Confirmatory Factor Analysis

Theory Construction and Statistical Modeling

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Outline

South African Personality Inventory Project

Carin Hill
Leon Jackson
Deon Meiring
J. Aleweyn Nel

Ian Rothmann
Michael Temane
Velichko H. Valchev
Fons J. R. van de Vijver

Nel, J. A., Valchev, V. H., Rothmann, S., van de Vijver, F. J. R., Meiring, D., & de Bruin, G. P. (2012). Exploring the personality structure in the 11 languages of South Africa. *Journal of Personality*, *80*, 915–948.

SAPI details

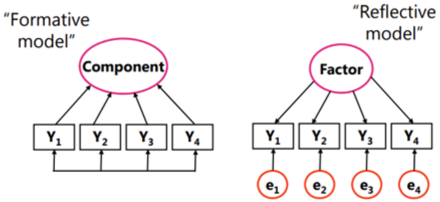
- 1216 participants from 11 official language groups
- From about 50,000 descriptive responses to 262 personality items
- Nine personality clusters:
  - Conscientiousness
  - Emotional Stability
  - Extraversion
  - Facilitating
  - Integrity
  - Intellect
  - Openness
  - Relationship Harmony
  - Soft-Heartedness (Ubuntu)
- Our data: selection of 1000 participants

Factor Analysis

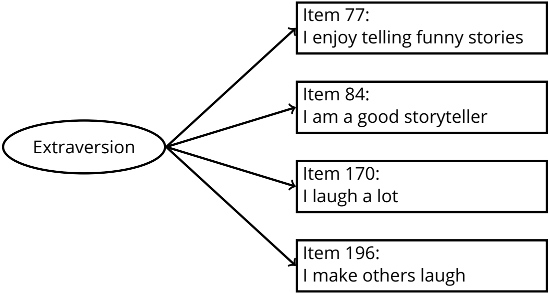
Factor Analysis: Modeling measurement of a latent variable

- EFA: Exploratory Factor Analysis.
- CFA: Confirmatory Factor Analysis.

Both EFA and CFA use a ”reflective” measurement model, not a ”formative” model.

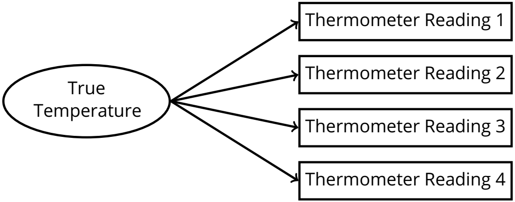


Reflective Constructs



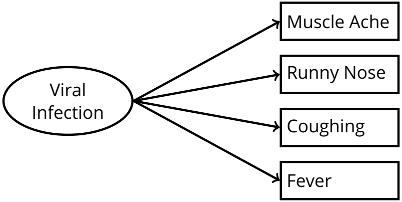
- Items are dependent variables, caused by the factor!
- Latent variable ‘extraversion’ explains item correlations: The factor is the reason for the covariances/correlations.

Reflective Constructs



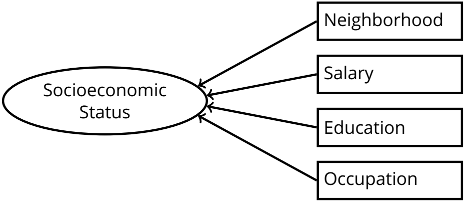
Thermometer readings are the dependent variables, caused by the temperature!

Reflective Constructs



Symptoms are the dependent variables, caused by the viral infection!

Formative Constructs



SES is an *index* defined as a (weighted) sum of the observed items.

- SES is the (latent) dependent variable, predicted by the items.
- This model is not empirically testable.

Interesting read

Interesting read on theory & latent variables:

Borsboom, D., Mellenbergh, G.J., & Van Heerden, J. (2003). The theoretical status of latent variables. *Psychological review, 110*(2), 203.

Confirmatory or Exploratory?

Two Subscales of Extraversion

Having Fun

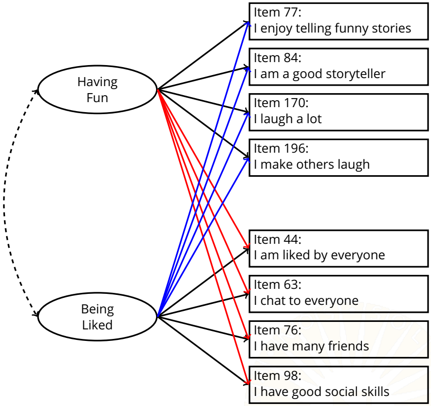
- Item 77: I enjoy telling funny stories
- Item 84: I am a good storyteller
- Item 170: I laugh a lot
- Item 196: I make others laugh

Being Liked

- Item 44: I am liked by everyone
- Item 63: I chat to everyone
- Item 76: I have many friends
- Item 98: I have good social skills

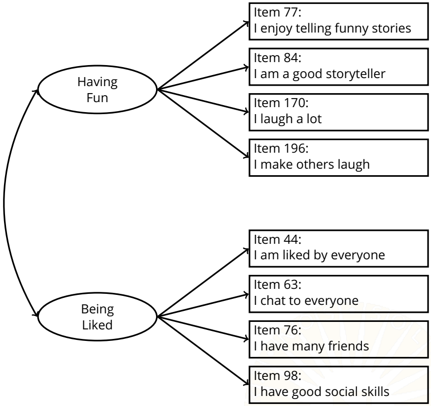
EFA

- All items load onto all factors
- No hypothesized measurement model
- Estimating latent covariances is optional
  - Oblique factors → Estimated
  - Orthogonal factors → Fixed
- Solution is not unique
- Use rotation to improve interpretability



CFA

- The statistical model represents the hypothesized measurement model
- No cross-loadings unless they're predicted by theory
- Almost always estimate the latent covariances
- A unique solution exists



CFA in R

Example: Estimate a CFA Model

Load the SAPI data.

```
dataDir <- "../data/"
sapi <- read.table(paste0(dataDir, "sapi.txt"),
                   header = TRUE,
                   na.strings = "-999")
```

Specify the **lavaan** model syntax for the SAPI extraversion CFA.

```
mod1 <- '
fun   =~ Q77 + Q84 + Q170 + Q196
liked =~ Q44 + Q63 + Q76  + Q98
'
```

Use the `cfa()` function to estimate the model.

```
library(lavaan)
out1 <- cfa(mod1, data = sapi)
```

Example: Summarize the Fitted CFA

```
partSummary(out1, 1:4)

lavaan 0.6-19 ended normally after 30 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        17

                                                  Used       Total
  Number of observations                           959        1000

Model Test User Model:

  Test statistic                               130.193
  Degrees of freedom                                19
  P-value (Chi-square)                           0.000
```

Example: Summarize the Fitted CFA

```
partSummary(out1, 5:7)

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  fun =~
    Q77               1.000
    Q84               0.761    0.051   14.902    0.000
    Q170              0.634    0.047   13.558    0.000
    Q196              0.795    0.046   17.381    0.000
  liked =~
    Q44               1.000
    Q63               1.512    0.147   10.278    0.000
    Q76               1.483    0.149    9.955    0.000
    Q98               1.243    0.119   10.462    0.000
```

Example: Summarize the Fitted CFA

```
partSummary(out1, 8:9)

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  fun ~~
    liked             0.231    0.025    9.234    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .Q77               0.548    0.038   14.389    0.000
   .Q84               0.727    0.039   18.703    0.000
   .Q170              0.687    0.035   19.572    0.000
   .Q196              0.364    0.025   14.731    0.000
   .Q44               0.662    0.034   19.291    0.000
   .Q63               0.807    0.048   16.943    0.000
   .Q76               0.966    0.054   17.931    0.000
   .Q98               0.469    0.029   16.121    0.000
    fun               0.627    0.056   11.303    0.000
    liked             0.182    0.029    6.290    0.000
```

Example: Model Fit Statistics

Example: Model Fit Statistics

```
fitMeasures(out1)

               npar                fmin               chisq
             17.000               0.068             130.193
                 df              pvalue      baseline.chisq
             19.000               0.000            1574.886
        baseline.df     baseline.pvalue                 cfi
             28.000               0.000               0.928
                tli                nnfi                 rfi
              0.894               0.894               0.878
                nfi                pnfi                 ifi
              0.917               0.622               0.929
                rni                logl   unrestricted.logl
              0.928          -10147.587          -10082.491
                aic                 bic              ntotal
          20329.175           20411.895             959.000
               bic2               rmsea      rmsea.ci.lower
          20357.903               0.078               0.066
     rmsea.ci.upper      rmsea.ci.level        rmsea.pvalue
              0.091               0.900               0.000
     rmsea.close.h0 rmsea.notclose.pvalue  rmsea.notclose.h0
              0.050               0.421               0.080
```

Example: Visualize the Fitted CFA

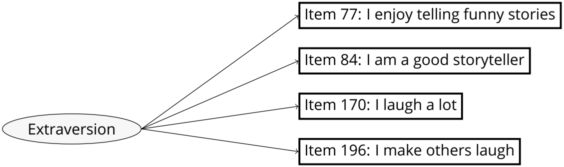
```r
library(lavaanPlot)
lavaanPlot(model = out1,
           node_options = list(shape = "box",
                               fontname = "Helvetica"),
           edge_options = list(color = "grey"),
           coefs = TRUE,
           stand = TRUE,
           covs = TRUE)
```

Example: Visualize the Fitted CFA

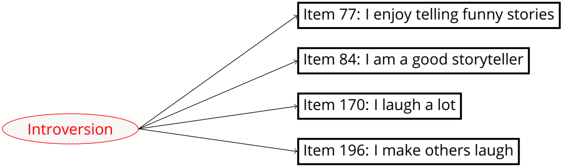
```
Error in path.expand(path):  invalid 'path' argument
```

Latent variable scaling

Latent variables are not observed, thus no inherent scale.



Latent variable scaling Ctd.



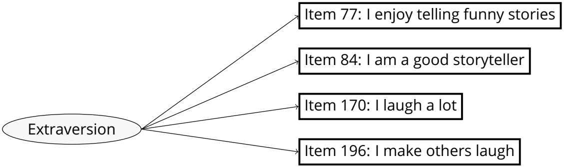
Therefore, set up model such that scale of latent variable is clear.

Two common ways

1. Marker-variable method
Constrain one of the factor loadings (default).

2. Reference group method:
Constrain the factor variance.

3. Effect coding:
Constrain the average of the loadings.



1. Marker-variable method (default)

Default parameterization:

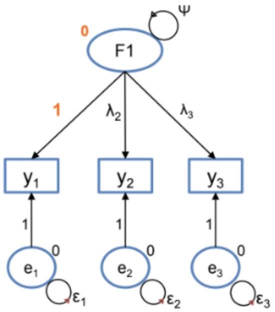
- First factor loading constrained at 1.
- Factor mean constrained at 0.

Other defaults:

- Mean of residuals is by definition 0.
- Residuals have a loading of 1.

Estimated:

- factor variance (Ψ),
- 'other' factor loadings (λ_2, λ_3),
- all item intercepts (ν_1, ν_2, ν_3),
- all residual variances (ϵ_1, ϵ_2, ϵ_3).



1. Default marker-variable method - lavaan

```
# Model
model.1CFA <- '
 Extraversion =~ Q77 + Q84 + Q170 + Q196
'

# Fit model
fit_1CFA <- cfa(model.1CFA, data=sapi,
                missing='fiml', fixed.x=F)   # use FIML
```

- First factor loading constrained at 1:

```
Extraversion =~
  Q77                  1.000
```

- Factor mean constrained at 0:

```
Extraversion       0.000
```

1. Default marker-variable method - lavaan Ctd

```
parameterEstimates(fit_1CFA)[1:4,-c(5,6,7)]

           lhs op  rhs   est ci.lower ci.upper
1 Extraversion =~  Q77 1.000    1.000    1.000
2 Extraversion =~  Q84 0.708    0.616    0.799
3 Extraversion =~ Q170 0.567    0.466    0.668
4 Extraversion =~ Q196 0.742    0.640    0.845
```

Factor loading of first indicator fixed to 1.
all other loadings are relative to that.

If reference category changed, other loadings also change.

2. Reference-group method

Parameterization:

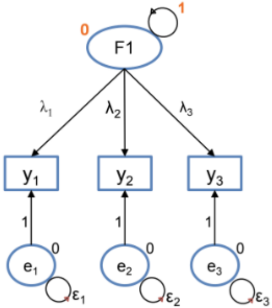
- Factor variance constrained at 1.
- Factor mean constrained at 0.

Defaults:

- Mean of residuals is by definition 0.
- Residuals have a loading of 1.

Estimated:

- all factor loadings (λ_1, λ_2, λ_3),
- all item intercepts (ν_1, ν_2, ν_3),
- all residual variances (ϵ_1, ϵ_2, ϵ_3).



2. Reference-group method - lavaan

```
# Model
model.1CFA_RefGr <- '
  # Free first factor loading, using: NA*
  Extraversion =~ NA*Q77 + Q84 + Q170 + Q196

  # Set factor variance to 1, using: 1*
  Extraversion ~~ 1*Extraversion
 '

# Fit model
fit_1CFA_RefGr <- cfa(model.1CFA_RefGr, data=sapi,
                missing='fiml', fixed.x=F)  # use FIML
```

- Factor variance constrained at 1:

  `Extraversion        1.000`

- Factor mean constrained at 0:

  `Extraversion        0.000`

2. Reference-group method - lavaan Ctd

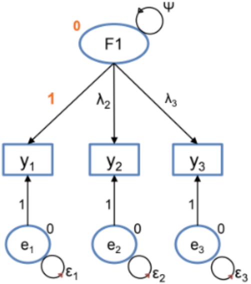
```
parameterEstimates(fit_1CFA_RefGr)[1:4,-c(5,6,7)]

           lhs op  rhs   est ci.lower ci.upper
1 Extraversion =~  Q77 0.835    0.759    0.910
2 Extraversion =~  Q84 0.591    0.520    0.662
3 Extraversion =~ Q170 0.473    0.404    0.543
4 Extraversion =~ Q196 0.619    0.559    0.680
```

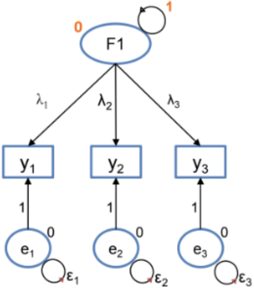
Advantage:
All factor loadings and scores on standardized metric.

Which method to choose?

1. Marker-variable method



2. Reference-group method



Does not matter for substantive conclusions.
Sometimes, pragmatic reasons.

MODEL ESTIMATION

Model-Implied Statistics

Most statistical estimation algorithms operate by minimizing the difference between two key reference points:

1. The *model-implied* statistics/predictions/fitted values
   - The sufficient statistics implied by the structure of your model.
   - Predicted/fitted values produced by your model.

2. The *observed* statistics/values
   - The sufficient statistics calculated from the observed data.
   - The raw outcome values from your dataset.

The predictions/implied statistics produced by a good model must be simpler than the analagous quantities in the observed data.

- A model that exactly replicates the obvserved data is overfitting.
- The inferences from such models won't generalize to the population.

Model-Implied Statistics

You should already be familiar with this idea from OLS regression.

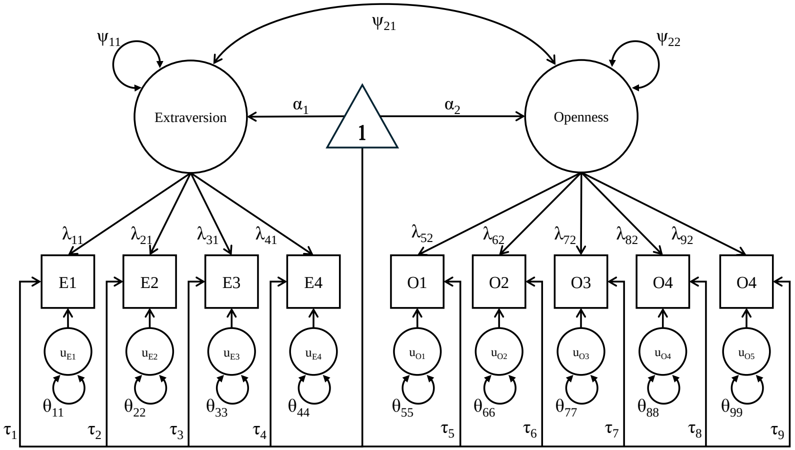
- The fitted values are the model implied statistics.

$$\hat{Y}_n = \hat{\beta}_0 + \sum_{p=1}^{P} \hat{\beta}_p X_{n,p}$$

- The raw outcome variable, Y, contains the observed values.
- Minimize the difference between $\hat{Y}$ and Y to estimate the model.

$$RSS = \sum_{n=1}^{N} \left(Y_n - \hat{Y}_n \right)^2$$

Fully Specified Path Diagram



Parameter Matrices

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} \quad \Psi = \begin{bmatrix} \psi_{11} & \\ \psi_{21} & \psi_{22} \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \\ \tau_6 \\ \tau_7 \\ \tau_8 \\ \tau_9 \end{bmatrix} \quad \Theta = \begin{bmatrix} \theta_{11} & & & & & & & & \\ 0 & \theta_{22} & & & & & & & \\ 0 & 0 & \theta_{33} & & & & & & \\ 0 & 0 & 0 & \theta_{44} & & & & & \\ 0 & 0 & 0 & 0 & \theta_{55} & & & & \\ 0 & 0 & 0 & 0 & 0 & \theta_{66} & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \theta_{77} & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{88} & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{99} \end{bmatrix} \quad \Lambda = \begin{bmatrix} \lambda_{11} & 0 \\ \lambda_{21} & 0 \\ \lambda_{31} & 0 \\ \lambda_{41} & 0 \\ 0 & \lambda_{52} \\ 0 & \lambda_{62} \\ 0 & \lambda_{72} \\ 0 & \lambda_{82} \\ 0 & \lambda_{92} \end{bmatrix}$$

Model-Implied Statistics

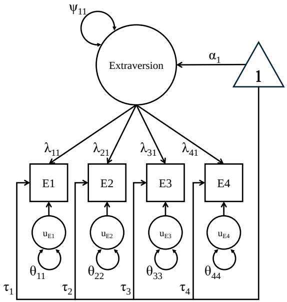
Model estimation for CFA/SEM follows the same principle.

$$\Sigma = \begin{bmatrix} \lambda_{11}\psi_{11}\lambda_{11} + \theta_{11} & & \\ \lambda_{11}\psi_{11}\lambda_{21} + \theta_{21} & \lambda_{21}\psi_{11}\lambda_{21} + \theta_{22} & \\ \lambda_{11}\psi_{11}\lambda_{31} + \theta_{31} & \lambda_{21}\psi_{11}\lambda_{31} + \theta_{32} & \lambda_{31}\psi_{11}\lambda_{31} + \theta_{33} \end{bmatrix}$$

$$\mu = \begin{bmatrix} \tau_1 + \lambda_{11}\alpha_1 & \tau_2 + \lambda_{22}\alpha_1 & \tau_3 + \lambda_{33}\alpha_1 \end{bmatrix}$$

Tracing Rules

Blah, blah, blah



Maximum Likelihood Estimation

ML estimation simply finds the parameter values that are “most likely” to have given rise to the observed data.

- The *likelihood* function is just a probability density (or mass) function with the data treated as fixed and the parameters treated as random variables.
- Having such a framework allows us to ask: “Given that I’ve observed these data values, what parameter values most probably describe these data?”

Maximum Likelihood Estimation

ML estimation is usually employed when there is no closed form solution for the parameters we seek.

- This is why you don't usually see ML used to fit general linear models.

After choosing a likelihood function, we iteratively optimize the function to produce the ML estimated parameters.

- In practice, we nearly always work with the natural logarithm of the likelihood function (i.e., the *loglikelihood*).

ML Intuition

Let's say we have the following $N = 10$ observations.

- We assume these data come from a normal distribution with a known variance of $\sigma^2 = 1$.
- We want to estimate the mean of this distribution, μ.

```
(y <- rnorm(n = 10, mean = 5, sd = 1))

 [1] 5.060983 3.364836 4.968344 6.696222 3.610013
 [6] 6.627266 4.165329 4.615346 4.537332 6.024850
```

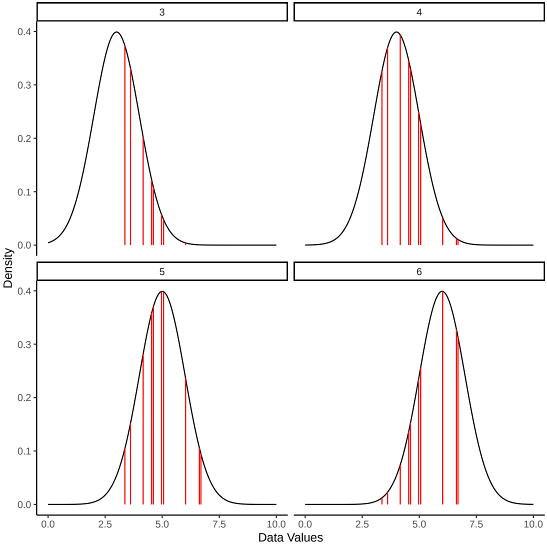
ML Intuition

In ML estimation, we would define different normal distributions.

- Every distribution would have $\sigma^2 = 1$.
- Each distribution would have a different value of μ.

We then compare the observed data to those distributions and see which distribution best fits the data.

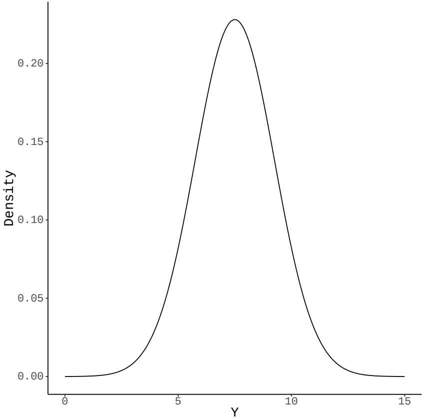
ML Intuition



Likelihoods

Suppose we have the following model:

$$Y \sim \mathrm{N}\left(\mu, \sigma^2\right).$$



Likelihoods

For a given Y_n, we have:

$$P\left(Y_n|\mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(Y_n-\mu)^2}{2\sigma^2}}. \tag{1}$$

If we plug estimated parameters into Equation 1, we get the probability of observing Y_n given $\hat{\mu}$ and $\hat{\sigma}^2$:

$$P\left(Y_n|\hat{\mu}, \hat{\sigma}^2\right) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} e^{-\frac{(Y_n-\hat{\mu})^2}{2\hat{\sigma}^2}}. \tag{2}$$

Applying Equation 2 to all N observations and multiplying the results produces a *likelihood*:

$$\hat{L}\left(\hat{\mu}, \hat{\sigma}^2\right) = \prod_{n=1}^{N} P\left(Y_n|\hat{\mu}, \hat{\sigma}^2\right).$$

Likelihoods

We generally want to work with the natural logarithm of Equation 2. Doing so gives the *loglikelihood*:

$$\begin{aligned}\hat{\mathcal{L}}\left(\hat{\mu}, \hat{\sigma}^2\right) &= \ln \prod_{n=1}^{N} P\left(Y_n \mid \hat{\mu}, \hat{\sigma}^2\right) \\ &= -\frac{N}{2} \ln 2\pi - N \ln \hat{\sigma} - \frac{1}{2\hat{\sigma}^2} \sum_{n=1}^{N} \left(Y_n - \hat{\mu}\right)^2\end{aligned}$$

ML tries to find the values of $\hat{\mu}$ and $\hat{\sigma}^2$ that maximize $\hat{\mathcal{L}}\left(\hat{\mu}, \hat{\sigma}^2\right)$.

- Find the values of $\hat{\mu}$ and $\hat{\sigma}^2$ that are *most likely*, given the observed values of Y.

Likelihoods

Suppose we have a linear regression model:

$$Y = \beta_0 + \beta_1 X + \varepsilon,$$

$$\varepsilon \sim \mathrm{N}\left(0, \sigma^2\right).$$

This model can be equivalently written as:

$$Y \sim \mathrm{N}\left(\beta_0 + \beta_1 X, \sigma^2\right)$$

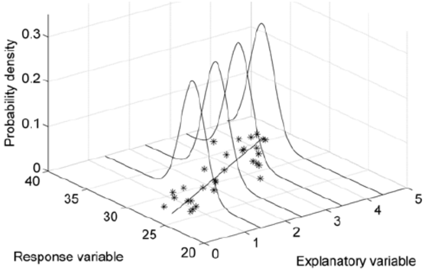


Image retrieved from:
`http://www.seaturtle.org/mtn/archives/mtn122/mtn122p1.shtml`

Likelihoods

For a given $\{Y_n, X_n\}$, we have:

$$P\left(Y_n | X_n, \beta_0, \beta_1, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(Y_n - \beta_0 - \beta_1 X_n)^2}{2\sigma^2}}. \tag{3}$$

If we plug our estimated parameters into Equation 3, we get the probability of observing Y_n given $\hat{Y}_n = \hat{\beta}_0 + \hat{\beta}_1 X_n$ and $\hat{\sigma}^2$.

$$P\left(Y_n | X_n, \hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2\right) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} e^{-\frac{\left(Y_n - \hat{\beta}_0 - \hat{\beta}_1 X_n\right)^2}{2\hat{\sigma}^2}} \tag{4}$$

Likelihoods

So, our final loglikelihood function would be the following:

$$\hat{\mathcal{L}}\left(\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2\right) = \ln \prod_{n=1}^{N} P\left(Y_n | X_n, \hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2\right)$$

$$= -\frac{N}{2} \ln 2\pi - N \ln \hat{\sigma} - \frac{1}{2\hat{\sigma}^2} \sum_{n=1}^{N} \left(Y_n - \hat{\beta}_0 - \hat{\beta}_1 X_n\right)^2 .$$

Example

```r
## Fit a model:
out1 <- lm(ldl ~ bp + glu + bmi, data = diabetes)

## Extract the predicted values and estimated residual standard error:
yHat <- predict(out1)
s    <- summary(out1)$sigma

## Compute the row-wise probabilities:
pY <- dnorm(diabetes$ldl, mean = yHat, sd = s)

## Compute the loglikelihood, and compare to R's version:
sum(log(pY)); logLik(out1)[1]

[1] -2109.939
[1] -2109.93
```

Multivariate Normal Distribution

The PDF for the multivariate normal distribution is:

$$P(\mathbf{Y}|\boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^P |\Sigma|}} e^{-\frac{1}{2}(\mathbf{Y}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{Y}-\boldsymbol{\mu})}.$$

So, the multivariate normal loglikelihood is:

$$\mathcal{L}(\boldsymbol{\mu}, \Sigma) = -\left[\frac{P}{2}\ln(2\pi) + \frac{1}{2}\ln|\Sigma| + \frac{1}{2}\right] \sum_{n=1}^{N} (\mathbf{Y}_n - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{Y}_n - \boldsymbol{\mu}).$$

Which can be further simplified if we multiply through by -2:

$$-2\mathcal{L}(\boldsymbol{\mu}, \Sigma) = \left[P\ln(2\pi) + \ln|\Sigma|\right] \sum_{n=1}^{N} (\mathbf{Y}_n - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{Y}_n - \boldsymbol{\mu}).$$

Steps of ML

1. Choose a probability distribution, $f(Y|\theta)$, to describe the distribution of the data, Y, given the parameters, θ.
2. Choose some estimate of θ, $\hat{\theta}^{(i)}$.
3. Compute each row's contribution to the loglikelihood function by evaluating: $\ln\left[f\left(Y_n|\hat{\theta}^{(i)}\right)\right]$.
4. Sum the individual loglikelihood contributions from Step 3 to find the loglikelihood value, $\hat{\mathcal{L}}$.
5. Choose a "better" estimate of the parameters, $\hat{\theta}^{(i+1)}$, and repeat Steps 3 and 4.
6. Repeat Steps 3 – 5 until the change between $LL^{(i-1)}$ and $LL^{(i)}$ falls below some trivially small threshold.
7. Take $\hat{\theta}^{(i)}$ as your estimated parameters.

MODEL FIT

MODEL EVLUATION