Incomplete Data Analysis

Assignment 1

Callum Abbott

Code

The Rmd file used to create this pdf can be found in the following repo https://github.com/c-abbott/ida under the sub-directory assignment-1/ida-assignment-1.Rmd

Question 1

Let us suppose that on a (hypothetical) survey there is a question about alcohol consumption and that the variable ALQ records the respondent's answer to the following specific question: "In the past year, have you had at least 12 drinks of any type of alcoholic beverage?". The possible answers are 'Yes' or 'No'. Not all participants respond to this question, that is, the ALQ variable has some missing values. Further, and again hypothetically, suppose that we only have additional data on the gender of the participants in the survey (which is fully observed). For each of the following situations, choose, justifying, the correct answer.

Question (1a)

- **Description**: Suppose that ALQ is MCAR. The probability of ALQ being missing for those with ALQ=Yes is 0.3. What is the probability of ALQ being missing for those with ALQ=No?
- **Answer**: In our hypothetical survey about alcohol consumption two variables are recorded: ALQ and gender. We note that the ALQ variable contains some missing values whilst the gender variable is fully observed. We now suppose that ALQ is MCAR, indicating that the missingness of the ALQ variable is independent of any data. This implies that the probability of the ALQ being missing for those with ALQ=No must **also be 0.3**. If it were any other value, the probability of the data being missing would then have an implied dependence on the ALQ variable rendering the data to **not** be MCAR.

Question (1b)

- **Description**: ALQ being MAR given gender means:
- **Answer**: (ii) The probability of ALQ being missing is independent of the Yes/No value of ALQ after adjusting for gender.

Question (1c)

- **Description**: Suppose again that ALQ is MAR given gender, and that the probability of ALQ being missing for men is 0.1. What is the probability of ALQ being missing for women?

- **Answer**: (iii) It is impossible to conclude from the information given. I believe this because answer (i) of 0.1 would imply that the data would *actually* be MCAR. Meanwhile, answer (ii) seems to indicate that some conservation of probability must be present which is incorrect to assume. This leaves answer (iii).

Question 2

- **Description**: Suppose that a dataset consists of 100 subjects and 10 variables. Each variable contains 10% of missing values. What is the largest possible subsample under a complete case analysis? What is the smallest? Justify.
- **Answer - Largest CCA Subset:** Let us begin with first defining how our data is presented and stored. We can view the data in a table format with 100 rows and 10 columns. The question does not state the total number of missing values in our dataset of $100 \times 10 = 1000$ values, and so let us first assume a scenario where we have 10 missing values (out of 1000) in our dataset. In order to satisfy the criterion that "*each variable (column) contains 10% of missing values*", then we must have 1 missing value in each of our 10 variables. We know that in a complete case analysis that if any one of a subject's 10 variables contains a missing value, then that subject will not be included in the analysis. Therefore, if all our 10 missing values align along a single row in our dataset, then we would only have to discard a single subject from our complete case analysis leaving a total of **99 subjects**.
- **Answer - Smallest CCA Subset:** We now imagine the converse situation where we have a much greater number of missing values in our dataset. For argument's sake, let us propose that this number is now 100. This time, in order to satisfy the criterion that "*each variable (column) contains 10% of missing values*", we must have 10 missing values in each of our 10 variables. Now suppose that rows 1-10 were missing for variable 1, rows 11-20 were missing for variable 2, rows 21-30 were missing for variable 3 and so on up to variable 10. This kind of structure in our dataset would result in a missing value being present for every subject in our dataset leaving a total of **0 subjects** under a complete case analysis.

Question 3

- **Description**: Consider a two variable (Y_1, Y_2) problem, with each variable defined as follows:

$$Y_1 = 1 + Z_1,\ Y_2 = 5 + 2 \times Z1 + Z2$$

  where Y_1 is fully observed but Y_2 is subject to missingness. Further consider that $Y2$ is missing if $a \times (Y_1 - 1) + b \times (Y_2 - 5) + Z_3 < 0$, where Z_1, Z_2, and Z_3 follow independent standard normal (that is, mean 0 and variance 1) distributions.

Question (3a)

- **Description**: Start by simulating a (complete) dataset of size 500 on (Y_1, Y_2). Then, and considering $a = 2$ and $b = 0$, simulate the corresponding observed dataset (by imposing missingness on Y_2 as instructed above). Is this mechanism MCAR, MAR, or MNAR? Display the marginal distribution of Y_2 for the complete (as originally simulated) and observed (after imposing missingness) data. Comment.
- **Answer**: Given that we know Y_1 is completely observed and Y_2 is only partially observed, we see that our missingness indicator, r, for the parameters $a = 2$ and $b = 0$, is only dependent on Y_1. This, by definition, tells us that the missing data for Y_2 will be MAR. Our beliefs are then confirmed after observing that the distributions (density and box) of the complete, missing and observed data of $Y2$ are **not congruent** (see below).

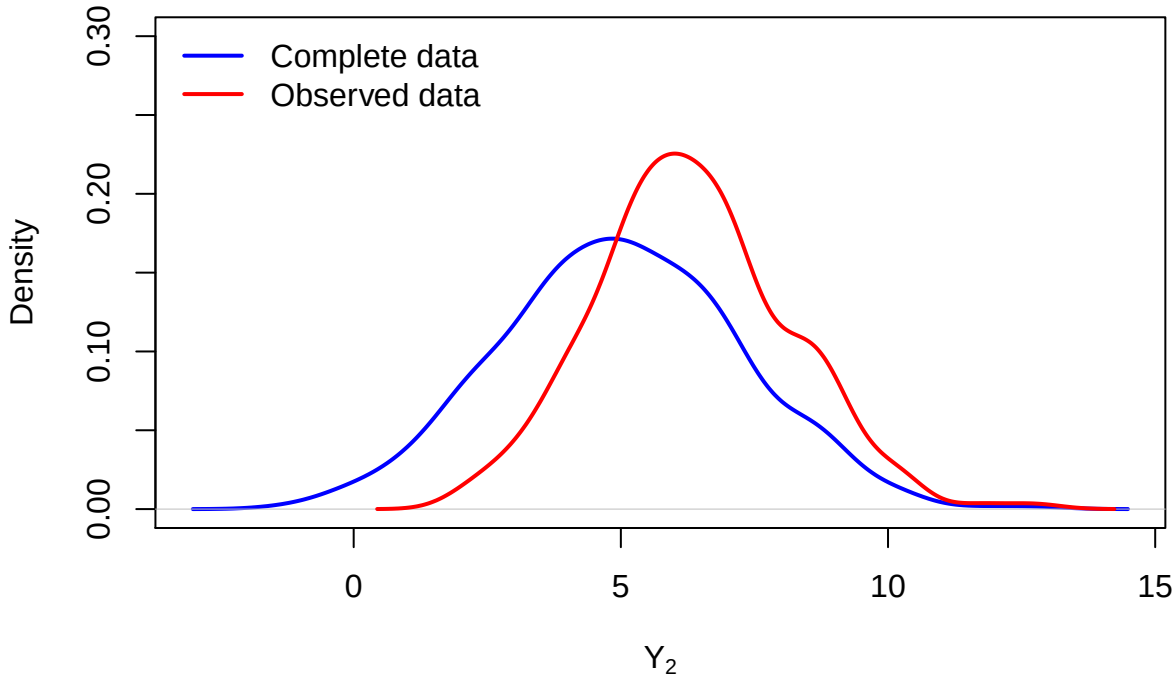
Helper Functions

```
plot_densities = function(data1, data2, ylim = NULL) {
  # Calculate y limit of both densities
  ylim = if (is.null(ylim)) range(d1$d$y, d2$d$y) else ylim

  # Plotting
  plot(data1$d, lwd = 2, col = data1$col, xlab = expression(Y[2]), ylim = ylim, main = NA)
  lines(data2$d, lwd = 2, col = data2$col)
  legend('topleft',
    legend = c(data1$name, data2$name),
    col = c(data1$col, data2$col),
    lty = c(1, 1), lwd = c(2, 2), bty = 'n'
  )
}

# Helper function for calculating the mean and standard error on a dataset
get_statistics =  function(data) {
  # Create an empty list to store the statistics
  results = NULL
  # Computing CCA statistics
  results$mean = mean(data, na.rm = TRUE)
  results$se = sd(data, na.rm = TRUE) / sqrt(sum(Rfunc(a=2, b=0)))
  results$sd = sd(data, na.rm = TRUE)
  return(results)
}
```

Y_2 Marginal Densities

```
# Simulating Gaussian data
nsim = 500
Z1 = rnorm(500, 0, 1)
Z2 = rnorm(500, 0, 1)
Z3 = rnorm(500, 0, 1)

# Define Y1 and Y2 from Z samples
Y1 = 1 + Z1
Y2 = 5 + 2 * Z1 + Z2
Y2_miss = 5 + 2 * Z1 + Z2

# Create missingness indicator for Y2
Rfunc = function(a, b) { a * (Y1 - 1) + b * (Y2 - 5) + Z3 >= 0 }
R = Rfunc(a = 2, b = 0)

# Impose missingness
Y2_miss[!R] = NA
Y2_obs = Y2[R]

# Plot densities for the complete and observed data (after imposing missingness) for Y2
plot_densities(
  list(d = density(Y2), name = 'Complete data', col = 'blue'),
```

```
  list(d = density(Y2_obs), name = 'Observed data', col = 'red'),
  ylim = c(0, 0.30)
)
```

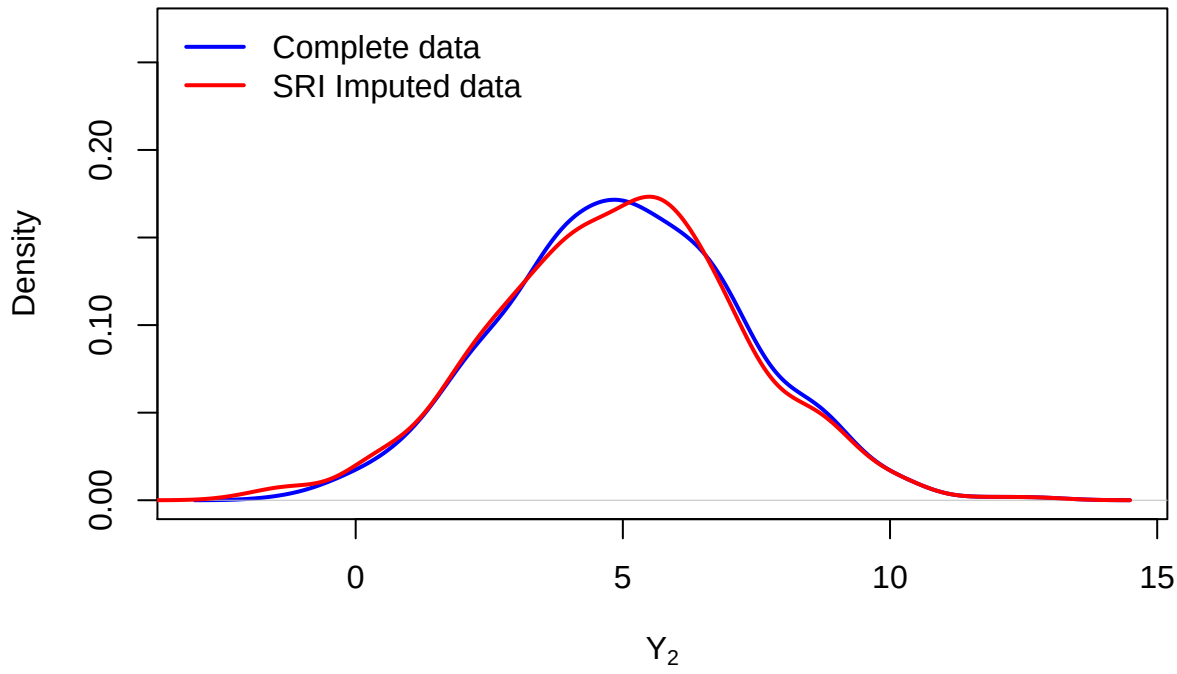


Question (3b)

- **Description**: For the observed dataset simulated in (a), impute the missing values using stochastic regression imputation. Display the marginal distribution of Y2 for the complete (as originally simulated) and completed (after imputation) data. Comment.
- **Answer**: We see that both the complete data density plot, and the density plot after imposing missingness and then imputing the missing values using stochastic regression imputation, are very similar. The only distinguishing difference between the two is that the "Data w/ SRI" density plot is more noisy and less smooth than the complete data density plot as one would expect. I have quantified the similarity between these distributions by computing the mean of both distributions for which I observe their means to both be ≈ 5. I have also computed the standard deviations for both distributions and observe the standard deviation of `Y2_sri` to be slightly larger (2.31) than the complete `Y2` (2.24) distribution which is reflective of the stochastic noise we added to `Y2_sri`. Overall, I believe SRI has done a good job at imputing the missing data values due to the relatively unbiased estimates of the mead and standard deviation.

```
# Defining variables needed for SRI
Y1_miss = Y1[!R]
n_miss = sum(!R)
```

```r
Y2_sri = Y2_miss

# Conducting SRI
fits  = lm(Y2_miss ~ Y1)
# Noise for stochastic regression imputation
noise = rnorm(n_miss, mean = 0, sd = sigma(fits))
Y2_sri[!R] = predict(fits, data.frame(Y1 = Y1_miss)) + noise

# Plot densities for complete and SRI imputed data
plot_densities(
  list(d = density(Y2), name = 'Complete data', col = 'blue'),
  list(d = density(Y2_sri), name = 'SRI Imputed data', col = 'red'),
  ylim = c(0, 0.27)
)
```




```r
# Get statistics
Y2_sri_stats = get_statistics(Y2_sri)

# Print statistics for comparison
cat(sprintf("Complete Mean: %f\nComplete Standard Deviation %f\n", mean(Y2), sd(Y2)))
```

```
## Complete Mean: 4.999348
## Complete Standard Deviation 2.244424
```

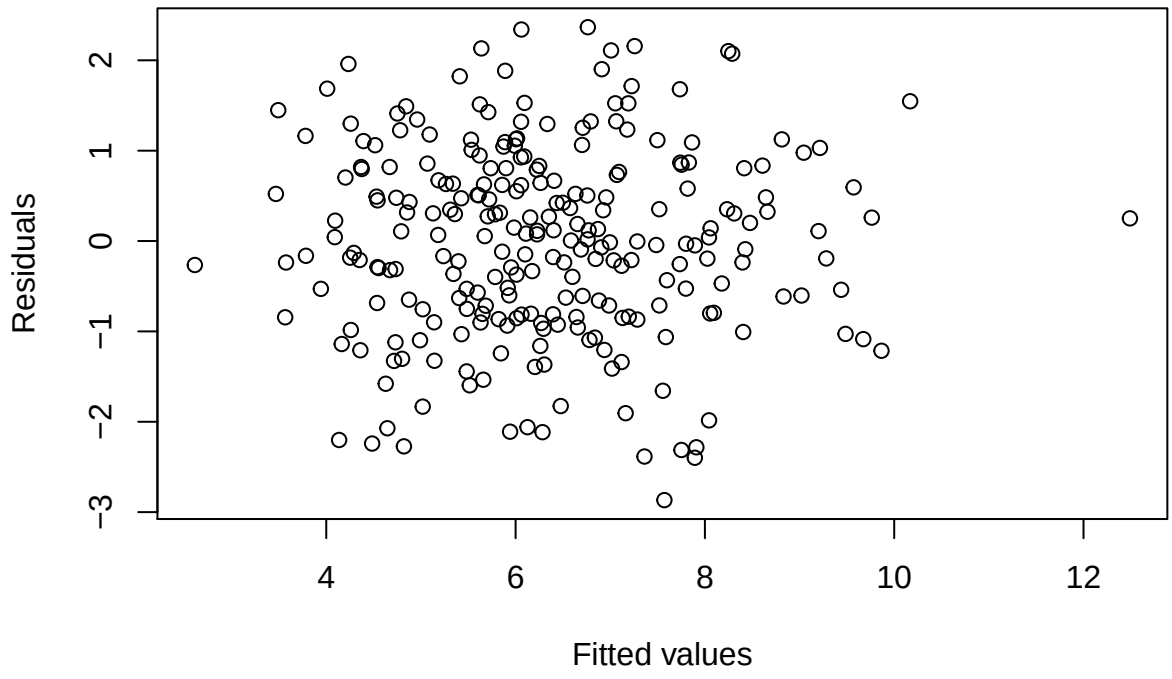
```
cat(sprintf("SRI Mean: %f\nSRI Standard Error: %f\nSRI Standard Deviation: %f\n",
            Y2_sri_stats$mean, Y2_sri_stats$se, Y2_sri_stats$sd))
```

```
## SRI Mean: 4.903801
## SRI Standard Error: 0.144975
## SRI Standard Deviation: 2.310519
```

Linear Model Check

We also check that our linear model fit on Y2 with Y1 is valid before moving on by plotting the residuals. We see no correlation between the residuals and the fitted values, and hence, the linear regression fit was valid.

```
plot(fits$fitted.values, residuals(fits), xlab = "Fitted values", ylab = "Residuals")
```

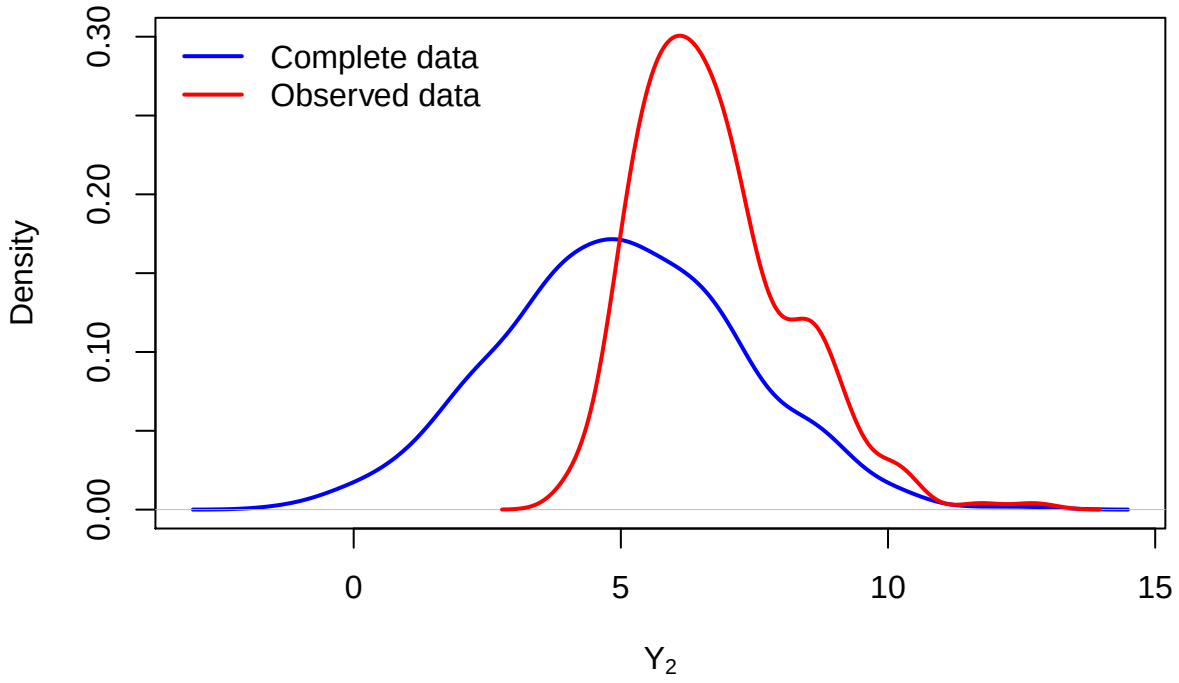


Question (3c)

- **Description**: Using the complete dataset simulated in (a), now impose missingness on Y2 by considering $a = 0$ and $b = 2$. Is this mechanism MCAR, MAR, or MNAR? Display the marginal distribution of Y2 for the complete (as originally simulated) and observed (after imposing missingness) data. Comment.

- **Answer**: Given that we know Y_1 is completely observed and Y_2 is only partially observed, we see that our missingness indicator, r, for the parameters $a = 0$ and $b = 2$, is only dependent on Y_2. This, by definition, tells us that the missing data for Y_2 will be MNAR since the missingess depends on the missing variable itself. Our beliefs are then confirmed after observing that the distributions (density and box) of the complete, missing and observed data of $Y2$ are **even less congruent** than the plots seen in 3(a) (see below).

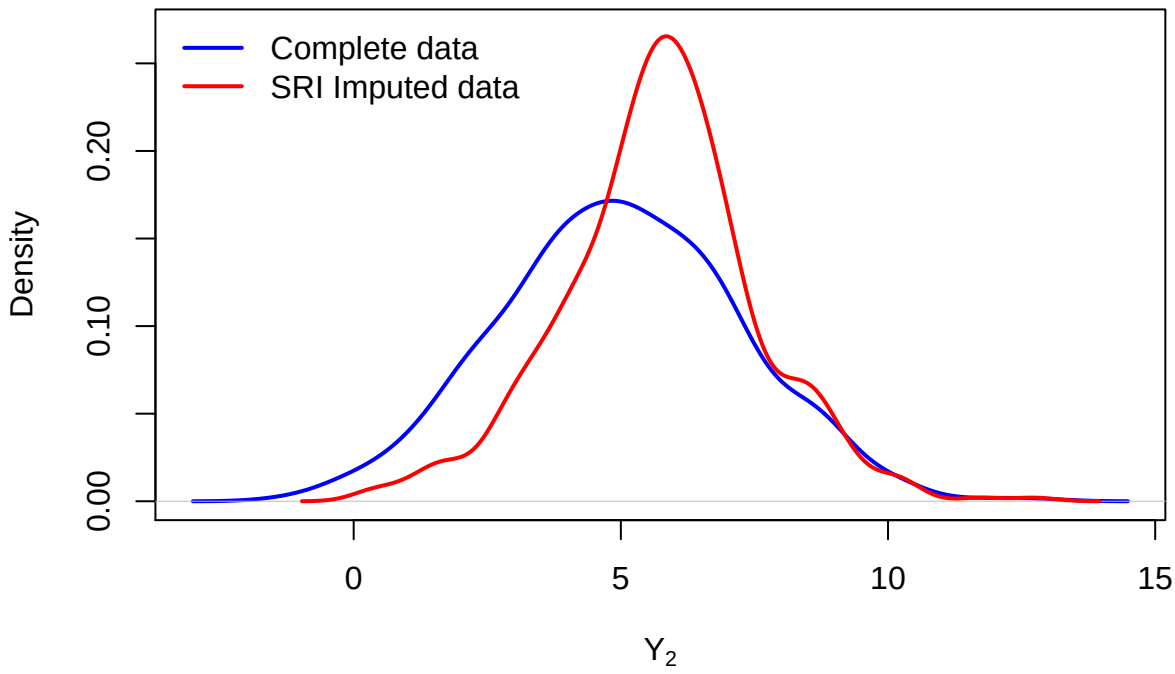
```
# Define Y1 and Y2 from Z samples
Y1 = 1 + Z1
Y2 = 5 + 2 * Z1 + Z2
Y2_miss = 5 + 2 * Z1 + Z2

# Create missingness indicator for Y2
R = Rfunc(a = 0, b = 2)

# Impose missingness
Y2_miss[!R] = NA
Y2_obs = Y2[R]

# Plot densities for the complete and observed data (after imposing missingness) for Y2
plot_densities(
  list(d = density(Y2), name = 'Complete data', col = 'blue'),
  list(d = density(Y2_obs), name = 'Observed data', col = 'red'),
  ylim = c(0, 0.30)
)
```



Question (3d)

- **Description:** The same as in (b) but for the observed data generated in (c).
- **Answer**: Now that the missingness of the variable Y_2 is MNAR, we observe significant deviations of the SRI imputed dataset (mean = 5.72638, sd = 1.814302) from the complete dataset (mean = 4.999348, sd = 2.244424). This indicates that SRI produces **biased estimates** of missing values for MNAR data and so we should look towards methods such as MICE in order to reduce these biases.

```
# Defining variables needed for SRI
Y1_miss = Y1[!R]
n_miss = sum(!R)

# Noise for stochastic regression imputation
noise = rnorm(n_miss, mean = 0, sd = sigma(fits))

# Conducting SRI
fits  = lm(Y2_miss ~ Y1)
Y2_sri = Y2_miss
Y2_sri[!R] = predict(fits, data.frame(Y1 = Y1_miss)) + noise

# Plot densities for complete and SRI imputed data
plot_densities(
  list(d = density(Y2), name = 'Complete data', col = 'blue'),
  list(d = density(Y2_sri), name = 'SRI Imputed data', col = 'red'),
  ylim = c(0, 0.27)
)
```




```
Y2_sri_stats = get_statistics(Y2_sri)

# Print statistics for comparison
cat(sprintf("Complete Mean: %f\nComplete Standard Deviation %f\n", mean(Y2), sd(Y2)))
```

```
## Complete Mean: 4.999348
## Complete Standard Deviation 2.244424
```

```
cat(sprintf("SRI Mean: %f\nSRI Standard Error: %f\nSRI Standard Deviation: %f\n",
            Y2_sri_stats$mean, Y2_sri_stats$se, Y2_sri_stats$sd))
```

```
## SRI Mean: 5.726381
## SRI Standard Error: 0.113839
## SRI Standard Deviation: 1.814302
```

Question 4

- **Description:** It is sometimes necessary to lower a patient's blood pressure during surgery, using a hypotensive drug. Such drugs are administrated continuously during the relevant phase of the operation; because the duration of this phase varies, so does the total amount of drug administered. Patients also vary in the extent to which the drugs succeed in lowering blood pressure. The sooner the blood pressure rises again to normal after the drug is discontinued, 2 the better. The dataset databp.Rdata available on Learn, a partial missing value version of the data presented by Robertson

and Armitage (1959), relate to a particular hypotensive drug and give the time in minutes before the patient's systolic blood pressure returned to 1000mm of mercury (the recovery time), the logarithm (base 10) of the dose of drug in milligrams (you can use this variable as is, no need to transform it to the original scale), and the average systolic blood pressure achieved while the drug was being administered.

Question (4a)

- **Description:** Carry out a complete case analysis to find the mean value of the recovery time (and associated standard error) and to find also the (Pearson) correlations between the recovery time and the dose and between the recovery time and blood pressure.
- **Answer:**

```
# Missingness indicator for the recovery time variable
R_func = function(data) {!is.na(data$recovtime)}
R = R_func(databp)


get_statistics = function(data) {
  # Storing statistics
  results = NULL

 # Computing relevant statistics
  results$mean = mean(data$recovtime, na.rm = TRUE)
  results$se = sd(data$recovtime, na.rm = TRUE) / sqrt(sum(R_func(data)))
  results$cor = NULL
  results$cor$bloodp = cor(data$recovtime, data$bloodp, use = 'complete.obs',
                           method = 'pearson')
  results$cor$logdose = cor(data$recovtime, data$logdose, use = 'complete.obs',
                            method = 'pearson')

  # Display and return the statistics
  cat(sprintf("Mean: %f\nStandard Error: %f\nCorrelation (log-dose):
              %f\nCorrelation (BP): %f", results$mean, results$se,
              results$cor$logdose, results$cor$bloodp))
  return(results)
}
```

```
cca_stats = get_statistics(databp)
```

```
## Mean: 19.272727
## Standard Error: 2.603013
## Correlation (log-dose):
##               0.239126
## Correlation (BP): -0.019529
```

Question (4b)

- **Description:** The same as in (a) but using mean imputation.
- **Answer:**

```r
# Copy the dataset
mi_data = databp
# Mean imputation for recovery time
mi_data[!R, 3] = cca_stats$mean
# Store and report the statistics of the imputed data
mi_stats = get_statistics(mi_data)
```

```
## Mean: 19.272727
## Standard Error: 2.284135
## Correlation (log-dose):
##               0.215061
## Correlation (BP): -0.019341
```

Question (4c)

- **Description:** The same as in (b) but using mean regression imputation.
- **Answer:**

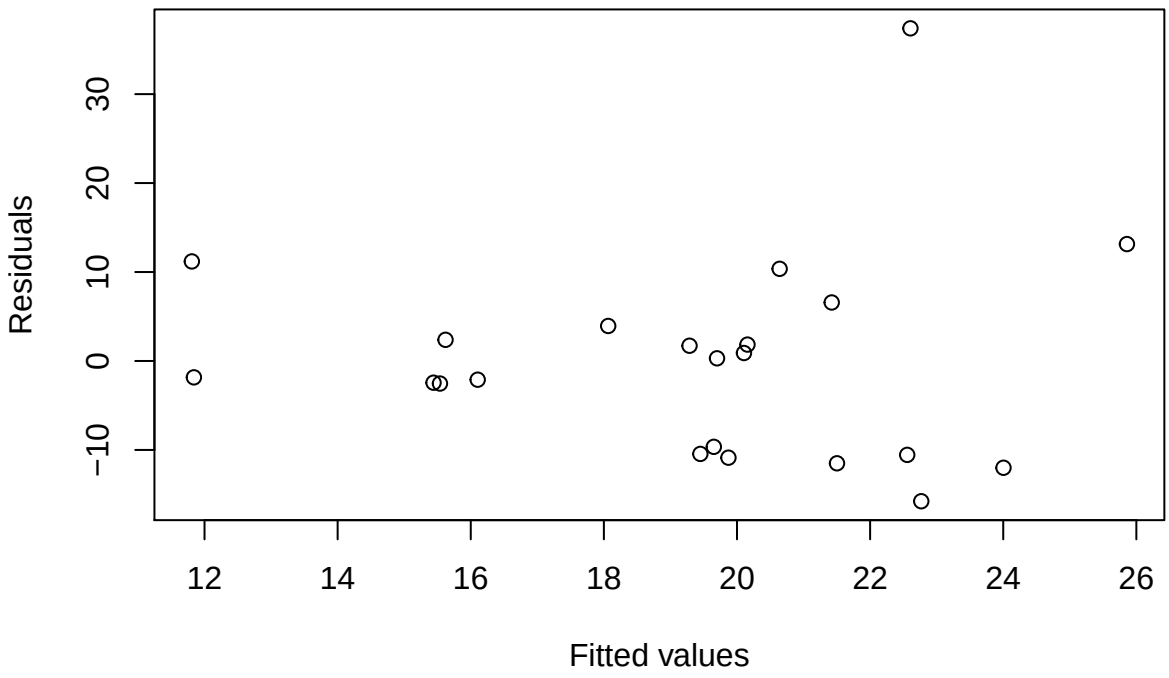
```r
# Copy the dataset
ri_data = databp
# Fit linear model for recovery time with blood pressure and log-dose as covariates
ri_fits = lm(recovtime ~ logdose + bloodp, data = ri_data)
# Impute missing recovery time values using model
ri_data[!R, 3] = predict(ri_fits, ri_data[!R, 1:2])
# Store and report the statistics of the imputed data
ri_stats = get_statistics(ri_data)
```

```
## Mean: 19.444285
## Standard Error: 2.312845
## Correlation (log-dose):
##               0.280184
## Correlation (BP): -0.011136
```

Linear Model Check

We also check that our linear model fit on the recovery time with blood pressure and log-dose is valid before moving on by plotting the residuals. We see no correlation between the residuals and the fitted values, and hence, the linear regression fit was valid.

```r
plot(ri_fits$fitted.values, residuals(ri_fits), xlab = "Fitted values", ylab = "Residuals")
```



Question (4d)

- **Description:** The same as in (c) but using stochastic regression imputation.
- **Answer:**

```
set.seed(1)
# Copy the dataset
sri_data = databp
# Fit linear model for recovery time with blood pressure and log-dose as covariates
sri_fits = lm(recovtime ~ logdose + bloodp, data = sri_data)
# Noise for stochastic regression imputation
noise = rnorm(n = sum(!R), mean = 0, sd = sigma(sri_fits))
# Impute missing recovery time values using model
sri_data[!R, 3] = predict(sri_fits, sri_data[!R, 1:2]) + noise
# Store and report the statistics of the imputed data
sri_stats = get_statistics(sri_data)
```

```
## Mean: 18.817714
## Standard Error: 2.351041
## Correlation (log-dose):
##               0.256621
## Correlation (BP): -0.008132
```

Question (4e)

- **Description:** You will now conduct the same analysis but applying another technique called predictive mean matching (Little, 1988), which is a special type of hot deck imputation. In the simplest form of this method (and the one you will use here), a regression model is used to predict the variables with missing values from the other (complete) variables. For each subject with a missing value, the donor is chosen to be the subject with a predicted value of her or his own that is closest (to be measured by the squared difference) to the prediction for the subject with the missing value.
- **Answer:**

```
set.seed(1)
# Copy the dataset
reg_data = databp
# Recycling sri_fits to get prediction for every row
reg_data[, 3] = predict(sri_fits, reg_data[, 1:2])
# Empty list to store predictions
preds = NULL

# ----------------- PREDICTIVE MEAN MATCHING ALGORITHM  ----------------- #
# Loop over missing subjects
for (i_miss in which(!R)) {
  # Predictions of recovery time for subjects with missing values
  recov_time = reg_data[i_miss, 3]

  # Store distances
  distances = vector(, nrow(reg_data))

  # Preventing the use of donors who originally has missing values
  distances[!R] = Inf

  # Compute distances for observed values
  distances[R] = (rep(recov_time, sum(R)) - reg_data[R, 3])^2

  # Get donor who has smallest distance
  donor = which.min(distances)
  cat(sprintf('Donor for subject %s is %s\n', i_miss, donor))

  # Store the original value of the recovery time from donor
  preds = c(preds, reg_data[donor, 3])
}
```

```
## Donor for subject 4 is 6
## Donor for subject 10 is 2
## Donor for subject 22 is 17
```

```
# Copy data for predictive mean matching
pmm_data = databp
# Overwrite NAs with predictions from above
pmm_data[!R, 3] = preds
# Store and report the statistics of the imputed data
pmm_stats = get_statistics(pmm_data)
```

```
## Mean: 19.472062
```

```
## Standard Error: 2.306657
## Correlation (log-dose):
##               0.273245
## Correlation (BP): -0.011460
```

Question (4f)

- **Description:** What is an advantage of predictive mean matching over stochastic regression imputation? Based on your analysis, can you foresee any potential problem of predictive mean matching?
- **Answer:** An advantage of predictive mean matching over stochastic regression imputation is that the imputed value that is used to replace the missing value originates from a subject who is *similar* (this is quantified using the squared difference). However, predictive mean matching would begin to produce very biased estimates, in a similar manner to mean imputation, if the same donor value is used to impute many missing values. This is a plausible situation when the sample size is small.