METRUM
RESEARCH GROUP

Please grab some lunch and eat here or in the next room.

We'll start the workshop at 12:30.

https://demo.metworx.com

metworx

Enter your organization's fingerprint. If you don't have a fingerprint, select "No Fingerprint".

f1e3fd4a41

NO FINGERPRINT

SUBMIT

Sign in with your email and password

Email

userxy@metrumrg.com

Password

••••••••••

Sign in

metworx

Dashboard | My Info | Sign Out | Support | Contact

My Active Workflows 1 of 3 Active

kylebworx

UPDATE

SHUT DOWN

Created: 3 days ago

Blueprint: metworx-22.09 - ACOP

Disk:

Status: Running

RSTUDIO | RSCONNECT | GUACAMOLE

Head Node 2 vCPU / 8 GB RAM (~$0.12/hr) GB

PERFORMANCE

Cluster Nodes 2 vCPU / 4 GB RAM (~$0.09/hr) 0 nodes (of 100) 0 pending jobs

Start a New Workflow

metworx-22.09 - ACOP LAUNCH

Workflow Info

All Workflows are launched with the following software pre-installed: RStudio Workbench, RStudio Connect, RStudio Desktop, R, NONMEM, PsN, Python, Pirana, Monolix Suite, Matlab, Git, Apache Subversion, pkgr, GNU Fortran, TexLive (LaTex), Ubuntu (OS), NginX, and Sun Grid Engine

MetrumRG Account Info

As part of your Metworx account, you can have up to 3 Workflows actively running at any given time

© 2022 Metrum Research Group | acop-training | DEVELOPMENT

Schedule

12:00 pm - 12:30 pm Lunch and login

12:30 pm - 1:00 pm Introductions and introduction of the ecosystem (Matt)

1:00 pm - 1:45 pm Model Output (Kyle, Sam)

- Introduction to yspec and pmplots
- Model Diagnostics (bbr, yspec, pmplots)
- Reporting templates using Rmarkdown

1:45 pm - 2:30 pm Hands-on examples with yspec and pmplots (Kyle, Sam)

2:30 pm - 2:45 pm Break

2:45 pm - 3:30 pm Model management with bbr (Seth)

3:30 pm - 4:15 pm Hands-on examples with bbr (Seth)

4:15 pm - 5:00 pm Q&A (everyone)

Introductions

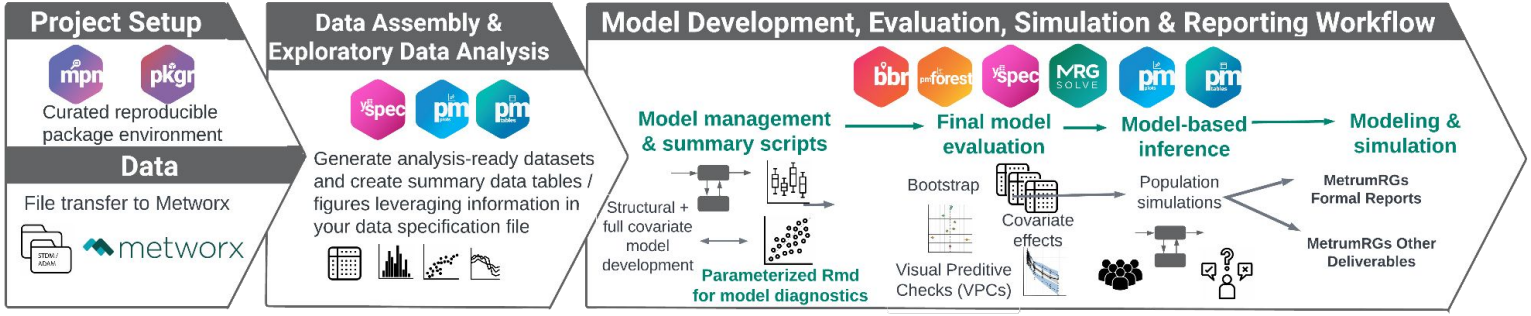
- Matthew Riggs – Chief Science Officer
- Seth Green – Manager of Data Science Engineering
- Kyle Baron – Principal Scientist II, Scientific Advisor to PKPD
- Sam Callisto – Research Scientist

MeRGE

Metrum **R**esearch **G**roup **E**cosystem

METRUM RESEARCH GROUP

We introduce MeRGE through a **user-friendly Expo** that showcases a suite of tools in the context of a simplified population pharmacokinetic (PPK) model. It demonstrates how to proceed step-by-step through a PPK modeling and simulation (M&S) analysis, using the same process and suite of tools that we use at Metrum Research Group, to ensure traceable and reproducible pharmacometrics research.



How ACoP12 Motivated MeRGE…

ACoP12 Preconference: Integrating Standardization and Innovation in Your Organization: Find a Workflow That Works for You!

Preconference chairs: Jace Nielsen, Chris Penland, Mike K. Smith, Stacey Tannenbaum

What is a “workflow”?

What’s the worst that could happen?

- What are the dangers of not having a workflow?
- What are the scenarios that reveal weak points in your workflows?
- How do I know what I am missing? Identifying blindspots in your workflow.

Making design choices in your workflow

- What’s in the toolbox? Environments, software, scripts already available in PMX
- …

Why MeRGE?

Support traceable, reproducible, and scalable pharmacometric analyses

Example 1: working on a project with a team … consistency, efficiency

Example 2: working with stakeholders, I'd like to give them an update … consistency, expectations, ease of communications

Example 3a: work done X years ago, new reg submission and we need to recreate (or update) an analysis

Example 3b: work done X years ago, new/additional will continue the work, we need to be able to follow what was done (how and why) to continue…

Why MeRGE?

Support traceable, reproducible, and scalable pharmacometric analyses

Example 1: working on a project with a team … consistency, efficiency

For example, tables can be VERY time-consuming to make, especially in a traceable manner. The look and content of tables can vary considerably when made by different individuals, or even by the same person at different times!

yspec + pmtables makes this MUCH easier!!

Same goes for figures:
yspec + pmplots + mrgsolve + pmforest also makes this MUCH easier!!

Why MeRGE?

Support traceable, reproducible, and scalable pharmacometric analyses

Example 2: working with stakeholders, I'd like to give them an update … consistency, expectations, ease of communications

"Hey Matt, explain to me why you chose a two vs a one compartmental model? And what about that variance structure, are we certain that we have appropriate random effects for IIV and residual variability?"

bbr + yspec + pmtable + pmplots + Rmarkdown makes this easy!!

Why MeRGE?

Support traceable, reproducible, and scalable pharmacometric analyses

Example 3a: work done 3 years ago, new reg submission and we need to recreate (or update) an analysis

"Hey Kyle, remember that empagliflozin work we did a few years ago [1]; we need to some more work to bridge into T1DM [2]…"

mpn + pkgr + yspec + bbr + pmtable + pmplots + Rmarkdown (would have) made this easy (ier)!!

[1] https://link.springer.com/article/10.1007/s13300-016-0174-y
[2] https://accp1.onlinelibrary.wiley.com/doi/abs/10.1002/jcph.1051

Why MeRGE?

Support traceable, reproducible, and scalable pharmacometric analyses

Example 3b: work done 3 years ago, the people that did the work are not available, we need to track down what they did to continue on for a new indication…

“Hey Curtis, guess what, we need you to do some work on the empagliflozin program…”

Why MeRGE?

Support traceable, reproducible, and scalable pharmacometric analyses

In summary … without specifically thinking about it, MetrumRG has been working on developing MeRGE for years (through many cross-functional teams) … this has come together, through an evolution, to form our ecosystem. Inspired by the ACoP12 precon, we realize that there's a need beyond our individual work, so we want to share what we've developed with you, the PMx community.

(audience discussion?)

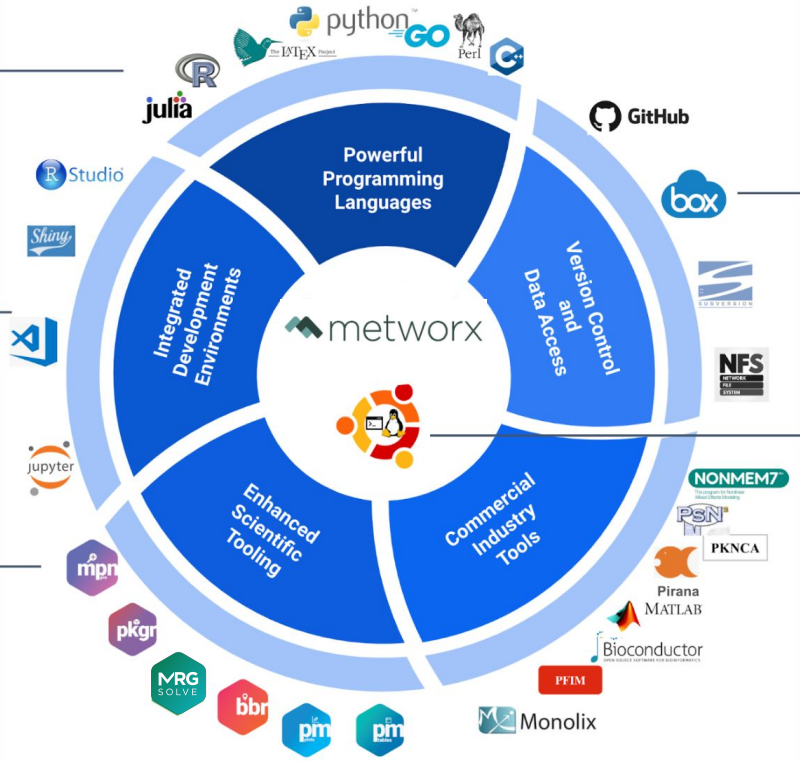
Pull in learnings from last years ISoP pre-con??

The Quantitative Decision Support Ecosystem

Powerful Programming
Explore and deliver with raw access to proven languages to optimize productivity and performance.

Familiar Development Environments
R, Python, Julia - teams can develop in familiar IDEs and easily launch interactive dashboards and scientific applications with Shiny.

Better Reproducibility, Reduced Risk, Enhanced Efficiencies
Immutable library and dependency snapshots, solvers, package management, publication-ready tables and figures and an easier life with NONMEM.



Versioning / Data Access
Integrate with networked data repositories whether name-brand, on-premise or as a managed service.

Powerful Operating System
Metworx offers a complete operating system at its core. access to rock-solid utilities and services already familiar to IT. Access options included web, remote desktop, command line and more.

Metworx is a secure, highly-scalable, cloud-native Platform-as-a-Service that brings reproducible tools and computing to scientific teams of all sizes.

metworx

Value	How	Metworx 4.0 *coming enhancements*
Scientific Excellence Built-In	• MIDD at its core • Designed, maintained and guided by the scientific excellence of MRG • Comes with industry-leading tools and technologies	• Inclusion of best-practice examples via MeRGE
Reproducible / Traceable	• Rapid validation • Consistent, controllable state of compute environments • MPN: Immutable snapshots of packages and dependencies for long-term reproducibility	• Tighter integration with MPN • Inclusion of best-practice examples via MeRGE • Enhanced audit trails
Scaleable	• No shared clusters • Each workflow is its own scalable grid • Allows multiple workflows per user • Fast ramp-up across large, distributed teams	• Improved visibility and cost-efficiency of cloud resources • More controls across large groups with different usage needs
Security	• Secure data and compute isolation • Client-controlled permissions • SSO integration	• Enhanced user/group administration

Introduction to the Ecosystem

Untitled1

Source on Save | Run | Source

```
1
print("There is much more to MeRGE than an Rstudio prompt. How do we
bring this to life? We'll show you today and give you a basic pop PK
modeling project as an example.")
```

1:1 (Top Level) R Script

Console | Terminal | Jobs | Launcher

R 4.1.1 · ~/

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Introduction to the Ecosystem

Untitled1

Source on Save | Run | Source

```
1
print("in addition to these slides, we have a site: https://merge.metrumrg.com/expo/expo1-nonmem-foce/index.html that we'd love for you to use — this site is for you and for you to help us – feedback is encouraged and welcomed!")
```

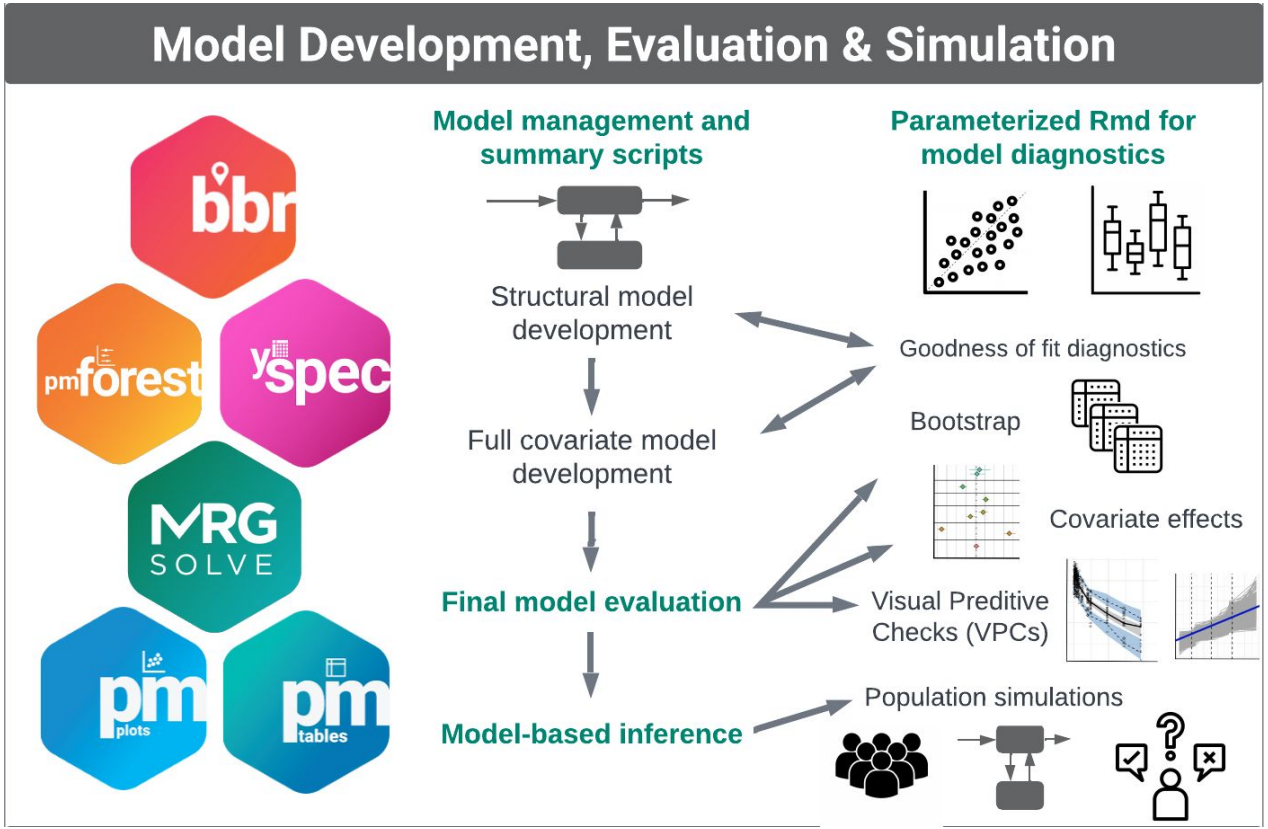
1:1 (Top Level) R Script

Console | Terminal | Jobs | Launcher

R 4.1.1 · ~/

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

A pop PK workflow using NONMEM

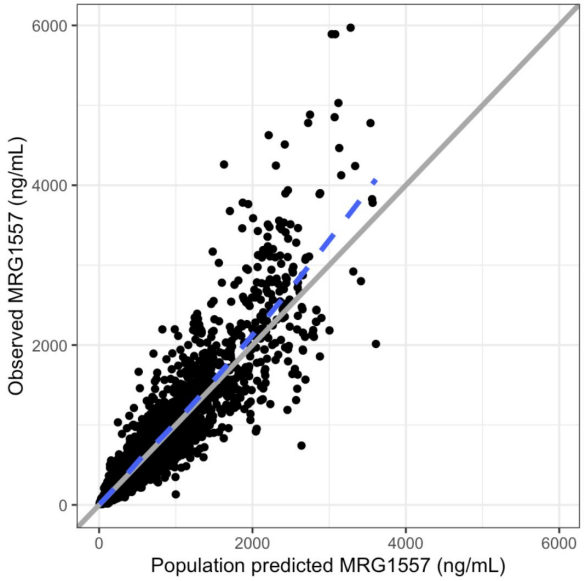


METRUM RESEARCH GROUP

Introduction to pmplots


```
dv_pred(df, yname = .yname)
```

- Standardized plots
  - Exploratory
  - Diagnostic
- Simple / efficient syntax
- Expects standard inputs
  - TIME
  - DV
  - PRED
  - IPRED
  - CWRES
- Batch processing
- "Enough" customization
- Not a new grammar of graphics



METRUM RESEARCH GROUP

Introduction to pmplots

Conditional weighted residuals versus time

```
p1 <- cwres_time(data)
```

Residuals versus population predicted value

```
p2 <- res_pred(data)
```

NPDE boxplots in each study

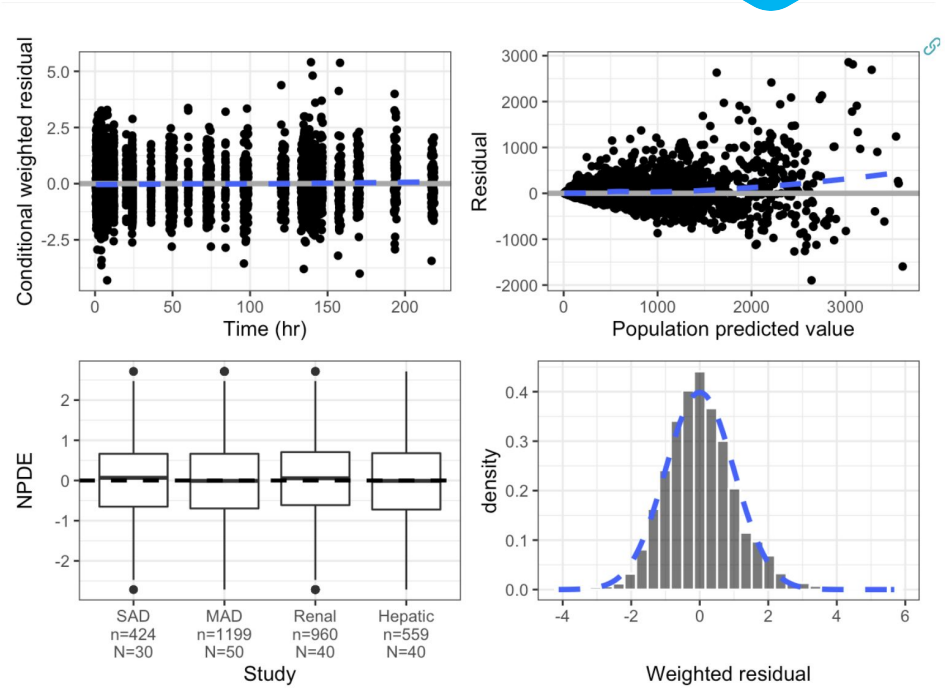
```
p3 <- npde_cat(data, x = "STUDYc//Study")
```

Histogram of weighted residuals

```
p4 <- wres_hist(data)
```

With output

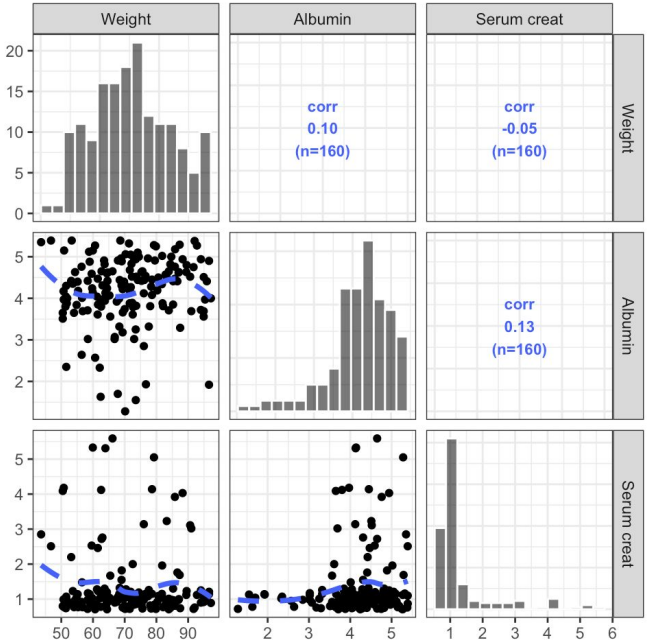
```
(p1+p2)/(p3+p4)
```



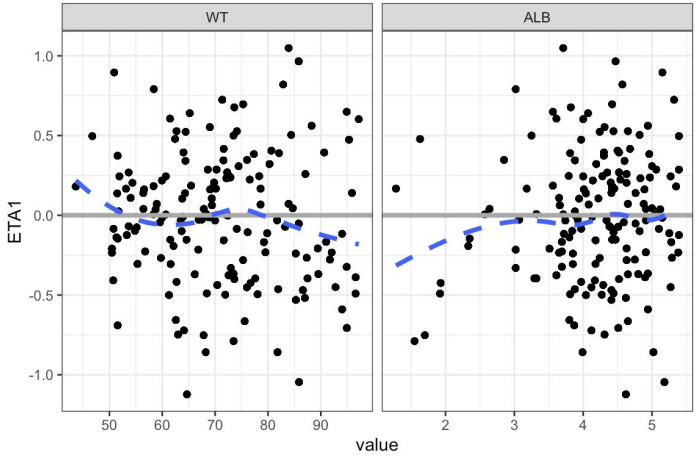
METRUM RESEARCH GROUP

Introduction to pmplots


```
cols <- c("WT//Weight", "ALB//Albumin", "SCR//Serum creat")
pairs_plot(id, cols)
```




```
wrap_eta_cont(
  df,
  y = "ETA1",
  x = c("WT", "ALB"),
  scales = "free_x"
)
```



METRUM RESEARCH GROUP

pmplots gallery

https://metrumresearchgroup.github.io/pmp-book/

The pmplots Gallery

Related

The pmplots Gallery

Plots for Pharmacometrics

AUTHOR
Kyle Baron, Pharm.D., Ph.D.

PUBLISHED
Jun 23, 2022

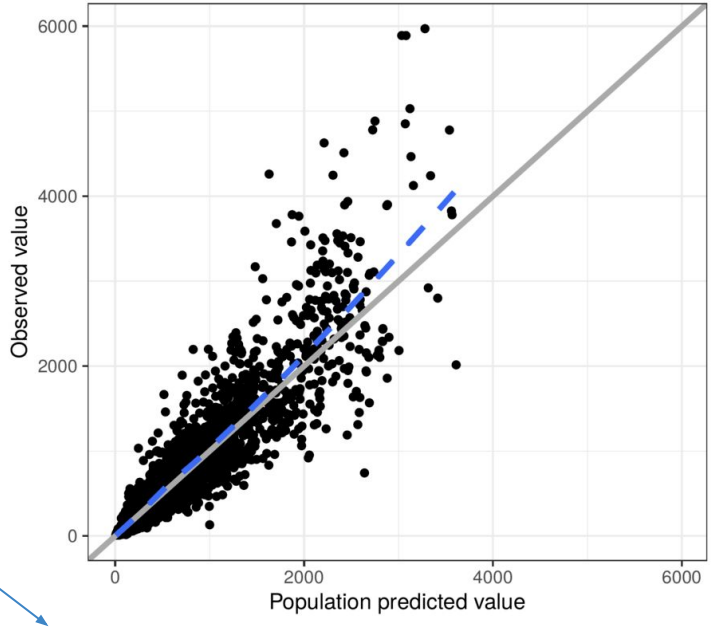
This is a simple introduction to the pmmplots package for R. I hope this will be useful for those who are new to the package and those who just need a reminder on the syntax. The goal with this package isn't to create a new grammar of graphics, but rather to create a standard set of commonly-used plots in pharmacometrics analyses.

This is truly intended to be a Gallery. In some chapters, you will see a great deal of repetion in plots (like CWRES versus TIME, WRES versus TIME, RES versus TIME). This is by design with the intention to make the reader aware of the different functions available in the package. One exception to this is the page on customization. Please take a moment to look through this page; it is long but you will find some very helpful examples of what you can do with pmplots.

You can find `documentation` for pmplots here.

mrggsave - save annotated images


```
mrggsave(p, stem = "intro-1", dir = tempdir(), script = "mrggsave.Rmd")
```



Source code: mrggsave.Rmd
Source graphic: intro-1.pdf

- Annotation
  - Source code file name
  - Image output file name
- Save lists of plots
- Interpolate variables into file names
- Save to multiple devices
  - pdf, png, both …
- Set height and width with sensible defaults

mrggsave - save lists of plots


```
run <- 101

p <- list(
  dv_pred(
  npde_tir
  cwres_h:
)

ans <- mrg

basename(a
```

```
[1] "diagn
[3] "diagn
```

```
p <- list(
  `dv-v-pr
  `npde-v-
  `cwres-h
)

ans <- mrg

basename(a
```

```
[1] "dv-v-p
[3] "cwres-
```

```
p <- named
  dv_pred(
  npde_tir
  cwres_h:
)

ans <- mrg

basename(a
```

```
[1] "dv-pr
```

```
dv_versus_pred <- dv_pred(data)

p <- named_plots(dv_versus_pred)

ans <- mrggsave(p, tag = run, dev = "png", use_names = TRUE)

basename(ans)
```

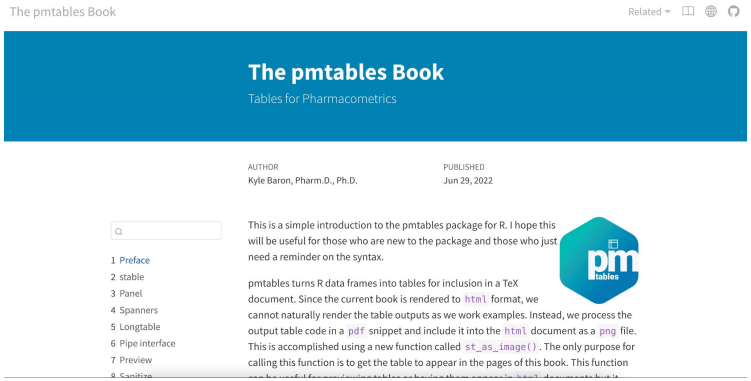
```
[1] "dv-versus-pred-101.png"
```

pmtables - tables for latex

	Study				
Statistic	12-DEMO-001 n = 30	12-DEMO-002 n = 50	11-DEMO-005 n = 40	13-DEMO-001 n = 40	Summary n = 160
Weight					
Mean (SD)	72.2 (14.3)	72.4 (11.5)	68.9 (14.5)	69.4 (11.6)	70.7 (12.8)
Min / Max	50.9 / 97.2	51.5 / 96.6	43.6 / 92.8	50.7 / 96.6	43.6 / 97.2
Missing	1	1	1	0	3
CRCL					
Mean (SD)	106 (9.46)	103 (8.35)	58.8 (29.7)	102 (8.19)	92.1 (25.5)
Min / Max	93.2 / 126	90.6 / 121	15.4 / 103	90.7 / 119	15.4 / 126
Missing	1	1	1	3	6
Sex					
male	10 (33.3)	18 (36.0)	29 (72.5)	23 (57.5)	80 (50.0)
female	20 (66.7)	32 (64.0)	11 (27.5)	17 (42.5)	80 (50.0)
Formulation					
tablet	25 (83.3)	42 (84.0)	30 (75.0)	33 (82.5)	130 (81.2)
capsule	3 (10.0)	6 (12.0)	3 (7.5)	3 (7.5)	15 (9.4)
troche	2 (6.7)	2 (4.0)	7 (17.5)	4 (10.0)	15 (9.4)

Categorical summary is count (percent)
n: number of records summarized
SD: standard deviation
Min: minimum; Max: maximum
Source code: _snippets.Rmd

```
pt_demographics(
  pmt_first,
  cols_cont = c(Weight = "WT", "CRCL"),
  cols_cat = c(Sex = "SEXf", Formulation = "FORM"),
  span = c(Study = "STUDYf")
) %>% stable() %>% st_as_image()
```



Introduction to yspec

- Documentation of analysis data sets
  - Write definitions in yaml format
  - Load into R as object
- Use along all phases of project work
  - Interactive query during DA
  - Generate define.pdf
  - Annotate plots and tables
  - Generate table for report

1 Datasets

Description	Location
Example PopPK analysis data set	analysis3.xpt

1.1 Example PopPK analysis data set (`analysis3.xpt`)

VARIABLE	LABEL	TYPE	CODES
C	comment character	character	C = comment, . = non-comment
NUM	record number	numeric	
ID	subject identifier	numeric	
TIME	time after first dose (unit: hour)	numeric	
SEQ	data type	numeric	0 = dose, 1 = observation
CMT	compartment number	numeric	
EVID	event ID	numeric	0 = observation, 1 = dose
AMT	dose amount (unit: mg)	numeric	
DV	dependent variable	numeric	
AGE	age (unit: years)	numeric	
WT	weight (unit: kg)	numeric	
HT	height (unit: cm)	numeric	
EGFR	estimated glomerular filtration rate (unit: mL/min/1.73m^2)	numeric	
ALB	albumin (unit: g/dL)	numeric	
BMI	BMI (unit: kg/m^2)	numeric	
SEX	SEX	numeric	0 = male, 1 = female

Coding data definitions in yaml fo


```yaml
WT:
  short: weight
  unit: kg
  range: [40, 140]

ARM:
  short: treatment arm
  type: character
  values: [200 mg qd, 400 mg qd]
  make_factor: true

FORM:
  short: formulation
  values: [0, 1]
  decode: [tablet, capsule]

STUDY:
  short: study number
  type: numeric
  values: [101, 102, 201]
  decode:
    - DRGX-55-101
    - DRGX-55-102
    - DRGX-66-201
```

Column name

Continuous data item

Categorical data items

Using a project-wide lookup file

Lookup file
(all definitions used on the project)

```yaml
WT:
  short: weight
  unit: kg
  range: [40, 140]

ARM:
  short: treatment arm
  type: character
  values: [200 mg qd, 400 mg qd]
  make_factor: true

FORM:
  short: formulation
  values: [0, 1]
  decode: [tablet, capsule]

STUDY:
  short: study number
  type: numeric
  values: [101, 102, 201]
  decode:
    - DRGX-55-101
    - DRGX-55-102
    - DRGX-66-201
```

In the PK file

```yaml
ARM: !look
STUDY: !look
FORM: !look
DV:
  short: concentration
  unit: ng/mL
```

In the AE file

```yaml
ARM: !look
STUDY: !look
DV:
  short: grade 4 thrombocytopenia
  values: {no: 0, yes: 1}
```

Load, preview and validate

Load

```r
data <- read_csv("my-data-file.csv")

spec <- ys_load("my-data-spec.yml")
```

Preview

```r
head(spec)
```

```
   name info         unit              short        source
1     C  cd-            .  comment character ysdb_internal
2   NUM  ---            .      record number ysdb_internal
3    ID  ---            . subject identifier ysdb_internal
4  SUBJ  c--            . subject identifier ysdb_internal
5  TIME  ---         hour               TIME          look
6   SEQ  -d-            .                SEQ             .
7   CMT  ---            . compartment number ysdb_internal
8  EVID  -d-            .           event ID ysdb_internal
9   AMT  ---           mg        dose amount ysdb_internal
10   DV  --- micrograms/L dependent variable ysdb_internal
```

Validate

```r
ys_check(data, spec, error_on_fail = FALSE)
```

```
Messages:
 - spec has more items than cols in the data
 - names in spec but not in data:
    - AAG

#----------------------------------------

[1] FALSE
```

Access data as list or through api

Query Continuous

```
spec$WT
```

```
name  value
col   WT
type  numeric
short weight
unit  kg
range 40 to 100
```

Query Categorical

```
spec$FORM
```

```
name  value
col   FORM
type  numeric
short formulation
value 0 : tablet
      1 : capsule
```

The yspec Book

https://metrumresearchgroup.github.io/ysp-book/

The ypsec Book

Related

The ypsec Book

Dataset specification for pharmacometrics

AUTHOR
Kyle Baron, Pharm.D., Ph.D.

PUBLISHED
Jun 21, 2022

yspec is an R package to help you manage and utilize documentation for analysis data sets of the kind that are frequently used in pharmacometrics. The `y` in yspec stands for `yaml`: data set definitions are written in a standard format using `yaml` language.

You can find `documentation` for yspec here.

Source

The yspec package is maintained here. The code for this book is maintained here.

lastdose - calculate time after dose


```
data
```

	ID	SUBJ	TIME	CMT	EVID	AMT	II	ADDL
1	1	1	0.00	1	1	5	6	23
2	1	1	0.61	2	0	0	0	0
3	1	1	1.15	2	0	0	0	0
4	1	1	1.73	2	0	0	0	0
5	1	1	2.15	2	0	0	0	0
6	1	1	3.19	2	0	0	0	0
7	1	1	4.21	2	0	0	0	0
8	1	1	5.09	2	0	0	0	0
9	1	1	6.22	2	0	0	0	0
10	1	1	8.09	2	0	0	0	0
11	1	1	12.03	2	0	0	0	0
12	1	1	20.07	2	0	0	0	0
13	1	1	24.20	2	0	0	0	0

```
lastdose(data)
```

	ID	SUBJ	TIME	CMT	EVID	AMT	II	ADDL	TAD	LDOS
1	1	1	0.00	1	1	5	6	23	0.00	5
2	1	1	0.61	2	0	0	0	0	0.61	5
3	1	1	1.15	2	0	0	0	0	1.15	5
4	1	1	1.73	2	0	0	0	0	1.73	5
5	1	1	2.15	2	0	0	0	0	2.15	5
6	1	1	3.19	2	0	0	0	0	3.19	5
7	1	1	4.21	2	0	0	0	0	4.21	5
8	1	1	5.09	2	0	0	0	0	5.09	5
9	1	1	6.22	2	0	0	0	0	0.22	5
10	1	1	8.09	2	0	0	0	0	2.09	5
11	1	1	12.03	2	0	0	0	0	0.03	5
12	1	1	20.07	2	0	0	0	0	2.07	5
13	1	1	24.20	2	0	0	0	0	0.20	5

https://github.com/metrumresearchgroup/lastdose

Model Diagnostics - Parameterized Reports

Report diagnostics

Purpose

To produce a set of diagnostic plots that will be included in the report. Please note that these plots are just meant to provide an example of what could be created and how. They are not an exhaustive list of every possible plot and were chosen with the project aims in mind.

While this *should* give users examples of plots generated with the most up-to-date packages and methods, we're always happy to have feedback. If you know of more efficient methods or want to suggest alternative ways of plotting the figures please open an issue with the details.

Set up

Model location

Define `modelName` and path to the model directory (`MODEL_DIR`).

Figure location

If saving figures out to pdf, define where those pdfs should be saved to. Here the figures are saved to `deliv > figure > model_run_number`

Model Diagnostics - Parameterized Reports

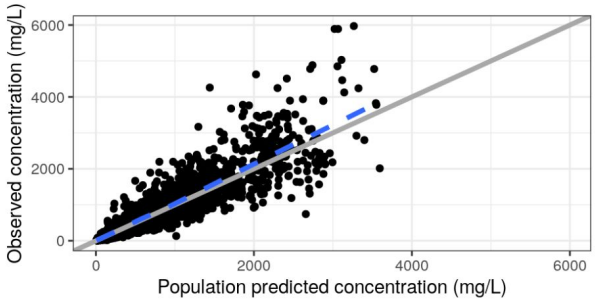
General diagnostic plots

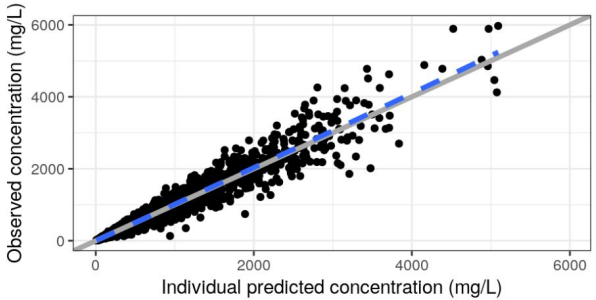
The following plots assume that the preferred x-axis labels are defined here.

DV vs PRED and IPRED

Create plots of DV vs PRED and IPRED for the full dataset and stratified by renal function and hepatic function.

```
## [1] "DV vs PRED and IPRED"
```





Model Diagnostics - Spec file

- Read in your spec file

```
spec <- ys_load(here("data","spec","analysis3.yml"))
head(spec)
```

```
   name info  unit                 short        source
1     C  cd-     .     comment character             .
2   NUM  ---     .         record number ysdb_internal
3    ID  ---     .    subject identifier ysdb_internal
4  TIME  ---  hour time after first dose             .
5   SEQ  -d-     .             data type             .
6   CMT                compartment number ysdb_internal
```

- Change the namespace

```
spec <- ys_namespace(spec, "plot")
```

- Use the spec flags

```
diagContCov <- pull_meta(spec, "flags")$diagContCov
diagCatCov <- pull_meta(spec, "flags")$diagCatCov
```

Model Diagnostics - Data

- Read in your model output
  - `read_model`
  - `model_summary`

```
mod <- read_model(here("model","pk","106"))
sum <- mod %>% model_summary()
```

- Read in your data
  - `nm_join` - to join your NONMEM tables with the original dataset
  - `filter` to the observation records
  - `yspec_add_factors` to decode categorical covariates

```
data0 <- nm_join(mod)
```

```
data <-
  data0 %>%
  filter(EVID==0) %>%
  yspec_add_factors(spec, .suffix = "")
```

Hands-on

Model Diagnostics

NPDE vs continuous covariates plot

- Get covariates of interest from the spec file and make a list of axis labels
  - `pull_meta` to pull in information about the flags and select the appropriate flag
  - `ys_select` those covariates
  - `axis_col_labs` will convert the selected covariates to column axis labels

```
diagContCov <- pull_meta(spec, "flags")$diagContCov

NPDEco <-
  spec %>%
  ys_select(all_of(diagContCov)) %>%
  axis_col_labs(title_case = TRUE, short_max = 10) %>%
  as.list()

NPDEco
```

```
$AGE
[1] "AGE//Age (years)"

$WT
[1] "WT//Weight (kg)"

$ALB
[1] "ALB//Albumin (g/dL)"

$EGFR
[1] "EGFR//EGFR (mL/min/1.73m2)"
```

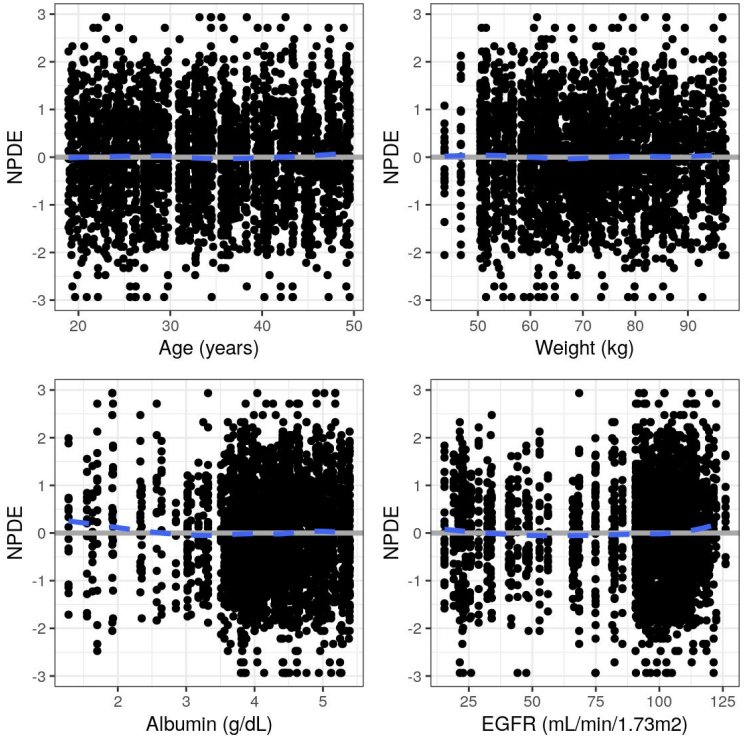
Model Diagnostics

NPDE vs continuous covariates plot

- Get covariates of interest from the spec file and make a list of axis labels

```
pList <- purrr::map(NPDEco, ~ npde_cont(data, x = .x))

pm_grid(pList, ncol = 2)
```

- `map` across the covariate list to create all plots using `npde_cont`
- `pm_grid` to display all plots in a grid



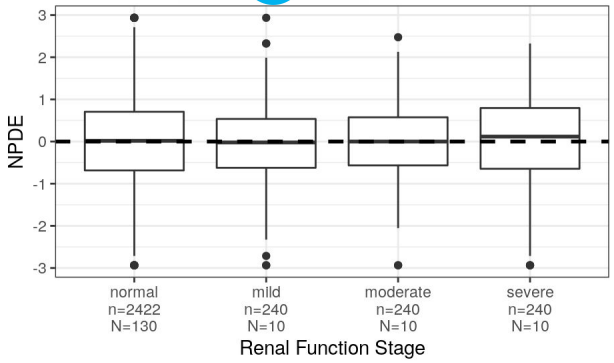
METRUM RESEARCH GROUP

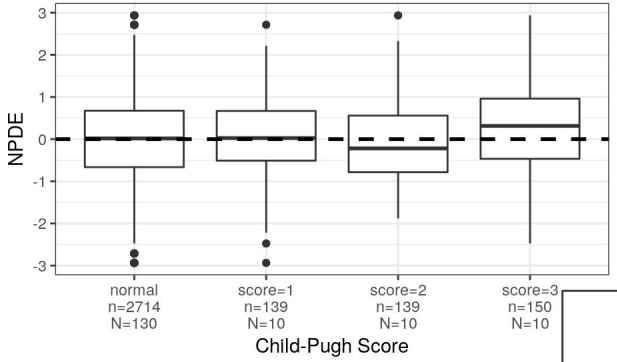
Model Diagnostics

NPDE vs categorical covariates plot

- Use similar methods to create NPDE plots for categorical covariates

```
NPDEca <-
  spec %>%
  ys_select("RF", "CP") %>%
  axis_col_labs(title_case = TRUE, short_max = 20) %>%
  as.list()
pList_cat = purrr::map(NPDEca, ~ npde_cat(data, x = .x))
pm_grid(pList_cat, ncol=1)
```





Hands-on

METRUM RESEARCH GROUP

Model Diagnostics

- The ETA based plots require a dataset filtered to one record per subject

```
id <- distinct(data, ID, .keep_all=TRUE)
```

- pmplots package has series of ETA plot functions
  - `eta_pairs` correlation and distribution of model ETAs
  - `eta_cont` ETA vs continuous covariates
  - `eta_cat` ETA vs categorical covariates
- Leverage the information in the spec object in several ways:
  - Extract covariates of interest from the spec flags with `pull_meta` and `ys_select`
  - Axis labels are renamed with the short label in the spec using `axis_col_labs`
  - Numerical categorical covariates are decoded with the `yspec_add_factors` function

Model Diagnostics

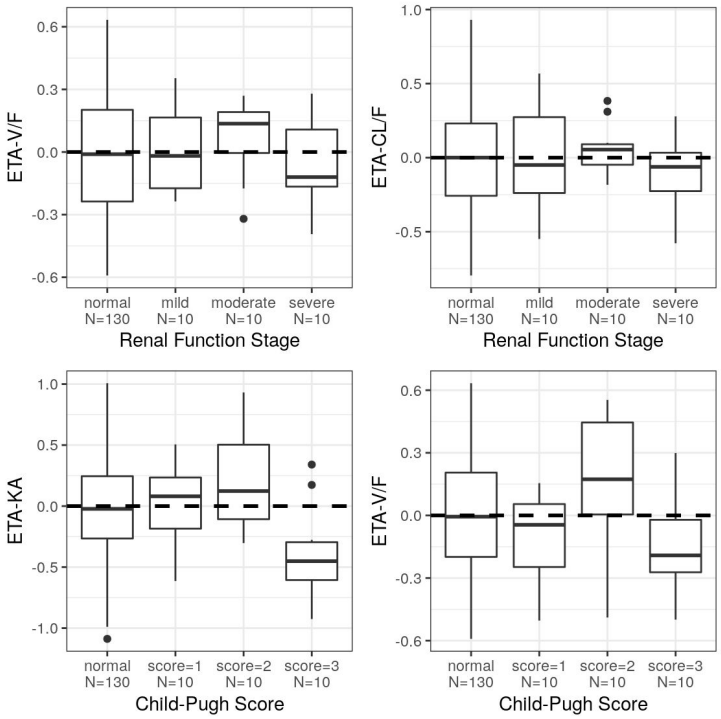
ETA vs categorical covariates plot

- Define the ETAs of interest

```
etas <- c("ETA1//ETA-KA", "ETA2//ETA-V/F", "ETA3//ETA-CL/F")
```

- Get the covariates from the spec file and use the `eta_cat` function to create a list of plots for each ETA and covariate pairing

```
ca <-
  spec %>%
  ys_select(diagCatCov) %>%
  axis_col_labs(title_case=TRUE, short_max = 20)
p <- eta_cat(id, ca, etas)
pRenal <- (p[[5]] + p[[6]]) / (p[[7]] + p[[8]])
pRenal
```

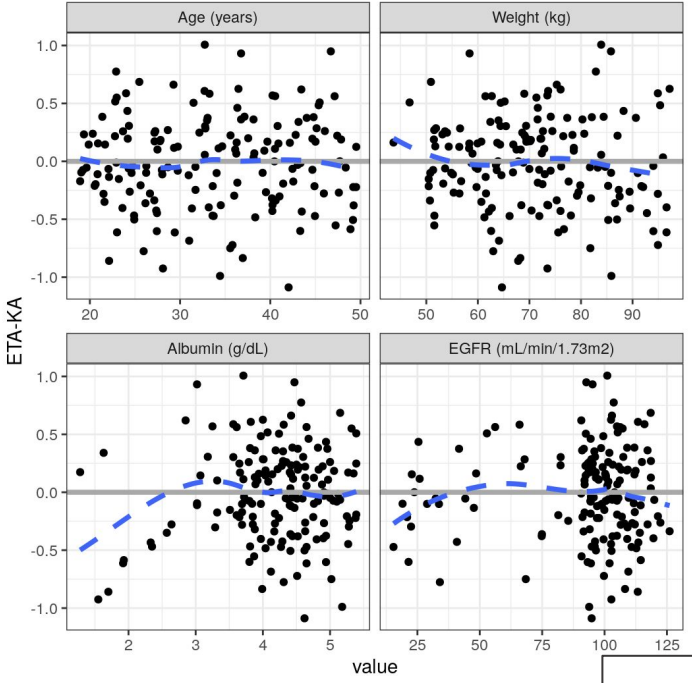


Model Diagnostics

ETA vs continuous covariates plot

- `wrap_eta_cont` makes an ETA plot faceted by continuous covariates
- `map` over the ETAs to create multiple plots

```
map_wrap_eta_cont = function(.id, .co, .etas){
  p <- wrap_eta_cont(.id,
                     x = .co, y = .etas,
                     use_labels = TRUE,
                     scales= "free_x")
}

p = purrr::map(.x = etas, ~ map_wrap_eta_cont(id, contCo, .x))
p[[1]]
```



Hands-on

METRUM RESEARCH GROUP

Reporting templates using Rmarkdown

Title for the page

Output to an html file with a floating table of contents

Parameters that can be updated each time the Rmarkdown is rendered

```
---
title: "Report diagnostics"
output:
  html_document:
    toc: true
    toc_float: true
    depth: 2
params:
  run: 102
  modelDir: "model/pk"
  script: "diagnostics-report.Rmd"
  yspec: "analysis3.yml"
  contCov: !r c("AGE","WT","ALB","EGFR")
  catCov: !r c("STUDY", "RF", "CP", "DOSE")
  etas: !r c("ETA1//ETA-KA", "ETA2//ETA-V/F", "ETA3//ETA-CL/F")
  drugNameUnits: "concentration (mg/L)"
  include_code: FALSE
  include_plots: TRUE
  run_mrggsave: TRUE
```

Rendering templates using R

- Several different ways to render the templates
  - Only need to define parameters that differ from the defaults provided in the template yaml section
  - Use our `model_diagnostics` helper function to render the plot and `browseURL` to pop open the html after creation

```
mod <- bbr::read_model(file.path(modelDir, 100))

mod %>%
  model_diagnostics(
    modelSpecifics,
    template = rmd_template
  )
```

```
model_diagnostics(
  file.path(modelDir, 102),
  modelSpecifics,
  template = rmd_template
) %>%
  browseURL()
```

Rendering templates using R

- Define the model specifics

```
modelSpecifics <- list(
  yspec = "analysis3.yml",
  contCov = c("AGE","WT","ALB","EGFR"),
  catCov = c("STUDY", "RF", "CP", "DOSE"),
  etas = c("ETA1//ETA-KA", "ETA2//ETA-V/F", "ETA3//ETA-CL/F"),
  include_code = TRUE,
  include_plots = TRUE,
  run_mrggsave = TRUE)
```

- Render the Rmd template

```
rmarkdown::render(
  here("script", "diagnostic-templates", "diagnostics-report.Rmd"),
  params = modelSpecifics,
  output_dir = here(modelDir, "100"),
  output_file = "diagnostic-report-100.html"
)
```

Hands-on

Break

Using bbr for model development

bbr is an R package developed by MetrumRG. It serves three primary purposes:

- Submit NONMEM models, particularly for execution in parallel and/or on a high-performance compute (HPC) cluster (e.g. Metworx).
- Parse NONMEM outputs into R objects to facilitate model evaluation and diagnostics in R.
- Annotate the model development process for easier and more reliable traceability and reproducibility.

Walk though:

- Creating and submitting a model
- Iterative model development
- Preview of model evaluation and diagnostics
- Annotation of models with tags, notes, etc.

Follow along on the "Model Management" page and associated code.

bbr: Creating and submitting a model

Creating a model object from a NONMEM control stream file:

```
# create the first model
mod100 <- new_model(file.path(MODEL_DIR, 100))
```

Submitting models for execution:

```
submit_model(mod100)
```

bbr: Creating and submitting a model

Creating a model object from a NONMEM control stream file:

```
# create the first model
mod100 <- new_model(file.path(MODEL_DIR, 100))
```

Submitting models for execution:

```
submit_model(mod100)
```

```
submit_model(
  mod,
  .bbi_args = list(
    overwrite = TRUE,
    parallel = TRUE,
    threads = 8
  )
)
```

These other arguments let you parallelize the run, too!

bbr: Iterative model development

Creating a new model based on an existing model:

```
mod101 <- copy_model_from(
    .parent_mod = mod100,
    .new_model = 101,
    .inherit_tags = TRUE
  )
```

This will copy an existing model ("100") and make a new one ("101"). You can then edit and save 101.ctl accordingly.

Housekeeping:

- it will "remember" the lineage (you'll see that later),
- and… can carry over tags.

bbr: Iterative model development

Once you've created a new model based on an existing model:

```
mod101 <- copy_model_from(
    .parent_mod = mod100,
    .new_model = 101,
    .inherit_tags = TRUE
  )
```

Compare that model to its "parent" model:

```
# shows the difference between control streams
model_diff(mod101)
```

bbr: Model evaluation and diagnostics

Parse NONMEM outputs into an R list object:

```
sum100 <- model_summary(mod100)
```

Create a simple tibble with parameter estimates:

```
# helper function to extract parameter table
sum100 %>% param_estimates()
```

bbr: Adding model annotation

Add notes to the model:

```
mod100 <- mod100 %>%
  add_notes("systematic bias, explore alternate compartme
```

Add tags to the model:

```
mod100 <- mod100 %>%
  add_tags(c(
    TAGS$one_compartment_absorption,
    TAGS$eta_cl,
    TAGS$eta_ka,
    TAGS$eta_v,
    TAGS$proportional_ruv
  ))
```

bbr: Leveraging model annotation

Create a “run log” table:

```
# create a run log and do some basic formatting
run_log(MODEL_DIR, .recurse = FALSE) %>%
  collapse_to_string(based_on, tags, notes) %>%
  select(run, based_on, description, tags, notes) %>% knitr::kable()
```

run	based_on	description	tags	notes
100	NA	NA	one-compartment + absorption, ETA-CL, ETA-KA, ETA-V, proportional RUV	systematic bias, explore alternate compartmental structure
101	100	NA	two-compartment + absorption, ETA-CL, ETA-KA, ETA-V2, proportional RUV	eta-V2 shows correlation with weight, consider adding allometric weight
102	101	Base Model	two-compartment + absorption, ETA-CL, ETA-KA, ETA-V2, CLWT-allo, V2WT-allo	Allometric scaling with weight reduces eta-V2 correlation with weight. Will consider additional RUV structures. Proportional

Quick start

Getting started with the example project

- Download the Github repository and upload it to your Metworx session
- Start an Rstudio session and open the ***expo1-nonmem-foce.Rproj*** project
- **Go to the terminal window** in project home directory: type > ***pkgr install***
  - Hit enter, then once packages have installed, restart your R session
- Then install bbi **in your R console** by running ***bbr::use_bbi()***
- You should now be ready to start running code given in the example project. Runnable code examples are in the *script/* folder

More details here: https://merge.metrumrg.com/zy8x3BETA7R5Ph/posts/about-the-repo.html

Additional Resources

- MeRGE Expo 1 website:

  http://merge.metrumrg.com/expo/expo1-nonmem-foce/
- Package management: MPN and pkgr
  - https://kb.metworx.com/Users/Managing_R_Packages/r-package-management/
    - Questions: File a Metworx help tickets:

      https://kb.metworx.com/Users/Getting_Started/create-support-ticket/
- VPCs using mrgsolve
  - https://merge.metrumrg.com/expo/expo1-nonmem-foce/posts/pk-vpc-final.html
- Right sizing workflow
  - https://kb.metworx.com/Users/Getting_Started/rightsizing-workflows/
  - https://metrumresearchgroup.github.io/bbr/articles/nonmem-parallel.html
- General bbr "cheat sheet":

  https://metrumresearchgroup.github.io/cheatsheets/bbr_nonmem_cheat_sheet.pdf