Contents

We provide the details of the estimation procedure for the LTMLE and the Cheap Bootstrap/Subsampling confidence interval in the following paragraph. The data generating mechanism may be found in the **data$_{\text{generatingmechanism.R}}$** file, where each regression coefficient corresponds to a regression coefficient in a logistic regression model. Basic implementations of the Cheap Subsampling/Bootstrap confidence interval and the LTMLE are given in the **bootstrap$_{\text{ci.R}}$** file. The specific details of how we use this to simulate from the setup in the simulation study are given in the **get$_{\text{lavamodel.R}}$** file. We now demonstrate how to use each of these files in the following R code:

```
suppressMessages(library(data.table))
library(foreach)
library(lava)
library(targets)
source("get_lava_model.R") ## the get_lava_model function and get_sim_data
source("bootstrap_ci.R") ## cheap bootstrap/subsampling confidence interval
source("data_generating_mechanism.R") ## loads data generating mechanism in
tar_source("Ltmle") ## use targets to source all of the files

## make lava model from coefficients
lava_model <- suppressMessages(get_lava_model(3, coefs))

## set up model object
mm <- list()
mm$outcome <- "heart_failure"
mm$censoring <- "Censored"
mm$comp.event <- "Dead"
mm$treatment <- "GS"
mm$time_covariates <- "insulin"
mm$time_horizon <- 3
mm$baseline <- c("sex","education","agegroups","tertile_income","index_hear
mm$model <- lava_model

## simulate data
sample_size <- 1000
set.seed(6)
data <- get_sim_data(mm, sample_size)
data
```

```
pnr    sex education agegroups tertile_income
      <int> <fctr>    <fctr>    <fctr>         <fctr>
   1:     1 Female    Medium     60-65      Income_q1
   2:     2   Male      High     60-65      Income_q3
   3:     3 Female    Medium     65-70      Income_q1
   4:     4 Female    Medium     80-85      Income_q2
   5:     5   Male    Medium     70-75      Income_q2
  ---
 996:   996 Female      High  below 45      Income_q1
 997:   997   Male      High     70-75      Income_q1
 998:   998 Female      High     45-50      Income_q3
 999:   999   Male    Medium     60-65      Income_q1
1000:  1000 Female    Medium     50-55      Income_q2
      index_heart_failure diabetes_duration
                   <fctr>            <fctr>
   1:                  No          above 10
   2:                  No              5-10
   3:                  No           below 5
   4:                  No              5-10
   5:                  No              5-10
  ---
 996:                  No          above 10
 997:                  No           below 5
 998:                  No              5-10
 999:                  No           below 5
1000:                 Yes          above 10
      secondline_duration first_2ndline insulin_0  GS_0
                   <fctr>        <fctr>     <int> <int>
   1:             above 3         other         1     1
   2:             above 3         other         0     0
   3:             below 1         other         0     0
   4:                 1-3         other         0     0
   5:                 1-3         other         0     1
  ---
 996:             above 3         other         1     1
 997:             above 3         other         0     0
 998:             above 3          glp1         0     0
 999:                 1-3         other         0     0
1000:             below 1         other         0     0
      Censored_1 heart_failure_1 Dead_1 insulin_1  GS_1
```

```
          <int>           <int>  <int>     <int> <int>
   1:         1               0      0         0     0
   2:         1               0      0         0     1
   3:         1               0      0         0     0
   4:         1               0      0         0     0
   5:         1               0      0         1     0
  ---
 996:         1               0      0         0     1
 997:         1               0      0         0     1
 998:         1               0      0         0     0
 999:         1               0      0         0     0
1000:         1               1     NA        NA    NA
      Censored_2 heart_failure_2 Dead_2 insulin_2  GS_2
          <int>           <int>  <int>     <int> <int>
   1:         1               1     NA        NA    NA
   2:         1               0      0         0     0
   3:         1               0      0         0     0
   4:         1               0      0         0     1
   5:         1               0      0         1     0
  ---
 996:         1               0      0         0     0
 997:         0              NA     NA        NA    NA
 998:         1               0      0         0     0
 999:         1               0      0         0     1
1000:         1               1     NA        NA    NA
      Censored_3 heart_failure_3
          <int>           <int>
   1:         1               1
   2:         1               1
   3:         1               0
   4:         1               0
   5:         1               0
  ---
 996:         1               0
 997:         0              NA
 998:         1               0
 999:         1               1
1000:         1               1
```

We now demonstrate how we find parameter estimates with the LTMLE

for our simulated data set. The default behavior of the **prepare$_{\text{Ltmle}}$** is that it uses the entire history, in an additive manner. Note that the LTMLE code has been taken from an altered version of the **tmle** package (Lendle et al., 2017), which may be found here: TMLE for breakfast.

```
## get data for use in Ltmle, i.., split up separately into the outcomes, r
data <- get_sim_data(mm, sample_size, TRUE, data)
## get LTMLE estimates
x <- prepare_Ltmle(
  outcome_data = data$outcome,
  regimen_data = data$regimen,
  baseline_data = data$baseline_covariates,
  timevar_data = data$time_covariates,
  time_horizon = 3,
  censored_label = 0,
  name_outcome = "heart_failure",
  name_regimen = "GS",
  name_censoring = "Censored",
  name_competing_risk = "Dead",
  abar = list(treat=c(1,1,1), control = c(0,0,0)),
  SL.library = "glm",
  verbose = FALSE,
  gbounds = c(0,1)
)
f<-summary(do.call("Ltmle", x))
f
```

```
Target_parameter Estimator   estimate    std.err
             <char>    <char>      <num>      <num>
1:        Mean(A=1)      tmle  0.3115874 0.04845545
2:        Mean(A=0)      tmle  0.4487031 0.05120906
3:              ATE      tmle -0.1371157 0.07003415
4:            Ratio      tmle  0.6944177 0.19167729
        lower        upper       pvalue
        <num>        <num>        <num>
1:  0.2166164 0.4065582972 1.272779e-10
2:  0.3483352 0.5490710128 1.915063e-18
3: -0.2743802 0.0001486821 5.024867e-02
4:  0.4769411 1.0110596389 5.709518e-02
```

We also provide a basic example, showing how the Cheap Subsampling/-

Bootstrap confidence intervals may be used in practice for $B = 5$ and $m = \lfloor 0.632n \rfloor$ for the treatment contrast between those that are treated continuously throughout the period and those that are never treated in the period:

```
bs <- 5
## cheap_subsampling_ci
k_m <- 0.632
m_val <- floor(k_m * sample_size)
res_subsampling <- list()
for (b in seq_len(bs)) {
  ## subsample data of size m
  subsample <- sample(1:sample_size, size = m_val, replace = FALSE)
  formatted_data_sub <-
    lapply(data, function(x) {
      x[subsample, ]
    })
  x <- prepare_Ltmle(
    outcome_data = formatted_data_sub$outcome,
    regimen_data = formatted_data_sub$regimen,
    baseline_data = formatted_data_sub$baseline_covariates,
    timevar_data = formatted_data_sub$time_covariates,
    time_horizon = 3,
    censored_label = 0,
    name_outcome = "heart_failure",
    name_regimen = "GS",
    name_censoring = "Censored",
    name_competing_risk = "Dead",
    abar = list(treat=c(1,1,1), control = c(0,0,0)),
    SL.library = "glm",
    verbose = FALSE,
    gbounds = c(0,1)
  )
  f_temp <- do.call("Ltmle", x)
  res_subsampling[[b]] <- summary(f_temp)[,c(1:3)]
}

res_subsampling <- rbindlist(res_subsampling)

## cheap bootstrap ci
```

```r
print("95% Cheap Subsampling CI:")
get_cheap_subsampling_ci(f[Target_parameter== "ATE", estimate], res_subsam

res_non_parametric_bootstrap <- list()
for (b in seq_len(bs)) {
  ## subsample data of size m
  bootstrap_sample <- sample(1:sample_size, size = sample_size, replace = T
  formatted_data_boot <-
    lapply(data, function(x) {
      temp <- x[bootstrap_sample, ]
      temp[, pnr:= 1:.N]
      temp
    })
  x <- prepare_Ltmle(
    outcome_data = formatted_data_boot$outcome,
    regimen_data = formatted_data_boot$regimen,
    baseline_data = formatted_data_boot$baseline_covariates,
    timevar_data = formatted_data_boot$time_covariates,
    time_horizon = 3,
    censored_label = 0,
    name_outcome = "heart_failure",
    name_regimen = "GS",
    name_censoring = "Censored",
    name_competing_risk = "Dead",
    abar = list(treat=c(1,1,1), control = c(0,0,0)),
    SL.library = "glm",
    verbose = FALSE,
    gbounds = c(0,1)
  )
  f_temp <- do.call("Ltmle", x)
  res_non_parametric_bootstrap[[b]] <- summary(f_temp)[,c(1:3)]
}

## cheap bootstrap_ci
res_non_parametric_bootstrap <- rbindlist(res_non_parametric_bootstrap)

print("95% Cheap Bootstrap CI:")
get_cheap_bootstrap_ci(f[Target_parameter== "ATE", estimate], res_non_paran
```

```
[1] "95% Cheap Subsampling CI:"
```

```
[1] -0.33335041  0.05911894
[1] "95% Cheap Bootstrap CI:"
[1] -0.22397980 -0.05025168
```

References

Lendle, S. D., J. Schwab, M. L. Petersen, and M. J. van der Laan (2017, October). Ltmle: An R Package Implementing Targeted Minimum Loss-Based Estimation for Longitudinal Data. *Journal of Statistical Software 81*, 1–21.