Univariate Modeling in *umx*

Timothy C. Bates

tim.bates@ed.ac.uk

umx: Twin and path-based structural equation modeling in R
TC Bates, H Maes, MC Neale
Twin Research and Human Genetics 22 (1), 27-41

14 2019

Roadmap for this interactive tutorial on Univariate Twin Analysis

1. Introducing and installing umx
2. Using umx to model genetic and environmental effects on total variance of a trait
   - **umxACEv** models with ACE and ADE decomposition
3. Comparing models
   - **umxCompare** to compare **umxACEv**() and **umxACE**() models
4. Testing fit of reduced models
   - **umxModify** to create AE, CE and E-Only Models

Installing umx

- Typically installing umx from CRAN will be fine
  - `install.packages("umx")`
- At the workshop, we might roll out new features, so using the development version will give access to these for everyone during the workshop
  - `devtools::install_github("tbates/umx")`
  - `Note: this requires install.packages("devtools")`
- Dependencies
  - umx will try and install all its dependencies, but cannot always catch everything it needs
  - The response is straightforward: **Read** those messages and just install the package(s) being requested
- So if the message “There is no package "XML" ”
  - `install.packages("XML")`
  - `devtools::install_github("tbates/umx")`

What version of umx, OpenMx, R do I have?

`umxVersion()`

umx version: 3.0.6

OpenMx version: 2.17.2.12 [GIT v2.17.2-12-gbe68a5e]

R version: R version 3.6.1 (2019-07-05)

Platform: x86_64-apple-darwin15.6.0

MacOS: 10.15.4

Default optimizer: CSOLNP

NPSOL-enabled?: Yes

OpenMP-enabled?: Yes

You can update OpenMx with:

install.OpenMx(c("NPSOL", "travis", "CRAN", "open travis build page")

General production line

- make → modify → compare → summarize → plot

```
m1 = umxACEv(mzData = mzData, dzData= dzData)
umxSummary(m1)
plot(m1)
m2 = umxACE(mzData = mzData, dzData= dzData)
umxCompare(m1, m2)
m3 = umxModify(m1, "c_r1c1", comparison = TRUE)
```

umx implements a wide range of Twin Modeling Functions

- umxACEv, umxACE
- Factor models:
  - umxCP()
  - umxIP()
- Gene-environment interaction
  - umxGxE(), umxGxEbiv(), umxGxE_window
- Others:
  - umxSexLim
  - umxMendelianRandomization()
  - umxSimplex
  - umxDoC
- Path-based SEM
  - umxTwinMaker

- Each has a umxSummary() and plot() method

- Many other functions, e.g.
  - umxReduce()
  - umxEFA()

- Explore ?umx
  - Look at the families of functions
- Look at the help!
  - umx help has path diagrams, lots of real examples

Many other handy functions

- Other Reporting Functions
  - parameters()
    - Show model parameters
  - umxAPA:
    - formats many things
- umx_set_optimizer()
- umx_set_auto_plot()
- `power.ACE.test(AA = .3, CC = 0, update = c"a", MZ_DZ_ratio= .85, n= 141, power = NULL, method = "empirical")`

- Twin Data functions
  - umx_long2wide()
  - umx_wide2long()
  - umx_make_TwinData()
    - Simulate twin data, inc GxE
    - umx_make_MR_data()
  - umx_residualize()
    - Flexible residualization, inc. twin data
  - umx_scale_wide_twin_data()
    - Allows scaling of wide data
  - umx_make_twin_data_nice

How to run an ACE analysis in umx

umxACEv()

Read the help ?umxACEv

umxACE()

Read the help ?umxACE

umxACEv:
Many parameters… most can be ignored!

- **umxACEv**(name = "ACE", **selDVs**, **dzData**, **mzData**, **sep**)

- Less-often used parameters: , dzAr = 0.5, dzCr = 1, type, addStd = TRUE, addCI = TRUE, numObsDZ, numObsMZ, boundDiag, weightVar, equateMeans = TRUE, bVector = FALSE, autoRun, optimizer

- nb: Some options not completed. e.g. fixed-effect covariates

umxACEv: just the parameters we need

m1 = umxACEv(**selDVs =** "wt", **suffix** ="",
dzData=dzData, **mzData**=mzData)

All variables continuous

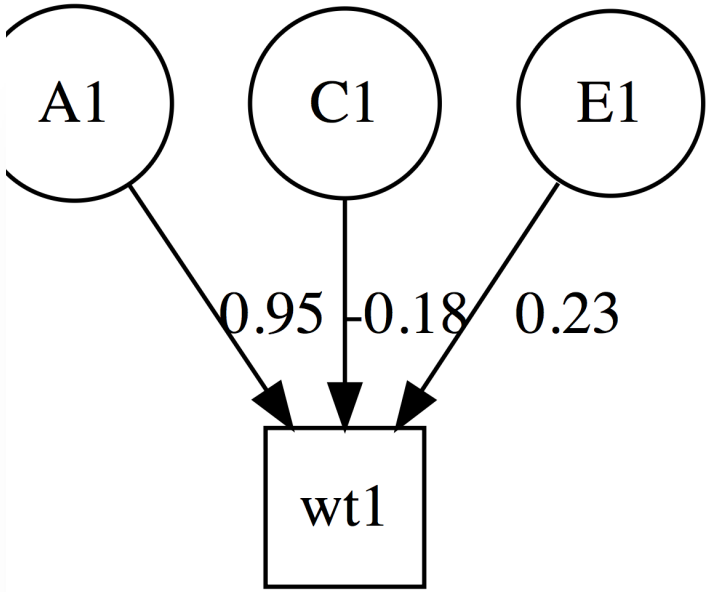
treating data as raw

Running ACE with 4 parameters

ACE -2×log(Likelihood) 27278.55 (df=4)

Standardized solution

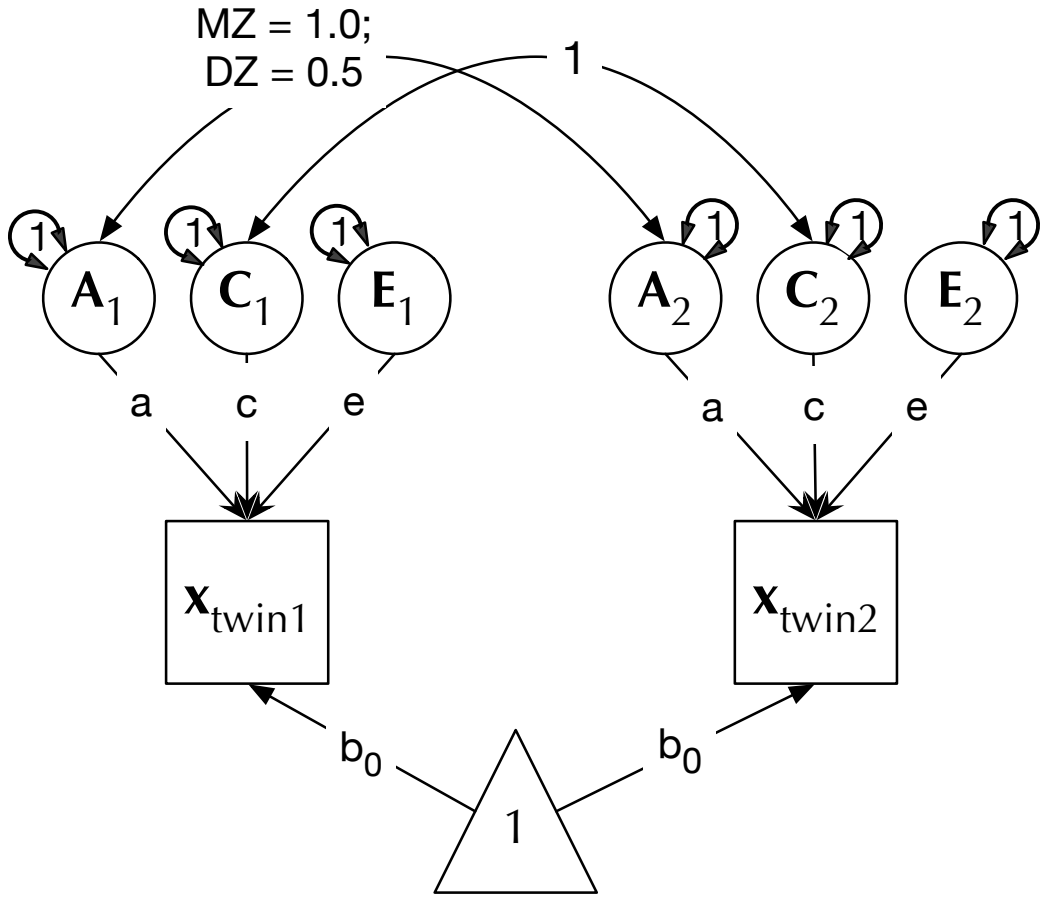
	A1	C1	E1
wt1	0.95	-0.18	0.23



plot(model)

- plot() will make a graph, and open it in your browser
- Can also output pdf or .dot
  - OmniGraffle or VISIO to edit plots for publication.

```
umx_set_auto_plot(TRUE/FALSE)
umx_set_plot_file_suffix()
```



What's in the file plot() saves?

Graphviz "dot" language (like S (R's parent), invented at Bell Laboratories

```
digraph G {
    splines = "FALSE";
    # Latents
    a1 [shape = circle];
    c1 [shape = circle];
    e1 [shape = circle];

    # Manifests
    ht1 [shape = square];
   a1 -> ht1 [label = "0.92"];
   c1 -> ht1 [label = "0.14"];
   e1 -> ht1 [label = "0.36"];
    {rank = same;  ht1 };
    {rank = min   ;  a1 };
    {rank = max   ;  c1; e1 };
}
```

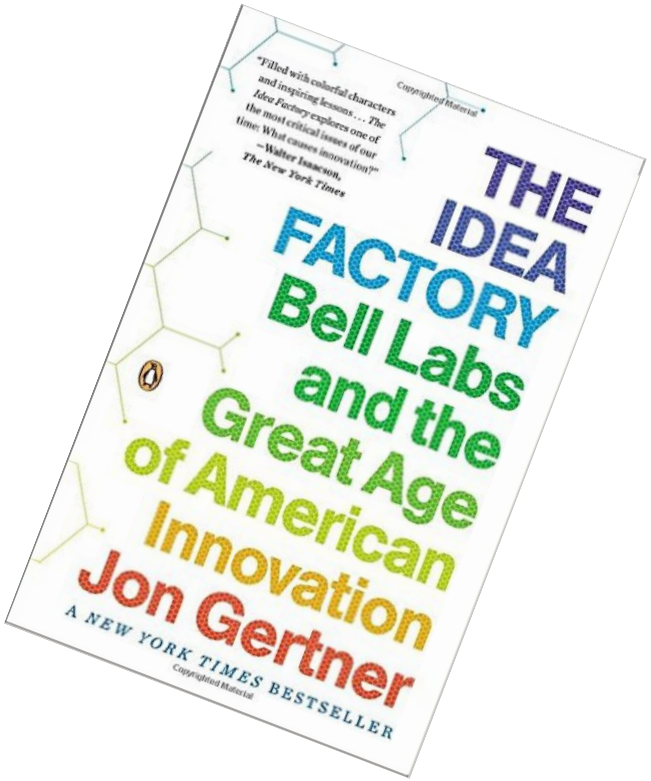
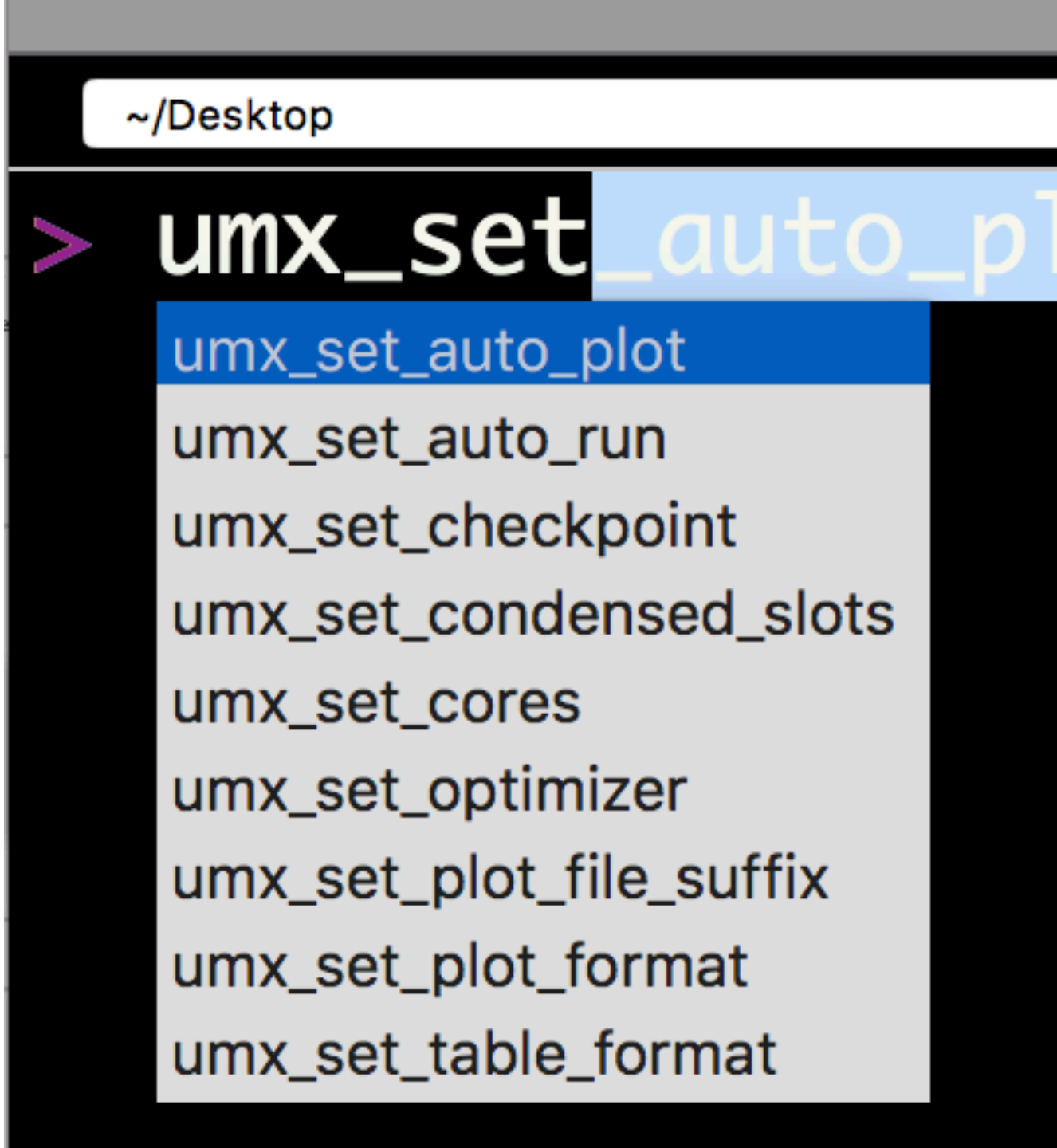


Table output

- Markdown by default
- Can be latex, html,…

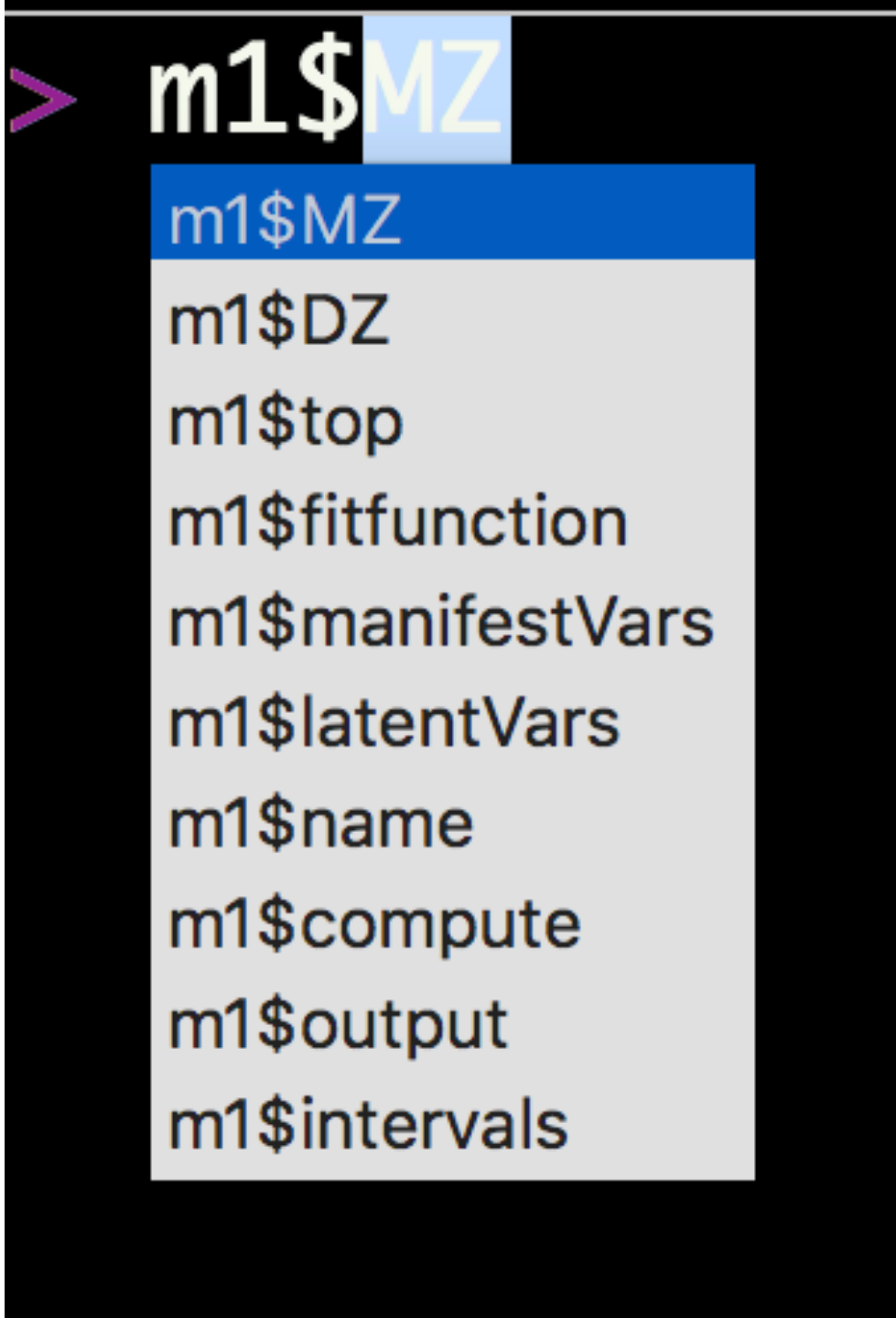
```
umx_set_table_format()

Current format is'markdown'.
Valid options are 'latex',
'html', 'markdown', 'pandoc',
and 'rst'>
```



What's inside this model?

- "**top**" is a model that contains all the shared matrices and algebras
- **MZ** is a model contains the MZ data and how to optimize this
- **DZ** is a model contains the DZ data and how to optimize this



What's inside top?

- A, C, and E are matrices for our three variance components.
- dzAr and dzCr are our .5 and 1 expected correlations for DZs

FullMatrix 'expMean’

```
$labels       wt1          wt2
   means "expMean_r1c1" "expMean_r1c1”

$values       wt1          wt2
   means      58.81        58.81

$free         wt1          wt2
   means      TRUE         TRUE
```

SymmMatrix 'A'

```
$labels   [,1]
     [1,] "A_r1c1”

$values   [,1]
     [1,]   82.36

$free     [,1]
     [1,] TRUE

$lbound: No lower bounds assigned.
$ubound: No upper bounds assigned.
```

m1topACE

```
mxAlgebra 'ACE’

$formula:  A + C + E

$result: [,1]
     [1,] 86.61276

dimnames: NULL
```

m1topexpCovMZ

```
mxAlgebra 'expCovMZ’
$formula:  rbind(cbind(ACE, AC), cbind(AC, ACE))

$result:          wt1      wt2
           wt1 86.61   66.35
           wt2 66.35   86.61
dimnames: [1] "wt1" "wt2"
          [2] "wt1" "wt2"
```

Univariate model of weight

```
#' # ========================
#' # = Univariate model of weight =
#' # ========================
#' require(umx)
#' data(twinData) # ?twinData from Australian twins.
#'
#' # Things to note: ACE model of weight will return a NEGATIVE variance in C.
#' #  This is why we have ACEv! It suggests we need a different model
#' #  In this case: ADE.
#' # Other things to note:
#' # 1. umxACEv can figure out variable names: provide "sep" and "wt" -> "wt1" "wt2"
#' # 2. umxACEv picks the variables it needs from the data.
#'

#' mzData = twinData[twinData$zygosity %in% "MZFF", ]
#' dzData  = twinData[twinData$zygosity %in% "DZFF", ]

#' m1 = umxACEv(selDVs = "wt", sep = "", dzData = dzData, mzData = mzData)
```

ADE model

- ADE model
  - What's the difference for ACE and ADE?
  - **Just change dzCr to .25**

Evidence for dominance ? (DZ correlation set to .25)

```
m2 = umxACEv(selDVs ="wt", sep = "", dzData = dzData, mzData = mzData, dzCr = .25)
```

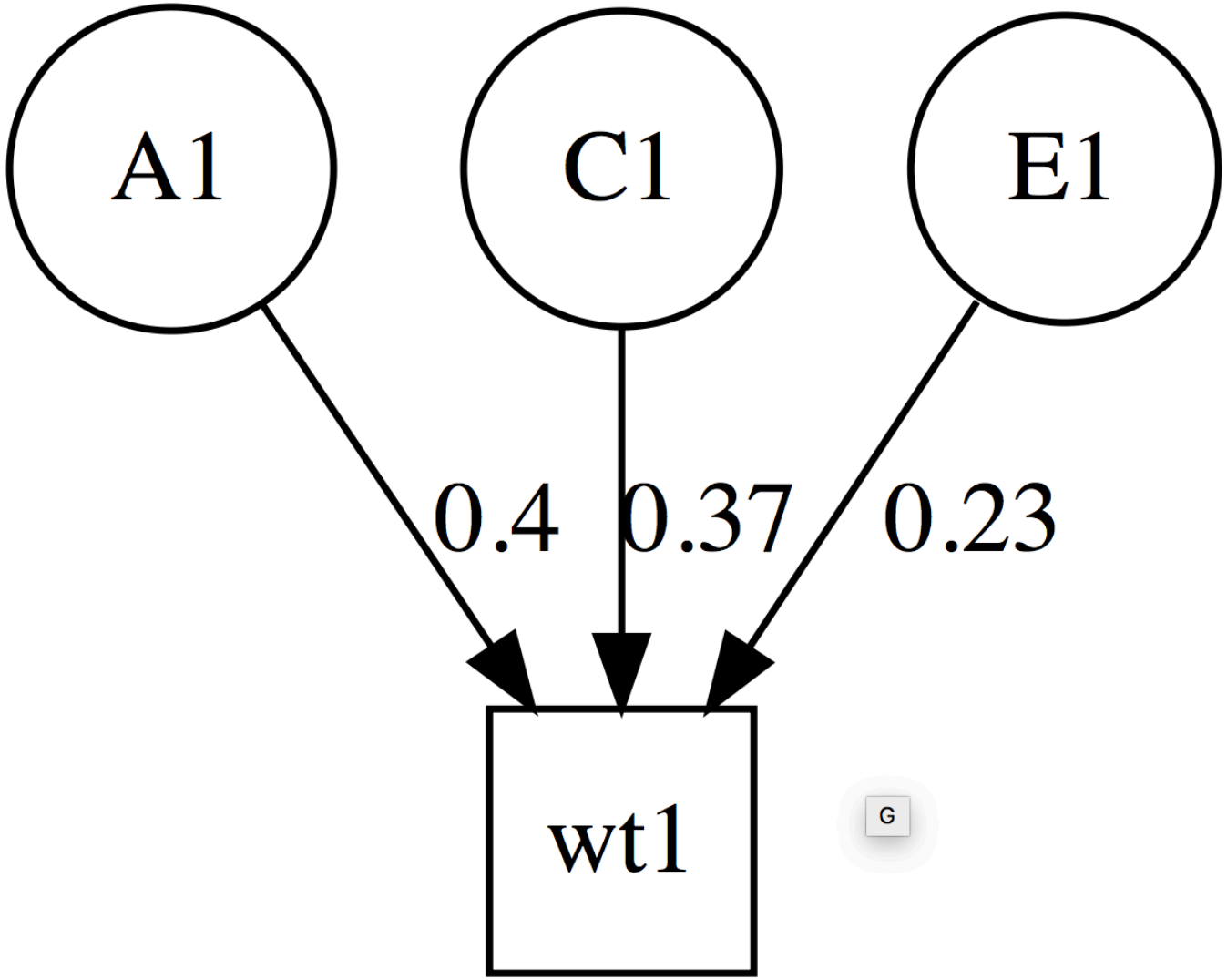
note: the underlying matrices are still called A, C, and E.

I catch this in the summary table, so columns are labeled A, D, and E.

However, currently, the plot will say A, C, E.

Table and plot: what changed?

	A1	D1	E1
wt1	0.4	0.37	0.23



Something to Try:

```
umxSummary(m2, report="html")
```

- What happened?

umxACE

- ?umxACE

```
m1=umxACE(selDVs="wt", sep="",
dzData=dzData, mzData=mzData)
```

```
ACE -2 × log(Likelihood)'log
Lik.' 27287.23 (df=4)
```

```
Standardized solution
|    |   a1|c1 |   e1|
|:---|----:|:--|----:|
|wt1 | 0.87|.  | 0.49|
```

umxACE with dominance instead of C

```
m2 = umxACE("ADE", selDVs = selDVs, sep="", dzData = dzData, mzData = mzData, dzCr = .25)
umxCompare(m2, m1) # ADE seems better?
```

	a1	d1	e1
wt1	0.63	0.61	0.48

What about if we compare umxACEv and umxACE?

```
m1 = umxACEv("ADEvari", selDVs = selDVs, sep="", dzData = dzData, mzData = mzData, dzCr = .25)

m2 = umxACE("ADEchol", selDVs = selDVs, sep="", dzData = dzData, mzData = mzData, dzCr = .25)

umxCompare(m1,m2)
```

Model	EP	Δ -2LL	Δ df	p	AIC	Compare with Model
ADEvari	4				19506.55	
ADEchol	4	0	0		19506.55	ADEvari

Modify model

- **umxModify** allows you to modify, re-run and summarize a model.

```
new= umxModify(m1, update="label",
   name = "newName", comparison=TRUE)
```

- Less commonly needed parameters
  - master = NULL, regex = FALSE, free = FALSE, value = 0, newlabels = NULL, freeToStart = NA, name = NULL, verbose = FALSE, intervals = FALSE, **comparison** = FALSE, autoRun = TRUE)

Modify model

```
m3 = umxModify(m2, update = "C_r1c1", name = "AE",
comparison = T)
```

- This will:
  - Make a new model “m3” with name “AE”
  - Drop parameter "C_r1c1” from this model
  - Run the new model
  - Print a comparison of fit of m2 and m3

We can modify this model, dropping dominance component (matrix is still called C)

```
m3= umxModify(m2,update="C_r1c1",comparison=T, name="AE")
```

	A1	D1	E1
wt1	0.76	.	0.24

Model	EP	Δ -2LL	Δ df	p	AIC	Compare with Model
ADE	4				19506.55	
AE	3	8.675	1	0.003	19513.23	ADE

We can attempt to drop A

```
m4= umxModify(m2,update="A_r1c1",comparison=T, name="DE")
umxCompare(m2, c(m3, m4))
```

Model	EP	Δ -2LL	Δ df	p	AIC	Compare with Model
ADE	4				19506.55	
AE	3	8.6750179	1	0.003	19513.23	ADE
DE	3	8.2983642	1	0.004	19512.85	ADE

Easter egg: umxReduce()

- Try umxReduce(m1) where m1 is your umxACE model
  - (or a umxGxE() model…)
- It will do all the sensible things an expert might wonder about:
  - C or D?
  - Drop C?
  - Drop A?

Well done!

```
#=============================================
# = Well done!
# = Now you can make and modify twin models
# = in umx ☺
#=============================================
```

umx: Twin and path-based structural equation modeling in R 14 2019
TC Bates, H Maes, MC Neale
Twin Research and Human Genetics 22 (1), 27-41