Univariate 5 ways

Sarah Medland and Lucía Colodro Conde

2020

Today - sarah/2020/tuesday2

We start with the univariate from yesterday

We will look at:

- some extensions of the model
- some different parameterisations

Pay particular attention to the variance covariance model!

Important structural stuff

- openMx has a very fluid and flexible stucture
- Each code snippet is being saved as an object
- We tend to reuse the object names in our scripts
  - There are very few ‘reserved’ names
  - Naming a matrix “mean” does not make it a mean.
- Remember the project also contains the data so these files can become very large.

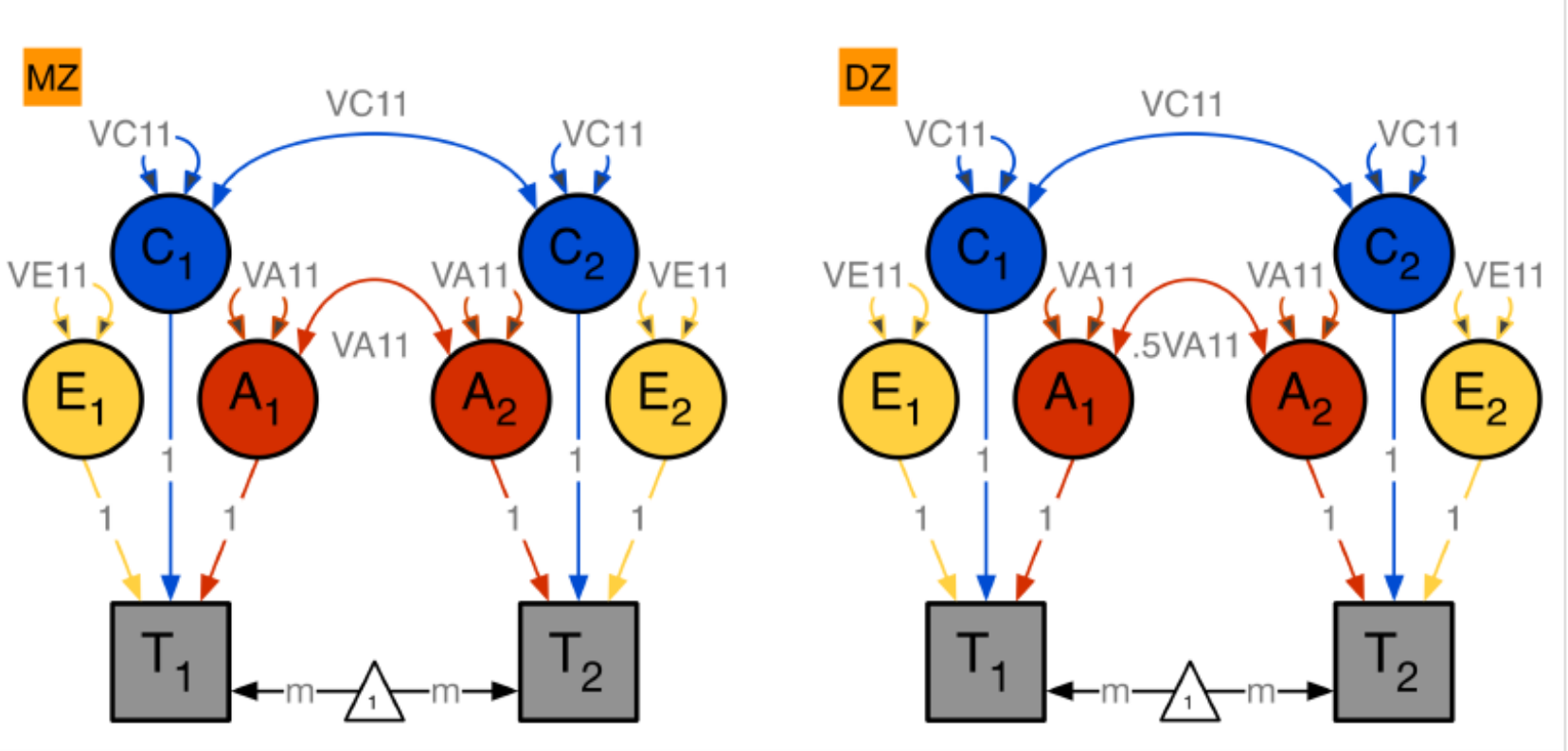
Matrices are the building blocks

```
mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE,
values=.6, label="a11", name="a" ), #X
```

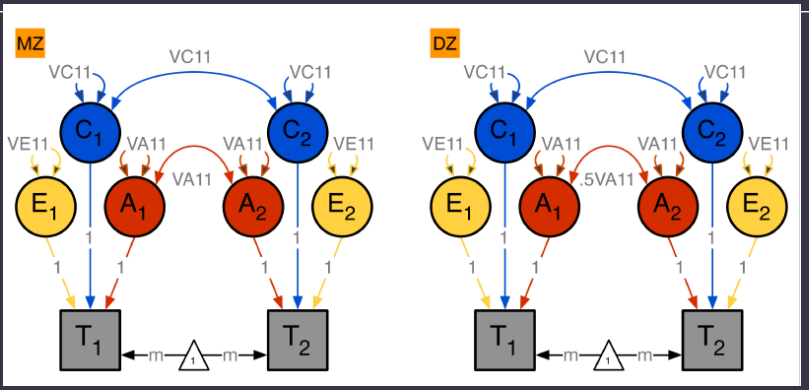
- Many types eg. type="Lower"
- Denoted by names eg. name="a“
- Size eg. nrow=nv, ncol=nv
- All estimated parameters must be placed in a matrix & Mx must be told what type of matrix it is

Yesterday's model

- MZ and DZ pairs – estimating A, C and E



Yesterday's model



- MZ and DZ pairs – estimating A, C and E

MZ

A+C+E	**A+C**
A+C	**A+C+E**

DZ

A+C+E	**.5⊗A+C**
.5⊗A+C	**A+C+E**

MZ

A+C+E	**A+C**
A+C	**A+C+E**

```
covP      <- mxAlgebra( expression= VA+VC+VE, name="V" )
covMZ     <- mxAlgebra( expression= VA+VC, name="cMZ" )
```

V	**cMZ**
cMZ	**V**

```
expCovMZ  <- mxAlgebra( expression= rbind( cbind(V, cMZ),
                                           cbind(t(cMZ), V)),
                                           name="expCovMZ" )
```

DZ

A+C+E	.5⊗A+C
.5⊗A+C	A+C+E

```
covP      <- mxAlgebra( expression= VA+VC+VE, name="V" )
covDZ     <- mxAlgebra( expression= 0.5%x%VA+VC, name="cDZ" )
```

V	cDZ
cDZ	V

```
expCovDZ  <- mxAlgebra( expression= rbind( cbind(V, cDZ),
                                           cbind(t(cDZ), V)),
                                           name="expCovDZ" )
```

00_ACEvc.R

- Run the ACE model
- Look at the output
  - (type “sumACE”)
- Record the output in Tuesday2.xls

- Any questions about this model or script?

Next step – add a sibling

- Let's include 1 extra sibling in the analysis
- Assume that this is a non-twin full sibling
  - What would the variance of the sibling be in the ACE model we just ran? (trick question)
  - What would the covariance be between the sibling and twin1? (trick question)
    - Is this the same for MZ and DZ families? (trick question)

Next step – add a sibling

- MZ

A+C+E	A+C	**.5⊗A+C**
A+C	A+C+E	**.5⊗A+C**
.5⊗A+C	**.5⊗A+C**	**A+C+E**

Next step – add a sibling

- expCovMZ <- mxAlgebra(expression=

rbind(cbind(V, cMZ, cDZ),
cbind(t(cMZ), V, cDZ),
cbind(t(cDZ), t(cDZ), V)), name="expCovMZ")

A+C+E	A+C	**.5⊗A+C**
A+C	A+C+E	**.5⊗A+C**
.5⊗A+C	**.5⊗A+C**	**A+C+E**

Next step – add a sibling

- DZ

A+C+E	.5⊗A+C	**.5⊗A+C**
.5⊗A+C	A+C+E	**.5⊗A+C**
.5⊗A+C	**.5⊗A+C**	**A+C+E**

Next step – add a sibling

- expCovDZ <- mxAlgebra(expression=

rbind(cbind(V, cDZ, cDZ),
cbind(t(cDZ), V, cDZ),
cbind(t(cDZ), t(cDZ), V)), name="expCovDZ")

A+C+E	.5⊗A+C	**.5⊗A+C**
.5⊗A+C	A+C+E	**.5⊗A+C**
.5⊗A+C	**.5⊗A+C**	**A+C+E**

Next step – add a sibling

- Q: What about if some families have siblings and others don't?
- A: That is fine because we use full information maximum likelihood (FIML) methods
  - model your biggest family size
  - missing phenotypes for the non-exsistent sibs BUT you do need to give them covariates
  - assumes missing at random

01_extrasib.R

- Two versions – if you have some mx experience try challenge_01_extrasib.R

- Run the ACE model
- Look at the output
  - (type “sumACE”)
- Record the output in Tuesday2.xls

- Any questions about this model or script?

Variation on this theme

- Although it can be helpful to write out the full variance/covariance matrix it quickly becomes unwieldy
  - imagine doing this if you largest family = 10 sibs…
- expCovMZ <- mxAlgebra(expression=

```
rbind(cbind(V,     cMZ,   cDZ),
      cbind(t(cMZ), V,     cDZ),
      cbind(t(cDZ), t(cDZ), V)), name="expCovMZ" )
```

Alternate parameterisation

Lets think about A for an MZ family

A+C+E	A+C	.5⊗A+C
A+C	A+C+E	.5⊗A+C
.5⊗A+C	.5⊗A+C	A+C+E

A	A	.5⊗A
A	A	.5⊗A
.5⊗A	.5⊗A	A

= A ⊗

1	1	.5
1	1	.5
.5	.5	1

Alternate parameterisation

Lets think about A for an DZ family

A+C+E	.5⊗A+C	.5⊗A+C
.5⊗A+C	A+C+E	.5⊗A+C
.5⊗A+C	.5⊗A+C	A+C+E

A	.5⊗A	.5⊗A
.5⊗A	A	.5⊗A
.5⊗A	.5⊗A	A

= A ⊗

1	.5	.5
.5	1	.5
.5	.5	1

Alternate parameterisation

What about C?

A+C+E	.5⊗A+C	.5⊗A+C
.5⊗A+C	A+C+E	.5⊗A+C
.5⊗A+C	.5⊗A+C	A+C+E

C	C	C
C	C	C
C	C	C

= c ⊗

1	1	1
1	1	1
1	1	1

Alternate parameterisation

What about E?

A+C+E	.5⊗A+C	.5⊗A+C
.5⊗A+C	A+C+E	.5⊗A+C
.5⊗A+C	.5⊗A+C	A+C+E

E	E	E
E	E	E
E	E	E

= E ⊗

1	0	0
0	1	0
0	0	1

How do we do this in the script?

- relMZ <- mxMatrix(type="Symm", nrow=ntv, ncol=ntv, free=FALSE, values=c(1,1,.5,1,.5,1), name="rAmz")
- relDZ <- mxMatrix(type="Symm", nrow=ntv, ncol=ntv, free=FALSE, values=c(1,.5,.5,1,.5,1), name="rAdz")

relMZ (rAmz)

1	**1**	.5
1	1	.5
.5	.5	1

relDZ (rAdz)

1	**.5**	.5
.5	1	.5
.5	.5	1

How do we do this in the script?

- relMZ <- mxMatrix(type="Symm", nrow=ntv, ncol=ntv, free=FALSE, values=c(1,1,.5,1,.5,1), name="rAmz")
- relDZ <- mxMatrix(type="Symm", nrow=ntv, ncol=ntv, free=FALSE, values=c(1,.5,.5,1,.5,1), name="rAdz")

r here is the ***coefficient of relatedness***

relMZ (rAmz)

1	**1**	.5
1	1	.5
.5	.5	1

relDZ (rAdz)

1	***.5***	.5
.5	1	.5
.5	.5	1

How do we do this in the script?

- relC <- mxMatrix(type="Unit", nrow=ntv, ncol=ntv, free=FALSE, name="rC")
- relE <- mxMatrix(type="Iden", nrow=ntv, ncol=ntv, free=FALSE, name="rE")

relC (rC)

1	1	1
1	1	1
1	1	1

relE (rE)

1	0	0
0	1	0
0	0	1

How do we do this in the script?

- expCovMZ <- mxAlgebra(expression=
  VA%x%**rAmz** + VC%x%rC + VE%x%rE,
  name="expCovMZ")

- expCovDZ <- mxAlgebra(expression=
  VA%x%**rAdz** + VC%x%rC + VE%x%rE,
  name="expCovDZ")

02_extrasib2.R

- Run the ACE model
- Look at the output
  - (type “sumACE”)
- Record the output in Tuesday2.xls

- Any questions about this model or script?

Can we make this even more efficient?

What are the differences between the MZ and DZ groups?

- relMZ <- mxMatrix(type="Symm", nrow=ntv, ncol=ntv, free=FALSE, values=c(1,**1**,.5,1,.5,1), name="rAmz")
- relDZ <- mxMatrix(type="Symm", nrow=ntv, ncol=ntv, free=FALSE, values=c(1,**.5**,.5,1,.5,1), name="rAdz")

relMZ (rAmz)

1	**1**	.5
1	1	.5
.5	.5	1

relDZ (rAdz)

1	**.5**	.5
.5	1	.5
.5	.5	1

Is there another way we could do this?

How about we read this coefficient from the data and only have one group?

- relA <- mxMatrix(type="Stand", nrow=ntv, ncol=ntv, free=FALSE, labels=c("data.zyg","data.zyg2","data.zyg2"), name="rA")

relA (rA)

1	**zyg**	zyg2
zyg	1	zyg2
zyg2	zyg2	1

Putting **data.** in the label tells openMx that this is a definition variable and should be updated dynamically for each case in the data

Is there another way we could do this?

How about we read this coefficient from the data and only have one group?

- relA <- mxMatrix(type="Stand", nrow=ntv, ncol=ntv, free=FALSE, labels=c("data.zyg","data.zyg2","data.zyg2"), name="rA")

zyg = 1 for MZs
zyg = .5 for DZs

zyg2 =.5 for everyone

1	zyg	zyg2
zyg	1	zyg2
zyg2	zyg2	1

Twin1	Twin2	Sib	s1	s2	s3	a1	a2	a3	sex1	sex2	sex3	zyg	zyg2	zygosity
-1.554	-1.370	-2.385	0.52463	0.511	0.466	30.506	29.866	34.205	0	0	1	0.5	0.5	2
-1.968	-1.470	-2.279	1	0.482	0.522	24.630	32.214	17.769	0	1	1	1	0.5	1
-1.605	-1.991	-2.184	0.47602	0.573	0.501	30.298	36.711	29.852	1	1	0	0.5	0.5	2
-0.501	-0.758	-2.182	1	0.468	0.535	24.435	20.991	25.800	1	0	0	1	0.5	1
-0.844	-0.500	-2.162	1	0.496	0.520	38.463	24.808	18.607	1	0	0	1	0.5	1
-0.654	-1.172	-2.161	1	0.539	0.463	38.219	37.255	20.900	0	0	1	1	0.5	1
-0.687	-1.058	-2.104	0.51559	0.485	0.509	23.963	32.485	28.586	1	1	0	0.5	0.5	2

03_zygdef.R

- Run the ACE model
- Look at the output
  - (type “sumACE”)
- Record the output in Tuesday2.xls

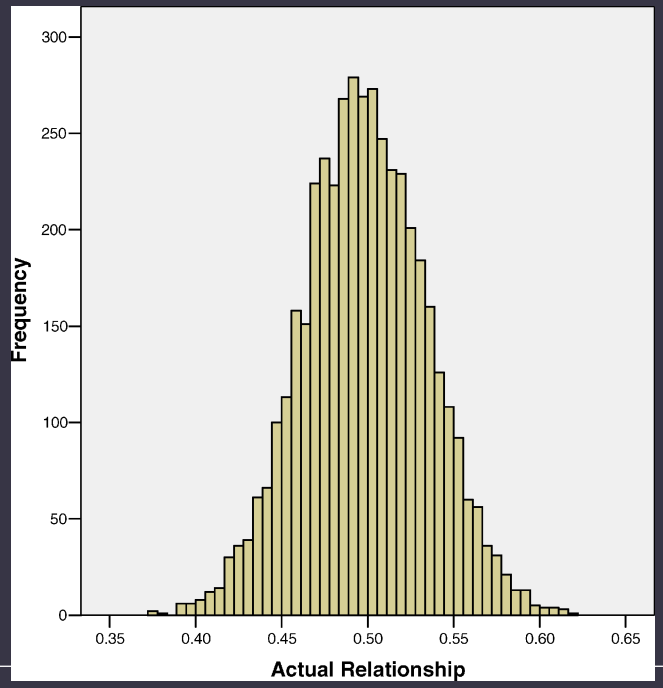
- Any questions about this model or script?

Is this more efficient?

- 01_extrasib.R
  - With CIs: Wall clock time: 49.40628 secs
  - Without CIs: Wall clock time: 1.26036 secs
- 02_extrasib2.R
  - With CIs: Wall clock time: 29.92903 secs
  - Without CIs: Wall clock time: 1.300223 secs
- 03_zygdef.R
  - With CIs: Wall clock time: 39.61322 secs
  - Without CIs: Wall clock time: 1.577673 secs

Variations on this theme

- How about including actual genetic relatedness instead of the .5 or 1?
  - Estimate genetic relatedness by computing a GRM in PLINK or GCTA



Variations on this theme

- How about including actual genetic relatedness instead of the .5 or 1?
  - Estimate genetic relatedness by computing a GRM in PLINK or GCTA

- relA <- mxMatrix(type="Stand", nrow=ntv, ncol=ntv, free=FALSE, labels=c("data.s1","data.s2","data.s3"), name="rA")

1	S1	S2
S1	1	S3
S2	S3	1

Twin1	Twin2	Sib	s1	s2	s3	a1	a2	a3	sex1	sex2	sex3	zyg	zyg2	zygosity
-1.554	-1.370	-2.385	0.52463	0.511	0.466	30.506	29.866	34.205	0	0	1	0.5	0.5	2
-1.968	-1.470	-2.279	1	0.482	0.522	24.630	32.214	17.769	0	1	1	1	0.5	1
-1.605	-1.991	-2.184	0.47602	0.573	0.501	30.298	36.711	29.852	1	1	0	0.5	0.5	2
-0.501	-0.758	-2.182	1	0.468	0.535	24.435	20.991	25.800	1	0	0	1	0.5	1
-0.844	-0.500	-2.162	1	0.496	0.520	38.463	24.808	18.607	1	0	0	1	0.5	1
-0.654	-1.172	-2.161	1	0.539	0.463	38.219	37.255	20.900	0	0	1	1	0.5	1
-0.687	-1.058	-2.104	0.51559	0.485	0.509	23.963	32.485	28.586	1	1	0	0.5	0.5	2

04_relatedness.R

- Run the ACE model
- Look at the output
  - (type "sumACE")
- Record the output in Tuesday2.xls

- Any questions about this model or script?
- How do the answers compare to the previous scripts?

Final variation…

- Once we include measured relationships the model we don't technically need MZs to make the model identified

Final variation…

- When would we do this?
  - If the equal environments assumption was problematic for your trait
  - If we only had sibling pairs
  - (If we want to show we're super clever…)

05_relatednessDZonly.R

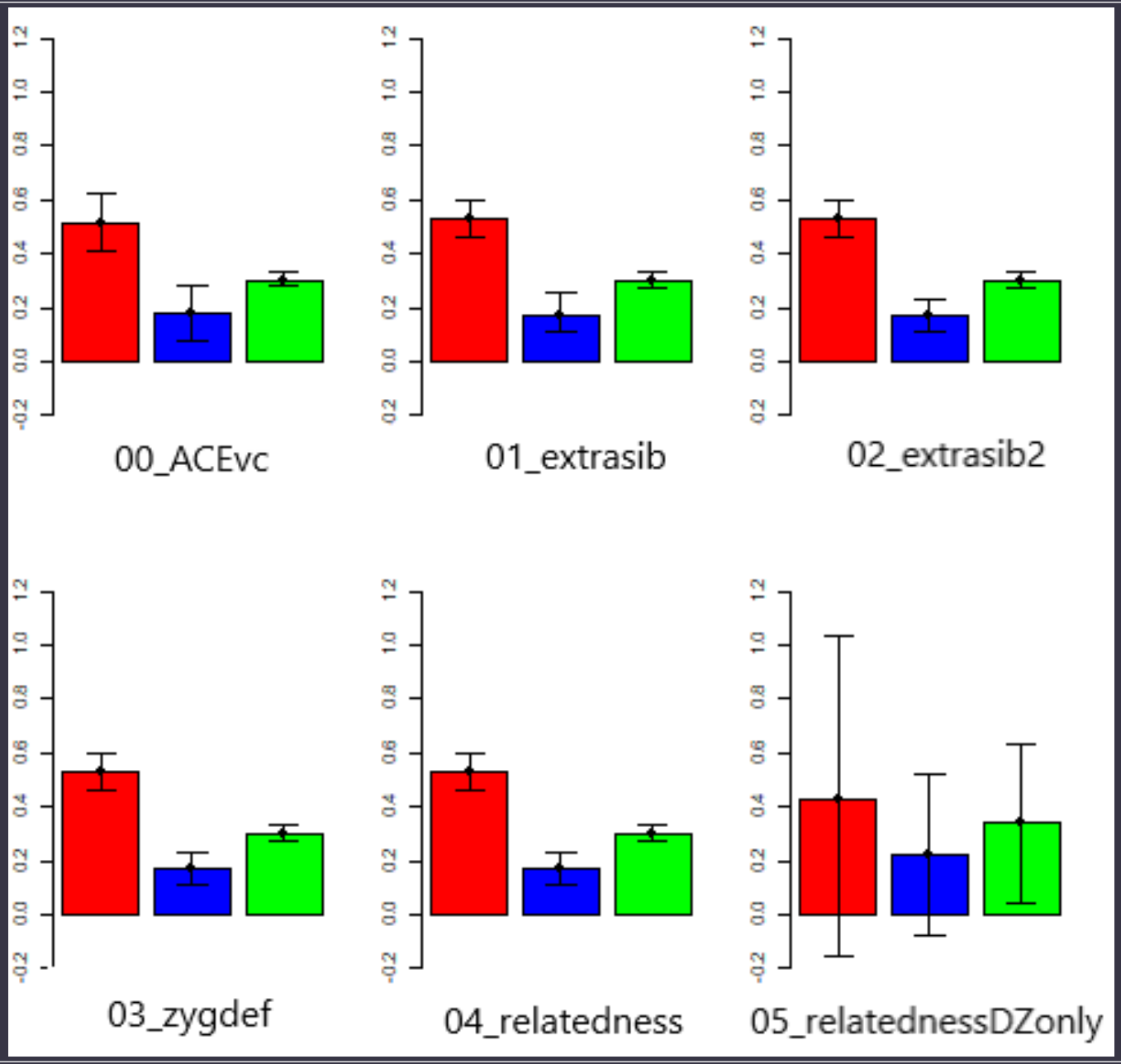
- Run the ACE model
- Look at the output
  - (type "sumACE")
- Record the output in Tuesday2.xls

- Any questions about this model or script?
- How do the answers compare to the previous scripts?
  - If you have MZ twins is this a good use of your data?

In summary

	A	C	E
00_ACEvc	0.52 (0.41, 0.62)	0.18 (0.08,0.28)	0.30 (0.28, 0.33)
01_extrasib	0.53 (0.46, 0.60)	0.17 (0.11, 0.23)	0.30 (0.27, 0.33)
02_extrasib2	0.53 (0.46, 0.60)	0.17 (0.11, 0.23)	0.30 (0.27, 0.33)
03_zygdef	0.53 (0.46, 0.60)	0.17 (0.11, 0.23)	0.30 (0.27, 0.33)
04_relatedness	0.53 (0.46, 0.60)	0.17 (0.11, 0.22)	0.30 (0.27, 0.33)
05_relatednessDZonly	0.43 (-0.15, 1.03)	0.22 (-0.08, 0.52)	0.34 (0.05, 0.64)

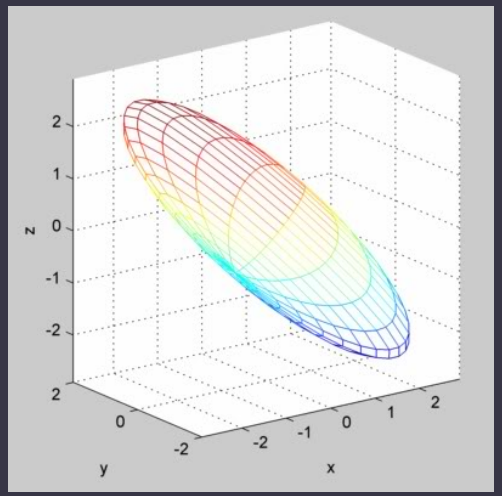
This is a different simulation run so pay more attention to the width of the CIs than the point estimates


00_ACEvc
01_extrasib
02_extrasib2
03_zygdef
04_relatedness
05_relatednessDZonly
A
C
E

Thinking out side the box…

Rather than thinking about estimates as fixed points I like to think about parameter space…

Imagine an ACE model as a solution space bounded by CIs



"Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful"

George E P Box and Norman R Draper. 1986. *Empirical Model-Building and Response Surface*. John Wiley & Sons, Inc., New York, NY, USA.

Questions?