R Function: missingMatrix

Austin Putz

May 18, 2016

1 Introduction

This function was inspired after struggling with understanding the data's missing structure. It's important to be able to uncover data missingness for modeling purposes especially. Linear models need a response value and all predictors to be available to be used in the analyses.

2 Description

Give the function a data frame or matrix and it will output a matrix with counts or percentages of missing values.

On the **diagonal**, it will print count/percentage of missing values in that column. So element 1,1 will be the count/percent missing in column 1 of the data frame/matrix.

In the **upper triangle**, it will calculate counts/percentages of missing values that occur in either column. So element 1,2 will give the count/percentage that either column 1 or 2 has a missing value.

In the **lower triangle**, it will print counts/percentages when *both* values are missing, not only a single column is missing.

3 Inputs

1. Data frame or matrix

4 Options

1. percent

   logical; default TRUE; do you want in a count or a percent, TRUE will give you the output in percentages, FALSE will give you output in counts

2. missing

   logical; default TRUE; do you want counts in not-NA (FALSE) or NA (TRUE)

3. digits

   integer; default 2; number you want the output rounded to; only for percent=TRUE

5 Example

Table 1 will be used in this example. It has 7 records and 3 fields. Colors: red is for diagonals, dark orchid is for upper triangular values, and dodger blue is for lower triangular values.

Table 1: Example data set used

	col1	col2	col3
1	1.00		1.00
2	2.00		
3	3.00	5.00	
4	4.00	7.00	5.00
5	5.00		1.00
6	6.00		3.00
7	9.00		4.00

5.1 percent=TRUE, missing=TRUE

```
> missingMatrix(my_df, percent=TRUE, missing=TRUE)

Data Frame
     col1  col2  col3
col1    0 71.43 28.57
col2    0 71.43 85.71
col3    0 14.29 28.57
```

Interpretation:

1. 1,1: There are 0% (0/7) missing values in column 1.
2. 1,2: There are 71.43% (5/7) missing at least 1 record between columns 1 and 2.
3. 1,3: There are 28.57% (2/7) missing at least 1 record between columns 1 and 3.
4. 2,1: There are 0% (0/7) missing both records between columns 1 and 2.
5. 2,2: There are 71.43% (5) missing values in column 2.
6. 2,3: There are 85.71% (6/7) missing at least 1 record between columns 2 and 3.
7. 3,1: There are 0% (0/7) missing both records between columns 1 and 3.
8. 3,2: There are 14.29% (1/7) missing both records between columns 2 and 3.
9. 3,3: There are 28.57% (2) missing values in column 3.

We want the diagonals to be close to 0, indicating that there are few missing values down a column. We want upper triangular values to be closer to 0 for complete records with both values present. You also want lower triangular values close to 0, indicating that most or all don't have double missing. In this example it's clear that 1,3 and 3,1 are the closest to 0 so it's the most complete field. You can validate by looking at Table 1.

5.2 percent=TRUE, missing=FALSE

```
> missingMatrix(my_df, percent=TRUE, missing=FALSE)
```

```
Data Frame
     col1  col2  col3
col1  100 28.57 71.43
col2  100 28.57 14.29
col3  100 85.71 71.43
```

This is simply 100 - percent missing. Therefore we want all values to be as close to 100% as possible.
Interpretation:

1. 1,1: 100% (all 7) of the records are present in column 1.
2. 1,2: 28.57% (2/7) have records in *both* columns 1 and 2.
3. 1,3: 71.43% (5/7) have records in *both* columns 1 and 3.
4. 2,1: 100% (7/7) have at least 1 record in *either* columns 1 and 2.
5. 2,2: 28.57% (2) records are present in column 2.
6. 2,3: 14.29% (1/7) have records in *both* columns 1 and 3.
7. 3,1: 100% (7/7) have at least 1 record in *either* columns 1 and 3.
8. 3,2: 85.71% (6/7) have at least 1 record in *either* columns 2 and 3.
9. 3,3: 71.43% (5) records are present in column 3.

5.3 percent=FALSE, missing=TRUE

Will give counts of missing values.

```
> missingMatrix(my_df, percent=FALSE, missing=TRUE)
```

```
Data Frame
     col1 col2 col3
col1    0    5    2
col2    0    5    6
col3    0    1    2
```

5.4 percent=FALSE, missing=FALSE

Will give counts of non-missing values.

```
> missingMatrix(my_df, percent=FALSE, missing=FALSE)
```

```
Data Frame
     col1 col2 col3
col1    7    2    5
col2    7    2    1
col3    7    6    5
```