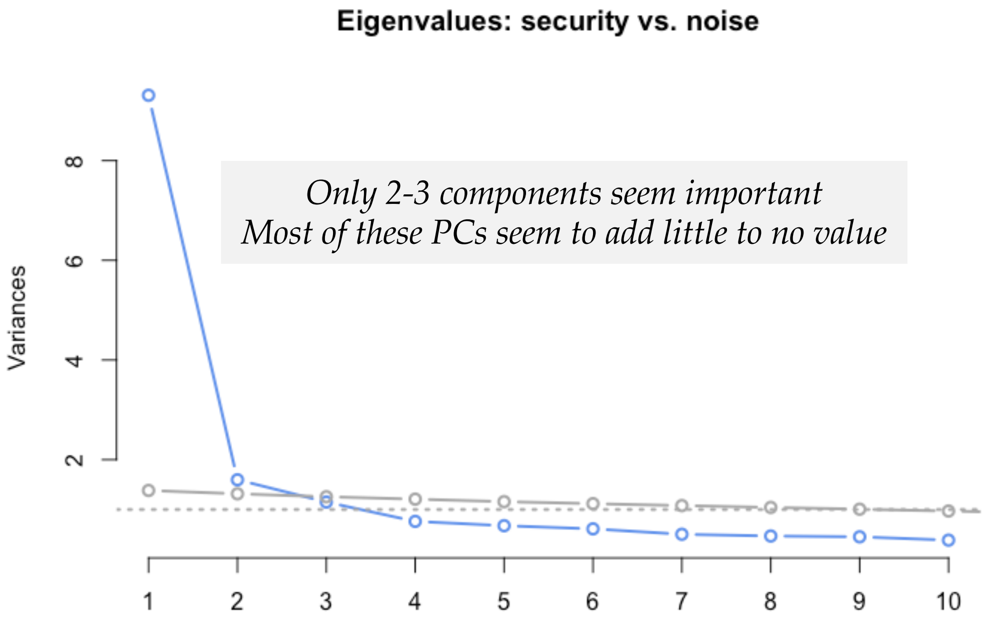
Parallel Analysis: Security Example

```
sec <- read.csv("security_questions.csv")
sec_eigen <- eigen(cor(sec))
sec_eigen$values
```

```
 [1] 9.31 1.60 1.15 0.76 0.68 0.61 0.50 0.47 0.45 0.39 0.35 0.30 0.29 0.26 0.23 0.23 0.21 0.20
```

Parallel Analysis

```
sim_noise <- function(n, p) {
  noise <- data.frame(replicate(p, rnorm(n)))
  return( eigen(cor(noise))$values )
}

set.seed(42)
evalues_noise <- replicate(1000,
                   sim_noise(nrow(sec), ncol(sec)))
evalues_mean <- apply(evalues_noise, 1, mean)

screeplot(sec_pca, type="lines", ...)
lines(evalues_mean, type="b", ...)
abline(h=1, lty="dotted")
```



Only 2-3 components seem important
Most of these PCs seem to add little to no value

☹️

It is difficult to estimate the meaning of components from a simple PCA

Unrotated Components: Three Factors vs. Two Factors

Three factor solution

```
principal(r = sec, nfactors = 3, rotate = "none")
```

```
     PC1   PC2   PC3   h2   u2  com
Q1   0.82 -0.14  0.00 0.69 0.31 1.1
Q2   0.67 -0.01  0.09 0.46 0.54 1.0
Q3   0.77 -0.03  0.09 0.60 0.40 1.0
Q4   0.62  0.64  0.11 0.81 0.19 2.1
Q5   0.69 -0.03 -0.54 0.77 0.23 1.9
Q6   0.68 -0.10  0.21 0.52 0.48 1.2
Q7   0.66 -0.32  0.32 0.64 0.36 2.0
Q8   0.79  0.04 -0.34 0.74 0.26 1.4
Q9   0.72 -0.23  0.20 0.62 0.38 1.4
Q10  0.69 -0.10 -0.53 0.76 0.24 1.9
Q11  0.75 -0.26  0.17 0.66 0.34 1.4
Q12  0.63  0.64  0.12 0.82 0.18 2.1
Q13  0.71 -0.06  0.08 0.52 0.48 1.0
Q14  0.81 -0.10  0.16 0.69 0.31 1.1
Q15  0.70  0.01 -0.33 0.61 0.39 1.4
Q16  0.76 -0.20  0.18 0.65 0.35 1.3
Q17  0.62  0.66  0.11 0.83 0.17 2.0
Q18  0.81 -0.11 -0.07 0.67 0.33 1.1

                       PC1  PC2  PC3
SS loadings           9.31 1.60 1.15
Proportion Var        0.52 0.09 0.06
Cumulative Var        0.52 0.61 0.67
Proportion Explained  0.77 0.13 0.10
Cumulative Proportion 0.77 0.90 1.00
```

Meaning of three factors?

🤔

***Three factors** seems to offer finer nuance of meanings?*
*(better for **explanation and understanding**)*

Two factor solution

```
principal(r = sec, nfactors = 2, rotate = "none")
```

```
     PC1   PC2   h2   u2  com
Q1   0.82 -0.14 0.69 0.31 1.1
Q2   0.67 -0.01 0.45 0.55 1.0
Q3   0.77 -0.03 0.59 0.41 1.0
Q4   0.62  0.64 0.80 0.20 2.0
Q5   0.69 -0.03 0.48 0.52 1.0
Q6   0.68 -0.10 0.48 0.52 1.0
Q7   0.66 -0.32 0.53 0.47 1.4
Q8   0.79  0.04 0.62 0.38 1.0
Q9   0.72 -0.23 0.58 0.42 1.2
Q10  0.69 -0.10 0.48 0.52 1.0
Q11  0.75 -0.26 0.64 0.36 1.2
Q12  0.63  0.64 0.80 0.20 2.0
Q13  0.71 -0.06 0.51 0.49 1.0
Q14  0.81 -0.10 0.67 0.33 1.0
Q15  0.70  0.01 0.50 0.50 1.0
Q16  0.76 -0.20 0.62 0.38 1.1
Q17  0.62  0.66 0.82 0.18 2.0
Q18  0.81 -0.11 0.66 0.34 1.0

                       PC1  PC2
SS loadings           9.31 1.60
Proportion Var        0.52 0.09
Cumulative Var        0.52 0.61
Proportion Explained  0.85 0.15
Cumulative Proportion 0.85 1.00
```

More items have low communality and high uniqueness…

…but not a great loss in variance explained

🤔

***Two factors** seems enough to capture most variance?*
*(good enough for **data reproduction and prediction**,*
backed up by parallel analysis)

Rotated Components: Three Factors vs. Two Factors

Three factor solution

```
principal(r = sec, nfactors = 3, rotate = "varimax")
```

	RC1	RC3	RC2	h2	u2	com
Q1	**0.66**	0.45	0.22	0.69	0.31	2.0
Q2	0.54	0.29	0.29	0.46	0.54	2.1
Q3	**0.62**	0.34	0.31	0.60	0.40	2.1
Q4	0.22	0.19	**0.85**	0.81	0.19	1.2
Q5	0.24	**0.83**	0.16	0.77	0.23	1.3
Q6	**0.65**	0.20	0.23	0.52	0.48	1.5
Q7	**0.79**	0.10	0.06	0.64	0.36	1.0
Q8	0.38	**0.71**	0.30	0.74	0.26	2.0
Q9	**0.74**	0.23	0.14	0.62	0.38	1.3
Q10	0.28	0.82	0.10	0.76	0.24	1.3
Q11	**0.76**	0.28	0.12	0.66	0.34	1.3
Q12	0.23	0.19	**0.85**	0.82	0.18	1.2
Q13	**0.59**	0.32	0.26	0.52	0.48	1.9
Q14	**0.72**	0.31	0.28	0.69	0.31	1.7
Q15	0.34	**0.66**	0.24	0.61	0.39	1.8
Q16	**0.74**	0.27	0.17	0.65	0.35	1.4
Q17	0.21	0.19	**0.87**	0.83	0.17	1.2
Q18	0.61	0.50	0.23	0.67	0.33	2.2

Meaning of three factors?
- ***Confidentiality*** *(data kept secreet)*
- ***Authentication*** *(identities are verified)*
- ***Non-repudiation*** *(cannot deny transactions)*

Rotation changes variance explained by each component

	RC1	RC3	RC2
SS loadings	5.61	3.49	2.95
Proportion Var	0.31	0.19	0.16
Cumulative Var	0.31	0.51	**0.67**
Proportion Explained	0.47	0.29	0.24
Cumulative Proportion	0.47	0.76	1.00

Rotation doesn't change total variance explained

🤔

Three factors seems to offer finer nuance of meanings?
*(better for **explanation and understanding**)*

Two factor solution

```
principal(r = sec, nfactors = 2, rotate = "varimax")
```

	RC1	RC2	h2	u2	com
Q1	**0.78**	0.27	0.69	0.31	1.2
Q2	**0.60**	0.31	0.45	0.55	1.5
Q3	**0.69**	0.34	0.59	0.41	1.5
Q4	0.24	**0.86**	0.80	0.20	1.1
Q5	**0.62**	0.31	0.48	0.52	1.5
Q6	**0.65**	0.24	0.48	0.52	1.3
Q7	**0.73**	0.04	0.53	0.47	1.0
Q8	**0.67**	0.42	0.62	0.38	1.7
Q9	**0.75**	0.15	0.58	0.42	1.1
Q10	**0.65**	0.24	0.48	0.52	1.3
Q11	**0.79**	0.13	0.64	0.36	1.1
Q12	0.25	**0.86**	0.80	0.20	1.2
Q13	**0.65**	0.29	0.51	0.49	1.4
Q14	**0.76**	0.30	0.67	0.33	1.3
Q15	**0.61**	0.35	0.50	0.50	1.6
Q16	**0.76**	0.19	0.62	0.38	1.1
Q17	0.22	**0.88**	0.82	0.18	1.1
Q18	**0.76**	0.29	0.66	0.34	1.3

Confidentiality and Authentication are lumped together

Non-repudiation is still distinct

	RC1	RC2
SS loadings	7.52	3.39
Proportion Var	0.42	0.19
Cumulative Var	0.42	**0.61**
Proportion Explained	0.69	0.31
Cumulative Proportion	0.69	1.00

Not a great loss in variance explained

🤔

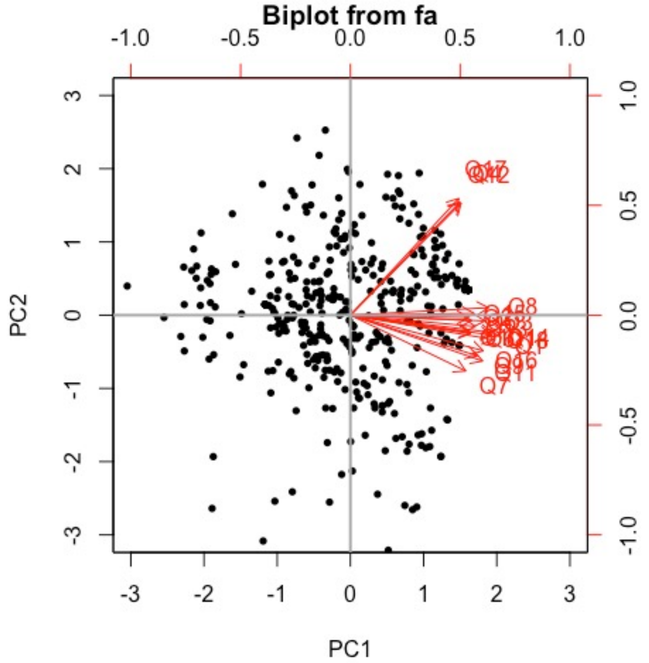
Two factors seems enough to capture most variance?
*(good enough for **data reproduction and prediction**, backed up by parallel analysis)*

Rotation and Interpretability

```
principal(r = sec, nfactors = 2, rotate = "none")
```

	PC1	PC2	h2	u2	com
Q4	**0.62**	**0.64**	0.80	0.20	2.0
Q12	**0.63**	**0.64**	0.80	0.20	2.0
Q17	**0.62**	**0.66**	0.82	0.18	2.0

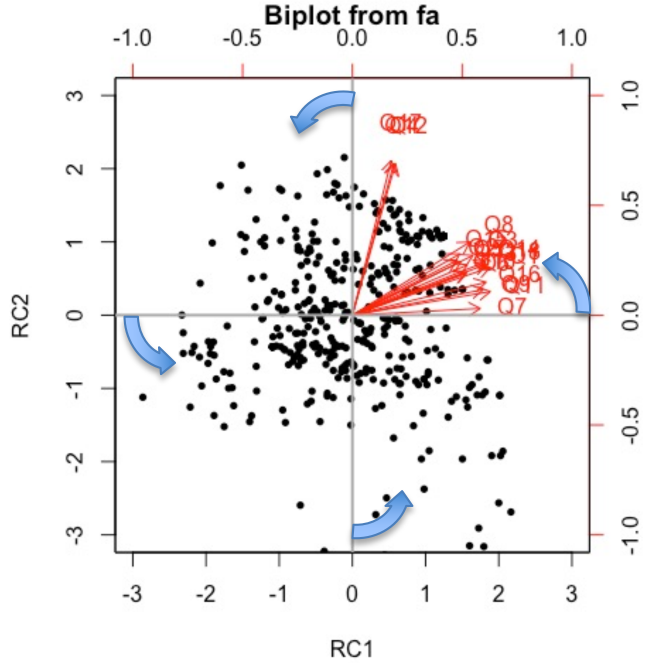
```
biplot(sec_pca2)
abline(h=0, v=0, lwd=2, col="darkgray")
```




```
principal(r = sec, nfactors = 2, rotate = "varimax")
```

	RC1	RC2	h2	u2	com
Q4	0.24	**0.86**	0.80	0.20	1.1
Q12	0.25	**0.86**	0.80	0.20	1.2
Q17	0.22	**0.88**	0.82	0.18	1.1

```
biplot(sec_pca2_rot)
abline(h=0, v=0, lwd=2, col="darkgray")
```



Rotation has made items that were ambivalent to line up more with one of the principal components

Measurement Models

Measuring Phenomena from Observed Variables

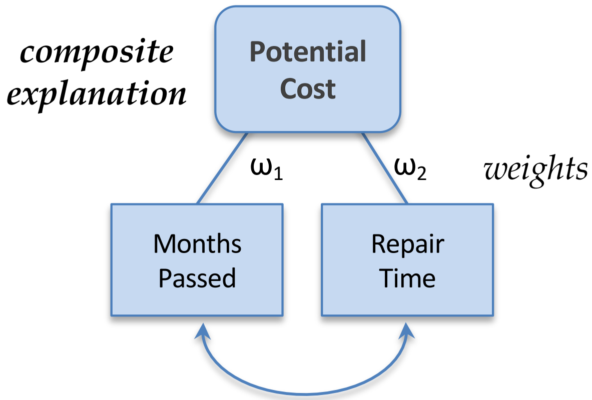
PCA with eigenvectors does this!

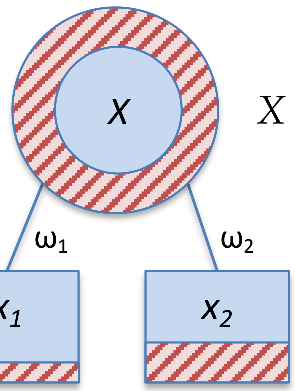
PCA with "factors" estimates this!
(but doesn't exactly do common factors)

Composite Model

Formative Measurement

Composite constructs are aggregates of items
(useful for prediction/recreating data)





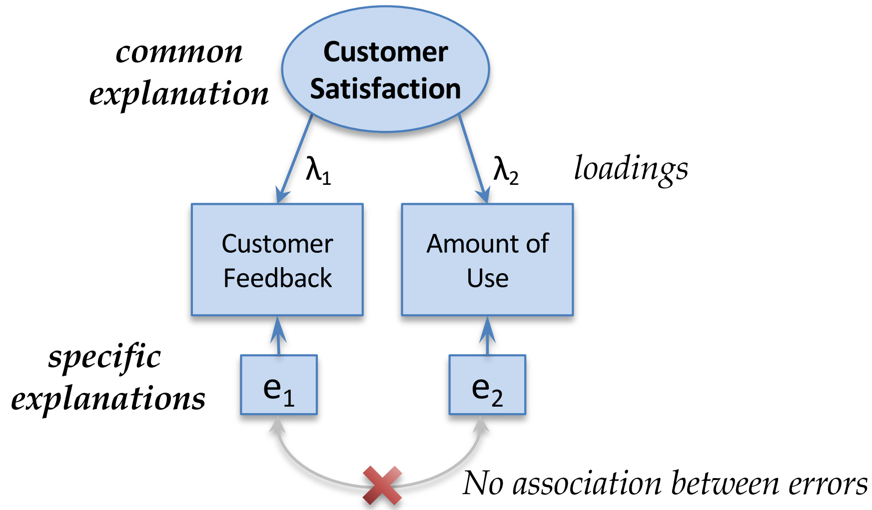
$$X = w_1 x_1 + w_2 x_2 + \varepsilon$$

composite measurement combines shared variance, unique variance

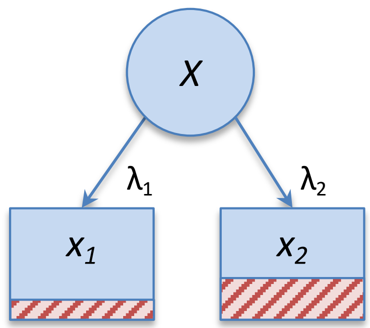
Common Factor Model

Reflective Measurement

Latent constructs are free of measurement error!
(useful for explanation/interpretation)



$$x_1 = \lambda_1 X + \varepsilon$$
$$x_2 = \lambda_2 X + \varepsilon$$



Reflective Measurement captures only the common variance of items, and not its specific variances (measurement errors?)

Structural model

Estimating the Relationship Between Constructs

$$Y = b_0 + b_1 X + \varepsilon$$

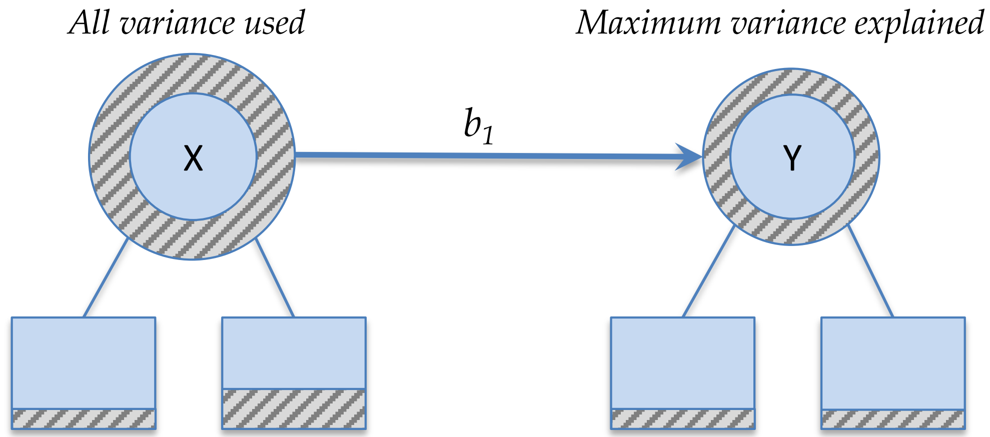
Composite Models

Suitable for Explanation

Correct way to model known composites

Better for Prediction

Composite model maximizes captured variance
Better for predicting actual outcomes



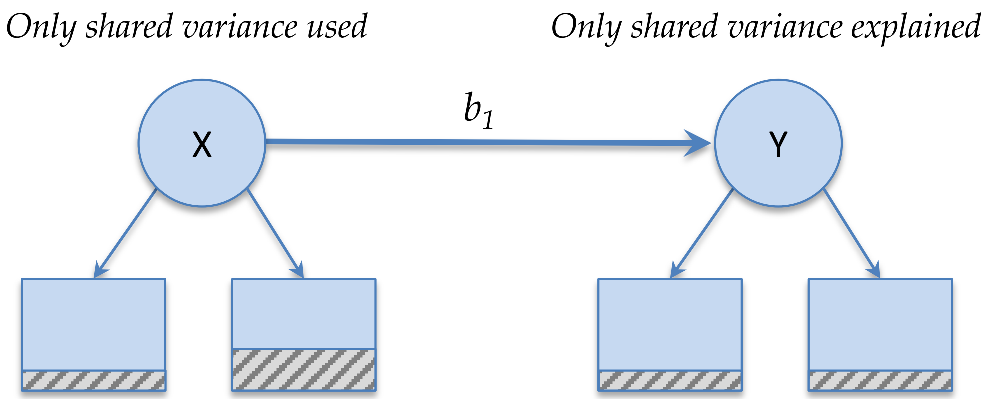
Factor Models

Better for Explanation

Factor model maximizes shared variance
Useful for modeling pure latent, abstract factors

NOT suitable for prediction

Factors scores are not determinable

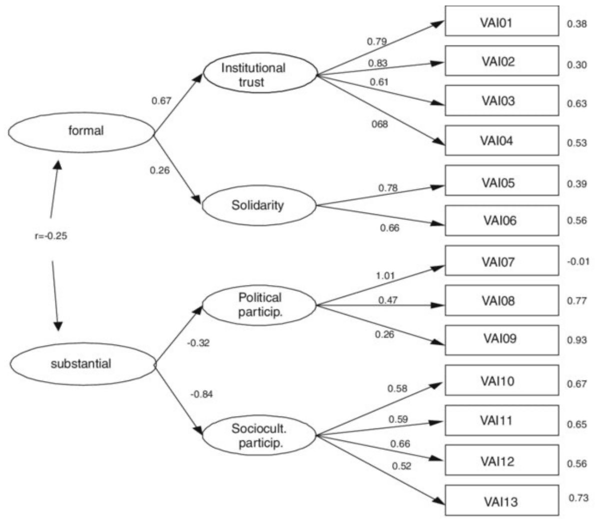


Structural Equation Models (SEM)

Using constructs to model concepts measured by multiple items

Social Sciences

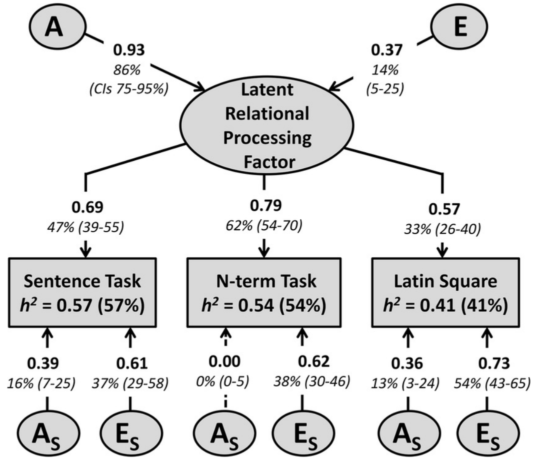
(Psychology/Sociology/Management)



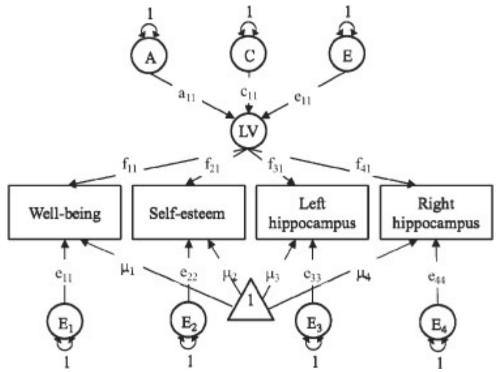
https://www.researchgate.net/publication/257664169_Construction_Validation_and_Application_of_the_Measurement_of_Social_Cohesion_in_47_European_Countries_and_Regions/figures?lo=

Life Sciences

(Genetics/Neuroscience)



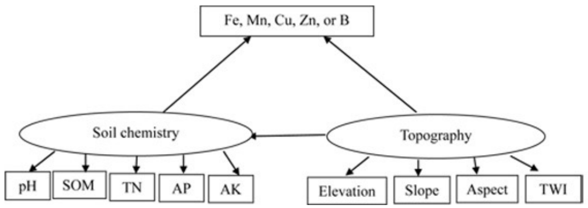
https://openi.nlm.nih.gov/detailedresult.php?img=PMC4393228_pone.0123886.g002&req=4



https://www.researchgate.net/publication/223984074_A_multivariate_twin_study_of_hippocampal_volume_self-esteem_and_well-being_in_middle-aged_men

Natural Sciences

(Chemistry)

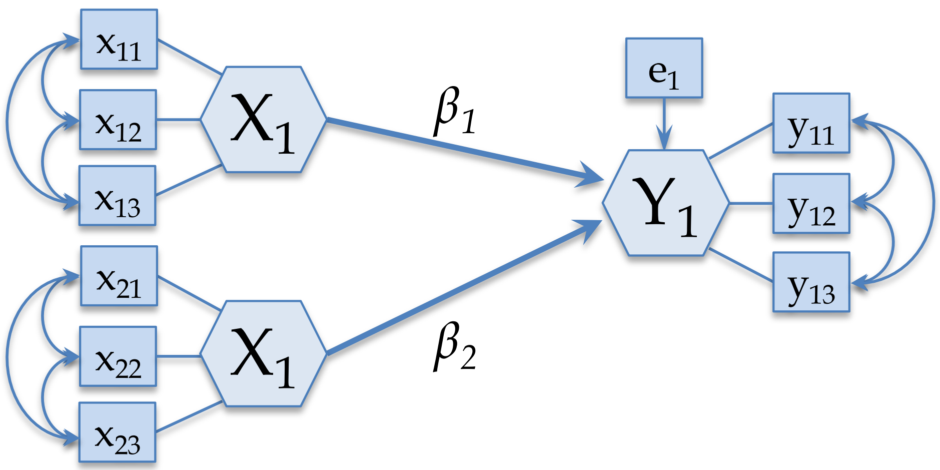


http://www.scielo.cl/scielo.php?pid=S0718-95162016005000076

SEM with Composites

Partial Least Squares – Path Modeling (PLS-PM)

Composite Measurement
(but factors can be estimated)



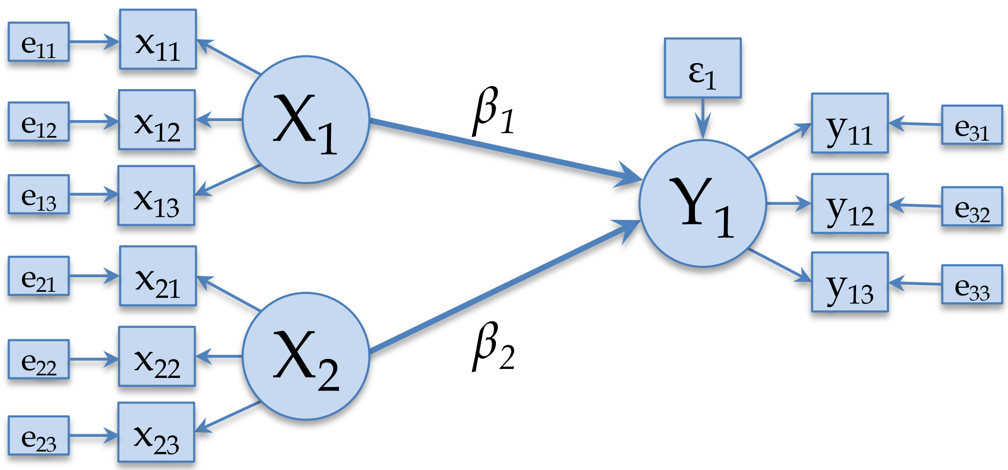
Non-parametric:
- *no global solution formula*
- *no global goodness-of-fit*
- *no inferential statistics (standard errors, etc.)*

"oriented to optimizing explained variances"

SEM with Common Factors

Covariance-based SEM (CB-SEM)

Common Factor Measurement
(factor scores are indeterminate)



Parametric:
- *Global solution and goodness-of-fit*
- *Inferential statistics (standard errors, etc.)*

"oriented towards modeling latent constructs"

SEM Example: *Security Survey*

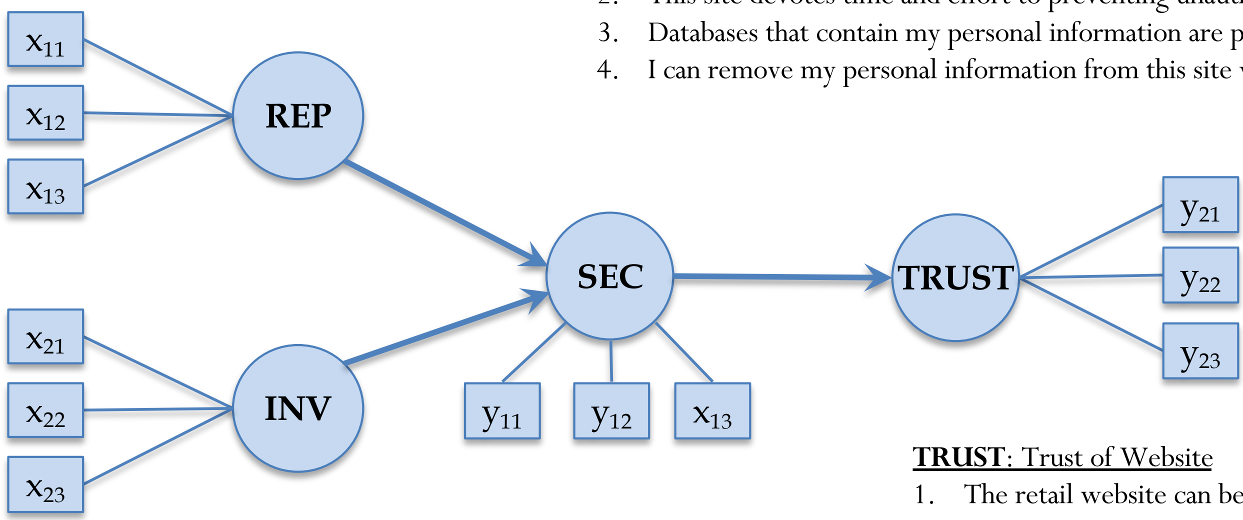
Survey Measurement Items

<u>REP</u><u>: Perceived Reputation</u>
1. This company is well known
2. This company has a good reputation
3. This company has a good reputation in its market
4. Feedback, ratings and/or reviews of this website are available

<u>INV</u><u>: Perceived Investment in Website</u>
1. A lot of time seems to have been invested in developing this website.
2. A lot of effort seems to have been invested in developing this website.
3. A lot of money seems to have been invested in developing this website.

Should we model these using ***composites*** *or* ***common factors****?*

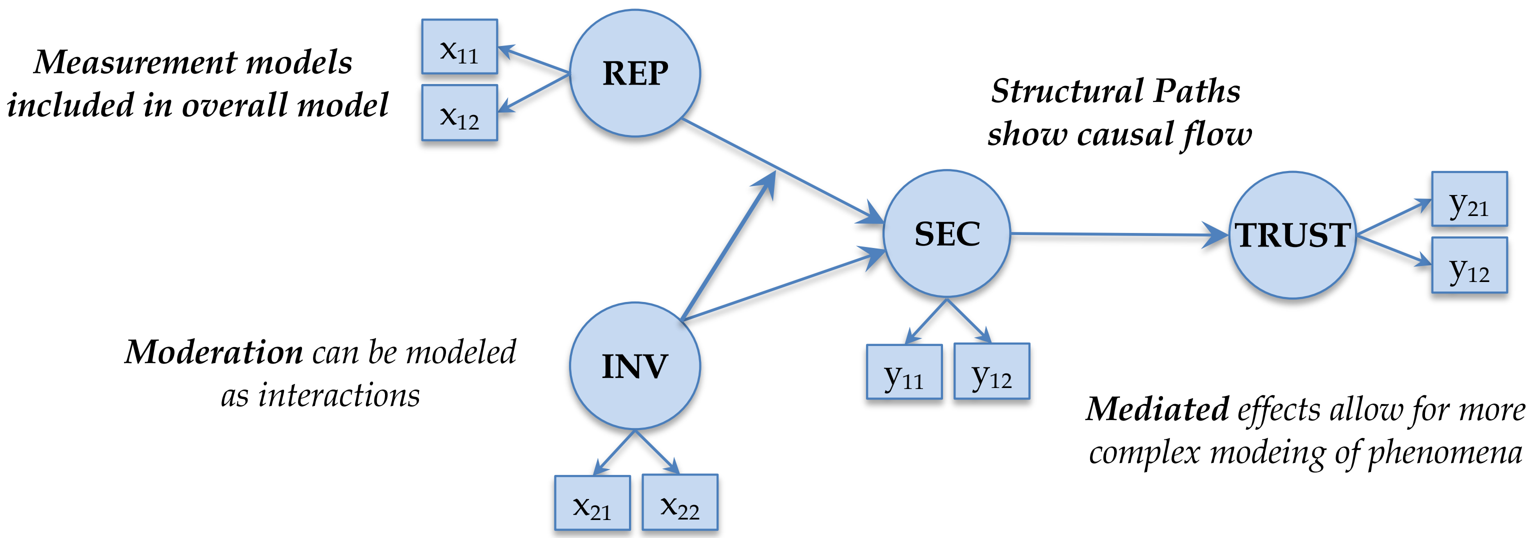
<u>SEC</u><u>: Perception of privacy protection</u>
1. This site never sells my personal information in their computer databases to other companies
2. This site devotes time and effort to preventing unauthorized access to my personal information
3. Databases that contain my personal information are protected from unauthorized access
4. I can remove my personal information from this site when I want to



<u>TRUST</u><u>: Trust of Website</u>
1. The retail website can be trusted at all times
2. The retail website can be counted on to do what is right
3. The retail website has high integrity
4. The retail website is competent and knowledgeable

Measurement, Multiple Outcomes, Moderation, Mediation

Allows us to create more complex regression models



SEM Software

R packages

Commercial Software

PLS-PM

sem-pls

```
R> data("ECSIsm")
R> ECSIsm

      source         target
 [1,] "Image"        "Expectation
 [2,] "Expectation"  "Quality"
 [3,] "Expectation"  "Value"
 [4,] "Quality"      "Value"
 [5,] "Image"        "Satisfactio
 [6,] "Expectation"  "Satisfactio
 [7,] "Quality"      "Satisfactio
 [8,] "Value"        "Satisfactio
 [9,] "Satisfaction" "Complaints"
[10,] "Image"        "Loyalty"
[11,] "Satisfaction" "Loyalty"
[12,] "Complaints"   "Loyalty"
```

plspm

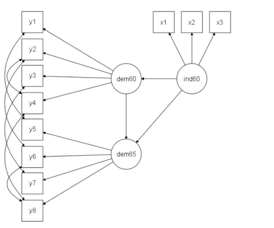
```
# path matrix (inner model realtionships)
AGRIN = c(0, 0, 0)
INDEV = c(0, 0, 0)
POLINS = c(1, 1, 0)
rus_path = rbind(AGRIN, INDEV, POLINS)

# add optional column names
colnames(rus_path) = rownames(rus_path)
```

SmartPLS

Smart™ Next Generation Path Modeling
P L S

CB-SEM

Lavaan




```
myModel <- '
   # latent variables
     ind60 =~ x1 + x2 + x3
     dem60 =~ y1 + y2 + y3 + y4
     dem65 =~ y5 + y6 + y7 + y8
   # regressions
     dem60 ~ ind60
     dem65 ~ ind60 + dem60
   # residual covariances
     y1 ~~ y5
     y2 ~~ y4 + y6
     y3 ~~ y7
     y4 ~~ y8
     y6 ~~ y8
'
fit <- sem(model = myModel,
           data  = PoliticalDemocracy)
summary(fit)
```

AMOS

SPSS
AMOS 24

Advantages
Open source - extendable with code
Free!

Disadvantages
Hard to write syntax

Advantages:
Very easy to use!

Disadvantages
Not free
Cannot extend their features

SEMinR – SEM Modeling, Estimation, and Visualization for R

https://github.com/sem-in-r/seminr

Install from CRAN

```
install.packages("seminr")
```

2016-08	Started work on SEMinR PLS-PM only
2018-01	Released on CRAN
2019-09	v1.0 – PLS-PM + CBSEM
2021-03	v2.0 – visualizations of graphical models
2022-01	v2.3 – latest release!

Created by former BACS students!

Juan Manuel Estrada Velasquez
National Tsing Hua University
Developed initial code to implement PLS algorithm

Nicholas Danks
Trinity College Dublin
Primary package maintainer

Soumya Ray
National Tsing Hua University
Package creator and CB-SEM additions

André Calero Valdez
RWTH Aachen University
Graphics and Design Applications

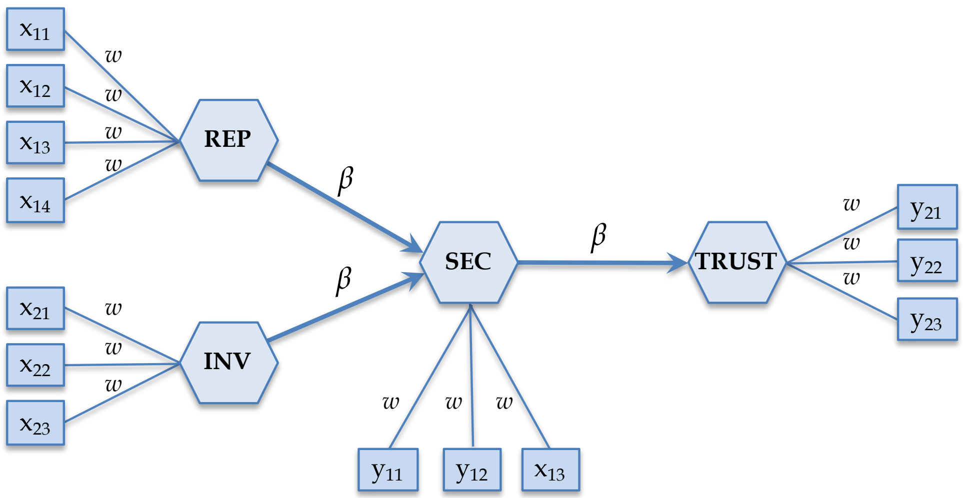
SEMinR: SEM Modeling, Estimation, and Visualization for R

Load library and data

```
library(seminr)
sec = read.csv("security_data.csv")
```

Define measurement and structural models

```
# Measurement Model
sec_mm <- constructs(
  composite("REP", multi_items("PREP", 1:4)),
  composite("INV", multi_items("PINV", 1:3)),
  composite("SEC", multi_items("PSEC", 1:4)),
  composite("TRUST", multi_items("TRST", 1:4))
)

# Structural Model
sec_sm <- relationships(
  paths(from = c("REP", "INV"), to = "SEC"),
  paths(from = "SEC", to = "TRUST")
)
```



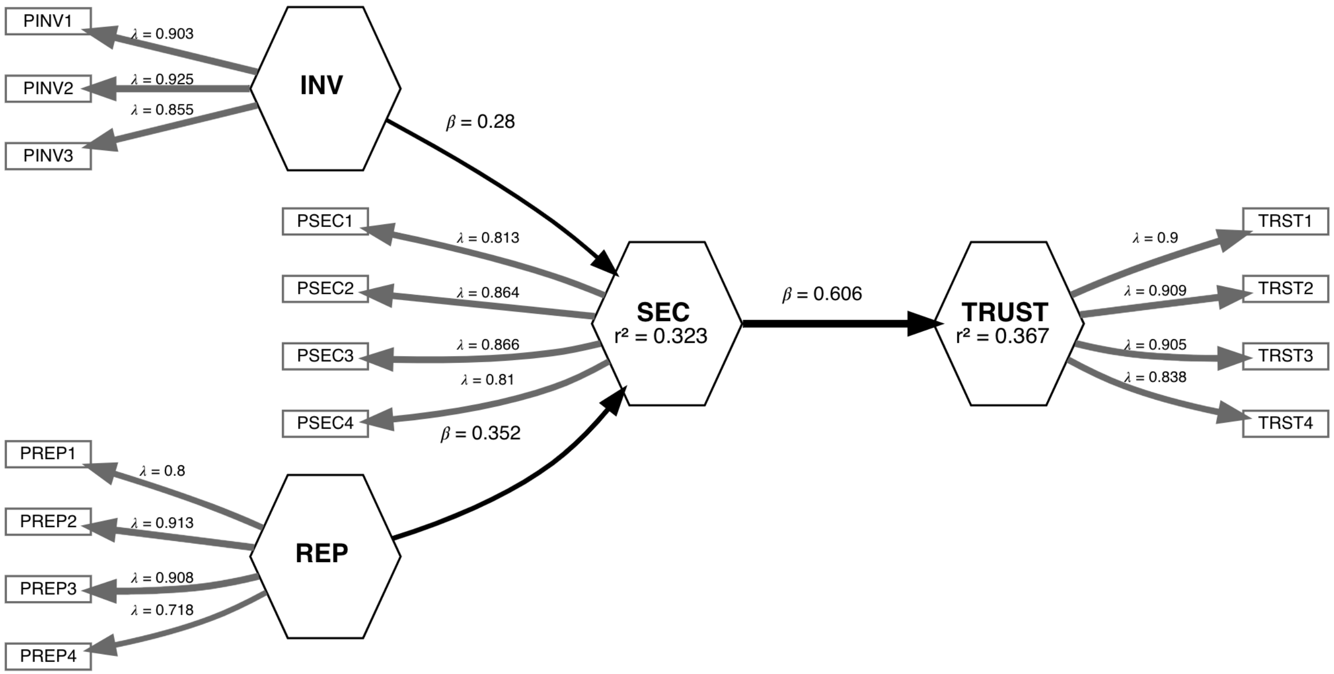
Run estimation algorithm

```
sec_pls <- estimate_pls(data = sec,
                        measurement_model = sec_mm,
                        structural_model = sec_sm)
```

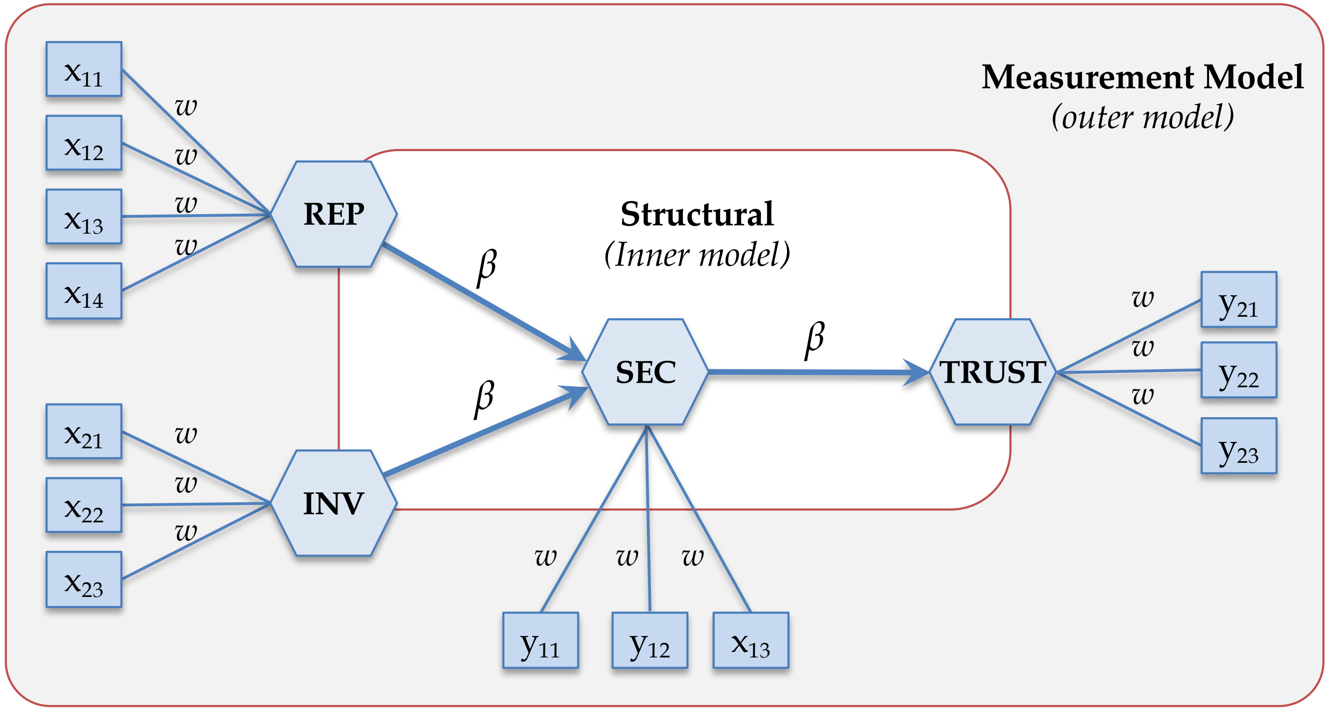
estimate** the models using **PLS

Visualize Results

```
plot(sec_pls)
```



PLS-SEM: Non-parametric, Iterative Estimation



PLS cannot estimate parameters (ω, β) directly – the model is too complex

Instead, it tries to computationally guess, and re-guess until its answers stabilize.

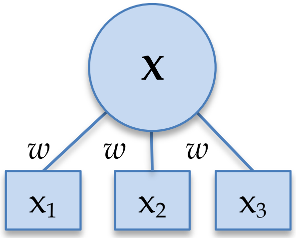
Iterative algorithm (loop):

1st stage: solves blocks of measurement model (weights, loadings)

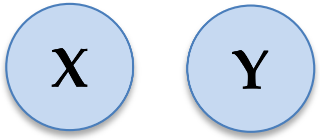
2nd stage: solves structural model (path coefficients)

Stops when estimates (λ, β) stop changing

Weights




```
sec_report$weights

        REP   INV   SEC TRUST
PREP1 0.215 0.000 0.000 0.000
PREP2 0.334 0.000 0.000 0.000
PREP3 0.349 0.000 0.000 0.000
PREP4 0.287 0.000 0.000 0.000
PINV1 0.000 0.363 0.000 0.000
PINV2 0.000 0.395 0.000 0.000
PINV3 0.000 0.359 0.000 0.000
PSEC1 0.000 0.000 0.277 0.000
PSEC2 0.000 0.000 0.314 0.000
PSEC3 0.000 0.000 0.302 0.000
PSEC4 0.000 0.000 0.298 0.000
TRST1 0.000 0.000 0.000 0.282
TRST2 0.000 0.000 0.000 0.280
TRST3 0.000 0.000 0.000 0.286
TRST4 0.000 0.000 0.000 0.278
```

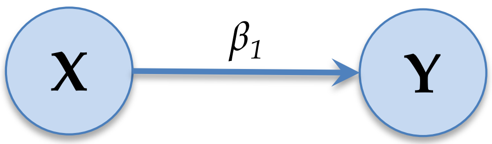


Composite Scores

```
sec_report$composite_scores

           REP        INV         SEC      TRUST
1   -0.2413125 -0.3992500 -0.28441353  0.6008351
2    0.6206341  0.2760630  0.09051111  0.7724682
3    0.2038340 -0.2185641  0.45053250  0.4136877
         .          .           .          .
         .          .           .          .
403 -2.5151146 -1.3410739 -0.89774320 -0.6820062
404  0.8707435  0.3426908 -0.06138858  1.1506011
405  0.2360766  0.2760631 -0.86793669 -1.6060672
```

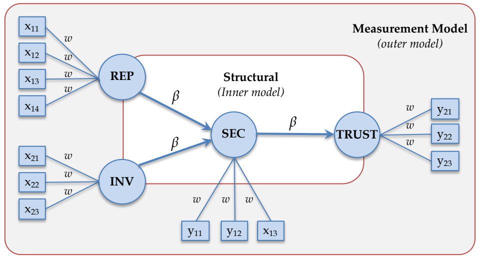
Path Coefficients

Path Fit (R^2)




```
sec_report$paths

         SEC TRUST
R^2    0.323 0.367
AdjR^2 0.319 0.365
REP    0.352    NA
INV    0.280    NA
SEC       NA 0.606
```

SEMinR: Bootstrapping for Significance



Non-parametric:
- no inferential statistics (no standard errors of ω, β)
- no global goodness-of-fit

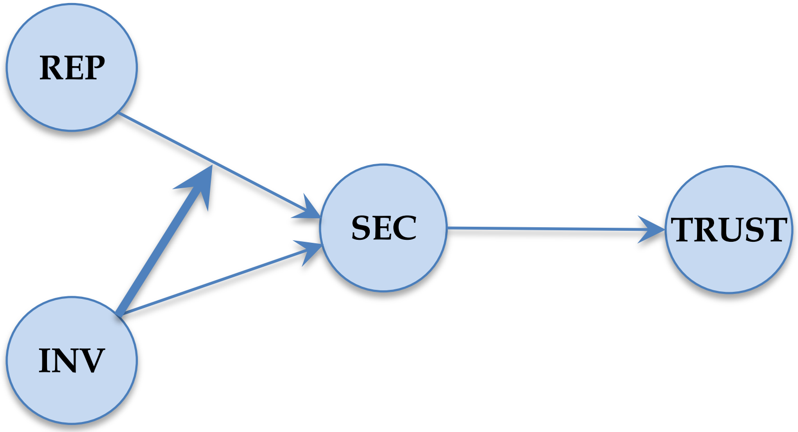
Bootstrapping
We can re-estimate our model on bootstrapped resamples of the data:

```
boot_pls <- bootstrap_model(sec_pls, nboot = 1000)

summary(boot_pls)

Results from Bootstrap resamples:  1000

Bootstrapped Structural Paths:
                Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI 97.5% CI
REP  ->  SEC            0.352          0.352        0.053   6.684   0.248    0.452
INV  ->  SEC            0.280          0.281        0.059   4.727   0.164    0.395
SEC  ->  TRUST          0.606          0.607        0.034  17.685   0.537    0.671
```

SEMinR: Interaction Terms

Conceptual Interaction Model

Our researcher believes that the look-and-feel of a website (INV) is so important that people care less about the reputation of the company (REP) as the quality of its look-and-feel gets better.

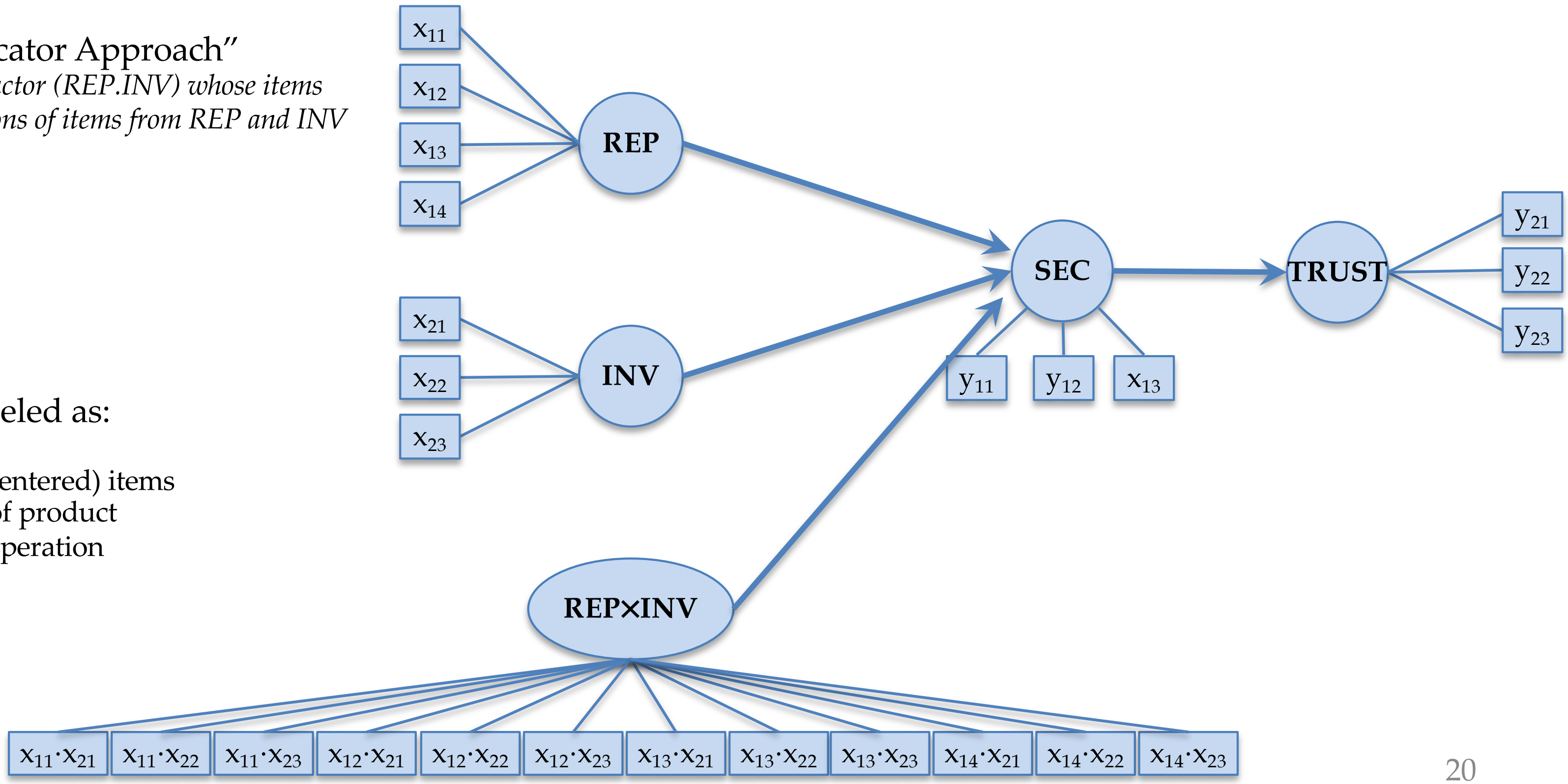


Operationalizing Interactions in PLS

"Repeated Product-Indicator Approach"
Create a new interaction term factor (REP.INV) whose items are the product of all combinations of items from REP and INV

Interactions can be modeled as:

- Product of ***original*** items
- Product of ***scaled*** (mean-centered) items
- ***Orthogonalized*** residual of product
- Computed in a *two-stage* operation



SEMinR: Interaction Terms

Orthogonalized interaction term

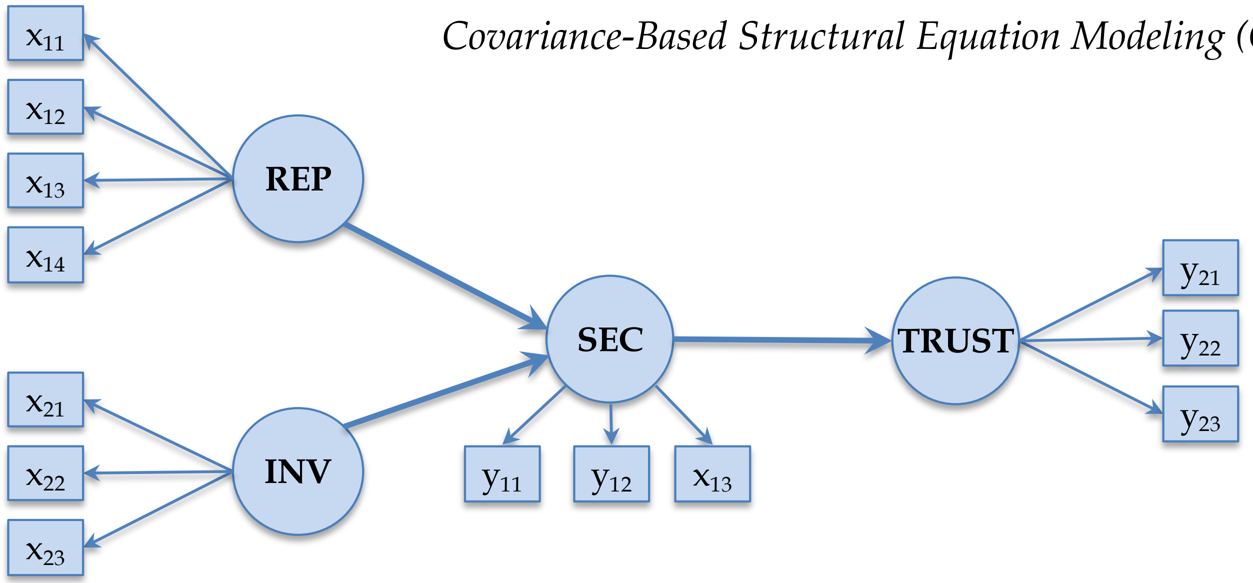
```
sec_intxn_mm <- constructs(
  composite("REP", multi_items("PREP", 1:4)),
  composite("INV", multi_items("PINV", 1:3)),
  interaction_term(iv="REP", moderator="INV", method=orthogonal),
  composite("SEC", multi_items("PSEC", 1:4)),
  composite("TRUST", multi_items("TRST", 1:4))
)

sec_intxn_sm <- relationships(
  paths(from = c("REP", "INV", "REP*INV"), to = "SEC"),
  paths(from = "SEC", to = "TRUST")
)

sec_intxn_pls <- estimate_pls(
  data = sec,
  measurement_model = sec_intxn_mm,
  structural_model = sec_intxn_sm
)

summary(sec_intxn_pls)
```

```
 Total Iterations: 5
Path Coefficients:
           SEC TRUST
R^2      0.329 0.367
AdjR^2   0.324 0.365
REP      0.352     .
INV      0.280     .
REP*INV -0.071     .
SEC          . 0.606
```

```
boot_intxn <- bootstrap_model(sec_intxn_pls, nboot = 1000)

summary(boot_intxn)

 Bootstrapped resamples: 1000

Structural Path p-values:
          SEC TRUST
REP     0.000     .
INV     0.000     .
REP*INV 0.127     .
SEC         . 0.000
```

Common Factor Models

Covariance-Based Structural Equation Modeling (CB-SEM)



Define all factors as "**reflective**"

```
sec_cf_mm <- constructs(
  reflective("REP", multi_items("PREP", 1:4)),
  reflective("INV", multi_items("PINV", 1:3)),
  reflective("SEC", multi_items("PSEC", 1:4)),
  reflective("TRUST", multi_items("TRST", 1:4))
)
```

Estimate model using CB-SEM

```
sec_cf_pls <- estimate_cbsem(
  data = sec,
  measurement_model = sec_cf_mm,
  structural_model = sec_sm
)
```

```
sec_cf_report$paths

$coefficients
           SEC     TRUST
R^2 0.4419771 0.5007483
REP 0.4194503        NA
INV 0.3103883        NA
SEC        NA 0.7076357

$pvalues
              SEC  TRUST
REP 2.139136e-09    NA
INV 1.276164e-05    NA
SEC           NA     0
```

Creating a **Domain-Specific Language** (DSL)

$\underbrace{\text{Domain-Specific Language}}_{\textit{syntax that is specialized to a particular domain or field}}$

Most of SEMinR's functions simply make data structures for you:

```
multi_items("PREP", 1:4)
[1] "PREP1" "PREP2" "PREP3" "PREP4"
```

is the same as:

```
c("PREP1", "PREP2", "PREP3", "PREP4")
[1] "PREP1" "PREP2" "PREP3" "PREP4"
```

🤔 *What do `constructs()`, `composite()`, `paths()`, and `relationships()` do?*

```
# Measurement Model
sec_mm <- constructs(
  composite("REP", multi_items("PREP", 1:4)),
  composite("INV", multi_items("PINV", 1:3)),
  composite("SEC", multi_items("PSEC", 1:4)),
  composite("TRUST", multi_items("TRST", 1:4))
)

# Structural Model
sec_sm <- relationships(
  paths(from = c("REP", "INV"), to = "SEC"),
  paths(from = "SEC", to = "TRUST")
)
```

syntax now reads like the model:

composite constructs *using* ***multiple items***

relationships *between constructs with*
paths from ____ to ____

A Closer look at **Functions in SEMinR**

```
sec_intxn_mm <- constructs(
  composite("REP", multi_items("PREP", 1:4)),
  composite("INV", multi_items("PINV", 1:3)),
  interaction_term(iv="REP", moderator="INV", method=orthogonal),
  composite("SEC", multi_items("PSEC", 1:4)),
  composite("TRUST", multi_items("TRST", 1:4))
)
```

What kind of object is "`orthogonal`"?

```
> orthogonal
function (iv, moderator, weights)
{
  …
}
<bytecode: 0x10fa03d28>
<environment: namespace:seminr>
```

Users can plug in their own functions!

Less if-then-else code in SEMinR
(just runs the supplied function)

```
interaction_term(iv="REP", moderator="INV", method=orthogonal)
```

What does the function `interaction_term()` return?

Function Currying

Haskell Brooks **Curry**
Mathemetician, Logician

Usual use of functions:

```r
set.seed(42)
my_x_data <- data.frame(x1 = rnorm(100), x2 = rnorm(100))

# somewhere else in our code:
y1 = rnorm(100)
lm(y1 ~ x1 + x2, data=my_x_data)

# yet another place in our code:
y2 = rnorm(100)
lm(y2 ~ x1 + x2, data=my_x_data)
```

We have to ***pass*** *our predictors and* ***repeat*** *our model everywhere they are needed.*

`lm(y2 ~ x1 + x2, data=my_x_data)`

let's put this into a curried function…

```r
y_regr <- function(x_data) {
  invisible(
    function(y) {
      lm(y ~ x1 + x2, data = x_data)
    }
  )
}
```

Curried function:

We ***create a function*** *that processes some of our parameters…*

`invisible(…)` *- hides output of some code*

… and it ***defines and return a new function*** *aksing for more parameters*

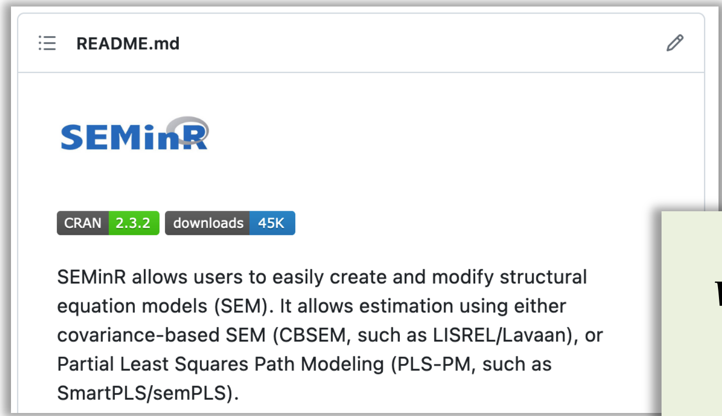
```r
set.seed(42)
my_x_data <- data.frame(x1 = rnorm(100), x2 = rnorm(100))
lm_with_x <- y_regr(my_x_data)

# somewhere else in our code:
y1 = rnorm(100)
lm_with_x(y1)

# yet another place in our code:
y2 = rnorm(100)
lm_with_x(y2)
```

Our code is now ***cleaner****, and*
doesn't require passing *data and model everywhere*

This is a powerful feature when you are making your own library of functions for others to put together

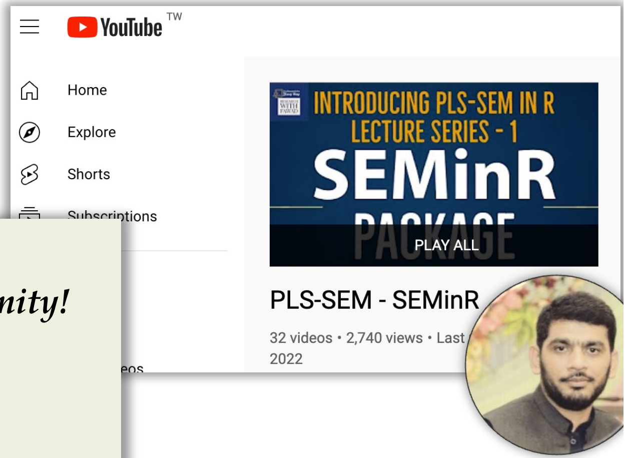
The SEMinR Community

Github Repository



https://github.com/sem-in-r/seminr

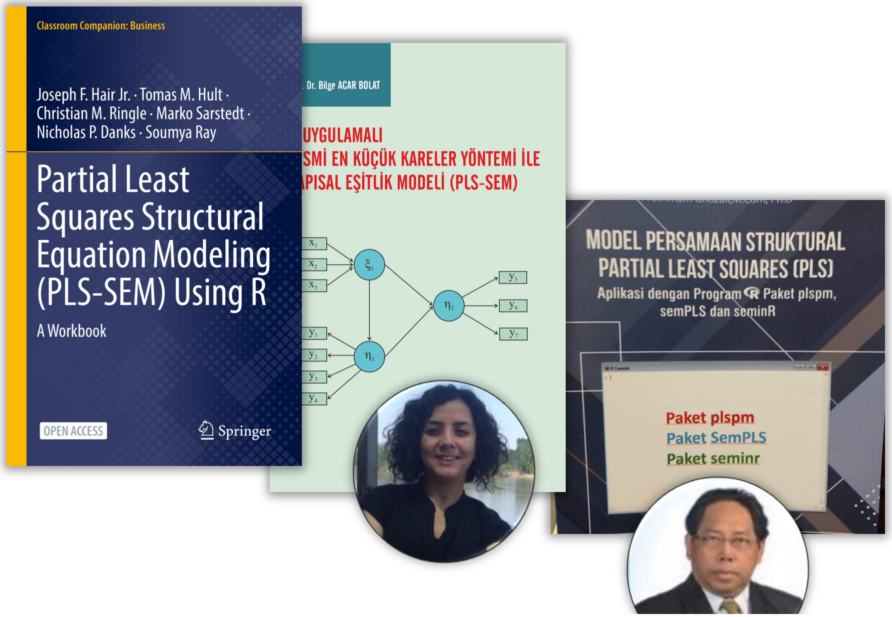
Video Series



Writing open-source code is building a community!

Others contribute by:
Helping other users
Bug reporting / fixes
Documentation (books, videos)
Spreading the word!

Published Books



Facebook Group

https://www.facebook.com/groups/seminr