# Lesson 1: Getting to know R

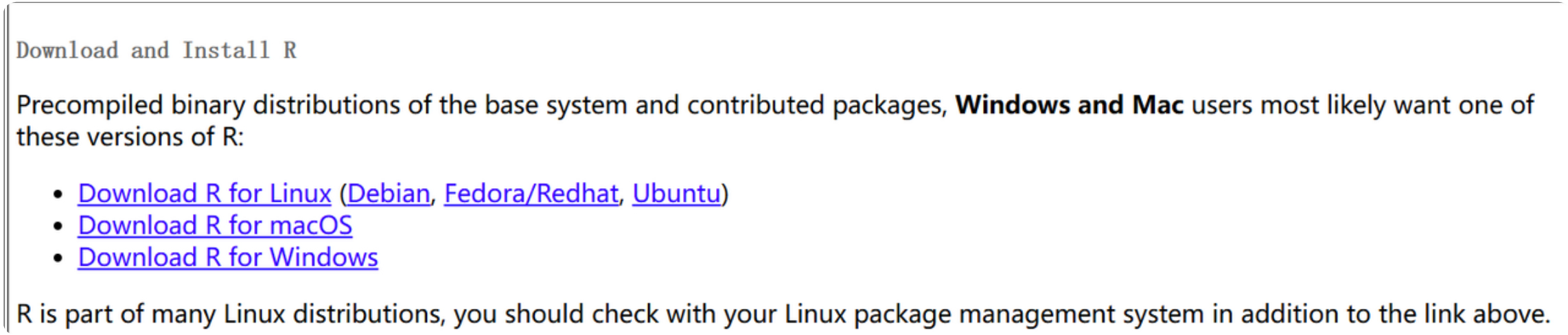
**Ziqi Chen**

PhD Candidate

Faculty of Applied Sciences, Macao Polytechnic University

Macao SAR

Twitter: https://x.com/chenziqi0506
ResearchGate: https://www.researchgate.net/profile/Ziqi-Chen-35
Google Scholar: https://scholar.google.com/citations?hl=zh-CN&user=O-Y9Dy8AAAAJ




# R installation

- The R software is a free, open-source software platform with powerful statistical analysis and visualization capabilities.
- **Download R:** https://cloud.r-project.org/

DOWNLOAD

# RStudio Desktop

Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python.

Don’t want to download or install anything? Get started with RStudio on Posit Cloud for free. If you're a professional data scientist looking to download RStudio and also need common enterprise features, don't hesitate to book a call with us.

Want to learn about core or advanced workflows in RStudio? Explore the RStudio User Guide or the Getting Started section.

| OS | Download | Size | SHA-256 |
|---|---|---|---|
| Windows 10/11 | RSTUDIO-2024.04.2-764.EXE | 262.79 MB | 09E1E38A |
| macOS 12+ | RSTUDIO-2024.04.2-764.DMG | 664.40 MB | D0DDD395 |
| Ubuntu 20/Debian 11 | RSTUDIO-2024.04.2-764-AMD64.DEB | 194.73 MB | 87B20155 |
| Ubuntu 22/Debian 12 | RSTUDIO-2024.04.2-764-AMD64.DEB | 196.64 MB | 1D0BD2F5 |
| OpenSUSE 15 | RSTUDIO-2024.04.2-764-X86_64.RPM | 196.89 MB | CC0E1D88 |
| Fedora 34/Red Hat 8 | RSTUDIO-2024.04.2-764-X86_64.RPM | 219.85 MB | DC097731 |
| Fedora 36/Red Hat 9 | RSTUDIO-2024.04.2-764-X86_64.RPM | 210.75 MB | 38140ED7 |

Windows 系统

苹果系统

# R installation

- **Download RStudio:**
  https://posit.co/download/rstudio-desktop/

# ! Here is an important tip:

- Don't use a Chinese username when the computer starts up; otherwise, RStudio will bar the Chinese username.
- Try to install on a non-system disc, for example, you can choose to install in the D disc.
- Installation path without Chinese and spaces. For example, this is better
  - D:/R
  - D:/Rstudio

# Install packages

- The power of the R language also lies in the various macro packages, which are generally downloaded and installed from The Comprehensive R Archive Network (CRAN).

**Examples:**

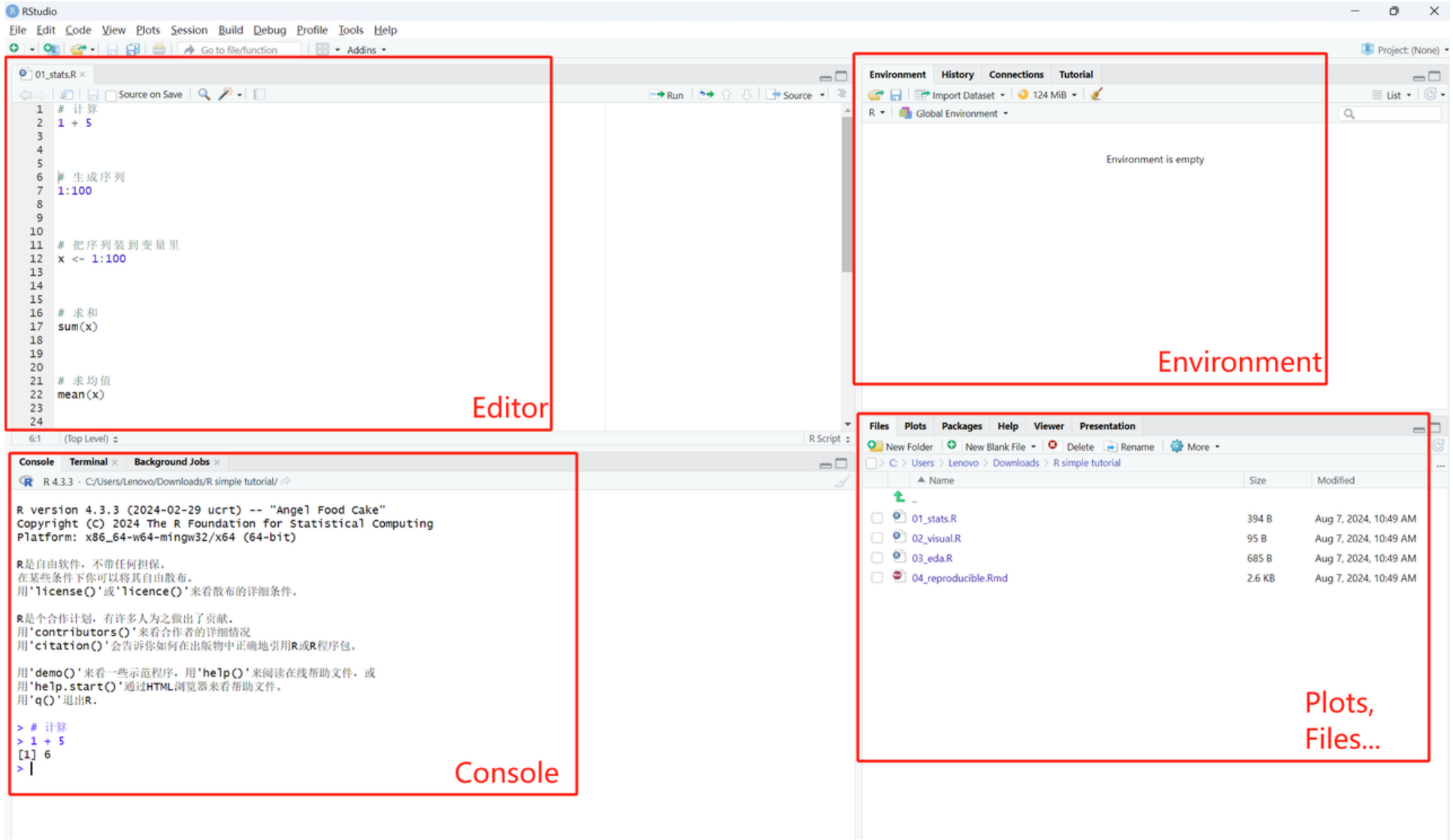
- **Installing a single package:**

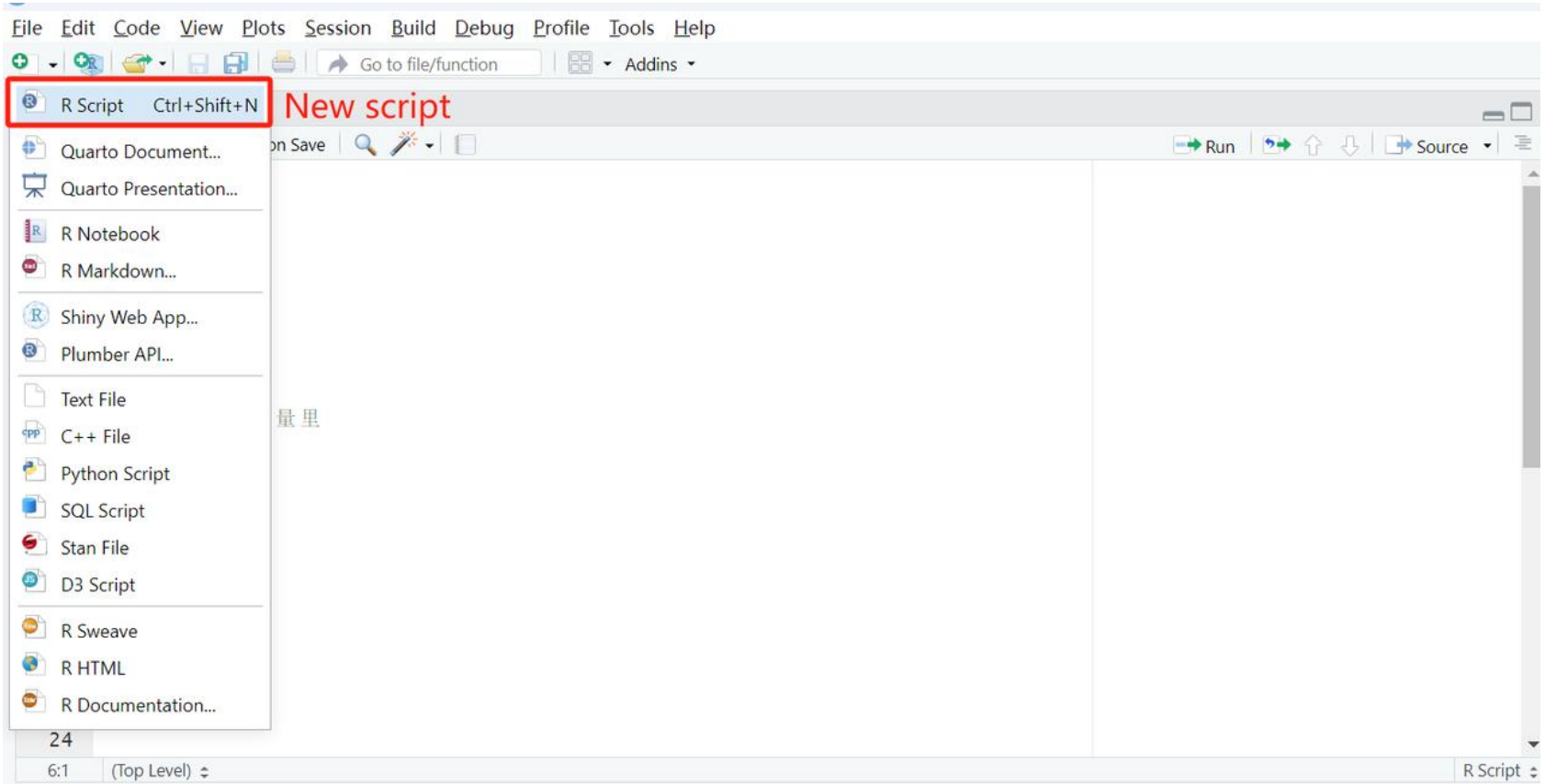
```
install.packages("tidyverse")
```

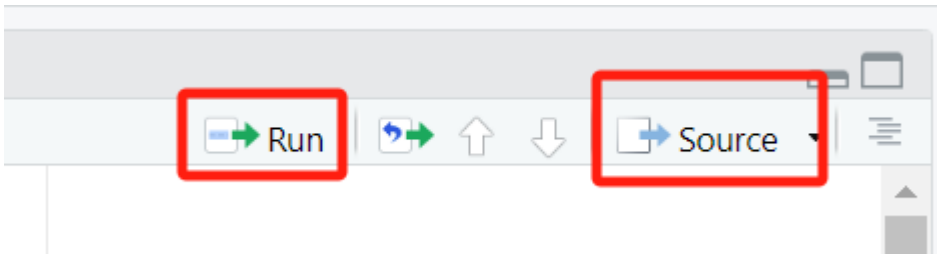
- **Installing multiple packages:**

```
install.packages(c("palmerpenguins", "patchwork", "gapminder", "ggridges", "readxl"))
```

# Let's get started!




File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
R Script Ctrl+Shift+N
New script
Quarto Document...
Quarto Presentation...
R Notebook
R Markdown...
Shiny Web App...
Plumber API...
Text File
C++ File
Python Script
SQL Script
Stan File
D3 Script
R Sweave
R HTML
R Documentation...
Run
Source
24
6:1
(Top Level)
R Script

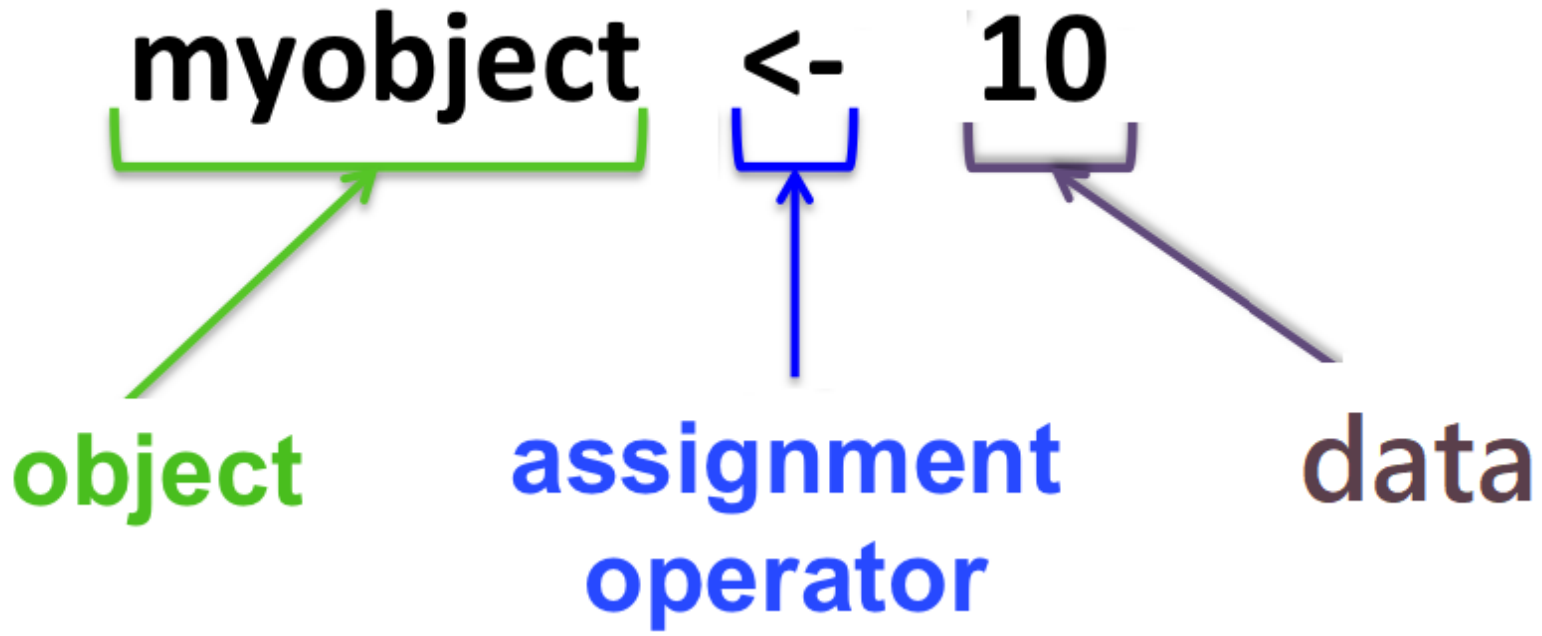

Run
Source

# Object

Try to type these into your console…

```
1 + 2
5 - 3
3 * 4
12/3
2^3
(2 + 4)/3
```

# Object



# Tibble data

```
> library(tidyverse)
── Attaching core tidyverse packages ──────────────────────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.4     ✓ readr     2.1.5
✓ forcats   1.0.0     ✓ stringr   1.5.1
✓ ggplot2   3.5.1     ✓ tibble    3.2.1
✓ lubridate 1.9.3     ✓ tidyr     1.3.1
✓ purrr     1.0.2
── Conflicts ────────────────────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package to force all conflicts to become errors
```

The tidyverse has designed a data format for tidy data called ***tibble***, which is a combination of the words tidy and table and, as the name implies, refers specifically to tidy data. It is the equivalent of a table in Excel (only if it is tidy), data in Stata, Pandas dataframes in Python, etc.

# Print *Tibble*

```
# A tibble: 349,940 × 8
   Date       Song                              Artist          Rank Last.Week Peak.Position Weeks.in.Charts Image.URL
   <chr>      <chr>                             <chr>          <int> <chr>              <int> <chr>           <chr>
 1 1958-08-06 Poor Little Fool                  Ricky Nelson       1 1                      1 2               #
 2 1958-08-06 Nel Blu Dipinto Di Blu (Volare)   Domenico Modu…     2 54                     2 2               #
 3 1958-08-06 Patricia                          Perez Prado A…     3 2                      2 2               #
 4 1958-08-06 Splish Splash                     Bobby Darin        4 3                      3 2               #
 5 1958-08-06 When                              Kalin Twins        5 5                      5 2               #
 6 1958-08-06 My True Love                      Jack Scott         6 8                      6 2               #
 7 1958-08-06 Hard Headed Woman                 Elvis Presley…     7 4                      4 2               #
 8 1958-08-06 Rebel-'rouser                     Duane Eddy Hi…     8 6                      6 2               #
 9 1958-08-06 Just A Dream                      Jimmy Clanton…     9 12                     9 2               #
10 1958-08-06 Willie And The Hand Jive          The Johnny Ot…     9 9                      9 2               #
# ℹ 349,930 more rows
# ℹ Use `print(n = ...)` to see more rows
```

**Sample data 1:**
**Billboard Hot 100 song data (349940 rows, 8 columns)**

# Print *Tibble*

In the R interface, the output format of ***tibble*** data will look like the above, from top to bottom.

```
# A tibble: 53,940 × 11
      X carat cut       color clarity depth table price     x     y     z
  <int> <dbl> <chr>     <chr> <chr>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1     1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
2     2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
3     3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
4     4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
5     5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
6     6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
7     7  0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
8     8  0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
9     9  0.22 Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
10   10  0.23 Very Good H     VS1      59.4    61   338  4     4.05  2.39
# ℹ 53,930 more rows
# ℹ Use `print(n = ...)` to see more rows
```

**Sample data 2:**
**Diamonds data (53940 rows, 11 columns)**

- # A tibble: 10 × 10 means the current tibble has 10 rows and 10 columns.
- carat cut ... This line is the variable name
- <dbl> <ord> ... This line is the variable type, e.g.
  - numeric (<dbl>)
  - integer (<int>)
  - Ordered category (<ord>)
- 1 From this line on, it's all diamond-specific data.
  - 1 is the row number, the rest are the parameters of the diamond.

If you have a large *tibble* with many rows and columns, the *tibble* will automatically **hide the extra rows and columns** for you when printing.

# Types of columns

Columns, or ‘variables’, are stored inside R in *vectors*.
Some common types of variables are described below:

- **numeric**, including real (double, real) and integer.
- **logical**, including TRUE and FALSE.
- **character**, including various lengths of text
- **factor**, **haven label**, usually numeric variables with labels
- **date**, **datetime**, which is essentially a numeric value
- **geometry**, from the *sf package*, including points, lines, surfaces, etc., used for spatial analysis and drawings
- Other **one-dimensional objects**
- **2D and even multidimensional objects**, such as a column of *tibbles*, each cell stores one *tibble*.

# Thank you!

**For more information:**

- https://chenziqi0506-ai.github.io/ziqichenRsimpletutorial/index.html
- https://github.com/chenziqi0506-AI/ziqichenRsimpletutorial/tree/main