

Yan, Yuxing

Article

Teaching programming skills to finance students: how to design and teach a great course

Financial Innovation

Provided in Cooperation with:

Springer Nature

Suggested Citation: Yan, Yuxing (2017) : Teaching programming skills to finance students: how to design and teach a great course, Financial Innovation, ISSN 2199-4730, Springer, Heidelberg, Vol. 3, Iss. 32, pp. 1-14,
<https://doi.org/10.1186/s40854-017-0081-x>

This Version is available at:

<https://hdl.handle.net/10419/176471>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<http://creativecommons.org/licenses/by/4.0/>

METHODOLOGY

Open Access



Teaching programming skills to finance students: how to design and teach a great course

Yuxing Yan

Correspondence: yany@canisius.edu
Department of Economics and
Finance, Canisius College, 2001
Main Street, Buffalo, NY 14028, USA

Abstract

A motivated finance-major student should master at least one programming language. This is especially true for students from quantitative finance, business analytics, those attending a Master of Science in Finance or other financial engineering programs. Among the preferred languages, R holds one of the first places. This paper explains seven critical factors for designing and teaching a programming course: strong motivation, a good textbook, hands-on learning environment, being data intensive, a challenging term project, multiple supporting R datasets, and an easy way to upload such R datasets.

Keywords: Programming skills, Quantitative-finance, Financial engineering, R, Open-source finance, Data analytics

JEL: A2, I22, G00

Background

Nowadays, we are overwhelmed by large amounts of information (e.g., see Shi et al. (2017), Fang and Zhang (2016)), the catchphrase being “big data.” However, defining it is still controversial, since many explanations are available. Simply, college students should have one or two memory sticks, each with a capacity of 4 GB, and the simple answer is that if the student can generate and process 4 GB of data, he/she has the ability of dealing with “big data.”¹ Obviously, this is not an accurate definition. Nevertheless, it is a start and a practical one at that. In this context, the question is how could instructors arm their students with the ability of processing 4 GB of data. The answer is teaching them one programming language.

Any motivated finance-major student should master at least one programming language, and this holds especially true for students from quantitative-finance, business analytics, computational finance, data science, and those following a Master of Science in Finance or any other financial engineering programs. For example, recently, a job advertisement from the Department of Finance at the University of San Francisco specifically mentions MATLAB, R, SAS, and Python, as shown below:

The Department of Finance in the School of Management at the University of San Francisco is seeking to fill a position at the Assistant or Associate Professor level in

derivatives, fixed income, and mathematical finance with a subfield in an area of corporate finance and investments. Applicants should have mastery of Stata. Familiarity with MATLAB, R, SAS, and Python is desirable. The position is anticipated to begin in fall 2018.

This is not an isolated exception. Actually, 2017 is a turning point, since keywords such as “big data” and “data analytics” are prevalent in the job advertisements for assistant or associate professors in finance. Below is part of the job advertisement for an assistant or associate professor in finance by Wake Forest University (2017):

The Wake Forest University School of Business is seeking qualified candidates for a Tenure-Track Assistant (senior) or Associate Professor in Finance starting Fall 2018. This individual will be expected to teach introductory and advanced quantitative methods courses in Finance. Candidates should have a Ph.D. (or its equivalent) in Finance or a related field and should have a strong grasp of the roles and tools of big data and analytics in the practice of contemporary financial management.

Additionally, the title of a job advertisement by the University of California at Riverside (2017) is “Associate/Full professors in Finance/Business Analytics Cluster,” and that for an assistant professor in finance by the University of Central Florida (2017) specifically mentions that “*Preference will be given to candidates with strong skills in quantitative and empirical methods. A computer science background with demonstrated skills such as fluency in R, Python, and data visualization is highly desirable.*” Another advertisement, from the SUNY Polytechnic Institute New York (2017) looking for an assistant/associate professor in finance, states that “*Areas such as data analytics (‘big data’), financial analytics, financial innovation, regulatory policy, and other emergent and interdisciplinary finance areas are favored.*” A similar trend or turning point is also encountered in other subject areas in business schools, such as economics and marketing.

To answer which big data programming language to use, Pointer (2016) compares several languages. On his list, R and Python occupy the top two places, while Fig. 1 shows what programming/statistics languages researchers/practitioners use for analytics/data mining/data science work (for further details, see the survey results of KDnuggets Home (2014)).

This shows that R, SAS, and Python are top three. Since SAS is rather expensive, free software such as R and Python are extremely attractive. Figure 2 presented the results of another survey, conducted by CrowdFlower (2016): for the two categories of jobs that require programming and coding, or statistical tools, both Python and R occupy the first place.

Two other examples come from finance professors. In 2015, finance Prof. Sheng Xiao, from Westminster College, taught a course titled Financial Analytics and used the textbook “Python for Finance” by Yan (2014). In 2016, another finance professor, Premal Vora, at Penn State University, also adopted “Python for Finance” (Yan 2014) as a textbook. The textbook “Financial Modeling using R,” by Yan (2016), was adopted as a textbook by finance professors at Loyola University Maryland (2011–2013), Canisius College (2015–2016), University at Buffalo (2014–2017), and Rowan University (2016).

Since 2010, the author has spent a significant amount of time and effort applying R to finance. When teaching at Loyola University of Maryland, R was applied to financial

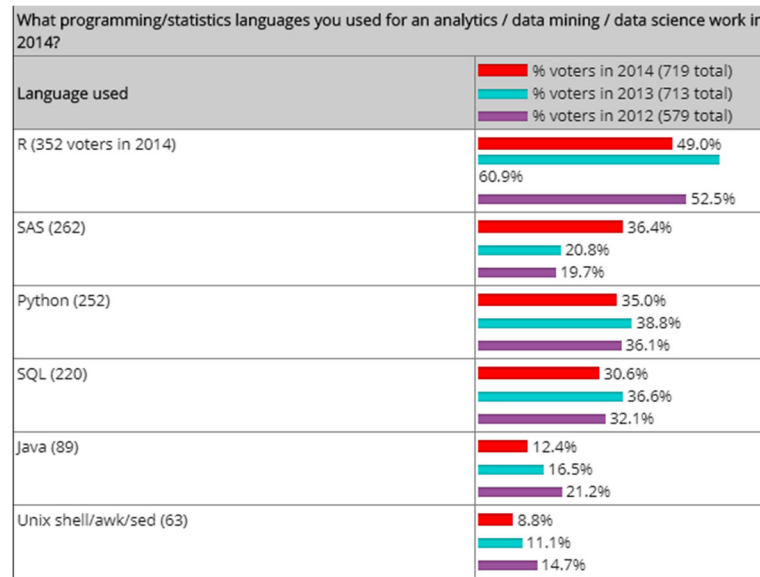


Fig. 1 What programming languages used

modeling for the first time, and numerous students adopted R as a computational tool. However, some students still preferred Excel. Here is a typical example: one student complained that Excel was not used and his answer was stunning when asked why he preferred Excel: “Because I am good at Excel.” As such, if a student is not motivated, then he/she has no incentive to learn R and apply it to finance.²

To help a potential instructor design and teach such a course, this brief paper summarizes the numerous determinants into seven unique and significant ones: strong motivation, good textbook, hands-on environment, being data intensive, challenging term project, making supporting R datasets available, and an easy way to upload those R datasets quickly. Some factors are easy to recognize, such as a good textbook. Moreover, strong motivation is vital for a finance-major student to learn a programming language, while interesting term projects could be an ideal indicator of an instructor teaching a course successfully (i.e., a performance measure). An example is presented in the following, where three factors related to data: being data intensive, making numerous datasets available, and uploading data efficiently, are shown to be closely related to the core of any course using R to process financial/economic and accounting data toward a specific set of goals. We thus elaborate each factor in more detail below.

Among jobs that require programming and coding:

77% require Python

40.6% require Java / JavaScript

25.1% require C / C++ / C# / Objective-C

Among jobs that require statistical tools:

59.6% require R

43.1% require SAS

25.5% require SPSS

Fig. 2 Languages and jobs

Strong motivation (factor #1)

Because of the current environment focusing on big data, business analytics, and data science, motivating a finance-major student to learn one programming language is not difficult. Obviously, the first lecture is the best time to motivate students, when comparing and contrasting different potential languages is an easy way to motivate them. There are several languages available, such as SAS, R Python, MATLAB, and C++. Since the language chosen is R in this example, it should be compared with other languages. Nowadays, SAS, R, and Python are top three languages in terms of superior ability to process data. Table 1 shows their scores based on cost, ease of learning, data handling and graphical capacities, tools advancement, job market preference, and customer service support and community (see Jain (2017) and Data science salary review (2015) for further details).

Most scores are reasonable such as cost (availability), with only a few problematic. For example, SAS is superior to R in terms of its data handling capacity, while the most critical disadvantage for both R and Python is their lack of support. This should however be tolerable when compared with expensive software, such as SAS and MATLAB. The second way to motivate students is to mention that many prestigious schools, such as New York University and Harvard, have also adopted R.³

For many new learners of a programming language, it is difficult to associate “motivation” with “intimidation.” For a finance-major student, even thinking about learning a programming language could be intimidating. Therefore, many one-line programs were generated. For example, to write a present value function, we need just one line of R code:

```
> pv_f<-function (fv, r, n) fv/(1+r)^n
```

Calling such as a function is trivial, shown below:

```
> pv_f(100,0.1, 1)
[1] 90.90909
> pv_f(100,0.08,2)
[1] 85.73388
>
```

A second one-line program is used to download historical daily price data from the author’s website.

```
> x<-read.csv("http://canisius.edu/~yany/data/ibm.csv")
```

Table 1 Comparisons between R, SAS and Python (5 being the best)

Language	SAS	R	Python
Availability	2	5	5
Ease of learning	4.5	2.5	3.5
Data handling capability	4	4	4
Graphical capability	3	4.5	4
Advancements in tool	4	4.5	4
Job opportunities	4.5	3.5	2.5
Customer services support and community	4	3.5	3

To view the first several lines, the `head()` function is used:

```
> head(x)
```

	Date	Open	High	Low	Close	Adj.Close	Volume
1	1962-01-01	7.71333	7.71333	7.00333	7.22667	2.077532	8760000
2	1962-02-01	7.30000	7.48000	7.09333	7.16000	2.058365	5737600
3	1962-03-01	7.18667	7.41333	7.07000	7.10333	2.042351	5344000
4	1962-04-01	7.10000	7.10000	6.00000	6.05333	1.740457	12851200
5	1962-05-01	6.05333	6.53000	4.73333	5.23333	1.504688	49307200
6	1962-06-01	5.21333	5.21333	4.00000	4.52333	1.300755	68451200

Additionally, for the last several lines, the `tail()` function could be applied:

```
> tail(x)
```

	Date	Open	High	Low	Close	Adj.Close	Volume
665	2017-05-01	160.05	160.42	149.79	152.63	149.5731	103329100
666	2017-06-01	152.80	157.20	150.80	153.83	152.2217	83977000
667	2017-07-01	153.58	156.03	143.64	144.67	143.1575	93293500
668	2017-08-01	145.00	145.67	139.13	143.03	141.5346	80268700
669	2017-09-01	142.98	147.42	141.64	145.08	145.0800	78296700
670	2017-10-01	145.35	148.95	145.21	146.54	146.5400	37011700

To load a monthly Fama-French dataset, one line R codes is again sufficient⁴:

```
> load("c:/temp/ffMonthly.RData")
```

For schools with a Center for Research in Security Prices (CRSP 2018) subscription, students could upload a monthly R dataset, with the codes below.⁵ The dataset contains trading data for 31,219 unique stocks (PERMNOs) from 1925 to 2015:

```
> load("c:/temp/stockMonthly.RData")
```

The logic is simple: after a new learner has learnt several one-line R programs, how could he/she not gain confidence? To motivate students, they are shown how to write an R program to randomly call their names and show their photos on the screen. After launching R, an R package called “png” is first loaded:

```
> library(png)
```

If we receive an error message, we then have to install the R package:

```
> install.packages('png')
```

After the package is installed, we type the following one-line R code:

```
> source("http://canisius.edu/~yany/randomCall.R")
```

The second moment to motivating students is when discussing the two chapters related to R packages.⁶ Anything taught before this moment provides students with basic concepts and skills, such as R being case-sensitive, writing simple programs, calling various pre-written R programs, running various loops and if- else- if conditions, and so on. This is similar to training new soldiers in basic skills: when they are good enough, the commander will lead them to a warehouse with all types of advanced weaponry, such as missiles and helicopters. Similarly, students become excited to find out about the many finance-related R packages at their disposal. As a result, they will be motivated.

The final opportunity to motivate students is when discussing potential topics for their term projects. The major purpose of a term project is to help students apply what they have learnt to a real-world, challenging task. After the mid-term, students are given about 40 potential topics. For example, “Does the January effect exist,” “Which party, Democratic or Republican, could manage the economy better,” and “What is the momentum trading strategy?” After finishing Chapter 8: T-test, F-test, Durbin-Watson, Normality, and Granger Causality Tests, students are shown how to test the January effect by randomly choosing one stock. However, for a good term project, students have to use all stocks. For this reason, for schools with (CRSP 2018) subscription, students are offered several hundreds of R datasets from (CRSP 2018) and Compustat.

Good textbook (factor #2)

A good R textbook for students majoring in finance should possess the following properties. First, it should apply R to finance. Second, it should review basic financial concepts, formulae, and the like. Third, it should use economics, financial, and accounting data, especially from public sources. As such, “Financial Modeling using R” by Yan (2016) was adopted. To encourage students to learn R, the book offers many one-line or few-line R programs. For example, below is the program to price a call option based on the famous Black-Scholes-Merton model:

```
bs_call<-function(s,x,T,r,sigma){
  d1 = (log(s/x)+(r+sigma*sigma/2.)*T)/(sigma*sqrt(T))
  d2 = d1-sigma*sqrt(T)
  s*pnorm(d1)-x*exp(-r*T)*pnorm(d2)
}
```

Calling the function is easy as well (see below). Using just five lines of R code for such a complex quant model definitively boosts a student's confidence dramatically.

```
> bs_call(40,42,0.5,0.1,0.2)
[1] 2.27778
```

Hands-on learning environment (factor #3)

Since hands-on is critical, all lectures should be conducted in a computer lab. The first few weeks are crucial. After giving the students a task by showing them code lines, an instructor should walk around to help individual students with their code. To prevent the students from copying and pasting from slides, the R codes shown

on PowerPoint slides are images. Coding should not be that difficult if students spend enough time on it.⁷ Again, overcoming intimidation is the first priority. Consequently, teaching speed should be controlled for the first several weeks. Additionally, since it is hands-on approach we are targeting, the ideal class size should be below 15.

In-class exercises also play an important role. During each lecture, students do at least one in-class exercise. Over the first several weeks, the exercises are rather simple: an instructor could discuss the tasks and type code, with the students simply typing those codes to complete the task. Gradually, students are given basic steps or a flow chart and asked to write their own codes. Since term-projects play a big role in this course, after the mid-term, most of the in-class exercises should be around subjects related to the term projects. For example, if students need to estimate equal- or value-weighted portfolio returns, an in-class exercise is designed to replicate S&P 500 monthly returns.

Another practice is to encourage students to generate one text file, which they constantly add new programs to over the entire semester. By the end of the semester, such a file should contain around 50 to 100 short programs. There are several advantages of doing this. First, students could use old programs for their homework without “reinventing the wheel.” Second, they save time during the mid-term and final. Finally, they could use many of their own programs in the future. The text format is also beneficial, in that anyone can open it without difficulty.

Being data intensive, especially when using public data (factor #4)

Nowadays, there are numerous sources of publicly available economic, accounting, and financial data. Table 2 shows a partial list:

Using publicly available data has several advantages: they are free and we can obtain recent data, such as yesterday's closing stock price, while this is not possible for expensive financial databases, such as (CRSP 2018). From the SEC (Securities and Exchange Commission) platform, students could download a public company's latest filings. It is also relatively easy to write an R program to access such public data. Finally, students from teaching schools could learn R effectively even if their schools have no subscription to expensive financial databases.

Challenging term-project (factor #5)

There are several objectives for a challenging term-project. First, it represents a great opportunity to summarize what students have learnt. Second, they could learn skills related to data processing, especially when dealing with dozens of stocks.⁸ Third, introducing various term topics widens a student's horizon, since many term projects came from seminal papers. For many topics, in addition to explaining their underlying logic, concrete steps are also described. Whether students have reached the same results as the original papers is not critical, but the process they follow is. The Table 3 below shows a few challenging term projects.

Finally, completing a good term project would benefit a student who is job hunting, as during a job interview, the student can talk about his/her term-project by explaining why the topic is interesting, what types of data he/she used, and how to retrieve and

Table 2 Open data sources^a

Name	Web page	Data types	Related topics
Yahoo Finance	http://finance.yahoo.com	Current and historical pricing, analyst forecast, options, balance sheet, income statement	CAPM, portfolio theory, liquidity measure, momentum strategy, VaR, options
Google Finance	http://www.google.com/finance	Current, historical trading prices	Stock trading data
Federal Reserve	http://www.federalreserve.gov/releases/h15/data.htm	interest rates, rates for AAA, AA rated bonds	fixed income, bond, term structure
Marketwatch	http://www.marketwatch.com	Financial statements	Corporate finance, investment
SEC filing	http://www.sec.gov/edgar.shtml	Balance sheet, income statement, holdings	Ratio analysis, fundamental analysis
Oanda	http://www.oanda.com	Foreign Exchange rates, price for precious metals	International finance, commodity trading
Prof. French data library	http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html	Fama-French factors, market index, risk-free rate, industry classification	Factor models, CAPM
Census Bureau	http://www.census.gov/ http://www.census.gov/compendia/statab/hist_stats.html	Census data	Real income, trading strategy
US. Dept. Treasury	http://www.treas.gov	US. Treasury yield	Fixed income
FINRA	http://cxa.marketwatch.com/finra/BondCenter/Default.aspx	Bond price and yield	Fixed income
Bureau of Labor Statistics	http://www.bls.gov/ http://download.bls.gov/	Inflation, Employment, unemployment, pay and benefits	Macro economics
Bureau of Economic Analysis	http://www.bea.gov/	GDP (Gross Domestic Product) and others .	Macro economics
National Bureau of Economic Research	http://www.nber.org/	Business cycles, vital statistics, report of Presidents	Macroeconomics, financial stability

^aTable 5.1 from Yan (2016). The author devotes an entire chapter to discussing many of those data sources

process data, as well as the conclusions reached.⁹ To successfully complete many challenging topics, instructor's strong support is critical, especially in terms of data. Oral presentations are also an integral part of term projects. Students have benefited greatly from other groups' presentations because of the variety of topics and their depth. For presentations, each group has 15 min for the presentation and five minutes of Q&A. Before the presentation, each student is given a "peer-evaluation" sheet. Based

Table 3 A partial list of potential topics for term projects

Using CRSP or TAQ	31	Estimate spread using CRSP daily data (Chung and Zhang 2014)
	32	Is liquidity factor priced? (Amihud 2002)
	33	What is the color of your firm, blue or red? (Yan 2014)
	34	Which model is the best, CAPM, FF3, FFC4, or FF5?
	35	Estimate spread, relative spread, expected spread etc. by using TAQ
	36	Process TAQ efficiently, how to process 30 year MTAQ data efficiently?
	37	Replicate momentum trading strategy (Jegadeesh and Titman 1993)
	38	Replicate industry momentum trading strategy (Moskowitz and Grinblatt 1999)
	39	Replicate 52-week high trading strategy (George and Huang 2004)
	40	Replicate max- trading strategy (Bali et al. 2011)
	41	Impact of business cycle on the above four trading strategies, (Yan and Zhang 2015)

on the feedback, this is a good practice to help students sharpen their presentation and critical thinking skills.¹⁰

Making several supporting R datasets available (factor #6)

Making several good R datasets available is a necessary condition for completing a challenging term project. There are two types of datasets: from public sources and from expensive financial databases, such as (CRSP 2018). In the US, most schools with a quantitative-finance program have subscribed to (CRSP 2018). Whenever possible, it is a good idea to use (CRSP 2018) to finish many challenging term projects. In 2015, the author first introduced (CRSP 2018) during the course “Financial Analysis with R” at University at Buffalo and, surprisingly, nine of the 10 term projects used (CRSP 2018). This further supports the use of (CRSP 2018) for teaching.

For the first few weeks, students learn how to generate R datasets for small amounts of data, such as the Fama-French three-factor, Fama-French-Carhart four-factor, and Fama-French five-factor time series. Students thus benefit from learning how to generate those datasets themselves. However, for a challenging term project, such as replicating the so-called MAX trading strategy suggested by Bali et al. (2011), it is not feasible to ask students to generate (CRSP 2018)-related R datasets themselves. To help students complete such challenging term projects, over 400 R datasets were generated, with some examples shown in Table 4.

Table 4 A partial list of R data sets related to term projects

* List of R data sets for Fama-French *		

* ffMonthly	# Fama-Fench monthly 3 factors	*
* ffcMonthly	# Fama-French-Carhart 4 factors	*
* ffMonthly5	# Fama-French-Carhart 5 factors	*
* ffDaily	# Fama-French daily 3 factors	*
* ffcDaily	# Fama-French-Carhart daily 3 factors	*
* ffDaily5	# Fama-French 5 daily factors	*

* List of R data sets for CRSP *		

* crspInfo	# CRSP header file	*
* tradingDaysM	# trading days for monthly data	*
* tradingDaysD	# trading days for daily data	*
* stockMonthly	# monthly stock data	*
* indexMonthly	# index monthly data	*
* sp500add	# sp500 constituents	*
* sp500monthly	# monthly S&P500 returns	*

Easy and quick uploading of R datasets (factor #7)

Two words could be used to summarize this approach: *trivial* and *second*. The former means that uploading various R datasets is trivial. It also means that an instructor does not have to explain how to upload various datasets, since he/she could just provide the students one line of R code. For the latter, uploading each dataset should take just a few seconds. Since (CRSP 2018), Compustat, and TAQ are proprietary databases, while the Fama-French data is free, Fama-French datasets will be used as an illustration. Readers will observe how easy it is to load any dataset by simply typing the following one-line R code:

```
>source("http://canisius.edu/~yany/ff.R")
```

After hitting the return key, we will see the instructions below:

```
*-----*
* Fama-French data   7/11/2016                               *
*-----*
* Functions          Utilities                                *
*-----*
* pv_f              loadFF                                   *
*                  severalRcommands                         *
*-----*
* >pv_f             # find its usage                         *
* >ff               # back to this menu                     *
*-----*
```

To determine how to load various Fama-French datasets, we just type the function name, *loadFF*, with the following output:

```
Objective: upload R data sets related to Fama-French data
n : an integer

  n  R data set  Description
-----
  1  ff3Monthly  monthly Fama-French      3 factor
  2  ff4Monthly  monthly Fama-French-Carhart 4 factor
  3  ff5Monthly  monthly Fama-French      5 factor
  4  ff3Daily    daily   Fama-French      3 factor
  5  ff4Daily    daily   Fama-French-Carhart 4 factor
  6  ff5Daily    daily   Fama-French      5 factor

Example #1:>loadFF      # show how to use this function

Example #2:> loadFF(1)
Note: Loaded data set is called ff3Monthly
> head(ff3Monthly,3)
      DATE MKT RF      SMB      HML      RF
1 1926-07-01 0.0296 -0.0230 -0.0287 0.0022
2 1926-08-01 0.0264 -0.0140  0.0419 0.0025
3 1926-09-01 0.0036 -0.0132  0.0001 0.0023
> tail(ff3Monthly,2)
      DATE MKT RF      SMB      HML      RF
1078 2016-04-01 0.0091  0.0084  0.0312 0.0001
1079 2016-05-01 0.0178 -0.0028 -0.0185 0.0001

Example #3:> loadFF(3)
Note: Loaded data set is called ff5Monthly
";loadFF_(n)
}
```

This simple method is also valid for other datasets (or databases). Assume that we have 200 datasets from (CRSP 2018). During a lecture, the students can easily upload those datasets by typing the one-line R code below. Since (CRSP 2018) is a proprietary database, the datasets could not be made public, so the following image is just an illustration:

```

function(n){
  "Objective: upload one R data set from CRSP
  n : an integer
  -----
  n   R data set   Description
  ---
  1   crspInfo     CRSP information data
  2   tradingDaysM trading days for monthly data
  3   tradingDaysD trading days for daily data
  4   stockMonthly CRSP monthly data
  5   sp500Monthly S&P500 monthly data
  6   sp500Daily   S&P500 daily data
  7   sp500add     stocks added/deleted from the index
  8   indexMonthly index monthly data
  9   indexDaily   index daily data
  1925 d_          daily data for 1925
  1926 d_          daily data for 1926
  ....
  2013 d_          daily data for 2013
  2014 d_          daily data for 2014
  19252 d_         daily data for 1925 (more variables)
  19262 d_         daily data for 1926
  ....
  20132 d_         daily data for 2013
  20142 d_         daily data for 2014

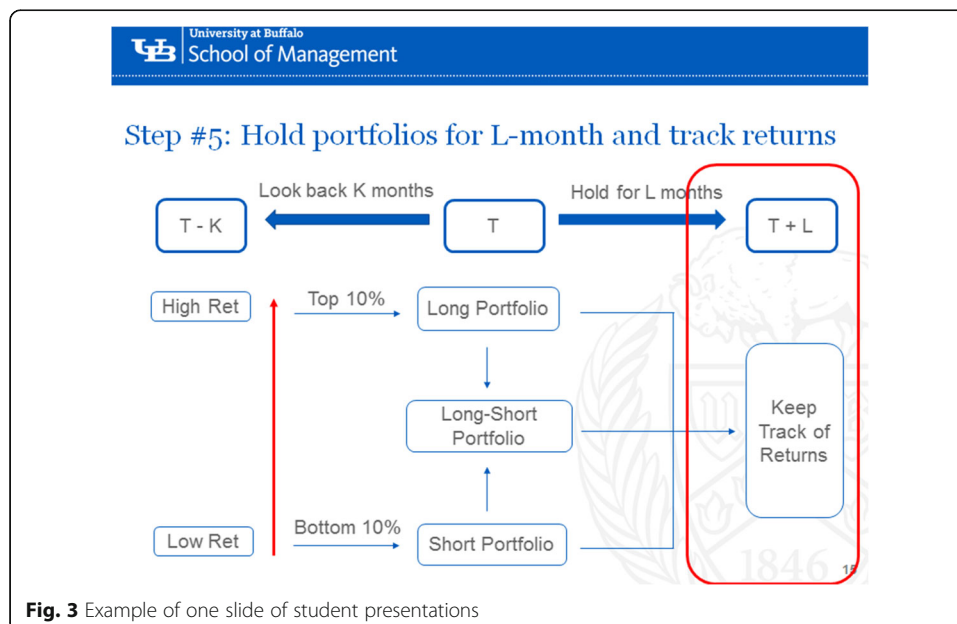
  Example #1: show how to use this function
  >loadCRSP

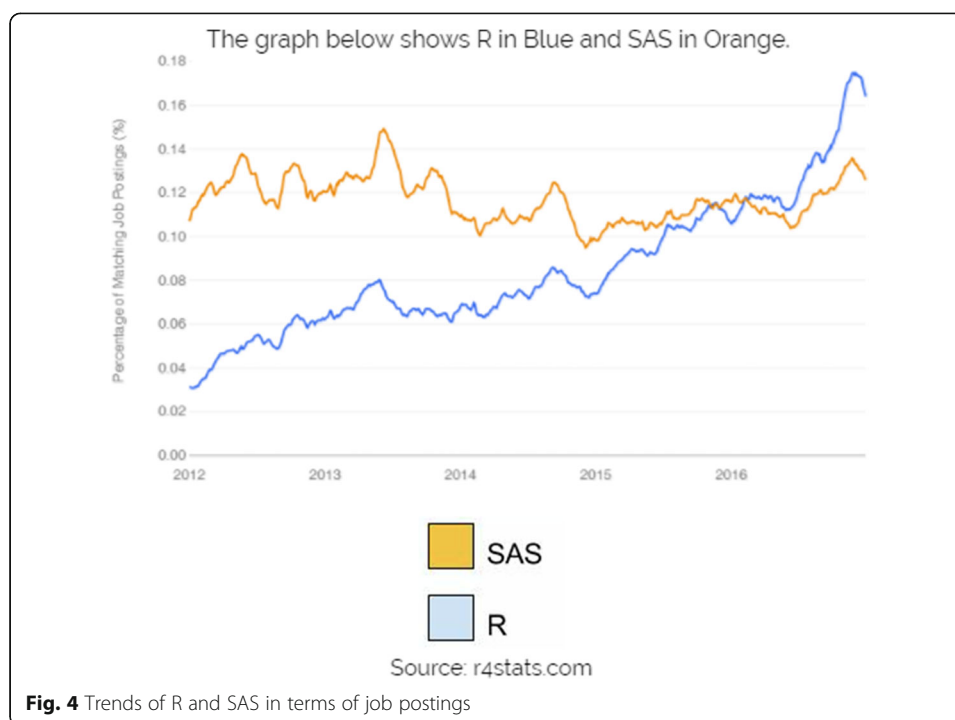
  Example #2: load crspInfo.RData
  >setwd('c:/temp')
  >loadCRSP(1)
  > head(crspInfo,2)
  PERMNO PERMCO CUSIP FIRMNAME TICKER EXCHANGE BEGDATE ENDDATE
  1 10000 7952 68391610 OPTIMUM MANUFACTURING INC OMFGA 3 1986-01-31 1987-06-30
  2 10001 7953 36720410 GAS NATURAL INC EGAS 2 1986-01-31 2014-12-31
  > tail(crspInfo,2)
  PERMNO PERMCO CUSIP FIRMNAME TICKER EXCHANGE BEGDATE ENDDATE
  30657 93435 53452 82936620 SINO CLEAN ENERGY INC SCEI 3 2010-06-30 2012-05-31
  30658 93436 53453 88160R10 TESLA MOTORS INC TSLA 3 2010-06-30 2014-12-31
  "loadCRSP_(n)
}

```

In terms of the best outcome from taking this course, we use a term project by three students who replicate Jegadeesh and Titman (1993) momentum strategy after taking this course (one of their presentation slides is shown in Fig. 3). They use R to process (CRSP 2018) monthly data to generate the same results as Jegadeesh and Titman (1993). After submitting their report, PowerPoint presentation, R code, and output results, they made a presentation in front of the class.

Regarding the job market, Jain (2017) affirms that “Globally, SAS is still the market leader in available corporate jobs. Most of the big organizations still work on SAS. R/Python, on the other hand are better options for start-ups and





companies looking for cost efficiency. Also, number of jobs on R/Python have been reported to increase over last few years. Here is a trend widely published on internet, which shows the trend for R and SAS jobs. Python jobs for data analysis will have similar or higher trend as R jobs (Fig. 4):”

Conclusions

This brief paper shows potential finance instructors how to design a course for teaching finance-major students a programming language such as R and apply it to finance. There are seven important determinants: 1) strong motivation, 2) a good textbook, 3) hands-on learning environment, 4) being data intensive, 5) challenging term project plus oral presentation, 6) making available several supporting R datasets for term projects, and 7) an easy and quick way to upload those R datasets. For a finance major student, programming skills will become a necessity in the foreseeable future. Moreover, once a user masters one computer language, it is much easier/quicker for him/her to learn a second one. This holds true for other languages such as SAS, MATLAB, or Python. As such, instructors from various finance departments might find this short paper helpful. The syllabi, slides, and various R datasets are available upon request. However, since the sample size is quite small, there might exist systematic bias.

Endnotes

¹Some researchers find that each company has, on average, about 30 T of data. Therefore, another way to define “big data” is the ability to store and process data at such a scale.

²Eventually, the student withdrew from the course.

³See the links at http://www.nyu.edu/projects/politicsdatalab/learning_students.html, <http://online-learning.harvard.edu/course/data-analysis-life-sciences-1-statistics-and-r-0>

⁴The R dataset could be downloaded from <http://canisius.edu/~yany/RData/ffMonthly.RData>.

⁵The CRSP database contains trading data for all listed stocks in US from 1926 onwards.

⁶Chapter 31: Introduction to R packages and Chapter 16: Two dozen R packages related to Finance, from Yan (2016).

⁷Usually, students are required to spend at least one hour per day, including weekends, outside the classroom working on R.

⁸When students use the CRSP for R datasets, they are using over 30,000 stocks for their term projects.

⁹Actually, this indeed happened. One of my students told me that during his job interview with Goldman Sachs, he was talking about this course plus one of my working paper related to PIN (Probability of Informed Trading). I was extremely thrilled after hearing that.

¹⁰An example of peer feedback can be found at <http://canisius.edu/~yany/doc/peerFeedback2016.txt>

Acknowledgements

I thank Karyl Leggio, Qiyu Zhang, Lisa Fairchild and Kee Chung for their helps when I was teaching at Loyola University and University at Buffalo. I thank James Yan for his editorial efforts.

Authors' contributions

I have contributed 100%.

Competing interests

The author declares that he has no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 11 March 2016 Accepted: 7 November 2017

Published online: 08 December 2017

References

- Amihud, Yakov, 2002, Illiquidity and Stock returns, *Journal of Financial Markets* 5, 31–56.
- Bali, Turan G., Nusret Cakici, and Robert F. Whitelaw, 2011, Masing Out: Stocks as Lotteries and the Cross-Section of Expected Returns, *J Financ Econ* 99 427–446
- Chung, Kee H. and Hao Zhang, 2014, A Simple Approximation of Intraday Spreads Using Daily Data, *Journal of Financial Markets* 17, 94–120.
- CrowdFlower, 2016, Data Scientist Report, http://visit.crowdfower.com/rs/416-ZBE-142/images/Crowdfower_Data_Scientist_Survey2015.pdf
- CRSP 2018, <http://crsp.com/>
- Data Science salary review, 2015 <https://www.analyticsvidhya.com/blog/2017/09/sas-vs-vs-python-tool-learn/>
- Fang, Bing and Peng Zhang, Big Data in Finance, in *Big Data Concepts, Theories, and Applications*, S. Yu and S. Guo, Eds., ed Cham: Springer, 2016, pp. 391–412
- Jain, Kunal, 2017, SAS vs. R (vs. Python) – which tool should I learn?, <https://www.analyticsvidhya.com/blog/2017/09/sas-vs-vs-python-tool-learn/>
- Jegadeesh N, Titman S (1993) Returns to buying winners and selling losers: implications for stock market efficiency. *J Financ* 48:65–91
- KDnuggets Home, Polls, 2014, Languages for analytics/data mining (2014), <https://www.kdnuggets.com/polls/2014/languages-analytics-data-mining-data-science.html>
- Moskowitz, Tobias, and Mark Grinblatt, 1999, Do industries explain momentum? *Journal of Finance* 54, 2017–2069.
- Pointer, Ian, 2016, Which freaking big data programming language should I use? <http://www.infoworld.com/article/3049672/application-development/which-freaking-big-data-programming-language-should-i-use.html>
- Shi, Xiang, Peng Zhang and Samee U. Khan, 2017, Quantitative Data Analysis in Finance, in *Handbook of Big Data Technologies*, A. Y. Zomaya and S. Sakr, Eds., Springer, 2017
- SUNY Polytechnic Institute in New York job ad for Assistant/Associate Professor of Finance (Tenure Track Position), (2017) <https://chroniclevitae.com/jobs/0000370015-01>
- University of California at Riverside job ad for associate or full professor in finance/ business analytics cluster (2017), <https://aprecruit.ucr.edu/apply/JPF00833>
- University of Central Florida job ad for assistant professor (2017), <https://www.jobswithucf.com/postings/51058>

Wake Forest University job ad for assistant/associate professor in finance (2017), <http://careers.afajof.org/job/305049/tenure-track-assistant-or-associate-professor-in-finance/>

Yan, Yuxing, 2014, Python for finance, 1st edition, Packt Publishing

Yan Y (2016) Financial Modeling using R. Publishing, Tate ISBN: 978-1-68187-530-9

Yan, Yuxing and Shaojun Zhang, 2016, Business cycle, investors' preferences and trading strategies, *Frontiers of Business Research in China*, Vol. 10, Issue (4) : 525–547.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
