

BINF 5003: Data Mining, Modeling, and Biostatistics

Week 1

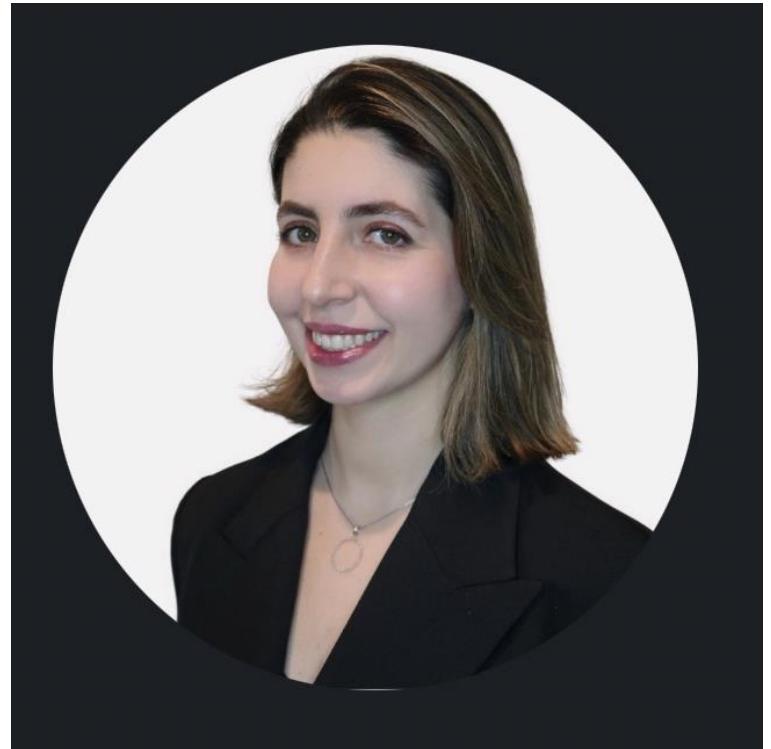
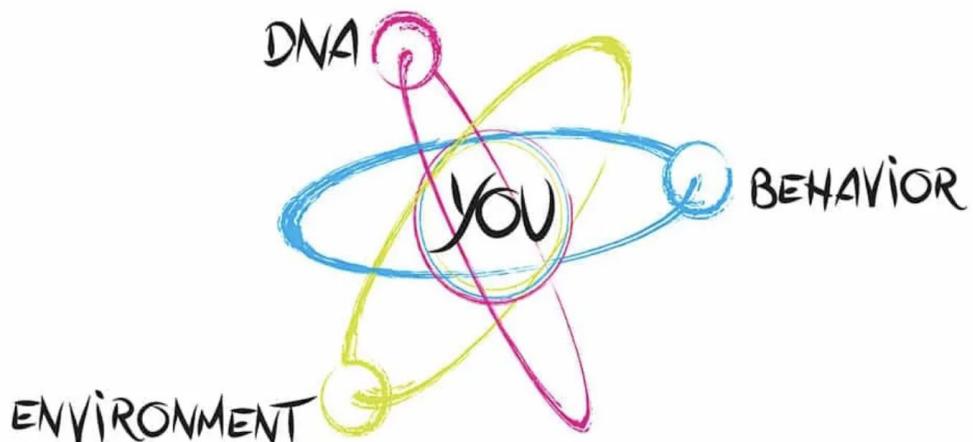
09/04/2025

Module 1 – Introduction to R

WELCOME!

- Mahshid, PhD Candidate Integrative Life Science (Virginia Commonwealth University)

MEng at University of Birmingham(UK):
Biomedical Engineering
MSc: Human Genetics (Concentration on
Genomics Data Science) VCU
PhD : Behavioural Statistical Genetics/ VCU



Lecture overview

- Expectations of the course and classroom
- Why bioinformatics is necessary with emerging and rapidly developing fields of Science
- What is R and Rstudio, and how to use these tools together to write code
- Basic syntax of R

Course Structure

Weeks	Module	Topic
1	1	Introduction to R and data analysis
2,3	2	Data Wrangling and Management
4,5	3	Data Visualization
6,7	4	Predictive Data Modeling
8,9,10	5	Dimensionality Reduction and Clustering
11,12	6	Research Project
13,14	7	Interactive Data Visualization

Course Assessments-Worksheets

- ~ One per module
- No answer keys will be posted, there are multiple correct ways to solve the questions(The "Best" solutions will be written with efficient code, minimal lines)
- Everyone will receive personalized feedback
- Read the questions carefully, review the rubric before you get started
- Concepts build on top of each other,it is important to keep up with the practice!
- 5% late penalty per day
 - 1 no-questions-asked 48-hour extension permitted if needed

Course Assessments-Independent Project

- More information on these assessments later in the semester after we get some tools under our belt!
- Identify a published dataset you are interested in analyzing using the skills developed in this class. Contrast the results with the currently published report
- Proposal(15%)
- Proposal Presentation (10%)
- Final Submission(35%)

Code of Student Community Standards

<https://humber.ca/knowthecode/codes-community-standards>

1.1 Our Community Rights & Responsibility

All community members have a responsibility to conduct themselves in a manner that is consistent with the core values embraced by the College community and reflected in its various codes and policies.

Support

- Dedicated time at the end of each workshop for discussion, review, or answer to questions
- Post questions about course content on the discussion board
- Feel free to respond to your peer's questions, instructors will review all posts
- Do not post completed code in your question or answer
- Sensitive or personal questions can be directed to your instructor's email

Why programming?

Is there a relationship between head size and brain weight?

Head Size(cm ³)	Brain Weight(grams)
4512	1530
3738	1297
4261	1335



Is there a relationship between head size and brain weight?

Need computational tools to analyze large datasets!



Head Size(cm ³)	Brain Weight(grams)
4512	1530
3738	1297
4261	1335
3777	1282
4177	1590
3585	1300
3785	1400
3559	1255
3613	1355
3982	1375
3443	1340
3993	1380

The Human Genome Project

The complete mapping and understanding of all the genes of human beings

- First genome took 13 years and \$2.7 billion USD to complete
- Since then, increasing affordability and accessibility of technology has made genetics and sequencing household terms



Elymas/Shutterstock

R – a language and environment for statistical computing and graphics

- Open-source programming language made by statisticians based on the S language
- Originally developed by Robert Gentleman (a Canadian!) and Ross Ihaka
- High quality and reproducible statistics and graphics
- Large and active online community of users and developers

RStudio Tour

Let's go!

RStudio

Program we use to simplify R programming.



R



RStudio

RStudio

An integrated development environment(IDE) for R

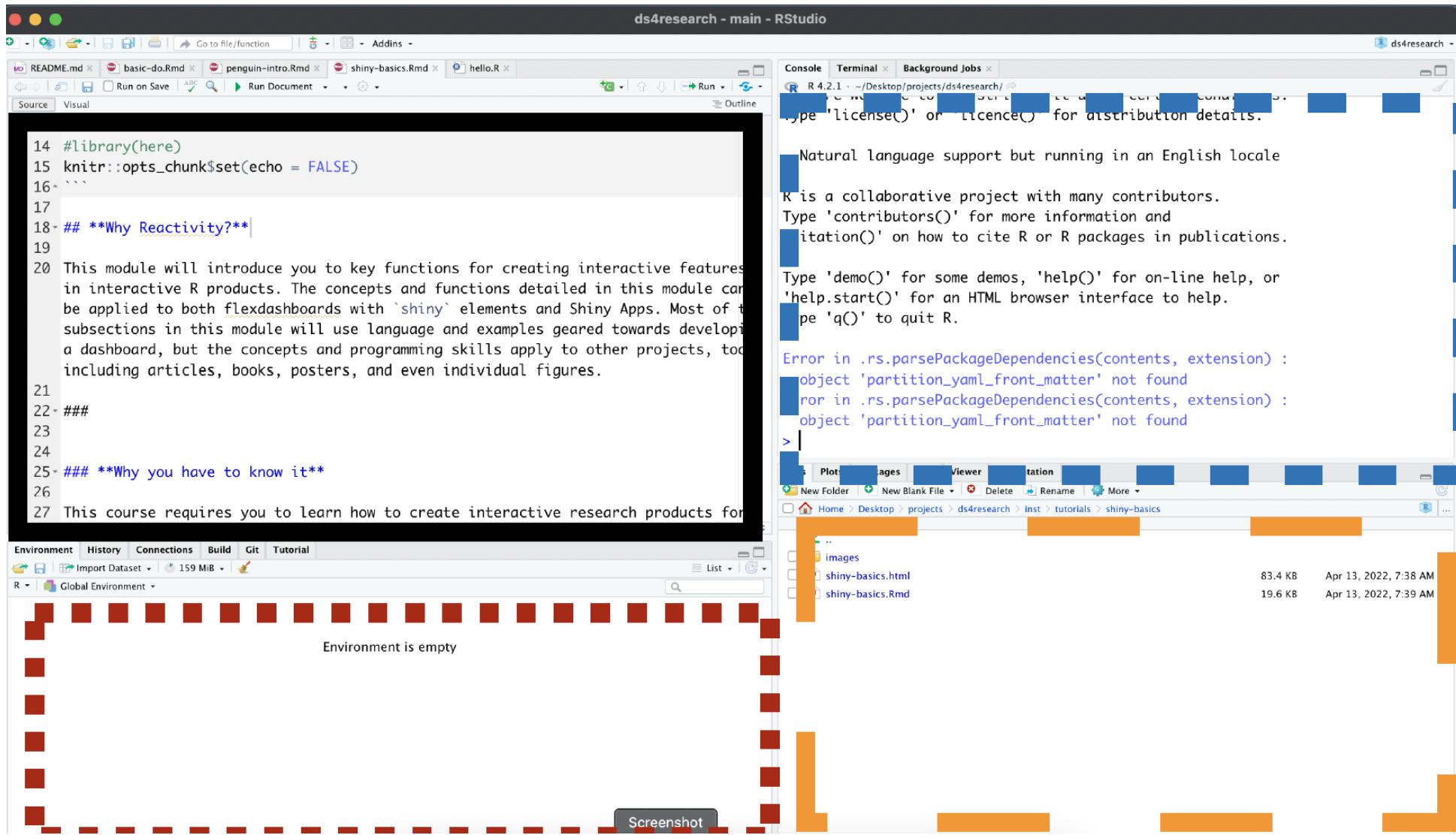
Why Use RStudio?

- Simplifies coding in R with autocomplete, Syntax highlighting, and code snippets
- Easy to visualize data and results Within same interface

Console vs Script Vs Markdown

- Console
 - The brain of the operation, the engine
- Script
 - Organized and annotated record of your code that you will save
- Markdown
 - Is its own language
 - R markdown files integrate the script with the output in a single file

RStudio has multiple panes.



Text Editor

The screenshot shows the RStudio interface with the following components:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Go to file/function, Addins.
- Version:** R 4.0.2.
- Left Panel (Text Editor):** Displays code in `02-vectors.R`. A yellow box highlights the text "Here in the text editor."

```
16 # of R objects. A foundational knowledge of how to create, modify, reorder,  
17 # and search vectors is essential for working with more complex data types.  
18  
19  
20  
21  
22  
23 # Vectors -----  
24  
25 # Vectors are collections of 1 or more *homogenous* value types.  
26 # Vectors can be numeric, character, or logical.  
27 # If you try to combine values of different types into a single vector,  
28 # R will coerce (i.e., force)  
29  
30  
31  
32 # Ways to combine numbers into a numeric vector  
33  
34 # concatenate  
23:1 # Vectors
```
- Right Panel (Environment):** Shows the Global Environment with variables `chipmunks`, `x`, `y`, and `z`.
- Bottom Left (Console):** Displays R session history with commands like `rev` and `length`.
- Bottom Right (Help):** Shows the documentation for `rev`, including the description, usage, and arguments.

Text Editor

The screenshot shows the RStudio interface. A large orange box highlights the top navigation bar and the main code editor area. In the top left of the editor, two script files are listed: "02-vectors.R" and "01-language-basics.R". A black arrow points from the text "You can load multiple scripts at once." to the tab for "01-language-basics.R". The code editor displays R code related to vectors. Below the editor is a console window showing R session history. To the right of the editor is the Global Environment pane, which lists variables like "chipmunks", "x", "y", and "z" with their corresponding data types and values. Further right is the Help pane, specifically the "Reverse Elements" documentation page.

You can load multiple scripts at once.

```
16 # of R objects. A
17 # additional knowledge of how to create, modify, reorder,
18 # and search vectors is essential for working with more complex data types.
19
20
21
22
23 # Vectors -----
24
25 # Vectors are collections of 1 or more *homogenous* value types.
26 # Vectors can be numeric, character, or logical.
27 # If you try to combine values of different types into a single vector,
28 # R will coerce (i.e., force)
29
30
31
32 # Ways to combine numbers into a numeric vector
33
34 # concatenate
```

Console

```
/cloud/project/
> chipmunks[2] <- "Aaron"
> chipmunks
[1] "Alvin"    "Aaron"     "Theodore"
> chipmunks
[1] "Alvin"    "Aaron"     "Theodore"
> chipmunks[c(1, 3)] <- c("Theodore", "Alvin")
> chipmunks
[1] "Theodore" "Aaron"     "Alvin"
> length(chipmunks)
[1] 3
> # Adding onto a vector
> chipmunks[4] <- "Simon"
> |
```

Global Environment

	values
chipmunks	chr [1:4] "Theodore" "Aaron" "Alvin" "Simon"
x	int [1:3] 1 2 3
y	chr [1:3] "a" "b" "c"
z	logi [1:3] TRUE TRUE FALSE

Help

Reverse Elements

Description

`rev` provides a reversed version of its argument. It is generic function with a default method for vectors and one for [dendrogram](#)s.

Note that this is no longer needed (nor efficient) for obtaining vectors sorted into descending order, since that is now rather more directly achievable by `sort(x, decreasing = TRUE)`.

Usage

`rev(x)`

Arguments

... A vector or another object for which `reverser` is defined.

Environment Pane

≡ Your Workspace / hgen611_2020

⚙️ 🌐 Dana Lapato

The screenshot shows the RStudio interface with several panes visible:

- File Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Project Explorer:** Shows files 02-vectors.R and 01-language-basics.R.
- Code Editor:** Displays R code related to vectors and numeric vectors.
- Console:** Shows R commands and their outputs, including a session where the 'chipmunks' vector is modified.
- Environment Pane (highlighted):** Shows the global environment with objects chipmunks, x, y, and z. A large orange box highlights this pane with the text "Here in the environment pane."
- Help/Documentation:** Shows the documentation for the rev function, including its description, usage, and arguments.

Environment Pane

Your Workspace / hgen611_2020

File Edit Code View Plots Session Build Debug Profile Tools Help

02-vectors.R 01-language-basics.R

16 # of R objects. A foundational knowledge of how to create, modify, reorder,
17 # and search vectors is essential for working with more complex data types.
18
19
20
21
22
23 # Vectors
24
25 # Vectors are collections of 1 or more *homogenous* value types.
26 # Vectors can be numeric, character, or logical.
27 # If you try to combine values of different types into a single vector,
28 # R will coerce (i.e., force)
29
30
31
32 # Ways to combine numbers into a numeric vector
33
34 # concatenate
23:1 # Vectors :

As you create and modify objects,
they will appear here.

Run Source

Global Environment

Values

chipmunks	chr [1:4] "Theodore" "Aaron" "Alvin" "Simon"
x	int [1:3] 1 2 3
y	chr [1:3] "a" "b" "c"
z	logi [1:3] TRUE TRUE FALSE

R: Reverse Elements - Find in Topic

rev {base} R Documentation

Reverse Elements

Description

`rev` provides a reversed version of its argument. It is generic function with a default method for vectors and one for [dendograms](#).

Note that this is no longer needed (nor efficient) for obtaining vectors sorted into descending order, since that is now rather more directly achievable by `sort(x, decreasing = TRUE)`.

Usage

`rev(x)`

Arguments

... a vector or another object for which reversal is defined

Console Terminal Jobs

/cloud/project/

```
> chipmunks[2] <- "Aaron"
> chipmunks
[1] "Alvin"    "Aaron"     "Theodore"
> chipmunks
[1] "Alvin"    "Aaron"     "Theodore"
> chipmunks[c(1, 3)] <- c("Theodore", "Alvin")
> chipmunks
[1] "Theodore" "Aaron"     "Alvin"
> length(chipmunks)
[1] 3
> # Adding onto a vector
> chipmunks[4] <- "Simon"
> |
```

Console

Your Workspace / hgen611_2020

File Edit Code View Plots Session Build Debug Profile Tools Help

02-vectors.R x 01-language-basics.R x

```
16 # of R objects. A foundational knowledge of how to create, modify, reorder,
17 # and search vectors is essential for working with more complex data types.
18
19
20
21
22
23 # Vectors -----
24
25 # Vectors are collections of 1 or more *homogenous* value types.
26 # Vectors can be numeric, character, or logical.
27 # If you try to combine values of different types into a single vector,
28 # R will coerce (i.e., force)
29
30
31
32 # Ways to combine numbers into a numeric vector
33
34 # concatenate
```

23:1 # Vectors :

Console Terminal Jobs

```
/cloud/project/ 
> chipmunks[2] <- "Aaron"
> chipmunks
[1] "Alvin"    "Aaron"     "Theodore"
> chipmunks
[1] "Alvin"    "Aaron"     "Theodore"
> chipmunks[c(1, 3)] <- c("Theodore", "Alvin")
> chipmunks
[1] "Theodore" "Aaron"     "Alvin"
> length(chipmunks)
[1] 3
> # Adding onto a vector
> chipmunks[4] <- "Simon"
>
```

Run Source

R 4.0.2

Environment History Connections Tutorial

Import Dataset

Global Environment

Values	
chipmunks	chr [1:4] "Theodore" "Aaron" "Alvin" "Simon"
x	int [1:3] 1 2 3
y	chr [1:3] "a" "b" "c"
z	logi [1:3] TRUE TRUE FALSE

Files Plots Packages Help Viewer

R: Reverse Elements - Find in Topic

rev {base} R Documentation

Reverse Elements

description

rev provides a reversed version of its argument. It is generic function with a default method for vectors and one for [dendograms](#).

note that this is no longer needed (nor efficient) for obtaining vectors sorted to descending order, since that is now rather more directly achievable by [sort\(x, decreasing = TRUE\)](#).

usage

rev(x)

arguments

a vector or another object for which reversal is defined

Console



If I can run code in the console, why would I bother using the text editor?

≡ Your Workspace / hgen611_2020

The screenshot shows the RStudio interface. In the top-left, there's a navigation bar with tabs like File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below it is a toolbar with icons for file operations like Open, Save, and Print. A search bar says "Go to file/function". A dropdown menu says "Addins". On the right, it says "R 4.0.2". In the center-left, there are two tabs: "02-vectors.R" and "01-language-basics.R". The "02-vectors.R" tab is active, displaying R code. Lines 16-34 are visible, including comments about vectors and numeric vectors. The "Global Environment" panel on the right shows variables: chipmunks (character vector), x (integer vector), y (character vector), and z (logical vector). At the bottom, the "Console" tab is active, showing a session history with commands like `> chipmunks[2] <- "Aaron"`, `> chipmunks`, and `> rev(chipmunks)`. A yellow callout box highlights the text: "Code typed into the console is not saved. TERRIBLE for reproducibility!"

File Edit Code View Plots Session Build Debug Profile Tools Help

02-vectors.R 01-language-basics.R

Run Source

R 4.0.2

Environment History Connections Tutorial

Import Dataset

Global Environment

Values

chipmunks	chr [1:4] "Theodore" "Aaron" "Alvin" "Simon"
x	int [1:3] 1 2 3
y	chr [1:3] "a" "b" "c"
z	logi [1:3] TRUE TRUE FALSE

Files Plots Packages Help Viewer

R: Reverse Elements Find in Topic

rev {base} R Documentation

Reverse Elements

Description

rev provides a reversed version of its argument. It is generic function with a default method for vectors and one for [dendrogram](#)s.

Note that this is no longer needed (nor efficient) for obtaining vectors sorted in decreasing order, since that is now rather more directly achievable by `decreasing = TRUE`.

Arguments

Console Terminal Jobs

/cloud/project/

> chipmunks[2] <- "Aaron"
> chipmunks
[1] "Alvin" "Aaron" "Theodore"
> chipmunks
[1] "Alvin"
> chipmunks[c(1, 4)]
> chipmunks
[1] "Theodore" "
> length(chipmunks)
[1] 3
> # Adding onto
> chipmunks[4] <- "Simon"
>

Code typed into the console is not saved.
TERRIBLE for reproducibility!

Navigation Bar

The screenshot shows the RStudio interface with a large blue callout pointing from the text box below to the 'Files' tab in the top navigation bar.

Your Workspace / hgen611_2020

File Edit Code View Plots Session Build Debug Profile Tools Help

02-vectors.R x 01-language-basics.R x

```
16 # of R objects. A foundational knowledge of how to create, modify, reorder,
17 # and search vectors is essential for working with more complex data types.
18
19
20
21
22
23 # Vectors -----
24
25 # Vectors are collections of 1 or more *homogenous* value types.
26 # Vectors can be numeric, character, or logical.
27 # If you try to combine values of different types into a single vector,
28 # R will coerce (i.e., force)
29
30
31
32 # Ways to combine numbers into a numeric vector
33
34 # concatenate
35 # Vectors
36
```

Run Source

Environment History Connections Tutorial

Import Dataset

Global Environment

Values	
chipmunks	chr [1:4] "Theodore" "Aaron" "Alvin" "Simon"
x	int [1:3] 1 2 3
y	chr [1:3] "a" "b" "c"
z	logi [1:3] TRUE TRUE FALSE

Files Plots Packages Help Viewer

R: Reverse Elements - Find in Topic

rev {base} R Documentation

Reverse Elements

Description

rev provides a reversed version of its argument. It is generic function with a default method for vectors and one for [dendrograms](#).

Note that this is no longer needed (nor efficient) for obtaining vectors sorted into descending order, since that is now rather more directly achievable by `sort(x, decreasing = TRUE)`.

Usage

`rev(x)`

Arguments

`x` a vector or another object for which reversal is defined.

The Navigation Bar. It can help you find specific places in your code.

File Pane

The screenshot shows the RStudio interface with the following components:

- Header Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Go to file/function, Addins.
- Environment Tab:** Shows objects in the Global Environment: chipmunks (character vector), x (integer vector), y (character vector), z (logical vector).
- Help Tab:** Shows the help page for the `rev` function, which provides a reversed version of its argument.
- Console Tab:** Displays R code and its output, including the creation of a character vector `chipmunks` and its reversal.
- Text Overlay:** A large black box contains the text:
 - This pane does a lot of stuff.**
 - Displays files in your laptop or server, plots, helps you install packages, displays help pages.**

A large orange rectangle highlights the Help tab area, and a black arrow points from the text overlay towards it.

```
16 # of R objects. A foundational knowledge of how to create, modify, reorder,
17 # and search vectors is essential for working with more complex data types.
18
19
20
21
22
23 # Vectors -----
24
25 # Vectors are collections of 1 or more *homogenous* value types.
26 # Vectors can be numeric, character, or logical.
27 # If you try to combine values of different types into a single vector
28 # R will coerce them to a common type.
29
30
31
32 # Ways to combine vectors
33
34 # concatenate
23:1 # Vectors
  
```

```
/cloud/project/
> chipmunks[2] <- "Aaron"
> chipmunks
[1] "Alvin"     "Aaron"
> chipmunks
[1] "Alvin"     "Aaron"    "Theodore"
> chipmunks[c(1, 3)] <- c("Theodore", "Alvin")
> chipmunks
[1] "Theodore"  "Aaron"    "Alvin"
> length(chipmunks)
[1] 3
> # Adding onto a vector
> chipmunks[4] <- "Simon"
> 
```

rev {base}

Reverse Elements

Description

`rev` provides a reversed version of its argument. It is generic function with a default method for vectors and one for [dendrogram](#)s.

Note that this is no longer needed (nor efficient) for obtaining vectors sorted into descending order, since that is now rather more directly achievable by [`sort\(x, decreasing = TRUE\)`](#).

Usage

`rev(x)`

Arguments

`x` – a vector or another object for which `reverser` is defined.

Running Code

Your Workspace / hgen611_2020 Dana Lapato

File Edit Code View Plots Session Build Debug Profile Tools Help

02-vectors.R x 01-language-basics.R x

Run

16 # of R objects. A foundational knowledge of how to create, modify, reorder,
17 # and search vectors is essential for working with more complex data types.
18
19
20
21
22
23 # Vectors -----
24
25 # Vectors are collections of 1 or more *homogenous* value types.
26 # Vectors can be numeric, character, or logical.
27 # If you try to combine values of different types into a single vector,
28 # R will coerce (i.e., force)
29
30
31
32 # Ways to combine numbers into a numeric vector
33
34 # concatenate
23:1 # Vectors :

chipmunks

x

y

z

Environment History Connections Tutorial

Import Dataset

Global Environment

Values

chipmunks chr [1:4] "Theodore" "Aaron" "Alvin" "Simon"
x int [1:3] 1 2 3
y chr [1:3] "a" "b" "c"
z logi [1:3] TRUE TRUE FALSE

Files Plots Packages Help Viewer

R: Reverse Elements - Find in Topic

rev {base} R Documentation

Reverse Elements

Description

rev provides a reversed version of its argument. It is generic function with a default method for vectors and one for [dendrogram](#)s.

Note that this is no longer needed (nor efficient) for obtaining vectors sorted into descending order, since that is now rather more directly achievable by [sort\(x, decreasing = TRUE\)](#).

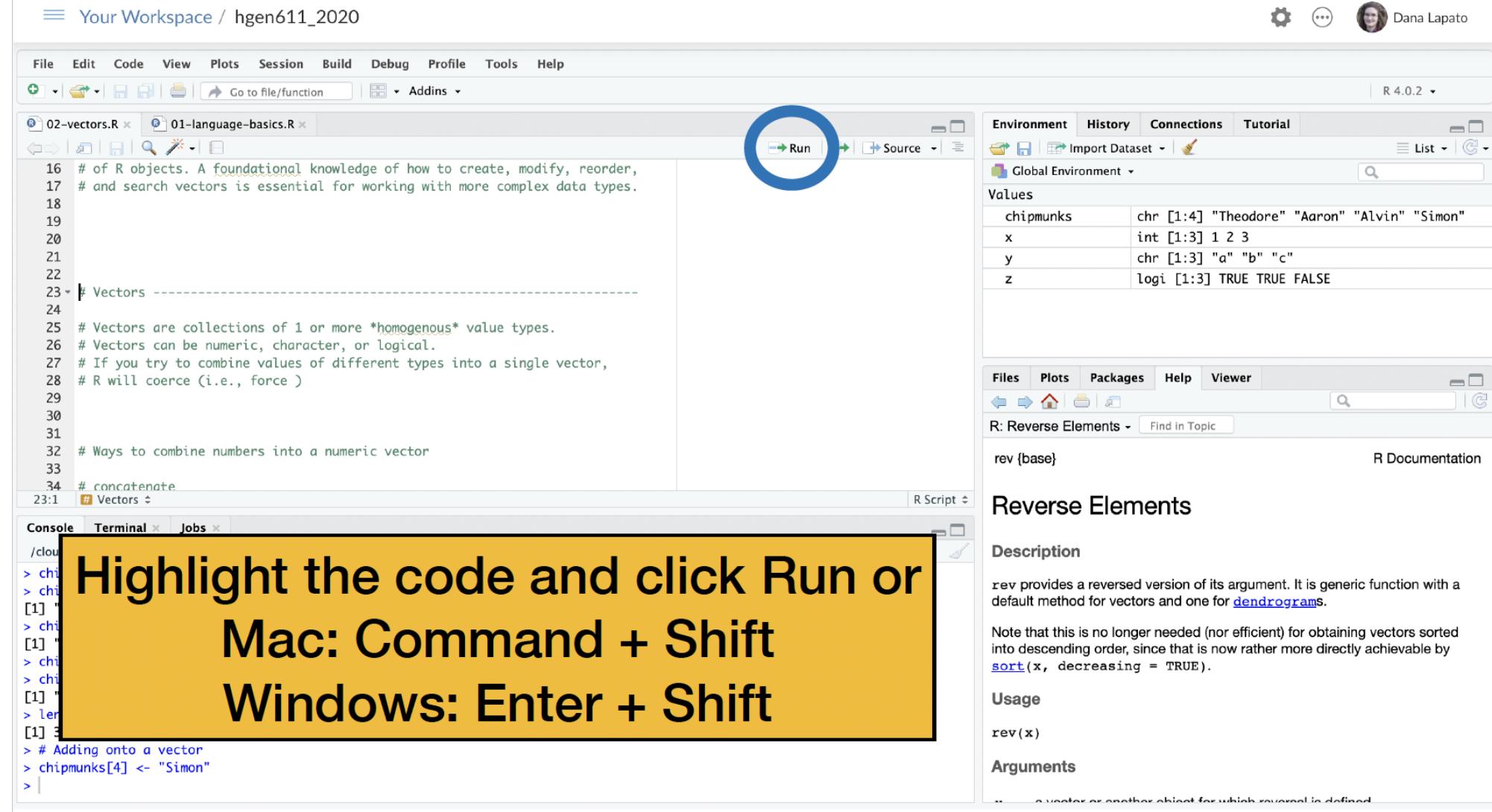
Usage

rev(x)

Arguments

... A vector or another object for which reversal is defined.

Highlight the code and click Run or
Mac: Command + Shift
Windows: Enter + Shift



After I run my code where should I look to see the output?

- A. Text Editor Pane
- B. Environment Pane
- C. Files Pane
- D. Console

After I run my code where should I look to verify it ran and see the output?

- A. Text Editor Pane
- B. Environment Pane
- C. Files Pane
- D. Console 

The best answer is the console. You *might* see changes in the environment pane depending on the code you run, but console is still the best answer.

R Language Basics

R Language

At its core, R is a calculator.

$$2 + 4$$

$$3 * 5$$

R will perform basic math calculations and follow the

R Language

R can evaluate logical comparisons.

$2 > 4$

R will return FALSE
in the console.

R is an object-oriented language

“put” arrow,
synonymous to = but NOT ==

Object

x <- 5

y <- 1:10

z <- c(1, 3, 5, 7, 11)

Concatenate/combine

Functions are commands applied to objects

Predefined
function

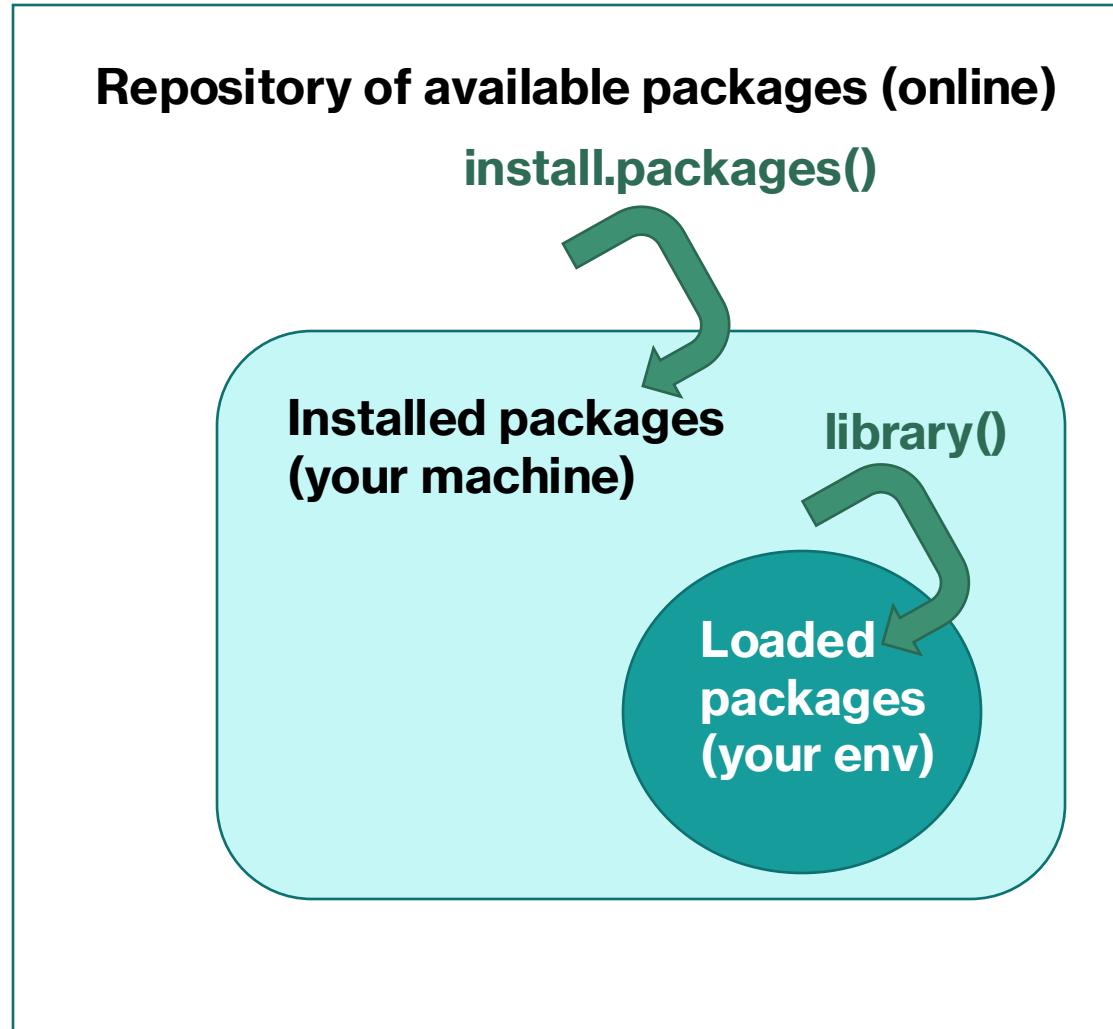
mean(z)

Object on which
the function is
applied on to

meanOfX <- mean(z)

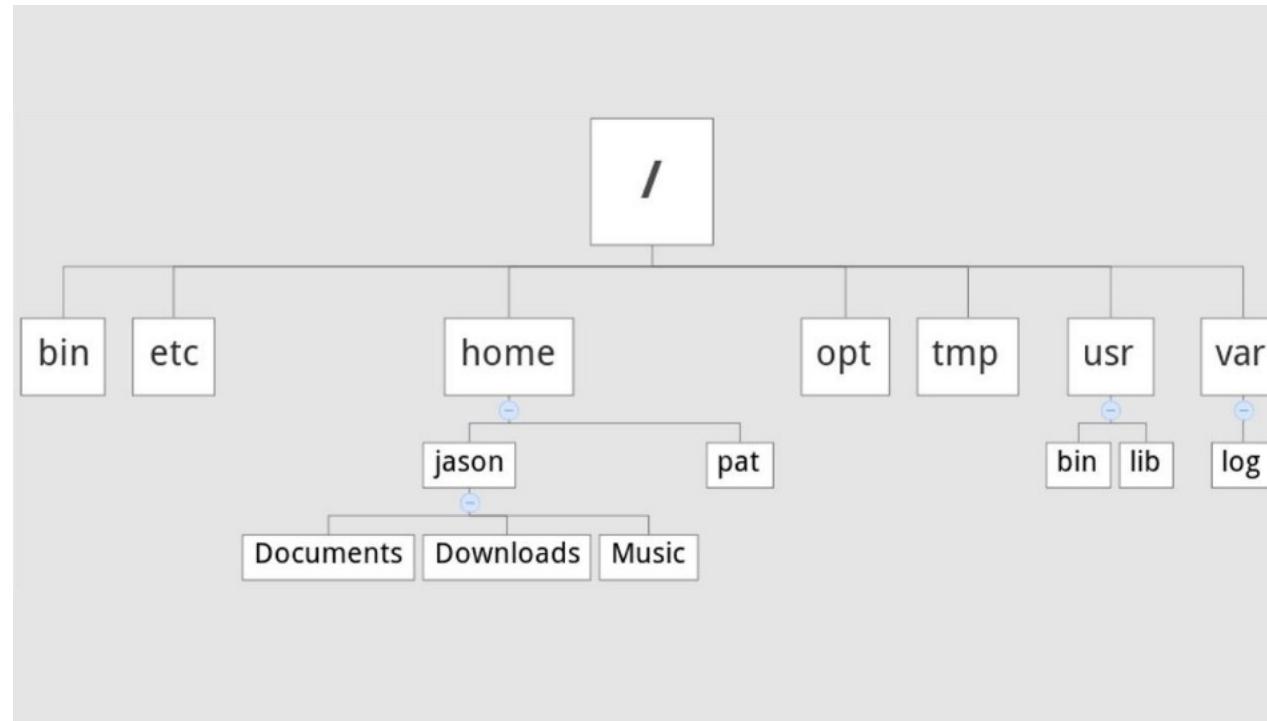
Store output in
object

Packages are collections of functions



Directories are folders

They are navigated by text using file paths



Check Your Understanding

Where can I view and edit my scripts?

The screenshot shows the RStudio interface with the following components:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- File Explorer:** Shows files 02-vectors.R and 01-language-basics.R.
- Code Editor:** Displays R code related to vectors and their manipulation.
- Console:** Shows a session history with commands like `rev` and `length` applied to the `chipmunks` vector.
- Environment:** Shows the global environment with variables `chipmunks`, `x`, `y`, and `z`.
- Documentation:** A detailed page for the `rev` function, including its description, usage, and arguments.

Which window pane shows the list of R objects I've created?

The screenshot displays the RStudio interface with the following components:

- Top Bar:** Your Workspace / hgen611_2020, gear icon, user profile (Dana Lapato), and R 4.0.2 dropdown.
- Code Editor:** Shows a script named "02-vectors.R" with R code related to vectors. The code includes comments explaining vector creation, modification, and concatenation.
- Environment Browser:** Shows the Global Environment pane with objects defined:

Object	Type	Value
chipmunks	chr	[1:4] "Theodore" "Aaron" "Alvin" "Simon"
x	int	[1:3] 1 2 3
y	chr	[1:3] "a" "b" "c"
z	logi	[1:3] TRUE TRUE FALSE
- Help Documentation:** Shows the "Reverse Elements" page from the base package. It includes sections for Description, Usage, and Arguments.

If I can run code in the console, why would I bother using the text editor?

The screenshot shows the RStudio interface with the following components:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- File Explorer:** Shows files 02-vectors.R and 01-language-basics.R.
- Code Editor:** Displays R code related to vectors. Lines 23-34 are expanded to show examples of vector creation and manipulation.
- Console:** Shows a session history with commands like `rev(chipmunks[2])` and `length(chipmunks)`.
- Environment Tab:** Shows the global environment with objects chipmunks, x, y, and z.
- Help Documentation:** A modal window for the `rev` function. It includes:
 - Description:** `rev` provides a reversed version of its argument. It is generic function with a default method for vectors and one for [dendrogram](#)s.
 - Usage:** `rev(x)
 - Arguments:** x – a vector or another object for which reversal is defined.

What is this thing?

The screenshot shows the RStudio interface with a large blue arrow pointing from the code editor towards the documentation pane.

Code Editor: Displays two files: `02-vectors.R` and `01-language-basics.R`. The `02-vectors.R` file contains R code related to vectors, including concatenation and reversing elements. The `01-language-basics.R` file is partially visible.

```
16 # of R objects. A foundational knowledge of how to create, modify, reorder,
17 # and search vectors is essential for working with more complex data types.
18
19
20
21
22
23 # Vectors -----
24
25 # Vectors are collections of 1 or more *homogenous* value types.
26 # Vectors can be numeric, character, or logical.
27 # If you try to combine values of different types into a single vector,
28 # R will coerce (i.e., force)
29
30
31
32 # Ways to combine numbers into a numeric vector
33
34 # concatenate
35 # Vectors $
```

Console: Shows a series of R commands and their outputs. The user is demonstrating how to reverse the elements of a vector named `chipmunks`.

```
> chipmunks[2] <- "Aaron"
> chipmunks
[1] "Alvin"    "Aaron"     "Theodore"
> chipmunks
[1] "Alvin"    "Aaron"     "Theodore"
> chipmunks[c(1, 3)] <- c("Theodore", "Alvin")
> chipmunks
[1] "Theodore" "Aaron"     "Alvin"
> length(chipmunks)
[1] 3
> # Adding onto a vector
> chipmunks[4] <- "Simon"
> |
```

Environment: Shows the global environment with variables `chipmunks`, `x`, `y`, and `z`.

Variables	Type	Value
chipmunks	chr	[1:4] "Theodore" "Aaron" "Alvin" "Simon"
x	int	[1:3] 1 2 3
y	chr	[1:3] "a" "b" "c"
z	logi	[1:3] TRUE TRUE FALSE

Documentation: The `rev` function is being documented. It provides a reversed version of its argument. It is generic function with a default method for vectors and one for [dendrograms](#). Note that this is no longer needed (nor efficient) for obtaining vectors sorted into descending order, since that is now rather more directly achievable by [sort\(x, decreasing = TRUE\)](#).

Usage: `rev(x)`

Arguments: `x` – a vector or another object for which reversal is defined.

Wrap-up

- Review the syllabus, mark your calendars!
- Increasing accessibility and affordability of technology is allowing access to larger and more complex datasets. Bioinformatics is required to handle these datasets
- R is the language, RStudio is the program we will be writing R code in
- R is an object-oriented language