data_frame_example_4.2

Mike Kozlowski

R Markdown

This is set of basic Examples of working with Data Tables of various types in R. A table is the type of structure that we often see in Excel files, or in data base tables (as implemented in SQL as an example).

In most typical data tables, the rows are observations or records of one event, or person, or measurement. In statistical terms this is an observation on an individual. The columns are then variables recorded for each individuals.

Here is a really simple table

Name Age Income Tax Rate Gender Bob 31 150,000 0.12 M Jane 45 85,000 0.14 F Larry 53 62,000 0.18 M

Notice that we have 4 types of data here, a text string (name), Age (integer), Income and Tax rate (floating point) and Gender

We could enter these into a data.frame in R, this is the “stock” way of handling this type of data

```
Name=c("Bob","Jane","Larry")
Age=c(31,45,53)
Income=c(150000,85000,62000)
TaxRate=c(0.12,0.14,0.18)
Gender=as.factor(c("M","F","M"))
QTable1=data.frame(Name=Name,Age=Age,Income=Income,TaxRate=TaxRate,Gender=Gender)
```

We have some different printing options in RMarkDown, which gives us a nice looking result, with labelling

```
QTable1
```

```
##    Name Age Income TaxRate Gender
## 1   Bob  31 150000    0.12      M
## 2  Jane  45  85000    0.14      F
## 3 Larry  53  62000    0.18      M
```

There is also a tibble data format, which is an updated version of a data table. The tibble class is in the tibble package or the tidyverse package. It fixes some of the quirks of the data.frame structure.

Have a look at

https://cran.r-project.org/web/packages/tibble/README.html (https://cran.r-project.org/web/packages/tibble/README.html)

for more info on tibbles

```
require(tibble)
```

```
## Loading required package: tibble
```

```
myParabola=tibble(x=1:5, y=x^2)
print(myParabola)
```

```
## # A tibble: 5 × 2
##       x     y
##   <int> <dbl>
## 1     1     1
## 2     2     4
## 3     3     9
## 4     4    16
## 5     5    25
```

```
QTable2=tibble(Name=Name,Age=Age,Income=Income,TaxRate=TaxRate,Gender=Gender)
print(QTable2)
```

```
## # A tibble: 3 × 5
##   Name    Age Income TaxRate Gender
##   <chr> <dbl>  <dbl>   <dbl> <fct>
## 1 Bob      31 150000    0.12 M
## 2 Jane     45  85000    0.14 F
## 3 Larry    53  62000    0.18 M
```

Matrices

In R, we also may need to use matrices of data, which have the requirements that the values in the matrices, the contents of the cells of the matrix, all have the same type, typically floating point, but not always.

Many linear algebra methods require the matrix format.

```
z=matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,ncol=3,byrow=TRUE)
print(z)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

We can convert using as.matrix and as.data.frame

```
zd=as.data.frame(z)
zd
```

```
##   V1 V2 V3
## 1  1  2  3
## 2  4  5  6
## 3  7  8  9
```

we can add row and column names or labels to a data.frame or a matrix

```
colnames(zd)<-c("A","B","C")
rownames(zd)<-c("Doc","Bashful","Sleepy")
z
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

Loading tabular data from a CSV file into R

This is done using the read.table() function or the read.csv() function- either works fine. There are a lot of commands related to whether or not the input file has row and column headers or not, run help(read.table) to see what options are available.

I often use the file.choose() function with the read.table() or read.csv function, to allow me to browse to locate the file I want to load, here's what this looks like, loading a file called Tows.csv on the City of Buffalo Automobile tows

I got from the Open Data Buffal website at https://data.buffalony.gov/ (https://data.buffalony.gov/)

Initially, I tried opening this with read.table()

my_Tows=read.table(file.choose(),header=TRUE,sep=",",stringsAsFactors=FALSE,skipNul=TRUE)

but this wouldn't run, due to missing data on line 5297 (and probably other places as well!)

read.csv did work

my_Tows=read.csv(file.choose(),header=TRUE,sep=",",stringsAsFactors=FALSE,skipNul=TRUE)

{note: to get RMarkDown to create a result file, I had to enter the file name here}

```
my_Tows=read.csv("C:\\Users\\Mike\\Documents\\DAT511\\1-25 class\\Tows.csv",header=TRUE,sep
=",",stringsAsFactors=FALSE,skipNul=TRUE)
```

We can see what we have in the file using str(my_Tows)

```
str(my_Tows)
```

```
## 'data.frame':    70879 obs. of  22 variables:
##  $ UNIQUE.KEY        : chr  "201011014DPE" "20150129600SBA" "201105017DPE" "2015022324RIV"
...
##  $ TOW.DATE          : chr  "11/01/2010" "01/29/2015" "05/01/2011" "02/23/2015" ...
##  $ AGENCY            : chr  "DPE" "SBA" "DPE" "RIV" ...
##  $ VEHICLE.YEAR      : int  1996 0 98 2004 1994 2010 1885 2000 2000 91 ...
##  $ VEHICLE.MAKE      : chr  "PLYM" "FORD" "JEEP" "JEEP" ...
##  $ LICENSE.PLATE     : chr  "" "" "" "" ...
##  $ LICENSE.STATE     : chr  "NY" "NY" "NY" "NY" ...
##  $ TOW.LOCATION      : chr  "693 EAST FERRY" "ECMC RAMP" "ALLEN" "80 COLLAGE" ...
##  $ CITY              : chr  "Buffalo" "Buffalo" "Buffalo" "Buffalo" ...
##  $ STATE             : chr  "NY" "NY" "NY" "NY" ...
##  $ ZIP               : chr  "14211" "" "" "14201" ...
##  $ TOW.REASON        : chr  "IL" "IM" "AI" "IL" ...
##  $ TOW.DESCRIPTION   : chr  "ILLEGAL VEHICLE" "IMPOUNDED" "ACCIDENT" "ILLEGAL VEHICLE" ...
##  $ LOCATION          : chr  "(42.91453442930668, -78.83628415637776)" "" "" "(42.899120131
819565, -78.87898678325882)" ...
##  $ LATITUDE          : num  42.9 NA NA 42.9 42.9 ...
##  $ LONGITUDE         : num  -78.8 NA NA -78.9 -78.9 ...
##  $ COUNCIL.DISTRICT  : chr  "MASTEN" "" "" "FILLMORE" ...
##  $ POLICE.DISTRICT   : chr  "District C" "" "" "District B" ...
##  $ CENSUS.TRACT      : chr  "34" "" "" "68" ...
##  $ CENSUS.BLOCK.GROUP: chr  "4" "" "" "3" ...
##  $ CENSUS.BLOCK      : chr  "4001" "" "" "3005" ...
##  $ NEIGHBORHOOD      : chr  "UNKNOWN" "UNKNOWN" "UNKNOWN" "UNKNOWN" ...
```