Estimating Variance of Simple Defined Variable Main and Low-Order Interaction Effects

Felix Kapulla

```
knitr::opts_chunk$set(fig.width=14, fig.height=8)
```

```
library(Matrix)
library(tidyverse)
library(ggplot2)
library(ggpubr)
library(ranger)
library(MixMatrix)
library(mvtnorm)
library(stringr)
library(parallel)

cores <- detectCores()
clust <- makeCluster(4)

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

parallel::clusterEvalQ(clust,
                       expr = {source('C:/Users/feix_/iCloudDrive/Studium Master/CQM - Thesis Internshi
```

Simulation

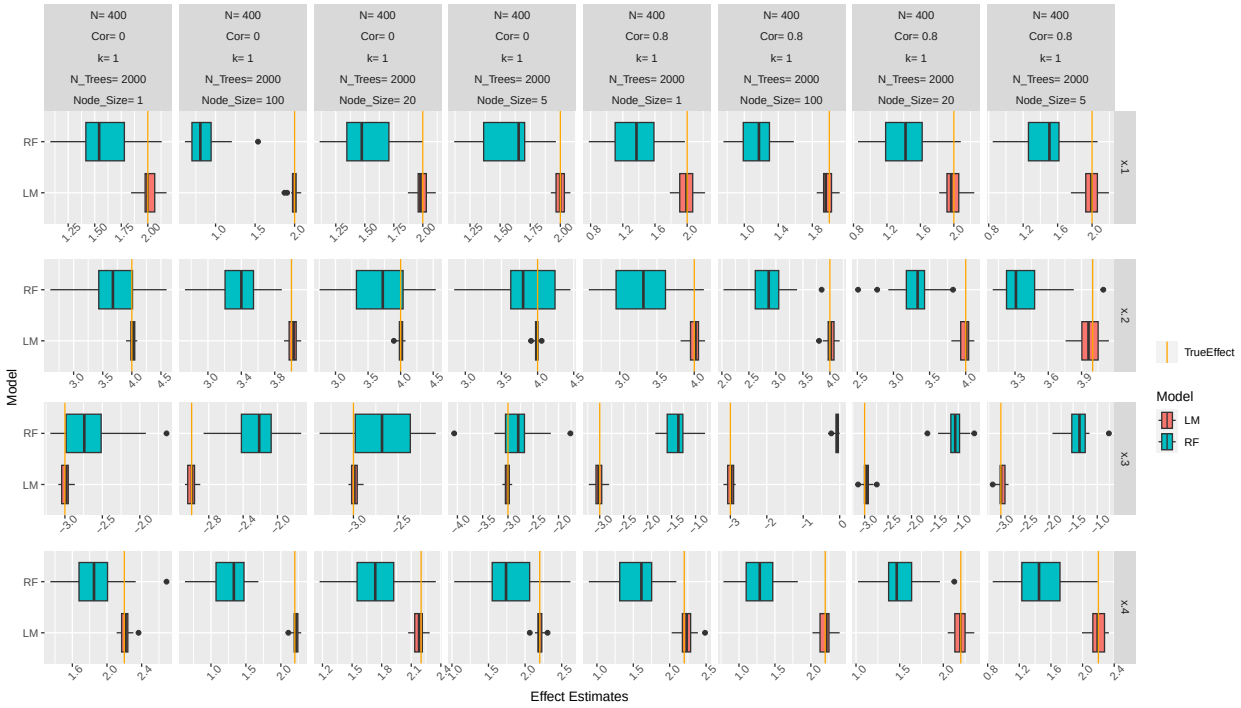
```
n <- c(400) ; num.trees <- 2000 ; repeats <- 25; cor <- c(0, 0.8)
k <- c(1); node_size <- c(1, 5, 20, 100); pdp <- T

formulas <- c("2*x.1+4*x.2-3*x.3+2.2*x.4")
parallel::clusterExport(cl = clust, varlist = 'formulas')
scenarios <- data.frame(expand.grid(n, num.trees, formulas, repeats,
                                    cor, k, node_size, pdp))
colnames(scenarios) = c("N", "N_Trees", "Formula", "Repeats",
                        "Correlation", "k", "Node_Size", "pdp")
scenarios$k_pdp <- (scenarios$k == unique(scenarios$k)[1])
scenarios[,"Formula"] <- as.character(scenarios[,"Formula"]) ### Formula became Factor
scenarios <- split(scenarios, seq(nrow(scenarios)))
#system.time(result <- lapply(X = scenarios, FUN = sim_multi))
#Run Simulation
system.time(result <- parLapply(cl = clust,
                                X = scenarios,
                                fun = sim_multi))
```

```
##    user  system elapsed
##    0.02    0.06  927.17
```

```
if (!pdp) {
 print_results(result)
}
effect_plots <- plot_effects(result)
```

```
## `summarise()` has grouped output by 'N', 'cor', 'k', 'num.trees', 'node_size'.
## You can override using the `.groups` argument.
```

```
#se_plot <- plot_se(result)
effect_plots
```

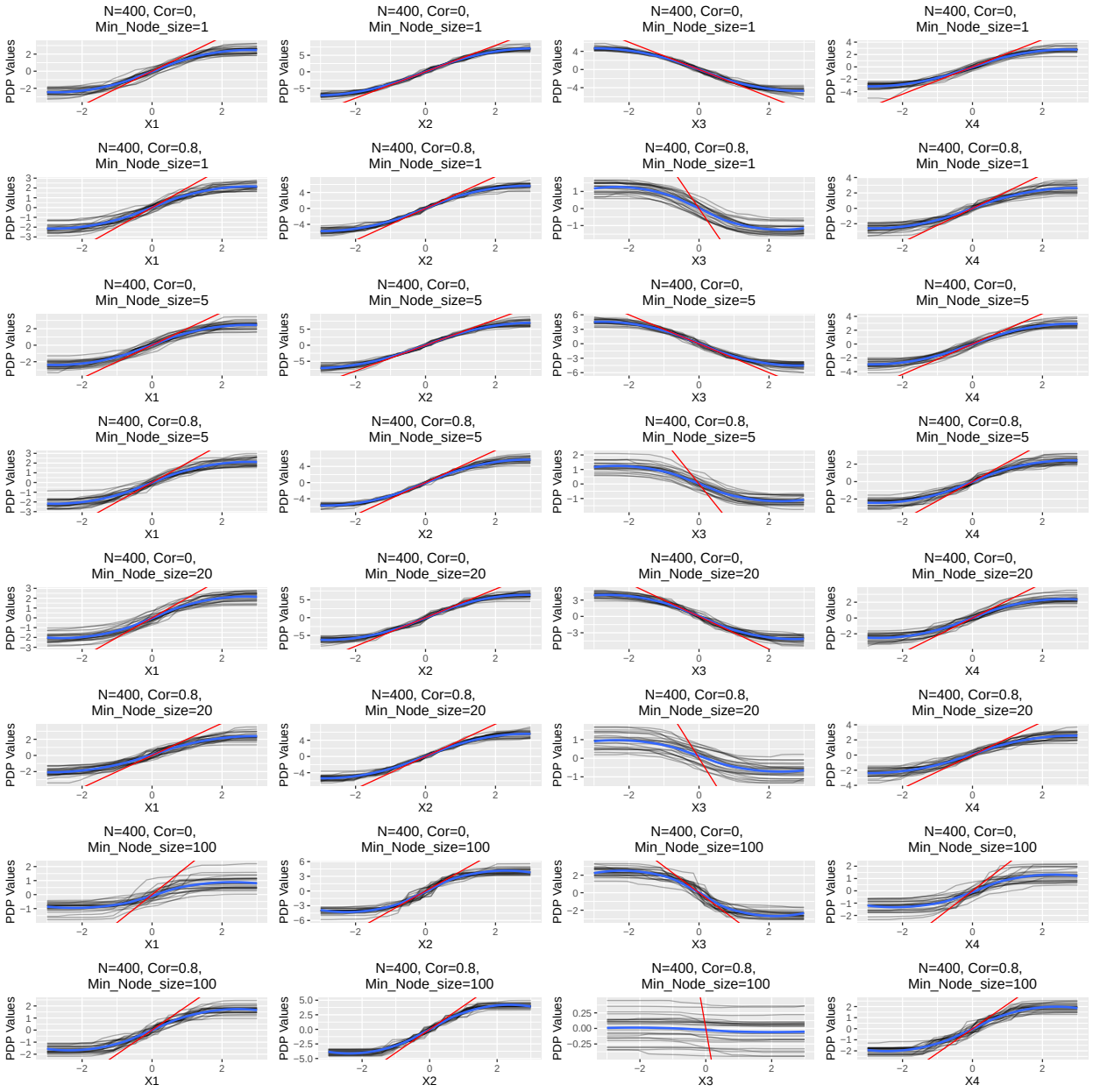



```
#se_plot
```

```
if (pdp) {
  plot_pdps(result)
}
```

```
## `summarise()` has grouped output by 'N', 'cor', 'k', 'num.trees', 'node_size'.
## You can override using the `.groups` argument.
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation ideoms with `aes()`
```


N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
PDP Values
X1
X2
X3
X4

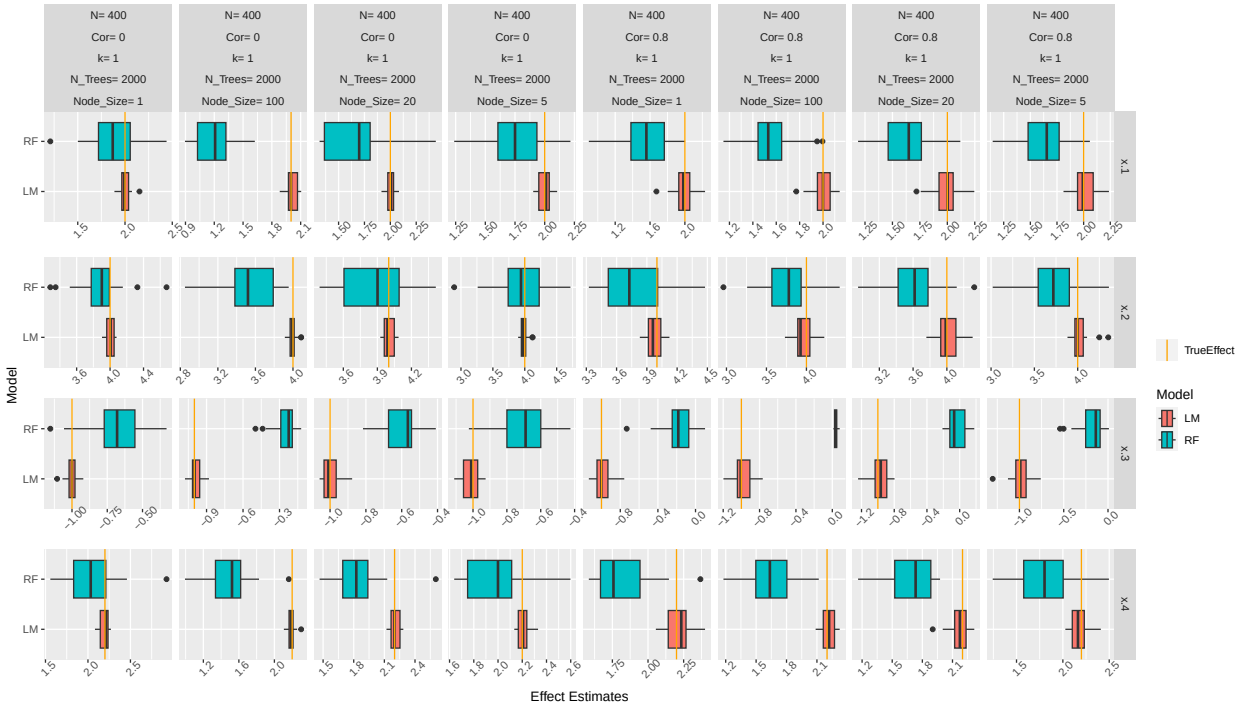
```
n <- c(400) ; num.trees <- 2000 ; repeats <- 25; cor <- c(0, 0.8)
k <- c(1); node_size <- c(1, 5, 20, 100); pdp <- T

formulas <- c("2*x.1+4*x.2-x.3+2.2*x.4")
#parallel::clusterExport(cl = clust, varlist = 'formulas')
scenarios <- data.frame(expand.grid(n, num.trees, formulas, repeats,
                                    cor, k, node_size, pdp))
colnames(scenarios) = c("N", "N_Trees", "Formula", "Repeats",
                        "Correlation", "k", "Node_Size", "pdp")
scenarios$k_pdp <- (scenarios$k == unique(scenarios$k)[1])
scenarios[,"Formula"] <- as.character(scenarios[,"Formula"]) ### Formula became Factor
scenarios <- split(scenarios, seq(nrow(scenarios)))
#system.time(result <- lapply(X = scenarios, FUN = sim_multi))
#Run Simulation
system.time(result <- parLapply(cl = clust,
                                X = scenarios,
                                fun = sim_multi))
```

```
##    user  system elapsed
##    0.01    0.05  946.98
```

```
if (!pdp) {
 print_results(result)
}
effect_plots <- plot_effects(result)
```

```
## `summarise()` has grouped output by 'N', 'cor', 'k', 'num.trees', 'node_size'.
## You can override using the `.groups` argument.
```

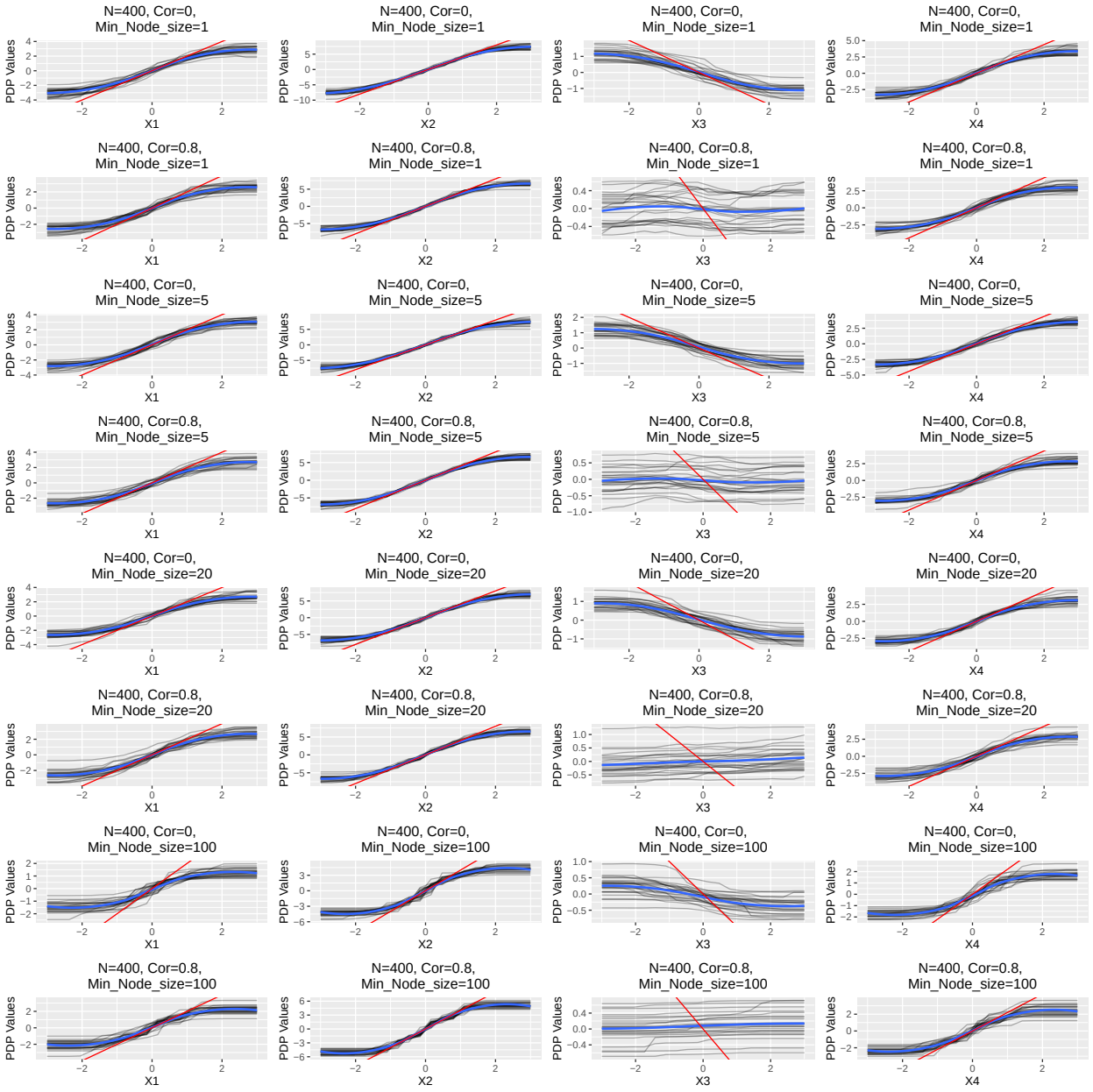
```
#se_plot <- plot_se(result)
effect_plots
```




```
#se_plot
```

```
if (pdp) {
  plot_pdps(result)
}
```

```
## `summarise()` has grouped output by 'N', 'cor', 'k', 'num.trees', 'node_size'.
## You can override using the `.groups` argument.
```


N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
PDP Values
X1
X2
X3
X4

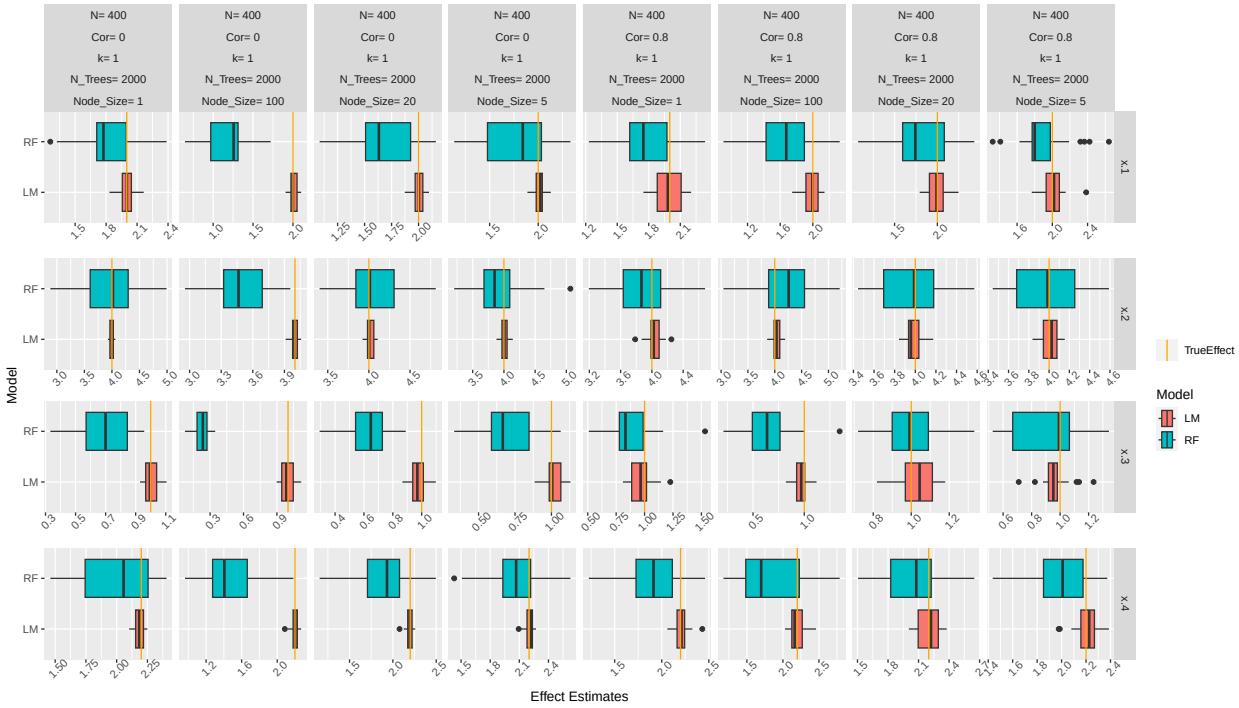
```
n <- c(400) ; num.trees <- 2000 ; repeats <- 25; cor <- c(0, 0.8)
k <- c(1); node_size <- c(1, 5, 20, 100); pdp <- T

formulas <- c("2*x.1+4*x.2+x.3+2.2*x.4")
#parallel::clusterExport(cl = clust, varlist = 'formulas')
scenarios <- data.frame(expand.grid(n, num.trees, formulas, repeats,
                                    cor, k, node_size, pdp))
colnames(scenarios) = c("N", "N_Trees", "Formula", "Repeats",
                        "Correlation", "k", "Node_Size", "pdp")
scenarios$k_pdp <- (scenarios$k == unique(scenarios$k)[1])
scenarios[,"Formula"] <- as.character(scenarios[,"Formula"]) ### Formula became Factor
scenarios <- split(scenarios, seq(nrow(scenarios)))
#system.time(result <- lapply(X = scenarios, FUN = sim_multi))
#Run Simulation
system.time(result <- parLapply(cl = clust,
                                X = scenarios,
                                fun = sim_multi))
```

```
##    user  system elapsed
##    0.00    0.06 1015.17
```

```
if (!pdp) {
 print_results(result)
}
effect_plots <- plot_effects(result)
```

```
## `summarise()` has grouped output by 'N', 'cor', 'k', 'num.trees', 'node_size'.
## You can override using the `.groups` argument.
```

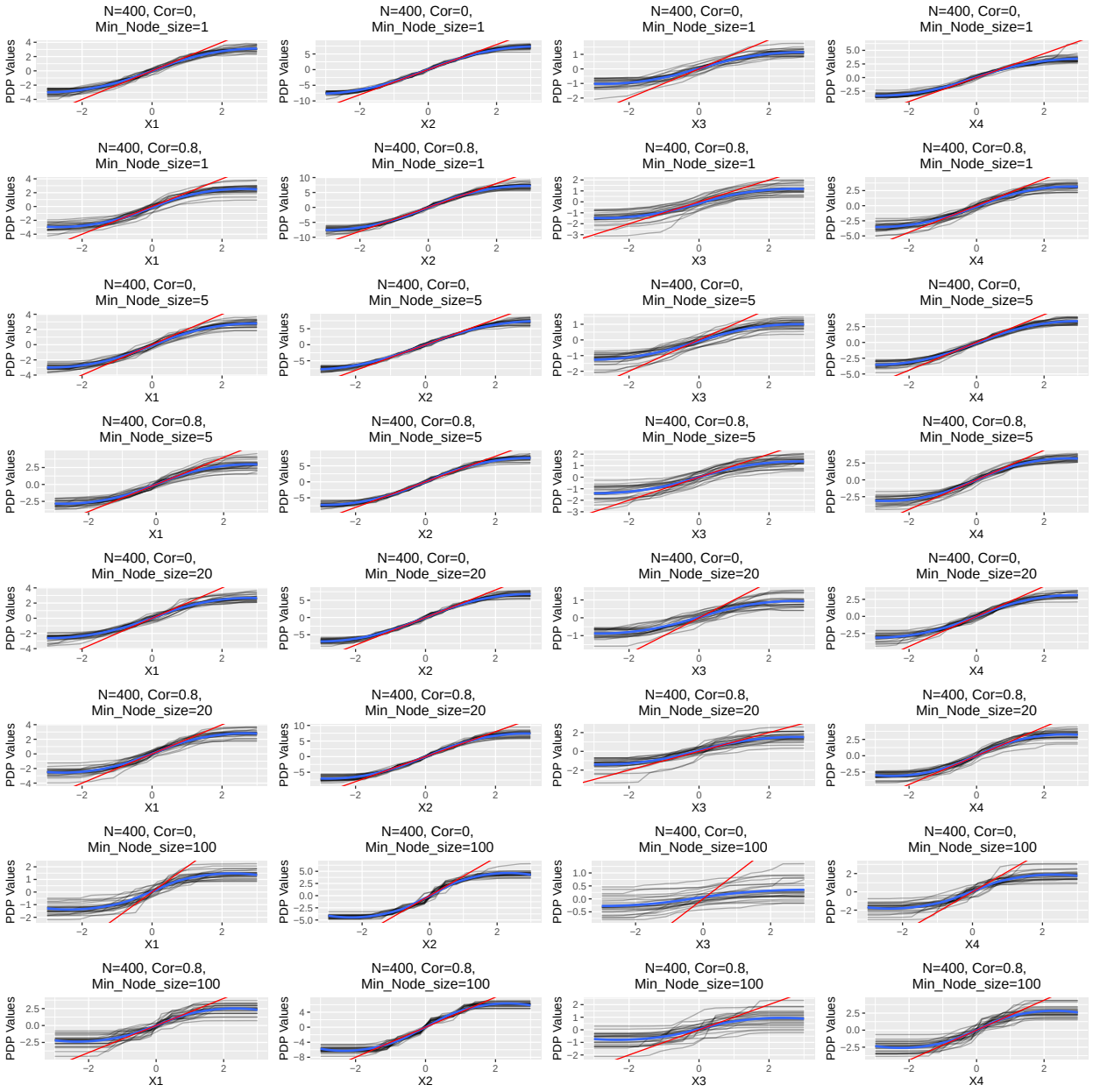
```
#se_plot <- plot_se(result)
effect_plots
```




```
#se_plot
```

```
if (pdp) {
  plot_pdps(result)
}
```

```
## `summarise()` has grouped output by 'N', 'cor', 'k', 'num.trees', 'node_size'.
## You can override using the `.groups` argument.
```


N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
PDP Values
X1
X2
X3
X4

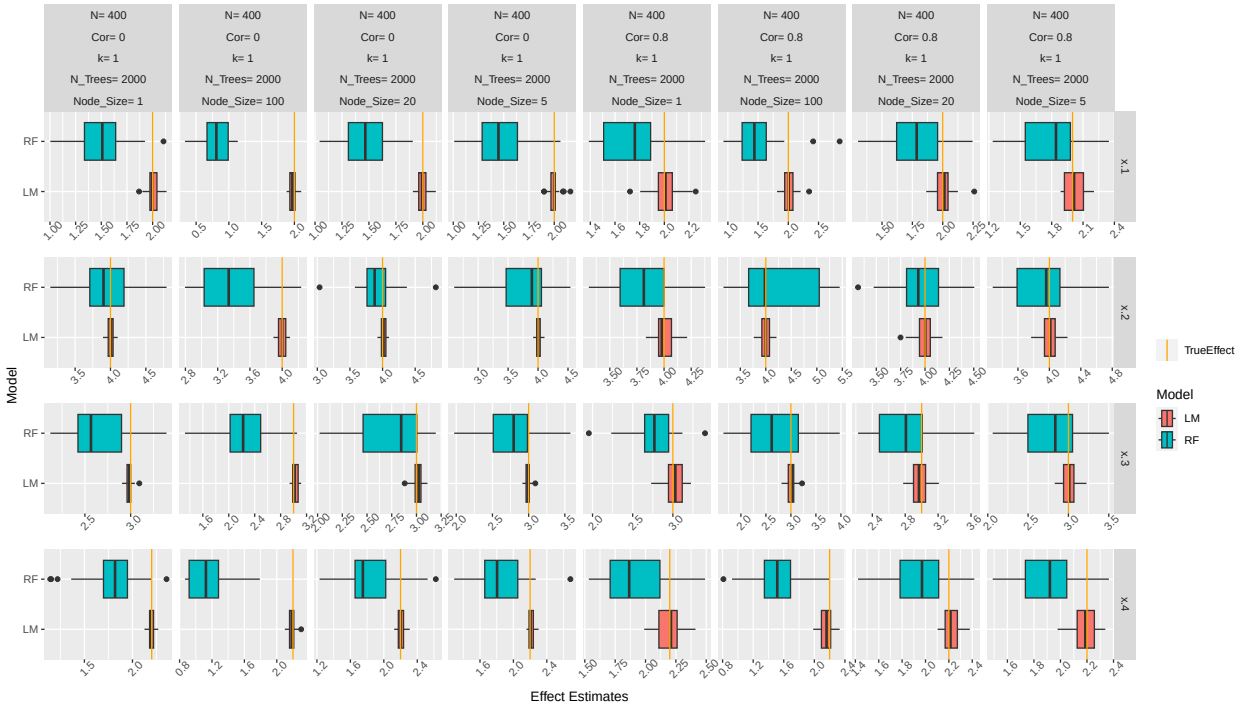
```r
n <- c(400) ; num.trees <- 2000 ; repeats <- 25; cor <- c(0, 0.8)
k <- c(1); node_size <- c(1, 5, 20, 100); pdp <- T

formulas <- c("2*x.1+4*x.2+3*x.3+2.2*x.4")
#parallel::clusterExport(cl = clust, varlist = 'formulas')
scenarios <- data.frame(expand.grid(n, num.trees, formulas, repeats,
                                    cor, k, node_size, pdp))
colnames(scenarios) = c("N", "N_Trees", "Formula", "Repeats",
                        "Correlation", "k", "Node_Size", "pdp")
scenarios$k_pdp <- (scenarios$k == unique(scenarios$k)[1])
scenarios[,"Formula"] <- as.character(scenarios[,"Formula"]) ### Formula became Factor
scenarios <- split(scenarios, seq(nrow(scenarios)))
#system.time(result <- lapply(X = scenarios, FUN = sim_multi))
#Run Simulation
system.time(result <- parLapply(cl = clust,
                                X = scenarios,
                                fun = sim_multi))
```

```
##    user  system elapsed
##    0.00    0.08 1176.20
```

```r
if (!pdp) {
 print_results(result)
}
effect_plots <- plot_effects(result)
```

```
## `summarise()` has grouped output by 'N', 'cor', 'k', 'num.trees', 'node_size'.
## You can override using the `.groups` argument.
```

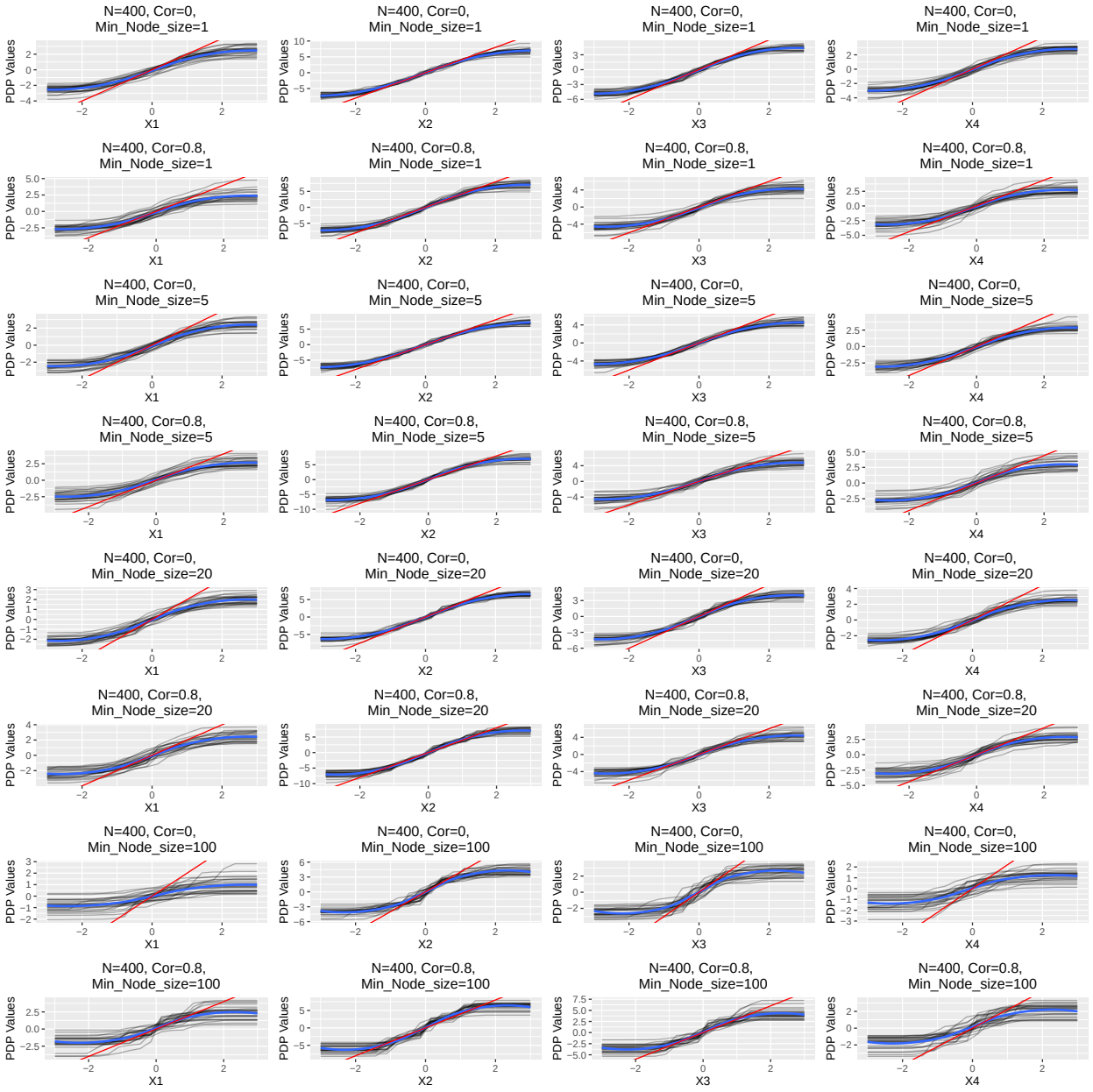
```r
#se_plot <- plot_se(result)
effect_plots
```




```
#se_plot
```

```
if (pdp) {
  plot_pdps(result)
}
```

```
## `summarise()` has grouped output by 'N', 'cor', 'k', 'num.trees', 'node_size'.
## You can override using the `.groups` argument.
```


N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0.8, Min_Node_size=1
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0.8, Min_Node_size=5
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0.8, Min_Node_size=20
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
N=400, Cor=0.8, Min_Node_size=100
PDP Values
X1
X2
X3
X4