

R Examples Generate Tibble Dataframes

Fan Wang

2022-07-24

Contents

1	Generate Dataframe	1
1.1	Simple Meshed Dataframe Name Columns	1
1.2	Generate Tibble given Matrixes and Arrays	2
1.3	Rename Tibble with Numeric Column Names	3
1.4	Tibble Row and Column and Summarize	3
1.5	Sorting and Rank	4
1.5.1	Sorting	4
1.5.2	Create a Ranking Variable	5
1.6	REconTools Summarize over Tibble	6

1 Generate Dataframe

Go to the [RMD](#), [R](#), [PDF](#), or [HTML](#) version of this file. Go back to [fan's REconTools](#) research support package, [R4Econ](#) examples page, [PkgTestR](#) packaging guide, or [Stat4Econ](#) course page.

1.1 Simple Meshed Dataframe Name Columns

```
# 5 by 3 matrix
mt_rnorm_a <- matrix(rnorm(4,mean=0,sd=1), nrow=5, ncol=3)

# Column Names
ar_st_varnames <- c('id','var1','varb','vartheta')

# Combine to tibble, add name col1, col2, etc.
tb_combine <- as_tibble(mt_rnorm_a) %>%
  rowid_to_column(var = "id") %>%
  rename_all(~c(ar_st_varnames))

# Display
kable(tb_combine) %>% kable_styling_fc()
```

id	var1	varb	vartheta
1	1.8481608	-0.0136192	0.3298700
2	-0.0136192	0.3298700	-1.1755065
3	0.3298700	-1.1755065	1.8481608
4	-1.1755065	1.8481608	-0.0136192
5	1.8481608	-0.0136192	0.3298700

1.2 Generate Tibble given Matrixes and Arrays

Given Arrays and Matrixes, Generate Tibble and Name Variables/Columns

- naming tibble columns
- tibble variable names
- dplyr rename tibble
- dplyr rename tibble all variables
- dplyr rename all columns by index
- dplyr tibble add index column
- see also: [SO-51205520](#)

```
# Base Inputs
ar_col <- c(-1,+1)
mt_rnorm_a <- matrix(rnorm(4,mean=0,sd=1), nrow=2, ncol=2)
mt_rnorm_b <- matrix(rnorm(4,mean=0,sd=1), nrow=2, ncol=4)

# Combine Matrix
mt_combine <- cbind(ar_col, mt_rnorm_a, mt_rnorm_b)
colnames(mt_combine) <- c('ar_col',
                          paste0('matcolvar_grpa_', seq(1,dim(mt_rnorm_a)[2])),
                          paste0('matcolvar_grpb_', seq(1,dim(mt_rnorm_b)[2])))

# Variable Names
ar_st_varnames <- c('var_one',
                   paste0('tibcolvar_ga_', c(1,2)),
                   paste0('tibcolvar_gb_', c(1,2,3,4)))

# Combine to tibble, add name col1, col2, etc.
tb_combine <- as_tibble(mt_combine) %>% rename_all(~c(ar_st_varnames))

# Add an index column to the dataframe, ID column
tb_combine <- tb_combine %>% rowid_to_column(var = "ID")

# Change all gb variable names
tb_combine <- tb_combine %>%
  rename_at(vars(starts_with("tibcolvar_gb_")),
            funs(str_replace(., "_gb_", "_gbrenamed_")))

# Tibble back to matrix
mt_tb_combine_back <- data.matrix(tb_combine)

# Display
kable(mt_combine) %>% kable_styling_fc_wide()
```

ar_col	matcolvar_grpa_1	matcolvar_grpa_2	matcolvar_grpb_1	matcolvar_grpb_2	matcolvar_grpb_3	matcolvar_grpb_4
-1	0.9666212	-0.227653	0.3012623	0.6214326	0.3012623	0.6214326
1	0.7706524	1.192903	0.4534995	-1.0177555	0.4534995	-1.0177555

```
kable(tb_combine) %>% kable_styling_fc_wide()
```

ID	var_one	tibcolvar_ga_1	tibcolvar_ga_2	tibcolvar_gbrenamed_1	tibcolvar_gbrenamed_2	tibcolvar_gbrenamed_3	tibcolvar_gbrenamed_4
1	-1	0.9666212	-0.227653	0.3012623	0.6214326	0.3012623	0.6214326
2	1	0.7706524	1.192903	0.4534995	-1.0177555	0.4534995	-1.0177555

```
kable(mt_tb_combine_back) %>% kable_styling_fc_wide()
```

ID	var_one	tibcolvar_ga_1	tibcolvar_ga_2	tibcolvar_gbrenamed_1	tibcolvar_gbrenamed_2	tibcolvar_gbrenamed_3	tibcolvar_gbrenamed_4
1	-1	0.9666212	-0.227653	0.3012623	0.6214326	0.3012623	0.6214326
2	1	0.7706524	1.192903	0.4534995	-1.0177555	0.4534995	-1.0177555

1.3 Rename Tibble with Numeric Column Names

After reshaping, often could end up with variable names that are all numeric, integers for example, how to rename these variables to add a common prefix for example.

```
# Base Inputs
ar_col <- c(-1,+1)
mt_rnorm_c <- matrix(rnorm(4,mean=0,sd=1), nrow=5, ncol=10)
mt_combine <- cbind(ar_col, mt_rnorm_c)

# Variable Names
ar_it_cols_ctr <- seq(1, dim(mt_rnorm_c)[2])
ar_st_varnames <- c('var_one', ar_it_cols_ctr)

# Combine to tibble, add name col1, col2, etc.
tb_combine <- as_tibble(mt_combine) %>% rename_all(~c(ar_st_varnames))

# Add an index column to the dataframe, ID column
tb_combine_ori <- tb_combine %>% rowid_to_column(var = "ID")

# Change all gb variable names
tb_combine <- tb_combine_ori %>%
  rename_at(
    vars(num_range('1', ar_it_cols_ctr)),
    funs(paste0("rho", ., 'var'))
  )

# Display
kable(tb_combine_ori) %>% kable_styling_fc_wide()
```

ID	var_one	1	2	3	4	5	6	7	8	9	10
1	-1	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216
2	1	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920
3	-1	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975
4	1	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323
5	-1	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216

```
kable(tb_combine) %>% kable_styling_fc_wide()
```

ID	var_one	rho1var	rho2var	rho3var	rho4var	rho5var	rho6var	rho7var	rho8var	rho9var	rho10var
1	-1	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216
2	1	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920
3	-1	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975
4	1	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323
5	-1	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216	0.6994920	1.1923975	-0.1758323	1.2479216

1.4 Tibble Row and Column and Summarize

Show what is in the table: 1, column and row names; 2, contents inside table.

```
tb_iris <- as_tibble(iris)
print(rownames(tb_iris))
```

```
## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15" "16"
## [21] "21" "22" "23" "24" "25" "26" "27" "28" "29" "30" "31" "32" "33" "34" "35" "36"
## [41] "41" "42" "43" "44" "45" "46" "47" "48" "49" "50" "51" "52" "53" "54" "55" "56"
## [61] "61" "62" "63" "64" "65" "66" "67" "68" "69" "70" "71" "72" "73" "74" "75" "76"
## [81] "81" "82" "83" "84" "85" "86" "87" "88" "89" "90" "91" "92" "93" "94" "95" "96"
## [101] "101" "102" "103" "104" "105" "106" "107" "108" "109" "110" "111" "112" "113" "114" "115" "116"
## [121] "121" "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132" "133" "134" "135" "136"
## [141] "141" "142" "143" "144" "145" "146" "147" "148" "149" "150"
```

```
colnames(tb_iris)
```

```
## [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
```

```
colnames(tb_iris)
```

```
## [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
```

```
summary(tb_iris)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100 setosa :50
## 1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300 versicolor:50
## Median :5.800 Median :3.000 Median :4.350 Median :1.300 virginica :50
## Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199
## 3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800
## Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500
```

1.5 Sorting and Rank

1.5.1 Sorting

- dplyr arrange desc reverse
- dplyr sort

```
# Sort in Ascending Order
```

```
tb_iris %>% select(Species, Sepal.Length, everything()) %>%
  arrange(Species, Sepal.Length) %>% head(10) %>%
  kable() %>% kable_styling_fc()
```

Species	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa	4.3	3.0	1.1	0.1
setosa	4.4	2.9	1.4	0.2
setosa	4.4	3.0	1.3	0.2
setosa	4.4	3.2	1.3	0.2
setosa	4.5	2.3	1.3	0.3
setosa	4.6	3.1	1.5	0.2
setosa	4.6	3.4	1.4	0.3
setosa	4.6	3.6	1.0	0.2
setosa	4.6	3.2	1.4	0.2
setosa	4.7	3.2	1.3	0.2

```
# Sort in Descending Order
```

```
tb_iris %>% select(Species, Sepal.Length, everything()) %>%
```

```
arrange(desc(Species), desc(Sepal.Length)) %>% head(10) %>%
kable() %>% kable_styling_fc()
```

Species	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
virginica	7.9	3.8	6.4	2.0
virginica	7.7	3.8	6.7	2.2
virginica	7.7	2.6	6.9	2.3
virginica	7.7	2.8	6.7	2.0
virginica	7.7	3.0	6.1	2.3
virginica	7.6	3.0	6.6	2.1
virginica	7.4	2.8	6.1	1.9
virginica	7.3	2.9	6.3	1.8
virginica	7.2	3.6	6.1	2.5
virginica	7.2	3.2	6.0	1.8

1.5.2 Create a Ranking Variable

We use dplyr's [ranking](#) functions to generate different types of ranking variables.

The example below demonstrates the differences between the functions [row_number\(\)](#), [min_rank\(\)](#), and [dense_rank\(\)](#).

- *row_number*: Given 10 observations, generates index from 1 to 10, ties are given different ranks.
- *min_rank*: Given 10 observations, generates rank where second-rank ties are both given “silver”, and the 4th highest ranked variable not given medal.
- *dense_rank*: Given 10 observations, generates rank where second-rank ties are both given “silver” (2nd rank), and the 4th highest ranked variable is given “bronze” (3rd rank), there are no gaps between ranks.

Note that we have “desc(var_name)” in order to generate the variable based on descending sort of the the “var_name” variable.

```
tb_iris %>%
  select(Species, Sepal.Length) %>% head(10) %>%
  mutate(row_number = row_number(desc(Sepal.Length)),
         min_rank = min_rank(desc(Sepal.Length)),
         dense_rank = dense_rank(desc(Sepal.Length))) %>%
  kable(caption = "Ranking variable") %>% kable_styling_fc()
```

Ranking variable

Species	Sepal.Length	row_number	min_rank	dense_rank
setosa	5.1	2	2	2
setosa	4.9	5	5	4
setosa	4.7	7	7	5
setosa	4.6	8	8	6
setosa	5.0	3	3	3
setosa	5.4	1	1	1
setosa	4.6	9	8	6
setosa	5.0	4	3	3
setosa	4.4	10	10	7
setosa	4.9	6	5	4

1.6 REconTools Summarize over Tible

Use R4Econ's summary tool.

```
df_summ_stats <- ff_summ_percentiles(tb_iris)
kable(t(df_summ_stats)) %>% kable_styling_fc_wide()
```

stats	n	unique	NAobs	ZEROobs	mean	sd	cv	min	p01	p05	p10	p25	p50	p75	p90	p95	p99	max
Petal.Length	150	43	0	0	3.758000	1.7652982	0.4697441	1.0	1.149	1.300	1.4	1.6	4.35	5.1	5.80	6.100	6.700	6.9
Petal.Width	150	22	0	0	1.199333	0.7622377	0.6355511	0.1	0.100	0.200	0.2	0.3	1.30	1.8	2.20	2.300	2.500	2.5
Sepal.Length	150	35	0	0	5.843333	0.8280661	0.1417113	4.3	4.400	4.600	4.8	5.1	5.80	6.4	6.90	7.255	7.700	7.9
Sepal.Width	150	23	0	0	3.057333	0.4358663	0.1425642	2.0	2.200	2.345	2.5	2.8	3.00	3.3	3.61	3.800	4.151	4.4