



Beginning Mathematica and Wolfram for Data Science

Applications in Data Analysis, Machine
Learning, and Neural Networks

—
Second Edition

—
Jalil Villalobos Alva

Apress®

Beginning Mathematica and Wolfram for Data Science

**Applications in Data Analysis,
Machine Learning,
and Neural Networks**

Second Edition

Jalil Villalobos Alva

Apress®

Beginning Mathematica and Wolfram for Data Science: Applications in Data Analysis, Machine Learning, and Neural Networks

Jalil Villalobos Alva
Mexico City, Mexico

ISBN-13 (pbk): 979-8-8688-0347-5
<https://doi.org/10.1007/979-8-8688-0348-2>

ISBN-13 (electronic): 979-8-8688-0348-2

Copyright © 2024 by Jalil Villalobos Alva

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Melissa Duffy
Development Editor: James Markham
Editorial Project Manager: Gryffin Winkler
Copyeditor: Kim Burton

Cover designed by eStudioCalamar

Cover image designed by Mathew Schwartz on Unsplash (www.unsplash.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, Suite 4600, New York, NY 10004-1562, USA. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub. For more detailed information, please visit <https://www.apress.com/gp/services/source-code>.

If disposing of this product, please recycle the paper

To my family, who supported me in all aspects

Table of Contents

About the Author	xiii
About the Technical Reviewer	xv
Acknowledgments	xvii
Introduction	xix
Chapter 1: Introduction to Mathematica	1
Why Mathematica?	1
The Wolfram Language	2
Structure of Mathematica	3
Design of Mathematica	6
Mathematica Environment.....	9
Notebook Interface	9
Text Processing	12
Palettes.....	14
Notebook Style and Features	15
Expression in Mathematica.....	18
Assigning Values.....	19
Built-in Functions	22
Dates and Time.....	23
Strings	25
Basic Plotting.....	27
Logical Operators and Infix Notation	30
Algebraic Expressions	33
Solving Algebraic Equations	34
Using Wolfram Alpha Inside Mathematica	37
Delayed and Immediate Expressions.....	40

TABLE OF CONTENTS

- Improving Code 41
 - Code Performance 42
 - Handling Errors 43
 - Debugging Techniques 45
- How Mathematica Works 47
 - How Computations are Made (Form of Input)..... 47
 - Searching for Assistance 50
 - Notebook Security 53
- Summary..... 54
- Chapter 2: Data Manipulation 55**
 - Lists 55
 - Types of Numbers..... 56
 - Working with Digits 60
 - A Few Mathematical Functions 61
 - Numeric Function 63
 - Lists of Objects 64
 - List Representation..... 65
 - Generating Lists..... 65
 - Arrays of Data..... 68
 - Nested Lists..... 71
 - Vectors..... 72
 - Matrixes..... 73
 - Matrix Operations 75
 - Restructuring a Matrix..... 76
 - Manipulating Lists..... 77
 - Retrieving Data 77
 - Assigning or Removing Values 79
 - Structuring List 82
 - Criteria Selection 84
 - Summary..... 87

Chapter 3: Working with Data and Datasets	89
Operations with Lists	89
Arithmetic Operations to a List.....	90
Applying Functions to a List	91
Defining Own Functions	93
Pure Functions.....	95
Indexed Tables	96
Tables with the Wolfram Language	96
Associations	101
Dataset Format	103
Constructing Datasets	103
Accessing Data in a Dataset.....	109
Adding Values.....	113
Dropping Values.....	117
Filtering Values	119
Applying Functions.....	122
Functions by Column or Row.....	127
Joining and Merging Datasets.....	132
Customizing a Dataset.....	134
Generalization of Hash Tables	138
Summary.....	145
Chapter 4: Import and Export	147
Importing Files	148
CSV and TSV Files.....	148
XLSX Files.....	150
JSON Files	153
Web Data	156
Semantic Import	157
Quantities	159
Datasets with Quantities	160
Costume Import (Dealing with Large Datasets).....	164

TABLE OF CONTENTS

- Export..... 166
 - Other Formats..... 169
 - XLS and XLSX Formats 173
 - JSON Formats..... 174
 - Content File Objects 177
 - Searching Files with Wolfram Language 178
- Connecting to External Services 179
 - External Connections..... 179
 - External Resources..... 181
 - Database and File Operations (SQL) 184
- Summary..... 186
- Chapter 5: Data Visualization 187**
 - Basic Visualization 187
 - 2D Plots 187
 - Plotting Data..... 191
 - Plotting Defined Functions 196
 - Customizing Plots 197
 - Adding Text to Charts..... 197
 - Frame and Grids 200
 - Filled Plots 202
 - Filling Patterns and Gradient..... 204
 - Combining Plots 206
 - Multiple Plots..... 207
 - Multiaxis Plots 210
 - Coloring Plot Grids..... 211
 - Colors Palette..... 216
 - 3D Plots..... 218
 - Customizing 3D Plots..... 219
 - Hue Color Function and List3D 220

Contour Plots.....	222
3D Plots and 2D Projections	226
Plot Themes	227
Summary.....	231
Chapter 6: Statistical Data Analysis	233
Random Numbers	233
Random Sampling	235
Systematic Sampling.....	237
Commons Statistical Measures	239
Measures of Central Tendency	239
Measures of Dispersion.....	240
Statistical Charts.....	242
Bar Charts.....	242
Histograms	245
Pie Charts and Sector Charts.....	250
Box Plots.....	252
Distribution Chart	254
Charts Palette.....	256
Ordinary Least Squares Method.....	262
Pearson Coefficient	264
Linear Fit	265
Model Properties	266
Summary.....	269
Chapter 7: Data Exploration.....	271
Wolfram Data Repository	271
Wolfram Data Repository Website	272
Selecting a Category	274
Extracting Data from the Wolfram Data Repository	275
Accessing Data Inside Mathematica	278
Data Observation and Querying.....	281

TABLE OF CONTENTS

- Descriptive Statistics 287
 - Table and Grid Formats 289
 - Dataset Visualization 293
 - Data Outside Dataset Format 296
 - 2D and 3D Plots 297
- Summary..... 301
- Chapter 8: Machine Learning with the Wolfram Language..... 303**
 - Gradient Descent Algorithm 303
 - Getting the Data..... 304
 - Algorithm Implementation 305
 - Multiple Alphas 307
 - Linear Regression 308
 - Predict Function 308
 - Boston Dataset 309
 - Model Creation 310
 - Model Measurements 316
 - Model Assessment 318
 - Retraining Model Hyperparameters..... 319
 - Logistic Regression..... 321
 - Titanic Dataset..... 321
 - Data Exploration 325
 - Classify Function 327
 - Testing the Model 332
 - Data Clustering 338
 - Clusters Identification..... 338
 - Choosing a Distance Function 340
 - Identifying Classes 342
 - K-Means Clustering..... 343
 - Dimensionality Reduction..... 345
 - Applying K-Means 347
 - Changing the Distance Function..... 349

Different k's	350
Cluster Classify	353
Summary	357
Chapter 9: Neural Networks with the Wolfram Language	359
Layers	360
Input Data	360
Linear Layer	360
Weights and Biases	361
Initializing a Layer	362
Retrieving Data	364
Mean Squared Layer	365
Activation Functions	368
Softmax Layer	372
Function Layer	374
Encoder and Decoders	375
Encoder	375
Pooling Layer	379
Decoders	380
Applying Encoder and Decoders	382
NetChains and Graphs	383
Containers	383
Multiple Chains	386
NetGraphs	387
Combining Containers	393
Network Properties	395
Exporting and Importing a Model	399
Summary	402
Chapter 10: Neural Networks Framework	403
Training a Neural Network	403
Data Input	403
Training Phase	405

TABLE OF CONTENTS

- Model Implementation..... 406
- Batch Size and Rounds..... 408
- Training Method (NetTrain) 412
- Measuring Performance 413
- Model Assessment 415
- Exporting a Neural Network 417
- Wolfram Neural Net Repository..... 418
 - Selecting a Neural Net Model..... 419
 - Accessing Inside Mathematica..... 421
 - Retrieving Relevant Information 422
- LeNet Neural Network..... 423
 - LeNet Model 423
 - MINST Dataset..... 424
 - LeNet Architecture..... 425
 - MXNet Framework..... 426
 - Preparing LeNet..... 428
 - LeNet Training..... 430
 - LeNet Model Assesment..... 432
 - Testing LeNet..... 434
- GPT and LLM Basics..... 436
 - A Brief Overview 437
 - LLM in the Wolfram Language..... 437
 - Chat Notebooks 438
 - Wolfram Prompt Repository 441
 - LLM Functionalities 444
 - GTP-1 and GPT-2 Models..... 446
- Final Remarks 448
- Index..... 451**

About the Author

Jalil Villalobos Alva is a Wolfram Language programmer and Mathematica user. He graduated with a degree in engineering physics from the Universidad Iberoamericana in Mexico City. His research background comprises quantum physics, bioinformatics, proteomics, and protein design. His academic interests cover the topics of quantum technology, bioinformatics, machine learning, artificial intelligence, stochastic processes, and space engineering. During his idle hours, he likes playing soccer, swimming, and listening to music.

About the Technical Reviewer



Andrew Yule is a co-founder and managing partner of Pontem Analytics, a global consulting company in the energy industry specializing in combining domain expertise with data-driven solutions. Andrew has over 13 years of professional experience leveraging the Wolfram Language and was the recipient of the Wolfram Innovator Award in 2017. He is an editor for the Society of Petroleum Engineers, The Way Ahead magazine, and is also currently a member of the Young Entrepreneurial Council. His technical background includes a bachelor's degree in chemical engineering from the Colorado School of Mines and a master's degree in data science from Southern Methodist University.

Acknowledgments

I want to thank the collective support and guidance received throughout the development of this project's second edition. The contributions of numerous past and present individuals have played an integral role in shaping this work. Their assistance, feedback, and mentorship have been invaluable, enriching the content and presentation of this edition. I also want to thank the technical and staff reviewers for their valuable comments and feedback during this manuscript. They both helped me improve the material's presentation and theoretical work. And finally, I would like to thank Las Des Nestor and "Los Betos" for teaching me great mastery.

Introduction

Welcome to *Beginning Mathematica and Wolfram for Data Science*.

Why is data science important nowadays? Data science is an active topic that is evolving daily; new methods, techniques, and data are created daily. Data science is an interdisciplinary field involving scientific methods, algorithms, and systematic procedures to extract data sets and thus better understand the data in its different structures. It is a continuation of some theoretical data analysis fields such as statistics, data mining, machine learning, and pattern analysis. With a unique objective, to extract quantitative and qualitative information of value from the data being recollected from various sources, and thus be able to objectively count an event for decision-making, product development, pattern detection, or identification of new business areas.

Data Science Roadmap

Data science carries out a series of processes to solve a problem, which includes data acquisition, data processing, model construction, communication of results, and data monitoring or model improvement. The first step is to formalize an objective in the investigation. From the object of the investigation, you can proceed to the data acquisition sources. This step focuses on finding the right data sources. The product of this path is usually raw data, which must be processed before it can be handled. Data processing includes transforming the data from a raw form to a state in which it can be reproduced to construct a mathematical model. Proceeding to the construction of the model, a stage that intends to obtain the information by making predictions in accordance with the conditions established in the early stages. Here, the appropriate techniques and tools, which consist of different disciplines, are used. The objective is to obtain a model that provides the best results. The next step is to present the outcome of the study. Which consists of reporting the results obtained and whether they are congruent with the established research objective. Finally, it comes to data monitoring, with the intention of keeping the data updated because data can change constantly and in different ways.

Data Science Techniques

Data science includes analysis techniques from different disciplines, such as mathematics, statistics, computer science, and numerical analysis. The following are some disciplines and techniques used.

- Statistics (linear, multiple regressions, least squares method, hypothesis testing, analysis of variance (ANOVA), cross-validation, resampling methods)
- Graph theory (network analysis, social network analysis)
- Artificial intelligence
- Machine learning
- Supervised learning (natural language processing, decision trees, naive bayes, nearest neighbors, support vector machine)
- Unsupervised learning (cluster analysis, anomaly detection, K-means cluster)
- Deep learning (artificial neural networks, deep neural networks)
- Stochastic processes (Monte Carlo methods, Markov chains, time series analysis, nonlinear models)

Even though many techniques exist, this list only shows a part of it since research on data science, machine learning, and artificial neural networks is constantly increasing.

Prerequisites

This book is intended for readers who want to learn about Mathematica / Wolfram Language and implement it in data science; it focuses on the basic principles of data science as well as for programmers outside of software development, that is, people who write code for their academic and research projects, including students, researchers, teachers, and many others. The general audience is not expected to be familiar with Wolfram Language or with the front-end program Mathematica, but little or any experience is welcome. Previous knowledge of the syntax would be an advantage in

understanding how the commands work in Mathematica. If this is not the case, the book provides the basic concepts of the Wolfram Language syntax. The fundamental structure of expressions in the Wolfram Language. Basic handling and understanding of Mathematica notebooks.

Prior knowledge or some experience with programming, mathematical concepts such as numbers, trigonometric functions, and basic statistics are useful, along with some understanding of mathematical modeling, which is also helpful but not compulsory.

Wolfram Language is different from many other languages but very intuitive and user-friendly to learn.

The book aims to teach the general structure of the Wolfram Language, data structures, objects, and rules for writing efficient code, and at the same time, teach data management techniques that allow them to solve problems in a simple and effective way. Provide the reader with the basic tools of the Wolfram Language, such as creating structured data, to support the construction of future practical projects.

For this new version, all the programming was carried out on a MacBook Air M1 with Sonoma 14 environment with the installation of version 13.3.1.0 and 14 of Wolfram Mathematica. Wolfram Mathematica is currently supported in other environments such as Linux, Windows, and macOS. The code found in the book works with both the Pro and Student versions.

Book Conventions

Throughout the book, you may come across different words written distinctly from others. Throughout the book, the words command, built-in functions, and functions may be used as synonyms that mean Wolfram Language commands written in Mathematica. So, a function will be written in the form of the real name; for example, RandomInteger.

The evaluation of expressions appears in the Mathematica In/Out format; the same applies to blocks of code.

```
In[#]:= "Hello World!"
```

```
Out[#]= "Hello World!"
```

The Layout

The book is written in a compact and focused way to cover the basic ideas behind the Wolfram Language and cover details on more complex topics. Some chapters have been revised and redesigned in this new version to focus on novice and advanced topics.

Chapter 1 discusses the starting topics of the Wolfram Language, basic syntax, and basic concepts with some example application areas, followed by an overview of the basic operations and debugging techniques, and concludes by discussing security measures within a Mathematica session.

Chapter 2 provides the key concepts and commands for data manipulation, sampling, types of objects, and some concepts of linear algebra—the introduction to lists, an important concept to understand in the Wolfram Language.

Chapter 3 discusses how to work properly with data and the initiation of the core structures for creating a dataset object, introducing concepts like associations and association rules are discussed with a conclusion remarking how associations and dataset constructions can be interpreted as a generalization of a hash table aiming to expose a better understanding of internal structures inside the Wolfram Language, including an overview of performing operations on a list and between lists and then discussing various techniques applied to dataset objects.

Chapter 4 exposes the main ideas behind importing and exporting data with examples throughout the chapter with common and newly added file formats. It also presents a very powerful command known as `SemanticImport`, which can import data elements that are natural language.

Chapter 5 covers the topic areas for new data visualization, common data plots, data colors, data markers, and how to customize a plot. Basic commands for 2D plots and 3D plots are presented, too.

Chapter 6 introduces the statistical data analysis. Starting with random data generation begins by introducing some standard statistical measures, followed by a discussion on creating statistical charts and performing an ordinary least square method.

Chapter 7 exposes the basis for data exploration and reviews a central discussion on the Wolfram Data Repository. Performing descriptive statistics and data visualization inside Fisher's Irises dataset objects is also covered.

Chapter 8 starts with machine learning concepts and techniques, such as gradient descent, linear regression, logistic regression, and cluster analysis, including examples from various datasets like the Boston and Titanic datasets and newly implemented features.

Chapter 9 introduces the key ideas and the basic theory to understand the construction of neural networks in the Wolfram Language, such as layers, containers, and graphs. The MXNet framework in the Wolfram Language scheme is also discussed.

Chapter 10 concludes the book by discussing training neural networks in the Wolfram Language. In addition, the Wolfram Neural Net Repository is discussed with an example application, examining how to access data inside Mathematica and the retrieval of information, such as credit risk modeling fraud detection, and concluding with the example of the LeNet neural network, reviewing the idea behind this neural network and exposing the main points on the architecture with the help of the MXNet graph operations and a final road map on the creation, evaluation, and deployment of predictive models with the Wolfram Language. In this new version, LLM (large language model) features are introduced with the connection to GPT services, use of chat cells, and presentation of the GPT-1 and GPT-2 models.

CHAPTER 1

Introduction to Mathematica

The chapter begins with a preliminary introduction to why Mathematica is a useful and practical tool. It explores the core concepts of the Wolfram Language and its syntax. It starts by explaining the internal structure of Mathematica and how to add code effectively. The concept of a notebook is introduced, which is important to understand the type of format that Mathematica handles. The chapter examines this interface class and demonstrates how notebooks simultaneously support code and text. In this way, a notebook is a computable text file. Next, you inspect various add-ons that can be employed within a notebook to help the user maximize their code's capabilities.

The next section demonstrates how to write expressions in Mathematica, examining topics such as arithmetic, algebra, symbols, global and local variables, built-in functions, date and time formats, plotting functions, logical operators, performance measures, delayed expressions, and accessing Wolfram Alpha. You then look at how Mathematica performs code computations, including its accepted varieties of inputs and the evaluation of these inputs. This chapter concludes with tips for seeking support within Mathematica, managing and handling errors, searching for solutions, and safely dealing with security concerns in notebooks that incorporate dynamic content.

Why Mathematica?

Mathematica is a mathematical software package created by Stephen Wolfram more than 35 years ago. Its first official version (Mathematica 1.0) emerged in 1988 and was created as an algebraic computational system capable of handling symbolic computations. However, Mathematica has established itself as a tool capable of performing complex tasks efficiently, automatically, and intuitively. Mathematica is

widely used in many disciplines like engineering, optics, physics, graph theory, financial engineering, game development, and software development.

Mathematica provides a complete, integrated platform to import, analyze, and visualize data. Mathematica does not require plug-ins. It also has a mixed syntax, performing both symbolic and numerical calculations. It provides an accessible way to read the code with the implementation of notebooks as a standard format, which also serves to create detailed reports of the processes carried out. Mathematica can be characterized as a powerful platform enabling efficient and concise forms of work. Among computer languages, the Wolfram Language falls into the group of programming languages classified as a high-level, multi-paradigm interpreted language. Unlike conventional programming languages, the Wolfram Language adheres to unique rules, facilitating order and clear, compact code composition.

The Wolfram Language

Mathematica is powered by the Wolfram Language, an interpreted high-level programming language that covers both symbolic and numeric capabilities. To understand the Wolfram Language, it is necessary to remember that the language's core nature resembles a normal mathematical text, as opposed to other programming languages' syntax. The following describes some remarkable features of the Wolfram Language.

- The first letter of a built-in function word is uppercase and is also human-readable.
- Any element introduced in the language is taken as an expression.
- Expressions take values consisting of the Wolfram Language atomic expressions.
 - A symbol made up of letters, numbers, or alphanumeric contents
 - Four types of numbers: integers, rational, real, and complex
 - The default character string is written within the quotation marks (“ ”)

- In Mathematica, there are three ways to group expressions.
 - Parentheses group terms within an expression $(\text{expr1} + \text{expr2}) + (\text{expr3})$.
 - Command entries are enclosed by brackets $[]$. Also, square brackets enclose the arguments of a built-in function, $F[x]$.
 - Mathematica uses curly braces $\{ \}$ (e.g., $\{a, b, c\}$) to represent lists, arrays, matrixes, and other collections.

Structure of Mathematica

Before entering code, you need to get the layout of Mathematica. To launch Mathematica, go to your Applications folder and select the Mathematica icon. This action brings up the new welcome screen, illustrated in Figure 1-1.

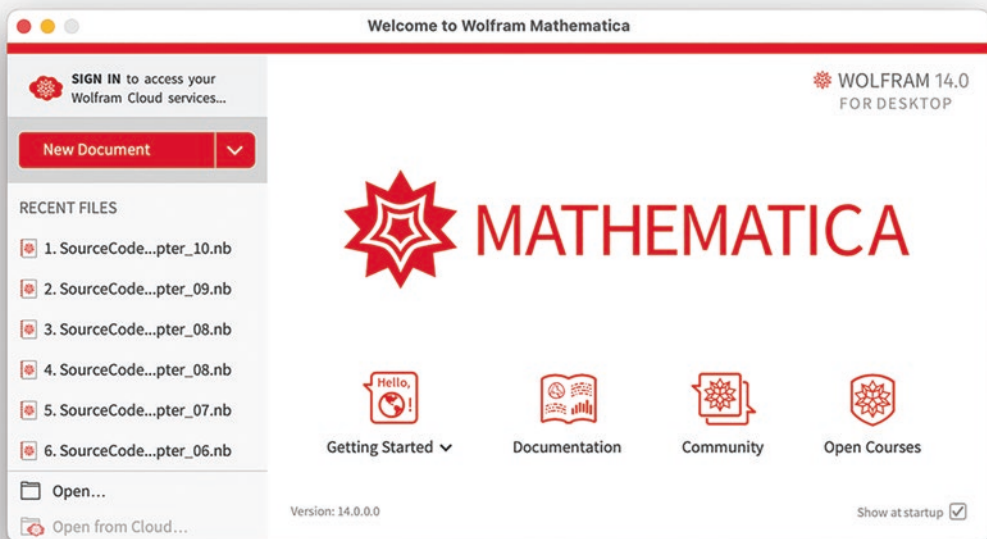


Figure 1-1. The default welcome screen for Mathematica's latest version

Tip The startup window offers valuable information for new and adept users, including the Mathematica version, access to documentation, resources, and the Wolfram community, among other things.

After the startup screen appears, you can create a new notebook by selecting the New Document button, and a blank page should appear like the one shown in Figure 1-2. New documents can also be created by selecting File ► New ► Notebook or with the ⌘+N (macOS) or Ctrl+N (Win) keyboard shortcut command.

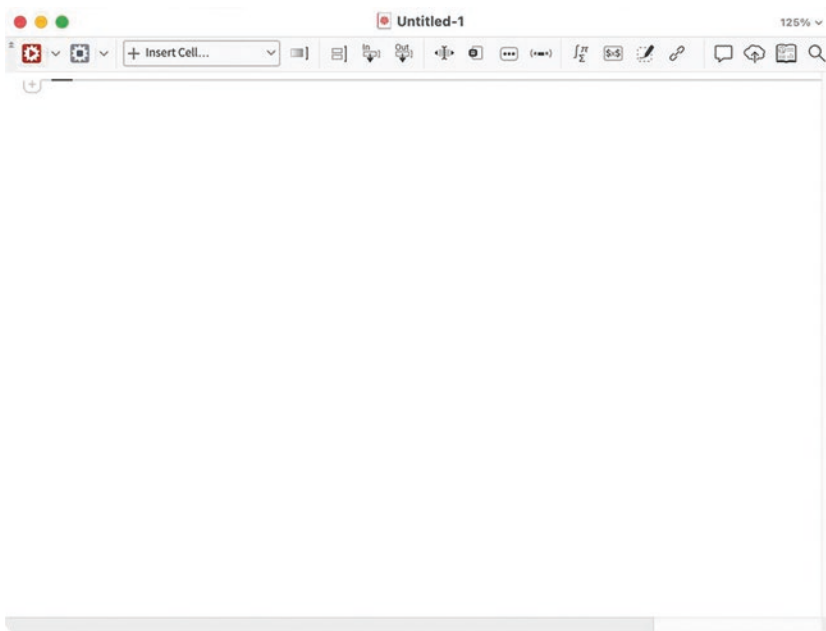


Figure 1-2. A blank notebook ready to receive input

The blank document that appears is called a notebook, and it's the core interaction between the user and Mathematica. Notebooks can be saved locally from the menu bar by selecting File ► Save (or Save as). Initializing Mathematica always exhibits an untitled notebook. Notebooks serve as the standard document format. They can be customized to display text alongside computations. However, the key feature of Mathematica lies in its capacity to perform computations, extending beyond numerical calculations, regardless of the notebook's purpose.

Note Mathematica version 13.1 introduced a new default assistant toolbar.

Mathematica's notebooks are separated into input spaces called cells. Cells are represented by the square brackets on the notebook's right side. Each input and output cell has its bracket. Brackets enclosed by larger brackets are related computations, whether input or output. Grouped cells are represented by nested brackets that contain the whole evaluation cell. Other cells can be grouped by selecting and grouping them with the right-click option. Cells can also have the capability to show or hide input by simply double-clicking the cells. To add a new cell, move the text cursor down, and a flat line should appear, marking the new cell ready to receive input expressions. The plus tab in the line is the assistant input tab, showing the various types of input supported by Mathematica. Figure 1-3 displays grouped input (In[-]) and output (Out[-]) cells.

```
In[ ]:= "Hello World" ] ]  
Out[ ]:= Hello World ] ]
```

Figure 1-3. Expression cells are grouped by input and output

There are four main input types. The default input is the Wolfram Language code input. Free-form input is involved with Wolfram knowledge-base servers, and the results are shown in Wolfram Language syntax. Wolfram Alpha query is associated with results explicitly shown on the Wolfram Alpha website. External Language Input is built-in support for common external programming supported by Mathematica.

There are four main input types.

- Default input: Wolfram Language code input
- Free-form input: involved with Wolfram knowledge-base servers and the results are shown in Wolfram Language syntax
- Wolfram Alpha query: associated with results explicitly shown on the Wolfram Alpha website
- External language input: built-in support for common external programming supported by Mathematica

These are illustrated in Figure 1-4.

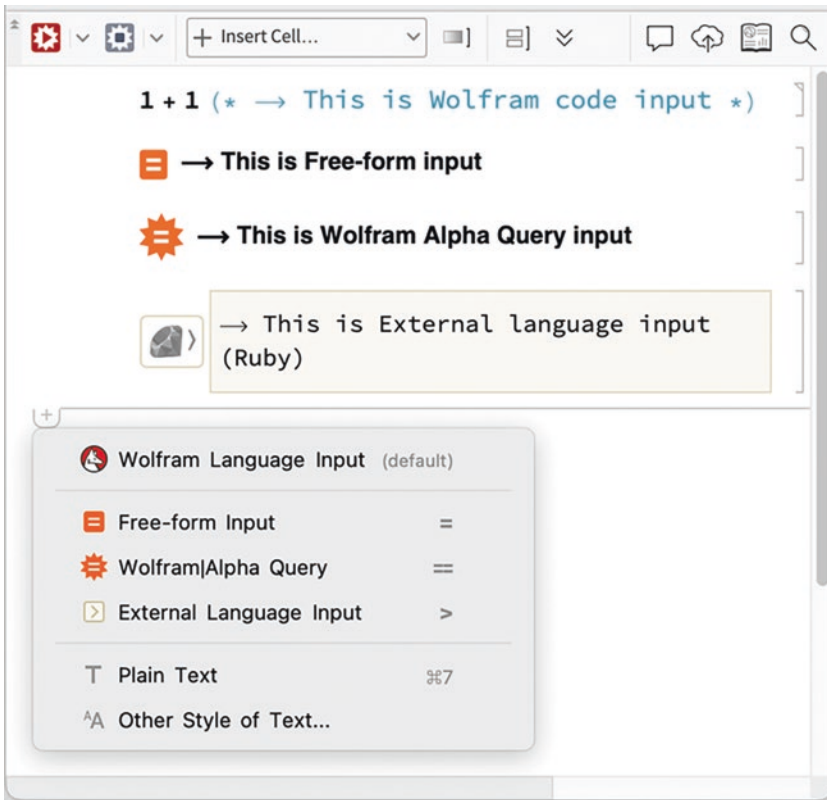


Figure 1-4. Main input types in Mathematica

Tip Keyboard shortcuts for front-end instruction commands are shown on the right or left side of each panel.

Design of Mathematica

Now that you have the lay of the land of Mathematica’s basic format, you can learn the internal structure of how Mathematica works. Inside Mathematica, there are two fundamental processes: the Mathematica kernel and the graphical interface. The Mathematica kernel is the one that takes care of performing the programming computations; it is where the Wolfram Language is interpreted and is associated with each Mathematica session. The Mathematica interface allows the user to interact with the Wolfram Language functions and, at the same time, document your progress.

Each notebook contains cells, where the commands that the Mathematica kernel receives are written and then evaluated. Each cell has an associated number. There are two types of cells: the Input cell and the Output cell. These are associated with each other and have the following expressions: In[n]:= Expression and Out [n]: = Result or (“new expr”). The evaluations are listed according to which cell is evaluated first and continue in ascending order. When quitting the kernel session, all the information, computations made, and stored variables are relinquished, and the kernel is restarted, including the cell expressions. To quit a kernel session, select Evaluation ► Quit Kernel ► Local.

Tip To start a new kernel session, click Evaluation ► Start Kernel ► Local.

To begin, try typing the following computation.

```
In[1] := (11*17) + (4/2)
Out[1] = 189
```

The computation shows that In and Out have a number enclosed. This number is the number associated with the evaluated expression.

A suggestion bar appears after every expression is evaluated (see Figure 1-5). The suggestion bar in Mathematica is always visible unless the user hides it. But the suggestion bar offers suggestions for possible new commands or functions to be applied to the generated output. The suggestion bar can sometimes be helpful if you are unsure what to code next; if used wisely, it might be helpful.

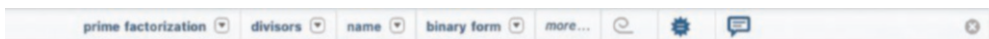


Figure 1-5. Suggestion bar for more possible evaluations

The input form of Mathematica is intuitive; to write in a Mathematica notebook, you just have to put the cursor in a blank space, and the cursor indicates that you are inside a cell that has not been evaluated. To evaluate a cell, click the keys [Shift + Enter], instructing Mathematica kernel to evaluate the expression written. The next chapter looks at the new form to evaluate expressions using the new toolbar.

To evaluate the whole notebook, go to the Evaluation tab on the toolbar and select Evaluate Notebook. If the execution of calculations takes more time than expected, you make a wrong execution of code, or if you want to seize a computation, Mathematica provides several ways to stop calculations. To abort a computation, go to

Evaluation ► Abort Evaluation. Alternatively, use the keyboard shortcut in Windows [Alt + .] or macOS [⌘ + .].

When a new notebook is created, the default settings are applied to every cell (input style). Nevertheless, preferences can be edited in Mathematica with various options. To access them, go to Edit ► Preferences. On macOS, the Preferences (settings) menu is located in the application menu, go to Mathematica ► Settings.

Once opened, a pop-up window appears (see Figure 1-6) with multiple tabs (Interface, Appearance, AI Settings, etc.). Basic customizations involve magnification, language settings, and other general instructions. The Appearance tab is related to code syntax color (i.e., symbols, strings, comments, errors, etc.). The AI Settings tab is the new tab associated with the LLM (large language model) evaluator. Other options belong to advanced settings that are not used in this book. Feel free to navigate each option.

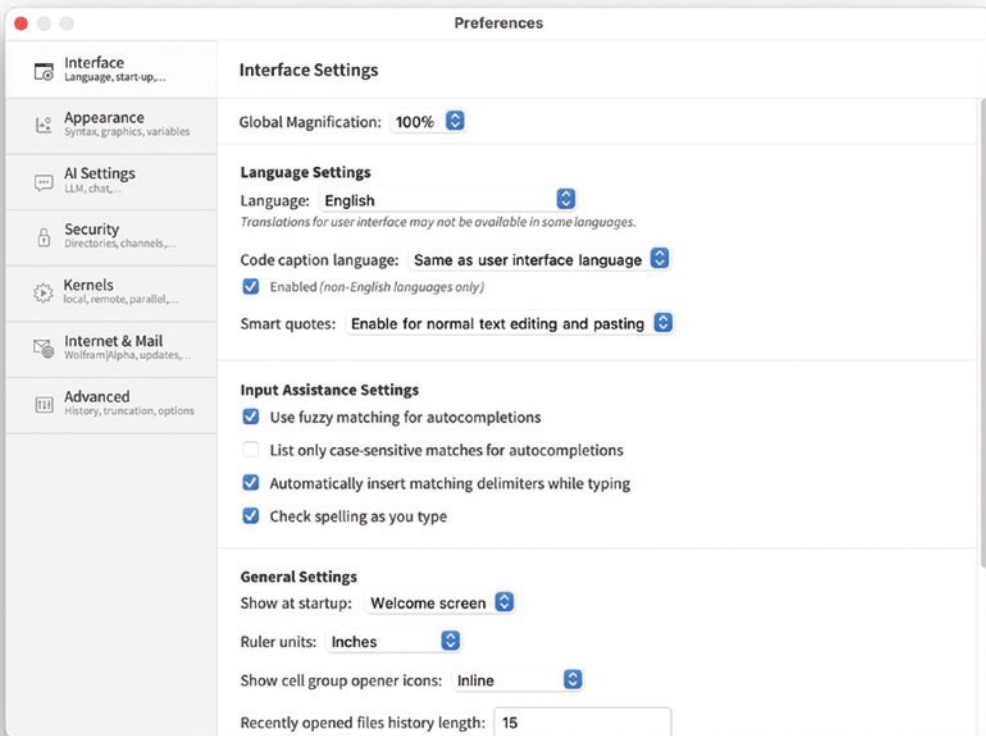


Figure 1-6. Preferences window

Later, you learn about in-depth settings and customization options for the notebook interface that allow you to tailor preferences.

Mathematica Environment

This section explores the user interface of Mathematica, with a focus on the notebook interface, as well as the other user experience functionalities.

Notebook Interface

Mathematica is always on the quest to improve user experience and boost productivity. In version 13.1, a big enhancement has been introduced—the seamless integration of a default toolbar (see Figure 1-7) across all standard notebook user interfaces (UIs).

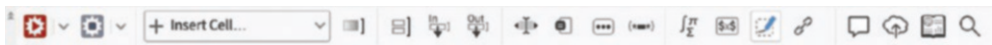


Figure 1-7. *The new UI default toolbar showcases essential tools and functionalities for efficient code development. Toolbar icons may vary by Mathematica version*

This new toolbar (described left to right) includes several new features to enhance user experience. Evaluate allows users basic and costume code evaluation. Abort lets users cancel queued cells and remove chosen ones; both options are shown in Figure 1-8. These features can also be accessed via the keyboard, as previously mentioned.

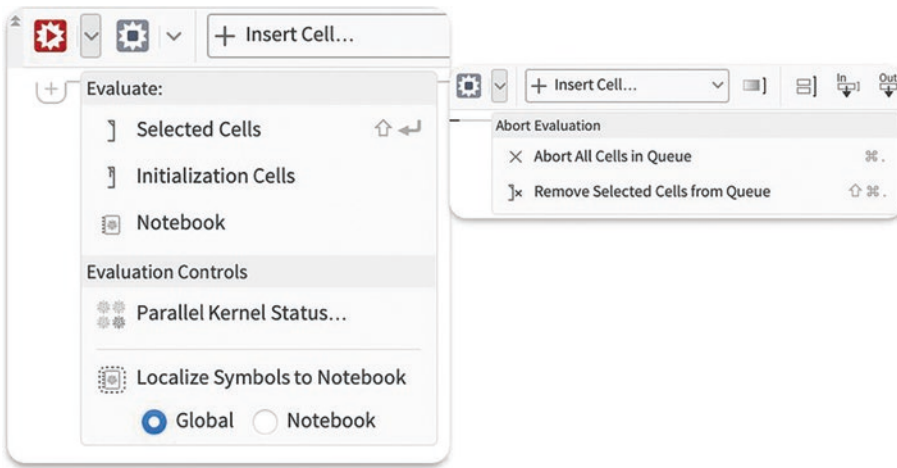


Figure 1-8. Extensive options for code evaluation and abort options in a notebook interface; the double arrow-like shape hides the toolbar

The other options integrate text cell formatting, offering styling options like cell style (title, subtitle, etc.) and cell color background. Users also benefit from the convenient cell management functions, such as grouping, dividing, merging cells, and inserting input/output of cells, all reduced to simple buttons. Continuing to options like extend selection, convert natural language into Wolfram Language code, collapse cells, insert comment, math form input, and LATEX rendering, users also have access to drawing canvas and hyperlink features. Finally, the rightmost section of the toolbar includes buttons for chat notebooks (utilizing LLM features), saving or publishing to the Wolfram Cloud, accessing documentation (local or web-based), and searching within the notebook.

Note Different buttons may appear based on the selected cell type where the cursor resides, ranging from text to code formatting. Figure 1-7 shows for Wolfram Language code input cell. Figure 1-9 shows for text display cell.

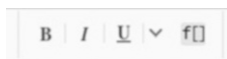


Figure 1-9. Text cell options for bold, italic, and underline and insert code text evaluation and abort options in a notebook interface