Main Effects Linear Regression and Random Forest Diagnostics

Amos Okutse

27 December, 2022

Contents

0.0.1 LOAD DATA

Model diagnostics for the multiple linear regression and the tuned random forest regression models are performed using full and and observed data for the case when $n = 2000$ and the residual standard deviation in the outcome model varied as 1 and 45, respectively to identify potential issues in these models. We investigate potential non-linearity and homoscedasticity (unequal variance) using residuals versus fitted value plots and examine the suitability of the fitted functional forms of the covariates using plots of the predicted values against each of the covariates of interest in the analysis.

0.1 Main effects linear regression

0.2 Plots of residuals versus fitted values

- these plots can be used to identify non-linearity, non-constant variance, and outliers in the data.
- non-linearity occurs when the residuals deviate from the 0 line in some systematic manner. Non-random patterns suggest that the regression function may be non-linear (a pattern of the blue line).
- Non constant variance appears as *fanning* effect where residuals are close to 0 for small predictor values and spread out for larger predictor values or as *funneling* where the residuals are spread out for small predictor values and close to 0 for larger predictor values.
- Outliers stand out from the random pattern of the other residuals.

0.2.1 Case 1 [n = 2000, SD = 1]

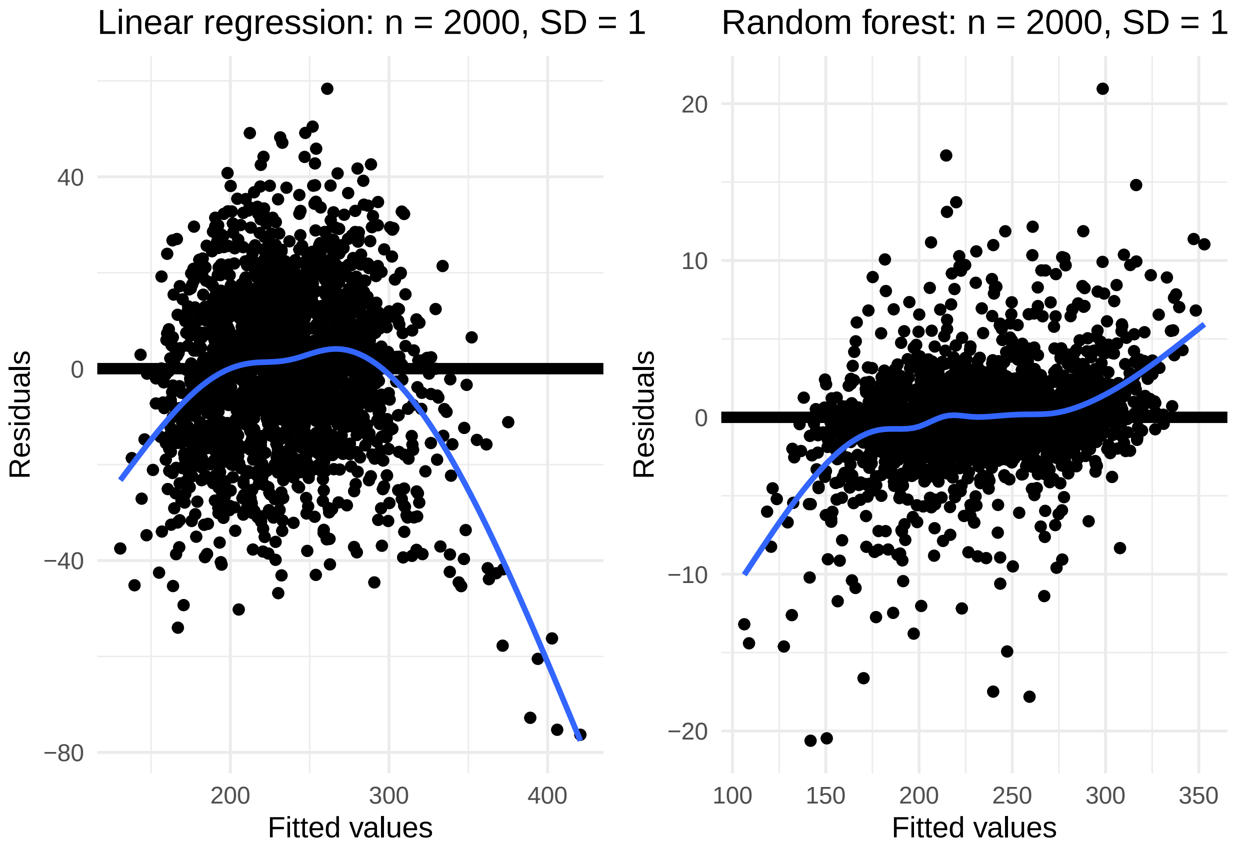

Figure 1: Residuals versus fitted values for linear and random forest regression models with n = 2000, SD = 1.

0.2.2 Case 2 [n = 2000, SD = 45]

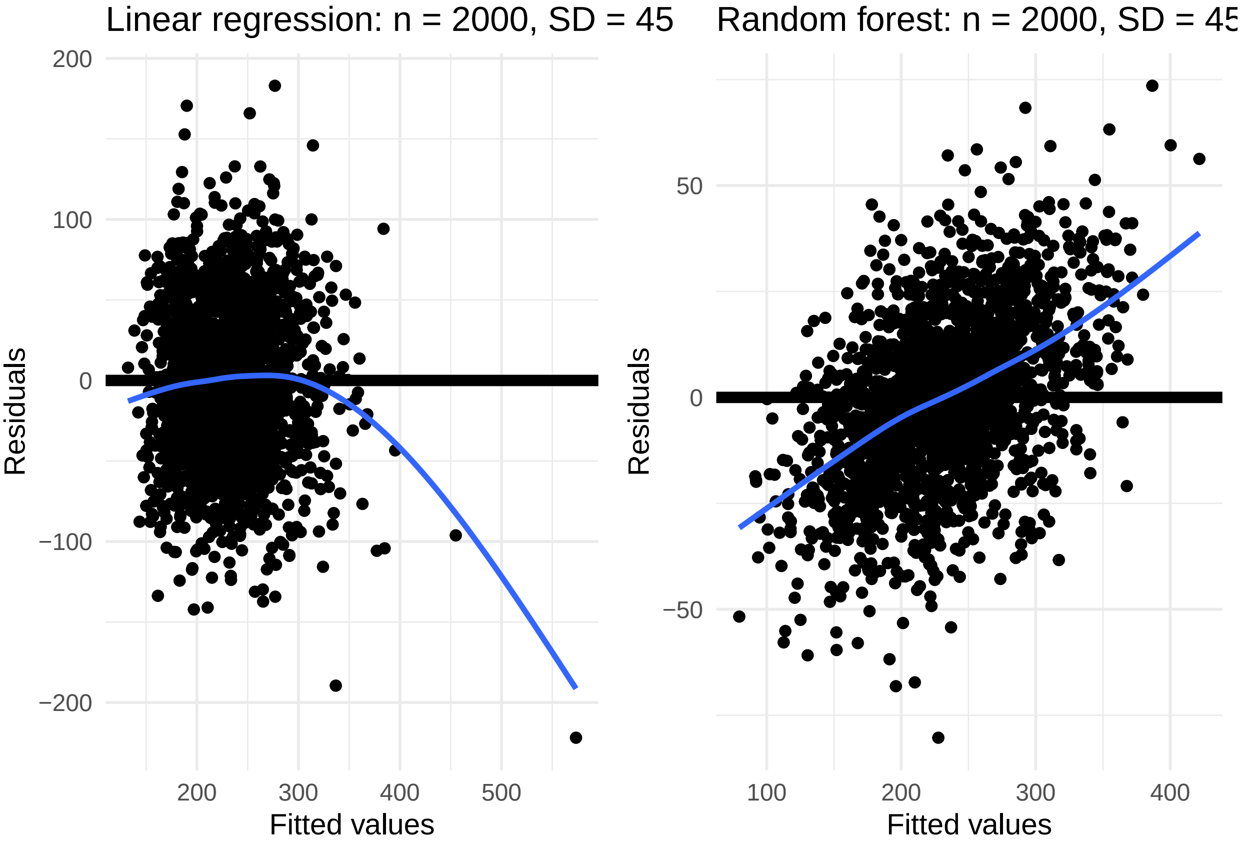

Figure 2: Residuals versus fitted values for linear and random forest regression models with n = 2000, SD = 45.

0.3 Plots of predicted outcome against covariates

0.3.1 Case 1 [n = 2000, SD = 1]

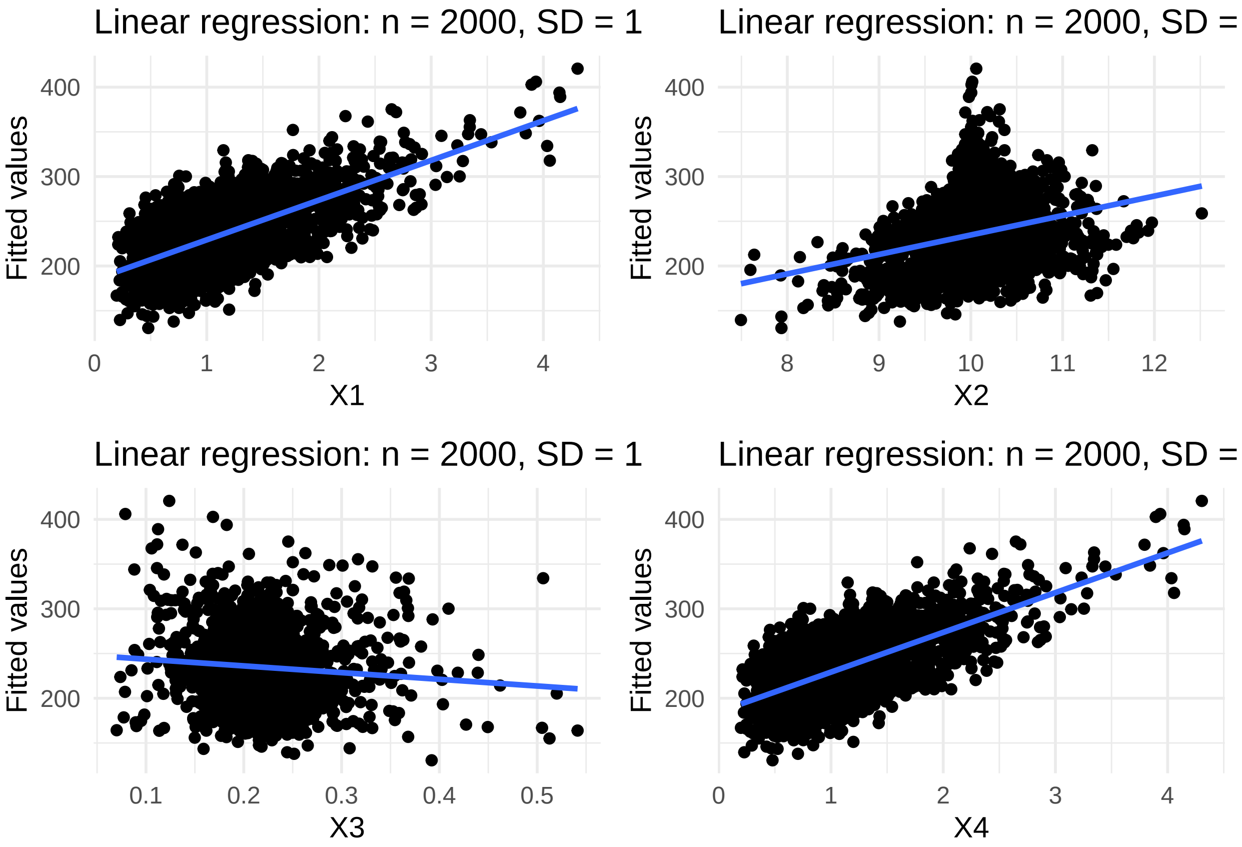

Figure 3: Plots of the covariates against fitted values for the linear and random forest regression models with n = 2000 and SD = 1

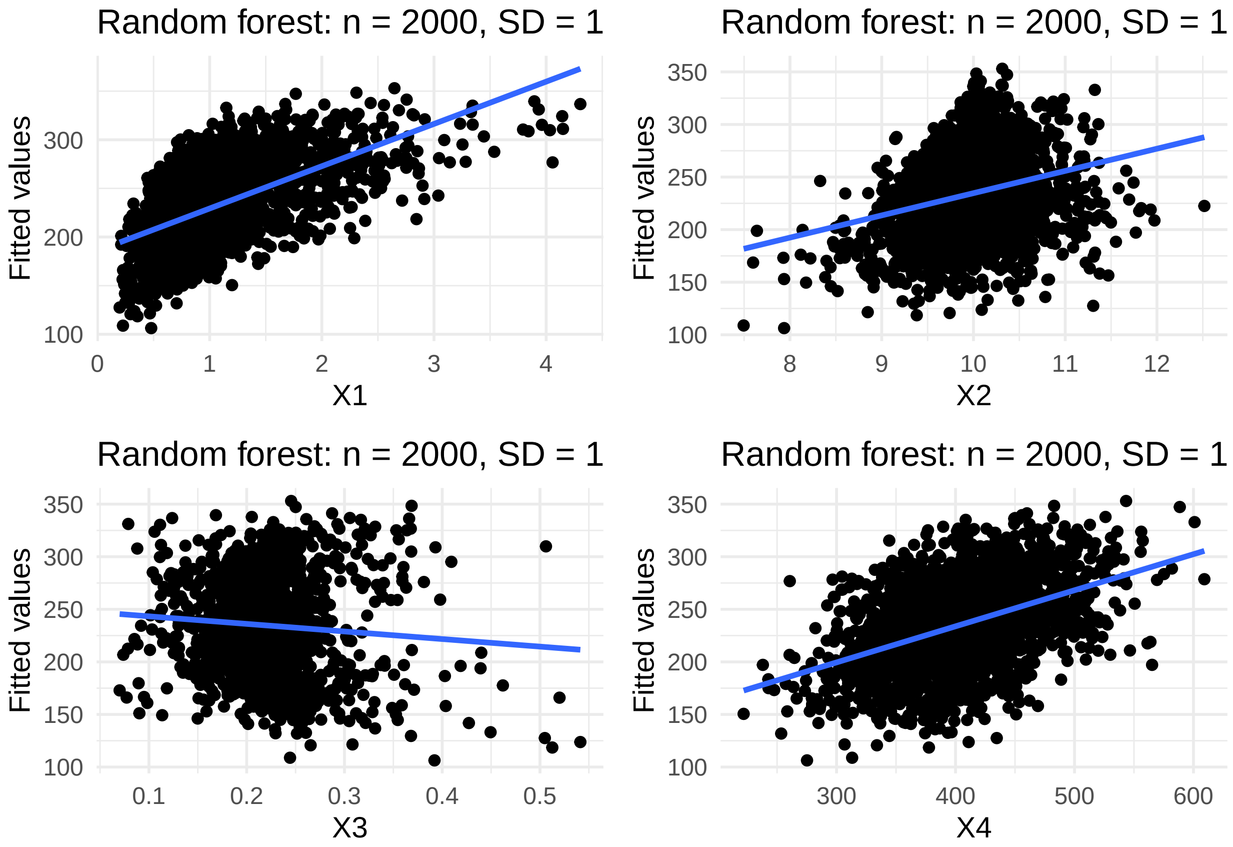

Figure 4: Plots of the covariates against fitted values for the linear and random forest regression models with n = 2000 and SD = 1

0.3.2 Case 2 [n = 2000, SD = 45]

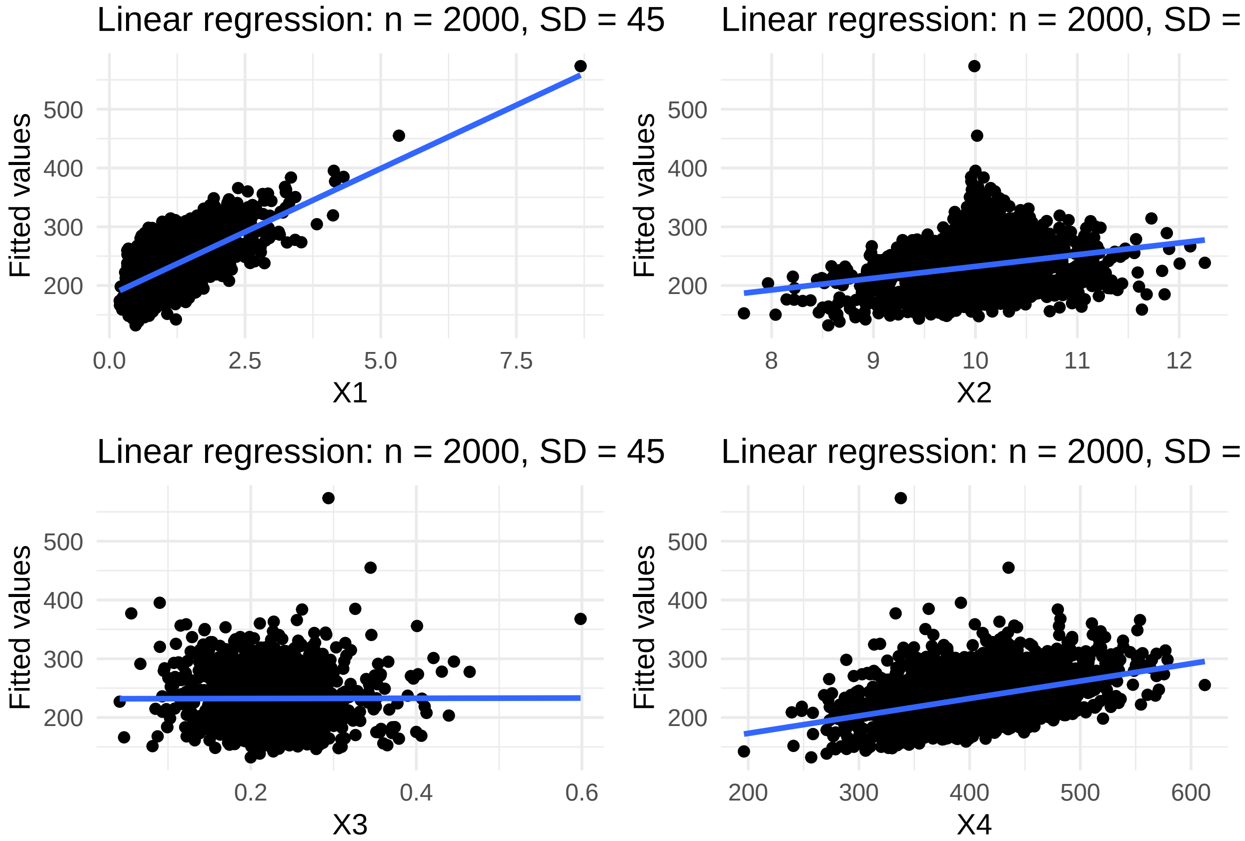

Figure 5: Plot of the covariates against the fitted values ($\hat{y}$) with n = 2000 and SD = 45 for the linear and random forest regression models

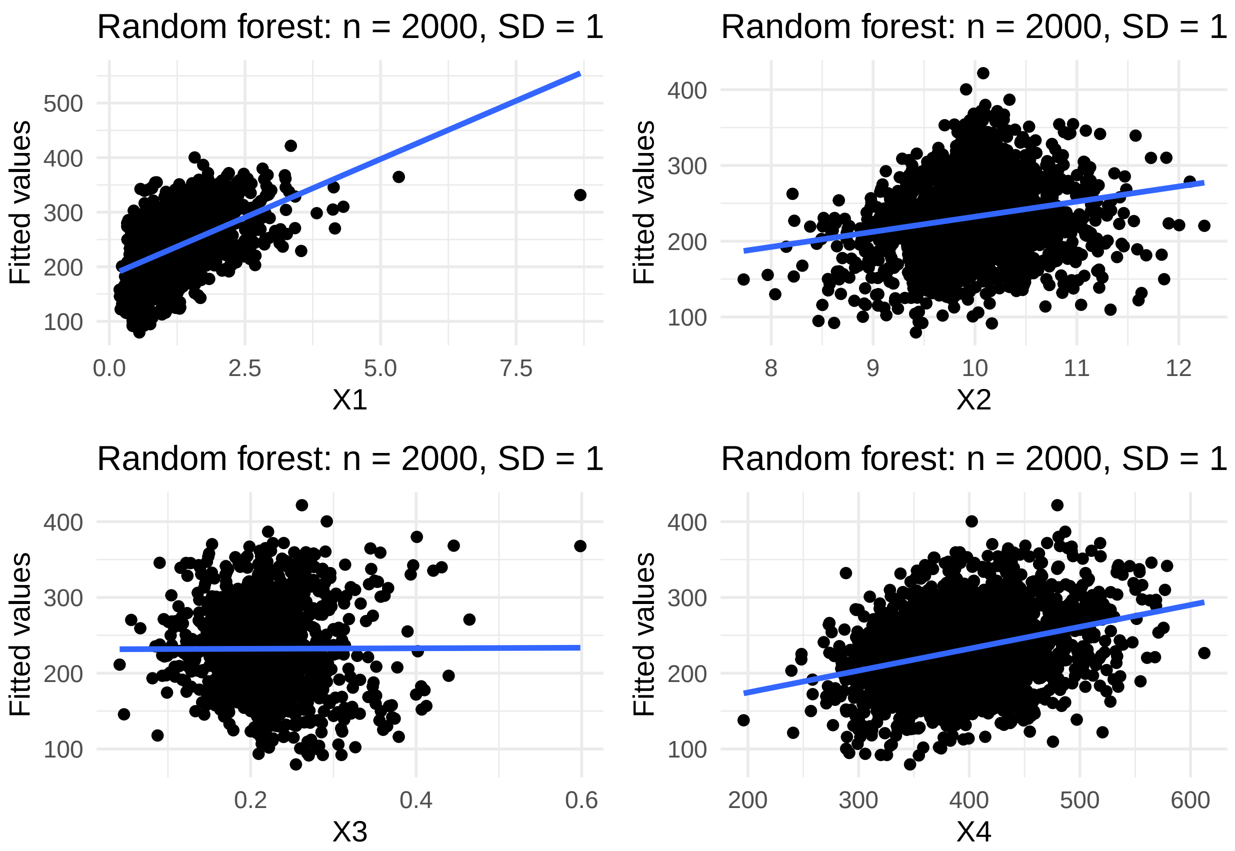

Figure 6: Plot of the covariates against the fitted values ($\hat{y}$) with n = 2000 and SD = 45 for the linear and random forest regression models

```r
knitr::opts_chunk$set(
    echo = FALSE,
    cache = FALSE,
    message = FALSE,
    warning = FALSE,
    fig.align = 'center',
    fig.pos = 'H',
    dpi = 350,
    tidy.opts = list(width.cutoff = 80, tidy = TRUE)
)
# function to install missing packages
ipak <- function(pkg){
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE, repos='http://cran.rstudio.com/')
  sapply(pkg, require, character.only = TRUE)
}
packages =c( "tidyverse","knitr", "kableExtra","skimr", "MatchIt", "RItools","optmatch",
 ↪  "ggplot2", "tufte", "tufterhandout", "plotly", "snowfall", "rstan", "gridExtra",
 ↪  "knitr", "gtsummary", "data.table", "GGally", "MASS", "broom", "boot", "foreach",
 ↪  "doParallel", "glmnet", "tidymodels" , "usemodels", "magrittr", "modelr")
ipak(packages)
```

```r
rm(list = ls())
load("G:\\Shared drives\\amos\\ThesisResults\\data\\df_one.RData")
load("G:\\Shared drives\\amos\\ThesisResults\\data\\df_two.RData")
load("G:\\Shared drives\\amos\\ThesisResults\\data\\df_three.RData")
load("G:\\Shared drives\\amos\\ThesisResults\\data\\df_four.RData")
## linear regression
df_lma = df_three
lma <- lm(formula = y ~ A + x1 + x2 + x3 + x4, data = df_lma)
## add the residuals and predicted values to the data set
df_lma = broom::augment(lma, df_lma)

## random forest
df_rfa <- df_three
rfa <- rand_forest(trees = 1000, mtry = 5, min_n = 4) %>%
  set_mode("regression") %>%
  set_engine("ranger") %>%
  fit(formula = y ~ A + x1 + x2 + x3 + x4, data = df_rfa)

## add predicted random forest results to the data frame
df_rfa <- broom::augment(rfa, df_rfa)

# linear regression
a1 <- ggplot(data = df_lma, aes(.fitted, .resid))+
  geom_ref_line(h = 0, colour = "black") +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "Linear regression: n = 2000, SD = 1", x = "Fitted values", y =
   ↪  "Residuals")+
  theme_minimal()

# random forest
a2 <- ggplot(data = df_rfa, aes(.pred, .resid))+
  geom_ref_line(h = 0, colour = "black") +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "Random forest: n = 2000, SD = 1", x = "Fitted values", y = "Residuals")+
  theme_minimal()

gridExtra::grid.arrange(a1, a2, ncol = 2)
## linear regression
df_lmb = df_four
lmb <- lm(formula = y ~ A + x1 + x2 + x3 + x4, data = df_lmb)
## add the residuals and predicted values to the data set
df_lmb = broom::augment(lmb, df_lmb)

## random forest
df_rfb <- df_four
rfb <- rand_forest(trees = 1000, mtry = 5, min_n = 4) %>%
  set_mode("regression") %>%
  set_engine("ranger") %>%
  fit(formula = y ~ A + x1 + x2 + x3 + x4, data = df_rfb)

## add predicted random forest results to the data frame
```

```
df_rfb <- broom::augment(rfb, df_rfb)

# linear regression
b1 <- ggplot(data = df_lmb, aes(.fitted, .resid))+
  geom_ref_line(h = 0, colour = "black") +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "Linear regression: n = 2000, SD = 45", x = "Fitted values", y =
   ↪  "Residuals")+
  theme_minimal()

# random forest
b2 <- ggplot(data = df_rfb, aes(.pred, .resid))+
  geom_ref_line(h = 0, colour = "black") +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "Random forest: n = 2000, SD = 45", x = "Fitted values", y = "Residuals")+
  theme_minimal()

gridExtra::grid.arrange(b1, b2, ncol = 2)

## Linear regression
one <- ggplot(df_lma, aes(x1, .fitted))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Linear regression: n = 2000, SD = 1", x = "X1", y = "Fitted values")+
  theme_minimal()
two <- ggplot(df_lma, aes(x2, .fitted))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Linear regression: n = 2000, SD = 1", x = "X2", y = "Fitted values")+
  theme_minimal()
three <- ggplot(df_lma, aes(x3, .fitted))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Linear regression: n = 2000, SD = 1", x = "X3", y = "Fitted values")+
  theme_minimal()
four <- ggplot(df_lma, aes(x1, .fitted))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Linear regression: n = 2000, SD = 1", x = "X4", y = "Fitted values")+
  theme_minimal()
gridExtra::grid.arrange(one, two, three, four, ncol = 2)

## Random forest

rfone <- ggplot(df_rfa, aes(x1, .pred))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Random forest: n = 2000, SD = 1", x = "X1", y = "Fitted values")+
  theme_minimal()
rftwo <- ggplot(df_rfa, aes(x2, .pred))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Random forest: n = 2000, SD = 1", x = "X2", y = "Fitted values")+
  theme_minimal()
rfthree <- ggplot(df_rfa, aes(x3, .pred))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Random forest: n = 2000, SD = 1", x = "X3", y = "Fitted values")+
  theme_minimal()
rffour <- ggplot(df_rfa, aes(x4, .pred))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Random forest: n = 2000, SD = 1", x = "X4", y = "Fitted values")+
  theme_minimal()
gridExtra::grid.arrange(rfone, rftwo, rfthree, rffour, ncol = 2)
## Linear regression
oneb <- ggplot(df_lmb, aes(x1, .fitted))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Linear regression: n = 2000, SD = 45", x = "X1", y = "Fitted values")+
  theme_minimal()
twob <- ggplot(df_lmb, aes(x2, .fitted))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Linear regression: n = 2000, SD = 45", x = "X2", y = "Fitted values")+
  theme_minimal()
threeb <- ggplot(df_lmb, aes(x3, .fitted))+geom_point() + geom_smooth(se = FALSE, method
 ↪  = "lm") +
  labs(title = "Linear regression: n = 2000, SD = 45", x = "X3", y = "Fitted values")+
  theme_minimal()
fourb <- ggplot(df_lmb, aes(x4, .fitted))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Linear regression: n = 2000, SD = 45", x = "X4", y = "Fitted values")+
  theme_minimal()
gridExtra::grid.arrange(oneb, twob, threeb, fourb, ncol = 2)

## Random forest

rfoneb <- ggplot(df_rfb, aes(x1, .pred))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Random forest: n = 2000, SD = 1", x = "X1", y = "Fitted values")+
  theme_minimal()
rftwob <- ggplot(df_rfb, aes(x2, .pred))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Random forest: n = 2000, SD = 1", x = "X2", y = "Fitted values")+
  theme_minimal()
rfthreeb <- ggplot(df_rfb, aes(x3, .pred))+geom_point() + geom_smooth(se = FALSE, method
 ↪  = "lm") +
  labs(title = "Random forest: n = 2000, SD = 1", x = "X3", y = "Fitted values")+
  theme_minimal()
rffourb <- ggplot(df_rfb, aes(x4, .pred))+geom_point() + geom_smooth(se = FALSE, method =
 ↪  "lm") +
  labs(title = "Random forest: n = 2000, SD = 1", x = "X4", y = "Fitted values")+
  theme_minimal()
gridExtra::grid.arrange(rfoneb, rftwob, rfthreeb, rffourb, ncol = 2)
```