IPW for Categorical Exposure with 4 leves with stremr package

Soudeh

December 3, 2017

## stremr version 0.8.99 and

Installing stremr package from GitHub and loading the packages

# ----------------------------------------------------------------------  
# Instal stremr Version 0.8.99 Data  
# ----------------------------------------------------------------------  
#knitr::opts\_chunk$set(echo = TRUE)  
  
library(devtools)  
#install\_github("osofr/stremr", ref = "experimental\_master")  
  
# ----------------------------------------------------------------------  
# Get Libraries  
# ----------------------------------------------------------------------  
library(stremr)  
library(data.table)  
library(magrittr)  
library(h2o)

##   
## ----------------------------------------------------------------------  
##   
## Your next step is to start H2O:  
## > h2o.init()  
##   
## For H2O package documentation, ask for help:  
## > ??h2o  
##   
## After starting H2O, you can use the Web UI at http://localhost:54321  
## For more information visit http://docs.h2o.ai  
##   
## ----------------------------------------------------------------------

##   
## Attaching package: 'h2o'

## The following objects are masked from 'package:data.table':  
##   
## hour, month, week, year

## The following objects are masked from 'package:stats':  
##   
## cor, sd, var

## The following objects are masked from 'package:base':  
##   
## %\*%, %in%, &&, ||, apply, as.factor, as.numeric, colnames,  
## colnames<-, ifelse, is.character, is.factor, is.numeric, log,  
## log10, log1p, log2, round, signif, trunc

options(stremr.verbose=TRUE)  
sessionInfo()

## R version 3.4.2 (2017-09-28)  
## Platform: x86\_64-w64-mingw32/x64 (64-bit)  
## Running under: Windows 10 x64 (build 16299)  
##   
## Matrix products: default  
##   
## locale:  
## [1] LC\_COLLATE=English\_United States.1252   
## [2] LC\_CTYPE=English\_United States.1252   
## [3] LC\_MONETARY=English\_United States.1252  
## [4] LC\_NUMERIC=C   
## [5] LC\_TIME=English\_United States.1252   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] h2o\_3.14.0.3 magrittr\_1.5 data.table\_1.10.5 stremr\_0.8.99   
## [5] devtools\_1.13.4   
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_0.12.14 assertthat\_0.2.0 digest\_0.6.12 withr\_2.1.0   
## [5] rprojroot\_1.2 bitops\_1.0-6 R6\_2.2.2 jsonlite\_1.5   
## [9] backports\_1.1.1 evaluate\_0.10.1 stringi\_1.1.6 rmarkdown\_1.8   
## [13] tools\_3.4.2 pander\_0.6.1 stringr\_1.2.0 RCurl\_1.95-4.8   
## [17] yaml\_2.1.14 compiler\_3.4.2 memoise\_1.1.0 htmltools\_0.3.6   
## [21] knitr\_1.17

# Get Source Data from another Github Rasp

Read sampleAD.RData from Soudi00 GitHub Rasp

library(repmis)  
  
source\_data("https://github.com/Soudi00/Multi-Treatment-Causal-Modeling/blob/master/sampleAD.RData?raw=True")

## Downloading data from: https://github.com/Soudi00/Multi-Treatment-Causal-Modeling/blob/master/sampleAD.RData?raw=True

## SHA-1 hash of the downloaded data file is:  
## 89e884ef132014c520d5a0bd50ed7c05757ff6e0

## [1] "AD"

AD = as.data.table(AD, key= c(ID\_MPI,SEQ))

use importData to prepare the data to get porpensity scores, also define censoring and exposrure regressions

# ----------------------------------------------------------------------  
# Import Data  
# ----------------------------------------------------------------------  
OData.2 <- importData(AD, ID = "ID", t\_name = "SEQ",   
 covars = c("CAT\_VAR1","CAT\_VAR2","CONT\_VAR1"),   
 CENS = c("CNS","ADM\_CNS"),   
 TRT = "TRTN",  
 MONITOR = NULL, OUTCOME = "STATUS",  
 weights = NULL, remove\_extra\_rows = TRUE,  
 verbose = getOption("stremr.verbose"))

## [1] "stremr will use the following options as defaults: "  
##   
## List of 7  
## $ fit\_method : chr "none"  
## $ fold\_column : NULL  
## $ lower\_bound\_zero\_Q: logi TRUE  
## $ skip\_update\_zero\_Q: logi TRUE  
## $ up\_trunc\_offset : num 20  
## $ low\_trunc\_offset : num -10  
## $ eps\_tol : num 1e-05

## ...converting the following factor(s) to binary dummies (and droping the first factor levels): TRTN,CAT\_VAR1,CAT\_VAR2

# ----------------------------------------------------------------------  
# Look at the input data object  
# ----------------------------------------------------------------------  
print(OData.2)

## <DataStorageClass>  
## Public:  
## addYnode: function (YnodeVals, det.Y)   
## backup.savedGstarsDT: active binding  
## backupNodes: function (nodes)   
## check\_norows\_after\_event: function ()   
## clone: function (deep = FALSE)   
## convert.to.wide: function (bslcovars)   
## curr\_data\_A\_g0: TRUE  
## dat.sVar: active binding  
## define.stoch.nodes: function (NodeNames)   
## define\_CVfolds: function (nfolds = 5, fold\_column = "fold\_ID", seed = NULL)   
## det.Y: NULL  
## emptydat.sVar: active binding  
## eval\_follow\_rule: function (NodeName, gstar.NodeName)   
## eval\_follow\_rule\_each\_t: function (NodeName, gstar.NodeName)   
## eval\_uncensored: function ()   
## eval\_uncensored\_idx: function ()   
## evalsubst: function (subset\_vars, subset\_exprs = NULL)   
## fold\_column: NULL  
## follow\_rule: active binding  
## g\_holdout\_preds: NULL  
## g\_preds: NULL  
## get.dat.sVar: function (rowsubset = TRUE, covars)   
## get.outvar: function (rowsubset = TRUE, var)   
## get.sVar: function (name.sVar)   
## gstarNodes\_stoch: NULL  
## initialize: function (Odata, nodes, YnodeVals, det.Y, noCENScat, ...)   
## interventionNodes.g0: NULL  
## interventionNodes.gstar: NULL  
## IPWeights\_info: NULL  
## IPwts\_by\_regimen: active binding  
## is.sVar.CENS: function (name.sVar)   
## make\_origami\_fold\_from\_column: function (subset\_idx)   
## max.t: active binding  
## min.t: active binding  
## modelfit.gA: NULL  
## modelfit.gC: NULL  
## modelfit.gN: NULL  
## modelfits.g0: NULL  
## names.sVar: active binding  
## ncols.sVar: active binding  
## new.factor.names: list  
## nfolds: NULL  
## nobs: active binding  
## noCENScat: 0  
## nodes: active binding  
## noNA.Ynodevals: active binding  
## nuniqueIDs: active binding  
## nuniquets: active binding  
## Qlearn.fit: NULL  
## replaceNodesVals: function (subset\_idx, nodes\_to\_repl = intervened\_NODE, source\_for\_repl = NodeNames)   
## rescaleNodes: function (subset\_idx, nodes\_to\_rescale, delta)   
## restoreNodes: function (nodes)   
## set.sVar: function (name.sVar, new.sVarVal)   
## swapNodes: function (current, target)   
## type.sVar: function (var\_names, pcontinuous = 0.05)   
## uncensored: active binding  
## YnodeVals: NULL  
## Private:  
## .active.bin.sVar: NULL  
## .follow\_rule: NULL  
## .IPwts: NULL  
## .mat.bin.sVar: NULL  
## .mat.sVar: data.table, data.frame  
## .nodes: list  
## .ord.sVar: NULL  
## .protected.YnodeVals: NULL  
## .saveGstarsDT: NULL  
## .uncensored: NULL

# ----------------------------------------------------------------------  
# Access the input data  
# ----------------------------------------------------------------------  
get\_data(OData.2)  
  
# ----------------------------------------------------------------------  
# Rgression formula for Right Censroing and Adminstrative Censoring and   
# Exposure  
# ----------------------------------------------------------------------  
gform\_CENS <- "CNS + ADM\_CNS ~ CAT\_VAR1 + CONT\_VAR1"  
gform\_TRT = "TRTN ~ CAT\_VAR1 + CAT\_VAR2 + CONT\_VAR1"

# Error in fitPropensity (not meaningful for factors)

I tried diffrent options in fitPropensity but none of them works for categoircal with more than 2 levels

# ----------------------------------------------------------------------  
# Estimate Propensity Scores  
# fitPRopensity score with all defult option has an error  
# ----------------------------------------------------------------------  
  
OData.2 <- fitPropensity(OData.2, gform\_CENS = gform\_CENS,ngform\_TRT = gform\_TRT )

## Using the default regression formula: TRTN ~ CAT\_VAR1\_2+CAT\_VAR1\_3+CAT\_VAR1\_4+CAT\_VAR1\_6+CAT\_VAR1\_7+CAT\_VAR2\_2+CAT\_VAR2\_3+CAT\_VAR2\_4+CAT\_VAR2\_5+CONT\_VAR1

## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(CNS|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(ADM\_CNS|CNS, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: CNS == 0;\\ N: NA"  
## [1] "New 'ModelCategorical' regression defined:"  
## [1] "P(TRTN|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: categorical;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(CNS|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(ADM\_CNS|CNS, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: CNS == 0;\\ N: NA"  
## [1] "fitting the model: P(TRTN|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: categorical;\\ Stratify: ;\\ N: NA"

## Failed on Lrnr\_condensier\_c("equal.mass", "equal.len", "dhist")\_5\_20\_FALSE\_NA\_FALSE\_NULL

##   
## sl3 error debugging info:  
## [1] "Error in Summary.factor(structure(c(3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, : \n 'max' not meaningful for factors\n"  
## attr(,"class")  
## [1] "try-error"  
## attr(,"condition")  
## <simpleError in Summary.factor(structure(c(3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 2L, 1L, 2L, 1L, 1L, 2L, 2L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 1L, 2L, 2L, 2L, 2L, 1L, 2L, 2L, 2L, 1L, 1L, 1L, 2L, 2L, 1L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 3L, 1L, 3L, 1L, 3L, 1L, 1L, 1L, 1L, 3L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 1L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 2L, 2L, 3L, 1L, 1L, 1L, 2L, 2L, 3L, 4L, 2L, 2L, 2L, 2L, 3L, 1L, 2L, 3L, 3L, 1L, 1L, 1L, 2L, 2L, 3L, 4L, 3L, 2L, 3L, 1L, 2L, 3L, 2L, 3L, 1L, 1L, 2L, 2L, 2L, 2L, 3L, 1L, 2L, 2L, 3L, 2L, 2L, 2L, 3L, 1L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 2L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 3L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 3L, 1L, 3L, 3L, 3L, 1L, 2L, 1L, 1L, 2L, 1L, 3L, 4L, 2L, 2L, 2L, 2L, 2L, 4L, 2L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 4L, 4L, 2L, 4L, 2L, 2L, 4L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 4L, 2L, 2L, 2L), .Label = c("1", "2", "3", "4"), class = "factor"), na.rm = FALSE): 'max' not meaningful for factors>

## ...trying to run Lrnr\_glm\_fast as a backup...

## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Warning in Ops.factor(eta, offset): '-' not meaningful for factors

## Warning in Ops.factor(y, mu): '-' not meaningful for factors  
  
## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Warning in Ops.factor(weights, y): '\*' not meaningful for factors

## Warning in Ops.factor(y, mu): '-' not meaningful for factors  
  
## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Error in private$PsAsW.models[[k\_i]]$predictAeqa(newdata = newdata, n = n, : some of the modeling predictions resulted in NAs, which indicates an error in a prediction routine

# tried modeing treatmtnet with Gradient Boosting machines same error

# ----------------------------------------------------------------------  
# Fitting treatment model with Gradient Boosting machines:  
# ----------------------------------------------------------------------  
require("h2o")  
h2o::h2o.init(nthreads = -1)

## Connection successful!  
##   
## R is connected to the H2O cluster:   
## H2O cluster uptime: 23 minutes 4 seconds   
## H2O cluster version: 3.14.0.3   
## H2O cluster version age: 2 months and 11 days   
## H2O cluster name: H2O\_started\_from\_R\_Soudeh\_hzo489   
## H2O cluster total nodes: 1   
## H2O cluster total memory: 0.81 GB   
## H2O cluster total cores: 4   
## H2O cluster allowed cores: 4   
## H2O cluster healthy: TRUE   
## H2O Connection ip: localhost   
## H2O Connection port: 54321   
## H2O Connection proxy: NA   
## H2O Internal Security: FALSE   
## H2O API Extensions: Algos, AutoML, Core V3, Core V4   
## R Version: R version 3.4.2 (2017-09-28)

gform\_CENS <- "CNS + ADM\_CNS ~ CAT\_VAR1 + CONT\_VAR1"  
models\_TRT <- sl3::Lrnr\_h2o\_grid$new(algorithm = "gbm")  
OData.2 <- fitPropensity(OData.2, gform\_CENS = gform\_CENS,  
 gform\_TRT = gform\_TRT,  
 models\_TRT = models\_TRT)

## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(CNS|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(ADM\_CNS|CNS, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: CNS == 0;\\ N: NA"  
## [1] "New 'ModelCategorical' regression defined:"  
## [1] "P(TRTN|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: categorical;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(CNS|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(ADM\_CNS|CNS, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: CNS == 0;\\ N: NA"  
## [1] "fitting the model: P(TRTN|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: categorical;\\ Stratify: ;\\ N: NA"

## Failed on Lrnr\_condensier\_c("equal.mass", "equal.len", "dhist")\_5\_20\_FALSE\_NA\_FALSE\_NULL

##   
## sl3 error debugging info:  
## [1] "Error in Summary.factor(structure(c(3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, : \n 'max' not meaningful for factors\n"  
## attr(,"class")  
## [1] "try-error"  
## attr(,"condition")  
## <simpleError in Summary.factor(structure(c(3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 2L, 1L, 2L, 1L, 1L, 2L, 2L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 1L, 2L, 2L, 2L, 2L, 1L, 2L, 2L, 2L, 1L, 1L, 1L, 2L, 2L, 1L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 3L, 1L, 3L, 1L, 3L, 1L, 1L, 1L, 1L, 3L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 1L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 2L, 2L, 3L, 1L, 1L, 1L, 2L, 2L, 3L, 4L, 2L, 2L, 2L, 2L, 3L, 1L, 2L, 3L, 3L, 1L, 1L, 1L, 2L, 2L, 3L, 4L, 3L, 2L, 3L, 1L, 2L, 3L, 2L, 3L, 1L, 1L, 2L, 2L, 2L, 2L, 3L, 1L, 2L, 2L, 3L, 2L, 2L, 2L, 3L, 1L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 2L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 3L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 3L, 1L, 3L, 3L, 3L, 1L, 2L, 1L, 1L, 2L, 1L, 3L, 4L, 2L, 2L, 2L, 2L, 2L, 4L, 2L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 4L, 4L, 2L, 4L, 2L, 2L, 4L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 4L, 2L, 2L, 2L), .Label = c("1", "2", "3", "4"), class = "factor"), na.rm = FALSE): 'max' not meaningful for factors>

## ...trying to run Lrnr\_glm\_fast as a backup...

## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Warning in Ops.factor(eta, offset): '-' not meaningful for factors

## Warning in Ops.factor(y, mu): '-' not meaningful for factors  
  
## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Warning in Ops.factor(weights, y): '\*' not meaningful for factors

## Warning in Ops.factor(y, mu): '-' not meaningful for factors  
  
## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Error in private$PsAsW.models[[k\_i]]$predictAeqa(newdata = newdata, n = n, : some of the modeling predictions resulted in NAs, which indicates an error in a prediction routine

# Use `H2O-3` distributed implementation of GLM for treatment model estimator:  
models\_TRT <- sl3::Lrnr\_h2o\_glm$new(family = "multinomial")  
OData.2 <- fitPropensity(OData.2, gform\_CENS = gform\_CENS,  
 gform\_TRT = gform\_TRT,  
 models\_TRT = models\_TRT)

## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(CNS|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(ADM\_CNS|CNS, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: CNS == 0;\\ N: NA"  
## [1] "New 'ModelCategorical' regression defined:"  
## [1] "P(TRTN|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: categorical;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(CNS|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(ADM\_CNS|CNS, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: CNS == 0;\\ N: NA"  
## [1] "fitting the model: P(TRTN|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: categorical;\\ Stratify: ;\\ N: NA"

## Failed on Lrnr\_condensier\_c("equal.mass", "equal.len", "dhist")\_5\_20\_FALSE\_NA\_FALSE\_NULL

##   
## sl3 error debugging info:  
## [1] "Error in Summary.factor(structure(c(3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, : \n 'max' not meaningful for factors\n"  
## attr(,"class")  
## [1] "try-error"  
## attr(,"condition")  
## <simpleError in Summary.factor(structure(c(3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 2L, 1L, 2L, 1L, 1L, 2L, 2L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 1L, 2L, 2L, 2L, 2L, 1L, 2L, 2L, 2L, 1L, 1L, 1L, 2L, 2L, 1L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 3L, 1L, 3L, 1L, 3L, 1L, 1L, 1L, 1L, 3L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 1L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 2L, 2L, 3L, 1L, 1L, 1L, 2L, 2L, 3L, 4L, 2L, 2L, 2L, 2L, 3L, 1L, 2L, 3L, 3L, 1L, 1L, 1L, 2L, 2L, 3L, 4L, 3L, 2L, 3L, 1L, 2L, 3L, 2L, 3L, 1L, 1L, 2L, 2L, 2L, 2L, 3L, 1L, 2L, 2L, 3L, 2L, 2L, 2L, 3L, 1L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 2L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 3L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 3L, 1L, 3L, 3L, 3L, 1L, 2L, 1L, 1L, 2L, 1L, 3L, 4L, 2L, 2L, 2L, 2L, 2L, 4L, 2L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 4L, 4L, 2L, 4L, 2L, 2L, 4L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 4L, 2L, 2L, 2L), .Label = c("1", "2", "3", "4"), class = "factor"), na.rm = FALSE): 'max' not meaningful for factors>

## ...trying to run Lrnr\_glm\_fast as a backup...

## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Warning in Ops.factor(eta, offset): '-' not meaningful for factors

## Warning in Ops.factor(y, mu): '-' not meaningful for factors  
  
## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Warning in Ops.factor(weights, y): '\*' not meaningful for factors

## Warning in Ops.factor(y, mu): '-' not meaningful for factors  
  
## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Error in private$PsAsW.models[[k\_i]]$predictAeqa(newdata = newdata, n = n, : some of the modeling predictions resulted in NAs, which indicates an error in a prediction routine

# Use Deep Neural Nets:  
models\_TRT <- sl3::Lrnr\_h2o\_grid$new(algorithm = "deeplearning")  
OData.2 <- fitPropensity(OData.2, gform\_CENS = gform\_CENS,  
 gform\_TRT = gform\_TRT,  
 models\_TRT = models\_TRT)

## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(CNS|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "New 'ModelBinomial' regression defined:"  
## [1] "P(ADM\_CNS|CNS, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: CNS == 0;\\ N: NA"  
## [1] "New 'ModelCategorical' regression defined:"  
## [1] "P(TRTN|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: categorical;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(CNS|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: ;\\ N: NA"  
## [1] "fitting the model: P(ADM\_CNS|CNS, CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CONT\_VAR1);\\ outvar.class: binomial;\\ Stratify: CNS == 0;\\ N: NA"  
## [1] "fitting the model: P(TRTN|CAT\_VAR1\_2, CAT\_VAR1\_3, CAT\_VAR1\_4, CAT\_VAR1\_6, CAT\_VAR1\_7, CAT\_VAR2\_2, CAT\_VAR2\_3, CAT\_VAR2\_4, CAT\_VAR2\_5, CONT\_VAR1);\\ outvar.class: categorical;\\ Stratify: ;\\ N: NA"

## Failed on Lrnr\_condensier\_c("equal.mass", "equal.len", "dhist")\_5\_20\_FALSE\_NA\_FALSE\_NULL

##   
## sl3 error debugging info:  
## [1] "Error in Summary.factor(structure(c(3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, : \n 'max' not meaningful for factors\n"  
## attr(,"class")  
## [1] "try-error"  
## attr(,"condition")  
## <simpleError in Summary.factor(structure(c(3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 2L, 1L, 2L, 1L, 1L, 2L, 2L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 2L, 1L, 2L, 2L, 2L, 2L, 1L, 2L, 2L, 2L, 1L, 1L, 1L, 2L, 2L, 1L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 3L, 1L, 3L, 1L, 3L, 1L, 1L, 1L, 1L, 3L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 1L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 2L, 2L, 3L, 1L, 1L, 1L, 2L, 2L, 3L, 4L, 2L, 2L, 2L, 2L, 3L, 1L, 2L, 3L, 3L, 1L, 1L, 1L, 2L, 2L, 3L, 4L, 3L, 2L, 3L, 1L, 2L, 3L, 2L, 3L, 1L, 1L, 2L, 2L, 2L, 2L, 3L, 1L, 2L, 2L, 3L, 2L, 2L, 2L, 3L, 1L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 2L, 1L, 3L, 3L, 3L, 1L, 1L, 3L, 3L, 3L, 1L, 3L, 3L, 3L, 3L, 1L, 3L, 1L, 1L, 3L, 1L, 3L, 3L, 3L, 1L, 2L, 1L, 1L, 2L, 1L, 3L, 4L, 2L, 2L, 2L, 2L, 2L, 4L, 2L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 4L, 4L, 2L, 4L, 2L, 2L, 4L, 2L, 4L, 4L, 4L, 2L, 2L, 2L, 4L, 4L, 2L, 2L, 2L), .Label = c("1", "2", "3", "4"), class = "factor"), na.rm = FALSE): 'max' not meaningful for factors>

## ...trying to run Lrnr\_glm\_fast as a backup...

## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Warning in Ops.factor(eta, offset): '-' not meaningful for factors

## Warning in Ops.factor(y, mu): '-' not meaningful for factors  
  
## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Warning in Ops.factor(weights, y): '\*' not meaningful for factors

## Warning in Ops.factor(y, mu): '-' not meaningful for factors  
  
## Warning in Ops.factor(y, mu): '-' not meaningful for factors

## Error in private$PsAsW.models[[k\_i]]$predictAeqa(newdata = newdata, n = n, : some of the modeling predictions resulted in NAs, which indicates an error in a prediction routine