

Overview

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

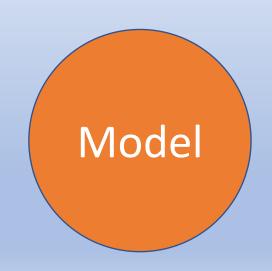




Pmetrics basics

- Pmetrics allows you to fit a model to data
- A model is a series of equations relating input to output
- Population modeling means estimating the probability distribution of the values of the model variables (parameters) in your study population.

Data



Create PM_data object

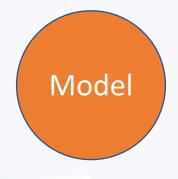
- Data should be in delimited format
 - Default is csv, but can be others using setPMoptions()
 - At minimum needs to have these columns:
 - ID subject ID
 - Time relative time in decimal hours for each event, starting at 0 for each ID
 - Dose input amount
 - Out output value

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

Create PM_data object

- Ensure that R is pointing to the same folder the data file is in
- > setwd(...)
- Create the data object by reading the file into memory for use in R
- > data1 <- PM_data\$new("myFile.csv")

Create PM_model object



- Read a .txt file
 - Easier format
 - Working directory or path to file must be specified
- Define directly in R
 - Harder format
 - Model definition and details easily visible when working in R
 - Updating/changing therefore easier

```
#Pri
Ka, 0.1, 0.9
Ke, 0.001, 0.1
V, 30, 120
Tlag1, 0, 4
#Cov
\mathsf{WT}
AFRICA
AGE
GENDER
HEIGHT
#Lag
TLAG(1) = Tlag1
#Out
Y(1) = X(2)/V
#Err
G=5
0.02, 0.05, -0.0002, 0
```

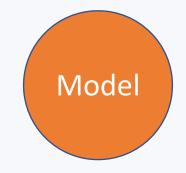


Create PM_model object

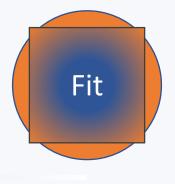


- Ensure that R is pointing to the same folder the model file is in
- > setwd(...)
- Create the data object by reading the file into memory for use in R
- > mod1 <- PM_model\$new("myModel.txt")

```
mod1 <- PM_model$new(list(
 pri = list(
  Ka = range(0.1, 0.9),
  Ke = range(0.001, 0.1),
  V = range(30, 120),
  Tlag1 = range(0, 4)
 cov = list("WT", "AFRICA", "AGE", "GENDER", "HEIGHT"),
 lag = list("Tlag(1) = Tlag1"),
 out = list(
  Y1 = list(
   value = X(2)/V,
   err = list(
     model = proportional(5),
     assay = c(0.02, 0.05, -0.0002, 0)
```

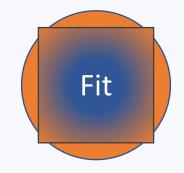


PM_fit



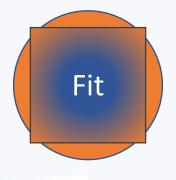
• Bring the data and model together in a PM_fit object, ready to run.

> fit1 <- PM_fit\$new(data1, mod1)



```
> fit1
<PM_fit>
 Public:
  check: function ()
  clone: function (deep = FALSE)
  initialize: function (data = data, model = model, ...)
  run: function (..., engine = "npag")
  save: function (file_name = "PMfit.rds")
 Private:
  data: PM_data, R6
  engine: NPAG
  model: PM_model_list, PM_model, R6
```

PM_fit



- Execute the run
- > fit1\$run()
- No arguments required
- Does not need to be assigned to a variable
 - Results saved directly to hard drive

PM_load()

• Grab the results from your hard drive to make them available in R

```
> res1 <- PM_load(1)
```

```
> res1
<PM result>
 Public:
  clone: function (deep = FALSE)
  cov: PM cov, R6
  cycle: PM_cycle, R6
  data: PM data, R6
  errfile:
  final: NPAG, PM_final, R6
  initialize: function (out, quiet = T)
  make_valid: function (...)
  MM opt: function (...)
  model: PM_model_file,
  PM model list, PM model, R6
  nca: function (...)
  npdata: NPAG
```

```
op: PM op, R6
plot: function (type, ...)
pop: PM_pop, R6
post: PM post, R6
save: function (file_name = "PMresult.rds")
sim: function (...)
step: function (...)
success: TRUE
summary: function (type, ...)
valid: NULL
```