More on thresholds

Sarah Medland

A plug for OpenMx?

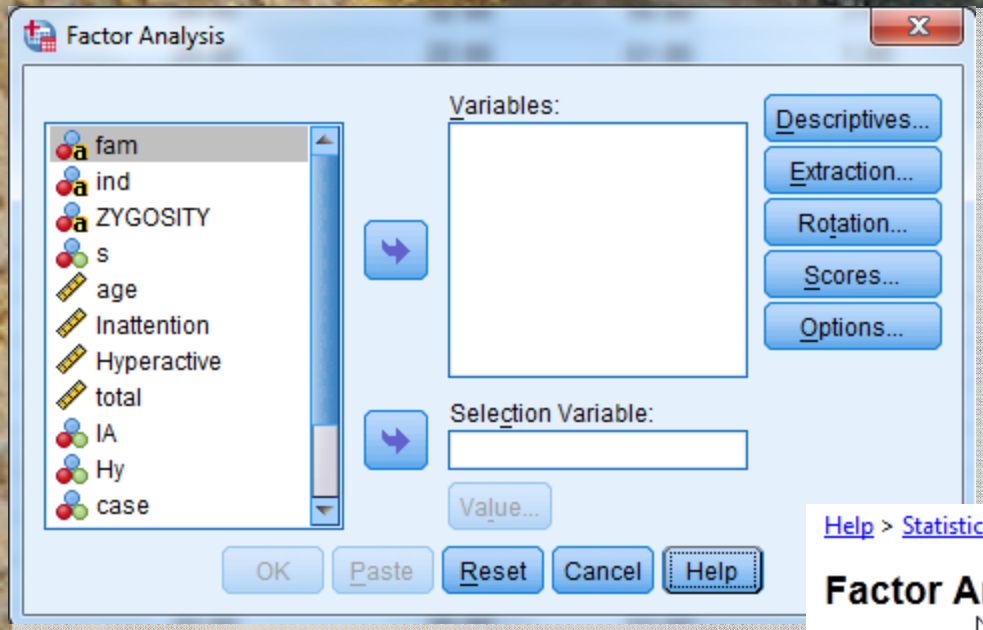
- Very few packages can handle ordinal data adequately…
- OpenMx can also be used for more than just genetic analyses
  - Regression
  - Polycohoric correlations
  - Factor analysis...

Names for different types of thresholds

Table 2.2: Classification of correlations according to their observed distribution.

Measurement	Two Categories	Three or more Categories	Continuous
Two	Tetrachoric	Polychoric	Biserial
Three or more	Polychoric	Polychoric	Polyserial
Continuous	Biserial	Polyserial	Product Moment

- http://ibgwww.colorado.edu/workshop2004/cdrom/HTML/book2004a.pdf



Help > Statistics Base Option

Factor Analysis

Previous Next

Factor analysis attempts to identify underlying variables, or **factors**, that explain the pattern of correlations within a set of observed variables. Factor analysis is often used in data reduction to identify a small number of factors that explain most of the variance that is observed in a much larger number of manifest variables. Factor analysis can also be used to generate hypotheses regarding causal mechanisms or to screen variables for subsequent analysis (for example, to identify collinearity prior to performing a linear regression analysis).

Data. The variables should be quantitative at the **interval** or **ratio** level. Categorical data (such as religion or country of origin) are not suitable for factor analysis. Data for which Pearson correlation coefficients can sensibly be calculated should be suitable for factor analysis.

Two approaches to the liability threshold model

- Problem
  - Ordinal data has 1 less degree of freedom
    - MZcov, DZcov, Prevalence
    - No information on the variance
  - Thinking about our ACE/ADE model
    - 4 parameters being estimated
    - A C E mean
  - ACE/ADE model is unidentified without adding a constraint

Two approaches to the liability threshold model with binary data

- Solution?
- Traditional
  - Maps data to a standard normal distribution
  - Total variance constrained to be 1
- Alternate
  - Fixes an alternate parameter (usually E)
  - Estimates the remaining parameters

Traditional Approach

- Imagine we have a set of binary data
- Trait – lifetime cannabis use
  - Never Smoked/Ever Smoked

Zyg	twin1	twin2	Age	Sex
1	0	0	25.80	1
1	0	0	21.10	1
1	0	0	21.79	1
1	0	0	21.12	1
1	0	0	32.05	1
1	0	0	37.41	1
1	0	0	33.56	0

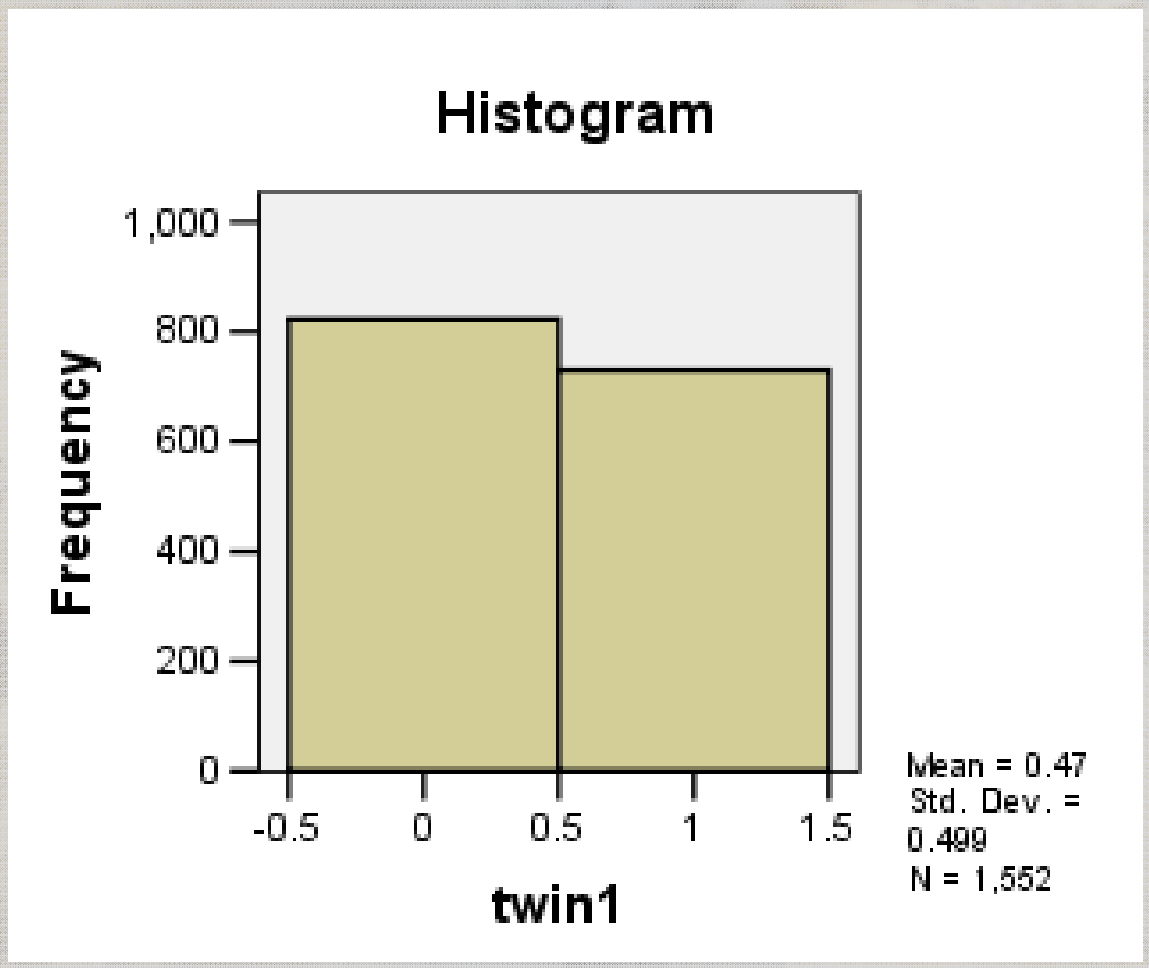
Twin 1 cannabis use

- 0 = never used

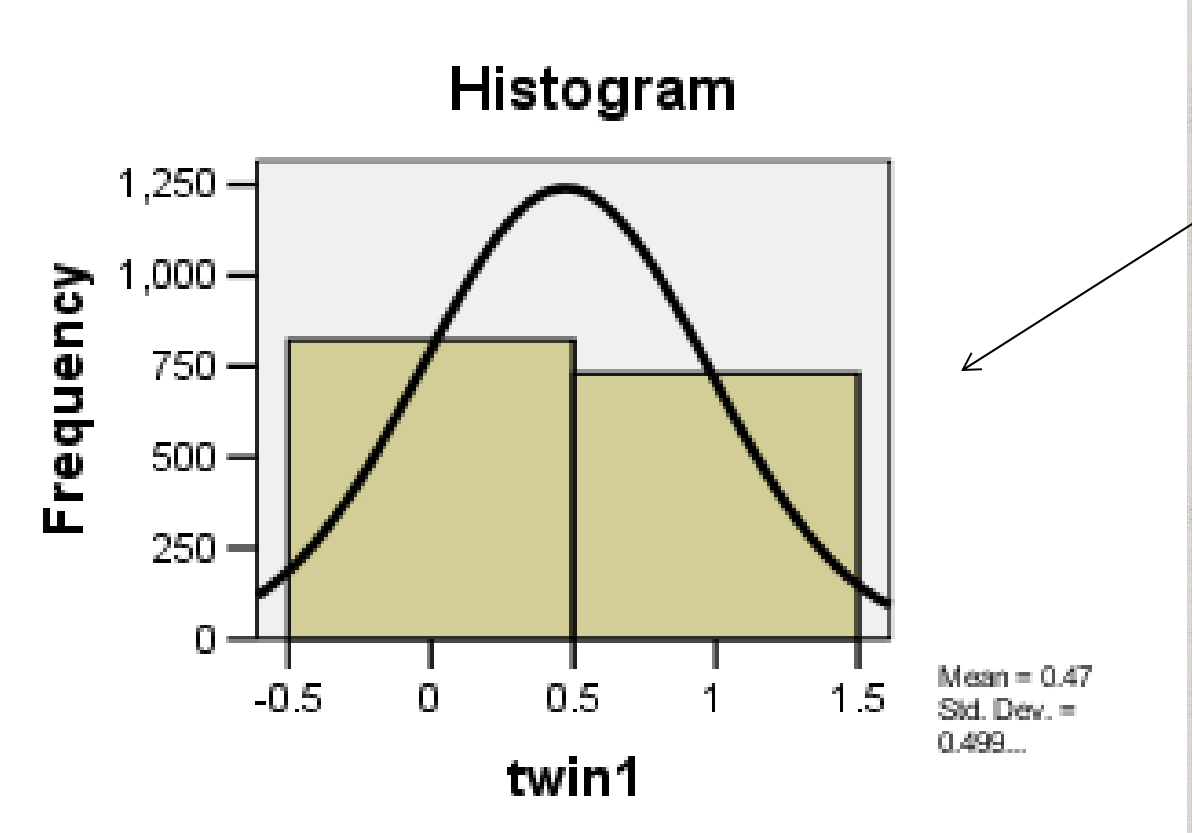
twin1

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0	822	47.5	53.0	53.0
	1	730	42.2	47.0	100.0
	Total	1552	89.7	100.0	
Missing	System	179	10.3		
Total		1731	100.0		

Twin 1 cannabis use

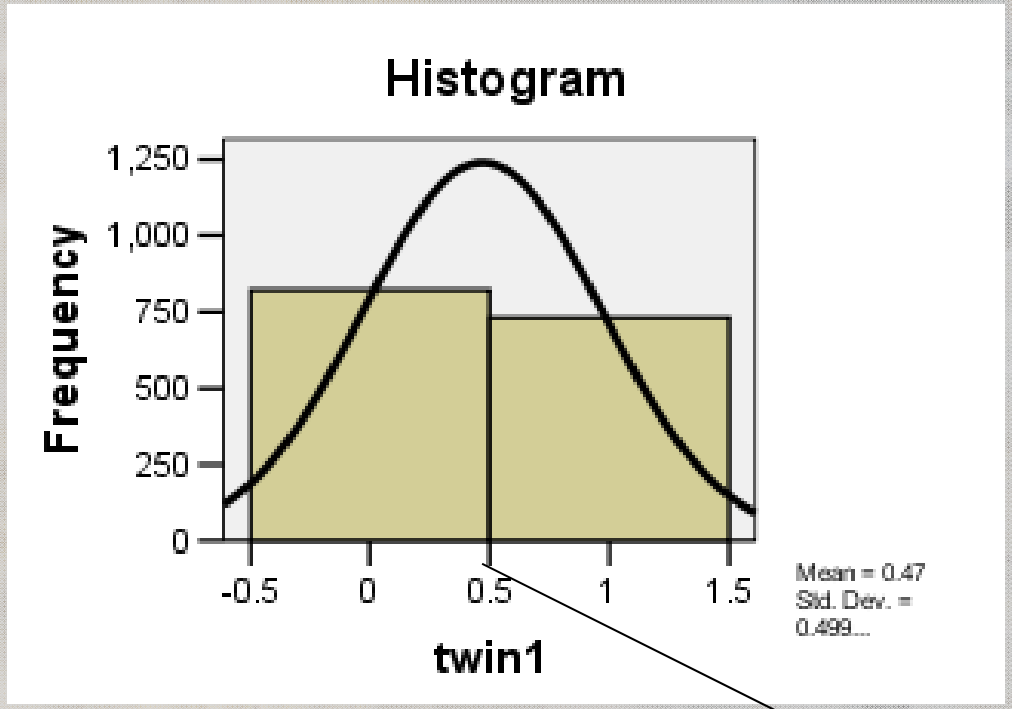


Twin 1 cannabis use



Liability or 'risk' of initiation distribution

Just because an individual has never used cannabis does not mean their 'risk' of initiation is zero



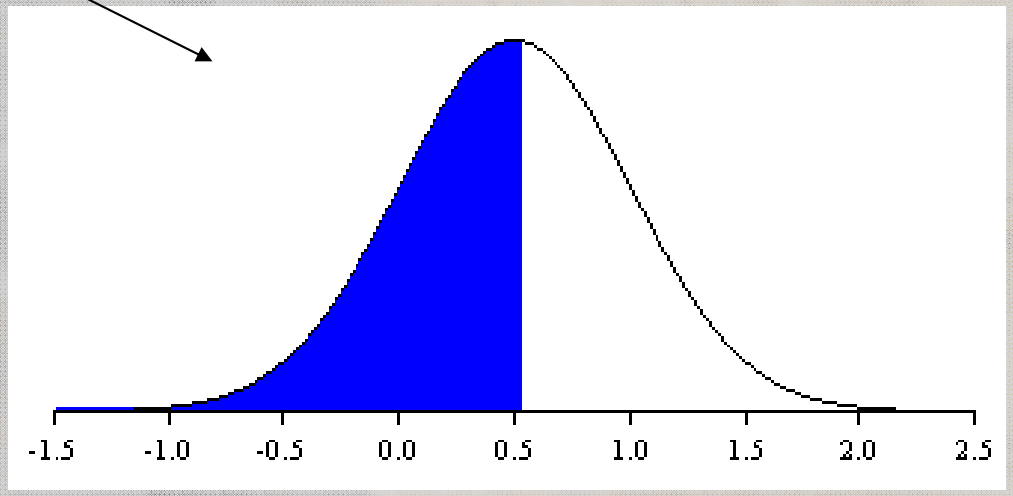
Mean = .47

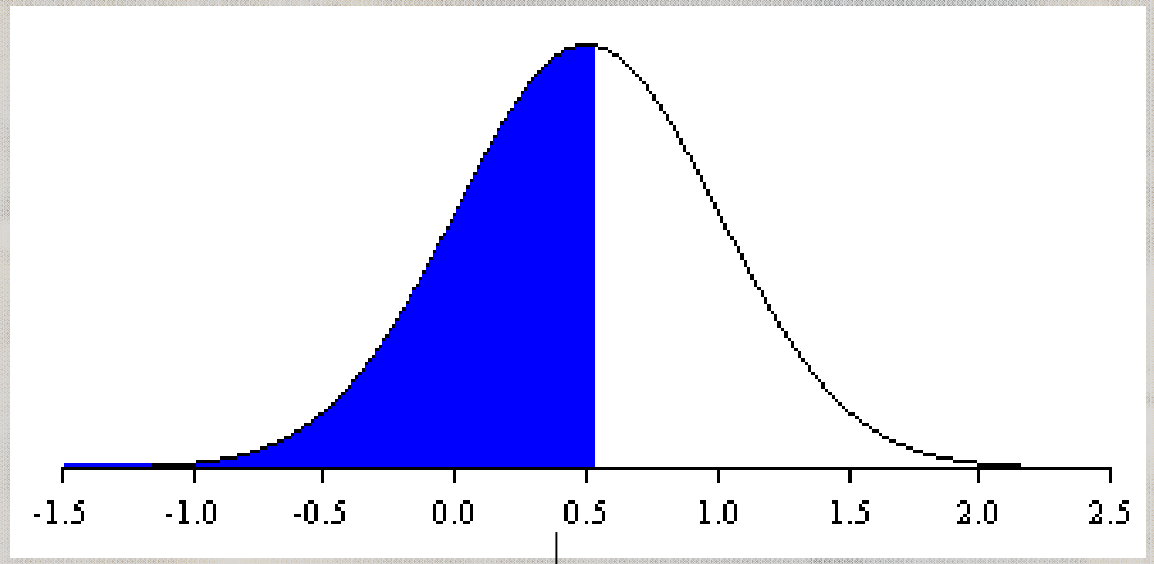
SD =.499

Non Smokers =53%

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

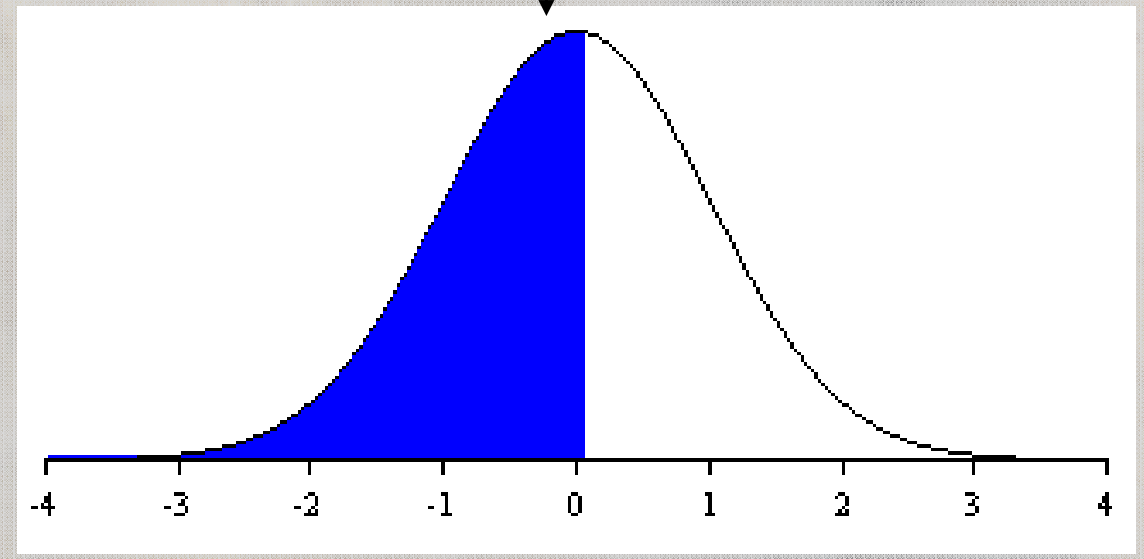
ie Obesity vs BMI
MDD vs quantitative depression scales





Raw data distribution

Mean	= .47
SD	=.499
Non Smokers	=53%
Threshold	=.53

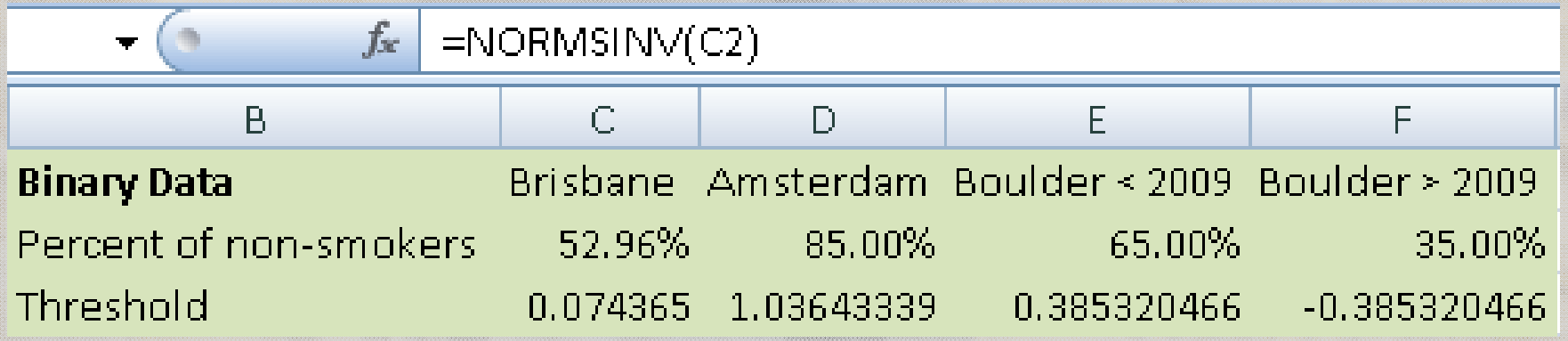


Standard normal distribution

Mean	= 0
SD	=1
Non Smokers	=53%
Threshold	=.074

Threshold = .074 – Huh what?

- How can I work this out
  - Excell
    - =NORMSINV()
    - Thresholds.xls



=NORMSINV(C2)

B	C	D	E	F
Binary Data	Brisbane	Amsterdam	Boulder < 2009	Boulder > 2009
Percent of non-smokers	52.96%	85.00%	65.00%	35.00%
Threshold	0.074365	1.03643339	0.385320466	-0.385320466

  - R
    - qnorm(.5296)

Why rescale the data this way?

- Convenience
  - Variance always 1
  - Mean is always 0
  - We can interpret the area under a curve between two z-values as a probability or percentage

Threshold.R

```
require(OpenMx)
Canabis <- read.table ('two_cat.dat', header=T )

# Print Descriptive Statistics
# ------------------------------------------------------------------
summary(Canabis$twin1)
table(Canabis$twin1)

# Select data
# ------------------------------------------------------------------
Canabis1 <-data.frame(Canabis$twin1)
print( "Note no subset command because I want to use all the data")
head(Canabis1)
print( "This won't work because data names cannot contain '.'")
names(Canabis1) <- "twin1"
head(Canabis1)
```

Threshold.R

```
# Estimate threshold from RawData
# ------------------------------------------------------------------------
nvar <- 1
nthresh <- 1
Vars <-('twin1')
selVars <- ('twin1')
Canabis1$twin1 <- mxFactor(Canabis1$twin1, levels=c(0:1) )


Means       <-   mxMatrix(type="Zero", nrow=nvar, ncol=nvar, name="expMean" )
Thresholds  <-  mxMatrix(type="Full", nrow=nvar, ncol=nthresh, free=TRUE, values=0,
                name="expThresh", label="threshold1",dimnames=list('th1',selVars) )
Corelation  <-   mxMatrix(type="Stand", nrow=nvar, ncol=nvar, name="expCor")
data        <-  mxData(Canabis1, type="raw")
exp         <-      mxExpectationNormal( covariance="expCor", means="expMean", dimnames=selVars,
                thresholds="expThresh" )
funML       <-  mxFitFunctionML()

ThresholdModel <- mxModel("checkThreshold", Means, Thresholds, Corelation, data, exp, funML)

checkThresholdFit <- mxRun(ThresholdModel)
checkThresholdSumm <- summary(checkThresholdFit)
checkThresholdSumm
```

B2 =NORMSINV(B1)

	A	B
1	Percent of non-smokers	0.52964
2	Threshold	0.074365

```
> checkThresholdFit$expThresh
FullMatrix 'expThresh'

@labels
    twin1
th1 "threshold1"

@values
          twin1
th1 0.07436543

@free
    twin1
th1  TRUE

@lbound: No lower bounds assigned.

@ubound: No upper bounds assigned.
```

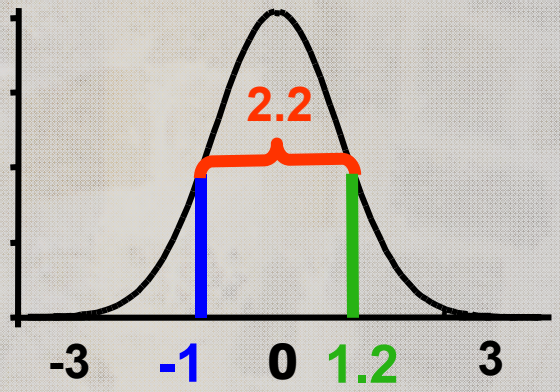
```
> checkThresholdFit$expMean
ZeroMatrix 'expMean'

@labels: No labels assigned.

@values
     twin1
[1,]     0

@free: No free parameters.

@lbound: No lower bounds assigned.

@ubound: No upper bounds assigned.
```

```
> checkThresholdFit$expCor
StandMatrix 'expCor'

@labels: No labels assigned.

@values
      twin1
twin1     1

@free: No free parameters.

@lbound: No lower bounds assigned.

@ubound: No upper bounds assigned.
```

What about more than 2 categories?

- Very similar
  - We create a matrix containing the 1st threshold and the displacements between subsequent matrices
  - We then add the 1st threshold and the displacement to obtain the subsequent thresholds

Mx Threshold Specification: 3+ Cat.



Threshold matrix: `T Full 2 2 Free`

Twin 1 Twin 2

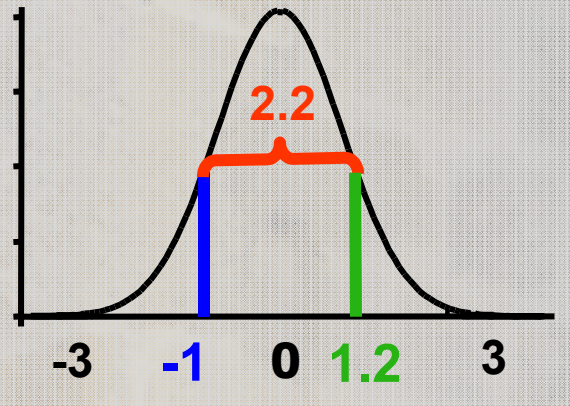
$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$$

1st threshold

increment

Mx Threshold Specification: 3+ Cat.

Threshold matrix: `T Full 2 2 Free`



Twin 1 Twin 2

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$$

1st threshold

increment

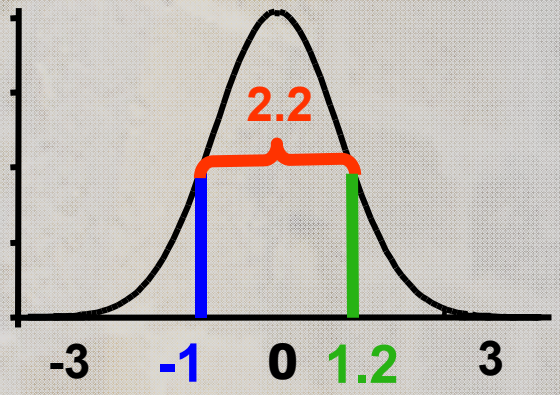
MxAlgebra **L%*%T**

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

$$= \begin{bmatrix} t_{11} & t_{12} \\ t_{11} + t_{21} & t_{12} + t_{22} \end{bmatrix}$$

Mx Threshold Specification: 3+ Cat.

Threshold matrix: `T Full 2 2 Free`



Twin 1 Twin 2

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$$

1st threshold

increment

MxAlgebra L%*%T

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

$$= \begin{bmatrix} t_{11} & t_{12} \\ t_{11} + t_{21} & t_{12} + t_{22} \end{bmatrix}$$

2nd threshold

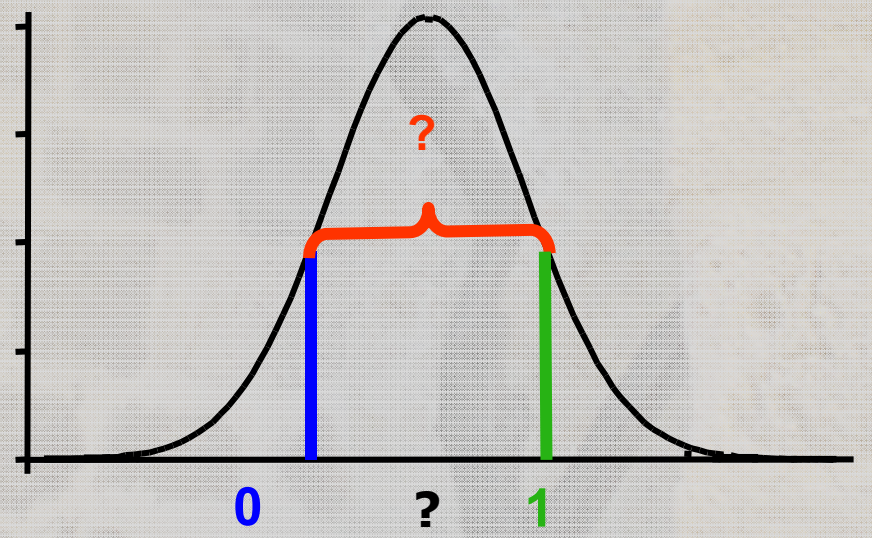
Check the xls spreadsheet…

Ordinal Data	Brisbane	Amsterdam	Boulder < 2009	Boulder > 2009
Never used	52.96%	85.00%	65.00%	35.00%
used >10 times	5.00%	5.00%	5.00%	5.00%
used<10 times	42.04%	10.00%	30.00%	60.00%
Threshold 1	0.074365	1.03643339	0.385320466	-0.385320466
Threshold 2	0.200973	1.28155157	0.524400513	-0.253347103
T11 (1st threshold)	0.074365	1.03643339	0.385320466	-0.385320466
T21 (displacement)	0.126608	0.24511818	0.139080046	0.131973363
T11+T21	0.200973	1.28155157	0.524400513	-0.253347103

Two approaches to the liability threshold model

- Solution?
- Traditional
  - Maps data to a standard normal distribution
  - Total variance constrained to be 1
- Alternate
  - Fixes an alternate parameter
    - Binary or Ordinal data fix E
    - Ordinal data fix 1st two thresholds (aka invariant threshold approach)
  - Estimate the remaining parameters

Fixed Thresholds



Models are equivalent, but…

- Alternate approach means the data is no longer mapped to a standard normal
- No easy conversion to %
- Makes it difficult to compare between groups as the scaling is now arbitrary

- We are going to run traditional and Fixed Thresholds ACE models with ordinal data
  - twinAceOrd-Traditional.R
  - twinAceOrd-FixThreshold.R
- There is are other scripts in the folder – take a look later
  - twinAceBin-Traditional.R
  - twinAceBin-FixE.R
  - twinAceOrd-FixE.R


Lisbon Castle