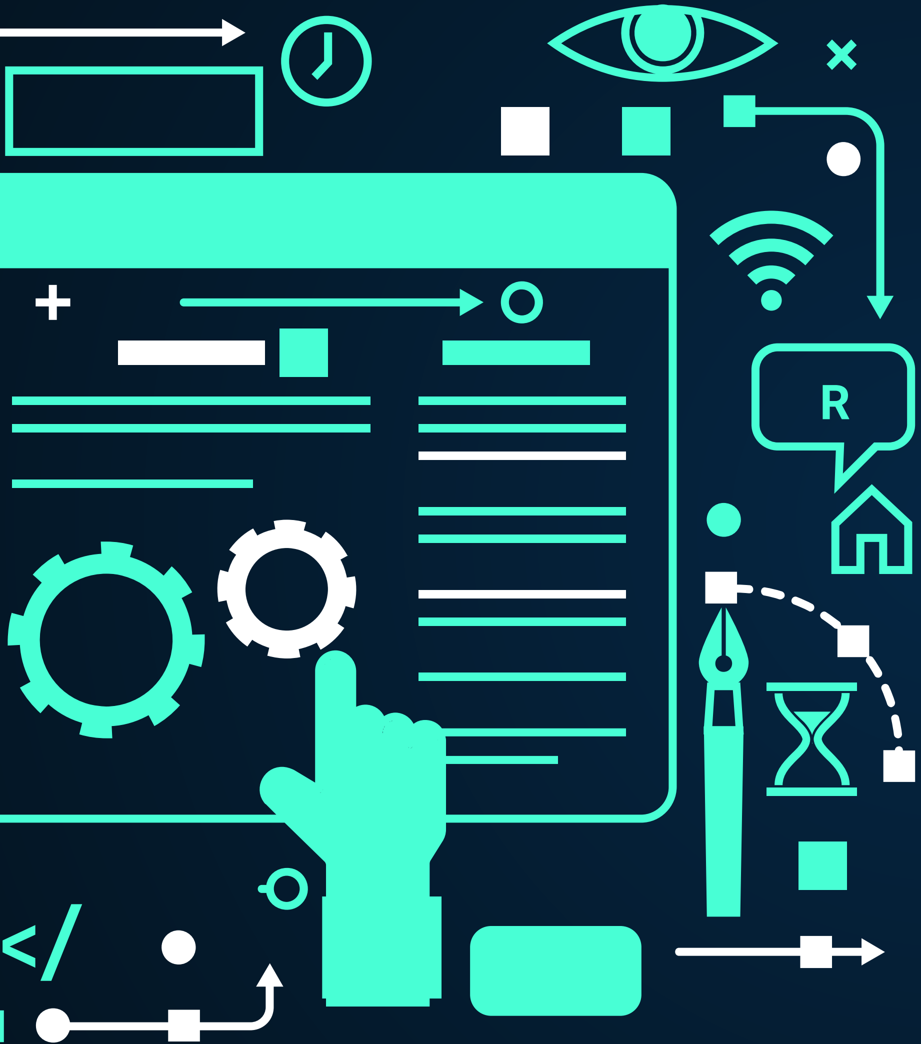


Statistical Bioinformatics Lab: Introduction to R

January 11, 2023

(slides adapted from Jasmine Chong)

TABLE OF CONTENTS

R Basics 01

R Studio - Commands in R 02

R Data Types 03

04 R Data Structures

05 Data Manipulation

06 Real World Data

R Basics

- R is case-sensitive! **A** and **a** are different.
- Separate commands with ; or a new line.
- Add comments with # (@ start of line)
  - Useful to document code so others (and future you!) can understand what the code is doing
- Exit R with **q()**

R Basics II

- Assignment operator **<-**

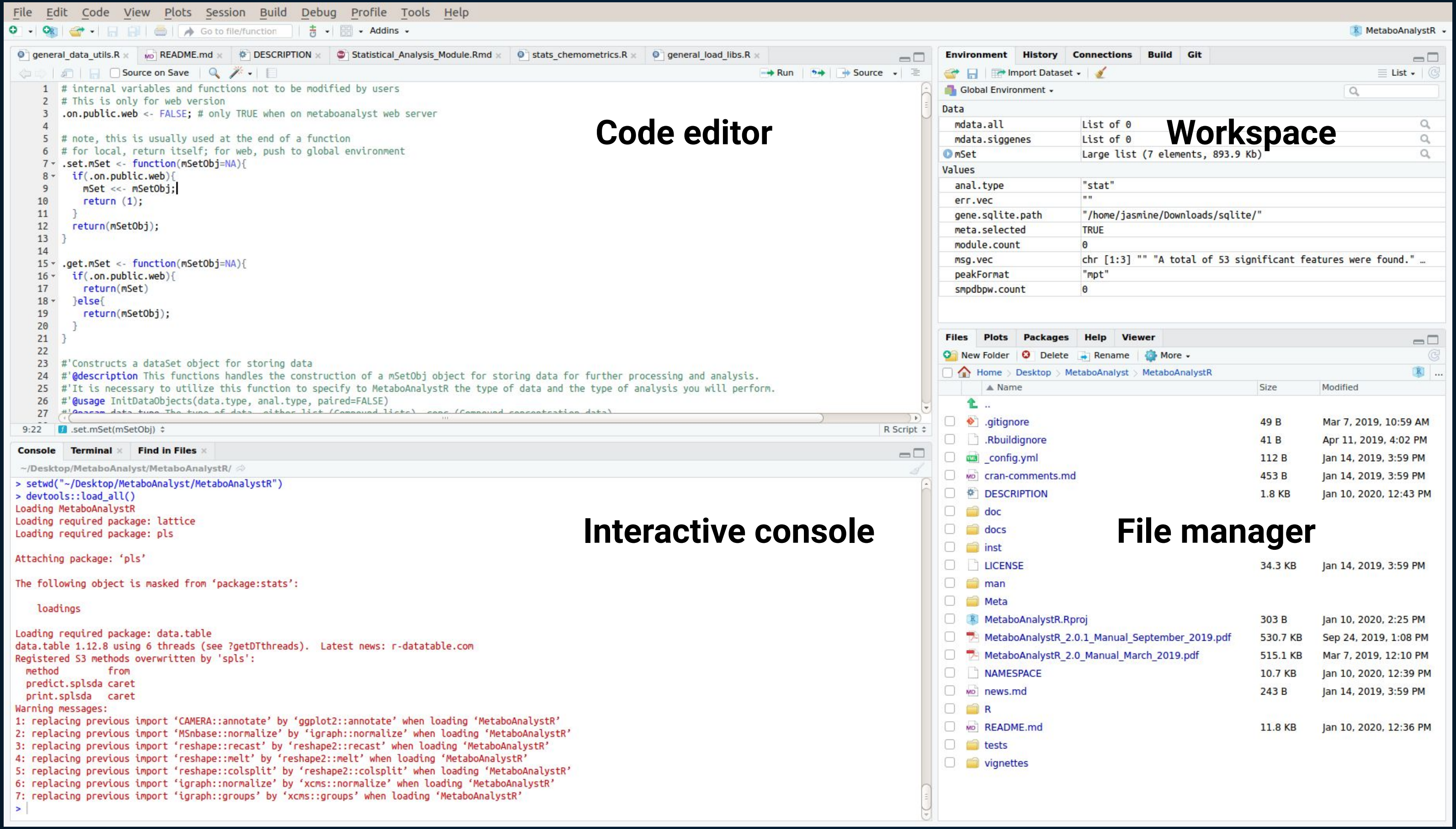
```
> x <- 1
> x = 1
```

- Equivalent to the function:

```
> assign(“x”, 1)
```

R Studio Basics

- Top left window ~ **Code editor**
  - Create R script containing your code
  - Cut and paste into R console or Edit > Run
- Bottom left window ~ **Interactive console**
  - Type directly into command line
  - R code runs 1 line at a time
- Top right window ~ **Workspace**
  - All objects in current working environment / history
- Bottom right window ~ **Data visualization, working directory, help files**



Code editor

```r
# internal variables and functions not to be modified by users
# This is only for web version
.on.public.web <- FALSE; # only TRUE when on metaboanalyst web server

# note, this is usually used at the end of a function
# for local, return itself; for web, push to global environment
.set.mSet <- function(mSetObj=NA){
  if(.on.public.web){
    mSet <<- mSetObj;
    return (1);
  }
  return(mSetObj);
}

.get.mSet <- function(mSetObj=NA){
  if(.on.public.web){
    return(mSet)
  }else{
    return(mSetObj);
  }
}

#'Constructs a dataSet object for storing data
#'@description This functions handles the construction of a mSetObj object for storing data for further processing and analysis.
#'It is necessary to utilize this function to specify to MetaboAnalystR the type of data and the type of analysis you will perform.
#'@usage InitDataObjects(data.type, anal.type, paired=FALSE)
#'@param data.type The type of data, either list (Compound lists), conc (Compound concentration data)
```

Workspace

Data	
mdata.all	List of 0
mdata.siggenes	List of 0
mSet	Large list (7 elements, 893.9 Kb)
Values	
anal.type	"stat"
err.vec	""
gene.sqlite.path	"/home/jasmine/Downloads/sqlite/"
meta.selected	TRUE
module.count	0
msg.vec	chr [1:3] "" "A total of 53 significant features were found." …
peakFormat	"mpt"
smpdbpw.count	0

Interactive console

```
> setwd("~/Desktop/MetaboAnalyst/MetaboAnalystR")
> devtools::load_all()
Loading MetaboAnalystR
Loading required package: lattice
Loading required package: pls

Attaching package: ‘pls’

The following object is masked from ‘package:stats’:

    loadings

Loading required package: data.table
data.table 1.12.8 using 6 threads (see ?getDTthreads).  Latest news: r-datatable.com
Registered S3 methods overwritten by 'spls':
  method          from
  predict.splsda caret
  print.splsda   caret
Warning messages:
1: replacing previous import ‘CAMERA::annotate’ by ‘ggplot2::annotate’ when loading ‘MetaboAnalystR’
2: replacing previous import ‘MSnbase::normalize’ by ‘igraph::normalize’ when loading ‘MetaboAnalystR’
3: replacing previous import ‘reshape::recast’ by ‘reshape2::recast’ when loading ‘MetaboAnalystR’
4: replacing previous import ‘reshape::melt’ by ‘reshape2::melt’ when loading ‘MetaboAnalystR’
5: replacing previous import ‘reshape::colsplit’ by ‘reshape2::colsplit’ when loading ‘MetaboAnalystR’
6: replacing previous import ‘igraph::normalize’ by ‘xcms::normalize’ when loading ‘MetaboAnalystR’
7: replacing previous import ‘igraph::groups’ by ‘xcms::groups’ when loading ‘MetaboAnalystR’
>
```

File manager

▲ Name	Size	Modified
..		
.gitignore	49 B	Mar 7, 2019, 10:59 AM
.Rbuildignore	41 B	Apr 11, 2019, 4:02 PM
_config.yml	112 B	Jan 14, 2019, 3:59 PM
cran-comments.md	453 B	Jan 14, 2019, 3:59 PM
DESCRIPTION	1.8 KB	Jan 10, 2020, 12:43 PM
doc		
docs		
inst		
LICENSE	34.3 KB	Jan 14, 2019, 3:59 PM
man		
Meta		
MetaboAnalystR.Rproj	303 B	Jan 10, 2020, 2:25 PM
MetaboAnalystR_2.0.1_Manual_September_2019.pdf	530.7 KB	Sep 24, 2019, 1:08 PM
MetaboAnalystR_2.0_Manual_March_2019.pdf	515.1 KB	Mar 7, 2019, 12:10 PM
NAMESPACE	10.7 KB	Jan 10, 2020, 12:39 PM
news.md	243 B	Jan 14, 2019, 3:59 PM
R		
README.md	11.8 KB	Jan 10, 2020, 12:36 PM
tests		
vignettes		

Getting help in R

- For any R function:
  - **help(sum)**
  - **?sum**
- Many R functions are extensively documented - manuals, tutorials, examples. Most important part are the commands’ description and parameters.
- To view parameters of any functions use **args()**
  - **args(matrix)**
  - **args(cor)**

R Basics III

- To view the command history
  - Use the vertical arrow keys on the keyboard
- R objects
  - Remove objects ~ **rm(x,y)**
- Save workspace
  - **save.image(file="me.RData")** # saves all objects in current R session
  - **save(city, country, file="me.RData")** # save particular objects
  - **load(file="me.RData")** # load objects in new R session
  - **saveRDS(city, file="city.rds")** # save 1 object
  - **city <- readRDS("city.rds")** # load object and assign to a variable

R Data Types

- Numeric
- Character
- Logical
- Factor
- Check class of data with **class()**

```
> x <- 1
> class(x)
[1] “numeric”
```

Logical variables

- **>** greater than
- **<** less than
- **==** equal to each-other
- **!=** not equal to each-other
- **!X** (not X)
- **X & Y** (X and Y)
- **X | Y** (X or Y)

R Basics: Sequences of Numbers

- Generate sequences of numbers

```
> 1:30
> 30:1
```

- Sequence function: **seq(from=1, to=1, by=((to - from)/(length.out - 1)), length.out=NULL)**

```
> seq(1,30) # equivalent to 1:30
> x <- seq(1, 30, by=0.2)
> x <- seq(1, 30, length.out=12)
```

R Basics: Sequences of Numbers

- Repeating sequences

```
> rep(1:4, times=2)
[1] 1 2 3 4 1 2 3 4

> rep(1:4, each=2)
[1] 1 1 2 2 3 3 4 4
```

R Data Structures

- **Vector**
  - Basic data structure, containing zero or more elements, all of the same type.
- **Matrix**
  - Two dimensional data structure where all elements are the same type.
- **List**
  - Contain zero or more elements of any type.
- **Data frame**
  - Two dimensional data structure where columns may be different types, but all elements in the column must be the same type.

Vectors

- Created with the combine function **c()**
- *Numerical*

  ```
  > num.vec <- c(1, 2, 3)
  ```
- *Logical*

  ```
  > log.vec <- c(TRUE, FALSE, NA)
  ```
- *Character*

  ```
  > char.vec <- c(“ice”, “rain”, “rock”)
  ```

Naming vectors

- Why? It’s important to understand your data!
- Use clear names to understand what the elements refer to.

```
> x <- c(7.5, 6.1, 5.9, 6.2)
> baby.weights <- c(7.5, 6.1, 5.9, 6.2)
```

- Names function: **names()**

```
#opt1
> names(baby.weights) <- c("Amanda", "Nick", "Jessica", "Andy")
#opt2
> baby.names <- c("Amanda", "Nick", "Jessica", "Andy")
> names(baby.weights) <- baby.names
```

Vector Arithmetic

- Can be used in arithmetic calculations, element by element

```
> c <- c(11, 6, 12, 7, 8)
> d <- c(13, 10, 9, 8, 11)
> e <- c + d
[1] 24 16 21 15 19
```

- Sum function, total of elements in a variable: **sum()**

```
> sum(c)
[1] 44
```

More functions on vectors

- > c <- c(11, 6, 12, 7, 8)
- > d <- c(13, 10, 9, 8, 11)
- > length(c) # number of elements in vector
- > mean(c) # mean of vector
- > var(c) # variance of vector
- > sqrt(var(c))
- > summary(c) # shows min, median, max, etc.
- > summary(d)
- > cor(c,d) # correlation between c and d

Indexing 101

- Indexing in R starts at 1 (0 is common in other programming languages)
- Select specific elements within a vector

```
> x <- c(1, 2, 3, 4, 5)
> x[1] # first element
```

- Select multiple elements within a vector

```
> x[c(1, 4)] # first and fourth element
[1] 1 4
> x[1:3] # first 3 elements
[1] 1 2 3
```

Indexing 101

- Index by element name

```
> x <- c(1, 2, 3, 4, 5)
> names(x) <- c("blue", "green", "orange", "yellow", "pink")
> x[c("green", "pink")]
green pink
 2 5
```

Indexing 101

- By logical comparisons, subset vector by elements that are TRUE

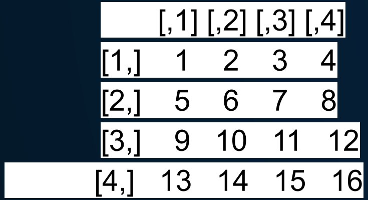
```
> baby.weights <- c(7.5, 6.1, 5.9, 6.2)
> big.baby.weights <- baby.weights>6
Amanda   Nick   Jessica   Andy
   TRUE   TRUE   FALSE   TRUE
> big6.baby.weights <- baby.weights[big.baby.weights]
Amanda   Nick   Andy
      7.5     6.1     6.2
```

Practice

Matrixes

- A matrix is a collection of data of the same type (numeric, character, logical)
- Fixed number of rows and columns
- Created using the function **matrix()**

> matrix(1:16, byrow=TRUE, nrow=4, ncol=4) #byrow is logical



```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
```

Creating a matrix

- Vectors with birth information

```
> baby_weights <- c(7.5, 6.1, 5.9, 6.2)
> baby_gest_age <- c(37, 35, 36, 38)
> baby_length <- c(51.3, 47.7, 42.1, 49.6)
```

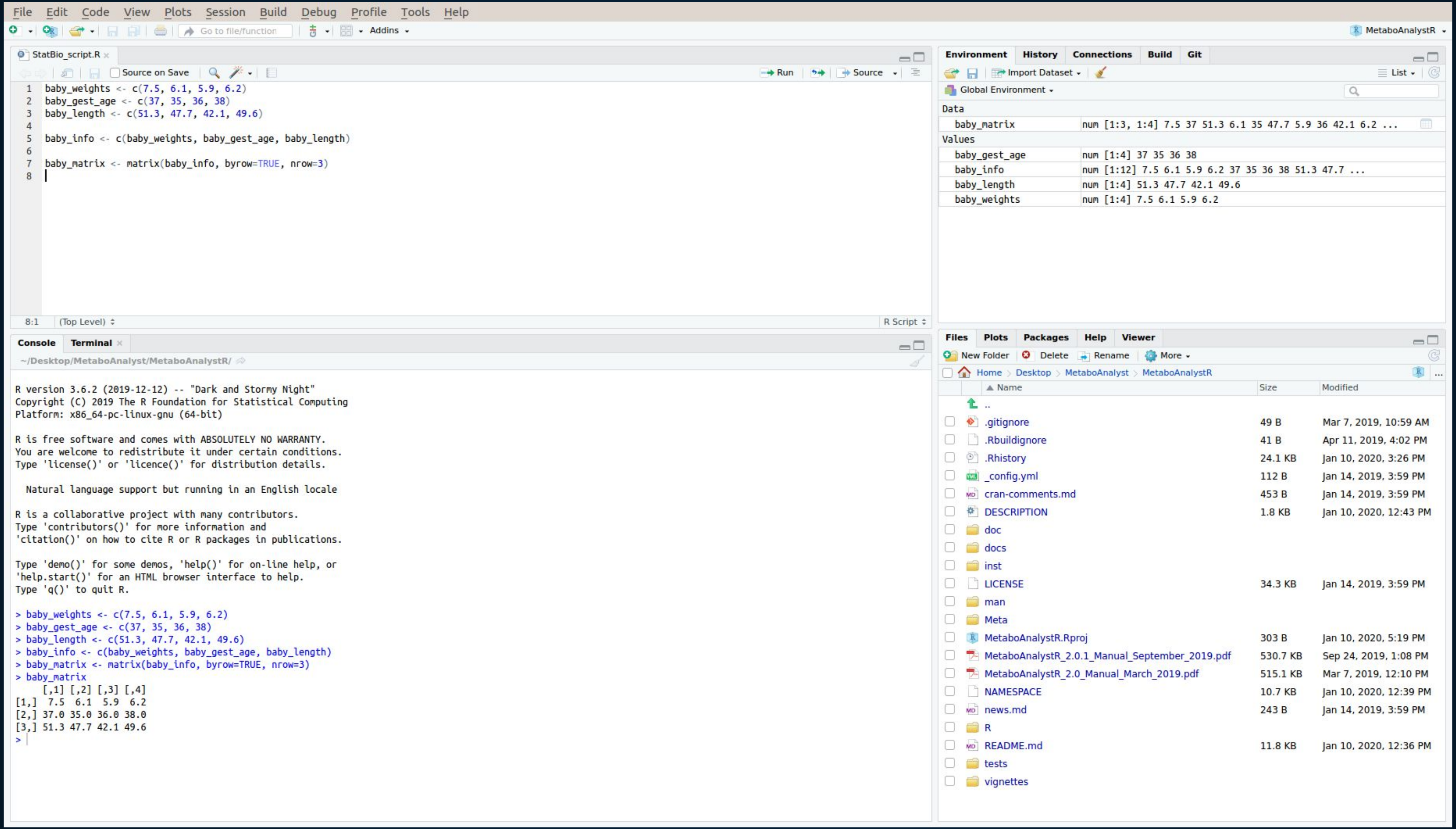
- Combine 3 vectors into 1

```
> baby_info <- c(baby_weights, baby_gest_age, baby_length)
```

- Create matrix

```
> baby_matrix <- matrix(baby_info, byrow=TRUE, nrow=3)
```

- What happens if **byrow=F**?



```r
baby_weights <- c(7.5, 6.1, 5.9, 6.2)
baby_gest_age <- c(37, 35, 36, 38)
baby_length <- c(51.3, 47.7, 42.1, 49.6)

baby_info <- c(baby_weights, baby_gest_age, baby_length)

baby_matrix <- matrix(baby_info, byrow=TRUE, nrow=3)

```

Data

baby_matrix	num [1:3, 1:4] 7.5 37 51.3 6.1 35 47.7 5.9 36 42.1 6.2 ...

Values

baby_gest_age	num [1:4] 37 35 36 38
baby_info	num [1:12] 7.5 6.1 5.9 6.2 37 35 36 38 51.3 47.7 ...
baby_length	num [1:4] 51.3 47.7 42.1 49.6
baby_weights	num [1:4] 7.5 6.1 5.9 6.2

~/Desktop/MetaboAnalyst/MetaboAnalystR/

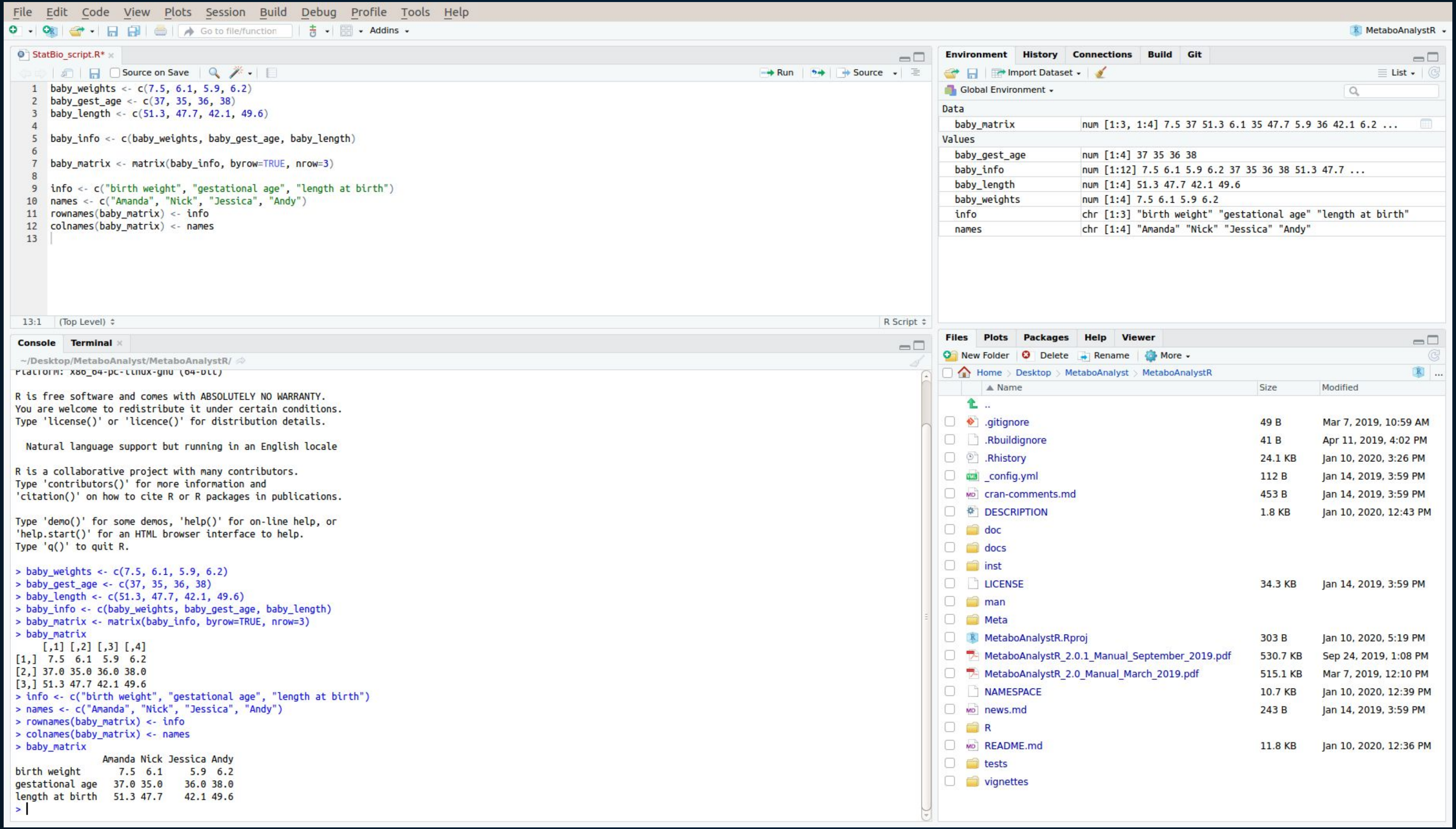
```
R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> baby_weights <- c(7.5, 6.1, 5.9, 6.2)
> baby_gest_age <- c(37, 35, 36, 38)
> baby_length <- c(51.3, 47.7, 42.1, 49.6)
> baby_info <- c(baby_weights, baby_gest_age, baby_length)
> baby_matrix <- matrix(baby_info, byrow=TRUE, nrow=3)
> baby_matrix
     [,1] [,2] [,3] [,4]
[1,]  7.5  6.1  5.9  6.2
[2,] 37.0 35.0 36.0 38.0
[3,] 51.3 47.7 42.1 49.6
> 
```

▲ Name	Size	Modified
..		
.gitignore	49 B	Mar 7, 2019, 10:59 AM
.Rbuildignore	41 B	Apr 11, 2019, 4:02 PM
.Rhistory	24.1 KB	Jan 10, 2020, 3:26 PM
_config.yml	112 B	Jan 14, 2019, 3:59 PM
cran-comments.md	453 B	Jan 14, 2019, 3:59 PM
DESCRIPTION	1.8 KB	Jan 10, 2020, 12:43 PM
doc		
docs		
inst		
LICENSE	34.3 KB	Jan 14, 2019, 3:59 PM
man		
Meta		
MetaboAnalystR.Rproj	303 B	Jan 10, 2020, 5:19 PM
MetaboAnalystR_2.0.1_Manual_September_2019.pdf	530.7 KB	Sep 24, 2019, 1:08 PM
MetaboAnalystR_2.0_Manual_March_2019.pdf	515.1 KB	Mar 7, 2019, 12:10 PM
NAMESPACE	10.7 KB	Jan 10, 2020, 12:39 PM
news.md	243 B	Jan 14, 2019, 3:59 PM
R		
README.md	11.8 KB	Jan 10, 2020, 12:36 PM
tests		
vignettes		

Naming Matrices

- Need to name baby_matrix, if not data is hard to interpret.
- Name rows and columns separately.

```
> info <- c("birth weight", "gestational age", "length at birth")
> names <- c("Amanda", "Nick", "Jessica", "Andy”)
> rownames(baby_matrix) <- info
> colnames(baby_matrix) <- names
```



```r
baby_weights <- c(7.5, 6.1, 5.9, 6.2)
baby_gest_age <- c(37, 35, 36, 38)
baby_length <- c(51.3, 47.7, 42.1, 49.6)

baby_info <- c(baby_weights, baby_gest_age, baby_length)

baby_matrix <- matrix(baby_info, byrow=TRUE, nrow=3)

info <- c("birth weight", "gestational age", "length at birth")
names <- c("Amanda", "Nick", "Jessica", "Andy")
rownames(baby_matrix) <- info
colnames(baby_matrix) <- names

```

~/Desktop/MetaboAnalyst/MetaboAnalystR/

```
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> baby_weights <- c(7.5, 6.1, 5.9, 6.2)
> baby_gest_age <- c(37, 35, 36, 38)
> baby_length <- c(51.3, 47.7, 42.1, 49.6)
> baby_info <- c(baby_weights, baby_gest_age, baby_length)
> baby_matrix <- matrix(baby_info, byrow=TRUE, nrow=3)
> baby_matrix
     [,1] [,2] [,3] [,4]
[1,]  7.5  6.1  5.9  6.2
[2,] 37.0 35.0 36.0 38.0
[3,] 51.3 47.7 42.1 49.6
> info <- c("birth weight", "gestational age", "length at birth")
> names <- c("Amanda", "Nick", "Jessica", "Andy")
> rownames(baby_matrix) <- info
> colnames(baby_matrix) <- names
> baby_matrix
                Amanda Nick Jessica Andy
birth weight       7.5  6.1     5.9  6.2
gestational age   37.0 35.0    36.0 38.0
length at birth   51.3 47.7    42.1 49.6
> 
```

```
Data
  baby_matrix      num [1:3, 1:4] 7.5 37 51.3 6.1 35 47.7 5.9 36 42.1 6.2 ...
Values
  baby_gest_age    num [1:4] 37 35 36 38
  baby_info        num [1:12] 7.5 6.1 5.9 6.2 37 35 36 38 51.3 47.7 ...
  baby_length      num [1:4] 51.3 47.7 42.1 49.6
  baby_weights     num [1:4] 7.5 6.1 5.9 6.2
  info             chr [1:3] "birth weight" "gestational age" "length at birth"
  names            chr [1:4] "Amanda" "Nick" "Jessica" "Andy"
```

▲ Name	Size	Modified
..		
.gitignore	49 B	Mar 7, 2019, 10:59 AM
.Rbuildignore	41 B	Apr 11, 2019, 4:02 PM
.Rhistory	24.1 KB	Jan 10, 2020, 3:26 PM
_config.yml	112 B	Jan 14, 2019, 3:59 PM
cran-comments.md	453 B	Jan 14, 2019, 3:59 PM
DESCRIPTION	1.8 KB	Jan 10, 2020, 12:43 PM
doc		
docs		
inst		
LICENSE	34.3 KB	Jan 14, 2019, 3:59 PM
man		
Meta		
MetaboAnalystR.Rproj	303 B	Jan 10, 2020, 5:19 PM
MetaboAnalystR_2.0.1_Manual_September_2019.pdf	530.7 KB	Sep 24, 2019, 1:08 PM
MetaboAnalystR_2.0_Manual_March_2019.pdf	515.1 KB	Mar 7, 2019, 12:10 PM
NAMESPACE	10.7 KB	Jan 10, 2020, 12:39 PM
news.md	243 B	Jan 14, 2019, 3:59 PM
R		
README.md	11.8 KB	Jan 10, 2020, 12:36 PM
tests		
vignettes		

Matrix Manipulation

- Add a column with function **cbind()**
- Merges matrixes and/or vectors

```
> new_baby <- c(7.1, 39, 50.1)
> baby_matrix_c <- cbind(baby_matrix, new_baby)
```

- Add a row with function **rbind()**

```
> baby_head <- c(37.8, 35.2, 34.2, 36.1)
> baby_matrix_r <- rbind(baby_matrix, baby_head)
```

Practice

Lists in R

- Similar to vectors, but instead of having one data type, you can have all of them in one (numeric, character, logical, etc.) + all data structures (even lists in lists)
- Useful to store output from analyses
- Can also be named

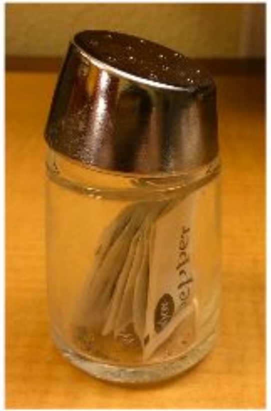
```
> apples <- c("McIntosh", "Fuji", "Gala", "Honeycrisp", "Granny Smith")
> trees <- c(100, 200, 300, 150, 175)
> good_yield <- c(F, T, T, F, T)
> apple_list <- list(apples, trees, good_yield)
> names(apple_list) <- c("a", "b", "c")
```

Lists in R

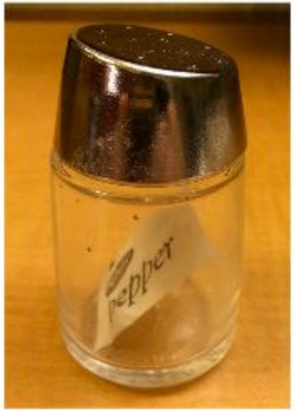
- **> apple_list[1]** #sub-list of vector 1 - [ ] always a list
- **> apple_list[c(1,3)]** #sub-list of vector 1 & 3
- **> apple_list[[2]][1]** #returns element 1 in 2nd vector - no longer a list
- **> apple_list[[2]][1]** <- 150 #directly change element
- **> apple_list_named$a** #subset named elements, works like [[ ]]

Subsetting a list

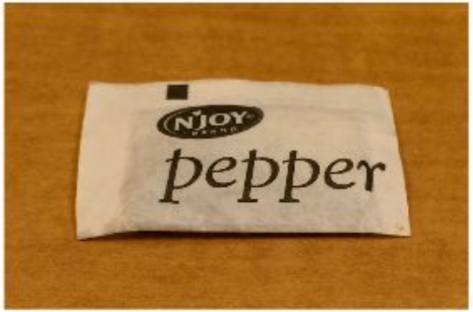
apple_list



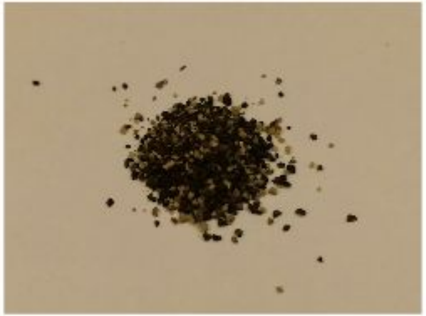
apple_list[1]



apple_list[[1]]



apple_list[[1]][1:100]



Practice

Data Frames in R

- For storing data tables, most datasets entered in R are DFs
- Characteristics:
  - List of vectors of equal length – categorical, numerical, logical
  - Column-names !empty, row-names unique
  - Create a df with function **data.frame(x, y, z)**

- **> mtcars #pre-built DF in R**
  - Header = top line, column names
  - Rows = begins with row names. Default numeric if none given.

Importing Data in R

- First, set your working directory!
  - Why? To find files to work with, save R code to a specific folder, save images to a specific folder...
- **getwd()** # shows current position
- **setwd()** # set working directory
- **dir()** # lists all files in current directory
- In R Studio, 3 ways to set working directory:
  - Session > Set working directory
  - setwd("~/Desktop")
  - Bottom right window: Files > More > Set working directory

From Excel to R

- Excel lets you save files as “comma separated values”
- Strips formatting, but content still there for R
- Function to read csv files: **read.csv(file=“x.csv”)**
- Function to create csv files: **write.csv(file=“x.csv”)**

From Excel to R

- In the dataset **bps.csv**, you will find systolic and diastolic blood pressure measurements for women and men.

  - **> setwd("~/Desktop/R/R workshop")**
  - **> bps <- read.csv(file="bps.csv", header=T) #header means that the columns are true titles**
  - **> head(bps)**
  - **Q: How many rows are there?**

Data Visualization in R

- **> attach(bps)** # negates need for dataframe $variable
- **> detach(bps)** # removes attachment
- **> help(plot)**
- **One variable plots**
  - Histograms
    - **> hist(SYSTBP)**
  - Scatter plots
    - **> plot(SYSTBP[1:5], DIASTBP[1:5])**
  - Box plots
    - **> boxplot(SYSTBP)**

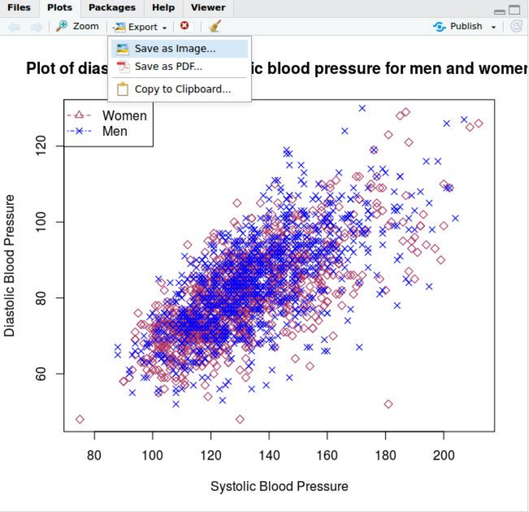
Two Variable Plots in R

- Scatterplots
  - **> plot(SYSTBP, DIASTBP)** #not that informative, all points at once
- Plot different points for men and women
- Use the function: **points(x, y)** #pch = plotting characters

```
> plot(SYSTBP, DIASTBP)

> plot(SYSTBP[SEX==0], DIASTBP[SEX==0], ylab="Diastolic Blood Pressure", xlab="Systolic Blood Pressure", main="Plot of diastolic against systolic blood pressure for men and women", col="maroon", pch=5)
> points(SYSTBP[SEX==1], DIASTBP[SEX==1], col="blue", pch=4) #points and lines will add to the previous plot, plot will give you a new plot!
> legend('topleft', c("Women","Men"), pch=c(2,4), col=c("maroon","blue"), lty=c(2,4))
```

Saving Plots in R

- Need to open a graphic device
- R Studio Plots Panel: Export > Save as Image or Save as PDF



Saving Plots in R

- Specify the format to save images using functions such as: **jpeg()**, **png()**, **svg()** or **pdf()**.
  - Also specify the height and width of the images in these functions.
- Create the plot.
- Close the file with **dev.off()**
  - **> png(file=“x.png”, width=350, height=350)**
  - **> plot(1:10)**
  - **> dev.off()**

Practice