Welcome! Introduction to R

Outline

Links for getting started

Download R and Rstudio, respectively:
https://cran.r-project.org/index.html
https://rstudio.com/products/rstudio/download/

Quick guides for R:
https://rstudio.com/resources/cheatsheets/

Book - Hadley Wickham's R for data science:
https://r4ds.had.co.nz/

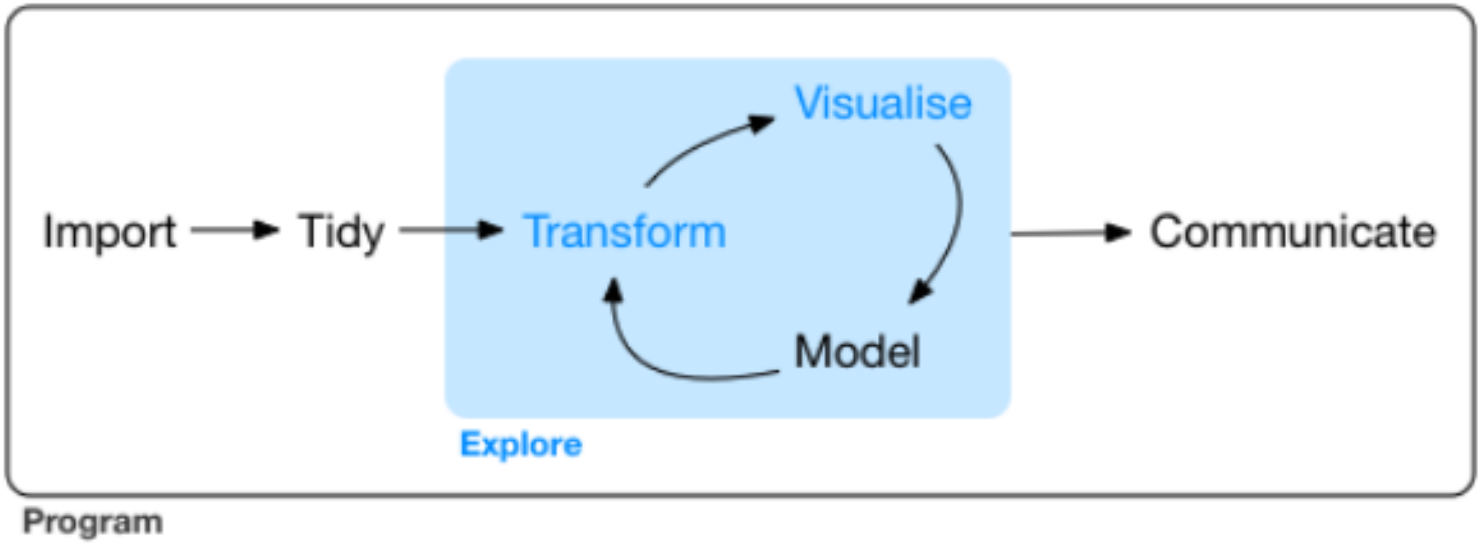
Swirl - Interactive learning for R:
https://swirlstats.com/

Why R?

Free and open-source
Interdisciplinary
Reproducibility
Publication-quality output
Rstudio available as IDE for R

Introduction

Data analysis: extraction of knowledge from data



Source: R for Data Science (Wickham & Grolemund, 2017)

Import

Types of data

.csv (comma-separated files)
.dat (generic data files)
.txt (text files)
.xls (Excel files)
.sas (SAS files)
.sav and .por (SPSS files)
.dta (Stata files)

...many others!

Tidy

Data wrangling (also called "munging" or "tidying") is assembling, organizing, cleaning, and transforming the data format in order to be more appropriate for specific tasks.

missing data does not spark joy

Tidy

One example: wide format versus long format

Name	Height	Weight
John	160	67
Christopher	182	78

Name	Attribute	Value
John	Height	160
John	Weight	67
Christopher	Height	182
Christopher	Weight	78

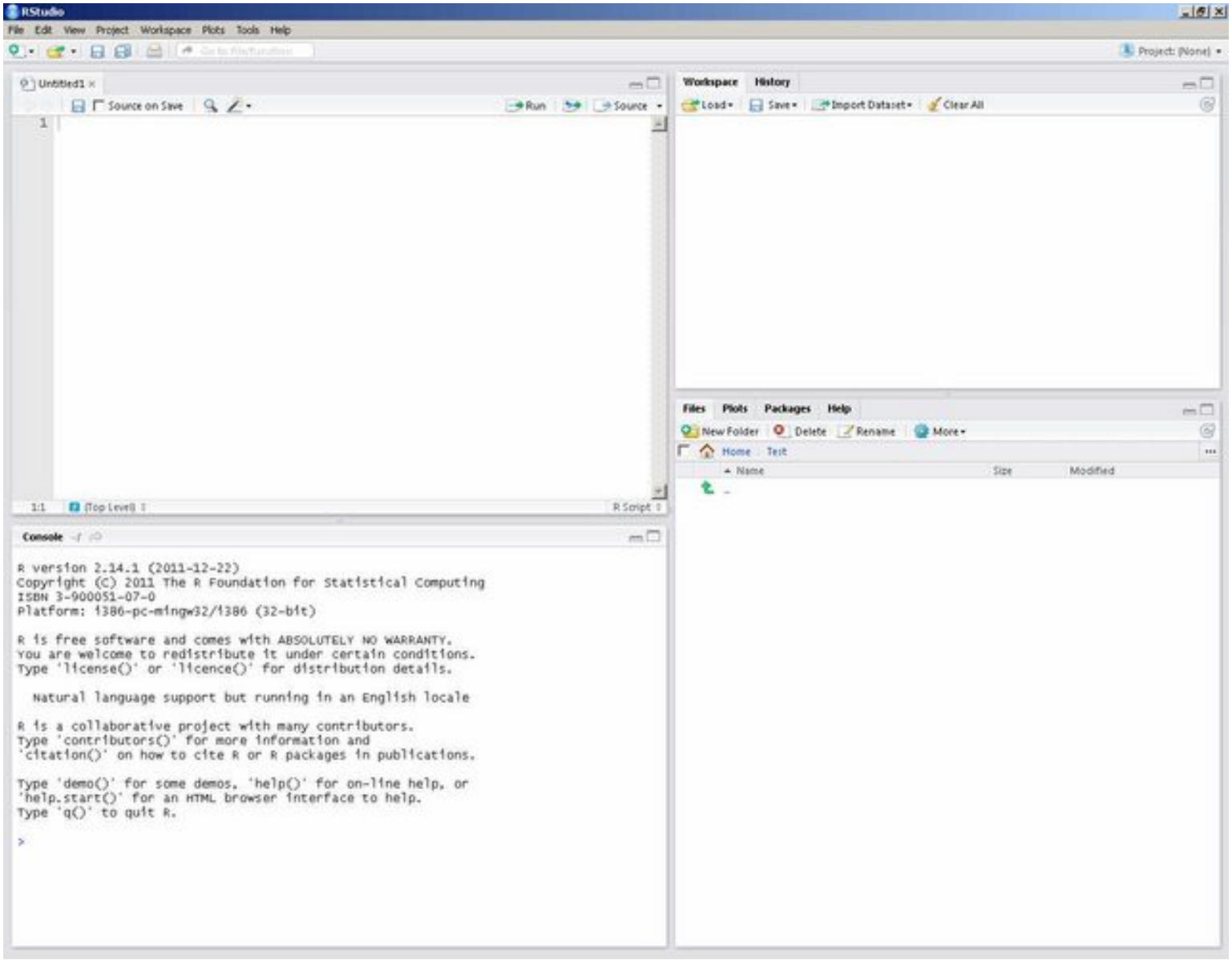
Source: https://www.quora.com/What-is-long-and-wide-format-data

Download software

Download R :
https://cran.r-project.org/index.html

Download R studio:
https://rstudio.com/products/rstudio/download/

Rstudio


RStudio
File Edit View Project Workspace Plots Tools Help
Project: (None)
Untitled1
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
Workspace History
Load
Save
Import Dataset
Clear All
Files Plots Packages Help
New Folder
Delete
Rename
More
Home Test
Name
Size
Modified
Console
R version 2.14.1 (2011-12-22)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-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.
>

Getting started

Project

Create self-contained working directory with all files. Packaged to share with collaborators and organize different tasks for a better work flow.

Script

A file that allows you to save code instead of working directly from the console.

Beyond the scope of this presentation, but worth mentioning is Rmarkdown. Allows high-quality documents to be produced so that figures/plots/graphics/code can be inserted directly into a Microsoft Word document (or HTML and PDF documents).

Cheat sheets

Cheat Sheets are quick references.
Start with: Base R

Cheat sheets

For more information about the Rstudio environment: Rstudio IDE cheat sheet.

Assignment

Calculations: PEMDAS, and be explicit (to multiply * must be used in equations).

Objects:
R is object-oriented, *everything* is an object. R uses $<-$ as a local assignment operator and is best to avoid using $=$ which is common in other programming languages.

Object Assignment

```
x <- 3
y <- 1
```

Vectorized language

Data structures:
Vectors, lists, matrices, data frames, tibbles, factors, tables.

<u>Vectors are atomic or non-atomic</u>

Atomic vectors have the same class or type:
logical (TRUE or FALSE)
integer (1, 2, 3); For integer place an L after the value (e.g., 2L)
double (1.0, 2.0, 3.0)
numeric (real or decimial)
complex (1 + 1i)
character ("apple")

note that character values must be placed within double quotes

Basic R functions

c() concatenates or joins objects

Object Assignment

```
grocerylist <- c("Lysol", "water", "Purell", "food")
numbers <- c(5, 4, 3, 2, 1)
```

Basic R and packages

Base R and packages

Using Packages

`install.packages('dplyr')`
Download and install a package from CRAN.

`library(dplyr)`
Load the package into the session, making all its functions available to use.

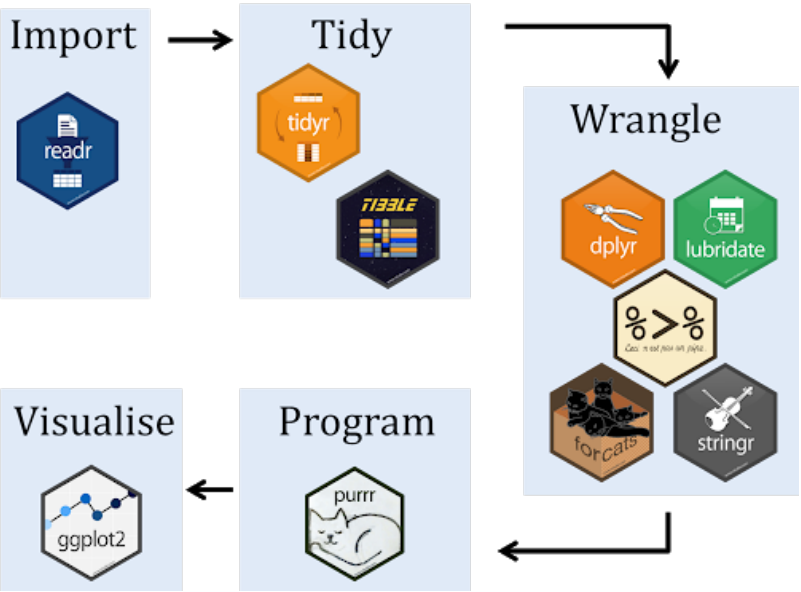
`dplyr::select`
Use a particular function from a package.

`data(iris)`
Load a built-in dataset into the environment.

https://rstudio.com/resources/cheatsheets/

Tidyverse

Tidyverse



http://www.seec.uct.ac.za/r-tidyverse

Import data

Import a .csv file:
read.csv or read_csv
Base R uses read.csv and the better functioning, updated import format uses read_csv in the "readr" package (included in Tidyverse)

Point-and-click also an option.
In the "Environment" window, click "Import Dataset."

Inspect data

Display the first few rows:

```
head(dataset)
head(dataset, n =3)
```

n = 3 shows the first 3 rows

Display the last few rows:

```
tail(dataset)
```

Find number of rows and columns:

```
dim(dataset)
```

Summary information through structure function:

```
str(dataset)
```

Working with variables

The $ references a column in the data set

dataset$age

Another way is by using [row, colum]
If age is the second column of the dataset then it can be also referenced by dataset[,2]

Function and package help

The ? can be typed in front of the function or package for more information.

Example "mean" is a base R function to calculate the arithmetic mean for an R object. Also has other options such as removing missing values which can be found by typing "?mean"

Using R functions

mean(dataset$age)
or the long way
sum(age) / length(age)

"psych" package's "describe" function produces descriptive statisitics: sample size, mean, median, mad: median absolute deviation, range, skew, kurtosis, standard error.

R Code

```
install.package("psych")
library("psych") or require("psych")
psych::describe(dataset)
```

Note– psych::describe() can be used to avoid masking issues that can be encountered by describe() if another package has a function describe().

Notes to your future self (and collaborators)

is used to comment out codes. These can be used as notes or to prevent running code not needed or used.

R Code

```
install.package("psych") #install psych package

library("psych") #load psych package

psych::describe(dataset) #code to produce decriptive statistics
```

Also helpful is to use # as quick way to locate chunk of code by typing 4 pound signs (hashtags) after the chunk label: #Correlation####

Functions

round(3.14159265359, digits = 2) #round pi to 2 decimal places
or
round(3.14159265359, 2)

sqrt(9) #square root of 9

R Code

```
data <- c(10, 20, 40, 80, 160) #create a vector of numbers

GT50_data <- data[data > 50] #subset values into new
dataframe containing values greater than 50
```

Plotting data

1. ggplot2
2. lattice
3. plotrix
4. plotly
5. highcharter
6. Rcolorbrewer
7. sunburstR

R Datasets:

PlantGrowth

mtcars

iris

ChickWeight

to find a list of built in datasets:

$data()$

to find all available datasets in R packages:

$data(package = .packages(all.available = TRUE))$

ggplot2

R Code

```
?PlantGrowth
```

PlantGrowth {datasets} R Documentation

Results from an Experiment on Plant Growth

Description

Results from an experiment to compare yields (as measured by dried weight of plants) obtained under a control and two different treatment conditions.

Usage

```
PlantGrowth
```

Format

A data frame of 30 cases on 2 variables.

[, 1] weight numeric
[, 2] group factor

The levels of `group` are 'ctrl', 'trt1', and 'trt2'.

ggplot2

R Code

Plant <- PlantGrowth
#assign dataset object

str(Plant)
#check structure

View(Plant)
#views the dataframe

```
> str(Plant)
'data.frame':   30 obs. of  2 variables:
 $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17
 $ group : Factor w/ 3 levels "ctrl","trt1",..: 1
```

	weight	group
1	4.17	ctrl
2	5.58	ctrl
3	5.18	ctrl
4	6.11	ctrl
5	4.50	ctrl

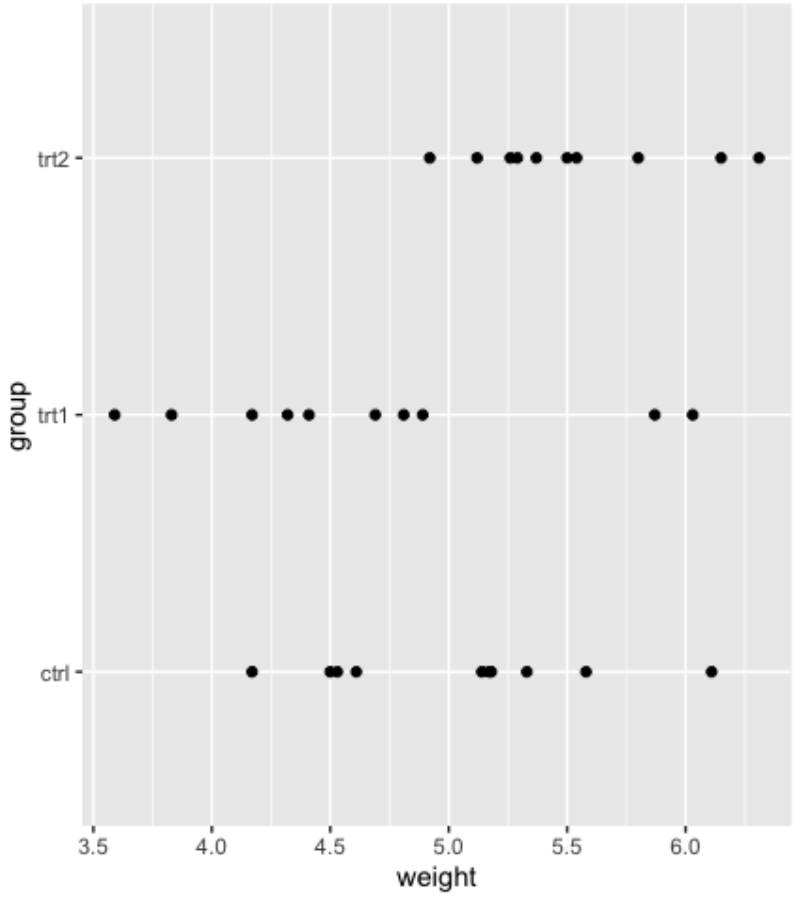
ggplot2

R Code

```
ggplot(data = PlantGrowth) +
#initialize layer for plot

aes(x = weight, y = group) +
#construct aesthetic mapping

geom_point()
#plot data as points
```



ggplot2

R Code

```
ggplot(data = PlantGrowth) +

aes(x = weight, y = group) +

labs(x= "Dried Weight", y =
"Groups", title = "Plant
Growth")+

geom_point(aes(colour =
factor(group)))
```