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

Yuxing Yan[1]
1/5/2016

Abstract

It is always my firm belief that an ambitious finance-major student should master at least one computer language. This is especially true for students from quantitative-finance, MSF, business analytics or financial engineering programs. Among many good languages, R and Python are the two best ones. Based on my 6-year experience of teaching R to finance students at 3 schools, the following 7 factors are critical for designing and teaching such a course: strong motivation, a good textbook, a hands-on environment, data intensive, a challenging term project, tons of supporting R data sets and an easy way to upload those R data sets.

JEL: A2, I22, G00

Keywords: programming kills, quantitative-finance, financial engineering, R, open-source finance, data analytics

[1] Department of Economics and Finance, Richard J. Wehle School of Business, Canisius College, 2001 Main St., Buffalo, NY 14208. Email: yany@canisius.edu. Tel: (716) 888-2604.

1 Introduction

Nowadays, we have heard numerous times about the phrase "Big Data". However, few know how to define it since many so different explanations are available. In fact, many of them are contradict each other. For instance, over many meetings at my school, we have discussed this phrase repeatedly. Unfortunately, even up to today, I still don't have a clear one-sentence (or a few sentences) definition. This must be true for professors at other schools as well. When my students asked me the definition of big data, usually I tell them to "search your school bags for your memory stick". There is a good chance my students could have one or two memory sticks with a capacity of 4GB each. My answer is very simple, if one could generate and process 4GB data, then one could have the ability of dealing with "Big Data".[2] Obviously, this is not an accurate definition. Nevertheless, it is better than nothing and it is practical. How could students arm themselves with such an ability of processing 4GB data? The answer is: learn one computer language!

It is always my firm belief that an ambitious finance-major student should master at least one computer language.[3] This is especially true for students from quantitative-finance, MSF (Master of Science in Finance), business analytics, computational finance, data science or financial engineering programs. Among many good languages, R and Python are the two top choices.[4] Since 2010, I have spent a huge amount of time and efforts to apply R to finance. When teaching at Loyola University Maryland, I introduced R to Financial Modeling for the very first time. Many students liked my approach of using R as our computational tool. Nevertheless, some students still preferred Excel. I remember vividly my conversation with one MBA student who was taking my R course. He complained that our course didn't use Excel. I asked him why Excel was preferred and his answer really stunned me: "Because I am good at Excel". If a student is not motivated, then he/she has no incentive to learn R and apply it to finance.[5]

---

[2] Some researchers find that each company has, on average, about 30T data. Thus, another way to define 'Big Data' is the ability to store and process data with such a scale.

[3] My background: Ph.D. in finance, good at SAS, R, Python, Matlab and C, and an expert on financial data.

[4] Many schools, with quantitative-finance programs, still teach C++, see an example related to MIT Quant Club web page at http://quantfinanceclub.wix.com/mitqfc, then click "IAP Introduction to C++". The problem is that even after one-year (two semester) C++ courses, those students could not compete with my students who took my one-semester course related to R. Just look at Appendix B (A list of topics for term projects) which was offered to my students.

[5] Eventually, the student withdrew from the course.

Over the past six years, I taught several similar courses, 6 times, at 3 different schools (Loyola, University at Buffalo (UB) and Canisius) at both graduate and undergraduate levels. Those courses were Financial Theory and Modeling (Loyola, 2011-2012, MSF/MBA, GB725), Financial Modeling using R (Loyola, 2011, graduate, GB729), Quantitative Financial Analysis (Canisius, 2013, undergraduate, FIN457), and Financial Analysis with R (UB, 2014-2015, graduate, MGF690). The feedback was overwhelmingly positive. When teaching R to undergraduate students the first time in 2013 at my school, I was so worried about the student's evaluation. It was really a surprise to me that I received the highest student's evaluation among all my finance classes![6] The latest two courses were offered to the MSF students at UB (2014 and 2015). Again, the feedback was marvelous. One student commented that if I offer another course similar to this one, such as applying SAS or Python to finance, he would definitively take my course again!

In this short paper, I summarize my experiences of introducing R to finance at both graduate and undergraduate levels. In total, there are 7 quite unique and important factors: strong motivation, a good textbook, a hands-on environment, data intensive, a challenging term project, making hundreds of supporting R data sets available, and an easy way to upload those R data sets quickly. In the next few sections, I will explain each of those seven factors in more detail.

2 A strong motivation (factor #1)

Because of the current environment-big data, business analytics, computational finance, data science, quantitative-finance and financial engineering, it seems that motivating a finance-major student to learn one computer language is not that difficult. Obviously, the first lecture is the best time to motivate students. Comparing and contrasting different potential languages is an easy way to motivate. There are many good languages available, such as SAS, R Python, Matlab and C++.[7] Since the language used for my course is R, I need to compare it with other languages. Table 1 below shows scores based on the cost, ease of learning, capability of data handling, graph capability, level of the advancement in tools, job market preference, and customer services support and community.

[6] I got 4.76 (out of 5). The latest two teaching evaluations for the course of "Financial Analysis with R" at University Buffalo to MSF students are 4.286 (3.43/4*5 for Fall 2014) and 4.43 (Fall 2015).

[7] Most financial engineering programs still teach their students one year C++. Currently, I am writing a paper titled "A question to many financial engineering programs: Is it time to replace your one-year C++ courses with R and Python?"

Table 1: Comparison between R, SAS and Python[8] (5 being the best)

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

Many rankings (scores) are quite reasonable such as cost. However, I disagree with many of the other rankings. For example, SAS is superior to R in terms of data handling capacity[9], while for custom support, R is overrated compared with SAS. Based on my experience of applying various computer languages to finance,[10] my comparison is given below:

Table 2: comparisons by me (5 being the best)

	SAS	R	Python	Matlab
Cost	2	5	5	2
Easy of learning	2	5	4	2
Data handling	5	4	4	2
Graph	3	4.5	4	4,5
Current job	5	4	4	5
Future trend	3	5	5	4
Support	5	2	2	5
Total score	23	29.5	28	20

In total, R achieves the highest score of 29.5 while Python is number 2 (28). The worst disadvantage of both R and Python is their lack of support. I guess that if we are offered a 'free lunch', we should not complain that it is not as tasty as that at a trendy restaurant. This lack of support should be tolerable when compared with expensive software such as SAS and Matlab. The second way to motivate students is to mention that many good schools, such as New York University and Harvard, have adopted R as well.[11] In addition to motivate students at my first lecture, over the years I find two more occasions when I could do so: after finishing two chapters

[8] http://www.analyticsvidhya.com/blog/2014/03/sas-vs-vs-python-tool-learn

[9] At least based on my experience of using both languages. Even I published a book related to R, I still think that my best language is SAS in terms of skills and understanding.

[10] When teaching at NTU (Nangyang Technological University, Singapore), I taught C++ to doctoral students major in finance (the name of the course is "Introduction to Financial Databases") 4 times. For the NTU's Financial Engineering program, I taught "Advanced Financial Modeling" 3 times. The language of the course is C++ and Matlab. When working at Wharton School, I helped researchers to debug their programs written in SAS, Matlab and C++. Last year, I published a book titled "Python for Finance".

[11] See the links at http://www.nyu.edu/projects/politicsdatalab/learning_students.html, http://projects.iq.harvard.edu/rtc/event/introduction-r, http://projects.iq.harvard.edu/rtc/r-prog,

related to R packages and when I discuss various topics related to term projects (after the mid-term). I will come back to those two later in the paper.

To many new learners of a computer language, it is difficult to associate "motivation" with "intimation". For a finance-major student, even thinking about learning a computer language could be quite intimidating. Unfortunately, this is a reality: I heard a story about the encountering of an America-born-Chinese (ABC). He was born in the US and did not speak a single sentence of Chinese. During one summer vacation, he visited China with his classmates. He found two amazing experiences: everyone thought he could speak Chinese, and many Chinese claim that his English was easier to understand than his non-ABC classmates. The first was easy to explain because of his appearance. However, it was difficult to comprehend the second one. Personally, I had a similar experience: when teaching at Hofstra, several students wanted to transfer to my class from another professor. After more than three requests, I asked them the true reason and they told me that my English is easier to follow than that professor. I was undoubtedly sure that that professor spoke much better English than me. The reason is with those students themselves. When their minds tell them that "this person's English is easy to follow", they eventually understand. This is true for learning a programming language for non-computer science student.

With this in mind, it becomes my first priority to remove intimidation from my finance students. I have generated many one-line programs. For example, to write a present value function, we need just one line R codes.

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

Calling such a function is trivial, shown below:

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

The second one-line program is used to download historical daily price data from Yahoo!Finance.

```
> x<-read.csv("http://chart.yahoo.com/table.csv?s=IBM",header=T)
```

To view the first several lines, we use the head() function:

```
> head(x)
```

```
        Date   Open   High    Low  Close  Volume Adj.Close
1 2015-12-18 136.41 136.96 134.27 134.90 8983100    134.90
2 2015-12-17 139.35 139.50 136.31 136.75 4048600    136.75
3 2015-12-16 139.12 139.65 137.79 139.29 4313300    139.29
4 2015-12-15 137.40 138.97 137.28 137.79 4207900    137.79
5 2015-12-14 135.31 136.14 134.02 135.93 5103800    135.93
6 2015-12-11 135.23 135.44 133.91 134.57 5315200    134.57
```

For the last several lines, we apply the tail() function.

```
> tail(x)
            Date     Open     High      Low    Close Volume Adj.Close
13581 1962-01-09 552.0000 563.0003 552.0000 555.9998 491200  2.280983
13582 1962-01-08 559.5000 559.5000 545.0003 549.5003 544000  2.254319
13583 1962-01-05 570.5002 570.5002 558.9998 560.0003 363200  2.297395
13584 1962-01-04 576.9997 576.9997 570.9997 571.2503 256000  2.343548
13585 1962-01-03 572.0002 576.9997 572.0002 576.9997 288000  2.367136
13586 1962-01-02 578.4997 578.4997 572.0002 572.0002 387200  2.346625
```

To load the monthly Fama-French data set, we still have one line R codes. [12]

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

This is true when loading CRSP stock monthly data which contains 30,658 unique stocks (PERMNOs) from 1925 to 2014.

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

My logic is very simple: after a new learner has learnt 100 one-line programs, how could he/she not gain confidence? To motivate my students, I show them that we could write an R program to call students randomly. After launching R, we load an R package called 'png':

```
> library(png)
```

If you receive an error message, then install the package by issuing the following command:

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

After the package is installed, just type the following one-line R codes.

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

The second time to motivate my students is when discussing the two chapters related to R packages.[13] Any things taught before this moment is to prepare my students with basic concepts

[12] The R data set could be downloaded at http://canisius.edu/~yany/RData/ffMonthly.RData.
[13] Chapter 31: Introduction to R packages and Chapter 16: Two dozen R packages related to Finance, Yan 2016.

and skills, such as R is case-sensitive, writing simple programs, calling various pre-written R programs, running various loops and if- else- if conditions, and so on. It's like training a new soldier with basic skills; When he/she is good enough, the commander will lead them to a warehouse which contains all kinds of advanced weaponry, such as missiles and helicopters. Students get really excited when they find out about the many finance-related R packages at their disposal. If they got excited, they were motivated.

The last time to motivate my students is when discussing potential topics for their term projects. The major purpose of doing a term project is to help students apply what they have learnt to a real-world and challenging task. After the mid-term, I give students about 40 potential topics. The logic and concept of many topics are easy to explain and understand. For example, "Does January Effect Exist?", "Which Party, Democratic Party or Republican Party 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, I show students how to test the January Effect by randomly choosing one stock. However, for a good term project, students have to choose ALL stocks. For this very reason, when teaching at UB to their MSF students the second time in 2015, I offer my students several hundred R data sets related to CRSP.

3 A good textbook (factor #2)

When preparing to teach R to my finance major students for the first time in 2010, I spent considerable time trying to locate a textbook with a reasonable quality. Unfortunately, I found none. Almost all programing books are written by professors from the computer sciences.[14] This does not dispute the quality of many good books. It only means that those books were not written for my students who major in finance. Here are several reasons. First, those books are too detail-oriented, i.e., they contain too much information. Most of the information (chapters) is not relevant in the view of a finance student. Second, those textbooks are too intimidating. In the previous section, I argue that overcoming intimidation is vital to motivate my finance-major students to learn R. Third, those textbooks are not related to finance. A few of them have one or two financial applications. But none of them devoted completely to finance. And last but not

[14] On the back cover of my book (Financial Modeling using R), the first sentence reads "This is a programming book written by a finance professor". Hopefully, after finish this paper and my book, readers would appreciate its true meaning.

least, those books don't use data intensively. Moe importantly, they don't use the economic and financial data that is publicly available.

Because of those limitations, I started to write my own lecture notes. I remember a request when I wrote my first book "Python for Finance". The external expert, hired by the publisher, asked me repeatedly to drop the section related to Python installation since he considered it trivial. However, I refused: if a student could not install Python or R, there is no chance that he/she would continue to learn such a language. This is the reason that I use the basic R environment. To encourage students to learn R, I wrote many one-sentence or a few-sentence R programs to show them how simple R could be. Below is the program to price a call options 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 (shown below). If we use just 5 lines of R codes for such a complex quant model, it would definitively boost my students' confidence dramatically.

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

Three features for my book include free software; the whole book is devoted to finance, and it uses data intensively, especially publicly-available economic and financial data.[15]

4 A hands-on learning environment (factor #3)

All my lectures are conducted in a computer lab. The first few weeks are crucial: after offering students a task, I type my codes on the screen.[16] Then, I walk around the classroom to help any individual students debug his/her codes. To prevent students from copying and pasting from my slides, the R codes shown on my PowerPoint slides are images. Coding should not be that difficult if we spend enough time on it.[17] Again, overcoming intimidation is my first priority.

[15] Here is the table of Contents of the book http://www3.canisius.edu/~yany/doc/tocFMuR.pdf.
[16] At this moment, I don't copy-and-paste, I type those codes.
[17] Usually, I require my students to spend at least one hour per day, including weekend, outside classroom on R.

Because of this, I move quite slowly for the first several weeks. Also, since it's hands-on, the ideal class size should be less than 15. [18]

In-class exercises also play an important role. For each lecture, I usually give one in-class exercise. For the first several weeks, the exercises are quite simple: I discuss the task and type my codes, [19] and students just type those codes to complete the task. Gradually, I give several basic steps like a flow chart and ask students to write their own codes. Since term-projects play a big role for this course, after the mid-term, most of my in-class exercises are related to various term projects. For example, students need to estimate equal-weighted or value-weighted portfolio returns for many term-projects. Because of this, I design an in-class exercise to replicate S&500 monthly returns.

Another practice of mine is to encourage my students to generate one text file. Over the whole semester, they continue adding various programs to it. For example, when teaching "Financial Analysis with R" at Buffalo University, I asked each student to generate text file called mgf690.txt. By the end of the semester, such a file should contain 50 to 100 small programs. There are several advantages of doing so: First, students could use the old programs for their homework without reinventing the wheel. Second, they save lots of time during the mid-term and final. And lastly, they could use many of their own programs in the future. The text format is also beneficial since anyone could open it without any trouble.

5 Data intensive, especially the usage of public data (factor #4)

Nowadays, there are so many publicly available economic and financial data. The following table shows a partial list:

Table 3: a list of open data sources[20]

Name	Web page	Data types	Related topics
Yahoo Finance	http://finance.yahoo.com	Current & 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

[18] Unfortunately, because of the popularity of this course, the class size of my latest class at UB is capped at 30, twice of my ideal one. Because of this, the effectiveness of hands-on is decreased since I spend less time on each student.

[19] Again, for simple codes and during the first few weeks, I type codes instead of copy-and-paste.

[20] Table 5.1, Financial Modeling using R, Yuxing Yan, 2016. In my R book, I have devoted a whole chapter discuss many of those data sources.

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
Bondsonline	http://www.bondsonline.com/	Bond data	Fixed income, trading strategy
US. Dept. Treasure	http://www.treas.gov	US. Treasure 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 etc.	Macro economics
National Bureau of Economic Research	http://www.nber.org/	Business cycles, vital statistics, report of Presidents	Macroeconomics, financial stability

Using publicly available data has several advantages: they are free and we could get the recently data such as yesterday's closing stock price. This is not possible for expensive financial database, such as CRSP. From SEC, we could get a company latest filings, but we would have to wait for such information if a school subscribed the Computata (CapitalIQ) database. It is also quite easy to write an R program to access those public data. Last but not least, students from teaching schools could learn R quite effectively if they could process a huge amount of data.

6 A challenging term-project (factor #5)

There are several objectives of doing a challenging term-project. First, this is a great opportunity to summarize what students have learnt. Second, they could learn more about processing various data sets, especially when dealing with several thousand stocks. Third, introducing various term topics would open students' horizons since many term projects came from seminal papers. For many topics, in addition to the explanation of underlying logic, I offer concrete steps to finish those projects. Whether students have achieved the same results as the original papers is not that critical, the process is. For example, I explain in detail how the momentum strategy works (Jegadeesh and Titman, 1993) and steps to replicate it. Table 3 shows a few quite challenging topics for a term project.

Table 4: 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, Cakici and Whitelaw, 2011)
	41	Impact of business cycle on the above four trading strategies, (Yan and Zhang , 2015)

Lastly, finishing a good term project would benefit their job hunting. I like to tell my students: "during your job interview, brag about your term-project by explaining why your topic is interesting, what kind of data you used and how to retrieve and process data, as well as the conclusions reached".[21] In 2015, I had 30 students for the course of "Financial Analysis with R" at Buffalo University. Since each group could have up to 3 group members, I ended up with 10 groups, see Appendix C for the topics chosen.

To finish many challenging topics, the instructor's strong support is critical, especially in terms of data. Whenever my students have any issues related to programs or data, I will try my best to satisfy their needs. After students have submitted their term projects before deadline, I set aside one day for their presentations. Over the years, I find that students have benefited greatly from other groups' presentations because of their various topics and depth. For the presentations, each group has 15 minutes for their presentation plus 5 minutes of Q&A. Before the presentation, I gave each student a 'peer evaluation' sheet. Based on the feedback, I find that this is a very good practice to help students sharpen their critical thinking. [22]

7 Making many supporting R data sets available (factor #6)

Making many good R data sets available is a necessary condition for finishing a challenging term project. There are two types of data sets: from public sources and from expensive financial databases, such as CRSP. In the US, almost all schools with a quantitative-

[21] Actually, 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.

[22] For example, after students' presentation in the fall 2015, I spent a whole day to type their comments , see the link below for the fall 2015, http://canisius.edu/~yany/peerComment2015.txt

finance program have subscribed to CRSP. Thus, it is a great idea to use CRSP to finish many challenging term projects. In 2015, I introduced CRSP into my course called "Financial Analysis with R" at UB to their MSF students for the first time. To me, it is a big surprise to see that 9 of the 10 term projects are involved with CRSP. This suggests that it is a huge waste of money for those schools, with CRSP subscription, that they don't use the database for their teaching.[23]

For the first few weeks, I show students how to generate R data sets for many small data sets, such as Fama-French 3 factor, Fama-French-Carhart 4 factor and Fama-French 5-factor time series. Students would benefit greatly from learning how to generate those data sets themselves. However, for a challenging term project, such as replicating so-called MAX trading strategy suggested in Bali, Cakici and Whitelaw (2011), it is not feasible to ask students to generate CRSP related R data sets themselves. To help students finish those 10 challenging term projects, I have generated over 400 R data sets. A few examples are shown in Table 4:

Table 5: 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 constitutients           *
* sp500monthly         # monthly S&P500 returns         *
*-------------------------------------------------------*
```

When teaching financial modeling or financial analysis in North America, it is impossible not to discuss or use a huge amount of data. When using financial data, it is not possible not to discuss CRSP and Compustat. Because of this, I generated over 300 R data sets from CRSP and Compustat.

8. An easy way and quick way to upload those R data sets (factor #7)

[23] Currently I am writing a short paper called "CRSP for Teaching", see the abstract here http://canisius.edu/~yany/crsp4teachingAbstract.pdf

Two words could be used to summarize my approach: *trivial* and *second*. For 'trivial', uploading various R data sets is trivial. It also means that an instructor does not have to explain how to upload various data sets, since he/she could just give students one line of R codes. For '*second*', uploading each data set should take just a few seconds (most of times, just 2 seconds).

In 2015, when teaching "Financial Analysis with R" the second time, I introduced CRSP, Compustat and TAQ into my teaching and support various term projects. In total, I have generated over 300 individual R data sets. Since CRSP, Compustat and TAQ are proprietary databases, I use Fama-French data sets as an example. You will see how easy it is to load any data set. First, just type the following one-line R codes:

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

After hitting the return key, we will see the instruction, shown below:

```
> source("http://canisius.edu/~yany/loadFF.R")
function(n){
"
Objective: retrieve several Fama-Frech R data sets
    n    : an integer, see the names of data sets below.

    n  R data set            Description
   --- ------------  ----------------------------------
   1   ffMonthly     monthly Fama-French          3 factors
   2   ffcMonthly    monthly Fama-French-Carhart  4 factors
   3   ff5Monthly    monthly Fama-French          5 factors
   4   ffDaily       daily   Fama-French          3 factors
   5   ffcDaily      daily   Fama-French-Carhart  4 factors
   6   ff5Daily      daily   Fama-French          5 factors

 Example #1: # know the usage of this function
             >loadFF

 Example #2: # to load the first FF data set
             >loadFF(1)
             >head(ffMOnthly,2)
                     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

 Example #3:>loadFF(6)
            >head(ff5Daily,2)

";loadFF_(n)
}
>
```

To load the Fama-French monthly data factors, we issue `loadFF(1)`, shown below:

```
> loadFF(1)
Data set loaded is    ffMonthly
> head(ffMonthly,2)
        DATE MKT_RF    SMB     HML     RF
1 1926-07-01 0.0296 -0.023 -0.0287 0.0022
2 1926-08-01 0.0264 -0.014  0.0419 0.0025
> tail(ffMonthly,2)
            DATE  MKT_RF     SMB     HML RF
1071 2015-09-01 -0.0308 -0.0271  0.0073  0
1072 2015-10-01  0.0775 -0.0186 -0.0032  0
>
```

This simple method is true for other data sets (databases) as well. Assume that we have 200 data sets from CRSP. In my lecture, we load those data sets quite easily: just type one-line of R codes, shown below. Obviously since CRSP is a proprietary database, I could not make those data sets publicly available. The following image is just an illustration.

```
> source("http://canisius.edu/~yany/loadCRSP.R")
function(n){
"Objective: upload one R data set from CRSP
        n : an integer
    n   R data set    Description
   ---  ------------  -------------
    1   crspInfo      CRSP infomation 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/deleved 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 82936G20 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)
}
>
```

The names of `loadCRSP()`, `loadCOMP()`, `loadFF()` and `loadTAQ()` are quite clear. However, we could use one-letter functions to save our typing time. We could use a, b, c, and d for those 4 types of data sets. For example, to load the first Fama-French data set, we use `c(1)` instead of `loadFF(1)`:

```
>a=loadCRSP; b=loadCOMP; c=loadFF;d=loadTAQ
```

This method is quit flexible. Alternatively, we could use two-letter functions, shown below:

```
>cr=loadCRSP; co=loadCOMP; ff=loadFF;ta=loadTAQ
```

In a sense, my method works even for a new user who has no experience in R nor CRSP. After a few minutes of trial-and-error, he/she will become a master for loading those data sets.[24]

[24] For my teaching, I made just a few R data sets available online. When discussing CRSP, I give my students a zip file. After they download and unzip it into a local subdirectory, such as c:/temp/, I offer them a function to load each of those data sets locally.

8 Conclusions

In this short paper, I show potential instructors how to design a good course to teach finance-major students to learn R and apply it to finance. Based on my past 6-year teaching experience at three universities, there are seven important factors: 1) strong motivation, 2) a good textbook, 3) a hands-on teaching/learning environment, 4) data intensive, 5) a challenging term project plus student presentation; 6) tons of supporting R data sets available for those term projects, and 7) an easy and quick way to upload those R data sets. It is my belief that programming skills will become a necessity for all finance-major students in the foreseeable future. Instructors from various finance departments around the world would hopefully find this short paper helpful. If an instructor wanted to teach courses such as "Financial Analysis with R", I could offer my slides and many of my R data sets.[25] My syllabus, for Financial Analysis with R taught twice at University Buffalo, is shown in Appendix A.

References

Altman, Edward, 1968, Financial ratios, discriminant analysis and the prediction of corporate bankruptcy, *Journal of Finance*, 23(4), pp 598-608.

Amihud, Yakov, 2002, Illiquidity and Stock returns, *Journal of Financial Markets* 5, 31-56.

Bali, Turan G., Nusret Cakici, and Robert F. Whitelaw, 2011, Maxing Out: Stocks as Lotteries and the Cross-Section of Expected Returns, *Journal of Financial Economics* 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.

CRSP, CRSP user manual, University of Chicago

George, Thomas J, and Chuan-Yang Huang, 2004, The 52-week High and Momentum Investing, *Journal of Finance* 54, 5, 2145-2176.

Jegadeesh, N., and S. Titman, 1993, Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency, *Journal of Finance* 48, 65-91.

Moskowitz, Tobias, and Mark Grinblatt, 1999, Do industries explain momentum? *Journal of Finance* 54, 2017-2069.

Pastor, L. & Stambaugh, R., 2003, Liquidity risk and expected stock returns, *Journal of Political Economy* 111, 642-685.

[25] So far, two finance professors have adopted my first book "Python for Finance" their two courses: Financial Analytics (2015) and Data Science in Finance (2016). If some schools plan to offer a summer course, such as Financial Analysis with R, I love to be the instructor.

Roll, Richard, 1984, A simple implicit measure of the effective bid-ask spread in an efficient market, *Journal of Finance* 39, 1127-1139.

Yan, Yuxing, 2016, Financial Modeling using R, *Tate Publishing*, ISBN: 978-1-68187-530-9 https://www.tatepublishing.com/bookstore/book.php?w=9781681875309 http://www.amazon.com/Financial-Modeling-using-Yuxing-Yan/dp/1681875306/ref=sr_1_2?ie=UTF8&qid=1452098908&sr=8-2&keywords=financial+modeling+using+R

Yan, Yuxing, 2016, CRSP for Teaching, working paper, Canisius College, the abstract is available at http://canisius.edu/~yany/doc/crsp4teachingAbstract.pdf

Yan, Yuxing, 2014, Python for Finance, 2014, *Packt Publishing*, *ISBN: 978-1-78328-437-5*, http://www.packtpub.com/python-for-finance/book

Appendix A: My syllabus

MGF 690 F2F
Special Topics: Financial Analysis with R
Course Syllabus
Department of Finance and Managerial Economics
University at Buffalo Fall 2015

Instructor: Paul Yan
Office: 359 Jacobs Management Center (Tuesday only)
Emails: yyan6@buffalo.edu and yany@canisius.edu
Phones: 716 645 3277 (Tuesday), 716 888 2604 (other weekdays)

Textbook: *Financial Modeling using R* by Yuxing Yan
Publisher: Tate Publishing, ISBN: 978-1-68187-530-9
https://www.tatepublishing.com/bookstore/book.php?w=9781681875309

References: *An Introduction to R* http://canisius.edu/~yany/doc/R-intro.pdf
The R Language Definition http://canisius.edu/~yany/doc/R-lang.pdf

Web sites: https://ublearns.buffalo.edu/ and http://canisius.edu/~yany/R.shtml
> source("http://canisius.edu/~yany/mgf690.R") Note: I will explain this line in week 2

Software: R, open source statistical and computational software[26]. Students are expected to spend at least 1 hour per day on R outside the classroom.

Prerequisites: Minimum two finance courses, such as Corporate Finance and Portfolio Analysis
Lecture: Tuesdays: 8:00am ~ 10:50am @ Jacobs B30 (a computer lab)
Office hours: Tuesdays 11:00am ~ 1:00pm @ 359 Jacobs Management Center

Four objectives:

1) Learn/review basic financial concepts such as Ratio Analysis, Portfolio Theory, CAPM, Fama-French-Carhart Factor Model, Monte Carlo simulation, Options Theory, VaR (Value at Risk) and Market microstructure
2) Learn and apply R to finance
3) Focus on publicly available financial data such as Yahoo Finance, Google Finance, Prof. French's Data Library and Federal Reserve Economic Data Library (FRED).
4) Use CRSP and/or TAQ (Trade and Quote) databases. [27]

Note: CRSP and TAQ are for the term-project if some groups or individual student want to use CRSP or TAQ databases for their projects.

[26] www.r-project.org
[27] The last objective is true only for schools that have subscribed to those expensive financial databases.

Teaching Method:

Each class will be consist of two parts: lecture (including discussion of homework) and hands-on.

Grading Policy:

Homework	30%
Midterm	20%
Final exam	25%
Group project	10%
Group presentation	5%
Class participation	10%
Total	100%

Mid-term and Final:

All exams (midterm and final) will be conducted in the computer lab. Those are open book exams with three types of questions: related to 1) finance; 2) R or 3) financial data.

Group project

Each group can have up to three members. A topic should be closely associated with this course. The maximum number of pages of your report is 15 with 12-point font. Please discuss with me your topic before you start to work on it. Some basic criterions are listed below.

Real world topics are especially encouraged. Three parts are essential:

1) theory and background of the topic,
2) R programs with a short explanation of the codes,
3) final data set (plus the codes to process the data, the source of raw data)
   Note: please do not send me your raw data.

The second type of projects is to study one of R packages. Three parts are critical:

1) why this specific package is useful in finance
2) a summary of all or most important functions offered by the package
3) examples to use them

Note: see a list of potential topics, at the end of the syllabus, for the group projects.

References (most of them are for term projects)

After mid-term, I will post all papers, manuals and PowerPoint presentations online.

[Note: since most references are the same as the papers mentioned in this paper, this part is removed for brevity]

Tentative schedule

#	Date	Topics	Description (F for Finance)
1	9/1	Syllabus discussion, introduction to R	A short survey, self-intro, syllabus, course structure, mid-term and final F: Review of basic financial concepts, financial formulae, risk vs. return, how to measure risk, time value of money, present value (PV), future value (FV), PV(annuity), etc. R: Installation, assignment, basic math functions: mean(), min(),max(), median(), sd() and use R as a scientific calculator
2	9/8	Review of basic finance concepts and formulae (functions)	F: How to estimate returns? PV(bond), simple and compound interest, conversion of returns for different frequencies R: How to write an R function? double_f(), pv_f(), fv_f(), IRR(), pv_annuity(), fv_annuity(), pv(perpetuity), pv(perpetuity_due) How to call your functions? several ways to input data, matrix, differential operator and use R as a financial calculator
3	9/15	Options	F: Black-Scholes-Merton option model, trading strategies with options, implied volatility, Black's approximation for an American option, Greeks, put-call parity, hedging strategy, risk-neutral evaluation R: pnorm() [cumulative standard normal distribution], bs_f.R, Implied_vol.R, greeks.R
4	9/22	Financial Statement Analysis	F: Financial statement analysis, ratio analysis, profitability ratio: operating margin, net profit margin, ROA, ROE, current ratio, book debt-equity ratio R: Retrieve BS, IS, CF, current ratio, debt equity ratio,
5	9/29	Open data [Yahoo finance etc.]	F: daily vs. monthly returns, Yahoo finance, Rf, French's Data Library R: How to input data from a text file? simple programs to download historical price data from Yahoo finance, download data from Prof. French's data library , download risk-free rate
6	10/6	CAPM	F: CAPM, β estimation, rolling/portfolio β, hedging portfolio market risk R Several functions: as.Date, week_day_effect.R , data.frame, ,beta.R, rolling_beta.R, replicate
7	10/13	Multi-factor models, Sharpe Treynor ratios	F: Fama-French 3-factor model, momentum strategy, Sharpe ratio, Treynor's ratio, 52-weeks high, Jensen's α, R: ff3factor, ff4, weekday/January effect. sharpe.R, Treynor.R
8	10/20		Midterm
9	10/27	T-test, F-test, Autocorrelation, Causality	F: T-test for significance, equality of means, F-test for difference of volatility, Granger causality test, Durbin-Watson autocorrelation test R: t.test(), var.test(), dwtest(), Wilcoxon.test(), granger_test(). Note: post two dozen topics related to term projects (see Appendices B and C for two examples)
10	11/3	Monte Carlo Simulation	F: simulation and assumptions, normality test, estimate variance-covariance matrix, conversion variances between different frequencies, path dependent options, sensitivity analysis, scenario analysis R: rnorm(x), random number from normal, uniform distribution Several functions: as.Date, week_day_effect.R,In class-exercise: find 1 C among 500 Os and Monte Carlo Simulation to price European and Asian options

Continued

Lecture	Date	Topic	F (finance) and R
11	11/10	CRSP for teaching using R	F: What is CRSP? CRSP monthly, daily time series data, event data (more topics for term projects) R: stockMonthly, indexMonthly, indexDaily, stockD1925 to stockD2014, various R program to retrieve/process data efficiently
12	11/17	Portfolio theory	F: variance, standard deviation, correlation, return matrix, portfolio return, portfolio volatility of 2-stock (n-stock) portfolio, variance-covariance matrix, portfolio optimization R: package "fPortfolio"
13	11/24	Value at Risk	F: introduction to VaR, (standard) normal distribution, thick tail distribution R: standard normal distribution, VaR_01.R, VaR_02.R, introduction to packages in R.
14	12/1	Group Presentation	3 to 4 groups
15	12/8	Group Presentation	All other groups
	Extra	TAQ for teaching using R	TAQ (Trade and Quote) are high frequency database by NYSE MTAQ (up to second) DTAQ (up to millisecond) R data sets: TAQct, TAQcq, DTAQ22ct, DTAQ22cq TORQct, TORQcq, TORQcd, TORQsod R: loadTAQ, filterCT, filterCQ, spread, relativeSpread,leeRead
	Extra	Credit risk	F: Credit rating, credit score, probability of default, Z-score for predicting Bankruptcy, Moody's, S&P, Fitch, Best crediting systems, credit transition matrix R: prob_default.R, Z_score.R, KMV.R
	Extra	Spread estimation from low-frequency	F: spread estimation from low frequency data, Roll (1984), Corwin and Schultz (2012) high-low spread R: Roll.R, Corwin_Schultz.R
	Extra	Liquidity measure	F: Amihud (2002) illiquidity measure, Pastor and Stambaugh (2002) liquidity measure R: Amihud.R, PS.R
	TBA	Final	Final-exam

Appendix B: A list of potential topics for term projects

Warm up		1	Financial statement analysis
		2	An interned connected financial calculator (Yan, 2012)
		3	Correlations among stocks in US, UK, Canada, France, China, Japan and Australia
		4	A Business cycle indicator (Yan and Zhang, 2015)
		5	Use journal ranking data efficiently (SCImago Journal and Country Ranking)
		6	Find an optimal portfolio
		7	How much you need when you retire? Social Security Benefit calculator
		8	Which party, Republican or Democratic, could manage the economy better?
		9	Monte Carlo Simulation (standard normal distribution, one variables) VaR
		10	PCA (Principal Component Analysis)
Data	Public data	11	Generate R data sets for 200 stocks, CPI, GDP, Unemployment rate etc.
		12	Generate R data sets for Fama-French 3-factors, 5 factors etc.
		13	Generate R data sets for all SEC 10Q and 10K index files from SEC (1993-2015)
		14	Generate R data sets for TORQ (Trade, Order, Report and Quote) database
		15	Generate R data sets for TDAQ (millisecond by millisecond transaction data)
		16	Parse 10K data from SEC filings, generate related R data sets
	CRSP/TAQ	17	Generate R data sets for one month's TAQ data (MTAQ)
		18	Generate R data sets for crspInfo, stockMonthly, indexMonthly for CRSP
		19	Generate R data sets for stockDaily, indexDaily for CRSP
		20	Generate R data sets for TDAQ for several months
Using public data		21	Are annual beta mean reversion?
		22	Test the January and weekday effects
		23	Does size effect exist?
		24	Tracking errors
		25	Z-score (bankruptcy prediction, Altman, 1968)
		26	52-week High trading strategy using more than 200 stocks
		27	estimate Roll spread from daily data (Roll, 1984)
		28	Assessment of multiple choice questions using R
		29	Monte Carlo Simulation (capital budgeting, replicate a Slot Machine)
		30	Monte Carlo Simulation (one variables) VaR, n correlated stocks
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, Cakici and Whitelaw, 2011)
		41	Impact of business cycle on the above four trading strategies, (Yan and Zhang , 2015)

Appendix C: Topics chosen by my students, MGF690 (Financial Analysis with R) 2015

Topic	Group members
Topic #30: Monte Carlo Simulation (one variables) VaR, n correlated stocks.	Xi, Xunyi, and Zhen Long
Topic #36: Momentum trading strategy	Qun, Yi and Jiawei
Topic #26: 52-week High trading strategy using more than 200 Stocks	Jiaqi, Chen and Yiwen
Topic #7: which party manages the economy better?	Siddhartha, Snehal, Vinayak and Mohd
Topic #46 : Roll spread	Jiaying, Jing and Siqi
Topic # 44 Is liquidity a price factor?	Yuhao, Shuhan and Yang
Topic #42: Which model is the best, CAPM, FF3, FFC4, or FF5?	Kai, Ye and Cheng
Topic#42 Which model is the best	Yixin, Qiaowen, Han
Topic #25 (Z-score Bankruptcy Prediction)	Lu and Mingzhe
Topic # 22 Test the January and weekday effects.	Chenxi, Yutu, Jinhao

Appendix D: A complete list of R data sets related to term projects

```
*-------------------------------------------------------*
* 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 constitutients           *
* sp500monthly         # monthly S&P500 returns         *
* sp500daily           # daily S&P500 daily returns     *
* indexDaily           # daily index                    *
* d1925                # daily stock for 1925           *
* d1926                # daily stock for 1926           *
* .....                                                 *
* d2013                # daily stock for 2013           *
* d2014                # daily stock for 2014           *
* eventDaily           # daily event data set           *
* eventMonthly         # monthly event data set         *
*-------------------------------------------------------*
* 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 TAQ (MTAQ and DTAQ)           *
*-------------------------------------------------------*
* ct30_20041101 # CT (Consolidated Trade)               *
* cq30_20041101 # CQ (Consolidated Quote)               *
* DTAQct50      # CT (Ultra-high frequency,millisecond) *
* DTAQcq50      # CQ (Ultra-high frequency)             *
* TORQct        # CT from TORQ                          *
 *TORQcq        # CQ from TORQ                          *
* TORQcd        # CD (audit) from TORQ                  *
* TORQcod       # COD  from TORQ                        *
*-------------------------------------------------------*
* Accounting data etc.                                  *
*-------------------------------------------------------*
* compInfo      # links between GVKEY,CUSIP,TICKER etc  *
* varDecription # descriptions of data items (vars)     *
* deletionCodes # deletion codes                        *
* acc1950       # accounting data for 1950              *
* acc1950       # accounting data for 1951              *
*  …                                                    *
* acc2015       # accounting data for 2015              *
* sec10K        # SEC quarterly indices (1993-2015)     *
*-------------------------------------------------------*
```

Appendix E: Topic #8 which political party manages the economy better?

Objectives:

1) understand value-weighted and equal-weighted portfolios
2) understand T-test

Data sets needed:

1) sp500monthly.RData or
2) indexMonthly.RData

Currently we are seeing many presidential debates among potential presidential nominees for Republican and Democratic parties. One question a potential voter likes to ask is which party could manage the economy better. With this term project, we try to asker this question: which party could manage the economy better in terms of the performance of the stock market. Here we use VWRETD (Value-weighted) and EWRETD (Equal-weighted) S &P500 market returns. According to the web page of http://www.enchantedlearning.com/history/us/pres/list.shtml, we could find to which party a US president belongs.

Table 1. Presents and their party affiliations

President	which party	time period
30. Calvin Coolidge (1872-1933)	Republican	1923-1929
31. Herbert C. Hoover (1874-1964)	Republican	1929-1933
32. Franklin Delano Roosevelt (1882-1945)	Democrat	1933-1945
33. Harry S Truman (1884-1972)	Democrat	1945-1953
34. Dwight David Eisenhower (1890-1969)	Republican	1953-1961
35. John Fitzgerald Kennedy (1917-1963)	Democrat	1961-1963
36. Lyndon Baines Johnson (1908-1973)	Democrat	1963-1969
37. Richard Milhous Nixon (1913-1994)	Republican	1969-1974
38. Gerald R. Ford (1913- 2006)	Republican	1974-1977
39. James (Jimmy) Earl Carter, Jr. (1924-)	Democrat	1977-1981
40. Ronald Wilson Reagan (1911- 2004)	Republican	1981-1989
41. George H. W. Bush (1924-)	Republican	1989-1993
42. William (Bill) Jefferson Clinton (1946-)	Democrat	1993-2001
43. George W. Bush (1946-)	Republican	2001-2009
44. Barack Obama (1961-)	Democrat	2009-

Thus, we could generate the following table. The PARTY and RANGE variables are from the web page. YEAR2 is the second number of RANGE minus 1, except the last row.

Table 1: Parties and Presidents since 1923

PARTY	RANGE	YEAR1	YEAR2
Republican	1923-1929	1923	1928
Republican	1929-1933	1929	1932
Democrat	1933-1945	1933	1944
Democrat	1945-1953	1945	1952
Republican	1953-1961	1953	1960
Democrat	1961-1963	1961	1962
Democrat	1963-1969	1963	1968
Republican	1969-1974	1969	1973
Republican	1974-1977	1974	1976
Democrat	1977-1981	1977	1980
Republican	1981-1989	1981	1988
Republican	1989-1993	1989	1992
Democrat	1993-2001	1993	2000
Republican	2001-2009	2001	2008
Democrat	2009-2012	2009	2014

Detailed procedure is given below:

Step 1: retrieve an R data set called sp500monthly.RData

```
> head(sp500monthly)
        DATE    VWRETD    VWRETX    EWRETD    EWRETX    VALUE  N      VAL500
1 1925-12-31      <NA>      <NA>      <NA>      <NA> 15236830 89        <NA>
2 1926-01-30 -0.001783 -0.003980  0.006457  0.003250 15277664 89 15236829.50
3 1926-02-27 -0.033296 -0.037876 -0.039979 -0.042451 14712895 89 15277664.00
4 1926-03-31 -0.057708 -0.062007 -0.067915 -0.073275 14012079 89 14712894.90
5 1926-04-30  0.038522  0.034856  0.031441  0.027121 14500482 89 14012079.20
6 1926-05-28  0.013623  0.009070  0.012011  0.009515 14778796 89 14500482.20
  N500 SP500INDEX  SP500RET
1 <NA>      12.46      <NA>
2   79      12.74  0.022472
3   81      12.18 -0.043956
4   81      11.46 -0.059113
5   82      11.72  0.022688
6   82      11.81  0.007679


> tail(sp500monthly)
           DATE    VWRETD    VWRETX    EWRETD    EWRETX       VALUE   N
1064 2014-07-31 -0.014274 -0.015582 -0.022790 -0.023728 17655206100 501
1065 2014-08-29  0.039593  0.037275  0.042378  0.040379 18287631200 502
1066 2014-09-30 -0.013867 -0.015410 -0.025746 -0.027661 18004701800 502
1067 2014-10-31  0.024587  0.023411  0.030985  0.030032 18392036300 502
1068 2014-11-28  0.028013  0.025552  0.024561  0.022099 18823339700 502
1069 2014-12-31 -0.002489 -0.004270  0.003412  0.001396 18803349400 502
             VAL500 N500 SP500INDEX  SP500RET
1064 17937192900.00  501    1930.67 -0.015080
1065 17661342600.00  502    2003.37  0.037655
1066 18301794000.00  502    1972.29 -0.015514
1067 18004701800.00  502    2018.05  0.023201
1068 18392036300.00  502    2067.56  0.024534
1069 18835770400.00  502    2058.90 -0.004189
>
```

Step 2: Classify VWRETD (returns) into two groups according to YEAR1 and YEAR2: under Republican and under Democratic

Step 3: Test the null hypothesis: two group means are equal.

$$\bar{R}_{Democratic} = \bar{R}_{Republican} \quad (1)$$

Step 4: Discuss your results and answer the following question: are the monthly mean returns under both parties equal?

Note: repeat the above process by using EWRETD (equal-weighted market index).

Appendix F: Topic #36: Momentum strategy

We could use a simple phrase to summarize the so-called momentum trading strategy: buy winners and sell losers. Here, we have an implied assumption: within a short-term (between 3 months and 12 months), the winner will remain a winner while a loser would continue to be a loser. Two related questions: 1) how to define a winner from a loser? 2) how to conduct a test?

Objectives of this term project:

1) Understand the CRSP database
2) Understand how to use R to retrieve and process data using CRSP monthly stock data
3) Prove or disapprove so-called momentum strategy by replicating Table 1 of Jegadeesh and Titman (1993)

Prerequisites: access to an R data set called stockMonthly.RData (I will supply this data set)

Basic logic: According to Jegadeesh and Titman (1993) it is a profitable trading strategy if we buy the past winners and sell the past losers.

Notations: Check the past K-month returns, and then form a portfolio for L months,
Where K=3,6,9 and 12 and L=3, 6, 9 and 12. Below we use K=L=6 as an example.

Trading strategy: Estimate all stocks' past 6- month returns and sort stocks into 10 groups (deciles) according to their 6-month total returns. Long the top decile (winners) and short the bottom decile (losers) for the next 6 months.

Procedure:

Step 0: Starting month: January 1927

Step 1: Retrieve CRSP data (PERMNO, DATE and RET)

Step 2: Estimate past 6-month cumulative returns R_t^{6month}

Step 3: Sort all stocks into deciles according to their cumulative 6-month returns

Step 4: Long winners (best return group) and short losers for the next 6-month

Step 5: Estimate portfolio returns

Step 6: Move to the next month and repeat the above steps until the last month (July, 2013 since the last month for the current CRSP monthly data is December 2014)

References

Jegadeesh Narasimhan and Sheridan Titman, 1993, Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency, *Journal of Finance* 48 (1), 65-91.

Appendix A: Table 1 from Jegadeesh and Titman (1993).

Table I

Returns of Relative Strength Portfolios

The relative strength portfolios are formed based on J-month lagged returns and held for K months. The values of J and K for the different strategies are indicated in the first column and row, respectively. The stocks are ranked in ascending order on the basis of J-month lagged returns and an equally weighted portfolio of stocks in the lowest past return decile is the *sell* portfolio and an equally weighted portfolio of the stocks in the highest return decile is the *buy* portfolio. The average monthly returns of these portfolios are presented in this table. The relative strength portfolios in Panel A are formed immediately after the lagged returns are measured for the purpose of portfolio formation. The relative strength portfolios in Panel B are formed 1 week after the lagged returns used for forming these portfolios are measured. The t-statistics are reported in parentheses. The sample period is January 1965 to December 1989.

		Panel A				Panel B			
J		K = 3	6	9	12	K = 3	6	9	12
3	Sell	0.0108	0.0091	0.0092	0.0087	0.0083	0.0079	0.0084	0.0083
		(2.16)	(1.87)	(1.92)	(1.87)	(1.67)	(1.64)	(1.77)	(1.79)
3	Buy	0.0140	0.0149	0.0152	.0156	0.0156	0.0158	0.0158	0.0160
		(3.57)	(3.78)	(3.83)	(3.89)	(3.95)	(3.98)	(3.96)	(3.98)
3	Buy-sell	0.0032	0.0058	0.0061	0.0069	0.0073	0.0078	0.0074	0.0077
		(1.10)	(2.29)	(2.69)	(3.53)	(2.61)	(3.16)	(3.36)	(4.00)
6	Sell	0.0087	0.0079	0.0072	0.0080	0.0066	0.0068	0.0067	0.0076
		(1.67)	(1.56)	(1.48)	(1.66)	(1.28)	(1.35)	(1.38)	(1.58)
6	Buy	0.0171	0.0174	0.0174	0.0166	0.0179	0.0178	0.0175	0.0166
		(4.28)	(4.33)	(4.31)	(4.13)	(4.47)	(4.41)	(4.32)	(4.13)
6	Buy-sell	0.0084	0.0095	0.0102	0.0086	0.0114	0.0110	0.0108	0.0090
		(2.44)	(3.07)	(3.76)	(3.36)	(3.37)	(3.61)	(4.01)	(3.54)
9	Sell	0.0077	0.0065	0.0071	0.0082	0.0058	0.0058	0.0066	0.0078
		(1.47)	(1.29)	(1.43)	(1.66)	(1.13)	(1.15)	(1.34)	(1.59)
9	Buy	0.0186	0.0186	0.0176	0.0164	0.0193	0.0188	0.0176	0.0164
		(4.56)	(4.53)	(4.30)	(4.03)	(4.72)	(4.56)	(4.30)	(4.04)
9	Buy-sell	0.0109	0.0121	0.0105	0.0082	0.0135	0.0130	0.0109	0.0085
		(3.03)	(3.78)	(3.47)	(2.89)	(3.85)	(4.09)	(3.67)	(3.04)
12	Sell	0.0060	0.0065	0.0075	0.0087	0.0048	0.0058	0.0070	0.0085
		(1.17)	(1.29)	(1.48)	(1.74)	(0.93)	(1.15)	(1.40)	(1.71)
12	Buy	0.0192	0.0179	0.0168	0.0155	0.0196	0.0179	0.0167	0.0154
		(4.63)	(4.36)	(4.10)	(3.81)	(4.73)	(4.36)	(4.09)	(3.79)
12	Buy-sell	0.0131	0.0114	0.0093	0.0068	0.0149	0.0121	0.0096	0.0069
		(3.74)	(3.40)	(2.95)	(2.25)	(4.28)	(3.65)	(3.09)	(2.31)

Appendix G: Explanation of the loadTAQ() function

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

 Just type the following command to see its usage
 loadTAQ
> loadTAQ
function(n){
" Objective: load various TAQ (TORQ, DTAQ) data sets

   n   Description      Name of R data set
  ---  -------------    -----------
   1   ct30_20041101      ct_
   2   cq30_20041101      cq_
   3   TORQct             ct_
   4   TORQcq             cq_
   5   TORQcod            cod_
   6   DTAQct             ct_
   7   taqInfo200411      taqInfo

Example 1> # assume data sets under c:/temp
         >setwd('c:/temp/')
         >loadTAQ(1)
         >head(ct_)
            SYMBOL     DATE    TIME PRICE  SIZE G127 CORR COND EX   TSEQ
          1      A 20041101 9:30:18 24.95 19600   40    0       N 220946
          2      A 20041101 9:30:19 24.95   400    0    0       M      0
          3      A 20041101 9:30:19 24.95   100    0    0       M      0
          4      A 20041101 9:30:19 24.95   200    0    0       M      0
          5      A 20041101 9:30:22 24.95   200    0    0       B      0
          6      A 20041101 9:30:22 24.95   400    0    0       B      0

";loadTAQ_(n)
}
> setwd("c:/temp")
> loadTAQ(1)
> head(ct_,3)
  SYMBOL     DATE    TIME PRICE  SIZE G127 CORR COND EX   TSEQ
1      A 20041101 9:30:18 24.95 19600   40    0       N 220946
2      A 20041101 9:30:19 24.95   400    0    0       M      0
3      A 20041101 9:30:19 24.95   100    0    0       M      0
> tail(ct_,3)
       SYMBOL     DATE     TIME PRICE SIZE G127 CORR COND EX  TSEQ
221171     ZL 20041101 15:55:24  2.78  300   40    0       N 31079
221172     ZL 20041101 15:56:02  2.75  100    0    0       P     0
221173     ZL 20041101 15:56:02  2.75  100    0    0       T     0
>
```

Note: Appendices related to *loadCOMP()* is omitted for brevity.