Supplementary Material

Using the SuperLearner Package for Dose-Response Estimation

```
packages <- c("data.table","tidyverse","skimr","here","rmutil")


for (package in packages) {
  if (!require(package, character.only=T, quietly=T)) {
    install.packages(package, repos='http://lib.stat.cmu.edu/R/CRAN')
  }
}


for (package in packages) {
  library(package, character.only=T)
}


remotes::install_github("ecpolley/SuperLearner")
library(SuperLearner)


thm <- theme_classic() +
  theme(
    legend.position = "top",
    legend.background = element_rect(fill = "transparent", colour = NA),
    legend.key = element_rect(fill = "transparent", colour = NA)
  )
theme_set(thm)


library(here)
```

```r
# set the seed for reproducibility
set.seed(123)

# generate the observed data
n=5000
x = runif(n,0,8)
y = 5 + 4*sqrt(9 * x)*as.numeric(x<2) + as.numeric(x>=2)*(abs(x-6)^(2)) + rlaplace(n)

# to plot the true dose-response curve, generate sequence of 'doses' from 0 to 8 at every 0.1,
#   then generate the true outcome
xl<-seq(0,8,.1)
yl<-5 + 4 * sqrt(9 * xl)*as.numeric(xl<2) + as.numeric(xl>=2)*(abs(xl-6)^(2))

D<-data.frame(x,y)      # observed data
Dl<-data.frame(xl,yl)   # for plotting the true dose-response curve

# Specify the number of folds for V-fold cross-validation
folds= 5
index<-split(1:1000,1:folds)
splt<-lapply(1:folds,function(ind) D[index[[ind]],])

#-------------------------------------------------------------------------------
# Fit using the SuperLearner Package
#-------------------------------------------------------------------------------
# Create the 5 df GAMs and size = 5 NNET using functions called from programs "sourced" above

ranger_learner = create.Learner("SL.ranger", params = list(min.node.size = 50),
                                tune = list(num.trees = c(200,500,1000)))
xgboost_learner = create.Learner("SL.xgboost", params = list(minobspernode = 50),
                                 tune = list(shrinkage = c(.01,.1,.2)))
gam_learner = create.Learner("SL.gam", tune = list(deg.gam = c(4,5)))
```

```r
# Specifying the SuperLearner library of candidate algorithms
sl.lib <- c(gam_learner$names,"SL.nnet","SL.earth",ranger_learner$names,xgboost_learner$names)

# Fit using the SuperLearner package, specify
#       outcome-for-prediction (y), the predictors (x), the loss function (L2),
#       the library (sl.lib), and number of folds
fitY<-SuperLearner(Y = y,
                   X = data.frame(x),
                   newX = data.frame(x),
                   family=gaussian,
                   method="method.NNLS",
                   SL.library=sl.lib,
                   verbose=T)

# View the output: 'Risk' column returns the CV-MSE estimates
#       'Coef' column gives the weights for the final SuperLearner (meta-learner)
fitY
```

```
##
## Call:
## SuperLearner(Y = y, X = data.frame(x), newX = data.frame(x),
##     family = gaussian, SL.library = sl.lib,
##     method = "method.NNLS", verbose = T)
##
##
##                       Risk       Coef
## SL.gam_1_All     3.094141 0.00000000
## SL.gam_2_All     2.409914 0.04359822
## SL.nnet_All      8.785827 0.00000000
## SL.earth_All     2.238790 0.10822286
## SL.ranger_1_All  2.108977 0.00000000
## SL.ranger_2_All  2.106633 0.00000000
## SL.ranger_3_All  2.106755 0.00000000
```

```
## SL.xgboost_1_All 1.998723 0.84817892
## SL.xgboost_2_All 2.052846 0.00000000
## SL.xgboost_3_All 2.097306 0.00000000
```

```
# Now predict the outcome for all possible x 'doses'
pred_obj <- predict(fitY,newdata=data.frame(x=xl),onlySL=F)


head(pred_obj$library.predict)
```

```
##      SL.gam_1_All SL.gam_2_All SL.nnet_All
## [1,]     11.42452     9.981661    16.76303
## [2,]     11.98981    10.738574    16.76289
## [3,]     12.55409    11.493800    16.76272
## [4,]     13.11512    12.243789    16.76248
## [5,]     13.66968    12.983878    16.76216
## [6,]     14.21398    13.708757    16.76173
##      SL.earth_All SL.ranger_1_All
## [1,]     8.525209        6.477074
## [2,]     9.417256        8.648719
## [3,]    10.309303       11.119712
## [4,]    11.201350       11.634780
## [5,]    12.093397       12.431109
## [6,]    12.985444       13.385973
##      SL.ranger_2_All SL.ranger_3_All
## [1,]        6.515110        6.508634
## [2,]        8.611866        8.609163
## [3,]       11.130376       11.154168
## [4,]       11.769592       11.776413
## [5,]       12.448360       12.440098
## [6,]       13.474584       13.514588
##      SL.xgboost_1_All SL.xgboost_2_All
## [1,]         7.328457         7.307293
## [2,]         8.693849         8.725897
## [3,]        11.012263        11.813244
```

```
## [4,]        11.846663        11.568146
## [5,]        12.557858        12.474278
## [6,]        13.559600        13.374663
##      SL.xgboost_3_All
## [1,]         7.298950
## [2,]         8.692555
## [3,]        11.927782
## [4,]        11.488048
## [5,]        12.443556
## [6,]        13.310889
```

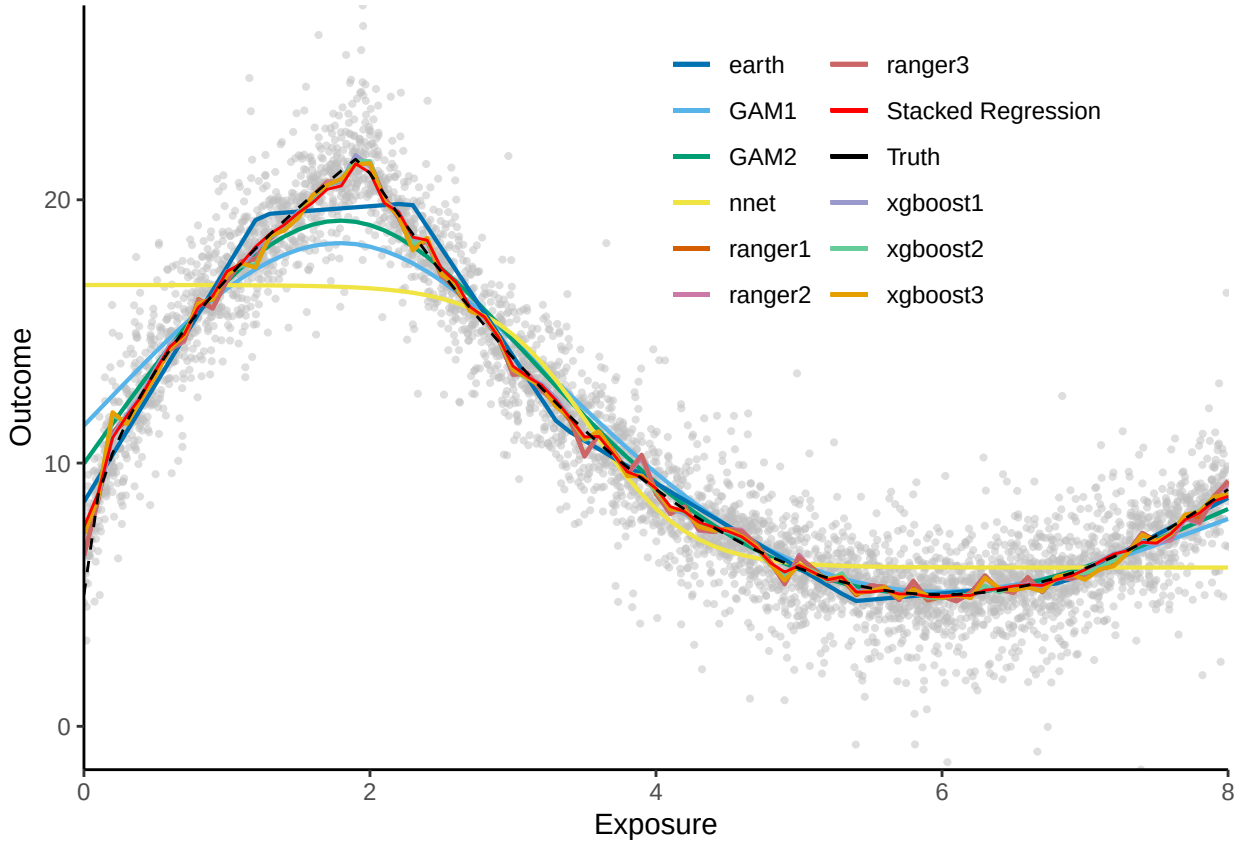
```
head(pred_obj$pred)
```

```
##           [,1]
## [1,]  7.573648
## [2,]  8.861284
## [3,] 10.957181
## [4,] 11.794140
## [5,] 12.526166
## [6,] 13.503966
```

```
yS<-pred_obj$pred


# Create a dataframe of all x 'doses' and predicted SL responses
Dl1       <- data.frame(xl,yS)
Dgam1     <- data.frame(xl,y_pred=pred_obj$library.predict[,1])
Dgam2     <- data.frame(xl,y_pred=pred_obj$library.predict[,2])
Dnnet     <- data.frame(xl,y_pred=pred_obj$library.predict[,3])
Dearth    <- data.frame(xl,y_pred=pred_obj$library.predict[,4])
Dranger1  <- data.frame(xl,y_pred=pred_obj$library.predict[,5])
Dranger2  <- data.frame(xl,y_pred=pred_obj$library.predict[,6])
Dranger3  <- data.frame(xl,y_pred=pred_obj$library.predict[,7])
Dxgboost1 <- data.frame(xl,y_pred=pred_obj$library.predict[,8])
Dxgboost2 <- data.frame(xl,y_pred=pred_obj$library.predict[,9])
```

```
Dxgboost3 <- data.frame(xl,y_pred=pred_obj$library.predict[,10])


cols <- c("Truth"="black",
          "Stacked Regression"="red",
          "GAM1"="#56B4E9",
          "GAM2"= "#009E73",
          "nnet"="#F0E442",
          "earth"="#0072B2",
          "ranger1"="#D55E00",
          "ranger2"="#CC79A7",
          "ranger3"="#CC6666",
          "xgboost1"="#9999CC",
          "xgboost2"="#66CC99",
          "xgboost3"="#E69F00")


ggplot() +
  geom_point(data=D, aes(x,y),color="gray75",alpha=.5,size=.75) +
  geom_line(data=Dgam1, aes(xl,y_pred,color="GAM1"),size=.75, linetype=1) +
  geom_line(data=Dgam2, aes(xl,y_pred,color="GAM2"),size=.75, linetype=1) +
  geom_line(data=Dnnet, aes(xl,y_pred,color="nnet"),size=.75, linetype=1) +
  geom_line(data=Dearth, aes(xl,y_pred,color="earth"),size=.75, linetype=1) +
  geom_line(data=Dranger1, aes(xl,y_pred,color="ranger1"),size=.75, linetype=1) +
  geom_line(data=Dranger2, aes(xl,y_pred,color="ranger2"),size=.75, linetype=1) +
  geom_line(data=Dranger3, aes(xl,y_pred,color="ranger3"),size=.75, linetype=1) +
  geom_line(data=Dxgboost1, aes(xl,y_pred,color="xgboost1"),size=.75, linetype=1) +
  geom_line(data=Dxgboost2, aes(xl,y_pred,color="xgboost2"),size=.75, linetype=1) +
  geom_line(data=Dxgboost3, aes(xl,y_pred,color="xgboost3"),size=.75, linetype=1) +
  geom_line(data=Dl1, aes(xl,yS,color="Stacked Regression"),size=.5, linetype=1) +
  geom_line(data=Dl, aes(xl,yl,color="Truth"),size=.5,linetype=2) +
  scale_colour_manual(name="",values=cols) +
  scale_x_continuous(expand=c(0,0)) +
  scale_y_continuous(expand=c(0,0)) +
```

```
  theme_classic() + theme(legend.position=c(.7,.8)) +
  guides(color=guide_legend(ncol=2)) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  labs(x = "Exposure",y = "Outcome")
```




```
ggsave(here("figures","promo_fig1.pdf"),
       width=6,
       height=5,
       units="in")
```