Introduction to R

Pwani R Workshop

8th July 2024

Welcome!

- Introductions

Overall learning objectives and scope

- Understand basic programming syntax
- Import and manipulate data in R
- Visualize data in R
- Explore relationships between variables

In this session, we will:

- Introduce R and R studio
- Set up and interact with R and R Studio
- Introduce terminology and basic R syntax
- Learn about ‘packages’

What is R?

- R is a language and environment for statistical computing and graphics developed in 1991
- **R**oss Ihaka and **R**obert Gentleman at the University of Auckland, New Zealand developed R
- R is both open source and open development

Why R?

- Free (open source)
- High level language designed for statistical computing
- Powerful and flexible - especially for data wrangling and visualization
- Extensive add-on software (packages)
- Strong community

Expectations

What do you hope to get out of the class?

Course format

- Interactive lecture in the morning.
- Hands-on practical experience in the afternoon.

Installing R and R Studio

- Install the latest R version
- Install RStudio

RStudio is an **integrated development environment** (IDE) that makes it easier to work with R.

More on that soon!

Basic terms

Glossary of R jargon

Package - a package in R is a bundle or “package” of code (and or possibly data) that can be loaded together for easy repeated use or for **sharing** with others.

Packages are analogous to a software application like Microsoft Word on your computer. Your operating system allows you to use it, just like having R installed (and other required packages) allows you to use packages.

[source]

Working with R – RStudio

RStudio is an Integrated Development Environment (IDE) for R

- Helps you write code - makes suggestions
- Helps you view the output of your code
- Helps you find errors
- Is NOT a dropdown statistical tool (such as Stata or SPSS)

[source]

RStudio used to be the name of a company that is now called Posit.

RStudio

Easier working with R

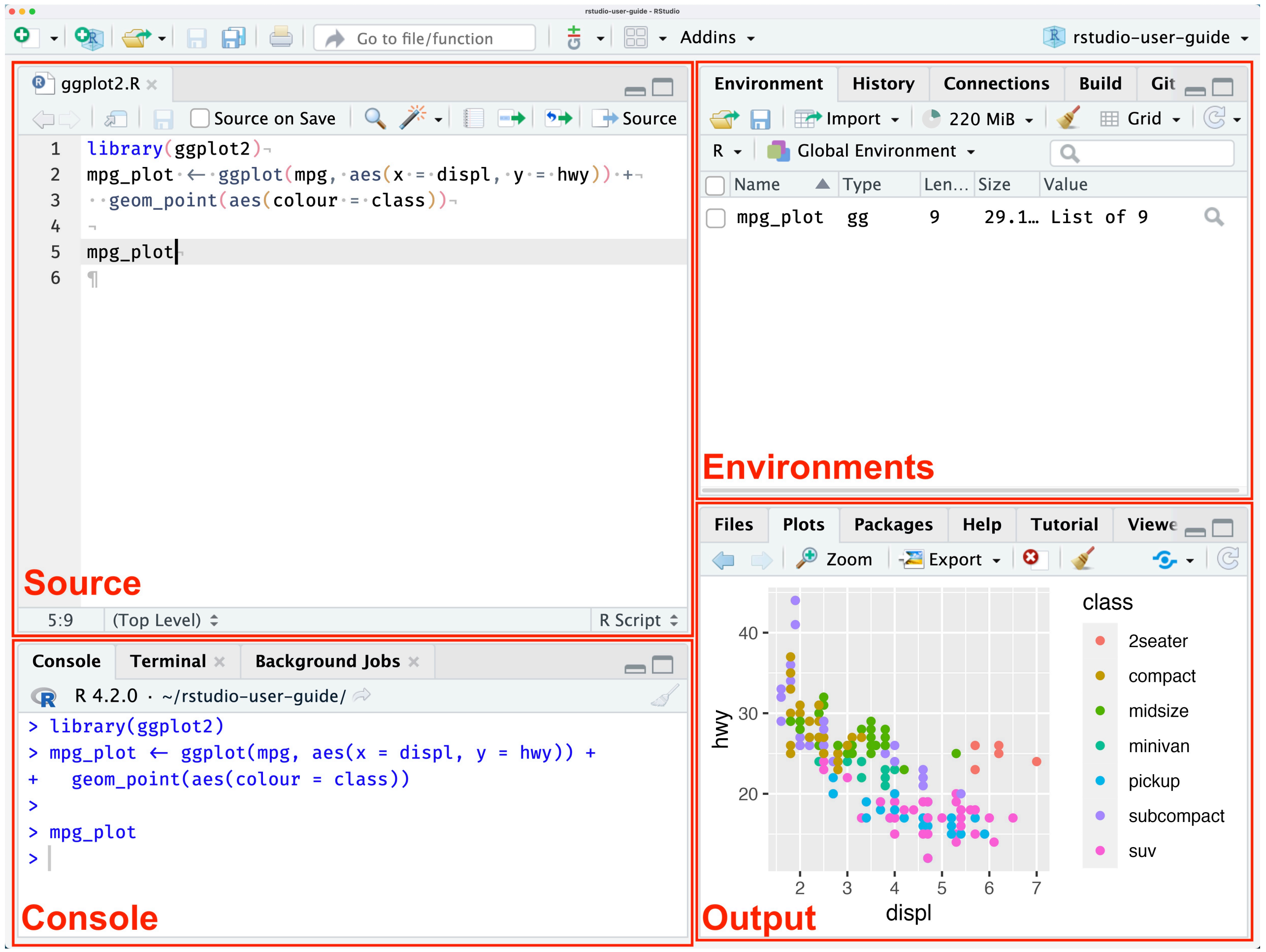
- Syntax highlighting, code completion, and smart indentation
- Easily manage multiple working directories and projects

More information

- Workspace browser and data viewer
- Plot history, zooming, and flexible image and file export
- Integrated R help and documentation

RStudio

First it is important to be familiar with the layout.



```r
library(ggplot2)
mpg_plot <- ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(colour = class))

mpg_plot
```

Name	Type	Len…	Size	Value
mpg_plot	gg	9	29.1…	List of 9

```
R 4.2.0 · ~/rstudio-user-guide/
> library(ggplot2)
> mpg_plot <- ggplot(mpg, aes(x = displ, y = hwy)) +
+   geom_point(aes(colour = class))
>
> mpg_plot
>
```

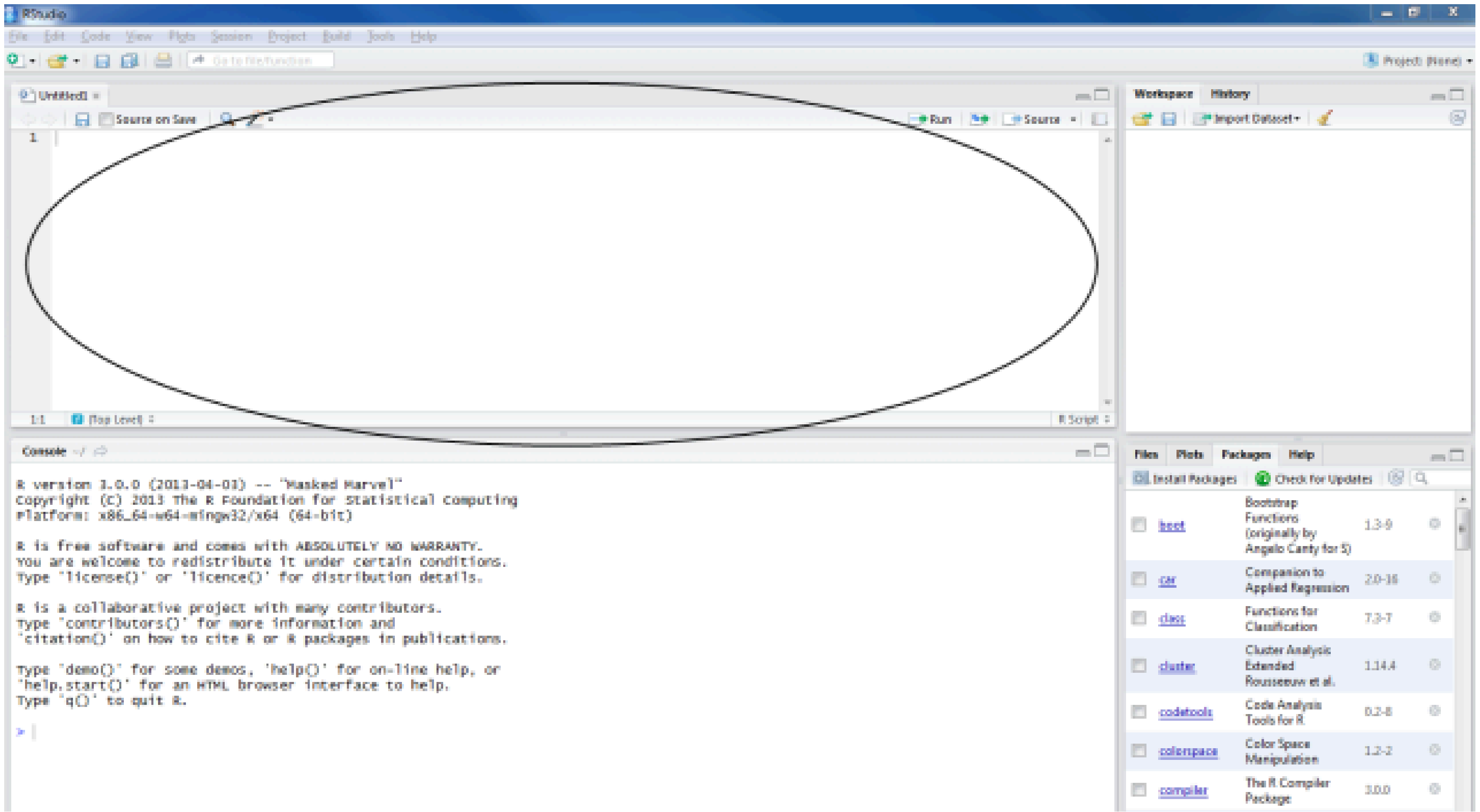
Working with R in R Studio - 2 major panes:

1. The **Source/Editor**: “Analysis” Script + Interactive Exploration
    - Static copy of what you did (reproducibility)
    - Top by default
2. The **R Console**: “interprets” whatever you type
    - Calculator
    - Try things out interactively, then add to your editor
    - Bottom by default

Source / Editor

- Where files open to
- Have R code and comments in them

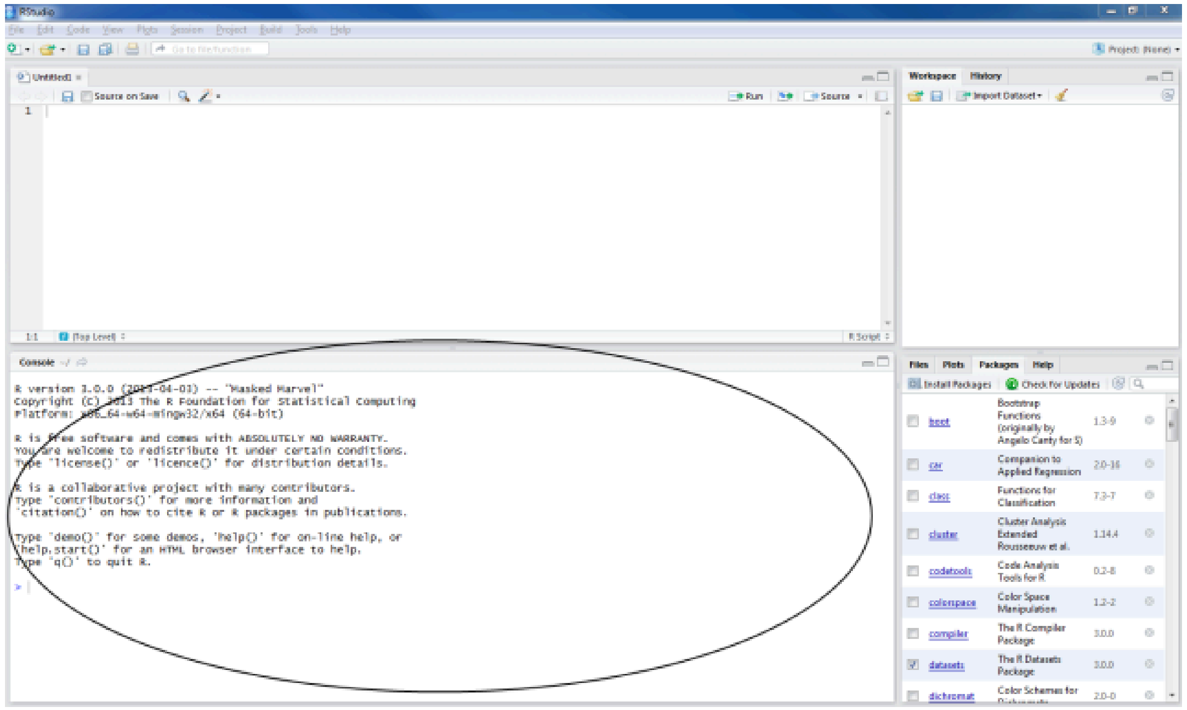
In a .R file (we call a script), code is saved on your disk



```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

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.

>
```

☐	boot	Bootstrap Functions (originally by Angelo Canty for S)	1.3-9
☐	car	Companion to Applied Regression	2.0-16
☐	class	Functions for Classification	7.3-7
☐	cluster	Cluster Analysis Extended Rousseeuw et al.	1.14.4
☐	codetools	Code Analysis Tools for R	0.2-8
☐	colorspace	Color Space Manipulation	1.2-2
☐	compiler	The R Compiler Package	3.0.0

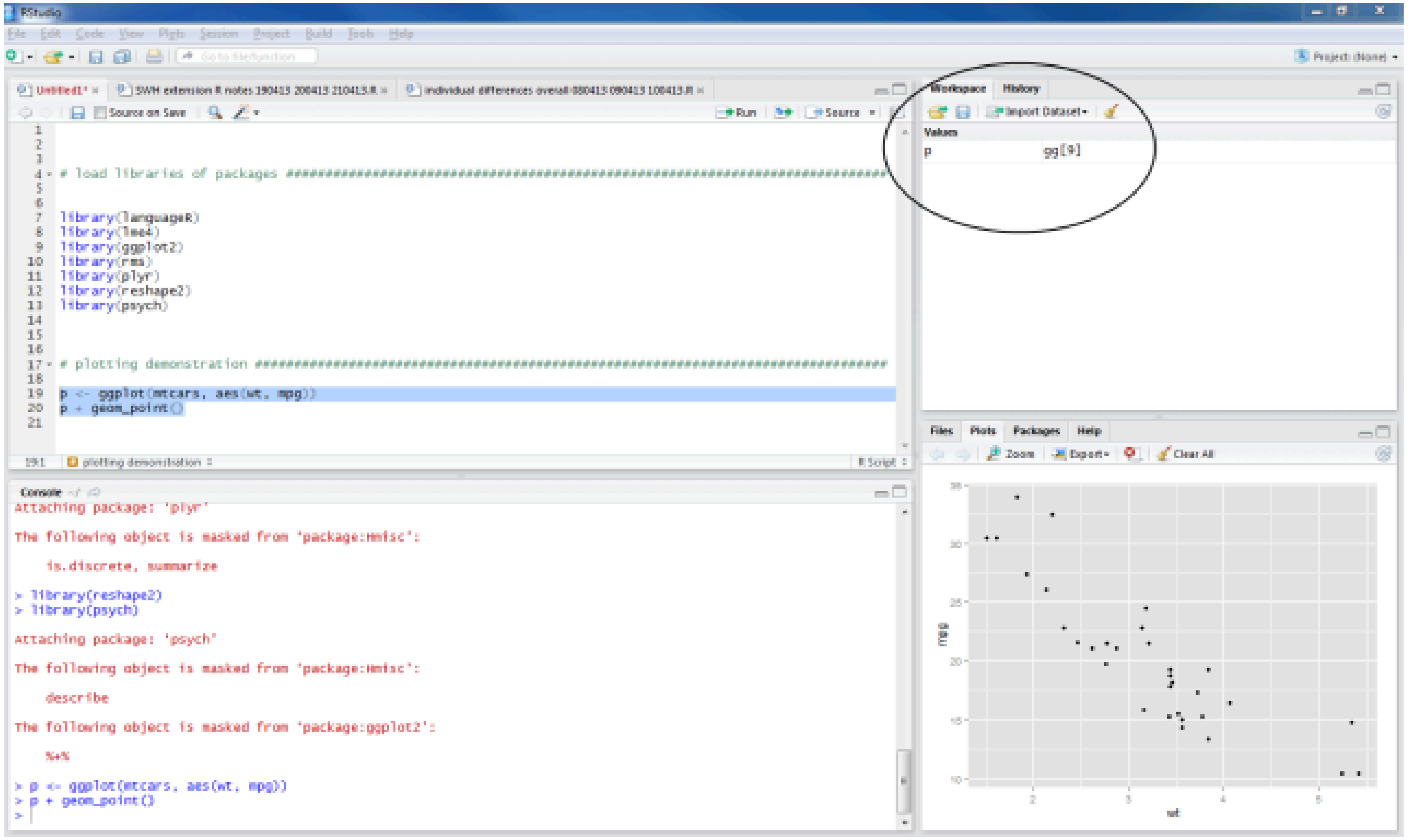
R Console



```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

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.

>
```

- Where code is executed (where things happen)
- You can type here for things interactively to test code
- Code is **not saved** on your disk

Workspace/Environment



```


# load libraries of packages ##########################################################################


library(languageR)
library(lme4)
library(ggplot2)
library(rms)
library(plyr)
library(reshape2)
library(psych)



# plotting demonstration ##############################################################################

p <- ggplot(mtcars, aes(wt, mpg))
p + geom_point()

```

```
Attaching package: 'plyr'

The following object is masked from 'package:Hmisc':

    is.discrete, summarize

> library(reshape2)
> library(psych)

Attaching package: 'psych'

The following object is masked from 'package:Hmisc':

    describe

The following object is masked from 'package:ggplot2':

    %+%

> p <- ggplot(mtcars, aes(wt, mpg))
> p + geom_point()
> 
```

Values

p	gg[9]

Workspace/Environment

- Tells you what **objects** are in R
- What exists in memory/what is loaded?/what did I read in?

History

- Shows previous commands.
- Also type the “up” key in the Console to scroll through previous commands

Other Panes

- **Files** - shows the files on your computer of the directory you are working in
- **Plots** - pictures and figures
- **Packages** - list of R packages that are loaded in memory
- **Help** - shows help of R commands
- **Viewer** - displays HTML content e.g interactive visualizations
- **Presentation** - used to display HTML slides

Basic R terms

Function - a function is a piece of code that allows you to do something in R. You can write your own, use functions that come directly from installing R, or use functions from additional packages.

You can think of a function as **verb** in R.

A function might help you add numbers together, create a plot, or organize your data. More on that soon!

```
sum(4,3)
```

```
## [1] 7
```

Basic terms

Argument - what you pass to a function

- can be data like the number 1 or 20234

```
sum(4, 3)
```

```
## [1] 7
```

- can be options about how you want the function to work such as `digits`

```
round(0.532, digits = 2)
```

```
## [1] 0.53
```

Basic terms

Object - an object is something that can be worked with or on in R - can be lots of different things! You can think of objects as **nouns** in R.

- a matrix of numbers
- a plot
- a function
- data

… many more

Variable and Sample

- **Variable**: something measured or counted that is a characteristic about a sample

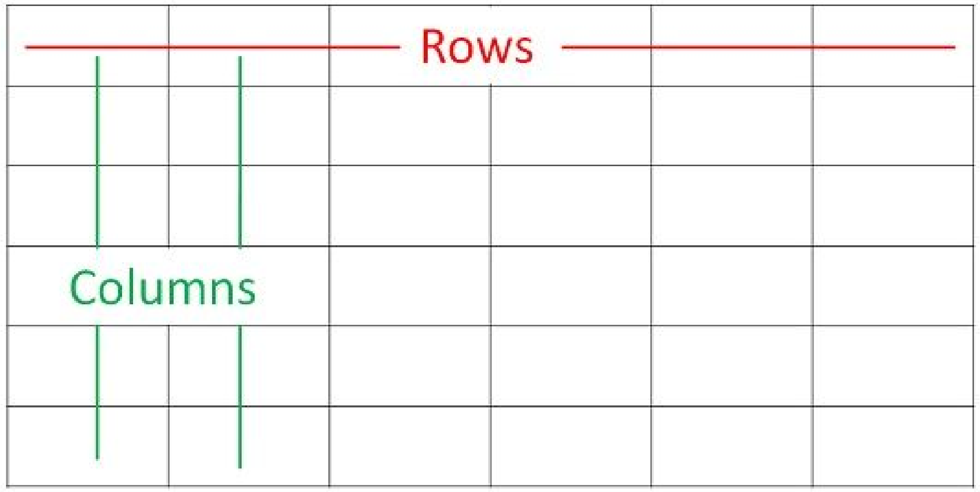
examples: temperature, length, count, color, category

- **Sample**: individuals that you have data about -

examples: people, houses, viruses etc.

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Columns and Rows



[source]

Sample = Row
Variable = Column

A data object that looks like this is called a **data frame**.

More on Functions and Packages

- When you download R, it has a “base” set of functions/packages (**base R**)
  - You can install additional packages for your uses from CRAN or GitHub
  - These additional packages are written by RStudio or R users/developers
  - There are also packages for bioinformatics available at Bioconductor

Using Packages

- Not all packages available on CRAN or GitHub are trustworthy!
- Posit makes many useful packages
- How to trust an R package
- Many packages have accompanying academic papers published in peer-reviewed journals
- Widely used packages have better documentation (official and in forums) and are more likely free of errors

Tidyverse and Base R: Two Dialects

We will mostly show you how to use tidyverse packages and functions.

This set of packages designed for data science make your code more **intuitive** as compared to the original older Base R.

Tidyverse advantages:
- **consistent structure** - making it easier to learn how to use different packages
- particularly good for **wrangling** (manipulating, cleaning, joining) data
- more flexible for **visualizing** data

Tidyverse packages are managed by a team of respected data scientists at Posit.

See this article for more info.

Package Installation

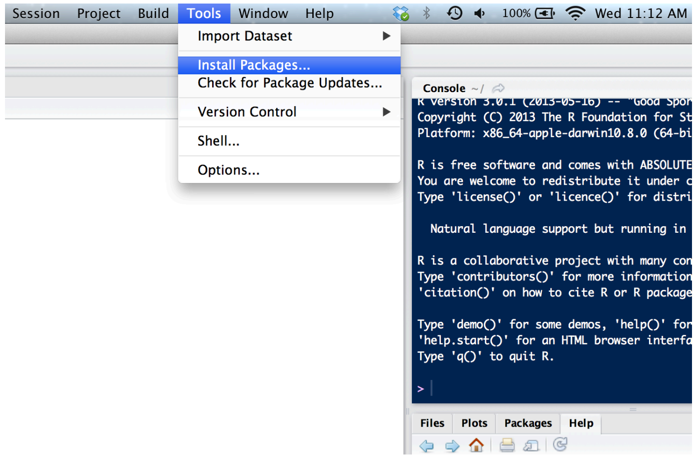
We will practice this in the hands-on session

Differs depending on the source (CRAN, GitHub, etc)

Must be done **once** for each installation of R (e.g., version 4.3 >> 4.4).

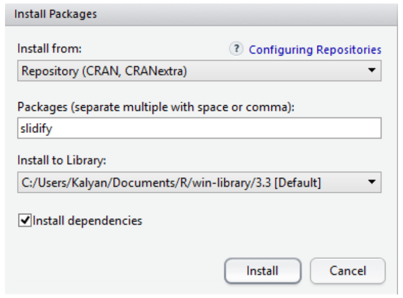
Package Installation: R Studio Menu

tools > Install Packages



```
R version 3.0.1 (2013-05-16) -- "Good Spor
Copyright (C) 2013 The R Foundation for St
Platform: x86_64-apple-darwin10.8.0 (64-bi

R is free software and comes with ABSOLUTE
You are welcome to redistribute it under c
Type 'license()' or 'licence()' for distri

  Natural language support but running in

R is a collaborative project with many con
Type 'contributors()' for more information
'citation()' on how to cite R or R package

Type 'demo()' for some demos, 'help()' for
'help.start()' for an HTML browser interfa
Type 'q()' to quit R.

>
```

Type in the package name to install.



Install from: Repository (CRAN, CRANextra)

Packages (separate multiple with space or comma): slidify

Install to Library: C:/Users/Kalyan/Documents/R/win-library/3.3 [Default]

[✓] Install dependencies

Package Installation: Using Code

We use a function called `install.packages()` for CRAN packages.

Here is an example where we “install” the `dplyr` package:

```r
install.packages("dplyr")
```

The package name is enclosed in quotation marks.

Loading packages

After installing packages, you will need to “load” them into memory so that you can use them.

This must be done **every time** you start R.

We use a function called `library` to load packages.

Here is an example where we “load” the `dplyr` package:

```
library(dplyr)
```

Quotation marks are optional.

Help!!!

Error messages can be scary!

We will show you how to resolve any installation issues

Summary

- R is a powerful data visualization and analysis software language.
- Add-on **packages** like the `tidyverse` can help make R more intuitive.
- **Functions** (like verbs) perform specific tasks in R and are found within packages.
- **Arguments** within functions specify how to perform a function.
- **Objects** (like nouns) are data or variables.
- RStudio makes working in R easier
- the Editor (top) is for static code like scripts or R Markdown documents
- The console is for testing code (bottom) - best to save your code though!