### Introduction

Why use R?

Why not use a spreadsheet?

Todays workshop

* A common scenario
* A friend has emailed you her data in a spreadsheet
* Todays workshop is about how to get started.
* It’s 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?

* R is free
* R is 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

### References

[1]   James P. Howard. R Cookbook. O’Reilly Media, Inc, 2011.

[2]   Phil Spector. Data Manipulation with R. Use R series Springer, 2008

Getting Started

Todays Files

Workshop files on Github

<https://github.com/pechang03/SizeDoesMatter>

* The slides. main.pdf
* The handouts handout.pdf
* The R code SizeDoesMatterEg.R
* The spreadsheets
  + [1\_RWkshp\_dummydata\_OTUtable.xlsx](1_R%20Wkshp_dummy%20data_OTU%20table.xlsx)
  + [2\_RWkshp\_dummydata\_EnvData\_incl2outliersMK.xlsx](2_R%20Wkshp_dummy%20data_Env%20Data_incl2outliersMK.xlsx)
  + [3\_Followupdatafromcontaminatedsite\_MK.xlsx](3_Follow%20up%20data%20from%20contaminated%20site_MK.xlsx)

Getting Started

Installing R!

Download it

* Open [http://www.r-\_project.org](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

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](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>

#### Some manners

Iterative development

Working Creatively

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

* Keep the ‘manager’ away
* Work sustainably
* People over process
* Iterative development

#### Basic R Data types

R Data types

Lists, frames and tables

Vectors

* Vectors l ← 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

-

### Reading files

Let’s read the first 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>

Reading a table from a file

Reading an excel 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/](http://www.thertrader.com/2014/02/11/a-million-ways-to-connect-r-and-excel/)
* Installing the R package xlsx
* CRAN mirror <http://cran.csiro.au>
* Change in preferences

### Getting help on packages

R Packages

CRAN

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-\_other-\_way](http://www.lib.uts.edu.au/question/5955/how-can-i-import-bibliography-endnote-bibtex-latex-what-about-converting-other-way)

  x←citation()   
  x1←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

Reading an excel 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

-

Reading an excel table

The columns types 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 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Reading an excel 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://stackoverflow.com/questions/17288197/reading-\_a-\_csv-\_file-\_organized-\_horizontally](http://stackoverflow.com/questions/17288197/reading-a-csv-file-organized-horizontally)

Fields across many columns

Replicating first column

TDD – First 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    
##  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    
##  1    
## Levels:  1 2 3 4 FALSE TRUE    
## X9    
##  1    
## Levels:  1 2 3 4 FALSE TRUE    
## X10    
##   1    
## Levels:  1 2 3 4 FALSE TRUE    
## X11    
##   2    
## Levels:  1 2 3 4 FALSE TRUE    
## X12    
##   2    
## Levels:  1 2 3 4 FALSE TRUE    
## X13    
##   2    
## Levels:  1 2 3 4 FALSE TRUE    
## X14    
##   1    
## Levels:  1 2 3 4 FALSE TRUE    
## X15    
##   1    
## Levels:  1 2 3 4 FALSE TRUE    
## X16    
##   1    
## Levels:  1 2 3 4 FALSE TRUE    
## X17

##   2    
## Levels:  1 2 3 4 FALSE TRUE    
## X18    
##   2    
## Levels:  1 2 3 4 FALSE TRUE    
## X19    
##   2    
## Levels:  1 2 3 4 FALSE TRUE    
## X20    
##   1    
## Levels:  1 2 3 4 FALSE TRUE    
## X21    
##   1    
## Levels:  1 2 3 4 FALSE TRUE    
## X22    
##   1    
## Levels:  1 2 3 4 FALSE TRUE    
## X23    
##   2    
## Levels:  1 2 3 4 FALSE TRUE    
## X24    
##   2    
## Levels:  1 2 3 4 FALSE TRUE    
## X25    
##   2    
## Levels:  1 2 3 4 FALSE TRUE    
## X26    
##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X27    
##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X28    
##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X29    
##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## X30    
##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## X31    
##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## X32

##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X33    
##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X34    
##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X35    
##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## X36    
##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## X37    
##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## X38    
##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X39    
##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X40    
##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X41    
##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## X42    
##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## X43    
##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## X44    
##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X45    
##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X46    
##   3    
## Levels:  1 2 3 4 FALSE TRUE    
## X47

##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## X48    
##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## X49    
##   4    
## Levels:  1 2 3 4 FALSE TRUE    
## rowNames    
##    FALSE    
## Levels:  1 2 3 4 FALSE TRUE    
## transpose    
##      TRUE    
## Levels:  1 2 3 4 FALSE TRUE

-

### Strings

How to work with strings

stringr package

* require(stringr)

A look at the stringer 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)

-

Reading Tables

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://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>
* Input files from Stata

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

Morning Tea Time

Back in 20min

Need coffee !!

### Types

Let’s read the next table

Reading a table using xlxs

setwd("/Users/pcru/SizeDoesMatter1")   
#dir()   
table2<-read.xlsx2("2\_R Wkshp\_dummy data\_Env Data\_incl2outliersMK.xlsx", sheetName ="Sheet2",header=TRUE,rowNames=FALSE)

-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  | 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 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Reading the next table

Reading a table I

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 1 1 1 ...    
##  $ Site                  : Factor w/ 4 levels "1","2","3","4": 1 1 1 2 2 2 1 1 1 2 ...    
##  $ Sample.ID             : Factor w/ 18 levels "10000","10001",..: 1 2 3 4 5 6 7 8 9 1 ...    
##  $ Rep                   : Factor w/ 9 levels "1","2","3","A",..: 1 2 3 1 2 3 7 8 9 7 ...    
##  $ Spill.date            : Factor w/ 2 levels "14-May-14","N/A": 1 1 1 1 1 1 1 1 1 1 ...    
##  $ Sample.collection.date: Factor w/ 4 levels "15.5.14","17/5/14",..: 1 1 1 1 1 1 2 2 2 2 ...    
##  $ labnum                : Factor w/ 36 levels "2000","2001",..: 1 2 3 4 5 6 7 8 9 19 ...    
##  $ phosphate..ppb.       : Factor w/ 39 levels "10","105","108",..: 27 30 28 26 25 27 12 15 13 7 ...    
##  $ ammonia..ppb.         : Factor w/ 41 levels "10","103","1042",..: 10 14 15 6 7 4 31 34 32 28 ...    
##  $ chlorophyll..ug.L.    : Factor w/ 38 levels "1","10","11",..: 20 23 21 25 17 18 16 14 15 12 ...    
##  $ DO....                : Factor w/ 31 levels "100","120","31",..: 5 4 3 7 6 5 8 7 9 11 ...

-

Reading the next table

Reading a table II

Break it down

First read a few rows only

table2 <- read.xlsx2("2\_R Wkshp\_dummy data\_Env Data\_incl2outliersMK.xlsx", sheetName = "Sheet2",   
    header = TRUE, rowNames = FALSE, as.Data.frame = FALSE, colIndex = c(1:5),   
    stringsAsFactors = FALSE, colClasses = c("character", "numeric", "numeric",   
        rep("character", 2)), endRow = 4)   
sapply(table2, mode)

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

sapply(table2, class)

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

-

Reading 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 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\_incl2outliersMK.xlsx",   
sheetName = "Sheet2",header=TRUE,rowNames=FALSE,as.Data.frame=FALSE,   
colIndex=c(1:11),stringsAsFactors=FALSE,   
colClasses=c("character",rep("numeric",2),"character",rep("character",2),rep("numeric",6)))   
sapply(table2,class)

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

-

Reading table 2

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"

-

Reading table 2

Setting the Date Type Correctly

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")   
cdate1<-as.Date(table2bf$Sample.collection.date,"%d.%m.%y")   
cdate2<-as.Date(table2bf$Sample.collection.date,"%d/%m/%y")   
table2bf$Sample.collection.date<-as.Date(ifelse   
(!is.na(cdate1),as.Date(cdate1),as.Date(cdate2)), origin="1970-01-01")   
table2bf$Group<-as.factor(table2bf$Group)   
table2bf$Rep<-as.factor(table2bf$Rep)   
dated<-table2bf$Sample.collection.date-table2bf$Spill.date

-

Reading table 2

Setting the Date Type Correctly

Count the NAs

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    
##          as.Data.frame    
##                      0

-

### Strings

Reading table 2

Just fix the Rep column using the stringer 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 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 ...

-

### Merging Tables

How to merge two data sets?

Using the merge command

The inbuilt command merge

* R has a command merge
* To begin, start looking at the first 9 lines of the tables and merge them
* Need to use Group, Site, Sample.ID because otherwise it’s not unique

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 do I append two data sets?

To begin load the third data set

Follow up data from contaminated site

table3←read.xlsx2(”3\_Follow up data from contaminated site\_MK.xlsx”,   
 sheetName =”Sheet1”,header=TRUE,rowNames=FALSE,   
 colClasses=c(rep(”character”,3),   
   
rep(”character”,2),rep(”numeric”,18)))   
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 do I append two data sets?

Loading the third data set

Joining table 3 to the other 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 porphyromondaceae
* Should these values be NA or 0?
* We will do one of each.
* Generally we would use NA but in this case 0 is better as its likely the rows were missing as none were detected

How do I append two data sets?

Appending the third set

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))   
porphyromondaceae←rep(NA,nrow(table3fii))   
table3fiii←cbind(table3fii, corynebacteriaceae, porphyromondaceae)   
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)

## Loading required package: plyr

## [1] 24    
## [1] 27    
## [1] "Sample ID"          "corynebacteriaceae" "porphyromondaceae"    
## character(0)

-

Another Break

### Fat or wide

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 what a box plot to see if there are outliers
* We will use ggplot2 box plot
* The 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