



Multivariate Ordinal Analysis

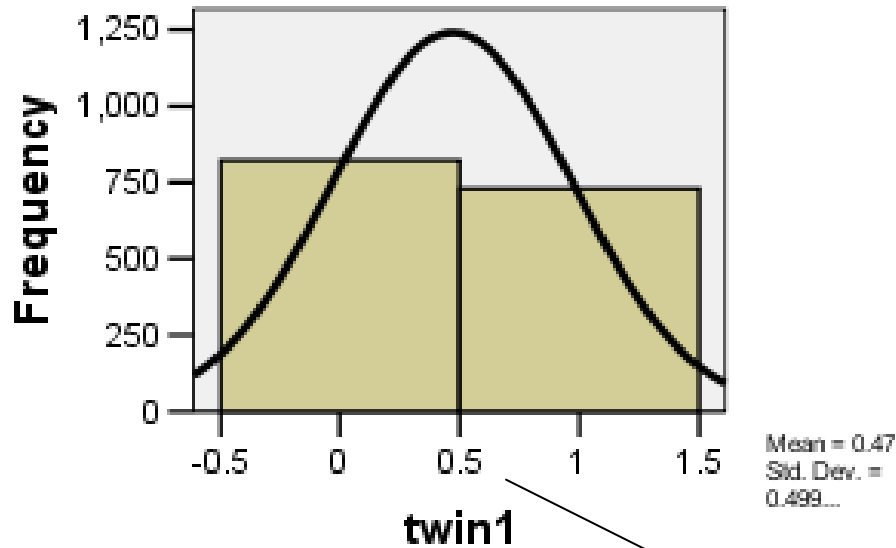
Sarah Medland – Boulder 2010



Today

- Recap thresholds
- MV analysis when all Vs have the same number of thresholds
- MV analysis when all Vs have different number of thresholds

Histogram



Standard normal distribution

Mean = 0

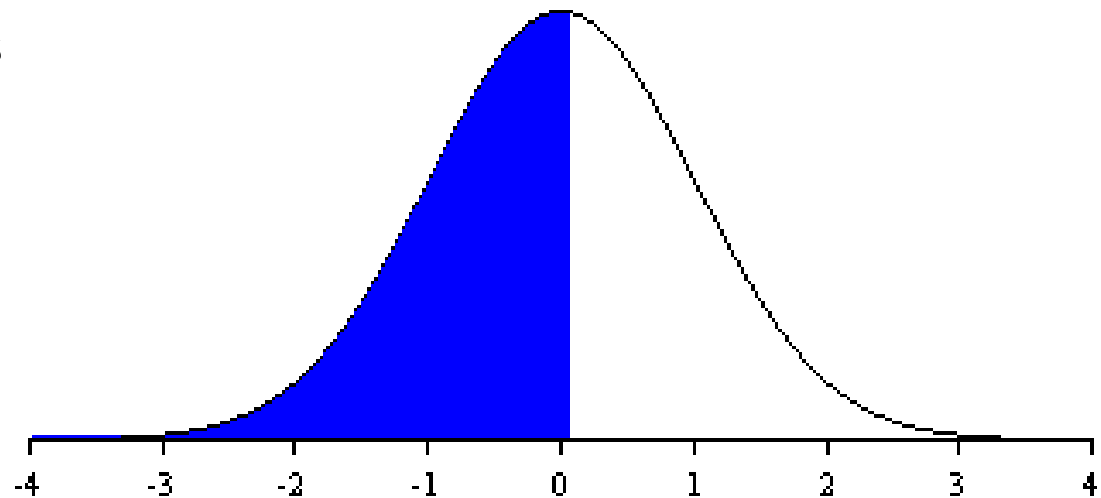
SD = 1

Non Smokers = 53%

Threshold = .074

The observed phenotype is an *imperfect* measurement of an underlying continuous distribution

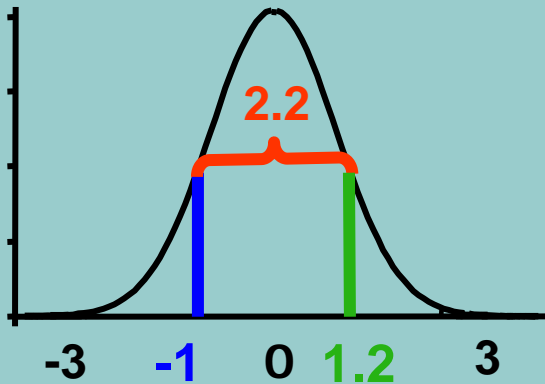
ie Obesity vs BMI
MDD vs quantitative depression scales





MV analysis when all Vs have
the same number of thresholds

Mx Threshold Specification: 3+ Cat.



$$\mathbf{T} = \begin{array}{c} \text{Twin 1} \quad \text{Twin 2} \\ \left[\begin{array}{cc} t_{11} & t_{12} \\ t_{21} & t_{22} \end{array} \right] \end{array}$$

1st threshold increment

Mx Threshold Model: Thresholds $L * T$ /

$$\begin{aligned} L * T &= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \\ &= \begin{bmatrix} t_{11} & t_{12} \\ t_{11} + t_{21} & t_{12} + t_{22} \end{bmatrix} \end{aligned}$$

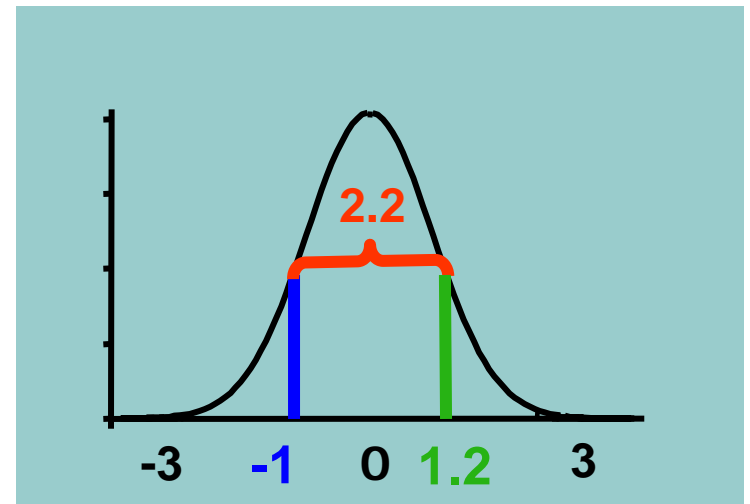
OpenMx code

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} = \begin{bmatrix} \text{zscore1 twin1} & \text{zscore1 twin2} \\ \text{zscore2 twin1} - \text{zscore1 twin1} & \text{zscore2 twin2} - \text{zscore1 twin2} \end{bmatrix}$$

```
mxMatrix( type="Full", nrow=2, ncol=2, free=TRUE,  
values=.5, lbound= c(-3,0.0001, -3, 0.0001), name="T" ),
```

The bounds stop the thresholds going 'backwards'

ie they preserve the ordering of the data



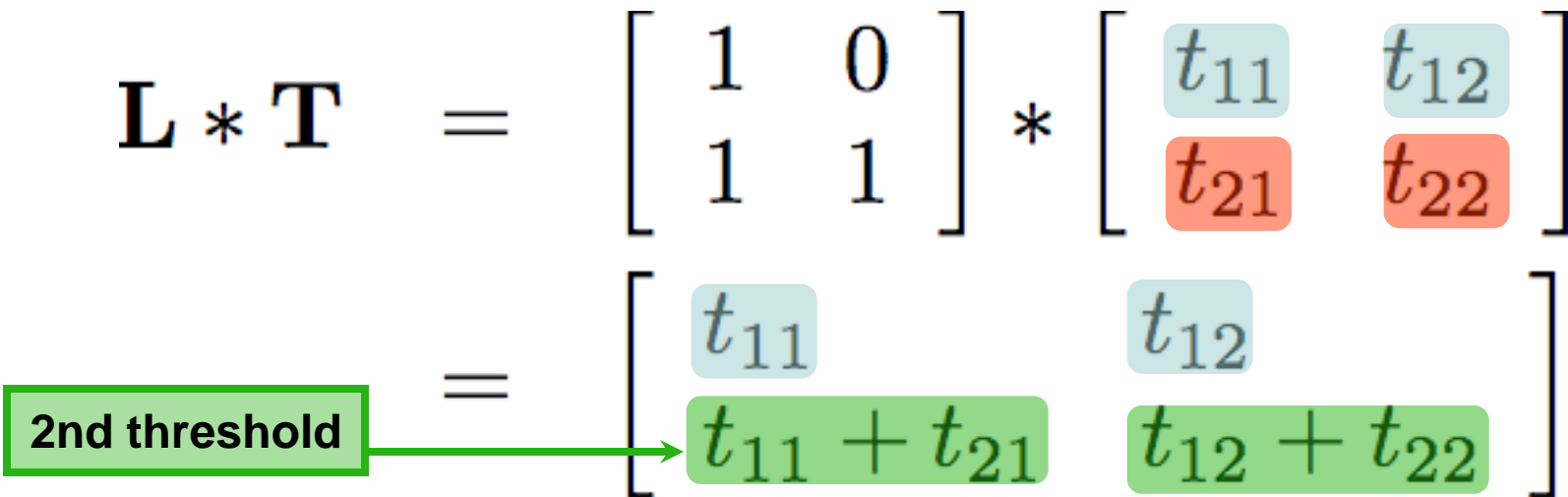
OpenMx code

```
mxMatrix( type="Lower", nrow=2, ncol=2, free=FALSE,  
values=1, name="L" ),
```

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

OpenMx code

```
mxAlgebra( expression= L %*% T, dimnames=list(thRows, selVars)  
, name="expThmz"),
```

$$\mathbf{L} * \mathbf{T} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$$
$$= \begin{bmatrix} t_{11} & t_{12} \\ t_{11} + t_{21} & t_{12} + t_{22} \end{bmatrix}$$


2nd threshold

OpenMx code

```
mxMatrix( type="Full", nrow=2, ncol=2, free=TRUE,  
values=.5, lbound= c(-3,0.0001, -3, 0.0001), name="T" ),
```

```
mxMatrix( type="Lower", nrow=2, ncol=2, free=FALSE,  
values=1, name="L" ),
```

```
mxAlgebra( expression= L %*% T, dimnames=list(thRows,selVars)  
, name="expThmz"),
```

$$\mathbf{L} * \mathbf{T} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$$
$$= \begin{bmatrix} t_{11} & t_{12} \\ t_{11} + t_{21} & t_{12} + t_{22} \end{bmatrix}$$

2nd threshold →

SAME_THRESHOrdinalMultivariate.r

```
FamData <- read.table ("DHBQ_bs.dat", header=T, na=-999)

OFamData <-data.frame(FamData$zyg2, FamData$gff1, FamData$gff2, FamData$hap1, FamData$hap2)
names(OFamData) <- c("zyg2", "gff1", "gff2", "hap1", "hap2")

head(OFamData)
OFamData$Ogff1 <- as.numeric(cut(OFamData$gff1, breaks=c(0, 38, 60), labels=c(0,1)))
OFamData$Ogff2 <- as.numeric(cut(OFamData$gff2, breaks=c(0, 38, 60), labels=c(0,1)))
OFamData$Ohap1 <- as.numeric(cut(OFamData$hap1, breaks=c(0, 22, 60), labels=c(0,1)))
OFamData$Ohap2 <- as.numeric(cut(OFamData$hap2, breaks=c(0, 22, 60), labels=c(0,1)))

head(OFamData)
table(OFamData$Ogff1)
table(OFamData$Ogff2)
table(OFamData$Ohap1)
table(OFamData$Ohap2)
```

SAME_THRESHOrdinalMultivariate.r

```
Vars <- c('Ogff', 'Ohap')
nv <- 2
maxthresh <- 1
selVars <- paste(Vars, c(rep(1, nv), rep(2, nv)), sep="")
thresh <- c("Th1")
ntv <- nv*2
mzData <- subset(OFamData, zyg2==1, selVars)
dzData <- subset(OFamData, zyg2==2, selVars)
```

SAME_THRESHOrdinalMultivariate.r

```
# Fit Multivariate Saturated Model
# -----
multTwinSatModel <- mxModel("multTwinSat",
  mxModel("MZ",
    mxMatrix( type="Stand", nrow=ntv, ncol=ntv, free=TRUE, values=.3, name="expCovMZ" ),
    mxMatrix( type="Full", nrow=1, ncol=ntv, free=FALSE, values=0, name="expMeanMZ" ),

    mxMatrix( type="Full", nrow=maxthresh, ncol=ntv, free=TRUE, values=c(0.1,-.3,0.1,-.3),
      lbound=-4, name="cutpoints" ),
    mxMatrix( type="Lower", nrow=maxthresh, ncol=maxthresh, free=FALSE, values=1, name="Increment" ),
    mxAlgebra( expression= (Increment %*% cutpoints), dimnames=list(thresh,selVars), name="expThresh" ),

    mxData( observed=mzData, type="raw" ),
    mxFIMLObjective( covariance="expCovMZ", means="expMeanMZ", dimnames=selVars, thresholds="expThresh" )
  ),
  mxAlgebra( MZ.objective, name="-2sumll" ),
  mxAlgebraObjective("-2sumll")
)

multTwinSatFit <- mxRun(multTwinSatModel)
multTwinSatSumm <- summary(multTwinSatFit)
multTwinSatSumm
```

SAME_THRESHOrdinalMultivariate.r

```
> multTwinSatFit$MZ@matrices
$expCovMZ
StandMatrix 'expCovMZ'

@labels: No labels assigned.

@values
      Ogff1      Ohap1      Ogff2      Ohap2
Ogff1 1.0000000 0.4305299 0.5602075 0.3110542
Ohap1 0.4305299 1.0000000 0.3298978 0.4933827
Ogff2 0.5602075 0.3298978 1.0000000 0.4200501
Ohap2 0.3110542 0.4933827 0.4200501 1.0000000
```

SAME_THRESHOrdinalMultivariate.r

```
# Fit Multivariate ACE Model using Cholesky Decomposition
# -----
multACEModel <- mxModel("multACE",
  mxModel("ACE",
    # Matrices a, c, and e to store a, c, and e path coefficients
    mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=.6,label=c("a11", "a21", "a22"), name="a" ),
    mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=.6,label=c("c11", "c21", "c22"), name="c" ),
    mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=.6,label=c("e11", "e21", "e22"), name="e" ),
    # Matrices A, C, and E compute variance components
    mxAlgebra( expression=a %*% t(a), name="A" ),
    mxAlgebra( expression=c %*% t(c), name="C" ),
    mxAlgebra( expression=e %*% t(e), name="E" ),
    # Algebra to compute total variances and standard deviations (diagonal only)
    mxAlgebra( expression=A+C+E, name="V" ),
    mxMatrix( type="Iden", nrow=nv, ncol=nv, name="I"),
    # Constraint on variance of ordinal variables
    mxConstraint( alg1="V", "=", alg2="I", name="ConstrainedV"),
```

SAME_THRESHOrdinalMultivariate.r

```
## Note that the rest of the mxModel statements do not change for bivariate/multivariate case
# Matrix & Algebra for expected means vector
  mxMatrix( type="Full", nrow=1, ncol=ntv, free=FALSE, values= 0, name="expMean"),
# Matrix & Algebra for expected thresholds vector
  mxMatrix(type="Full", nrow=maxthresh, ncol=ntv, free=TRUE,
    values=c(0.1,-.3,0.1,-.3), lbound=-4,name="cutpoints"),
  mxMatrix( type="Lower", nrow=maxthresh, ncol=maxthresh, free=FALSE,
    values=1, name="Increment" ),
  mxAlgebra( expression= (Increment %*% cutpoints), dimnames=list(thresh,selVars),
    name="expThresh"),
```

$$\mathbf{L} * \mathbf{T} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$$
$$= \begin{bmatrix} t_{11} & t_{12} \\ t_{11} + t_{21} & t_{12} + t_{22} \end{bmatrix}$$

SAME_THRESHOrdinalMultivariate.r

```
# Algebra for expected variance/covariance matrix in MZ
  mxAlgebra( expression= rbind ( cbind(A+C+E , A+C),
                                cbind(A+C , A+C+E)), name="expCovMZ" ),
# Algebra for expected variance/covariance matrix in DZ, note use of 0.5, converted to 1*1 m
  mxAlgebra( expression= rbind ( cbind(A+C+E , 0.5*x%A+C),
                                cbind(0.5*x%A+C , A+C+E)), name="expCovDZ" )

),

mxModel("MZ",
  mxData( observed=mzData, type="raw" ),
  mxFIMLObjective( covariance="ACE.expCovMZ", means="ACE.expMean", dimnames=names(mzData),
                    thresholds="ACE.expThresh")
),
mxModel("DZ",
  mxData( observed=dzData, type="raw" ),
  mxFIMLObjective( covariance="ACE.expCovDZ", means="ACE.expMean", dimnames=names(mzData),
                    thresholds="ACE.expThresh")
),
mxAlgebra( expression=MZ.objective + DZ.objective, name="-2sumll" ),
mxAlgebraObjective("-2sumll")
)
multACEFit <- mxRun(multACEModel)
multACESumm <- summary(multACEFit)
multACESumm
```


SAME_THRESHOrdinalMultivariate.r

```
$A
mxAlgebra 'A'
@formula: a %*% t(a)
@result:
           [,1]      [,2]
[1,] 0.32466579 0.01959946
[2,] 0.01959946 0.45095750
dimnames: NULL
```

```
$C
mxAlgebra 'C'
@formula: c %*% t(c)
@result:
           [,1]      [,2]
[1,] 0.21606832 0.04997648
[2,] 0.04997648 0.01155965
dimnames: NULL
```

```
$E
mxAlgebra 'E'
@formula: e %*% t(e)
@result:
           [,1]      [,2]
[1,] 0.45926590 -0.06957594
[2,] -0.06957594 0.53748367
dimnames: NULL
```

```
$V
mxAlgebra 'V'
@formula: A + C + E
@result:
           [,1]      [,2]
[1,] 1.000000e+00 -2.656830e-09
[2,] -2.65683e-09 1.000001e+00
dimnames: NULL
```

```
> multTwinSatSumm
  name      matrix  row  col  Estimate
1 <NA>  MZ.expCovMZ  Ogff1 Ohap1  0.43052990
2 <NA>  MZ.expCovMZ  Ogff1 Ogff2  0.56020748
3 <NA>  MZ.expCovMZ  Ohap1 Ogff2  0.32989782
4 <NA>  MZ.expCovMZ  Ogff1 Ohap2  0.31105425
5 <NA>  MZ.expCovMZ  Ohap1 Ohap2  0.49338270
6 <NA>  MZ.expCovMZ  Ogff2 Ohap2  0.42005014
7 <NA>  MZ.cutpoints    1    1 -0.06566605
8 <NA>  MZ.cutpoints    1    2 -0.49270273
9 <NA>  MZ.cutpoints    1    3 -0.06243746
10 <NA> MZ.cutpoints    1    4 -0.54130749
```



MV analysis when Vs have different numbers of thresholds

DIFF_THRESHOrdinalMultivariate.r

- THIS SCRIPT IS NOT GENERALISABLE
- TO CHANGE THE NUMBER OF VARIABLES OR THRESHOLDS YOU WILL NEED TO REWRITE THE MATRIX AND ALGEBRA STATEMENTS!!!!!!

DIFF_THRESHOrdinalMultivariate.r

```
FamData <- read.table ("DHBQ_bs.dat", header=T, na=-999)

OFamData <-data.frame(FamData$zyg2, FamData$gff1, FamData$gff2, FamData$hap1, FamData$hap2)
names(OFamData) <- c("zyg2", "gff1", "gff2", "hap1", "hap2")

head(OFamData)
OFamData$Ogff1 <- as.numeric(cut(OFamData$gff1, breaks=c(0, 38, 60), labels=c(0,1)))
OFamData$Ogff2 <- as.numeric(cut(OFamData$gff2, breaks=c(0, 38, 60), labels=c(0,1)))
OFamData$Ohap1 <- as.numeric(cut(OFamData$hap1, breaks=c(0, 20, 26, 60), labels=c(0,1,2)))
OFamData$Ohap2 <- as.numeric(cut(OFamData$hap2, breaks=c(0, 20, 26, 60), labels=c(0,1,2)))

head(OFamData)
table(OFamData$Ogff1)
table(OFamData$Ogff2)
table(OFamData$Ohap1)
table(OFamData$Ohap2)

Vars <- c('Ogff', 'Ohap')
nv <- 2
maxthresh <- 2
selVars <- paste(Vars,c(rep(1,nv),rep(2,nv)),sep="") #c('gff1', 'hap1','gff2' 'hap2')
thresh <- c("Th1","Th2")
ntv <- nv*2
mzData <- subset(OFamData, zyg2==1, selVars)
dzData <- subset(OFamData, zyg2==2, selVars)
```

DIFF_THRESHOrdinalMultivariate.r

```
# Fit Multivariate Saturated Model
# -----
multTwinSatModel <- mxModel("multTwinSat",
  mxModel("MZ",
    mxMatrix( type="Stand", nrow=ntv, ncol=ntv, free=TRUE, values=.5, name="expCovMZ" ),
    mxMatrix( type="Full", nrow=1, ncol=ntv, free=FALSE, values=0, name="expMeanMZ" ),

    mxMatrix( type="Full", nrow=maxthresh, ncol=1, free=c(T,F), values=c(.1,NA), name="V1T" ),
    mxMatrix( type="Full", nrow=maxthresh, ncol=1, free=TRUE, values=c(-.3,.6),
      lbound=c(-4,0.001), name="V2T" ),
    mxMatrix( type="Lower", nrow=maxthresh, ncol=maxthresh, free=FALSE, values=1, name="Increment" ),
    mxAlgebra( expression= cbind(V1T,(Increment %*% V2T),V1T,(Increment %*% V2T)),
      dimnames=list(thresh,selVars), name="expThresh" ),

    mxData( observed=mzData, type="raw" ),
    mxFIMLObjective( covariance="expCovMZ", means="expMeanMZ", dimnames=selVars, thresholds="expThresh" )
  ),
  mxAlgebra( MZ.objective, name="-2sumll" ),
  mxAlgebraObjective( "-2sumll" )
)

multTwinSatFit <- mxRun(multTwinSatModel)
multTwinSatSumm <- summary(multTwinSatFit)
multTwinSatSumm
```

DIFF_THRESHOrdinalMultivariate.r

○ Huh what?

```
mxMatrix(type="Full", nrow=maxthresh, ncol=1, free=c(T,F),  
         values=c(.1,NA), name="V1T"),
```

```
mxMatrix(type="Full", nrow=maxthresh, ncol=1, free=TRUE, values=c(-.3,.6),  
         lbound=c(-4,0.001), name="V2T"),
```

```
mxMatrix( type="Lower", nrow=maxthresh, ncol=maxthresh, free=FALSE,  
         values=1, name="Increment" ),
```

$$V1T = \begin{bmatrix} .1 \\ NA \end{bmatrix}$$

$$V2T = \begin{bmatrix} -.3 \\ .6 \end{bmatrix} \text{ with lower bounds } \begin{bmatrix} -4 \\ .001 \end{bmatrix}$$

$$\text{Increment} = \begin{bmatrix} 1 & \\ 1 & 1 \end{bmatrix}$$

DIFF_THRESHOrdinalMultivariate.r

```
mxAlgebra( expression= cbind(V1T,(Increment %*% V2T),V1T,(Increment %*% V2T)),  
           dimnames=list(thresh,selVars), name="expThresh"),
```

$$V1T = \begin{bmatrix} .1 \\ NA \end{bmatrix}$$

$$(\text{Increment \% \% } V2T) = \begin{bmatrix} 1 & \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} -.3 \\ .6 \end{bmatrix}$$

$$V2T = \begin{bmatrix} -.3 \\ .6 \end{bmatrix} \text{ with lower bounds } \begin{bmatrix} -4 \\ .001 \end{bmatrix}$$

$$= \begin{bmatrix} -.3 \\ -.3 + .6 \end{bmatrix}$$

$$\text{Increment} = \begin{bmatrix} 1 & \\ 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -.3 \\ .3 \end{bmatrix}$$

$$\text{cbind}(V1T,(\text{Increment \% \% } V2T),V1T,(\text{Increment \% \% } V2T)) = \begin{bmatrix} .1 & & -.3 & & .1 & & -.3 \\ NA & & .3 & & NA & & .3 \end{bmatrix}$$

DIFF_THRESHOrdinalMultivariate.r

○ Wow...

`cbind(V1T,(Increment % ** V2T), V1T,(Increment % ** V2T)) =` $\begin{bmatrix} .1 & \begin{bmatrix} -.3 \\ .3 \end{bmatrix} \\ NA & \begin{bmatrix} -.3 \\ .3 \end{bmatrix} \end{bmatrix}$

Vs

$$\mathbf{L} * \mathbf{T} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$$
$$= \begin{bmatrix} t_{11} & t_{12} \\ t_{11} + t_{21} & t_{12} + t_{22} \end{bmatrix}$$

with missing thresholds fixed

DIFF_THRESHOLDOrdinalMultivariate.r

- implications...

`cbind(V1T,(Increment % *% V2T), V1T,(Increment % *% V2T)) =` $\begin{bmatrix} .1 & \begin{bmatrix} -.3 \\ .3 \end{bmatrix} \\ NA & \begin{bmatrix} -.3 \\ .3 \end{bmatrix} \end{bmatrix}$

- The code is not generalisable
- TO CHANGE THE NUMBER OF VARIABLES OR THRESHOLDS YOU WILL NEED TO REWRITE THE MATRIX AND ALGEBRA STATEMENTS!!!!!!
- Addition of covariate effects is painfull

DIFF_THRESHOLDOrdinalMultivariate.r

Hmm ...

`cbind(V1T,(Increment % *% V2T), V1T,(Increment % *% V2T)) =` $\begin{bmatrix} .1 & \begin{bmatrix} -.3 \\ .3 \end{bmatrix} \\ NA & \begin{bmatrix} -.3 \\ .3 \end{bmatrix} \end{bmatrix}$

- This is temporary solution
- Hopefully get this fixed soon



A couple of words about Cholesky

André-Louis Cholesky

Note sur une methode de resolution des equation normales provenant de l'application de la methode des moindres carrés a un systeme d'equations lineaires en nombre inferieure a celui des inconnues. Application de la methode a la resolution d'un systeme defini d'equations lineaires (Procédé du Commandant Cholesky), published in the Bulletin geodesique in 1924.



André-Louis Cholesky (October 15, 1875 – August 31, 1918) was a [French](#) military officer and [mathematician](#), born in [Montguyon, France](#).

His paternal family was descendant from the Cholewski family who emigrated from Poland at the beginning of XIX century. He was educated at the [Ecole polytechnique](#). He worked in [geodesy](#) and map-making, was involved in surveying in Crete and North Africa before World War I. But he is primarily remembered for the development of a matrix decomposition known as the [Cholesky decomposition](#) which he used in his surveying work. He served the French military as engineer officer and was killed in battle a few months before the end of [World War I](#); his discovery was published posthumously by his fellow officer in the "Bulletin Geodesique".



Cholesky Decomposition

- Ambiguous terminology
 - Problematic in writing up results
- Not a model
 - Not a structural model
 - Data summary
 - However, the ACE Cholesky is a genetic model
- Solution
 - Always refer to it as the Cholesky decomposition NOT as the Cholesky model



Cholesky Decomposition

- The Factors within a Cholesky decomposition are orthogonal
- Implications of having N var genetic factors
 - Suggesting at a theoretical level that there are N var sets of genes influencing the traits
 - More realistic to think of the genetic influence on a factor as the sum of many small effects that act in similar fashions than a large single gene effect



Cholesky Decomposition

- Implications of having N var environmental factors
 - Suggesting at a theoretical level that there are N var sets of non-genetic effects influencing the traits
 - May be large single source effects
 - Day/Time of testing effects
 - Major Disaster
 - May be the sum of multiple small effects
 - cumulative consequences of growing up in an abusive family situation



Cholesky Decomposition

- Theoretically we can test if these influences are due to our favourite genetic or environmental 'candidate' effects by
 - sequentially including the effects as covariates in the model
 - examining the change in the factor structure (or lack there of)
 - Proportion of variance explained by the candidate gene effect