



# Data manipulation in R

A program to use when size matters

Peter Shaw

September 29, 2015



# Why use R?

Why not use a spreadsheet?



## Today's workshop

- A common scenario
- A friend has emailed you her data in a spreadsheet
- Today's workshop is not about impressing with R code

## Why not use a spreadsheet?

- Data manipulation in Excel is VERY risk and time consuming
- A range 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?



### 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



# Getting Started

Some References



**James P. Howard.**

*R Cookbook.*

O'Reilly Media, Inc, 2011.



**Phil Spector.**

*Data Manipulation with R.*

Use R series

Springer, 2008



# Getting Started

## Installing R!



### 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)



# Getting Started

## Installing a GUI



### How about RStudio

- <https://www.rstudio.com/products/rstudio/download/>
- Its also on your thumb drive



# Getting Started

## Basic steps



```
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.
##         2         4         6         6         8        10
```

```
# use q() to quit
```



# Getting Started

## Help Functions



### To access the documentation type

```
help.start()  
help(summary)  
args(summary)  
example(sd)  
??package
```





# Help Functions

Search the Web



## To search R documentation

- `RSiteSearch("key phrase")`
- `help(adf.test, package="tseries")`
- To search for a tutorial for a package  
`vignette(package="packagename")`
- For an intro to vignettes see  
<https://cran.r-project.org/web/packages/sos/vignettes/sos.pdf>
- Examples on the web  
<http://shiny.rstudio.com/gallery/>

## Custom Google search focused on R-specific websites

<http://rseek.org>

## Coding Q&A site

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



Research on how to work creatively based on case studies of successful R&D projects developed into Agile

- Keep the managers away
- Work sustainably
- People over process
- Iterative development



# R Data types

Lists, frames and tables



## Vectors

- Vectors  $l \leftarrow c(1, 3, 4, 7, 11)$
- Refer to elements using array  $l[c(2, 5)]$  2nd and 5th elements of  $l$

## Data Frames

```
a <- c(35,23,24,65)
e <- c("Peter", "John", "Mark", NA)
f <- c(TRUE,TRUE,TRUE,FALSE)
team <- data.frame(a,e,f)
names(team) <- c("Age","Names","Passed") # variable names
str(team)

## 'data.frame': 4 obs. of 3 variables:
## $ Age : num 35 23 24 65
## $ Names : Factor w/ 3 levels "John","Mark",...: 3 1 2 NA
## $ Passed: logi TRUE TRUE TRUE FALSE
```



## Lets read the table

Check the current directory



### Where are we

```
getwd()  
setwd("/Users/pcru/SizeDoesMatter1")  
dir() #This lists the files  
ls() #This lists the variables
```

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



# Lets read the table

## Reading a table



### To read a csv table as a table try

```
tab1 <- as.matrix(read.csv(file="filetable.csv", sep="," , header=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

- Citing packages
- Getting the bibtex entry into endnote
- [http://www.lib.uts.edu.au/question/5955/  
how-can-i-import-bibliography-endnote-bibtex-latex-what-about-converting-othe](http://www.lib.uts.edu.au/question/5955/how-can-i-import-bibliography-endnote-bibtex-latex-what-about-converting-othe)

```
x←citation()
xl←citation(package="RSQLite")
toBibtex(x)

sessionInfo()
packages_in_use ← c( sessionInfo()$basePkgs, names( sessionInfo()$loadedOnly ) )
the_citations_list ← lapply( X=packages_in_use, FUN=citation )
the_citations_list
```



# Lets read the table

An example



```
table1<-read.xlsx2("1-R Wkshp_dummy data_OTU table.xlsx", sheetName =  
"Sheet1", header=FALSE, rowNames=FALSE, transpose=TRUE, endRow=18)
```

## Loading the xlsx package

```
## 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
```



# Lets read the table

The columns are wrong

	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





# Lets read the table

## Transpose the table



### Transposing

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

```
table1t=setNames(data.frame(t(table1[,-1])),table1[,1])
```

[http://rgm3.lab.nig.ac.jp/RGM/R\\_rdfile?f=Ecdat/man/read.transpose.Rd&d=R\\_CC](http://rgm3.lab.nig.ac.jp/RGM/R_rdfile?f=Ecdat/man/read.transpose.Rd&d=R_CC)

<http://stackoverflow.com/questions/17288197/>

reading-a-csv-file-organized-horizontally



# Fields across many columns

Replicating first column



Lets do it the easy way first

```
ctridx<-which(table1t$Group=="Control")
table1t$Group[1:48]<-"Contaminated"
table1t$Group[(ctridx+1):48]<-"Control"
```

```
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
```



# How to work with strings

## stringr package



- `require(stringr)`

Lets look at this package

- `stri_c(str1, str2)`

concatenates two string

- `str_len(str)`

```
require(stringr)
```

```
## Loading required package: stringr
```

```
table1t$Rep<-str_replace(table1t$Rep,"[rep]{3}?", "\\1")
table1t$Rep<-str_replace(table1t$Rep,"A","1")
table1t$Rep<-str_replace(table1t$Rep,"B","2")
table1t$Rep<-str_replace(table1t$Rep,"C","3")
table1t$Rep<-as.factor(table1t$Rep)
```



# Lets read the table

## Reading a table of other types



- <http://www.statmethods.net/input/importingdata.html>
- <http://stackoverflow.com/questions/17288197/reading-a-csv-file-organized-horizontally>
- [http://rgm3.lab.nig.ac.jp/RGM/R\\_rdfile?f=Ecdat/man/read.transpose.Rd&d=R\\_CC](http://rgm3.lab.nig.ac.jp/RGM/R_rdfile?f=Ecdat/man/read.transpose.Rd&d=R_CC)
- Input files from Stata

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



# Morning Tea Time

Back in 20min



**Need coffee !!**



# Lets read the next table

Reading a table using xlsx



```
setwd("/Users/pcru/SizeDoesMatter1")  
#dir()  
table2<-read.xlsx2("2_R Wkshp_dummy data_Env Data_incl2outliersMK.xlsx", sheetName)
```

	Group	Site	Sample.ID	Rep	Spill.date	Sample.collection.date
1	Contaminated	1	10000	1	14-May-14	15.5.14
2	Contaminated	1	10001	2	14-May-14	15.5.14
3	Contaminated	1	10002	3	14-May-14	15.5.14
4	Contaminated	2	10003	1	14-May-14	15.5.14
5	Contaminated	2	10004	2	14-May-14	15.5.14
6	Contaminated	2	10005	3	14-May-14	15.5.14



# Lets read the next table

Reading a table



Oh NO

- All columns have been set to factors
- Dates have different formats

```
str(table2[,1:11])
```

```
## 'data.frame': 48 obs. of 11 variables:
```

```
## $ Group           : Factor w/ 2 levels "Contaminated",...: 1 1 1 1 1 1 1
## $ Site            : Factor w/ 4 levels "1","2","3","4": 1 1 1 2 2 2 1 1
## $ Sample.ID       : Factor w/ 18 levels "10000","10001",...: 1 2 3 4 5 6
## $ Rep             : Factor w/ 9 levels "1","2","3","A",...: 1 2 3 1 2 3
## $ Spill.date       : Factor w/ 2 levels "14-May-14","N/A": 1 1 1 1 1 1 1
## $ Sample.collection.date: Factor w/ 4 levels "15.5.14","17/5/14",...: 1 1 1 1
## $ labnum           : Factor w/ 36 levels "2000","2001",...: 1 2 3 4 5 6 7
## $ phosphate..ppb.   : Factor w/ 39 levels "10","105","108",...: 27 30 28 20
## $ ammonia..ppb.     : Factor w/ 41 levels "10","103","1042",...: 10 14 15 6
## $ chlorophyll..ug.L. : Factor w/ 38 levels "1","10","11",...: 20 23 21 25 1
## $ DO...            : Factor w/ 31 levels "100","120","31",...: 5 4 3 7 6
```

## Lets break it down

First lets reed a few rows only

```
table2<-read.xlsx2("2_R Wkshp_dummy data_Env Data_incl2outliersMK.xlsx", sheetName=
sapply(table2,mode))
```



# Lets read the next table

Setting the data types



## colClasses

- The variable `colClasses` can be used to specify the row types.
- We need to set **`stringsAsFactor=FALSE`** or all columns will 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_incl2outliersMK.xlsx", sheetName="Data",
  as.is=TRUE, colClasses=colClasses,
  apply=apply(table2,colClasses))
```

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





# Lets read the next table

Setting the Date Type



```
table2f<-table2
table2f$Spill.date<-as.Date(table2f$Spill.date,"%d-%b-%y")
table2f$Sample.collection.date<-as.Date(table2f$Sample.collection.date,"%d.%m.%y")

## 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.date
## "character"      "numeric"    "numeric"      "character"    "Date"
##      rowNames as.Data.frame
##      "logical"      "logical"
```



# Lets read the next table

Setting the Date Type



## colClasses

- The `as.Date` 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")
cdate1<-as.Date(table2bf$Sample.collection.date,"%d.%m.%y")
cdate2<-as.Date(table2bf$Sample.collection.date,"%d/%m/%y")
table2bf$Sample.collection.date<-as.Date(ifelse(!is.na(cdate1),as.Date(cdate1),as
table2bf$Group<-as.factor(table2bf$Group)
table2bf$Rep<-as.factor(table2bf$Rep)
na_count <-sapply(table2bf, function(y) sum(length(which(is.na(y)))))
na_count
```

##	Group	Site	Sample.ID
##	0	0	0
##	Rep	Spill.date	Sample.collection.date
##	0	24	0
##	labnum	phosphate..ppb.	ammonia..ppb.
##	0	0	0
##	chlorophyll..ug.L.	DO...	rowNames
##	0	0	0



# Lets read the next table

Just fix the Rep column using the stringr package again



```
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)
str(table2bf)

## 'data.frame': 48 obs. of 13 variables:
## $ Group : Factor w/ 2 levels "Contaminated",...: 1 1 1 1 1 1 1
## $ Site : num 1 1 1 2 2 2 1 1 1 2 ...
## $ Sample.ID : num 10000 10001 10002 10003 10004 ...
## $ Rep : Factor w/ 3 levels "1","2","3": 1 2 3 1 2 3 1 2 3 1
## $ Spill.date : Date, format: "2014-05-14" "2014-05-14" ...
## $ Sample.collection.date: Date, format: "2014-05-15" "2014-05-15" ...
## $ labnum : num 2000 2001 2002 2003 2004 ...
## $ phosphate..ppb. : num 3020 3253 3169 2999 2879 ...
## $ ammonia..ppb. : num 13880 14598 14676 10984 11657 ...
## $ chlorophyll..ug.L. : num 302 323 315 352 289 296 254 248 250 220 ...
## $ DO... : num 34 33 31 38 36 34 40 38 41 45 ...
## $ rowNames : logi FALSE FALSE FALSE FALSE FALSE FALSE ...
## $ as.Data.frame : logi FALSE FALSE FALSE FALSE FALSE FALSE ...
```



# How to I merge two data sets

## Using the merge command



### 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

```
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, ...)
```

```
tab1c<-table1t[1:9,]
tab2c<-table2b[1:9,]
m1<-merge(tab1c,tab2c,by.x="Sample ID",by.y="Sample.ID")
m2<-merge(table1t,table2bf,by.x=c("Group","Site","Sample ID"),by.y=c("Group","Site","Sample ID"))
m3<-merge(table1t,table2bf,by.x=c("Group","Site","Sample ID","Rep"),by.y=c("Group","Site","Sample ID","Rep"))
```



Lunch Time  
Back in 30 min



Provided



# How to I append two data sets

Lets load a third data set



## Follow up data from contaminated site

```
table3f←read.xls×2("3_Follow up data from contaminated site_MK.xlsx", sheetName ="Sheet1", head
table3f←table3
table3f$Spill.date←as.Date(table3f$Spill.date,"%d.%m.%y")
table3f$Sample.collection.date←as.Date(table3f$Sample.collection.date,"%d.%m.%y")
sapply(table3f,mode)
sapply(table3f,class)
```



# How to append two data sets

Lets load a third data set



## Joining table 3 to are merged tables

- We need to be careful to match everything
- Install the plyr package This has lots of useful functions for renaming var etc
- This means we need columns for corynebacteriaceae and porphyromonadaceae
- should these be NA or 0
- we will do one of each. generally we would use NA but in this case 0 is perhaps better

```
require(plyr)
Sample.ID←rep(20000,3)
table3fi←cbind(table3f, Sample.ID)
#how many columns I can't count
ncol(table3fi)
ncol(m3)
#now get the cols all right
table3fii←table3fi[c(1,2,24,3,4:23)]
m3i←m3[c(1:4,19:20,5:18,21:26)]
setdiff(names(m3i),names(table3fii))
m3ii←rename(m3i,c("Sample.ID"="Sample.ID"))
corynebacteriaceae←rep(0,nrow(table3fii))
porphyromonadaceae←rep(NA,nrow(table3fii))
table3fiii←cbind(table3fii, corynebacteriaceae, porphyromonadaceae)
setdiff(names(m3ii),names(table3fiii))

m3ii[,c(7:24)] ← sapply(m3ii[,c(7:24)], as.numeric)
m3ii[,c(1:4)] ←sapply(m3ii[,c(1:4)], as.character)
#m3ii[,c("Site")] ←sapply(m3ii[,c("Site")], as.character)

table3fiii[,c(1:4)] ← sapply(table3fiii[,c(1:4)], as.character)
table3fiii[,c(7:24)] ← sapply(table3fiii[,c(7:24)], as.numeric)
table4←rbind(m3ii, table3fiii)
table4[,1] ← sapply(table4[,1], as.factor)
```



## Another Break







# Reshaping Tables

## reshape2



### reshape2

- `vignette(reshape)` doesn't work
- try <http://had.co.nz/reshape/>
- and <http://seananderson.ca/2013/10/19/reshape.html>

### A small example for melt

- Suppose we want a box plot to see if there are outliers
- We will use `ggplot2` box plot
- but box plot needs data in long format to use this
- first melt the data
- We need to specify the unique key, the variable name and the value name
- The key is not unique.
- Then plot it



# Reshaping Tables

melt and boxplot



## The code

```
matable4<-melt(table4[,c(1:4,6:25)], variable.name = "microbe", value.name ="abundance", \\
id=c("Group", "Site", "Sample.ID", "Rep"), factorsAsStrings=FALSE, rm.na=TRUE)
```

```
require(reshape2)
```

```
## Loading required package: reshape2
```

```
matable4<-melt(table4[,c(1:4,7:25)], variable.name = "microbe", value.name ="abundanc
```



# Reshaping Tables

boxplot



## Using ggplot

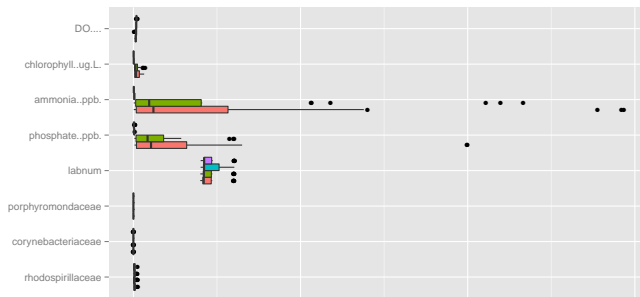
- As we have keys we need to specify the x and y
- lets make the sites different colors
- The variable names are long so lets flit it with `coord_flip()`
- Looks like we have outliers...hmm

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
ggplot(matable4,aes(x=microbe,y=abundance,fill=Site)) + geom_boxplot() + coord_flip()
```

```
## Warning: Removed 24 rows containing non-finite values (stat_boxplot).
```





# Reshaping Tables

boxplot



## Removing Outliers

- Outliers are defined 1.5 times the interquartile range above the upper quartile
- lets fix the one with phosphate as the 9 is typed twice
- Still issues with ammonia to explore

```
phosphate<-table4[, "phosphate..ppb."]
upper.limit <- quantile(phosphate)[4] + 1.5*IQR(phosphate)
lower.limit <- quantile(phosphate)[2] - 1.5*IQR(phosphate)
#table4[phosphate> upper.limit, c("Site", "phosphate..ppb." )]
```

	Site	phosphate..ppb.
1	1	3020.00
2	1	3253.00
3	1	3169.00
12	1	9982.00
14	1	9982.00
16	1	1542.00

```
table4[12, "phosphate..ppb."] <- -982
table4[14, "phosphate..ppb."] <- -982
```



# Outliers check

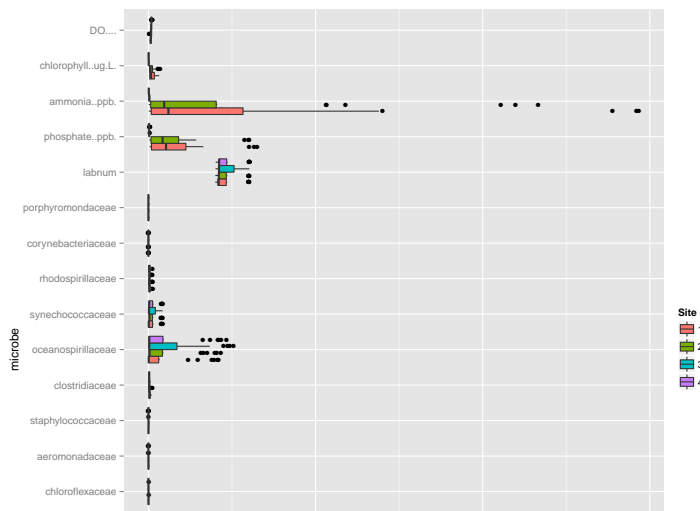
Redo the boxplot



## Look again ggplot

```
matable4<-melt(table4[,c(1:4,7:25)],variable.name = "microbe",value.name = "abundance")
ggplot(matable4,aes(x=microbe,y=abundance,fill=Site)) + geom_boxplot() + coord_flip()

## Warning: Removed 24 rows containing non-finite values (stat_boxplot).
```





## RSQLite

- Suppose merge is not enough? I know about SQL and want to do joins
- Lets Install RSQLite
- We also need to install DBI

```
## Loading required package: RSQLite
```

```
db <- dbConnect(SQLite(), dbname="Test.sqlite")
#getConfig()$staged.queries
# sqldf(attach "Test1.sqlite" as new)
dbBegin(db)
```

```
## [1] TRUE
```

```
dbWriteTable(db, "table1", table1t, overwrite=TRUE)
```

```
## [1] TRUE
```

```
dbReadTable(db, "table1")
```

```
##           Group Site Sample.ID Rep phormidiaceae
## X2      Contaminated    1   10000    1      24872
## X3      Contaminated    1   10001    2      24872
## X4      Contaminated    1   10002    3       5822
```



Another important component of TDD is refactoring and unit tests

- Refactoring <http://refactoring.com/>
- <http://www.r-bloggers.com/my-experience-of-learning-r-from-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/>



# Dropping row and columns

Dropping selected variables



## Dropping Row and Columns with too many NAs

```
numNAs_inData4_rows <- apply(rawData4, 1, function(z) sum(is.na(z)))
numNAs_inData4_col <- apply(table4, 2, function(z) sum(is.na(z))) # count NAs in Data4
lessThan20 <- table4[!(numNAs_inData4_rows > 20),] #only select the rows contain less Than
lessThan20col <- table4[,!(numNAs_inData4_col > 20)]
```





# Dropping row and columns

Dropping selected variables



## Tidy Data

In tidy data:

- Each variable forms a column.
- Each observation forms a row.
- Each type of observational unit forms a table.
- <https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html>
- <http://pj.freefaculty.org/R/Rtips.html#toc-Subsection-1.11>

## Spit out the dates and numbers

```
dates4←table4[,c(5,6)]  
abundance←table4[,c(7:25)]
```



## Adding a new column

Calculating the number of days



### Calculating the number of days

We can just subtract as.Date fields

```
dates4<-table4[,c(5,6)]  
abundance<-table4[,c(7:25)]  
days<-dates4[,2]-dates4[,1]
```



# Setting the Relative abundance

## Normalizing data



### sapply

- Also known as centring the data
- Ecological percentage of the sum of the variables
- We can use sweep to centre the data
- `options(digits = 1)` Just to make things pretty

```
sweepOutContinu←sweep(abundance,2,apply(abundance,2,min,na.rm=TRUE))
afterSweepContinu←sweep(sweepOutContinu,2,apply(sweepOutContinu,2,max,na.rm=TRUE),"/")
table5←cbind(table4[,c(1:6)],afterSweepContinu,days)
options(digits=1)
sweep(abundance, 2, colSums(abundance), FUN="/")
scale(abundance, center=FALSE, scale=colSums(abundance))
```



# Now lets have some fun

## Graphics in R



### R has nice graphs

- Lets create a correlation heat map
- Heat map.
- A graphical output
- <http://rcharts.io/gallery/>
- R Graph gallery currently down try <http://rgraphgallery.blogspot.com/>
- A reference on where to go R thumbnails
- ggplot2 (scatter plot of 2 var and then 3 plots)

```
library(corrplot)
abuncor<-cor(t5lessThan20col[,c(6:22)])
require(corrplot)
corrplot(abuncor, method = "circle")
```

```
## [1] 23
```

```
## Loading required package: corrplot
```



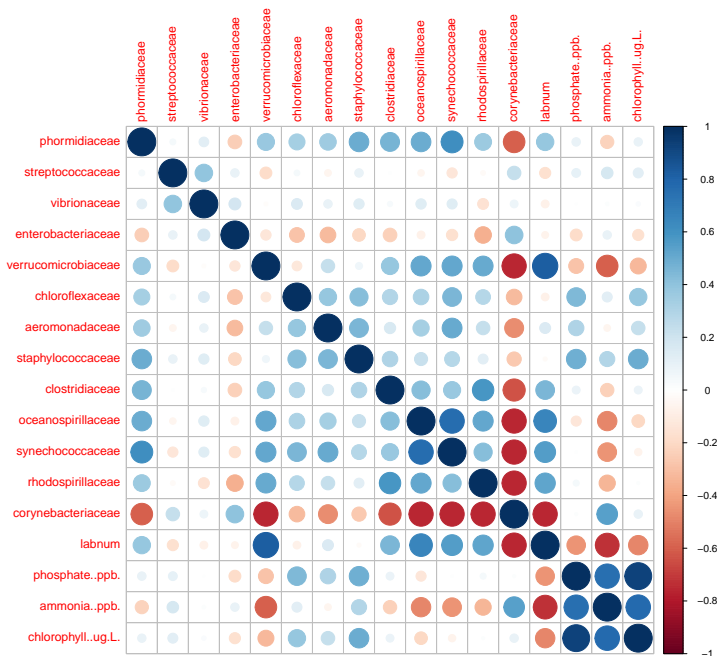
# Now lets have some fun

## Making a heat map



### A heat map

Spearman Correlations





## What next

Proposed future talks



### Help is on the way

- Parameterized Complexity Research Unit (PCRU) PhD students
- PhD student in Bioinformatics from Central South Uni

### Your feedback on some ideas

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



## Resources

If you want to improve this style



LaTeX Beamer

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



Sharelatex Site

<https://www.sharelatex.com>



A Data Cleaning Mooc

<https://www.sharelatex.com>



# R Packages Used

Session Info



## Output of sessionInfo

```

sessionInfo()

## R version 3.1.2 (2014-10-31)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
##
## locale:
## [1] C
##
## attached base packages:
## [1] methods      stats      graphics  grDevices  utils      datasets  base
##
## other attached packages:
## [1] corrplot_0.73  RSQLite_1.0.0  DBI_0.3.1    ggplot2_1.0.0
## [5] reshape2_1.4.1 plyr_1.8.1    stringr_0.6.2 xtable_1.7-4
## [9] xlsx_0.5.7     xlsxjars_0.6.1 rJava_0.9-7  knitr_1.11
##
## loaded via a namespace (and not attached):
## [1] MASS_7.3-39      Rcpp_0.11.5     colorspace_1.2-6 digest_0.6.8
## [5] evaluate_0.7.2   formatR_1.2     grid_3.1.2     gtable_0.1.2
## [9] highr_0.5        labeling_0.3    munsell_0.4.2  proto_0.3-10
## [13] scales_0.2.4     tools_3.1.2

```