CRSP for Teaching

Yuxing Yan
Canisius College

Most finance professors, at least in the North America, are quite familiar with the CRSP database, which contains historical trading data for all American listed stocks. However, few use it in their classrooms for three reasons: it is difficult to retrieve data; time consuming to explain and does not have recently available data, i.e. for yesterday or last month. In this paper, I present an effective way to solve those shortcomings. Two words could be used to describe my method: "trivial" and "fast". "Trivial" means that the method is so simple that it needs just a one-page long instruction. "Fast" means that loading a data set takes just a few seconds. For example, it takes just 5 seconds to load CRSP monthly data (which contains 32,676 stocks over 92 years). Another advantage of my approach is that we could retrieve IBM's data from 1926 up to yesterday by combining CRSP with Yahoo!Finance. Over the last 5 years, more than 50 schools have launched Business Analytics/Data Analytics programs. For those programs, CRSP is an ideal database they should use for machine learning, data visualization, cluster analysis, portfolio optimization and bankruptcy prediction.

INTRODUCTION

CRSP stands for "the Center for Research in Security Prices". It contains all trading data, such as closing price, trading volume, and shares outstanding, for all listed stocks in the US from 1926 and onward. Because of its quality and long history, it has been used intensively by academic researchers and practitioners. For example, Jegadeesh and Titman (1993) use the CRSP database for their famous momentum trading strategy. Another example is the famous Fama-French 3 factor model, where Fama and French (1992, 2993) use CRSP plus another accounting database to construct their factor series. Thus, almost all finance professors, at least in the US, are quite familiar with this database. It seems natural that a finance professor would use it for his/her teaching. However, CRSP was seldom used in our classrooms and I use myself as an example. Over the years, I have been teaching various finance courses at 8 schools, such as McGill and Wilfrid Laurier University (in Canada), Nanyang Technological University (NTU, in Singapore), Loyola University Maryland, UMUC, Hofstra University, University at Buffalo and Canisius College (in the US). In addition, I am an expert on financial databases including CRSP since I had been a consultant for 8 years to help doctoral students and other researchers, and I had answered thousands of questions related to CRSP and other financial databases. Despite this, among my two dozen finance courses, I have only used CRSP for two courses. The first course is called "Introduction to Financial Databases" taught at NTU. The second course is called "Financial Analysis with R" taught at University at Buffalo.

It is a trend that many schools have launched Business Analytics, such as Canisius College, University at Buffalo and University of San Francisco, just name a few. For the courses related to data inputting, manipulation, output and the like, offered by those programs, CRSP is an ideal database that instructors could use. The obvious motivation is that those schools might have a valid subscription already. There are several reasons why CRSP is seldom used in our classrooms.

Firstly, the CRSP database is not user friendly in terms of retrieving. The software used to retrieve data is called CRSP SIFT which is very difficult to use with a 65-page manual. Secondly, it does not have the latest observations. For example, today is 11/7/2018. The last trading day from the annual CRSP subscription is December 31, 2017. This gap increases until the next update. Thirdly, the design of the software used to retrieve data from CRSP is really outdated. It ignores the characteristics of our current students. Assume that we give a new cell phone to each of our students. Do we expect our students to read the manual in order to use it? For this generation of students, there is a good chance that no student should look for a manual. In a sense, my new approach matches the characteristics of this generation quite well: letting them explore to understand.

Over the years, I have constantly polished my method to retrieve various data sets based on the CRSP database. Guess how many pages of my introduction distributed to my students? Just one (see Appendix A). The paper is organized in the following way. The next section shows how to download and install R, the free software used. Since CRSP is a proprietary database, it is a good idea to use other publicly available data sets as an illustration. Since the Fama-French data is used quite intensively, I use the data sets I generated from Prof. French's Data Library as an example. The CRSP data sets work the same way as the Fama-French data, see section 7 for more detail.

Section 3 uses two simple examples showing how easily to load and use CRSP R data sets. Section 4 lists all the functions associated with using the CRSP database. In sections 5 and 6, functions `.dumpCRSP()` and `.loadCRSP()` will be discussed. It is always my firm belief that an easy way to download or upload CRSP data sets is the first step to use the database effectively. In the paper, it will be shown that loading any CRSP data set is really trivial. Since CRSP database is a proprietary product, the Fama-French data sets will be used as an illustration showing in section 7. Second 8 offers more explanations for the CRSP database. Sections 9, 10 and 11 show potential applications of the CRSP database by using Excel for undergraduate students or MBA students, R and Python for graduate students, respectively. The last section concludes the paper.

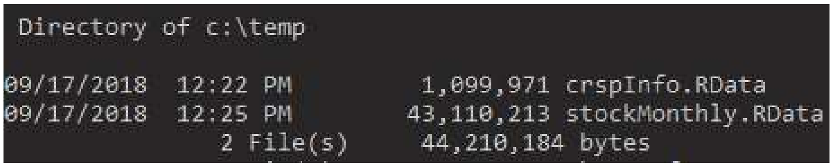
R INSTALLATION

To install R, we have the following procedure.

```
1: Go to http://www.r-project.org
2: Click "CRAN" under "Download,Packages" (left-hand side)
3: Choose a mirror address
4: Choose appropriate software (PC, Mac)
5: Click "base"
```

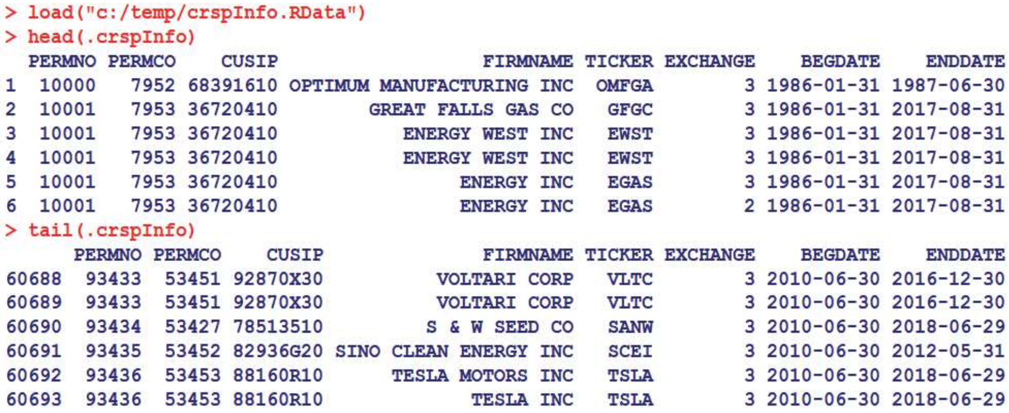
TWO EXAMPLES FOR TWO CRSP DATA SETS

It is a good idea to show how to use one or two R data sets. For this case, we show how to upload and explore two R data sets called `crspInfo` and `stockMonthly`. Assume that those two data sets are located under c:/temp, shown below.



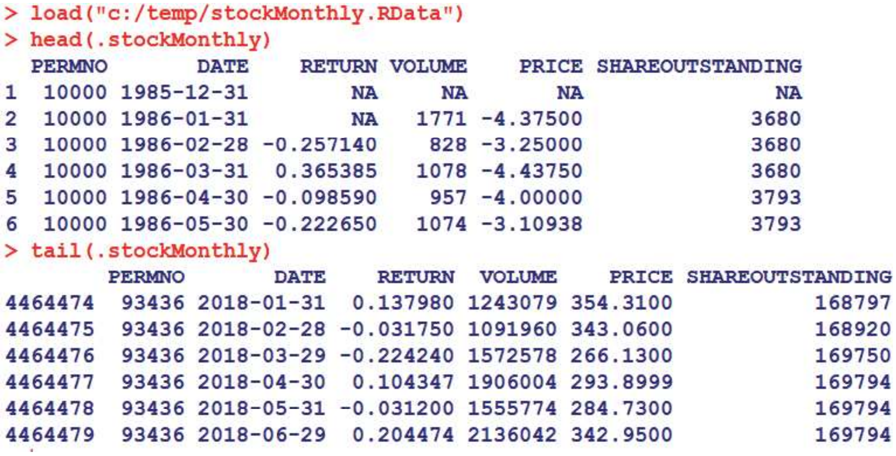
```
Directory of c:\temp

09/17/2018  12:22 PM         1,099,971 crspInfo.RData
09/17/2018  12:25 PM        43,110,213 stockMonthly.RData
               2 File(s)     44,210,184 bytes
```

The size of `crspInfo.RData` is about 1.1 M (mega bite) while it is 43.1M for the monthly stock data set. The R function `load()` can be used to upload them. In addition, the `head()`, `tail()` and `dim()` functions can be used to view the first and last several lines plus the dimensions of the datasets, shown below.



```
> load("c:/temp/crspInfo.RData")
> head(.crspInfo)
  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        GREAT FALLS GAS CO   GFGC        3 1986-01-31 2017-08-31
3  10001   7953 36720410           ENERGY WEST INC   EWST        3 1986-01-31 2017-08-31
4  10001   7953 36720410           ENERGY WEST INC   EWST        3 1986-01-31 2017-08-31
5  10001   7953 36720410                ENERGY INC   EGAS        3 1986-01-31 2017-08-31
6  10001   7953 36720410                ENERGY INC   EGAS        2 1986-01-31 2017-08-31
> tail(.crspInfo)
      PERMNO PERMCO    CUSIP                  FIRMNAME TICKER EXCHANGE    BEGDATE    ENDDATE
60688  93433  53451 92870X30              VOLTARI CORP   VLTC        3 2010-06-30 2016-12-30
60689  93433  53451 92870X30              VOLTARI CORP   VLTC        3 2010-06-30 2016-12-30
60690  93434  53427 78513510             S & W SEED CO   SANW        3 2010-06-30 2018-06-29
60691  93435  53452 82936G20 SINO CLEAN ENERGY INC   SCEI        3 2010-06-30 2012-05-31
60692  93436  53453 88160R10         TESLA MOTORS INC   TSLA        3 2010-06-30 2018-06-29
60693  93436  53453 88160R10                TESLA INC   TSLA        3 2010-06-30 2018-06-29
```

In the above output, the *PERMNO* is the CRSP's stock *ID*, *PERMCO* is firm ID. In total, there are 31,599 observations.



```
> load("c:/temp/stockMonthly.RData")
> head(.stockMonthly)
  PERMNO       DATE    RETURN VOLUME    PRICE SHAREOUTSTANDING
1  10000 1985-12-31        NA     NA       NA               NA
2  10000 1986-01-31        NA   1771 -4.37500             3680
3  10000 1986-02-28 -0.257140    828 -3.25000             3680
4  10000 1986-03-31  0.365385   1078 -4.43750             3680
5  10000 1986-04-30 -0.098590    957 -4.00000             3793
6  10000 1986-05-30 -0.222650   1074 -3.10938             3793
> tail(.stockMonthly)
        PERMNO       DATE    RETURN  VOLUME    PRICE SHAREOUTSTANDING
4464474  93436 2018-01-31  0.137980 1243079 354.3100           168797
4464475  93436 2018-02-28 -0.031750 1091960 343.0600           168920
4464476  93436 2018-03-29 -0.224240 1572578 266.1300           169750
4464477  93436 2018-04-30  0.104347 1906004 293.8999           169794
4464478  93436 2018-05-31 -0.031200 1555774 284.7300           169794
4464479  93436 2018-06-29  0.204474 2136042 342.9500           169794
```

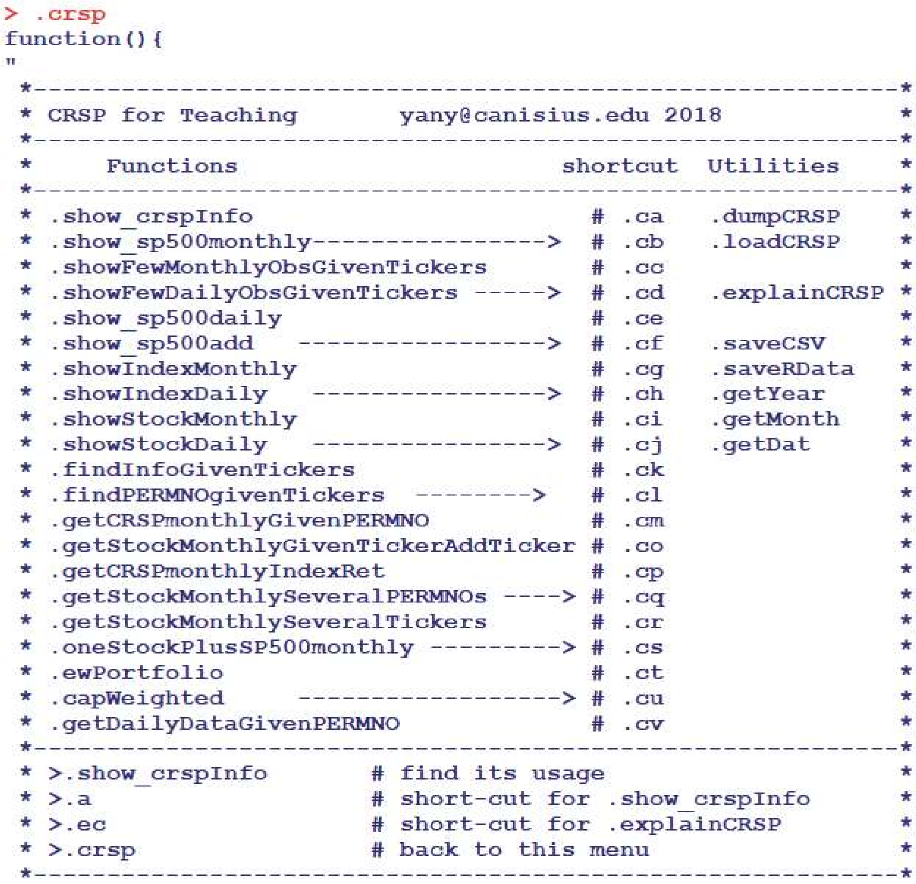
The above method is rather simple. Just imagine how long a student, who has never used R before and never even heard about CRSP, could load and explore those two R data sets? Our guess is that it might at most 20 minutes: 5 minutes to download R, 5 minutes to launch and quite R; 2 minutes to copy the data sets to a subdirectory; 8 minutes to explore those two major CRSP data sets. Later, we show that many functions were written to help students explore various data sets, such as retrieving header information for a given ticker or company name.

ALL FUNCTIONS RELATED TO CRSP

It is a good idea to show potential readers how trivial it is to retrieve data from the CRSP. After launching R, an instructor can just give students one-line R codes. An example is given below. Note that R is case sensitive.

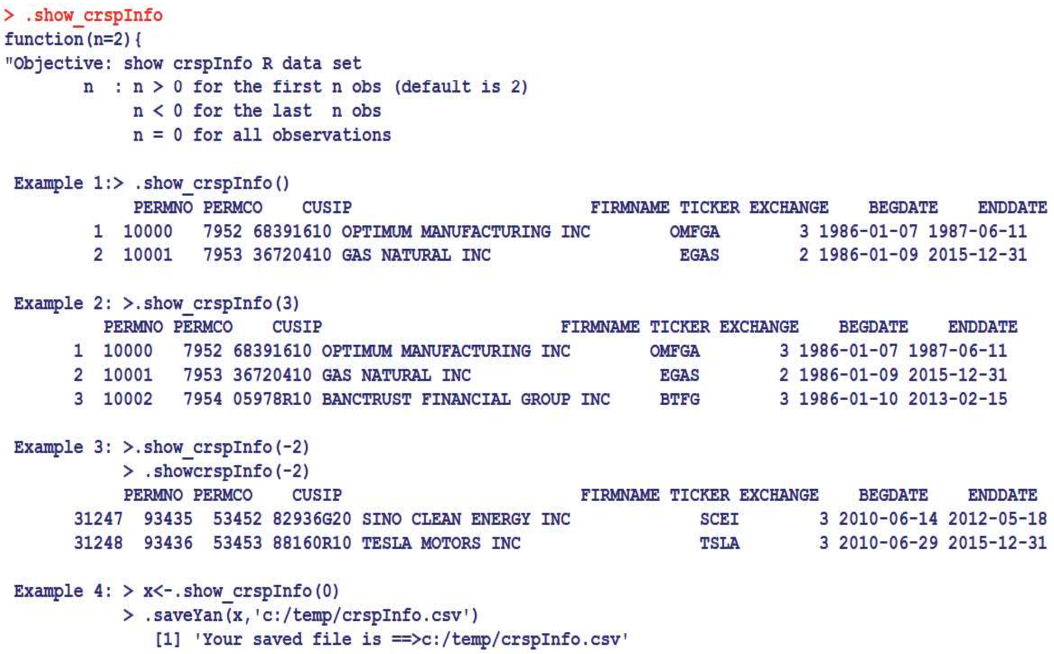
```
source(“http://datayyy.com/crsp.R”)
```

After inputting the above code, the following window pops up:



```
> .crsp
function(){
"
 *-----------------------------------------------------------*
 * CRSP for Teaching       yany@canisius.edu 2018             *
 *-----------------------------------------------------------*
 *     Functions                      shortcut  Utilities    *
 *-----------------------------------------------------------*
 * .show_crspInfo                      # .ca   .dumpCRSP    *
 * .show_sp500monthly---------------->  # .cb   .loadCRSP    *
 * .showFewMonthlyObsGivenTickers       # .cc                *
 * .showFewDailyObsGivenTickers ----->  # .cd   .explainCRSP *
 * .show_sp500daily                     # .ce                *
 * .show_sp500add   ----------------->  # .cf   .saveCSV     *
 * .showIndexMonthly                    # .cg   .saveRData   *
 * .showIndexDaily   ---------------->  # .ch   .getYear     *
 * .showStockMonthly                    # .ci   .getMonth    *
 * .showStockDaily   ---------------->  # .cj   .getDat      *
 * .findInfoGivenTickers                # .ck                *
 * .findPERMNOgivenTickers  -------->   # .cl                *
 * .getCRSPmonthlyGivenPERMNO           # .cm                *
 * .getStockMonthlyGivenTickerAddTicker # .co                *
 * .getCRSPmonthlyIndexRet              # .cp                *
 * .getStockMonthlySeveralPERMNOs ----> # .cq                *
 * .getStockMonthlySeveralTickers       # .cr                *
 * .oneStockPlusSP500monthly ---------> # .cs                *
 * .ewPortfolio                         # .ct                *
 * .capWeighted     ------------------> # .cu                *
 * .getDailyDataGivenPERMNO             # .cv                *
 *-----------------------------------------------------------*
 * >.show_crspInfo       # find its usage                     *
 * >.a                   # short-cut for .show_crspInfo       *
 * >.ec                  # short-cut for .explainCRSP         *
 * >.crsp                # back to this menu                  *
 *-----------------------------------------------------------*
```

Every function is self-explanatory. Thus, instructors don't need to spend any time to explain its usage to his/her students. By typing the name of any function, users can see its purpose, input variables with a few examples. For example, typing `.show_crspInfo` leads to the following info.

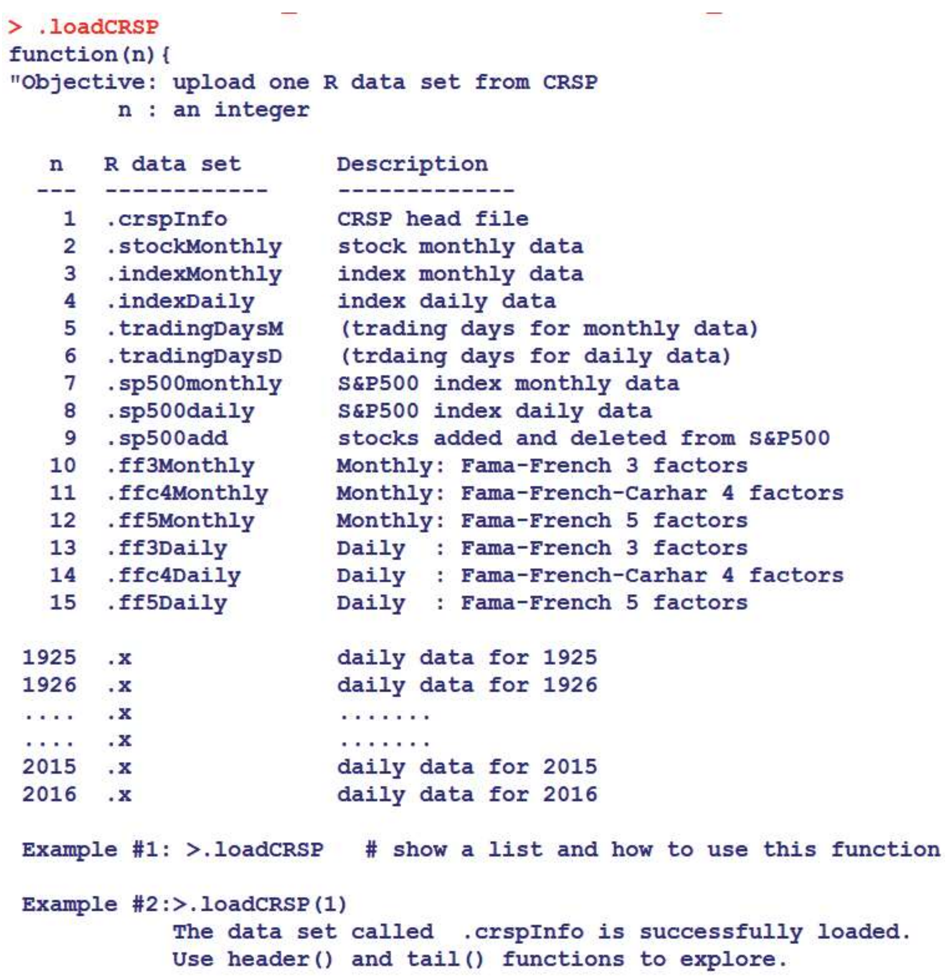


```
> .show_crspInfo
function(n=2){
"Objective: show crspInfo R data set
        n  : n > 0 for the first n obs (default is 2)
             n < 0 for the last  n obs
             n = 0 for all observations

 Example 1:> .show_crspInfo()
             PERMNO PERMCO    CUSIP                         FIRMNAME TICKER EXCHANGE    BEGDATE    ENDDATE
         1  10000   7952 68391610 OPTIMUM MANUFACTURING INC        OMFGA        3 1986-01-07 1987-06-11
         2  10001   7953 36720410 GAS NATURAL INC                   EGAS        2 1986-01-09 2015-12-31

 Example 2: >.show_crspInfo(3)
          PERMNO PERMCO    CUSIP                         FIRMNAME TICKER EXCHANGE    BEGDATE    ENDDATE
       1  10000   7952 68391610 OPTIMUM MANUFACTURING INC        OMFGA        3 1986-01-07 1987-06-11
       2  10001   7953 36720410 GAS NATURAL INC                   EGAS        2 1986-01-09 2015-12-31
       3  10002   7954 05978R10 BANCTRUST FINANCIAL GROUP INC     BTFG        3 1986-01-10 2013-02-15

 Example 3: >.show_crspInfo(-2)
            > .showcrspInfo(-2)
            PERMNO PERMCO    CUSIP                         FIRMNAME TICKER EXCHANGE    BEGDATE    ENDDATE
       31247  93435  53452 82936G20 SINO CLEAN ENERGY INC            SCEI        3 2010-06-14 2012-05-18
       31248  93436  53453 88160R10 TESLA MOTORS INC                 TSLA        3 2010-06-29 2015-12-31

 Example 4: > x<-.show_crspInfo(0)
            > .saveYan(x,'c:/temp/crspInfo.csv')
               [1] 'Your saved file is ==>c:/temp/crspInfo.csv'
```

In the following sections, we will introduce several most important functions such as `.loadCRSP`, `.explainCRSP` and `.saveCRSPcsv`. To make each function more meaningful, we adopt long names, such as `.showFewMonthlyBosGivenTickers`. However, to make students' lives a bit easier, we have generated many short-cuts (see the list after we type `.crsp`). For example, .cc is equivalent to `.showFewMonthlyBosGivenTickers`.

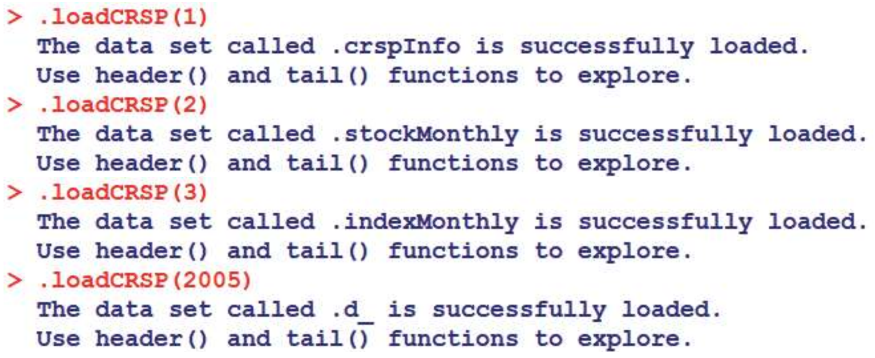
Loading CRSP Data Set Using .loadCRSP

Loading CRSP data sets separately or manually is quite time consuming since we have around 200 data sets. Another issue is how to type those names especially for CRSP daily data set. Another potential issue is the memory issues. For this purpose, we have generated a function called `.loadCRSP()` which makes our task rather simple. By typing its name, we learn how to use the function, its input and learn a few examples (see the following image).



```
> .loadCRSP
function(n){
"Objective: upload one R data set from CRSP
        n : an integer

   n   R data set        Description
  ---  ------------      -------------
    1  .crspInfo         CRSP head file
    2  .stockMonthly     stock monthly data
    3  .indexMonthly     index monthly data
    4  .indexDaily       index daily data
    5  .tradingDaysM     (trading days for monthly data)
    6  .tradingDaysD     (trdaing days for daily data)
    7  .sp500monthly     S&P500 index monthly data
    8  .sp500daily       S&P500 index daily data
    9  .sp500add         stocks added and deleted from S&P500
   10  .ff3Monthly       Monthly: Fama-French 3 factors
   11  .ffc4Monthly      Monthly: Fama-French-Carhar 4 factors
   12  .ff5Monthly       Monthly: Fama-French 5 factors
   13  .ff3Daily         Daily  : Fama-French 3 factors
   14  .ffc4Daily        Daily  : Fama-French-Carhar 4 factors
   15  .ff5Daily         Daily  : Fama-French 5 factors

 1925  .x                daily data for 1925
 1926  .x                daily data for 1926
 ....  .x                .......
 ....  .x                .......
 2015  .x                daily data for 2015
 2016  .x                daily data for 2016

 Example #1: >.loadCRSP   # show a list and how to use this function

 Example #2:>.loadCRSP(1)
            The data set called  .crspInfo is successfully loaded.
            Use header() and tail() functions to explore.
```

For example, if we want to load `.crspInfo, .stockMonthly, .indexMonthly` plus daily data in 2005, we simply issue `.loadCRSP(1)`, `.loadCRSP(2)`, `.loadCRSP(3)` and `.loadCRSP(2005)` , shown below.

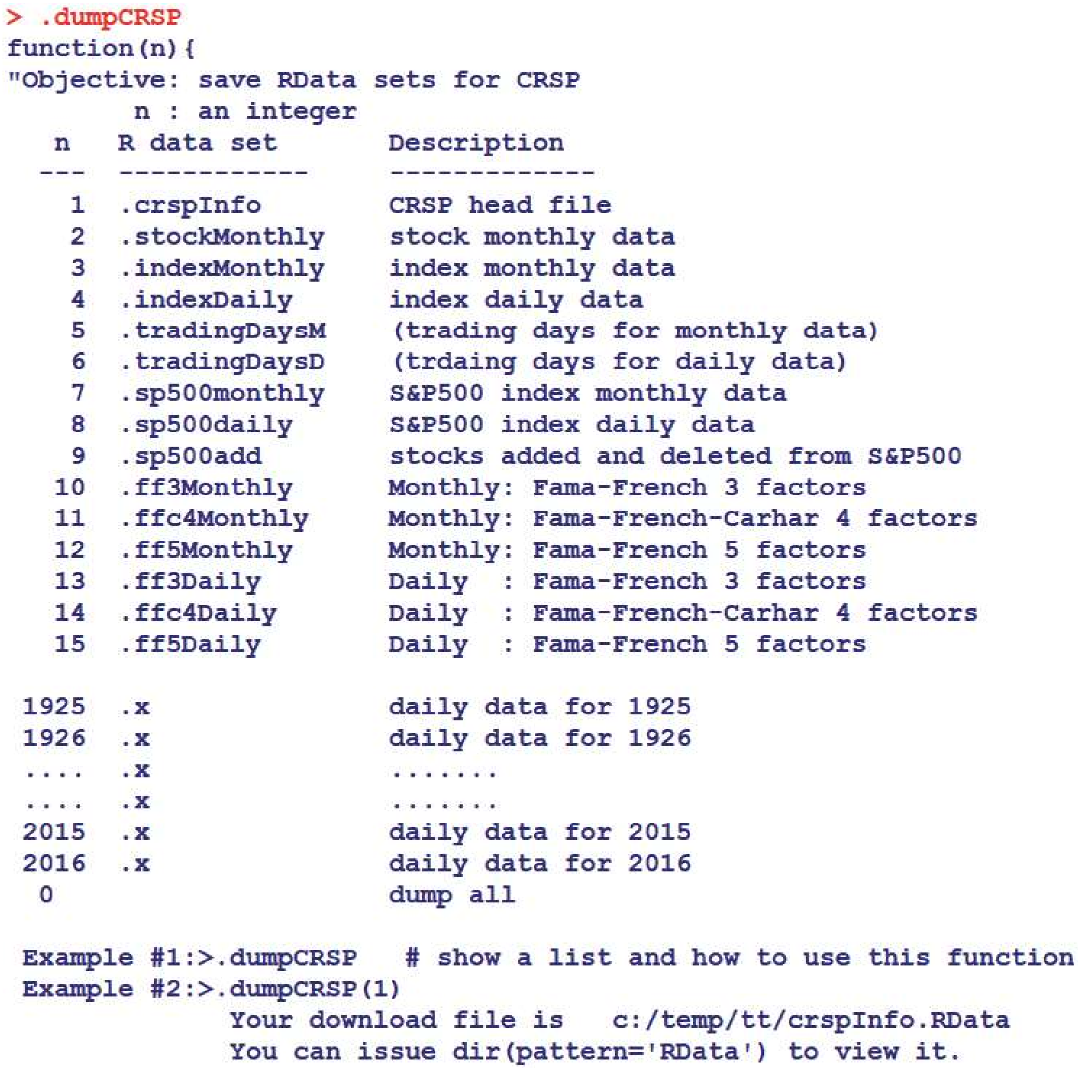


```
> .loadCRSP(1)
  The data set called .crspInfo is successfully loaded.
  Use header() and tail() functions to explore.
> .loadCRSP(2)
  The data set called .stockMonthly is successfully loaded.
  Use header() and tail() functions to explore.
> .loadCRSP(3)
  The data set called .indexMonthly is successfully loaded.
  Use header() and tail() functions to explore.
> .loadCRSP(2005)
  The data set called .d_ is successfully loaded.
  Use header() and tail() functions to explore.
```

Again, since the CRSP is a proprietary database, I cannot post them publicly. Below, we show a similar loading function called `.loadFF` which loads around two dozen R data sets, generated based on the data downloaded from the Prof. French Data Library.

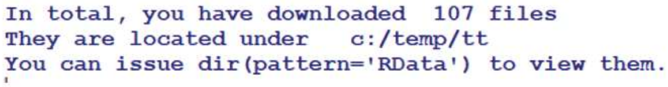
Dumping CRSP Data Set Using .dumpCRSP

Many times, students prefer to download one, a few or all R data sets for further exploration. For this purpose, `.dumpCRSP` is designed to achieve this task.



```
> .dumpCRSP
function(n){
"Objective: save RData sets for CRSP
        n : an integer
   n   R data set       Description
  ---  ------------     -------------
    1  .crspInfo        CRSP head file
    2  .stockMonthly    stock monthly data
    3  .indexMonthly    index monthly data
    4  .indexDaily      index daily data
    5  .tradingDaysM    (trading days for monthly data)
    6  .tradingDaysD    (trdaing days for daily data)
    7  .sp500monthly    S&P500 index monthly data
    8  .sp500daily      S&P500 index daily data
    9  .sp500add        stocks added and deleted from S&P500
   10  .ff3Monthly      Monthly: Fama-French 3 factors
   11  .ffc4Monthly     Monthly: Fama-French-Carhar 4 factors
   12  .ff5Monthly      Monthly: Fama-French 5 factors
   13  .ff3Daily        Daily  : Fama-French 3 factors
   14  .ffc4Daily       Daily  : Fama-French-Carhar 4 factors
   15  .ff5Daily        Daily  : Fama-French 5 factors

 1925  .x               daily data for 1925
 1926  .x               daily data for 1926
 ....  .x               .......
 ....  .x               .......
 2015  .x               daily data for 2015
 2016  .x               daily data for 2016
  0                     dump all

 Example #1:>.dumpCRSP   # show a list and how to use this function
 Example #2:>.dumpCRSP(1)
              Your download file is   c:/temp/tt/crspInfo.RData
              You can issue dir(pattern='RData') to view it.
```

If the input value is 0, then we would download all R data sets related to CRSP.



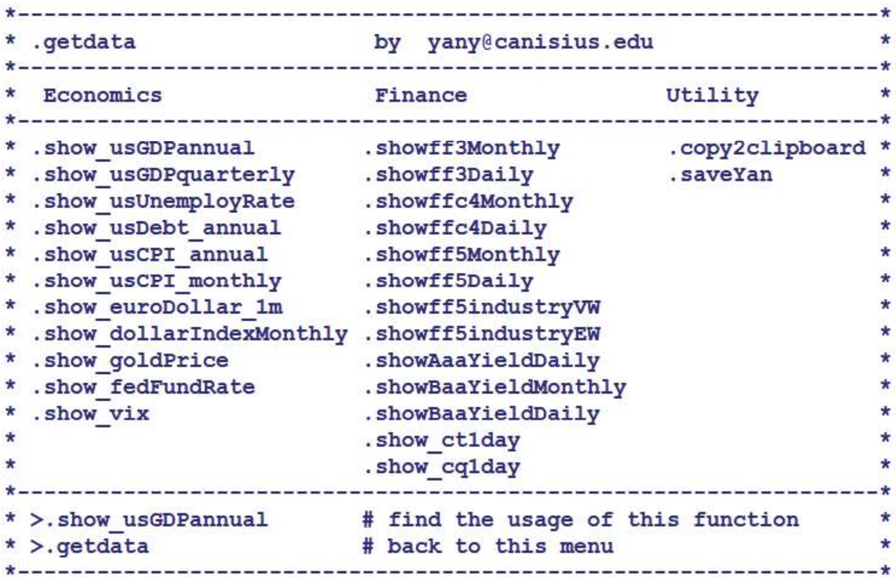
```
In total, you have downloaded  107 files
They are located under   c:/temp/tt
You can issue dir(pattern='RData') to view them.
```

LOADING FAMA-FRENCH DATA SETS EFFICIENTLY: .loadFF FUNCTION

In this section, I show how to load various Fama-French data sets efficiently. First, type the following R code.

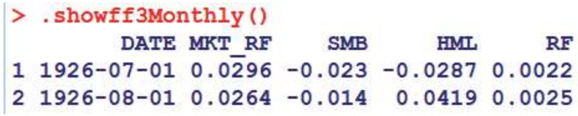
```
source('http://datayyy.com/getdata.R')
```

After hitting the return key, we see the instruction.



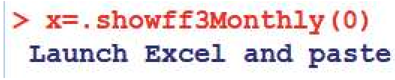
```
*-------------------------------------------------------------------*
* .getdata                  by  yany@canisius.edu                   *
*-------------------------------------------------------------------*
*  Economics                 Finance               Utility          *
*-------------------------------------------------------------------*
* .show_usGDPannual         .showff3Monthly        .copy2clipboard *
* .show_usGDPquarterly      .showff3Daily          .saveYan         *
* .show_usUnemployRate      .showffc4Monthly                        *
* .show_usDebt_annual       .showffc4Daily                          *
* .show_usCPI_annual        .showff5Monthly                         *
* .show_usCPI_monthly       .showff5Daily                           *
* .show_euroDollar_1m       .showff5industryVW                      *
* .show_dollarIndexMonthly .showff5industryEW                       *
* .show_goldPrice           .showAaaYieldDaily                      *
* .show_fedFundRate         .showBaaYieldMonthly                    *
* .show_vix                 .showBaaYieldDaily                      *
*                           .show_ct1day                            *
*                           .show_cq1day                            *
*-------------------------------------------------------------------*
* >.show_usGDPannual        # find the usage of this function      *
* >.getdata                 # back to this menu                     *
*-------------------------------------------------------------------*
```

To view the first 2 lines for the Fama-French monthly 3- factors, we issue `.showff3Monthly()`, shown below.



```
> .showff3Monthly()
        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
```

To retrieve all Fama-French 3 factor data, we enter 0 for its input value, shown below.

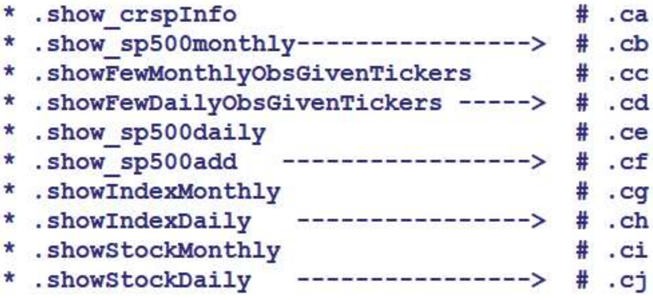


```
> x=.showff3Monthly(0)
 Launch Excel and paste
```

MORE ABOUT CRSP

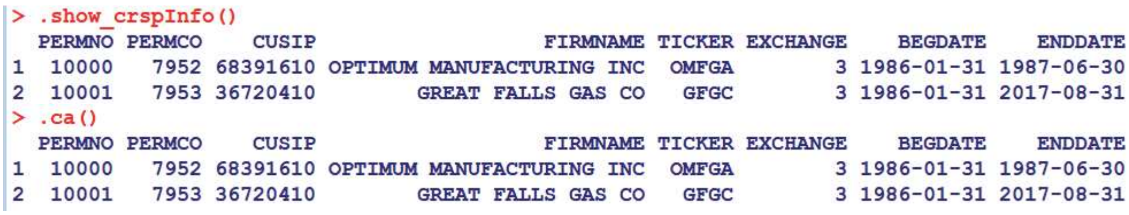
Seeing Is Believing

To help students understand CRSP, we have generated about a dozen functions staring with `.show`. The sole objective of those functions is to allow students to explore those data sets.



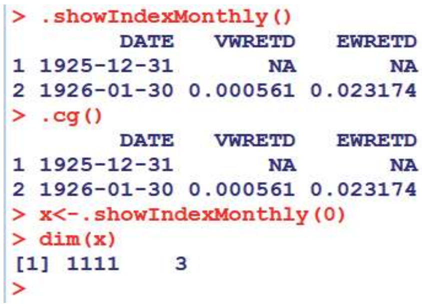
```
* .show_crspInfo                          # .ca
* .show_sp500monthly----------------->   # .cb
* .showFewMonthlyObsGivenTickers          # .cc
* .showFewDailyObsGivenTickers ----->    # .cd
* .show_sp500daily                        # .ce
* .show_sp500add   ----------------->    # .cf
* .showIndexMonthly                       # .cg
* .showIndexDaily   ---------------->    # .ch
* .showStockMonthly                       # .ci
* .showStockDaily   ---------------->    # .cj
```

The structure of those functions are the same with just one input variable n. If n is positive, then we show the first n observations of the data set. For a negative n, we show the last n observations. If n is zero, we show the whole data set. Usually, taking n=0 is for outputting the whole data set. Below is an example related to the `.show_crspInfo` function.



```
> .show_crspInfo()
  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          GREAT FALLS GAS CO   GFGC        3 1986-01-31 2017-08-31
> .ca()
  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          GREAT FALLS GAS CO   GFGC        3 1986-01-31 2017-08-31
```

Note that `.ca` is the short-cut of this function. Below is another example.

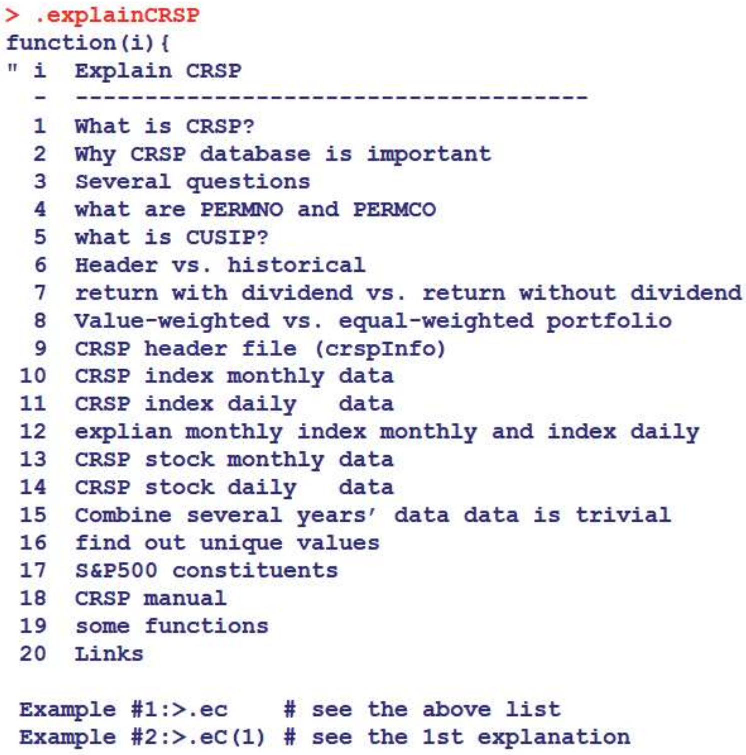


```
> .showIndexMonthly()
        DATE   VWRETD   EWRETD
1 1925-12-31       NA       NA
2 1926-01-30 0.000561 0.023174
> .cg()
        DATE   VWRETD   EWRETD
1 1925-12-31       NA       NA
2 1926-01-30 0.000561 0.023174
> x<-.showIndexMonthly(0)
> dim(x)
[1] 1111    3
>
```

We will discuss this data set further in the section 10 called “Exploring CRSP data sets further”.

Explain CRSP (`.explainCRSP`)

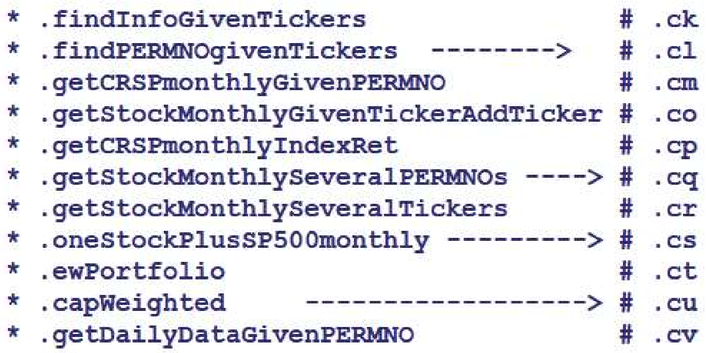
To help students understand better, we have generated a set of Q&As, through the function of `.explainCRSP`. Again, we simply type the function name without any input value (see the image below).



```
> .explainCRSP
function(i){
" i  Explain CRSP
  -  -------------------------------------
  1  What is CRSP?
  2  Why CRSP database is important
  3  Several questions
  4  what are PERMNO and PERMCO
  5  what is CUSIP?
  6  Header vs. historical
  7  return with dividend vs. return without dividend
  8  Value-weighted vs. equal-weighted portfolio
  9  CRSP header file (crspInfo)
 10  CRSP index monthly data
 11  CRSP index daily   data
 12  explian monthly index monthly and index daily
 13  CRSP stock monthly data
 14  CRSP stock daily   data
 15  Combine several years' data data is trivial
 16  find out unique values
 17  S&P500 constituents
 18  CRSP manual
 19  some functions
 20  Links

 Example #1:>.ec    # see the above list
 Example #2:>.eC(1) # see the 1st explanation
```

Exploring CRSP Further

In order to understand the CRSP database better, students can explore the data sets according to their own needs.

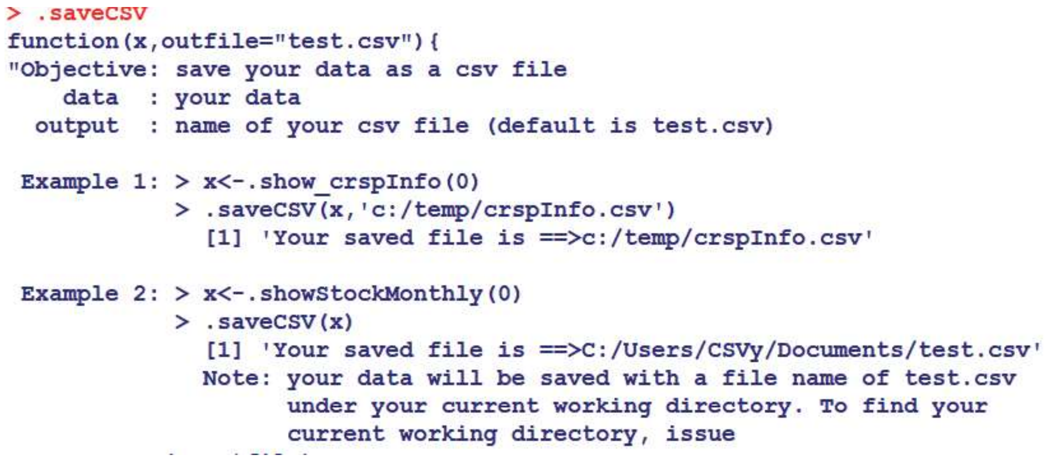


```
* .findInfoGivenTickers                  # .ck
* .findPERMNOgivenTickers  -------->     # .cl
* .getCRSPmonthlyGivenPERMNO             # .cm
* .getStockMonthlyGivenTickerAddTicker # .co
* .getCRSPmonthlyIndexRet                # .cp
* .getStockMonthlySeveralPERMNOs ----> # .cq
* .getStockMonthlySeveralTickers         # .cr
* .oneStockPlusSP500monthly ---------> # .cs
* .ewPortfolio                           # .ct
* .capWeighted     -----------------> # .cu
* .getDailyDataGivenPERMNO               # .cv
```

Of course, we could supply more functions to explore the CRSP database.

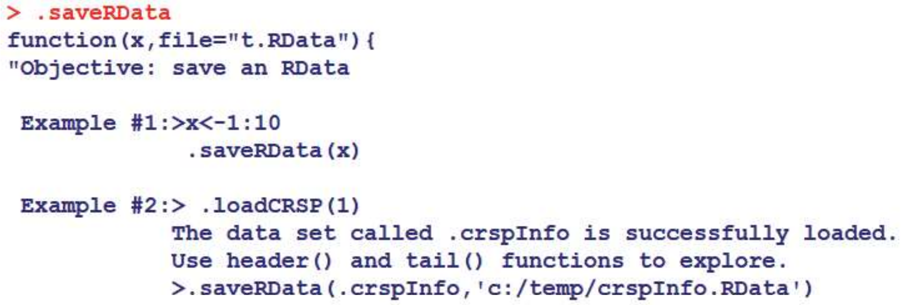
Saving CRSP Individual Data Sets

Different business analytics, data analytics or data science might adopt different programming languages, such as R, Python, Excel, Octave or even SAS. For this purpose, we generate several useful tools (functions). The objective of `.saveCSV()` function is to save CRSP data set as a CSV (Comma Separated Value) format, to be used by Excel or other software for further analysis. Again, to know its usage, simply type its name (see the image below).



```
> .saveCSV
function(x,outfile="test.csv"){
"Objective: save your data as a csv file
    data   : your data
  output   : name of your csv file (default is test.csv)

 Example 1: > x<-.show_crspInfo(0)
            > .saveCSV(x,'c:/temp/crspInfo.csv')
              [1] 'Your saved file is ==>c:/temp/crspInfo.csv'

 Example 2: > x<-.showStockMonthly(0)
            > .saveCSV(x)
              [1] 'Your saved file is ==>C:/Users/CSVy/Documents/test.csv'
             Note: your data will be saved with a file name of test.csv
                   under your current working directory. To find your
                   current working directory, issue
```

Obviously, `.saveRData` is to save an R dataset. Below are a few examples to download several R data sets.



```
> .saveRData
function(x,file="t.RData"){
"Objective: save an RData

 Example #1:>x<-1:10
             .saveRData(x)

 Example #2:> .loadCRSP(1)
            The data set called .crspInfo is successfully loaded.
            Use header() and tail() functions to explore.
            >.saveRData(.crspInfo,'c:/temp/crspInfo.RData')
```

CRSP FOR EXCEL

For most undergraduate students, Excel is probably the most widely used computation tool for finance, accounting and economics at various business schools. There is a good chance it is used for Corporate Finance, Financial Modeling, Portfolio Theory and Investments. From the previous discussion, we know that we could easily download and save CRSP monthly or daily stock and index data. Below, one of my numerous data cases is used to show how to use the CRSP index data. For example, I have designed a few data cases for my undergraduate students. For data case is to test which party, Democratic or Republican, could manage the economy better. The market indices used are EVWRTD (Equal-weighted market index with dividend) and VWRETD (value-weighted market index with dividend) which are come from the CRSP database.

CRSP FOR R PROGRAMMING

Since CRSP is the proprietary database, we use other public data sets to show how easily to download various data sets easily by using R. The first data set is related to the Fama-French 3 factor time series. The following R code shows how to download from the author's website.

```
path<-"http://canisius.edu/~yany/RData/"
dataSet<-"ff3Monthly"
link<-paste(path,dataSet,".RData",sep='')
load(url(link))
head(.ff3Monthly)
        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
4 1926-10-01 -0.0324  0.0004  0.0051 0.0032
5 1926-11-01  0.0253 -0.0020 -0.0035 0.0031
6 1926-12-01  0.0262 -0.0004 -0.0002 0.0028
```

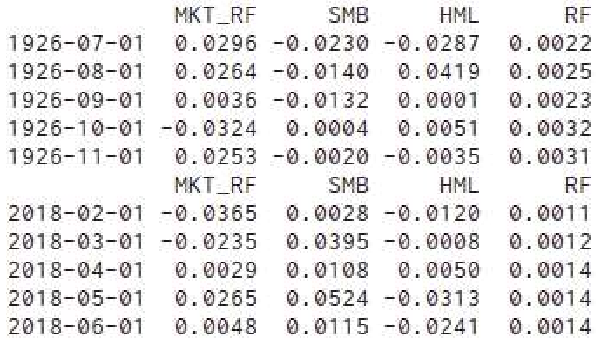
The original Fama-French 3-factor time series could be downloaded and updated from Prof. French's Data Library, at http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html.

CRSP FOR PYTHON PROGRAMMING

For most undergraduate students, Excel is probably the most widely used computation tool for most business analytics, data analytics or data science programs. Since the CRSP is a proprietary database, I use the Fama-French data set as an example. The following pickle data set could be downloaded at http://canisius.edu/~yany/python/ff3Monthly.pkl. We could use the following Python program to retrieve the data set sand print the first and last several lines.

```
import pandas as pd
x=pd.read_pickle("c:/temp/ff3Monthly.pkl")
print(x.head())
print(x.tail())
```

The related output is shown below.



```
             MKT_RF     SMB     HML     RF
1926-07-01  0.0296 -0.0230 -0.0287  0.0022
1926-08-01  0.0264 -0.0140  0.0419  0.0025
1926-09-01  0.0036 -0.0132  0.0001  0.0023
1926-10-01 -0.0324  0.0004  0.0051  0.0032
1926-11-01  0.0253 -0.0020 -0.0035  0.0031
             MKT_RF     SMB     HML     RF
2018-02-01 -0.0365  0.0028 -0.0120  0.0011
2018-03-01 -0.0235  0.0395 -0.0008  0.0012
2018-04-01  0.0029  0.0108  0.0050  0.0014
2018-05-01  0.0265  0.0524 -0.0313  0.0014
2018-06-01  0.0048  0.0115 -0.0241  0.0014
```

CONCLUSION

In this short paper, I show how to easily retrieve CRSP data for our undergraduate and graduate students. The only requirement is that your school has a valid CRSP subscription. Any student with no knowledge about CRSP is able to retrieve data from CRSP within 30 minutes of instruction as well as downloading and installing R. First, download and install R. Second, copy those CRSP related R data sets. Third, issue one-line R command to activate all related functions. In the fall 2015, I incorporate CRSP into my teaching. The title of the course is called "Financial Analysis with R", see Appendix B for my syllabus used at the University at Buffalo. In total, I have 10 groups for the term projects. It is not a surprise to find out that 9 of 10 groups choose a topic using CRSP. See Appendix C for a list of potential term projects.

ACKNOWLEDGEMENT

My contact information: Department of Economics and Finance, Canisius College, 2001 Main Street, Buffalo, NY 14208. Email: yany@canisius.edu. The paper was presented at the 2018 FMA annual conference. I thank the comments and suggestions offered by the conference participants, especially Gulnara R. Zaynutdinova for her valuable comments. In addition, I thank James Yan for his editorial help. Part of the work was done when I was teaching at Loyola University, Hofstra University and University at Buffalo.

REFERENCES

CRSP website http://crsp.com/

CRSP user manual (n.d.). Retrieved from http://www.crsp.com/documentation

Fama, E., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33, 3056.

Fama, E., & French, K. R. (1992). The cross-section of expected stock returns, *Journal of Finance*, 47, 427-465.

Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance*, 48, 65-91.

Yan, Y. (2016). *Financial Modeling using R.* Tate Publishing. ISBN: 978-1-68187-530-9. Retrieved from https://www.tatepublishing.com/bookstore/book.php?w=9781681875309

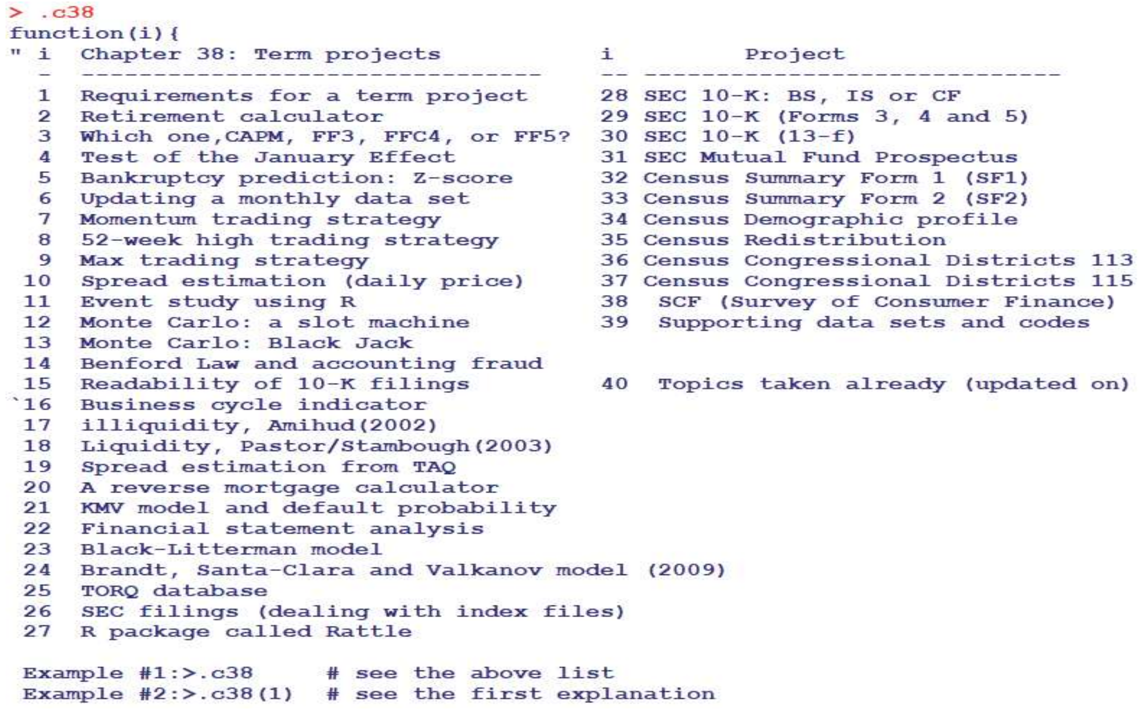
Yan, Y. (2017). Teaching programming skills to finance students: How to design and teach a great course? *Financial Innovation*, (forthcoming).

APPENDICES

Appendix A: A One-page Introduction Distributed to Students

To download and install R (free computational software), we have the following 5 steps.
Step 1: Go to http://www.r-project.org
Step 2: Click "CRAN" under "Download" (left-hand side)
Step 3: Choose a mirror address close to your location.
Step 4: Choose the appropriate software (PC, Mac)
Step 5: Click "base". For example, for Windows, we have the following result.

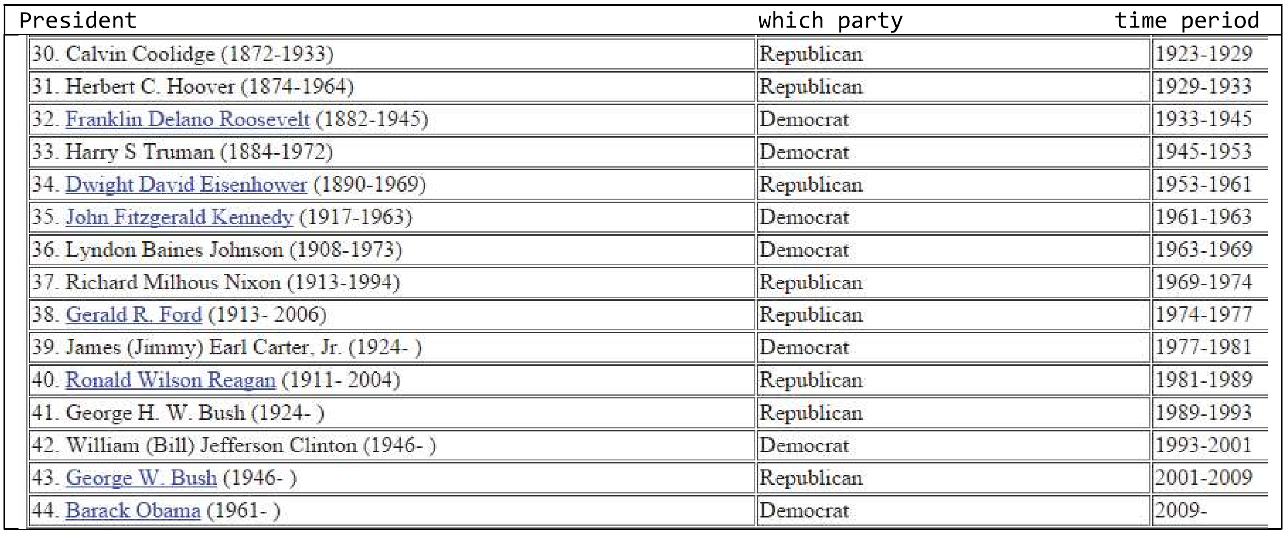
Appendix B: A List of Potential Topics for my Course “Financial Modeling Using R”



```
> .c38
function(i){
" i  Chapter 38: Term projects           i         Project
  -  --------------------------------    -- -----------------------------
  1  Requirements for a term project     28 SEC 10-K: BS, IS or CF
  2  Retirement calculator               29 SEC 10-K (Forms 3, 4 and 5)
  3  Which one,CAPM, FF3, FFC4, or FF5?  30 SEC 10-K (13-f)
  4  Test of the January Effect          31 SEC Mutual Fund Prospectus
  5  Bankruptcy prediction: Z-score      32 Census Summary Form 1 (SF1)
  6  Updating a monthly data set         33 Census Summary Form 2 (SF2)
  7  Momentum trading strategy           34 Census Demographic profile
  8  52-week high trading strategy       35 Census Redistribution
  9  Max trading strategy                36 Census Congressional Districts 113
 10  Spread estimation (daily price)     37 Census Congressional Districts 115
 11  Event study using R                 38  SCF (Survey of Consumer Finance)
 12  Monte Carlo: a slot machine         39  Supporting data sets and codes
 13  Monte Carlo: Black Jack
 14  Benford Law and accounting fraud
 15  Readability of 10-K filings         40  Topics taken already (updated on)
`16  Business cycle indicator
 17  illiquidity, Amihud(2002)
 18  Liquidity, Pastor/Stambough(2003)
 19  Spread estimation from TAQ
 20  A reverse mortgage calculator
 21  KMV model and default probability
 22  Financial statement analysis
 23  Black-Litterman model
 24  Brandt, Santa-Clara and Valkanov model (2009)
 25  TORQ database
 26  SEC filings (dealing with index files)
 27  R package called Rattle

 Example #1:>.c38     # see the above list
 Example #2:>.c38(1)  # see the first explanation
```

Appendix C: A Data Case for Using the CRSP Index Data

Which political party manages the economy better? According to the web page of http://www.enchantedlearning.com/history/us/pres/list.shtml, we could find which party the US presidents belong to.



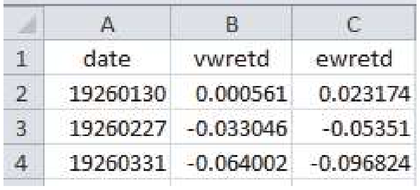
President	which party	time period
30. Calvin Coolidge (1872-1933)	Republican	1923-1929
31. Herbert C. Hoover (1874-1964)	Republican	1929-1933
32. Franklin Delano Roosevelt (1882-1945)	Democrat	1933-1945
33. Harry S Truman (1884-1972)	Democrat	1945-1953
34. Dwight David Eisenhower (1890-1969)	Republican	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 can generate the following table.

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

Step 1: Open an Excel file called which_party_data.xlsx which has 3 columns: date, `VWRETD` (value-weighted market index) and `EWRETD` (equal-weighted market index), see a few observations below.



	A	B	C
1	date	vwretd	ewretd
2	19260130	0.000561	0.023174
3	19260227	-0.033046	-0.05351
4	19260331	-0.064002	-0.096824

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

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

$$\bar{R}_{Democratic} = \bar{R}_{Republican} \tag{1}$$

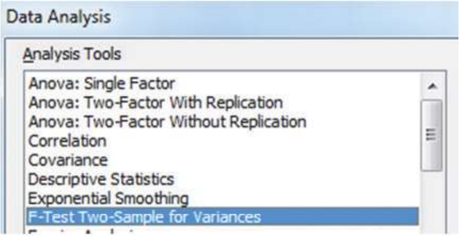
Step 4: Discuss your results and answer the following question: are the monthly mean returns under both parties equal?

Note 1: Repeat the above process using EWRETD (equal-weighted market index).

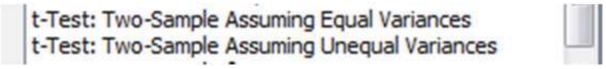
Note 2: How do we test whether two groups have the same means?

Click "Data" on the menu bar then "Data Analysis"

Step A: Test if the variances of two monthly returns are the same.

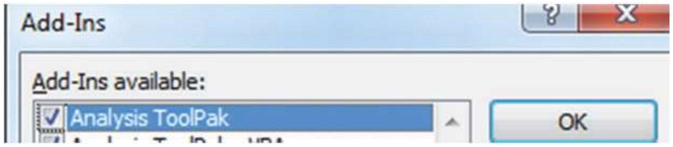


Step B: Depending on the result of Step A, we have the following choices.


t-Test: Two-Sample Assuming Equal Variances
t-Test: Two-Sample Assuming Unequal Variances

Note 3: If "Data Analysis" is not available after clicking "Data" on the menu bar

Click "File" ==> "Options" ==> "Add ins" (on the left-hand side) ==> "Go" (on the right-hand side of 'Manage [Excel Add-in]" ==> (activate Analysis ToolPak) ==>"OK"



Appendix D: Replicating 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. From that, we have two related questions: 1) how to define a winner from a loser? And 2) how do we 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 the 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 the CRSP data (PERMNO, DATE and RET)
Step 2: Estimate the 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