Random/Randomer Forest Bootstrap vs Subsample

Random Forest Bootstrap vs Subsampling

Variables swept over: mtrys - p^(1/4, 1/2, 3/4, 1) replace - T or F

RerF Bootstrap vs Subsampling

Variables swept over: mtrys - p^(1/4, 1/2, 3/4, 1, 2) sparsity - 1/p, 2/p, 3/p, 4/p, 5/p replacement - T or F

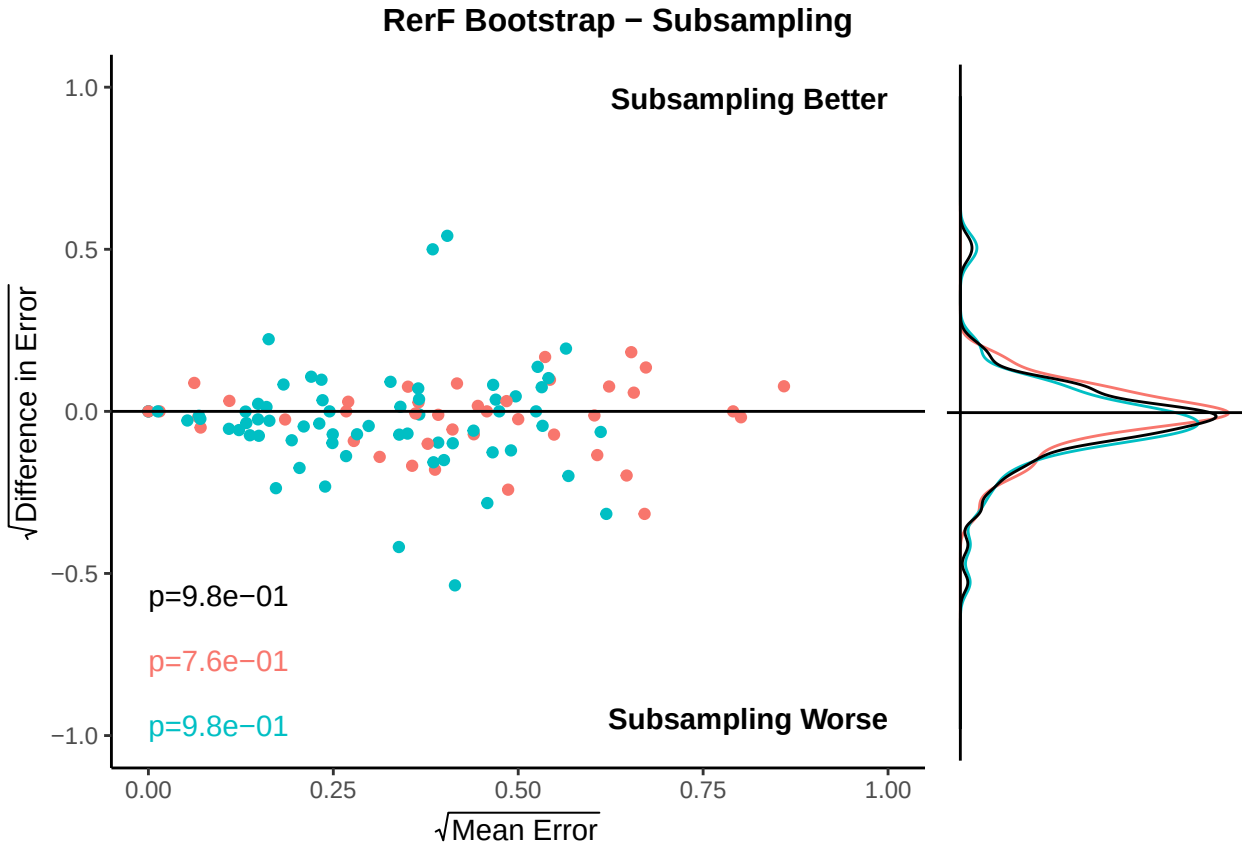
```
library('ggplot2')
library('grid')
library('gridExtra')
library('gtable')

plotResults <- function(df, classifiers, y.min = -1, y.max = 1) {
  categories <- df[['Category']][seq(1, nrow(df), 5)]
  error.cls.1 <- df[[classifiers[1]]]
  error.cls.2 <- df[[classifiers[2]]]

  error.cls.1 <- rowMeans(t(matrix(error.cls.1, nrow = 5)))
  error.cls.2 <- rowMeans(t(matrix(error.cls.2, nrow = 5)))

  # Compute One sided Wilcox Rank Test
  alt <- 'g'

  wilcox.all <- wilcox.test(error.cls.1, error.cls.2, paired = T, alternative = alt, exact = F)
  wilcox.categorical <- wilcox.test(error.cls.1[categories == 'categorical'],
                                    error.cls.2[categories == 'categorical'],
                                    paired = T,
                                    alternative = alt,
                                    exact = F)
  wilcox.numeric <- wilcox.test(error.cls.1[categories == 'numeric'],
                                error.cls.2[categories == 'numeric'],
                                paired = T,
                                alternative = alt,
                                exact = F)

  pvalue.all <- format(round(wilcox.all$p.value, 2), scientific = T)
  pvalue.categorical <- format(round(wilcox.categorical$p.value, 2), scientific = T)
  pvalue.numeric <- format(round(wilcox.numeric$p.value, 2), scientific = T)

  mean.error <- sqrt((error.cls.1 + error.cls.2) / 2)
  difference.error <- sqrt(abs(error.cls.1 - error.cls.2)) * sign(error.cls.1 - error.cls.2)

  df <- data.frame(mean.error, difference.error, categories)
  names(df) <- c("mean", "diff", "category")
  df$category <- factor(df$category)

  # Plot scatter
  fig <- ggplot(df, aes(x = mean, y = diff, color = category)) + geom_point() +
    theme(
```

```
      panel.background = element_blank(), axis.line = element_line(colour = "black")
    ) +
    labs(
      x = expression(sqrt("Mean Error")),
      y = expression(sqrt("Difference in Error"))
    ) +
    geom_hline(yintercept = 0) +
    xlim(0, 1) +
    ylim(-1, 1) +
    annotate("text", label = 'bold("Subsampling Better")', x = 1, y = 1, parse = T, hjust = 'inward', v
    annotate("text", label = 'bold("Subsampling Worse")', x = 1, y = -1, parse = T, hjust = 'inward', v
    # annotate("text", label = paste0("p=", pvalue.all, "\np=", pvalue.categorical, "\np=", pvalue.nume
    #           x = 0, y = -1, vjust = 'inward', hjust = 'inward')
    annotate("text", label = paste0("p=", pvalue.all),
             x = 0, y = -.6, vjust = 'inward', hjust = 'inward', color = "black") +
    annotate("text", label = paste0("p=", pvalue.categorical),
             x = 0, y = -.8, vjust = 'inward', hjust = 'inward', color = "#F8766D") +
    annotate("text", label = paste0("p=", pvalue.numeric),
             x = 0, y = -1, vjust = 'inward', hjust = 'inward', color = "#00BFC4")

  # Plot KDE
  kde <- ggplot(df, aes(x = diff, color = category)) +
    stat_density(geom = 'line', position = 'identity') +
    stat_density(aes(x = diff, color = 'all'),  geom = 'line') +
    theme(panel.background = element_blank(),
          axis.line = element_blank(),
          axis.ticks = element_blank(),
          axis.text = element_blank(),
          axis.title = element_blank(),
          legend.direction = "horizontal",
          legend.position = "bottom") +
    geom_hline(yintercept = 0) +
    geom_vline(xintercept = 0) +
    xlim(-1, 1) +
    coord_flip() +
    scale_color_manual(values=c('#000000','#F8766D','#00BFC4'))

  # print(fig)
  return(list(fig = fig, kde = kde))
}


load('./../2018.07.02/uci_results.RData')
load('./../2018.07.04/df.rf.RData')

res <- plotResults(df, c('RerF','RerF.subsample'))
fig.1 <- res$fig
kde.1 <- res$kde + scale_fill_manual(name = "Dataset Type", values = c('black', '#F8766D','#00BFC4'), la

res <- plotResults(df.rf, c('rf.bag', 'rf.subsample'), -.5, .5)
fig.2 <- res$fig
kde.2 <- res$kde
```

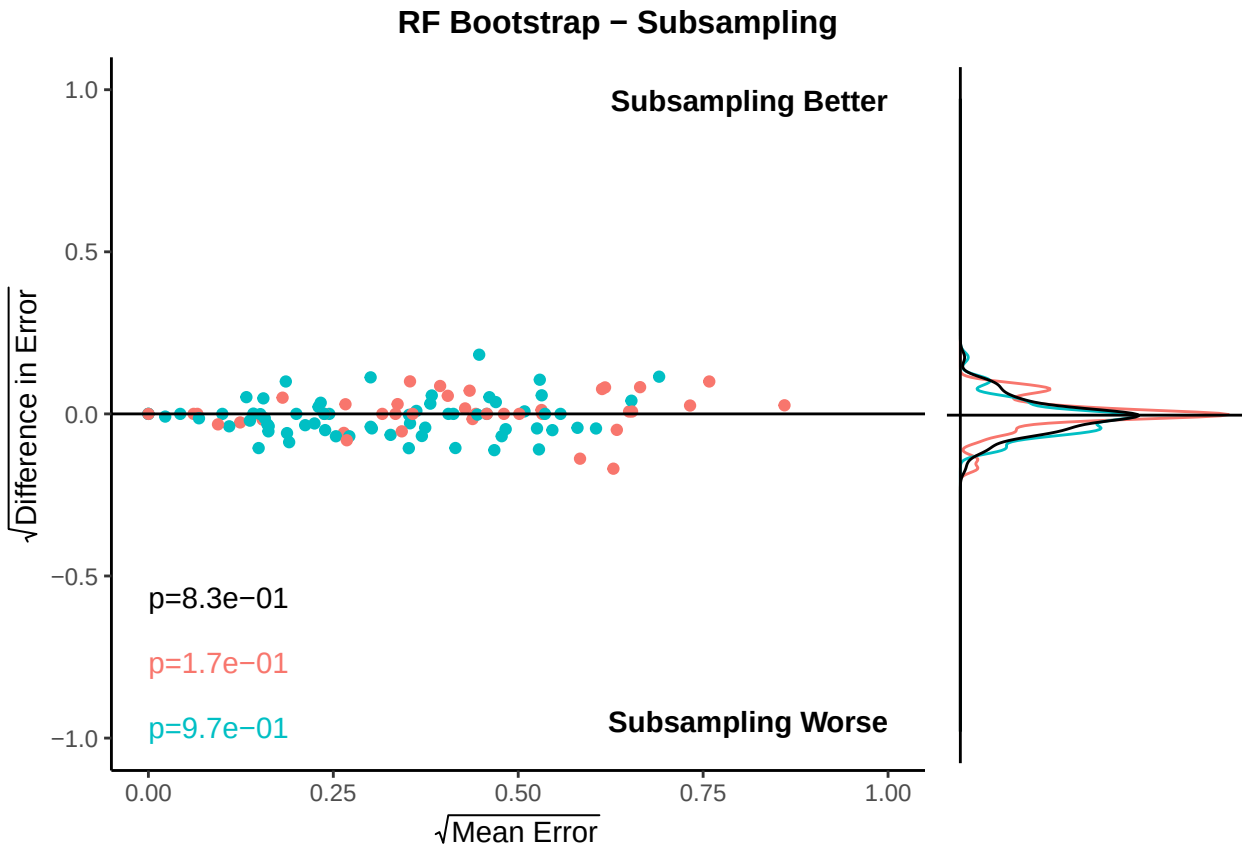
```r
# Get legend for separate plotting
g_legend<-function(a.gplot){
  tmp <- ggplot_gtable(ggplot_build(a.gplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  return(legend)}

leg <- g_legend(kde.1 + scale_fill_manual("Dataset Type", values = c('black', '#F8766D','#00BFC4'), lab
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```

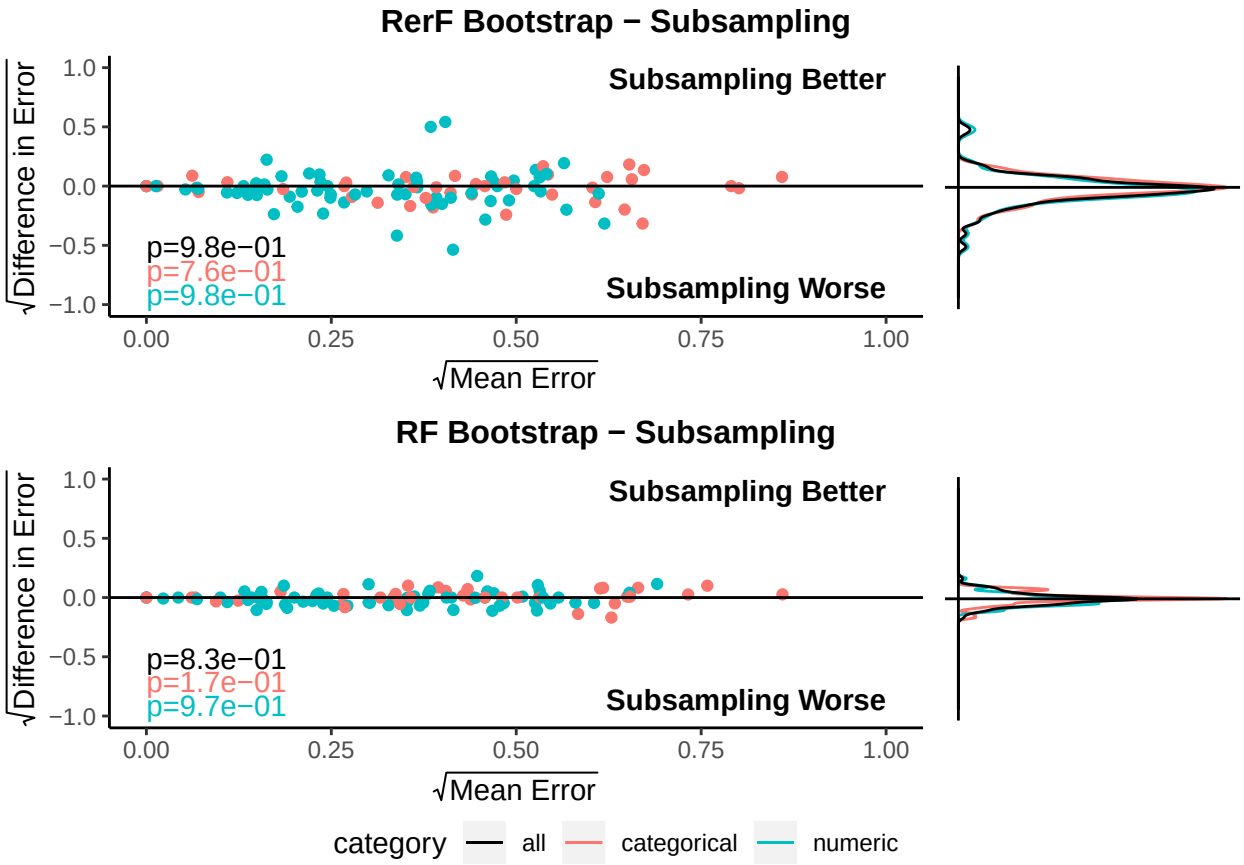
```r
# Combine figures
g.1 <- ggplotGrob(fig.1 + theme(legend.position = 'none'))
panel_id <- g.1$layout[g.1$layout$name == "panel",c("t","l")]
g.1 <- gtable_add_cols(g.1, unit(4,"cm"))
g.1 <- gtable_add_grob(g.1, ggplotGrob(kde.1 + theme(legend.position = 'none', plot.margin = unit(c(.13
                       t = panel_id$t, l = ncol(g.1))

g.2 <- ggplotGrob(fig.2 + theme(legend.position = 'none'))
panel_id <- g.2$layout[g.2$layout$name == "panel",c("t","l")]
g.2 <- gtable_add_cols(g.2, unit(4,"cm"))
g.2 <- gtable_add_grob(g.2, ggplotGrob(kde.2 + theme(legend.position = 'none', plot.margin = unit(c(.13
                       t = panel_id$t, l = ncol(g.2))

top <- grid.arrange(g.1, nrow = 1, top = textGrob("RerF Bootstrap - Subsampling", gp=gpar(fontface = "b
```



```
bottom <- grid.arrange(g.2, nrow = 1, top = textGrob("RF Bootstrap - Subsampling", gp=gpar(fontface = "
```




```
output <- grid.arrange(top, bottom,leg, nrow = 3, heights=c(1, 1, .1))
```




```
ggsave(filename = './result.pdf', plot = output, width = 7, height = 7)
```