



Phân tích dữ liệu với Lập trình R



Nhóm biên soạn:

1. Lê Ngọc Thành
2. Nguyễn Ngọc Thảo
3. Phạm Trọng Nghĩa
4. Nguyễn Thái Vũ
5. Trương Tấn Khoa

Năm 2022

Mô tả khóa học

- Chủ đề của khóa học là **tìm hiểu ngôn ngữ lập trình R**.
- Người học sẽ tìm hiểu
 - Cách sử dụng RStudio – môi trường để nhà phân tích dữ liệu làm việc với R
 - Các ứng dụng phần mềm và công cụ dành riêng cho R (chẳng hạn các gói R)
 - Cách thức R làm sạch, tổ chức, phân tích, trực quan hóa và báo cáo dữ liệu.



Nội dung chính của khóa học



- Tìm hiểu lợi ích của việc sử dụng ngôn ngữ lập trình R
- Khám phá cách sử dụng RStudio để ứng dụng R vào tác vụ phân tích
- Khảo sát các khái niệm cơ bản liên quan đến lập trình trên R
- Khảo sát nội dung và thành phần của các gói R, cụ thể là gói Tidyverse
- Hiểu rõ về khung dữ liệu (data frames) và cách sử dụng chúng trong R
- Tìm hiểu các lựa chọn để tạo trực quan hóa trong R
- Học về R Markdown để tạo tài liệu cho việc lập trình R



1 Lập trình và Phân tích dữ liệu



Thế giới lập trình

Lập trình máy tính

- **Lập trình máy tính** (Coding) là việc đưa ra các chỉ thị cho máy tính để thực hiện một hành động hoặc tập hợp các hành động theo cú pháp của một ngôn ngữ lập trình cụ thể.
- Việc **chọn lựa ngôn ngữ lập trình** tùy thuộc vào dự án bạn đang thực hiện hoặc vấn đề bạn muốn giải quyết.



Ngôn ngữ lập trình

- **Ngôn ngữ lập trình** (programming language) là hệ thống các từ và ký hiệu dùng để viết các chỉ thị mà máy tính theo đó thực hiện.
- **Cú pháp** (syntax) là bộ quy tắc riêng về cách sử dụng các từ và ký hiệu để chúng có ý nghĩa với máy tính.
- Ngôn ngữ lập trình là **cầu nối cho phép con người và máy tính giao tiếp**.



Ngôn ngữ lập trình R

- R là một ngôn ngữ lập trình được sử dụng cho phân tích thống kê, trực quan hóa và các tác vụ phân tích dữ liệu khác.
- Nhà phân tích dữ liệu sử dụng R cho nhiều nhiệm vụ liên quan đến quá trình phân tích dữ liệu.
- Hiểu cách thức hoạt động của R và lý do sử dụng R là điều quan trọng để phát triển thành thạo kỹ năng phân tích dữ liệu.



Cuộc cạnh tranh giữa R và Python

Ngôn ngữ	R	Python
Đặc điểm chung	<ul style="list-style-type: none">- Mã nguồn mở.- Dữ liệu được lưu trữ trong khung dữ liệu (data frames).- Công thức và chức năng có sẵn.- Có cộng đồng phát triển và hỗ trợ mã.	
Ưu điểm riêng	<ul style="list-style-type: none">- Thao tác dữ liệu, trực quan hóa dữ liệu và gói thống kê.- Phương pháp tiếp cận "Scalpel" cho dữ liệu: tìm các gói để thực hiện những gì bạn muốn với dữ liệu.	<ul style="list-style-type: none">- Cú pháp dễ dàng cho nhu cầu học máy.- Tích hợp với các nền tảng đám mây như Google Cloud, Amazon Web Services và Azure.
Thử thách riêng	<ul style="list-style-type: none">- Quy ước đặt tên không nhất quán → làm người mới bắt đầu khó chọn hàm.- Cách thao tác biến hơi phức tạp đối với người mới bắt đầu.	<ul style="list-style-type: none">- Người mới bắt đầu phải quyết định nhiều thứ như đầu vào / đầu ra dữ liệu, cấu trúc, biến, gói, và đối tượng.- Phương pháp tiếp cận "Swiss army knife" cho với dữ liệu: tìm ra cách để thực hiện những gì bạn muốn với dữ liệu.

Câu hỏi thảo luận

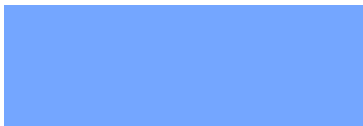
Viết 2-3 câu (tổng cộng 20-60 từ) để trả lời các câu hỏi sau:

- Nền tảng chuyên môn của bạn là gì?
- Điều gì khiến bạn đăng ký tham gia khóa học này?



Câu hỏi thảo luận

- Tiếp theo, viết 3-5 câu (tổng cộng 60-100 từ) để chia sẻ suy nghĩ của bạn về nền tảng trong lập trình.
 - Bạn đã từng làm việc với ngôn ngữ lập trình nào?
 - Bạn đã sử dụng lập trình cho các dự án cá nhân hay chuyên nghiệp nào?
 - Bạn thích điều gì nhất về lập trình?
 - Nếu bạn là người mới học lập trình, bạn có cảm nghĩ gì về việc học lập trình bằng R: bạn thấy hào hứng, có chút lo lắng, hay cả hai?



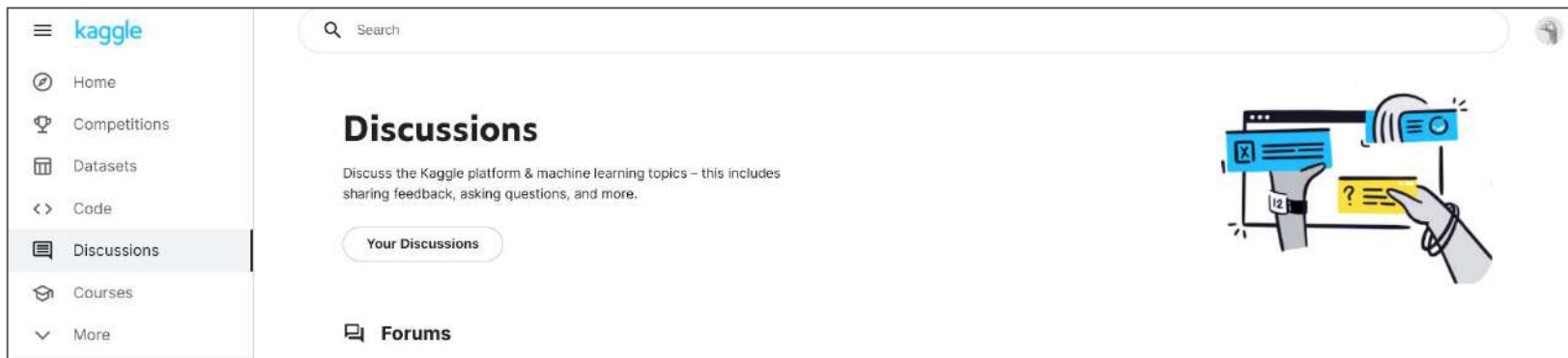
Câu hỏi thảo luận

- **Viết 2-3 câu (40-60 từ) trả lời cho mỗi câu hỏi dưới đây:**
 - Điều gì khiến bạn quyết định tìm hiểu về R?
 - Bạn hào hứng tìm hiểu những phần nào của R? Những phần nào có vẻ khó?



Sử dụng Kaggle để đặt câu hỏi về R

1. Đăng nhập tài khoản Kaggle và điều hướng đến tab Thảo luận (Discussions) trên menu.



Sử dụng Kaggle để đặt câu hỏi về R

2. Chọn danh mục con **Bắt đầu (Getting Started)**.

 **Forums**

**General**
Announcements, resources, and interesting discussions
last post 6 minutes ago by ArnuldOnData

**Getting Started**
The first stop for new Kagglers
last post an hour ago by KritiDoneria

**Product Feedback**
Tell us what you love, hate, and wish for
last post 15 hours ago by HuzaifaMS

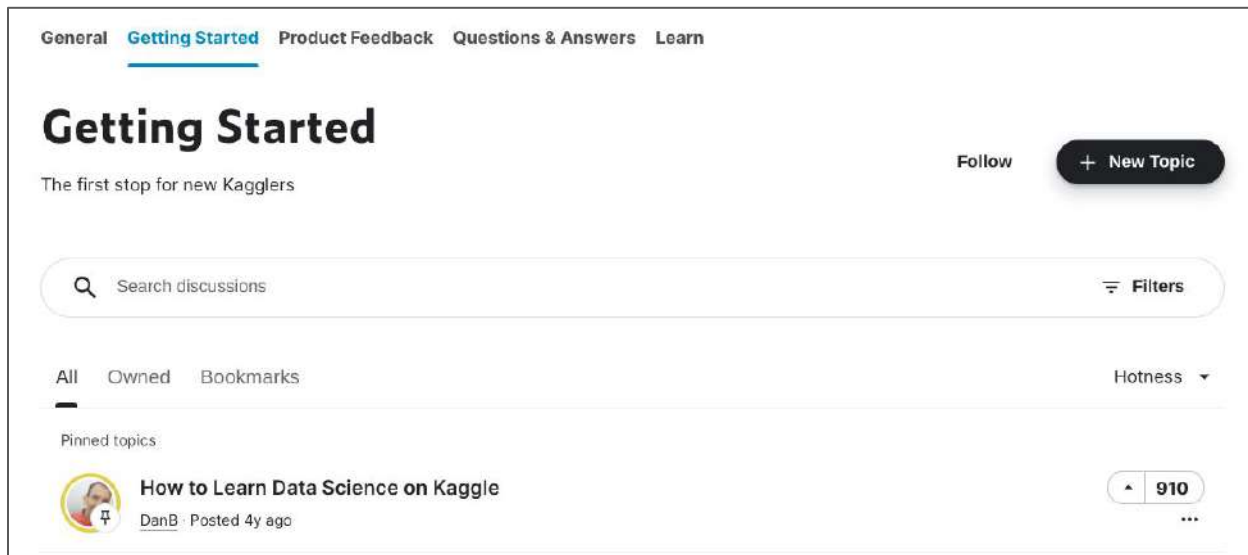

Recent topics by ArnuldOnData, Tushar Mishra, vardhan SIRAMDASU


Recent topics by KritiDoneria, Shivani Rana 63, Raghu Nilgal


Recent topics by HuzaifaMS, Björn, ibexorigin

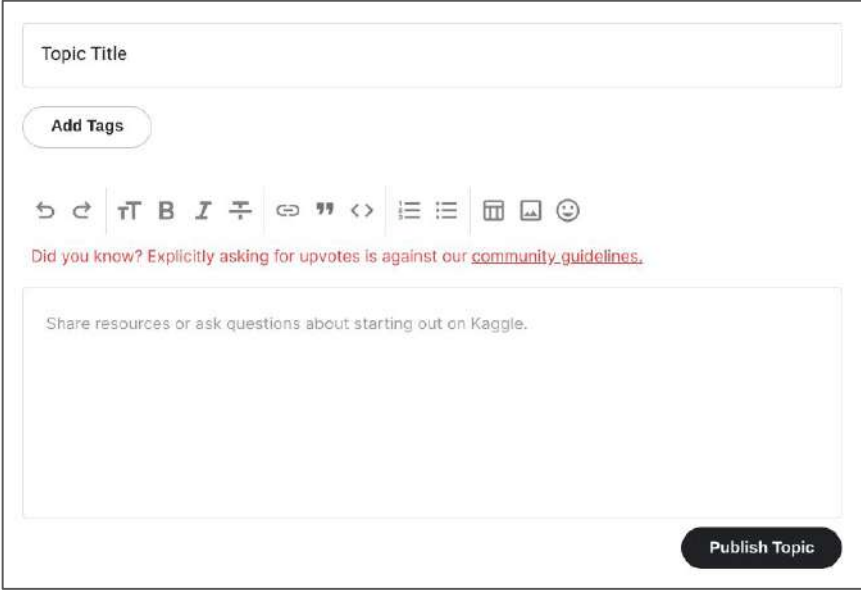
Sử dụng Kaggle để đặt câu hỏi về R

3. Sau đó, nhấp nút **Chủ đề mới (New Topic)** để tạo chủ đề mới trên diễn đàn.



Sử dụng Kaggle để đặt câu hỏi về R

4. Viết **Tiêu đề chủ đề (Topic Title)** và nội dung câu hỏi, sau đó nhấp nút **Xuất bản chủ đề (Publish Topic)**.



The image shows a screenshot of the Kaggle 'Ask a Question' form. It includes a text input field for the 'Topic Title', an 'Add Tags' button, a rich text editor with various formatting icons (bold, italic, link, etc.), a warning message in red text: 'Did you know? Explicitly asking for upvotes is against our [community guidelines](#).', a large text area for the question content with the placeholder text 'Share resources or ask questions about starting out on Kaggle.', and a 'Publish Topic' button at the bottom right.

Sử dụng Kaggle để đặt câu hỏi về R

- **Hãy xem xét câu hỏi bạn đã viết, bất kỳ câu trả lời nào bạn nhận được, và vai trò của Kaggle trong việc học R của bạn.**
 - Phản hồi của ai đó có giúp trả lời câu hỏi của bạn không? Tại sao có hoặc tại sao không? Nếu không, bạn hy vọng sẽ học được gì từ phản hồi của những người trong cộng đồng?
 - Bạn sử dụng Kaggle như thế nào để hỗ trợ bạn trong khi học R?

Lập trình như nhà phân tích dữ liệu

Ngôn ngữ lập trình theo ngành nghề

- **Nhà phân tích dữ liệu** thu thập, biến đổi và sắp xếp dữ liệu để đưa ra kết luận hay dự đoán, và thúc đẩy việc ra quyết định sáng suốt.
- **Nhà thiết kế Web** chịu trách nhiệm về kiểu dáng và bố cục của trang web chứa văn bản, đồ họa và video.
- **Nhà phát triển ứng dụng di động:** sử dụng lập trình để tạo các ứng dụng được sử dụng trên máy tính xách tay, điện thoại di động và máy tính bảng.



Học ngôn ngữ lập trình

- Xác định một dự án thực tế và sử dụng ngôn ngữ để giúp hoàn thành nó
- Ghi nhớ các khái niệm và quy tắc lập trình trước đây
- Tạo và lưu giữ các ghi chú hay cheat sheets ở định dạng phù hợp với bạn nhất
- Tạo hệ thống tài liệu thông tin trực tuyến để truy cập khi làm việc trong những môi trường lập trình khác nhau



Bảng tính, SQL, và R: Điểm chung

- **Điểm chung:** Cả ba công cụ đều sử dụng bộ lọc và hàm
- **Điểm khác**

Câu hỏi chính	Bảng tính	SQL	R
Đây là gì?	Chương trình sử dụng cột và dòng để tổ chức dữ liệu, phân tích và thao tác dữ liệu với công thức, hàm và các chức năng hỗ trợ sẵn	Ngôn ngữ lập trình cơ sở dữ liệu để giao tiếp với cơ sở dữ liệu cho việc phân tích dữ liệu	Ngôn ngữ lập trình chung cho việc phân tích thống kê, trực quan hóa dữ liệu và những phân tích dữ liệu khác
Ưu thế chính là gì?	Gồm các công cụ và tính năng trực quan hóa	Cho phép người dùng thao tác và nhận diện dữ liệu cần cho phân tích	Cung cấp ngôn ngữ truy cập được để tổ chức, hiệu chỉnh, và làm sạch khung dữ liệu, và tạo trực quan hóa sâu sắc về dữ liệu
Loại tập dữ liệu nào là tốt nhất?	Tập dữ liệu nhỏ	Tập dữ liệu lớn	Tập dữ liệu lớn

Bảng tính, SQL, và R: Điểm khác

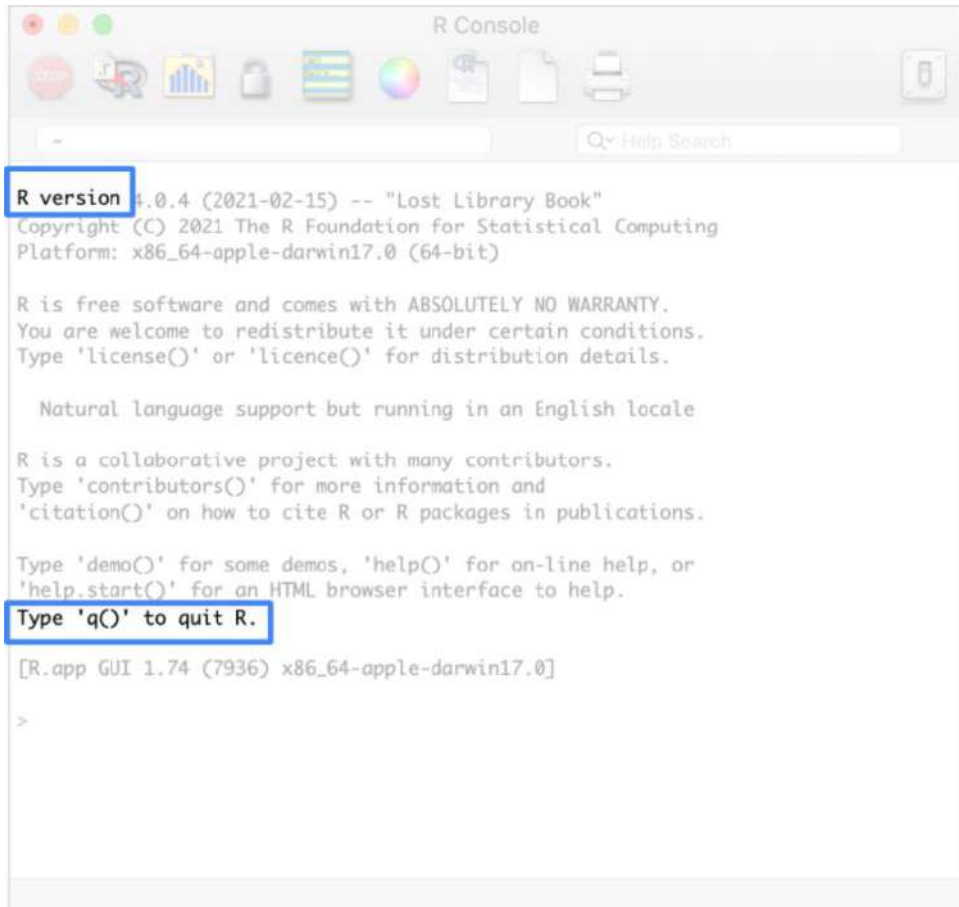
Câu hỏi chính	Bảng tính	SQL	R
Nguồn dữ liệu nào?	Nhập thủ công hoặc nhập từ nguồn ngoại vi	Truy cập từ một cơ sở dữ liệu ngoại vi	Được tải bằng R từ máy tính, hoặc từ nguồn ngoại vi
Dữ liệu phân tích thường được lưu ở đâu?	Trong tập tin bảng tính trên máy tính của bạn	Trong các bảng trên cơ sở dữ liệu đã truy cập	Trong tập tin R trên máy tính của bạn
Sử dụng công thức và hàm?	Có	Có	Có
Tạo trực quan hóa?	Có	Có, bằng công cụ hỗ trợ như hệ quản trị cơ sở dữ liệu (RDBMS) và công cụ trí tuệ kinh doanh (BI)	Có

Sử dụng R Console

- R Console là cửa sổ chương trình trong R để sử dụng ngôn ngữ lập trình R.
 - Đây là giao diện cho phép bạn xem, viết, chỉnh sửa và thực thi mã R.
- RStudio là môi trường phát triển tương tác (IDE) để lập trình R, sử dụng R Console và các công cụ khác để giúp viết và thực thi mã R dễ dàng hơn.
- Bạn có thể cài đặt RStudio [trực tiếp trên máy tính \(RStudio Desktop\)](#) hoặc sử dụng trên [nền tảng đám mây \(RStudio Cloud\)](#)



Sử dụng R Console



The screenshot shows the R Console window with a title bar 'R Console' and standard macOS window controls. The main text area displays the R startup message. A blue box highlights the text 'R version' at the beginning of the first line. Another blue box highlights the instruction 'Type 'q()' to quit R.' at the bottom of the main text area. The text in the console is as follows:

```
R version 4.0.4 (2021-02-15) -- "Lost Library Book"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (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.

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.

[R.app GUI 1.74 (7936) x86_64-apple-darwin17.0]

>
```


Sử dụng R Console

- Đây là dòng lệnh nhắc và bất kỳ thứ gì bạn nhập sau nó sẽ được đọc dưới dạng mã R thực thi khi bạn nhấn Enter (Windows) hoặc Return (Mac).

```
[R.app GUI 1.74 (7936) x86_64-apple-darwin17.0]
```

```
> |
```

```
> 4+5
```

```
[1] 9
```

```
> 5-4
```

```
[1] 1
```

```
> 10*2
```

```
[1] 20
```

```
> 10/2
```

```
[1] 5
```

```
> |
```

Câu hỏi thảo luận

- Bạn đã tải xuống và cài đặt các tập tin cho ngôn ngữ lập trình R. Hãy viết 2-3 câu (40-60 từ) để trả lời cho mỗi câu hỏi sau.
 - Ưu điểm của việc cài đặt R thay vì sử dụng nó trên nền tảng trực tuyến là gì?
 - Học R sẽ giúp bạn xây dựng kỹ năng phân tích dữ liệu như thế nào?



Câu hỏi thảo luận

- Bạn đã sử dụng R Console để viết một số chức năng cơ bản. Hãy viết 2-3 câu (40-60 từ) để trả lời cho mỗi câu hỏi sau.
 - R Console cho bạn biết điều gì về lập trình trong giao diện R?
 - Sự khác biệt giữa việc sử dụng R Console so với việc viết mã R trong tập tin văn bản là gì?



Học lập trình với RStudio

Khởi động với RStudio

- Bạn có thể tiếp cận RStudio thông qua [RStudio Desktop](#) hoặc [RStudio Cloud](#).
- **RStudio Desktop** là phiên bản mã nguồn mở theo giấy phép công cộng, được cài đặt trên máy tính
 - Bản dùng thử miễn phí RStudio Pro có tất cả các tính năng của phiên bản nguồn mở và giấy phép thương mại.
- **RStudio Cloud** cho phép người dùng làm việc trực tuyến từ trình duyệt, và do đó không phụ thuộc vào hệ điều hành.
 - Bạn cần đường truyền ổn định để có thể sử dụng Rstudio Cloud thông suốt.

Khởi động với RStudio

- **Gói** (package) là các đơn vị mã R có thể tái tạo, tài liệu mô tả, các bài kiểm tra mã nguồn và những tập dữ liệu mẫu.
 - Cộng đồng R tạo ra các gói để quản lý các chức năng R mà họ viết và sử dụng lại.
- **Tidyverse** là gói R với một triết lý thiết kế chung để thao tác, khám phá và trực quan hóa dữ liệu, và do đó là công cụ cần thiết đối với phân tích dữ liệu.



Cài đặt và nạp gói tidyverse

- Gõ `library(tidyverse)` để nạp tidyverse
- Cài đặt gói tidyverse từ console

```
> library(tidyverse)
— Attaching packages — tidyverse 1.3.1 —
✓ ggplot2 3.3.5      ✓ purrr  0.3.4
✓ tibble  3.1.2      ✓ dplyr  1.0.7
✓ tidyr   1.1.3      ✓ stringr 1.4.0
✓ readr   1.4.0      ✓ forcats 0.5.1
— Conflicts — tidyverse_conflicts() —
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
Console Terminal x Jobs x
R 4.0.3 · /cloud/project/ ↗

> install.packages("tidyverse")

* installing *binary* package 'tidyr' ...
* DONE (tidyr)
* installing *binary* package 'broom' ...
* DONE (broom)
* installing *binary* package 'modelr' ...
* DONE (modelr)
* installing *binary* package 'tidyverse' ...
* DONE (tidyverse)

The downloaded source packages are in
  '/tmp/RtmpEASzb7/downloaded_packages'
> |
```

Câu hỏi thảo luận

Bạn đã truy cập Rstudio như một IDE để lập trình R. Hãy viết 2-3 câu (40-60 từ) để trả lời cho mỗi câu hỏi sau.

- Trải nghiệm sử dụng RStudio khác với các môi trường khác như chương trình R chuẩn như thế nào? (Nếu bạn không cài đặt R vào thiết bị của mình, làm thế nào để so sánh các tính năng?)
- Lợi thế của việc sử dụng RStudio Cloud là gì? Ngược lại, lợi thế của việc sử dụng RStudio Desktop là gì?
- Bạn có gặp những trở ngại nào hay không?



Khi nào sử dụng RStudio

- Bạn cần chuyển đổi dữ liệu thô thành thông tin hữu ích → có thể khó thực hiện khi dữ liệu thô phức tạp.
- R và RStudio được thiết kế để xử lý các tập dữ liệu lớn, mà các bảng tính cũng có thể không xử lý được.
- RStudio cũng giúp bạn dễ dàng tái tạo công việc trên bộ dữ liệu khác nhau và tạo hình ảnh trực quan chi tiết.



Khi nào sử dụng RStudio

- Bạn cần quản lý phân tích, trực quan hóa xu hướng, và xây dựng đồ họa cho dữ liệu được trải rộng trên nhiều danh mục hoặc nhóm.
 - Ví dụ: phân tích dữ liệu bán hàng cho mọi thành phố trên toàn bộ quốc gia, mỗi thành phố có một nhóm dữ liệu riêng.
- Nhiệm vụ càng khó khăn hơn khi càng có nhiều nhóm dữ liệu cần làm việc.



Câu hỏi thảo luận

- RStudio cung cấp cho bạn một không gian làm việc duy nhất, nơi bạn có thể sử dụng R cho tất cả các giai đoạn của quá trình phân tích dữ liệu. Hãy viết một văn bản gồm hai hoặc nhiều đoạn văn (150-200 từ) mô tả những suy nghĩ ban đầu của bạn về RStudio.
 - Bạn nghĩ RStudio có thể giúp bạn như thế nào trong vai trò nhà phân tích dữ liệu trong tương lai?
 - Bạn yêu thích những tính năng nào của RStudio?
 - Nếu bạn chưa quen với R và RStudio, bạn nghĩ tính năng nào sẽ hữu ích nhất cho bạn với tư cách là một người học?



Kết nối với cộng đồng R

- **Cộng đồng trực tuyến** cho phép bạn kết nối với những người dùng R khác bất kể bạn sống ở đâu.
 - Diễn đàn, kênh thảo luận, thẻ truyền thông xã hội (hashtag) trên nền tảng truyền thông xã hội.
- **Buổi gặp mặt** (meetups): Nhiều tổ chức tổ chức cả buổi gặp mặt trực tiếp và trực tuyến cho người dùng R.
 - Tuy nhiên, bạn phải luôn thận trọng và an toàn khi trực tiếp tham dự các buổi gặp mặt.





2 Lập trình với RStudio



Khái niệm lập trình cơ bản

Khái niệm cơ bản trong R: Hàm

- **Hàm** (function) là phần mã có thể sử dụng lại để thực hiện các tác vụ cụ thể.
- Hàm bắt đầu bằng tên hàm (ví dụ, print hay paste) và thường được theo sau bởi một hoặc nhiều đối số trong dấu ngoặc đơn.
- **Đối số** (argument) là thông tin mà một hàm R cần để chạy.
- Ví dụ

```
> print("Coding in R")  
[1] "Coding in R"  
>
```

```
> print?
```

```
print(x, ...)  
print prints its argument and returns it invisibly (via  
invisible(x)). It is a generic function which means that new  
printing methods can be easily added for new classes.  
Press F1 for additional help
```

print	{base}
print.AsIs	{base}
print.by	{base}
print.condition	{base}
print.connection	{base}
print.data.frame	{base}
print.Date	{base}
print.default	{base}

Khái niệm cơ bản trong R: Biến

- **Biến** (variable) là một đại diện của một giá trị trong R có thể được lưu trữ để sử dụng sau này trong quá trình lập trình.
 - Biến cũng có thể được gọi là đối tượng.
- Tên biến thường là cụm từ ngắn có **phân biệt hoa thường, bắt đầu bằng một chữ cái** và cũng có thể chứa số và dấu gạch dưới.



Khái niệm cơ bản trong R: Chú thích





- **Chú thích** (comment) là những gì bạn cần mô tả hoặc giải thích về đoạn mã nguồn tương ứng, luôn bắt đầu bằng dấu # (hashtag).
- Bạn nên sử dụng chú thích nhiều nhất có thể để tập lệnh R dễ đọc hơn và mọi người đều có thể hiểu được mã nguồn.
- Ví dụ,

```
> #Here is an example of a variable  
> first_variable <- "This is my variable"  
> first_variable  
[1] "This is my variable"
```

Khái niệm cơ bản trong R: Vectơ

- **Vectơ** (vector) là một nhóm các phần tử dữ liệu cùng kiểu được lưu trữ thành chuỗi trong R.

```
> vec_1 = c(13, 48.5, 71, 101.5, 2)
> vec_1
[1] 13.0 48.5 71.0 101.5 2.0
```

Environment	History	Connections	Tutorial
  Import Dataset ▾	 137 MiB ▾		List ▾
R ▾	Global Environment ▾		
Values			
first_variable	"This is my variable"		
vec_1	num [1:5] 13 48.5 71 101.5 2		

Khái niệm cơ bản trong R: Đường ống

- **Đường ống** (pipe) là công cụ thể hiện một chuỗi gồm nhiều phép toán, được sử dụng để áp dụng đầu ra của một hàm vào một hàm khác.
- Sử dụng đường ống giúp mã nguồn dễ đọc và dễ hiểu hơn.

Đường ống này lọc
và sắp xếp dữ liệu

```
ToothGrowth %>%  
filter(dose == 0.5) %>%  
arrange(len)
```

Cấu trúc dữ liệu trong R

- **Cấu trúc dữ liệu** (data structures) là định dạng để tổ chức và lưu trữ dữ liệu.
- Các cấu trúc dữ liệu phổ biến nhất trong bao gồm:
 - **Vectơ**
 - **Khung dữ liệu (data frames)**
 - **Ma trận**
 - **Mảng**

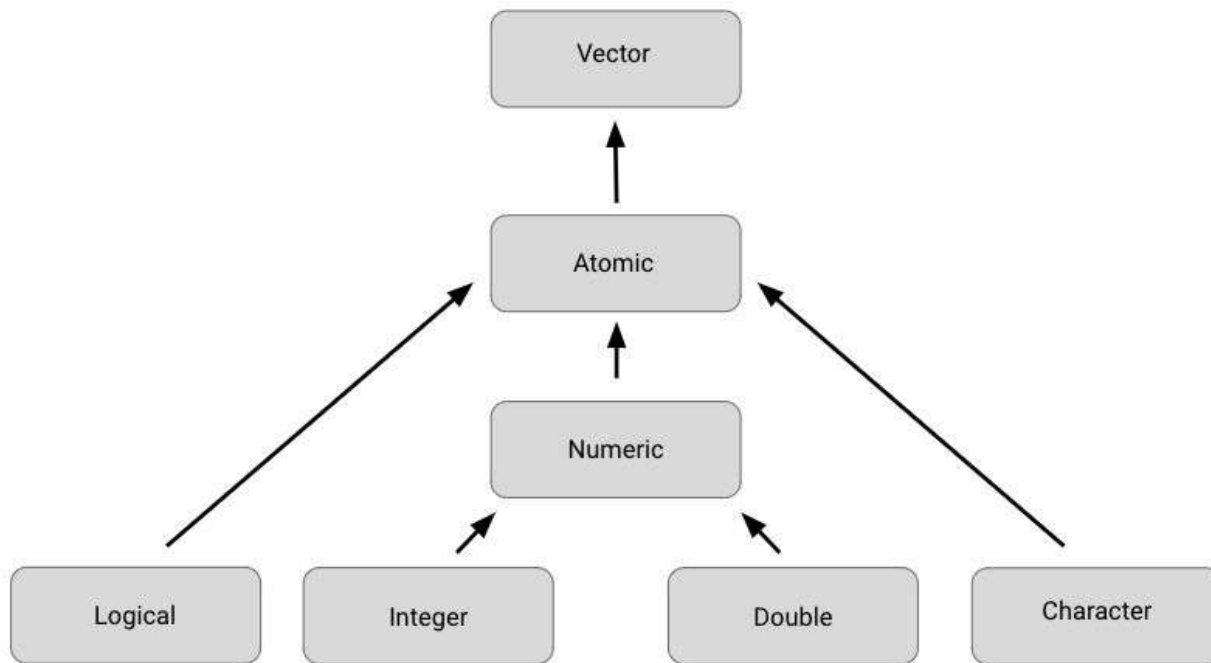


Vectơ trong R

- **Vectơ** (vector) là tập hợp phần tử dữ liệu cùng kiểu, được lưu trữ trong một chuỗi trong R.
- Có **sáu loại vectơ nguyên tử chính**: logical, integer, double, character (kể cả strings), phức (complex), và thô (raw).

Loại	Mô tả	Ví dụ
Logical	True/False	TRUE
Integer	Giá trị nguyên dương và âm	3
Double	Giá trị thực	101.175
Character	Giá trị chuỗi / ký tự	"Coding"

Vectơ trong R



Vectơ trong R: Thao tác thông dụng

- Tạo vectơ bằng hàm `combine c()`
- Kiểm tra loại vectơ bằng hàm `typeof()` hoặc kiểm tra loại cụ thể bằng hàm `is.`
- Kiểm tra số phần tử trong vectơ bằng hàm `length()`

```
c(2.5, 48.5, 101.5)      # Vectơ gồm số thực
c("Sara" , "Lisa" , "Anna")  # Vectơ
                             # gồm chuỗi
c(TRUE, FALSE, TRUE)      # Vectơ gồm giá
                             # trị Boolean
```

```
x <- c(33.5, 57.75, 120.05)
typeof(x)
#> [1] "double"
is.double(x)
#> [1] TRUE
length(x)
#> [1] 3
```

Vectơ trong R: Thao tác thông dụng

- Tất cả các loại vectơ đều có thể được đặt tên, điều này rất hữu ích để mã nguồn dễ đọc và mô tả các đối tượng được rõ ràng.
- Đặt tên cho các phần tử của vectơ bằng hàm `names()`

```
> x <- c(1, 3, 5)
> names(x) <- c("a", "b", "c")
> x
a b c
1 3 5
```


Danh sách trong R: Thao tác thông dụng

- **Danh sách** (list) khác với vectơ nguyên tử vì các phần tử của chúng có thể thuộc bất kỳ loại nào — như ngày tháng, khung dữ liệu, vectơ, ma trận, v.v.
 - Danh sách thậm chí có thể chứa các danh sách khác.
- Tạo danh sách bằng hàm **list()** và xem cấu trúc danh sách bằng hàm **str()**

```
str(list("a", 1L, 1.5, TRUE))  
#> List of 4  
#> $ : chr "a"  
#> $ : int 1  
#> $ : num 1.5  
#> $ : logi TRUE
```

```
list(list(list(1 , 3, 5)))  
#> List of 1  
#> $ :List of 1  
#> ..$ :List of 3  
#> .. ..$ : num 1  
#> .. ..$ : num 3  
#> .. ..$ : num 5
```

Kiểu dữ liệu thời gian trong R

- Bạn cần nạp cả hai gói, **tidyverse** và **lubridate** (thuộc gói tidyverse), để có thể làm việc với ngày và giờ trong R.
- Trong R, có ba loại dữ liệu để cập đến một thời điểm trong thời gian:
 - Ngày tháng ("2016-08-16")
 - Thời gian trong ngày ("20:11:59 UTC")
 - Ngày tháng-thời gian ("2018-03-31 18:15:48 UTC")



Thời gian trong R: Thao tác thông dụng

- Lấy ngày hiện tại bằng hàm `today()`
- Lấy ngày-giờ hiện tại bằng hàm `now()`
- Sử dụng hàm `as_date()` để chuyển đổi ngày-giờ thành ngày tháng.

```
today()  
#> [1] "2021-01-20"
```

```
now()  
#> [1] "2021-01-20 16:25:05 UTC"
```

```
as_date(now())  
> [1] "2021-01-20"
```

Thời gian trong R: Thao tác thông dụng

- Bạn có ba cách để tạo định dạng ngày-giờ trong R
 - Từ một chuỗi
 - Từ một ngày tháng đơn lẻ
 - Từ một đối tượng ngày / giờ hiện có
- R tạo ngày ở định dạng `yyyy-mm-dd` chuẩn theo mặc định.

```
ymd("2021-01-20")  
#> [1] "2021-01-20"  
mdy_hm("01/20/2021 08:01")  
#> [1] "2021-01-20 08:01:00  
UTC"  
ymd(20210120)  
#> [1] "2021-01-20"
```

Khung dữ liệu trong R

- **Khung dữ liệu** (data frames) là tập hợp các cột – tương tự như bảng tính hoặc bảng SQL – giúp tóm tắt và sắp xếp dữ liệu ở định dạng dễ đọc và sử dụng.
 - Mỗi cột có một tên ở trên cùng đại diện cho một biến và bao gồm một quan sát trên mỗi hàng.

```
> data.frame(x = c(1, 2, 3) , y = c(1.5, 5.5, 7.5))
```

```
  x  y
1 1 1.5
2 2 5.5
3 3 7.5
```



The screenshot shows a data frame with 21 rows and 10 columns. The columns are: carat, cut, color, clarity, depth, table, price, x, y, and z. The data represents various diamonds with their physical and financial attributes.

	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.23	Ideal	E	S12	61.5	55.0	326	3.95	3.98	2.43
2	0.21	Premium	E	S11	59.8	61.0	326	3.89	3.84	2.31
3	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
4	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
5	0.31	Good	J	S12	63.3	58.0	335	4.34	4.35	2.75
6	0.24	Very Good	J	VVS2	62.8	57.0	336	3.94	3.96	2.48
7	0.24	Very Good	I	VVS1	62.3	57.0	336	3.95	3.98	2.47
8	0.26	Very Good	H	S11	61.9	55.0	337	4.07	4.11	2.53
9	0.22	Fair	E	VS2	65.1	61.0	337	3.87	3.78	2.49
10	0.23	Very Good	H	VS1	59.4	61.0	338	4.00	4.05	2.39
11	0.30	Good	J	S11	64.0	55.0	339	4.25	4.28	2.73
12	0.23	Ideal	J	VS1	62.8	56.0	340	3.93	3.90	2.46
13	0.22	Premium	F	S11	60.4	61.0	342	3.88	3.84	2.33
14	0.31	Ideal	J	S12	62.2	54.0	344	4.35	4.37	2.71
15	0.20	Premium	E	S12	60.2	62.0	345	3.79	3.75	2.27
16	0.32	Premium	E	I1	60.9	58.0	345	4.38	4.42	2.68
17	0.30	Ideal	I	S12	62.0	54.0	348	4.31	4.34	2.68
18	0.30	Good	J	S11	63.4	54.0	351	4.23	4.29	2.70
19	0.30	Good	J	S11	63.8	56.0	351	4.23	4.26	2.71
20	0.30	Very Good	J	S11	62.7	59.0	351	4.21	4.27	2.66
21	0.30	Good	I	S12	63.3	56.0	351	4.26	4.30	2.71

Showing 1 to 22 of 53,940 entries, 10 total columns

Tìm hiểu về Tibbles

- Tibbles giống như **khung dữ liệu được sắp xếp hợp lý**, được tự động thiết lập để **chỉ kéo lên 10 hàng đầu tiên của tập dữ liệu** và **số cột vừa với màn hình**.
- Tibbles không bao giờ thay đổi tên của các biến hoặc kiểu dữ liệu đầu vào.
 - **Khung dữ liệu cho phép nhiều thay đổi hơn, nhưng Tibbles dễ sử dụng hơn.**
- Tibbles là một phần của gói tidyverse.

Tìm hiểu về Tibbles: Ví dụ minh họa

- Tập dữ liệu diamonds có 10 cột và hàng nghìn hàng. Hình ảnh này hiển thị một phần của khung dữ liệu.

	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
2	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
3	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
4	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
5	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
6	0.24	Very Good	J	VVS2	62.8	57.0	336	3.94	3.96	2.48
7	0.24	Very Good	I	VVS1	62.3	57.0	336	3.95	3.98	2.47
8	0.26	Very Good	H	SI1	61.9	55.0	337	4.07	4.11	2.53
9	0.22	Fair	E	VS2	65.1	61.0	337	3.87	3.78	2.49
10	0.23	Very Good	H	VS1	59.4	61.0	338	4.00	4.05	2.39
11	0.30	Good	J	SI1	64.0	55.0	339	4.25	4.28	2.73
12	0.23	Ideal	J	VS1	62.8	56.0	340	3.93	3.90	2.46
13	0.22	Premium	F	SI1	60.4	61.0	342	3.88	3.84	2.33
14	0.31	Ideal	J	SI2	62.2	54.0	344	4.35	4.37	2.71
15	0.20	Premium	E	SI2	60.2	62.0	345	3.79	3.75	2.27
16	0.32	Premium	E	I1	60.9	58.0	345	4.38	4.42	2.68
17	0.30	Ideal	I	SI2	62.0	54.0	348	4.31	4.34	2.68
18	0.30	Good	J	SI1	63.4	54.0	351	4.23	4.29	2.70
19	0.30	Good	J	SI1	63.8	56.0	351	4.23	4.26	2.71
20	0.30	Very Good	J	SI1	62.7	59.0	351	4.21	4.27	2.66
21	0.30	Good	I	SI2	63.3	56.0	351	4.26	4.30	2.71

Showing 1 to 22 of 53,940 entries, 10 total columns

Tìm hiểu về Tibbles: Ví dụ minh họa

- Tạo một tibble từ dữ liệu hiện có bằng hàm `as_tibble()`

```
# A tibble: 53,940 x 10                                as_tibble(diamonds)
  carat cut    color clarity depth table price      x      y      z
  <dbl> <ord> <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1  0.23 Ideal E      SI2      61.5    55   326   3.95   3.98   2.43
2  0.21 Prem... E      SI1      59.8    61   326   3.89   3.84   2.31
3  0.23 Good  E      VS1      56.9    65   327   4.05   4.07   2.31
4  0.290 Prem... I      VS2      62.4    58   334   4.2    4.23   2.63
5  0.31 Good  J      SI2      63.3    58   335   4.34   4.35   2.75
6  0.24 Very... J      VVS2     62.8    57   336   3.94   3.96   2.48
7  0.24 Very... I      VVS1     62.3    57   336   3.95   3.98   2.47
8  0.26 Very... H      SI1      61.9    55   337   4.07   4.11   2.53
9  0.22 Fair  E      VS2      65.1    61   337   3.87   3.78   2.49
10 0.23 Very... H      VS1      59.4    61   338   4      4.05   2.39
# ... with 53,930 more rows
```

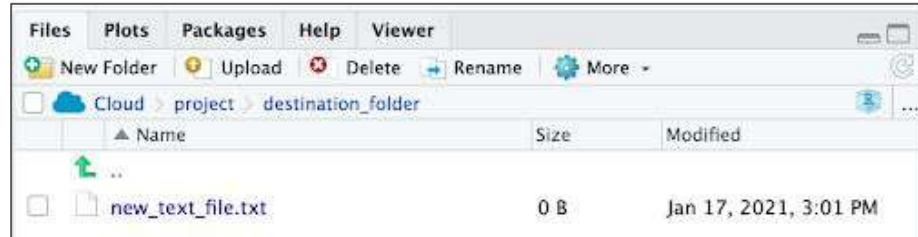

Tập tin trong R

- **dir.create()**: tạo thư mục mới
- **file.create()**: tạo tập tin trống
- **file.copy()**: sao chép tập tin
- **unlink()**: xóa tập tin

```
dir.create ("destination_folder")
```

```
file.create ("new_csv_file.csv")
```

```
file.copy ("new_text_file.csv" , "destination_folder")
```



```
unlink ("new_text_file.csv")
```

Ma trận trong R

- **Ma trận** (matrix) là một tập hợp hai chiều của các phần tử dữ liệu thuộc một kiểu dữ liệu duy nhất.
- Có thể tạo ma trận bằng hàm **matrix()**

```
> matrix(c(3:8), nrow = 2)
```

	[,1]	[,2]	[,3]
[1,]	3	5	7
[2,]	4	6	8

```
> matrix(c(3:8), ncol = 2)
```

	[,1]	[,2]
[1,]	3	6
[2,]	4	7
[3,]	5	8

Khám phá lập trình trong R

Toán tử và phép tính

- **Toán tử** (operator) là một ký hiệu đặt tên cho kiểu thao tác hoặc phép tính sẽ được thực hiện trong một công thức.
- **Toán tử gán** (assignment operator) gán giá trị cho các biến và vectơ.
 - Ví dụ, `x <- c(33.5, 57.75, 120.05)`
- **Toán tử số học** (arithmetic operator) được sử dụng để hoàn thành các phép tính toán học quen thuộc: + (cộng), - (trừ), * (nhân), và / (chia).



Toán tử logic

- Toán tử logic (logical operator) trả về kiểu dữ liệu logic như TRUE hay FALSE.
- Có ba loại toán tử logic chính: AND (ký hiệu & hoặc &&), OR (ký hiệu | hoặc ||) và NOT (ký hiệu !)



Toán tử logic: Ví dụ minh họa

	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	36	118	8.0	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4

- `Solar.R > 150 & Wind > 10`

	Ozone	Solar.R	Wind	Temp	Month	Day
1	18	313	11.5	62	5	4

Toán tử logic: Ví dụ minh họa

- `Solar.R > 150 | Wind > 10`

	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	12	149	12.6	74	5	3
3	18	313	11.5	62	5	4

- `!(Solar.R > 150 | Wind > 10)`

	Ozone	Solar.R	Wind	Temp	Month	Day
1	36	118	8.0	72	5	2

Phát biểu điều kiện

- **Câu lệnh điều kiện** (conditional statement) là câu khai báo rằng nếu một điều kiện nhất định đúng, thì một sự kiện nhất định phải diễn ra.
 - Ví dụ: "Nếu nhiệt độ trên mức đóng băng, thì tôi sẽ ra ngoài đi dạo." Nếu điều kiện đầu tiên là đúng (nhiệt độ trên mức đóng băng), thì điều kiện thứ hai sẽ xảy ra (tôi sẽ đi dạo).
- Các câu lệnh điều kiện trong R có logic tương tự.



Câu lệnh điều kiện: if

- Câu lệnh **if** đặt một điều kiện và nếu điều kiện cho giá trị là TRUE, thì mã R liên kết với câu lệnh if sẽ được thực thi.

```
if (condition) {  
    expr  
}
```

```
if (x > 0) {  
    print("x is a positive number")  
}
```

Câu lệnh điều kiện: else

- Câu lệnh **else** được sử dụng kết hợp với câu lệnh **if**.

```
if (condition) {  
    expr 1  
} else {  
    expr 2  
}
```

```
if (x > 0) {  
    print("x is a positive number")  
} else {  
    print(" x is a non-positive number")  
}
```

Câu lệnh điều kiện: else if

- Câu lệnh **else if** nằm giữa câu lệnh **if** và câu lệnh **else**.

```
if (condition1) {  
    expr1  
} else if (condition2) {  
    expr2  
} else {  
    expr3  
}
```

Câu hỏi thảo luận: Truy vấn và Lập trình

- Bây giờ bạn đã viết các truy vấn bằng SQL và sử dụng mã để lập trình trong R, bạn có thể nhận thấy một số điểm tương đồng giữa hai điều này.
- Hãy viết một thảo luận gồm hai hoặc nhiều đoạn văn (tổng cộng 100-150 từ) về bất kỳ điểm tương đồng nào mà bạn có thể gặp phải.



Tìm hiểu gói R

Gói trong R

- **Gói trong R** (R packages) bao gồm các hàm R có thể tái sử dụng, tài liệu về các hàm chức năng và cách sử dụng chúng.
- Base R là một tập hợp các gói có sẵn trong Rstudio
- Một số khác được khuyến nghị nhưng chưa có sẵn → cần phải gọi lệnh thư viện để tải về trước khi sử dụng.



Gói trong R

```
> installed.packages()
```

	Package	LibPath	Version	Priority
base	"base"	"/opt/R/4.2.1/lib/R/library"	"4.2.1"	"base"
boot	"boot"	"/opt/R/4.2.1/lib/R/library"	"1.3-28"	"recommended"
class	"class"	"/opt/R/4.2.1/lib/R/library"	"7.3-20"	"recommended"
cluster	"cluster"	"/opt/R/4.2.1/lib/R/library"	"2.1.3"	"recommended"
codetools	"codetools"	"/opt/R/4.2.1/lib/R/library"	"0.2-18"	"recommended"
compiler	"compiler"	"/opt/R/4.2.1/lib/R/library"	"4.2.1"	"base"
datasets	"datasets"	"/opt/R/4.2.1/lib/R/library"	"4.2.1"	"base"
foreign	"foreign"	"/opt/R/4.2.1/lib/R/library"	"0.8-82"	"recommended"
graphics	"graphics"	"/opt/R/4.2.1/lib/R/library"	"4.2.1"	"base"
grDevices	"grDevices"	"/opt/R/4.2.1/lib/R/library"	"4.2.1"	"base"
grid	"grid"	"/opt/R/4.2.1/lib/R/library"	"4.2.1"	"base"

Chọn đúng gói R

- **Tidyverse**: là tập hợp các gói R được thiết kế đặc biệt để làm việc với dữ liệu. Đây là thư viện tiêu chuẩn cho hầu hết các nhà phân tích dữ liệu,.
- **Danh sách nhanh các gói R hữu ích (Quick list of useful R packages)**: danh sách gói hữu ích của Bộ phận Hỗ trợ RStudio với hướng dẫn cài đặt và mô tả chức năng.
- **CRAN Task Views**: chỉ mục của các gói CRAN được sắp xếp theo tác vụ



Tìm hiểu về tidyverse

- **Tidyverse** là tập hợp các gói trong R với một triết lý thiết kế chung cho thao tác, khám phá và trực quan hóa dữ liệu.
- Bạn cần phải tải tidyverse và nạp gói này mỗi khi sử dụng.

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

```
library("tidyverse")
```

```
— Attaching packages —
✓ ggplot2 3.3.6      ✓ purrr  0.3.4
✓ tibble  3.1.7      ✓ dplyr   1.0.9
✓ tidyr   1.2.0      ✓ stringr 1.4.0
✓ readr   2.1.2      ✓ forcats 0.5.1

— Conflicts —
✗ dplyr::filter() masks stats::filter()
✗ dplyr::lag()    masks stats::lag()
```

- **Xung đột** (conflicts) xảy ra khi gói có các hàm trùng tên với các hàm khác.

Tìm hiểu tidyverse

Phân tích dữ liệu với tidyverse

- tidyverse bao gồm tám gói, bốn trong số đó có vai trò thiết yếu với quy trình làm việc của nhà phân tích dữ liệu: ggplot2, dplyr, tidyr và readr.
- **ggplot2**: trực quan hóa dữ liệu với các biểu đồ.
- **tidyr**: dọn dẹp dữ liệu để làm cho dữ liệu gọn gàng hơn.
- **readr**: được sử dụng để nhập dữ liệu, hàm phổ biến nhất là read_csv để thao tác này sẽ nạp tập tin CSV vào R.
- **dplyr**: cung cấp một bộ chức năng nhất quán để hoàn thành một số tác vụ thao tác dữ liệu thông thường.

Đường ống và Mã nguồn lồng nhau

- **Lồng nhau** (nested) mô tả mã nguồn thực hiện một chức năng cụ thể và được chứa trong mã nguồn thực hiện một chức năng rộng hơn.
- Đường ống là một cách để triển khai khái niệm trên

Mã nguồn không lồng nhau

```
filtered_tg <- filter(ToothGrowth, dose =  
0.5)  
arrange(filtered_tg, len)
```

Mã nguồn có lồng nhau

```
arrange(filter(ToothGrowth, dose = 0.5),  
len)
```

Mã nguồn dùng đường ống

```
filtered_toothgrowth <- ToothGrowth %>%  
  filter(ToothGrowth, dose = 0.5) %>%  
  arrange(len)
```

Tài nguyên hỗ trợ R

- Cộng đồng R có nhiều người dùng tận tâm giúp nhau tìm ra giải pháp cho các vấn đề và cách sử dụng R.
- Ngoài ra còn có rất nhiều blog tuyệt vời, nơi bạn có thể tìm thấy các hướng dẫn và các tài nguyên khác, chẳng hạn như
 - **RStudio**
 - **RStudio Blog**
 - **Stack Overflow**
 - **R-Bloggers**
 - **R-Bloggers' tutorials for learning R**





3 Làm việc với dữ liệu trong R



Khảo sát dữ liệu trong R

Lưu ý về khung dữ liệu trong R

- Phải đặt tên cột, dùng tên cột trống có thể gây ra vấn đề với kết quả sau này.
- Dữ liệu lưu trữ trong khung dữ liệu có thể ở nhiều kiểu khác nhau, chẳng hạn như số, hệ số hoặc ký tự.
- Mỗi cột phải chứa cùng một số mục dữ liệu, ngay cả khi một số mục dữ liệu bị thiếu.



Khung dữ liệu trong R: Ví dụ minh họa

- Ta sử dụng tập dữ liệu **diamonds** có sẵn trong gói ggplot2 của tidyverse.

- Nạp tập dữ liệu vào R:

```
data("diamonds")  
View(diamonds)
```

- Dùng hàm **head()** để xem nhanh một số hàng đầu tiên

```
> head(diamonds)  
# A tibble: 6 × 10  
  carat cut      color clarity depth table price      x      y      z  
  <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>  
1  0.23 Ideal    E      SI2     61.5    55   326   3.95   3.98   2.43  
2  0.21 Premium E      SI1     59.8    61   326   3.89   3.84   2.31  
3  0.23 Good    E      VS1     56.9    65   327   4.05   4.07   2.31  
4  0.29 Premium I      VS2     62.4    58   334   4.2    4.23   2.63  
5  0.31 Good    J      SI2     63.3    58   335   4.34   4.35   2.75  
6  0.24 Very Good J      VVS2     62.8    57   336   3.94   3.96   2.48
```

Khung dữ liệu trong R: Ví dụ minh họa

- Xem cấu trúc của bảng bằng hàm `str()` hay `colnames()`

```
> str(diamonds)
tibble [53,940 × 10] (S3: tbl_df/tbl/data.frame)
 $ carat   : num [1:53940] 0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
 $ cut     : Ord.factor w/ 5 levels "Fair"<"Good"<...: 5 4 2 4 2 3 3 3 1 3 ...
 $ color   : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<...: 2 2 2 6 7 7 6 5 2 5 ...
 $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<...: 2 3 5 4 2 6 7 3 4 5 ...
 $ depth   : num [1:53940] 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
 $ table   : num [1:53940] 55 61 65 58 58 57 57 55 61 61 ...
 $ price   : int [1:53940] 326 326 327 334 335 336 336 337 337 338 ...
 $ x       : num [1:53940] 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
 $ y       : num [1:53940] 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
 $ z       : num [1:53940] 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

```
> colnames(diamonds)
[1] "carat" "cut" "color" "clarity" "depth" "table" "price" "x" "y" "z"
```

Khung dữ liệu trong R: Ví dụ minh họa

- Dùng hàm `mutate()` để thêm một cột và tính toán dữ liệu mới của cột.

```
> mutate(diamonds, carat_2 = carat*100)
# A tibble: 53,940 × 11
  carat cut      color clarity depth table price     x     y     z carat_2
  <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl> <dbl>
1  0.23 Ideal     E    SI2     61.5    55   326   3.95   3.98   2.43    23
2  0.21 Premium  E    SI1     59.8    61   326   3.89   3.84   2.31    21
3  0.23 Good     E    VS1     56.9    65   327   4.05   4.07   2.31    23
4  0.29 Premium  I    VS2     62.4    58   334   4.2    4.23   2.63    29
5  0.31 Good     J    SI2     63.3    58   335   4.34   4.35   2.75    31
6  0.24 Very Good J    VVS2     62.8    57   336   3.94   3.96   2.48    24
7  0.24 Very Good I    VVS1     62.3    57   336   3.95   3.98   2.47    24
8  0.26 Very Good H    SI1     61.9    55   337   4.07   4.11   2.53    26
9  0.22 Fair     E    VS2     65.1    61   337   3.87   3.78   2.49    22
10 0.23 Very Good H    VS1     59.4    61   338   4      4.05   2.39    23
# ... with 53,930 more rows
```

Quan sát dữ liệu

Làm việc với dữ liệu

- Thao tác trên dữ liệu có thể được thực hiện thông qua các toán tử R cơ bản: Số học (arithmetic), Quan hệ (relational), Logic (logical), và Gán (assignment).
- Tổ chức dữ liệu bằng các hàm như: `arrange()`, `group_by()`, `filter()`.
- Biến đổi dữ liệu bằng cách hàm như: `separate()`, `unite()`, `mutate()`.



Làm việc với dữ liệu

- Thao tác trên dữ liệu có thể được thực hiện thông qua các toán tử R cơ bản: Số học (arithmetic), Quan hệ (relational), Logic (logical), và Gán (assignment)
- Tổ chức dữ liệu bằng các hàm như: `arrange()`, `group_by()`, `filter()`,...
- Biến đổi dữ liệu bằng cách hàm như: `separate()`, `unite()`, `mutate()`,...
- Hiệu chỉnh tên cột với `rename()`, `rename_with()`, `clean_names()`,...
- Phân tích thống kê dữ liệu: `mean()`, `sd()`, `cor()`

Giải quyết dữ liệu lệch với R

- Tình huống sau được chia sẻ bởi nhà phân tích định lượng.
- Nhóm phân tích dữ liệu đang thực hiện một cuộc khảo sát so sánh song song: hiển thị cho người dùng hai quảng cáo cạnh nhau và yêu cầu họ chọn quảng cáo nào thích hơn.
- Sau nhiều lượt khảo sát, họ nhận thấy sự thiên vị nhất quán có lợi cho mục đầu tiên. Cũng có một sự sụt giảm có thể đo lường được trong sở thích nếu chúng tôi đổi vị trí của nó sang vị trí thứ hai.
- Vì vậy, họ quyết định hiển thị vị trí của hai quảng cáo một cách ngẫu nhiên với tần suất tương đương.



Giải quyết dữ liệu lệch với R

- Hàm **sample()** đưa vào yếu tố ngẫu nhiên vào mã nguồn R, cho phép bạn lấy một mẫu phần tử ngẫu nhiên từ một tập dữ liệu.
- Việc thêm mã này sẽ xáo trộn hàng trong tập dữ liệu một cách ngẫu nhiên.
- Hàm `sample()` chỉ là một trong nhiều hàm và phương thức trong R mà bạn có thể sử dụng để giải quyết sự sai lệch trong dữ liệu.





4

Trực quan hóa Tính mỹ thuật Chú thích



Trực quan hóa trong R

Tạo trực quan hóa trong R

- **Trực quan hóa dữ liệu** (visualization) là một trong những phần quan trọng nhất của phân tích dữ liệu.
- Các gói tương tác với R bao gồm ggplot2, Plotly, Lattice, RGL, Dygraphs, Leaflet, Highcharter, Patchwork, gganimate và ggridges.
- **ggplot2** là công cụ được đánh giá là linh hoạt và phổ biến nhất trong số đó.

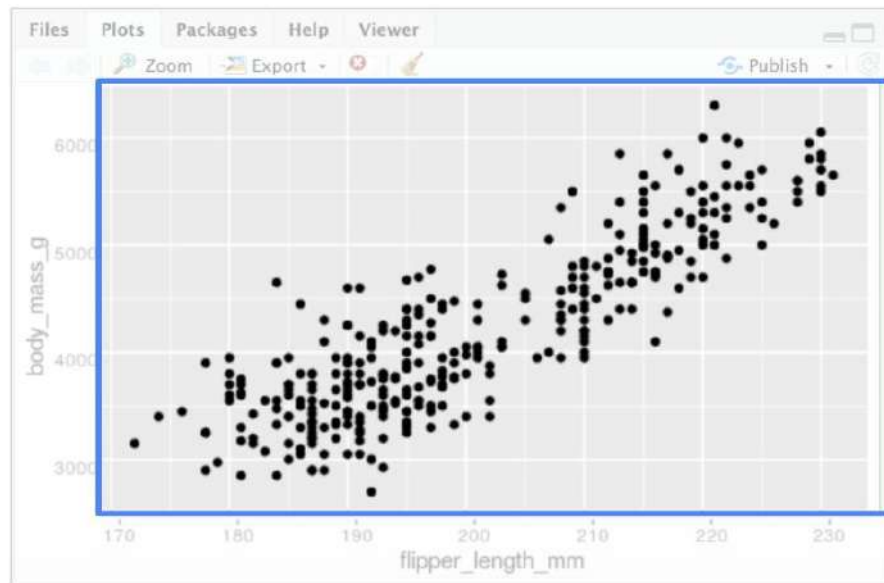


Trực quan hóa dữ liệu với ggplot2

- **ggplot2** là một hệ thống dựa trên ngữ pháp đồ họa để mô tả và xây dựng các hình ảnh trực quan hóa dữ liệu.
- Ta có thể xây dựng bất kỳ cốt truyện nào từ những thành phần cơ bản giống nhau, gọi là khối xây dựng (building block), bao gồm
 - Tập dữ liệu
 - Tập hợp hình học
 - Tập hợp các thuộc tính thẩm mỹ
- Tạo biểu đồ với ggplot2: Chọn tập dữ liệu → chọn tập hình học để đại diện cho các điểm dữ liệu và tính thẩm mỹ để lập biểu đồ các biến → vẽ biểu đồ.

Trực quan hóa dữ liệu với ggplot2

```
ggplot(data = penguins) + geom_point(mapping = aes(x = flipper_length_mm, y =  
body_mass_g))
```



Câu hỏi thảo luận

- Vui lòng viết một đến hai đoạn văn (tổng cộng 150-200 từ) mô tả suy nghĩ ban đầu của bạn về sự khác biệt giữa Tableau và ggplot2 khi nói đến trực quan hóa dữ liệu.
- Suy ngẫm về những câu hỏi sau:
 - Điểm mạnh và hạn chế của Tableau khi nói đến trực quan hóa dữ liệu là gì? Bạn yêu thích những tính năng nào của Tableau?
 - Nếu chưa quen với ggplot2, bạn nghĩ tính năng nào hữu ích nhất để hiển thị dữ liệu?
 - Công cụ trực quan hóa trong Tableau khác với công cụ trong ggplot2 như thế nào?

Khám phá tính mỹ thuật trong phân tích

Thuộc tính mỹ thuật

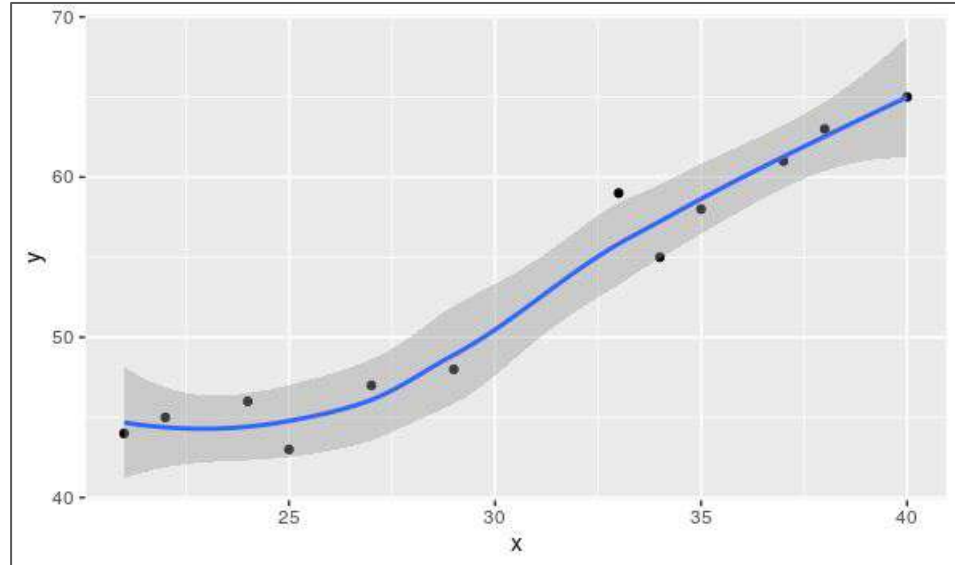
- **Màu sắc (color):** tùy chỉnh màu của tất cả các điểm trên biểu đồ hoặc màu của từng nhóm dữ liệu
- **Kích thước (size):** tùy chỉnh kích thước của các điểm theo nhóm dữ liệu
- **Hình dạng (shape):** tùy chỉnh hình dạng của các điểm theo nhóm dữ liệu

```
ggplot(data, aes(x=distance, y= dep_delay, color=carrier,  
size=air_time, shape = carrier)) + geom_point()
```


Làm mịn trên biểu đồ

- **Làm mịn (smoothing)** cho phép phát hiện xu hướng dữ liệu vốn không dễ dàng nhận thấy được từ các điểm dữ liệu trên biểu đồ.

```
ggplot(data,  
  aes(x=distance,  
    y= dep_delay)) +  
  geom_point() +  
  geom_smooth()
```



Làm mịn trên biểu đồ

- **Làm mịn Loess:** phù hợp nhất để làm mịn cho biểu đồ dưới 1000 điểm.
- **Làm mịn Gam (generalized additive model):** phù hợp để làm mịn cho biểu đồ với lượng điểm lớn.



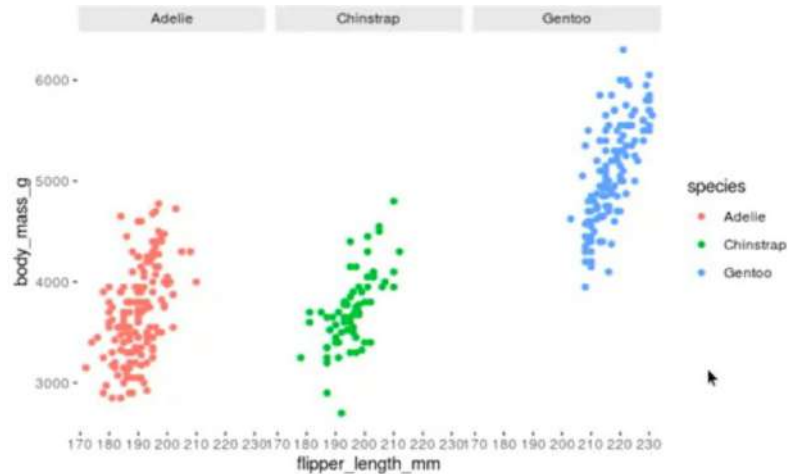
Facet để hiển thị tập con dữ liệu

- Hàm `facet` giúp hiển thị các nhóm nhỏ hơn hoặc tập hợp con dữ liệu.
- Facet hiển thị các mặt khác nhau của dữ liệu bằng cách đặt mỗi tập hợp con trên biểu đồ của riêng nó, từ đó giúp khám phá mẫu trong dữ liệu và tập trung vào mối quan hệ giữa các biến khác nhau.
- `facet_wlaps()` và `facet_grid()`



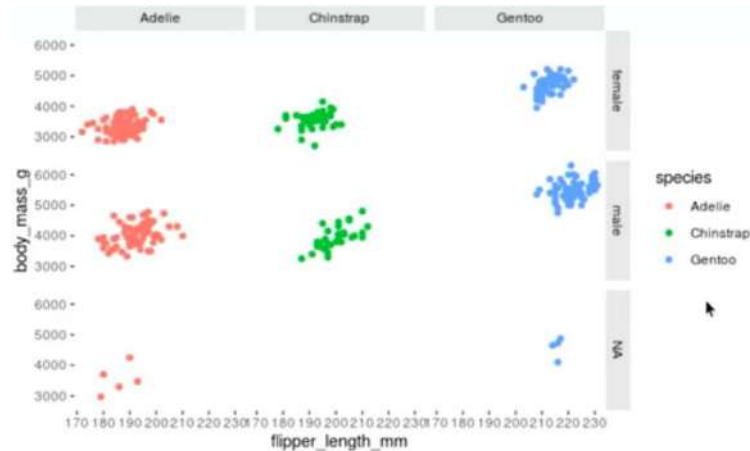
Facet để hiển thị tập con dữ liệu

```
ggplot(data=penguins)+  
  geom_point(mapping=aes(x=flipper_length_mm,y=body_mass_g,color=species))+  
  facet_wrap(~species)
```



Facet để hiển thị tập con dữ liệu

```
ggplot(data=penguins)+  
  geom_point(mapping=aes(x=flipper_length_mm,y=body_mass_g,color=species))+  
  facet_grid(sex~species)
```



Lọc và vẽ biểu đồ

- **Lọc dữ liệu** (filter) trước khi vẽ biểu đồ giúp tập trung vào tập con cụ thể của dữ liệu và có được nhiều chi tiết hướng đến mục tiêu hơn.
- Sử dụng hàm dplyr **filter()** trong cú pháp ggplot.

```
data %>%  
  filter(variable1 == "DS") %>%  
  ggplot(aes(x = weight, y = variable2, colour = variable1)) +  
  geom_point(alpha = 0.3, position = position_jitter()) +  
  stat_smooth(method = "lm")
```

Chú thích và lưu trực quan hóa

Chú thích biểu đồ

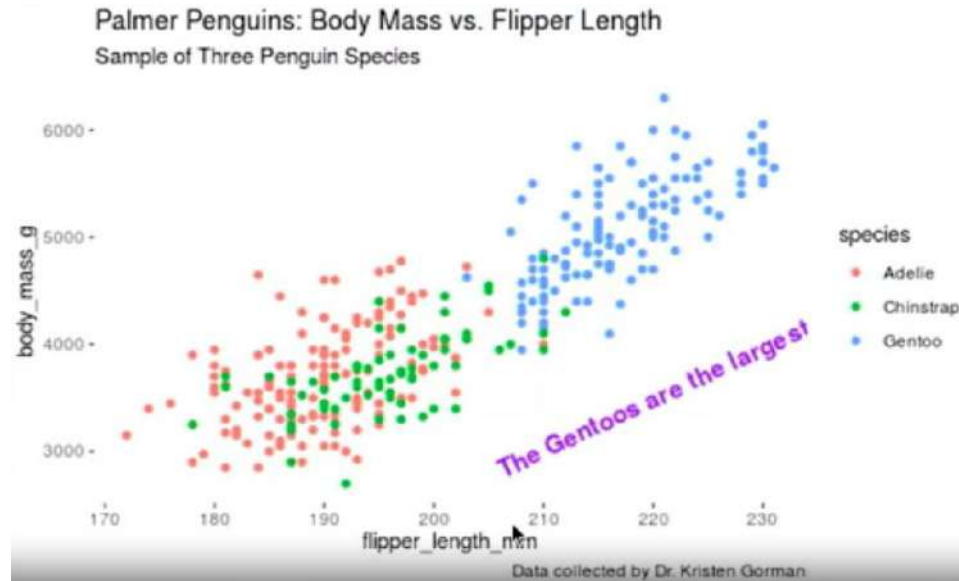
- **Nhãn** (label) và **chú thích** (annotation) có thể thực sự hữu ích khi làm nổi bật các phần quan trọng trong dữ liệu của bạn và truyền đạt các điểm chính.
- Ví dụ, thêm tiêu đề vào biểu đồ:

```
ggplot(data=penguins)+  
  geom_point(mapping=aes(x=flipper_length_mm,y=body_mass_g,color=species))+  
  labs(title="Palmer Penguins: Body Mass vs. Flipper Length")
```

- Ví dụ thêm chú thích tại vị trí (x,y) trên biểu đồ:

```
annotate("text", x=220,y=3500,label="The Gentoos are the largest")
```


Chú thích biểu đồ



Lưu trực quan hóa

- Sử dụng tùy chọn Export trong tab biểu đồ của RStudio hoặc chức năng ggsave được cung cấp bởi gói ggplot2.
- Bạn có thể mở một thiết bị đồ họa R như **png()** hoặc **pdf()** để lưu biểu đồ dưới dạng PNG hoặc PDF và kết xuất bằng lệnh **dev.off()**.

```
pdf(file = "/Users/username/Desktop/example.pdf", width = 4, height = 4)
plot(x = 1:10, y = 1:10)
abline(v = 0)
text(x = 0, y = 1, labels = "Random text")
dev.off()
```



5 Tài liệu và báo cáo



Xây dựng tài liệu và báo cáo trong RStudio

Tổng quan về R Markdown

- **R Markdown** là định dạng tập tin để tạo tài liệu động theo cú pháp Markdown.
 - **Markdown** là cú pháp để định dạng tập tin văn bản trơn.
- R Markdown cho phép bạn tạo bản phân tích và kết luận trong một tài liệu.
 - Nó liên kết mã và báo cáo với nhau để bạn có thể chia sẻ từng bước phân tích.
- **R Notebook** cho phép người dùng chạy mã nguồn và hiển thị các biểu đồ và biểu đồ trực quan hóa mã.
 - Bất kỳ tài liệu R Markdown nào cũng có thể được sử dụng như một notebook.

Tổng quan về R Markdown

- R Markdown cho phép chuyển đổi nhiều định dạng tập tin.
- Ngôn ngữ Markdown ban đầu được thiết kế cho đầu ra HTML, do đó R Markdown có nhiều tính năng khả dụng nhất cho định dạng này.
- [HTML](#) là tập hợp ký hiệu và mã đánh dấu được sử dụng để tạo trang web.
- Có thể tìm hiểu R Markdown thông qua:
 - Tài liệu về R Markdown trong RStudio
 - Tài liệu tham khảo R Markdown
 - Sách R for Data Science và R Markdown: The Definitive Guide

Tạo tài liệu R Markdown

Cấu trúc của tài liệu Markdown

- Tài liệu Markdown được lưu ở định dạng tập tin RMD.



YAML header dành cho siêu dữ liệu hoặc dữ liệu về dữ liệu trong phần còn lại của tập tin.

```
---
title: "Untitled"
output: html_document
date: '2022-07-15'
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the Knit button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```
```


Hiểu đoạn mã và kết xuất tài liệu

Sử dụng đoạn mã trong tài liệu Markdown

- **Dấu phân tách** (delimiter) là một ký tự chỉ ra phần đầu hoặc phần cuối của một mục dữ liệu.
- Phần chèn mã nguồn sẽ được đặt trong cụm ``` và ```.

```
```{r loading packages}  
library(tidyverse)
data(diamonds)
view(diamonds)
```
```

2022-07-15

```
library(tidyverse)
```

```
## — Attaching packages — tidyverse 1.3.1 —
```

```
## ✓ ggplot2 3.3.6   ✓ purrr  0.3.4  
## ✓ tibble  3.1.7   ✓ dplyr  1.0.9  
## ✓ tidyr   1.2.0   ✓ stringr 1.4.0  
## ✓ readr   2.1.2   ✓ forcats 0.5.1
```

```
## — Conflicts — tidyverse_conflicts() —
```

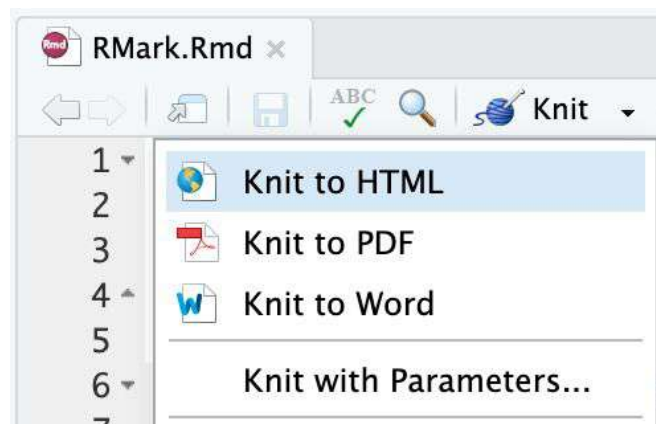
```
## X dplyr::filter() masks stats::filter()  
## X dplyr::lag()    masks stats::lag()
```

```
data(diamonds)  
view(diamonds)
```

Kết xuất tài liệu R Markdown

Tài liệu R Markdown được kết xuất theo **định dạng chỉ định trong YAML header** hoặc **định dạng khác thông qua chức năng Knit** trên giao diện.

```
---  
title: "Untitled"  
output: html_document  
date: '2022-07-15'  
---
```





Tổng kết



Những điểm cần nắm:

- Hiểu những lợi ích của việc sử dụng ngôn ngữ lập trình R.
- Khám phá cách sử dụng RStudio để ứng dụng R vào tác vụ phân tích.
- Khảo sát các khái niệm cơ bản liên quan đến lập trình trên R.
- Khảo sát nội dung và thành phần của các gói R, bao gồm gói Tidyverse.
- Hiểu rõ về khung dữ liệu (data frames) và cách sử dụng chúng trong R.
- Tìm hiểu các lựa chọn để tạo trực quan hóa trong R.
- Học về R Markdown để tạo tài liệu cho việc lập trình R.



THANK YOU

