

# 2025 National Taiwan University - Population Pharmacokinetics workshop

- 1 Load Packages
- 2 Dataset
- 3 Showcase 4
  - 3.1 Overlay First Dose data with the built model - Individual fits
  - 3.2 Overlay First Dose data with the built model - pcVPC
  - 3.3 Fit First Full
  - 3.4 Fit First 2
  - 3.5 Fit First 1
  - 3.6 Pred First 0
  - 3.7 Compare across sampling schemes
  - 3.8 Bias and Precision

## 1 Load Packages

Here we will import our installed packages into our R environment

```
library(renv)
```

```
##  
## Attaching package: 'renv'
```

```
## The following objects are masked from 'package:stats':  
##  
##      embed, update
```

```
## The following objects are masked from 'package:utils':  
##  
##      history, upgrade
```

```
## The following objects are masked from 'package:base':  
##  
##      autoload, load, remove, use
```

```
library(tidyverse)
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats    1.0.0      ✓ stringr    1.5.1
## ✓ ggplot2     3.5.2      ✓ tibble     3.2.1
## ✓ lubridate  1.9.4      ✓ tidyr      1.3.1
## ✓ purrr      1.0.4
```

```
## — Conflicts — tidyverse_conflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## ✖ purrr::modify() masks renv::modify()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(ggplot2)
library(nlmixr2)
```

```
## Loading required package: nlmixr2data
```

```
library(xpose4)
```

```
## Loading required package: lattice
```

```
library(xpose.nlmixr2)
```

```
## Loading required package: xpose
##
## Attaching package: 'xpose'
##
## The following object is masked from 'package:stats':
##
##   filter
```

```
library(rxode2)
```

```
## rxode2 3.0.4 using 1 threads (see ?getRxThreads)
##   no cache: create with `rxCreateCache()`
## =====
## rxode2 has not detected OpenMP support and will run in single-threaded mode
## This is a Mac. Please read https://mac.r-project.org/openmp/
## =====
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##      combine
```

```
library(ggPMX)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2
##
## Attaching package: 'ggPMX'
##
## The following object is masked from 'package:xpose':
##
##      get_data
```

```
library(ggpubr)
library(mrgsolve)
```

```
##
## Attaching package: 'mrgsolve'
##
## The following object is masked from 'package:renv':
##
##      init
##
## The following object is masked from 'package:stats':
##
##      filter
```

```
library(vpc)
```

```
##
## Attaching package: 'vpc'
##
## The following object is masked from 'package:xpose':
##
##      vpc
```

```
library(patchwork)
```

## 2 Dataset

```
# Import busulfan two compartment dataset
busulfan_First_Dose_dataset <- read.csv("dataset/busulfan_First_Dose.csv", na.strings =
".")
```

## 3 Showcase 4

### 3.1 Overlay First Dose data with the built model - Individual fits

```
model_code <- 'model/busulfan_test_dose_model.mod'
mod <- mread('busulfan_test_dose', file = model_code)
```

```
## Building busulfan_test_dose ... done.
```

```

sim <- function(rep, data, model,
               recover = c('AMT', 'DV', 'Sex', 'BSA')) {
  mrgsim(
    model,
    data = data,
    recover = recover,
    Req = c('CP_no_RUV', 'CP', 'no_obs_AUC'),
    output = 'df',
  ) %>% mutate(irep = rep)
}

isim <- seq(1000)

set.seed(86486)

sims <- lapply(
  isim, sim,
  data = busulfan_First_Dose_dataset,
  mod = mod
) %>% bind_rows()

indv_fits <- sims %>% filter(irep == 1)

p1 <- ggplot() +
  geom_line(data = indv_fits, aes(TIME/60, CP_no_RUV, group = ID)) +
  geom_point(data = busulfan_First_Dose_dataset, aes(TIME/60, DV, group = ID)) +
  facet_wrap_paginate(~ID, nrow = 4, ncol = 4, page = 1) +
  xlim(2.5, 10) +
  labs(x = 'Time (horus)', y = "Busulfan Concentration (mg/L)") +
  theme_bw()

p2 <- ggplot() +
  geom_line(data = indv_fits, aes(TIME/60, CP_no_RUV, group = ID)) +
  geom_point(data = busulfan_First_Dose_dataset, aes(TIME/60, DV, group = ID)) +
  facet_wrap_paginate(~ID, nrow = 4, ncol = 4, page = 2) +
  xlim(2.5, 10) +
  labs(x = 'Time (horus)', y = "Busulfan Concentration (mg/L)") +
  theme_bw()

p1

```

```

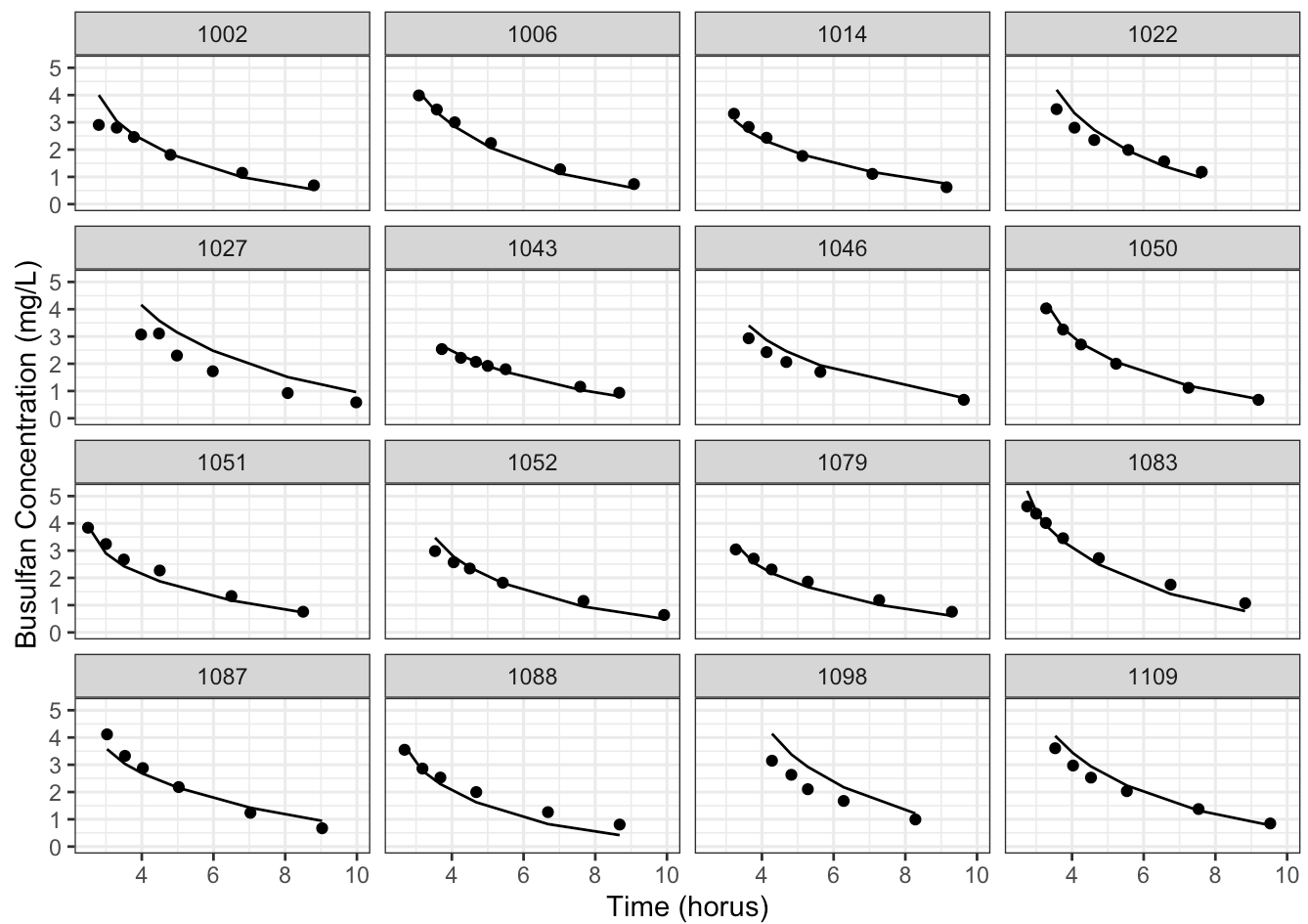
## Warning: Removed 24 rows containing missing values or values outside the scale range
## (`geom_line()`).

```

```

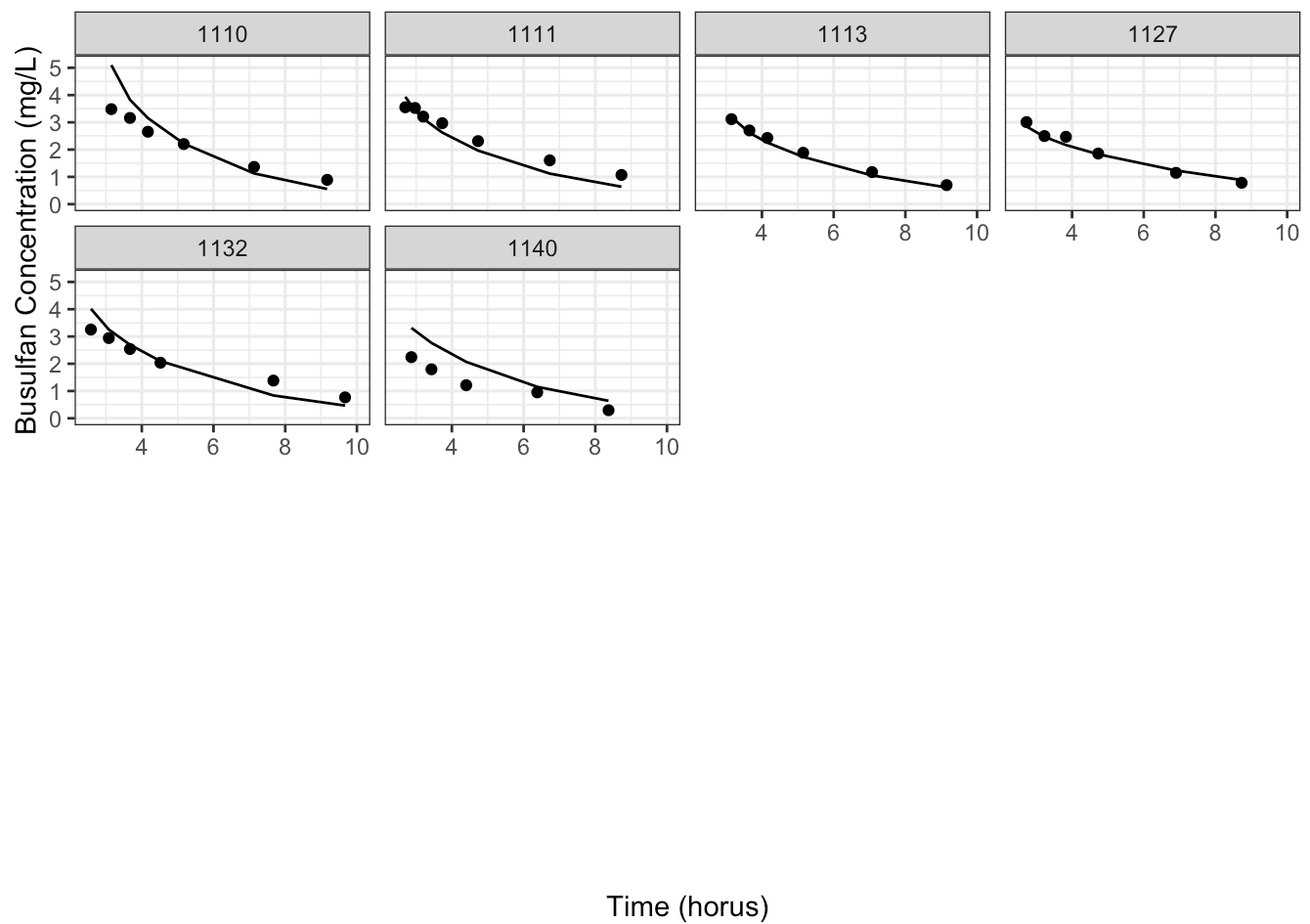
## Warning: Removed 24 rows containing missing values or values outside the scale range
## (`geom_point()`).

```



p2

```
## Warning: Removed 24 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 24 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



### 3.2 Overlay First Dose data with the built model -

# pcVPC

```
mrg_vpc_theme = new_vpc_theme(list(
  sim_pi_fill = "steelblue3", sim_pi_alpha = 0.5,
  sim_median_fill = "grey60", sim_median_alpha = 0.5
))

vpc_plot <- vpc(obs = busulfan_First_Dose_dataset,
  obs_cols = list(dv = 'DV',
    idv = 'TIME',
    id = 'ID'
  ),
  sim = sims,
  bins = "pretty",
  n_bins = 10,
  sim_cols = list(dv = 'CP',
    idv = 'TIME',
    id = 'ID',
    sim = 'irep'
  ),
  #log_y = TRUE,
  #stratify = 'DOSCOV',
  pi = c(0.025, 0.975),
  ci = c(0.025, 0.975),
  show = list(obs_dv = TRUE
    #obs_ci = FALSE
    #obs_median = FALSE
  ),
  vpc_theme = mrg_vpc_theme,
  ylab = 'Busuflan Concentration (mg/L)',
  xlab = 'Time (hours)',
  title = "Prediction-Corrected Visual Predictive Check - pcVPC"
) +
theme_bw() +
scale_x_continuous(limits = c(150, 600),
  breaks = seq(0, 650, by = 30),
  labels = function(x) x / 60)

vpc_plot +
  # Add dummy geoms to create legend entries
  geom_rect(aes(xmin = 0, xmax = 0, ymin = 0, ymax = 0, fill = "PI (95%)"), alpha = 0.5)
+
  geom_rect(aes(xmin = 0, xmax = 0, ymin = 0, ymax = 0, fill = "Median"), alpha = 0.5) +

  # Define manual fill scale
  scale_fill_manual(name = "Simulated Data",
    values = c("PI (95%)" = "steelblue3", "Median" = "grey60")) +

  # Customize the legend position inside plot
  theme(
    legend.position = c(0.90, 0.80), # x and y coordinates (0~1 scale)
    legend.background = element_rect(fill = "white", color = "black"),
```



```
    legend.title = element_text(size = 10),  
    legend.text = element_text(size = 9)  
  ) +  
  
  guides(fill = guide_legend(override.aes = list(alpha = 0.5)))
```

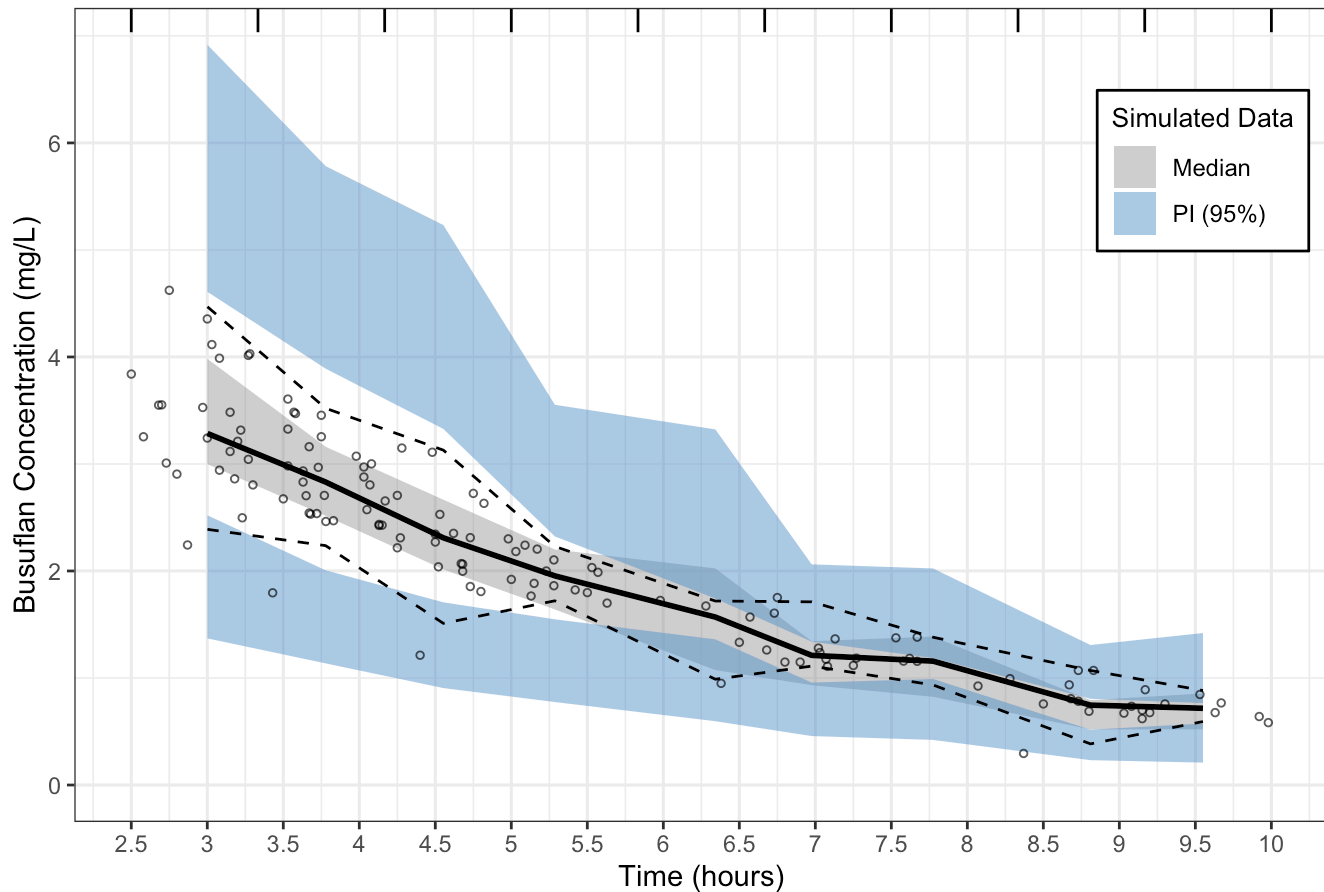
```
## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2  
## 3.5.0.  
## i Please use the `legend.position.inside` argument of `theme()` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was  
## generated.
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range  
## (`geom_line()`).  
## Removed 2 rows containing missing values or values outside the scale range  
## (`geom_line()`).  
## Removed 2 rows containing missing values or values outside the scale range  
## (`geom_line()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range  
## (`geom_point()`).
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range  
## (`geom_rect()`).  
## Removed 12 rows containing missing values or values outside the scale range  
## (`geom_rect()`).
```

## Prediction-Corrected Visual Predictive Check - pcVPC



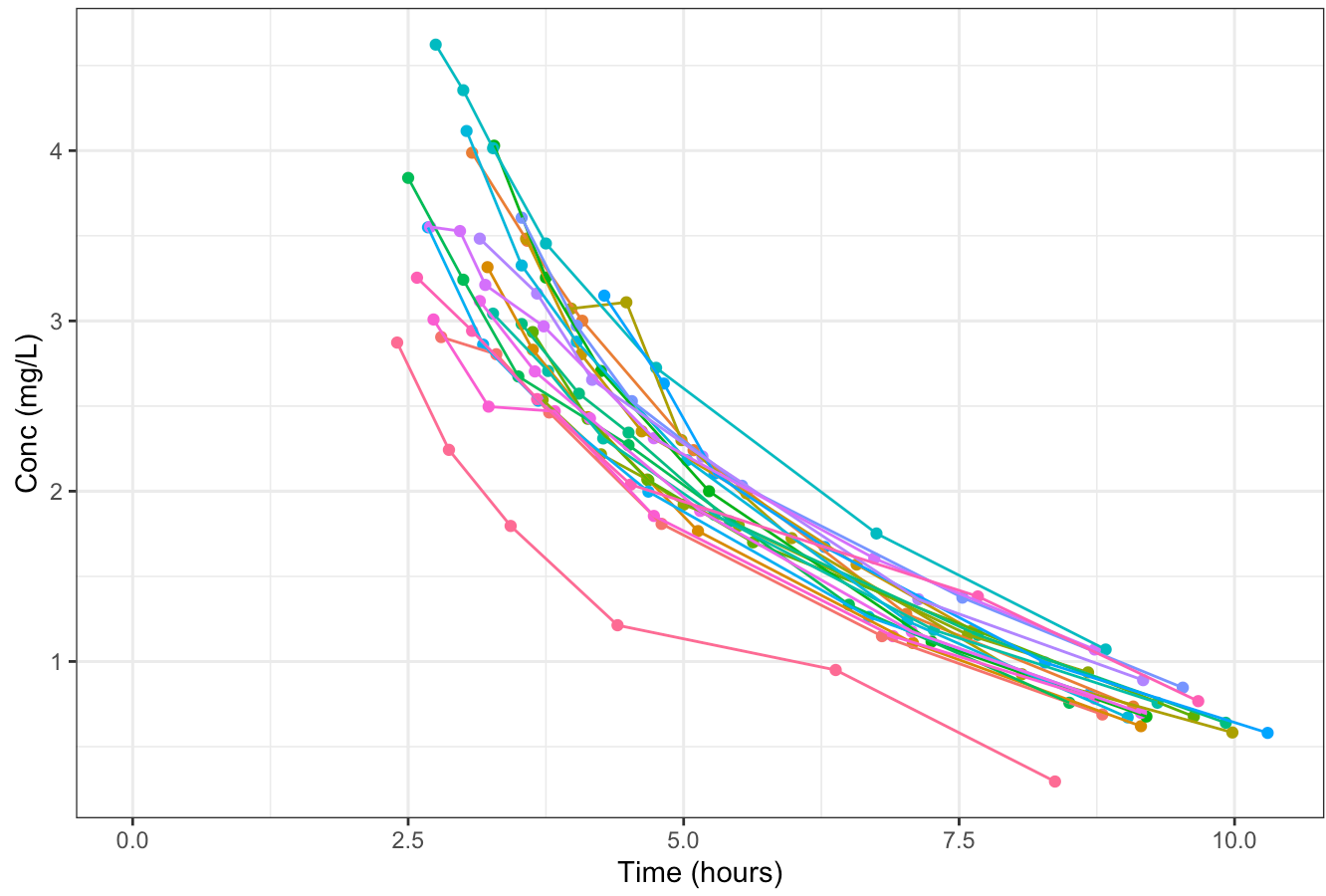
### 3.3 Fit First Full

```
# Plasma drug concentration decline in two phases shown in semi-log plot
ggplot(busulfan_First_Dose_dataset, aes(TIME/60, DV, group = ID, color = as.factor(ID)))
+
  geom_point() +
  geom_line() +
  theme_bw() +
  labs(title = "Busulfan First Dose semi-log plot", x = "Time (hours)", y = "Conc (mg/
L)") +
  theme(legend.position = "blank")
```

```
## Warning: Removed 22 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 22 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

Busulfan First Dose semi-log plot



```

busulfan_first_dose_2cmt_model <- function() {
  ini({
    # Typical value (THETAs)
    tvcl  <- log(0.187)
    tvv1  <- log(29)
    tvq   <- log(0.41)
    tvv2  <- log(17.3)

    covbsav1 <- log(2.32)
    covbsacl <- log(1.30)
    covsexv2 <- log(0.8)

    # Interindividual variability (OMEGAs)
    eta_cl  ~ 0.0222
    eta_v1  ~ 0.0222
    eta_v2  ~ 0.0241

    # Residual variability
    prop.err <- 0.0955
  })

  model({
    # Individual value
    cl <- exp(tvcl + eta_cl) * (BSA/2.01)^covbsacl
    v1 <- exp(tv1 + eta_v1) * (BSA/2.01)^covbsav1
    q  <- exp(tvq)
    v2 <- exp(tv2 + eta_v2)

    # Sex effect
    if (Sex == 1) {
      v2 <- v2 * covsexv2
    } else {
      v2 <- v2 * 1
    }

    # Conversion
    k10 <- cl / v1
    k12 <- q / v1
    k21 <- q / v2

    # ODEs
    d / dt(central) = -k10 * central - k12 * central + k21 * peripheral
    d / dt(peripheral) = -k21 * peripheral + k12 * central

    # Concentration
    cp = central / v1

    # Error model (on log scale)
    IPRED = cp
    IPRED ~ prop(prop.err)
  })
}

```

```
# Model Fitting
full_tp_pk_fit <- nlmixr2(
  busulfan_first_dose_2cmt_model,
  busulfan_First_Dose_dataset,
  "posthoc",
  control = foceiControl(maxOuterIterations=0),
  table = list(cwres = TRUE, npde = TRUE)
)
```

```
## i parameter labels from comments are typically ignored in non-interactive mode
```

```
## i Need to run with the source intact to parse comments
```

```
## → Calculating residuals/tables
```

```
## ✓ done
```

```
## → compress origData in nlmixr2 object, save 9504
```

```
## Full-Estimation
full_tp_indiv_cl <- data.frame(ID = full_tp_pk_fit$ID, Clearance = full_tp_pk_fit$cl)

full_tp_indiv_exposure <- merge(busulfan_First_Dose_dataset, full_tp_indiv_cl, by = 'ID')
full_tp_indiv_exposure <- full_tp_indiv_exposure %>% mutate(AUC = AMT/Clearance) %>% select(ID, AUC)

full_tp_AUC <- full_tp_indiv_exposure %>%
  group_by(ID) %>%
  summarize(full_tp_AUC = first(AUC), .groups = "drop")

full_tp_AUC
```

```
## # A tibble: 22 × 2
##       ID full_tp_AUC
##   <int>     <dbl>
## 1  1002     1051.
## 2  1006     1171.
## 3  1014     1113.
## 4  1022     1364.
## 5  1027     1640.
## 6  1043     1212.
## 7  1046     1158.
## 8  1050     1240.
## 9  1051       972.
## 10 1052     1179.
## # i 12 more rows
```

## 3.4 Fit First 2

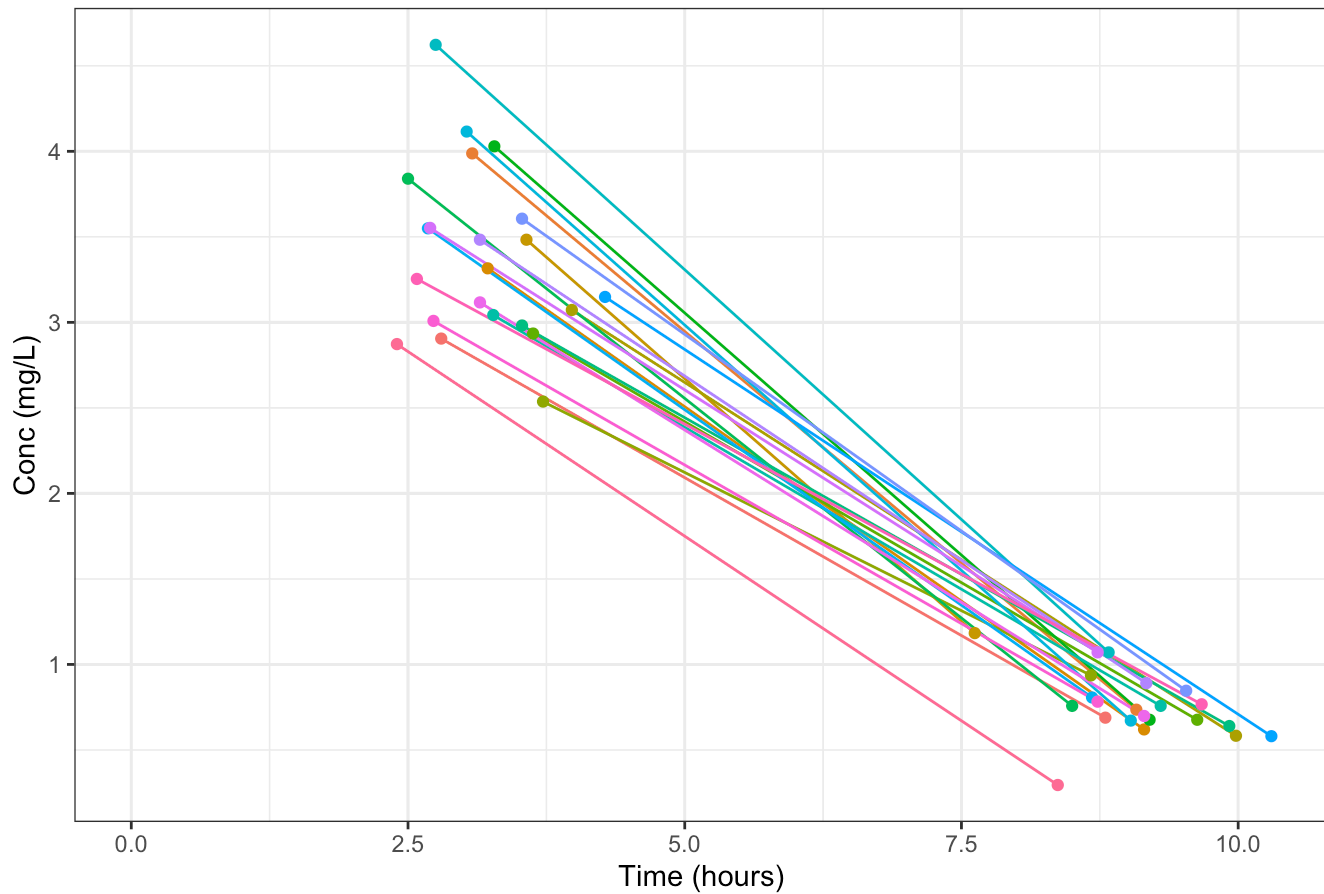
```
# Plasma drug concentration decline in two phases shown in semi-log plot
busulfan_tdm_dataset <- busulfan_First_Dose_dataset %>%
  group_by(ID) %>%
  filter(row_number() == 1 | row_number() == 2 | row_number() == n()) %>% ungroup()

ggplot(busulfan_tdm_dataset, aes(TIME/60, DV, group = ID, color = as.factor(ID))) +
  geom_line() +
  geom_point() +
  theme_bw() +
  labs(title = "Busulfan First Dose semi-log plot", x = "Time (hours)", y = "Conc (mg/
L)") +
  theme(legend.position = "blank")
```

```
## Warning: Removed 22 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 22 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

## Busulfan First Dose semi-log plot



```
# Model Fitting - Need MaxEval zero
two_cmt_tdm_pk_fit <- nlmixr2(
  busulfan_first_dose_2cmt_model,
  busulfan_tdm_dataset,
  "posthoc",
  control = foceiControl(maxOuterIterations=0),
  table = list(cwres = TRUE, npde = TRUE)
)
```

```
## i parameter labels from comments are typically ignored in non-interactive mode
```

```
## i Need to run with the source intact to parse comments
```

```
## → Calculating residuals/tables
```

```
## ✓ done
```

```
## → compress origData in nlmixr2 object, save 4432
```

```
## TDM Point
orig_data <- two_cmt_tdm_pk_fit$origData
indv_cl <- data.frame(ID = two_cmt_tdm_pk_fit$ID, Clearance = two_cmt_tdm_pk_fit$cl)

tdm_indiv_exposure <- merge(orig_data, indv_cl, by = 'ID')
tdm_indiv_exposure <- tdm_indiv_exposure %>% mutate(AUC = AMT/Clearance) %>% select(ID,
AUC)

tdm_tp_AUC <- tdm_indiv_exposure %>%
  group_by(ID) %>%
  summarize(tdm_AUC = first(AUC), .groups = "drop")

tdm_tp_AUC
```

```
## # A tibble: 22 × 2
##       ID tdm_AUC
##   <int> <dbl>
## 1  1002  1051.
## 2  1006  1170.
## 3  1014  1091.
## 4  1022  1363.
## 5  1027  1638.
## 6  1043  1221.
## 7  1046  1185.
## 8  1050  1225.
## 9  1051   971.
## 10 1052  1159.
## # i 12 more rows
```

## 3.5 Fit First 1

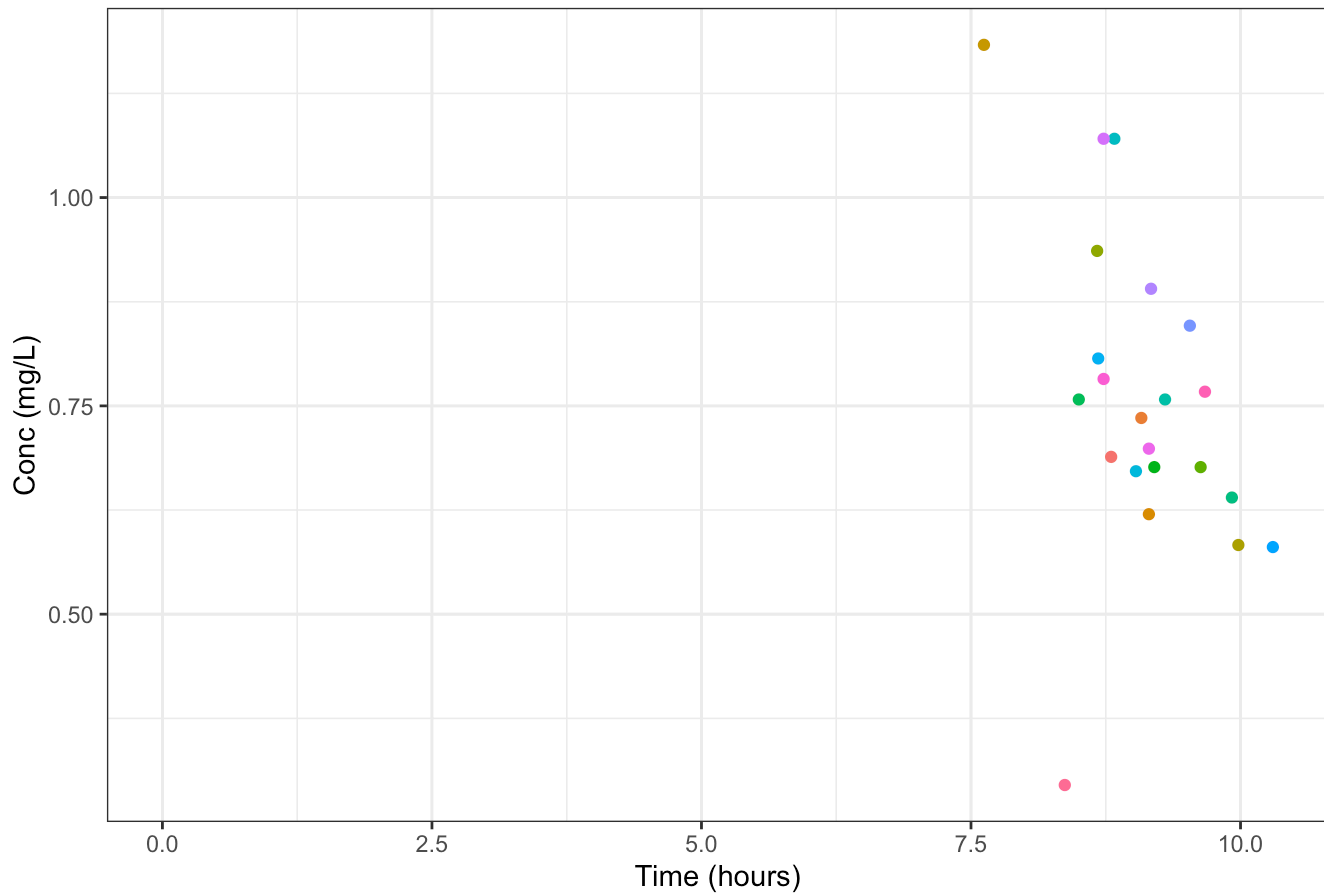
```
last_time_point <- busulfan_tdm_dataset %>%
  group_by(ID) %>%
  filter(row_number() == 1 | row_number() == n())

# Plasma drug concentration decline in two phases shown in semi-log plot
ggplot(last_time_point, aes(TIME/60, DV, group = ID, color = as.factor(ID))) +
  geom_point() +
  theme_bw() +
  labs(title = "Busulfan First Dose semi-log plot", x = "Time (hours)", y = "Conc (mg/
L)") +
  theme(legend.position = "blank")
```

```
## Warning: Removed 22 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



## Busulfan First Dose semi-log plot



```
# Model Fitting
two_cmt_last_tp_pk_fit <- nlmixr2(
  busulfan_first_dose_2cmt_model,
  last_time_point,
  "posthoc",
  control = foceiControl(maxOuterIterations=0),
  table = list(cwres = TRUE, npde = TRUE)
)
```

```
## i parameter labels from comments are typically ignored in non-interactive mode
```

```
## i Need to run with the source intact to parse comments
```

```
## → Calculating residuals/tables
```

```
## ✓ done
```

```
## → compress origData in nlmixr2 object, save 3496
```

```
## Last Time Point
last_tp_indv_cl <- data.frame(ID = two_cmt_last_tp_pk_fit$ID, Clearance = two_cmt_last_t
p_pk_fit$cl)

last_tp_indiv_exposure <- merge(last_time_point, last_tp_indv_cl, by = 'ID')
last_tp_indiv_exposure <- last_tp_indiv_exposure %>% mutate(AUC = AMT/Clearance) %>% sel
ect(ID, AUC)

last_tp_AUC <- last_tp_indiv_exposure %>%
  group_by(ID) %>%
  summarize(last_tp_AUC = first(AUC), .groups = "drop")

last_tp_AUC
```

```
## # A tibble: 22 × 2
##       ID last_tp_AUC
##   <int>    <dbl>
## 1  1002    1052.
## 2  1006    1171.
## 3  1014    1066.
## 4  1022    1363.
## 5  1027    1638.
## 6  1043    1238.
## 7  1046    1214.
## 8  1050    1180.
## 9  1051     973.
## 10 1052    1182.
## # i 12 more rows
```

## 3.6 Pred First 0

```
## No observation
no_tp_AUC <- indiv_fits %>%
  group_by(ID) %>%
  summarize(no_tp_AUC = first(no_obs_AUC), .groups = "drop")

no_tp_AUC
```

```
## # A tibble: 22 × 2
##       ID no_tp_AUC
##   <dbl>   <dbl>
## 1  1002     1089.
## 2  1006     1200.
## 3  1014     1137.
## 4  1022     1246.
## 5  1027     1601.
## 6  1043     1157.
## 7  1046     1281.
## 8  1050     1329.
## 9  1051     1150.
## 10 1052     1151.
## # i 12 more rows
```

## 3.7 Compare across sampling schemes

```
compare_auc <- merge(full_tp_AUC, tdm_tp_AUC, by = "ID")
compare_auc <- merge(compare_auc, last_tp_AUC, by = "ID")
compare_auc <- merge(compare_auc, no_tp_AUC, by = "ID")

compare_auc[] <- lapply(compare_auc, function(x) if(is.numeric(x)) round(x, 0) else x)
print(compare_auc)
```

```
##       ID full_tp_AUC tdm_AUC last_tp_AUC no_tp_AUC
## 1  1002         1051    1051         1052     1089
## 2  1006         1171    1170         1171     1200
## 3  1014         1113    1091         1066     1137
## 4  1022         1364    1363         1363     1246
## 5  1027         1640    1638         1638     1601
## 6  1043         1212    1221         1238     1157
## 7  1046         1158    1185         1214     1281
## 8  1050         1240    1225         1180     1329
## 9  1051          972     971          973     1150
## 10 1052         1179    1159         1182     1151
## 11 1079         1145    1096         1071     1062
## 12 1083         1182    1182         1183     1470
## 13 1087         1222    1222         1222     1330
## 14 1088          812     811          812       948
## 15 1098         1207    1248         1300     1467
## 16 1109         1325    1368         1382     1390
## 17 1110         1386    1310         1329     1405
## 18 1111         1109    1109         1111     1134
## 19 1113          857     857          858     1040
## 20 1127         1093    1080         1084     1139
## 21 1132         1018    1017         1019     1131
## 22 1140          964     963          965     1115
```

```

# Convert to long format
df_long <- compare_auc %>%
  pivot_longer(cols = -ID, names_to = "Method", values_to = "AUC")

# Order methods for nicer plotting
df_long$Method <- factor(df_long$Method, levels = c("full_tp_AUC", "tdm_AUC", "last_tp_AUC", "no_tp_AUC"))

# Set common y-axis limits
y_limits <- c(750, 1750)

# --- Full vs Two TP
df_last_two <- compare_auc %>%
  select(ID, Full = full_tp_AUC, Two_TP = tdm_AUC) %>%
  pivot_longer(cols = -ID, names_to = "Method", values_to = "AUC") %>%
  mutate(Method = factor(Method, levels = c("Full", "Two_TP")))

p_last_two <- ggplot(df_last_two, aes(x = Method, y = AUC)) +
  geom_boxplot(alpha = 0.3, aes(fill = Method, color = Method)) +
  geom_jitter(aes(color = Method), width = 0.1, alpha = 0.7, size = 2) +
  geom_line(aes(group = ID), color = "grey60", alpha = 0.6) +
  scale_fill_manual(values = c("Full" = "#F8766D", "Two_TP" = "#00BFC4")) +
  scale_color_manual(values = c("Full" = "#F8766D", "Two_TP" = "#00BFC4")) +
  theme_bw() +
  labs(title = "Two TP", x = "", y = "AUC") +
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5)
  ) +
  scale_y_continuous(limits = y_limits)

# --- Full vs Last TP
df_last <- compare_auc %>%
  select(ID, Full = full_tp_AUC, Last_TP = last_tp_AUC) %>%
  pivot_longer(cols = -ID, names_to = "Method", values_to = "AUC") %>%
  mutate(Method = factor(Method, levels = c("Full", "Last_TP")))

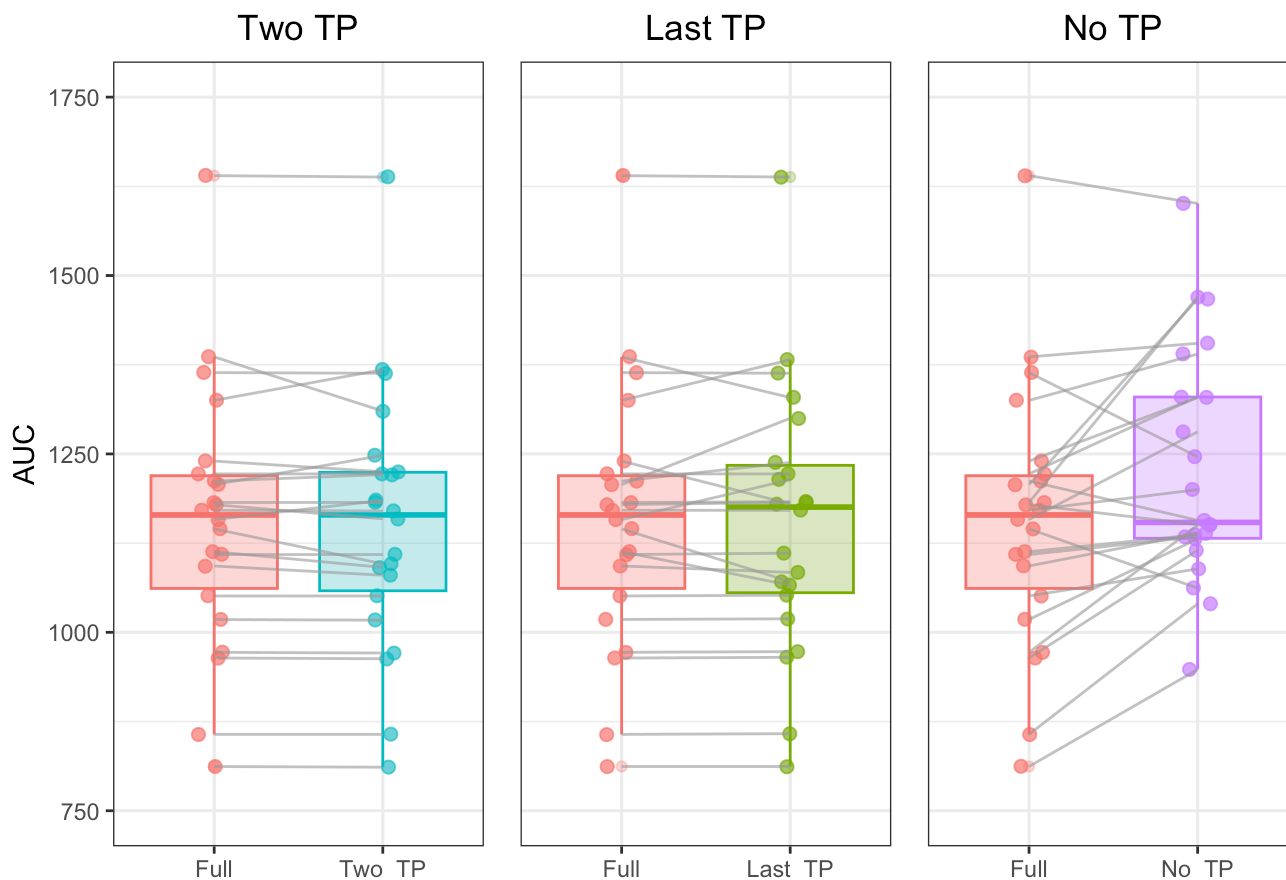
p_last <- ggplot(df_last, aes(x = Method, y = AUC)) +
  geom_boxplot(alpha = 0.3, aes(fill = Method, color = Method)) +
  geom_jitter(aes(color = Method), width = 0.1, alpha = 0.7, size = 2) +
  geom_line(aes(group = ID), color = "grey60", alpha = 0.6) +
  scale_fill_manual(values = c("Full" = "#F8766D", "Last_TP" = "#7CAE00")) +
  scale_color_manual(values = c("Full" = "#F8766D", "Last_TP" = "#7CAE00")) +
  theme_bw() +
  labs(title = "Last TP", x = "", y = "AUC") +
  theme(legend.position = "none",
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        plot.title = element_text(hjust = 0.5)
  ) +
  scale_y_continuous(limits = y_limits)

```

```
# --- Full vs No TP
df_no <- compare_auc %>%
  select(ID, Full = full_tp_AUC, No_TP = no_tp_AUC) %>%
  pivot_longer(cols = -ID, names_to = "Method", values_to = "AUC") %>%
  mutate(Method = factor(Method, levels = c("Full", "No_TP")))

p_no <- ggplot(df_no, aes(x = Method, y = AUC)) +
  geom_boxplot(alpha = 0.3, aes(fill = Method, color = Method)) +
  geom_jitter(aes(color = Method), width = 0.1, alpha = 0.7, size = 2) +
  geom_line(aes(group = ID), color = "grey60", alpha = 0.6) +
  scale_fill_manual(values = c("Full" = "#F8766D", "No_TP" = "#C77CFF")) +
  scale_color_manual(values = c("Full" = "#F8766D", "No_TP" = "#C77CFF")) +
  theme_bw() +
  labs(title = "No TP", x = "", y = "AUC") +
  theme(legend.position = "none",
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        plot.title = element_text(hjust = 0.5)
  ) +
  scale_y_continuous(limits = y_limits)

wrap_plots(p_last_two, p_last, p_no, ncol = 3)
```



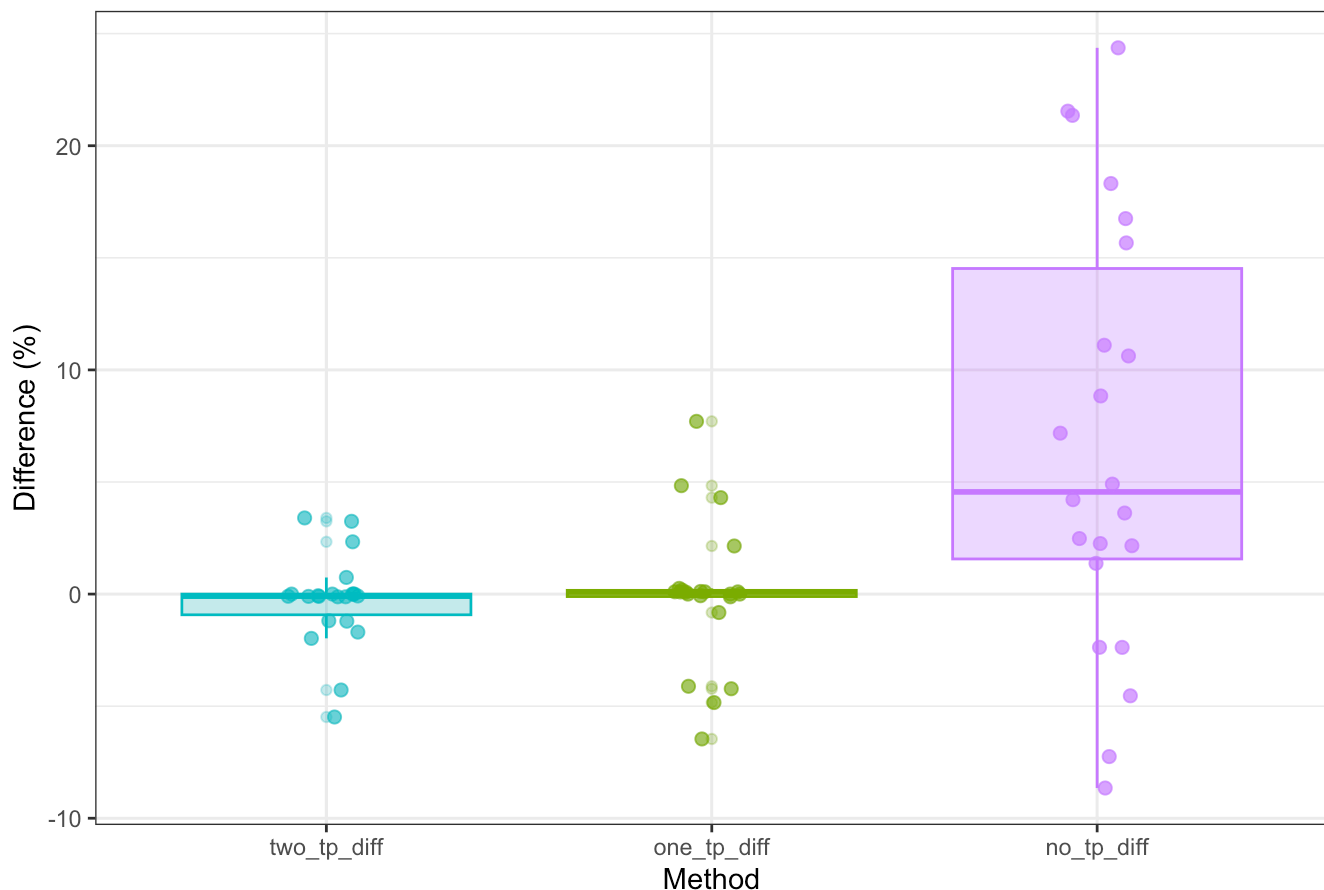
## 3.8 Bias and Precision

```
# Boxplot
compare_auc <- compare_auc %>%
  mutate(two_tp_diff = (tdm_AUC - full_tp_AUC) / full_tp_AUC * 100,
         one_tp_diff = (last_tp_AUC - full_tp_AUC) / full_tp_AUC * 100,
         no_tp_diff = (no_tp_AUC - full_tp_AUC) / full_tp_AUC * 100,
         )

first_dose_boxplot <- compare_auc %>%
  select(ID, ends_with("diff")) %>%
  pivot_longer(-ID, names_to = "Method", values_to = "Value") %>%
  mutate(Method = factor(Method, levels = c("two_tp_diff", "one_tp_diff", "no_tp_diff")))

ggplot(first_dose_boxplot, aes(x = Method, y = Value)) +
  geom_boxplot(alpha = 0.3, aes(fill = Method, color = Method)) +
  geom_jitter(aes(color = Method), width = 0.1, alpha = 0.7, size = 2) +
  theme_bw() +
  labs(y = "Difference (%)", title = "Percentage difference across sampling schemes") +
  theme(legend.position = "none") +
  scale_fill_manual(values = c("two_tp_diff" = "#00BFC4", "one_tp_diff" = "#7CAE00", "no_tp_diff" = "#C77CFF")) +
  scale_color_manual(values = c("two_tp_diff" = "#00BFC4", "one_tp_diff" = "#7CAE00", "no_tp_diff" = "#C77CFF"))
```

Percentage difference across sampling schemes



```

first_bias_prec_vals <- compare_auc %>%
  summarize(
    two_tp_bias = mean((tdm_AUC - full_tp_AUC) / full_tp_AUC) * 100,
    one_tp_bias = mean((last_tp_AUC - full_tp_AUC) / full_tp_AUC) * 100,
    no_tp_bias = mean((no_tp_AUC - full_tp_AUC) / full_tp_AUC) * 100,
    two_tp_rmse = sqrt(mean(((tdm_AUC - full_tp_AUC) / full_tp_AUC)^2)) * 100,
    one_tp_rmse = sqrt(mean(((last_tp_AUC - full_tp_AUC) / full_tp_AUC)^2)) * 100,
    no_tp_rmse = sqrt(mean(((no_tp_AUC - full_tp_AUC) / full_tp_AUC)^2)) * 100
  )

# Convert the summarized data frame to a named list, then enframe it
first_value_long <- tibble::enframe(as.list(first_bias_prec_vals)) %>%
  mutate(
    value = as.numeric(unlist(value)), # Ensures `value` is a numeric column
    type = ifelse(grepl("bias", name), "Bias", "RMSE"),
    method = case_when(
      grepl("two_tp", name) ~ "Two TP",
      grepl("one_tp", name) ~ "One TP",
      grepl("no_tp", name) ~ "No TP"
    )
  )

first_value_long <- first_value_long %>% mutate(method = factor(method, levels = c("Two
TP", "One TP", "No TP")))

# Bias plot
first_bias_plot <- first_value_long %>%
  filter(type == "Bias") %>%
  ggplot(aes(x = method, y = value, fill = method)) +
  geom_bar(stat = "identity", width = 0.6) +
  labs(title = "Bias by Method", y = "Bias (%)", x = NULL) +
  theme_minimal() +
  theme(legend.position = "none") +
  scale_fill_manual(values = c("Two TP" = "#00BFC4", "One TP" = "#7CAE00", "No TP" = "#C
77CFF")) +
  scale_color_manual(values = c("Two TP" = "#00BFC4", "One TP" = "#7CAE00", "No TP" = "#
C77CFF"))
first_bias_plot

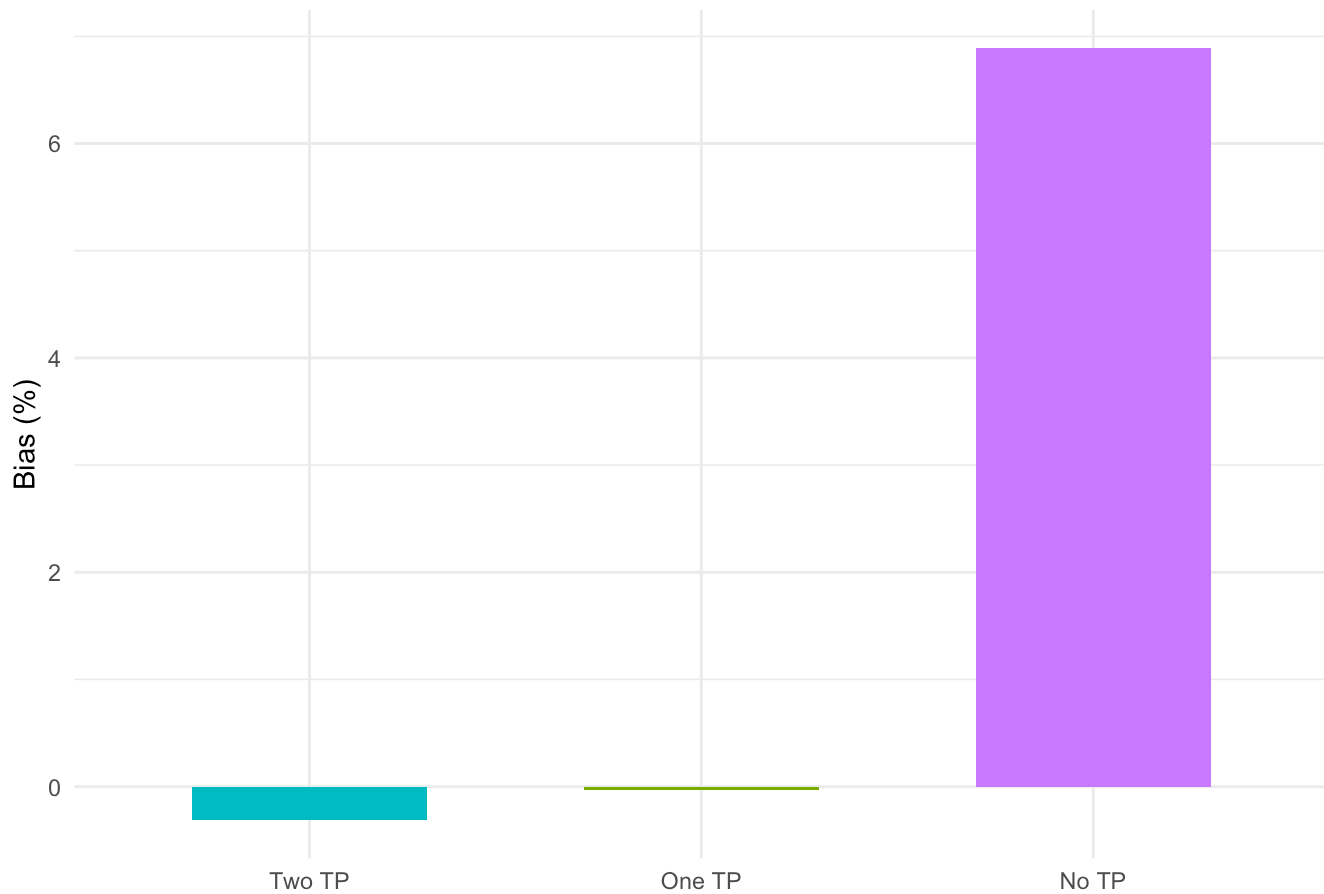
```

```

## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.

```

## Bias by Method



```
first_rmse_plot <- first_value_long %>%
  filter(type == "RMSE") %>%
  ggplot(aes(x = method, y = value, fill = method)) +
  geom_bar(stat = "identity", width = 0.6) +
  labs(title = "RMSE by Method", y = "RMSE (%)", x = NULL) +
  theme_minimal() +
  theme(legend.position = "none") +
  scale_fill_manual(values = c("Two TP" = "#00BFC4", "One TP" = "#7CAE00", "No TP" = "#C77CFF")) +
  scale_color_manual(values = c("Two TP" = "#00BFC4", "One TP" = "#7CAE00", "No TP" = "#C77CFF"))

first_rmse_plot
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
```



RMSE by Method

