Financial Modeling Using R

(the second edition)

Preface

The first edition of this book was published in 2016. Since R is an evolving programming language, many functions and packages have changed over the past eight years. Below are some reasons I wrote this new version.

In this edition, I have made a few changes.

Firstly, I switched the order of learning R and applying it to finance. In the old version, the first part was related to using R in finance, while the second part was related to learning R. Now, in the first half, students will learn R and then apply it to finance in the second half.

Secondly, one chapter related to ChatGPT was added. This tool is being added to help students apply R to finance. This chapter shows how to offer different prompts to get answers from ChatGPT. Students will find that ChatGPT is an excellent tool that can help them in multiple ways. This chapter ends the book's first half since ChatGPT will help students with more complex R programs.

Thirdly, a new chapter called "GitHub and Git Bash" was added. The logic behind it is that students can search GitHub to find a good topic and related code for their term project after studying this book. In addition, they learn version control and how to collaborate with their fellow students. For this reason, GitHub is an excellent place for students to upload their projects, including a readme, an explanation, code, and other related materials. Git Bash is fantastic for downloading and uploading files to GitHub.

Fourthly, instructors and students can use this book or take a similar course in an R-assisted learning environment. Learners can access hundreds of data sets, over 1,000 programs written in R, five dozen websites, and 40 related videos. I will elaborate on this later in this preface.

I have rewritten many chapters to ensure a comprehensive and up-to-date learning experience. For instance, I have added two extra parts to Chapter 6: Open Data. The first part covers over 600 data sets from various R packages, while the second part includes several hundred CSV and R.Data data sets I have generated. This comprehensive approach aims to provide students and learners with a thorough understanding of the subject matter.

Finally, I omitted four chapters since they are less relevant to applying R to finance. I have also corrected many typos and mistakes.

Many thanks to those who adopted my R/Python books or offered helpful comments.

Ronald Crowe	Our Lady of Lake University
Frank P. D'Souza	Loyola University

Daniel Folkinshteyn	Rowan University
Shaobo Ji	Carlton University
Lawrence Kryzanowski	Concordia University
Mark Lennon	PennWest University
Premal Vora	Penn State University
Russ Wermers	University of Maryland
Tomasz Mlynowski	King's Business School
Shaojun Zhang	Hong Kong Polytechnic University
James Zheng	University of Massachusetts Lowell
Sheng Xiao	Westminster College

Why is R adopted as the computational tool?

Adopting R as the computational tool for financial modeling offers several advantages, making it a popular choice among finance professionals and researchers. Here are a few key benefits.

Open Source and Cost-Effective:

R is an open-source programming language that is freely available for anyone to use and modify. This makes it a cost-effective choice for individuals, academic institutions, and businesses, as it eliminates licensing fees associated with proprietary software.

Rich Statistical and Data Analysis Libraries:

R has a vast ecosystem of packages and libraries specifically designed for statistical analysis, econometrics, and data visualization. This extensive collection enables finance professionals to perform sophisticated modeling and analysis, including risk management, time series analysis, and portfolio optimization.

Community Support and Collaboration:

R has a vibrant and active community of users and developers. This community support facilitates knowledge-sharing, collaboration, and access to a wealth of resources, including forums, online tutorials, and contributed packages. Users can leverage this community to seek assistance, share best practices, and stay updated on the latest developments.

Integration with Financial Data Sources:

R integrates with various financial data sources, including APIs, databases, and direct data downloads. This makes it easy to retrieve and manipulate financial data for modeling purposes. Popular packages like `quantmod` and `tidyquant` provide convenient interfaces to access financial data.

Reproducibility and Documentation:

R promotes reproducibility in research and modeling. Scripts and analyses can be documented and shared transparently, allowing others to reproduce the results. This is particularly important in finance, where transparency and auditability are crucial for regulatory compliance and decision-making.

Extensive Visualization Capabilities:

R offers powerful data visualization capabilities through packages like `ggplot2`. These are essential for creating informative and visually appealing charts and graphs, which aid in interpreting and communicating financial models and results.

Advanced Modeling Techniques:

R supports advanced modeling techniques and machine learning algorithms, enabling finance professionals to implement sophisticated models for forecasting, risk assessment, and algorithmic trading.

Compatibility with Other Tools:

R can be easily integrated with other tools and languages, allowing for a flexible and comprehensive approach to financial modeling. For example, R Markdown facilitates the creation of dynamic reports that combine code, results, and narrative text, enhancing the communication of economic analyses.

By leveraging these advantages, R becomes a powerful and versatile financial modeling, analysis, and research tool. It offers efficiency, flexibility, and a rich set of capabilities to meet the diverse needs of the finance industry.

Starting from scratch

We assume students have no prior knowledge of R when using this book. However, if students have some basic knowledge related to finance, it will benefit them when applying R to finance. More specifically, if students take one or two finance courses, such as Corporate Finance and Portfolio Theory, they will feel more comfortable studying the book's second part.

How do learners use this book?

This book is perfect for a one-semester (15 weeks) course. The course can be divided evenly between learning R and applying it to finance.

In an R-assisted learning environment

They will use this book in a so-called R-assisted learning environment to help readers and students learn R and apply it to finance. After launching R, issue one of the following lines of code.

```
source("http://datayyy.com/fmr/week1.txt")
source("http://datayyy.com/fmr/week2.txt")
    …
source("http://datayyy.com/fmr/week14.txt")
source("http://datayyy.com/fmr/week15.txt")
```

For example, the following menu would appear after issuing the first line above.

```
*----------------------------------------------------*
* Financial Modeling using R (2nd edition)  2026 Yan  *
*----------------------------------------------------*
*  .c1  R Basics                                      *
*  .c2  R functions                                   *
*----------------------------------------------------*
* >.c2      # go to chapter 1 (a dot in front of c1) *
* >.uu      # go to utility submenu                   *
* >.fm      # back to this menu                       *
*----------------------------------------------------*
```

Figure 0-1 The menu for Week 1

For the first chapter, type `.chapter1`, or `.c1`, and we will get the following menu.

```
> .chapter1
function(i){
" i  Chapter 1: R basics
  -  -------------------------------------
  1  Download and install R
  2  Launch/quit R, one line for this course
  3  Comment line and comment paragraph
  4  3 ways to assign a value/values & how to show its value
  5  Use the up and down arrow keys to recall the previous commands
  6  R is case sensitive
  7  Normal operations: +, -, /, ^ (power)
  8  listing function ls()
  9  remove a variable or several variables
 10  remove all variables
 11  use meaningful variable names
 12  current working directory
 13  head() and tail() functions
 14  mean() for calculating mean and sd() for standard deviation
 15  put several commands on one line
 16  The simplest function
 17  use .nLetterFunctions() to show all n-letter functions
 18  use help() to find out more information about a specific function
 19  Videos
 20  Links

 Example #1:>.c1    # see the above list
 Example #2:>.c1(1) # see the first explanation
```

Figure 0-2 The contents of Chapter 1

Many helpful utility functions

The .uu section of the above menu is for utility functions. For the first week, we have the following menu. Over the semester, more utility functions will be added to the menu.

```
> .uu
function(){
"
 *------------------------------------------*
 * Utilities           -- short-cut --      *
 *------------------------------------------*
 * .allChapters        # .all               *
 * .calendar           # .cal               *
 * .getdata            #. gd                *
 * .inClassEx          # .ice               *
 * .macUsers           # .mac               *
 * .nLetterFunctions   # .nlf               *
 * .watchVideos        # .v2                *
 * .videos             # .v                 *
 *------------------------------------------*
 * >.ice               # see a list of ice  *
 * >.uu                # back to utilities  *
 * >.fm                # back to main menu  *
 *------------------------------------------*
```

Figure 0-3 The contents of the utility menu (.uu)

The first menu item from the above menu is .allChapter, abbreviated `.all`. After typing `.allChapters`, we will see the following menu.

```
> .allChapters
function(i=0){
"
 *-----------------------------------------------------------------------*
 * Financial Modeling using R (2nd ed.)              2026  by Dr. Yan     *
 *-----------------------------------------------------------------------*
 *          Learning R               |        Applying R to finance       *
 *-----------------------------------------------------------------------*
 *  .c1  R Basics                .c14  Finance Basics                     *
 *  .c2  Functions               .c15  Financial Statement Analysis       *
 *  .c3  Introduction to R packages .c16  Distributions, T-, F-tests, etc.  *
 *  .c4  Data Frame, list, and date .c17  Linear regressions, Sharpe ratio  *
 *  .c5  R loops and conditions  .c18  Portfolio Theory                   *
 *  .c6  Open data               .c19  VaR (Value at Risk)                *
 *  .c7  Data input              .c20  Options and Futures                *
 *  .c8  Simple data manipulations  .c21  Monte Carlo Simulations         *
 *  .c9  Data output                   --- Chapters below are online only *
 * .c10  Simple plots and graphs .c22  Liquidity measure/credit risk      *
 * .c11  Matrix manipulation     .c23  GitHub and Git Bash                *
 * .c12  Excel and R             .c24  ChatGPT: write and bug R code      *
 * .c13  String manipulation     .c25  Term projects                      *
 * ----------------------------------------------------------------------*
```

Figure 0-4 The names or titles of 25 chapters (`.allChapters`)

Our utility functions make learning Python and ChatGPT more efficient. We have 15 lists (see the last list, Week 15, below). To help readers and students use this book or take a related course, we have generated over 800 small programs written in Python. The above list is only for the first week. Over 15 weeks, the number of items on the list would vary. The following list is for the last week, Week 15.

```
> .uu
function(){
"
 *--------------------------------------------*
 * Utilities          -- short-cut --         *
 *--------------------------------------------*
 * .allChapters          # .all               *
 * .calendar             # .cal               *
 * .code                 # subdirectory       *
 * .dictionary           # .dict              *
 * .inClassEx            # .ice               *
 * .macUsers             # .mac               *
 * .niceFormula          # .nf                *
 * .nLetterFunctions     # .nlf               *
 * .searchDataInPackages # .sdip              *
 * .searchRpackages      # .srp               *
 * .termProjects         # .tp       .c25     *
 * .videos               # .v                 *
 * .watchVideos          # .v2                *
 *--------------------------------------------*
 * >.ice                 # see a list of ice  *
 * >.uu                  # back to utilities  *
 * >.fm                  # back to main menu  *
 *--------------------------------------------*
```
Figure 0-5 The hidden functions for the utility menu

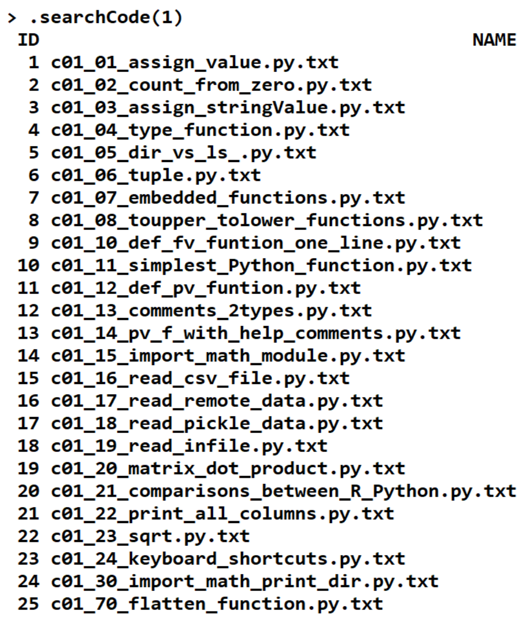
We developed a few utility functions to help students use our generated programs (see the next section for more details).

Code-related functions

We have designed three related utility functions, `.searchCode()`, `.showCode()`, and `.dictionary()`, to help users search, display, and download these programs.

```
> .code
function(){
"
 *---------------------------------------------------*
 *  Functions     Short-cut                          *
 *---------------------------------------------------*
 *  .searchCode    .sc   search code ID              *
 *  .showCode      .show [show on your screen]       *
 *  .dictionary    .dict                             *
 *---------------------------------------------------*
 * >.code          # back to this menu               *
 * >.algo          # back to the main menu           *
 *---------------------------------------------------*
```
Figure 0-6 The code menu

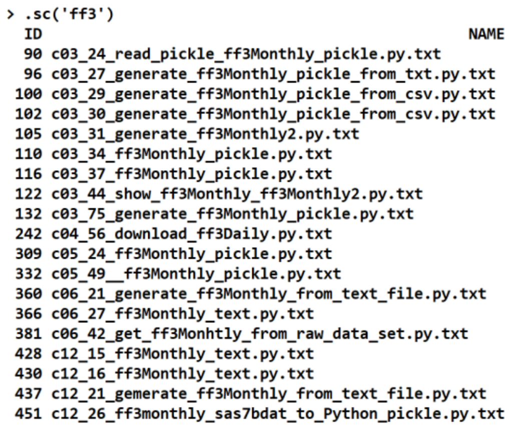
If we have 800 programs, their IDs will be from 1 to 800. Readers can use the `.searchCode()` function to find a program ID by entering a chapter number or a keyword.



```
> .searchCode(1)
 ID                                         NAME
  1 c01_01_assign_value.py.txt
  2 c01_02_count_from_zero.py.txt
  3 c01_03_assign_stringValue.py.txt
  4 c01_04_type_function.py.txt
  5 c01_05_dir_vs_ls_.py.txt
  6 c01_06_tuple.py.txt
  7 c01_07_embedded_functions.py.txt
  8 c01_08_toupper_tolower_functions.py.txt
  9 c01_10_def_fv_funtion_one_line.py.txt
 10 c01_11_simplest_Python_function.py.txt
 11 c01_12_def_pv_funtion.py.txt
 12 c01_13_comments_2types.py.txt
 13 c01_14_pv_f_with_help_comments.py.txt
 14 c01_15_import_math_module.py.txt
 15 c01_16_read_csv_file.py.txt
 16 c01_17_read_remote_data.py.txt
 17 c01_18_read_pickle_data.py.txt
 18 c01_19_read_infile.py.txt
 19 c01_20_matrix_dot_product.py.txt
 20 c01_21_comparisons_between_R_Python.py.txt
 21 c01_22_print_all_columns.py.txt
 22 c01_23_sqrt.py.txt
 23 c01_24_keyboard_shortcuts.py.txt
 24 c01_30_import_math_print_dir.py.txt
 25 c01_70_flatten_function.py.txt
```

Figure 0-7 The R programs related to Chapter 1

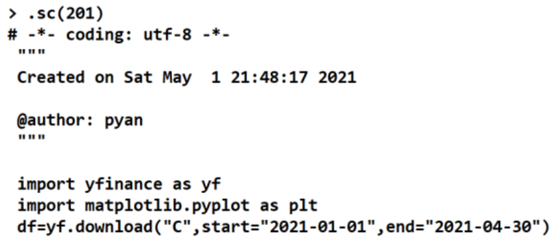
The second way to search is to use a keyword. For example, assuming we are interested in a Python program related to the Fama-French 3-factor model, we use the keyword "`ff3`". Note that `.sc()` is the shortcut function for the `.searchCode()` function.



```
> .sc('ff3')
  ID                                                NAME
  90 c03_24_read_pickle_ff3Monthly_pickle.py.txt
  96 c03_27_generate_ff3Monthly_pickle_from_txt.py.txt
 100 c03_29_generate_ff3Monthly_pickle_from_csv.py.txt
 102 c03_30_generate_ff3Monthly_pickle_from_csv.py.txt
 105 c03_31_generate_ff3Monthly2.py.txt
 110 c03_34_ff3Monthly_pickle.py.txt
 116 c03_37_ff3Monthly_pickle.py.txt
 122 c03_44_show_ff3Monthly_ff3Monthly2.py.txt
 132 c03_75_generate_ff3Monthly_pickle.py.txt
 242 c04_56_download_ff3Daily.py.txt
 309 c05_24_ff3Monthly_pickle.py.txt
 332 c05_49__ff3Monthly_pickle.py.txt
 360 c06_21_generate_ff3Monthly_from_text_file.py.txt
 366 c06_27_ff3Monthly_text.py.txt
 381 c06_42_get_ff3Monhtly_from_raw_data_set.py.txt
 428 c12_15_ff3Monthly_text.py.txt
 430 c12_16_ff3Monthly_text.py.txt
 437 c12_21_gemerate_ff3Monthly_from_text_file.py.txt
 451 c12_26_ff3monthly_sas7bdat_to_Python_pickle.py.txt
```

Figure 0-8 The programs related to the keyword "`ff3`"

Again, the function. searchCode () aims to get an ID for a function of interest. After getting an ID, we can use the `.showCode()`, `.showCodeTxt()`, or `.downloadCode()` functions. The difference between `.showCode()` and `.showCodeTxt()` is that programs will show on our screen for the former, and a web browser will be launched to open a text file for the latter.

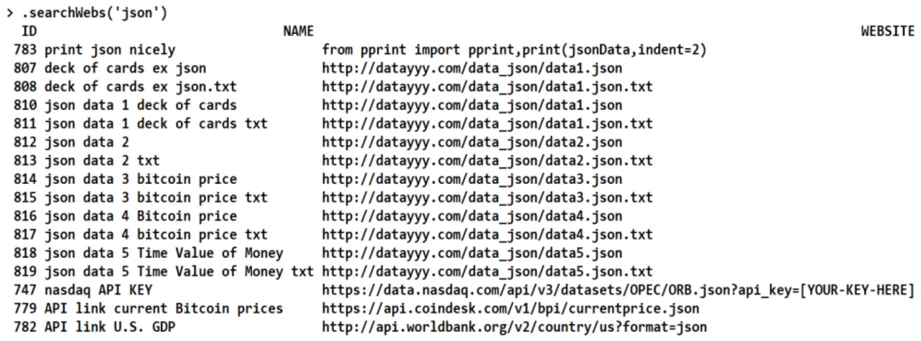


```
> .sc(201)
# -*- coding: utf-8 -*-
 """
 Created on Sat May  1 21:48:17 2021

 @author: pyan
 """

 import yfinance as yf
 import matplotlib.pyplot as plt
 df=yf.download("C",start="2021-01-01",end="2021-04-30")
```

Figure 0-9 The R program with 201 as its ID

The `.searchWebs()` function for convenience

For this book/course, we have generated several hundred small data sets in CSV, text, pickle (Python data), and RData (R data set). The `.searchWebs()` function makes accessing those data sets more efficient. With this function, students and instructors can access a given data set in seconds by writing a Python program. For example, we are interested in the data sets related to the Fama-French 3-factor model, shown below.



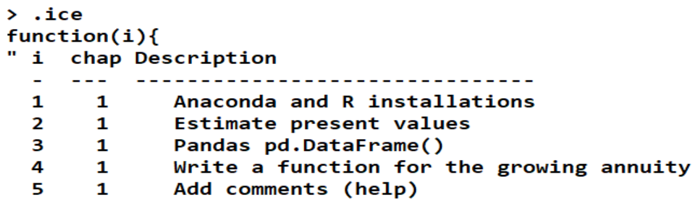
```
> .searchWebs('json')
  ID                              NAME                                                                     WEBSITE
 783 print json nicely                 from pprint import pprint,print(jsonData,indent=2)
 807 deck of cards ex json             http://datayyy.com/data_json/data1.json
 808 deck of cards ex json.txt         http://datayyy.com/data_json/data1.json.txt
 810 json data 1 deck of cards         http://datayyy.com/data_json/data1.json
 811 json data 1 deck of cards txt     http://datayyy.com/data_json/data1.json.txt
 812 json data 2                       http://datayyy.com/data_json/data2.json
 813 json data 2 txt                   http://datayyy.com/data_json/data2.json.txt
 814 json data 3 bitcoin price         http://datayyy.com/data_json/data3.json
 815 json data 3 bitcoin price txt     http://datayyy.com/data_json/data3.json.txt
 816 json data 4 Bitcoin price         http://datayyy.com/data_json/data4.json
 817 json data 4 bitcoin price txt     http://datayyy.com/data_json/data4.json.txt
 818 json data 5 Time Value of Money   http://datayyy.com/data_json/data5.json
 819 json data 5 Time Value of Money txt http://datayyy.com/data_json/data5.json.txt
 747 nasdaq API KEY                    https://data.nasdaq.com/api/v3/datasets/OPEC/ORB.json?api_key=[YOUR-KEY-HERE]
 779 API link current Bitcoin prices   https://api.coindesk.com/v1/bpi/currentprice.json
 782 API link U.S. GDP                 http://api.worldbank.org/v2/country/us?format=json
```

Figure 0-10 The result of websites with a keyword of 'json'

Using an ID allows us to launch a website quickly. For example, by issuing `.searchWebs(810)`, we will go to http://datayyy.com/data_json/data1.json.txt. Note that since we are constantly updating this dataset, the website associated with `810` might be different.

Many in-class exercises

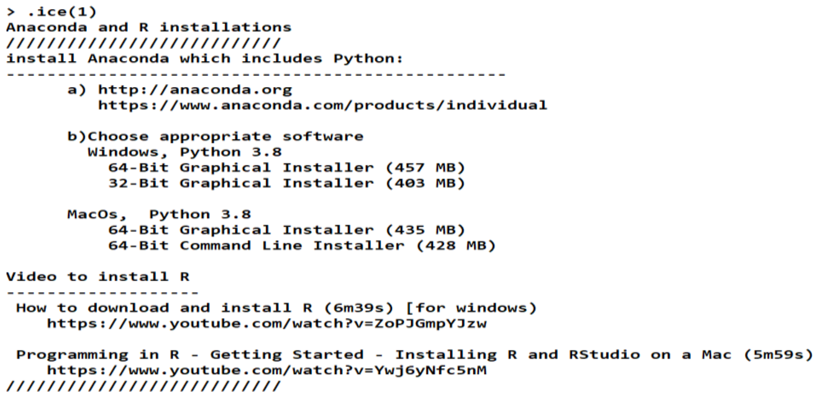
The best way to learn a programming language is via hands-on learning. For this reason, we have developed close to 100 in-class exercises. For each lecture, students will do at least two in-class exercises. To access a list of in-class exercises, type `.inClassEx` or `.ice`. Over a 15-week semester, the number of in-class exercises will increase from 4 to 60. For the first week, we have just 5, shown below.



```
> .ice
function(i){
" i  chap Description
  -  ---  -------------------------------
  1    1     Anaconda and R installations
  2    1     Estimate present values
  3    1     Pandas pd.DataFrame()
  4    1     Write a function for the growing annuity
  5    1     Add comments (help)

Example 1:>.ice        # show all the exercises
Example 2:>.ice(1)     # see the first one
```

Figure 0-11 The usage of the function for in-class exercises (`.ice`)

We have the first-in-class exercise to help students download and install Anaconda (Python) and R.

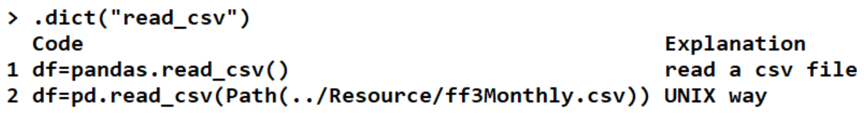


```
> .ice(1)
Anaconda and R installations
///////////////////////////
install Anaconda which includes Python:
-------------------------------------------------
      a) http://anaconda.org
         https://www.anaconda.com/products/individual

      b)Choose appropriate software
        Windows, Python 3.8
          64-Bit Graphical Installer (457 MB)
          32-Bit Graphical Installer (403 MB)

      MacOs,  Python 3.8
          64-Bit Graphical Installer (435 MB)
          64-Bit Command Line Installer (428 MB)

Video to install R
-------------------
 How to download and install R (6m39s) [for windows)
    https://www.youtube.com/watch?v=ZoPJGmpYJzw

 Programming in R - Getting Started - Installing R and RStudio on a Mac (5m59s)
    https://www.youtube.com/watch?v=Ywj6yNfc5nM
///////////////////////////
```

Figure 0-12 The content of the first in-class exercise

`.dictionary()` for searching commands

To help readers/students search for a corresponding Python function of a given R function or vice versa, we have designed a `.dictionary()` function (with an abbreviation of `.dict()`). Both students and instructors use this function intensively. For example, since R is taught before Python, we could enter '`ls`' to find the corresponding Python function, shown below.

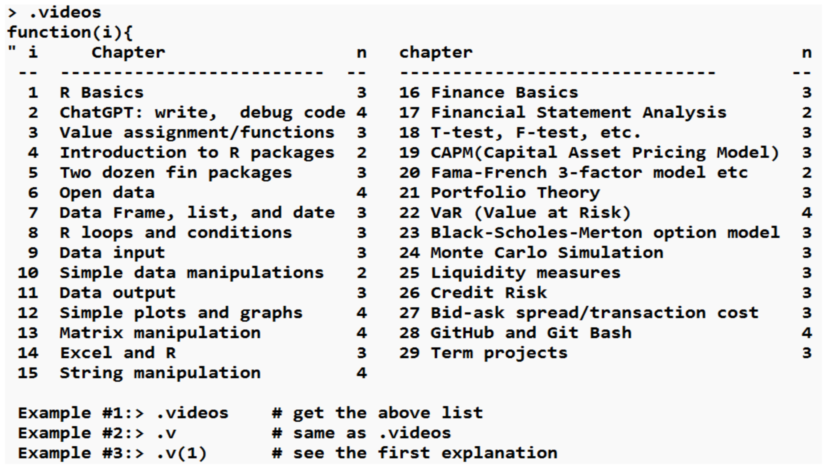


```
> .dict("read_csv")
  Code                                          Explanation
1 df=pandas.read_csv()                          read a csv file
2 df=pd.read_csv(Path(../Resource/ff3Monthly.csv)) UNIX way
```

Figure 0-13 The output from the dictionary with a keyword of "`read_csv`"

Good videos for new learners

There are many excellent YouTube videos on R and applying R to finance. Certain students learn better by watching these videos rather than sitting in a classroom listening to dry lectures. For programming, hands-on exercises are pretty important. On the other hand, after doing a few in-class exercises, many students prefer to view related videos. For those students, we have generated videos (see a list below).

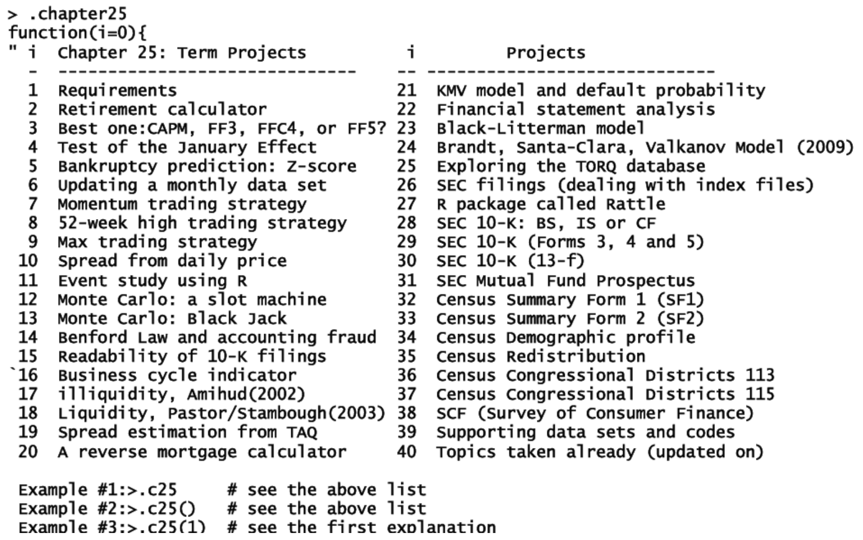


```
> .videos
function(i){
" i     Chapter                  n   chapter                           n
 --   ------------------------  --   -----------------------------    --
  1  R Basics                   3   16 Finance Basics                  3
  2  ChatGPT: write,  debug code 4   17 Financial Statement Analysis    2
  3  Value assignment/functions 3   18 T-test, F-test, etc.            3
  4  Introduction to R packages 2   19 CAPM(Capital Asset Pricing Model) 3
  5  Two dozen fin packages     3   20 Fama-French 3-factor model etc   2
  6  Open data                  4   21 Portfolio Theory                3
  7  Data Frame, list, and date 3   22 VaR (Value at Risk)             4
  8  R loops and conditions     3   23 Black-Scholes-Merton option model 3
  9  Data input                 3   24 Monte Carlo Simulation          3
 10  Simple data manipulations  2   25 Liquidity measures              3
 11  Data output                3   26 Credit Risk                     3
 12  Simple plots and graphs    4   27 Bid-ask spread/transaction cost 3
 13  Matrix manipulation        4   28 GitHub and Git Bash             4
 14  Excel and R                3   29 Term projects                   3
 15  String manipulation        4

 Example #1:> .videos    # get the above list
 Example #2:> .v         # same as .videos
 Example #3:> .v(1)      # see the first explanation
```

Figure 0-14 The help (usages) of the function showing videos

Term projects

It's a great idea to do a term project to test a learner's grasp of Python knowledge and skills. This part is treated as extra credit for our teaching since learning R and Python within one semester is already a heavy task. After typing .c28, the following list will pop up.



```
> .chapter25
function(i=0){
" i  Chapter 25: Term Projects          i          Projects
  -  ------------------------------    -- -----------------------------
  1  Requirements                      21  KMV model and default probability
  2  Retirement calculator             22  Financial statement analysis
  3  Best one:CAPM, FF3, FFC4, or FF5? 23  Black-Litterman model
  4  Test of the January Effect        24  Brandt, Santa-Clara, Valkanov Model (2009)
  5  Bankruptcy prediction: Z-score    25  Exploring the TORQ database
  6  Updating a monthly data set       26  SEC filings (dealing with index files)
  7  Momentum trading strategy         27  R package called Rattle
  8  52-week high trading strategy     28  SEC 10-K: BS, IS or CF
  9  Max trading strategy              29  SEC 10-K (Forms 3, 4 and 5)
 10  Spread from daily price           30  SEC 10-K (13-f)
 11  Event study using R               31  SEC Mutual Fund Prospectus
 12  Monte Carlo: a slot machine       32  Census Summary Form 1 (SF1)
 13  Monte Carlo: Black Jack           33  Census Summary Form 2 (SF2)
 14  Benford Law and accounting fraud  34  Census Demographic profile
 15  Readability of 10-K filings       35  Census Redistribution
`16  Business cycle indicator          36  Census Congressional Districts 113
 17  illiquidity, Amihud(2002)         37  Census Congressional Districts 115
 18  Liquidity, Pastor/Stambough(2003) 38  SCF (Survey of Consumer Finance)
 19  Spread estimation from TAQ        39  Supporting data sets and codes
 20  A reverse mortgage calculator     40  Topics taken already (updated on)

 Example #1:>.c25     # see the above list
 Example #2:>.c25()   # see the above list
 Example #3:>.c25(1)  # see the first explanation
```

Figure 0-15 The contents of Chapter 25 (Term projects)

What this book covers

Part I: Learning R

Chapter 1: R Basics teaches the basic concepts and code. First, students learn ways to install R and RStudio and some basic concepts, such as how to assign values to a new variable, whether R is case-sensitive, how to use embedded R functions, and how to find help.

Chapter 2: R Functions teaches students/learners how to write functions in R, such as its basic structure, three ways to input values, adding help to make our functions self-explanatory, and default values.

Chapter 3: Introduction to R Packages first discusses the importance of R packages. We will discuss finding relevant R packages and installing/updating individual packages.

Chapter 4: Data Frame, List, and Date introduces two important data types (data frame and list) and then explains how to generate a data frame, add column and row names, retrieve columns or rows, and merge data sets. Defining a true date variable is essential when working with time series since we can easily retrieve year, month, and date from such a variable and merge different time series based on date.

Chapter 5: R Loops and Conditions discusses various loops, such as `for` and `while` loops. We will also explain how to use multiple conditions to direct and redirect the flows of our programs.

Chapter 6: Open Data first discusses accessing over 600 data points embedded in various data sets, since this is the easiest way to access data. After that, we will explain Yahoo!Finance, FRED (Federal Reserve Bank's Data Library), Professor French's Data Library, Census Data, TORQ, and TAQ (both are high-frequency data), and over 500 data sets generated by me.

Chapter 7: Data input discusses various ways to input data, such as data sets embedded in multiple R packages and using functions `scan()`, `seq()`, and `c()` to input simple data sets from our keyboard. We discuss several functions, such as read.csv(), reading external relative big data sets, `table()`, `load()`, `readRDS()`, and `load(url())` to input CSV (Comma Separated Values), text, and R.Data.

Chapter 8: Simple Data Manipulations will discuss many basic techniques for data manipulation, such as selecting different columns and rows and merging data sets.

Chapter 9: Data output discusses various ways to input data, such as CSV (Comma Separated Values), text, pickle, and other types. We will focus on the functions write.csv() and write to achieve this goal.`table()`, `write()`, `save()`, `saveRDS()`, `sink()`, and `save.image()`.

Chapter 10: Simple Plots and Graphs discusses how to draw various graphs such as lines, curves, bar charts, histograms, moving GIFs, and interactive graphs.

Chapter 11: Matrix manipulation explains how to generate a matrix, find its dimensions, add and subtract two types of matrix multiplications, and use matrix manipulation to simplify our operations, such as return estimation and forming various portfolios.

Chapter 12: R and Excel discusses various ways to input data from Excel files since Excel is one of the corporations' most frequently used computational tools, big or small.

Chapter 13: Simple String Manipulations discusses various ways to manipulate string variables, such as combining two strings and choosing an upper- and lowercase substring. In addition, we quietly introduce several of the most frequently used regular expressions, such as replacing all the leading (trailing) parts before (after) a given substring.

Part II: Applying R to Finance

Chapter 14: Finance Basics explains fundamental concepts and functions related to finance, such as the Time Value of Money, functions for present value, future value, present value of a perpetuity, present value of an annuity, and the present value of a growing annuity. In addition, we will discuss several decision rules: NPV (Net Present Value), IRR (Internal Rate of Return), Payback period, and their associated decision rules.

Chapter 15: Financial Statement Analysis first explains the basic concepts. Then, we show how to download the balance sheet, income statement, and cash flow statement from Yahoo!Finance then manually processes them to estimate ratios such as Current Ratio, ROA (Return on Assets), and ROE (Return on Equity). As a difficult task, we will discuss how to process data called SEC Financial Statement Data Sets. Based on those data sets, students get estimated ratios for a given set of companies over 14 years.

Chapter 16: Distributions, T-test, and F-test starts with several vital distributions, such as normal, student-t, F—, and Chisq distributions. Then, we will discuss how to conduct various tests, one-sided and two-sided, and three decision rules, and check whether our results are statistically significant.

Chapter 17:Linear CLinear Models discusses a simple way to estimate the required rate of return for a given stock. Its logic is based on the one-factor linear model. The slope of the linear model (beta) represents the market risk of the underlying stock. The Fama-French 3-factor model is a simple extension of the CAPM model by adding two more factors: SMB (small company portfolio returns minus big company portfolio returns) and HML (returns of a High book-to-market ratio portfolio minus returns of a low book-to-market ratio portfolio). In addition, we will discuss the Fama-French-Carhart 4-factor model.

Chapter 18: Portfolio Theory starts from the most straightforward 2-stock portfolio: how to measure portfolio risk, what role correlation plays, and how to minimize portfolio risk. Then, we discuss optimizing an n-stock portfolio by choosing its weights to minimize specific risk-return measures such as the Sharpe Ratio, Treynor, or Sortino ratios.

Chapter 19: VaR (Value at Risk) first discusses the definition of VaR and then demonstrates two ways to measure it: a formula based on the normality assumption and sorting returns. In addition, normality is discussed to test whether stocks and market indices' returns follow a normal distribution.

Chapter 20: Options and Futures will explain the basic concepts first, such as definitions of call and put options, their payoff, corresponding profit/loss functions, and various hedging, speculative, and other trading strategies. We show how to use five lines to generate an R program for the famous Black-Scholes-Merton options model.

Chapter 21: Monte Carlo Simulations first shows how to generate random numbers from uniform and normal distributions. Then, the applications to finance are discussed in more detail. One implication is to price some exotic path-dependent options.

Part III: Online only chapters

Chapter 22: Liquidity Measures and Credit Risk Analysis discusses two liquidity measures: Armihud's illiquidity and Paster and Stambaugh's liquidity measure. For Credit Analysis, we first show several types of credit ratings, Standard Poor's, Moody's, and Pitch, then explain the pricing of corporate bonds based on the rating and yield curve.

Chapter 23: GitHub and Git Bash will first show how to search GitHub for useful topics, projects, and essential Python programs. Students learn to leverage others' existing projects to shorten their learning curve and make new learners more productive. We then show how to download it and several of the most frequently used commands using Git Bash, such as `git clone https://github.com/paulyxy/mortgage_calculator.git`.

Chapter 24: ChatGPT: Write, extend, and debug code will show how to use ChatGPT to write and debug our R programs. This will save us time and effort dramatically when applying R to finance. When using ChatGPT to help us apply R to finance, the leverage effect of understanding basic R concepts and functionalities will be at least 10-fold compared to no knowledge associated with R.

Chapter 25: Term Projects will discuss several types of projects students could do as a term project. We will present a few meaningful projects related to this book by searching GitHub. After learners/students understand the requirements, they can choose from a list of potential topics.