



Data manipulation in R

A program to use when size matters

Peter Shaw

September 19, 2015



Why not use a spreadsheet?

- Data manipulation in Excel is VERY risk and time consuming
- A rage of software packages are available for Excel
- Large data sets can exceed the size limits of standard programs
- Spreadsheets don't have the inherent understanding of statistics that R has
- For example handling of NA's
- R is hot!



Why use R?

- Its free
- Its available on most operating systems Windows, OS X, Linux
- There are huge numbers of packages available
- Its becoming the international standard for statistics



- James P. Howard.
 R Cookbook.
 O'Reilly Media, Inc, 2011.
- Phil Spector.
 Data Manipulation with R.
 Use R series
 Springer, 2008



Download it

- Open http://www.r-project.org
- Click CRAN (Under download on Top Left)
- Click http://cran.ms.unimelb.edu.au/ University of Melbourne

Windows

- Select Windows
- Select Base
- Download R (suggest latest version)

OS X

- Select Select OS X
- Select R-3.2.2.pkg (or the version that matches your OS version)



How about RStudio

https://www.rstudio.com/products/rstudio/download/



```
2+5
## [1] 7
# Create a sequence of numbers
X = 2:10
# Display basic statistical measures
summary(X)
##
     Min. 1st Qu. Median Mean 3rd Qu. Max.
##
                        6
                                              10
# use q() to quit
```





To access the documentation type

help.start()
help(summary)
args(summary)
example(sd)





To search R documentation

RSiteSearch("key phrase")

Custom Google search focused on R-specific websites

http://rseek.org

Coding Q&A site

http://stackoverflow.com http://stats.stakexchange.com



Some discussion recently on how to work creatively. Research out of successful R&D projects developed into Agile

- Keep the manages away
- work sustainably
- people over process
- iterative development

Lists

- I = c(1,3,4)
- bbb



Where are we

```
getwd()
setwd("/Users/pcru")
dir() #This lists the files
ls() #This lists the variables
```

http://www.statmethods.net/input/contents.html



To read a csv table as a table try

 $\texttt{tab1} \leftarrow \textbf{as.matrix} (\texttt{read.csv} (\texttt{file} = \texttt{"filetable.csv"}, \texttt{sep} = \texttt{","}, \texttt{header} = \texttt{FALSE}))$

But our table is an excel file

- What about a package?
- http://www.thertrader.com/2014/02/11/a-million-ways-to-connect-r-and-excel/
- Lets use the R package xlsx



Where from

- install command
- install.packages(pkgs)

Citing Packages

https://cran.r-project.org/web/packages/RefManageR/vignettes/TestRmd.html



```
x<-citation()
toBibtex(x)
## @Manual{,
##
    title = {R: A Language and Environment for Statistical Computing
     author = {{R Core Team}},
##
     organization = {R Foundation for Statistical Computing},
##
     address = {Vienna, Austria},
##
    year = \{2014\},\
##
     url = {http://www.R-project.org/},
##
## }
```



```
table 1 \leftarrow \textbf{read} . \times ls \times 2 \ ("1\_R \ Wkshp\_dummy \ data\_OTU \ table . \times ls \times ", \ sheetName = "Sheet1", header=FALSE, rowNames=FALSE, transpose=TRUE, endRow=18)
```

```
## Loading required package: xlsx
## Warning: package 'xlsx' was built under R
version 3.1.3
## Loading required package: rJava
## Warning: package 'rJava' was built under R
version 3.1.3
## Loading required package: methods
## Loading required package: xlsxjars
## Loading required package: xtable
```





	X1	X2	X3	X4	X5	X6	X7
1	Group	Contaminated					
2	Site	1			2		
3	Sample ID	10000	10001	10002	10003	10004	10005
4	Rep	1	2	3	1	2	3
5	phormidiaceae	24872	24872	5822	7538	7201	7538
6	streptococcaceae	11	7	14	8	10	8





Transposing

We need to transpose the table and set the column names correctly

```
table1t=setNames(data.frame(t(table1[,-1])),table1[,1])
ctridx<-which(table1t$Group=="Control")
table1t$Group[1:48]<-"Contaminated"
table1t$Group[(ctridx+1):48]<-"Control"</pre>
```



```
ttt—table1t$Site
for(i in c(2:length(table1t$Site)))
{
temp—as.character(table1t$Site[i])
tempb—as.character(ttt[i-1])
if(table1t$Site[i]=="")
{
   ttt[i]—tempb
   }
if(!table1t$Site[(i)]=="")
{
   ttt[i]—temp
}
}
table1t$Site—ttt
```

```
## X3
## 1
```

Levels: 1 2 3 4 FALSE TRUE

X4 ## 1

Levels: 1 2 3 4 FALSE TRUE

X5 ## 2

Levels: 1 2 3 4 FALSE TRUE

X6

2

Levels: 1 2 3 4 FALSE TRUE

X7

2

Levels: 1 2 3 4 FALSE TRUE

X8



- http://www.statmethods.net/input/importingdata.html
- Input files from Stata

```
library (foreign )
mydata ← read.dta("c:/mydata.dta")
```



```
setwd("/Users/pcru/SizeDoesMatter1")
#dir()
```

table2<-read.xlsx2("2_R Wkshp_dummy data_Env Data.xlsx", sheetName ="</pre>

	Group	Site	Sample.ID	Rep	Spill.date
1	Contaminated	1	10000	1	14-May-14
2	Contaminated	1	10001	2	14-May-14
3	Contaminated	1	10002	3	14-May-14
4	Contaminated	2	10003	1	14-May-14
5	Contaminated	2	10004	2	14-May-14
6	Contaminated	2	10005	3	14-May-14





Oh NO All columns have been set to factors

lets break it down

First lets reed a few rows only

table2<-read.xlsx2("2_R Wkshp_dummy data_Env Data.xlsx", sheetName =
sapply(table2,mode)</pre>

```
Site
                                  Sample.ID
                                                      Rep
##
          Group
                                                             Spill.d
    "character"
                    "numeric"
                                  "numeric" "character"
                                                           "charact
##
##
       rowNames as.Data.frame
##
       "logical"
                "logical"
```

sapply(table2,class)

"logical"

```
## Group Site Sample.ID Rep Spill.d
## "character" "numeric" "character" "charact
## rowNames as.Data.frame
```

"logical"

##



colClasses

- The variable colClasses can be used to specify the row types.
- We need to set stringsAsFactor=FALSE or all columns with be loaded as factors
- The dates are in a non standard format so we need to read them as chars first

```
table2b<-read.xlsx2("2_R Wkshp_dummy data_Env Data.xlsx", sheetName =
sapply(table2,class)
##
           Group
                          Site
                                    Sample.ID
                                                         Rep
                                                                Spill.d
                     "numeric"
                                    "numeric" "character"
##
     "character"
                                                               "charact
        rowNames as.Data.frame
##
##
       "logical"
                     "logical"
```





```
table2f<-table2
table2f$Spill.date<-as.Date(table2f$Spill.date,"%d-%b-%y")
table2f$Sample.collection.date<-as.Date(table2f$Sample.collection.dat
## Error in
as.Date.default(table2f$Sample.collection.date,
"%d.%m.%y"): do not know how to convert
'table2f$Sample.collection.date' to class "Date"
#sapply(table2f, mode)
sapply(table2f,class)
##
           Group
                          Site
                                   Sample.ID
                                                        Rep
                                                               Spill.d
```

```
"character"
                "numeric"
                                  "numeric"
                                             "character"
##
                                                                 "Da
##
       rowNames as.Data.frame
```

##

"logical" "logical"

colClasses

- The as.Data method can take a format string as the second variable
- The format strings are described in help on strptime
- But Spill.data has two formats
- We can use the if else function to combine them

table2bf<-table2b

table2bf\$Spill.date<-as.Date(table2bf\$Spill.date,"%d-%b-%y")

#table2bf£Sample.collection.date<-as.Date(table2bf£Sample.collectioncdate1<-as.Date(table2bf£Sample.collection.date,"%d.%m.%y")

cdate2<-as.Date(table2bf\$Sample.collection.date,"%d/%m/%y")

table2bf\$Sample.collection.date<-as.Date(ifelse(!is.na(cdate1),as.Dat
table2bf\$Group<-as.factor(table2bf\$Group)</pre>

table2bf\$Rep<-as.factor(table2bf\$Rep)

na_count <-sapply(table2bf, function(y) sum(length(which(is.na(y)))))</pre>

na_count
Group Site Sample.

0 0 0 ## Rep Spill.date Sample.collection.da ## 0 24

labnum phosphate..ppb. ammonia..pp



- require(stringer)
- \blacksquare $stri_c(str1, str2)$ concatenates two string
- \blacksquare $str_len(str)$

: Date, format: "2014-05-14" "2014-05-14"

```
require(stringr)
table2bf$Rep<-str_replace(table2bf$Rep, "[rep]{3}?", "\\1")
table2bf$Rep<-str_replace(table2bf$Rep, "A", "1")
table2bf$Rep<-str_replace(table2bf$Rep, "B", "2")
table2bf$Rep<-str_replace(table2bf$Rep, "C", "3")
table2bf$Rep<-as.factor(table2bf$Rep)</pre>
str(table2bf)
   'data frame': 48 obs. of 13 variables:
                              : Factor w/ 2 levels "Contaminated",..: 1
##
    $ Group
    $ Site
                                     1 1 1 2 2 2 1 1 1 2 ...
##
                                     10000 10001 10002 10003 10004 ...
##
    $ Sample.ID
    $ Rep
                              : Factor w/ 3 levels "1", "2", "3": 1 2 3 1
##
```

\$ Sample.collection.date: Date, format: "2014-05-15" "2014-05-15"

\$ labnum : num 2000 2001 2002 2003 2004 ...

\$ Spill.date

##

The inbuilt command merge

- R has a command merge
- Lets start looking at the first 9 lines of the tables and merge them using the Sample ID
- Because otherwise its not uniques

```
\begin{split} \text{merge}(x,\ y,\ by = intersect(names(x),\ names(y)), \\ by.x = by,\ by.y = by,\ all = FALSE,\ all.x = all,\ all.y = all, \\ sort = TRUE,\ suffixes = c(".x",".y"), \\ incomparables = NULL,\ \ldots) \end{split}
```

```
tab1c<-table1t[1:9,]
tab2c<-table2b[1:9,]
m1<-merge(tab1c,tab2c,by.x="Sample ID",by.y="Sample.ID")</pre>
```



reshape2

vignette(reshape)

using RSQLite

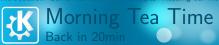






Another important component of TDD is refactoring and unit tests

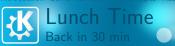
- Refactoring http://refactoring.com/
- http://www.r-bloggers.com/my-experience-of-learning-rfrom-basic-graphs-to-performance-tuning/
- TDD in R http://www.slideserve.com/andrew/test-driven-development-in-r
- Version Control tortiseSVN http://tortoisesvn.net/
- GitHub https://github.com/



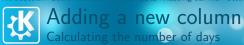
Need coffee

SQLite

dd



Provided



Using the is. Date command









Your feedback on some ideas

- Using Sweave or Knitr
- Advanced Data Cleaning
- Network Centric data analysis



■ LaTeX Beamer

http://latex-beamer.sourceforge.net/

Sharelatex Site

https://www.sharelatex.com

A Data Cleaning Mooc

https://www.sharelatex.com





The chunk below will not be printed