Machine Learning Methods for Site-specific Input Management

Shunkei Kakimto

library(knitr)  
# library(here)  
  
knitr::opts\_chunk$set(  
 cache = FALSE,  
 warning = FALSE,  
 message = FALSE  
 )  
  
# opts\_knit$set(root.dir=here())  
opts\_knit$set(root.dir = "~/Dropbox/ResearchProject/ML\_VRA")

# === packages ===#  
library(sf)  
library(data.table)  
library(RColorBrewer)  
library(patchwork)  
library(grf)  
library(magrittr)  
library(ggplot2)  
library(ggthemes)  
library(viridis)  
library(tidyverse)  
library(ggpubr)  
library(flextable)  
library(officer)  
library(officedown)  
library(modelsummary)  
library(here)  
library(gridExtra)  
library(DiagrammeR)  
library(latex2exp)

# Load the data

# setwd(here())  
source("./GitControlled/Codes/0\_0\_functions.R")  
# source("./GitControlled/Codes/functions\_stepwise\_vs\_base.R")

pCorn <- price\_table[2, pCorn]  
pN <- price\_table[2, pN]  
  
sp\_range <- 400  
  
field <- readRDS("./Shared/Data/for\_Simulations/field\_padding.rds") %>%  
 filter(padding==1)%>%  
 dplyr::select(unique\_cell\_id)  
  
  
####==== (1) cell-level dataset ====####  
# coef\_data <- readRDS("./Shared/Data/for\_Simulations/cnn\_coefficients\_sprange\_400.rds")  
  
# x=1  
# coef\_data\_m <- coef\_data[sim == x, ]  
# coef\_data\_t <- coef\_data[sim == ifelse(x + 1 >= max(sim), 1, x + 1), ]  
  
# sample\_dt <- prepare\_raw\_data(1, field=field, coef\_data\_m=coef\_data\_m, coef\_data\_t=coef\_data\_t, app\_error="no")  
  
# sample\_training\_cell <- sample\_dt$reg\_raw\_data  
  
# saveRDS(sample\_training\_cell, "./Shared/Data/for\_writing/sample\_reg\_raw\_data.rds")  
  
  
field\_cell\_dt <- readRDS("./Shared/Data/for\_writing/sample\_reg\_raw\_data.rds")%>%  
 .[,plot\_id := ceiling(subplot\_id/4)]%>%  
 .[padding==1,]  
  
field\_cell\_sf <- left\_join(field, field\_cell\_dt, by="unique\_cell\_id")%>%  
 na.omit()  
  
# ggplot(field\_cell\_sf)+  
# geom\_sf(aes(fill=factor(padding)), size=0)  
  
##== area of the field ==##  
# st\_area(st\_union(field\_cell\_sf%>%filter(padding==1)))   
  
# nrow(field\_cell\_sf)  
  
  
  
####==== (2) plot-level dataset without padding area ====####  
  
field\_plot\_sf <- field\_cell\_sf%>%  
 group\_by(plot\_id, strip\_id, rate)%>%  
 summarise()  
  
# nrow(field\_plot\_sf)  
# ggplot(field\_plot\_sf)+geom\_sf()  
  
  
  
####==== (3) subplot-level field without padding area ====####  
  
# field\_subplot\_dt <- field\_cell\_dt[, .(  
# sim = mean(sim),  
# yield = mean(yield),  
# opt\_N = mean(opt\_N),  
# rate = mean(rate),  
# aa\_n = mean(aa\_n),   
# alpha = mean(alpha),  
# beta = mean(beta),  
# ymax = mean(ymax),  
# alpha1 = mean(alpha1),  
# alpha2 = mean(alpha2),  
# beta1 = mean(beta1),  
# beta2 = mean(beta2),  
# ymax1 = mean(ymax1),  
# ymax2 = mean(ymax2),  
# theta\_1 = mean(theta\_1),  
# theta\_2 = mean(theta\_2)  
# ), by = .(subplot\_id, strip\_id)]%>%  
# .[,unique\_subplot\_id:=paste0(strip\_id,"\_",subplot\_id)]%>%  
# .[, `:=`(  
# strip\_id = NULL,  
# subplot\_id = NULL  
# )]  
  
# saveRDS(field\_subplot\_dt, here("Shared", "Data", "for\_writing", "sample\_field\_subplot\_dt.rds"))  
  
field\_subplot\_dt <- readRDS("Shared/Data/for\_writing/sample\_field\_subplot\_dt.rds")  
  
field\_subplot\_sf <- field\_cell\_sf%>%  
 group\_by(subplot\_id, strip\_id)%>%  
 summarise()%>%  
 mutate(unique\_subplot\_id=paste0(strip\_id,"\_",subplot\_id))%>%  
 dplyr::select(unique\_subplot\_id)%>%  
 left\_join(., field\_subplot\_dt, by="unique\_subplot\_id")  
  
# nrow(field\_subplot\_sf)  
# ggplot(field\_subplot\_sf)+geom\_sf()

# Sample CT figure

# library(causalTree)  
# library(here)  
# library(rpart)  
# library(rattle)  
  
# ===================================  
# Get tree   
# ===================================  
  
# data\_sf <- st\_read(here("Shared/Data/snider\_high\_res.gpkg")) %>%  
# na.omit() %>%  
# dplyr::rename(tgtn=NPlan,tgts=SRPlan)  
  
# data\_dt <- data.table(data\_sf) %>%  
# setnames(names(.),tolower(names(.)))  
  
# tgts\_ls <- data\_dt[,tgts] %>% unique() %>% sort  
  
# ctree\_data <- copy(data\_dt) %>%  
# .[tgts %in% tgts\_ls[1:2],] %>%  
# .[,treat:=ifelse(tgts==tgts\_ls[2],1,0)] %>%  
# #--- Corn Yield: 1 bu/acre -> 62.77 kg/ha---#  
# .[,yield:= yield\*62.77]  
  
  
# tree <- causalTree(  
# yield ~ slope+ecs,  
# data = ctree\_data,  
# treatment = ctree\_data$treat,  
# split.Rule = "CT",  
# cv.option = "CT",  
# split.Honest = T,  
# cv.Honest = T,  
# split.Bucket = F,  
# xval = 5,  
# cp = 0,  
# minsize = 20,  
# propensity = 0.5  
# )  
  
# opcp <- tree$cptable[,1][which.min(tree$cptable[,4])]  
# opfit <- prune(tree, opcp)  
  
  
# pdf(here("GitControlled/Writing/JournalSubmission/ctree\_ex.pdf"))  
# fancyRpartPlot(opfit,sub='')  
# dev.off()

variogram\_tb <- data.frame(  
 Parameters = c("ymax\_ij", "alpha\_ij", "beta\_ij", "varepsilon\_ij"),  
 Range = c(400, 400, 400, 400),  
 Mean = c(12000, -0.5, format(c(0,0))),  
 Nugget = format(c(0, 0, 0, 0)),  
 Sill =c(2000000, 0.02, format(1, nsmall = 1), 0.015)  
 ) %>%  
 flextable() %>%  
 compose(i = 1, j = 1, part = "body", value = as\_paragraph(as\_i("ymax"), as\_sup(as\_i("i,j")))) %>%  
 compose(i = 2, j = 1, part = "body", value = as\_paragraph(as\_i("\U03B1"), as\_sup(as\_i("i,j")))) %>%  
 compose(i = 3, j = 1, part = "body", value = as\_paragraph(as\_i("\U03B2"), as\_sup(as\_i("i,j")))) %>%  
 compose(i = 4, j = 1, part = "body", value = as\_paragraph(as\_i("\U03B5"), as\_sup(as\_i("i,j")))) %>%  
 compose(i = 1, j = 3, part = "body", value = as\_paragraph("1.2", "\U2A2F", "10", as\_sup("4"))) %>%  
 compose(i = 1, j = 5, part = "body", value = as\_paragraph("2.0", "\U2A2F", "10", as\_sup("6"))) %>%  
 compose(i = 1, j = 3, part = "body", value = as\_paragraph("1.2", "\U2A2F", "10", as\_sup("4"))) %>%  
 compose(i = 2, j = 5, part = "body", value = as\_paragraph("2.0", "\U2A2F", "10", as\_sup("-2"))) %>%  
 compose(i = 4, j = 5, part = "body", value = as\_paragraph("1.5", "\U2A2F", "10", as\_sup("-2"))) %>%  
 align(align = "center", part = "all")%>%  
 align(j=1, align = "left", part = "all")%>%  
 # --- change headers labels--- #  
 set\_header\_labels(values = list(  
 Parameters = "Parameters",  
 Range = "Range (m)",  
 Mean = "Mean",  
 Nugget = "Nugget",  
 Sill = "Sill"  
 ))%>%  
 #- change the borders just for consistency with other tables -#  
 hline\_bottom(part="all") %>%  
 hline\_top(part="header") %>%  
 footnote(  
 value = as\_paragraph("NOTE: About ymax, the units of of Mean, Nugget, and Sill are kg."),  
 ref\_symbols = NA  
 ) %>%  
 autofit()

# sample MB curve

# gen\_yield\_MB <- function(ymax, alpha, beta, N) {  
# yield <- ymax \* (1 - exp(alpha + beta \* N))  
# return(yield)  
# }  
  
N\_rates <- field\_cell\_dt[,rate]%>%unique()%>%sort()  
  
N\_seq <- 0:300  
  
 # alpha beta ymax  
# 1: -0.6605052 -0.01572583 10886.94  
  
dt\_sample\_MB\_curve <- field\_cell\_dt[1,.(alpha,beta,ymax)]%>%  
 .[rep(1:nrow(.), each = length(N\_seq)), ]%>%  
 .[,N:= N\_seq]%>%  
 .[,Yield := gen\_yield\_MB(ymax, alpha, beta, N)]  
  
sample\_MB\_curve <- ggplot()+  
 geom\_point(data=dt\_sample\_MB\_curve,   
 aes(x=N, y=Yield))+  
 theme\_few()+  
 xlab("N(kg/ha)") + ylab("Yield(kg/ha)")

# Field Maps

## Division of field: plots, subplots and cells (Visualization)

##== plot-level field without padding area ==##  
# vis\_field\_plot <- ggplot(field\_plot\_sf) +  
# geom\_sf(fill=NA) +  
# theme\_void() +  
# labs(caption="Bottom Title") +   
# theme(plot.title = element\_text(hjust = 0.5))   
  
  
  
####==== Example map of plot and subplots and cells ====####  
##== Preparation ==##  
# ex\_plot <- field\_plot\_sf[224,]  
# ex\_subplots <- field\_subplot\_sf[ex\_plot, , op = st\_within]  
  
# ex\_one\_subplot <- ex\_subplots[1,]  
# ex\_cells <- field\_cell\_sf[ex\_one\_subplot, ,op=st\_within]  
  
  
# #-- (1) plot-level field map (base: vis\_field\_withoutPadding) --#  
# library(latex2exp)  
  
# plot <-   
# ggplot()+  
# geom\_sf(data=field\_plot\_sf)+  
# geom\_sf(data=ex\_plot, fill="green", size = 1)+  
# # coord\_sf(expand = F) +  
# theme\_void() +  
# ggtitle(TeX("$12 \\times 32$ plots")) +  
# theme(plot.title = element\_text(hjust = 0.5))  
  
  
# grob\_plot <- ggplotGrob(plot)  
  
# # #-- (2) create a map of subplots in a plot --##  
# subplots\_inPlot <- ggplot()+  
# geom\_sf(data=ex\_plot, fill="green")+  
# geom\_sf(data = ex\_subplots, fill = NA, size = 1)+  
# theme\_void() +  
# ggtitle(TeX("$4 \\times 1$ subplots")) +  
# theme(plot.title = element\_text(hjust = 0.5))  
  
# grob\_subplots\_inPlot <- ggplotGrob(subplots\_inPlot)  
  
# # #-- (3) create a map of cells in a subplot --#  
# cells\_inSubplot <-   
# ggplot()+  
# geom\_sf(data=ex\_one\_subplot, fill="green", size = 1)+  
# geom\_sf(data=ex\_cells, fill=NA)+  
# theme\_void() +  
# ggtitle(TeX("$6 \\times 6$ cells")) +  
# theme(  
# plot.title = element\_text(hjust = 0.5, margin=margin(0,0,10,0)))  
  
# grob\_cells\_inSubplot <- ggplotGrob(cells\_inSubplot)  
  
# # #-- (4)put them on the same map --#  
# g\_inset <- ggplot() +  
# coord\_equal(xlim = c(0, 1), ylim = c(0, 1), expand = FALSE)  
  
# map\_all\_together <-   
# #/\*----------------------------------\*/  
# #' ## base maps   
# #/\*----------------------------------\*/  
# g\_inset +  
# # --- plot level --- #  
# annotation\_custom(grob\_plot,   
# xmin = 0, xmax = 0.4, ymin = 0, ymax = 1) +  
# # --- subplot level --- #   
# annotation\_custom(grob\_subplots\_inPlot,   
# xmin = 0.57, xmax = 0.67, ymin = 0.3, ymax = 0.8) + #hight is 5  
# # --- cell level --- #   
# annotation\_custom(grob\_cells\_inSubplot,   
# xmin = 0.8, xmax = 1, ymin = 0.5, ymax = 0.7) +  
# # xmin = 0.75, xmax = 1, ymin = 0.45, ymax = 0.7) +  
# #/\*----------------------------------\*/  
# #' ## add line segments  
# #/\*----------------------------------\*/  
# # --- plot-subplot (above) --- #  
# # geom\_segment(aes(x = 0.386, xend = 0.55, y = 0.547, yend = 0.73),   
# geom\_segment(aes(x = 0.387, xend = 0.57, y = 0.534, yend = 0.717),   
# lineend = "round", color = "grey",  
# size = 1.5  
# ) +  
# # --- plot-subplot (below) --- #  
# # geom\_segment(aes(x = 0.386, xend = 0.55, y = 0.5, yend = 0.37),   
# geom\_segment(aes(x = 0.387, xend = 0.57, y = 0.485, yend = 0.352),   
# # data = arrowA,   
# lineend = "round", color = "grey",   
# size = 1.5  
# )+  
# # --- subplot-cell (above) --- #  
# # geom\_segment(aes(x = 0.65, xend = 0.805, y = 0.64, yend = 0.692),   
# geom\_segment(aes(x = 0.671, xend = 0.82, y = 0.626, yend = 0.661),   
# # data = arrowA,   
# lineend = "round", color = "grey",  
# size = 1.5  
# ) +  
# # --- subplot-cell (below) --- #  
# geom\_segment(aes(x = 0.671, xend = 0.82, y = 0.532, yend = 0.506),   
# # data = arrowA,   
# lineend = "round", color = "grey",  
# size = 1.5  
# ) +   
# theme\_void()  
   
# # theme(plot.margin=unit(c(0,0,0,0),"mm"))  
  
# file <- here("GitControlled/Writing/Re\_JournalSubmission/field\_devision\_v2.pdf")  
# ggsave(file)  
# knitr::plot\_crop(file)  
# dev.off()

# Distribution Map of Field Characteristics

## Experimental N design

field\_Ndesign <- ggplot() +  
 # geom\_sf(data=field\_plot\_sf, color='blue', size=0.8, fill=NA) +  
 geom\_sf(data=field\_plot\_sf,  
 aes(fill = factor(rate)), size = 0,  
 inherit.aes = FALSE,  
 # alpha=0.9  
 ) +  
 scale\_fill\_viridis\_d() +  
 labs(fill = "Nitrogen rate\n (kg/ha)") +  
 theme\_void()  
 # theme(legend.position = "bottom",  
 # legend.title = element\_text(size = 0.9))

## Yield map

# yield\_dist <- ggplot(field\_cell\_dt)+  
# geom\_density(aes(x=yield))  
  
##== cell level ==##  
vis\_yield\_cell <- ggplot(field\_cell\_sf) +  
 geom\_sf(aes(fill = yield), size = 0) +  
 scale\_fill\_viridis\_c()+  
 labs(fill = "Yield Level\n (kg/ha)")+  
 theme\_void()  
  
  
##== subplot yield level ==##  
vis\_yield\_subplot <- ggplot(field\_subplot\_sf) +  
 geom\_sf(aes(fill = yield), size = 0) +  
 scale\_fill\_viridis\_c()+  
 labs(fill = "Yield Level\n (kg/ha)")+  
 theme\_void()

## opt\_N map

opt\_N\_dist <- ggplot(field\_cell\_dt)+  
 geom\_density(aes(x=opt\_N))  
  
##== cell level ==##  
vis\_optN\_cell <- ggplot(field\_cell\_sf) +  
 geom\_sf(aes(fill = opt\_N), size = 0) +  
 scale\_fill\_viridis\_c()+  
 labs(fill = "EONR (kg/ha)")+  
 theme\_void()  
  
##== subplot yield level ==##  
vis\_optN\_subplot <- ggplot(field\_subplot\_sf) +  
 geom\_sf(aes(fill = opt\_N), size = 0) +  
 scale\_fill\_viridis\_c()+  
 labs(fill = "Yield Level\n (kg/ha)")+  
 theme\_void()

## Field Characteristics Map

####==== ymax map ====####  
ymax\_dist <- ggplot(field\_cell\_dt)+  
 geom\_density(aes(x=ymax))  
  
field\_ymax <- ggplot(field\_cell\_sf) +  
 geom\_sf(aes(fill = ymax), size = 0) +  
 scale\_fill\_viridis\_c()+  
 ggtitle('(1) ymax')+  
 labs(fill = "kg/ha")+  
 theme\_void()  
 # theme(legend.title = element\_blank())  
  
####==== alpha map ====####  
alpha\_dist <- ggplot(field\_cell\_dt)+  
 geom\_density(aes(x=alpha))  
  
field\_alpha <- ggplot(field\_cell\_sf) +  
 geom\_sf(aes(fill = alpha), size = 0) +  
 scale\_fill\_viridis\_c() +  
 ggtitle(expression("(2) " ~alpha)) +  
 theme\_void() +  
 theme(legend.title = element\_blank())  
  
####==== beta map ====####  
beta\_dist <- ggplot(field\_cell\_dt)+  
 geom\_density(aes(x=beta))  
  
field\_beta <- ggplot(field\_cell\_sf) +  
 geom\_sf(aes(fill = beta), size = 0) +  
 scale\_fill\_viridis\_c()+  
 ggtitle(expression("(3) " ~beta)) +  
 # labs(fill = expression(~beta))+  
 theme\_void() +  
 theme(legend.title = element\_blank())  
  
  
####==== m\_error map ====#### (this should be m\_error\*det\_yield, not just m\_error)  
# m\_error\_sf <- left\_join(field, coef\_data\_m, by="unique\_cell\_id")  
  
field\_m\_error <- ggplot(field\_cell\_sf) +  
 geom\_sf(aes(fill = yield\_error), size = 0) +  
 scale\_fill\_viridis\_c()+  
 ggtitle(expression("(d) " ~epsilon)) +  
 labs(fill = "kg/ha")+  
 theme\_void()  
 # theme(legend.title = element\_blank())  
  
  
# grid.arrange(field\_ymax, field\_alpha, field\_beta, field\_m\_error, ncol=2, nrow=2)

# Results

## Source Results and Preparation

# ===================================  
# Forest results   
# ===================================  
  
res\_forest\_all\_honest <- readRDS("./Shared/Results/Summary\_SimRes\_Honest\_full\_tune.rds")  
  
# ===================================  
# CNN results   
# ===================================  
  
res\_CNN\_all <- readRDS("./Shared/Results/SimRes\_CNN\_RMSE.rds")  
  
# /\*-------------------------------------------------------\*/  
#' ## Organize the data  
# /\*-------------------------------------------------------\*/   
#--- for reporting the results of yield prediction and EONR estimation ---#  
report\_res\_allML <- rbind(res\_forest\_all\_honest, res\_CNN\_all)%>%  
 .[, Method := factor(Method, levels = c("RF", "BRF", "CNN", "CF\_stepwise", "CF\_base"))] %>%  
 .[, Model := factor(Model, levels = c("aby", "abytt", "aabbyy", "aabbyytt"))]  
  
report\_res\_subsetML <- report\_res\_allML %>%  
 .[Method != "CF\_stepwise", ]

# Yield prediction results

## Report R^2 of of Predicted Yield (RF, BRF, CNN)

rmse\_y\_all <- copy(report\_res\_allML)%>%  
 .[Method %in% c("RF", "BRF", "CNN")]  
  
#==== Distribution of RMSE of predicted yields ====#  
plot\_dis\_y <- copy(rmse\_y\_all)%>%  
 .[, Method:=factor(Method, levels = c("RF", "BRF", "CNN"))]%>%  
 ggplot()+  
 geom\_density(aes(x=rmse\_y, fill=Method), alpha=0.6)+  
 facet\_wrap(~Model, ncol = 1)+  
 labs(x = "RMSE")+  
 # labs(x = "Mean R-squared")+  
 theme\_few()+  
 theme(  
 strip.text.x = element\_text(size=12,face="bold"),  
 legend.title = element\_text(size=12,face="bold"),  
 legend.text = element\_text(size=12, face="bold"),  
 legend.position = "bottom")  
  
#==== Summary Table ====#  
table\_y\_prep <- copy(rmse\_y\_all) %>%  
 .[,.(rmse\_y = mean(rmse\_y)), by=.(Method, Model)] %>%  
 .[,rmse\_y := format(round(rmse\_y,1), nsmall=1)] %>%  
 dcast(Model~Method, value.var = "rmse\_y")  
  
report\_table\_y <- copy(table\_y\_prep)%>%  
 .[, CF\_base := "-"]%>%  
 mutate(  
 across(  
 everything(),  
 as.character  
 )  
 )%>%  
 flextable(.) %>%  
 set\_header\_labels(values = list(  
 Model = "Model",  
 RF = "RF",  
 BRF = "BRF",  
 CNN = "CNN",  
 CF\_base = "CF-base"  
 ))%>%  
 align(align = "center", part = "all")%>%  
 align(j=1, align = "left", part = "all")%>%  
 #- change the borders just for consistency with other figures -#  
 hline\_bottom(part="all") %>%  
 hline\_top(part="header") %>%  
 autofit()

# Figure (RMSE of EONR vs RMSE of yield for RF abd BRF)

# rmse\_y\_all\_v2 <- rmse\_y\_all %>%  
# .[,`:=`(  
# rmse\_y\_scale = scales::rescale(rmse\_y),  
# rmse\_optN\_scale = scales::rescale(rmse\_optN)  
# ), by= .(Model, Method)]  
  
fig\_y\_optN <- rmse\_y\_all[Method %in% c("RF", "BRF", "CNN")] %>%  
 ggplot(aes(x=rmse\_y, y=rmse\_optN))+  
 geom\_point(size=0.5)+  
 facet\_grid(Model ~ Method) +  
 # geom\_smooth(method='lm', se=FALSE, aes(colour="Regression line"), size=1) +  
 # scale\_colour\_manual(values= "red") +  
 geom\_smooth(method='lm', se=FALSE, color = "red", size=1) +  
 stat\_regline\_equation(  
 # label.x.npc = "right",  
 # label.y.npc = "top",   
 label.x = 2150, label.y = 85,  
 aes(label = ..rr.label..)) +  
 # geom\_abline(slope=1, intercept=0, color="red")+  
 guides(  
 fill = guide\_legend(keywidth = 1, keyheight = 1),  
 linetype = guide\_legend(keywidth = 3, keyheight = 1),  
 colour= guide\_legend(keywidth = 3, keyheight = 1))+  
 labs(y = " RMSE of EONR Estimation (kg/ha)")+  
 labs(x = " RMSE of Yield Prediction (kg/ha)")+  
 theme\_few() +  
 theme(  
 strip.text.x = element\_text(size=12,face="bold"),  
 strip.text.y = element\_text(size=12,face="bold"),  
 legend.title = element\_blank(),  
 legend.text = element\_text(size=12, face="bold"),  
 legend.position = "bottom")  
  
# print(fig\_y\_optN, preview = "docx")  
# dfabline <- data.frame(  
# x = 0:3500, y=0:3500)  
  
# figure\_te <- readRDS(here("Shared/Data/for\_writing/dt\_TEcomparison.rds"))%>%  
# .[, Method:= case\_when(  
# Method=="RF" ~ "RF",  
# Method=="BRF" ~ "BRF",  
# Method=="CF\_base" ~ "CF-base")] %>%  
# .[, Method:= factor(Method, levels = c("RF", "BRF", "CF-base"))]%>%  
# ggplot() +  
# geom\_point(aes(x=true\_tau\_base, y=tau\_base), size=0.5)+  
# # geom\_abline(aes(intercept = 0, slope = 1, color = "1-to-1 line"), show.legend=TRUE) +  
# geom\_line(data=dfabline, aes(x=x, y=y, color="1-to-1 line")) +  
# scale\_colour\_manual(labels = "1-to-1 line", values="red") +  
# facet\_grid(Treatment ~ Method) +  
# guides(  
# fill = guide\_legend(keywidth = 1, keyheight = 1),  
# linetype = guide\_legend(keywidth = 3, keyheight = 1),  
# colour= guide\_legend(keywidth = 3, keyheight = 1),  
# direction = "horizontal")+  
# labs(y = "Estimated Treatment Effect")+  
# labs(x = "True Treatment Effect")+  
# theme\_few() +  
# theme(  
# strip.text.x = element\_text(size=12,face="bold"),  
# strip.text.y = element\_text(size=12,face="bold"),  
# legend.title = element\_blank(),  
# legend.text = element\_text(size=12, face="bold"),  
# legend.position = "bottom")

## Yield Prediction vs EONR estimation : Count Table in terms of profit loss

prepare\_count\_tab <- rmse\_y\_all %>%  
 dcast(sim + Model ~ Method, value.var = c("Mean", "rmse\_y"))%>%  
 .[,which\_ML\_y := case\_when(  
 rmse\_y\_BRF < rmse\_y\_RF & rmse\_y\_BRF < rmse\_y\_CNN ~ "BRF",  
 rmse\_y\_RF < rmse\_y\_BRF & rmse\_y\_RF < rmse\_y\_CNN ~ "RF",  
 rmse\_y\_CNN < rmse\_y\_BRF & rmse\_y\_CNN < rmse\_y\_RF ~ "CNN"  
 )]%>%  
 .[,which\_ML\_piLoss := case\_when(  
 Mean\_BRF < Mean\_RF & Mean\_BRF < Mean\_CNN ~ "BRF",   
 Mean\_RF < Mean\_BRF & Mean\_RF < Mean\_CNN ~ "RF",   
 Mean\_CNN < Mean\_BRF & Mean\_CNN < Mean\_RF ~ "CNN"  
 )] %>%  
 .[,index\_cnst\_y\_piLoss := ifelse(which\_ML\_piLoss==which\_ML\_y, 1, 0)]%>%  
 .[,ML\_cnst\_y\_piLoss:= ifelse(index\_cnst\_y\_piLoss==1, which\_ML\_piLoss, NA)]  
  
 # # RF vs BRF  
 # .[,rmse\_y\_ratio\_RF\_BRF := rmse\_y\_RF/rmse\_y\_BRF]%>%  
 # .[,rmse\_optN\_ratio\_RF\_BRF := rmse\_optN\_RF/rmse\_optN\_BRF]%>%  
 # # CNN vs RF  
 # .[,rmse\_y\_ratio\_CNN\_RF := rmse\_y\_CNN/rmse\_y\_RF]%>%  
 # .[,rmse\_optN\_ratio\_CNN\_RF := rmse\_optN\_CNN/rmse\_optN\_RF]%>%  
 # # CNN vs BRF   
 # .[,rmse\_y\_ratio\_CNN\_BRF := rmse\_y\_CNN/rmse\_y\_BRF]%>%  
 # .[,rmse\_optN\_ratio\_CNN\_BRF := rmse\_optN\_CNN/rmse\_optN\_BRF]  
  
# nrow(.SD[ML\_cnst\_y\_piLoss=="BRF"])  
# prepare\_count\_tab[ML\_cnst\_y\_piLoss=="RF",] %>% nrow()  
# prepare\_count\_tab[ML\_cnst\_y\_piLoss=="BRF",] %>% nrow()  
# prepare\_count\_tab[ML\_cnst\_y\_piLoss=="CNN",] %>% nrow()  
  
# Summary Table   
summary\_res\_CNN\_RF\_BRF <- prepare\_count\_tab%>%  
 .[,.(  
 count\_BRF = nrow(.SD[ML\_cnst\_y\_piLoss=="BRF",]),  
 count\_RF = nrow(.SD[ML\_cnst\_y\_piLoss=="RF",]),  
 count\_CNN = nrow(.SD[ML\_cnst\_y\_piLoss=="CNN",]),  
 count\_y\_BRF = nrow(.SD[which\_ML\_y=="BRF",]),  
 count\_y\_RF = nrow(.SD[which\_ML\_y=="RF",]),  
 count\_y\_CNN = nrow(.SD[which\_ML\_y=="CNN",]),  
 Total= sum(index\_cnst\_y\_piLoss)  
 ), by=Model] %>%  
 .[,`:=`(  
 blank1 = NA, blank2 = NA, blank3 = NA, blank4 = NA  
 )] %>%  
 .[,.(Model, blank1, count\_y\_RF, count\_RF, blank2, count\_y\_BRF, count\_BRF, blank3, count\_y\_CNN, count\_CNN ,blank4, Total)]  
   
# summary\_res\_CNN\_RF\_BRF[Model=="aby", count\_y\_RF]  
  
#- table creation -#  
report\_summary\_res\_CNN\_RF\_BRF <- copy(summary\_res\_CNN\_RF\_BRF)%>%  
 flextable(.) %>%  
 border\_remove() %>%  
 delete\_part(part = "header") %>%  
 add\_header(  
 Model="Model",   
 blank1 = "", count\_y\_RF = "RF", count\_RF = "RF",  
 blank2 = "", count\_y\_BRF = "BRF", count\_BRF = "BRF",  
 blank3 = "", count\_y\_CNN = "CNN", count\_CNN = "CNN",  
 blank4 = "", Total = "Total",  
 top = TRUE) %>%  
 merge\_h(part = "header") %>%  
 hline\_bottom(j=c(3:4, 6:7, 9:10), part = "header") %>%  
 add\_header(  
 Model="",   
 blank1 = "", count\_y\_RF = "#Y", count\_RF = "#YP",  
 blank2 = "", count\_y\_BRF = "#Y", count\_BRF = "#YP",  
 blank3 = "", count\_y\_CNN = "#Y", count\_CNN = "#YP",  
 blank4 = "", Total = "",  
 top = FALSE) %>%  
 hline\_bottom(part="all") %>%  
 hline\_top(part="header") %>%  
 align(align = "center", part = "all")%>%  
 align(j=1, align = "left", part = "all")%>%  
 footnote(  
 value = as\_paragraph("NOTE: #Y indicates the number of simulation rounds in which the model provided the lowest RMSE for yield prediction. #YP indicates the number of simulation rounds in which the model provided the lower RMSE of yield prediction and the highest profit at the same time."),  
 ref\_symbols = NA  
 ) %>%  
 # set\_table\_properties(width = .7, layout = "autofit") %>%  
 fontsize(i = NULL, j = NULL, size = 9, part = "footer") %>%  
 autofit() %>%  
 width(j = c(3,4,6,7,9,10,12), width=0.6)%>%  
 width(j = c(2,5,8,11), width=0.1)  
  
 # set\_table\_properties(width = .6, layout = "autofit")  
  
# print(report\_summary\_res\_CNN\_RF\_BRF, preview = "docx")  
# in terms of RMSE of yield prediction and ", "\U1D70B\U0302", as\_sub("loss"), ", respectively.

# EONR Estimation

##== Distribution ==##  
plot\_dis\_optN <- copy(report\_res\_subsetML)%>%  
 .[Method %in% c("RF", "BRF", "CF\_base"), ]%>%  
 .[, Method := case\_when(  
 # Method == "CF\_stepwise" ~ "CF-stepwise",  
 Method == "CF\_base" ~ "CF-base",  
 Method == "RF" ~ "RF",  
 Method == "BRF" ~ "BRF"  
 )]%>%  
 .[, Model := case\_when(  
 Model == "aby" ~ "Scenario: aby",  
 Model == "abytt" ~ "Scenario: abytt",  
 Model == "aabbyy" ~ "Scenario: aabbyy",  
 Model == "aabbyytt" ~ "Scenario: aabbyytt"  
 )]%>%  
 # .[, Method:=factor(Method, levels = c("RF", "BRF", "CF-stepwise", "CF-base"))]%>%  
 .[, Method:=factor(Method, levels = c("RF", "BRF", "CF-base"))]%>%  
 .[, Model:=factor(Model,   
 levels = c("Scenario: aby", "Scenario: abytt", "Scenario: aabbyy", "Scenario: aabbyytt"))] %>%  
 ggplot()+  
 geom\_density(aes(x=rmse\_optN, fill=Method), alpha=0.7)+  
 # scale\_fill\_viridis\_d()+  
 facet\_wrap(~Model, ncol = 1)+  
 # labs(x = expression(R^2))+  
 labs(x = "RMSE (kg/ha)")+  
 theme\_few()+  
 theme(  
 strip.text.x = element\_text(size=12,face="bold"),  
 legend.title = element\_text(size=12,face="bold"),  
 legend.text = element\_text(size=12, face="bold"),  
 legend.position = "bottom")  
  
  
##== Summary Table ==##  
  
#- preparation -#  
table\_optN\_prep <- copy(report\_res\_subsetML)%>%  
 .[, .(  
 rmse\_optN = mean(rmse\_optN),  
 pi\_loss = mean(Mean)  
 ), by=.(Method, Model)]%>%  
 .[,`:=`(  
 rmse\_optN = format(round(rmse\_optN,1), nsmall=1),  
 pi\_loss = format(round(pi\_loss,2), nsmall=2)  
 )] %>%  
 dcast(Model~Method, value.var = c("rmse\_optN", "pi\_loss")) %>%  
 .[,`:=`(  
 blank1 = NA, blank2 = NA, blank3 = NA, blank4 = NA, blank5 = NA  
 )] %>%  
 .[,.(Model, blank1, rmse\_optN\_RF, pi\_loss\_RF, blank2, rmse\_optN\_BRF, pi\_loss\_BRF, blank3, rmse\_optN\_CNN, pi\_loss\_CNN ,blank4, rmse\_optN\_CF\_base, pi\_loss\_CF\_base)]  
  
# table\_optN\_prep[Model=="aby", rmse\_optN\_RF]  
  
# avg\_optN <- copy(report\_res\_subsetML)%>%  
# .[, .(  
# Avg\_rmse\_optN = mean(rmse\_optN),  
# Avg\_pi\_loss = mean(Mean)  
# ), by=Method] %>%  
# .[, `:=`(  
# Avg\_rmse\_optN = format(round(Avg\_rmse\_optN, digits=1), nsmall = 1),  
# Avg\_pi\_loss = format(round(Avg\_pi\_loss, digits=2), nsmall=2)  
# )]   
  
#- table creation -#  
report\_table\_optN <- copy(table\_optN\_prep)%>%  
 mutate(  
 across(  
 everything(),  
 as.character  
 )  
 )%>%  
 # add\_row(.after = 4) %>%  
 flextable(.) %>%  
 border\_remove() %>%  
 delete\_part(part = "header") %>%  
 add\_header(  
 Model="Model",   
 blank1 = "", rmse\_optN\_RF = "RF", pi\_loss\_RF = "RF",  
 blank2 = "", rmse\_optN\_BRF = "BRF", pi\_loss\_BRF = "BRF",  
 blank3 = "", rmse\_optN\_CNN = "CNN", pi\_loss\_CNN = "CNN",  
 blank4 = "", rmse\_optN\_CF\_base = "CF-base", pi\_loss\_CF\_base = "CF-base",  
 top = TRUE) %>%  
 merge\_h(part = "header") %>%  
 hline\_bottom(j=c(3:4, 6:7, 9:10, 12:13), part = "header") %>%  
 add\_header(  
 Model="",   
 blank1 = "", rmse\_optN\_RF = "RMSE", pi\_loss\_RF = "pi\_loss",  
 blank2 = "", rmse\_optN\_BRF = "RMSE", pi\_loss\_BRF = "pi\_loss",  
 blank3 = "", rmse\_optN\_CNN = "RMSE", pi\_loss\_CNN = "pi\_loss",  
 blank4 = "", rmse\_optN\_CF\_base = "RMSE", pi\_loss\_CF\_base = "pi\_loss",  
 top = FALSE) %>%  
 compose(i = 2, j = c(4, 7, 10, 13), part = "header", value = as\_paragraph("\U1D70B\U0302", as\_sub(as\_i("def")))) %>%  
 hline\_bottom(part="all") %>%  
 hline\_top(part="header") %>%  
 align(align = "center", part = "all")%>%  
 align(j=1, align = "left", part = "all")%>%  
 # autofit() %>%  
 # width(j = c(2,5,8,11), width=0.3) %>%  
 # fix\_border\_issues()   
 footnote(  
 # i = 1, j = c(4, 7, 10, 13), part = "header",  
 value = as\_paragraph("NOTE: \U1D70B\U0302", as\_sub(as\_i("def")), " indicates profit-deficit ($/ha) relative to the true maximum profit at the subplot level. The maximized profit is the profit under the true yield response functions evaluated at ", as\_i("\U004E\U2071"), as\_sub(as\_i("opt")),"."),  
 ref\_symbols = NA  
 ) %>%  
 fontsize(i = NULL, j = NULL, size = 9, part = "footer") %>%  
 autofit() %>%  
 width(j = c(3,4,6,7,9,10,12,13), width=0.6)%>%  
 width(j = c(2,5,8,11), width=0.1)  
   
 # print(report\_table\_optN, preview = "docx")  
 # autofit()   
 # width(j = c(2,5,8,11), width=0.5)  
 # width(j = c(2,5,8,11), width=0.5)  
  
  
  
  
# print(report\_table\_optN, preview = "docx")  
  
# typology <- data.frame(  
# col\_keys = names(table\_optN\_prep),  
# what = c(" ", " ", "RF", "RF", " ", "BRF", "BRF", " ", "CNN", "CNN", " ", "CF-base", "CF-base"),  
# measure = c(" ", " ", "RMSE", "pi\_loss", " ", "RMSE", "pi\_loss", " ", "RMSE", "pi\_loss", " ", "RMSE", "pi\_loss"),  
# stringsAsFactors = FALSE )  
  
  
# copy(table\_optN\_prep)%>%  
# mutate(  
# across(  
# everything(),  
# as.character  
# )  
# )%>%  
# add\_row(.after = 4) %>%  
# flextable(.) %>%  
# border\_remove() %>%  
# add\_body(  
# Model="Avg.",   
# blank1 = NA, rmse\_optN\_RF = avg\_optN[Method=="RF",Avg\_rmse\_optN], pi\_loss\_RF = avg\_optN[Method=="RF",Avg\_pi\_loss],  
# blank2 = NA, rmse\_optN\_BRF = avg\_optN[Method=="BRF",Avg\_rmse\_optN], pi\_loss\_BRF = avg\_optN[Method=="BRF",Avg\_pi\_loss],  
# blank3 = NA, rmse\_optN\_CNN = avg\_optN[Method=="CNN",Avg\_rmse\_optN], pi\_loss\_CNN = avg\_optN[Method=="CNN",Avg\_pi\_loss],  
# blank4 = NA, rmse\_optN\_CF\_base = avg\_optN[Method=="CF\_base",Avg\_rmse\_optN], pi\_loss\_CF\_base = avg\_optN[Method=="CF\_base",Avg\_pi\_loss],  
# top = FALSE  
# )%>%  
# # delete\_part(part = "header") %>%  
# set\_header\_df(mapping = typology, key = "col\_keys") %>%  
# compose(i = 2, j = c(4, 7, 10, 13), part = "header", value = as\_paragraph("\U1D70B\U0302", as\_sub("loss"))) %>%  
# merge\_h(part = "header") %>%  
# hline\_bottom(part="all") %>%  
# hline\_top(part="header") %>%  
# align(align = "center", part = "all")%>%  
# align(j=1, align = "left", part = "all")%>%  
# autofit() %>%  
# width(j = c(2,5,8,11), width=0.5)

# Profit Loss

## Profit loss summary and distributions

# piLoss\_summary <- report\_res\_subsetML[,select(.SD, Method, Model, Min.:Max., cumulative\_pi\_loss)]%>%  
# .[, lapply(.SD, mean), by=.(Method, Model)] %>%  
# .[, lapply(.SD, function(x) as.numeric(format(mean(x), digits=3))), by=.(Method, Model)] %>%  
# flextable()  
  
  
piLoss\_density <-   
 report\_res\_subsetML[Method!="CNN",] %>%  
 .[,Method := case\_when(  
 Method == "CF\_base" ~ "CF-base",  
 Method == "RF" ~ "RF",  
 Method == "BRF" ~ "BRF"  
 )]%>%  
 .[, Model := case\_when(  
 Model == "aby" ~ "Scenario: aby",  
 Model == "abytt" ~ "Scenario: abytt",  
 Model == "aabbyy" ~ "Scenario: aabbyy",  
 Model == "aabbyytt" ~ "Scenario: aabbyytt"  
 )]%>%  
 .[,Method := factor(Method, levels = c("RF", "BRF", "CF-base"))] %>%  
 .[, Model:=factor(Model,   
 levels = c("Scenario: aby", "Scenario: abytt", "Scenario: aabbyy", "Scenario: aabbyytt"))] %>%  
 ggplot() +   
 geom\_density(aes(x=Mean, fill=Method), alpha=0.5)+  
 # scale\_fill\_viridis(discrete = TRUE, alpha=0.5) +  
 # scale\_fill\_viridis\_d()+  
 # labs(x = expression(hat(pi)["loss"], " $/ha")))+  
 labs(x = TeX("$\\hat{pi}\_{loss}$ (\\$/ha)"))+  
 facet\_wrap(~Model, ncol=1) +  
 theme\_few()+  
 theme(  
 strip.text.x = element\_text(size=12,face="bold"),  
 legend.title = element\_text(size=12,face="bold"),  
 legend.text = element\_text(size=12, face="bold"),  
 legend.position = "bottom")

# Treatment effect comparison (CF-base vs RF vs BRF)

####=== The original code for this figure is in "1\_3\_CompTeEstimation.R" ===####  
  
figure\_te <- readRDS("./Shared/Data/for\_writing/dt\_TEcomparison.rds")%>%  
 .[, Method:= case\_when(  
 Method=="RF" ~ "RF",  
 Method=="BRF" ~ "BRF",  
 Method=="CF\_base" ~ "CF-base")] %>%  
 .[, Method:= factor(Method, levels = c("RF", "BRF", "CF-base"))]%>%  
 ggplot() +  
 geom\_point(aes(x=true\_tau\_base, y=tau\_base), size=0.5)+  
 geom\_abline(aes(intercept = 0, slope = 1), color="red", show.legend=TRUE) +  
 facet\_grid(Treatment ~ Method) +  
 guides(  
 fill = guide\_legend(keywidth = 1, keyheight = 1),  
 linetype = guide\_legend(keywidth = 3, keyheight = 1),  
 colour= guide\_legend(keywidth = 3, keyheight = 1))+  
 labs(y = "Estimated Treatment Effect (kg/ha)")+  
 labs(x = "True Treatment Effect (kg/ha)")+  
 theme\_few() +  
 theme(  
 strip.text.x = element\_text(size=12,face="bold"),  
 strip.text.y = element\_text(size=12,face="bold"),  
 legend.title = element\_blank(),  
 legend.text = element\_text(size=12, face="bold"),  
 legend.position = "bottom")

# Examination of Treatment Effects (CF-base vs CF-stepwise)

# ======================================================================  
# Examination of te esimation (CF-stepwise vs CF-base)  
# ======================================================================  
  
# te\_res <- readRDS(here("Shared/Results/res\_eval\_tre\_effects.rds"))%>%  
# .[,.(method, N\_index, R\_squared)]%>%  
# .[, N\_index := case\_when(  
# N\_index == "1-2" ~ "(N1,N2)",  
# N\_index == "2-3" ~ "(N2,N3)",  
# N\_index == "3-4" ~ "(N3,N4)",  
# N\_index == "4-5" ~ "(N4,N5)",  
# N\_index == "1-3" ~ "(N1,N3)",  
# N\_index == "1-4" ~ "(N1,N4)",  
# N\_index == "1-5" ~ "(N1,N5)"  
# )]%>%  
# .[,sim:=rep(1:1000, each=8)]  
  
##== Treatment Effects Summary Table ==##  
  
# te\_res\_summary <- te\_res[,.(R\_squared=mean(R\_squared)), by=c("method", "N\_index")]%>%  
# .[, R\_squared:= round(R\_squared,3)]%>%  
# flextable(.)%>%  
# set\_header\_labels(  
# values = list(  
# method="Method",  
# N\_index ="N Rates Combinations",  
# R\_squared = "R-squared"  
# ))%>%  
# compose(i = 1, j = 2, part = "header", value = as\_paragraph("(N", as\_sup("con"), ", N", as\_sup("tre"), ")"))%>%  
# compose(i = 1, j = 3, part = "header", value = as\_paragraph("R",as\_sup("2")))%>%  
# align(align = "center", part = "all")%>%  
# align(j=1, align = "left", part = "all")%>%  
# # align(align = "center", part = "header")%>%  
# # align\_text\_col(align = "center", header = TRUE, footer = TRUE)%>%  
# autofit()  
  
##== Graph ==##  
#== tre calculation ==#  
  
# reg\_data\_all <- readRDS(here("Data/for\_Simulations/reg\_data.rds"))  
# test\_agg\_data\_all <- readRDS(here("Data/for\_Simulations/test\_agg\_data.rds"))  
  
# # estimate\_tre() is defined in "functions\_stepwise\_vs\_base.R"  
# sim\_ex <- estimate\_tre(x=32, reg\_data\_all=reg\_data\_all, test\_data\_all=test\_agg\_data\_all)  
  
# sim\_ex$r2  
  
# saveRDS(sim\_ex, here("Shared", "Data","for\_writing","tre\_demo.rds"))  
  
# sim\_output <- readRDS(here("Shared","Data","for\_writing","tre\_demo.rds"))  
  
# # r2 <-sim\_output$r2  
# eval <- sim\_output$output  
  
# cf\_step <- eval[Method=="CF-stepwise"]%>%  
# ggplot()+  
# geom\_density(aes(x=value, fill=factor(variable)),alpha=0.8)+  
# facet\_wrap(~N\_index, ncol = 2)+  
# theme\_few()+  
# theme\_few()+  
# theme(  
# legend.position="bottom",  
# plot.title = element\_text(hjust = 0.5),  
# panel.background = element\_rect(fill = "white"),  
# axis.ticks = element\_blank(),  
# legend.title = element\_blank(),  
# strip.text.x = element\_text(size=12,face="bold"),  
# legend.text = element\_text(size=12, face="bold"),  
# )+  
# labs(  
# x = "Treatment Effects(kg/ha)",  
# colour = "Cylinders",  
# shape = "Transmission"  
# )  
  
  
# cf\_base <- eval[Method=="CF-base"]%>%  
# ggplot()+  
# geom\_density(aes(x=value, fill=factor(variable)),alpha=0.8)+  
# facet\_wrap(~N\_index, ncol = 2)+  
# theme\_few()+  
# theme\_few()+  
# theme(  
# legend.position="bottom",  
# plot.title = element\_text(hjust = 0.5),  
# panel.background = element\_rect(fill = "white"),  
# axis.ticks = element\_blank(),  
# legend.title = element\_blank(),  
# strip.text.x = element\_text(size=12,face="bold"),  
# legend.text = element\_text(size=12, face="bold"),  
# )+  
# labs(  
# x = "Treatment effects(kg/ha)",  
# colour = "Cylinders",  
# shape = "Transmission"  
# )

# Appendix: CF-base vs CF-stepwise

# ==========================================================================  
# RMSE  
# ==========================================================================  
##== Distribution ==##  
plot\_dis\_CF\_optN <- copy(report\_res\_allML)%>%  
 .[Method %in% c("CF\_stepwise", "CF\_base"), ]%>%  
 .[, Method := case\_when(  
 Method == "CF\_stepwise" ~ "CF-stepwise",  
 Method == "CF\_base" ~ "CF-base"  
 )]%>%  
 .[, Model := case\_when(  
 Model == "aby" ~ "Scenario: aby",  
 Model == "abytt" ~ "Scenario: abytt",  
 Model == "aabbyy" ~ "Scenario: aabbyy",  
 Model == "aabbyytt" ~ "Scenario: aabbyytt"  
 )]%>%  
 .[, Method:=factor(Method, levels = c("CF-stepwise", "CF-base"))] %>%  
 .[, Model:=factor(Model,   
 levels = c("Scenario: aby", "Scenario: abytt", "Scenario: aabbyy", "Scenario: aabbyytt"))] %>%  
 ggplot()+  
 geom\_density(aes(x=rmse\_optN, fill=Method), alpha=0.7)+  
 # scale\_fill\_viridis\_d()+  
 facet\_wrap(~Model, ncol = 1)+  
 labs(x = "RMSE")+  
 theme\_few()+  
 theme(  
 strip.text.x = element\_text(size=12,face="bold"),  
 legend.title = element\_text(size=12,face="bold"),  
 legend.text = element\_text(size=12, face="bold"),  
 legend.position = "bottom")  
  
##== barplot ==##  
CF\_optN\_boxplot <- copy(report\_res\_allML)%>%  
 .[Method %in% c("CF\_stepwise", "CF\_base"), ]%>%  
 .[, Method := case\_when(  
 # Method == "CF\_stepwise" ~ "CF-stepwise",  
 Method == "CF\_stepwise" ~ "CF-stepwise",  
 Method == "CF\_base" ~ "CF-base"  
 )]%>%  
 .[, Method:=factor(Method, levels = c("CF-stepwise", "CF-base"))]%>%  
 ggplot() +   
 geom\_boxplot(aes(x=Method, y=rmse\_optN, fill=Method), alpha=0.5)+  
 geom\_jitter(aes(x=Method, y=rmse\_optN, fill=Method), size=0.3, alpha=0.9) +  
 # scale\_fill\_viridis(discrete = TRUE, alpha=0.6) +  
 scale\_fill\_viridis\_d()+  
 labs(y = " RMSE")+  
 facet\_grid(~Model)+  
 theme\_few()+  
 theme(  
 strip.text.x = element\_text(size=12,face="bold"),  
 legend.title = element\_text(size=12,face="bold"),  
 legend.text = element\_text(size=12, face="bold"),  
 legend.position = "bottom")  
  
  
##== Summary Table ==##  
  
#- preparation -#  
table\_CF\_optN\_prep <- copy(report\_res\_allML)%>%  
 .[, .(  
 rmse\_optN = mean(rmse\_optN),  
 pi\_loss = mean(Mean)  
 ), by=.(Method, Model)]%>%  
 .[,`:=`(  
 rmse\_optN = format(round(rmse\_optN,1), nsmall=1),  
 pi\_loss = format(round(pi\_loss,2), nsmall=2)  
 )] %>%  
 dcast(Model~Method, value.var = c("rmse\_optN", "pi\_loss")) %>%  
 .[,`:=`(  
 blank1 = NA, blank2 = NA  
 )] %>%  
 .[,.(Model, blank1, rmse\_optN\_CF\_base, pi\_loss\_CF\_base, blank2, rmse\_optN\_CF\_stepwise, pi\_loss\_CF\_stepwise)]  
  
# avg\_CF\_optN <- copy(report\_res\_allML)%>%  
# .[, .(  
# Avg\_rmse\_optN = mean(rmse\_optN),  
# Avg\_pi\_loss = mean(Mean)  
# ), by=Method] %>%  
# .[, `:=`(  
# Avg\_rmse\_optN = format(round(Avg\_rmse\_optN, digits=1), nsmall = 1),  
# Avg\_pi\_loss = format(round(Avg\_pi\_loss, digits=2), nsmall=2)  
# )]   
  
  
#- table creation -#  
report\_table\_CF\_optN <- copy(table\_CF\_optN\_prep)%>%  
 mutate(  
 across(  
 everything(),  
 as.character  
 )  
 )%>%  
 # add\_row(.after = 4) %>%  
 flextable(.) %>%  
 border\_remove() %>%  
 delete\_part(part = "header") %>%  
 add\_header(  
 Model="Model",   
 blank1 = "", rmse\_optN\_CF\_base = "CF-base", pi\_loss\_CF\_base = "CF-base",  
 blank2 = "", rmse\_optN\_CF\_stepwise = "CF-stepwise", pi\_loss\_CF\_stepwise = "CF-stepwise",  
 top = TRUE) %>%  
 merge\_h(part = "header") %>%  
 hline\_bottom(j=c(3:4, 6:7), part = "header") %>%  
 add\_header(  
 Model="",   
 blank1 = "", rmse\_optN\_CF\_base = "RMSE", pi\_loss\_CF\_base = "pi\_loss",  
 blank2 = "", rmse\_optN\_CF\_stepwise = "RMSE", pi\_loss\_CF\_stepwise = "pi\_loss",  
 top = FALSE) %>%  
 compose(i = 2, j = c(4, 7), part = "header", value = as\_paragraph("\U1D70B\U0302", as\_sub("loss"))) %>%  
 hline\_bottom(part="all") %>%  
 hline\_top(part="header") %>%  
 align(align = "center", part = "all")%>%  
 align(j=1, align = "left", part = "all")%>%  
 # set\_table\_properties(width = .5, layout = "autofit")  
 autofit() %>%  
 width(j = c(2,5), width=0.3) %>%  
 fix\_border\_issues()   
  
  
# ==========================================================================  
# profit-loss  
# ==========================================================================  
  
##== Distribution ==##  
piLoss\_density\_CF <- copy(report\_res\_allML)%>%  
 .[Method %in% c("CF\_stepwise", "CF\_base"), ]%>%  
 .[, Method := case\_when(  
 # Method == "CF\_stepwise" ~ "CF-stepwise",  
 Method == "CF\_stepwise" ~ "CF-stepwise",  
 Method == "CF\_base" ~ "CF-base"  
 )]%>%  
 .[, Model := case\_when(  
 Model == "aby" ~ "Scenario: aby",  
 Model == "abytt" ~ "Scenario: abytt",  
 Model == "aabbyy" ~ "Scenario: aabbyy",  
 Model == "aabbyytt" ~ "Scenario: aabbyytt"  
 )]%>%  
 .[, Method:=factor(Method, levels = c("CF-stepwise", "CF-base"))]%>%  
 .[, Model:=factor(Model,   
 levels = c("Scenario: aby", "Scenario: abytt", "Scenario: aabbyy", "Scenario: aabbyytt"))] %>%  
 ggplot() +   
 geom\_density(aes(x=Mean, fill=Method), alpha=0.5)+  
 # scale\_fill\_viridis(discrete = TRUE, alpha=0.5) +  
 # scale\_fill\_viridis\_d()+  
 labs(x = expression(hat(pi)["loss"]))+  
 facet\_wrap(~Model, ncol=1) +  
 theme\_few()+  
 theme(  
 strip.text.x = element\_text(size=12,face="bold"),  
 legend.title = element\_text(size=12,face="bold"),  
 legend.text = element\_text(size=12, face="bold"),  
 legend.position = "bottom")  
  
  
##== barplot ==##  
CF\_piLoss\_boxplot <- copy(report\_res\_allML)%>%  
 .[Method %in% c("CF\_stepwise", "CF\_base"), ]%>%  
 .[, Method := case\_when(  
 # Method == "CF\_stepwise" ~ "CF-stepwise",  
 Method == "CF\_stepwise" ~ "CF-stepwise",  
 Method == "CF\_base" ~ "CF-base"  
 )]%>%  
 .[, Method:=factor(Method, levels = c("CF-stepwise", "CF-base"))]%>%  
 ggplot() +   
 geom\_boxplot(aes(x=Method, y=Mean, fill=Method), alpha=0.5)+  
 # geom\_jitter(aes(x=Method, y=Mean, fill=Method), size=0.3, alpha=0.9) +  
 # scale\_fill\_viridis(discrete = TRUE, alpha=0.6) +  
 scale\_fill\_viridis\_d()+  
 labs(y = " Mean Profit-loss")+  
 facet\_grid(~Model)+  
 theme\_few()+  
 theme(  
 strip.text.x = element\_text(size=12,face="bold"),  
 legend.title = element\_text(size=12,face="bold"),  
 legend.text = element\_text(size=12, face="bold"),  
 legend.position = "bottom")