Exercise3

Miriam Fischer

12 10 2020

Creating Dataset

We first create our random variables, and the data of Y1 and Y2:

```
set.seed(40)
Z1 <- rnorm(500, 0, 1)
Z2 <- rnorm(500, 0, 1)
Z3 <- rnorm(500, 0, 1)

Y1 <- 1 + Z1
Y2 <- 5 + 2 * Z1 + Z2
```

We then set the function when Y2 should be missing:

```
a <- 2
b <- 0
v <- a * (Y1 - 1) + b * (Y2 - 5) + Z3
```

3a: Impose missingness on Y2

We impose missingness on Y2 for Exercise 3a

```
ind_a <- which(v < 0)
Y2_MAR_obs <- Y2[-ind_a]
Y2_MAR_mis <- Y2[ind_a]
```

Note that this is MAR, missing at random. The reason is that with a=2 and b=0, the function which determines missingness on Y2 is

$$2(Y_1 - 1) + Z_3 < 0$$

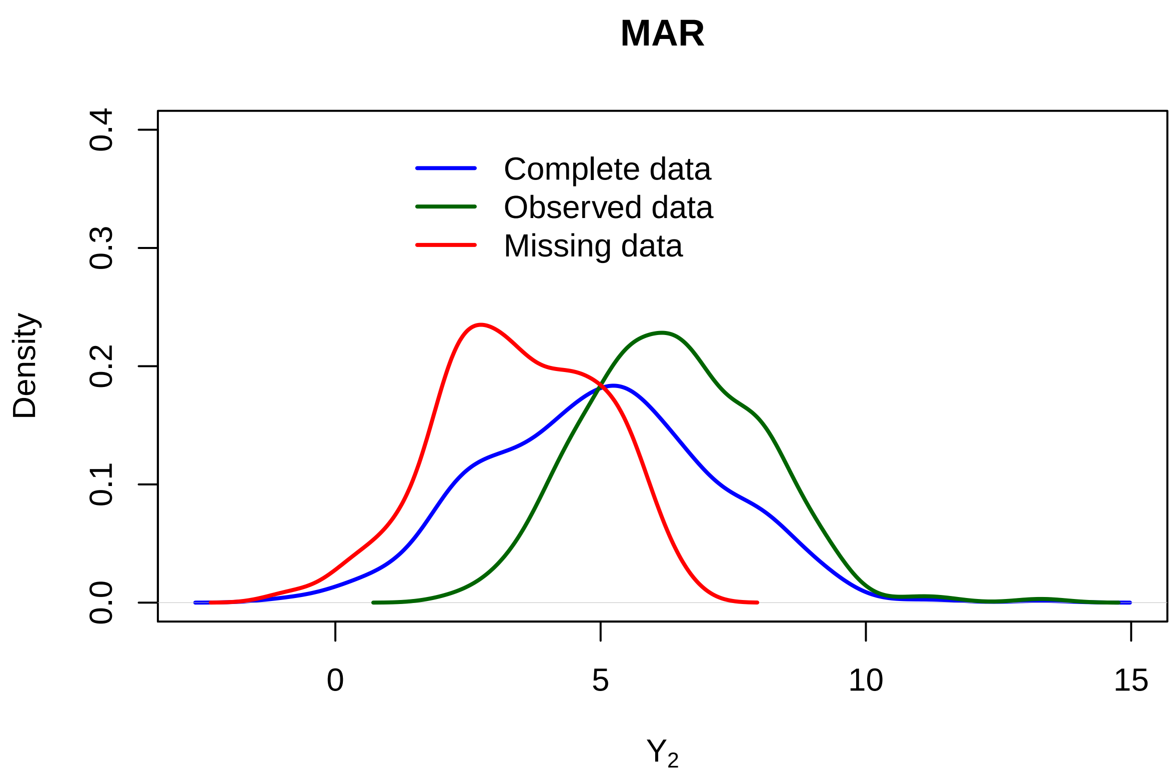
This function clearly depends on Y1, but not on Y2. It also depends on a random variable Z3 (which is without connection to Y1 and Y2). We note that Y1 is fully observed. Thus, missingness of Y2 depends on the value of Y1, which we have available in our dataset, making it MAR. Most importantly, missingness does not depend on Y2 itself. We further argue that the missingness depending on Z3 is not seen as missingness depending on an un-observed variable (in which case we would have MNAR), as Z3 is solely random and thus the missingness conditioned on Z3 is random.

3a: Plots

We now plot the missing data for 3a.

```
plot(density(Y2), lwd = 2, col = "blue", xlab = expression(Y[2]), main = "MAR", ylim = c(0,
    0.4))
lines(density(Y2_MAR_obs), lwd = 2, col = "darkgreen")
lines(density(Y2_MAR_mis), lwd = 2, col = "red")
```

```
legend(1, 0.4, legend = c("Complete data", "Observed data", "Missing data"), col = c("blue",
    "darkgreen", "red"), lty = c(1, 1, 1), lwd = c(2, 2, 2), bty = "n")
```



As we can see, the plots of the observed and missing data are not the same. However, they are shifted: if we ignore the fact that their respective mean and mode are at different values of Y2, the shape of the plot is similar. With that I mean that if we would shift the curve of the missing data to the right, we would roughly get the curve of the observed data. This is commonly seen for MAR data: Unconditioned on observed variables, missingness looks MNAR, but if we condition on observed variables, the missingness follows the same patterns.

3b: Regression imputation

We perform linear regression and assume that the necessary requirements for linear regression are fulfilled. We want to estimate the effect of Y1 on Y2.

```
Y2_MAR_na <- ifelse(v < 0, NA, Y2)
data_b <- data.frame(Y1_reg_b = Y1, Y2_reg_b = Y2_MAR_na)
reg_fit_b <- lm(Y2_reg_b ~ Y1_reg_b, data <- data_b)
```

The regression is:

```
reg_fit_b$coefficients
```

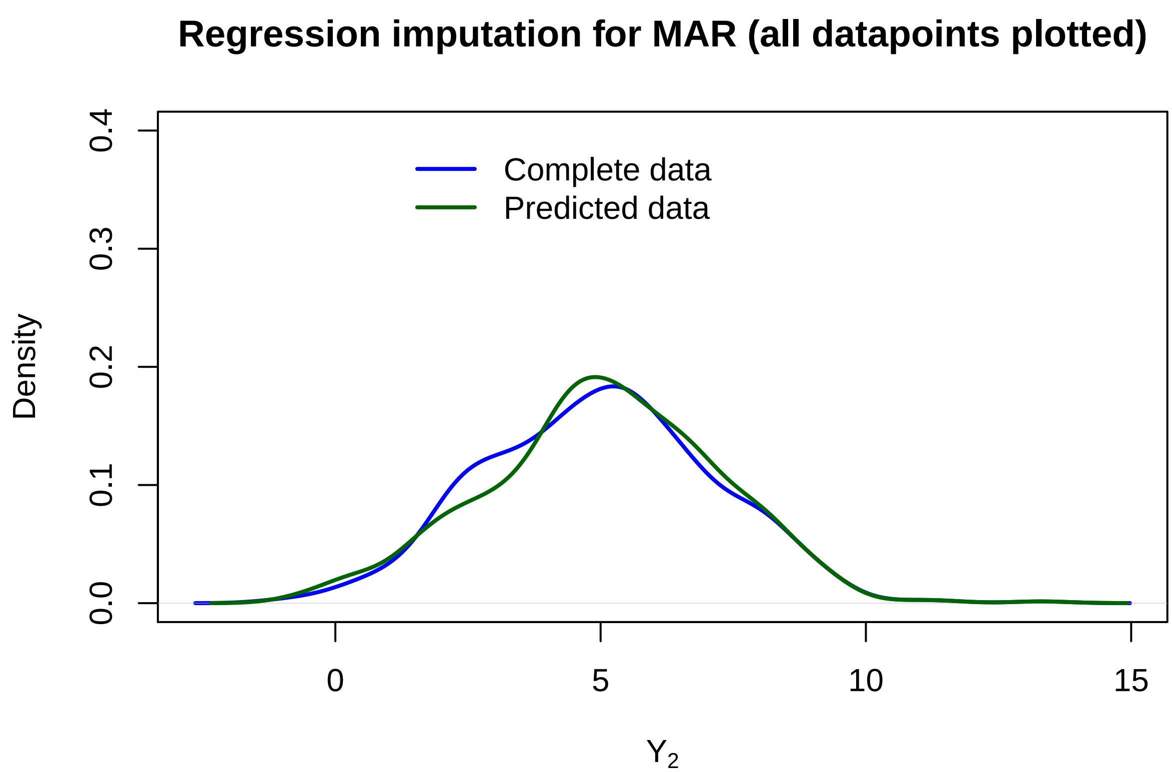
```
## (Intercept)    Y1_reg_b
##    2.989912    1.942356
```

Our predicted dataset will have the known values of Y2 for the values where we did not impose missingness, and will use the values predicted with our regression for the values which are missing. Further, as we use stochastic imputation, we include a random noise.

```
predicted_b <- predict(reg_fit_b, newdata = data_b) + rnorm(nrow(data_b), 0, sigma(reg_fit_b))
Y2_MAR_pre <- ifelse(is.na(data_b$Y2_reg_b), predicted_b, Y2)
```

Plots

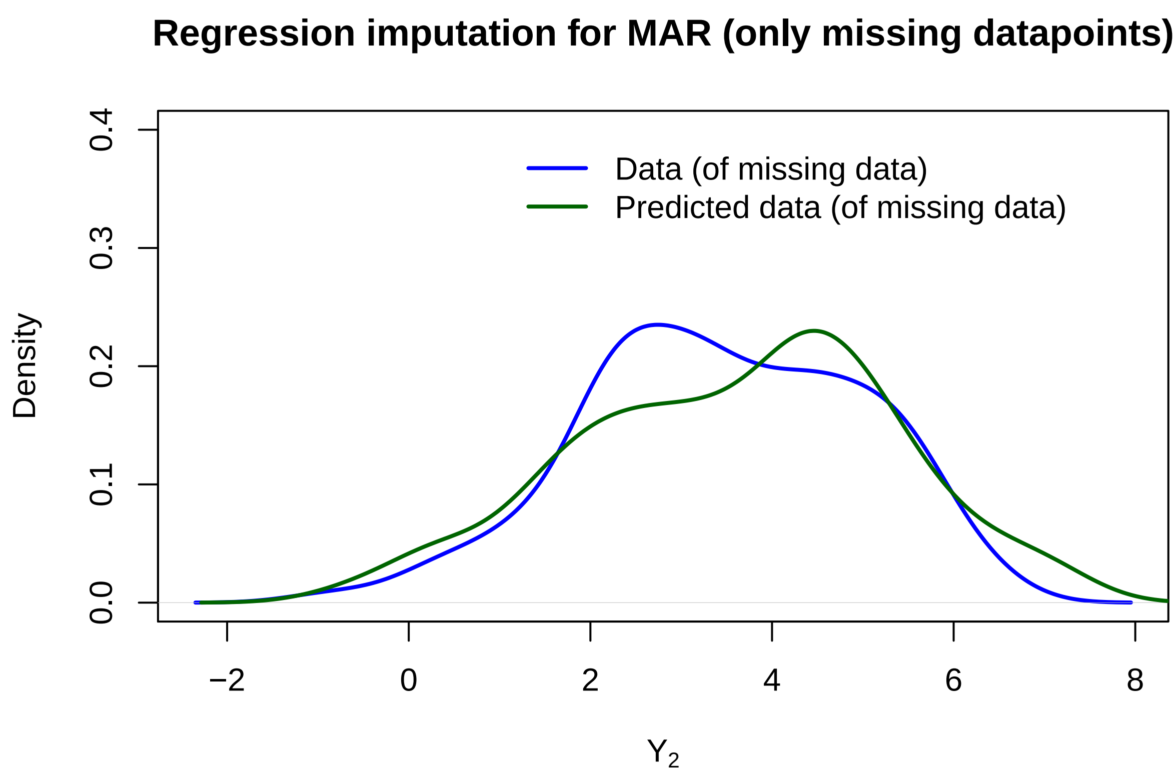
```
# plot
plot.new()
frame()
plot(density(Y2), lwd = 2, col = "blue", xlab = expression(Y[2]), main = "Regression imputation for MAR
    ylim = c(0, 0.4))
lines(density(Y2_MAR_pre), lwd = 2, col = "darkgreen")
legend(1, 0.4, legend = c("Complete data", "Predicted data"), col = c("blue", "darkgreen"),
    lty = c(1, 1), lwd = c(2, 2), bty = "n")
```



We note that for datapoints where Y2 is not missing, the "predicted" value and the correct value are equal (as, looking at our code, we our predicted data does not predict the values for given datapoints, but just takes their known values). We therefore use a second graph, which only has the data which we actually predict.

```
plot.new()
frame()
Y2_MAR_onlypre <- Y2_MAR_pre[ind_a]
plot(density(Y2_MAR_mis), lwd = 2, col = "blue", xlab = expression(Y[2]), main = "Regression imputation
    ylim = c(0, 0.4))
lines(density(Y2_MAR_onlypre), lwd = 2, col = "darkgreen")
```

```
legend(1, 0.4, legend = c("Data (of missing data)", "Predicted data (of missing data)"),
    col = c("blue", "darkgreen"), lty = c(1, 1), lwd = c(2, 2), bty = "n")
```



3c: Impose missingness

We change the parameters when Y2 should be missing:

```
a <- 0
b <- 2
vv <- a * (Y1 - 1) + b * (Y2 - 5) + Z3
```

We impose missingness on Y2 for Exercise 3c

```
ind_c <- which(vv < 0)
Y2_MNAR_obs <- Y2[-ind_c]
Y2_MNAR_mis <- Y2[ind_c]
```

Note that this is MNAR, missing not at random. The reason is that with a=0 and b=2, the function which determines missingness on Y2 is

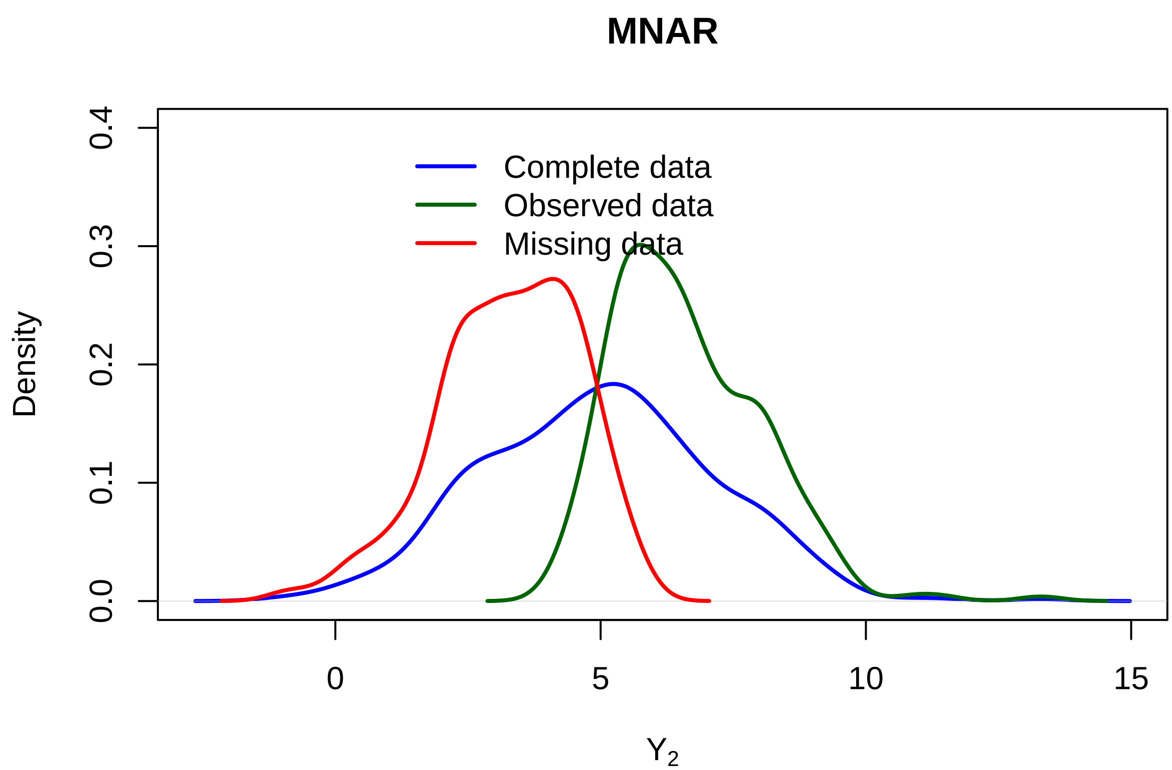
$$2(Y_2 - 5) + Z_3 < 0$$

This function clearly depends on Y2, but not on Y1. Thus, must crucially, missingness of Y2 depends on the value of Y2 itself, making it MNAR.

3c: Plots

We now plot the missing data for 3c.

```r
plot(density(Y2), lwd = 2, col = "blue", xlab = expression(Y[2]), main = "MNAR",
    ylim = c(0, 0.4))
lines(density(Y2_MNAR_obs), lwd = 2, col = "darkgreen")
lines(density(Y2_MNAR_mis), lwd = 2, col = "red")
legend(1, 0.4, legend = c("Complete data", "Observed data", "Missing data"), col = c("blue",
    "darkgreen", "red"), lty = c(1, 1, 1), lwd = c(2, 2, 2), bty = "n")
```



Most importantly, the missing data and the observed data are very different to each other. This will prove to be different to predict the missing data later.

3d: Regression imputation

We again use linear regression and assume the underlying assumptions to be met.

```r
Y2_MNAR_na <- ifelse(vv < 0, NA, Y2)
data_d <- data.frame(Y1_reg_d = Y1, Y2_reg_d = Y2_MNAR_na)
reg_fit_d <- lm(Y2_reg_d ~ Y1_reg_d, data <- data_d)
```

The regression is:

```r
reg_fit_d$coefficients
```

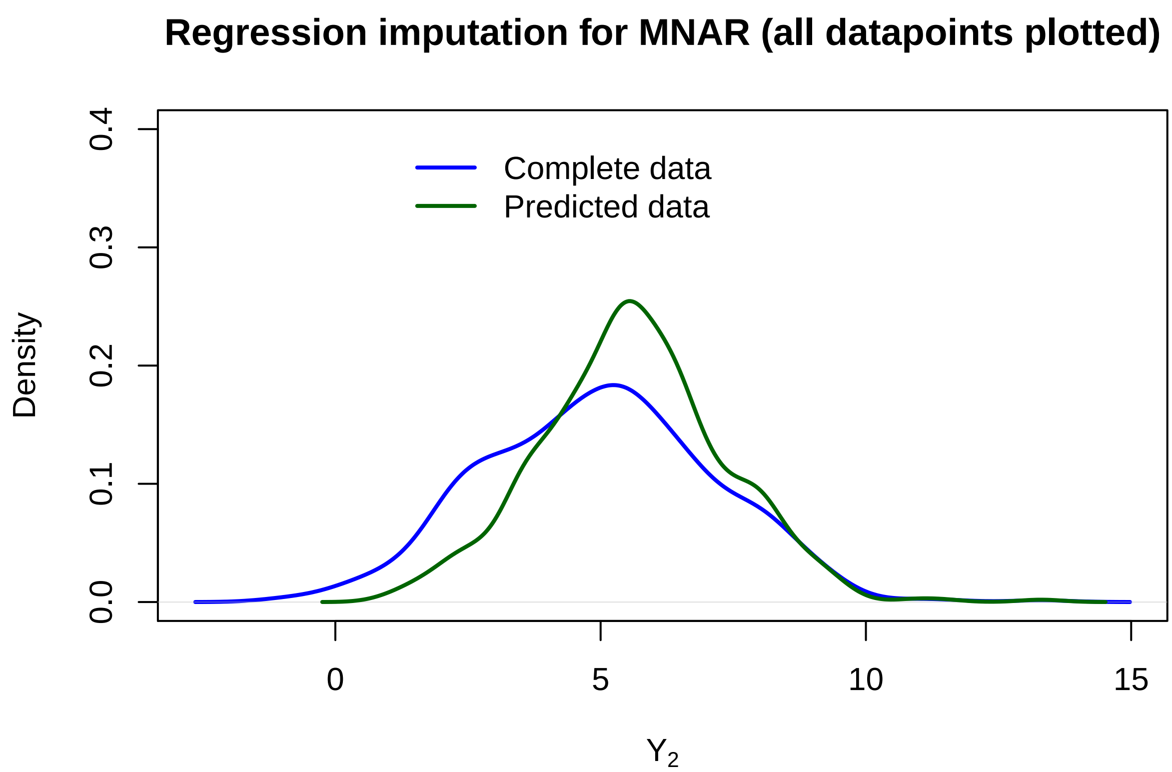
```
## (Intercept)    Y1_reg_d
##    4.018839    1.500059
```

Our predicted dataset will have the known values of Y2 for the values where we did not impose missingness, and will use the values predicted with our regression for the values which are missing.

```r
predicted_d <- predict(reg_fit_d, newdata = data_d) + rnorm(nrow(data_d), 0, sigma(reg_fit_d))
Y2_MNAR_pre <- ifelse(is.na(data_d$Y2_reg_d), predicted_d, Y2)
```

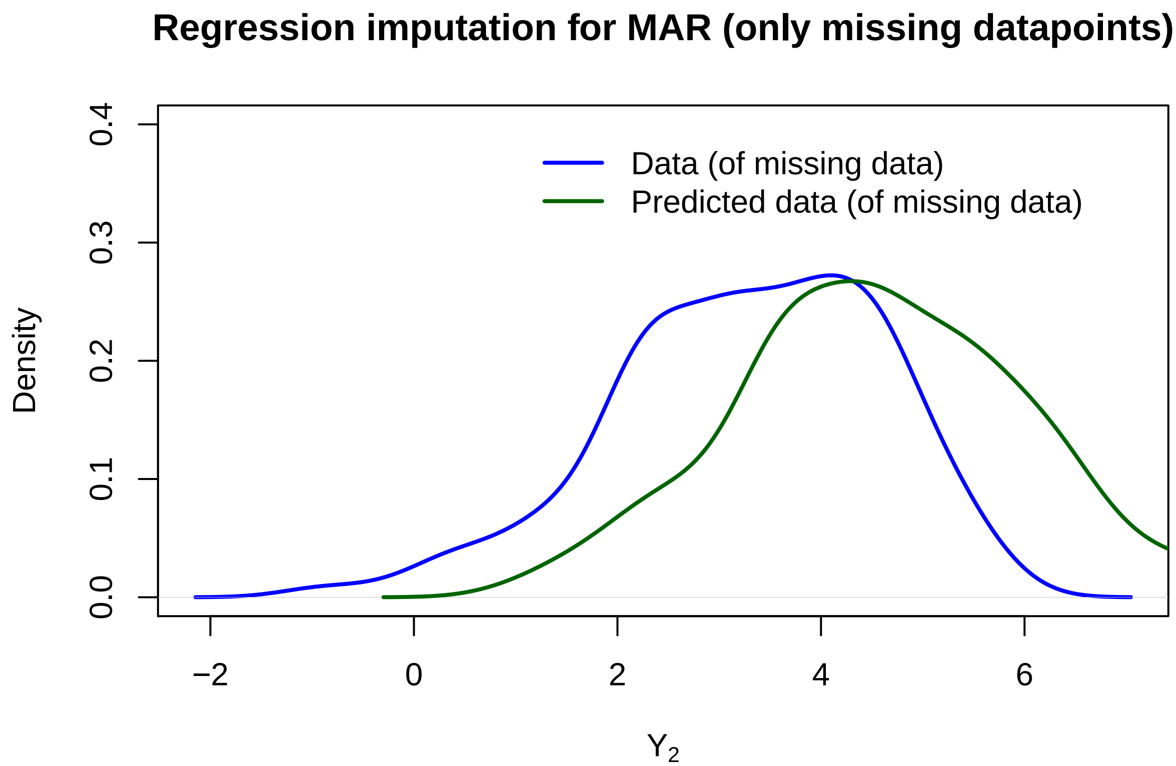
Plots

```r
# plot
plot.new()
frame()
plot(density(Y2), lwd = 2, col = "blue", xlab = expression(Y[2]), main = "Regression imputation for MNAR
    ylim = c(0, 0.4))
lines(density(Y2_MNAR_pre), lwd = 2, col = "darkgreen")
legend(1, 0.4, legend = c("Complete data", "Predicted data"), col = c("blue", "darkgreen"),
    lty = c(1, 1), lwd = c(2, 2), bty = "n")
```



We note that for datapoints where Y2 is not missing, the "predicted" value and the correct value are equal (as, looking at our code, we our predicted data does not predict the values for given datapoints, but just takes their known values). We therefore use a second graph, which only has the data which we actually predict.

```r
plot.new()
frame()
Y2_MNAR_onlypre <- Y2_MNAR_pre[ind_c]
plot(density(Y2_MNAR_mis), lwd = 2, col = "blue", xlab = expression(Y[2]), main = "Regression imputatio
    ylim = c(0, 0.4))
lines(density(Y2_MNAR_onlypre), lwd = 2, col = "darkgreen")
legend(1, 0.4, legend = c("Data (of missing data)", "Predicted data (of missing data)"),
    col = c("blue", "darkgreen"), lty = c(1, 1), lwd = c(2, 2), bty = "n")
```



Compared to the plots of the prediction for MAR, the predictions we are doing here are much worse: for MAR, our predictions were somewhat accurate (or at least reasonable), but the predictions we have here are far from being accurate. Thus, for MNAR, our prediction gives very poor results in predicting Y2. This is not a surprise: Our regression tries to predict Y2 based on Y1, but for our model, missingness on Y2 is not based on Y1, but on Y2. It is no surprise that the prediction is not good, as it tries to find a causality between two variables which do not have a causal connection to each other! More badly, for the regression the noise we add is probably doing better in predicting Y2 than Y1 itself (I haven't checked this though, but I find it likely to be the case)!