A short demonstration of missRanger, for missing data imputation

"Alternative implementation of the beautiful 'MissForest' algorithm used to impute mixedtype data sets by chaining tree ensembles, introduced by Stekhoven, D.J. and Buehlmann, P. (2012) . Under the hood, it uses the lightning fast random jungle package 'ranger'. Between the iterative model fitting, we offer the option of using predictive mean matching. This firstly avoids imputation with values not already present in the original data (like a value 0.3334 in 0- 1 coded variable). Secondly, predictive mean matching tries to raise the variance in the resulting conditional distributions to a realistic level. This would allow e.g. to do multiple imputation when repeating the call to missRanger()."

Functions to look at include:

- *generateNA()*
- *imputeUnivariate()*
- *missRanger(), and*
- *pmm()*

generateNA()

This takes a dataset and add a specified proportion of random NAs by replacement. For example, the fictitious data called *dat* in this example is complete (no missingness). Now we will use the *generateNA()* function to replace 30% of the values with NA. But, first, let's look at the data itself.

```
> head(dat)
  year value outcome factor
1    1   307   20.89      B
2    2   436    6.28      A
3    3   348   16.22      B
4    4   454   15.56      A
5    5   551   13.59      A
6    6   586   10.51      A

> dim(dat)
[1] 100   4
```

```
> datNA <- generateNA(dat, p = 0.3)
> head(datNA)
  year value outcome factor
1   NA   307   20.89      B
2   NA    NA    6.28      A
3    3   348   16.22      B
4   NA   454   15.56      A
5    5   551   13.59   <NA>
6   NA    NA   10.51      A
```

```
> identical(dim(dat), dim(datNA)) # compare the size of the two data
[1] TRUE
```

imputeUnivariate()

"Fills missing values of a vector of any type by sampling with replacement from the non-missing values.

Requires at least one non-missing value to run." Let's see an example.

```
> library(gtools) # loading package gtools
> numbers <- permute(c(1:10, rep(NA,4)))
> numbers
 [1]  1 NA NA  6 NA  8 NA 10  5  3  7  9  4  2
> num.impute <- imputeUnivariate(numbers)
> num.impute
 [1]  1  5  4  6  9  8  5 10  5  3  7  9  4  2
```

Single imputation of datNA using *imputeUnivariate()*

```
> imputeUnivariate(datNA)
Error: is.atomic(x) is not TRUE
```

This generates an error message. *imputeUnivariate()* will not work on a data.frame as it does on vectors.

This can be overcome with the *sapply()* function.

```
> imp.datNA <- as.data.frame(sapply(datNA, imputeUnivariate))
> head(imp.datNA)
  year value outcome factor
1   14   307   20.89      2
2    8   502    6.28      1
3    3   348   16.22      2
4   62   454   15.56      1
5    5   551   13.59      2
6   85   235   10.51      1
```

Factors A and B have become 1 and 2, respectively. To fix that, we another column, factor2, where 1 and

2 will be coded as A and B, respectively.

```
> library(dplyr) # load package "dplyr"
> imp.datNA <- imp.datNA %>% mutate(factor2 = ifelse(factor == 1, "A", "B"))
> head(imp.datNA)
  year value outcome factor factor2
1   14   307   20.89      2       B
2    8   502    6.28      1       A
3    3   348   16.22      2       B
4   62   454   15.56      1       A
5    5   551   13.59      2       B
6   85   235   10.51      1       A
```

Single imputation using *missRanger()*

```
> head(datNA)
  year value outcome factor
1   NA   307   20.89      B
2   NA    NA    6.28      A
3    3   348   16.22      B
4   NA   454   15.56      A
5    5   551   13.59   <NA>
6   NA    NA   10.51      A
```

```
> head(missRanger(datNA))

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done
      year    value outcome factor
1 66.39740 307.0000   20.89      B
2 61.76420 569.1995    6.28      A
3  3.00000 348.0000   16.22      B
4 29.79907 454.0000   15.56      A
5  5.00000 551.0000   13.59      B
6 67.01770 565.2405   10.51      A
```

Multiple imputation using *missRanger()*

```
> library(mosaic) # loading the package

> imp.reg <- function(da){
+   daata <- da
+   daata <- missRanger(daata)
+   reg <- lm(outcome ~ 1 + year + value, daata)
+   return(c(coef(reg), summary(reg)$coefficients[,2]))
+ }
```

This function *imp.reg()* takes the data with missing values, performs single imputation, and extract the parameters of interest and their respective standard errors. Example:

```
> imp.reg(datNA)

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done
  missRanger iteration 2:....done
  missRanger iteration 3:....done
  missRanger iteration 4:....done
 (Intercept)         year        value  (Intercept)         year        value

19.766778325 -0.033905364 -0.009682270  2.145737384  0.021246546  0.003892548
```

Replicating the process will generate different estimates. Example:

```
> library(mosaic) # loading the package

> do(3) * imp.reg(datNA)

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done
  missRanger iteration 2:....done
  missRanger iteration 3:....done
  Intercept       year        value Intercept.1     year.1     value.1
1  14.93511 0.01460099 -0.005724894    2.499202 0.02443774 0.004401453
2  11.72002 0.01397208  0.002643992    2.475530 0.02494745 0.004472413
3  13.28706 0.01837385 -0.002024648    2.294922 0.02229022 0.004060234
```

Organizing the result for multiple imputation using *missRanger()*

```
> my_table <- as.data.frame(do(5) * imp.reg(datNA))

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done
  missRanger iteration 2:....done
  missRanger iteration 3:....done

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done

Missing value imputation by chained tree ensembles
  missRanger iteration 1:....done

> names(my_table) <- c("intercept", "year", "value", "se.int",
+                      "se.year", "se.value")
> final<- colMeans(my_table)
> final
   intercept         year        value       se.int      se.year     se.value

15.370306069 -0.005577015 -0.003776427  2.336270564  0.023531725  0.004235149
```