permute_reg

Kelvyn Bladen

2024-05-24

Title

```
library(GGally)
library(randomForest)
library(dplyr)
library(randomForestVIP)
```

```
set.seed(123)
perm_LOCO <- vector(length = 6)
drop_LOCO <- vector(length = 6)
PaP <- vector(length = 6)
DaP <- vector(length = 6)
mrep <- 1
for (j in seq_len(mrep)) {
  sig <- diag(1, 6, 6)

  df <- MASS::mvrnorm(1000, mu = rep(0, 6), Sigma = sig)

  y <- 5 * df[, 1] + 4 * df[, 2] + 3 * df[, 3] +
    2 * df[, 4] + 1 * df[, 5] + rnorm(1000, mean = 0, sd = 1)
  df <- data.frame(cbind(df, y))

  dfv <- MASS::mvrnorm(1000, mu = rep(0, 6), Sigma = sig)
  yv <- 5 * dfv[, 1] + 4 * dfv[, 2] + 3 * dfv[, 3] +
    2 * dfv[, 4] + 1 * dfv[, 5] + rnorm(1000, mean = 0, sd = 1)
  dfv <- data.frame(cbind(dfv, yv))

  reg_full <- lm(y ~ ., data = df)
  s <- summary(reg_full)
  # m <- mean(s$residuals^2)

  p = predict(reg_full, dfv)
  mv = mean((p-dfv$yv)^2)

  imp <- vector(length = 6)
  impv <- vector(length = 6)
  lp <- list()
  for (i in seq_len(6)) {
    df_new <- df
    df_new[i] <- df_new[sample(1:1000), i]
```

```
  reg <- lm(y ~ ., data = df_new)
  sp <- summary(reg)
  lp[[i]] <- sp
  names(lp)[i] <- paste0("s", i)
  # imp[i] <- (new_m - m)/m

  p = predict(reg, dfv)
  new_mv = mean((p-dfv$yv)^2)
  imp[i] <- (new_mv - mv)#/mv

  dfv_new <- dfv
  dfv_new[i] <- dfv_new[sample(1:1000), i]
  p = predict(reg_full, dfv_new)
  new_mv = mean((p-dfv$yv)^2)
  impv[i] <- (new_mv - mv)#/mv
}

imp1 <- pmax(imp, 0)
simp <- sqrt(imp1)
perm_LOCO <- perm_LOCO + simp / mrep

impv1 <- pmax(impv, 0)
simpv <- sqrt(impv1)
PaP <- PaP + simpv / mrep

drop_imp <- vector(length = 6)
drop_impv <- vector(length = 6)
ld <- list()
for (i in seq_len(6)) {
  df_new <- df
  df_new[, i] <- 0
  reg <- lm(y ~ ., data = df_new)
  sd <- summary(reg)
  ld[[i]] <- sd
  names(ld)[i] <- paste0("s", i)
  # drop_imp[i] <- new_m - m

  p = predict(reg, dfv)
  new_mv = mean((p-dfv$yv)^2)
  drop_imp[i] <- (new_mv - mv)#/mv

  dfv_new <- dfv
  dfv_new[, i] <- 0
  p = predict(reg_full, dfv_new)
  new_mv = mean((p-dfv$yv)^2)
  drop_impv[i] <- (new_mv - mv)#/mv
}

drop_imp1 <- pmax(drop_imp, 0)
drop_simp <- sqrt(drop_imp1)
drop_LOCO <- drop_LOCO + drop_simp / mrep

drop_impv1 <- pmax(drop_impv, 0)
```

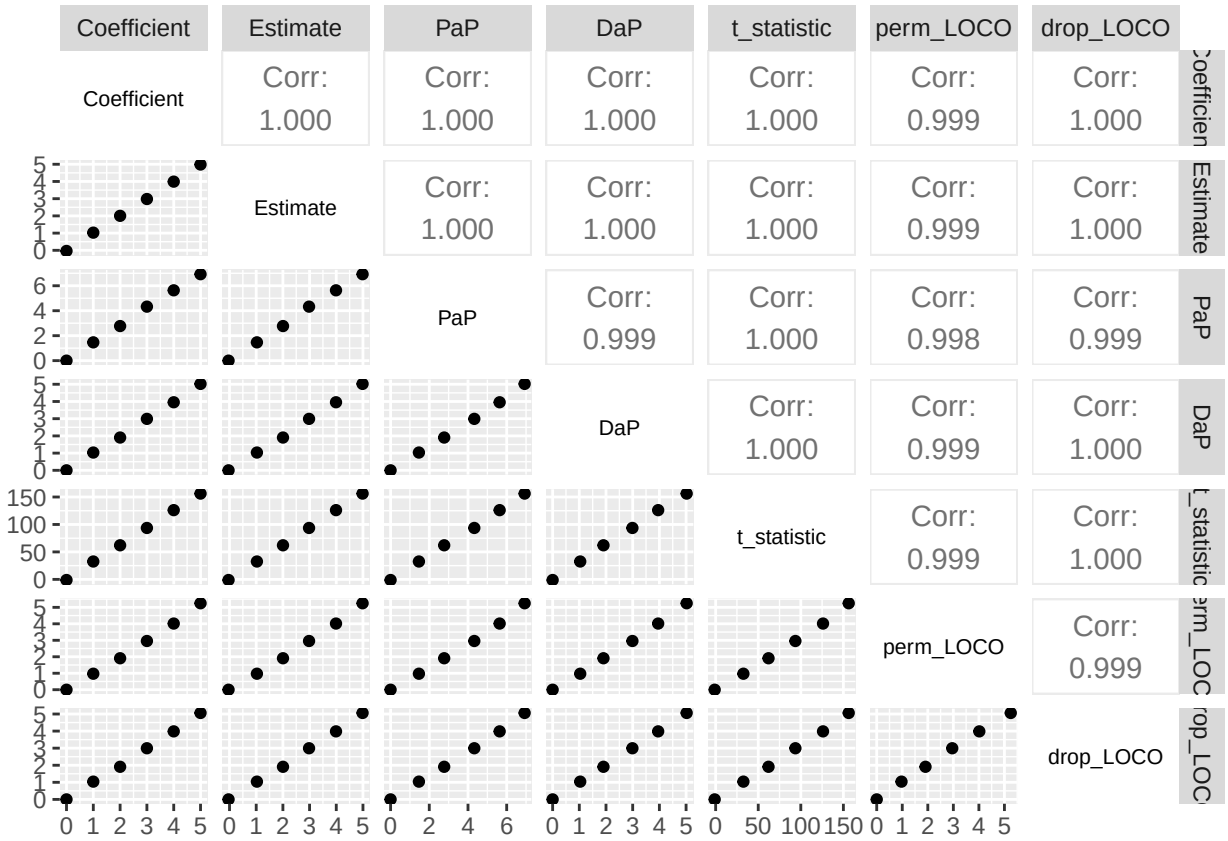
```
  drop_simpv <- sqrt(drop_impv1)
  DaP <- DaP + drop_simpv / mrep
}
```

```
## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases

## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases

## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases

## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases

## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases

## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases
```

```
both <- as.data.frame(cbind(Coefficient = 5:0,
                            Estimate = s$coefficients[-1, 1],
                            PaP, DaP,
                            t_statistic = s$coefficients[-1, 3],
                            perm_LOCO, drop_LOCO))
# why do these match t rather than t^2? Because I square rooted them
# ggpairs(both)
#cor(both)
both
```

```
##    Coefficient    Estimate      PaP      DaP t_statistic perm_LOCO drop_LOCO
## V1           5  4.98543430 6.925753 5.022245   156.24740 5.2471332  5.061451
## V2           4  3.98991925 5.634592 3.954879   126.08679 4.0163690  3.983921
## V3           3  2.98037840 4.326099 2.990007    93.64010 2.9572169  2.995342
## V4           2  2.00427694 2.770332 1.905021    62.06814 1.9096194  1.909263
## V5           1  1.02695296 1.464473 1.033678    32.58579 0.9707473  1.040029
## V6           0 -0.03671856 0.000000 0.000000    -1.15053 0.0000000  0.000000
```

```
g <- ggpairs(both,
             upper = list(continuous = wrap(ggally_cor,
                                            stars = F)),
             diag = list("continuous" = function(data, mapping, ...){
               ggally_text(rlang::as_label(mapping$x), col="black", size = 2.8) +
                theme_void()
       })
     )
g
```




```
ggsave("ind.pdf", g, dpi = 2400, width = 8, height = 8)
```

```
set.seed(123)
perm_LOCO <- vector(length = 6)
drop_LOCO <- vector(length = 6)
PaP <- vector(length = 6)
DaP <- vector(length = 6)
mrep <- 1
for (j in seq_len(mrep)) {
  sig <- diag(1, 6, 6)

  df <- MASS::mvrnorm(1000, mu = rep(0, 6), Sigma = sig)

  y <- 5 * df[, 1] + 4 * df[, 2] + 3 * df[, 3] +
    2 * df[, 4] + 1 * df[, 5] + rnorm(1000, mean = 0, sd = 10)
  df <- data.frame(cbind(df, y))

  dfv <- MASS::mvrnorm(1000, mu = rep(0, 6), Sigma = sig)
  yv <- 5 * dfv[, 1] + 4 * dfv[, 2] + 3 * dfv[, 3] +
    2 * dfv[, 4] + 1 * dfv[, 5] + rnorm(1000, mean = 0, sd = 10)
  dfv <- data.frame(cbind(dfv, yv))

  reg_full <- lm(y ~ ., data = df)
  s <- summary(reg_full)
  # m <- mean(s$residuals^2)
```

```
p = predict(reg_full, dfv)
mv = mean((p-dfv$yv)^2)

imp <- vector(length = 6)
impv <- vector(length = 6)
lp <- list()
for (i in seq_len(6)) {
  df_new <- df
  df_new[i] <- df_new[sample(1:1000), i]
  reg <- lm(y ~ ., data = df_new)
  sp <- summary(reg)
  lp[[i]] <- sp
  names(lp)[i] <- paste0("s", i)
  # imp[i] <- (new_m - m)/m

  p = predict(reg, dfv)
  new_mv = mean((p-dfv$yv)^2)
  imp[i] <- (new_mv - mv)#/mv

  dfv_new <- dfv
  dfv_new[i] <- dfv_new[sample(1:1000), i]
  p = predict(reg_full, dfv_new)
  new_mv = mean((p-dfv$yv)^2)
  impv[i] <- (new_mv - mv)#/mv
}

imp1 <- pmax(imp, 0)
simp <- sqrt(imp1)
perm_LOCO <- perm_LOCO + simp / mrep

impv1 <- pmax(impv, 0)
simpv <- sqrt(impv1)
PaP <- PaP + simpv / mrep

drop_imp <- vector(length = 6)
drop_impv <- vector(length = 6)
ld <- list()
for (i in seq_len(6)) {
  df_new <- df
  df_new[, i] <- 0
  reg <- lm(y ~ ., data = df_new)
  sd <- summary(reg)
  ld[[i]] <- sd
  names(ld)[i] <- paste0("s", i)
  # drop_imp[i] <- new_m - m

  p = predict(reg, dfv)
  new_mv = mean((p-dfv$yv)^2)
  drop_imp[i] <- (new_mv - mv)#/mv

  dfv_new <- dfv
  dfv_new[, i] <- 0
  p = predict(reg_full, dfv_new)
```

```
    new_mv = mean((p-dfv$yv)^2)
    drop_impv[i] <- (new_mv - mv)#/mv
  }

  drop_imp1 <- pmax(drop_imp, 0)
  drop_simp <- sqrt(drop_imp1)
  drop_LOCO <- drop_LOCO + drop_simp / mrep

  drop_impv1 <- pmax(drop_impv, 0)
  drop_simpv <- sqrt(drop_impv1)
  DaP <- DaP + drop_simpv / mrep
}
```

```
## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases

## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases

## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases

## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases

## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases

## Warning in predict.lm(reg, dfv): prediction from rank-deficient fit; attr(*,
## "non-estim") has doubtful cases
```

```
s
```

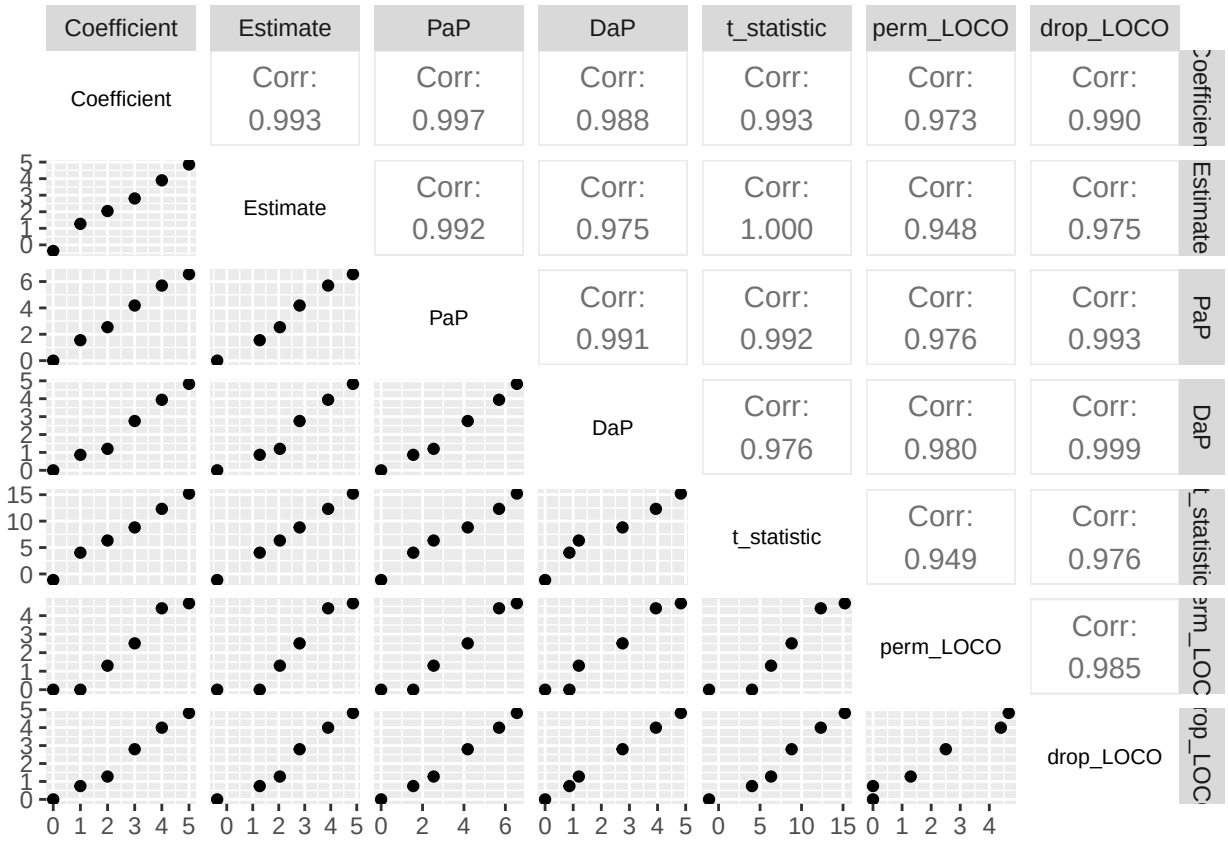
```
##
## Call:
## lm(formula = y ~ ., data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.832  -6.587  -0.045   7.062  33.574
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3337     0.3158  -1.057    0.291
## V1            4.8543     0.3191  15.214  < 2e-16 ***
## V2            3.8992     0.3164  12.322  < 2e-16 ***
## V3            2.8038     0.3183   8.809  < 2e-16 ***
## V4            2.0428     0.3229   6.326 3.80e-10 ***
## V5            1.2695     0.3152   4.028 6.05e-05 ***
## V6           -0.3672     0.3191  -1.151    0.250
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 9.96 on 993 degrees of freedom
## Multiple R-squared:  0.3518, Adjusted R-squared:  0.3478
## F-statistic: 89.81 on 6 and 993 DF,  p-value: < 2.2e-16
```

```
both <- as.data.frame(cbind(Coefficient = 5:0,
                            Estimate = s$coefficients[-1, 1],
                            PaP, DaP,
                            t_statistic = s$coefficients[-1, 3],
                            perm_LOCO, drop_LOCO))
# why do these match t rather than t^2? Because I square rooted them
# ggpairs(both)
#cor(both)
both
```

```
##    Coefficient   Estimate      PaP       DaP t_statistic perm_LOCO drop_LOCO
## V1           5  4.8543430 6.557488 4.8283590   15.213889  4.668101 4.8173287
## V2           4  3.8991925 5.695811 3.9413427   12.321970  4.396956 3.9971468
## V3           3  2.8037840 4.183191 2.7548718    8.809171  2.504448 2.7978296
## V4           2  2.0427694 2.537634 1.2004227    6.326017  1.297354 1.2739812
## V5           1  1.2695296 1.550587 0.8655113    4.028288  0.000000 0.7436914
## V6           0 -0.3671856 0.000000 0.0000000   -1.150530  0.000000 0.0000000
```

```
g1 <- ggpairs(both,
              upper = list(continuous = wrap(ggally_cor,
                                             stars = F)),
              diag = list("continuous" = function(data, mapping, ...){
                ggally_text(rlang::as_label(mapping$x), col="black", size = 2.8) +
                 theme_void()
        })
      )
g1
```




```
ggsave("indsd.pdf", g1, dpi = 2400, width = 8, height = 8)
```

```r
perm_LOCO <- vector(length = 8)
drop_LOCO <- vector(length = 8)

PaP <- vector(length = 8)
DaP <- vector(length = 8)

perm_LOCO_rf <- vector(length = 8)
drop_LOCO_rf <- vector(length = 8)

PaP_rf <- vector(length = 8)
DaP_rf <- vector(length = 8)

set.seed(123)
mrep <- 1
sdv = 1
n_size = 1000

for (j in seq_len(mrep)) {
  sig <- diag(1, 12, 12)

  for (i in 1:4) {
    for (k in 1:4) {
      sig[i, k] <- ifelse(i == k, 1, 0.95)
    }
  }
  strobl <- MASS::mvrnorm(n_size, mu = rep(0, 12), Sigma = sig)

  y <- 5 * strobl[, 1] + 5 * strobl[, 2] + 2 * strobl[, 3] +
    5 * strobl[, 5] + 5 * strobl[, 6] + 2 * strobl[, 7] +
    rnorm(n_size, mean = 0, sd = sdv)
  strobl <- data.frame(cbind(strobl, y))

  dfv <- MASS::mvrnorm(n_size, mu = rep(0, 12), Sigma = sig)
  yv <- 5 * dfv[, 1] + 5 * dfv[, 2] + 2 * dfv[, 3] +
    5 * dfv[, 5] + 5 * dfv[, 6] + 2 * dfv[, 7] +
    rnorm(n_size, mean = 0, sd = sdv)
  dfv <- data.frame(cbind(dfv, yv))

  reg_full <- lm(y ~ ., data = strobl)
  sc <- summary(reg_full)
  m <- mean(sc$residuals^2)

  p = predict(reg_full, dfv)
  mv = mean((p-dfv$yv)^2)

  # set.seed(123)
  rf <- randomForest(y ~ ., data = strobl, importance = TRUE, mtry = 10)
  imp = sqrt(as.data.frame(pmax(importance(rf), 0)))
  imp_df = imp[1:8, ]

  pt = predict(rf, strobl)
  mvt = mean((pt - strobl$y)^2)
  pv = predict(rf, dfv)
```

```
mvr = mean((pv - dfv$yv)^2)

impr <- vector(length = 8)
impv <- vector(length = 8)
imprfr <- vector(length = 8)
imprfv <- vector(length = 8)

for (i in seq_len(8)) {
  df_new <- strobl
  df_new[i] <- df_new[sample(1:n_size), i]
  reg <- lm(y ~ ., data = df_new)
  spc <- summary(reg)

  prv = predict(reg, dfv)
  new_m2 = mean((prv-dfv$yv)^2)
  impr[i] <- (new_m2 - mv)

  rf1 = randomForest(y ~ ., data = df_new, mtry = 10)

  prfv = predict(rf1, dfv)
  new_rfm2 = mean((prfv-dfv$yv)^2)
  imprfr[i] <- (new_rfm2 - mvr)

  dfv_new <- dfv
  dfv_new[i] <- dfv_new[sample(1:n_size), i]

  pp_PaP = predict(reg_full, dfv_new)
  sp_PaP = (pp_PaP-dfv_new$yv)
  new_mv = mean((pp_PaP-dfv_new$yv)^2)
  impv[i] <- (new_mv - mv)

  pr_PaP = predict(rf, dfv_new)
  new_rfmv = mean((pr_PaP-dfv_new$yv)^2)
  imprfv[i] <- (new_rfmv - mvr)
}

imp1 <- pmax(impr, 0)
simp <- sqrt(imp1)
perm_LOCO <- perm_LOCO + simp / mrep

impv1 <- pmax(impv, 0)
simpv <- sqrt(impv1)
PaP <- PaP + simpv / mrep

imprf1 <- pmax(imprfr, 0)
simprf <- sqrt(imprf1)
perm_LOCO_rf <- perm_LOCO + simprf / mrep

imprfv1 <- pmax(imprfv, 0)
simprfv <- sqrt(imprfv1)
PaP_rf <- PaP + simprfv / mrep

drop_impr <- vector(length = 8)
```

```
  drop_impv <- vector(length = 8)
  drop_imprf <- vector(length = 8)
  drop_imprfv <- vector(length = 8)

  for (i in seq_len(8)) {
    df_new <- strobl
    df_new[, i] <- 0
    reg <- lm(y ~ ., data = df_new)
    sdc <- summary(reg)

    prv = predict(reg, dfv)
    new_m2 = mean((prv-dfv$yv)^2)
    drop_impr[i] <- (new_m2 - mv)

    rf2 = randomForest(y ~ ., data = df_new, mtry = 10)

    prfv = predict(rf2, dfv)
    new_rfm2 = mean((prfv-dfv$yv)^2)
    drop_imprf[i] <- (new_rfm2 - mvr)

    dfv_new <- dfv
    dfv_new[, i] <- 0

    dp_PaP = predict(reg_full, dfv_new)
    sd_PaP = (dp_PaP-dfv_new$yv)
    new_mv = mean((dp_PaP-dfv_new$yv)^2)
    drop_impv[i] <- (new_mv - mv)

    pr_PaP = predict(rf, dfv_new)
    new_rfmv = mean((pr_PaP-dfv_new$yv)^2)
    drop_imprfv[i] <- (new_rfmv - mvr)
  }

  imp1 <- pmax(drop_impr, 0)
  simp <- sqrt(imp1)
  drop_LOCO <- drop_LOCO + simp / mrep

  drop_impv1 <- pmax(drop_impv, 0)
  drop_simpv <- sqrt(drop_impv1)
  DaP <- DaP + drop_simpv / mrep

  imprf1 <- pmax(drop_imprf, 0)
  simprf <- sqrt(imprf1)
  drop_LOCO_rf <- drop_LOCO + simprf / mrep

  drop_imprfv1 <- pmax(drop_imprfv, 0)
  drop_simprfv <- sqrt(drop_imprfv1)
  DaP_rf <- DaP + drop_simprfv / mrep
}

# df = as.data.frame(cbind(pp_res, dp_res, pp_PaP, dp_PaP))
# ggpairs(df)
# df = as.data.frame(cbind(sp_res, sd_res, sp_PaP, sd_PaP))
```

```r
# ggpairs(df)
sc
```

```
##
## Call:
## lm(formula = y ~ ., data = strobl)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.94053 -0.64838 -0.00269  0.64161  2.99716
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.062709   0.032084  -1.955   0.0509 .
## V1           4.961914   0.126744  39.149   <2e-16 ***
## V2           4.935223   0.125017  39.476   <2e-16 ***
## V3           2.103607   0.124992  16.830   <2e-16 ***
## V4          -0.017054   0.125289  -0.136   0.8918
## V5           5.049327   0.030788 164.003   <2e-16 ***
## V6           5.024257   0.032156 156.246   <2e-16 ***
## V7           2.044826   0.032303  63.302   <2e-16 ***
## V8           0.011242   0.032381   0.347   0.7285
## V9           0.003436   0.032213   0.107   0.9151
## V10          0.003801   0.032362   0.117   0.9065
## V11         -0.016438   0.032817  -0.501   0.6165
## V12         -0.045409   0.032006  -1.419   0.1563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.009 on 987 degrees of freedom
## Multiple R-squared:  0.995,  Adjusted R-squared:  0.9949
## F-statistic: 1.632e+04 on 12 and 987 DF,  p-value: < 2.2e-16
```

```r
rf$rsq[500]
```

```
## [1] 0.9452793
```

```r
sdf <- as.data.frame(cbind(Coefficient = c(5, 5, 2, 0, 5, 5, 2, 0),
                           Estimate = sc$coefficients[2:9, 1],
                           PaP, DaP,
                           t_statistic = sc$coefficients[2:9, 3],
                           perm_LOCO, drop_LOCO))
sdf
```

```
##    Coefficient   Estimate      PaP      DaP t_statistic perm_LOCO drop_LOCO
## V1           5  4.9619141 7.077902 5.013704  39.1491862 1.3512067  1.350120
## V2           5  4.9352225 7.224171 4.985636  39.4764131 1.2553710  1.264738
## V3           2  2.1036069 2.911051 2.133678  16.8299299 0.5233233  0.516049
## V4           0 -0.0170542 0.000000 0.000000  -0.1361185 0.0000000  0.000000
## V5           5  5.0493265 7.194511 4.982229 164.0026438 5.2490694  5.023065
## V6           5  5.0242571 7.133430 4.986592 156.2460951 5.1382030  5.033660
## V7           2  2.0448264 2.812301 1.988222  63.3019926 1.9651379  1.999274
## V8           0  0.0112421 0.000000 0.000000   0.3471839 0.0000000  0.000000
```

```r
# ggpairs(sdf)

status = as.factor(rep(c("Corr", "Orth"), each = 4))

g <- ggpairs(sdf, legend = c(2,1),
             lower = list(mapping = aes(shape = status)),
             upper = list(continuous = wrap(ggally_cor,
                                            stars = F)),
             diag = list("continuous" = function(data, mapping, ...){
               ggally_text(rlang::as_label(mapping$x), col="black", size = 2.8) +
                theme_void()
        })
      )

ggsave("strobl.pdf", g, dpi = 2400, width = 8, height = 8)

# set.seed(123)
#vimp = vip::vi_firm(rf, train = strobl)
#pdpImp <- vimp$Importance
#imp_df = cbind(imp, pdpImp)[1:8, ]

sdf1 <- as.data.frame(cbind(Coefficient = c(5, 5, 2, 0, 5, 5, 2, 0),
                            Estimate = sc$coefficients[2:9, 1],
                            PaP = PaP_rf,
                            DaP = DaP_rf,
                            t_statistic = sc$coefficients[2:9, 3],
                            OOB_PaP = imp_df$`%IncMSE`,
                            perm_LOCO = perm_LOCO_rf,
                            drop_LOCO = drop_LOCO_rf))
sdf1
```

```
##    Coefficient   Estimate         PaP        DaP t_statistic   OOB_PaP
## V1           5  4.9619141 13.82830562 10.8104176  39.1491862  5.369155
## V2           5  4.9352225 12.20472196  9.3330441  39.4764131  4.620046
## V3           2  2.1036069  6.60654862  5.5215857  16.8299299  4.295767
## V4           0 -0.0170542  2.51510092  2.2430007  -0.1361185  3.724003
## V5           5  5.0493265 13.51093054  9.7974535 164.0026438 12.444552
## V6           5  5.0242571 13.10158384  9.7985968 156.2460951 12.188858
## V7           2  2.0448264  4.49267944  3.4484191  63.3019926  5.870911
## V8           0  0.0112421  0.02185911  0.2846968   0.3471839  1.196691
##    perm_LOCO drop_LOCO
## V1 2.5211147  2.480050
## V2 2.5279944  2.489376
## V3 0.6918654  0.516049
## V4 0.3777965  0.000000
## V5 9.9758237  9.745394
## V6 9.8630451  9.711194
## V7 3.3113348  3.229603
## V8 0.0000000  0.000000
```

```r
g1 <- ggpairs(sdf1, legend = c(2,1),
             lower = list(mapping = aes(shape = status)),
             upper = list(continuous = wrap(ggally_cor,
```

```
                                          stars = F)),
            diag = list("continuous" = function(data, mapping, ...){
              ggally_text(rlang::as_label(mapping$x), col="black", size = 2.8) +
               theme_void()
        })
      )

ggsave("stroblrf.pdf", g1, dpi = 2400, width = 9, height = 9)
```

Repeat with High SD

```
perm_LOCO <- vector(length = 8)
drop_LOCO <- vector(length = 8)

PaP <- vector(length = 8)
DaP <- vector(length = 8)

perm_LOCO_rf <- vector(length = 8)
drop_LOCO_rf <- vector(length = 8)

PaP_rf <- vector(length = 8)
DaP_rf <- vector(length = 8)

set.seed(123)
mrep <- 1
sdv = 10
n_size = 1000

for (j in seq_len(mrep)) {
  sig <- diag(1, 12, 12)

  for (i in 1:4) {
    for (k in 1:4) {
      sig[i, k] <- ifelse(i == k, 1, 0.95)
    }
  }
  strobl <- MASS::mvrnorm(n_size, mu = rep(0, 12), Sigma = sig)

  y <- 5 * strobl[, 1] + 5 * strobl[, 2] + 2 * strobl[, 3] +
    5 * strobl[, 5] + 5 * strobl[, 6] + 2 * strobl[, 7] +
    rnorm(n_size, mean = 0, sd = sdv)
  strobl <- data.frame(cbind(strobl, y))

  dfv <- MASS::mvrnorm(n_size, mu = rep(0, 12), Sigma = sig)
  yv <- 5 * dfv[, 1] + 5 * dfv[, 2] + 2 * dfv[, 3] +
    5 * dfv[, 5] + 5 * dfv[, 6] + 2 * dfv[, 7] +
    rnorm(n_size, mean = 0, sd = sdv)
  dfv <- data.frame(cbind(dfv, yv))

  reg_full <- lm(y ~ ., data = strobl)
  sc <- summary(reg_full)
  m <- mean(sc$residuals^2)

  p = predict(reg_full, dfv)
  mv = mean((p-dfv$yv)^2)

  # set.seed(123)
  rf <- randomForest(y ~ ., data = strobl, importance = TRUE, mtry = 10)
  imp = sqrt(as.data.frame(pmax(importance(rf), 0)))
  imp_df = imp[1:8, ]

  pt = predict(rf, strobl)
```

```
mvt = mean((pt - strobl$y)^2)
pv = predict(rf, dfv)
mvr = mean((pv - dfv$yv)^2)

impr <- vector(length = 8)
impv <- vector(length = 8)
imprfr <- vector(length = 8)
imprfv <- vector(length = 8)

for (i in seq_len(8)) {
  df_new <- strobl
  df_new[i] <- df_new[sample(1:n_size), i]
  reg <- lm(y ~ ., data = df_new)
  spc <- summary(reg)

  prv = predict(reg, dfv)
  new_m2 = mean((prv-dfv$yv)^2)
  impr[i] <- (new_m2 - mv)

  rf1 = randomForest(y ~ ., data = df_new, mtry = 10)

  prfv = predict(rf1, dfv)
  new_rfm2 = mean((prfv-dfv$yv)^2)
  imprfr[i] <- (new_rfm2 - mvr)

  dfv_new <- dfv
  dfv_new[i] <- dfv_new[sample(1:n_size), i]

  pp_PaP = predict(reg_full, dfv_new)
  sp_PaP = (pp_PaP-dfv_new$yv)
  new_mv = mean((pp_PaP-dfv_new$yv)^2)
  impv[i] <- (new_mv - mv)

  pr_PaP = predict(rf, dfv_new)
  new_rfmv = mean((pr_PaP-dfv_new$yv)^2)
  imprfv[i] <- (new_rfmv - mvr)
}

imp1 <- pmax(impr, 0)
simp <- sqrt(imp1)
perm_LOCO <- perm_LOCO + simp / mrep

impv1 <- pmax(impv, 0)
simpv <- sqrt(impv1)
PaP <- PaP + simpv / mrep

imprf1 <- pmax(imprfr, 0)
simprf <- sqrt(imprf1)
perm_LOCO_rf <- perm_LOCO + simprf / mrep

imprfv1 <- pmax(imprfv, 0)
simprfv <- sqrt(imprfv1)
PaP_rf <- PaP + simprfv / mrep
```

```
  drop_impr <- vector(length = 8)
  drop_impv <- vector(length = 8)
  drop_imprf <- vector(length = 8)
  drop_imprfv <- vector(length = 8)

  for (i in seq_len(8)) {
    df_new <- strobl
    df_new[, i] <- 0
    reg <- lm(y ~ ., data = df_new)
    sdc <- summary(reg)

    prv = predict(reg, dfv)
    new_m2 = mean((prv-dfv$yv)^2)
    drop_impr[i] <- (new_m2 - mv)

    rf2 = randomForest(y ~ ., data = df_new, mtry = 10)

    prfv = predict(rf2, dfv)
    new_rfm2 = mean((prfv-dfv$yv)^2)
    drop_imprf[i] <- (new_rfm2 - mvr)

    dfv_new <- dfv
    dfv_new[, i] <- 0

    dp_PaP = predict(reg_full, dfv_new)
    sd_PaP = (dp_PaP-dfv_new$yv)
    new_mv = mean((dp_PaP-dfv_new$yv)^2)
    drop_impv[i] <- (new_mv - mv)

    pr_PaP = predict(rf, dfv_new)
    new_rfmv = mean((pr_PaP-dfv_new$yv)^2)
    drop_imprfv[i] <- (new_rfmv - mvr)
  }

  imp1 <- pmax(drop_impr, 0)
  simp <- sqrt(imp1)
  drop_LOCO <- drop_LOCO + simp / mrep

  drop_impv1 <- pmax(drop_impv, 0)
  drop_simpv <- sqrt(drop_impv1)
  DaP <- DaP + drop_simpv / mrep

  imprf1 <- pmax(drop_imprf, 0)
  simprf <- sqrt(imprf1)
  drop_LOCO_rf <- drop_LOCO + simprf / mrep

  drop_imprfv1 <- pmax(drop_imprfv, 0)
  drop_simprfv <- sqrt(drop_imprfv1)
  DaP_rf <- DaP + drop_simprfv / mrep
}

# df = as.data.frame(cbind(pp_res, dp_res, pp_PaP, dp_PaP))
# ggpairs(df)
```

```r
# df = as.data.frame(cbind(sp_res, sd_res, sp_PaP, sd_PaP))
# ggpairs(df)
sc
```

```
##
## Call:
## lm(formula = y ~ ., data = strobl)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.4053  -6.4838  -0.0269   6.4161  29.9716
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.62709    0.32084  -1.955 0.050919 .
## V1           4.61914    1.26744   3.644 0.000282 ***
## V2           4.35223    1.25017   3.481 0.000521 ***
## V3           3.03607    1.24992   2.429 0.015318 *
## V4          -0.17054    1.25289  -0.136 0.891755
## V5           5.49327    0.30788  17.842  < 2e-16 ***
## V6           5.24257    0.32156  16.304  < 2e-16 ***
## V7           2.44826    0.32303   7.579    8e-14 ***
## V8           0.11242    0.32381   0.347 0.728527
## V9           0.03436    0.32213   0.107 0.915065
## V10          0.03801    0.32362   0.117 0.906515
## V11         -0.16438    0.32817  -0.501 0.616550
## V12         -0.45409    0.32006  -1.419 0.156282
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.09 on 987 degrees of freedom
## Multiple R-squared:  0.6732, Adjusted R-squared:  0.6692
## F-statistic: 169.4 on 12 and 987 DF,  p-value: < 2.2e-16
```

```r
rf$rsq[500]
```

```
## [1] 0.626135
```

```r
sdf <- as.data.frame(cbind(Coefficient = c(5, 5, 2, 0, 5, 5, 2, 0),
                           Estimate = sc$coefficients[2:9, 1],
                           PaP, DaP,
                           t_statistic = sc$coefficients[2:9, 3],
                           perm_LOCO, drop_LOCO))
sdf
```

```
##    Coefficient  Estimate      PaP      DaP t_statistic perm_LOCO drop_LOCO
## V1           5  4.619141 7.124743 5.027586   3.6444731 1.5630356 1.6805038
## V2           5  4.352225 5.863992 4.602371   3.4813069 1.0337857 0.7910235
## V3           2  3.036069 4.502083 3.256003   2.4290101 0.8761659 0.0000000
## V4           0 -0.170542 0.000000 0.000000  -0.1361185 0.0273760 0.0000000
## V5           5  5.493265 7.603963 5.003245  17.8421824 4.5686947 5.0811228
## V6           5  5.242571 6.816003 4.451636  16.3035292 4.2241815 4.4557516
```

```
## V7           2  2.448264 2.882627 1.597242   7.5791268 1.3310369 1.7284773
## V8           0  0.112421 0.000000 0.000000   0.3471839 0.0000000 0.0000000
```

```
g <- ggpairs(sdf, legend = c(2,1),
             lower = list(mapping = aes(shape = status)),
             upper = list(continuous = wrap(ggally_cor,
                                            stars = F)),
             diag = list("continuous" = function(data, mapping, ...){
               ggally_text(rlang::as_label(mapping$x), col="black", size = 2.8) +
                theme_void()
        })
      )

ggsave("stroblsd.pdf", g, dpi = 2400, width = 8, height = 8)

# set.seed(123)

#vimp = vip::vi_firm(rf, train = strobl)
#pdpImp <- vimp$Importance
#imp_df = cbind(imp, pdpImp)[1:8, ]

sdf1 <- as.data.frame(cbind(Coefficient = c(5, 5, 2, 0, 5, 5, 2, 0),
                            Estimate = sc$coefficients[2:9, 1],
                            PaP = PaP_rf,
                            DaP = DaP_rf,
                            t_statistic = sc$coefficients[2:9, 3],
                            OOB_PaP = imp_df$`%IncMSE`,
                            perm_LOCO = perm_LOCO_rf,
                            drop_LOCO = drop_LOCO_rf))
sdf1
```

```
##    Coefficient  Estimate        PaP       DaP t_statistic  OOB_PaP  perm_LOCO
## V1           5  4.619141 12.0180278 9.0552234   3.6444731 4.669251 3.40099314
## V2           5  4.352225  9.9249693 8.6582168   3.4813069 4.954401 2.26883603
## V3           2  3.036069  9.4365809 7.8870562   2.4290101 4.800424 1.68938667
## V4           0 -0.170542  3.6680317 2.9827334  -0.1361185 4.346653 1.06069194
## V5           5  5.493265 14.5747828 9.7433790  17.8421824 7.757175 9.21437012
## V6           5  5.242571 12.7726018 9.2813500  16.3035292 7.098502 8.70190299
## V7           2  2.448264  5.1943858 3.3625700   7.5791268 4.500052 3.05470220
## V8           0  0.112421  0.6273754 0.5884071   0.3471839 0.000000 0.07114153
##    drop_LOCO
## V1 3.5360363
## V2 2.0689739
## V3 0.7251835
## V4 0.8334416
## V5 9.8509538
## V6 9.0501492
## V7 3.2869063
## V8 1.0766890
```

```
g1 <- ggpairs(sdf1, legend = c(2,1),
             lower = list(mapping = aes(shape = status)),
             upper = list(continuous = wrap(ggally_cor,
```

```
                                          stars = F)),
             diag = list("continuous" = function(data, mapping, ...){
               ggally_text(rlang::as_label(mapping$x), col="black", size = 2.8) +
                theme_void()
         })
       )

ggsave("stroblrfsd.pdf", g1, dpi = 2400, width = 9, height = 9)
```

Repeat with Higher SD

```
# perm_LOCO <- vector(length = 8)
# drop_LOCO <- vector(length = 8)
#
# perm_LOCO_old <- vector(length = 8)
# drop_LOCO_old <- vector(length = 8)

perm_LOCO <- vector(length = 8)
drop_LOCO <- vector(length = 8)

PaP <- vector(length = 8)
DaP <- vector(length = 8)

set.seed(123)
mrep <- 1
sdv = 30
n_size = 1000

for (j in seq_len(mrep)) {
  sig <- diag(1, 12, 12)

  for (i in 1:4) {
    for (k in 1:4) {
      sig[i, k] <- ifelse(i == k, 1, 0.95)
    }
  }
  strobl <- MASS::mvrnorm(n_size, mu = rep(0, 12), Sigma = sig)

  y <- 5 * strobl[, 1] + 5 * strobl[, 2] + 2 * strobl[, 3] +
    5 * strobl[, 5] + 5 * strobl[, 6] + 2 * strobl[, 7] +
    rnorm(n_size, mean = 0, sd = sdv)
  strobl <- data.frame(cbind(strobl, y))

  dfv <- MASS::mvrnorm(n_size, mu = rep(0, 12), Sigma = sig)
  yv <- 5 * dfv[, 1] + 5 * dfv[, 2] + 2 * dfv[, 3] +
    5 * dfv[, 5] + 5 * dfv[, 6] + 2 * dfv[, 7] +
    rnorm(n_size, mean = 0, sd = sdv)
  dfv <- data.frame(cbind(dfv, yv))

  reg_full <- lm(y ~ ., data = strobl)
  sc <- summary(reg_full)
  m <- mean(sc$residuals^2)

  p = predict(reg_full, dfv)
  mv = mean((p-dfv$yv)^2)

  # imp <- vector(length = 8)
  # impo <- vector(length = 8)
  impr <- vector(length = 8)
  impv <- vector(length = 8)
  lpc <- list()
  for (i in seq_len(8)) {
```

```
  df_new <- strobl
  df_new[i] <- df_new[sample(1:n_size), i]
  reg <- lm(y ~ ., data = df_new)
  spc <- summary(reg)
  # new_m <- mean(spc$residuals^2)
  lpc[[i]] <- spc
  names(lpc)[i] <- paste0("s", i)
  # imp[i] <- new_m - m
  #
  # pp_res = predict(reg, df_new)
  # sp_res = (pp_res-df_new$y)
  #
  # pr = predict(reg, strobl)
  # new_m1 = mean((pr-strobl$y)^2)
  # impo[i] <- new_m1 - m

  prv = predict(reg, dfv)
  new_m2 = mean((prv-dfv$yv)^2)
  impr[i] <- (new_m2 - mv)#/mv

  dfv_new <- dfv
  dfv_new[i] <- dfv_new[sample(1:n_size), i]
  pp_PaP = predict(reg_full, dfv_new)
  sp_PaP = (pp_PaP-dfv_new$yv)
  new_mv = mean((pp_PaP-dfv_new$yv)^2)
  impv[i] <- (new_mv - mv)#/mv
}

# imp1 <- pmax(imp, 0)
# simp <- sqrt(imp1)
# perm_LOCO <- perm_LOCO + simp / mrep
#
# # new
# imp1 <- pmax(impo, 0)
# simp <- sqrt(imp1)
# perm_LOCO_old <- perm_LOCO_old + simp / mrep

imp1 <- pmax(impr, 0)
simp <- sqrt(imp1)
perm_LOCO <- perm_LOCO + simp / mrep
# end new

impv1 <- pmax(impv, 0)
simpv <- sqrt(impv1)
PaP <- PaP + simpv / mrep

# drop_imp <- vector(length = 8)
# drop_impo <- vector(length = 8)
drop_impr <- vector(length = 8)
drop_impv <- vector(length = 8)
ldc <- list()
for (i in seq_len(8)) {
  df_new <- strobl
```

```
    df_new[, i] <- 0
    reg <- lm(y ~ ., data = df_new)
    sdc <- summary(reg)
    # new_m <- mean(sdc$residuals^2)
    ldc[[i]] <- sdc
    names(ldc)[i] <- paste0("s", i)
    # drop_imp[i] <- new_m - m
    #
    # dp_res = predict(reg, df_new)
    # sd_res = (dp_res-df_new$y)
    #
    # pr = predict(reg, strobl)
    # new_m1 = mean((pr-strobl$y)^2)
    # drop_impo[i] <- new_m1 - m

    prv = predict(reg, dfv)
    new_m2 = mean((prv-dfv$yv)^2)
    drop_impr[i] <- (new_m2 - mv)#/mv

    dfv_new <- dfv
    dfv_new[, i] <- 0
    dp_PaP = predict(reg_full, dfv_new)
    sd_PaP = (dp_PaP-dfv_new$yv)
    new_mv = mean((dp_PaP-dfv_new$yv)^2)
    drop_impv[i] <- (new_mv - mv)#/mv # do this everywhere
  }

  # drop_imp1 <- pmax(drop_imp, 0)
  # drop_simp <- sqrt(drop_imp1)
  # drop_LOCO <- drop_LOCO + drop_simp / mrep
  #
  # # new
  # imp1 <- pmax(drop_impo, 0)
  # simp <- sqrt(imp1)
  # drop_LOCO_old <- drop_LOCO_old + simp / mrep

  imp1 <- pmax(drop_impr, 0)
  simp <- sqrt(imp1)
  drop_LOCO <- drop_LOCO + simp / mrep
  # end new

  drop_impv1 <- pmax(drop_impv, 0)
  drop_simpv <- sqrt(drop_impv1)
  DaP <- DaP + drop_simpv / mrep
}

# df = as.data.frame(cbind(pp_res, dp_res, pp_PaP, dp_PaP))
# # ggpairs(df)
#
# df = as.data.frame(cbind(sp_res, sd_res, sp_PaP, sd_PaP))
# ggpairs(df)
sc
```

```r
sdf <- as.data.frame(cbind(Coefficient = c(5, 5, 2, 0, 5, 5, 2, 0),
                           Estimate = sc$coefficients[2:9, 1],
                           PaP, DaP,
                           t_statistic = sc$coefficients[2:9, 3],
                           perm_LOCO, drop_LOCO))
#cor(sdf)
sdf
status = as.factor(rep(c("Corr", "Orth"), each = 4))
ggpairs(sdf, showStrips = FALSE, axisLabels = "internal")
ggpairs(sdf, aes(col = status))

g <- ggpairs(sdf, legend = c(2,1),
             lower = list(mapping = aes(shape = status)),
             upper = list(continuous = wrap(ggally_cor,
                                            stars = F)),
             diag = list("continuous" = function(data, mapping, ...){
               ggally_text(rlang::as_label(mapping$x), col="black", size = 2.8) +
                theme_void()
        })
) #+ scale_shape_manual(values = c(1, 0))

g


#ggsave("stroblsd5.pdf", g, dpi = 2400, width = 8, height = 8)

# sdfb <- as.data.frame(cbind(Coefficient = c(5, 5, 2, 0, 5, 5, 2, 0),
#              sc$coefficients[2:9, c(1, 3)],
#              # perm_LOCO, drop_LOCO,
#              # perm_LOCO_old, drop_LOCO_old,
#              perm_LOCO, drop_LOCO,
#              PaP, DaP))
# ggpairs(sdfb)


# set.seed(123)
# rf <- randomForest(y ~ ., data = strobl, importance = TRUE)
# rf$rsq[500]
# #vimp = vip::vi_firm(rf, train = strobl)
# imp = sqrt(as.data.frame(pmax(importance(rf), 0)))
# #pdpImp <- vimp$Importance
# #imp_df = cbind(imp, pdpImp)[1:8, ]
# imp_df = imp[1:8, ]

# pd <- pdp_compare(rf)
# pd$full_num
# imp <- pd$imp
#o <- order(as.numeric(gsub("V", "", imp$var)))[1:8]

#sdd <- cbind(sdf, pd$imp[o, c(2, 4, 6)])

# sdd <- as.data.frame(cbind(Coefficient = c(5, 5, 2, 0, 5, 5, 2, 0),
#              sc$coefficients[2:9, c(1, 3)],
```

```
#              perm_LOCO,
#              PaP, rf_permute = imp_df$`%IncMSE`))
#sdd
#cor(sdd)
#data.frame(v1 = cor(sdd)[1,])
#ggpairs(sdd)
#ggsave("stroblrf.jpg", ggpairs(sdd), dpi = 2400, width = 8, height = 8)

# sddb <- as.data.frame(cbind(Coefficient = c(5, 5, 2, 0, 5, 5, 2, 0),
#              sc$coefficients[2:9, c(1, 3)],
#              # perm_LOCO,
#              # perm_LOCO_old,
#              perm_LOCO,
#              PaP, imp_df))
# ggpairs(sddb)


# set.seed(123)
# rf12 <- randomForest(y ~ .,
#   data = strobl, importance = TRUE,
#   mtry = 12
# )
# rf12$rsq[500]
# #vimp12 = vip::vi_firm(rf12, train = strobl)
# imp12 = sqrt(as.data.frame(pmax(importance(rf12), 0)))
# #pdpImp12 <- vimp12$Importance
# #imp_df12 = cbind(imp12, pdpImp12)[1:8, ]
# imp_df12 = imp12[1:8, ]

# pd12 <- pdp_compare(rf, trellis = F)
# pd12$full_num
# imp <- pd12$imp
# o <- order(as.numeric(gsub("V", "", imp$var)))[1:8]

#sdd1 <- cbind(sdf, pd12$imp[o, c(4, 6)])

# sdd1 <- as.data.frame(cbind(Coefficient = c(5, 5, 2, 0, 5, 5, 2, 0),
#              sc$coefficients[2:9, c(1, 3)], perm_LOCO,
#              PaP, RF_permute = imp_df$`%IncMSE`,
#              RF12_permute = imp_df12$`%IncMSE`))
#cor(sdd1)
# sdd1
# data.frame(v1 = cor(sdd1)[1,])
#ggpairs(sdd1)
# ggpairs(sdd1)
# ggsave("stroblsdrf.jpg", ggpairs(sdd1), dpi = 2400, width = 8, height = 8)
```