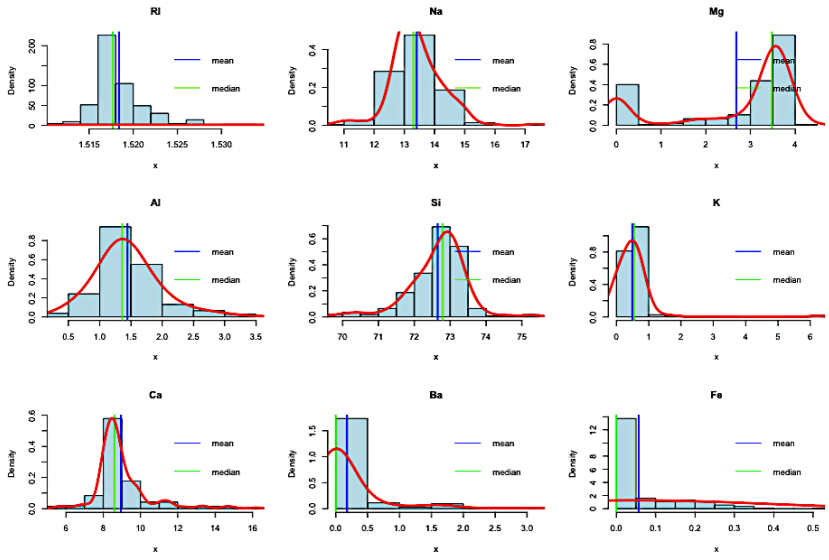
ISE 5103 Intelligent Data Analytics **Homework #4**

By
Akhila Podupuganti

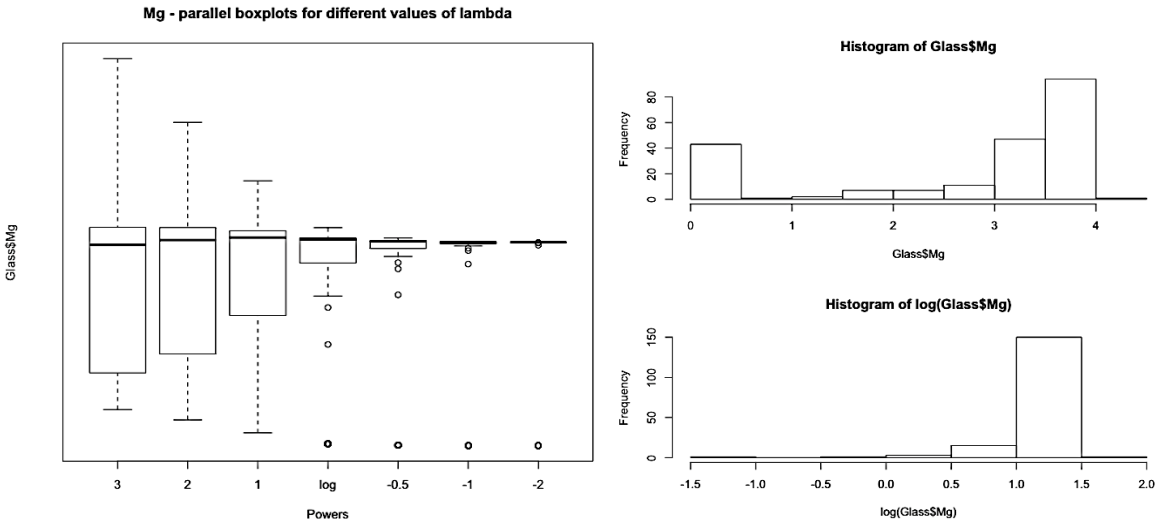
1. Glass data transformations

- Identify three attributes that benefit from skew transformation.

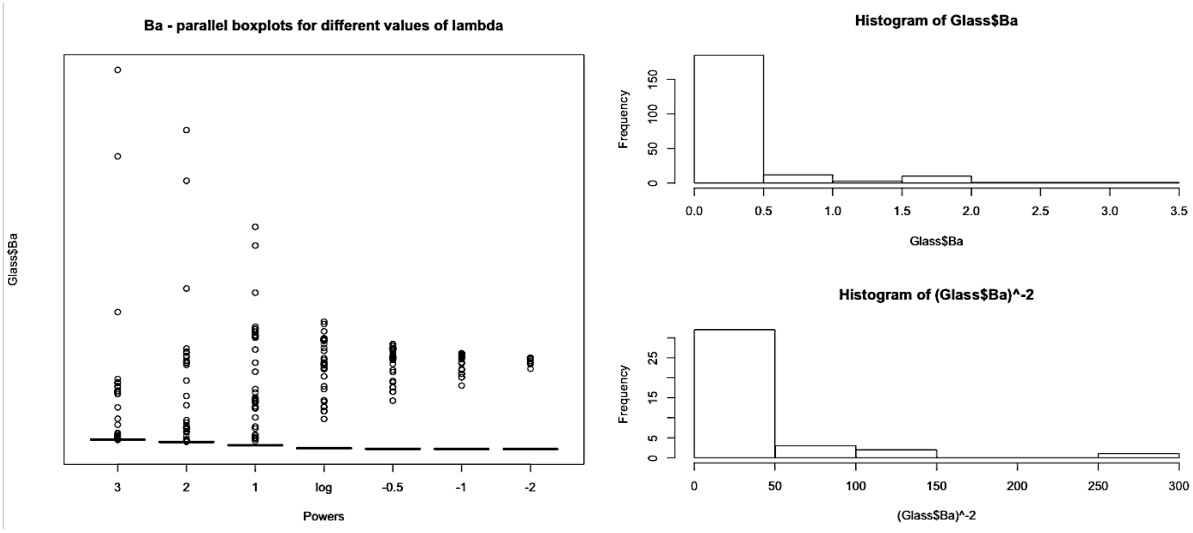


From the above hist plots I think Mg, Ba, Fe attributes benefit more from skew transformation. Because I can see there is a difference in between mean and medium. We can also see skewness from other attributes like K, Ca, RI where it is because of outliers.

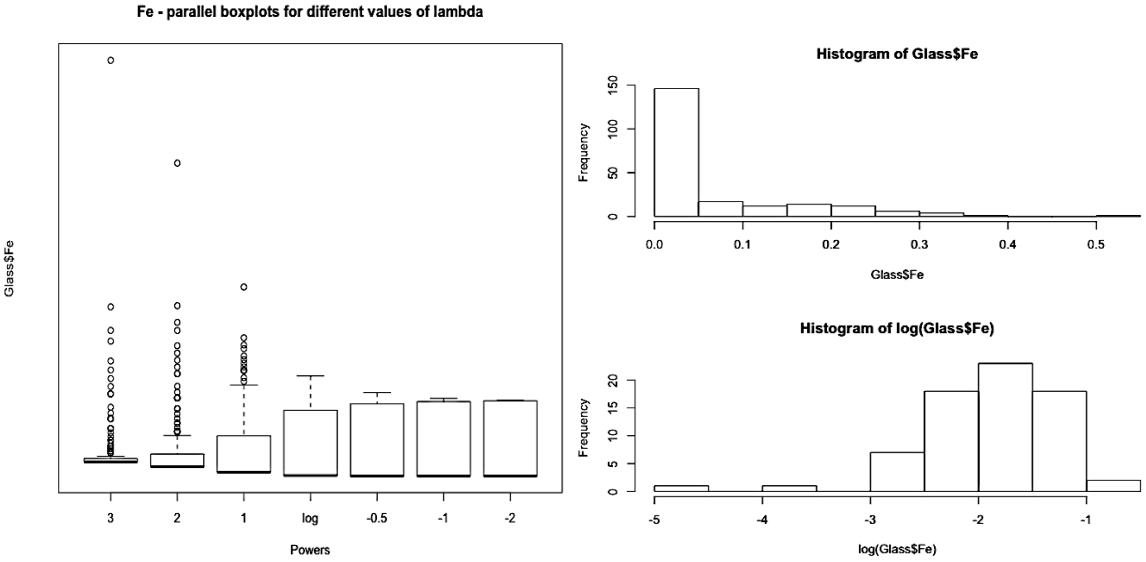
i. Use symbox and find the optimum power for transforming the data for identified attributes



From the symbox parallel boxplots with different lambda values we could see log can reduce very little skewness, though it is not a great result.

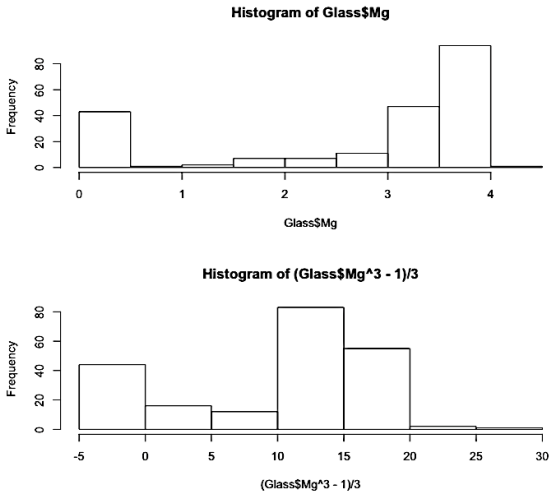


From the plot we could see the data is skewed a lot. By applying power -2 we can reduce very little skewness. Again though it is not a great result.



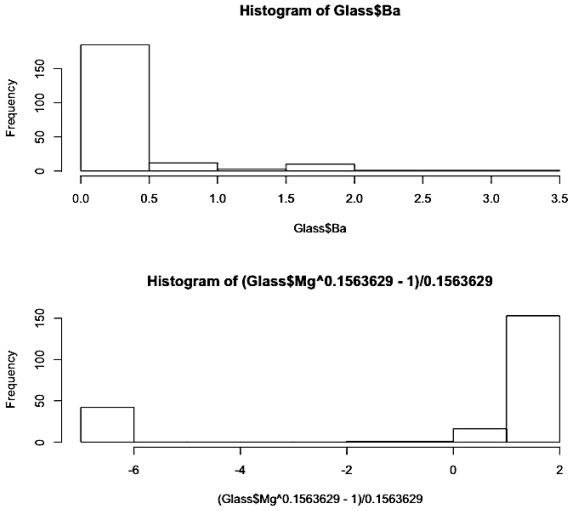
From the plot we could see the for log we got good result by removing skewness. Still may not be a perfect result.

ii. Use Boxcox and find the optimum probability for transforming the data from identified attributes

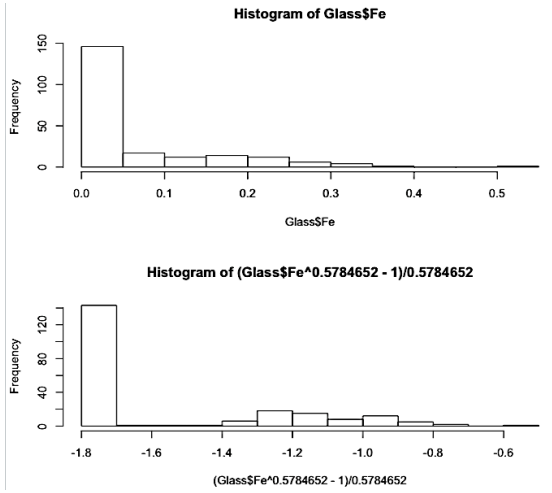


From boxcox I got optimum value lambda 3, for which transformed to get distribution as in left below plot.

There are many 0 values. Keeping it aside I think it did transformed a bit by reducing the skewness.



From boxcox I got optimum value lambda as 0.1563629 for which transformed histogram is in left below plot.



From boxcox got optimum lambda value as 0.5784652. For which transformed histogram is in left below plot.

2. Missing Data

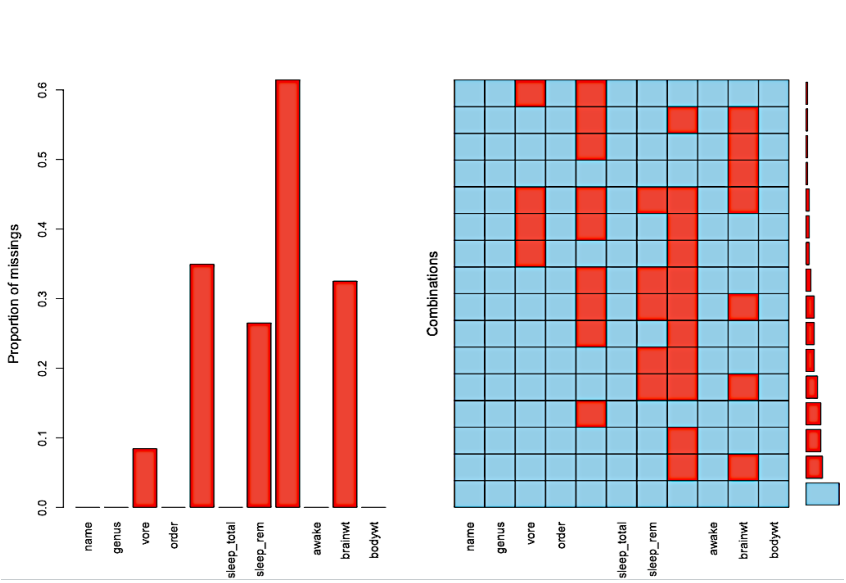
a) Different packages to explore the missing data

Missing data percent for numeric columns using *tidyvese* library. In which the data is missing in *sleep_rem, sleep_cycle* and *brainwt* attributes.

```
> msleep %>% select_if(is.numeric) %>% mutate_all(is.na) %>% summarise_all(mean)
# A tibble: 1 x 6
  sleep_total sleep_rem sleep_cycle awake brainwt bodywt
        <dbl>     <dbl>       <dbl> <dbl>   <dbl>  <dbl>
1           0     0.265       0.614     0   0.325      0
```

VIM package

- Aggr




```
> summary(a)

Missings per variable:
    Variable Count
        name     0
       genus     0
        vore     7
       order     0
conservation    29
 sleep_total     0
   sleep_rem    22
 sleep_cycle    51
       awake     0
     brainwt    27
     bodywt     0

Missings in combinations of variables:
        Combinations Count   Percent
0:0:0:0:0:0:0:0:0:0:0    20 24.096386
0:0:0:0:0:0:0:0:0:1:0     1  1.204819
0:0:0:0:0:0:0:1:0:0:0     9 10.843373
0:0:0:0:0:0:0:1:0:1:0    10 12.048193
0:0:0:0:0:0:1:1:0:0:0     5  6.024096
0:0:0:0:0:0:1:1:0:1:0     7  8.433735
0:0:0:0:1:0:0:0:0:0:0     9 10.843373
0:0:0:0:1:0:0:0:0:1:0     1  1.204819
0:0:0:0:1:0:0:1:0:0:0     5  6.024096
0:0:0:0:1:0:0:1:0:1:0     1  1.204819
0:0:0:0:1:0:1:1:0:0:0     3  3.614458
0:0:0:0:1:0:1:1:0:1:0     5  6.024096
0:0:1:0:0:0:0:1:0:0:0     2  2.409639
0:0:1:0:1:0:0:0:0:0:0     1  1.204819
0:0:1:0:1:0:0:1:0:0:0     2  2.409639
0:0:1:0:1:0:1:1:0:1:0     2  2.409639
```

Missing's (The plot on left)- Containing the amount of missing or imputed values in each variable.

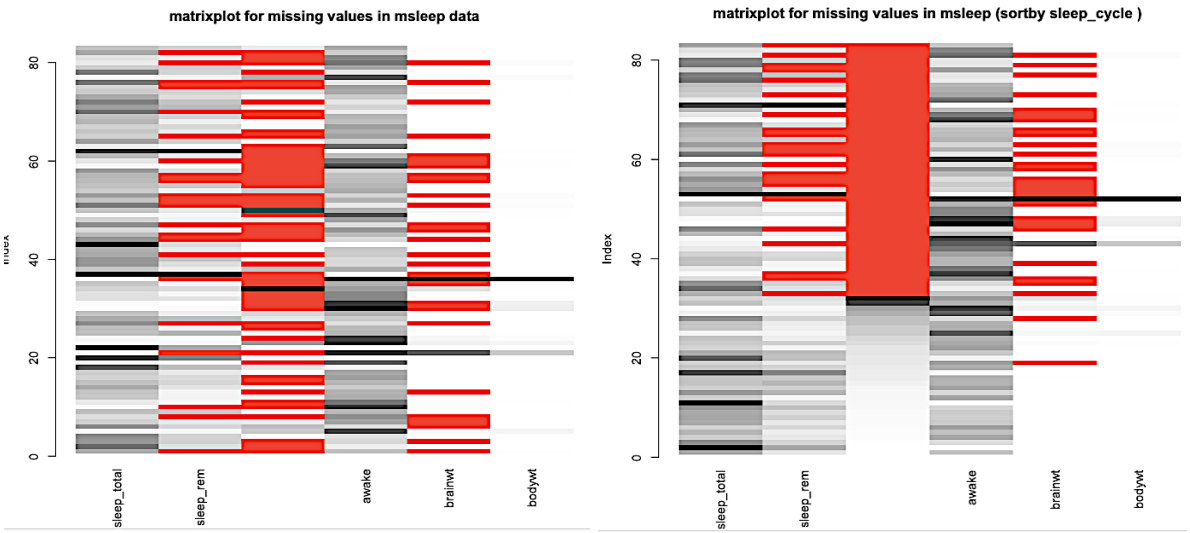
Combinations (The plot on right) - The combinations of variables along with their frequencies and percentages. Understanding most frequencies from the combinations

- 20 rows which means 24% of rows shows contain no NA values (no missing data)
- Similarly 10 rows which means of 12% of rows contain NA's in both *brainwt* and *sleep_cycle.* same goes with all other combinations.

This helps to study the amount and distribution of our missing values. Statistics of missing values can be very valuable in a data quality process.

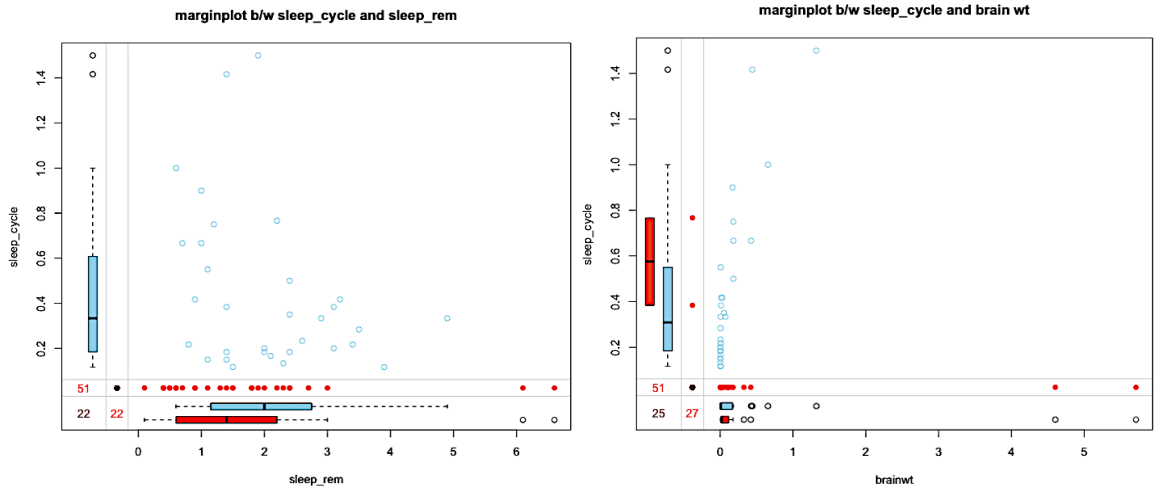
- Matrixplot

  The VIM function matrixplot creates a matrix plot in which all cells of a data matrix are visualized by rectangles/ by horizontal lines.



- Available data is coded according to a continuous color scheme (gray scale), For the grey scale the data are first scaled to the interval [0,1] for each variable by subtracting the mean and dividing by the range. Lighter colors indicate lower values, darker colors suggest larger values and values equal to 0 are colored in white.
- While missing / imputed data is visualized by a clearly distinguishable color (red).
- If you use Rstudio the plot is not interactive , but if you use R directly, you can click on a column of your choice: the rows are sorted (decreasing order) of the values of this column. This is useful to check if there is an association between the value of a variable and the missingness of another one. As we see on the right plot which is sorted by *sleep_cycle* we can see the most of the missing data combinations are associated with *sleep_rem and Brainwt* attributes.

- Marginplot



The marginplot creates a scatterplot with additional information on the missing values. If you plot the variables (x,y), the points with no missing values are represented as in a standard scatterplot. The points for which x (resp. y) is missing are represented in red along the y (resp. x) axis. In addition, boxplots of the x and y variables are represented along the axes with and without missing values (in red all variables x

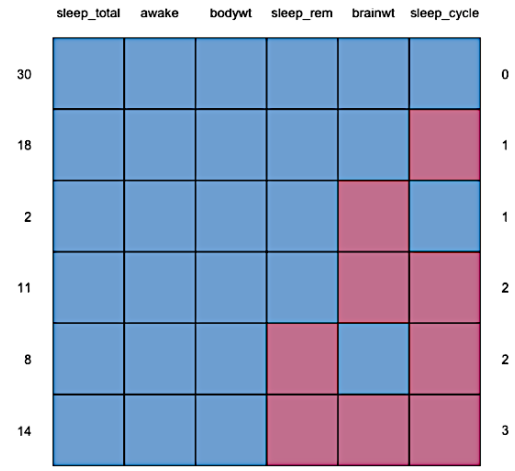
where y is missing, in blue all variables x where y is observed). The plot above allows you to examine the pattern and distribution of complete and incomplete observations.

From the plots above (I considered the missing attributes which seems to be associated with each other from above matrix) both seems to be missing at random (MAR).

Mice package

- Md.pattern

  The mice package provides a nice function md.pattern() to get a better understanding of the pattern of missing data




```
> md.pattern(x, plot = TRUE)
   sleep_total awake bodywt sleep_rem brainwt sleep_cycle
30           1     1      1         1       1           1   0
18           1     1      1         1       1           0   1
2            1     1      1         1       0           1   1
11           1     1      1         1       0           0   2
8            1     1      1         0       1           0   2
14           1     1      1         0       0           0   3
             0     0      0        22      27          51 100
```

The output tells us that 30 samples are complete, 18 samples miss only the sleep_cycle measurement, 2 samples miss only the brainwt value and so on. Perhaps more helpful visual representation can be obtained using the VIM package aggr.

- Md.pairs

  In practice, md.pattern() is primarily useful for datasets with a small number of columns. Alternative measures start from pairs of variables. A pair of variables (Y j , Y k) can have four missingness patterns like below:

  - both Y j and Y k are observed (pattern rr);
  - Y j is observed and Y k is missing (pattern rm);
  - Y j is missing and Y k is observed (pattern mr);
  - both Y j and Y k are missing (pattern mm).

```
> md.pairs(x)
$rr
            sleep_total sleep_rem sleep_cycle awake brainwt bodywt
sleep_total          83        61          32    83      56     83
sleep_rem            61        61          32    61      48     61
sleep_cycle          32        32          32    32      30     32
awake                83        61          32    83      56     83
brainwt              56        48          30    56      56     56
bodywt               83        61          32    83      56     83

$rm
            sleep_total sleep_rem sleep_cycle awake brainwt bodywt
sleep_total           0        22          51     0      27      0
sleep_rem             0         0          29     0      13      0
sleep_cycle           0         0           0     0       2      0
awake                 0        22          51     0      27      0
brainwt               0         8          26     0       0      0
bodywt                0        22          51     0      27      0

$mr
            sleep_total sleep_rem sleep_cycle awake brainwt bodywt
sleep_total           0         0           0     0       0      0
sleep_rem            22         0           0    22       8     22
sleep_cycle          51        29           0    51      26     51
awake                 0         0           0     0       0      0
brainwt              27        13           2    27       0     27
bodywt                0         0           0     0       0      0

$mm
            sleep_total sleep_rem sleep_cycle awake brainwt bodywt
sleep_total           0         0           0     0       0      0
sleep_rem             0        22          22     0      14      0
sleep_cycle           0        22          51     0      25      0
awake                 0         0           0     0       0      0
brainwt               0        14          25     0      27      0
bodywt                0         0           0     0       0      0
```

From the left md.pairs(x) where again I considered only numeric columns to check missingness.

Thus for example consider two variables sleep_cycle and sleep_rem.

- There are 32 complete observed pairs (in rr).
- No rows in which sleep_cycle is observed and sleep_rem is missing (in rm).
- 29 rows in which sleep_cycle is missing and sleep_rem is observed (in mr)
- 22 rows in which both are missing.

Therefore, these numbers add up to the total sample size. Similarly with other pair of combinations.

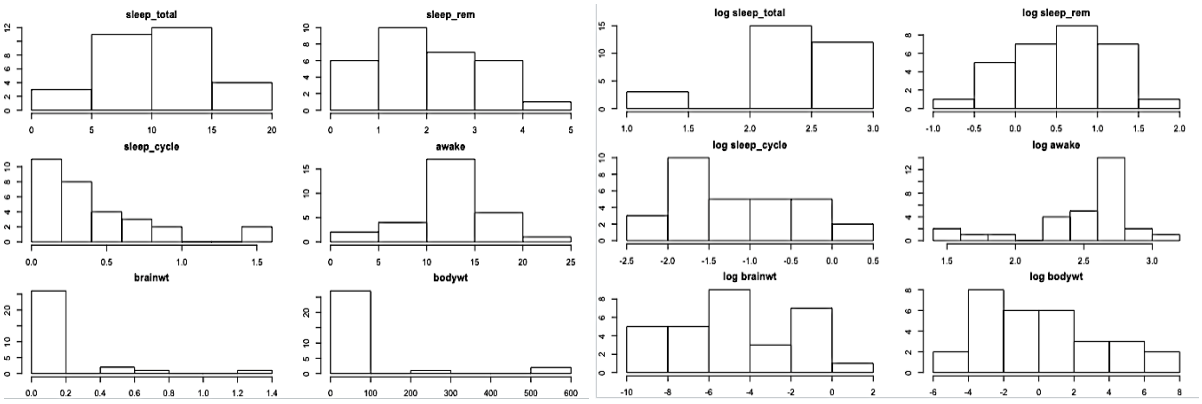
Naniar package

This package can also be used to evaluate and analyze missingness of data with great visualizations. Please refer in Podupuganti_HW4.R file

b) Read given paper and apply any one transformation to study the relationship between sleep attributes and body factors

From my skimming I got to see they apply logarithmic transformations to check between Sleep and Brain (grams), for which they got InSite that sleep scales on brain not on whole body.

Lets 1st see the hist plots for both before and after logarithmic transformations



Here we can say that the logarithmic works well. Let's play around with some similar which can effect sleep. Before and after transformation.

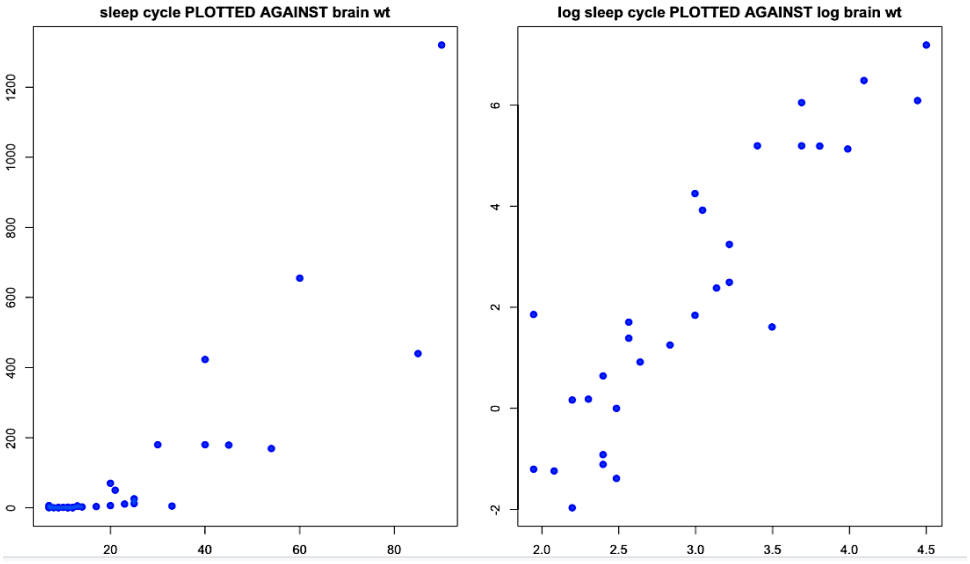
For correlation matric before transformation, There is 85% correlation b/w sleep cycle and brain.wt

```
> cor((completedata))
             sleep_total  sleep_rem sleep_cycle      awake    brainwt     bodywt
sleep_total    1.0000000  0.6618840  -0.5159194 -1.0000000 -0.4597432 -0.6035842
sleep_rem      0.6618840  1.0000000  -0.3522936 -0.6618840 -0.2820403 -0.4089389
sleep_cycle   -0.5159194 -0.3522936   1.0000000  0.5159194  0.8516203  0.4343020
awake         -1.0000000 -0.6618840   0.5159194  1.0000000  0.4597432  0.6035842
brainwt       -0.4597432 -0.2820403   0.8516203  0.4597432  1.0000000  0.4830048
bodywt        -0.6035842 -0.4089389   0.4343020  0.6035842  0.4830048  1.0000000
```

After transformation, The correlation increased to 89.9%

```
> cor(log(completedata))
             sleep_total  sleep_rem sleep_cycle      awake    brainwt     bodywt
sleep_total    1.0000000  0.7050120  -0.5883487 -0.8569896 -0.6312741 -0.6818323
sleep_rem      0.7050120  1.0000000  -0.3792345 -0.6154390 -0.3572937 -0.3722336
sleep_cycle   -0.5883487 -0.3792345   1.0000000  0.4921093  0.8994925  0.8532293
awake         -0.8569896 -0.6154390   0.4921093  1.0000000  0.5636381  0.5522135
brainwt       -0.6312741 -0.3572937   0.8994925  0.5636381  1.0000000  0.9616895
bodywt        -0.6818323 -0.3722336   0.8532293  0.5522135  0.9616895  1.0000000
```

Scatter plot b/w sleep_cycle and brain_wt, Where we can see clear relation after logarithm transformation.



c) Multiple imputation

i. Use mice to conduct multiple imputation

mice() imputes each missing value with a plausible value (simulates a value to fill-in the missing one) until all missing values are imputed and dataset is completed. Repeats the process for multiple times, say m times and stores all the m complete(d)/imputed datasets.

```
> impdata <- mice(msleep)
```

Default imputation method used here is *pmm*, with default no of imputations used is 5. Down below fig shows the output of impdata. As we see It applied *pmm* to missing variables only.

```
> impdata
Class: mids
Number of multiple imputations:  5
Imputation methods:
        name        genus         vore        order conservation  sleep_total    sleep_rem
          ""           ""           ""           ""           ""           ""       "pmm"
 sleep_cycle        awake      brainwt      bodywt
       "pmm"           ""       "pmm"           ""
PredictorMatrix:
              name genus vore order conservation sleep_total sleep_rem sleep_cycle awake brainwt
name             0     0    0     0            0           1         1           1     0       1
genus            0     0    0     0            0           1         1           1     0       1
vore             0     0    0     0            0           0         0           0     0       0
order            0     0    0     0            0           1         1           1     0       1
conservation     0     0    0     0            0           0         0           0     0       0
sleep_total      0     0    0     0            0           0         1           1     0       1
              bodywt
name               1
genus              1
vore               0
order              1
conservation       0
sleep_total        1
Number of logged events:  6
  it im dep      meth         out
1  0  0      constant        name
2  0  0      constant       genus
3  0  0      constant        vore
4  0  0      constant       order
5  0  0      constant conservation
6  0  0     collinear       awake
```

ii. Build and Evaluate a linear regression model based on the computed multiple imputation

```
> fit<-with(impdata,lm(sleep_total ~ sleep_cycle+sleep_rem))
> summary(fit)
```

```
> summary(fit<-with(impdata,lm(sleep_total~sleep_cycle+sleep_rem)))
# A tibble: 15 x 5
   term        estimate std.error statistic  p.value
   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
 1 (Intercept)    7.92      0.916      8.64 4.42e-13
 2 sleep_cycle   -3.77      1.11      -3.40 1.07e- 3
 3 sleep_rem      2.32      0.283      8.19 3.36e-12
 4 (Intercept)    8.48      0.937      9.05 6.90e-14
 5 sleep_cycle   -4.30      1.07      -4.01 1.37e- 4
 6 sleep_rem      2.23      0.293      7.63 4.33e-11
 7 (Intercept)   10.5       0.794     13.2  7.38e-22
 8 sleep_cycle   -6.15      0.880     -6.99 7.34e-10
 9 sleep_rem      1.46      0.225      6.49 6.68e- 9
10 (Intercept)    7.56      0.832      9.08 6.04e-14
11 sleep_cycle   -3.67      1.01      -3.61 5.25e- 4
12 sleep_rem      2.33      0.283      8.25 2.59e-12
13 (Intercept)    7.42      0.845      8.77 2.46e-13
14 sleep_cycle   -3.50      1.03      -3.40 1.04e- 3
15 sleep_rem      2.42      0.288      8.39 1.41e-12
```

with() analyses each of the 5 completed datasets separately based on the analysis model you want. Here we are considering linear regression model for analysis.

Summary gives the consolidated evaluation of all 5 datasets. Where we check for p-values, standard error and estimated values.

```
> combFit <- pool(fit)
> summary(combFit)
```

```
> summary(combFit)
             estimate std.error statistic       df     p.value
(Intercept)  8.700797 1.6177220  5.378425 6.280698 0.001467413
sleep_cycle -4.591771 1.9093859 -2.404842 6.220786 0.051489321
sleep_rem    2.069644 0.4770153  4.338737 6.829235 0.003604847
```

pool() combines (pools) all the results together based on Rubin's rules (Rubin, 1987).

iii. Compare regression coefficients and p-values with same regression model on complete cases

I took same linear regression model for complete data for p values and co-efficients. From the coefficients we can see that difference with complete data and imputation data, Where there is more difference with sleep_total, and cycle. But negligible for the sleep_rem. From which we can say sleep_total doesn't scale with sleep_rem.

```
> summary(fullfit<-lm(data=na.omit(msleep),sleep_total~sleep_cycle+sleep_rem)) # with complete data

Call:
lm(formula = sleep_total ~ sleep_cycle + sleep_rem, data = na.omit(msleep))

Residuals:
    Min      1Q  Median      3Q     Max
-4.3861 -1.1389  0.3269  0.9321  4.6103

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.8339     1.6506   5.352 5.28e-05 ***
sleep_cycle  -7.9351     2.2796  -3.481 0.002859 **
sleep_rem     2.2573     0.4723   4.779 0.000174 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.288 on 17 degrees of freedom
Multiple R-squared:  0.7732,    Adjusted R-squared:  0.7465
F-statistic: 28.97 on 2 and 17 DF,  p-value: 3.337e-06
```

iv. Repeat the multiple imputation analysis and linear model with mice using different imputation methods and compare results.

Regression analysis generates an equation to describe the statistical relationship between one or more predictor variables and the response variable.

```
- methods <- c('norm','norm.predict')
- for( m in methods)
- {
- impdata<-mice(msleep,meth=m)
- print(impdata)
- fit<-with(impdata,lm(sleep_total~sleep_cycle+sleep_rem))
- print(summary(fit<-
  with(impdata,lm(sleep_total~sleep_cycle+sleep_rem))))
- combFit <- pool(fit)
- print(summary(combFit))
- }
```

Norm imputation method

	term	estimate	std.error	statistic	p.value
1	(Intercept)	6.591430	0.6171145	10.681049	4.620817e-17
2	sleep_cycle	-1.321265	0.3346657	-3.948014	1.685054e-04
3	sleep_rem	2.340211	0.2478614	9.441611	1.193498e-14
4	(Intercept)	9.283333	0.5634583	16.475634	1.962489e-27
5	sleep_cycle	-4.873413	0.6123453	-7.958604	9.747818e-12
6	sleep_rem	1.754418	0.2159569	8.123923	4.623277e-12
7	(Intercept)	6.868765	0.6154196	11.161108	5.537411e-18
8	sleep_cycle	-1.734629	0.3609009	-4.806386	7.081469e-06
9	sleep_rem	2.396617	0.2503501	9.573064	6.596158e-15
10	(Intercept)	8.116923	0.5210144	15.579077	6.031169e-26
11	sleep_cycle	-3.297583	0.3919037	-8.414269	1.244459e-12
12	sleep_rem	2.062764	0.2084664	9.894951	1.549091e-15
13	(Intercept)	8.268218	0.5782134	14.299596	9.556871e-24
14	sleep_cycle	-3.810414	0.4618721	-8.249932	2.616464e-12
15	sleep_rem	2.170836	0.2362907	9.187141	3.767707e-14

Norm.predict imputation method

	term	estimate	std.error	statistic	p.value
1	(Intercept)	6.388901	0.5822397	10.972975	1.268741e-17
2	sleep_cycle	-2.010576	0.3349903	-6.001894	5.417834e-08
3	sleep_rem	2.662165	0.2458109	10.830137	2.385881e-17
4	(Intercept)	6.839052	0.5548966	12.324913	3.543800e-20
5	sleep_cycle	-2.667111	0.3615207	-7.377476	1.324408e-10
6	sleep_rem	2.577328	0.2285354	11.277588	3.319551e-18
7	(Intercept)	6.551126	0.5796605	11.301662	2.986895e-18
8	sleep_cycle	-2.281685	0.3754499	-6.077202	3.932741e-08
9	sleep_rem	2.634015	0.2379686	11.068752	8.315585e-18
10	(Intercept)	6.079997	0.6450378	9.425799	1.281791e-14
11	sleep_cycle	-1.293691	0.3668823	-3.526175	7.011311e-04
12	sleep_rem	2.656762	0.2641953	10.056053	7.516970e-16
13	(Intercept)	6.575605	0.6001316	10.956938	1.361847e-17
14	sleep_cycle	-1.885073	0.3338628	-5.646250	2.413862e-07
15	sleep_rem	2.582941	0.2446208	10.558959	7.950213e-17

The p-value for each term tests the null hypothesis that the coefficient is equal to zero (no effect). A low p-value (< 0.05) indicates that you can reject the null hypothesis. In other words, a predictor that has a low p-value is likely to be a meaningful addition to your model because changes in the predictor's value are related to changes in the response variable. Thus from this comparison I feel My linear regression model works best Norm.predict imputation method where there less very less effect from p value.