Pmetrics Data and Models

Michael Neely, MD
Professor of Pediatrics and Clinical Scholar
Director, Laboratory of Applied Pharmacokinetics and Bioinformatics
University of Southern California, Children’s Hospital Los Angeles

General Workflow

1. In R/Rstudio, use `setwd()` to ensure that your current working directory is the project /src folder
2. Use PM_data$new() to create a PM_data object in memory
3. Use PM_model$new() to create a PM_model object in memory
   - Source can be a file or a list in R
4. Combine the data and model with PM_fit$new()
5. Execute $run on the PM_fit object

Pmetrics Data Files

- One row, one event (dose or observation)
- All subjects continuous
- Column names are fixed
- Use “.” as placeholders in empty cells

Making Data files

- Manually, in Excel or another spreadsheet program
- Exporting from some other program and then manipulating in Excel or preferably, with R scripts
- Converting from NONMEM data files using `NM2PM()`
- Converting from old USC*PACK .wrk format using `PMwrk2csv()`
- Converting from old USC*PACK .mb format using `PMmb2csv()`

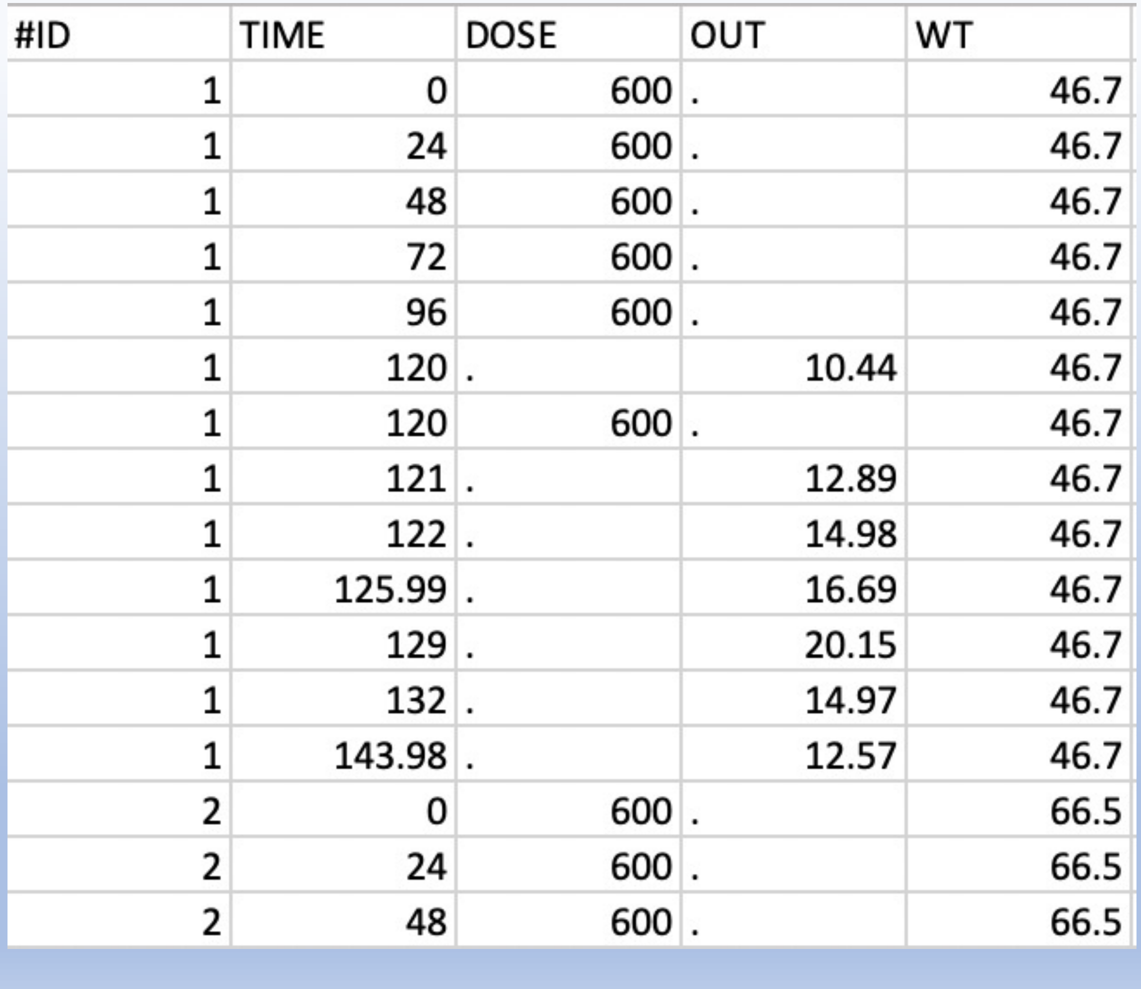
Why I like R scripts

- I never touch the original data file
- If the source data file is updated, I simply re-run the R script
- Can build in validation
- Useful R commands: `read.csv()`, `read.table()`, `read.xlsx()` [xlsx package], `read.spss()`, `read.ssd()`, `read.xport()`

More useful commands

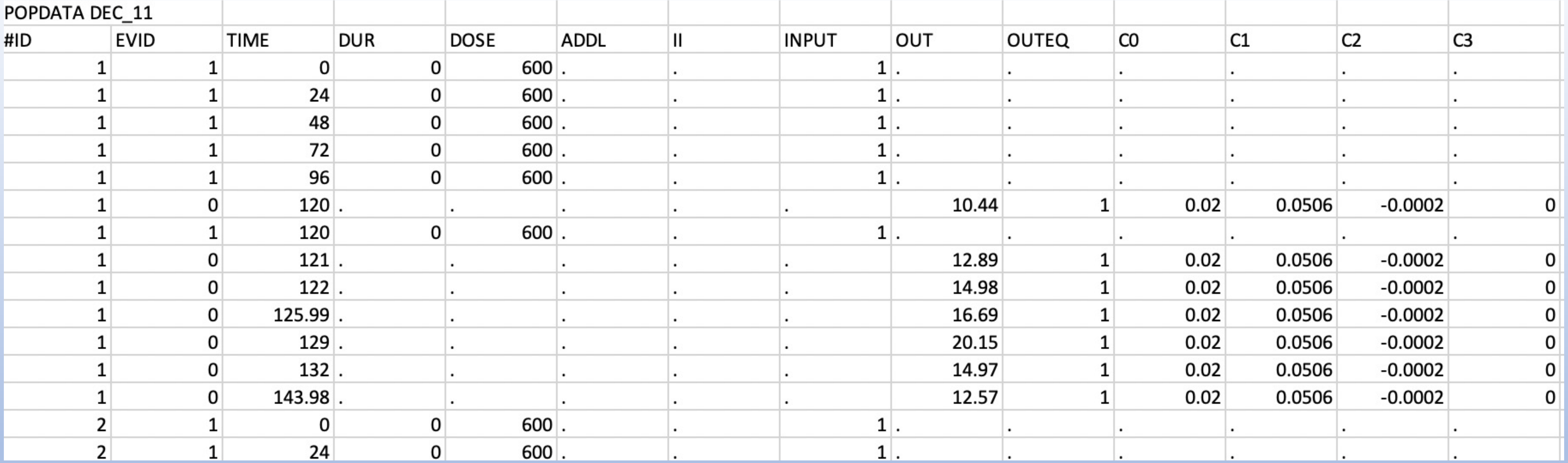
- `PM_data$new(file)` - read Pmetrics data file into R
- `PM_data$write(file)` - write a PM_data object to a file
- Both automatically call `PMcheck()` to check errors in the data and fix if possible, with report

Minimal format



#ID	TIME	DOSE	OUT	WT
1	0	600	.	46.7
1	24	600	.	46.7
1	48	600	.	46.7
1	72	600	.	46.7
1	96	600	.	46.7
1	120	.	10.44	46.7
1	120	600	.	46.7
1	121	.	12.89	46.7
1	122	.	14.98	46.7
1	125.99	.	16.69	46.7
1	129	.	20.15	46.7
1	132	.	14.97	46.7
1	143.98	.	12.57	46.7
2	0	600	.	66.5
2	24	600	.	66.5
2	48	600	.	66.5

Standardized format



POPDATA DEC_11

#ID	EVID	TIME	DUR	DOSE	ADDL	II	INPUT	OUT	OUTEQ	C0	C1	C2	C3
1	1	0	0	600	.	.	1	.	.	.	.	.	.
1	1	24	0	600	.	.	1	.	.	.	.	.	.
1	1	48	0	600	.	.	1	.	.	.	.	.	.
1	1	72	0	600	.	.	1	.	.	.	.	.	.
1	1	96	0	600	.	.	1	.	.	.	.	.	.
1	0	120	.	.	.	.	.	10.44	1	0.02	0.0506	-0.0002	0
1	1	120	0	600	.	.	1	.	.	.	.	.	.
1	0	121	.	.	.	.	.	12.89	1	0.02	0.0506	-0.0002	0
1	0	122	.	.	.	.	.	14.98	1	0.02	0.0506	-0.0002	0
1	0	125.99	.	.	.	.	.	16.69	1	0.02	0.0506	-0.0002	0
1	0	129	.	.	.	.	.	20.15	1	0.02	0.0506	-0.0002	0
1	0	132	.	.	.	.	.	14.97	1	0.02	0.0506	-0.0002	0
1	0	143.98	.	.	.	.	.	12.57	1	0.02	0.0506	-0.0002	0
2	1	0	0	600	.	.	1	.	.	.	.	.	.
2	1	24	0	600	.	.	1	.	.	.	.	.	.

#ID - mandatory

- Subject ID
- alphanumeric
- ≤11 characters
- ≤800 subjects / run

#EVID - optional

- EVent ID
- 0 = observation, i.e. concentration
- 1 = input, i.e. dose
- 2,3 are unused
- 4 = reset, all compartments are reset to 0, and time counter to 0, generating a new “block” in many Pmetrics objects

TIME - mandatory

- Elapsed decimal time, e.g. 1.5 or 2.34 or clock time e.g. 13:00.
  - If clock time, DATE must be present
- All subjects must start with TIME=0 if no DATE column
- All rows should have TIME entry
- Events for a subject should be sorted by TIME

DATE - optional

- If TIME is clock time, DATE must be present
- DATE and TIME formats are coerced if possible to YYYY-MM-DD and HH:MM by PM_data$new()

DUR - optional

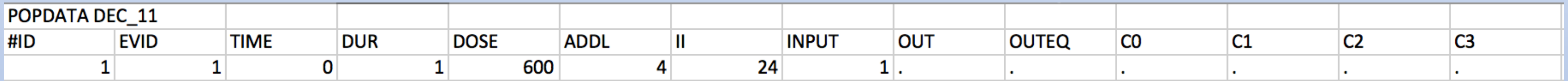
- DURation of an infusion for a dose (EVID=1) in decimal time, e.g. 2 or 1.5
- Only required for doses
- 0 = a bolus dose, given to compartment defined in model file
- >0 = infusion, given to compartment defined in model file
- Assumed to be 0 for all doses if missing

DOSE - mandatory

- What more can I say?

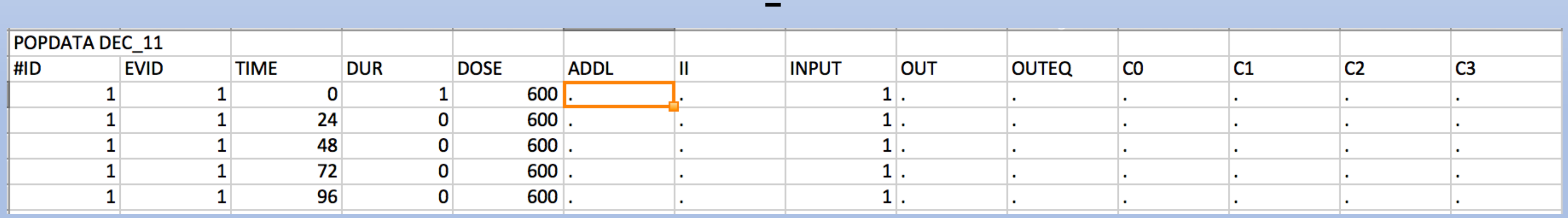
ADDL - optional

- For a given dose line, this is the number of ADDitionaL doses to be given at inter-dose interval (II)
- This shortcut is optional for EVID=1 and ignored for everything else



POPDATA DEC_11													
#ID	EVID	TIME	DUR	DOSE	ADDL	II	INPUT	OUT	OUTEQ	C0	C1	C2	C3
1	1	0	1	600	4	24	1	.	.	.	.	.	.

=



POPDATA DEC_11													
#ID	EVID	TIME	DUR	DOSE	ADDL	II	INPUT	OUT	OUTEQ	C0	C1	C2	C3
1	1	0	1	600	.	.	1	.	.	.	.	.	.
1	1	24	0	600	.	.	1	.	.	.	.	.	.
1	1	48	0	600	.	.	1	.	.	.	.	.	.
1	1	72	0	600	.	.	1	.	.	.	.	.	.
1	1	96	0	600	.	.	1	.	.	.	.	.	.

Additional ADDL

- Setting ADDL=-1 on the first dose at TIME=0, or a dose with EVID=4 simulates steady state conditions
- All compartments will be in a condition at the end of a steady state dosing interval of II

II - optional

- Interdose Interval is the interval in TIME units between ADDL doses
- Ignored if EVID ≠ 1 or ADDL is missing

INPUT - optional

- Defines the input, e.g. drug, number
- Only required for EVID=1
- Typically is 1 but could be >1 if multiple drugs are modeled simultaneously
- Assumed to be 1 if missing

OUT - mandatory

- OUTput, observation, concentration, etc.
- Only required for observations
- If an observation is missing, use -99, not “.”

OUTEQ - optional

- Output number for a corresponding OUT value
- Only required for observations
- Compartment that contains the output is defined in the model file
- The number is the number of the output equation in the model file
- Assumed to be 1 if missing

C0, C1, C2, C3 - optional

- Assay error coefficients
- Assay Error Polynomial (AEP): SD = $C_0[\text{out}]^0 + C_1[\text{out}]^1 + C_2[\text{out}]^2 + C_3[\text{out}]^3$
- If missing, model file values are used

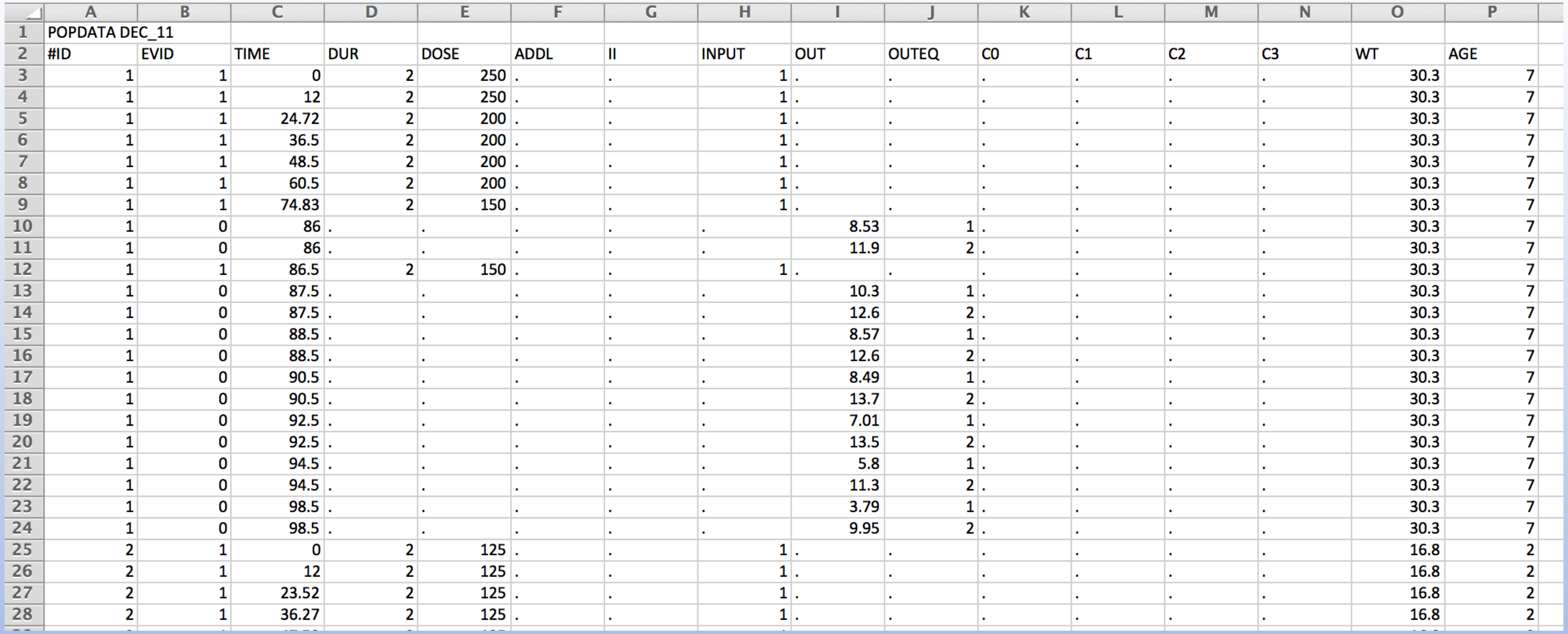
Covariates - optional

- Any column after C3 is a covariate
- Must be numeric
- Must have a value for every covariate at first time event for each subject, e.g. TIME=0 if DATE missing
- May have missing values after that; by default these missing values are linearly interpolated between bounding non-missing values

Covariates

- Covariates are applied at the time of doses
- Covariate values on observations are ignored
- To enter a changed covariate at a time other than a dose, use an dose record, with 0 DUR, 0 DOSE and the new covariate value

Example



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	POPDATA DEC_11															
2	#ID	EVID	TIME	DUR	DOSE	ADDL	II	INPUT	OUT	OUTEQ	C0	C1	C2	C3	WT	AGE
3	1	1	0	2	250	.	.	1	.	.	.	.	.	.	30.3	7
4	1	1	12	2	250	.	.	1	.	.	.	.	.	.	30.3	7
5	1	1	24.72	2	200	.	.	1	.	.	.	.	.	.	30.3	7
6	1	1	36.5	2	200	.	.	1	.	.	.	.	.	.	30.3	7
7	1	1	48.5	2	200	.	.	1	.	.	.	.	.	.	30.3	7
8	1	1	60.5	2	200	.	.	1	.	.	.	.	.	.	30.3	7
9	1	1	74.83	2	150	.	.	1	.	.	.	.	.	.	30.3	7
10	1	0	86	.	.	.	.	.	8.53	1	.	.	.	.	30.3	7
11	1	0	86	.	.	.	.	.	11.9	2	.	.	.	.	30.3	7
12	1	1	86.5	2	150	.	.	1	.	.	.	.	.	.	30.3	7
13	1	0	87.5	.	.	.	.	.	10.3	1	.	.	.	.	30.3	7
14	1	0	87.5	.	.	.	.	.	12.6	2	.	.	.	.	30.3	7
15	1	0	88.5	.	.	.	.	.	8.57	1	.	.	.	.	30.3	7
16	1	0	88.5	.	.	.	.	.	12.6	2	.	.	.	.	30.3	7
17	1	0	90.5	.	.	.	.	.	8.49	1	.	.	.	.	30.3	7
18	1	0	90.5	.	.	.	.	.	13.7	2	.	.	.	.	30.3	7
19	1	0	92.5	.	.	.	.	.	7.01	1	.	.	.	.	30.3	7
20	1	0	92.5	.	.	.	.	.	13.5	2	.	.	.	.	30.3	7
21	1	0	94.5	.	.	.	.	.	5.8	1	.	.	.	.	30.3	7
22	1	0	94.5	.	.	.	.	.	11.3	2	.	.	.	.	30.3	7
23	1	0	98.5	.	.	.	.	.	3.79	1	.	.	.	.	30.3	7
24	1	0	98.5	.	.	.	.	.	9.95	2	.	.	.	.	30.3	7
25	2	1	0	2	125	.	.	1	.	.	.	.	.	.	16.8	2
26	2	1	12	2	125	.	.	1	.	.	.	.	.	.	16.8	2
27	2	1	23.52	2	125	.	.	1	.	.	.	.	.	.	16.8	2
28	2	1	36.27	2	125	.	.	1	.	.	.	.	.	.	16.8	2

Pmetrics Model Files

Model Files

- Model files are at the heart of all Pmetrics functions
- You will spend most of your modeling time editing these model files
- Simple text documents
- Pmetrics expects them to have .txt extension

Model Files

- Contain up to 11 blocks marked by “#” and up to the first 3 letters of the block name
  - #PRImary variables Required
  - #COVariates
  - #SECondary variables
  - #BOLus inputs
  - #INItial conditions
  - #F (bioavailability)
  - #LAG time
  - #DIFferential equations
  - #OUTputs Required
  - #ERRor Required
  - #EXTra

#PRI

- The model parameters whose distributions are to be estimated
- Between 2-32 total may be specified, with 30 random and 20 fixed
- Random parameter format: *name, min, max*
- Fixed, unknown parameter format: *name, value*
- Fixed parameter format: *name, value!*💡
- Don’t use restricted words in names (see user manual)
- Once defined, primary variables can be used anywhere in the model file

💡 Hint: The “!” is used to fix the behavior of several model file blocks.

#PRI

- Behavior of the limits differs depending on the engine
  - NPAG - *min, max* are absolute
  - IT2B - *min, max* define the range, and the standard deviation of the prior is xsig*range, where xsig is an argument to `ITrun()`
  - Simulator - *min, max* are ignored unless limits argument to `SIMrun()` is `NA`, which uses min, max to truncate simulated parameter values

#PRI

- For IT2B only, adding a plus “+” to the parameter definition ensures that parameter values <0 will not be returned, even if the SD is sufficiently large to cross 0

#PRI Examples

#PRI
Ke, 0, 5
V, 0, 100+
Ka, 5
IC, 10!

#COV

- Declare the names of covariates
- Needs to match the names and order of covariates in the data file
- Covariates can be declared and not used (useful for covariate analysis later)
- Once defined, covariates can be used anywhere in the model file

#COV

- Default behavior is to linearly interpolate missing covariate values in the data file
- Use an “!” in the covariate declaration to change to piece-wise constant, i.e. carry forward previous value until new value is encountered

#COV Examples

#COV
wt
cyp
IC!

#SEC

- Define secondary variables
- Parameter value distributions are NOT generated for secondary variables; they are used only internally within the model file
- Once declared, secondary variables can be used anywhere in the model file

#SEC

- Need to be of the form X = Y
- For conditional variables or to include fortran code, preface lines with “&”

#SEC Examples

```
#Sec
CL = Ke * V * wt**0.75
& IF(cyp .GT. 1) CL = CL * cyp
```

#BOL

- By default, DUR=0 bolus inputs go into *compartment=input*, i.e. input 1 into compartment 1, and input 2 into compartment 2, etc.
- Use this block to override that default
- NBCOMP(n) = x, where n=input number, and x=compartment number

```
#BOL
NBCOMP(1)=2
```

#INI

- Set initial conditions of a compartment to be a non-zero amount
- Initial conditions are functions of fixed values, primary variables, secondary variables, and/or covariates, and conditional expressions may be used
- X(n), where n = compartment number

```
#INI
X(2) = IC*V
X(3) = IC3
X(4) = 50000
& IF(MALE==1) X(5) = 20
```

#F

- Define bioavailability (FA, fraction available) for a drug in a dataset with bolus and infusion dosing
- Bioavailability terms are functions of fixed values, primary variables, secondary variables, and/or covariates, and conditional expressions may be used
- FA(n), where n=input number

```
#F
FA(1) = F1
& IF(T>T0) FA(1) = F2
```

#LAG

- Define lag time for an input, i.e. the delay between bolus dose and beginning of absorption
- Not relevant for infusions
- Lag times are functions of fixed values, primary variables, secondary variables, and/or covariates, and conditional expressions may be used
- TLAG(n), where n=input number

```
#LAG
TLAG(1) = Lag1
& IF(DR==1) TLAG(1) = Lag2
```

#DIF

- Specify a model using differential equations in fortran format
  - XP(n) = dXn/dt, where n = compartment
  - X(n) = amount in compartment n
  - Use RATEIV(n) to place infusions for input n
  - Use any variable defined in #PRI, #SEC or any covariate
- Max of 20 equations

#DIF Example

```
#Dif
XP(1) = -KA*X(1)
XP(2) = RATEIV(1) + KA*X(1) - (KE+KCP)*X(2) + KPC*X(3)
XP(3) = KCP*X(2) - KPC*X(3)
```

#OUT

- Define the output equations as functions
- Y(n) = *function*, where n is 1, 2, 3,…
- n corresponds to OUTEQ in the data file
- *function* is comprised of fixed values, primary variables, secondary variables, and/or covariates, and conditional expressions may be used

#OUT Example

```
#Out
Y(1) = X(2)/V
```

#ERR

- Defines the error model to be used
- Two choices
  - error = (SD * gamma)^2
  - error = (SD + lambda)^2
- Recall SD = $C_0[\text{out}]^0 + C_1[\text{out}]^1 + C_2[\text{out}]^2 + C_3[\text{out}]^3$

#ERR

- First specify L/G = x, where x is the starting value for lambda or gamma
  - Lambda is not available in IT2B
  - Add an “!” to fix and not estimate; lambda can only be fixed to 0
- Then specify C0, C1, C2, C3 for each output equation in #OUT
  - Add an “!” to use these values regardless of those in data file; otherwise, datafile values will be used unless missing

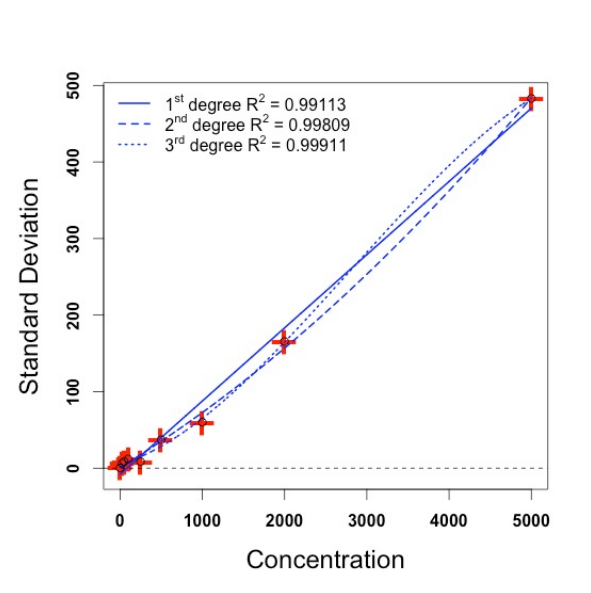
#ERR Examples

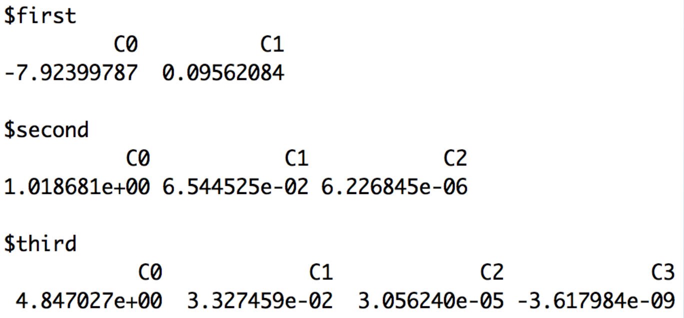
```
#ERR
L=0.4
0.1,0.1,0,0

#Err
G=2!
0.1,0.1,0,0
0.3,0.1,0,0!
```

Estimating Cs

1. Use assay validation data and `makeErrorPoly()`

Standard	SD
0	0.5
25	6.4
50	8.6
100	12
250	8.6
500	37.2
1000	60.1
2000	165.7
5000	483





```
$first
         C0          C1
-7.92399787  0.09562084

$second
          C0           C1           C2
1.018681e+00 6.544525e-02 6.226845e-06

$third
           C0            C1            C2            C3
 4.847027e+00  3.327459e-02  3.056240e-05 -3.617984e-09
```

```
obs <- c(0,25,50,100,250,500,1000,2000,5000)
sd <- c(0.5,6.4,8.6,12,8.6,37.2,60.1,165.7,483)
makeErrorPoly(obs=obs,sd=sd)
```

Estimating Cs

2. Estimate
   - C0 = 0.5 * LOQ or lowest concentration in data
   - C1 = 0.1 (assay with 10%CV)
   - C2 = 0
   - C3 = 0

Estimating Cs

- Use `ERRrun()`
- Similar to IT2B
- Model dependent, which makes this the least preferable way to estimate Cs
- Only with gamma, which will always be 1

Gamma/Lambda

- No particular rule on which to choose
- Interpreting Gamma
  - <<1…assay error polynomial likely too large; check to make sure C0 is in same units as concentration
  - 1-5…model OK, data OK
  - 5-10…model misspecification or data noisy, e.g. TDM data with errors in dose or concentration times
  - >10…severe model misspecification

Gamma/Lambda

- Interpreting Lambda
  - 0-3 x C0…model OK, data OK
  - 3-5 x C0…model misspecification or data noisy, e.g. TDM data with errors in dose or concentration times
  - >5 x C0…severe model misspecification

Gamma/Lambda

- Starting values
  - Gamma: 1 to 3
  - Lambda: C0 to 3xC0
- Increase if run ends with Hessian Error or “Probability of all points, given data for subject xx, is 0”
  - However, your model is likely misspecified (check model file very carefully) or structurally a bad choice (e.g. try adding another compartment)

#EXT

- Add Fortran subroutines for very advanced, customized model files

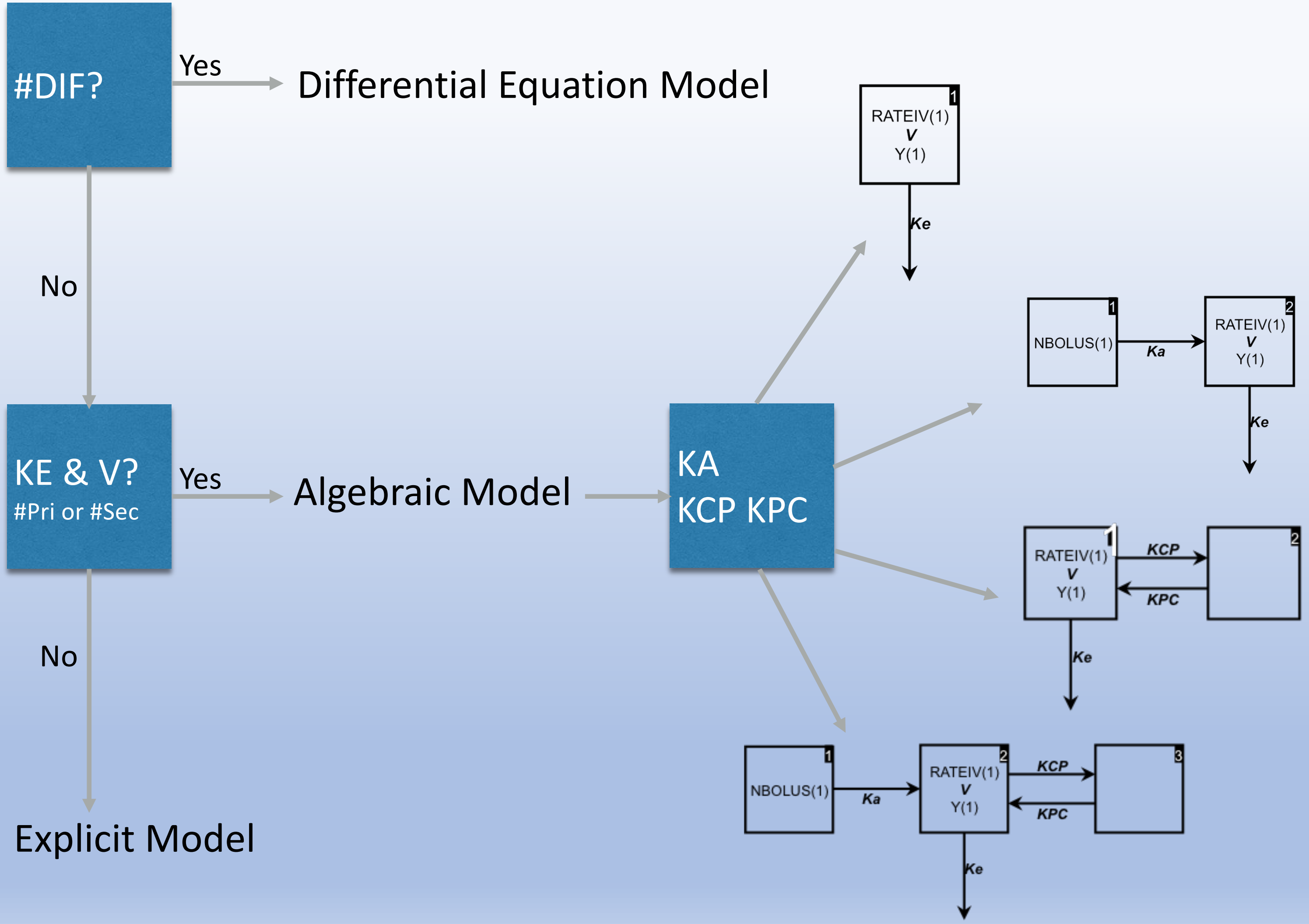
Comments

- Model files may contain comments
- Comment lines start with “C ”
- e.g. “C This weight is in kg”

Common model errors

- Forgetting ranges in #PRI block
- Forgetting covariates or not listing covariates in correct order in #COV block
- Forgetting “&” before lines in #SEC block that are not of the form A = B
- Not having minimum of #PRI, #OUT, #ERR in every model file
- Using a reserved name for a variable (see user manual)
- Mistakes in #DIF block
- Not matching number of output equations, Y(n) with maximum OUTEQ in data file
- Not matching number of C0,C1,C2,C3 in #ERR block to number of output equations

Model File Logic



#DIF?

Yes

Differential Equation Model

No

KE & V?

#Pri or #Sec

Yes

Algebraic Model

KA

KCP KPC

No

Explicit Model

Model library

- A basic model library is available on our website with downloadable files
- It is not exhaustive. The idea is to provide building blocks for more complicated models.
- http://www.lapk.org/ModelLib.php