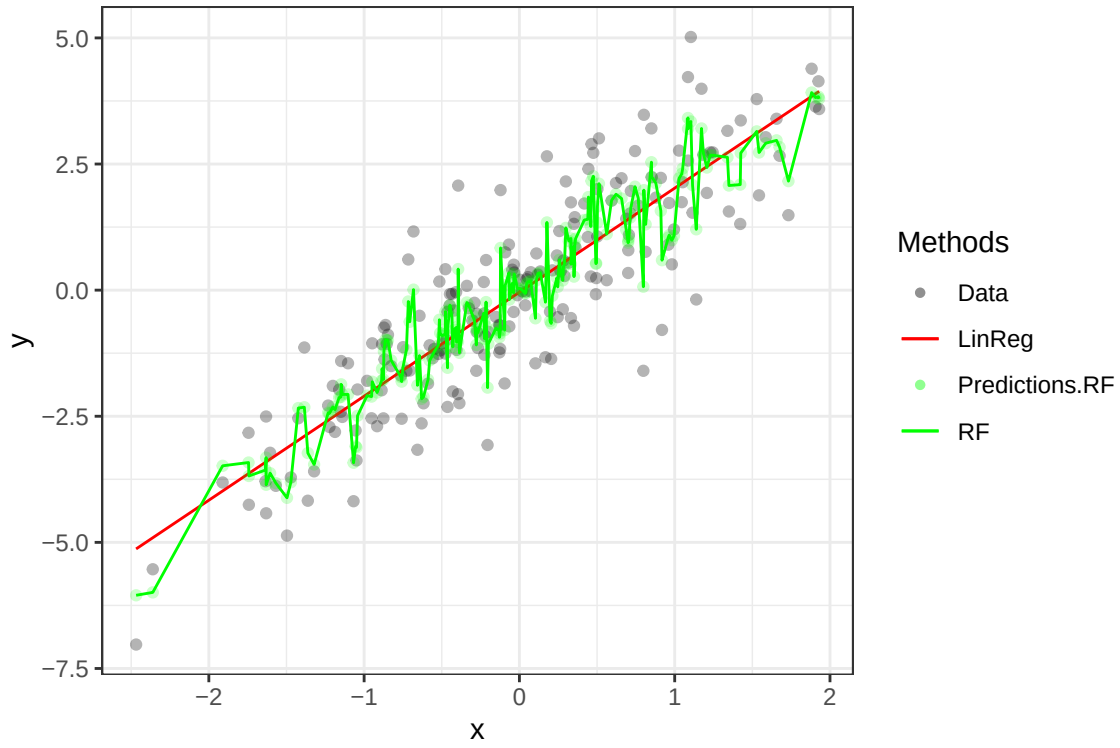
Estimating Variance of Simple Defined Variable Effect directly

Felix Kapulla

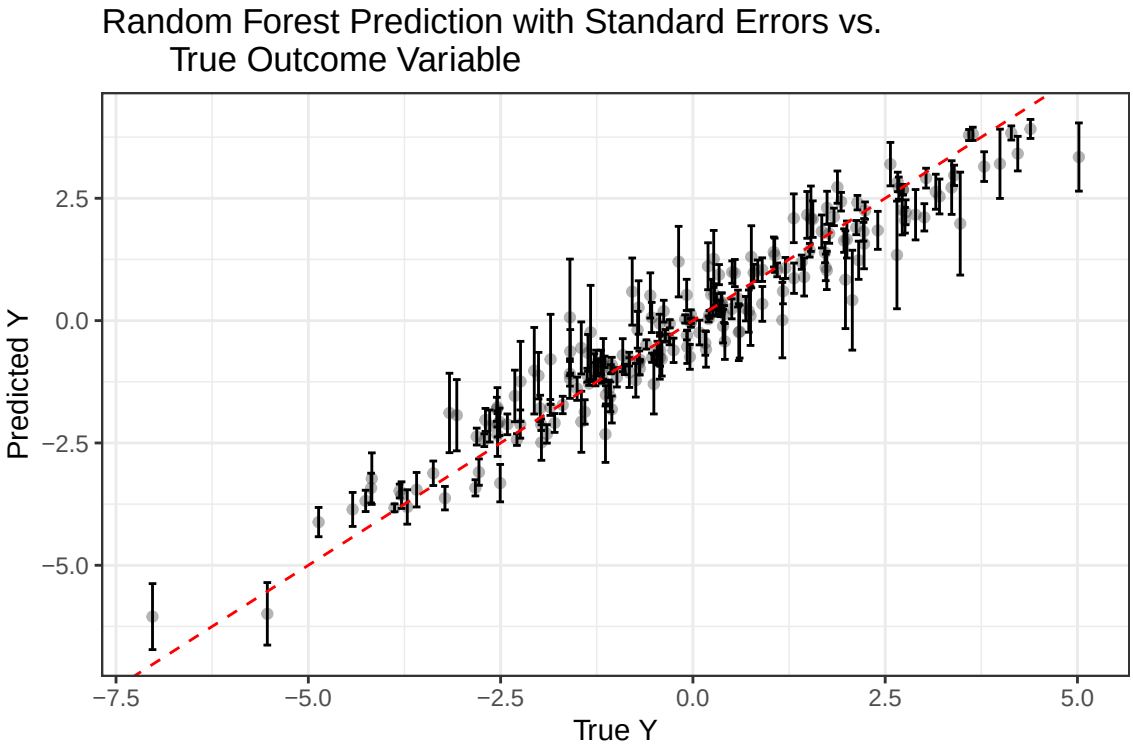
```
library(Matrix)
library(tidyverse)
library(ggplot2)
library(ggpubr)
library(ranger)
library(randomForest)
library(gridExtra)

source('C:/Users/feix_/iCloudDrive/Studium Master/CQM - Thesis Internship/Thesis-VariableEffects/Ranger
```

```
### Model Design
set.seed(124)
n <- 200
x <- rnorm(n, 0, 1)
e <- rnorm(n, 0, 1)
y <- 2*x + e
data <- data.frame(x, y)

### Fit Linear Regression Model
linreg <- lm(formula = y ~ x,
             data = data)
linreg.predictions <- linreg$fitted.values
linreg.se <- predict(linreg, newdata = data, se.fit = T)$se.fit

### Fit Random Forest Model from 'ranger'
rf <- ranger( formula = y ~ x,
              data = data,
              num.trees = 200,
              keep.inbag = T)


### Predict RF Responses (Training Data)
rf.predict <- RangerForestPredict(rf$forest,
                                  data,
                                  predict.all = T,
                                  inbag.counts = rf$inbag.counts)

# For each observation we get all predictions from all trees
rf.all.predictions <- rf.predict$predictions

# For each observation average predctions over all trees
rf.predictions <- rowMeans(rf.all.predictions)

#### Standard Error of Random Forest Predictions
```

```
rf.se <- RangerForestPredict(rf$forest,
                             data,
                             type = 'se',
                             se.method = 'jack',
                             inbag.counts = rf$inbag.counts)$se


data <- cbind(data, linreg.predictions, rf.predictions, linreg.se, rf.se)


# Plot Linear Regression and Random Forest fit
# Random Forest Predictions are plotted including standard errors
gg_fit <- ggplot(data = data) +
  geom_point(aes(x, y, colour='Data'), alpha=0.3) +
  geom_line(aes(x=x, y=linreg.predictions, colour='LinReg')) +
  geom_point(aes(x=x, y=rf.predictions, colour='Predictions.RF'), alpha=0.2) +
  geom_line(aes(x=x, y=rf.predictions, colour='RF')) +
  scale_color_manual(name='Methods',
                     values = c('Data'='black',
                                'LinReg'='red',
                                'Predictions.RF'='green',
                                'RF'='green'),
                     guide=guide_legend(override.aes = list(
                       linetype = c('blank', 'solid', 'blank', 'solid'),
                       shape = c(16, NA, 16, NA)
                     ))) +
  theme_bw()


gg_se <- ggplot(data = data, aes(x=y,y=rf.predictions)) +
  geom_point(alpha=0.3) +
  geom_errorbar(aes(ymin=rf.predictions - 0.5*rf.se,
                    ymax=rf.predictions + 0.5*rf.se), width=.1)+
  geom_abline(slope = 1, linetype = 2 , color = 'red')+
  labs(x = 'True Y', y = 'Predicted Y',
       title = 'Random Forest Prediction with Standard Errors vs.
       True Outcome Variable')+
  theme_bw()


gg_fit
```


5.0
2.5
0.0
−2.5
−5.0
−7.5
−2
−1
0
1
2
x
y
Methods
Data
LinReg
Predictions.RF
RF

```
gg_se
```


Random Forest Prediction with Standard Errors vs.
True Outcome Variable
2.5
0.0
−2.5
−5.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
True Y
Predicted Y

```r
#ggarrange(gg_fit, gg_se, nrow = 2)
```

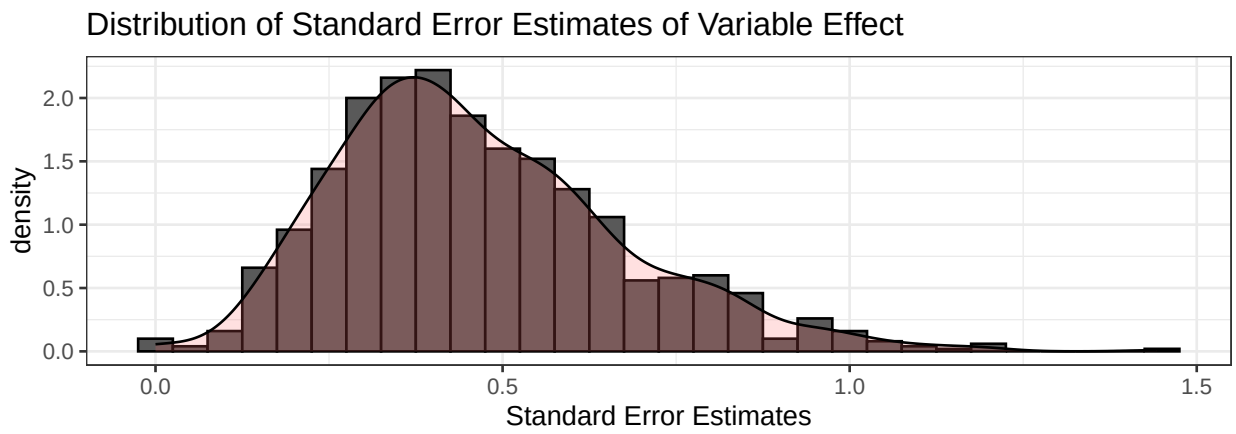
```r
###### Variable Effect and Standard Error #####
x_bar <- mean(x)
x_a <- mean(x) - sd(x)
x_b <- mean(x) + sd(x)
new_data <- data.frame(x = c(x_a, x_b))
```

```r
#### CoVariance Matrix of two Random Forest Predictions
rf.predict <- RangerForestPredict(rf$forest,
                                  new_data,
                                  type='se',
                                  se.method = 'jack_cov',
                                  predict.all = T,
                                  inbag.counts = rf$inbag.counts)

# Variance - Covariance Matrix between two predictions
rf.predict$cov
```

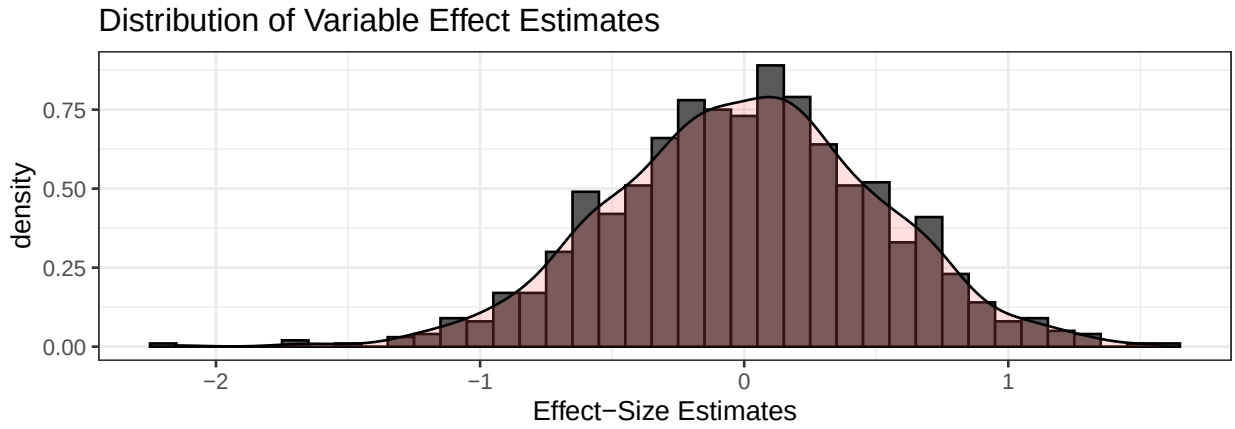
```
##            [,1]       [,2]
## [1,]  0.3043182 -0.1086148
## [2,] -0.1086148  0.5070766
```

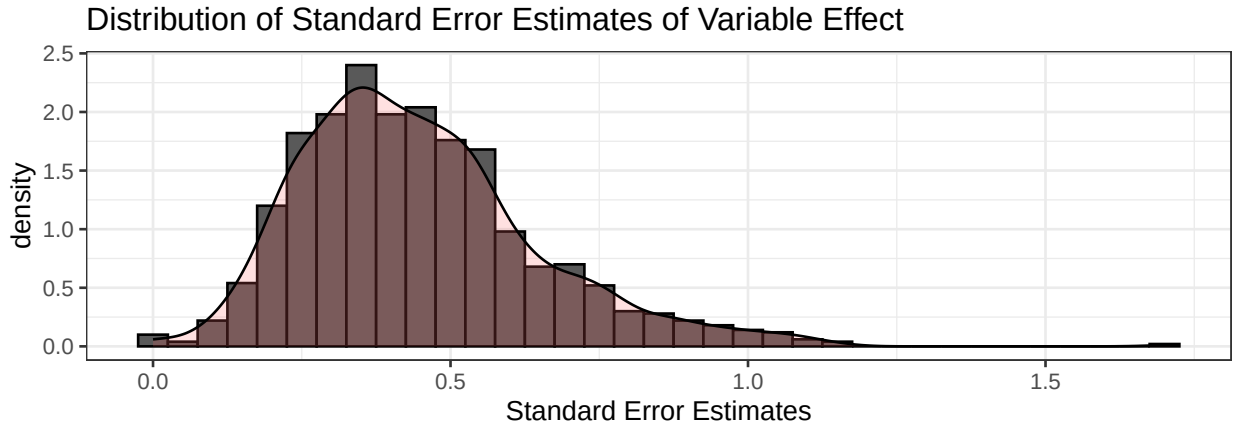
```
##### Simulation --- Distribution of Variable Effect and Standard Error
# Fit Random Forest n times
# Compute Variable Effect (as we defined it) n times
# Estimate Standard Error of Variable Effect n times directly
# ---> Based on estimated Variance-Covariance Matrix

# Compare Standard Error of simulated Variable Effects with mean
# of estimated Standard Errors of Variable Effects

sim_once <- function(n, num.trees, formula){


  x <- rnorm(n, 0, 1)
  e <- rnorm(n, 0, 1)
  formula <- parse(text = formula, list(x=x,e=e))
  y <- eval(formula)
  data <- data.frame(x, y)

  rf <- ranger( formula = y ~ x,
                data = data,
                num.trees = num.trees,
                keep.inbag = T)


  ### Variable Effect
  x_bar <- mean(x)
  x_a <- mean(x) - sd(x)
  x_b <- mean(x) + sd(x)
  new_data <- data.frame(x = c(x_a, x_b))

  rf.predict <- RangerForestPredict(rf$forest,
                                    new_data,
                                    type='se',
                                    se.method = 'jack_cov',
                                    predict.all = T,
                                    inbag.counts = rf$inbag.counts)

  ab_predictions <- rowMeans(rf.predict$predictions)
  effect <- (ab_predictions[2] - ab_predictions[1]) / 2*sd(x)
  ### VAR[(f(B) - f(A)) / 2*sd(x)]
  #   = ( 1 / 4*sd(x)^2 ) * ( VAR[f(B)] + VAR[f(B)] - 2*COV[f(A), f(B)] )
  effect.var <- pmax((rf.predict$cov[1,1] + rf.predict$cov[2,2] -
                        2*rf.predict$cov[1,2]) /  (2*sd(x))^2 , 0)
  effect.se <- sqrt(effect.var)

  return(list(effect = effect, effect.se = effect.se))

}


formulas <- c("2*x+e", "2*x^2+e")
for (formula in formulas) {
```

```
  res <- replicate(n = 1e3,
                   expr = sim_once(n = 200, num.trees = 200, formula = formula))


  ### Distribution of Test Statistics
  estimates <- data.frame(effect = unlist(res[1,]),
                          effect.se = unlist(res[2,]))


  g <- ggplot(estimates, aes(x = estimates[,1])) +
    geom_histogram(aes(y = after_stat(density)), binwidth = 0.1, color = 'black') +
    geom_density(alpha = .2, fill = "#FF6666") +
    labs(x = 'Effect-Size Estimates',
         title = 'Distribution of Variable Effect Estimates')+
    theme_bw()

  f <- ggplot(estimates, aes(x = estimates[,2])) +
    geom_histogram(aes(y = after_stat(density)), binwidth = 0.05, color = 'black') +
    geom_density(alpha = .2, fill = "#FF6666") +
    labs(x = 'Standard Error Estimates',
         title = 'Distribution of Standard Error Estimates of Variable Effect')+
    theme_bw()

  grid.arrange(g,f,nrow=2)




  cat('For functional relationship: ',
      formula,
      '\nMean of simulated Variable Effects: ',
      mean(unlist(res[1,1:ncol(res)])),
      '\nStandard Error of simulated Variable Effects: ',
      sd(unlist(res[1,1:ncol(res)])),
      '.\nMean of stimulated estimates of Standard Errors of Variable Effects: ',
      mean(estimates[,2]))

}
```




```
## For functional relationship:  2*x+e
## Mean of simulated Variable Effects:  2.002381
## Standard Error of simulated Variable Effects:  0.4604794 .
## Mean of stimulated estimates of Standard Errors of Variable Effects:  0.4648112
```






```
## For functional relationship:  2*x^2+e
## Mean of simulated Variable Effects:  0.00460506
## Standard Error of simulated Variable Effects:  0.5033887 .
## Mean of stimulated estimates of Standard Errors of Variable Effects:  0.4461205
```