mice trial

Laura

2024-11-20

mice trial Load data and install packages

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

Here are necessary functions to run the mice. These functions are there to handle data by e.g., calculating a mask to introduce missingness into a data set without missingness present, normalize data, and also calcualate the RMSE. These functions are based on the functions used in the GAIN(https://doi.org/10.48550/arXiv.1806.02920) paper.

```
# This function generates a mask for introducing the missing values
missing <- function(p, no, dim) {
  mm <- matrix(runif(no*dim),no,dim)
  mask <- ifelse((mm < p),1,NA)
  return(mask)
}


# this function normalize the data
normalization <- function(data, parameters){


  data_norm <- sweep(data, 2, parameters[,1], "-")
  data_norm <- sweep(data_norm, 2, parameters[,2], "/")

}

# this function calculate the RMSE
rmse_loss <- function(ori_data, imputed_data, data_m){

  #I changed this part becuase I was running into a problem since i already normalized so with the prev

  # Ensure that the original data and imputed data are already normalized
```

```
  if (is.null(parameters)) {
    parameters <- parameters_norm(ori_data)  # Calculate parameters only if not provided
  }

  # Normalize the data if parameters are not passed (initial normalization)
  ori_data <- normalization(ori_data, parameters)
  imputed_data <- normalization(imputed_data, parameters)



  # Only for missing values
  nominator <- sum(((1-data_m) * ori_data - (1-data_m) * imputed_data)**2)
  denominator <- sum(1-data_m)

  rmse <- sqrt(nominator/denominator)

  return(rmse)
}
```

Introducing missing data

```
set.seed(123)

# Introduce missing values
mask <- missing(0.2, nrow(data), ncol(data))

spam_data_with_missing <- data * mask
```

Impute missing data

```
##
##  iter imp variable
##    1   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    1   2  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    1   3  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    1   4  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    1   5  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    2   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    2   2  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    2   3  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    2   4  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    2   5  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    3   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    3   2  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    3   3  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    3   4  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    3   5  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    4   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    4   2  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
```

```
##   4   3  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##   4   4  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##   4   5  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##   5   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##   5   2  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##   5   3  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##   5   4  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##   5   5  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
```

```
## Warning: Number of logged events: 168
```

Calculate the overall rmse

```r
# RMSE calculation function
rmse <- function(original, imputed) {
  sqrt(mean((original - imputed)^2, na.rm = TRUE))
}

# Calculate RMSE for each column and get an average
rmse_values <- sapply(names(data), function(col) {
  rmse(data[[col]], spam_data_imputed[[col]])
})
rmse_overall <- mean(rmse_values, na.rm = TRUE)

print(rmse_overall)
```

```
## [1] 17.2708
```

I forgot to normalize the data, which is probably why the rmse is so high(the RMSE is related to the scale of the data lol). SO lets go again:))

Normalize

```r
set.seed(123) # For reproducibility

# Calculate the normalization parameters
parameters_norm <- function(data) {
  mean_sd <- apply(data, 2, function(x) c(mean = mean(x, na.rm = TRUE), sd = sd(x, na.rm = TRUE)))
  return(t(mean_sd))  # returns a matrix with means and sds
}

# Get parameters for normalization
parameters <- parameters_norm(data)

# Normalize the data using the function built above
data_normalized <- normalization(data, parameters)
```

Introduce missingness

```
set.seed(123)
# Introduce missing values
##mask <- missing(0.2, nrow(data_normalized), ncol(data_normalized))

data_with_missing <- data_normalized * mask
```

Imputing missing data

```
##
##  iter imp variable
##    1   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    2   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    3   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    4   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##    5   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co

## Warning: Number of logged events: 11
```

Calculate rmse

```
#I forgot there was already a function to calculate the RMSE above so I will use it from now on
# Calculate RMSE
rmse <- rmse_loss(data_normalized, imputed_data, mask)
print(paste("RMSE:", rmse))
```

```
## [1] "RMSE: NA"
```

```
# retunrs NA so got to change the code of the mask

mask_rmse <- ifelse(is.na(mask), 0, 1)  # Convert NA to 0, keeping valid entries as 1

# Calculate RMSE again
rmse <- rmse_loss(data_normalized, imputed_data, mask_rmse)
print(paste("RMSE:", rmse))
```

```
## [1] "RMSE: 3.99704543094645"
```

```
#looks better but still above 1, this might be due to different normmalization techniques
```

Redo the normalization

So the results above show that the RMSE is still outside of the [0,1] scale, which has been used for the GAIN. This means the two values are not comparable. I noticed looking back at the code that a different

normalization has been used than the one provided above. They used the minmax normalization so that the scale is [0,1] instead of the scale I had above [-1,1]. Fixing the normalization and keeping everything else in place should fix the problem and make our RMSE comparable to the one from GAIN:))

```
normalization <- function(data, parameters = NULL) {
  # If no parameters are provided, calculate min and max for each column
  if (is.null(parameters)) {
    min_val <- apply(data, 2, min, na.rm = TRUE)
    max_val <- apply(data, 2, max, na.rm = TRUE)

    # Normalize data using min-max formula
    norm_data <- sweep(data, 2, min_val, "-")
    norm_data <- sweep(norm_data, 2, max_val - min_val, "/")

    # Return normalized data and the min/max parameters for future renormalization
    return(list(norm_data = norm_data, parameters = list(min_val = min_val, max_val = max_val)))
  } else {
    min_val <- parameters$min_val
    max_val <- parameters$max_val

    # Normalize using pre-calculated min and max values
    norm_data <- sweep(data, 2, min_val, "-")
    norm_data <- sweep(norm_data, 2, max_val - min_val, "/")

    return(norm_data)
  }
}

norm_ <- normalization(data)
norm_data <- norm_$norm_data
parameters <- norm_$parameters  # parameters is a list, containing min_val and max

norms <- normalization(data, parameters)
```

Introduce missingness

```
set.seed(123)

# Introduce missing values by using the same mask generated above
#mask <- missig(0.2, nrow(norms), ncol(norms))

data_with_missing_2 <- norms * mask
```

Imputing missing data

```
##
##  iter imp variable
##   1   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  C
##   2   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  C
```

```
##   3   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##   4   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
##   5   1  Cont.1  Cont.2  Cont.3  Cont.4  Cont.5  Cont.6  Cont.7  Cont.8  Cont.9  Cont.10  Cont.11  Co
```

```
## Warning: Number of logged events: 232
```

Calculate rmse

```
# Calculate RMSE
rmse <- rmse_loss(norms, imputed_data, mask)
print(paste("RMSE:", rmse))
```

```
## [1] "RMSE: NA"
```

```
# retunrs NA so got to change the code of the mask

mask_rmse <- ifelse(is.na(mask), 0, 1)  # Convert NA to 0, keeping valid entries as 1

# Calculate RMSE again
rmse <- rmse_loss(norms, imputed_data, mask_rmse)
print(paste("RMSE:", rmse))
```

```
## [1] "RMSE: 0.0119416829808488"
```

the (hopefully) final rmse is 0.011 where as the one from gain is 0.051. So just by comparing these two numbers it looks like mice did perform better, i do not know whether it is okay to make that statement yet though. It is important to note that while the mask is generated in the same fashion, it is not ensured that it is the same mask. so we are not imputing the same missing values...