```
rm(list = ls())
time_doc1 <- Sys.time()
install.packages(c("devtools", "ggplot2", "repmis","xgboost", "e1071",
                   "RColorBrewer","SuperLearner","gam","glmnet","polspline","knitr"))
library(SuperLearner)
library(expose)
library(ggplot2)
library(expose)
library(ggplot2)
library(knitr)
library(RColorBrewer)
library(repmis)
library(RColorBrewer)
library(devtools)

devtools::install_github("itamuria/expose")

# Download data
source_data("https://github.com/itamuria/expose_dataset/blob/master/20180818_dtaset.RData?raw=true")
set.seed(22222)

# Define parameters

N <- dim(dtaset)[1]
Outcome="Y4"
seku <- c(0,0.1,0.2,0.25,0.3,0.4,0.5,0.6,0.7,0.75,0.8,0.9,1)
our.num.sim <- 50
delta=c(0,1)

fecha <- format(Sys.time(), "%Y%m%d%X")
fecha <- gsub(":","",fecha)

Exposures<- c("Var1","Var2","Var3","Var4","Var5")
Confounders<- c("sex")


###############################################################################################################

# Crossvalidation
###############################################################################################################


#Choose the library of algorithms to include in the SL

ourlibraries <- c("SL.glm", "SL.glm.interaction","SL.glmnet", "SL.gam","SL.nnet",
                  "SL.polymars","SL.svm","SL.xgboost")


#Performing SL Cross-validation

# Outcome 1
Outcome <- c("Y1")
m1 <- SuperLearner::CV.SuperLearner(Y = dtaset[,Outcome[1]],
                                    X = dtaset[, c(Confounders,Exposures)],V=10, SL.library = ourlibraries,
                                    family = "gaussian", method = "method.NNLS",
                                    verbose = FALSE)

m1$AllSL[[1]]

# Outcome 2
Outcome <- c("Y2")
m2 <- SuperLearner::CV.SuperLearner(Y = dtaset[,Outcome[1]],
                                    X = dtaset[, c(Confounders,Exposures)],V=10, SL.library = ourlibraries,
                                    family = "gaussian", method = "method.NNLS",
                                    verbose = FALSE)

m2$AllSL[[1]]

# Outcome 3
Outcome <- c("Y3")
m3 <- SuperLearner::CV.SuperLearner(Y = dtaset[,Outcome[1]],
                                    X = dtaset[, c(Confounders,Exposures)],V=10, SL.library = ourlibraries,
                                    family = "gaussian", method = "method.NNLS",
                                    verbose = FALSE)
```

```
m3$AllSL[[1]]

# Outcome 4
Outcome <- c("Y4")
m4 <- SuperLearner::CV.SuperLearner(Y = dtaset[,Outcome[1]],
                                    X = dtaset[, c(Confounders,Exposures)],V=10, SL.library = ourlibraries,
                                    family = "gaussian", method = "method.NNLS",
                                    verbose = FALSE)

m4$AllSL[[1]]

###############################################################################################################

# create basic data frame
###############################################################################################################


Outcome <- c("Y4")

gen <- general_function (dataset = dtaset, exposures = Exposures,
                         confounders = Confounders,
                         outcomes = Outcome[1], delta=delta, dr = seku)


summary_table_lines <- gen[[2]]



###############################################################################################################

# Simulations
###############################################################################################################


t1 <- Sys.time()
simu <- run_simulations (dataset = dtaset, exposures = Exposures,
                         confounders = Confounders, libraries = ourlibraries,
                         outcomes = Outcome[1], num.sim = our.num.sim, delta=delta, dr = seku,
                         newdata =gen[[1]], show_times = TRUE, verbose = FALSE, save_time = TRUE,
                         show_num_sim = TRUE, family = "gaussian", method ="method.NNLS")

t2 <- Sys.time()
(t2-t1)


###############################################################################################################

#  analysis
###############################################################################################################



################################# risk

ris <- simu[[2]]

len_sim <- dim(ris)[2]

lib <- gsub("SL.","",ris$libraries)
len_lib <- length(lib)

df1 <- data.frame(matrix(NA,1,2))
names(df1) <- c("Library","Value")

for(li in 1:len_lib)
{
  df2 <- data.frame(rep(lib[li],len_sim-1),t(ris[li,2:len_sim]))
  names(df2) <- c("Library","Value")
  df1 <- rbind(df1,df2)
}
df1 <- df1[-1,]

ggplot(df1, aes(x = Library, y = Value, fill = Library)) + geom_boxplot()  + theme_bw() +
  coord_flip() + ggtitle("Risk per method (minimize)") + theme(legend.position="none")
```

```
ris_table <- data.frame(lib,apply(ris[,-1],1,mean))
names(ris_table)<-c("Library","Performance")
knitr::kable(ris_table)


################################# Coefficient

coef <- simu[[3]]

coef_table <- data.frame(lib,apply(coef[,-1],1,mean))
names(coef_table)<-c("Library","Estimates")
knitr::kable(coef_table)

df1 <- data.frame(matrix(NA,1,2))
names(df1) <- c("Library","Value")

for(li in 1:len_lib)
{
  df2 <- data.frame(rep(lib[li],len_sim-1),t(coef[li,2:len_sim]))
  names(df2) <- c("Library","Value")
  df1 <- rbind(df1,df2)
}
df1 <- df1[-1,]

ggplot(df1, aes(x = Library, y = Value, fill = Library)) + geom_boxplot() + theme_bw() +
  coord_flip() + ggtitle("Coefficient per method (weight)") + theme(legend.position="none")


###############################################################################################################

# Naive ACE: Average causal effect
###############################################################################################################



ace.df.g <- naive_ace (allsim = simu[[1]], dataset = dtaset, ic_dis = "IC", st = summary_table_lines,
                       exposures = Exposures, delta = delta)

knitr::kable(ace.df.g)

# Use 95% confidence interval instead of SEM
pd <- position_dodge(0.1)
ggplot(ace.df.g, aes(x=Group, y=Mean, colour=Group)) +
  geom_errorbar(aes(ymin=ICa, ymax=ICb), width=.1, position=pd) +
  geom_line(position=pd) +
  geom_point(position=pd) + ylab("Estimated (Boostrapped 95% CI") +
  xlab("Library") + geom_hline(yintercept = 0)+ theme_bw() +
  ggtitle("Average Treatment Effect") + theme(legend.position="none")

###############################################################################################################

# Dose Respond
###############################################################################################################



drr.grp <- dose_resp (allsim = simu[[1]], dataset = dtaset, st = summary_table_lines,
                      dr = seku, exposures = Exposures)

knitr::kable(head(drr.grp))
drr.grp$dose <- as.numeric(gsub("DR_","",drr.grp$Quantile))

pd <- position_dodge(0.1)

ggplot(drr.grp, aes(x=dose, y=Mean)) +
  geom_errorbar(aes(ymin=Mean-SE, ymax=Mean+SE), width=.025, position=pd, size=0.5, color="blue") +
geom_hline(yintercept = 0) +
  geom_line(position=pd,col="blue") +
  geom_point(position=pd, size=2, shape=20, fill="black") + facet_grid(Exp ~ ., scales="free") +
  ggtitle("Dose response") + theme(legend.position="none") + theme_bw()


ggplot(drr.grp, aes(x=dose, y=Mean)) +
  geom_errorbar(aes(ymin=Mean-SE, ymax=Mean+SE), width=.05, position=pd, size=0.5, color="blue") +
  geom_line(position=pd,col="blue") +
  geom_point(position=pd, size=2, shape=20, fill="black") + facet_grid(Exp ~ .) + geom_hline(yintercept = 0)
```

```
+
  ggtitle("Dose response") + theme(legend.position="none") + theme_bw()

###############################################################################################################

# ice: individual causal effect
###############################################################################################################


ice_res <- ice(allsim = simu[[1]], dataset = dtaset, dr = seku, squem = summary_table_lines, remove_extrem =
FALSE)

# ploting
p=ggplot(ice_res,aes(X, Var1_pred,group=id))
p+geom_line()+theme_bw() +
  scale_x_discrete(breaks=sort(unique(ice_res$X)),
                   labels=as.character(seku))

# name to save
h <- 1
file_exp_name <- paste0("Invid_",h,".jpg")

ggsave(file_exp_name)
unlink(file_exp_name)

# coloring
p=ggplot(ice_res,aes(X, Var2_pred,group=id, color=Var3))
p+geom_line()+ theme_bw() + theme(legend.position="none") + ggtitle("Individual effect") +
  scale_x_discrete(breaks=sort(unique(ice_res$X)),
                   labels=as.character(seku))


p=ggplot(ice_res,aes(X, Var1_pred,group=id,col=Var3)) + labs(x="Var1 percentiles",y="Predicted Y")
p+geom_line()+theme_bw()+scale_color_gradientn(name="Var3",colours=rev(brewer.pal(9,"YlOrRd"))) +
  scale_x_discrete(breaks=sort(unique(ice_res$X)),
                   labels=as.character(seku)) + ggtitle("Individual effect")


p=ggplot(ice_res,aes(X, Var3_pred,group=id,col=Var5)) + labs(x="Var3 percentiles",y="Predicted Y")
p+geom_line()+theme_bw()+scale_color_gradientn(name="Var5",colours=rev(brewer.pal(9,"YlOrRd"))) +
  scale_x_discrete(breaks=sort(unique(ice_res$X)),
                   labels=as.character(seku)) + ggtitle("Individual effect")

# discrete variables

p=ggplot(ice_res,aes(X, Var3_pred,group=id,col=sex)) + labs(x="pp-Var3 percentiles",y="Predicted Y")
p+geom_line()+theme_bw()+scale_color_gradientn(name="Sex",colours=c("red","#FFFF00")) +
  theme(axis.text.x = element_text(face="bold",size=8, angle=0), legend.position = "none") +
  scale_x_discrete(breaks=sort(unique(ice_res$X)),
                   labels=as.character(seku)) + ggtitle("Individual effect")

###############################################################################################################

# Interactions
###############################################################################################################



it <- interact (allsim = simu[[1]], dataset = dtaset,exposures = Exposures,
                confounders = c("sex"), squem = summary_table_lines)

knitr::kable(head(it))

pd <- position_dodge(0.1)
ggplot(it, aes(x=Interaction, y=Mean, colour=Interaction)) +
  geom_errorbar(aes(ymin=Mean-SD, ymax=Mean+SD), width=.1, position=pd) +
  geom_line(position=pd) +
  geom_point(position=pd) + ylab("Estimated (Boostrapped 95% CI") +
  xlab("Interaction") + coord_flip() + geom_hline(yintercept = 0) +
  ggtitle("Interaction") + theme_bw() + theme(legend.position="none")

time_doc3 <- Sys.time()

print(time_doc3 - time_doc1)
```