



CREATING THE NEXT®

R Learning Guide

MGT 6203 – Data Analytics for Business

Prepared By:

Praneeth Pabba

Teaching Assistant

pabbapraneeth@gatech.edu

Table of Contents

Kickstarting R Preparation for this course	3
What do you need to know in R to help with the course?	3
Classifying students and their needs	3
Resources for learning R	4
R Learning Plans for Various Students	5
Basic building blocks	5
Working with the directory structure	6
Ranges, Lists and Vectors	6
Matrices and Dataframes:	7
Basic Graphing/Plotting	7
Getting Data from various sources:	8
Dplyr/Tidyverse (Data Cleaning and transformation):	8
Regression	9
Why R?	9
Installing R	10
Installing R Studio	13
Swirl	17
Why do we think swirl is great?	17
Installing swirl	18
Installing courses in swirl	19
Installing swirl courses useful for this class	20
Quick Tour of swirl courses	21

This document is intended to cover aspects related to R – installing R and RStudio, resources to learn R, R resources useful to excel in this course and using swirl package to learn R.

This document is **catered towards beginners**, with little or no introduction to programming, but is a very useful reference for everyone taking this course.

Kickstarting R Preparation for this course

What do you need to know in R to help with the course?

We believe that knowing the following concepts in R will help you breeze through initial couple of weeks in the course:

- Basic building blocks: R console input and evaluation, working with constants and variables, assignment operator, basic commands such as: ?, View, rm, etc.
- Working with directory structure: Concept of Working Directory, getting and setting working directory, navigating through directories, listing files, etc.
- Ranges, Lists, and Vectors: Creating and manipulating ranges of numbers, basic list/vector commands and functions (length, concatenation, etc.) and basic list/vector arithmetic.
- Matrices and Dataframes: Tabular data representation, manipulation such as sub-setting, binding and concatenating tables, creating new columns, mutating columns, filtering and selecting data, handling missing data, grouping data, etc.
- Basic Graphing/Plotting: Using plot() and ggplot2 packages for basic plotting
- Getting Data from various sources: Loading data from CSV, Excel, XML, etc. and checking if data is loaded appropriately
- Dplyr: Understanding pipe/chaining operator and using dplyr verbs for quickly cleaning and transforming datasets
- Regression: Functions to perform regression, checking regression statistics, etc.

Classifying students and their needs

Before we provide a detailed list of resources for each subtopic, we want to classify course students into different segments. A logical way of doing is using students' experience with programming – beginner, normal and advanced.

However, we take a different approach to this classification. We classify as:

- **Students with Extra Time/Energy**: Students who are willing to go the extra mile to learn R systematically and enjoy its long-term benefits, because they have/willing to put in extra time and effort as needed.
- **Students with Other Commitments**: We understand that many students doing this course have other work/family commitments and would want to do well in this course by learning an adequate amount of R needed for this course.

- **Students skimming through:** Some students might be auditing the course and might not be very serious about learning nitty-gritty details of the course/R language or maybe already have good programming knowledge and are just skimming through to understand the essence of the course.

Identifying yourself with one of these groups will help you understand what resources are most suitable for you to learn these topics.

Resources for learning R

Before we draft learning plans for all types of students, these are some of our strongly recommended resources:

- 1) Data Science Specialization by Johns Hopkins University on Coursera:
<https://www.coursera.org/specializations/jhu-data-science>
This is one of the oldest and best courses in R (I remember doing this course in 2014). Johns Hopkins is a strong advocate of R language, with a lot of their research producing wonderful R libraries.
It is an 8-course specialization. Courses [2,3,4](#) and [7](#) are most relevant to this course.
- 2) Swirl Courses: <https://swirlstats.com/students.html>
Swirl is a package to learn R in R. A lot of swirl courses are built using material from Coursera courses above. Check [more about swirl](#) and [why we really like it](#). So, if you ever feel that any part of the swirl course is too complicated, you can refer back to actual Coursera courses with the same name!!
Explore Swirl Course Network: <http://swirlstats.com/scn/>
- 3) Marin Stats Lectures (YouTube Channel):
<https://www.youtube.com/user/marinstatlectures/playlists>
A quick and very useful set of informative videos to cover various aspects of R programming
- 4) Datacamp Courses for R:
<https://www.datacamp.com/courses/free-introduction-to-r>
<https://www.datacamp.com/courses/intermediate-r>
<https://www.datacamp.com/community/tutorials/tidyverse-tutorial-r>
Another popular and must-check R learning resources similar to Coursera courses. These courses are also very interactive.
- 5) **[VERY IMPORTANT]** R Studio cheat sheets: I cannot stress more on the importance of these cheat sheets: <https://www.rstudio.com/resources/cheatsheets/>

They are the most essential tools for people beginning to learn R and for those who quickly want to refer to various concepts. They are extremely self-sufficient to learn concepts such as dplyr even though they are just “cheat sheets”. Keep all these cheat sheets handy.

Most useful cheat sheets:

- [Base R Cheat Sheet](#)
- [Data Import Cheat Sheet](#)

- [Data Table Cheat Sheet](#) (data.table data manipulations)
- [Data Transformation Cheat Sheet](#) (dplyr)
- [Data Visualization Cheat Sheet](#) (ggplot2)
- [Dates and Times Cheat Sheet](#) [Good to Have]
- [Apply functions Cheat Sheet](#) [Good to Have]
- [Data Analysis Cheat Sheet](#) [Quite Useful]
- [R Commands Cheat Sheet](#) [Quite Useful]

R Learning Plans for Various Students

Before checking out learning plans, we strongly recommend you to complete these steps:

- 1) Install Base R [\[Instructions below\]](#)
- 2) Install R Studio [\[Instructions below\]](#)
- 3) [Install Swirl](#) and [Swirl Courses](#) [Instructions below]
- 4) Check how to [navigate swirl courses](#) [Instructions below]

Learning plans for each section for various students:

Basic building blocks

Students with Extra Time/Energy:

Complete “R Programming” swirl course sub-topic: 1: Basic Building Blocks

Check following week 1 videos in [R Programming](#) Coursera Course:

Writing Code / Setting Your Working Directory (Windows) 7m

R Console Input and Evaluation 4m

Data Types - R Objects and Attributes 4m

Data Types - Vectors and Lists 6m

Data Types – Factors 4m

Data Types - Names Attribute 1m

If time permits, check all Week 1 videos of [R Programming](#) Coursera course

Students with Other Commitments:

Complete “R Programming” swirl course sub-topic: 1: Basic Building Blocks

Students skimming through:

Do nothing or refer cheat sheets below

All students, refer Cheatsheets: [Base R Cheat Sheet](#), [R Commands Cheat Sheet](#)

Working with the directory structure

Students with Extra Time/Energy, Students with Other Commitments:

Complete “R Programming” swirl course sub-topic: 2: Workspace and Files

Students skimming through:

Check “help” or R-documentation for: `getwd()`, `setwd()`, `ls()`, `rm()`, `list.files()`, `data()`, `View()`

All students, refer Cheatsheets: [Base R Cheat Sheet](#), [R Commands Cheat Sheet](#)

Ranges, Lists and Vectors

Students with Extra Time/Energy

Complete “R Programming” swirl course sub-topics: 3: Sequences of Numbers, 4: Vectors, 5: Missing Values, 6: Subsetting Vectors

Check following week 1 videos in [R Programming](#) Coursera Course:

Data Types - Vectors and Lists 6m

Data Types - Missing Values 2m

Subsetting – Basics 4m

Subsetting – Lists 4m

Subsetting - Removing Missing Values 3m

Vectorized Operations 3m

Students with Other Commitments:

Complete “R Programming” swirl course sub-topics: 3: Sequences of Numbers, 4: Vectors, 5: Missing Values, 6: Subsetting Vectors

Students skimming through:

Check following Marin Stats [Lecture](#)

All students, refer Cheatsheets: [Base R Cheat Sheet](#)

Matrices and Dataframes:

NOTE: Dataframe manipulations are important to know (check [Data Table Cheat Sheet](#)). However, dataframe operations such as selecting, filtering, sorting, etc. will also be covered as a part of dplyr learning and are not necessary to know. However, subsetting dataframes needs to be learned.

Students with Extra Time/Energy

Complete “R Programming” swirl course sub-topics: 7: Matrices and Data Frames, 12: Looking at Data

If time permits, also check swirl sub-courses: 10: lapply and sapply, 11: vapply and tapply

Check videos 8, 9, 10, 11 and 12 from Marin Stats R Programming YouTube Playlist:

<https://www.youtube.com/playlist?list=PLqzoL9-eJTNATicffatWXTEjwMq6N0Sf3>

Students with Other Commitments:

Complete “R Programming” swirl course sub-topics: 7: Matrices and Data Frames, 12: Looking at Data

Check video 11 from Marin Stats R Programming Playlist.

If time permits, check videos 8, 9, 10 and 12 from Marin Stats R Programming YouTube Playlist:

<https://www.youtube.com/playlist?list=PLqzoL9-eJTNATicffatWXTEjwMq6N0Sf3>

Students skimming through:

Use “help” or R-documentation to find about: `dim()`, `data.frame()`, `nrow()`, `ncol()`, `head()`, `tail()`

If time permits, check videos 9, 10 and 11 from Marin Stats R Programming YouTube Playlist:

<https://www.youtube.com/playlist?list=PLqzoL9-eJTNATicffatWXTEjwMq6N0Sf3>

All students, refer Cheatsheets: [Base R Cheat Sheet](#), [Data Analysis Cheat Sheet](#)

Basic Graphing/Plotting

NOTE: ggplot2() is a massive plotting package. Do not spend more time/energy on learning this package. It is better to refer [Data Visualization Cheat Sheet](#) for most types of graphs and graph configurations you may need.

Students with Extra Time/Energy, Students with Other Commitments, Students skimming through:

Complete “R Programming” swirl course sub-topic: 15: Base Graphics

Quickly refer to various aspects of this article for various ggplot2 customizations:

<http://r-statistics.co/ggplot2-Tutorial-With-R.html>

<http://r-statistics.co/Complete-Ggplot2-Tutorial-Part1-With-R-Code.html>

For most types of graphs and graph configurations, the best reference is [Data Visualization Cheat Sheet](#)
Quickly skimming through this material is enough for all types of readers.

Getting Data from various sources:

Students with Extra Time/Energy:

Check all week 1 videos on [Getting and Cleaning Data](#) Coursera Course. (Video Time: 1hr 3min)

Check the cheat sheet: [Data Analysis Cheat Sheet](#)

Students with Other Commitments:

Read the cheat sheet: [Data Analysis Cheat Sheet](#)

Refer cheat sheet: [Data Import Cheat Sheet](#)

Students skimming through:

All students, refer cheat sheets: [Data Import Cheat Sheet](#) , [Data Analysis Cheat Sheet](#)

Dplyr/Tidyverse (Data Cleaning and transformation):

NOTE: This is a very core topic for data analysis. Please spend extra time if needed to understand this well.

Students with Extra Time/Energy:

Check this video on YouTube: <https://www.youtube.com/watch?v=jWjqLW-u3hc&t=1510s>

This is a very good starting resource, explains dplyr very well and lasts for close to 40 mins.

For a hands-on experience, Complete “[Getting and Cleaning Data](#)” swirl course sub-topics: 1: Manipulating Data with dplyr (We would’ve also recommended completing 2: Grouping and Chaining with dplyr and 3: Tidying Data with tidyr but we noticed some technical issues while completing the swirl courses. You can try to do them but skip/exit if you run into issues)

Alternative Resource:

Check out data camp article/videos on introduction to tidyverse:
<https://www.datacamp.com/community/tutorials/tidyverse-tutorial-r>

If you’re interested enough, check out more about dplyr pipe operator here:
<https://www.datacamp.com/community/tutorials/pipe-r-tutorial>

Students with Other Commitments:

Check this video on YouTube until you understand piping and dplyr verbs in detail:

<https://www.youtube.com/watch?v=jWjqLW-u3hc&t=1510s>

For a hands-on experience, Complete “[Getting and Cleaning Data](#)” swirl course sub-topics: 1: Manipulating Data with dplyr

Students skimming through:

Skim through this video: <https://www.youtube.com/watch?v=jWjqLW-u3hc&t=1510s>

All students, definitely refer to this very well curated cheat sheet: [Data Transformation Cheat Sheet](#)

Regression

NOTE: Regression is covered as a part of this course. Please refer to these resources if you intend to explore it in greater depth.

Students with Extra Time/Energy:

Check out all 12 videos of Regression YouTube Playlist of Marin Stats Lectures:

<https://www.youtube.com/playlist?list=PLqzoL9-eJTnBJrvFcN-ohc5G13E7Big0e>

Approximate runtime for all videos: 73 min

If you need hands-on experience, complete all sub-courses of “Regression Models” swirl course.

Students with Other Commitments:

Check out videos 1,2,3,7,8 and 10 of Regression YouTube Playlist of Marin Stats Lectures:

<https://www.youtube.com/playlist?list=PLqzoL9-eJTnBJrvFcN-ohc5G13E7Big0e>

Approximate runtime: 40 min

If time permits, complete 1 to 6 sub-courses of “Regression Models” swirl course.

Students skimming through:

Skim through videos 1,2,3 and 10 of Regression YouTube Playlist of Marin Stats Lectures:

<https://www.youtube.com/playlist?list=PLqzoL9-eJTnBJrvFcN-ohc5G13E7Big0e>

Why R?

R is the most popular software for statistical and advanced data analytics in the past decade. Although other languages such as Python are catching up quickly, R is still most popular for statistical and data analysis. Python is a bit more useful for machine learning applications, but both of them are open source, provide numerous packages for many fields of study and have extensive developer community support.

R can also support large datasets and more complex data operations than Excel. Check [this](#) article and Google for interesting R-Excel and R-Python debates if interested.

Installing R

R language distribution needs to be installed to use the R language. This distribution is known as R base distribution.

R base has a minimal R code editor to write code. However, **we strongly recommend using R Studio for R programming** (we show how to get R Studio later). Note that **R base installation needs to be done before RStudio is installed.**

Instructions:

- 1) Go to R-official site: <https://www.r-project.org/>
- 2) Click on “download R” or “CRAN” hyperlinks in the page:



[Home]

Download

[CRAN](#)

R Project

About R

Logo

Contributors

What's New?

Reporting Bugs

Conferences

Search

Get Involved: Mailing Lists

Developer Pages

R Blog

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To download R, please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- **R version 3.6.1 (Action of the Toes)** has been released on 2019-07-05.
- useR! 2020 will take place in St. Louis, Missouri, USA.
- **R version 3.5.3 (Great Truth)** has been released on 2019-03-11.
- The R Foundation Conference Committee has released a [call for proposals](#) to host useR! 2020 in North America.
- You can now support the R Foundation with a renewable subscription as a [supporting member](#)

- 3) You will be led to CRAN (Comprehensive R Archive Network) mirrors page with various mirrors across the globe providing R software to download. Select the mirror that is closest to your location. United States users can choose any of these mirrors:

USA

<https://cran.cnr.berkeley.edu/> →

<http://cran.stat.ucla.edu/>

<https://mirror.las.iastate.edu/CRAN/>

<https://ftp.usgs.iu.edu/CRAN/>

<https://rweb.crmda.ku.edu/cran/>

<https://cran.mtu.edu/>

<https://repo.miserver.it.umich.edu/cran/>

<http://cran.wustl.edu/>

<http://archive.linux.duke.edu/cran/>

<https://cran.case.edu/>

<https://ftp.osuosl.org/pub/cran/>

<http://lib.stat.cmu.edu/R/CRAN/>

<http://cran.mirrors.hoobly.com/>

<https://mirrors.nics.utk.edu/cran/>

<https://cran.revolutionanalytics.com/>

European College Servers

University of California, Berkeley, CA
University of California, Los Angeles, CA
Iowa State University, Ames, IA
Indiana University
University of Kansas, Lawrence, KS
Michigan Technological University, Houghton, MI
MBNI, University of Michigan, Ann Arbor, MI
Washington University, St. Louis, MO
Duke University, Durham, NC
Case Western Reserve University, Cleveland, OH
Oregon State University
Statlib, Carnegie Mellon University, Pittsburgh, PA
Hoobly Classifieds, Pittsburgh, PA
National Institute for Computational Sciences, Oak Ridge, TN
Revolution Analytics, Dallas, TX

- 4) You will be led to the R-download page now for that mirror, with links to download for various operating systems. This article focuses on Windows, but the steps are almost the same for other operating systems too. macOS users can check this [link](#) for more guidance.

The Comprehensive R Archive Network

Download and Install R
Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#) →

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

- 5) We need to install the R “base”. We need not worry about other distributions in this window. Click either on “base” or “install R for the first time” hyperlinks:

R for Windows

Subdirectories:

[base](#)

[contrib](#)

[old contrib](#)

[Rtools](#)


Binaries for base distribution. This is what you want to [install R for the first time](#).

Binaries of contributed CRAN packages (for R \geq 2.13.x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.

Binaries of contributed CRAN packages for outdated versions of R (for R $<$ 2.13.x; managed by Uwe Ligges).

Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

- 6) Click on the latest R download hyperlink as shown below:


CRAN
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

About R
[R Homepage](#)
[The R Journal](#)

Software
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

Documentation
[Manuals](#)
[FAQs](#)
[Contributed](#)

R-3.6.1 for Windows (32/64 bit)

[Download R 3.6.1 for Windows](#) (81 megabytes, 32/64 bit)
[Installation and other instructions](#)
[New features in this version](#)

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of md5sum for windows; both [graphical](#) and [command line versions](#) are available.

Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is [CRAN_MIRROR/bin/windows/base/release.htm](#).

Last change: 2019-07-05

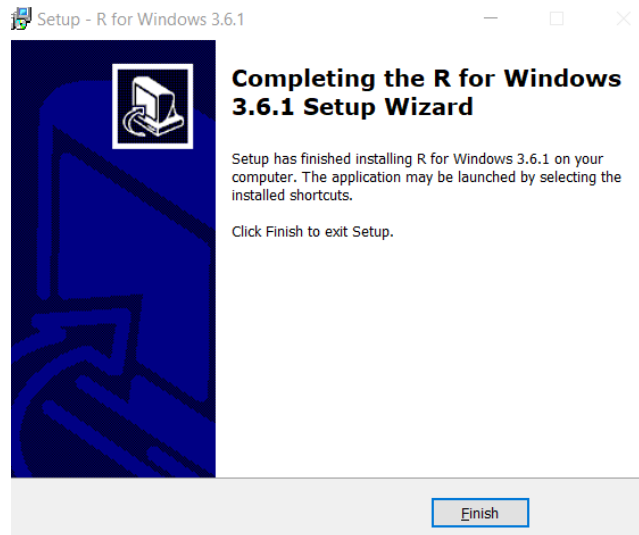


R-3.6.1-win.exe

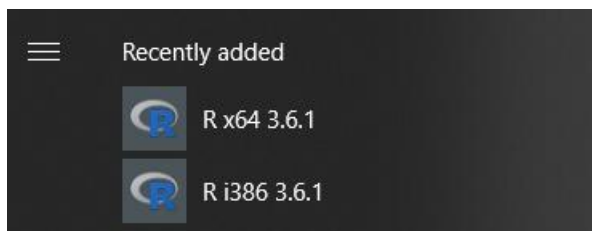
Show all

- 7) Open the R-installer after downloading:
- 8) Use all the recommended/default settings in the installer wizard and install the software:
- Accept the License Agreement
 - Keep the default Destination Location (Change if needed, but recommended not to change)
 - Install all components

- d. Do not customize startup options (advanced setting – do NOT select Yes)
- e. Create a Start Menu folder (Change if needed)
- f. Keep both registry entries options selected (Up to you if you want to create additional shortcuts)
- g. Click “Finish” to complete installation



- 9) Click on Windows or Start Button and search for R. You will find 2 versions of R:



R x64 3.6.1 is 64-bit R software.

R i386 3.6.1 is 32-bit R software.

You can use either 32-bit or 64-bit R software (64-bit is recommended). However, 32-bit operating systems may not support 64-bit R software.

You can check your operating system version by searching for “System information” in Start menu or Control Panel and look for System type property:

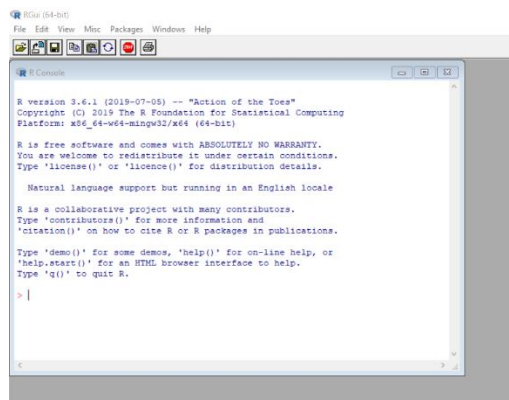
Item	Value
OS Name	Microsoft Windows 10 Pro
Version	10.0.18362 Build 18362
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	DESKTOP-U51RPP8
System Manufacturer	GIGABYTE
System Model	AERO 15XV8
System Type	<u>x64-based PC</u>
System SKU	P65XV8
Processor	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, 2208 Mhz, 6 C...

If System Type is x64-based PC, it is a 64-bit operating system.

If System Type is x86-based PC, it is a 32-bit operating system.

You can check [this](#) article for more information on this topic.

- 10) After clicking on either software (64-bit: R x64 3.6.1 say), you will see the R Gui window open. You can start writing R code here. (We strongly recommend installing R Studio on top of this. Check instructions below).



Installing R Studio

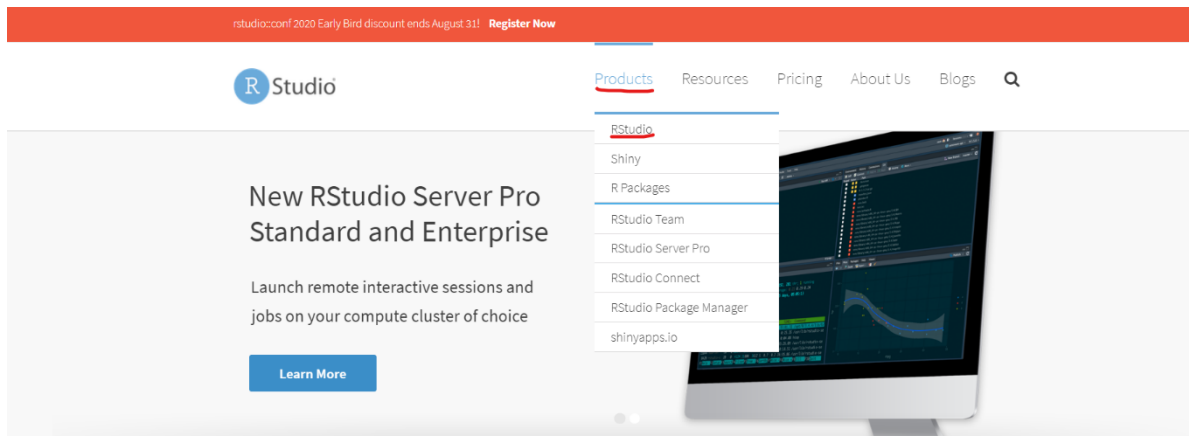
R Studio is an IDE (Integrated Development Environment) used on top of R. It provides a really useful development experience with many useful features for R programming. **We strongly recommend using R Studio for R programming assignments in this course.**

IMPORTANT NOTE: R Studio needs to be installed only after installing R properly. Please follow the above steps to install R and then install R Studio.

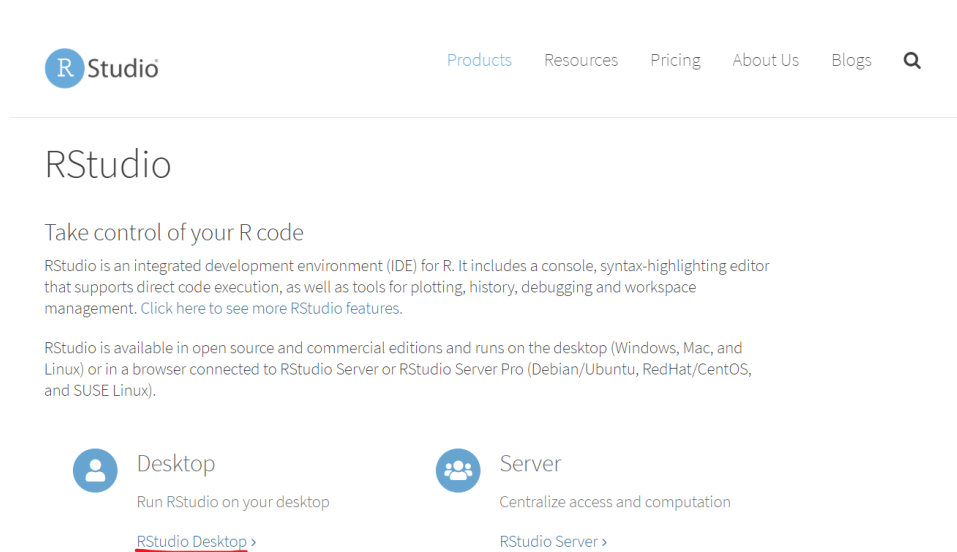
This [video](#) highlights some useful features of R Studio while comparing them with base R GUI.

- 1) Go to R-Studio official page:




2) Click on Products -> R Studio





3) Click on “R Studio Desktop” hyperlink:



4) Scroll a bit and click on “Download R Studio Desktop” button:


 <ul style="list-style-type: none"> Access RStudio locally Syntax highlighting, code completion and smart indentation 			Products Resources Pricing About Us Blogs 
Overview	<ul style="list-style-type: none"> Execute R code directly from the source editor Quickly jump to function definitions Easily manage multiple working directories using projects Integrated R help and documentation Interactive debugger to diagnose and fix errors quickly Extensive package development tools 	All of the features of open source; plus: <ul style="list-style-type: none"> A commercial license for organizations not able to use AGPL software Access to priority support 	
Support	Community forums only	<ul style="list-style-type: none"> Priority Email Support 8 hour response during business hours (ET) 	
License	AGPL v3	RStudio License Agreement	
Pricing	Free	\$995/year	
DOWNLOAD RSTUDIO DESKTOP 		BUY NOW	

5) Click on “Download” button under Free R Studio List:



[Products](#)
[Resources](#)
[Pricing](#)
[About Us](#)
[Blogs](#)


Choose Your Version of RStudio

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace. [Learn More about RStudio features.](#)



RStudio's new solution for every professional data science team. RStudio Team includes RStudio Server Pro, RStudio Connect and RStudio Package Manager. [LEARN MORE](#)

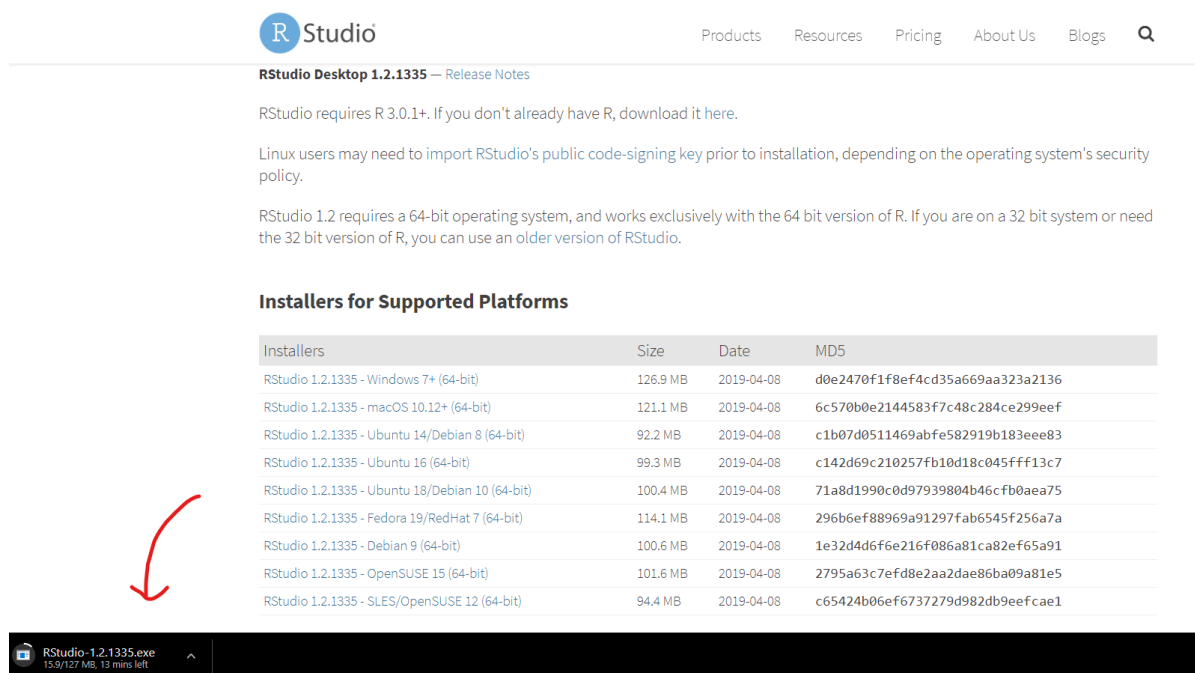
RStudio Desktop Open Source License FREE DOWNLOAD  Learn More	RStudio Desktop Commercial License \$995 per year BUY Learn More	RStudio Server Open Source License FREE DOWNLOAD Learn More	RStudio Server Pro Commercial License \$4,975 per year (5 Named Users) BUY Evaluation Learn More
---	---	--	--

6) Click on installer corresponding to your operating system (check for Windows below):

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.2.1335 - Windows 7+ (64-bit)	126.9 MB	2019-04-08	d0e2470f1f8ef4cd35a669aa323a2136
RStudio 1.2.1335 - macOS 10.12+ (64-bit)	121.1 MB	2019-04-08	6c570b0e2144583f7c48c284ce299eef
RStudio 1.2.1335 - Ubuntu 14/Debian 8 (64-bit)	92.2 MB	2019-04-08	c1b07d0511469abfe582919b183eee83
RStudio 1.2.1335 - Ubuntu 16 (64-bit)	99.3 MB	2019-04-08	c142d69c210257fb10d18c045fff13c7
RStudio 1.2.1335 - Ubuntu 18/Debian 10 (64-bit)	100.4 MB	2019-04-08	71a8d1990c0d97939804b46c fb0aea75
RStudio 1.2.1335 - Fedora 19/RedHat 7 (64-bit)	114.1 MB	2019-04-08	296b6ef88969a91297fab6545f256a7a
RStudio 1.2.1335 - Debian 9 (64-bit)	100.6 MB	2019-04-08	1e32d4d6f6e216f086a81ca82ef65a91
RStudio 1.2.1335 - OpenSUSE 15 (64-bit)	101.6 MB	2019-04-08	2795a63c7efd8e2aa2dae86ba09a81e5
RStudio 1.2.1335 - SLES/OpenSUSE 12 (64-bit)	94.4 MB	2019-04-08	c65424b06ef6737279d982db9eefcae1

7) Click on the installer when downloaded:



RStudio Desktop 1.2.1335 — Release Notes

RStudio requires R 3.0.1+. If you don't already have R, download it [here](#).

Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy.

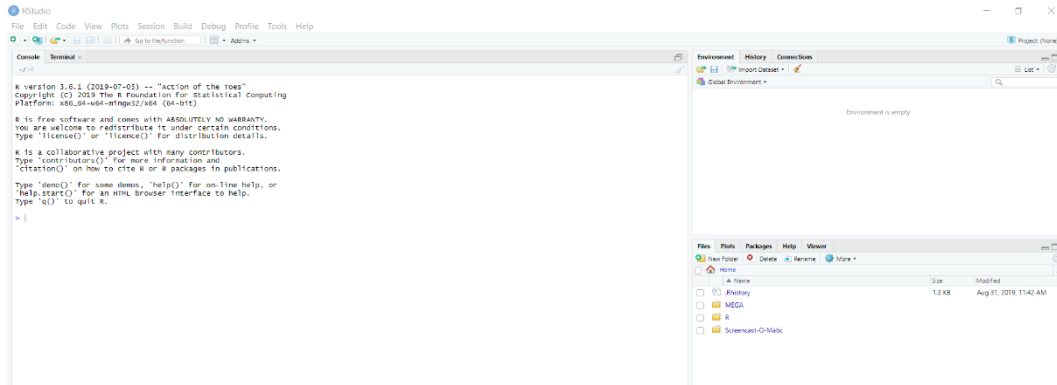
RStudio 1.2 requires a 64-bit operating system, and works exclusively with the 64 bit version of R. If you are on a 32 bit system or need the 32 bit version of R, you can use an older version of RStudio.

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.2.1335 - Windows 7+ (64-bit)	126.9 MB	2019-04-08	d0e2470f1f8ef4cd35a669aa323a2136
RStudio 1.2.1335 - macOS 10.12+ (64-bit)	121.1 MB	2019-04-08	6c570b0e2144583f7c48c284ce299eef
RStudio 1.2.1335 - Ubuntu 14/Debian 8 (64-bit)	92.2 MB	2019-04-08	c1b07d0511469abfe582919b183eee83
RStudio 1.2.1335 - Ubuntu 16 (64-bit)	99.3 MB	2019-04-08	c142d69c210257fb10d18c045fff13c7
RStudio 1.2.1335 - Ubuntu 18/Debian 10 (64-bit)	100.4 MB	2019-04-08	71a8d1990c0d97939804b46c fb0aea75
RStudio 1.2.1335 - Fedora 19/RedHat 7 (64-bit)	114.1 MB	2019-04-08	296b6ef88969a91297fab6545f256a7a
RStudio 1.2.1335 - Debian 9 (64-bit)	100.6 MB	2019-04-08	1e32d4d6f6e216f086a81ca82ef65a91
RStudio 1.2.1335 - OpenSUSE 15 (64-bit)	101.6 MB	2019-04-08	2795a63c7efd8e2aa2dae86ba09a81e5
RStudio 1.2.1335 - SLES/OpenSUSE 12 (64-bit)	94.4 MB	2019-04-08	c65424b06ef6737279d982db9eefcae1

RStudio-1.2.1335.exe
15.9/127 MB, 13 mins left

- 8) Select default settings and install R Studio. Click “Finish” to exit the installation wizard.
- 9) Launch R Studio (Click on Start/Windows and search for R Studio). You will see an application open similar to this:



Swirl

[Swirl](#) is an R library which creates a highly interactive learning experience to learn R. It is designed to learn/teach R in R. We strongly recommend that you install R Studio before using swirl so that most of the lectures on swirl can be completed without encountering issues.

Why do we think swirl is great?

There are many reasons why we support swirl over other good resources to learn R:

- It starts from ABSOLUTE BASIC !! swirl can be used to master the most basic concepts of R even for people who have never programmed in their life before!!
- It is interactive and feedback-based, making you write small pieces of code to move forward in the lessons (you can skip some of them too if you like). We think that it's a better way to learn a programming language rather than just keeping on watching videos. Videos are great!! But they need to be supplemented with practical exercises.
- You can move at your own pace. You can save your progress and restart where you left off!
- It covers most of the topics we believe will make your progress through the course a cakewalk!
- It's useful for us instructors too, and it's open-source!! Many useful lectures in swirl have been created by programmers like us, and when you learn R, you can contribute too.

The next parts of the guide show you how to install swirl, access and explore swirl course repository, explore swirl courses and also our recommendations of which swirl courses would be needed based on your programming background and proficiency.

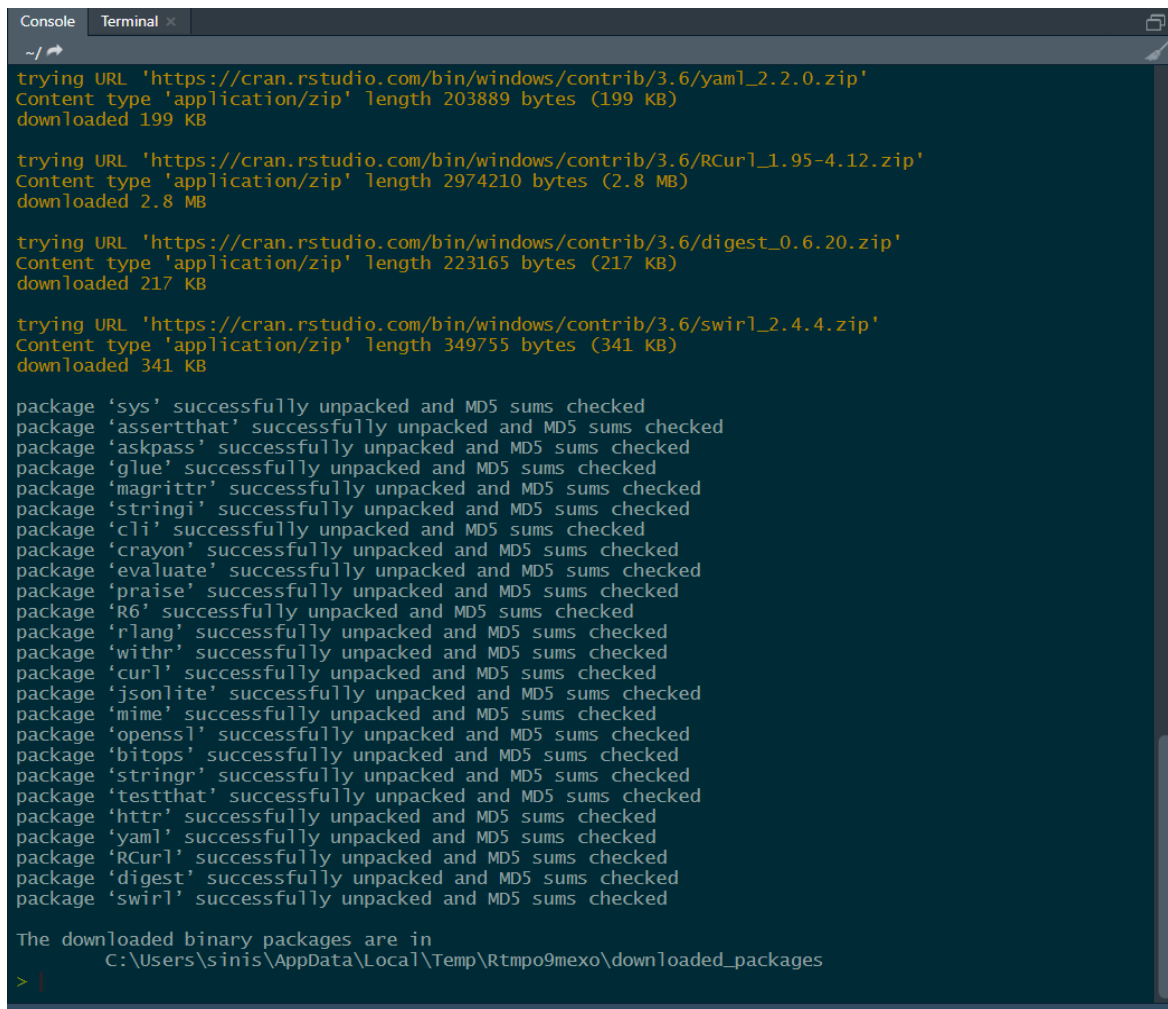
Installing swirl


Swirl can be installed in a jiffy!!

Open R Studio and go to R console.

Type the following in console: **install.packages("swirl")** and press enter.

Swirl and any other packages which are needed to run swirl will be automatically installed. You will see some windows opening and quickly downloading packages that may be necessary to run swirl. You will see something like below in console when swirl is successfully installed:



```
~/ 
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/yaml_2.2.0.zip'
Content type 'application/zip' length 203889 bytes (199 KB)
downloaded 199 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/RCurl_1.95-4.12.zip'
Content type 'application/zip' length 2974210 bytes (2.8 MB)
downloaded 2.8 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/digest_0.6.20.zip'
Content type 'application/zip' length 223165 bytes (217 KB)
downloaded 217 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/swirl_2.4.4.zip'
Content type 'application/zip' length 349755 bytes (341 KB)
downloaded 341 KB

package 'sys' successfully unpacked and MD5 sums checked
package 'assertthat' successfully unpacked and MD5 sums checked
package 'askpass' successfully unpacked and MD5 sums checked
package 'glue' successfully unpacked and MD5 sums checked
package 'magrittr' successfully unpacked and MD5 sums checked
package 'stringi' successfully unpacked and MD5 sums checked
package 'cli' successfully unpacked and MD5 sums checked
package 'crayon' successfully unpacked and MD5 sums checked
package 'evaluate' successfully unpacked and MD5 sums checked
package 'praise' successfully unpacked and MD5 sums checked
package 'R6' successfully unpacked and MD5 sums checked
package 'rlang' successfully unpacked and MD5 sums checked
package 'withr' successfully unpacked and MD5 sums checked
package 'curl' successfully unpacked and MD5 sums checked
package 'jsonlite' successfully unpacked and MD5 sums checked
package 'mime' successfully unpacked and MD5 sums checked
package 'openssl' successfully unpacked and MD5 sums checked
package 'bitops' successfully unpacked and MD5 sums checked
package 'stringr' successfully unpacked and MD5 sums checked
package 'testthat' successfully unpacked and MD5 sums checked
package 'httr' successfully unpacked and MD5 sums checked
package 'yaml' successfully unpacked and MD5 sums checked
package 'RCurl' successfully unpacked and MD5 sums checked
package 'digest' successfully unpacked and MD5 sums checked
package 'swirl' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\sinis\AppData\Local\Temp\Rtmpo9mexo\downloaded_packages
> |
```

Swirl is now installed. In order to use swirl, we first need to include “swirl” package in our code. We can do that by typing **library(swirl)** and pressing enter in R console. You will see a message like this:

```
The downloaded binary packages are in
  C:\Users\sinis\AppData\Local\Temp\Rtmpo9mexo\downloaded_packages
> library(swirl)

| Hi! Type swirl() when you are ready to begin.
> |
```

To begin, we type `swirl()` and press enter in R console.

Installing courses in swirl

We need to install swirl courses before we start using swirl. Swirl prompts will lead you to install “R Programming” course, but let us install all courses useful for this class directly.

The R command used to install a swirl course is **install_course(“course_name_in_quotes”)**.

Swirl courses can be found [here](#), and the course repository is found [here](#).

Let us say we want to install swirl course “Getting and Cleaning Data”:

- Go to swirl course network: <http://swirlstats.com/scn/>
- Click on “Browse Courses” and “By Title”
- Click on “Getting and Cleaning Data” hyperlink:

Swirl Courses Organized by Title

A

- [Advanced R Programming](#) by Roger Peng

C

- [ConoceR](#) by David Duncan

D

- [Data Science and R](#) by Wush Wu

E

- [Exploratory Data Analysis](#) by Team swirl

G

- [Getting and Cleaning Data](#) by Team swirl
- [Google Forms Course](#) by Sean Kross

P

- [Practical Data Science in R](#) by John Fox

- Use the command given on the page to install the swirl course:

Installation

```
swirl::install_course("Getting and Cleaning Data")
```

```
>
> swirl::install_course("Getting and Cleaning Data")
|=====| 100%
| Course installed successfully!
> |
```

NOTE: If you have already run **library(swirl)** command in R console, you need not use the prefix **swirl::** in front of **install_course()** command. However, using the prefix will install the course even when swirl package is not included and is a safer way to install a course.

Installing swirl courses useful for this class

We believe the following swirl courses are useful for this class:

- R Programming
- Getting and Cleaning Data
- Regression Analysis

To install all these courses, run the following commands in R console:

```
swirl::install_course("R Programming")
swirl::install_course("Getting and Cleaning Data")
swirl::install_course("Regression Models")
```

```
>
> swirl::install_course("Getting and Cleaning Data")
|=====| 100%
| Course installed successfully!
> swirl::install_course("R Programming")
|=====| 100%
| Course installed successfully!
> swirl::install_course("Regression Models")
|=====| 100%
| Course installed successfully!
> |
```

Quick Tour of swirl courses

After installing swirl and swirl courses, and including swirl (`library(swirl)` in the console), let us type `swirl()` and enter:

```
> swirl()
| Welcome to swirl! Please sign in. If you've been here before, use the same name as you did
| then. If you are new, call yourself something unique.
What shall I call you? Praneeth Pabba
| Thanks, Praneeth Pabba. Let's cover a couple of quick housekeeping items before we begin our
| first lesson. First of all, you should know that when you see '...', that means you should
| press Enter when you are done reading and ready to continue.
... <-- That's your cue to press Enter to continue
```

Enter your name at “what shall I call you?”

IMPORTANT NOTE: swirl saves your course progress and you can exit a course midway. When you come back to swirl again, **if you enter this same name again, swirl will automatically reload the latest course checkpoint.**

As mentioned above, **“...” in swirl is always a cue to press enter and continue.**

Swirl is an interactive course and such cues come in quite often.

```
... <-- That's your cue to press Enter to continue

| Also, when you see 'ANSWER:', the R prompt (>), or when you are asked to select from a list,
| that means it's your turn to enter a response, then press Enter to continue.

Select 1, 2, or 3 and press Enter

1: Continue.
2: Proceed.
3: Let's get going!

Selection: 3

| You can exit swirl and return to the R prompt (>) at any time by pressing the Esc key. If you
| are already at the prompt, type bye() to exit and save your progress. When you exit properly,
| you'll see a short message letting you know you've done so.

| When you are at the R prompt (>):
| -- Typing skip() allows you to skip the current question.
| -- Typing play() lets you experiment with R on your own; swirl will ignore what you do...
| -- UNTIL you type nxt() which will regain swirl's attention.
| -- Typing bye() causes swirl to exit. Your progress will be saved.
| -- Typing main() returns you to swirl's main menu.
| -- Typing info() displays these options again.

| Let's get started!
|
...
```

After pressing enter, you notice that any options given need to be selected with their numbers. Entering either of 1,2 or 3 in the options would've just continued the course.

Remember various options given to navigate the course!

```
| -- Typing skip() allows you to skip the current question.
| -- Typing play() lets you experiment with R on your own; swirl will ignore
what you do...
| -- UNTIL you type nxt() which will regain swirl's attention.
| -- Typing bye() causes swirl to exit. Your progress will be saved.
| -- Typing main() returns you to swirl's main menu.
| -- Typing info() displays these options again.
```

If you run into a difficult question and just want to skip it, use skip() in R prompt

If you want to run some code in R console but don't want to interfere with course flow, use play() to go into this “play mode”.

Use nxt() to come out of “play mode” above.

bye() just exits the swirl course. Normal R prompt resumes. You can use swirl() again to enter the course. swirl will resume at your latest checkpoint.

main() takes you to swirl main menu.

If you forget any of the options above, use info() in R console to get them again.

If you have various options and you want to select none of them, entering 0 often gets you out of that selection prompt.

You can navigate and complete swirl courses with these instructions.

```
Selection: 2
| Please choose a lesson, or type 0 to return to course menu.

1: Basic Building Blocks      2: Workspace and Files      3: Sequences of Numbers
4: Vectors                   5: Missing Values           6: Subsetting Vectors
7: Matrices and Data Frames  8: Logic                    9: Functions
10: lapply and sapply        11: vapply and tapply       12: Looking at Data
13: Simulation               14: Dates and Times         15: Base Graphics

Selection: 1
|
| 0%
| In this lesson, we will explore some basic building blocks of the R programming language.
...
|==
| 3%
| If at any point you'd like more information on a particular topic related to R, you can type
| help.start() at the prompt, which will open a menu of resources (either within RStudio or your
| default web browser, depending on your setup). Alternatively, a simple web search often yields
| the answer you're looking for.
...|
```

Once you enter a course, you will be seeing course progress on the right to indicate your progress.

Each swirl course typically takes 5-20 minutes based on your ease and programming proficiency.

Self-practice might push up this time.

===== End of Document =====