

PBA: 1. Course Introduction

Fall 2026

Jaewon (“Jay-one”) Yoo

National Tsing Hua University

Roadmap

1. Welcome and Motivation

Q1: Why do we want to learn programming and business analytics?

Q2: What do we do as business analysts?

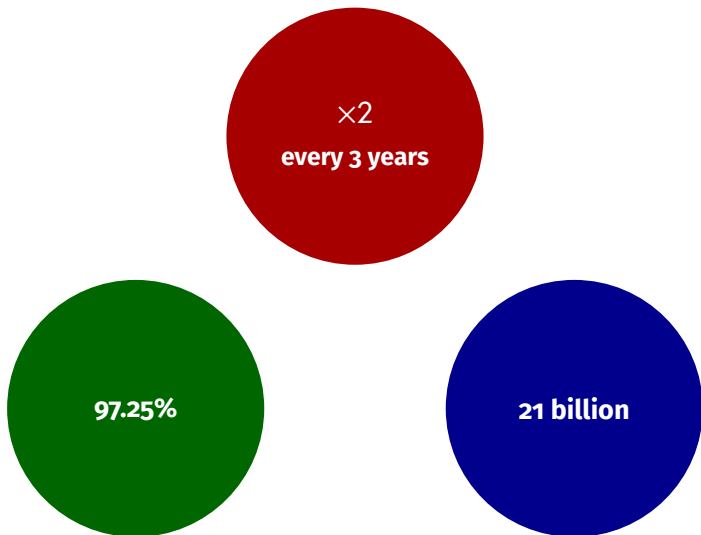
2. A First Look at R

3. Course Overview

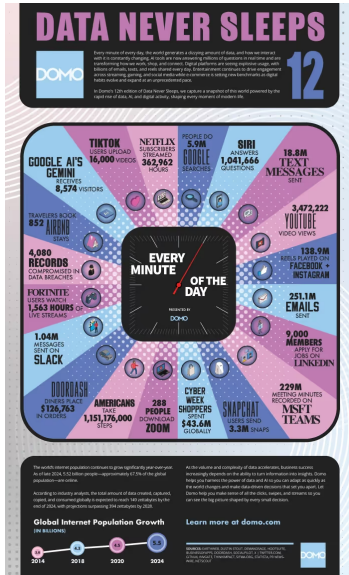
1/ Welcome and Motivation

Why Learn Programming & Business Analytics?

Three Circles w/ Numbers?



The Red Circle: Volume



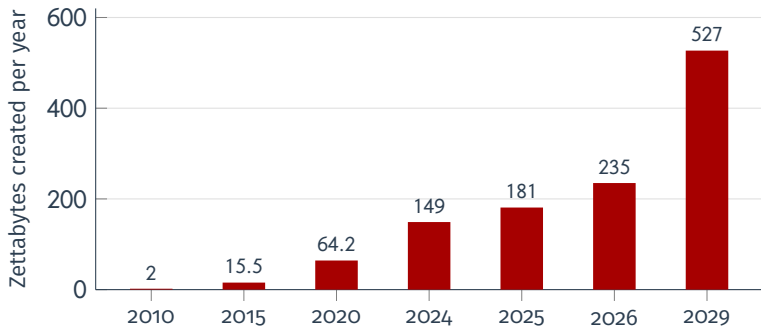
In one internet minute (2024):

- 5.9 million Google searches
- 3.5 million YouTube video views
- 139 million Reels played on Facebook and Instagram
- 251 million emails sent
- 363,000 hours streamed on Netflix

Every minute. Around the clock.

Source: Domo, Data Never Sleeps 12.0 (2024); Forbes, January 2025

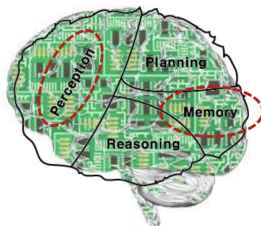
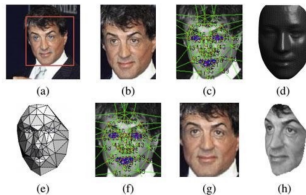
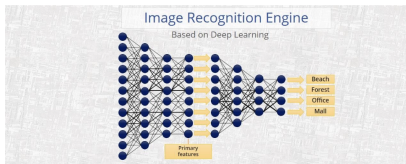
The Red Circle: How Fast It Adds Up



- 1 ZB is a trillion gigabytes: Giga < Tera < Peta < Exa < **Zetta** < Yotta
- About 235 ZB created in 2026 alone; 527 ZB by 2029 (IDC Global DataSphere forecast)
- The world's data roughly doubles every three years.

Source: Statista, from IDC Global DataSphere; 2026 and 2029 are forecasts

The Green Circle: Accuracy



- Facebook's DeepFace matched faces with **97.25%** accuracy in 2014, about the level of human eyes (97.53%) ([Forbes, 2014](#)) $\rightsquigarrow$ **Now?**



AI in 2026: Two Headlines

ANTHROPIC

Research ▾ Policy Commitments ▾ Learn ▾ News [Try Claude](#) ▾

Announcements

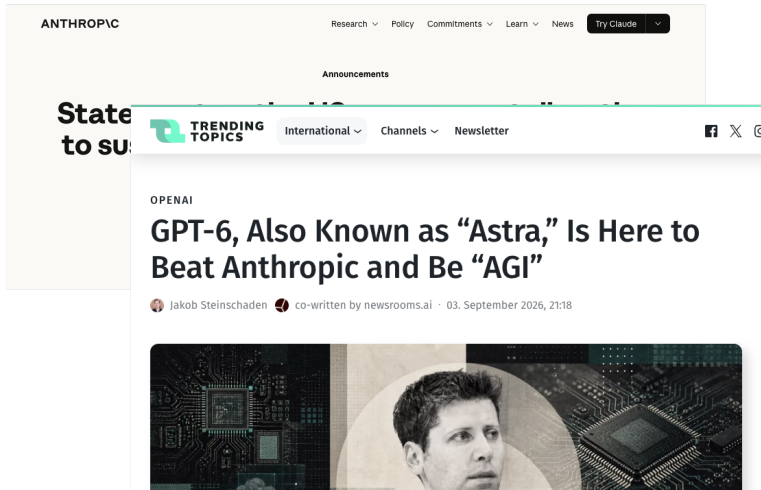
Statement on the US government directive to suspend access to Fable 5 and Mythos 5

Jun 12, 2026

The US government, citing national security authorities, has issued an export control directive to suspend all access to Fable 5 and Mythos 5 by any foreign national, whether inside or outside the United States, including foreign national Anthropic employees. The net effect of this order is that we must abruptly disable Fable 5 and Mythos 5 for **all** our customers to ensure compliance. **Access**

Anthropic, June 12, 2026 (lifted June 30); Trending Topics, September 3, 2026

AI in 2026: Two Headlines



Anthropic, June 12, 2026 (lifted June 30); Trending Topics, September 3, 2026

Recent Hype on ML/AI



Harvard
Business
Review

Sign In

Intellectual Property

Generative AI Has an Intellectual Property Problem

by Gil Appel, Juliana Neelbauer, and David A. Schweidel

April 07, 2023



HBR Staff/Pexels

Reference: <https://hbr.org/2023/04/generative-ai-has-an-intellectual-property-problem>



Recent Hype on ML/AI



Harvard
Business
Review

Sign In

Intellectual Property

Forbes

News Events Lists Life

Sign-in

Subscribe



Uncategorized

Major record labels sue AI music startups for allegedly copying songs to train AI

By Molly Bohannon

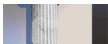
June 25, 2024



The three largest record labels are suing two artificial intelligence music companies, alleging the firms committed “mass infringement of



Reference: <https://www.forbes.com/sites/mollybohannon/2024/06/24/major-record-labels-sue-ai-music-startups-for-allegedly-copying-songs-to-train-ai/>



HBR Staff/Pexels

Reference: <https://hbr.org/2023/04/generative-ai-has-an-intellectual-property-problem>



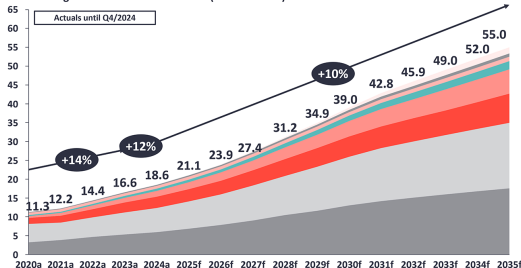
The Blue Circle: Connectivity



October 2025

Global IoT market forecast (in billions of connected IoT devices)

Number of global active IoT connections (installed base) in billions



Connectivity type	CAGR
	20–24 25–35
Cellular 5G/6G IoT	(208%) (35%)
Wireless Neighborhood Area Networks (WWAN)	(16%) (10%)
Wired IoT	(4%) (4%)
Other	(20%) (12%)
LPWA	(36%) (13%)
Cellular IoT (excl. 5G, LPWA)	(17%) (9%)
Wireless Personal Area Networks (WPAN)	(7%) (9%)
Wireless Local Area Networks (WLAN)	(16%) (10%)
XX% = CAGR	

Note: IoT connections do not include any computers, laptops, fixed phones, cell phones, or consumer tablets. Counted are active nodes/devices or gateways that concentrate the end-sensors, not every sensor/actuator. Simple one-directional communications technology (e.g., RFID or NFC) is not considered. Wired includes ethernet and field buses (e.g., connected industrial PLCs or I/O modules). The number of wired IoT aggregation nodes represents the primary connection point and excludes all wired end nodes. Cellular includes 2G, 3G, 4G, 5G. LPWA includes unlicensed and licensed low-power networks. WPAN includes Bluetooth, ZigBee, Z-Wave or similar. WWAN includes Wi-Fi and related protocols. WLAN includes non-short-range mesh, such as Wi-SUN. Other includes satellite and unclassified proprietary networks with any range.

Source: IoT Analytics Research 2025 – Global Cellular IoT Connectivity Tracker & Forecast. Conditions for republishing: Source citation with link to original post and company website.

- More than **21 billion** IoT devices connected worldwide in 2025, heading for 39 billion by 2030 (IoT Analytics, 2025)
- Not phones or laptops: smart TVs, earbuds, watches and rings, speakers, cameras, meters, cars, scooters. About 2.6 of them per person on Earth (8.2 billion people, UN, 2024)

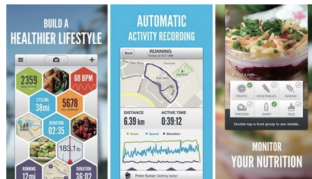


Wearable IOT & Sensor Data



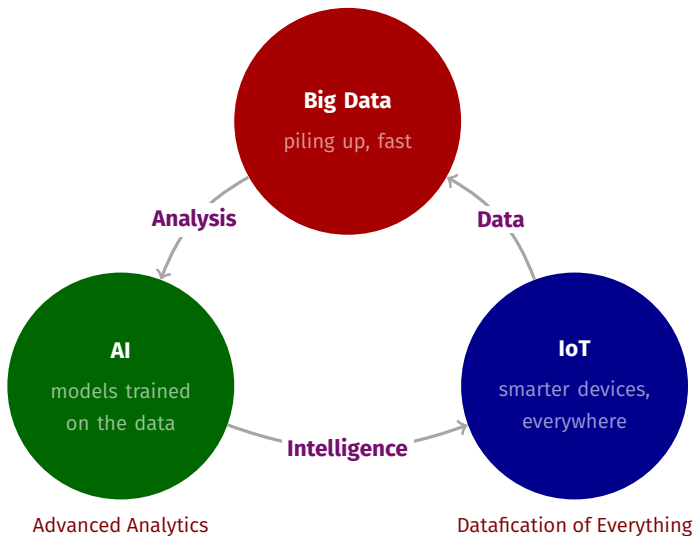
[Reference: Medium, Aug. 2018]

Type	Feature	Usage Purposes
Heart Rate Sensor	Quantify heartbeat per minute	Monitor health status
Pressure Sensor	Measure ambient pressure	Monitor altitude
CdS Sensor	Measure light intensity	Check the presence of light
Infrared Sensor	Detect infrared light emitted by a target	Monitor body temperature
Proximity Sensor	Measure relative distance to an object	Monitor surrounding environment
Gyro Sensor	Measure angular velocity using 3 axes	Monitor intensity of an activity
Velocity Sensor	Measure movement using 3 axes	Monitor intensity of an activity
GPS Sensor	Measure current location	Track location
Temperature Sensor	Measure temperature in the atmosphere	Monitor surrounding environment
Humidity Sensor	Measure humidity in the atmosphere	Monitor surrounding environment
Acoustic Sensor	Measure nearby acoustic waves	Monitor action
Glucose Sensor	Measure glucose level in blood	Monitor health



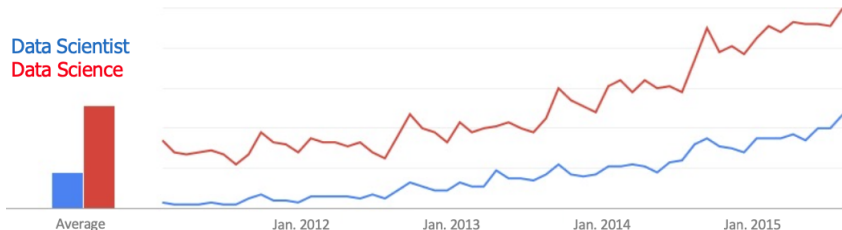
[Reference: Azumio]

Three Core Tech in the Era of Digitization



Promising Job in the Era of DT: Data Scientist

Data Scientist: The Sexiest Job of the 21st Century (HBR, 2012.10)



Fortune

Supply is down at merely 20% while approximately 4.5 million data scientists are needed globally.

Salaries for entry level data scientists had risen to \$91,000 nationally and \$110,000 in Silicon Valley.

– Glass Door Report

How About Now? A Follow-Up Article in HBR..

Is Data Scientist Still the Sexiest Job of the 21st Century?

by Thomas H. Davenport and DJ Patil

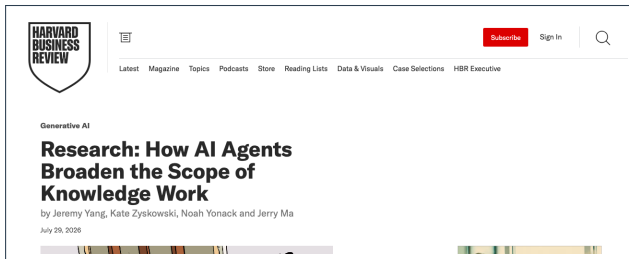
July 15, 2022



HBR Staff/StudioM/Moritz Otto/Getty Images

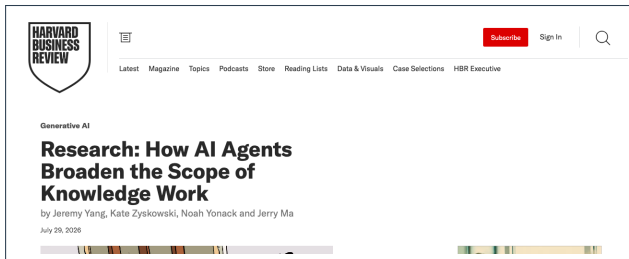
Reference: <https://hbr.org/2022/07/is-data-scientist-still-the-sexiest-job-of-the-21st-century>

2026: And Now?



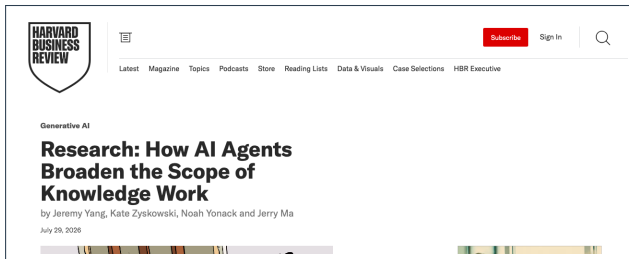
- AI agents widen what people attempt: 59% of agent requests cross occupational lines, 23% are tasks never tried before ([HBR, July 2026](#))

2026: And Now?



- AI agents widen what people attempt: 59% of agent requests cross occupational lines, 23% are tasks never tried before ([HBR, July 2026](#))
- **What does this mean?** $\rightsquigarrow$ lower entry barrier: people lacking fundamental background knowledge now run analyses, **prone to making logic errors!**

2026: And Now?

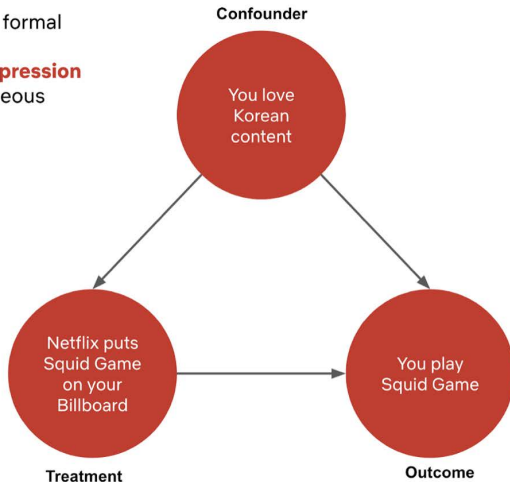


- AI agents widen what people attempt: 59% of agent requests cross occupational lines, 23% are tasks never tried before ([HBR, July 2026](#))
- **What does this mean?** $\rightsquigarrow$ lower entry barrier: people lacking fundamental background knowledge now run analyses, **prone to making logic errors!**
- Why learn data science in the age of AI? In business analytics, a wrong conclusion can cost a fortune.

What problems are data scientists/business analysts working on?

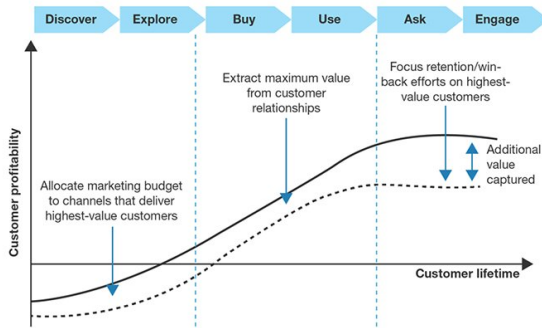
Causality

Causal Inference provides formal tools to tease out the true **incremental** value of an **impression** for each profile: Heterogeneous Treatment Effect (**HTE**)



Compared to machine learning, causal inference allows us to build a robust framework that controls for confounders in order to estimate the true incremental impact to members

Prediction



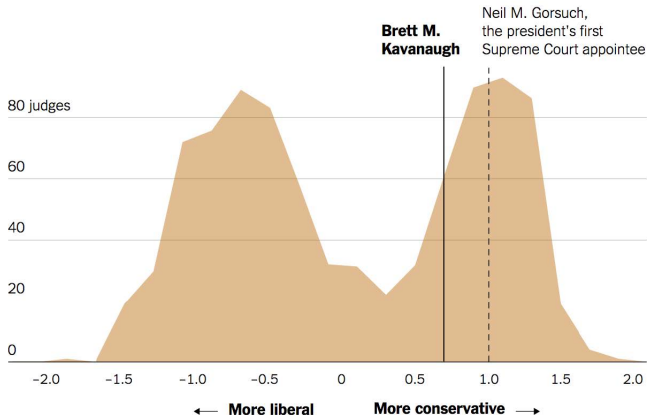
#CXNYC

FORRESTER®

- **Customer lifetime value (CLV):** how much you will be worth to a firm $\rightsquigarrow$ predicted from what you have done so far
- Treated accordingly: high CLV $\rightsquigarrow$ targeted for upselling, kept with retention offers; low CLV $\rightsquigarrow$ fewer offers, less effort

Measurement

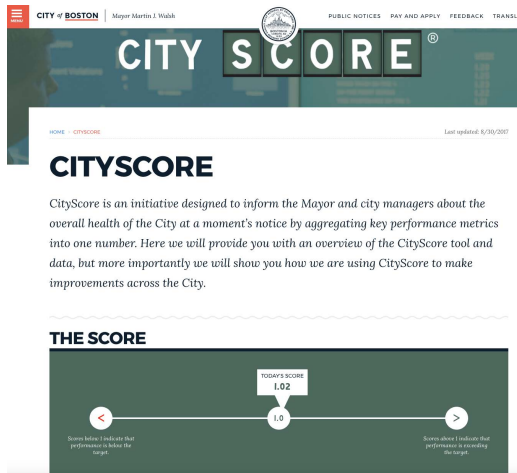
How Kavanaugh's Ideology Compares With Other Federal Judges



Based on the campaign finance scores of all current and former federal district and court of appeals judges nominated since 1980. | Source: [Database on Ideology, Money in Politics, and Elections](#); Adam Bonica, Stanford University Department of Political Science; Maya Sen, Harvard University, Kennedy School of Government; Adam Chilton and Kyle Rozema, University of Chicago Law School.

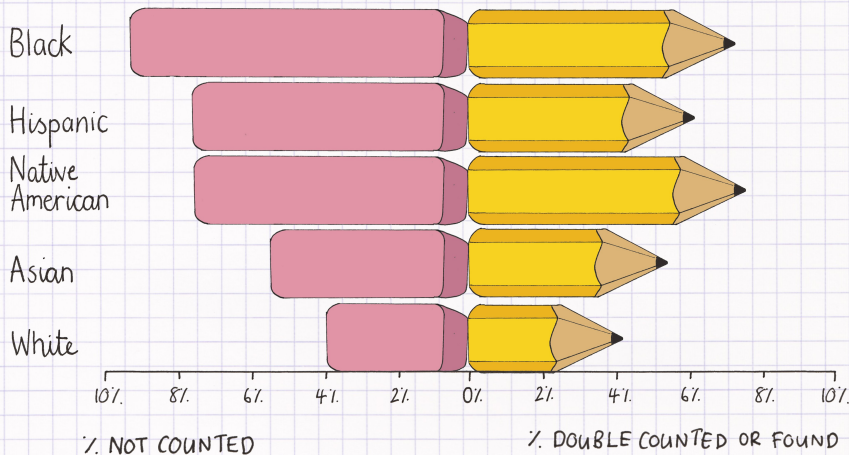
- “You can’t improve what you don’t measure.” - Peter Drucker

Making Organizations Work Better



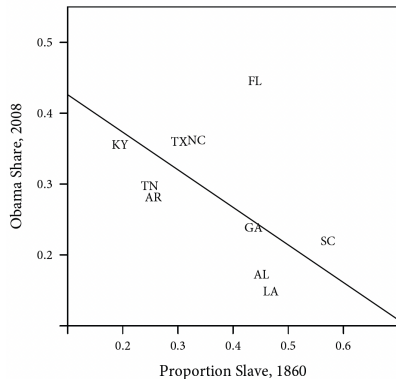
Combining Art and Data to Inform

Who Gets Miscounted In The Census?

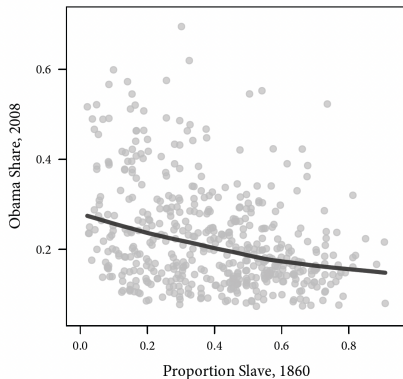


Understanding How the Past Matters

States



Counties



2/ A First Look at R

The Two Computer Revolutions



The Frontier of Computing

- Touch-based interfaces
- Single app at a time
- Little multitasking between apps
- Hides the file system



Where Statistical Computing Lives

- Windows and pointers
- Multi-tasking, multiple windows
- Works heavily with the file system
- Underneath it's UNIX and the command line

Plain-Text Tools for Data Analysis



- Often free, open-sourced, and powerful
 - e.g., R, Quarto, Git (for version control)
- Large, friendly communities around them
- Tons of resources
- But... far from the touch-based paradigm of modern computing

⇒ **So why use them?**

The process of data science is intrinsically messy

Office vs. Engineering Model of Computing

What's real in the project? How are changes managed?

In the Office Model

- Formatted documents are real
- Intermediate outputs copy/pasted into documents
- Changes are tracked inside files
- Final output is the file you're working on (e.g., Word or PDF)

In the Engineering Model

- Plain-text files are real
- Intermediate outputs are produced via code
- Changes are tracked outside files
- Final outputs are assembled programmatically and converted to the desired format

Pros and Cons to Each Approach

- Office model:
 - Everyone knows Word, Excel, Google Docs.
 - "Track changes" is powerful and easy.
 - Wait, how did I make this figure?
 - Which version of my code made this table?
 - Jaewon_Yoo_report_final_submitted_edits_FINAL_v2.docx
- Engineering model:
 - Plain text is universally portable.
 - Push button, recreate analysis.
 - Why won't R just do what I want!
 - Version control is a pain.
 - Object of type `'closure'` is not subsettable.

We'll tend toward the Engineering model because it's better suited to keep the mess in check.

R versus RStudio

```
R
R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

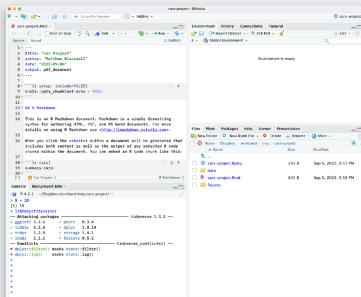
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

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.

>
```



An Overview

Write notes and codes in R Markdown or in an R Script

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the Render button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
(r)
1 + 1
```

You can add options to executable code like this

```
(r)
# echo: false
2 + 2
```

The `echo: false` option disables the printing of code (only output is displayed).

Environment is empty

Console

```
R 4.2.3 -- (4-bit)

R version 4.2.3 (2023-03-15) -- "Shortstop Beagle"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin20 (64-bit)

It 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.

Warning: macOS is reporting that you have not yet agreed to the Xcode license.
This can occur if Xcode has been updated or reinstalled (e.g. as part of a macOS update).
Some features (e.g. Git / SVN) may be disabled.
```

An Overview

The screenshot displays the RStudio interface with a Quarto document open. The document contains a title block, a paragraph about Quarto, a section on running code, and two code chunks. The first code chunk is a simple R expression `1 + 1`. The second code chunk is `2 + 2` with the `echo: false` option. The console on the right shows the R version (4.2.3) and various copyright and license information. A yellow box highlights the console area with the text: "Console: Run code, send code here, inspect output".

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the Render button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
(r)
1 + 1
```

You can add options to executable code like this

```
(r)
# echo: false
2 + 2
```

The `echo: false` option disables the printing of code (only output is displayed).

Console: Run code, send code here, inspect output

```
R version 4.2.3 (2023-03-15) -- "Shortstop Beagle"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin20 (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.

Warning: macOS is reporting that you have not yet agreed to the Xcode license.
This can occur if Xcode has been updated or reinstalled (e.g. as part of a macOS update).
Some features (e.g. Git / SVN) may be disabled.
```

An Overview

Interacting with R objects, running local tutorials/help, display plots and project files

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the Render button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
(r)
1 + 1
```

You can add options to executable code like this

```
(r)
2 + 2 echo: false
```

The `echo: false` option disables the printing of code (only output is displayed).

R version 4.2.3 (2023-03-15) -- "Shortstop Beagle"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin20 (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.

Warning: macOS is reporting that you have not yet agreed to the Xcode license.
This can occur if Xcode has been updated or reinstalled (e.g. as part of a macOS update).
Some features (e.g. Git / SVN) may be disabled.

A CloseR Look

- Code that you can type and run:

```
1  ## Any R code that begins with the # character is a comment
2  ## Comments are ignored by R
3  > my_numbers <- c(4, 8, 15, 16, 23, 42) # Anything after # is also a comment
```

- Output from code are prefixed with element number(s)/counter in brackets:

```
1  > my_numbers ## we just created this.
2  [1] 4 8 15 16 23 42
3
4  > letters ## this is built into R
5  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l"
6  [13] "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x"
7  [25] "y" "z"
8
9  > pi ## this is also built in
10 [1] 3.141593
```

We Do Things in R with Functions

- Functions take in objects, perform actions, and return outputs:

```
1 mean(x = my_numbers)
2 [1] 18
```

- where x is the argument name,
- my_number is what we are passing to that argument.

- If you omit the argument name, R will assume the default order:

```
1 mean(my_numbers)
2 [1] 18
```

Getting Help with R

- How do we know the default argument order? Examine the (built-in) help files:

```
1 help(mean)
2 ?mean ## shorter
```

- Sometimes inscrutable, so it helps to look else where:
 - Google, StackOverflow, Twitter, RStudio Community
 - Ask on eeclash discussion board!
 - Come talk to me & TAs for help
- Get help **early** before becoming too frustrated!
 - Easy to overlook small issues like missing commas, etc.

Functions in R ‘Live’ in Packages

- Packages are bundles of functions written by other users that we can use:

```
1 install.packages("ggplot2")
```

- Install packages using `install.packages()` to have them set up on your machine.

```
1 library(ggplot2) ## or require()
```

- Load them into your R session with `library()` or `require()`

~> We will now be able to use any function provided by `ggplot2`

Functions in R ‘Live’ in Packages (Cont)

- We can also use the `packageName::` prefix to access package functions without loading it:

```
1 knitr::kable(head(mtcars))
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.62	16.46	0	1	4	4
Mazda RX4	21.0	6	160	110	3.90	2.88	17.02	0	1	4	4
Wag											
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.22	19.44	1	0	3	1
Hornet Sport-	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
about											
Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1

The Acts of Coding

```
library(ggplot2)
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point()
```

Figure 1. Writing code

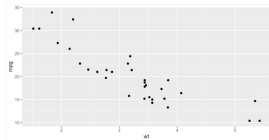


Figure 2. Looking at output

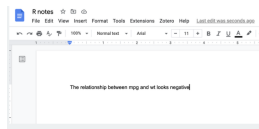
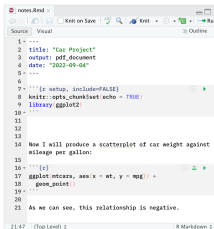


Figure 3. Taking notes

How to do all of these efficiently?

Quarto (or R Markdown) Files to the Rescue



```
1 ---  
2 title: "Car Project"  
3 output: pdf_document  
4 date: "2022-09-04"  
5 ---  
6  
7 '[ setup, include=FALSE]  
8 knitr::opts_chunk$set echo = TRUE  
9 library(ggplot2)  
10  
11  
12  
13  
14 Now I will produce a scatterplot of car weight against  
15 mileage per gallon:  
16  
17 '[x]  
18 ggplot(mtcars, aes(x = wt, y = mpg)) +  
19   geom_point()  
20  
21  
22 As we can see, this relationship is negative.  
23  
24  
25
```

Fig.: Quarto/Rmd file

Keep code and
notes together in
plain text

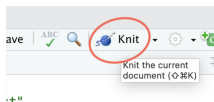


Fig.: Render

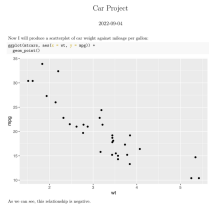


Fig.: PDF output

Produce
nice-looking
outputs in
different formats

Markdown: Formatting in Plain Text

Non-code text in .qmd/ .Rmd files is plain text with formatting instructions

syntax

```
Plain text
End a line with two spaces to start a new paragraph.
*italics* and _italics_
**bold** and __bold__
superscript^2^
~~strikethrough~~
[link](www.rstudio.com)

# Header 1
## Header 2
### Header 3
#### Header 4
##### Header 5
##### Header 6

endash: --
endash: ---
ellipsis: ...
inline equation: $A = \pi r^2$
image: 

horizontal rule (or slide break):
***

> block quote

* unordered list
* item 2
  + sub-item 1
  + sub-item 2

1. ordered list
2. item 2
  + sub-item 1
  + sub-item 2
```

becomes

Plain text
End a line with two spaces to start a new paragraph.
italics and *italics*
bold and **bold**
^{superscript²}
~~strikethrough~~
[link](#)

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

endash: –
endash: —
ellipsis: …
inline equation: $A = \pi * r^2$

image: 

horizontal rule (or slide break):

block quote

- unordered list
 - item 2
 - sub-item 1
 - sub-item 2
1. ordered list
 2. item 2
 - sub-item 1
 - sub-item 2

```
1 ---
2 title: "Car Project"
3 author: "Jaewon Yoo"
4 date: "2024-08-01"
5 output: pdf_document
6 ---
```

Header contains metadata
and sets options about the
whole document

```
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
```

Code chunk

Plain text with markdown
formatting

```
11
12 ## R Markdown
```

```
13
14 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.
```

```
15
16 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:
```

```
17
18 ```{r cars}
19 summary(cars)
20 ```
```

Chunks can have
names and options

Can "play" chunks
interactively

```
21
22 ## Including Plots
```

```
23
24 You can also embed plots, for example:
```

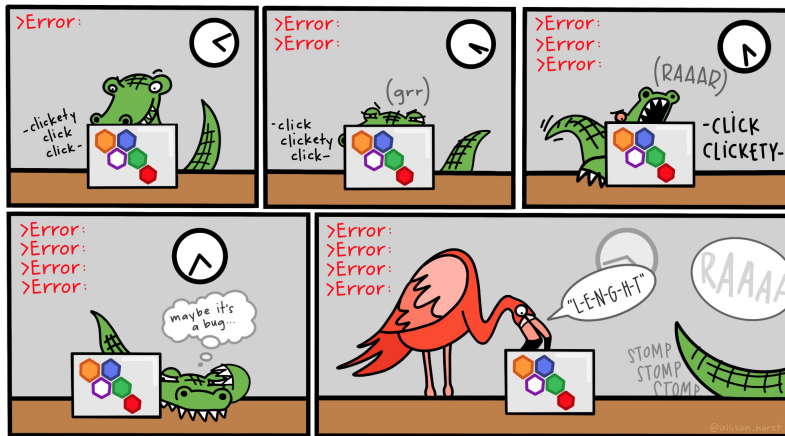
```
25
26 ```{r pressure, echo=FALSE}
27 plot(pressure)
28 ```
```

Code chunks replaced
with output when Knitted

```
29
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated
the plot.
```

Try to Type Your Code by Hand!

Typing Speeds Up the Try-Fail Cycle



Physically typing the code is best way to familiarize yourself with R and the try-fail-try-fail-try-succeed cycle!

3/ Course Overview

About Me



Figure 1: Illustration of Counterfactuals in CI vs. XAI

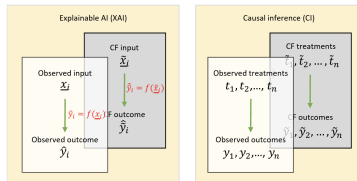


Table 1: Overview of differences in use of counterfactuals

Dimension	Causal Inference	Explainable AI
Purpose	Estimate causal effect	Explain a predicted outcome
Type of Relationship	Causal	Correlation
Quantity of Interest	Difference in outcomes	Input changes
Aggregation Level	Sample	Unit
Modified Object	Model	Data
Assumptions	On data generating function	On given predictive model
Performance Evaluation	Standard error, conf. interval, significance, robustness tests	Feasibility, sparseness, etc.

What will you learn in this class?

- Summarize and visualize data
- Wrangle messy data into tidy forms
- Evaluate claims about causality
- Be able to use linear regression to analyze data
- Understand uncertainty in data analysis and how to quantify it
- Use professional tools like R, RStudio, git, and GitHub

Teaching Assistants



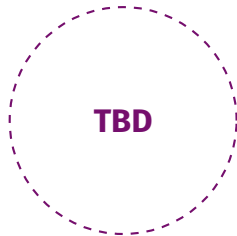
Anh Tran

anhtranpt@iss.nthu.edu.tw



Daffa Onielda

daffaonielda@gapp.nthu.edu.tw



TBD

Textbooks

- MD** Ismay, Chester and Albert Y. Kim. 2022. Statistical Inference via Data Science: A ModernDive into R and the Tidyverse.
- QSS** Imai, Kosuke and Nora Webb Willaims. 2022. Quantitative Social Science: An Introduction with Tidyverse, 2022. Princeton University Press.
- IMS** Mine Cetinkaya-Rundel and Johanna Hardin. 2021. Introduction to Modern Statistics. OpenIntro.
- AAG** Jared P. Lander, O'Reilly Media, 2017. R for Everyone: Advanced Analytics and Graphics, 2nd Edition.
- VT** Nathan Yau, John Wiley & Sons, 2011. Visualize This: The Flowing Data Guide to Design, Visualization, and Statistics.

Course Information

For the up-to-date syllabus, weekly schedule, grading policy, and readings, please visit the course website. It is updated during the semester.

ISS4066 - Fall 2026

OverviewScheduleMaterialsEvaluationPolicies← Back

Programming for Business Analytics (Undergraduate)

An undergraduate introduction to data analysis with R for business: wrangling and visualizing data, evaluating causal claims, linear regression, and quantifying uncertainty.

ISS4066 📅 Fall 2026 📍 TSMC Bldg. R421 🕒 Monday, 14:20-17:20



Syllabus PDF

Course Overview

This course teaches you to turn messy business data into defensible answers with R. You will learn to wrangle and visualize data, evaluate claims about causality, fit and interpret linear regressions, and quantify how much you should trust an estimate through sampling distributions, the bootstrap, confidence intervals, and hypothesis tests. Every method is practiced on real datasets, and the semester culminates in a team project published as a portfolio-ready article with a public GitHub repository.

Agentic AI now writes much of the code. The skill that matters has shifted from writing code to directing and verifying it: without the underlying building blocks, you cannot tell when an analysis is wrong, and you end up steered by the tool rather than steering it. The course therefore builds programming and visualization foundations first, then turns to statistical inference, and every in-class exercise ends with a verification step.

This is the undergraduate course (ISS4066, Mondays). Graduate students take ISS5066 on Thursdays, which carries research-oriented project requirements. Students may not receive credit for both.

Learning Objectives

- Data wrangling and visualization: filter, summarize, group, join, and reshape data

Course Details

📅 Sep 7 - Dec 26, 2026
🕒 Monday
🕒 14:20-17:20
📍 TSMC Bldg. R421

Communication

Lecture slides, handouts, in-class exercises, assignments, and grades are on [eclass](#) (NTHU's learning platform; enrolled students are added automatically). Announcements go out through eclass. For private matters, email the instructor. Slides are not posted on this public page.

What's New

This page is updated during the semester. Recently changed items are listed below.

<https://j1yoo.github.io/teaching/pba/>

Software Tools

- **R Statistical Programming Language:**

- R is an open-source language that can be used free of charge and can be downloaded from <https://www.r-project.org/>.

- **Integrated Development Environment (IDE):**

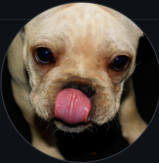
- For improved productivity and convenience, students will also use RStudio, a widely used integrated development environment (IDE) for R which is available at <https://www.rstudio.com/products/rstudio/> for download.
- If you use a MacBook with the M1 chip/Apple silicon, RStudio might not work on your machine. If it does not, you can also use Visual Studio Code (VS Code) to edit your codes which can be downloaded at <https://code.visualstudio.com/>.

git and GitHub



- Other core tools: git and GitHub
 - **Version control system:** an archive of project versions.
 - Allows you to revert back to old versions easily
 - Makes collaboration much more manageable.
- Will feel very odd at first, but you git used to it
- Why learn this now?
 - Knowing git/GitHub is a huge plus for data jobs.
 - Your GitHub profile can showcase your amazing new skills with data!

Sample GitHub profile



Mara Averick
batpigandme

[Follow](#)

810 followers · **137** following

 [@tidyverse](#)
 Missoula, MT
 <https://dataandme.com>
 @dataandme@mstdn.social
 <https://bsky.app/profile/dataandme.bsky.social>
 [@dataandme](#)
 [in/maraaverick](#)

Achievements





Overview **Repositories** 238 **Projects** **Packages** **Stars** 845

batpigandme / README.md

Hiieee! I'm Mara. 🐾

```
> # I'm the tidyverse developer advocate
> tidyverse::tidyverse_logo()
```



[Followers](#) 810 [Follow @dataandme](#)

 I'm also on Mastodon.

Popular repositories

night-owlish Public

 An RStudio, tmThemes, and Ace editor adaptation of @sdras' Night Owl VS Code theme...

 CSS  160  59

tverse_contributing Public

 "contributing to the tidyverse" from rstudio:conf(2018)

 22

tidymueller Public

thats-not-science Public

Academic Honesty and Plagiarism

- All work submitted for academic evaluation must be the student's own.
 - The penalty for violation of academic integrity will result in a zero for that assignment for the first time.
 - Subsequent violation(s) will result in a failing grade for the course.
- Academic dishonesty comprises of, but is not limited to, the following:
 - **Cheating:** Copying from other students' quizzes and assignments or allowing others to copy from one's own.
 - **Plagiarism:** Using other people's original work without giving appropriate credit or acknowledgment to the authors or sources.
 - **Self-plagiarism:** Submitting a piece of work in more than one course without the explicit permission of the instructors involved.
 - **Misrepresentation of authorship:** Submitting work as one's own, which has been prepared by or purchased from others. Students will be asked to upload their submission materials to eclass.

AI Use Policy**

- To maintain academic integrity, students who choose to use AI tools (e.g., GPT-4o, Claude 3.5, Gemini) must adhere to the following guidelines:
 - **Thorough Explanation:** Students must provide a detailed explanation of how the AI tool was used in the completion of the assignment.
 - **Original AI Responses:** Alongside their submission, students are required to submit the original responses generated by the AI tool.
 - **Critical Evaluation and Personal Input:** Students should critically evaluate the AI-generated content and clearly indicate the portions of the work that have been modified or expanded upon with their own insights and understanding.
 - **Proper Attribution:** Proper attribution should be given to the AI-generated content used, indicating clearly the sections that are AI-generated.
- Remember, attempting to cheat the system by **heavily relying on AI-generated content without substantial personal input** is ultimately the student's loss!

Final Project (40% undergraduate / 50% graduate)

- Open-ended data analysis project on a topic of your choice (e.g., voting trends, racial disparities in policing, NBA salaries, box-office economics of Marvel movies).
- Steps:
 1. Develop a clear research question
 2. Collect and prepare relevant data
 3. Apply methods from the course
 4. Present results in a clear and accessible way
- Goal: produce a polished, portfolio-ready project demonstrating your data skills.
- Deliverables include: research question, dataset overview, visualizations, regression analysis with plain-language interpretation, discussion of limitations, and a GitHub repo with all code.
- Milestones: GitHub repository (Oct 11), data and proposal (Oct 25), first visualization (Nov 15), first analysis (Dec 6), final report (Dec 20).

Assignments (20% undergraduate / 15% graduate)

- Five assignments spaced throughout the semester.
- Assignments connect directly to class topics and typically involve running and reporting a statistical method.
- Example: after covering OLS, you will analyze a provided dataset and report the results in a short article-style write-up.
- Deliverables: 2–3 page report generated using R Markdown/Quarto.
- Each assignment = 4% of the course grade (3% in the graduate section).
Late penalty: –1 percentage point per day (e.g., 2 days late caps at 2% out of 4%).

Problem Sets (15% / 10%) & In-Class Exercises (5%)

- **Problem Sets:**

- Regular problem sets drawn from recommended textbooks.
- Designed to reinforce understanding of key statistical/analytical concepts.

- **In-class Exercises:**

- No take-home quizzes this year.
- Five short in-class exercises (1% each): finish in class, upload to eeclass the same day.
- Each ends with a verification task: check what your (or your AI's) code actually did.

Participation (10%) & Attendance (10%)

- **Participation:**

- Stay engaged and active in discussions.
- Positive and thoughtful contributions improve your score.
- Be ready to share your perspectives and ideas.

- **Attendance:**

- Attend every class with laptops ready for exercises.
- Notify me and the TAs in advance if you must miss a class.
- You remain responsible for all missed material and required work.

Grading Policy

- **Breakdown:**
 - Final Project: 40% (graduate: 50%)
 - Assignments: 20% (graduate: 15%)
 - Problem Sets: 15% (graduate: 10%)
 - In-Class Exercises: 5%
 - Participation: 10%
 - Attendance: 10%
- **Bump-up:** Students close to cutoffs who stay engaged and make consistent contributions may receive a grade boost.

What Should You Do This Week?

- Install R and RStudio if you have not yet; open RStudio and type a line or two.
- Log in to eeclass and post your self-introduction on the “Introductions” board by Friday, September 11.
- Read through the course website: syllabus, weekly schedule, and the AI use policy.
- Next week: a first look at data visualization. Bring your laptop.

See You Next Week! :)

Contact Information:

jaewon.yoo@iss.nthu.edu.tw

<https://j1yoo.github.io/>

