lava REFERENCE CARD

Linear Latent Variable Models in R

MODEL BUILDING

• Initialize model (empty or multivarite regr. model)	`m <- lvm(); m <- lvm(c(y1,y2)~x+z)`
• Initialize from list of regression models	`m <- lvm(list(y~x,y~z,...))`
• Add extra regression associations (slopes)	`regression(m) <- c(y1,y3)~u`
• Add correlation between residual terms	`covariance(m) <- y1~y2+y3`
• Remove associations between variables	`cancel(m) <- ~y1+y2`
• Add variables	`addvar(m) <- ~y1+y2`
• Remove variables	`kill(m) <- ~y1+y2`
• Code as latent (reverse with arg. `clear=TRUE`)	`latent(m) <- ~y1+y2`
• Binary variables; `library(lava.tobit)`	`binary(m) <- ~y1+y2`

EQUALITY CONSTRAINTS

• **Intercepts**	
○ Constrain intercepts to be identical	`intercept(m) <- c(y1,y2)~f(a)`
○ Simultaneously fix several intercepts	`intercept(m,~y1+y2) <- list("a",2)`
• **Variance/covariance parameters**	
○ Fix variance term and covariance between residual to `v1` resp. 1	`covariance(m) <- y1~f(y1,v1)+f(y2,1)`
○ Fix multiple variance parameters	`covariance(m,~y1+y2) <- list("a",2)`
○ Simultaneously fix several covariance parameters	`covariance(m,c(y1,y2)~y2+y3) <- list(2,"a","b",1)`
• **Slope/regression parameters**	
○ $y_1 = x + az + \cdots$ $y_2 = x + bx + \cdots$	`regression(m,c(y1,y2)~x+z) <- list(1,"a",2,"b")`
○ $y_1 = x + \cdots, \quad y_2 = az + \cdots$	`regression(m,c(y1,y2)~x+z) <- list(1,"a")`
○ $y_i = ax + \cdots$	`regression(m) <- c(y1,y2,y3)~f(x,a)`
• Fix parameters defined by index (see `coef`)	`parfix(m,c(3,4,12)) <- list(1,"a",2)`
• Label all free parameters (see `multigroup`)	`m <- baptize(m)`
• Remove linear constraints by fixing to `NA` (applies also to the `intercept` and `covariance` methods)	`regression(m) <- c(y1,y2)~f(x,NA)`
• *Bracket notation.* Define intercept and variance of residual of y to 0 and 'v' and of x to 'a' and 1. And define $\mathbb{E}(y\|x) = b \cdot x$.	`regression(m) <- y[0:v]~f(x[a,1],b)`

SIMULATION

• Simulate 100 observations from model `m`	`sim(m,100,...)`
• Simulate with the slope-parameter of `x` on `y` set to -2, and intercept of y to 1 (see `coef`)	`sim(m,100,p=c("y"=1, "y<-x"=-2),...)`
• Define conditional distribution	`distribution(m,~y1+y2) <- function(n,mu,var,...) ...`
○ Predefined distributions	`binomial.lvm,,uniform.lvm,normal.lvm,poisson.lv`
• Define functional form of predictor on response	`functional(m,y~x) <-` *funct*

NON-LINEAR CONSTRAINTS

• Non-linear parameter constraints	`constrain(m,psi~beta+gamma) <-` *funct*
• Non-linear regression (covariate `x`)	`constrain(m,psi~beta+x) <-` *funct*
• Add extra parameters to model	`parameter(m) <- ~beta+gamma`
• Add predictor/exogenous variable to model	`exogenous(m) <- ~x1+x2`
• Random slopes (`x` name of covariate)	`regression(m) <- y1~f(eta,x)`
• Print non-linear constraints	`constrain(m)`

MODEL INSPECTION

• Examine parameter constraints	`intercept,covariance,regression,constrain`
• Extract variable names	`exogenous,endogenous,latent,manifest,vars`
• Submodel (see also `measurement`)	`subset(m,~y1+y2+eta+x)`
• List parameter names	`coef(m,mean=TRUE,labels=FALSE,...)`
• Parents and children of nodes (union))	`parents(m,~y1+y2);children(m,~x1+x2)`
• Extract (directed) pathways between variables	`path(m,y~x)`

PLOTTING

• Plot method (`lvm` and `lvmfit`)	`plot(m,labels=TRUE,...)`
• Change appearance of nodes	`nodecolor(m,~y1+x,labcol=c("red","blue"), border,lwd=2, ...) <- c("blue","red")`
• Change label and appearance of edges	`edgelabels(m,y~x+z,col,...) <- expression(rho)`
• Change labels of nodes (e.g. math expressions)	`labels(m) <- c(eta=expression(eta))`
• Extract `graphNEL` object (`library(Rgraphviz)`)	`Graph(m)`

STATISTICAL INFERENCE

• Estimate parameters (default MLE)	`e <- estimate(m,data,estimator,...)`
• Estimate multigroup model (default MLE)	`estimate(list(m,...),list(data,...),...)`
• Estimate under MAR assumption	`estimate(m,data,missing=TRUE,...)`
• Likelihood ratio tests vs. saturated model	`compare(e)`
• Likelihood ratio tests	`compare(e1,e2,e3,...)`
• Model indices based on score tests (or Wald tests)	`modelsearch(e,...)`
• Identify empirical equivalent models	`equivalence(e,y~x,k=1,...)`
• Calculate indirect and total effects of `x` on `y`	`effects(e, y~x)`
• Non-linear constraints and approx. std.errors	`constraints(e)`
• Mixtures of LLVM; `library(lava.mixture)`	`mixture(list(m1,m2),data)`
• Extract various likelihood summaries	`coef,score,information,logLik,AIC,gof`
• Clustered correlated data	`estimate(m,data,cluster="id",...)`
• Robust standard errors	`coef(e,type="robust")`
• Test for linearity; `library(gof)`	`cumres(e,...)`
• Non-parametric bootstrap	`bootstrap(e,R=100,...)`
• Likelihood-based confidence limits	`confint(e,idx,profile=TRUE,...)`

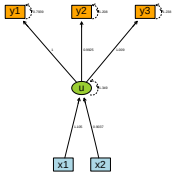
EXAMPLES

Linear Latent Variable Models in R

STRUCTURAL EQUATION MODEL

MIMIC model

```
> m <- lvm(c(y1,y2,y3)~u)
> regression(m) <- u ~ x1+x2
> latent(m) <- ~ u
> d <- sim(m,100)
> e <- estimate(m,d)
> plot(e)
```



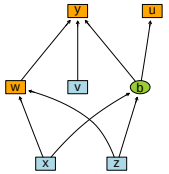
RANDOM REGRESSION

Random slopes allowing for unbalanced designs

```
> m <- lvm(c(y1,y2,y3)~f(eta,1))
> regression(m,c(y1,y2,y3)~u) <- list("x1","x2","x3")
> intercept(m,~y1+y2+y3) <- list("mu")
> covariance(m,~y1+y2+y3) <- list("v","v","v")
> latent(m) <- ~u+eta
> estimate(m,data,missing=TRUE)
```

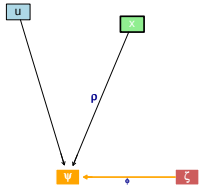
GRAPHICS I

```
> m <- lvm(list(y ~ b+v+w, c(b,w) ~ x+z, u ~ b))
> latent(m) <- ~b
> plot(m)
```



GRAPHICS II

```
> m <- lvm(y~x+z+u)
> labels(m) <- c(y=expression(psi), z=expression(zeta))
> nodecolor(m,~y+z+x,border=c("white","white","black"),
+   labcol="white", lwd=c(0,0,5)) <-
+   c("orange","indianred","lightgreen")
> edgelabels(m,y~z+x, cex=c(2,3), col=c("orange","black"),
+   labcol="darkblue",lwd=c(3,1)) <- expression(phi,rho)
> plot(m,layoutType="circo")
```



INSTRUMENTAL VARIABLE

IV estimator (not available with e.g. non-recursive structures)

```
> estimate(m,data,estimator="IV")
```

SIMULATION

Weibull with exponential distributed censoring

```
> m <- lvm(y~x1+x2+x3)
> distribution(m,~y) <- weibull.lvm(shape=0.5,cens=rexp)
> distribution(m,~x3) <- binomial.lvm()
> d <- sim(m,100)
```

MULTIVARIATE PROBIT

```
> m <- lvm(c(y1,y2)~f(x,b)+f(z,b))
> binary(m) <- ~y1+y2
> covariance(m) <- y1~y2
> estimate(m,data,control=list(trace=1))
```

MULTIGROUP ANALYSIS I

$$\log L(\theta|d) = \sum_i \log L_i(\theta|d_i)$$

```
> estimate(list(m1,m2,m3),list(d1,d2,d3))
```

INDIRECT EFFECTS. TOBIT/PROBIT MODEL

$$\mathbb{E}(y \mid x, z) = a(x + z)$$

```
> m <- lvm(list(y~z+x,z~x))
> d <- transform(sim(m,100),z=factor(z>0),y=Surv(ifelse(y<1,y,1),y<1))
> e <- estimate(m,d)
> effects(e,y~x)
```

NON-LINEAR REGRESSION

Bi-variate non-linear regression with random intercept. Estimated via Fischer scoring

```
> m0 <- lvm(c(y1,y2)~f(x,0)+f(eta,1))
> latent(m0) <- ~eta
> covariance(m0) <- c(y1,y2) ~ f(0.01)
> covariance(m0) <- c(eta) ~ f(0.01)
> d <- sim(m0,50)[,manifest(m0)]
> d <- transform(d,
                 y1=y1+pnorm(2*x),
                 y2=y2+pnorm(2*x))
> m <- m0
> parameter(m) <- ~ nu+alpha+xi
> intercept(m) <- c(y1,y2) ~ f(mu)
> covariance(m) <- c(y1,y2) ~ f(v)
> covariance(m) <- eta ~ f(zeta)
> constrain(m, mu ~ x + alpha + nu + xi) <- function(x) pnorm(x[1]*x[2]+x[4]) + x[3]
> e <- estimate(m,d,control=list(trace=1,method="NR",gamma=0.99))
```